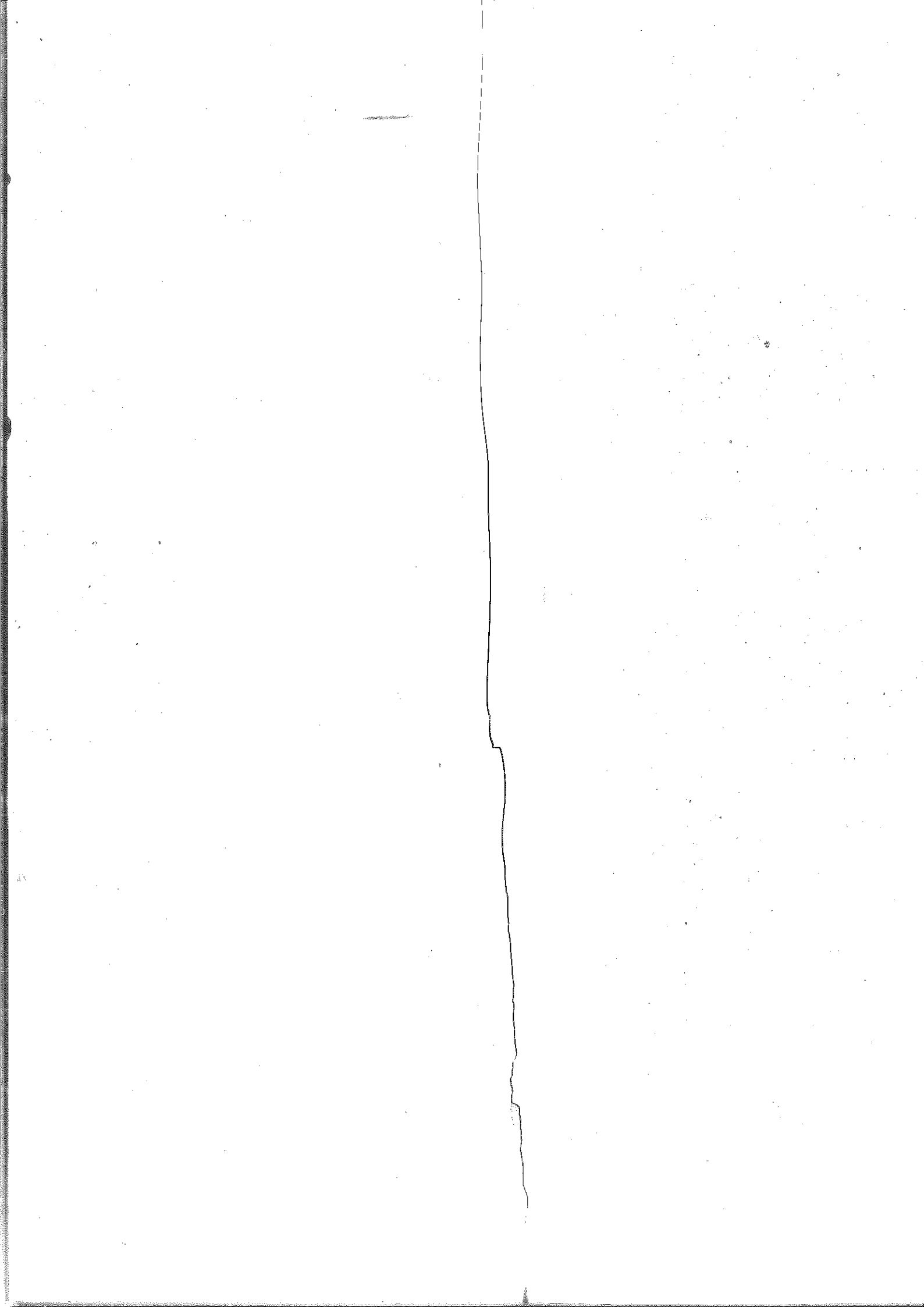


504.1

Incident Handling Step-by-Step and Computer Crime Investigation

SANS



504.1

Incident Handling Step-by-Step and Computer Crime Investigation

SANS

Copyright © 2019, Ed Skoudis, John Strand, Mike Murr. All rights reserved to Ed Skoudis, John Strand, Mike Murr, and/or SANS Institute.

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 THE SANS INSTITUTE FOR THE COURSEWARE. YOU AGREE THAT THIS AGREEMENT IS ENFORCEABLE LIKE ANY WRITTEN NEGOTIATED AGREEMENT SIGNED BY YOU.

With the CLA, the 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 the SANS Institute to the 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 The 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 THE SANS INSTITUTE, AND THAT THE 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 the 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 the 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 the 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.

SANS

Incident Handling Step-by-Step and Computer Crime Investigation

© 2019 Ed Skoudis, John Strand, Mike Murr | All Rights Reserved | Version E01_01

Hello, and welcome to SANS Security 504, *Hacker Tools, Techniques, Exploits, and Incident Handling*. We will spend the next several days discussing how attackers break into systems and, more importantly, how you can prevent, detect, and respond to such activities. This material is designed to help prepare you for the GIAC GCIH certification.

Our initial focus is on incident handling, as we discuss time-tested procedures for responding to computer attacks. The incident handling approach you will learn was originally developed by the United States Department of Energy and then adopted by the US Navy. Since then, this process has been further developed and refined by hundreds of incident handlers who have worked for over a decade to improve the state of practice. These efforts have centered on making the approach more generalized to support corporations, government agencies, educational institutions, and other organizations. The focus of this class is to prepare you to handle an incident.

SANS is serious about this training and certification, and we have invested a lot of time and effort into building a class that will prepare you to handle just about anything! You are going to have to work to win the certification, but you will know that it truly means something once you've achieved it. Incident handling is not a standalone skill; it builds on your system administration and network defense training. When you are under fire, you will appreciate having these skills at the ready!

Table of Contents

Page

Roadmap and Overview	10
Incident Handling Process	17
Preparation	19
Identification	42
- Cheat Sheets	57
- LAB 1.1: Windows Cheat Sheet	76
Containment	85
Eradication	103
Recovery	109
Lessons Learned	114
Enterprise-Wide IR	118
- LAB 1.2: Enterprise-Wide Identification and Analysis	124















This table of contents can be used for future reference.

Table of Contents

Page

Espionage	134
Unauthorized Use	140
Insider Threats	148
Legal Issues and Cybercrime Laws	154
- LAB 1.3: IR Tabletop	156
Appendix A: Intro to VMware and Linux Workshop	159

We wrap up today with a hands-on exercise.

<p>SEC560 Network Penetration Testing & Ethical Hacking GPEN</p> 	<p>SANS PENETRATION TESTING</p> <p>For more information: https://pen-testing.sans.org/ @SANSPenTest</p>  <p>SEC504 Hacker Tools, Techniques, Exploits & Incident Handling GCIH</p>  <p>SEC550 Active Defense, Offensive Countermeasures & Cyber Deception</p>	<p>SEC542 Web App Penetration Testing & Ethical Hacking GWAPT</p> 
<p>SEC660 Advanced Penetration Testing & Ethical Hacking GXPN</p> 		<p>SEC642 Advanced Web App Penetration Testing & Ethical Hacking</p> 
<p>SEC760 Advanced Exploit Development for Penetration Testers</p> 		<p>SEC575 Mobile Device Security & Ethical Hacking GMOB</p> 
<p>SEC561 Immersive Hands-On Hacking Techniques</p> 		<p>SEC617 Wireless Ethical Hacking, Penetration Testing, & Defenses GAWN</p> 
<p>SEC562 CyberCity Hands-on Kinetic Cyber Range Exercise</p> 		<p>NEW SEC567 2-Day Course Social Engineering for Penetration Testers</p> 
<p>SEC573 Automating Information Security with Python NEW GPYC</p> 		<p>SEC580 2-Day Course Metasploit Kung Fu for Enterprise Pen Testing</p> 

The SANS SEC504 course covers a variety of attacks and associated defenses. It explains how you can apply incident handling procedures to address each step of an attack. It is built around the philosophy that offense must inform defense. That is, to be a solid defender, you need to understand the attacks your systems and networks will face every day. Along with that notion is the concept that to be a good attacker (such as a penetration tester or red teamer), you need to know the defenses. There are two important reasons that professional attackers need to understand defenses. First, penetration testers need to be able to make recommendations about what kinds of defenses should be in place in their reports and recommendations. Furthermore, penetration testers need to understand defenses because they need to consider ways to thwart or bypass them.

For those reasons, this SANS SEC504 course is in a crucial location in the SANS Penetration Testing curriculum and the SANS Digital Forensics and Incident Response (DFIR) curriculum. In the slide, you can see its location in the Penetration Testing curriculum, covering insights into a variety of attacks (and their associated defenses) and providing security professionals detailed foundations and capabilities for understanding, analyzing, and launching attacks and applying practical defenses.

The slide features a central image of two baseball players in white uniforms, one swinging a bat. The background is a grid of text including 'OPERATING SYSTEMS', 'INCIDENT RESPONSE', 'DIGITAL FORENSICS', 'IN-DEPTH', 'THREAT HUNTING', and 'ANALYSIS'. At the top center, the text 'SANS DFIR' is prominently displayed, with 'DIGITAL FORENSICS' and 'INCIDENT RESPONSE' below it.

FOR500 Windows Forensics
GCFE

FOR518 Mac and iOS Forensic Analysis and Incident Response

FOR526 Advanced Memory Forensics & Threat Detection

FOR585 Smartphone Forensic Analysis In-Depth
GASF

FOR508 Advanced Incident Response and Threat Hunting
GCFA

FOR572 Advanced Network Forensics: Threat Hunting, Analysis, and Incident Response
GNFA

FOR578 Cyber Threat Intelligence
GCTI

FOR610 REM: Malware Analysis
GREM

SEC504 Hacker Tools, Techniques, Exploits, and Incident Handling
GCIH

At the bottom, there are social media and contact links: @sansforensics, sansforensics, dfir.to/DFIRCast, dfir.to/gplus-sansforensics, and dfir.to/MAIL-LIST.

As described on the previous slide, defenders need a solid understanding of attacks. When offense informs defense, security personnel, such as digital forensics experts, can anticipate an attacker's moves and analyze or counter them much more effectively. For that reason, this SEC504 course is also in a critical position in the SANS DFIR curriculum, as you can see in the slide.

SEC504 Course Roadmap

- Incident Handling 504.1
- Applied Incident Handling
- Attack Trends
- Step 1: Reconnaissance 504.2
- Step 2: Scanning
- Step 3: Exploitation 504.3
 - Gaining Access
 - Web App Attacks
 - Denial of Service
- Step 4: Keeping Access 504.4
- Step 5: Covering Tracks 504.5
- Conclusions and CtF 504.6

This course is made up of six parts. In the first part, 504.1, we discuss incident handling techniques, focusing on a well-established process for handling incidents in enterprises. We discuss the overall methodology, picking up useful processes and technical tips along the way. In addition, we go over computer crime topics and the legal system, discussing how they impact incident handlers.

After establishing a firm base of incident handling concepts, the course shifts into a discussion of how attackers exploit target systems and networks, from a step-by-step perspective. We go over reconnaissance and scanning in 504.2 to see how attackers get to know a target environment better. Next come two of the biggest components of the course, 504.3 and 504.4, in which we discuss how attackers gain access to target machines. This section is split over two course books because attackers have so many methods to infiltrate targets.

Section 504.5 looks at the next phases of many attacks after attackers gain access: maintaining access and covering their tracks. We address topics such as backdoors, rootkits, and log editing.

Our final component of the course is designed to help get attendees into the mindset of attackers. This gives you an idea of how an attacker views a target environment and the steps he or she would apply. This last topic, covered in 504.6, is our Capture the Flag (CtF) event, in which you get to apply all of the attack phases into a competition to hammer home lessons from throughout the course.

Lab Exercises

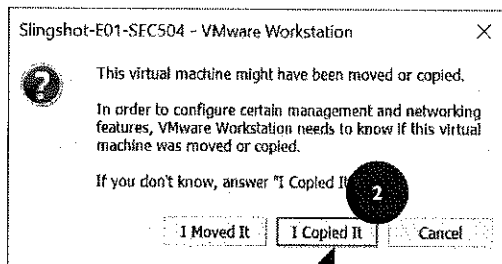
Throughout this course you will complete many different lab exercises to reinforce the course material and to build hands-on skills that you can apply immediately when you get back to the office.

- You will use two virtual machines for lab exercises, supplied on your SEC504 USB drive: Slingshot Linux and Windows 10
- Please take a moment to start copying the two VMs now
- After copying, extract compressed files using 7-zip
 - Install 7-zip (7z1900-x64.exe) if needed

In this course you will work on many lab exercises to reinforce the concepts we examine.

Boot and Login: Windows 10 and Slingshot Linux

- 1 Start the Windows 10 VM by double-clicking the VMX file



When prompted, select *I Copied It*

- 3 Log in with the username sec504 and the password sec504

- 4 Repeat this procedure with the Slingshot Linux VM

These VMs have been optimized for the lab exercises, please don't use them in a production environment!

After unzipping the VMs, double-click on the VMX file for the Windows 10 VM (1). When you first start the VMs, VMware will ask you if you copied or moved the VM. Please select *I Copied It* (2). When the Windows 10 VM has finished booting, log in with the username sec504 and the password sec504 (3).

After booting and logging in to the Windows 10 VM, repeat this procedure with the Slingshot Linux VM as well (4).

The VMs we use in the class are optimized for the lab exercises. Please do not use these VMs for any production-related activities. They should be used only for this class. They are not regularly updated or secured.

If you need to change your keyboard layout from the US default on Slingshot Linux, please select the following: Activities | Type keyboard | Then choose Region & Language.

Online Lab Access

All labs are completed on your local system using VMware and the supplied virtual machines for books 1 through 5.

If you are accessing the course online, you will complete some additional steps to participate in the book 6 Capture the Flag event.

Visit the *My Labs* link in your SANS Portal page at www.sans.org for setup directions.

All of the lab exercises for books 1 through 5 are completed locally using the supplied Slingshot Linux and Windows 10 virtual machines. These exercises are completed the same way if you are in the classroom or learning online.

For the book 6 Capture the Flag (CtF) event you will complete some additional steps to access the online lab environment. Visit the *My Labs* link in your SANS Portal page at www.sans.org for setup directions.

Course Roadmap

Incident Handling

- **Incident Handling**
- Applied Incident Handling
- Attack Trends
- Step 1: Reconnaissance
- Step 2: Scanning
- Step 3: Exploitation
 - Gaining Access
 - Web App Attacks
 - Denial of Service
- Step 4: Keeping Access
- Step 5: Covering Tracks
- Conclusions

1. **Overview**
2. Incident Handling Process
3. Preparation
4. Identification
 - Lab 1.1: Windows Cheat Sheet
5. Containment
6. Eradication
7. Recovery
8. Lessons Learned
9. Enterprise-Wide IR
 - Lab 1.2: Enterprise Ident. and Analysis

The first day's materials are broken into the following phases:

- **Overview:** This is an introduction to the material.
- **Detailed Incident Handling:** This is the heart of the 504.1 materials. It describes a six-step process for handling incidents in depth.
- **Applied Incident Handling:** This section presents tips and tricks for various scenarios. Each suggestion has served experienced incident handlers well.

We also include some additional material in the appendices that you'll need to know going forward for this course. If you are already familiar with VMware and Linux, you are ready to go with these appendices. If you are new to VMware and/or Linux, these appendices are essential, and we strongly recommend that you read them. Better yet, pop open a laptop and experiment with the items in Appendix A. Similarly, if you know VMware and Linux but haven't used them in a while, you might want to read the appendix as a refresher. Trust us—you need to know this material for the rest of this class!

If you have seen our *Incident Handling Step-by-Step* book and are wondering how the book and course relate, note that the book is the outline for the course. Additional material is covered here.

Without further ado, let's dive into the material!

Handwritten notes:
Lab 1.1: Windows Cheat Sheet
Lab 1.2: Enterprise Ident. and Analysis

- Incident handling is an action plan for dealing with the misuse of computer systems and networks, such as
 - Intrusions
 - Malicious code infection
 - Cyber theft
 - Denial of service
 - Other security-related events
- Keep written procedures and policies in place so you know what to do when an incident occurs

Incident handling is the action or plan for dealing with intrusions, cyber theft, denial-of-service, and other computer security-related events. Your incident handling plan should include hooks to your general disaster recovery and business continuity plans that deal with fire, floods, and other disastrous events. The scope of incident handling is greater than just intrusions; it covers insider crime and intentional and unintentional events that cause a loss of availability. Furthermore, intellectual property is becoming more important as we move into an information age. Types of intellectual property include brands, proprietary information, trade secrets, patents, copyrights, and trademarks.

The other key point of the definition is the notion of action. Sitting there watching is not incident handling. Identifying an incident is important, but you must act on that information to secure your systems in a timely manner. The best way to act on an incident and minimize your chance of a mistake is by having proper procedures in place. Well-documented procedures ensure that you know what to do when an incident occurs and minimize the chances that you will forget something.

Your incident handling plans and policies must comply with the applicable laws of your country. We discuss some of these legal aspects in more detail at the end of the day, after we describe the incident handling process itself.

- The term incident refers to an adverse event in an information system and/or network . . .
- . . . or the threat of the occurrence of such an event
- Focus is on detecting deviations from the normal state of the network and systems
- Examples of incidents include
 - Unauthorized use of another user's account
 - Unauthorized use of system privileges
 - Execution of malicious code that destroys data
- Incident implies harm or the attempt to harm

This slide and the next one are for the purpose of defining what we mean when we use a word like *incident* or *event*. Incident, as we use it, refers to actions that result in harm or the significant threat of harm to your computer systems or data. Looking for incidents involves finding deviations from the normal state of the network and systems. There are several important points for an incident handler that flow from this definition. First, because we are dealing with harm or potential harm, our task is to limit the damage. We want to be careful to choose courses of action that do not cause further harm.

Second, your organization may well have a right to redress. There are criminal and civil law remedies associated with computer incidents. In either case, the incident handler should proceed in a manner that does not preclude use of the evidence gathered in a court setting. A handler does not know in advance whether a given case will go to court. Although only a small fraction of most cases end up in court, you need to treat all of them from the outset as though they may go to court. Don't worry; that's not an enormous burden. It just means doing your job thoroughly and documenting your actions carefully.

- An *event* is any observable occurrence in a system and/or network
- Examples of events include
 - The system boot sequence
 - A system crash (could be normal behavior for that system)
 - Packet flooding within a network (could be bursty, legit traffic)
- These observable events provide the bulk of your organization's case if the perpetrator of an incident is caught and prosecuted
 - Must be recorded in notebooks and logs
 - Recording the same event in multiple places helps improve evidence— that's corroborating evidence

Events are observable, measurable occurrences in our computer systems. An *event* is something that happens that someone either directly experiences or that you can show actually occurred. An event is something that you see flash on the screen or that you hear. It can also be something that you know occurred because it was collected in a log or audit file.

At <http://www.sans.org/score/incidentforms>, you can find forms that help you document the information that should be documented; these forms help alert you to the things you should look for. The forms' copyright allows you to make all the copies you want.

If there is any chance of the incident ending in a court case, having corroborating information is better than a single source claiming that the event happened. For instance, if two people see a message flash on a screen, this fact will likely have more validity in court than if just one person saw it. Further, attackers sometimes use tools to alter or delete their traces in log files. If you can produce two independent sources for the information, your evidence has more validity. This is one reason we push intrusion analysts to become familiar with a large number of log formats. Let's look at an example on the next slide.

Corroborating Evidence: Microsoft IIS Attack?

Overview

```
[**] IIS vti_inf access attempt [**]  
06/25-05:36:17.833982 63.209.91.33:4791 -> 10.0.0.13:80  
TCP TTL:116 TOS:0x0 ID:6075 DF  
***PA* Seq: 0x1CB6779 Ack: 0xB58F0491 Win: 0x217C
```

Snort
Output

- Which corresponds directly with

```
[Wed Jun 25 05:36:13 2017] [error] [client 63.209.91.33]  
File does not exist: /usr/local/apache/htdocs/_vti_inf.html  
[Wed Jun 25 05:36:14 2017] [error] [client 63.209.91.33]  
File does not exist:  
/usr/local/apache/htdocs/_vti_bin/shtml.exe/_vti_rpc
```

Log
Output

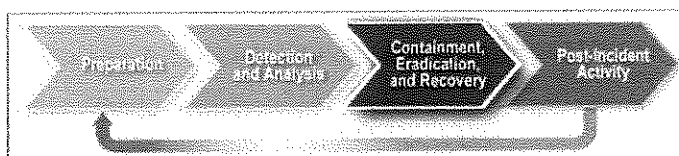
- Incident or event?
 - We must look at environment and context

SEC504 | Hacker Tools, Techniques, Exploits, and Incident Handling 14

On this slide is a bit of corroborating evidence that can help determine what is happening on our systems and even bolster a case. The main point is that two different systems have captured the same event of interest. The top block comes from a Snort intrusion detection system. It has a rule that causes it to alert if it sees the signature for a particular attack; in this case, it is an attack on the Windows web server IIS. This particular attack is against a weakness in a default script on Windows 2000 IIS servers named vti_inf. This is because scanning tools look for old and out-of-date software.

The bottom block comes from a UNIX web server running Apache software. So there isn't a lot of risk of harm here; a Microsoft IIS attack is unlikely to succeed against a UNIX system running Apache. Some people would classify this as an incident because the attacker probably did have malicious intent. Others would say that because no harm can be done, it should not be considered an incident. The point, however, is that the intrusion detection system and the web server are completely separate systems and they show the same event. If this went to court, having both logs of the event will make for stronger evidence. Also, when you have multiple sources of information, there is a good chance that you will be able to get data from one that might not be available in another. This is why the intrusion analyst and the incident handler should work hard to learn the types of log files available to them and develop the skills needed to read the logs.

- Incident handling is similar to first aid
 - The caregiver is under pressure, and mistakes can be costly
 - A simple, well-understood, documented approach is best
- Keep the six stages in mind: Preparation, Identification, Containment, Eradication, Recovery, and Lessons Learned
- Use predesigned forms and ask for help
- Additional materials are available in NIST's *Computer Security Incident Handling Guide*



Law enforcement agents tell story after story of the well-meaning system administrator who ruined the evidence—usually just a couple minutes after the incident. You do need to act, but take time to think.

This story has a crucial point. No one can run so fast that he can outrun a computer with a 3-GHz multi-core processor attached to a Gigabit Ethernet. More importantly, when one is working as root, administrator, or supervisor, many operations do not have an “undo” capability. Several times during this part of the talk, we will draw the analogy between incident handling and first aid. It is a solid analogy; in some ways, first aid is a form of incident handling.

To help you stick to the six-step process, use the forms at the www.sans.org website. They provide a template for useful information you need to capture during an incident. The free forms at this site include Incident Contact List, Identification Checklist, Survey, Containment Checklist, Eradication Checklist, and a Communications Log.

In addition, for more valuable materials, NIST has developed a *Computer Security Incident Handling Guide* (<https://csrc.nist.gov/publications/detail/sp/800-61/rev-2/final/>) that covers the same concepts we do here. It's a solid read and nicely complements this material. You can get it at no charge from the URL listed here. The graphic on the slide is excerpted from the NIST document and summarizes a flow for incident handling fully compatible with the process we cover in this course.

- If your corporate policy will allow it, share what you have learned with other incident handlers and incident response teams
 - Attacks against computers are happening everywhere, all the time
 - The bad guys share information; if we incident handlers do not share with each other, they'll stay a step ahead
 - Coordinating your efforts with those on other teams is a critical facet of incident response
 - Do as they told you to do in elementary school: share
 - The Internet Storm Center is a wonderful point of communication with a handler on duty every day
 - Check out the various *cons*, such as DEF CON, Black Hat, ShmooCon, and Wild West Hackin' Fest

The attacker community cooperates with one another (albeit sometimes in an antisocial manner). They share hacked accounts, exploits, and tricks of the trade.

We often don't share information in the security community. The fact that we may have come under attack seems to be a secret. This will not come as a big surprise, but virtually everyone connected to the internet comes under attack. Eventually, your organization is bound to take a hit. You can learn from that and you can share what you learn. By doing so, others can learn. If your attackers share and you don't, your organization is outnumbered big time!

So how can you share attack and incident information? You can post something to BUGTRAQ mailing list at www.securityfocus.com, or submit information to the handlers' list at the Internet Storm Center (isc.sans.edu). The handlers' list always has an experienced handler on duty, waiting for reports to come in. Each day, the handlers' diary is updated with the latest information about computer attacks. You should check it out! The ISC also displays statistics from the DShield sensor network, which has over 40,000 sensors distributed around the planet gathering information about scans and attacks against various ports. Their world map view shows the countries associated with the source IP addresses of the scans. Additionally, the ISC shows the top 10 rising target ports used in these scans.

There is also a large number of security/hacker gatherings called *cons* where great (and sometimes scary) information is shared. Following are some of our favorites:

<https://www.defcon.org/>

<https://www.blackhat.com/>

<http://www.shmoocon.org/>

<https://www.wildwesthackinfest.com/>

Course Roadmap

- **Incident Handling**
- Applied Incident Handling
- Attack Trends
- Step 1: Reconnaissance
- Step 2: Scanning
- Step 3: Exploitation
 - Gaining Access
 - Web App Attacks
 - Denial of Service
- Step 4: Keeping Access
- Step 5: Covering Tracks
- Conclusions

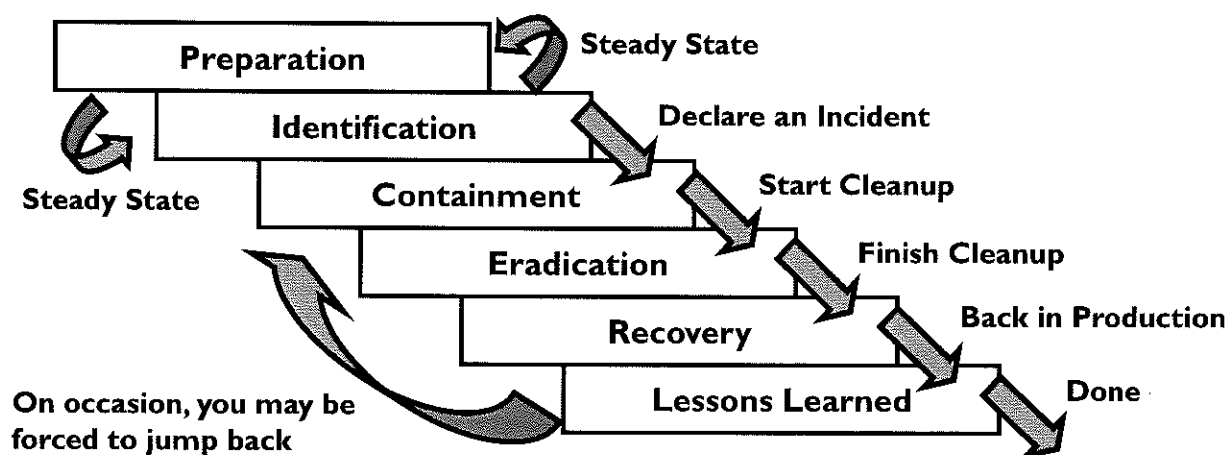
Incident Handling

1. Overview
2. **Incident Handling Process**
3. Preparation
4. Identification
 - Lab 1.1: Windows Cheat Sheet
5. Containment
6. Eradication
7. Recovery
8. Lessons Learned
9. Enterprise-Wide IR
 - Lab 1.2: Enterprise Ident. and Analysis

Here is our outline. We now move to the detailed incident handling process, the core of this first-day session, and the basis for all of our discussions for the rest of this class.

Six Primary Phases

Incident Handling Process



SEC504 | Hacker Tools, Techniques, Exploits, and Incident Handling 18

The six steps in incident handling are Preparation, Identification, Containment, Eradication, Recovery, and Lessons Learned. The steps serve as a compass or roadmap for the handler; the process is a way for the handler to keep in mind what he is trying to do and the things he needs to do next.

The steady-state, day-to-day practices of most incident handlers are the first two steps: Preparation and Identification. We spend a lot of our time getting ready to fight the next battle and looking for events that could be signs of trouble.

After we identify an incident (that is, events that indicate harm or the attempt to do harm), we move into Containment. Then, the general flow is down the page. You move from Containment to Eradication to Recovery to Lessons Learned. Don't skip steps! Also, we caution you: Try to complete an entire given step in the Containment and later phases before moving to the next phase for a single incident. In other words, for one incident, don't contain it partially on a few systems, and then move to Eradication on those machines while Containment on other systems begins. Do Containment first, and then move to Eradication, and so on. You will likely get organizational pushback on such an approach, but it is the best way to successfully handle incidents.

Also, although the general flow of this process is down the page, sometimes you have to jump back up when circumstances change. You might be in the midst of the Recovery phase when your attacker or malicious code sneaks back in. You've got to be flexible enough to jump back and redo the Containment phase, then Eradication, and then Recovery, for example.

Course Roadmap

- **Incident Handling**
- Applied Incident Handling
- Attack Trends
- Step 1: Reconnaissance
- Step 2: Scanning
- Step 3: Exploitation
 - Gaining Access
 - Web App Attacks
 - Denial of Service
- Step 4: Keeping Access
- Step 5: Covering Tracks
- Conclusions

Incident Handling

1. Overview
2. Incident Handling Process
3. **Preparation**
4. Identification
 - Lab 1.1: Windows Cheat Sheet
5. Containment
6. Eradication
7. Recovery
8. Lessons Learned
9. Enterprise-Wide IR
 - Lab 1.2: Enterprise Ident. and Analysis

Chance truly favors the prepared mind, and this is especially true with incident handling. As mentioned before, the middle of an incident is not a good time to ponder your incident response process, wonder if there is a command that will enable you to audit an operating system, or figure out how to create a trustworthy forensics image. Therefore, it is imperative to prepare and ensure you have the skills and resources that you need ready to go at a moment's notice.

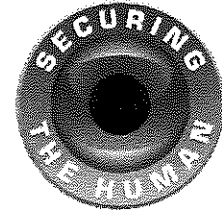
The goal of Preparation is to get the team ready to handle incidents

- People
- Policy
- Data
- Software/Hardware
- Communications
- Supplies
- Transportation
- Space
- Power and Environmental Controls
- Documentation

40

The goal of Preparation is to get the team ready to handle incidents. This slide serves as an overview of the elements needed to prepare the team for an incident; these are actually the fundamentals of contingency planning, and it is advisable to have these basics covered. As we move through the Preparation section, we will discuss these items further.

- One of the most overlooked aspects of our security posture
- Also, the most easily attacked
 - Via targeted email (spear phishing)
 - Via calls (social engineering)
- Reoccurring training can be a big help
 - Annual training tends to be ineffective
 - Constant reinforcement
 - SANS Securing the Human
- You can also regularly test your users with social engineering calls and phishing tests
 - Caller ID spoofing is a good test to employ
 - Phishing frameworks, such as Phishme



In information security, we tend to focus on the easy things—things like IDS, IPS, and AV. These are all technical and can be relatively easily implemented and evaluated. However, in many cases, these technologies are not how attackers target our organizations. Instead, many attackers target what is generally regarded as the easiest attack point: your people.

When attacking an organization, attackers can target users in a number of different ways. The most commonly used ways are via a phone call and through a malicious email.

The best way to prepare for these types of attacks is through constant training and assessment. Many organizations undergo quarterly testing of their technology; the same principle can be applied to people. Once a quarter, either call users via a social engineering (SE) campaign and/or utilize a spear phishing framework to test your user population's susceptibility to clicking malicious links. Projects and services like the sptoolkit and Phishme are excellent ways to create phishing campaigns and track the results.

For more information on training, check out the SANS Securing the Human at

<https://www.sans.org/security-awareness-training/>

Local SE + Phishme

- Establish policy and warning banners
 - Warning banners limit the presumption of privacy
- Warning banner should advise the user that:
 - Access to the system is limited to company-authorized activity
 - Any attempt at or unauthorized access, use, or modification is prohibited
 - The use of the system may be monitored and recorded **Crucial**
 - If the monitoring reveals possible evidence of criminal activity, the company can provide the records to law enforcement
- Have legal team review this banner, approving it in writing
- Be careful of local privacy laws, especially in Europe
 - European Data Privacy Directives may impact that crucial line

Don't go flying through this slide because this is such a familiar word.

Warning banners are very important to an incident handler. They make a major difference in the amount of trouble you have to go through to collect and use evidence. Everyone knows you should have them, but if your organization is lax on the implementation, start squawking! This is a battle worth fighting, but be certain to fight in a wise manner. The banner is one tool that can be used to explicitly define your organization's policy on the presumption of privacy. In a world of shades of gray, this is an issue we want to nail down; a handler must know his organization's policy about privacy.

If at all possible, an organization should retain the authority to monitor its networks and systems. That's why that fourth bullet is so important: "The use of the system may be monitored and recorded."

Have your legal team review the language of the warning banner to make sure it will support your monitoring activities and back you up in a court case. Have legal approve it in writing.

This language works well in the United States and several other countries; however, be careful in Europe. Various countries' interpretations of the European Data Privacy Directives may forbid you from monitoring your own data. Your best bet is to get local legal counsel to advise you on such matters.

- Establish an organizational approach to incident handling
- Decide generally how you will handle the "big issues" up front
 - Maintain secrecy or notify law enforcement
 - Most organizations maintain secrecy until they must notify law enforcement
 - That's not always the best policy, though
 - Contain and clear or watch and learn
 - Most organizations have a default pre-authorization to contain, but may handle it differently depending on the particulars of the case
- Get management buy-in and sign-off for your default practices
 - Document any purposeful deviations from your standard practice when you opt to do so

One thing you want to avoid is having an incident happen and finding yourself in a debate about whether to contain the incident and clean up or to watch the attackers and try to gather more evidence. Likewise, during the time an incident is occurring is a bad time to decide whether your policy is to involve law enforcement or maintain secrecy. The time to make these (career-affecting) decisions is before the incident, keeping senior management and your legal staff apprised.

If you want to consider watch and learn, you should probably spend some time reading about the Honeynet Project (www.honeynet.org). They probably have the most experience with this of any group on the internet.

- Requirement to report varies by jurisdiction
 - Threat to public health or safety
 - Substantial impact on third party
 - Legal requirement based on industry (e.g. FDIC, OCC, etc.)
- Many jurisdictions have breach notification laws
 - Usually focus on customer notification of compromised data (e.g. PII/PHI)
 - Scope of who must report can be surprisingly broad
- Optional reasons to notify law enforcement
 - To benefit from criminal discovery process
 - To be a good corporate citizen

Whether or not you are required by law to contact a law enforcement agency when an incident occurs depends on the jurisdiction. Unfortunately too many companies are too timid to call law enforcement. If we want to get tough with attackers, we need to be more willing to get law enforcement actively involved.

Some of the reasons you might be required to report are: a threat to public health or safety, substantial impact to a third party, or a legal requirement for your industry. Also, note that you may need to notify the public about an incident involving Personally Identifiable Information (PII) or Personal Healthcare Information (PHI). The scope of who must report under some of these laws (e.g. California's SB1386) can be surprisingly broad.

Another reason to notify law enforcement is a selfish one: to benefit from criminal discovery in a court case. The final reason involves just helping the community by making sure the attackers pay for their crimes.

- Several points to consider when interfacing with law enforcement
 - There will be two investigations (yours and theirs) with differing goals
 - They usually *don't* go to the media without victim consent
 - But they are usually under no strict legal obligation not to inform the media
 - Might ask to watch attackers, in order to gather more evidence
 - May ask for equipment if it is evidence
 - Though you usually don't have to provide it, especially if it hurts business operations
 - Not providing *any* evidence (e.g. not even copies) may hinder the investigation
 - Will need access to personnel with technical details
- Utilize the SANS SCORE: Law Enforcement FAQ
- Always consult with legal counsel

There are several things worth considering if you decide to contact law enforcement. First is that there will be two cases, *yours* and *theirs*. Just because law enforcement starts a criminal investigation does not mean you must stop yours. However the goals of a criminal investigation may differ from yours. Law enforcement investigations are usually in support of criminal prosecution, where as a business's investigation is often focused on getting business back up and running.

A common misconception about working with law enforcement is that they leak details of an incident to the press. When you notify law enforcement of an incident, they usually do *not* contact the press or media. Especially if it would hinder the investigation, or cause the victim to suffer further damage. Remember, law enforcement agents do not want to make the victim feel victimized again. With this in mind, in most jurisdictions law enforcement is not forbidden from contacting the media.

Law enforcement agents may *ask* you to allow the attackers to continue, so they can observe and gather more evidence. However short of a court order you usually don't have to comply. Similarly, they may *ask* for the equipment (computers, hard disks, etc.) that have evidence on them. Again, without a court order you usually don't have to comply, especially if it would negatively impact your business. Keep in mind that not providing *any* form of evidence (e.g. not even providing digital copies) can significantly hinder an investigation.

When law enforcement first contacts a victim, they have no pre-existing knowledge of the victim networks, what evidence exists, etc. So what law enforcement *will* need is access to people who have technical knowledge and details of the incident, as well as access to people who can provide (copies of) the evidence. The time that victim personnel spend working with law enforcement is time they necessarily are not spending on their job-assigned tasks. Of course, that's just part of the nature of reporting.

For more see the SANS Interfacing with Law Enforcement FAQ: <https://www.sans.org/score/law-enforcement-faq>

- Establish a policy for outside "peer" notification
- Establish a policy for dealing with incidents involving remote computers belonging to
 - Business partners and joint ventures
 - Your company
 - Your employees
 - Contractors and other employees who are not full time
- For VPN usage, include a warning banner saying that all systems connecting are subject to remote search
 - Include this notice in employee awareness initiatives

The unwritten policy on incident notification in some organizations is never to tell anyone anything for any reason. However, if there is any chance of the incident spreading and people finding out you had an incident because they were affected, this policy is not ideal.

What happens if the computer is not one that you own, but it has your data on it? The classic example is the employee who takes work home and has a system compromise at home, which involves a business system or even business data stored on his own home computer. Is the employee required to notify your organization? Are you going to do a full backup of that computer? It has three years of TurboTax data on it! Are you going to erase that hard drive?

For VPN usage by your employees, include a warning banner invoked upon VPN access that says that all systems connecting through the VPN are subject to remote search by the organization. Include a notice about this search possibility in employee awareness initiatives. Even though you won't often take this course of action, it is still useful to have it as an option.

What about a consultant who visits your organization, and the consultant's laptop is detected scanning the organization's file server? We know what you want to do, but what policy supports that?

What is your organization's road warrior policy? Do you have hard drive encryption enabled? Are you prepared in advance for such circumstances?

Remain Calm and Take Notes

Preparation

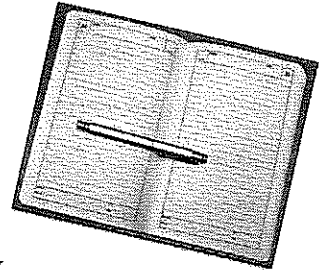
- Remain calm
 - Even a fairly mild incident tends to cause stress
 - Communication and coordination become difficult
- Do not hurry; mistakes can be very costly
- Notes, logs, and other evidence are crucial
- Handwritten notes can be a big help
 - Judges and juries resonate with them
 - The attacker cannot steal them from your machine or destroy them in a denial-of-service attack
 - They help you organize your thoughts and act as a governor on your speed
- If you are going too fast to take notes, you are going too fast!

Whenever people are under stress, communication tends to degrade; this is true even with experienced veterans. Have you ever wondered why folks in professions where communication is life-threatening and critical, such as police, firefighting, rescue, warfighters, and commercial pilots, all use a formalized language? You know what I mean: "Alpha Yankee Zulu, this is Popeye niner, I have a bogey on your six." They adopt a language that has explicit meaning so errors of interpretation are less likely. They practice speaking that language so they can do so when under stress.

A sure sign that you are in too much of a hurry is when you don't have time to take notes! This is one of the most common, least excusable errors incident handlers make. It is a sinking feeling when you get a phone call and the person on the other end is telling you that the perpetrator of an incident from six months ago has been arraigned. The court date is two months from now, and you realize you can't even come up with the date of the incident. How long should you keep your records? The best answer is to consult your organization's attorney, but in general, keep them as long as possible. We have been contacted by the FBI asking for logs of events that occurred years earlier.

Good record keeping is one of the most important skills that a handler must develop. This is also one of the more difficult things to test, and that is why your practical assignment is a detailed write-up of an incident or related vulnerability. Handwritten notes serve many purposes, including appealing to judges and juries, not being subject to electronic theft by the attacker, helping to organize the handler's thoughts, and governing your speed while handling an incident. In short, if you are going too fast to take good notes, you are just going too fast!

- Take excellent notes in a bound notebook with numbered pages
 - Your notes may become evidence in court . . . two years later
 - Record all of your actions (e.g. questions asked, commands typed, systems downed, etc.)
- Answer the Who, What, When, Where, Why, and How
 - Who and Why are often the most difficult in intrusions
- Date and timestamp each entry in your journal
 - Include date, time, and name of handler
- A small audio recorder and a still camera can be valuable
 - You may want to avoid video cameras, as they could be problematic



The handler must be certain not just to take notes, but to take good notes. The journalist's standard questions of who, what, where, when, and why are a bare minimum. Of the Ws, the hardest ones are who and why in many computer incidents. It is especially important to record all of your actions, including the questions you asked, the answers you received, the commands you typed, the systems you downed, and so on. Make sure you date and timestamp each entry in your journal, including the date, time, and handler's name for each element you write. The forms that are provided in the *Incident Handling Step-by-Step* book have most of the crucial words built in: date, time, location, operating system, and so forth. This can save a lot of time. The forms also remind notetakers of what information they should be collecting. Another advantage to forms is you are a bit less likely to doodle!

A video might contain far more information about your operation than you want to give away. A single-shot camera can certainly record the scene as you first saw it and is less dynamic than a video. I usually avoid using video cameras in my incident handling activities, favoring a still camera instead.

- Foster management support for an IR capability
 - Monthly or quarterly reports on brightly colored paper
 - Graphically illustrate an incident you faced
 - Show jump-off points used in your network
 - Collect historical support
- If it is a quiet month
 - Collect news articles on computer incidents and other related events, especially in organizations similar to yours
 - Watch the *Handler's Diary*, and *SANS NewsBites*
 - Look for similar organizations to yours that are being compromised in the news

Incident

Incident

Incident
Update

"I know hackers can be a nuisance, but they can't actually hurt anything, can they? I mean they can't harm anything serious." A senior security manager said this recently. Now we could call him dumb or short-sighted, but unless we reach him in ways he understands, he is going to be very tightfisted with resources for projects like incident handling. What will convince him? Seeing evidence of damage (especially significant harm that could affect his organization's capability to compete) done to organizations just like his is very powerful.

To deal with this issue, write a monthly or quarterly report on a single page of brightly colored paper. Call it your "Incident Report" and diligently prepare it for management to illustrate what your team has done in the previous month or quarter. If it's been a quiet quarter, list other security incidents in the press and describe how your team is taking proactive actions to deal with similar problems in your own environment.

Graphically illustrating an incident, in essence creating a cartoon, is also a very powerful technique. If a senior executive is able to "get it" and explain how an attack works to her peers, she is more likely to support your effort.

If we do not invest in communication, the only time an incident handling capability is appreciated is when there is an incident (that scares management) and it is handled well. Getting and keeping management and system administration support is a little like swimming upstream.

It is also helpful to look in the news for other similar organizations being compromised. There are great reddit.com lists like [r/netsec](#) and [r/pwned](#) for these news stories.

- Identify qualified people to join the team
- Choose local, centralized, or combination teams
- A multidisciplinary team is best
 - Security (both physical security and information security)
 - Network operations and management
 - Legal counsel, Human Resources, public affairs/public relations
 - Disaster recovery / business continuity planning
 - Union representation (if you are a union shop)
- Obviously, you won't get a full headcount for most of these
 - But at least make sure there is someone assigned to you, with a fraction (~10% or more) devoted to the incident handling cause

One of the challenging problems in building a world-class team is team members who are not hand-selected. Once you get some momentum, everybody wants to get in on the act. When someone wants to forcibly join, or a manager wants to force you to pick someone up on your team, this potentially can be a problem. There is more to a good handler than even desire and technical skill, although those skills are needed.

We recommend an inclusive approach: Anyone willing to study on his own time and able to qualify should have a fair shake at making the team. One solution that seems to work well is a core team and then a larger team that includes your legal and public affairs subject matter experts and security officers or all system administrators. If your organization has one or more unions, then be certain to analyze the contracts with the union before an incident occurs, and you will probably want union representation on the response team.

The demographics of your organization help determine whether a centralized team is at all reasonable. As a general rule, if you have to get on a plane to handle an incident, the structure you have chosen is not going to work well in practice.

Make sure that your team includes the following disciplines:

- Security (both computer and physical security!). You need skill sets of team members to include incident handling, forensics analysis, and malware analysis. Those skills may be embodied in a single person or divided among multiple people.
- Operations (system administration)
- Network management
- Legal counsel
- Human Resources
- Public affairs / public relations
- Disaster recovery / business continuity planning
- Union representation (if you are a union shop)

- Prepare system build checklists
 - Have most experienced system admins prepare a brief procedure for backing up and rebuilding systems under their control
 - One brief build document per system type
 - For example, standard Windows desktop, standard SAMBA file server, standard IIS web server, standard Apache server, etc.
 - They may have these already . . . if so, get a copy and even an image
 - If not, help make it happen
- Establish visibility and a compensation plan for the team
 - Work times and loads vary widely
 - Comp time is important; make sure management understands the need

Our computing environments are complex; no one knows every variant of UNIX and so forth. Although we are trying to make sure you have a solid grounding in the basics of handling systems, memory fades over time. It's useful to have the operations team in an organization prepare brief system build checklists that describe the standard build of each type of system in the environment, in 5 or 20 pages per system type. Not only will system administrators refer to these documents in their day-to-day work, but incident handlers will also find these documents to be immensely useful in understanding the environment better. If these documents already exist, get a copy for the incident handling team. If they do not exist, have the incident handling team work with system admins to help create them during preparation time.

In addition, you may want to get a virtual machine image of your standard builds in the environment so you can analyze them or at least compare discovered evidence and configuration information against them during an incident. These virtual images can help an experienced handler during analysis.

A large organization with over 10,000 computers is going to rack up some incidents. This can cause the incident handlers to burn out. It is kind of interesting; they tend to burn out just as they become good at their jobs. After training and seasoning, they do a great job on a couple of hot problems and the next thing you know, they are suffering from various stress effects. The solution seems to be a set of things, including rewards, compensation, and time off. This might run afoul of your organizational culture, but consider this: When do incidents occur? They often occur on Friday afternoons at 3:30 p.m. or later. Do the handlers and administrators go home and wait until Monday to start on the cleanup? No, in almost every case, they stay until the job is done. So, we need to reward these people and let them get some rest.

*While doing
a while ago
I don't know*

- Define incident handling team organization
 - On-site/on-location techie handlers
 - Often directly report to a business unit, with a dotted line to incident handling or even the security team
 - Command post with communications and management organization support
- Establish a response time baseline
 - Response time may vary (e.g. 15 to 90 minutes) depending on sensitivity
 - Have a skilled person respond within N minutes at all major facilities
 - May not report to incident handling team, but instead may be part of the business unit

This is one of the most important ideas that came out of the incident handling research process. In a fairly large incident, technically oriented "action" team members go to the site of the incident, collect data, evaluate the situation, and make decisions (or make recommendations, depending on their site's philosophy). However, this isn't half of the work required to handle a serious incident. A great deal of communications work needs to be done through coordination with other groups, and there are times when the decisions that need to be made must involve upper management.

The observation was made that police departments set up command posts when they need to handle large incidents, and this architecture works well for them. Disaster recovery specialists do the same thing while combating major fires, floods, or tornadoes. The command post needs to be identified in advance, and it should have plenty of communication methods, such as phones, faxes, networks, cell phones, batteries for cell phones, and staff who can collect the information coming in from the field and coordinate that information appropriately. This is the command post team.

In addition to the command post team, you'll also have local techie handlers. To help structure this group, establish a firm time frame (some firm time between 15 and 90 minutes) for your response team to have feet on the ground at the incident. Make sure you are able to have a technically savvy person respond within that time frame at all major facilities. These people may not report to the incident handling team but instead may be part of the business unit. Still, make sure that it's part of their job description to support your team.

- Develop an Emergency Communications Plan
 - Create a call list and establish methods of informing people quickly
 - Get a conference bridge number that can be set up with instant notice
 - Print (and laminate if you can) a credit card-sized list of incident response team contact information
 - Include the name and contact information for each member, and include the conference bridge number
 - Pass them out to everyone on the team
 - Test your call list and tree to make sure it works
 - Try "normal" times and "unusual" times
 - Use these tests to go through an incident scenario

When discussing the Emergency Action plan, we discussed how communications degrade when you are under stress; this is true on the organizational and individual level. So knowing in advance that in a serious incident, you may need fallback communications, it is possible to plan for this. Telephone communications include voice, fax, and voicemail. Organizations should have call lists to reach key people and "soccer mom style" call trees in case a large number of people need to be reached.

A shared voice mailbox for the core team allows each member access and is strongly recommended. The first member on the scene leaves a message for the others. As additional information becomes available, it is added. This way, the handlers on the scene do not have to stop and continue to give briefings. This works well but requires discipline to leave messages and discipline not to call the handler on the scene to request an "update."

Finally, consider getting a conference bridge number that can be set up with instant notice. Let your team know the conference bridge access number but perhaps not the control number necessary to set up a conference. You also should print (and laminate if you can) a credit card-sized list of incident response team contact information. Include each team member's name, role, phone number, fax number, and PGP key fingerprint (if you are using PGP). Also, make sure to include the incident response conference bridge number. Pass out this credit card-sized list to each member of the team. In fact, give each member several copies for safekeeping.

- The incident handling team needs to be able to access systems
 - Sometimes without the knowledge of the system admin
- Can be controversial
 - Still, the incident handling team needs controlled access to computing resources
- Passwords for critical systems and crypto keys
- Strike a bargain with the operations team
 - You'll notify ops management before logging in
 - You'll use only handlers with the skills needed to admin that type of operating system

When I have been the system administrator of a production system, I have never been really comfortable making privileged passwords available to others. However, in an emergency, a handler might need access to critical systems. One organization has a policy that says passwords are kept in sealed envelopes in locked containers. After several years of implementation, the organization has reported that, although sometimes cumbersome, this system has worked well for the company. Note that there is a two-fold responsibility here; the system administrators must make sure the envelopes are kept up to date, and the handlers must make sure they tread lightly on the systems, keep the administrators up to date on any changes they make, and above all, never use a privileged password unless they are qualified on that operating system. One thing that is nearly certain to make an incident worse is having someone who has no clue what he is doing fumbling around as administrator or root.

To help encourage your operations team to give you admin-level access to machines, promise the following and then live up to the promise:

- You will notify the operations personnel on your incident handling team before you log in with admin credentials.
- You will use only handlers who have enough experience to administer machines of that given type.

Not many of us can change the way our entire organization does business, but we can certainly be responsible for the way that we do business. Encourage people to write down critical passwords and encryption keys and store them safely so they can be accessed if required. As encryption becomes ever more prevalent, an organization must set policy about who owns the secret keys and passphrases and under what circumstances they can be used and accessed.

- Establish a primary point of contact and an incident command communications center
- In critical sites, establish secured communications
- Set up resource acquisition plans for the teams
 - In advance, you need to get permission
 - During an incident, you may need to quickly procure something; get ready
 - Set aside or get permission to spend \$5,000 to \$10,000 without going through a multi-month procurement process

One of the functions the command post needs to be ready to provide is the rapid acquisition of things needed by the teams. This can span the gamut from a drive to store forensics images to a backup computer to pizza and Coke for a team that has been on the job for 12 hours and hotel rooms near the site so the team has a place to crash when exhausted. If you are a manager and you are thinking "yeah, yeah," whose credit card do you think is going to be pressed into service if you are not prepared?

Large organizations, in particular, can get very rigid about how things can and cannot be procured. It is wise to set up the exceptions to the rules and execute an incident handling drill where these exceptions can be tested in practice before they are needed.

Remember the test we apply to any of the recommendations: Would I be sorry if I didn't do it? Secured communications can be commercially available, including encrypted pagers and cell phones. They are costly and the phones at least may not give you the quality you really want. That said, if you are in a large organization and the incident involves many millions of dollars, this can be a real comfort. Large or small, there simply is no excuse whatsoever for an incident team that has not established a method for exchanging encrypted email and files, such as PGP or GnuPG.

- Provide easy-to-use, convenient reporting facilities for anomalous activities
 - Educate users as they are hired
 - Publish a list of indicators of an incident
 - Use multiple mechanisms
 - Phone reporting: An incident response hotline
 - Email: A main incident response mailbox
 - Intranet website devoted to incident handling
 - Reward reporting: Controversial
 - Continually update management
- Establish a war room
 - Should be a place where you can safely display information

Employees (especially network operations people, system administrators, and help desk workers) are the eyes and ears of any organization. One organization rewards employees who report a suspicious (potential) incident with a small cash award. Another writes a small article with a picture when an employee detects and reports an incident. These and other methods of feedback encourage the employees to be alert. These types of things pay off time and time again for the organization.

Make it as easy as possible to report; publish a well-known voice and fax number (your incident response hotline), an email address, and a web address of an internal website devoted to incident handling. Employees can experience uncertainty about whether to report an incident; we want them to be as comfortable as possible. Some people prefer to talk to people on the telephone, whereas others are more introverted and prefer email. Give employees a choice. In a major attack, you do not know which of your communication mediums will still be available.

The war room should have a lockable door and a lockable file cabinet. You should have a war room, not a war cubicle. Given the close quarters, it's helpful if the room includes a thermostat for keeping the temperature comfortable for the incident handlers and their equipment. Finally, I prefer to have no windows in my war room, to avoid curious stares from the outside world. By windows, we don't mean the operating system! We mean transparent panes of glass on the walls.

- Conduct training for team members
 - Set up a planning/training meeting on scenarios
 - Set up tools and techniques training
 - Consider deploying an internal honeypot for analysis
 - Stock some high-capacity drives and practice forensics imaging
 - (Advanced) Conduct war games
 - Conduct a penetration test unannounced and see how your team responds
 - Do this only with a more experienced team that has worked together at least six months to a year
- Counter Hack Challenges has some fantastic resources
 - Holiday Hack Challenge is available for free, maintained all year round

The #1 training issues are:

- Creating forensics images under fire
- Keyboard skills under fire

To deal with these issues, have your team practice, practice, practice. It is easy to teach the incident handling process at a general level. What takes persistence and concentration are to hone the skills needed by the on-site, at-the-console, incident handler.

Knowing one method of image creation is not enough; you need to be ready to move to an alternate approach if something goes wrong.

Knowing how to read the audit logs and investigate a file system requires knowing the operating system. This is far harder than being able to read expert witness reports, and knowledge comes only with training and practice.

We like to walk into the Computer Incident Response Team room at work and start a drill unannounced; it helps your team stay "combat ready." Also, an internal honeypot can be a helpful tool to hone the analysis capabilities of your team.

You also might want to consider conducting a penetration test of your environment unannounced to see how your team detects and responds to it. However, I caution you: Only conduct such "war games" with a more experienced team that has worked together for at least six months. Otherwise, an inexperienced team may trip over itself and result in negative feelings with such an experiment.

There are also outstanding resources, such as Counter Hack Challenges (<http://www.counterhackchallenges.com>), which can serve to continuously train your team through programs such as NetWars and NetWars Continuous. Counter Hack Challenges also makes a free online hacking challenge available as the Holiday Hack Challenge, available at <https://www.holidayhackchallenge.com>

- Coordinate closely with help desks
 - Help desk personnel are often the incident handler's initial eyes and ears
- Pay particular attention to relationships with system administrators and network administrators
 - Involve system administrators in your team
 - Trust your experienced admins' sense of things that "just aren't right"
 - Conduct proactive training
 - Recognize "power" log file reading
 - Encourage regular system backups by sys admins

We need to be candid with one another for a second: Many technical people denigrate the help desk function. They are often entry-level positions and perhaps they do not have the system programming skills that are developed over time. Handlers that wish to be successful best get down off their high horse. There is no substitute for the thousands of eyes that your users have; it is a sensor network beyond compare. When they see something "funny," they tend to report it to the help desk. Also, if a group is going to try social engineering, it is likely to be tried at the help desk. Investing in your help desk, making sure they are trained to be part of the response process, is sound practice!

System administrators and network administrators are the wild cards in incident handling. If incident handlers find themselves in a culture where the team is at odds with the organization's system admins, the organization has a real problem and it will manifest itself during an incident. The most probable reason for this tension is if the incident handling team is primarily drawn from the organization's security department instead of equally from security and operations. Remember, if you do not trust each other during good times, you will not work together well when you are under fire.

You simply can't handle a large incident without system and network administrators, but they are likely to make those critical mistakes that happen in the first five minutes of an incident. The best thing to do is get them involved, get them trained, and make them an integral part of your formal response capability.

By the way, if a system administrator, especially one that has been around for a while, calls you and says, "Hey, this just doesn't look right," you dismiss that clue at your peril. These folks know their systems and should be rewarded well when they find the subtle clue that indicates a compromise or intrusion that would have otherwise been missed.

- Great shared tool between Admins and Security
 - Free, and maintained by Google
 - Runs on Linux, OS X, and Windows clients
- Python-based agent
 - Remote memory analysis via Rekal
 - Detailed monitoring of clients
- GRR has the ability to pull in-depth forensic artifacts from multiple systems
 - Because the pull is asynchronous, it allows you to pull information from computers that are not on the network at the initial request, but rather when they are online again; very good feature for laptops

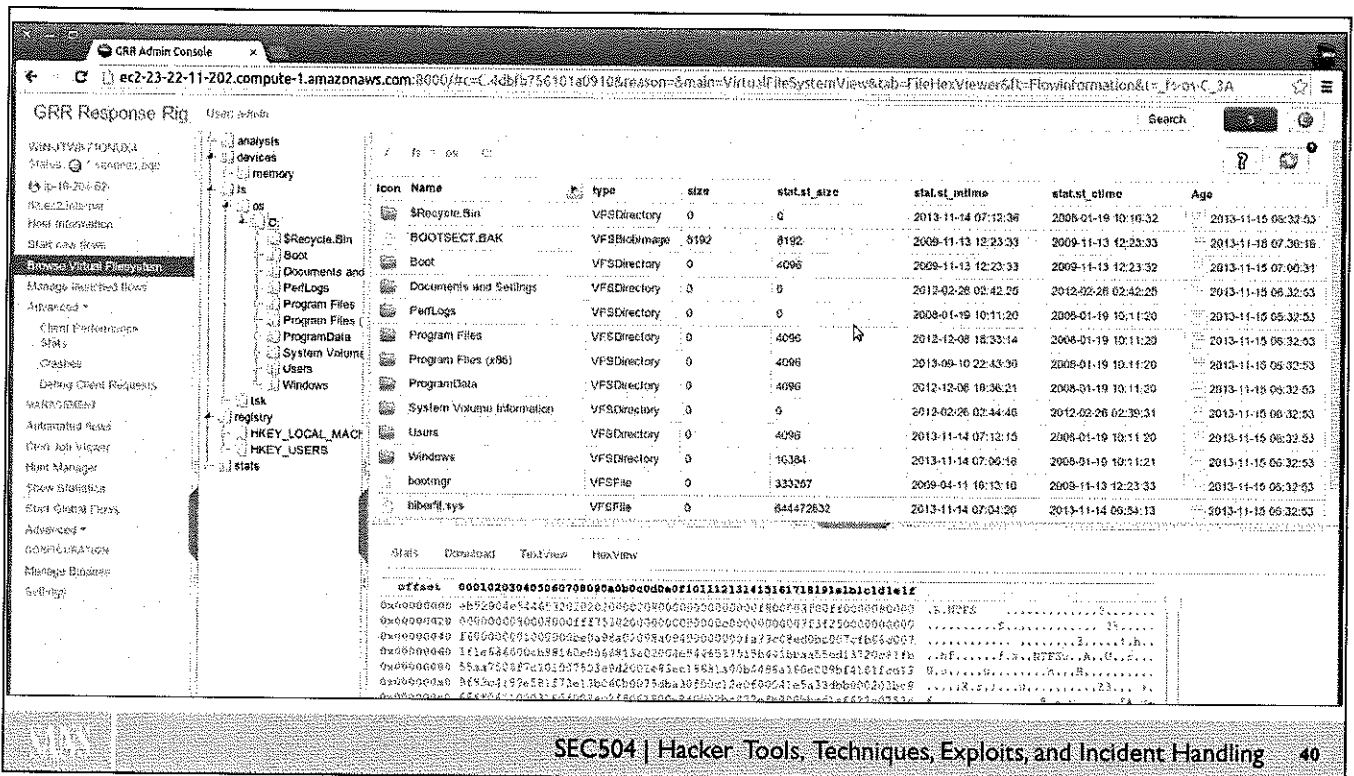


A fantastic tool for performing large-scale incident response and hunt teaming is GRR. Currently, this project is being maintained by Google and is free. It also runs on Windows, Linux, and OS X clients. One of our favorite pieces of functionality in this tool is the ability to perform memory analysis on remote hosts when coupled with Rekal.

It also has the ability to pull and store a wide selection of relevant data from a large number of hosts in an asynchronous manner. This means if a host is not on the network when a pull of information is requested, it can wait until that system is back online and connects in to gather the data. This is especially powerful for systems like laptops, which may not be online when the pull request is made.

You can get it here:

<https://github.com/google/grr>



One way to use GRR is to create a flow, which is a script that runs on the GRR server, but makes calls to the clients to perform various tasks. For example, you could create a flow to look for a file with a specific hash (or other properties). You could run this flow across several clients, even an entire enterprise, using a Hunt.

Once you've identified hosts that have the file of interest, you can use GRR to interrogate those systems further. For example, the screenshot on the slide above shows an analyst using GRR to examine the file system of a remote client.

- Get a duffle bag and keep it stocked with items for incident handling
 - Don't steal from your own jump bag
 - Always have it ready to roll
- Every jump bag is different, but some things to consider . . .
 - Fresh media for holding file system images
 - Evidence collection software (e.g. FTK Imager Lite)
 - Forensic analysis software (e.g. SIFT, EnCase, The Sleuth Kit)
 - Network taps, and every type of network cable you might ever need
 - PC repair kit tools (screwdrivers, tweezers, mechanic's mirrors)
 - Extra copies of forms, personal items (e.g. deodorant, aspirin)

A *jump bag* is the name for the bag (or container) that you use to carry your incident response tools. If you do not already have a jump bag, you might wish to try an exercise to see how long it would take to assemble one on the fly; you might find it can take weeks.

Remember, never steal from your own jump bag. You will regret it! Odds are, whatever you "borrow" from your jump bag "temporarily" will be the item you require on your next job. Of course, you likely forgot to replace it in your jump bag, so you are out of luck. Therefore, always keep your jump bag intact and replenished.

No two jump bags are alike, but some of the items commonly found in them include:

- Fresh media (e.g. blank USB thumb drives, blank hard drives) for storing evidence you collect.
- Evidence collection software such as FTK Imager Lite, or dd.
- Forensic analysis software such as the SANS Investigative Toolkit (SIFT), EnCase, The Sleuth Kit, and so on.
- Hardware for monitoring network traffic, such as a network tap.
- Every type of network cable you might ever need.
- PC repair toolkits which usually contain tools like screwdrivers, tweezers, and mechanic's mirrors.
- Extra copies of incident response forms.
- Personal items such as deodorant, aspirin, and a change of clothes.

Course Roadmap

- **Incident Handling**
- Applied Incident Handling
- Attack Trends
- Step 1: Reconnaissance
- Step 2: Scanning
- Step 3: Exploitation
 - Gaining Access
 - Web App Attacks
 - Denial of Service
- Step 4: Keeping Access
- Step 5: Covering Tracks
- Conclusions

Incident Handling

1. Overview
2. Incident Handling Process
3. Preparation
4. **Identification**
 - Lab 1.1: Windows Cheat Sheet
5. Containment
6. Eradication
7. Recovery
8. Lessons Learned
9. Enterprise-Wide IR
 - Lab 1.2: Enterprise Ident. and Analysis

This section focuses on the identification of an incident.

How do you detect an incident? The bulk of all detects will come from either sensor platforms or the things people just happen to notice. Sensors include firewalls, intrusion detection systems, and system logs (especially with Logwatcher software). To increase your chance of detection, consider burglar alarms sprinkled throughout your organization. These include personal firewalls and intrusion detection systems.

People can be your eyes and ears, and they are also spread around the organization. The trick is to give them the training to know that something is wrong and make sure they are aware of the risks and know to whom to report.

- Be willing to alert early!
- Don't be afraid to declare an incident
 - Even if there is no attack, you still help the organization
- Maintain situational awareness
 - Provide indications and warning
 - Provide current *intelligence* (up-to-date information) to incident handler
- Fuse or correlate information

There is a pronounced tendency to wait until we are sure something is wrong before we alert. This is death itself; the speed at which incidents occur requires us to field early. So, what if it is a false alarm? Use these as training opportunities.

For an incident capability to really perform, there needs to be a constant stream of information. Make sure that you construct your process to support and allow this constant stream.

This is one reason we recommend that you never send only one handler into the field unless you are very short of handlers. The second team member can maintain communications with the command center. This really helps with the situational awareness component.

- Assign a person to be the primary incident handler
 - Select a person to handle identification and assessment
 - Assign him a specific set of events on a specific set of systems to analyze
 - Empower him to escalate if needed
 - Call back to the incident handling chief if additional resources or expertise are required
- Ideally, assign a helper
 - If you have the resources, it's best to deploy two people to handle each incident to gather evidence

If one person isn't in charge, no person is in charge. For smaller incidents, often of the "would you check this out?" category, there is no need to send core incident handlers. Earlier, we discussed that a recommended practice was to have a core team of well-trained handlers, and have incident handling skills and training as part of the job for security officers or system administrators. An organization that does this benefits by having multiple levels of trained "firefighters." However, in such a case, it is important to set up assignments in a way that encourages the system administrator to succeed.

A known, full-time handler should be given the assignment in a way that it is clear what is expected of him: the quality of his investigation, what documentation he should produce, and when it is due. It is also important that he knows who he can call if he feels he needs additional support.

Ideally, it's best to deploy two people to handle each incident to gather evidence more thoroughly. Therefore, assign a primary handler and a helper.

- Enforce a *need to know* policy
- Tell the details of the incident to the minimum number of people possible
- Remind them that they are trusted individuals and that your organization counts on their discretion
- Inform them that they may be required to testify
 - This may scare them, but that's OK

Nothing spreads faster than a rumor! Let's be up front about this. In many organizations, the culture is "we trust our people." That is great. If you go to court, the defense has the right, even duty, to call as many witnesses as required. If 18 of the 20 people called to testify never understood what happened, failed to take notes, and are now recounting the event a year after it occurred, what do you think will happen? A legal disaster might ensue.

Also, a tremendous percent of the time, what you originally think is going on turns out not to be the case. This is common and is an expected part of the incident handling process; in fact, the process is explicitly designed to handle this. However, if those first clues and theories get published in the newspaper, it can be embarrassing to your organization.

Sometimes it takes a long time to bring an incident to closure; sometimes we are dealing with an insider. If someone blabs, that can be the tip-off that ruins your investigation.

Finally, many incidents occur because an individual made a mistake; perhaps someone was negligent in securing the systems they managed. This individual may need to be admonished, but if your team is the one that leaks the info, you will be regarded with mistrust from then on. If that happens, it is difficult to get the information you need to do your job.

- If the computers may have been compromised, avoid using them for incident handling discussions (email or chat)
- Rely on out-of-band communications
 - Use telephones and faxes
 - Be careful with VoIP, which can be sniffed and played back using a variety of tools (Wireshark, Cain, and VOMIT) if it is not encrypted
- Make sure the team can send encrypted email, such as GnuPG, PGP, S/MIME, and so on
 - Share keys in advance
- Possibly encrypted cloud storage, such as Tresorit or SecureSafe

It is possible to compromise a system, break root, and have a sniffer installed and running in less than 30 seconds. Once in control of a system, the attacker can monitor the email. In such a case, the attacker could reasonably track every move you make if you use the network to discuss it. On that note, you aren't really an incident handler if you don't have the ability to send encrypted and signed email and files, using tools like PGP or Gnu Privacy Guard. People can use your public key to send you sensitive information, and you can use your private key to "sign" instructions so people know they really come from you.

If the computers you'd use to send email or conduct a chat are infected or compromised, pick up the phone and give them a call instead. Be careful with VoIP connections, however, unless you have deployed VoIP with solid encryption. There are a variety of tools that can turn a sniffed packet capture file of a cleartext VoIP conversation into an audio file. This functionality is supported in Wireshark, Cain, and a tool called Voice Over Misconfigured Internet Telephones (VOMIT).

Faxes are a wonderful tool in incident handling. If at all possible, keep a directory with all the fax numbers in your organization and their locations. Make sure your people have actual paper-based fax machines, and not one of those free fax-to-email conversion services. First off, such services send data in email, in cleartext. Secondly, if your mail servers go down, you won't be able to communicate with your team.

Cell phones are important and handlers should be issued cell phones. In a long incident, several sets of batteries may be needed. On the global scale, the Internet Storm Center has set up an out-of-band network of ham radio operators. With worm class attacks, it is entirely possible that the internet can be disabled. If that happens and enough people fall over to dial-ups, it could affect the phone system. The reason it could affect the phone system is that there is a finite capacity of circuits. Think about it. Have you ever tried to call someone in an earthquake, hurricane, or any other significant event and received the "I'm sorry, all circuits are busy" error message? Incident handlers must think about out-of-band communications before the event!

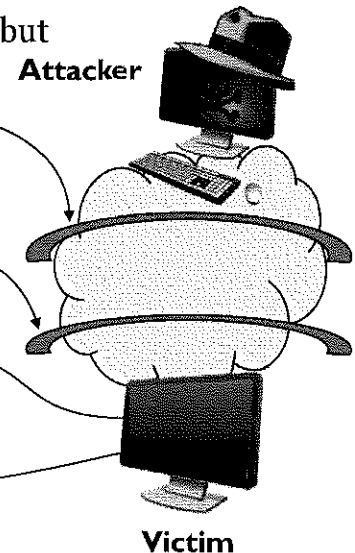
Make sure that the incident handling team has the ability to send and receive encrypted email among the team members. Consider procuring encryption tools, such as the free Gnu Privacy Guard (GnuPG), the commercial PGP, or the various S/MIME solutions available today. Be sure to exchange keys among the incident handling team in advance!

There are also encrypted cloud storage providers, such as Tresorit or SecureSafe. These providers store your encrypted IR files. Only systems with the proper authentication, encryption keys, and client will be able to access the data. Be careful, this may violate some corporate policies.

Where Does Identification Occur?

Identification

- Identification can happen anywhere in your environment, but especially helpful zones for gathering events are
- Network perimeter detection
 - Identification occurs on network
 - Firewalls, routers, external-facing network-based IDS, IPS, DMZ systems, etc.
- Host perimeter detection
 - Identification occurs when data enters or leaves a host
 - Personal firewalls/IPS, local firewalls, port sentry tools
- System-level (host) detection
 - Identification occurs based on activity on the host itself
 - Antivirus tools, endpoint security suites, file integrity tools, user noticing strange behavior
- Application-level detection
 - Application logs (web app, app server, cloud service, etc.)



SEC504 | Hacker Tools, Techniques, Exploits, and Incident Handling 48

When we consider a high-level view of a network architecture, identification can occur pretty much anywhere in your environment as you gather events. To help categorize the various zones in your environment to analyze for evidence of attacks, consider these four levels: network perimeter, host perimeter, host (or system), and application level.

Our network perimeter is monitored by firewalls, routers that generate logs, external-facing intrusion detection systems, intrusion prevention systems, and other machines on the DMZ. These systems can give us earlier warnings about attacks as they monitor our borders with the internet and other external networks.

The next layer down is the host perimeter, where we monitor activities across each host system's interface, analyzing what the machine is sending out to and receiving from the network. This border can be monitored using personal firewalls, host-based intrusion prevention systems, local firewalls, and port sentry tools.

The next level of detection is host-based, where we monitor the actions on the host systems themselves. Antivirus tools, file integrity checkers, and endpoint security suites often operate at this level. Also, a user noticing strange behavior on her desktop or laptop system falls into this category.

The final level is the application level, which is typically monitored via the logs generated by the application. The application may be a web application, a server-side application used by thick clients, or even a cloud-based service.

Ideally, you want to catch the attack at your perimeter, but sometimes (often, in fact), detection only occurs at the host or application level.

Network Detection Example

Identification

```
root@snort:~# tcpdump -nn port 27017
```

```
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
```

```
eth0, link-type EN10MB (Ethernet), capture size 262144 bytes
```

Scanning to see if TCP port 27017 is listening on each target in a range. This port is for Mongo DB. Two hosts (.140 and .143) are listening on 27017

```
804 IP 192.168.179.142.45796 > 192.168.179.145.27017: Flags [S]
862 IP 192.168.179.142.45796 > 192.168.179.146.27017: Flags [S]
882 IP 192.168.179.142.45796 > 192.168.179.148.27017: Flags [S]
909 IP 192.168.179.142.45796 > 192.168.179.140.27017: Flags [S]
929 IP 192.168.179.142.45796 > 192.168.179.147.27017: Flags [S]
946 IP 192.168.179.142.45796 > 192.168.179.143.27017: Flags [S]
980 IP 192.168.179.146.27017 > 192.168.179.142.45796: Flags [R.]
992 IP 192.168.179.140.27017 > 192.168.179.142.45796: Flags [S.]
994 IP 192.168.179.142.45796 > 192.168.179.140.27017: Flags [R]
995 IP 192.168.179.143.27017 > 192.168.179.142.45796: Flags [S.]
998 IP 192.168.179.142.45796 > 192.168.179.143.27017: Flags [R]
1015 IP 192.168.179.148.27017 > 192.168.179.142.45796: Flags [R.]
1044 IP 192.168.179.147.27017 > 192.168.179.142.45796: Flags [R.]
```

SNAN

SEC504 | Hacker Tools, Techniques, Exploits, and Incident Handling

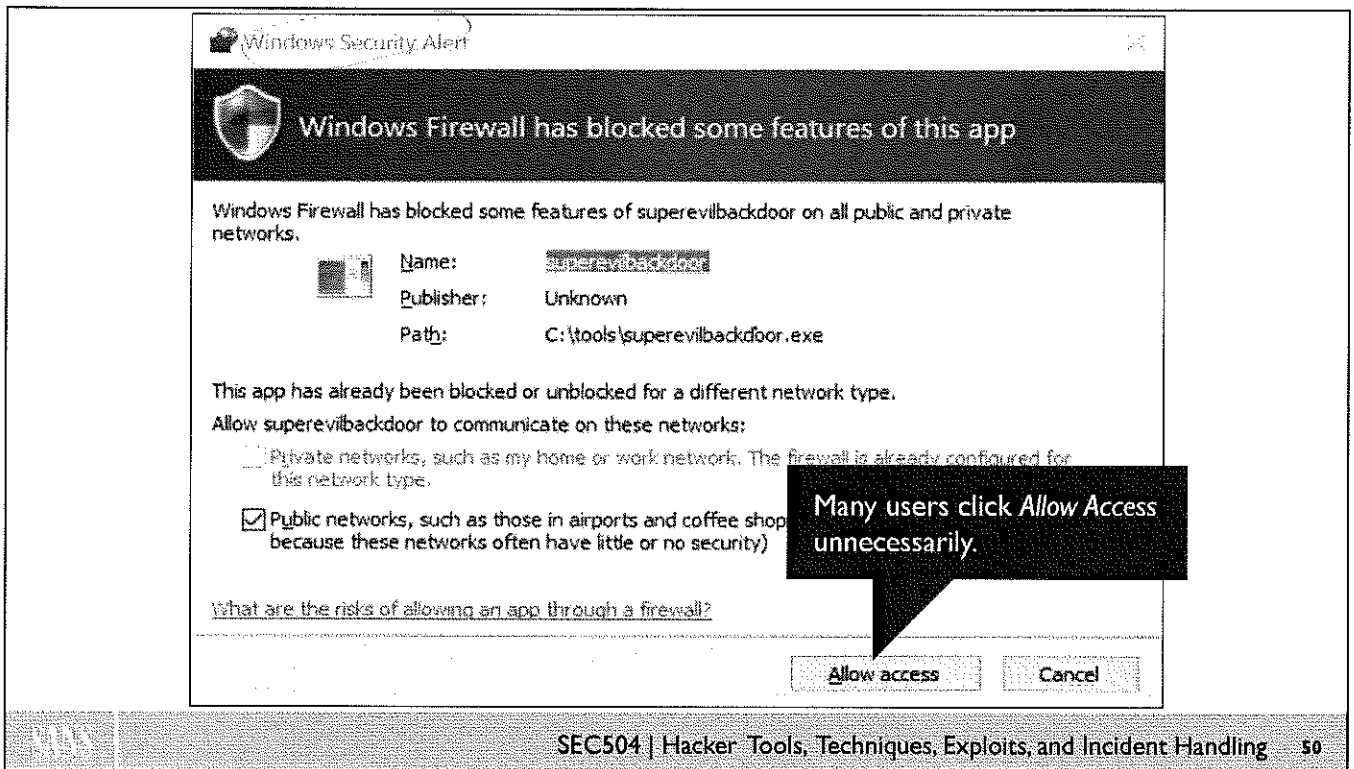
49

The slide above shows an internal IDS sensor capturing network activity from a scanning tool. Specifically, the scanning tool is attempting to find which hosts within a range of IP addresses are listening on TCP port 27017. Two hosts, 192.168.179.140, and 192.168.179.143 respond with SYN/ACK packets, indicating they are listening on TCP port 27017.

TCP port 27017 is usually associated with the MongoDB service. There have been some very costly (to the victim) ransomware incidents where the data on unprotected MongoDB servers was encrypted.

Looking at the output, there are some points worth noting. First is the odd TCP handshake with the two machines that are listening on TCP port 27017. After each listening host responds with a SYN/ACK, the scanning machine responds with a RESET instead of the usual ACK. Second, the source port for each SYN packet sent by the scanning machine is static. It's common for operating systems to increment the source port for each new connection.

Given the non-standard behavior, it is quite likely the attacker is using a tool to craft custom packets. If the scanning system were a random machine on the internet, this might represent network enumeration before an attack. However since the scanning system is internal, it strongly suggests this machine has been compromised. Even more troubling is that root or Administrator permissions are normally required to run packet crafting tools.



This graphic shows an example of a host perimeter detection by a personal firewall alerting us to the fact that `superevilbackdoor.exe` is trying to listen on the network.

Unfortunately, many users will blindly click *Allow Access*. However even if the user does allow the program to listen, we still have other options.

```
C:\> netstat -naob | more
[mqsvc.exe]
TCP        0.0.0.0:4444      0.0.0.0:0        LISTENING
[superevilbackdoor.exe]
TCP        0.0.0.0:5357     0.0.0.0:0        LISTENING
Can not obtain ownership information
TCP        0.0.0.0:16992    0.0.0.0:0        LISTENING
[LMS.exe]
TCP        0.0.0.0:49664    0.0.0.0:0        LISTENING
Can not obtain ownership information
TCP        0.0.0.0:49665    0.0.0.0:0        LISTENING
EventLog
svchost.exe]
```

TIP

Port 4444 is the default port for most Metasploit payloads

On this slide we see an example of using the `netstat` command for host perimeter detection. The `netstat` command shows us information about processes listening on the local system. The `-n` means to show port numbers and IP addresses instead of port names and host names. The `-a` tells `netstat` to show all processes. The `-o` and `-b` tell `netstat` to show the owning process id and executable (or DLL) name associated with a listening process.

You can see that `superevilbackdoor.exe` is listening on TCP port 4444. A point worth noting is that port 4444 is the default port for many Metasploit payloads. So while listening on port 4444 does not *guarantee* `superevilbackdoor.exe` is malicious, it's definitely worthy of further investigation.

At the host perimeter we can get valuable information about what is happening on that host. However our global view is much less than network perimeter detection.

- Look up the service (port list)
 - Internet Assigned Numbers Authority (IANA)
 - Also, Google . . .
- Does the destination host run the service?
 - Are you sure?
 - Is it included in your enterprise asset list?
- Could it be a backdoor? A service invoked by an attacker?
 - A useful port list for legitimate and malicious use of ports is available at SpeedGuide

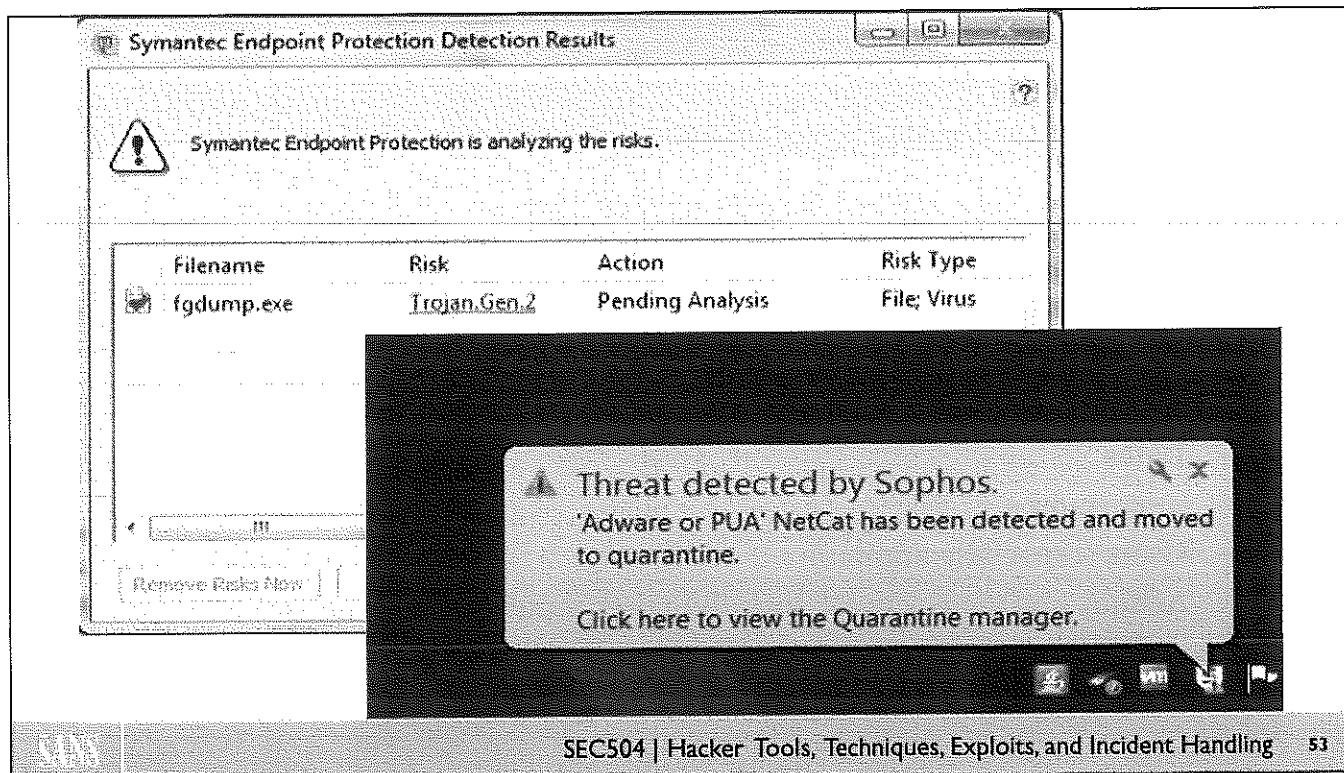
For network perimeter and host perimeter analysis, when you determine a listening port number, you should look up the port to see its official assignment, as well as the potential malicious use of that port. There are a couple of relevant port lists. The official port list is maintained by IANA (<http://www.iana.org/assignments/service-names-port-numbers>). You may also want to consult a port list for commonly used trojans and/or malicious code. Alternatively you can always type `port portnum` into Google (substituting `portnum` for the actual port number).

Can you be sure that the packet detected at the perimeter is what the port list says? No, of course not; trojans operate on TCP port 23, which is normally associated with the Telnet service. Additionally, many ports have more than one interpretation. However it is an important start.

Does the intended destination run the service that the packet had in its destination port field? If not, this probably doesn't indicate a good job of reconnaissance on the attacker's part, and the risk is probably very low. On the other hand, if it looks like the attacker knows what she is looking for, you may be in some trouble.

How can you find out if the service is running? Try `lsof -i` on UNIX and also `netstat -a` on UNIX or Windows. Also, you can scan the system with a port scanner like Nmap, but this may not be reliable. Some attacker software only answers specially formatted queries.

A handy list of legitimate and malicious use of ports can be found at <http://www.speedguide.net/ports.php>



This graphic shows system-level detection via an antivirus tool and an antispyware tool, both of which alerted when fgdump and netcat tried to run. The redundant layers of protection are a good idea, and we get even more information about what's happening at the system level.

- Application logs are especially useful from
 - Web apps
 - App servers for thick-client apps
 - Cloud-based services
- Particularly useful data
 - Dates
 - Timestamps
 - Users (especially admins)
 - Actions and transactions, including user input variable values

At the application level, incident handlers can analyze application logs to get a feel for unusual activity that could be associated with an attack that sometimes cannot be discerned by looking at the other layers of identification (network perimeter, host perimeter, and host-level). Such application logs are especially useful when gathered from web applications, application servers that support thick-client applications, and cloud-based services.

Incident handlers should make sure that they have access to such application log data. For the most important applications in their organization, they should also check in advance to ensure that the data includes useful elements of each action taken within the application. Vital data elements, such as dates, timestamps, users (especially administrative-level accounts), and actions/transactions, should all be recorded for later analysis. For logged actions and transactions, it is especially helpful if the logs record all user input variable values, a frequent avenue of application-level attack.

See the following resources for reviewing logs for other commonly used applications:

- Oracle: http://docs.oracle.com/cd/E27559_01/admin.1112/e27239/audit.htm
- Apache: <http://httpd.apache.org/docs/current/logs.html>
- IIS: <http://support.microsoft.com/kb/324091>

Application-Level Detect: Wordpress

Identification

Event ID	Date	Source IP	Message
1002	03-28-2019 5:44:56.606 PM	52.231.156.213	2 failed login(s) detected.
1002	03-28-2019 5:43:05.167 PM	58.229.206.215	1 failed login(s) detected.
1002	03-28-2019 5:42:12.821 PM	94.23.199.47	10 failed login(s) detected.
1002	03-28-2019 5:40:10.149 PM	151.80.185.177	1 failed login(s) detected.
1002	03-28-2019 5:39:05.401 PM	145.239.117.227	1 failed login(s) detected.
1002	03-28-2019 5:37:33.288 PM	54.37.208.190	1 failed login(s) detected.

From the network perspective, these are just a bunch of HTTP requests.

The application's audit log shows what was really happening.

This graphic shows password guessing attacks conducted against a publicly accessible Wordpress website.

The value of application-level detection becomes apparent when you consider what this would look like from a network perspective: a series of HTTP requests.

However, by looking at the application's audit logs it becomes apparent what was really going on.

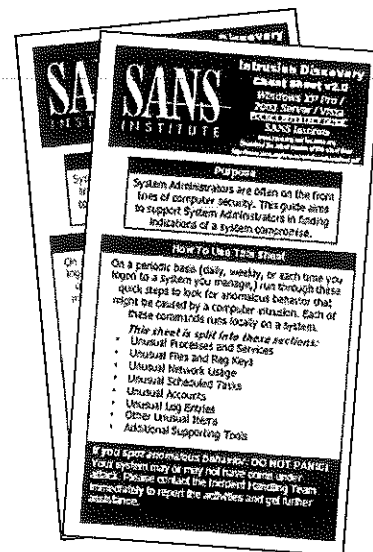
- Ideally, you want to detect attacks at your network perimeter
- Unfortunately, some attacks are stealthy and are detected only after infiltration occurs
- Many incidents are identified only when another site detects *your* site attacking them
- This can cause your site to be blocked or posted on a "shame" website
- That's why you want identification capabilities at all four levels: Network perimeter, host perimeter, host level, and application level

If you visit the website <https://isc.sans.edu/top10.html>, you notice a top-ten attacker's list of IP addresses. From time to time, we get irate emails from the owners of such an IP address until they realize the implications of being one of the internet's ten least wanted.

Many sites create ACLs directly from lists like this, so this could put an aspiring "dot.com" out of business faster than no business plan and no product.

For that reason, incident handlers should make sure that they have access to attack identification information across all the four levels we discussed: network perimeter, host perimeter, host level, and application level.

- SANS Intrusion Discovery cheat sheets can be helpful
 - Designed for system admins to spot trouble and call incident handling team for help
 - One page for Windows and one for Linux, each a trifold
 - Available at SANS Pen Testing Resources page
 - Also included on the Course USB in the Cheat_Sheets directory
 - Free . . . make as many copies as you'd like; just don't sell them



To help improve the identification process inside of organizations, SANS created its Intrusion Discovery cheat sheets for Windows and Linux. They are freely available online at the SANS Pen Testing Resources page: <https://pen-testing.sans.org/resources/downloads>. They are also included on the Course USB, located in the Cheat_Sheets directory. These sheets are designed to educate system administrators in actions they can take to look for anomalous behavior on their machines, including unusual processes, files, network usage, and so on. If system administrators spot trouble, the sheets instruct them to call the incident handling team (you). Put your name and phone number on a sticker, affix it to the front of the cheat sheets, and pass them out to all of your system administrators.

We'll spend much of the rest of this class (books 2–5) analyzing various events triggered by each attack we discuss. This initial list is an overview of the details we'll encounter for the rest of this class.

- No set of actions can detect every attack, but we shoot for the most common signs
 - Careful attackers using highly stealthy tools will be able to fly under the radar screen
 - But we'll still catch many attackers
 - Even the best attackers sometimes let down their guard
- These sheets expect system admins to know the "normal" state of their systems
 - So they can spot abnormal events
 - Cheat sheet tells them what to get familiar with and where to look for deviations from the norm

Our cheat sheets face some obvious limitations. First, no set of simple-to-perform commands is going to find every single attack. If the bad guy is especially careful to cover his tracks and employs extremely stealthy tools, we won't be able to detect his presence. Still, many attackers aren't all that careful, and even some of the most powerful tools leave tracks that we can spot. Even the more sophisticated bad guys accidentally leave a few interesting tidbits for an observant system administrator to discover.

Another major limitation associated with these cheat sheets is that they require the system administrators to know the "normal" state of their systems. The cheat sheets identify common areas of deviation from normal that a knowledgeable system admin can spot. However, without a good gut feel of the normal status, these techniques won't work. Therefore, over time, by exercising the tools and techniques covered in the cheat sheets, system admins gradually grow more comfortable with what is normal. After learning the normal state, they can better spot anomalies.

- Have the system administrators look for unusual (18)
 - Processes and services
 - Files
 - Network usage
 - Scheduled tasks
 - Accounts
 - Log entries
 - Other unusual items
 - Additional supporting (third-party) tools
- Let's look at Windows cheat sheets in more detail
 - The Linux cheat sheet is in your workbook (lab optional)

Beyond those common elements, the cheat sheets break down the specific technical activities into eight sections. We need administrators to periodically look for unusual processes, files, network usage (including TCP and UDP ports, as well as promiscuous mode where possible), scheduled tasks, and accounts.

We also need them to look at log entries for strange activities. Unfortunately, there are thousands of log entries that could be a sign of attack, yet we are confined to a single page. Therefore, we've listed a handful of the most common log items that might indicate an attack. We also list a handful of other unusual items an administrator should look for.

Finally, we list some supporting, third-party tools that go beyond the base operating system install to help secure the system. These tools are immensely helpful, and some of them should have been built into the operating system from the start. By adding them, we can significantly improve a system administrator's ability to view the status of a machine. NOTE: If you don't want your system administrators to install these items on their machines, make sure you delete this section from the cheat sheet before distributing it to them. We put this item on the back panel of the trifold so that it can easily be omitted from your copying process without looking strange.

To get into the details of these cheat sheets, we'll look at the Windows cheat sheet. You can take a look at the Linux cheat sheet as an optional lab exercise in your workbook.

- The latest cheat sheet applies to Windows XP Pro through Windows 10
 - The cheat sheet is on the Course USB, called `winsacheatsheet_2.0.pdf`
 - For earlier and less powerful versions of Windows (Windows 2000, XP Home, 10 Home), we have `win2ksacheatsheet.pdf`
 - There is also an older version on the USB called `winsacheatsheet_1.4.pdf`
 - That version is just for backward compatibility purposes

On the Course USB, in the `Cheat_Sheet` directory, there are several versions of the Windows cheat sheets. The latest version, called `winsacheatsheet_2.0.pdf`, applies from Windows XP Pro up to and including Windows 10. These versions of Windows include several new tools for analysis and troubleshooting built in, of which the cheat sheet takes advantage.

For earlier versions of Windows, including Windows 2000 Pro and Server, and less powerful versions of Windows, such as XP Home and Vista Home, we have included another version of the cheat sheet called `win2ksacheatsheet.pdf`. Given the limitations of these operating systems, this cheat sheet does not include as many commands simply because these operating systems don't have useful software for analysis built in. Yes, you will encounter these older systems in IR and security assessments.

Finally, some organizations still rely on the earlier version of the cheat sheets (v. 1.4). We keep that one on the USB for backward compatibility purposes.

For this class, we'll look at the latest and greatest version of the Windows cheat sheet (v. 2.0).

- On the Windows cheat sheet, we generally show the admins how to check a given item in the GUI
- Followed by one or more methods for checking the same item at the command line
- It's good to have multiple methods for checking the same thing
- There is no "best" place to start for every situation
 - Usually determined by the circumstances
 - When in doubt, start with either processes, or network connections
- Regardless of what you look at first, you always want to be thorough and examine all areas of a system

Generally speaking, on the Windows cheat sheet, we describe how to check a given item using the GUI first, followed by one or more methods for checking the same thing at the command line.

We offer multiple approaches for checking things for a few reasons. First, some admins are more comfortable with the GUI, while others prefer the command line. Next, the command line lends itself better to scripting, so system admins who want to write scripts to analyze their machines have a starting point for doing so. Also, with multiple methods for checking, the cheat sheets can help function as an educational tool, showing admins some of the features of their systems that they might not have known existed.

There is no "best" place to start any type of investigation or incident response process. Usually it is determined by the circumstances. For example, if you are using these commands during an incident that involves network communication, you might want to start by looking at network connections. If instead one of the indicators of the malware was a specific process name like `superevilbackdoor.exe`, you might start by looking at the running processes.

When in doubt, you can choose to start with either processes, or network connections since every system will have those.

Finally, regardless of what you look at first, you always want to be thorough in your work and examine all aspects of a system.

```
C:\> net view \\127.0.0.1
```

Look at file shares

```
C:\> net session
```

Look at inbound SMB sessions

```
C:\> net use
```

Look at outbound SMB sessions

```
C:\> nbtstat -S
```

Examine NetBIOS over TCP/IP activity

To look for unusual network activity on a Windows machine, we start out by looking at available shares using the `net view \\127.0.0.1` command. This action will show all file shares on the local machine. System administrators should check to make sure each share has a defined business need.

Additionally, administrators can see if anyone is connected to these shares by running the `net session` command. The output of this command shows any NetBIOS and/or SMB connections associated with file and print sharing and other activities.

The `net use` command shows whether this local machine has made any NetBIOS/SMB connections to other systems. `net session` shows connections *to* this machine, and the `net use` command shows connections this machine has *initiated*.

Finally, we have the administrators run the `nbtstat -S` command to focus on NetBIOS over TCP/IP activity. These connections and shares are likely included in the results of the earlier commands on this slide, but a final `nbtstat` check couldn't hurt. The `-S` indicates that we want to see systems connected to our machine, listed by IP address.

```
C:\> netstat -na
```

Look for unusual TCP and UDP ports

```
C:\> netstat -naob
```

Show owning process id and associated executables / DLLs

```
C:\> netstat -naob 5
```

Automatically refresh every 5 seconds

```
C:\> netsh advfirewall show currentprofile
```

Examine built-in firewall settings (Win7 – Win10)

Beyond the built-in Microsoft SMB and NetBIOS components, we also need to look at TCP and UDP activity. The `netstat -na` command shows listening and active TCP and UDP ports. By putting a number n after this command, Windows continuously runs the command and updates the display every n seconds.

An additional useful item is the `-o` option for `netstat` (as in `netstat -nao`). This flag displays the owner process ID associated with each listening TCP and UDP port.

With the `-b` option, `netstat` shows the EXE using the port and the DLLs that it has loaded to interact with the port.

For the `netstat` command to be useful, of course, the system administrator must have a good feel for the normal TCP and UDP activity of the machine.

Finally, an administrator can dump the detailed configuration of the built-in Windows personal firewall by running the following command:

On Windows 7 through Windows 10:

```
C:\> netsh advfirewall show currentprofile
```

On XP or Windows 2003:

```
C:\> netsh firewall show config
```

- Start Task Manager

- Look for unusual/unexpected processes
- Focus on processes with usernames SYSTEM, Administrator, or users in the administrator group

```
C:\> tasklist
```

Examine processes at the command line

```
C:\> tasklist /v
```

Run tasklist verbosely for more detailed output

```
C:\> tasklist /m /fi "pid eq pid"
```

Get command-line options & loaded DLLs for a process id

To look for unusual processes on Windows, the cheat sheet describes running the Task Manager tool by going to Start | Run and typing `taskmgr.exe`

At the command line, admins can also view which process is running on the machine by using either of the following commands:

```
C:\> tasklist
```

For more details, you can run `tasklist` verbosely as

```
C:\> tasklist /v
```

Of course, this requires the system administrator to know what processes are supposed to be running on the machine. Armed with the knowledge of the norm, she can then spot deviations.

And, you can get command-line options and loaded DLLs (modules):

```
C:\> tasklist /m /fi "pid eq pid"
```

Examining Processes with WMIC

Windows Cheat Sheet

```
C:\> wmic process list brief
```

Get brief information about running processes

```
C:\> wmic process list full
```

Get lots of information about running processes

```
C:\> wmic process get name,parentprocessid,processid
```

Get specific fields

```
C:\> wmic process where processid=pid get commandline
```

Focus on a specific process

Pay attention to base64-encoded command-line options!

SEC504 | Hacker Tools, Techniques, Exploits, and Incident Handling

65

While the `tasklist` command is good, the `wmic` command provides access to *very* detailed information about running processes.

To get just a few details about running processes you can type

```
C:\> wmic process list brief
```

Or to list the full details for each process:

```
C:\> wmic process list full
```

Since that is a lot of output, you might want to focus on just a few relevant fields. For instance you can get a process name, parent process ID, and process ID:

```
C:\> wmic process get name,parentprocessid,processid
```

If there is a particular process of interest, you can use a `where` filter:

```
C:\> wmic process where processid=pid get commandline
```

Pay close attention to any running processes that have base64-encoded command-line options, especially PowerShell. A number of different attackers and malware have used PowerShell's `-EncodedCommand` option to specify the *content* of a script to run at the command line. This way there are no extraneous `.ps1` files left lying around on disk. (In a few slides we'll see how to use your Windows VM to decode base64-encoded data.)

Of course not *every* program that uses `-EncodedCommand` is malicious, but they are definitely worthy of further investigation.

```
C:\> services.msc
```

Examine services using the services control panel GUI

```
C:\> net start
```

Examine running services

```
C:\> sc query | more
```

Get more detail about each service

```
C:\> tasklist /svc
```

Map running processes to services

The services control panel GUI, which shows various services and their status, can be invoked by typing at the Start | Run box or the command prompt:

```
C:\> services.msc
```

At the command line you can use the `net` or `sc` commands to get information about running services. To see a list of running services with the `net` command:

```
C:\> net start
```

The `sc` command provides more detail of the status of each service:

```
C:\> sc query | more
```

Finally, to see which services are running out of each process on your system, you could run:

```
C:\> tasklist /svc
```

- There are numerous registry and file locations that start software automatically
 - Called Autostart Extensibility Points (ASEPs)
- Registry keys commonly used by malware
 - `HKLM\Software\Microsoft\Windows\CurrentVersion\Run`
 - Also `RunOnce`, and `RunOnceEx`
 - Inspect both `HKLM` and `HKCU`

Query specific registry ASEPs at the command line with `reg`, or use the `regedit` GUI

```
C:\> reg query HKLM\Software\Microsoft\Windows\CurrentVersion\Run
C:\> reg query HKLM\Software\Microsoft\Windows\CurrentVersion\RunOnce
C:\> reg query HKLM\Software\Microsoft\Windows\CurrentVersion\RunOnceEx
```

Windows has numerous registry and file locations that can be used to start software without a user taking a specific action such as double-clicking on a program's icon. These locations are called Autostart Extensibility Points (ASEPs). Some sites also refer to these as Autostart Entry Points.

Despite the large number of ASEPs, quite a bit of malware uses the same few keys in the registry. These keys are responsible for executing programs when a system boots up or when a user logs on. These commonly used keys are:

```
HKLM\Software\Microsoft\Windows\CurrentVersion\Run
HKLM\Software\Microsoft\Windows\CurrentVersion\RunOnce
HKLM\Software\Microsoft\Windows\CurrentVersion\RunOnceEx
```

A diligent system administrator who suspects system compromise should check out the values assigned under these registry keys. The keys should be checked both in the `HKLM` (as shown above) and in the `HKCU` hives.

These items can be viewed in a GUI via the `Regedit` command:

```
C:\> regedit
```

Or, at the command line, their settings can be observed by running the `reg` command as follows:

```
C:\> reg query regkey
```

The `reg` command is case insensitive, which is nice if you have trouble remembering the capitalization of the various elements in the registry. Here is a command that reads settings of the `Run` registry key associated with the local machine (`HKLM`):

```
C:\> reg query HKLM\Software\Microsoft\Windows\CurrentVersion\Run
```

One of the best tools for reviewing the ASEPs on a Windows system is `autoruns.exe` from Microsoft. It's not built into Windows, but it's definitely something worth considering. It can be found here:

<https://docs.microsoft.com/en-us/sysinternals/downloads/autoruns>

```
C:\> dir /s /b "C:\Users\username\Start Menu"
```

Check user autostart folders

- Sometimes it's easier to use a tool to summarize ASEP information

```
C:\> start msconfig.exe
```

Use msconfig to examine startup items

```
C:\> wmic startup list full
```

Use wmic to examine startup items

The cheat sheets also list some of the autostart folders associated with users. These programs are automatically invoked each time the given user logs on to the system, and are sometimes altered by malware.

To list the contents of a user's autostart folder on Windows Vista and Windows 7 systems type:

```
C:\> dir /s /b "C:\Users\username\Start Menu"
```

On Windows 8 and newer the startup folder is located in: `C:\Users\username\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\Startup`. On systems before Windows Vista the equivalent command is:

```
C:\> dir /s /b "C:\Documents and Settings\username\Start Menu"
```

Sometimes it's easier to use automated tools to collect and summarize various ASEP folders and registry keys. The built-in tool `msconfig.exe` will do this for you. It's worth noting that on some versions of Windows `msconfig.exe` is not in the PATH used by `cmd.exe`. You can run it by clicking on the Start Menu, then click Run and type `msconfig.exe`. Alternatively you can run it at the command line using the `start` program.

```
C:\> start msconfig.exe
```

Instead of `msconfig`, you can use `wmic` with the `startup` option to inspect autostart programs.

```
C:\> wmic startup list full
```


- Look for new, unexpected accounts in the administrators group

```
C:\> lusrmgr.msc
```

Launch the GUI

OR

```
C:\> net user
```

List users

```
C:\> net localgroup administrators
```

Show who is in the administrators group

The Windows cheat sheet asks administrators to look at the users and groups defined on the machine using the `lusrmgr.msc` control. This local user manager interface can be used to check for unexpected accounts in the Administrators group.

At the command line, a list of users can be displayed by running the `net user` command. To see which accounts are in the administrators group, the following command can be run:

```
C:\> net localgroup administrators
```

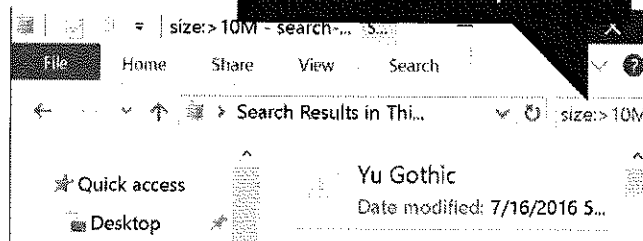
- Check file space usage for sudden major decreases in space

```
C:\> dir c:\
```

Look for major decreases in disk space

Search for files larger than 10 MB at the command line

Search for files larger than 10 MB in File Explorer



```
C:\> FOR /R C:\ %i in (*) do @if %~zi gtr 10000000 echo %i %~zi
```

Next, the Windows cheat sheet describes how to look for major decreases in free space on a machine. By using the GUI or the `dir` command, an administrator can spot normal disk utilization and look for deviations.

We also have them search for unusually large files by using the system search routine to look for files larger than 10 MB. Such files could contain an attacker's sniffer logs, stolen software, or pornography.

This search can be accomplished at a `cmd.exe` prompt by running

```
C:\> FOR /R C:\ %i in (*) do @if %~zi gtr 10000000 echo %i %~zi
```

This command is based on a FOR loop, which iterates over a set of something. The `/R` indicates that we are to iterate over files, recursively going through the file system. We start at `C:\`. Our iterator variable (`%i`) takes on the different names of various files. We'll iterate over files of any type (`in (*)`). For each file we iterate over, in our `do` clause, we turn off the display of commands (`@`) and use an IF statement. The iterator variable's file size is referred to as `%~zi`. We check to see if the file's size is greater than 10 MB (`gtr 10000000`). If it is, we display (`echo`) the file's name (`%i`) and its size (`%~zi`).

Alternatively, if we have GUI access to the machine, we can look for files with that size using built-in Windows search features. On Windows 10 you can type `size:>10M` into the search box on the upper right hand of the Windows File Explorer tool.

- Look for unusual scheduled tasks
 - Especially those that run as SYSTEM, as a user in the administrators group, or have a blank username
 - Can use the GUI or command line

```
!C:\> schtasks
```

TIP

The `at` command only shows tasks scheduled with `at`, while `schtasks` shows all scheduled tasks

Administrators also need a good feel for what tasks are normally scheduled to run on their systems. The Task Scheduler GUI and the `schtasks` command can be used to list scheduled tasks. The Windows cheat sheet requests that system administrators look for unexpected tasks that are scheduled to run with Administrator, SYSTEM, or blank privileges, which might be signs of an attack.

To launch the Task Scheduler GUI click Start | Programs | Accessories | System Tools | Scheduled Tasks

An older feature, the `at` command, can also be used to create tasks and display them, but it is limited. The `at` command only displays those tasks created using the `at` command itself, and not those scheduled tasks created with `schtasks`. The `schtasks` command shows a more comprehensive set of scheduled tasks, displaying those tasks created using `schtasks` itself, the Task Scheduler GUI, and the `at` command.

- Review the event log for suspicious events
 - "Event log service was stopped."
 - "Windows File Protection is not active on this system."
 - "The MS Telnet Service has started successfully."
 - Look for a large number of failed logon attempts or locked-out accounts

```
C:\> eventquery.vbs /L security
```

Works on systems before
Windows 7

```
C:\> wevtutil qe security /f:text
```

Works on Windows 7
through Windows 10

By running the Event Viewer control (`eventvwr.msc`), an administrator can look for anomalous event logs. Some of the most telling events to look for include:

- An indication that the event log service was stopped, which may have been done by an attacker to cover tracks
- A sign that the built-in Windows file integrity checker (Windows File Protection) was disabled
- A sign that the Microsoft Telnet service has been invoked
- An indication of a large number of failed logon attempts or locked-out accounts.

Some versions of Windows include the `eventquery.vbs`, which is a Visual Basic script. This tool can view all logs by running it as

```
C:\> eventquery.vbs
```

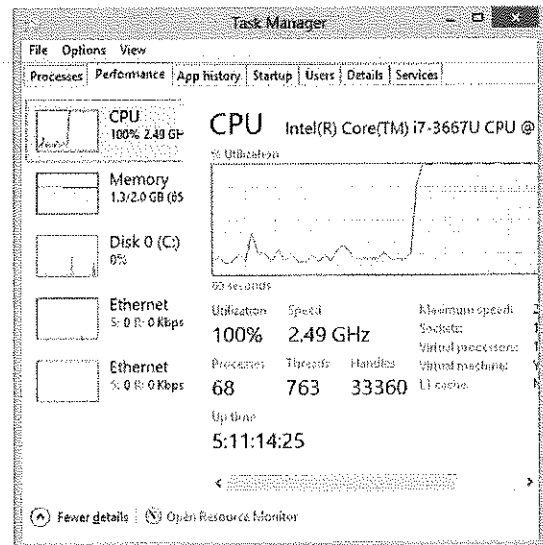
To narrow down just to security logs, the script can be run with

```
C:\> eventquery.vbs /L security
```

Unfortunately, Windows 7 through 10 do not include the `eventquery.vbs` command. Instead, on those more recent versions of Windows, you could run

```
C:\> wevtutil qe security /f:text
```

- The cheat sheets tell administrators to check the performance monitor
- . . . and look for unusual system crashes



As a final element, the cheat sheets remind the system administrator to look at the performance monitoring tool associated with the Task Manager (Task Manager | Performance Tab) to see how the system is performing. Also, it's recommended that system administrators look for unusual system crashes. Of course, neither circumstance is a guarantee of an attack, but these are items for the admins to keep in mind when analyzing their systems.

- It's not uncommon to encounter different types of encoding
 - Base64, Percent (URL) encoding, UTF-8, UTF-16 (little and big endian), etc.
- Newer Windows systems can run Bash
- To decode base64-encoded data in your Windows VM
 - Click on the Windows icon in the lower left corner
 - Type `bash` and click on Bash on Ubuntu on Windows
 - Use `echo` to pipe the encoded data to `base64` with the `--decode` option

TIP: Don't try to type the base64-encoded data, just use copy and paste

```
sec504@SEC504STUDENT:~$ echo base64-encoded-data | base64 --decode
```

Many times you will come across strange encodings in IT security. One of the more common ones is Base64.

Lucky for us, modern Windows 10 systems come with the ability to enable Bash as one of the utilities available in developer mode.

You can run this by simply selecting the Windows icon in the lower left-hand corner and typing `bash`. The user ID and password for your VM is `sec504`

Then, to decode base64 data, you simply enter the following:

```
✓ $ echo base64-encoded-data | base64 --decode
```

Remember, there are many, many different encoding, decoding, and analysis tools in Linux. Now we can use them in Windows!

You can also do this to remove Unicode spaces:

```
✓ $ echo base64-encoded-data | base64 --decode | iconv -f unicode
```

- The Sysinternals tools are *excellent* (and free!)
 - Process Explorer gives in-depth information about running processes
 - Process Monitor shows file system, registry, network, and process activity in real-time
 - TCPView shows listening ports (TCP and UDP) and maps them back to the owning process
- Center for Internet Security has templates and scoring tools

The Sysinternals suite of tools (<http://www.sysinternals.com>) are an excellent set of tools you should consider adding to your incident response and detection arsenal. A few favorites are:

- Process Explorer gives incredibly detailed information (including examining memory) for running processes.
- Process Monitor shows (and logs) file system, registry, network and process activity in real time.
- TCPView maps listening TCP and UDP ports back to the owning processes

The Center for Internet Security (<http://www.cisecurity.org>) also has hardening templates and scoring tools for Windows. They are amazingly useful starting points for hardening Windows systems.

Course Roadmap

- **Incident Handling**
- Applied Incident Handling
- Attack Trends
- Step 1: Reconnaissance
- Step 2: Scanning
- Step 3: Exploitation
 - Gaining Access
 - Web App Attacks
 - Denial of Service
- Step 4: Keeping Access
- Step 5: Covering Tracks
- Conclusions

Incident Handling

1. Overview
2. Incident Handling Process
3. Preparation
4. Identification
 - **Lab 1.1: Windows Cheat Sheet**
5. Containment
6. Eradication
7. Recovery
8. Lessons Learned
9. Enterprise-Wide IR
 - Lab 1.2: Enterprise Ident. and Analysis

Now that we've got a feel for the Windows cheat sheet, let's do some hands-on lab work with Windows. Some of these labs just look at our box in its normal state. In other lab steps, we actually create the condition we want to detect and then run the cheat sheet tip to detect it.

Lab Workbook

The lab workbook is available in printed form, and electronically as the home page for the Slingshot Linux and Windows 10 VMs

We recommend using the electronic copy of the lab for copy-paste access, but you can use either form for the exercises.

You can update the lab files by running `update-wiki` (on the Linux VM) or `update-wiki.ps1` (on the Windows VM).

You must be connected to the internet to get updates. See the *Connecting to the Network* instructions.

In this class you have two options for obtaining the directions for lab exercises: the printed workbook, or the electronic workbook.

A copy of the printed workbook is included with the rest of your course materials. Many people enjoy working from this printed resource since you can write on it, and you are allowed to bring it into the exam center with you.

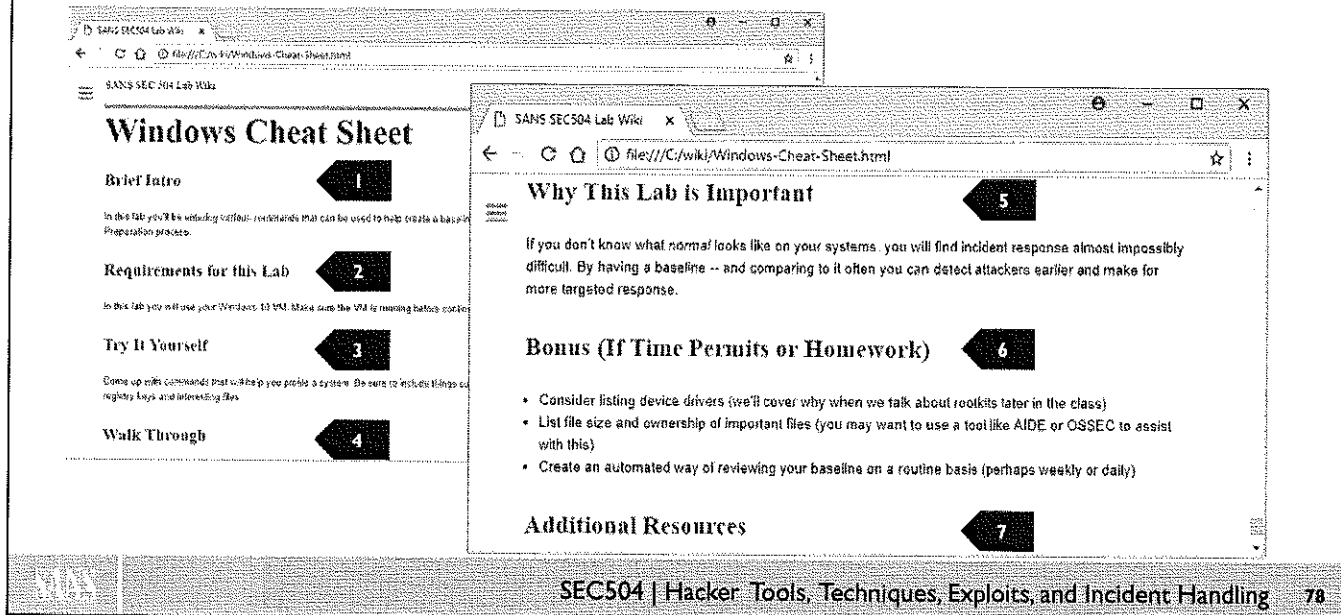
A printed workbook has limitations though. As authors, we can't update it as frequently as we would like, and it requires you to type many of the commands and attacks we'll complete manually. As an alternative, we recommend using the electronic workbook (dubbed *the wiki*), available from both the Windows 10 and Slingshot Linux VMs as the default browser home page.

In addition to copy-paste access, full-color images, and color style elements, the wiki also can be updated to collect the latest changes to lab exercises. This could be entirely new labs that we add to the course material, or it could be new techniques or typo correction to existing labs. From the Slingshot Linux VM, open a terminal and run the `update-wiki` command to download the latest wiki content. From the Windows 10 VM, open a PowerShell Command Prompt (not a standard `cmd.exe` Command Prompt) and run `update-wiki.ps1` to download the latest wiki content.

Note that your Slingshot Linux and Windows 10 VMs must be connected to the internet to download updated lab content with the `update-wiki` scripts. See the instructions documented in the *Connecting to the Network* guide (both in your printed lab workbook, and in the electronic workbook content) for instructions on configuring VMware and the individual VMs to connect to the internet.

update-wiki.ps1 (not cmd.exe)
update-wiki

Anatomy of a Lab



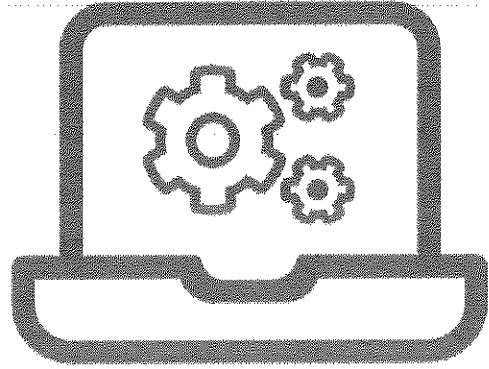
Before you start the first lab exercise, let's take a quick look at what we call the *anatomy of a lab*.

Each lab is designed to give you hands-on skills in a specific area that we believe will be immediately valuable to you when you get back to work. Although the specific steps for labs are different, they are all designed with the same core components:

1. Each lab starts with a brief introduction to what the lab is, and what you'll learn by working on the exercise in the *Brief Intro* section.
2. The lab virtual machine requirements (e.g. Windows 10, Slingshot Linux, or both) are defined in the *Requirements for this Lab* section.
3. If you're an expert, you may want to complete the lab on your own. The *Try It Yourself* section gives you just enough information to complete the exercise independently.
4. The *Walkthrough* section is the longest section, giving you step-by-step directions to complete the exercise, along with tips, notes, suggestions, illustrations, and screenshots to make the lab easy to follow and for the content to be immediately useful.
5. At the end of each lab you will find a *Why This Lab Is Important* section, reinforcing the lab learning elements in a way that can be applied to our job as defenders.
6. If you finish early, consider tackling one or more of the *Bonus* tasks. Most students don't get to finish all of these during a standard class day, but you can continue to work on these after class, or when you get back to the office.
7. Finally, the *Additional Resources* section gives you pointers and links to additional sources of information. This could be recommendations for other related SANS courses, but also includes free resources such as online learning resources, articles, and books that we recommend.

LAB 1.1

Please work on the lab exercise
Windows Cheat Sheet



This page intentionally left blank.

(P5)

1. `kernel - a20 > 4000000` *low a. via mif*

2. `kernel - a20 > 4000000` *second. 700*

3. `kernel - a20 > 4000000` *get process name via mif*

4. `kernel - a20 > 4000000` *no process (193) -> no flag*

5. `kernel - a20 > 4000000` *no process when process id is 0*

6. `kernel - a20 > 4000000` *available flag "imprisoned" to process*

7. `kernel - a20 > 4000000` *no process*

8. `kernel - a20 > 4000000` *no process*

9. `kernel - a20 > 4000000` *no process*

10. `kernel - a20 > 4000000` *no process*

- Determine whether an event is actually an incident
 - Check for simple mistakes by users, admins, or others
 - Assess the evidence in detail
 - Ask yourself, "what other possibilities are there?"
- We'll spend the rest of our class discussing the methodology used by attackers, to help familiarize you with items to watch for
- Maintain situational awareness, reporting to chief

Efficient handling of errors is part of the process

A large number of "incidents" turn out to be false positives, or perhaps a nicer term is events. One of the most important functions of an incident handler is to continue to assess the data to see if it indicates this could be something other than an incident. The process should be optimized to encourage reporting and then deal with false positives gracefully and efficiently. Keep in mind that this can be great training, especially for the second, or junior, person sent to the scene. A wise handler, who can already see what the situation is, can use the opportunity to help educate the less experienced handler.

One of the most important responsibilities of a senior incident handler is to maintain situational awareness. What is the effect of the vulnerability, can it be remotely exploited, is an exploit available, or is this a zero-day attack (an attack that was previously unannounced)? These types of questions help the handler come to a reasonable initial assessment.

- When looking at the situation, you need to determine how much damage could be caused:
 - How widely deployed is the affected platform or application?
 - What is the effect of vulnerability exploitation if a vulnerability is present?
 - What is the value of the systems impacted so far? What is the value of the data on those systems?
 - Can the vulnerability be exploited remotely (via a network connection)?
 - Is a public exploit available? Was one recently released?

Purists may choose to use a point system, but the main idea is that by asking the right questions, you can come to a reasonable initial assessment.

How widely deployed is the affected platform or application? Obviously, the more widely deployed, the greater the risk. A custom application means the containment is simple. A vulnerability affecting multiple Windows platforms is really scary.

In terms of the effect, is it denial-of-service, reconnaissance, large-scale reconnaissance, information compromise, user compromise, or privileged user (root or administrator) compromise?

What is the value of the systems impacted so far? What is the value of the data on those systems? Obviously, high-value victim machines or systems storing sensitive data represent a much more significant risk.

Can the vulnerability be exploited remotely (via a network connection)? If not, again containment is reasonable; if by the internal LAN, again containment is not too tough; if by the internet, you may need to configure a firewall or router rule fast.

Is a public exploit available for the vulnerability? One source to check is the Common Vulnerabilities and Exposures webpage at cve.mitre.org, also, bugtraq and isc.sans.edu. If there is no mention of a public exploit, you may be dealing with a zero-day (not previously announced) vulnerability and exploit. This could be evidence of a high-end attacker and raises the stakes.

Technique Matrix

- Main page
- Help
- Contribute
- References
- Using the API
- Tactics
 - Initial Access
 - Execution
 - Persistence
 - Privilege Escalation
 - Defense Evasion
 - Credential Access
 - Discovery
 - Lateral Movement
 - Collection
 - Exfiltration
 - Command and Control
- Techniques
 - Technique Matrix
 - All Techniques
 - Windows
 - Linux
 - macOS
- Groups
 - All Groups

Initial Access	Execution	Persistence	Privilege Escalation	Defense Evasion	Credential Access	Discovery	Lateral Movement	Collection	Exfiltration	Command and Control
Drive-by Compromise	AppleScript	bash_profile and bashrc	Access Token Manipulation	Access Token Manipulation	Account Manipulation	Account Discovery	AppleScript	Audio Capture	Automated Exfiltration	Commonly Used Port
Exploit Public-Facing Application	CMSTP	Accessibility Features	Accessibility Features	BITS Jobs	Bash History	Application Window Discovery	Application Deployment Software	Automated Collection	Data Compressed	Communication Through Removable Media
Hardware Additions	Command-Line Interface	AppCert DLLs	AppCert DLLs	Binary Padding	Brute Force	Browser Bookmark Discovery	Distributed Component Object Model	Clipboard Data	Data Encrypted	Connection Proxy
Replication Through Removable Media	Control Panel Items	AppInit DLLs	AppInit DLLs	Bypass User Account Control	Credential Dumping	File and Directory Discovery	Exploitation of Remote Services	Data Staged	Data Transfer Size Limits	Custom Command and Control Protocol
Spearphishing Attachment	Dynamic Data Exchange	Application Shimming	Application Shimming	CMSTP	Credentials in Files	Network Service Scanning	Logon Scripts	Data from Information Repositories	Exfiltration Over Alternative Protocol	Custom Cryptographic Protocol
Spearphishing Execution	Authentication	Bypass User Account	Bypass User Account	Clear Command History	Credentials in Registry	Network Share Discovery	Pass the Hash	Data from Local System	Exfiltration Over Command and	Data Encoding

One of the most effective tools to assist in incident response released in the last few years is the MITRE ATT&CK matrix. This is a collection of techniques used by actual malicious attack groups over the past few years. It is a great place to start looking for gaps in your preventative and detective capabilities.

One of the nice features of the matrix is that it breaks the techniques up into the various phases attackers go through. Further, for many of the different techniques, it also has example code to test if your organization can detect the various classes of attacks.

It can be found here:

https://attack.mitre.org/wiki/Technique_Matrix

If you want more scripts to practice with, check out the Red Canary Atomic Red Team:

<https://redcanary.com/blog/atomic-red-team-testing/>

- Ask yourself:
 - What level of skill and prerequisites are required by an attacker to exploit the vulnerability?
 - Is the vulnerability present in a default configuration?
 - Is a fix available for the vulnerability?
 - Do other factors exist that reduce or increase the vulnerability's risk or potential impact, such as the possibility it is a worm?
- Lenny Zeltser has prepared an Initial Security Incident Questionnaire for Responders
 - Series of cheat sheets to help assess an incident

What level of skill and prerequisites are required by an attacker to exploit the vulnerability? Also, does he have the skills to do anything to the system after he breaks in?

Is the vulnerability present in a default configuration? All too often, the answer to this is yes. The chances you are running default configurations are high, so this raises the risk level.

Is a fix available for the vulnerability? If so, this will be a major part of containment.

Do other factors exist that reduce or increase the vulnerability's risk or potential impact, such as the possibility that it is a worm? This can be the reason you drop your internet connection for your entire organization. In the case of Nimda, UUNET estimates it reached saturation in about two hours across the entire internet. For SQL Slammer, a massive outbreak occurred in 15 minutes.

Lenny Zeltser, SANS Instructor, has written a cheat sheet with these and more questions on it to ask while responding to security incidents. Lenny's cheat sheet is available at <https://zeltser.com/security-incident-questionnaire-cheat-sheet/>.

- Be careful to maintain a provable chain of custody
 - Do NOT delete ANY files until the case is closed out, and even then, if you have storage space, save them for a document retention time frame approved by your legal team
 - Identify every piece of evidence in your notebook
 - Control access to evidence
- Each piece of evidence must be under the control of one identified person at all times
 - Include a lined page with the evidence to record all hand-offs: who and when
 - Record when you lock it up in storage
- When turning over evidence to law enforcement, have them sign for it

A recommended practice is to keep everything together for a particular case. Before you touch anything, if there is reason to suspect this could go to court, it is wise to fill out attestation forms to the tune of "I, John Doe, 1 April 2013, am in room 23, 1416 Able St, and am looking at a Dell server, serial number XXX. This computer is suspected of being involved in criminal activity. At 21:45, we are disconnecting the network cord. We have done nothing else with this machine." And on it goes. To the extent possible, account for every action you take or command you type. Cameras, if allowed, can be useful. Often, all the evidence will fit in a gallon-size Ziploc bag, and you can read the evidence listing through the plastic. Keep it in a locked container that very few people can access.

To help maintain an inventory of all evidence you exchange with law enforcement and improve the chain of custody you maintain, have law enforcement officials sign for all evidence you hand over; make sure all the evidence is accounted for, and make sure the list includes some description of the "value" of the evidence. Also, give them a copy of your evidence, not the originals, unless they specifically require and ask for the originals. Most of the time, copies will suffice.

Course Roadmap

- **Incident Handling**
- Applied Incident Handling
- Attack Trends
- Step 1: Reconnaissance
- Step 2: Scanning
- Step 3: Exploitation
 - Gaining Access
 - Web App Attacks
 - Denial of Service
- Step 4: Keeping Access
- Step 5: Covering Tracks
- Conclusions

Incident Handling

1. Overview
2. Incident Handling Process
3. Preparation
4. Identification
 - Lab 1.1: Windows Cheat Sheet
5. **Containment**
6. Eradication
7. Recovery
8. Lessons Learned
9. Enterprise-Wide IR
 - Lab 1.2: Enterprise Ident. and Analysis

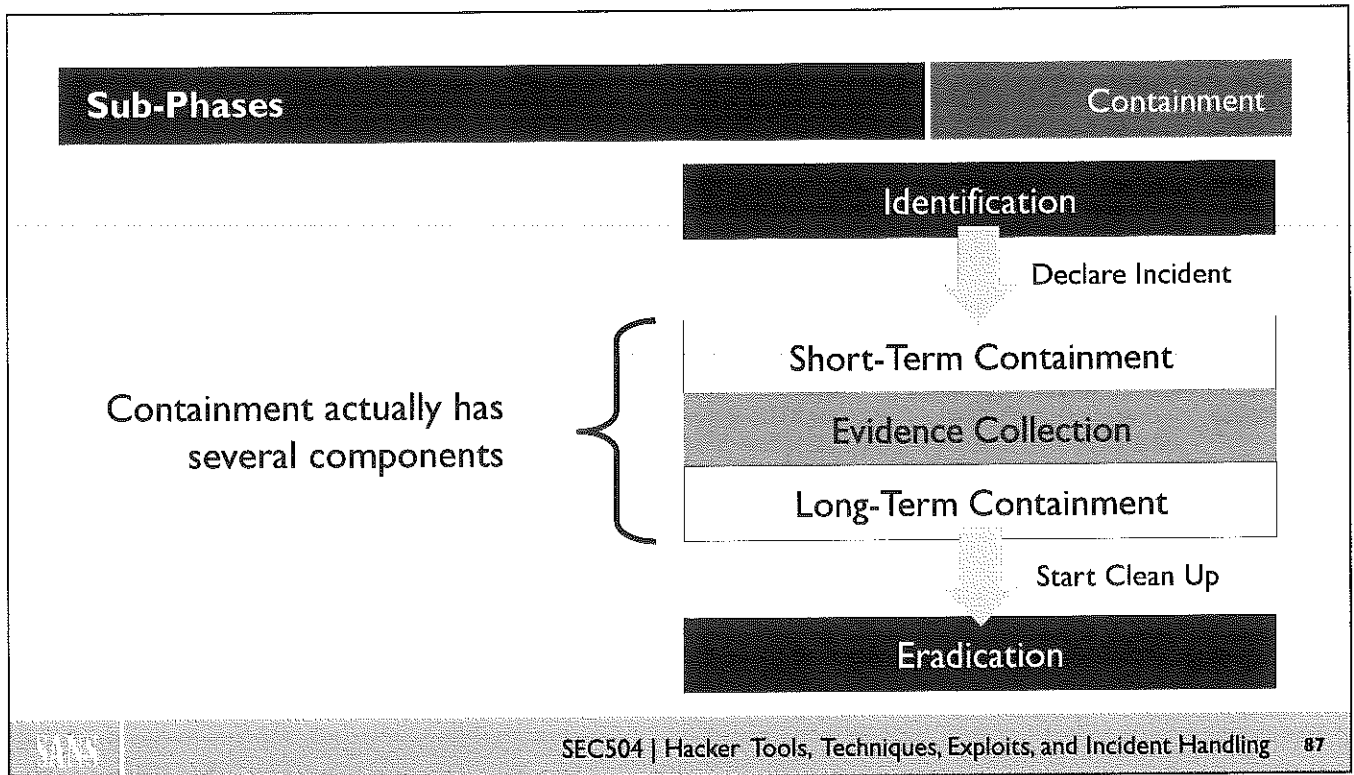
Let's review where we are in the process. We have prepared to some extent. We have identified a possible incident. We have gone on scene and we have introduced chain of custody.

The goal of containment is to keep the problem from getting worse. Before we fire, we should take the time to aim! Try to do a decent survey and review of the situation before altering the system.

When an incident handler first arrives on location, there is a chance that the system is pristine in terms of evidence and information. As soon as the handler starts to recover the system, there is a point in which the evidence starts to become contaminated. If at all possible, the file system image creation should come before this point so there is a copy of the unaltered system.

- The goal of the Containment phase is to stop the bleeding
 - Prevent the attacker from getting any deeper into the impacted systems or spreading to other systems
- We discuss
 - The sub-phases of containment
 - Methods for short-term containment
 - Evidence collection
 - Methods for long-term containment

For Containment, we want to stop the bleeding. How can we arrest the attacker in his/her tracks before he/she causes more damage? That's what Containment is all about!



Containment includes three sub-phases: short-term containment just to stop the damage, followed by collecting evidence, followed by long-term containment to make sure the bad guy is denied access. Let's look at each of these in more detail.

- Deploy a small on-site team to survey the situation
 - Typically, these will be the same personnel as the Identification team
 - Secure the area
 - If possible, use preprinted survey forms provided at www.sans.org/score/incidentforms
 - Review the information that was provided to you from the Identification phase

If you are dealing with a suspected crime, still or digital cameras can be used to record the scene, where you were, where things were located, and who was in the room.

A recommended practice during a survey is to trace down all the wires in a room; be especially alert for impromptu networks and anything that is telephony or wireless.

The incident actually started before you arrived; don't treat time zero as your arrival on the scene! What you see when you arrive may not be what the original user saw; things could change. For example, the RingZero trojan would start on the desktop but then remove itself from the desktop. If the user says he saw an icon on the desktop, and you looked and didn't see it and decided he was wrong, this doesn't make you a world-class handler, now does it? Take the time to review the evidence that was created before you arrived on the scene. This includes what everyone saw, heard, and did. It also includes whatever documentation was made.

- Once an incident has been declared, document various characteristics
- FIRST Case Classification is a good starting point
 - Incident Category
 - Denial of Service, Compromised Information, Compromised Asset, Internal Hacking, External Hacking, Malware, Policy Violations
 - Criticality (affects response time)
 - Level 1: Business critical systems, 60-minute response time
 - Level 2: Non-business critical systems, 4-hour response time
 - Level 3: Possible incident, non-business critical systems, 48-hour response time
 - Sensitivity (affects who should be notified)
 - Level 1: Extremely sensitive (CSIRT, management)
 - Level 2: Sensitive (CSIRT, management, system owners and operators)
 - Level 3: Less sensitive (CSIRT, affected employees)

In moving to the Containment phase, we have declared an incident. It is important to document various characteristics of the incident early on in our Containment phase. The FIRST organization distributes an incident Case Classification document that recommends characterizing an incident based on three areas: its general category, the criticality of impacted systems and data, and the sensitivity with which information about the case itself should be treated.

You can find the document at https://www.first.org/resources/guides/csirt_case_classification.html.

From a category perspective, most incidents fall into one or more areas of the list shown above. It is important to note that a single incident may be in multiple categories, such as Compromised Information, Malware, External Hacking, and Email, all in the same incident. You may want to add new categories here as attacks evolve.

The criticality rating of an incident will help determine how quickly you'll need to assign a team and deploy to handle the situation. For highly critical incidents, you may want to establish a baseline of response time at 60 minutes, or perhaps even less for some organizations with critical computing needs. Customize these time frames based on the type of information your organization handles and the criticality of its computing base.

The sensitivity metric here determines the types of personnel with whom information about the incident can be shared. For a case that is extremely sensitive, we may only want to share information with the incident response team and management. For sensitive cases, we may add in the system owners and the operations teams. For less sensitive cases, we may inform more employees, such as in the case of an isolated virus infection.

- Identify a senior management sponsor for your team
- CISO, CIO, legal counsel, etc.
- When you declare an incident, notify your management sponsor and get help to assist in the incident handling process
 - Notification may be a mere email
 - For more serious incidents, a phone call or visit
- Get a copy of the corporate phone book
- Assign a minimum of two people to each incident: a primary handler and a helper
 - Make sure both take notes of their actions and observations

If a person sitting next to you suddenly fell ill with a heart attack, what do you do first? Hopefully, you answered that you would alert the emergency medical system! Even though being treated in the first minutes from when your heart stops is your best chance of recovery, the time spent asking for help is not wasted.

Your incident handling team should have a senior member of management as its sponsor. This manager can help to clear out obstacles when you are under fire. To do that, you should strive to find a sympathetic senior manager, such as a Chief Information Security Officer (CISO), Chief Information Officer (CIO), senior legal counsel, or another related position that makes the most sense in your organizational structure.

Always let this management sponsor know that you are in incident mode, either via email or, for a more serious incident, with a phone call or visit. If you do not have a formal incident team reporting structure, advise your manager and the security point of contact at a minimum. I cannot count the number of times that, at ten or so minutes into the incident, I realized I was over my head and needed reinforcements or specialists. It takes time to mobilize people. As soon as the incident is identified, you may wish to put them on alert.

One of the things that I am learning is that I can't do it all. If you are the primary handler on site, it is really a challenge to take notes, secure the area, and so forth. This is not a lone ranger sport. Realize you will probably need help and arrange for it in advance. Assign a minimum of two people to each incident: a primary handler and a helper. Have them both take notes independently of the other. Sure, there might be some conflicts in their notes. However, I'd rather take that small chance while avoiding the chance for crucial evidence slipping through the cracks!

- Notify your local or organizational incident handling team
- Notify your manager and security officer
- Remember vertical and horizontal reporting
 - Inform management (of course)
 - Inform impacted business unit
- Create entry in incident tracking system
 - CyberSponse is a commercial IR tracking system
 - There's the free RTIR Incident Response Tracking tool

There is a dynamic tension in many organizations between security and line management. If this is an issue in your organization, it can be a good idea to make sure both groups are kept in the loop throughout an incident. Users should be aware that when handlers are under fire, they may drop important information. In a large-scale attack, a handler might see a message, think that he will get right to it, and have 60 other things come up. For this reason, it is good practice to encourage users to demand a reply.

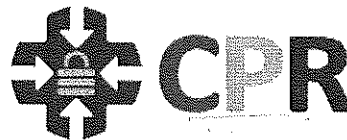
This is also a good reason why everything reported to a help desk or incident handling center should be given a trouble ticket; it helps prevent loss of critical information. From a user's perspective, the best examples of horizontal reporting might include the system administrator, help desk, and the incident reporting structure. Talk about good news, bad news! If he reports to all three (and many will), then three different trouble tickets get created, and there is a significant chance that actions will be taken that could damage the forensics evidence. On the other hand, additional eyes on the problem can be a good thing. The best solution is for the incident handling reporting structure to communicate closely with the help desk and system admin shops.

Some commercial Security Information Management (SIM) and Security Event Management (SEM) products include the ability to assign an incident number around a group of events to track them together. Some go so far as to include a collaborative environment for incident handlers to conduct analysis and share conclusions across a distributed team. One such product is CyberSponse. It is a commercial-grade IR tracking system that greatly simplifies the sharing and centralized collection of IR data across your team. It can be found at <http://cybersponse.com>.

Alternatively, the Request Tracker for Incident Response (RTIR) tool is a ticketing system targeted to incident handling tracking. Its focus is on helping incident handlers stay organized when conducting their work by providing an incident tracking number for each overall incident, plus tracking numbers for individual conversations.

You can find RTIR at <http://www.bestpractical.com/rtir/>

- Another option is CyberCPR
 - Brainchild of fellow SANS Instructor Steve Armstrong
- Web app that tracks incidents, systems, and evidence
 - Enforces need-to-know on incidents
 - All files are hashed and encrypted upon upload
 - Tracks user tasks and activity
 - Tracks attacker campaigns
 - Automates key analysis
 - Secure real-time OOB chat
- While a commercial tool, 1–3 users will be free



Another option is CyberCPR, the brainchild of fellow SANS Instructor Steve Armstrong and coded by several SANS Community Instructors and SEC504/560 and FOR408/508/610 alumni, it's designed around the DFIR user. Deployed as a hardened web application app (locally or with a cloud provider), it was designed to enforce need-to-know at the incident level, allowing the existence of sensitive investigations to be suppressed from those not involved.

It supports the incident handler by providing a central repository for evidence found and protects legal admissibility by encrypting and hashing (MD5 and SHA256) all data uploaded. This provides a secure alternative to emailing evidence and updates around the network. When system admins are provided accounts, they can upload large logs and files directly into the application, preventing what is known as evidence dispersal (where investigation evidence is scattered around various systems).

All items can have formal notes and comments added to them by that incident's users, and these notes are indexed and searchable. Incident tasks allow the Incident Manager to track actions they have delegated to other users. Together, these facilitate the secure coordination of evidence collection (the what), the recording of the reasons for that collection (the why), and the required timeline of analysis (the by when). Once analysis is complete, logs and artifacts can be uploaded for others' review, notes, and comment upon the analyst's actions.

For out-of-band (OOB) communications, the tool includes both shoutbox type chat and one-to-one messaging capabilities, allowing things like system admin or access passwords to be transmitted securely.

Best of all, as strong community supporters, Steve and his team have committed that the 1–3 user version will always be free. This allows SMEs to benefit from the tool's commercially funded development and new features without breaking the bank (<https://www.cybercpr.com>).

- **Keep a low profile**
 - Avoid looking for the intruder with obvious methods from the compromised machine (ping, traceroute, nslookup)
 - Don't tip your hand to the attacker
 - Maintain standard procedures
- Local handlers should keep making reports to the command center as they gather and analyze evidence

Rookie incident handlers can be spotted a mile away with a network logging system. They find an attack that appears to originate from an IP address. Then they ping the address, perform an nslookup, and traceroute to it. (Sometimes, they even telnet to it!) You know, that might just tip off the attackers.

Some handlers feel that we should maintain normal procedures; if system backups normally took place at 02:45, then when 02:45 comes around, if at all possible, do what you usually do. This may matter in some cases of high value, but in general, the attacker couldn't care less.

- Try to prevent the attacker from causing more damage
 - While minimizing the changes to data on the system(s)
 - We want untainted evidence and can only get that through our image creation process
- Some possible short-term containment actions
 - Disconnect network cable
 - Pull the power cable: Loses volatile memory and may damage drive
 - Using network-management tools, isolate the switch port so the system cannot receive or send data (or place on an "infected VLAN")
 - Apply filters to routers and/or firewalls
 - Change a name in DNS to point to a different IP address
- You could also use WordWebBugs to track the attacker
 - Documents that "call back" to identify where sensitive data is located
 - They are built into the Active Defense Harbinger Distribution

For short-term containment, we just want to stop the attacker's progress without making any changes to the impacted system itself. We want to keep the target machine's drive image intact until we can back it up. Therefore, this short-term containment typically involves disconnecting network access and/or power.

If you have the ability to control your switch infrastructure, you may want to consider isolating the switch port to which the impacted machine is connected, or even placing that system on an isolated, infected VLAN so that you can still communicate with it, but it cannot infect other machines.

Another option is based on the fact that most attackers target systems based on their IP address (such as 198.167.22.13) and not their domain name (www.[yourdomainname].com). If this is the case, another option for short-term containment involves altering DNS so that the domain name(s) for the impacted system(s) points to a different IP address, perhaps one where you have a newly installed, secured machine offering up the desired production service. Once that new DNS address record has propagated, your users relying on the domain name will be accessing the new system at the new IP address. The attacker, if he or she is using an IP address to hit the target, will continue to go to the old IP address. You could place a honeypot at that address, simply null route all traffic going to the address, or just leave it as a completely non-responsive, unused address.

You could also use WordWebBugs to track the attacker. These documents call back (preferably to a non-attributable system) so you can identify where your sensitive data is.

<http://sourceforge.net/projects/adhd/>

- If short-term containment disables the system (such as removing it from the network and/or denying legitimate users access to the machine)
 - Make sure you advise someone in the business unit responsible for the system
 - Usually the information or application owner
- Advise them in writing with a signed memo or at least an email that gets acknowledged
- They may disagree with your advice to drop the system
- In disagreements, the business unit almost always wins!

Containment (both short- and long-term) might stop the system from performing various business actions. Therefore, make sure you get approval before taking action that will impact business. Call the business unit teams before dropping a system.

- For external attacks, coordinate closely with your internet service provider
 - It may be able to assist you in identification, containment, and recovery
 - Especially for large packet floods, bot-nets, worms, and virulent spam
 - The information you provide may save someone else a lot of pain
 - We need to work together as a community to foil widespread attacks
- Also, you may need to rely on someone else's ISP to get a bot-infected system taken offline

Most organizations realize they need to coordinate with their ISPs to bring some incidents under control, especially in large packet floods, wide-scale botnets, and their communications channel, worms, and virulent spam. Many ISPs are experienced at network-based exploit and denial-of-service types of incidents and are efficient at handling them. ISPs often keep system and even network logs, at least for a reasonable time frame. Also, they may be able to spare other folks that get their service from the trouble that you had to go through.

By the way, you should be aware of one group that is extremely experienced in handling incidents; these are the computer and network staff of colleges and universities. As your organization is paying for employees' higher education and training, it can be a good idea to make contact with the computing staff at your local educational institution. If you build a relationship with them, they can really help you when you are under fire.

- Make forensics images of affected system(s) as soon as is practical
 - This initial image will be used as a source for forensics analysis
 - Grab an image of memory as well as the file system
 - Don't do graceful shutdown—you'll lose valuable data!
- Volatility Framework and Rekal can capture and analyze memory
- Use blank media
 - Old media often contain remnants
 - Newly purchased media may have some data on it, so beware
- If possible, make a bit-by-bit image to get all file system data
- Not all incidents will allow you to do a full backup and analysis
 - Time-sensitive incidents may require advanced network, domain, and live forensics
- Many forensics tools will automatically calculate hash of collected evidence

Failure to take complete notes is the most common error that incident handlers make.

Failure to make a good working forensics image is one of the most common errors. This is compounded because the incident handler is often called to work on a system that hasn't been backed up in a long time, sometimes years. Of course, the data is irreplaceable and mission essential. Many systems are being purchased today with multi-hundred gigabyte hard drives and no tape or other backup methods. If you do not make a good forensics of the system before you start doing detailed analysis, you drastically reduce the chance of that system information being usable in court. The other attorney could claim that you modified the system. If you must do things on the system before backing it up, and sometimes this is necessary, try to log each command you type and the system's response.

This initial forensics image in the Containment phase serves as a source for forensics analysis. Make sure to get a copy of both the memory and the file system. You will not be able to make forensics images for all systems in all incidents. Some incidents may require you to rely on network domain and live forensics.

The ideal image, however, is the binary, bit-by-bit image; this gets everything on the disk, including deleted and fragmentary files. One of the most popular tools for creating binary images on UNIX and Windows machines is `dd`. I use it all the time, and it is a major component of the SANS Forensics Track (SANS Security 508) for gathering system images for analysis. It is built into many UNIX and Linux distributions and is available in many incarnations for free for Windows. My favorite Windows version is available at <http://www.gmgsystemsinc.com/fau/>. Google has released a powerful tool called `Rekal` for capturing and analyzing memory on Windows machines. Volatility Systems released a tool called the `Volatility Framework` that likewise can be used to capture and analyze memory dumps.

One nice feature of using tools designed for digital forensics is they calculate cryptographic hashes for you automatically. This is useful for demonstrating that the evidence has not changed since you collected it.

- Consider using a write blocker if practical
 - Hardware write blockers sit between computer and hard disk to intercept write requests
 - Software write blockers run on the host directly
 - Usually can't use a write blocker on live systems
- Consider getting a hardware drive duplicator if you frequently image drives
 - Much faster than using a laptop with two external hard disks
- Drive storing evidence usually needs to be 10% bigger than the evidence
 - File system overhead, file format overhead

When collecting evidence, you may want to consider using a write blocker. This is a device that is designed to prevent processes from writing to disk (both internal and external). A hardware write blocker sits between your computer and the hard drive's controller to intercept (and block) write requests. A software write blocker achieves its goals by running on the host and hooking into write requests.

It's not always practical to use a write blocker, so don't consider this a requirement. If you are collecting evidence from a live system that cannot be shut down (e.g. business critical server) then you usually can't install a write blocker because you'd end up negatively impacting business.

If you image hard drives on a regular basis, you may want to consider investing in a drive duplicator. They're not cheap, but they are much faster than using a laptop with two usb cables connected to two external hard disks. Drive duplicators also usually calculate cryptographic hashes on the fly.

The drive you store your evidence on usually needs to be at least 10% larger than the original drive. This is to account for the file system overhead when imaging a drive and saving it as a file. Also, various different evidence file formats can include their own metadata.

Handwritten notes:
3 10
412
2019-09-01
2019-09-01

- Acquire logs and other sources of information How far did the attacker get?
- Review logs from neighboring systems
- Make a recommendation for longer-term containment
 - Document recommendation in signed memo
 - Ultimately it's a business decision, informed by the incident handler's input

Make sure to watch out for trust relationships concerning the affected system. The most important trust relationship is often the desktop of the system administrator to the system and to other systems they administer. Look over the logs from nearby systems and trusted machines. Try to get a feel for how far the bad guy may have penetrated.

This is that critical moment: Do you down the box? Is this a contain and clean or a watch and learn? Most people contain at this point; however, sometimes the affected system(s) are crucial to your organization's operations. While you probably wouldn't want to keep running them at all costs, it makes sense to accept a higher risk in such circumstances.

Remember, the ultimate decision for downing the machine is a business call. Make a recommendation, which should be documented in a signed memo to the business owner of the machine. Realize that you won't always get your way on such decisions.

- Once we've got our backup for forensics analysis, we can start making changes to the system
- Therefore, we can implement longer-term containment strategies
- Ideal: If the system can be kept offline, move to the Eradication phase
 - Get rid of the attacker's stuff
- Less than ideal, but sometimes necessary: If the system must be kept in production, perform long-term containment actions

Long-term containment might change the drive image of the impacted system. That's OK because we've already gotten our backup for forensics analysis.

Ideally, we can keep the compromised machine offline for an extended period of time while we fully eradicate the attacker's artifacts, possibly by rebuilding the system or restoring it from a backup. In such cases, our long-term containment steps actually involve detailed eradication. In other words, after creating forensics images, we move to the Eradication phase when extended downtime is acceptable. However, for critical production systems, extended downtime is usually not permissible. In such cases, the incident handlers must stay in the Containment phase, performing long-term containment actions on a live system to keep it running.

- Numerous potential actions, including
 - Patch the system, and possibly neighboring systems
 - Insert Intrusion Prevention System (IPS) or in-line Snort/Suricata
 - Null routing
 - Change passwords
 - Alter trust relationships
 - Apply firewall and router filter rules
 - Remove accounts used by attacker, shutting down their backdoor processes
- Remember, you still need to do eradication
- The idea for long-term containment is to apply a temporary Band-Aid to stay in production while you are building a clean system during eradication

There are several long-term containment activities, but the most likely, by far, is just patching the system if the attacker compromised it by exploiting a vulnerability. Handlers should also patch other nearby or similar systems to ensure that they do not get compromised. If patching is impractical over the short term, you may want to consider deploying an in-line intrusion prevention system, such as a commercial solution or even in-line Snort or Suricata, which can block some attacker activity. Other long-term containment options that allow us to keep the system in production include null routing, in which we configure routers to drop packets associated with a given source or destination IP address used in the attack. You may need to change passwords for accounts to introduce a discontinuity in the attacker's access. Likewise, trust relationships between machines may need to be altered and/or broken so that an attacker with access to one environment cannot simply access another set of systems without reauthenticating. Firewall rules and router access control lists may need to be tightened to prevent deeper attacks as well. Of course, you may need to remove accounts created by the bad guy and kill any processes that offer the attacker backdoor access to the machine.

Don't think you're done with the incident handling process, however, just because you applied a patch! You've still got the Eradication, Recovery, and Lessons Learned phases.

- Keep system owners and administrators briefed on progress
- Don't play the *blame game*
 - Never allow fault to be an issue during incident handling
 - Assigning fault now closes down important avenues of investigation
 - Sometimes, as you learn more, assumptions change
 - If fault absolutely must be assigned, do that during the Lessons Learned phase (Phase 6), not the Containment phase

One of the goals of a top-notch incident handler is to be as low impact as possible on the folks who own the computer with the incident. In some sense, we are guests or servants. We will be with the system for a short time, but it is not ours. We don't depend on it or administer it. It is important to keep the system administrator and the owner up to date on the situation. If you do not, they will try to get information from the command center and they may decide that they do not trust you. When this happens, it is harder for everyone.

Very often, assumptions change as more information becomes available during an incident. Early assumptions are often proved wrong. If you were to blame an individual and the facts later showed that person was not at fault, your credibility would be lost, at least in that part of the organization. There is more than just credibility at work here. Even though you are there to help, in some way, you are identified with your organization's security forces and you have access to their secrets. It is really important not to foster resentment; you need their trust and support to do your job.

Course Roadmap

- **Incident Handling**
- Applied Incident Handling
- Attack Trends
- Step 1: Reconnaissance
- Step 2: Scanning
- Step 3: Exploitation
 - Gaining Access
 - Web App Attacks
 - Denial of Service
- Step 4: Keeping Access
- Step 5: Covering Tracks
- Conclusions

Incident Handling

1. Overview
2. Incident Handling Process
3. Preparation
4. Identification
 - Lab 1.1: Windows Cheat Sheet
5. Containment
6. **Eradication**
7. Recovery
8. Lessons Learned
9. Enterprise-Wide IR
 - Lab 1.2: Enterprise Ident. and Analysis

Now we turn our attention to what is probably the hardest problem in incident handling—the complete and safe removal of any malicious code and other artifacts left by the attacker on the system, such as pirated software, pornography, and other illicit data. Although malicious code is not an issue in many incidents such as fires (or a denial-of-service attack), this is one of the hardest problems a handler faces. This is why the GIAC Certified Incident Handler (GCIH) invests so much time covering malicious code.

- With the bleeding stopped, the goal of the Eradication phase is to get rid of the attacker's artifacts on the machine
- Determine cause and symptoms of the incident
 - Use information gathered during identification and containment
 - Try to isolate the attack and determine how it was executed

Now, with the bleeding stopped, the goal of the Eradication phase is to get rid of the attacker's artifacts on the machine, including accounts, malicious code, pirated software, porn, or anything else the bad guy left on the machine.

Reformatting and reinstalling the operating system from scratch may be considered a valuable shortcut in the handling process. Although it is certainly true that total destruction of the contents of the disk takes care of any malevolent code, the opportunity for reinfection via the same channel after you reload the operating system still exists. There are many cases where handlers have taken systems down and reloaded the operating system only to have the box compromised again a few days later. The best course of action is to determine what the cause of the incident was, to find the vector of infection, and take action to prevent this from happening again.

The network system and forensic skills needed to do this are difficult to develop and, while we continue to try to make top-quality training available, there is no substitute for actual experience.

- Locate the most recent clean backup
 - Search for a recent backup before an intrusion
- In case of a rootkit-style attack (which modifies the operating system itself)
 - Wipe the drive (zeroing it out)
 - Reformat the drive
 - Rebuild the system from the original install media
 - Apply patches
 - Without a complete reformat, the attacker's residual data, tools, and access may linger

Backups simply do not happen as often as they should. They also are not tested in some cases. If an affected system has a recent clean backup and you can identify when the compromise occurred, recovery is a matter of wiping the drive (zeroing it out), reformatting the drive, reinstalling the operating system, reloading the data from backup, adding any lost data, and fixing the vulnerability that caused the problem in the first place.

If no backup is available, loading data by hand from a USB drive to a reinstalled OS is a tricky and expensive process. If your organization has a central incident handling team, you may want to establish a policy that this is the responsibility of the affected group or department.

In any case, here is the bottom line of eradication: There was an incident, you show up, you save the day, you are a hero! This is great. The problem comes back six hours later, and you are a goat! This is not great. Err on the side of caution; make sure all parties know that if the safest course is to down the system and scrub it, they are making a risky choice. Don't keep doing business exactly like you were before being compromised.

- Remove malware inserted by the attacker
 - E.g. viruses, backdoors, rootkits, etc.
- If a rootkit (user or kernel) was used, you should rebuild
 - Format the drive
 - Operating system (and PATCHES!)
 - Applications (and PATCHES!)
 - Data (the hardest one, possibly tainted backup)
- Encourage the impacted business unit to rebuild
 - Reviewed by the computer security team (including incident handlers)
- There may be times where the attacker did not use malware
 - Use legitimate services such as SSH and Remote Desktop

Viruses are fairly interesting. They are easy to deal with after the antivirus companies have analyzed them. You just let the software clean up the problem, and it does a great job. Dealing with viruses that don't yet have antivirus signatures is a much harder problem.

If the attackers change the operating system itself by installing a rootkit, you should rebuild from the original install media and install patches. Make sure you patch the system thoroughly, or the attacker will return.

Also, encourage the impacted business unit to do the rebuild under your supervision. That way, you can make sure they understand the build process. Furthermore, they can verify that the system is functioning properly. Finally, you can verify that all patches are installed when they rebuild the machine.

It is also common for an attacker to use common and legitimate services, such as SSH and Remote Desktop, to persist and spread. It is incredibly important to monitor the logs of these services for irregularities, like strange source IP addresses and multiple concurrent logins from a single user ID.

- Implement appropriate protection techniques
 - Applying firewall and/or router filters
 - Moving the system to a new name/IP address
 - Null routing particular IP addresses
 - Changing DNS names
 - Applying patches and hardening the system

Once your system is hacked, the word gets out and every pea-brained hacker on the planet lines up to take another shot at you. It is not enough to simply recover the system; the security of the affected system(s) needs to be upgraded. If it is a production system, you may hear arguments that the organization cannot risk modifying a production system. This is a valid and important argument. The flipside is that if the system was compromised, it must have some vulnerability. If we do not remove the vulnerability, the system may become compromised again.

The simple trick of changing the name and IP address of the system can solve many problems.

- Perform vulnerability analysis
 - Perform system vulnerability analysis
 - Perform network vulnerability analysis
 - Search for related vulnerabilities
 - If possible, scan your entire network for interesting ports with a port scanner, such as Nmap
 - A vulnerability scanner, such as Tenable's Nessus, OpenVAS, Rapid7 NeXpose, and Qualys can also be a big help
- Remember that attackers often use the same exploit and backdoors on multiple machines
 - Look for them throughout your environment

Vulnerability scanners, such as Tenable's Nessus, OpenVAS, Rapid7 NeXpose, Qualys, and others, can identify weaknesses in your organization's internal network. Nessus and Nmap, two free tools, are among my favorites for scanning.

After placing a suspect system on a small hub and doing the backup, I have sometimes found it helpful to run Nmap on the target computer from another system on the hub. Several times this has given me insight into the potential problems I may be dealing with by showing unexpected listening ports.

Running a security scanner on the neighboring systems in a compromise can help you make sure you have full and complete eradication. If one system is compromised, there is every chance the number is actually two or more.

Course Roadmap

- **Incident Handling**
- Applied Incident Handling
- Attack Trends
- Step 1: Reconnaissance
- Step 2: Scanning
- Step 3: Exploitation
 - Gaining Access
 - Web App Attacks
 - Denial of Service
- Step 4: Keeping Access
- Step 5: Covering Tracks
- Conclusions

Incident Handling

1. Overview
2. Incident Handling Process
3. Preparation
4. Identification
 - Lab 1.1: Windows Cheat Sheet
5. Containment
6. Eradication
7. **Recovery**
8. Lessons Learned
9. Enterprise-Wide IR
 - Lab 1.2: Enterprise Ident. and Analysis

Now it's time to get back in business. That's what the Recovery phase is all about.

- The goal of the Recovery phase is to put the impacted systems back into production in a safe manner
- Validate the system
 - Once the system has been restored, verify that the operation was successful and the system is back to its normal condition
 - Always ask for test plans and baseline documentation
 - Run through the tests, or better yet, have the business unit retest
- Run through user acceptance testing documentation

Remember that after you have touched the machine, everything that breaks is your fault. Be sure to get the owner of the machine to sign that it is back in full operation. Make every effort to ensure that the system is working properly before leaving the scene. If some functionality is not present, the default stance is usually to blame the incident handling team in some organizations. You need to proactively avoid such a situation by having the business unit test the machine before going back into production.

User acceptance testing documentation can also be fantastic to validate that applications are working properly. This is the documentation that an application has to go through before it can be officially deployed.

- **Decide when to restore operations**
 - Try for an off-hour time slot
 - It's easier to monitor carefully
 - You will often be overruled on this, with the business opting to restore service immediately once systems are rebuilt and ready
- **Put the final decision in the hands of the system owners**
 - Provide your advice, but they make the final call
 - Document your advice in a signed memo

The decision of when to put the system back into business has to be made by the system owner. As a handler, you can give them advice and be helpful, but this is their call. They are the ones that depend on this system.

Document your advice in a signed memo to the system owner. Remember, you may not get your way in this decision, but at least you document your recommendations.

- Monitor the systems
 - Once the system is back online, continue to monitor for backdoors that escaped detection
 - Utilize network and host-based intrusion detection systems and intrusion prevention systems
 - If possible, create a custom signature to trigger on the original attack vector because attackers will likely try the same thing again
 - Also, carefully check operating system and application logs

Needless to say, if the eradication was not complete or the infection vector was not closed off, the earlier you detect reinfection, the better off everyone is. It is also politically better if the handlers detect the problem and show up to fix it than if the problem comes to light because business operations are affected. This is a serious problem. Many times handlers take some shortcut along the way, or there is something you never discovered about the attack vector, and the problem comes back.

- It is critical to regularly check if the system compromised again
 - Attackers like to return to systems they've already compromised
 - Not always using malware, sometimes logging in via normal mechanisms
- Write a script that checks whether the artifacts left by the attacker have returned and run the script daily (or even more often) for several months
 - Look for changes to configuration via registry keys and values
 - Look for unusual processes
 - Look for accounts used by the attacker
 - Look for other artifacts we discussed during the earlier lab
 - Look for simultaneous logins
- In other words, apply the cheat sheet techniques
 - But now looking for specific attacker artifacts indicative of the attacker
 - Over 100 examples of checker commands at Command Line Kung Fu blog

One of the most important things for incident handlers to do in the Recovery phase and in the months following an incident is to check regularly to see if the attacker has returned. Most human attackers that compromise machines do return to the scene of the crime, making the same or similar changes to the system that they made upon initial compromise. Some attackers do not use malware, preventing handlers from using malware detection to identify the attacker's return. Instead, these attackers use normal login mechanisms to access the system and reconfigure the machine to maintain control. We need to look for these changes aggressively to detect an attacker's return.

Because of this fact, we urge handlers to write up one or more simple scripts that they can run on a daily basis to look for artifacts left by the attacker in the previous compromise cycle. If the attacker comes back and creates the same artifacts again, your script should detect that fact and notify you. We recommend running the script daily or even more often.

Your script could look for changes to the configuration of a Windows machine by checking for changes to registry keys and values using the `reg` command, which works remotely using the `reg \\MachineName` syntax. For example, if the attacker alters the configuration of an application via a registry key so that it logs credit card numbers, make sure you look for that kind of change in the registry.

Or you could look for unusual processes using the `wmic` command (which works remotely when run with `wmic /node:MachineName /user:Admin /password:password`) or the `tasklist` command (which can be run remotely using the `psexec` command from Microsoft Sysinternals). On Linux, you could use the `ps` command to get a list of processes.

Also, look for accounts the attacker created, which can be pulled via `wmic` with `wmic useraccount list brief` or the `net user` commands. On Linux, you can get this information via `cat /etc/passwd`.

Likewise, you should look for other artifacts covered by the Intrusion Discovery cheat sheets. The good news is that your script can look for *specific* items created by the bad guy during the earlier incident, so you can focus in on detecting certain changes to identify the attacker's return. The blog at <http://blog.commandlinekungfu.com/> includes more than 100 examples of commands that check various aspects of system status for Windows and Linux.

Course Roadmap

- **Incident Handling**
- Applied Incident Handling
- Attack Trends
- Step 1: Reconnaissance
- Step 2: Scanning
- Step 3: Exploitation
 - Gaining Access
 - Web App Attacks
 - Denial of Service
- Step 4: Keeping Access
- Step 5: Covering Tracks
- Conclusions

Incident Handling

1. Overview
2. Incident Handling Process
3. Preparation
4. Identification
 - Lab 1.1: Windows Cheat Sheet
5. Containment
6. Eradication
7. Recovery
8. **Lessons Learned**
9. Enterprise-Wide IR
 - Lab 1.2: Enterprise Ident. and Analysis

Suppose an incident was detected at 4:30 p.m. on a Friday afternoon, just as you were hoping to slide out the door. You stayed all night and finally got control of it at about midnight. While it was going on, you had all that adrenalin; you felt like you were part of something pretty darn exciting and now you just wish it would go away! The adrenalin has worn off and you are tired; in fact, you are sick of the whole situation and would prefer to live your life without ever hearing about it again!

It takes discipline to do the report. It takes even more to attend a Lessons Learned meeting, but it is important. Attackers are improving. We have to improve as well. One way to improve is to learn from our mistakes and move on to make new mistakes instead of repeating the old ones. This is the primary purpose of the follow-up part of the process!

- The goal of the Lessons Learned phase is to document what happened and improve our capabilities
- Develop a follow-up report
 - Start as soon as possible, right after recovery (i.e., going back into production)
 - Assign the task to the on-site team
 - SANS SCORE has sample forms to include
- Encourage all affected parties to review the draft
 - Attempt to reach consensus and get sign-off
 - Document any disagreements so there are no surprises in the future

The only one that really can or will write the report is the on-site handler. The handler submits the draft to the head of the incident handling team. This chief edits the document and interacts with the handler to make sure the document reflects what actually occurred, in light of the organization's culture. We should allow everyone involved to review the draft. Have everyone involved in handling the incident sign off on the report, agreeing to its contents.

The SANS SCORE project has sample incident handling forms at <https://www.sans.org/score/incident-forms>

If anyone has a strong disagreement about the facts of the matter, he can submit that, and his statement can remain a part of the incident record. It is far better to find out that you have a lack of consensus before going to court than during court!

- As soon as practical (within two weeks of resuming production)
- Review the report
- Finalize executive summary
- Keep it short and professional
 - Maximum length: Half day

After the report has been reviewed, schedule a Lessons Learned meeting. In general, the main purpose of such a meeting is to get consensus on the Executive Summary section of the report. This meeting should occur within two weeks of resuming production, while the events and report are still fresh in people's minds.

What is the most important thing for an executive summary to cover? How much the organization saved by having an effective incident handling procedure and team!

During every incident, mistakes occur. We learn from these, improve our process for the future, and move on. Sometimes we run into policy or other organizational problems that hinder bringing the incident to a close. We note these and submit them to management for its consideration.

Follow-up meetings are never the most popular of events. Everyone is tired; they have been under stress. The system is now back in operation and the last thing anyone wants to do is have a meeting to rehash painful memories.

However, this is a valuable tool for organizational improvement. This is the hardest time not to blame people. The focus should be on process improvement.

- Based on what you learned, get appropriate approval and funding to fix
 - Your processes
 - Your technology
 - Improved incident handling capabilities
- Do a root cause analysis
 - E.g. attacker was able to guess a weak password
 - Why was the weak password allowed in the first place?
 - Lack of policy
 - Lack of enforcement of policy
 - Weak policy

We want to have constant improvement so the cause of the incident can be eliminated or minimized. You must go to management and make a compelling case for fixing the problem that caused the incident in the first place. This may mean an alteration to the processes or technology in your environment.

When trying to improve, you will want to do a *root cause analysis*. This involves looking beyond just the immediate problem, and trying to find out why such a problem could even occur.

For example let's assume an attacker was able to gain access to a system because of a weak password. Obviously you will want to change the password, examine the compromised system, and so forth. However it's also important to ask *why* a weak password was able to be used in the first place?

Was there no password policy to begin with? Perhaps there is a password policy, but it wasn't properly enforced on the system the attacker compromised. Or perhaps there is a password policy, and it was enforced but the policy (unintentionally) allows weak passwords.

It's important to do this type of process because the differing underlying causes require different solutions.

Course Roadmap

- **Incident Handling**
- Applied Incident Handling
- Attack Trends
- Step 1: Reconnaissance
- Step 2: Scanning
- Step 3: Exploitation
 - Gaining Access
 - Web App Attacks
 - Denial of Service
- Step 4: Keeping Access
- Step 5: Covering Tracks
- Conclusions

Incident Handling

1. Overview
2. Incident Handling Process
3. Preparation
4. Identification
 - Lab 1.1: Windows Cheat Sheet
5. Containment
6. Eradication
7. Recovery
8. Lessons Learned
9. **Enterprise-Wide IR**
 - Lab 1.2: Enterprise Ident. and Analysis

Now let's discuss how to handle an enterprise-wide incident.

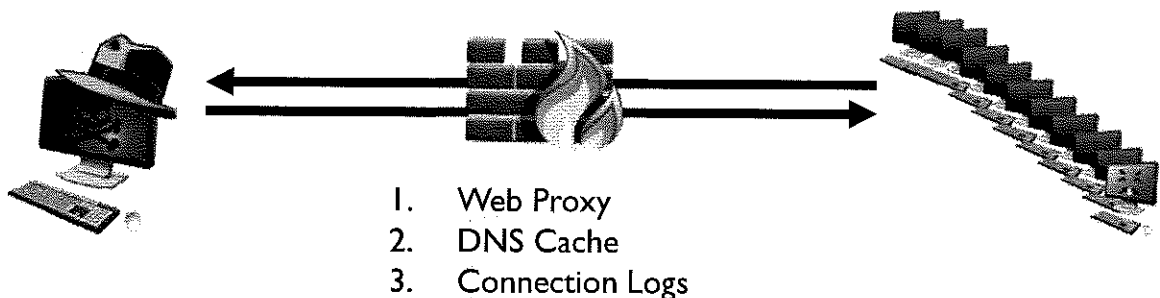
- Determining if one system is compromised can be difficult
- Doing this at scale across thousands of systems can seem impossible
- With the right tools and techniques, it is possible
- Many of the data points are already being collected in your environment
- You just need to know where to look

There are many different tools at the disposal of any incident responder who is working a large-scale incident. Some of these tools are commercial, and others most likely are being used actively by your enterprise right now.

However, the goal of any enterprise-wide incident response engagement is the same: to identify indicators of compromise across multiple machines and user accounts.

The vast majority of environments are already collecting logs and data for many of the components we will discuss over the next few slides. It is just an issue of tapping into those data sources and extracting the proper data.

- Need to start somewhere
- Connection data from the various points of presence is a good start



When we are working any incident, the first thing our responders ask for is the logs for the egress connections from the firewall, the DNS cache or DNS logs, and, finally, the logs from whatever external web-filtering device the organization is using.

There are a couple of reasons for this. First, the logs from these devices work well, as they serve as filter points for all traffic leaving the environment. Remember, in situations like an incident, the goal is not to stop an attack from coming in, but rather to be able to detect command and control points and identify additional internal systems that may be compromised. In addition, if a network is configured properly, all egress traffic should either flow through an egress firewall, resolve a domain via DNS, or connect through an egress web proxy.

- DNS can be very powerful
 - Simply reviewing a DNS server's logs and cache can reveal systems that are connected to known bad IP addresses and domains
- DNS Blacklists is a Python script by Ethan Robish to identify traffic to known malicious IP addresses and domains
 - Feed it DNS logs, cache, etc. Anything with IP addresses / domains
 - Uses regular expressions for flexibility
- Need to feed it a list of malicious hosts
- Malware Domain List is a great site for updated lists of known bad actors

DNS data can be one of the most powerful tools you have for detecting malicious traffic leaving your environment. First, there are many examples where traditional AV fails to detect well-known malicious programs from running. Second, many of the domains used by well-known botnets and C2 channels tend to be more static than the code base for the malware. Finally, it is relatively easy to compare the logs and/or the current cache for your DNS server with well-known evil domains and IP addresses.

One of the tactics we use as part of an initial incident response is to compare the current cache of the DNS server with a list of evil IPs and domains by using a tool like `dns-blacklists.py`. One of our favorite lists to compare with is the one from Malware Domain List.

You can find Ethan's Python script at: <https://bitbucket.org/ethanr/dns-blacklists>

Malware Domain List is at: <http://www.malwaredomainlist.com/mdl.php>

- Many organizations do not review these logs
 - Often, this is due to HR issues
- However, regular review can uncover compromised systems that are connecting to known bad C2 sites
- Review the length of URLs being visited
 - Many malicious URLs are very long
- Review user agent strings
 - Many malware specimens use older or odd user agent strings

Most every enterprise environment today has some sort of web proxy content filter to restrict employee access to sites with objectionable content. Further, these can be powerful incident response tools for any responder.

There are a couple of things to look for. First, you can once again see if there are any IP addresses or domains being accessed that are known bad actors. Sure, many of these providers have their own regular updated blacklists. However, it is good to get a second, or even a third, opinion.

Next, look at the length of the URLs being accessed. Many variants of malware will use long encoded URLs either as a command and control mechanism or as a way to deliver payloads. However, be careful because many legitimate sites also use long URLs. We recommend using this in conjunction with other indicators of compromise. For example, let's say one of your systems pops an AV alert. You then notice that some of the domains it is accessing are odd. Then, look at the URLs as another possible investigation item. This also has the added bonus of potentially identifying additional systems that are also compromised in the process.

Finally, review the user agent strings. If most of your systems are Windows 7 with the newest version of Internet Explorer, and you start seeing Windows XP with IE version 6, this could be malware. At the least, it is a system that needs to be upgraded. The reason for this is that some malware uses old user agent strings to "blend in." However, over time, they forget to upgrade or modify them.

- Sifting through thousands of packets can be daunting
- However, reviewing NetFlow data can reveal interesting patterns in connection statistics
 - Systems beaconing out every 30 seconds
 - Systems beaconing out at random intervals
 - Connections that live for far longer than they should
- Real Intelligence Threat Analytics by Active Countermeasures can do this
- Hunting for evil actors is a big part of SANS SEC511: Continuous Monitoring and Security Operations

Another, often overlooked, component is seeking out beaconing data. It is often assumed that malware makes a persistent connection back to the bad guys' command and control systems. However, with today's modern malware, this is just not the case. Rather, malware today tends to connect back at regular (or irregular) intervals. In order to detect this connection method, it requires the analysis of the connection logs of your egress firewall that performs Network Address Translation (NAT) from internal RFC 1918 IP addresses to externally rotatable IP addresses. Eric Conrad has an excellent tool to parse these log files to find connection attempts.

It can be found here:

<http://tinyurl.com/505and511>

It is also part of the SEC511 Continuous Monitoring and Security Operations class.

The Real Intelligence Threat Analytics (RITA) product from Active Countermeasures can also do this type of analysis. It can be found at <https://www.activecountermeasures.com/free-tools/rita>. We'll take a look at RITA in the next lab.

Course Roadmap

- **Incident Handling**
- Applied Incident Handling
- Attack Trends
- Step 1: Reconnaissance
- Step 2: Scanning
- Step 3: Exploitation
 - Gaining Access
 - Web App Attacks
 - Denial of Service
- Step 4: Keeping Access
- Step 5: Covering Tracks
- Conclusions

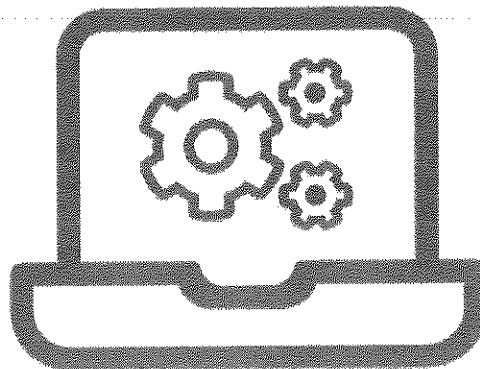
Incident Handling

1. Overview
2. Incident Handling Process
3. Preparation
4. Identification
 - Lab 1.1: Windows Cheat Sheet
5. Containment
6. Eradication
7. Recovery
8. Lessons Learned
9. Enterprise-Wide IR
 - **Lab 1.2: Enterprise Ident. and Analysis**

Now, let's have a quick enterprise-wide incident lab.

LAB 1.2

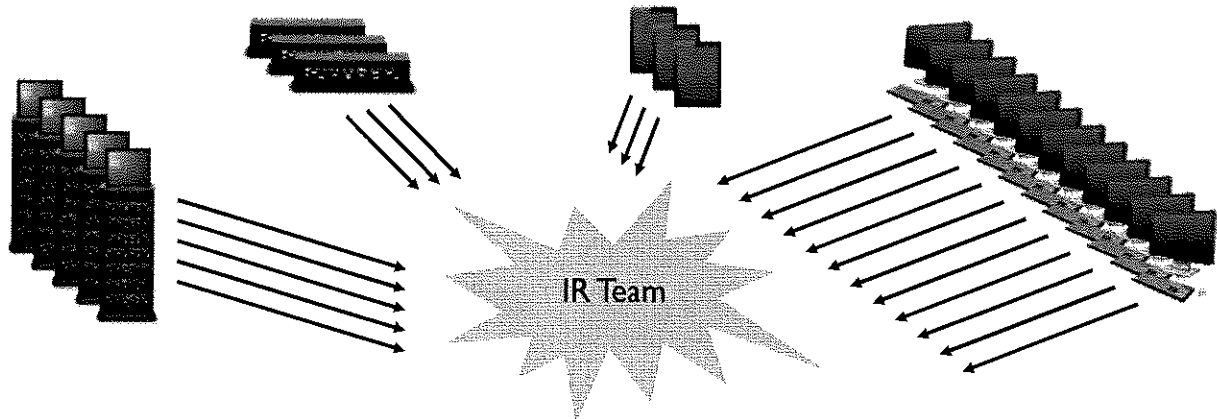
Please work on the lab exercise
*Enterprise-Wide Identification and
Analysis*



This page intentionally left blank.

Pulling Data from Multiple Systems

Enterprise-Wide IR



Expensive Third-Party Tools Not Required!

Next, we need to have a way to extract data from multiple internal computers to look for indications of compromise.

There are a number of useful techniques, and not all of them require an investment of third-party software. In fact, many of these tools are already incorporated in most enterprise environments.

Using WMIC to Pull Data from the Enterprise

Enterprise-Wide IR

```
C:\> wmic product get name,version,vendor
```

Get information about installed products on a local system

- The `/node:@systems.txt` allows you to run the same command on multiple systems

Get information about installed products from across the enterprise

```
C:\> wmic /node:@systems.txt product get name,version,vendor /format:csv > SoftwareInventory.txt
```

- All the commands we ran in our Windows Cheat Sheet lab can be run against multiple systems

SEC504 | Hacker Tools, Techniques, Exploits, and Incident Handling 127

Earlier, we discussed the awesome power of WMIC. There is no greater tool for quick-fire incident response than WMIC. During the Windows lab, you were able to see that WMIC can query just about everything. However, the real power of WMIC is the `/node` switch. With this switch, you can run WMIC commands on multiple systems at once and have the data reported back to you in CSV or XML format for easy parsing.

It requires you to be a domain admin in order to be completely effective. This is a consistent theme for all of these tools.

Above are just some of our favorite IR WMIC commands.

- Dependencies
 - PowerShell 3.0. A simple domain-wide install of 3.0 is required
 - For full functionality, install Handle.exe and autorunssc.exe from Sysinternals from Microsoft
 - For the machine launching the scripts, install Logparser from Microsoft
 - Enable Windows Remote Management on all targeted systems
- Add all hosts you want checked into a text file and loaded into the Kansa-Master directory

```
C:\> winrm quickconfig
```

Enable WinRM by running this on each system.
Alternatively, use Group Policy

Before we get started, some setup work is required. The required steps in the slide will configure Kansa properly.

The major thing to take from this is that all the required components for Kansa to run properly are from Microsoft. It is fairly easy to pull these files down and disperse them via Group Policy to the systems in your environment.

Note that autorunssc.exe and handle.exe need to be installed to the C:\Windows directory so they can be found by Kansa when it runs.

Also, this type of baselining is critical for effective incident response. The point is, run this before an incident occurs. This is a powerful tool, but it is only as good as the admins who run it.

```
PS C:\Kansa-master> .\kansa.ps1 -Targetlist .\hosts.txt -ModulePath  
.\Modules -Verbose -Analysis
```

```
VERBOSE: Found .\Modules\Modules.conf
```

```
VERBOSE: Running Modules:
```

```
Get-Netstat
```

```
Get-DNSCache
```

```
Get-Handle
```

```
... output truncated ...
```

```
VERBOSE: $Targets are clarence bob jackie james Jason.
```

```
VERBOSE: Waiting for Get-Netstat to complete.
```

Id	Name	PSJobTypeName	State	HasMoreData	Location
351	Job351	RemoteJob	Completed	True	clarenc...

```
VERBOSE: Waiting for Get-DNSCache to complete.
```

352	Job352	RemoteJob	Completed	True	clarenc...
-----	--------	-----------	-----------	------	------------

Use -Analysis to make it
easier to find outliers

The slide image is what Kansa looks like when it is running properly.

Note that it can be run on multiple systems by using the `-Targetlist` switch with a file listing all of the systems you want it to run on.

Also, it is helpful to use the `-Analysis` switch. This triggers all the stacking components and analysis components. It is far easier to do incident response when comparing and contrasting systems against each other rather than looking at each system as a one-off.

- Kansa supports the ability to pull the total count for specific things, such as Autostart Extensibility Points (ASEPs)
- In the following example, notice the count column on the left side:

cnt	Image Path	MD5
10	c:\windows\system32\cmd.exe	78af816051e512844aa98
10	c:\windows\system32\cmd.exe	54879ccbd9bd262f20b58
10	c:\windows\system32\cmd.exe	60668a25cfa2f1882bee8
2	c:\program files\gcat\gcat.exe	b79713939e97c80e204de
10	c:\program files\hpwbem\storage\service\hpwmistor.exe	202274cb14edae27862c
10	c:\hp\hpsmh\bin\smhstart.exe	5c74c7c4dc9f78255cae7
10	c:\msnipak\win2012sp0\asr\configureasr.vbs	197a28adb0b404fed01e9

Note: A callout box points to the 'cnt' column, stating: 'Pay attention to programs that are on only a few systems'.

In the slide image, there are a number of different autostart programs pulled from ten different systems. Rather than attempting to review each system's ASEPs, you can simply look at the total count of ASEPs, which exist on all systems.

Ideally, you should be looking for one-offs or entries that do not appear on all systems.

This can be a total nightmare in environments that do not practice consistent build processes and solid change management.

Kansa Things to Look For

Enterprise-Wide IR

ct Entry

```
1 clarence
1 231.1.12.10.in-addr.arpa
1 www.trulyevil550.com
1 www.monkeycheesepants.com
```

ct Account

```
5 IEUser
5 TEST\Domain Admins
5 Administrator
1 Support_31337
```

Look for things that are different, and things that show up only on a few systems.

ct Value

```
5 Microsoft.Windows.Defender
5 {D65231B0-B2F1-4857-A4CE-A8E7C6EA7027}\WindowsPowerShell\v.10\powershell.exe
5 {D65231B0-B2F1-4857-A4CE-A8E7C6EA7027}\OpenWith.exe
1 {D65231B0-B2F1-4857-A4CE-A8E7C6EA7027}\Fondue.exe
1 C:\Users\Administrator\Downloads\msf.exe
```

WAS

SEC504 | Hacker Tools, Techniques, Exploits, and Incident Handling 133

Ideally, malware jumps out at you. However, in reality, you are hoping to get two things from Kansa. First, you want to make the size of the data you are sifting through smaller. Think of the needle-in-the-haystack analogy. You are trying to make the haystack much smaller.

Second, you are trying to look at what exists on a smaller scale, that is, to look at what is installed and running on a few systems. This is where you will find malware and backdoors interesting.

Of course, your entire environment can be compromised with the same malware. Although this does happen, we usually find that bad guys use malware for initial compromise and then use existing credentials to pivot and pillage.

Course Roadmap

- Incident Handling
- **Applied Incident Handling**
- Attack Trends
- Step 1: Reconnaissance
- Step 2: Scanning
- Step 3: Exploitation
 - Gaining Access
 - Web App Attacks
 - Denial of Service
- Step 4: Keeping Access
- Step 5: Covering Tracks
- Conclusions

Applied Incident Handling

1. **Espionage**
2. Unauthorized Use
3. Insider Threats
4. Legal Issues and Cybercrime Laws
 - Lab 1.3: IR Tabletop
5. Appendix A: Intro to VMware and Linux Workshop

At this point, we completed the six-step incident handling process: Preparation, Identification, Containment, Eradication, Recovery, and Lessons Learned. This is a seasoned process that has worked for a large number of organizations and will work for you. Next, we work on some of the specific types of incidents that we may run into.

We look at several incident types and apply our six-phased process to them. We start with espionage.

- Stealing information to subvert the interests of an organization or government
 - Consider the high-profile Aurora, Titan Rain, and Night Dragon cases
 - Large organizations, high-value information theft
- Many cases of unauthorized access to corporate systems are for espionage purposes
 - Almost every espionage case prosecuted by the US government involved a trusted insider
- Espionage and insider criminal cases do not benefit from many helpers
 - Risk of information leak or evidence contamination rises as additional workers are added to the investigation
 - A senior member of management, such as the CIO or Chief Security Officer, must be advised, as well as the legal staff
- Many recent high-profile attacks were espionage

If you are thinking, "Espionage could never happen to *our* organization," think again. Espionage is not limited to governments and the military in any way. The focus of most espionage is not military; it is economic, and this includes the work done by government intelligence agencies.

Businesses routinely are involved in the activities of collecting information about their competition or trying to prevent the competition from getting information about their activities. As long as this is legal, we generally refer to this as competitive intelligence.

As handlers, we are primarily focused on the defensive strategies, of course, but we should take a minute to think about some of the obvious offensive techniques. Some common techniques include:

- Open source searches by your adversaries to see what information is publicly available
- Posing as a customer or potential customer to gain sensitive data
- Hiring critical employees as insiders, in effect working on behalf of your adversaries from the inside of your organization

There have also been a large number of high-profile compromises where the attackers were motivated to steal sensitive corporate secrets from large multinational companies.

- Ask what the most probable targets (information and processing capability) of the activity are
- For each probable target ask
 - What is the information worth?
 - Who (outside the organization) might benefit from having it?
 - What are all the possible ways to acquire these targets?
 - What are the two or three most likely ways to acquire these targets?

KODAK referred to its multilayer emulsion technology as the crown jewel. What are the jewels that are most valuable in your environment? These are quite likely the target(s).

This is critical; the physical differences between your organization and your competition are probably minimal. Those trade secrets, marketing contacts, business plans, and so forth make all the difference. The odds are fairly high that these crown jewels are the target. By now, you are learning to think like an attacker to some extent. What would you go after if you were interested in plundering your organization? How would you get to this information? Document the results of this exercise and share the information with management to get their input.

- Before-/after-hours access, work weekends, volunteering to empty paper recycling
- Pattern of access violations in audit trails
- Leak seeding (media leaks)
- Thumbprint critical files and search for keywords
 - Custom network-based IDS signatures
 - Custom firewall/IPS signature-matching technology
 - Google searches can be useful if attacker is storing information on publicly accessible websites

Activity that begins too early before the working day or goes on too late has always been a good indicator. Technology changes, not human nature. Many organizations have been sold out by trusted insiders; the trick is to be alert for this, to work to keep awareness up and keep running leads to ground.

A great way to thumbprint critical files is to invent an acronym that doesn't actually exist and plant it into the document. Then, if you have content-sensing firewalls, intrusion detection systems, or network-based intrusion prevention systems, they can be set to look for the string. Google searches can also be useful in finding data that has been leaked and placed on the web.

Intentional leaks work well, especially as you start to close in, and this is a standard practice used to hunt down the source of information being released.

- Ensure that access records of the affected facility are collected and protected
 - Records from badge access systems
 - Phone records from your organization's PBX (Private Branch Exchange)
 - Log books
 - System and Network logs
 - Surveillance videos
- Collect as much back data as possible
- Make sure you can access this type of data when you really need it
 - By asking for it periodically, as a course of doing business

One of the huge challenges is to get some degree of a chain of custody when you are dealing with a lot of data and in a large number of formats. This can quickly end up filling a dozen or more copy paper boxes. When possible, take cryptographic hashes of critical log files. These are very small and can be stored in multiple locations. I have even mailed them to trusted friends, simply asking them to store the hash or PGP signature of the file for me without telling them why. To the extent possible, make every effort to keep the data as pristine as possible.

Also, you want to use some care in the collection of the data; you can get away with one cover story, but at some point, people start to talk. This is why incident scenario training is such a valuable tool. If you are asking for door "badge swipe" records a couple of times a year and it is well known you are training, you not only make sure the logs are being kept and can be retrieved, but when you really need them, it will not arouse anywhere near as much suspicion or interest as it would if no one had ever asked for them before.

- If an outsider is collecting the information, you may be able to provide erroneous information and actually benefit from the incident
- If you suspect the information is being collected and distributed by an insider, this is less likely to work
 - However, the technique can be used to pinpoint the insider
 - Make up a fake activity called "Project XYZ" or a bogus bid for a client
 - Configure a network-based IDS and/or antivirus tool with custom signatures to look for this fake data

You may also be able to use erroneous or misleading information to detect that a leak exists. This is a classic trick and can sometimes not only help track down the insider, but if you are really clever, you can use this to your advantage.

One of the most difficult cases I have worked on involved contract bids where there was some reason to think that an insider, probably a system administrator, was accessing information on the bid. In this case, we constructed a completely fabricated bid (that turns out to be a lot of work, by the way) and really kept up the act, making changes as the bid deadline approached, sending information from the boss to the folks making up the bid. All the while, the real bid was being prepared by a different team composed of a couple individuals that were supposedly "on leave." The idea was to feed the other side enough wrong information to really screw them up. It didn't work; probably there was some error in the fake bid that tipped our hand. Either the other side was actually that sharp that they could spot the fake or perhaps we were dealing with a second person that was able to observe what we were doing. This is a hard sport!

Course Roadmap

- Incident Handling
- **Applied Incident Handling**
- Attack Trends
- Step 1: Reconnaissance
- Step 2: Scanning
- Step 3: Exploitation
 - Gaining Access
 - Web App Attacks
 - Denial of Service
- Step 4: Keeping Access
- Step 5: Covering Tracks
- Conclusions

Applied Incident Handling

1. Espionage
2. **Unauthorized Use**
3. Insider Threats
4. Legal Issues and Cybercrime Laws
 - Lab 1.3: IR Tabletop
5. Appendix A: Intro to VMware and Linux Workshop

For incidents involving unauthorized use, the attacker is allowed normal access in the course of doing business. However, the attacker abuses this access, using it in an unauthorized fashion.

- For unauthorized use, the user is allowed normal access but is abusing it
- Numerous possible categories
- but let's focus on two areas that incident handlers are frequently called upon to support:
 - Email problems
 - Inappropriate web surfing

Unauthorized use is something that every handler is likely to face. We deal with a couple of common problems: unsolicited, non-spam email and inappropriate use of computing resources. As we look at this section, keep in mind that you are an incident handler, not a cop, not a lawyer, not your organization's Human Resources department. Don't cross the line and do their jobs for them. On the other hand, you are likely to be coming from a technical perspective. Who better than to assess the situation and collect, protect, and analyze any evidence than the handler?

- Message(s) from the employee's machine
- All logs from employee's server(s) and email server(s)
- Logs from organization's mail relay(s), even if you are SURE it is internal
- Firewall/intrusion detection logs
- When comparing logs, timestamps matter
 - Be certain to account for clock skew
 - Are the timestamps UTC/GMT or local ("wall") time?

Keep in mind that you may face situations where the mail appears to be coming in externally. For example, it is coming from the internet, but it could still be someone internally sending it. The simplest example of this is when an employee surfs to Hotmail to send the note into the organization. If you have decent monitoring, this is easy to run to ground. If he uses a telephone and calls up to an ISP to send the mail, you may have to pull phone records or work with law enforcement to subpoena records from the ISP.

Many log files are perishable. As soon as you know you may need to collect evidence, you probably should. Have I mentioned that it is a good idea to SHA-1 and MD5 hash logs after they are closed? This point about perishable is important: Many times, the employee that is receiving the offensive mail doesn't say anything until he has received 15 or so over a six-week period. Then, he hands the stack to his manager, who doesn't know what to do, so she sits on it a week before giving it to HR, and HR waits two days until the manager calls back and then HR calls in a handler. Remember from the first part of this course, we were talking about awareness and knowing how to contact your organization's incident handling capability? Situations like this are where the rubber meets the road. If you don't know there is a problem for six weeks, it may not be possible to collect all the logs needed to do a complete investigation.

When collecting log files from multiple sources (firewalls, IDS systems, AV logs, etc.) it is important to keep track of clock drift/skew. This is when one system is 5 minutes faster than another. Another piece of information to keep track of is if the timestamps are in UTC/GMT time, or if they are local time (the time you would see if you looked at an analog clock on the wall). If they are in local time, you need to know the time zone the logs were written in, so you can adjust for things like daylight savings time.

From: JohnDoe@hotmail.com
To: JohnSmith@myorganization.com
Subject: Affair

John, everybody in the office knows
you are having an affair with Mary. What
if your wife finds out?

Handlers are not responsible for mending broken hearts, but they may have to deal with them. A common situation is for a female employee to receive unwanted email. It is usually pretty easy to determine the source in these cases. After all, the prospective suitor wants to be noticed and found. Once the data is in, this is an HR thing, not a handler issue.

Domestic disharmony is also usually pretty easy to run to ground. After all, the source will be someone close to the victim. Take a look at this message. Who is on your short list for people that might have sent this? John Smith's spouse and her close friends and relatives. If any of them work in the same organization, that could be a clue. For some reason, Hotmail has been the most often used service for insider-sent problem email messages.

Sorry to harp on how the initial data is often misleading, but I worked a case in Colorado where malicious code that damaged five computers was sent into the organization from an account with the name of a female employee@hotmail. By the time I was alerted, they had restored all the operating systems on the computers, which put the best evidence in doubt. They had also begun the inquisition of the female employee. We were able to pull the mail, firewall, and IDS records, but this facility had a tradition of people going to Hotmail during the day, so we had about 20 leads. We actually were able to solve this case and get a conviction, but it was by pure dumb luck, not by something we were able to do. It was a 15-year-old kid who bragged about what he did.

Email Scenario 2: Phishing

Unauthorized Use

From: security@bigbank.com
To: victim@yourISP.net
Subject: Critical Changes to Your Online BigBank Account

Dear BigBank valued customer,

In order to service you better, we have made some changes to the way you access your BigBank Online Account. From now on, access will be managed by BigBank Online Management Center.

Please note that BigBank Online Management Center will be the only way you can access your account.

Click on the following link to access your account:
www.bigbank.com/login.asp

TIP

You can report phishing to the bank (or other organization that appeared to send the email), the ISP, and the Anti-Phishing Working Group

SEC504 | Hacker Tools, Techniques, Exploits, and Incident Handling 144

This email represents a phishing attack. The perpetrators are trying to harvest account information from unsuspecting users by setting up a website that poses as the actual bank's site. Then, with an email blast often carried by worm-infected systems (see the previous slide), they trick users into surfing to the attacker's site. You should report phishing to the bank, the ISP, and www.antiphishing.org.

The links included in phishing emails are actually accessing the attacker's site but trick a user in any one of a variety of ways. Often the attackers use an `<A HREF>` tag to display certain text on an HTML-enabled email reader screen, with the link actually pointing somewhere else.

First, and perhaps most simply, the attacker could simply dupe the user by creating a link that displays the text www.goodwebsite.org but really links to an evil site. To achieve this, the attacker could compose a link like the following and embed it in an email or on a website:

```
<A HREF="http://www.evilwebsite.org">www.goodwebsite.org</A><p>
```

The browser screen will merely show a hot link labeled www.goodwebsite.org. When a user clicks it, however, the user will be directed to www.evilwebsite.org. Browser history files, proxy logs, and filters, however, will not be tricked by this mechanism at all, because the full evil URL is still sent in the HTTP request, without any obscurity. This technique is designed to fool human users. Of course, although this form of obfuscation can be readily detected by viewing the source HTML, it will still trick many victims and is commonly utilized in phishing schemes.

More subtle methods of disguising URLs can be achieved by combining the above tactic with a different encoding scheme for the evil website URL. The vast majority of browsers today support encoding URLs in a hex representation of ASCII or in Unicode (a 16-bit character set designed to represent more characters).

- Suppose a manager calls and complains
 - Employee spending too much time on the web
 - Access to sexually explicit material
- Advise manager that all such calls should be directed to Human Resources
- Only respond if HR requests such action in writing!

Unauthorized use comes up from time to time. An incident handler may receive a call from a manager saying, "I'm concerned about Bill Smith, his work appears to be off, can you:

- A) Access his email?
- B) Use the intrusion detection system to track his activities?
- C) Copy his hard drive while he is away from his desk?
- D) All of the above?"

Your answer, even in an organization with warning banners and no presumption of privacy, should be NO! If the same call comes in from Human Resources, and you have written a policy for when your organization might do any of the above, then ask them to confirm their request in writing. You may have to do these things because you are an expert in collecting and assessing evidence.

It is one thing to randomly monitor for certain strings, "SECRET," "CONFIDENTIAL," even "Supermodels." It is another thing entirely to use your skills to target an individual. Make sure that you are on firm, written, legal ground.

- Sexually explicit web access is a major problem for many organizations
- Such material is considered inappropriate because
 - It could be a factor in a sexual harassment case
 - It could embarrass your organization
 - These sites sometimes distribute spyware and other forms of malicious code
- Our jobs as incident handlers involve helping our organizations minimize damage from the misuse of computer systems
- We sometimes are called to get involved with this type of situation

You know the story about the elephant on the table? There is an elephant on the dining room table, but no one will talk about it. This metaphor illustrates a typical problem with a substance abuser in a family: everybody sees it, but no one wants to talk about it.

Sexually explicit access is one of the biggest money vectors on the internet. Yet no one wants to talk about it. A lot of your organization's time is lost. Further, if countermeasures are not applied, your organization could be in for a major lawsuit. That said, incident handlers are not responsible for the organization's policy. Senior management manages risk and, in the final analysis, this is simply one more risk management decision.

One thing you want to establish from a policy perspective is what to do if you happen to run into images that are illegal, such as child pornography. Do you call law enforcement? Probably; there is a better than even chance that they already know those images are in your facility. Speaking of possessing illegal materials, if your organization has a public FTP server, be sure and check it closely from time to time for hidden directories. The best place to store illegal files is on someone else's computer.

- If such activity is suspected, the wisest choice is to implement proxy countermeasures to block access to such sites
 - Forcepoint
 - Symantec Blue Coat
 - Numerous others
- This is not foolproof, but it works for flow reduction
 - By stopping 90% of the problem, it allows handlers to focus on the remaining 10% and other sensitive issues
- Also, it shows the organization's culture does not condone such activity

Nothing beats NetNanny-style filtering software to keep your organization out of hot water. For one thing, you just can't argue with a proxy about what is appropriate or not; it just does its thing and people seem to accept that. Also, you really don't want a sexual harassment suit. It takes a lot of time and money to deal with, and your organization will get a lot of negative press.

You should encourage your organization to filter out unwanted websites using a web filtering tool, such as Forcepoint, Blue Coat, or others.

Course Roadmap

- Incident Handling
- **Applied Incident Handling**
- Attack Trends
- Step 1: Reconnaissance
- Step 2: Scanning
- Step 3: Exploitation
 - Gaining Access
 - Web App Attacks
 - Denial of Service
- Step 4: Keeping Access
- Step 5: Covering Tracks
- Conclusions

Applied Incident Handling

1. Espionage
2. Unauthorized Use
3. **Insider Threats**
4. Legal Issues and Cybercrime Laws
 - Lab 1.3: IR Tabletop
5. Appendix A: Intro to VMware and Linux Workshop

This section covers the classifications of insider threats, how to identify potential insider threats, and how insider threats can be prevented.

- Simply stated, a threat from an entity with access to your data
 - An employee: including contract and temporary employees
 - A business partner: someone that has legitimate access, but is not an employee
- Such attackers usually have valid credentials and knowledge of the environment and its business practices

What is an insider threat? Simply stated, it's a threat from an entity with access to your data. This class deals with employee and business partner threats.

A well-intentioned employee makes mistakes that allow proprietary information to leak, or to allow access to your proprietary data.

In the physical realm, this could be an employee holding the door into a secure area for the person carrying lots of books or papers. Or someone that states she simply forgot her badge.

In the cyber realm, this could be an employee at a help desk being socially engineered out of an ID/password combination or a trusting soul that gives their ID/password combination to another.

The disgruntled employee could be someone who believes he may have been overlooked for a promotion, feels he is better than others, and has a need to show it.

The unnoticed employee (a.k.a. the secret thief) is probably the most serious threat you have to your infrastructure. This could be someone operating for a competitor, gathering data for sale to the highest bidder, or someone working for a foreign government.

One point I want to make is that when I use the term "employee," I include employees at all levels (even executives can be suspect), as well as contract and temporary employees.

- Since insider threat activity can involve employees, make sure they are aware of your monitoring
- Warning banner should advise the user that:
 - Access to the system is limited to company-authorized activity
 - Any attempt at or unauthorized access, use, or modification is prohibited
 - The use of the system may be monitored and recorded
 - If the monitoring reveals possible evidence of criminal activity, the company can provide the records to law enforcement
- Have legal team review this banner, approving it in writing
- Be careful of local privacy laws, especially in Europe
 - European Data Privacy Directives may impact that crucial line

Before any electronic detection of insider threats can begin, you must ensure that employees are aware of your monitoring policy.

Should your monitoring uncover any illegal activity, it will be hard, if not impossible, to pursue any criminal or civil charges without proper warning screens on the front doors of your systems. It is important that your company displays a warning message at each login point. This includes all points of remote access, be it VPN, SSH, FTP, dial-up, or all internal sign-on locations. This includes all single-user and shared terminals. This warning message should cover the following five points. It must advise the user that:

- 1) Access to the system is limited to company-authorized activity
- 2) Any attempted or unauthorized access, use, or modification is prohibited
- 3) Unauthorized users may face criminal or civil penalties
- 4) The use of the system may be monitored and recorded
- 5) If the monitoring reveals possible evidence of criminal activity, the company can provide the records to law enforcement

Make sure your legal staff reviews and approves of this wording in writing. That way, they'll support you if a case should ever be challenged. Also, be careful not to run afoul of the European Data Privacy Directives.

- Gathering intelligence on system activity
 - What websites and FTP sites are being visited?
 - Hacking tools, encryption tools, steganography software, free web-based email sites
- Monitor message boards for posted financial or merger information
- Gathering intelligence on your employees' activities
- General searches on the internet are acceptable
 - Casting a wide net is fine
 - Targeting a particular employee should only be done with written HR approval!

Now that you have warned your employees that they may be monitored, you can begin looking for threats.

The best way to identify insider activity is to gather intelligence through proactive scanning and monitoring. You should always be alert for anomalies and keep copious records of those anomalies. These records assist you in identifying a threat to the business.

Monitor message boards for posted financial or merger information.

Detection of anomalies can be accomplished by two means: intelligence gathering on your systems, and intelligence gathering on your employees' activities. Although no one wants to live in the world of Big Brother, it is important the activity be logged for accountability, company protection, and possible legal action.

- With written approval from HR, you can monitor an individual suspect's activity:
 - Identify equipment being used
 - Identify the operating system used
 - Identify the suspect's IP address
 - Begin monitoring HTTP activity
 - Monitor the IP address using IDS tools
 - Monitor email
- Working with HR before an incident to establish roles, responsibilities, and HR triggers is also very important

After getting explicit approval from HR, you can start monitoring a specific employee's actions. Make note of the following items:

Equipment identification

A laptop that the suspect takes home or a desktop PC: Is a modem attached? How about a wireless access point? Are there other wireless technologies, such as EVDO, in use?

Operating system identification

This helps you identify the tools to use.

Identify IP address

If dynamic, set to a static IP so you can more easily track the system's activities.

HTTP activity

Are the intranet and internet sites visited pertinent to the suspect's duties? Are web-based email services being used?

IDS use

Monitor the traffic to/from the IP to identify inbound and outbound traffic.

Monitor email

Grab copies of email to and from the suspect. Be particularly careful to view mail offline so that you do not send an auto reply to the message originator.

- Monitor phone numbers called
- Confirm background check data
- Monitor work habits
- Perform an after-hours visit
 - What is in/on the desk?
 - What equipment don't you know about?
 - Photograph your findings
 - Create a system image
- Review collected evidence
 - Summarize your findings, what does it all add up to?

Monitor telephone calls made and received

Look for patterns in the numbers dialed. Avoid monitoring conversations unless approved by legal counsel.

Confirm background check data

Ensure a background check was performed. Review the data so you can better understand the suspect. If one was not completed, consider a background check, but realize that a full check may take several weeks.

Monitor work habits

Is the suspect working late into the evening? Or working from remote locations more often?

After-hours visit

What is the suspect storing in his or her desk? Look for items that don't belong. If possible create an image of the suspect's system using industry-recognized software such as FTK Imager Lite.

Review the data

Review any evidence you collected using forensic tools such as EnCase, and The Sleuthkit. These tools will examine allocated space as well as unallocated space, slack space, swap space, etc. Such an examination can help determine if the suspect accessed data without authorization. The examination can also reveal what else the suspect may have been doing, such as using hacking tools.

Summarize your findings

Does it look like there is an actual threat? Actual threats need to be eliminated and possibly prosecuted. Perceived threats, someone exhibiting the trigger signs noted earlier, can be transferred and/or offered employee assistance.

Course Roadmap

- Incident Handling
- **Applied Incident Handling**
- Attack Trends
- Step 1: Reconnaissance
- Step 2: Scanning
- Step 3: Exploitation
 - Gaining Access
 - Web App Attacks
 - Denial of Service
- Step 4: Keeping Access
- Step 5: Covering Tracks
- Conclusions

Applied Incident Handling

1. Espionage
2. Unauthorized Use
3. Insider Threats
4. **Legal Issues and Cybercrime Laws**
 - Lab 1.3: IR Tabletop
5. Appendix A: Intro to VMware and Linux Workshop

Let's look at the legal issues surrounding incident handling and see how they impact the incident handler's job.

- In most countries, computer crime falls into two categories
 - Traditional crimes facilitated by a computer
 - Crimes in which the computer is the target
- The Department of Justice has a portal for US cybercrime laws
- Georgetown Law Library's International and Foreign Cyberspace Law Research Guide is a great resource for international computer crime laws
- Always incorporate your organization's legal department into any incident or interaction with law enforcement

Generally speaking, in most countries, computer crimes fall into two arenas: crimes in which the computer was used to facilitate traditional criminal activity (acting as a storage or analysis device for the criminal who might be dealing in drugs, participating in organized crime, trading child pornography, or engaging in terrorist activities), and crimes in which computers are the target of the attack (where criminals break into, steal information from, or crash computers). Some criminal activity fits into both arenas at the same time.

Many people who take this class are from other countries, whose laws we cannot cover due to time constraints. To address every country for each participant in this course, we'd take up an entire week just going over cybercrime laws. Thus, if your country is not covered on the list we'll address, and you have a significant interest in the cybercrime laws of your country, feel free to contact your instructor offline during a break, asking about the cybercrime situation in your country. Your instructor may have experience in that country or may be able to refer you to someone who does.

Most US laws related to computer crimes can be accessed via <http://www.justice.gov/criminal/cybercrime/>

Georgetown Law Library's International and Foreign Cyberspace Law Research Guide is a great resource on international laws and treaties related to cybercrime. You can find the guide at <http://guides.ll.georgetown.edu/c.php?g=363530&p=4715068>

Regardless of the country you are working in, always consult with a lawyer on sensitive or potentially sensitive issues.

and started

Course Roadmap

- Incident Handling
- **Applied Incident Handling**
- Attack Trends
- Step 1: Reconnaissance
- Step 2: Scanning
- Step 3: Exploitation
 - Gaining Access
 - Web App Attacks
 - Denial of Service
- Step 4: Keeping Access
- Step 5: Covering Tracks
- Conclusions

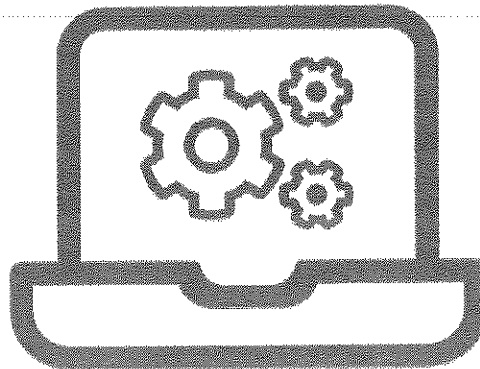
Applied Incident Handling

1. Espionage
2. Unauthorized Use
3. Insider Threats
4. Legal Issues and Cybercrime Laws
 - **Lab 1.3: IR Tabletop**
5. Appendix A: Intro to VMware and Linux Workshop

Next, to close out the day, let's do a lab on incident response tabletops.

LAB 1.3

Please work on the lab exercise *IR
Tabletop*



This page intentionally left blank.

504.1 Conclusions

- We completed the policy, procedure, and analytic fundamentals
 - These are crucial underpinnings for successful incident handling
- Tomorrow we focus on the technical attacks and defenses
 - Step-by-step
 - Offense must inform defense

So, there you have it: the incident handling process. Keep in mind that policy and procedure are absolutely essential in security, especially incident handling. Tomorrow we focus on how computer attackers undermine our systems and how to detect them and defend at each stage of the attack.

Course Roadmap

- Incident Handling
- **Applied Incident Handling**
- Attack Trends
- Step 1: Reconnaissance
- Step 2: Scanning
- Step 3: Exploitation
 - Gaining Access
 - Web App Attacks
 - Denial of Service
- Step 4: Keeping Access
- Step 5: Covering Tracks
- Conclusions

Applied Incident Handling

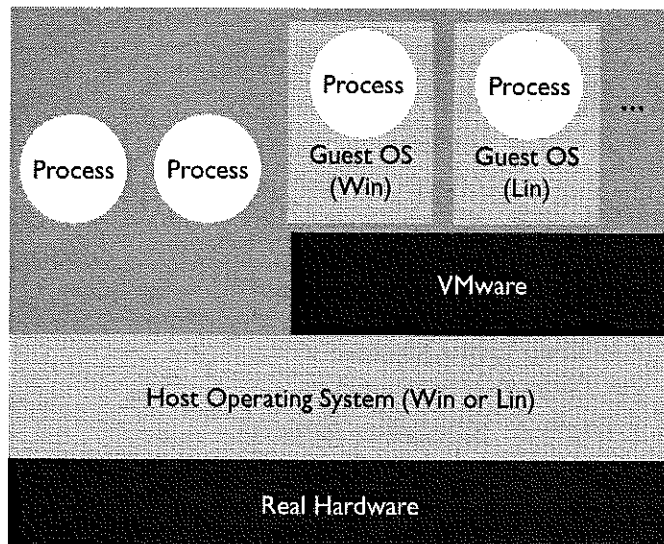
1. Espionage
2. Unauthorized Use
3. Insider Threats
4. Legal Issues and Cybercrime Laws
 - Lab 1.3: IR Tabletop
5. **Appendix A: Intro to VMware and Linux Workshop**

Next we are introduced to VMware Workstation and Player, which are virtual machine products from VMware.

What Is VMware?

Intro to VMware

- VMware is a virtual machine environment
 - Emulates various PC hardware components in software
- Single-host operating system
 - One or more guest operating systems



SEC504 | Hacker Tools, Techniques, Exploits, and Incident Handling 160

VMware is simply an emulator for a PC, all created in software. You install the VMware program on top of another operating system, known as the host operating system, such as Windows or Linux. Then you can boot up virtual computers within VMware, each of which is referred to as a guest operating system or a virtual machine. Each guest has its own memory allocation, virtualized network adapters, hard drive(s), and other hardware components. The different guests and the host appear to be truly independent operating systems, all running on the same hardware.

In my own incident handling and malware analysis activities, I rely extensively on VMware. I use it to practice hacking between virtual systems, perform forensics analysis of compromised machines, and safely test malware specimens. In fact, I don't know how I could do my job today without VMware or another similar virtual PC product.

- Virtual machines are inherently useful for
 - Incident response
 - Malware analysis
 - Digital forensics
 - Ethical hacking
 - Practice hacking
- This class uses VMware Workstation, VMware Player, or VMware Fusion
 - Other virtual machine platforms aren't officially supported, but you are free to try them

You can use VMware for so many different applications, including incident response, malware analysis, digital forensics, ethical hacking, and even practice hacking. VMware can be applied to many in-depth information security tasks.

- VMware machines consist of files in the host operating system, typically grouped into a single directory for each virtual machine
 - .vmx = Virtual machine's configuration
 - nvram = Stores the state of the virtual machine's BIOS
 - .vmdk = Stores the virtual disk file, the hard drive image(s) of the virtual machine
 - .vmss = Suspended state file, for a paused virtual machine
 - .vmsn = Snapshot file, used for taking a snapshot of the system state for restoring it later

So, what is a virtual machine inside of the VMware application? It consists of a directory with a bunch of files holding information about that guest operating system installation. The virtual files making up a VMware guest include files with these suffixes:

.vmx = Holds the virtual machine's configuration, including hardware and network settings.

nvram = Stores the state of the virtual machine's BIOS, including the BIOS boot program, clock settings, and so on.

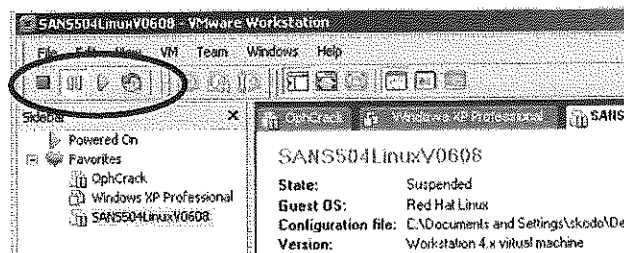
.vmdk = Stores the virtual disk file, that is, the hard drive image(s) of the virtual machine. A single virtual machine may have multiple virtual drives, so there could be more than one .vmdk file.

.vmss = Holds the suspended state file for a paused virtual machine. This suspended state includes a copy of RAM and the current console screen view.

.vmsn = Stores a snapshot file, used for, well, taking a snapshot of the system state for restoring it later. This snapshot feature is immensely useful, especially when analyzing malware. Before I run malware that could be destructive, I take a snapshot. Then, if the malware hoses the machine entirely, I can roll back to the most recent snapshot, restoring the system easily without having to rebuild!

You can back up the entire contents of a virtual machine, RAM, hard drive, and all, by simply creating a directory with a copy of all of these files. You can even zip it up for safekeeping.

- For VMware Player, invoke guest machine by simply opening it
 - When you close the Player, it suspends the guest
 - When you open that guest again, it resumes where you left off
- For VMware Workstation, using the VCR-like controls, you can
 - Stop a virtual machine
 - Suspend (pause) a virtual machine
 - Boot or resume a virtual machine
 - Reset a virtual machine (reboot)
 - Take a snapshot of a virtual machine
 - Revert to a snapshot



VMware Workstation includes VCR-like controls for running guest machines. VMware Player does not. With the Player, you can simply invoke a virtual machine by opening it. When you close the Player, it essentially suspends the guest machine. Upon reopening, the guest will be activated in the state where you last left off.

In VMware Workstation, each virtual machine can be easily controlled from within VMware using the VCR-like controls. You remember VCRs, right? You can stop a virtual machine by hitting the red Stop button. Hitting Stop, however, without gracefully shutting down the system is the equivalent of pulling the power cord on a physical system. Be careful! Your drive could get out of synch, and other problems could occur if you just hit Stop. Instead, you might want to suspend the virtual machine using the Pause button, which is akin to hibernating it. That operation is quite safe.

Also, you can boot a virtual machine by hitting the green Play button.

You can even reboot a virtual machine by hitting the red and green arrow button. Again, this is like applying a hardware reset on a real system, so be careful.

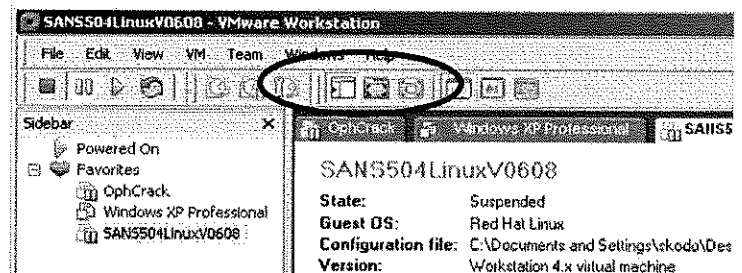
Finally, you can snapshot the virtual system or revert it to a previous snapshot using the Snapshot and Revert buttons.

- VMware Workstation supports three different screen view modes
 - **Navigation Bar Mode:** Shows favorite virtual machines
 - **Full-Screen Mode:** Virtual machine takes up entire screen
 - **Quick Switch Mode:** Provides tabs to switch between machines

TIP

Press CTRL+ALT to get out of a virtual machine.

On a Mac press Command (⌘)+CTRL



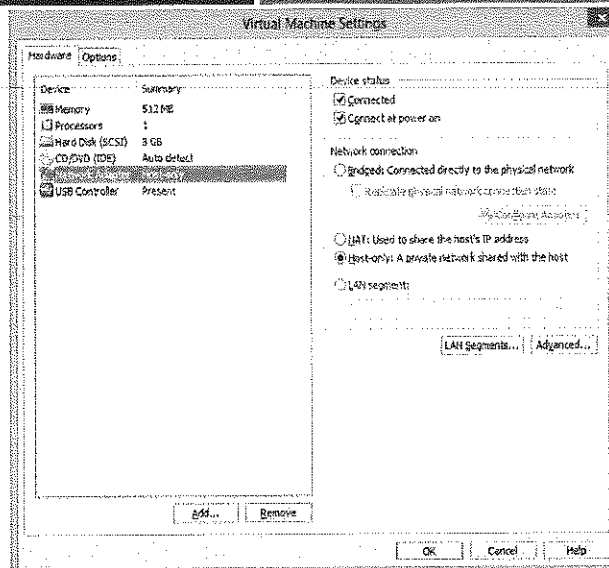
VMware Player does not offer many options for screen views. The guest is a window on top of the host operating system. When running VMware Workstation, the guest can interact with its user on the desktop using three different views:

- **Navigation Bar Mode:** Shows your favorite virtual machines defined on the system. These are essentially shortcuts to the guests that you use the most. This view is the easiest for jumping between different virtual systems running simultaneously.
- **Full-Screen Mode:** In this mode, the virtual machine takes up the entire screen. This mode makes it look like (from a user interface perspective) the virtual machine is the only thing running on the box. It hides the host operating system from the GUI entirely.
- **Quick Switch Mode:** Provides tabs to switch between machines at the top of the screen while hiding the host operating system GUI. It's like a blend of the other two modes.

Always remember that you can hit CTRL+ALT to jump out of virtual machines back into the host, regardless of the screen mode you are using. On a Mac press Command (⌘)+CTRL

Also, to send a virtual machine a CTRL+ALT+DEL, you need to go to the VMware window and select VM | Send Ctrl+Alt+Del.

- In VMware Workstation or Player
 - VM | Manage | Virtual Machine Settings
 - Or click CTRL+D
- Most useful controls:
 - CD/USB
 - Can change it to physical optical disk or mount ISO image
 - Network adapter
 - See next slide



In VMware Workstation or Player, to adjust any of the virtual hardware settings of a guest operating system, go to VM | Manage. There, you can adjust the amount of memory devoted to your virtual system. Also, you can mount a physical CD-ROM, or even an ISO image of a CD from your file system. That's particularly helpful.

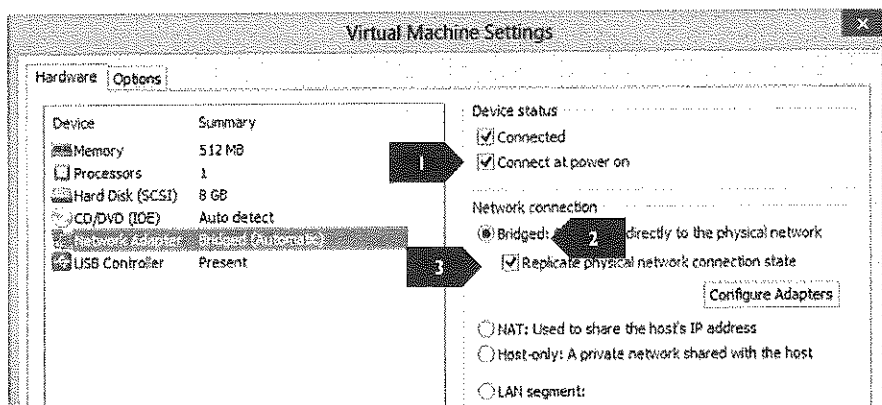
One of the most important settings in these configuration options is associated with the network adapter: your Ethernet configuration.

- Virtual machines can use one of three network options:
 - **Host-only network:** Nothing other than the host OS can get to the virtual machine across the network
 - **Bridged network:** The host and virtual machines behave as though they are sitting next to each other on a switch
 - Introduces virtual machine MAC addresses on the LAN
 - Puts host network interface in promiscuous mode (to capture traffic destined for the virtual machines)
 - **NAT:** The host acts as a NAT device, which the virtual machines sit behind

Virtual machines can use one of three network options, as follows:

- **Host-only network:** With this option, nothing other than the host OS can communicate with the virtual machine.
- **Bridged network:** With this configuration, the host and virtual machines behave as though they are sitting next to each other on a switch. This introduces the virtual machine MAC address on the LAN. Also, it puts the host network interface in promiscuous mode (to capture traffic destined for the virtual machines, the host will have to grab packets destined for MAC addresses that don't match the hardware address). Keep this in mind! I've had many students freak out when their network interface is in promiscuous mode, thinking an attacker installed a sniffer. However, in reality, these students put their interface into promiscuous mode themselves by selecting bridged networking. It's not really a security risk. Also, whenever I'm hacking from a virtual machine across a real network, I always use this mode. I don't want any network address translation to get in the way of my packet generation tools.
- **NAT:** In this mode, the host acts as a NAT device, which the virtual machines sit behind. All packets get their source IP translated so they appear to have come from the host instead of the guest operating system.

- Use bridged mode for labs that require direct network access
- In the VM settings, select Network Adapter on the Hardware tab
 - Make sure *Connect at power on* has a checkbox
 - Make sure *bridged* is selected
 - Make sure *Replicate physical network connection state* is checked



In the SEC504 and SEC560 classes, some of the labs will require your virtual machines to have direct access to a physical network.

To make this happen go to the Virtual Machine Settings (press CTRL+D) and select the Hardware | Network Adapter. Under *Device status* make sure the *Connect at power on* box is checked. If the virtual machine is currently running, also make sure the *Connected* box is checked.

Under *Network connection* make sure the *Bridged* radio button is selected and the *Replicate physical network connection state* box is checked.

- VMware is very powerful!
- Be careful with those network settings!
- Enjoy!

That's VMware. It will serve you well in this class, and I hope you find it useful on the job.

That concludes this introduction to VMware. You can either leave now or stay for the Intro to Linux mini-workshop.

- Linux is powerful but is also complex
- Still, even with little exposure to Linux, you can fully participate in the hacker tools workshop
- This course segment is designed to get you up to speed with Linux
- After this, you won't be an expert, but you'll be ready to go for the workshop
 - Our focus here is on practicality, not theory

To fully participate in this class, you need a basic working knowledge of Linux. We're not expecting you to be an expert by any means. Everything you need to know about Linux for the workshop will be covered in this introductory workshop.

We will not be covering Linux installation. You should have done that before coming to the session, as described in the class requirements.

- Bash is the default shell on many Linux distros
- Command history, accessible via up and down arrows
 - Use left and right arrows to position cursor to edit command
- Tab completion for directory and file names
 - Tab once to expand to unique
 - Tab twice to show non-unique matches
- CTRL+R to search command history
- CTRL+L to clear screen
- CTRL+C to abandon current command
- Home key to go to start of command line, End key to go to end

Throughout this session, we use bash as a command shell, one of the most common command shells in Linux distributions today. This shell includes many ease-of-use features that make interacting with Linux simpler. You should memorize each of these items, as they will save you much time and effort, making Linux a lot friendlier for you.

Bash, like many other shells, remembers your shell history, letting you access it by pressing the up and down arrows to access and edit recent commands, which you can rerun by simply pressing Enter.

After you choose a previous command, you can press the left and right arrow keys to position your cursor to edit the command.

Also, bash supports tab auto-complete for the names of directories and files. When accessing something in the file system, just press Tab for the shell to expand it to a unique name that matches what you've typed so far. If there are multiple items that match what you've typed (that is, there is nothing unique yet), you can press Tab again to show the names of all files or directories in your current working directory that match what you've typed so far. That is, Tab expands to a unique value, and Tab-Tab shows all items that match what you've typed so far if nothing is unique.

You can also search your history in bash by pressing CTRL+R at the start of a command line. Then start typing characters, and bash jumps back to the most recent command that has the characters you typed in that order. You can then press Enter to rerun that command or the left or right arrow keys to edit the command.

The CTRL+L option clears the screen, or you can simply type `clear`. The CTRL+C command lets you abandon the current command and get back to the command prompt. There is no need to delete the current command by holding down the backspace or Delete keys. Just press CTRL+C to get rid of the current command.

The Home key included on some keyboards lets you jump to the beginning of a command line, whereas the End key lets you jump to the end. These options can help you jump around in long commands to make altering them easier.

Account Stuff (logging in, useradd, passwd, su, whoami, terminal control)

- File System Stuff (structure, cd, pwd, ls, abs and rel referencing, mount, eject, mkdir, cp, find, locate, gedit, cat, less)
- Running Programs (PATH, which, ./, ps, jobs)
- Network Stuff (ifcfg-etho, restarting interfaces, ifconfig, ping, netstat)
- Building Tools (tar, configure, make)
- Other Odds and Ends (grep, man, info, shutdown)

Here's an outline describing the topics we'll cover. We'll start with Account Stuff.

- For almost all activities, you should log in as a non-root user
 - A # prompt means you are root
 - A \$ or other prompt means you aren't
- User's home directory is where that user is placed after logging in
 - Also stores that user's files

```
root@slingshot:~# useradd -d home-dir login
```

The `useradd` command adds a user, but does not set their password or create their home directory (for that use `adduser login`)

Go ahead and create a non-root account on your system.

Keep an eye on your prompt. If it's a \$, you just aren't root. If it's a #, you are root.

As root, type the following:

```
# useradd -d /home/fred fred
```

The login account "fred" will be created, with a home directory of /home/fred. The system will automatically assign a non-root userID to the account. The userID is just a number associated with this account for the purposes of assigning permissions. The home directory is where config files and other personal files for this account are stored.

By default the `useradd` command does not create the user's home directory, nor set their password and other information. However the `adduser` command will.

- Use the `passwd` command to change passwords

```
sec504@slingshot:~$ passwd
Changing password for sec504.
(current) UNIX password:
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
```

Any user can change their own password, as long as they know their current password

```
root@slingshot:~# passwd login
```

Root can change the password for any user

Currently, the fred account we created cannot be used because we haven't yet set a password. (The password isn't blank; the account is just disabled until we enter a password.) We need to set a password for the new fred account by typing:

```
# passwd fred
[type account password here]
[retype account password to verify]
```

If fred wanted to change his own password, fred would type (from the fred account):

```
$ passwd
```

Changing Accounts (sudo and whoami)

Account Stuff

- Only use root access when you truly need it
 - For most of the tools used in this class, you'll need root privs
 - If you do need root, use the sudo command
- To determine who you are currently logged in as, use whoami
- For more details, use the id command

Using sudo means you only need to remember your own password

```
sec504@slingshot:~$ sudo su -  
[sudo] password for sec504:  
root@slingshot:~#
```

```
# whoami  
root  
# id  
uid=0(root) gid=0(root) groups=0(root)
```

On many Linuxes, UID 0 accounts cannot ssh in directly. Ssh in as a regular user and use sudo

SEC504 | Hacker Tools, Techniques, Exploits, and Incident Handling 174

If you are logged in already, you run commands with the privileges of another account via the sudo command.

To get a root prompt, you could run:

```
$ sudo su -
```

Then you can type in your account's password, and if it has sudo rights to run a shell as root, you'll get a root prompt on your system.

The whoami command shows who you are currently logged in as.

Type the following:

```
# whoami
```

Given the # prompt at the beginning of this command, you will likely see root on the output. Try:

```
# su fred
```

(Notice that the prompt changed!)

```
$ whoami
```

Here, you should see that you are now fred. You can exit your most recent su by running:

```
$ exit
```

(The exit means that we are leaving the user fred and returning to root.)

```
# whoami
```

Now you should be root again (note the # prompt).

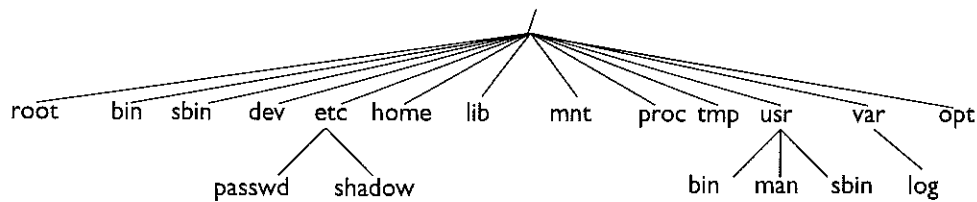
For even more details about your current user id and privileges, use the id command:

```
# id
```


- Account Stuff (logging in, useradd, passwd, su, whoami, terminal control)
- File System Stuff (structure, cd, pwd, ls, abs and rel referencing, mount, eject, mkdir, cp, find, locate, gedit, cat, less)
- Running Programs (PATH, which, ./, ps, jobs)
- Network Stuff (ifcfg-etho, restarting interfaces, ifconfig, ping, netstat)
- Building Tools (tar, configure, make)
- Other Odds and Ends (grep, man, info, shutdown)

Here's our pesky outline again. Let's cover File System Stuff next. This is the longest section simply because so much of Linux is oriented around its file system.

- The top of the file system is called /
- A bunch of things are under slash
- Here is a representative sample of what's under /
 - Varies for different versions of Linux



- Executable programs are stored in /bin and /sbin.
- /root is the root login account's home directory. This is hugely important because if you log in directly as root, this will be your initial location in the directory structure. If you log in as an individual user other than root, you'll be put in that user's directory, typically somewhere inside of /home.
- /dev stores devices (drives, terminals, etc.).
- /etc holds configuration items, like the account information (stored in /etc/passwd) and hashed passwords (stored in /etc/shadow).
- /home contains the user's home directories.
- /lib contains common libraries.
- /mnt is where various remote and temporary file systems (CD-ROMs, floppies, etc.) are attached.
- /proc is a virtual file system used to store kernel info.
- /tmp is for temporary data and is usually cleared at reboot.
- /usr holds user programs and other data.
- /var holds many different items, including logs (/var/log).
- /opt stores optional items and is often a location for specialized tools that have been added to a distribution.

Navigating the File System (cd and pwd)

File System Stuff

```
sec504@slingshot:~$ cd /tmp
sec504@slingshot:/tmp$ pwd
/tmp
```

To change directories use `cd directory`
To see where you are use `pwd`

```
sec504@slingshot:/tmp$ cd ..
sec504@slingshot:/$ pwd
/
```

The parent directory is `..` (called "dot dot").
To go up one directory type `cd ..`

```
sec504@slingshot:/$ cd ~
sec504@slingshot:~$ pwd
/home/sec504
```

To jump to your home directory type `cd ~` or
just `cd` by itself

SEC504 | Hacker Tools, Techniques, Exploits, and Incident Handling 177

If you are following along, let's change to the tmp directory:

```
$ cd /tmp
$ pwd
```

What do you see?

```
$ cd ..
$ pwd
```

What do you see?

```
$ cd ~
$ pwd
```

What do you see?

alister p...
no there's nothing there
cd
cd ~
ls

Looking at Directory Contents (ls)

File System Stuff

```
sec504@slingshot:/etc$ ls
acpi          host.conf    pm
adduser.conf hostname     pnm2ppa.conf
```

Use `ls dir` to list the contents of a directory. `ls` by itself shows what is in the current directory

The `-l` flag displays in long format: type and permissions, link count, owner, group, file size, timestamp, and name

```
sec504@slingshot:/etc$ ls -al
total 1448
drwxr-xr-x 165 root root 12288 Jun  8  2018 .
drwxr-xr-x  24 root root  4096 Jul 19  2018 ..
drwxr-xr-x  3 root root  4096 May 29  2017 acpi
-rw-r--r--  1 root root  3028 Feb 15  2017 adduser.conf
```

The `-a` flag shows all files (including files that start with a `.`)

If you are following along, type:

```
$ cd /etc
$ ls
```

What do you see?

```
$ ls -al
```

You now are looking at details associated with your `/etc` directory. System configuration information is stored here.

Handwritten notes:
ls -l shows details
ls -a shows hidden files
ls -la shows details for hidden files

- You can refer to files with their full path in the file system (absolute referencing; everything starts with "/")

```
sec504@slingshot:~$ cd /etc/init
sec504@slingshot:/etc/init$ pwd
/etc/init
```

Absolute referencing always goes to the same location, no matter where you currently are

- Or you can refer to files relative to your current working directory (everything starts assuming where you are currently located)

```
sec504@slingshot:~$ cd /etc
sec504@slingshot:/etc$ cd init
sec504@slingshot:/etc/init$ pwd
/etc/init
```

During a lab, make sure to pay attention to whether or not directories start with a /

For any file, you can refer to it using the relative reference (based on your current working directory), or the absolute reference.

Try the following, using absolute referencing for the directory:

```
$ cd /etc/init
$ pwd
```

Or you can do it in two steps, using relative references:

```
$ cd /etc
$ pwd
$ cd init ← Note that we dropped the leading /
$ pwd
```

What do you see?

- To create a new directory, use the mkdir command
- Make a temporary directory

```
sec504@slingshot:~$ cd /tmp
sec504@slingshot:/tmp$ pwd
/tmp
sec504@slingshot:/tmp$ mkdir test
sec504@slingshot:/tmp$ ls -al
total 60
drwxrwxrwt 15 root  root  4096 Mar 31 06:09 .
drwxr-xr-x 24 root  root  4096 Jul 19 2018 ..
-rw----- 1 sec504 sec504    0 Mar 31 05:08 config-err-jdKlul
drwxrwxr-x  2 sec504 sec504 4096 Mar 31 06:09 test
```

1. Change to /tmp
2. Print working directory
3. Make a directory called test
4. List detailed contents of current directory

To create a directory, use the mkdir command. Let's create a test directory in our /tmp directory.

```
$ cd /tmp
$ pwd
$ mkdir test
$ ls -al
```

What do you see?

- To find a file based on name, use *locate file*

```
sec504@slingshot:~$ locate whoami
/opt/lair-v1.0.5/npm-whoami.1
/opt/lair-v1.0.5/npm-whoami.md
/opt/lair-v1.0.5/whoami.js
/usr/bin/whoami
```

The *locate* command shows all file names that contain a string. If the OS complains about the database, type `updatedb`

- The *find* command exhaustively looks for stuff

```
sec504@slingshot:~$ find / -name whoami
/usr/bin/whoami
```

Use *find start-dir criteria* for a comprehensive search.

The *locate* command is an efficient way to determine where files are located on the system. It consults a local database installed and updated by the system administrator for files that are frequently sought. It runs quickly and doesn't consume a lot of resources. However, it cannot locate items that are not loaded into its database.

To try *locate*, type:

```
$ locate whoami
```

If your system complains that there isn't a *locate* database or that it's out of date, you can manually update the database by typing the command:

```
# updatedb
```

To do a comprehensive search of the directory, you can use the *find* command. This command consumes a lot of resources. Several *finds* running simultaneously will slow a Linux system to a crawl. Still, *find* is the best way to find something if *locate* doesn't work.

Let's try to find a file on the file system. Type the following:

```
$ find / -name whoami
```

What do you see?

- There are several editors included in most Linux variants:
 - vi, gnu-emacs, pico, mcedit, gedit
- For new users, gedit is easy to learn (and powerful!)

```
sec504@slingshot:/tmp$ cd ~
sec504@slingshot:~$ gedit test_file
```

Use `gedit file` for a text editor similar notepad



You may need to edit a file at some point. You can use any editor you are comfortable with. If you are new to Linux, you should consider using gedit, one of the easiest editing tools commonly installed in Linux. If you have a GUI, you can use gedit.

Let's create and edit a file:

```
$ cd ~                (change to the home directory)
$ gedit test_file     (let's edit and create a file named "test_file")
```

Now, edit your file. Type in a bunch of junk. Use the function keys to save it.

I told you gedit was easy!

Viewing File Contents (cat, head, and tail)

File System Stuff

```
sec504@slingshot:~$ cat ~/test_file
Using gedit is easy and fun!
sec504@slingshot:~$ cat /etc/passwd
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sec504@slingshot:~$ head /etc/passwd
root:x:0:0:root:/root:/bin/bash
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sec504@slingshot:~$ tail -n 2 /etc/passwd
fred:x:1003:1004::/home/fred:
mike:x:1004:1005::/home/mike:/bin/bash
```

Use `cat file` to display the contents of a file.

If there is too much on the screen, the `head` and `tail` commands allow you to see the first and last few lines of a file.

Use `-n num` to show `num` lines (also works for the `head` command)

SEC504 | Hacker Tools, Techniques, Exploits, and Incident Handling 183

So, you just edited a file. How can you see its contents? You can use the `cat` command:

```
$ cat ~/test_file
```

Also, you can look at other files:

```
$ cat /etc/passwd
```

This shows the contents of the password file! (Note that on most Linux installations, the passwords are stored in another file, `/etc/shadow`). Typically, in most modern UNIX installations, `/etc/passwd` just contains account information.

Alternatively, we can view portions of files using the `head` or `tail` commands. The `head` command shows the first 10 lines of a file by default. By specifying `head -n [n] [filename]`, we can view just the first `n` lines. Similarly, the `tail` command shows the last 10 lines of a file by default, or we can use the `-n [n]` syntax to view a different number of trailing lines. Consider the following commands:

```
$ head /etc/passwd
```

```
$ head -n 1 /etc/passwd
```

```
$ tail -n 2 /etc/passwd
```

- Use `less filename` to view files that are larger than one screen
 - Use up and down arrow keys to scroll through a file
 - Can also pipe the output of commands to less

```
sec504@slingshot:~$ less test_file
Using gedit is easy and fun!
test_file (END)
sec504@slingshot:~$ ls /dev
agpgart  loop1  snapshot  tty33  tty7  ttyS8
autofs   loop2  snd       tty34  tty8  ttyS9
block    loop3  sr0       tty35  tty9  uhid
sec504@slingshot:~$ ls /dev | less
agpgart
autofs
```

To quit less
press q

Use the | operator to feed the
output of a command into less

In addition to `cat`, there are other commands you can use to look at files. The `less` command is one of the best to use. Try typing:

```
$ less test_file
```

You should see the contents of the file.

In addition to looking at files, the `less` command can also be used to help look at lengthy output from a command. Try typing:

```
$ ls /dev
```

This shows you all the devices (virtual and otherwise) on your system. It's a long, unwieldy list. The `less` tool lets you interact with this output in a better way.

Type:

```
$ ls /dev | less
```

By piping the output of `ls` through `less`, you can now use the cursor keys to scroll up and down through the output. The space key jumps forward one page. Use the `q` key to quit. The pipe takes the output of one program and feeds it into the standard input of another program.

- Account Stuff (logging in, useradd, passwd, su, whoami, terminal control)
- File System Stuff (structure, cd, pwd, ls, abs and rel referencing, mount, eject, mkdir, cp, find, locate, gedit, cat, less)
- ▶ Running Programs (PATH, which, ./, ps, jobs)
- Network Stuff (ifcfg-etho, restarting interfaces, ifconfig, ping, netstat)
- Building Tools (tar, configure, make)
- Other Odds and Ends (grep, man, info, shutdown)

You guessed it . . . another outline slide. We will now discuss running programs. This section is important (not that the other ones aren't important). People frequently mess up on this stuff and get confused because Linux works differently from Windows in running programs.

- When you run a program, tries to find the program in the list of directories called your path
 - Accessible as the `$PATH` environment variable
- To see which directory a program runs from, use `which program`

```
sec504@slingshot:~$ echo $PATH
/home/sec504/bin:/home/sec504/.local/bin:/usr/local/
sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/u
sr/games:/usr/local/games:/snap/bin:/opt/JohnTheRipp
er/run:/opt/metasploit-4.11:/opt/course_www/vsagent-
504:/home/sec504/go/bin
sec504@slingshot:~$ which ls
/bin/ls
```

The directories in `$PATH` are separated by colons

When you type a command at the prompt, the system looks in your PATH to find the right program to run.

Look at your path by typing:

```
$ echo $PATH
```

The `echo` command means "type the following." The `$` before path means, "What follows isn't a string of characters; it's a variable." The variable we want to type is our PATH.

The result is a list of directories where the system searches for programs based on what we type at the command line. These directories are separated by a ":". If you type a program name at a command prompt, and the program isn't in your PATH, the system will tell you that it cannot find the program. You have to either refer to it absolutely or relatively or add its directory to your PATH.

If you want to see where in your PATH a command has been found, you can use the `which` command. Try typing:

```
$ which ls
```

That's where your `ls` program really is!

- Note that the current directory "." is not in your path!
 - This is good because you cannot be tricked into running a trojan horse
 - Think what would happen if I created a backdoor named ls

```
sec504@slingshot:~$ cp /bin/nc ~/superevilbackdoor
sec504@slingshot:~$ superevilbackdoor
superevilbackdoor: command not found
sec504@slingshot:~$ ./superevilbackdoor
Cmd line: ^C
sec504@slingshot:~$ /home/sec504/superevilbackdoor
Cmd line:
```

To run a program not in your path, use relative referencing: `./program`
You can also type the absolute path

This is an important point that confuses people because UNIX functions differently from Windows on this issue.

For security reasons, your current working directory (the one shown by `pwd`), also referred to as ".", is not in your PATH. That's a good thing! If "." were in your path, an evil attacker could name an evil trojan horse program `ls` and put it in your home directory. When you ran `ls` to look at your home directory's contents, you'd run the evil trojan horse! For this reason, "." isn't in the path by default and shouldn't be put in your path.

This also means that if you change directories to a place in which a program file is located, you cannot just type the program's name to run it. Instead, to run the program, you have to type `./program` to run it.

If the system ever complains that it cannot find a file but you can see the file in the current working directory using `ls`, you likely just need to start the program by typing:

```
$ ./program
```

On Windows machines, the current working directory is in your path. Therefore, if you change to a directory with an executable and type the executable's name on Windows, the program runs. Yes, it's convenient . . . However, it's a security hole!

- Updating your path changes the directories searched through for executables

```
sec504@slingshot:~$ PATH=$PATH:/home/sec504
sec504@slingshot:~$ superevilbackdoor
Cmd line:
```

Note the colon before
/home/sec504

This only affects the path
for the session you run it in

```
sec504@slingshot:~$ gedit ~/.bash_history
```

To make permanent changes edit
the ~/.bash_history file
(generally not necessary)

Although we DO NOT recommend it, you could add a directory to your PATH temporarily. Type the following:

```
$ echo $PATH
```

Look at your path. To change your path temporarily, you could type (NOT RECOMMENDED):

```
$ PATH=$PATH:/another_directory
```

Now type:

```
$ echo $PATH
```

Your path will now include the additional directory at its end.

This change applies only to this terminal and any processes started from this terminal. When you log out, this change goes away, and your path has its original settings.

To permanently change your path, you must edit the ~/.bash_profile file. I advise you to avoid editing this file if you are new to Linux. The default path setting is good for most purposes.

- To see running processes use `ps`
 - (sort of like the Windows Task Manager)

```
sec504@slingshot:~$ ps aux | less
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root         1  0.0  0.4 189480 10016 ?        Ss   06:46   0:02 /sbin/init
root         2  0.0  0.0      0      0 ?        S    06:46   0:00 [kthreadd]
root         3  0.0  0.0      0      0 ?        S    06:46   0:00 [ksoftirqd/0]
```

Unlike Task Manager, `ps` doesn't continuously update. However the `top` command does.

Sometimes you need to see what processes are running. The `ps` command shows you a bunch of info about all running processes. Try typing:

```
$ ps aux
```

You get an exhaustive (and exhausting) list of all running processes.

Let's use our little "less" trick to make this output more readable:

```
$ ps aux | less
```

Now you can scroll up or down and get a better feeling for what's running on your system.

Unlike Task Manager on Windows, the `ps` command does *not* update continuously. However the `top` command does.

- At a single command prompt, you can run and control multiple programs simultaneously

CTRL+C kills a program, CTRL+Z pauses a program and returns the prompt

```
sec504@slingshot:~$ find / -name ls
/bin/ls
^C
sec504@slingshot:~$ find / -name ls
/bin/ls
^Z
[1]+  Stopped
sec504@slingshot:~$ bg
[1]+ find / -name ls &
sec504@slingshot:~$ /usr/lib/klibc/bin/ls
[1]+  Exit 1          find / -name ls
```

The bg command allows a paused program to continue running in the background

You can temporarily pause programs with CTRL+Z and get your command prompt back. This is quite useful, because you can run more programs if you want.

Also, you can restart the paused program running in the background with the bg command. The fg command starts it running in the foreground, as you might expect.

Let's try it. Type:

```
$ find / -name ls
```

Before it finishes running, press CTRL+Z.

Now restart the program in the background by typing:

```
$ bg
```


More Job Control: &, jobs, and fg

Running Programs

The jobs command lists programs running in the background

```
sec504@slingshot:~$ nc -nlp 2600 &
[1] 2848
sec504@slingshot:~$ nc -nlp 4444 &
[2] 2849
sec504@slingshot:~$ jobs
[1]- 2848 Running                  nc -nlp 2600 &
[2]+ 2849 Running                  nc -nlp 4444 &
sec504@slingshot:~$ fg 1
nc -nlp 2600
```

The & runs a program in the background and returns the prompt

The fg command brings a job to the foreground
Default is the most recent job sent to the background

The jobs command gives you a list of all programs you have kicked off that are running in the background. The fg command can also be used to restart a specific paused program in the foreground by giving the job number after the fg command.

If the find command from the previous slide has finished, type the same command again, but this time run it in the background using the & after the command invocation.

```
$ nc -nlp 4444 &
```

As it runs in the background, type the jobs command:

```
$ jobs
```

Look at the job running. You can move it to the foreground by typing:

```
$ fg 1
```

- Account Stuff (logging in, useradd, passwd, su, whoami, terminal control)
- File System Stuff (structure, cd, pwd, ls, abs and rel referencing, mount, eject, mkdir, cp, find, locate, gedit, cat, less)
- Running Programs (PATH, which, ./, ps, jobs)
- ▶ Network Stuff (ifcfg-eth0, restarting interfaces, ifconfig, ping, netstat)
- Building Tools (tar, configure, make)
- Other Odds and Ends (grep, man, info, shutdown)

Another outline slide. Gee, these outlines are fun.

Now we will cover getting and staying networked in Linux.

- Network configuration is in the file `/etc/network/interfaces`
 - It's text, so use your favorite editor, such as `gedit`
 - Can set interfaces to static or DHCP, specific IP addresses, netmasks, etc.
- To apply changes, restart networking services

```
sec504@slingshot:~$ sudo su -
[sudo] password for sec504:
root@slingshot:~# gedit /etc/network/interfaces
** (gedit:2933): WARNING **: Set document metadata
failed: Setting attribute metadata::gedit-position
not supported
root@slingshot:~# service networking restart
root@slingshot:~#
```

You can safely ignore the `gedit` warnings

Notice there is no output if there are no errors

To set your network interface options in Linux, you can edit the `/etc/network/interfaces` file. By putting in the appropriate information, you can configure your interface for static addresses or `dhcp`.

If you change the interfaces file, your changes will not be applied to the interface immediately. Instead, you need to restart your interface. Type (as root):

```
# service networking restart
```

eth0 is the ethernet adapter and lo is the loopback adapter

```
sec504@slingshot:~$ ifconfig
eth0  Link encap:Ethernet  HWaddr 00:0c:29:23:04:5e
      inet addr:10.10.75.1  Bcast:10.10.255.255  Mask:255.255.0.0
lo    Link encap:Local Loopback
      inet addr:127.0.0.1  Mask:255.0.0.0
sec504@slingshot:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKN...
   link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
   inet 127.0.0.1/8 scope host lo
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_...
   link/ether 00:0c:29:23:04:5e brd ff:ff:ff:ff:ff:ff
   inet 10.10.75.1/16 brd 10.10.255.255 scope global eth0
```

Most modern systems support the ip command

Let's see if our interface changes were applied to the system. To look at your interface configuration, type:

```
# ifconfig
```

You see your IP address, netmask, MAC address, and various other nifty items. If you have one ethernet card, you see two interfaces, the local loopback interface with the address 127.0.0.1 and your ethernet interface, called eth0.

Instead of using ifconfig, you can also use the ip command with the a (for address) option on most modern Linux systems.

```
# ip a
```

- Ping sends ICMP Echo Request messages to another host
 - Prints out whether it gets a response
 - You can use it to verify that you are properly networked

Unlike on Windows, ping on Linux keeps going until you press CTRL+C

```
sec504@slingshot:~$ ping 127.0.0.1
PING 127.0.0.1 (127.0.0.1) 56(84) bytes of data:
64 bytes from 127.0.0.1: icmp_seq=1 ttl=64 time=0.098 ms
64 bytes from 127.0.0.1: icmp_seq=2 ttl=64 time=0.071 ms
^C
--- 127.0.0.1 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1001ms
rtt min/avg/max/mdev = 0.071/0.084/0.098/0.016 ms
```

To verify that you are properly networked, you can ping another machine. The ping command is similar, but not identical, to the Windows ping program. One of the biggest differences is that a Linux ping keeps sending pings until you press CTRL+C to stop it. By default, the Windows ping sends out four ICMP Echo Request packets and then stops. Linux just keeps going until you stop it.

- To show information about network usage use `netstat`
 - It can show routing tables, current connections, and listening ports
 - Can also use `lsof -i` or `ss`

```
root@slingshot:~# netstat -nap
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address   Foreign Address State  PID/Program name
tcp        0      0 127.0.0.1:3306  0.0.0.0:*       LISTEN 1004/mysqld
tcp        0      0 0.0.0.0:80     0.0.0.0:*       LISTEN 1013/nginx -g...
tcp        0      0 0.0.0.0:22     0.0.0.0:*       LISTEN 990/sshd
```

Look for LISTEN and
ESTABLISHED

Now look at what's using your various TCP and UDP ports. Type:

```
$ netstat -nap
```

There's a lot of stuff there. It can be a bit difficult to read as it scrolls by, so try this:

```
$ netstat -nap | less
```

You can scroll up and down through the output. We'll discuss how to do better searches through this later.

Note that various TCP and UDP ports are shown as LISTEN. These are waiting for a connection. Others may indicate that they are ESTABLISHED. These have existing connections.

You can also use the command `lsof -i` or the `ss` command from the `iproute2` suite of tools.

- Account Stuff (logging in, useradd, passwd, su, whoami, terminal control)
- File System Stuff (structure, cd, pwd, ls, abs and rel referencing, mount, eject, mkdir, cp, find, locate, gedit, cat, less)
- Running Programs (PATH, which, ./, ps, jobs)
- Network Stuff (ifcfg-etho, restarting interfaces, ifconfig, ping, netstat)
- ▶ Building Tools (tar, configure, make)
- Other Odds and Ends (grep, man, info, shutdown)

You know that to use Linux in the workshop, you have to run tools. To run them, in many cases you need to build and install them. As you see from the outline slide, we'll cover building and installing tools next.

Some programs are installed by using tar files. Others are RPMs. Still others use "configure" and "make." How do you know which tools use which format? Most of the tools include a README file. Look at the README file (using cat, less, or gedit) for instructions on installing the tool. The course notes for the main class also include directions for compiling and installing.

By the way, we have you compile and install the tools so that you can get experience with doing these tasks. In the wild, you may need to compile and install new versions of these and other tools, so we want to get you ready.

- If a file name ends in `.tar`, it is a tape archive image
- If a file name ends in `.tar.gz` or `.tgz` it is also compressed
- Extract using the `tar` command

```
sec504@slingshot:~# tar xvf file.tar
```

x means extract, v means verbose, f means read from a file

```
sec504@slingshot:~# tar zxvf file.tgz
```

z means unzip the file before extracting

Some files are stored as tape archives, abbreviated tar. This doesn't mean they were on physical tapes; the lingo just lingers from the olden days. Although tapes may or may not be used, tar files are used all the time. Think of them as being like ZIP archives in Windows. You take a bunch of files and glom them together in a tar file.

To open a tar file, you use the `tar` command with the `xvf` parameters. `x` means *extract*. `v` means *be verbose; give me a lot of output to let me know what's going on*. `f` means *get this from a file*.

If the tar file has been compressed using a tool called `gzip`, its name will end with a suffix of `.tar.gz` or simply `.tgz`. To open these, you need to use the `tar` command with the `xvf` and `z` flags. The `z` flag means *unzip this before you open the archive*.

When the archive opens, all files and directories associated with it will be automatically created in the current working directory and below.

We won't demo this during the Intro to Linux mini-workshop. You'll get a chance to use tar files during the main class.

- Some tools you need to compile yourself
 - Many tools include a script, typically called `configure`, to determine if dependencies are installed, where libraries are located, etc.
 - The `make` program is often used to compile, and then install the tool

```
root@slingshot:/opt/nmap-latest# ./configure
checking whether NLS is requested... yes
checking for gcc... gcc
checking whether the C compiler works... yes
root@slingshot:/opt/nmap-latest# make
Compiling liblua
make[1]: Entering directory '/opt/nmap-6.47/liblua'
root@slingshot:/opt/nmap-latest# make install
```

Although some programs ship as tar files and others as RPMs, many just ship with a script called `configure`. You need to run this script first, which checks your environment and creates a set of options necessary to get the tool compiled on your device. After running `configure`, you run the `make` command, which compiles and builds the tool. Then, by typing `make install`, the program is loaded into the appropriate place.

We won't demo this during the Intro to Linux mini-workshop. You'll get a chance to use `configure` and `make` during the main class.

- For some tools, there is no configure script
 - You simply use the `make` program to compile it
 - Or compile it yourself using `gcc`

Syntax:

`gcc -o output sourcecode.c`

```
sec504@slingshot:~$ gcc -o notabackdoor superevilbackdoor.c
sec504@slingshot:~$ ./notabackdoor
```

Some tools don't have a configure script. For these, you just run the `make` command.

Some tools don't even use `make`. Instead you compile them by hand using a compiler. The syntax to compile source code using the `gcc` (GNU C Compiler) tool is:

```
$ gcc -o output sourcecode.c
```

(Make sure to substitute the executable file name you want to create for `output`, and the name of the file that contains the source code for `sourcecode.c`)

- Account Stuff (logging in, useradd, passwd, su, whoami, terminal control)
 - File System Stuff (structure, cd, pwd, ls, abs and rel referencing, mount, eject, mkdir, cp, find, locate, gedit, cat, less)
 - Running Programs (PATH, which, ./, ps, jobs)
 - Network Stuff (ifcfg-eth0, restarting interfaces, ifconfig, ping, netstat)
 - Building Tools (tar, configure, make)
- Other Odds and Ends (grep, man, info, shutdown)

Here are some other odds and ends that can help us use Linux throughout this course.

- The `grep` command filters text that matches a pattern

```
sec504@slingshot:~$ cd /etc
sec504@slingshot:/etc$ grep root *
grep: acpi: Is a directory
aliases:postmaster: root
aliases:nobody: root
sec504@slingshot:/etc$ grep -ir root *
cups/cups-files.conf:#RequestRoot /var/sp
cups/cups-files.conf:#ServerRoot /etc/cups
```

The * means
all files

The `i` means ignore case, the
`r` means search recursively

Grep is a powerful tool for finding data. It can look through files or the output from commands to identify particular strings. We will just scratch the surface of its use.

To look for a given string in a set of files in a directory, you can type:

```
$ grep root *
```

This prints all occurrences of the word "root" and the file in which it appears in the current working directory. Try the following:

```
$ cd /etc
```

```
$ grep root *
```

See the word "root" in any files here? Which ones?

By default, `grep` is case sensitive. This means the strings `root`, `rOoT`, and `Root` are treated differently. To make `grep` ignore case use the `-i` option. Also `grep` by default does not recurse into subdirectories. To make `grep` search subdirectories use the `-r` (recursive) option. You can specify the options individually (`-i -r`) or combined together as shown below.

```
$ grep -ir root *
```

Search for things listening on port 7777

```
sec504@slingshot:~$ nc -nlp 7777 &
[1]: 28412
sec504@slingshot:~$ netstat -nap | grep 7777
tcp    0    0  0.0.0.0:7777    0.0.0.0:*    LISTEN  28412/nc
sec504@slingshot:~$ ps aux | grep nc
sec504  28412  0.0  6496 1832 pts/0  S   23:50   0:00 nc -nlp 7777
sec504  28417  0.0 1427 1088 pts/0  S+  23:51   0:00 grep --color=auto nc
```

Search for processes named nc

grep shows lines that contain a pattern, so you may see more than expected

Grep can help isolate information about the usage of particular ports and processes. First start a netcat listener on port 7777 (so we have something to search for).

```
$ nc -nlp 7777 &
```

At the command prompt, type:

```
$ netstat -nap | grep 7777
```

This says, "Run the netstat command to show me TCP and UDP port usage, send the output to grep and have grep show me any lines with the string 7777 in it." The results indicate if anything is listening on or using port 7777.

Likewise, you can use grep to help you find particular programs. At a command prompt, type:

```
$ ps aux | grep nc
```

This shows you all processes running the nc program on your system.

You might see more output than expected when using grep. This is because grep searches for a pattern in lines of text. The grep tool doesn't try to interpret the output of an arbitrary command. So it doesn't know you want a port number, or just the nc process (instead of the grep command you ran to search for nc processes).

- The man and info commands show detailed usage information for other commands

```
sec504@slingshot:~$ man ls
LS(1)                                User Commands                                LS(1)

NAME
ls - list directory contents
```

To exit the man or info pages, press q

```
sec504@slingshot:~$ info ls
10.1 'ls': List directory contents
=====

The 'ls' program lists information about files (of any type,
```

To learn more about Linux, you can use man or info.

Try the following:

```
$ man ls
```

Interesting . . . and chilling. The ls command is complex!

Also, try:

```
$ info ls
```

And, check this out to learn more about man:

```
$ man man
```

- For a one-line summary of a command use `whatis command`
- The command `apropos keyword` searches the man pages

```
sec504@slingshot:~$ whatis ifconfig
ifconfig (8) - configure a network interface
sec504@slingshot:~$ apropos network
interfaces (5) - network interface configuration for ifup and ifdown
aseqnet (1) - ALSA sequencer connectors over network
avahi-autoipd (8) - IPv4LL network address configuration daemon
```

The `apropos` command is equivalent to
`man -k keyword`

The `whatis` command is useful for getting hints from the system about what various commands do. It won't change your life, but it might just jog your memory about some esoteric command.

I usually just use the main page, but some people prefer `whatis`.

Try typing:

```
$ whatis ifconfig
```

You can also use the `apropos` command to search for topics and the commands related to those topics:

```
$ apropos network
```

This is the equivalent of `man -k` to look up something by keyword, as in:

```
$ man -k network
```

- Can shut down (or reboot) at the GUI or command line
 - Some Linuxes require you to be root to use the command line
 - Your Slingshot VM does not

```
sec504@slingshot:/tmp$ shutdown -h now
```

The **h** means halt and shut down

```
sec504@slingshot:/tmp$ shutdown -r now
```

The **r** means reboot, can also type
reboot

TIP

You can specify a time to shut down, such as
`shutdown -h +10M`
to shut down in 10 minutes from now

When you are done with Linux, you should shut it down gracefully. You can do this from the GUI, but I usually just do it from the command prompt.

Some Linux systems require you to be root to run the command-line tools to shut down your system. Your Slingshot VM however does not. To gracefully shut down your system, type:

```
$ shutdown -h now
```

The `-h` flag means "halt" the system. Of course, "now" means do it right away. You can actually schedule the system to shut down at another time using this command, too. For example, to shut down in 10 minutes from:

```
$ shutdown -h +10M
```

You can also use the shutdown or reboot command to reboot the machine by using `-r` instead of `-h`. To reboot, I usually just type:

```
# reboot
```


- You have the building blocks you need to participate in the full class
- Linux is powerful but sometimes frustrating
- Refer to this section during the main class
- Ask for help from instructors, teaching assistants, and mentors if required

You are now ready for the full class Linux labs! Use your newfound Linux skills for good, not evil!

Course Resources and Contact Information



AUTHOR CONTACT

Mike Murr
mike@socialexploits.com



SANS INSTITUTE

11200 Rockville Pike, Suite 200
N. Bethesda, MD 20852
301.654.SANS(7267)



PEN TESTING RESOURCES

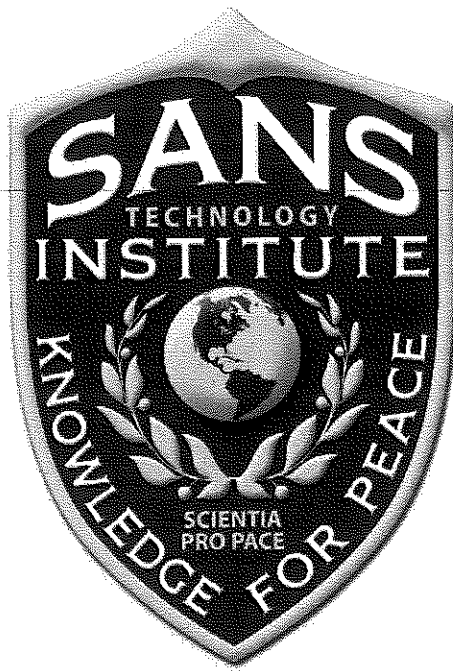
pen-testing.sans.org
Twitter: @SANSPen Test



SANS EMAIL

GENERAL INQUIRIES: info@sans.org
REGISTRATION: registration@sans.org
TUITION: tuition@sans.org
PRESS/PR: press@sans.org

This page intentionally left blank.



This Course Is Part of the SANS Technology Institute (STI) Master's Degree Curriculum.

If your brain is hurting from all you learned in this class but you still want more, consider applying for a master's degree from STI. We offer two hands-on, intensive master's degree programs:

- Master of Science in Information Security Engineering
- Master of Science in Information Security Management

If you have a bachelor's degree and are ready to pursue a graduate degree in information security, visit www.sans.edu for more information.

www.sans.edu

855-672-6733

info@sans.edu

"As usual, SANS courses pay for themselves by Day 2. By Day 3, you are itching to get back to the office to use what you've learned."

Ken Evans, Hewlett Packard Enterprise - Digital Investigation Services

SANS Programs
sans.org/programs

GIAC Certifications
Graduate Degree Programs
NetWars & CyberCity Ranges
Cyber Guardian
Security Awareness Training
CyberTalent Management
Group/Enterprise Purchase Arrangements
DoDD 8140
Community of Interest for NetSec
Cybersecurity Innovation Awards

SANS Free Resources
sans.org/security-resources

- E-Newsletters
 - NewsBites: Bi-weekly digest of top news
 - OUCH!: Monthly security awareness newsletter
 - @RISK: Weekly summary of threats & mitigations
- Internet Storm Center
- CIS Critical Security Controls
- Blogs
- Security Posters
- Webcasts
- InfoSec Reading Room
- Top 25 Software Errors
- Security Policies
- Intrusion Detection FAQ
- Tip of the Day
- 20 Coolest Careers
- Security Glossary



Search SANSInstitute

SANS Institute
11200 Rockville Pike | Suite 200
North Bethesda, MD 20852
301.654.SANS(7267)
info@sans.org