

530.5

Zero Trust Architecture: Addressing the Adversaries Already in Our Networks

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

SANS

THE MOST TRUSTED SOURCE FOR INFORMATION SECURITY TRAINING, CERTIFICATION, AND RESEARCH | sans.org

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

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

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

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

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

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

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

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

PMP and PMBOK are registered marks of PMI.

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

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

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



Zero Trust Architecture: Addressing the Adversaries Already in Our Networks

© 2019 Eric Conrad, Justin Henderson, & Ismael Valenzuela | All Rights Reserved | Version E01_01

Welcome to SEC530.5, Zero Trust Architecture!

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

Table of Contents

	Page
Zero Trust Architecture	3
Credential Rotation	13
Securing Traffic	25
EXERCISE: Network Isolation and Mutual Authentication.....	45
Host-Based Firewalls	46
Network Access Control (NAC)	52
Segmentation Gateways	69
Security Event Information Management (SIEM)	81
EXERCISE: SIEM Analysis and Tactical Detection	89
Log Collection	90
Audit Policies	113
Host Hardening	133
Patching	151
Tripwires and Red Herring Defenses	158
EXERCISE: Advanced Defense Strategies	176

530.5 Table of Contents

This table of contents outlines our plan for 530.5.

Course Roadmap

- Day 1: Defensible Security Architecture
- Day 2: Network Security Architecture
- Day 3: Network-Centric Application Security Architecture
- Day 4: Data-Centric Application Security Architecture
- **Day 5: Zero Trust Architecture**
- Day 6: Capstone: Design, Detect, Defend

CURRENT STATE ASSESSMENT, SOCS, AND SECURITY ARCHITECTURE

1. **Zero Trust Architecture**
2. Credential Rotation
3. Securing Traffic
4. **EXERCISE: Network Isolation and Mutual Authentication**
5. Host-Based Firewalls
6. Network Access Control (NAC)
7. Segmentation Gateways
8. Security Event Information Management (SIEM)
9. **EXERCISE: SIEM Analysis and Tactical Detection**
10. Log Collection
11. Audit Policies
12. Host Hardening
13. Patching
14. Tripwires and Red Herring Defenses
15. **EXERCISE: Advance Defense Strategies**

Course Roadmap

The next section covers the concept of Zero Trust Architecture.

Perimeter Security

Perimeter security is basic in principal

- **External** access is **untrusted**
- **Internal** access is **trusted**

Perimeter security has a major failing

- Inside access is not always benign
- Modern attacks are inside out
- Trusted system brings attacker in

Internal access is loosely regulated



Perimeter Security

Today security measures often focus on perimeter defenses. The concept is simple. Connections oriented from outside the network or VPN are untrusted, and connections within the network or from the VPN are trusted. And yet, this logic makes one critical mistake. Internal access should not be trusted.

Modern attacks such as phishing attacks trick internal assets into inviting an adversary into your network. With this model, once an adversary is on the inside they become trusted. While there might be some degree of least privilege internally the level of defense in depth is sorely lacking. Detection and prevention are not designed for a win against internal to internal attacks.

Zero Trust Architecture

Developed by Forrester's John Kindervag in 2010¹

- Data-centric focus

Basic principles of zero trust:

- Network is always hostile
- Internal and external threats are always present
- Internal network is not sufficient to equal trusted
- Every device, user, and network flow must be proven
- Log and inspect all traffic

Zero Trust Architecture

Prior books focused heavily on securing traffic on the network. While network-based solutions will still be discussed in this book, the focus is shifting towards securing what matters most to an organization. Often times, this is data. Whether the data is large amounts of credit card numbers or intellectual property, data needs to be understood and secured.

A zero trust architecture takes this to an extreme. The core concepts are that nothing is to be trusted and thus security must be designed to authenticate and secure all connections. With zero trust, the network and endpoints are treated as potential enemies, and all access must truly be verified and authorized. Part of the zero trust core is that all authorized connections must be known or come from expected conditions. Therefore, all traffic must be logged and inspected for verification.

[1] <https://www.forrester.com/search?N=21061+10001&sort=3&everything=true&source=browse>

Need for Zero Trust

Data security is compromised by multiple factors

- Rogue devices
 - Mobile devices / BYOD
 - Remote users
 - Business partners and contractors
- Compromised internal assets
- Insider threats
 - Accidental or intentional



Need for Zero Trust

The truth is any organization has previously had or is compromised at some level. Whether by automated malware or a targeted adversary the truth is it is inevitable. Rather than using this as fear uncertainty and doubt (FUD) use it as a truth to shed light on your internal controls.

Think about this a different way. Has your organization ever had unauthorized devices on its network? If you have never looked for unauthorized devices, just nod your head and say yes. How about this, does your organization have devices on your network that you cannot guarantee the asset's configuration or software? What are the odds you do not have 100% asset control and configuration? Let's think of another angle. Do your business partners or contractors truly have least privilege access that is only allowed during times of need?

The discussion is not over yet. What about your end users? Are they superhuman in that they always do what they should and never go to sites that are not necessary for business purposes? Are you sure they are all trustworthy? If you answered yes to all of the questions, then why are you in SEC530? The truth is everyone knows internal assets get compromised. The problem is our architectures do not reflect it.

Zero Trust Networks¹

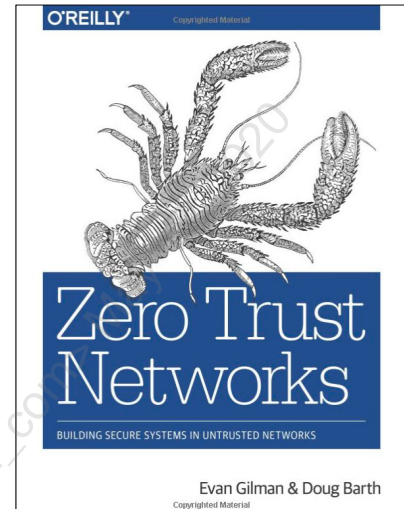
Zero trust concept is a strategy

- Strategy may be tough to swallow

Zero Trust Networks¹ goes deep into concepts and logical barriers

- 530 focus is on understanding the zero trust model
- Then practical application of it

Full zero trust is unlikely to be achieved



Zero Trust Networks¹

Treating all network access as untrusted and implementing verification for everything seems pretty difficult. It seems difficult because it is difficult. In fact, organizations are unlikely to implement zero trusts fully. The reason is not that it is impossible but rather because technologies and products today are still heavily designed for perimeter security. The book *Zero Trust Networks¹* goes heavily into detail on all the logical concepts and barriers to implementation and is a recommended read.

While Forrester founded the zero trust model there are different approaches to implementation based upon how granular the concept of zero trust is adopted. In truth, full zero trust implementation is unlikely and unpractical due to time and resource considerations. A balance of treating the network and systems as untrusted needs to be found. The mindset of zero trust alone allows a much more robust architecture so long as it is applied.

[1] <https://www.amazon.com/Zero-Trust-Networks-Building-Untrusted/dp/1491962194>

Zero Trust Mandates

1. All traffic must be secured
 - Traffic must be authenticated
 - Traffic must be encrypted
2. Least privilege must be enforced
 - Trust must be factored into least privilege
 - Trust is no longer binary (yes or no)
3. All data flows must be known and controlled
4. All assets must be scanned, hardened, and rotated

**Trust Nothing
Verify Everything**

Zero Trust Mandates

Zero trust falls into four specific mandates. First off, if the network is considered hostile, then it is necessary to secure traffic with authentication and encryption. Encryption provides confidentiality and authentication verifies traffic should be authorized. Next, once traffic is secured access must be enforced. Yet access is no longer a yes or no decision because trust is no longer implicit but earned. Therefore, access must be dynamic and variable.

Because of zero trust, there also is a requirement that all connections to and from data be known. Effectively, access is heavily restricted to only known, and expected data flow. Then because assets themselves are considered untrusted, each user, service, and device must be continuously scanned, hardened, and even rotated. Continuous remediation, hardening, and rotation are necessary because again trust is not implicit, and the more systems move outside control and time trust is lost.

Variable Trust

Access controlled by **variable trust**

- Similar to real-life credit scores



User authentication with username/password

10 points

Device authentication

10 points

Known device and location

10 points

Access to PCI database requires

40 points

Multifactor authentication with smart card

20 points

Access to PCI database

GRANTED

Variable Trust

This slide demonstrates an example of what is meant by variable trust. With a zero trust architecture, trust must be earned and can change dynamically. For example, a user accessing a PCI database needs enough trust to gain access. It is possible to quantify the trust requirements such as by giving user points for logging in with a username and password and using a known device and location. Yet access is not simply yes and no.

In this slide access to a PCI, the database requires 40 points. Yet the user and device combination initially only add up to 30 points. Rather than denying the connection variable trust can prompt or require an additional piece to increase trust. In this example, the user is prompted for smart card authentication. Supplying a smart card gives another 20 points for a total of 50 points. 50 points are enough trust to access the PCI database, so access is granted.

Keep in mind part of variable trust is continuously re-evaluating trust, so once access is granted, it is not permanently given. Also, the concept allows the trust to accumulate or be lost over time due to a user or device's behavior.

Trust Over Time

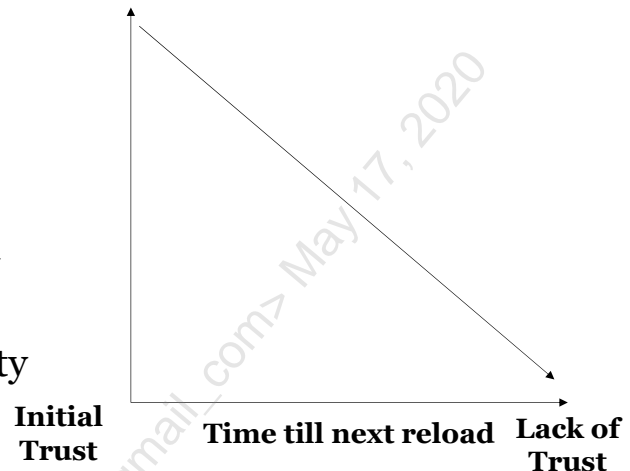
Risk to systems increase over time

- Systems need to be reloaded
 - Due to deviations from baseline
- Credentials need to be rotated
 - Limits compromise and reuse
- Certificates need to be replaced
 - Limits compromise and reuse

Cycling time increases with security

- But still should take place

Security Measures



Trust Over Time

With zero trust, trust is earned or lost. One concept is that assets can lose trust simply due to time. The reasoning behind this is that the longer a machine has been in production, the more likely it is compromised or deviates from baseline. This applies to more than systems. For example, user, service, and applications utilize credentials and/or certificates. The longer these are in production, the more likely they are to be stolen and used. As a result, the rotation is control necessary within a zero trust architecture.

Computers, credentials, and certificates need to be rotated within a period of time to limit risk. The more security measures in place to secure these assets the longer it can be before rotation. Yet a time period needs to be discussed and set. The diagram in this slide represents how trust is lost over time. The more security measures in place, the longer a device can be trusted, but eventually, trust is lost.

Zero Trust Review

Least privilege should be founded on zero trust

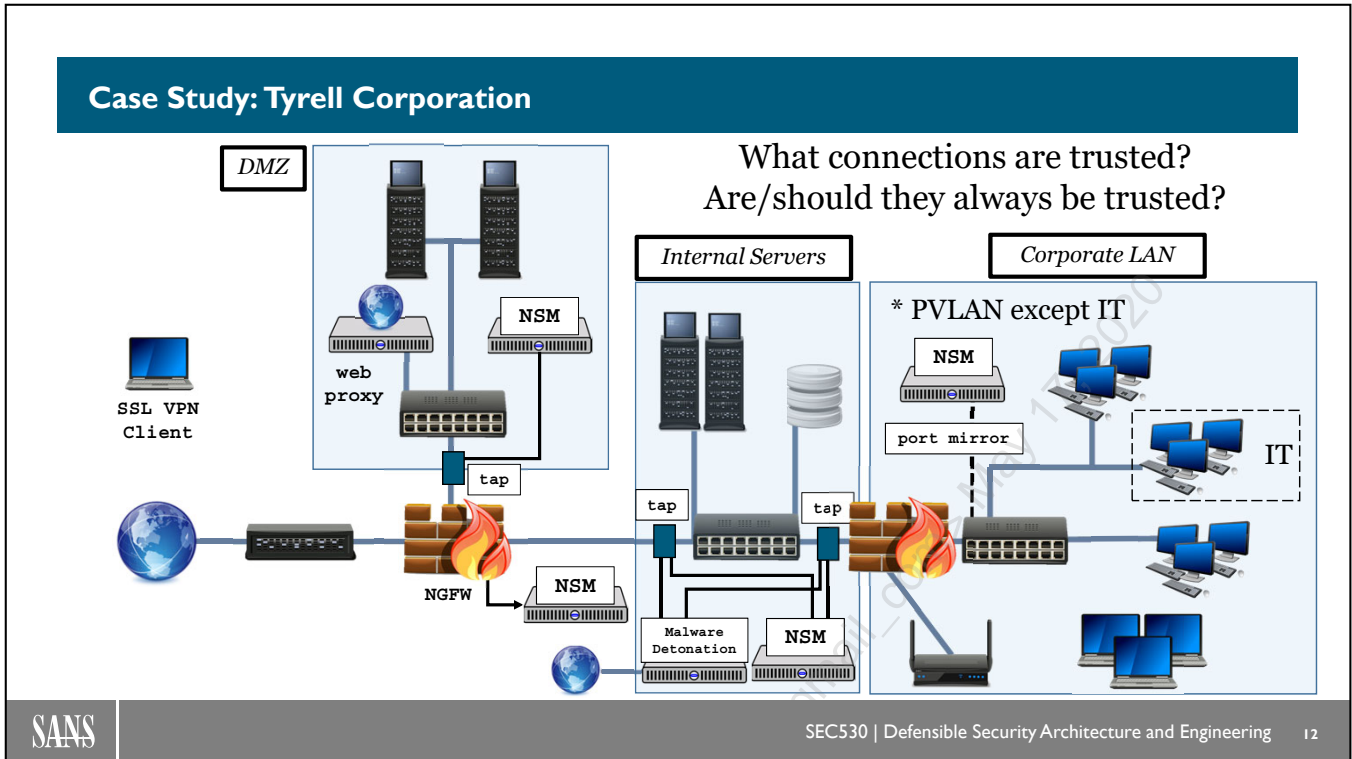
- All access should be authenticated and verified
- Trust should be earned and dynamically adapt
- Know thy network is a must

Implementations should start with the basics

- Securing systems with critical data
- Server to server communication (easier to start with)

Zero Trust Review

The concept and strategy of zero trust is an extension to the concept of least privilege. Zero trust simply extends the concept to make it more encompassing. Implementing zero trust can be daunting, so it generally is recommended to start with systems and networks you have full control over such as server networks in a data center. The rollout tends to be in stages and slowly matures an organization's internal defenses.



Case Study: Tyrell Corporation

This diagram represents the Tyrell Corporation's design. Internal to the organization there are servers and workstations. External to the organization are laptops and mobile devices that connect via a SSL VPN to access internal resources.

Course Roadmap

- Day 1: Defensible Security Architecture
- Day 2: Network Security Architecture
- Day 3: Network-Centric Application Security Architecture
- Day 4: Data-Centric Application Security Architecture
- **Day 5: Zero Trust Architecture**
- Day 6: Capstone: Design, Detect, Defend

CURRENT STATE ASSESSMENT, SOCS, AND SECURITY ARCHITECTURE

1. Zero Trust Architecture
2. **Credential Rotation**
3. Securing Traffic
4. **EXERCISE: Network Isolation and Mutual Authentication**
5. Host-Based Firewalls
6. Network Access Control (NAC)
7. Segmentation Gateways
8. Security Event Information Management (SIEM)
9. **EXERCISE: SIEM Analysis and Tactical Detection**
10. Log Collection
11. Audit Policies
12. Host Hardening
13. Patching
14. Tripwires and Red Herring Defenses
15. **EXERCISE: Advance Defense Strategies**

Course Roadmap

The next section covers credential rotation.

Zero Trust Credentials

Under zero trust credentials should be rotated

- Assumption is credentials are compromised
- Includes private keys and usernames and passwords

Problems created by credential rotation

- Irritates and inconveniences end users
- Potential to break critical services
- Goes against NIST 800-63B best practices

Zero Trust Credentials

In order to gain access to critical data and systems attackers commonly steal and reuse credentials. Combine this with the fact that credentials are never rotated or are rotated extremely rarely, and the situation just gets worse. Under zero trust credentials are assumed to have a higher risk over time. This means that credentials such as usernames and passwords and even things such as TLS certificates need to be rotated over time.

However, password rotation causes major issues. First and foremost, it inconveniences end users. For critical services, it can cause service outages. Not to mention that rotating passwords is against the best practices set in NIST 800-63B.

NIST 800-63B¹

NIST clearly states password rotation is not recommended
"Verifiers SHOULD NOT require memorized secrets to be changed arbitrarily (e.g., periodically). However, verifiers SHALL force a change if there is evidence of compromise of the authenticator."¹

- Based on studies that show password rotation increases the likelihood of poor passwords²



So, is password rotation good or bad?

NIST 800-63B¹

NIST 800-63B is on Digital Identity Guidelines: Authentication and Lifecycle Management. The document is a best practice guide for establishing and maintaining a secure digital identity. This document provides specific instructions on what all a digital identity comprises of or should comprise of and how to secure it. This includes practices such as combining user and device authentication and multifactor authentication.

Inside NIDS 800-63B is a controversial statement that states "Verifiers SHOULD NOT require memorized secrets to be changed arbitrarily (e.g., periodically). However, verifiers SHALL force a change if there is evidence of compromise of the authenticator."¹ This means that organizations should not force users to rotate passwords at all. This recommendation is based on studies that reflect that forcing users to change passwords causes them to gravitate towards weak passwords or write them down.²

[1] <https://pages.nist.gov/800-63-3/sp800-63b.html>

[2] <https://www.ftc.gov/news-events/blogs/techftc/2016/03/time-rethink-mandatory-password-changes>

Credential Rotation

If a **stolen certificate** is revoked and reissued, what happens?

- Adversary can no longer serve trusted content

If a **stolen credential** has a password change, what happens?

- Adversary loses access to account

Attacker can re-gain access through backdoors and local admin accounts/system-level accounts

- But beaconing and persistence can be detected

Rotation is beneficial but requires proper password policies

Credential Rotation

Credential rotation is recommended with a zero trust architecture because if credentials are stolen by an adversary, but then they become rotated the adversary loses access to them. For example, if an adversary had access to a user account and a cleartext password but the user changes their password then the adversary may lose access to the network or at least the assets that required that user's password. This example is why under a zero trust architecture credential rotation is recommended. The assumption is that credentials can and will become compromised therefore they need to be rotated.

Keep in mind rotating credentials alone is not enough to keep an adversary out. If backdoors exist that use other accounts such as local or system-level accounts or even with rootkits, then the attacker still has access and can potentially regain user credentials. However, this all falls into defense-in-depth. The existence of backdoor network access and the use of persistence mechanisms provide additional visibility for defenders to detect. So credential rotation can help with rotating adversaries off existing credentials and forcing them to use other methods that are easy to hone in on and detect.

Password Policies

Rotation increases chance of users picking weak passwords

- Primarily due to lack of password policy controls

Windows supports fine-grained password policies¹

- Allows different password policies per user or group

The screenshot shows the 'Password Settings' dialog box for a policy named 'High Security Password Policy'. The settings are as follows:

Setting	Value
Name	High Security Password Policy
Precedence	1
Enforce minimum password length	<input checked="" type="checkbox"/>
Minimum password length (characters)	10
Enforce password history	<input checked="" type="checkbox"/>
Number of passwords remembered	24
Password must meet complexity requirements	<input checked="" type="checkbox"/>
Store password using reversible encryption	<input type="checkbox"/>
Protect from accidental deletion	<input checked="" type="checkbox"/>
Enforce minimum password age	<input checked="" type="checkbox"/>
User cannot change the password within (days)	1
Enforce maximum password age	<input checked="" type="checkbox"/>
User must change the password after (days)	42
Enforce account lockout policy	<input checked="" type="checkbox"/>
Number of failed logon attempts allowed	3
Reset failed logon attempts count after (mins)	30
Account will be locked out	
For a duration of (mins)	30
Until an administrator manually unlocks the account	<input checked="" type="radio"/>

Password Policies

A major concern with password rotation is that end users who are forced to change passwords once a month or even once every couple of months end up choosing easier to guess passwords or permutations of previous passwords. This greatly increases the likelihood that an adversary can guess passwords or re-obtain previously compromised credentials. The results of this are primarily due to having poor password policies.

Many organizations use the built-in password policies capabilities built into Windows or Linux. Linux uses the pluggable authentication module (PAM) which is extensible and capable. However, Windows and Active Directory password policy managers only can enforce simple rules. The image in this slide shows an example of fine-grained password policies introduced in 2008. Fine-grained password policies allow password policies to be set per user or group so long as that user or group is a global object in Active Directory. Server 2012 and Windows 8 and later can set a fine-grained password policy via Active Directory Administrative Center.

[1] <https://blogs.technet.microsoft.com/canitpro/2013/05/29/step-by-step-enabling-and-using-fine-grained-password-policies-in-ad/>

[2] <https://www.ftc.gov/news-events/blogs/techftc/2016/03/time-rethink-mandatory-password-changes>

Third-Party Password Policy Management

Microsoft's password policy allows weak passwords

- **Winter2018!** is susceptible to cracking and guessing
- But meets complexity requirements

Third-party solutions provide granular policy requirements

- Cannot contain dictionary terms
- Disallow characters at beginning or end of password
- Present GUI breakdown of rules
- Disallow passwords using common permutations

Linux pluggable authentication module (PAM) has granular support

Third-Party Password Policy Management

The problem with fine-grained password policies or built-in Windows password policies, in general, is they are basic in nature. While they can enforce complexity such as requiring three out of four from upper case, lower case, number, or special character and can set length these settings can still result in extremely poor passwords. **Winter2018!** is an example of a poor password that would be allowed with a policy that allows a minimum character length of 10 or less and has password complexity enabled.

An alternative is to purchase and use a third-party password policy management tool. These products integrate into Windows by talking directly to the Local Security Authority and act as a custom passfilt.dll. This DLL handles password policy enforcement. When a third-party solution is used, then password policies become significantly more granular. Extra capabilities include but are not limited to the ability to block passwords that are an exact or partial match to a dictionary list, that begin with or end with certain characters such as special characters or numbers, or that use common permutations that password crackers automatically attempt.

Keep in mind that just because you can do something does not mean you should. Too many rules frustrate an end user. As a general rule, the more controls enabled in a password policy, the more training end users should be received. Also, it is important to explain why and notify users it is to protect them and that they should follow the same rules in their personal life. Also, some third-party solutions allow the installation of a client-side component that replaces the default password change GUI to include the password policy rules. This can be helpful to guide the end users to a proper password.

[1] <https://nfrontsecurity.com/products/nfront-password-filter/>

[2] <https://www.manageengine.com/products/self-service-password/password-policy-enforcer.html>

Password Auditing

Passwords should be evaluated for weaknesses

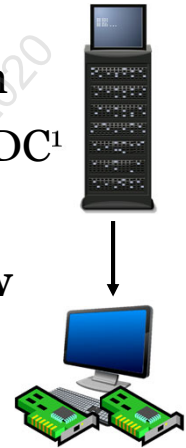
- Consider doing it before the adversary does

Possible to intentionally dump hashes and test them

- Meterpreter capable of grabbing all hashes from DC¹
- Hashes can be pulled from volume shadow copy
- Linux hashes in **/etc/passwd** and **/etc/shadow**

Hashes can be sent to system with GPUs

- Low-cost solution for password auditing



Password Auditing

An attacker who steals password hashes is likely to attempt to use a password cracker in attempt to find the cleartext password using the password hash. A password cracker works by hashing cleartext strings and seeing if the resulting hash matches the password hash it was given to crack. Penetration testers and attackers routinely use password cracking rigs or pre-calculated tables called rainbow tables. A rainbow table is the result of pre-computed password hashes saved in a database so that they can be quickly used to look up cleartext passwords. The reason a strong password policy is necessary is to prevent passwords that are easy to guess or crack.

If organizations know the bad guys are going to attempt to guess passwords and crack passwords, then why do they not do the same? Password crackers, rainbow tables, commonly used password dictionaries are all easy to find and use. Tools exist such as Meterpreter that can automate the process of dumping hashes from Windows systems and even Active Directory. On Linux, the `/etc/passwd` and `/etc/shadow` files contain password hashes and corresponding usernames. Automation of password hash dumping and evaluation with password cracking tools can be used to bolster security greatly. For example, a script could dump password hashes once a week and then attempt to crack them within a few days. If the password is cracked, the end user could be notified or forced to change their password.

If considering the implementation of a password auditing system a few things need to be considered. First off, cleartext passwords should never be saved or immediately destroyed if cracked. The purpose of the solution is not to see end user's passwords but instead to test them for weaknesses. Next, the password cracking solution should be hardened and heavily segmented. If the system ever were to be compromised an adversary would have access to company-wide password hashes. Also, the system should use graphics cards for password cracking. The math computations performed by a graphics card are thousands of times faster than CPUs, and so a system with one or more graphics card is more efficient. Lastly, you should have permission to implement this solution. Some countries or legal departments may object to the use of a password auditing solution.

[1] <https://blog.rapid7.com/2015/07/01/safely-dumping-domain-hashes-with-meterpreter/>

[2] <http://kestas.kuliukas.com/RainbowTables/>

Automatic Credential Rotation

Some accounts are high risk and need rotation frequently

- **Local administrator accounts** (if enabled)
- **Service Accounts**

Both account types are attacker favorites

- Gold images leave local admin password the same
- Service accounts often set and never touched again
 - Service accounts often have admin privileges

Both account types can integrate with automatic rotation

Automatic Credential Rotation

In some cases, credentials can be automatically rotated. This does not work for end users, but it does for critical accounts that may be in use such as local administrator accounts and service accounts. Both of these types of accounts are common targets. The local administrator account is recommended to be disabled, but organizations commonly enable it for troubleshooting purposes. Worse yet, the administrator account password is often the same across an organization. This typically is because the system was deployed using a gold image, or master image such as using Microsoft SCCM to deploy workstation images.

Service accounts are different. They often are purpose-built for a specific application. Because of this service account passwords often are never changed for fear of breaking the application. This can lead to problems as service accounts often have elevated privileges and are not locked down or monitored appropriately. Fortunately, both local administrator accounts and service accounts can be configured for automatic password rotation.

Local Administrator Password Solution (LAPS)¹

LAPS is a free tool from Microsoft

- Automatically rotates local admin password
- Requires agent or DLL registered on all systems
- Supports Server 2003 / Vista and later systems

LAPS is centrally controlled via Active Directory

- Group policy sets rotation constraints and timing
- Active Directory stores passwords
- Cleartext passwords visible through AD

LAPS UI	
ComputerName	hr01
Password	(71nnbL!p9VzgK\$ [I
Password expires	2/10/2018 3:44:50 PM
New expiration time	Saturday, February 3, 2018

Password Complexity	
Complexity	Large letters + small letters + numbers + specials
Password Length	64
Password Age (Days)	7

Local Administrator Password Solution (LAPS)¹

The local administrator password can be set using Microsoft Local Administrator Password Solution which is referred to as LAPS. LAPS automatically rotates local administrator passwords and stores them in Active Directory. Furthermore, the length of the password and how often it is rotated is set centrally with group policy. This means each computer has a cryptographically random password that frequently rotates. The password stored in Active Directory is protected with access control lists to limit access to only specific groups and supports auditing password retrieval. This means that the local administrator account can be enabled for troubleshooting without the risk of mass compromise via techniques like pass-the-hash using the local administrator account.

For LAPS to work an agent or DLL must be registered on all machines that LAPS will be used on. These systems must all be running Windows Server 2003 or Vista and later. Also, any group policies that attempt to set the local administrator password must be removed. These are no longer supported by Microsoft, and the option to set the local administrator account via group policy was removed with patch MS014-025. However, prior group policies settings may still exist and need to be removed for LAPS to work.

[1] <https://technet.microsoft.com/en-us/mt227395.aspx>

Managed Service Account (MSA)¹

Server 2008 R2, Windows 7, and later support MSAs

- MSA is a special service account dedicated to one system
- Password is rotated the same way computer accounts are
 - Defaults to every 30 days (can be changed)
 - Uses a random 120-character password
 - Cannot be locked out
 - Cannot perform interactive logons

MSA set per computer with PowerShell

Managed Service Account (MSA)¹

Beginning with Server 2008 R2 and Windows 7 Microsoft supports the use of a managed service account (MSA). MSAs are special service accounts that are dedicated to a single system. These accounts have passwords that are automatically changed using the same password mechanism the computer accounts use to rotate passwords. This mechanism uses random 120-character passwords. MSAs default to password rotation every 30 days but this can be changed between 1 and 1,000,000 days. MSAs must be created with PowerShell. The password age is controlled using the MaximumPasswordAge registry key found under the registry key location below:

HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NetLogon\Parameters

An MSA account cannot be locked out, and it cannot be used for interactive login. A 120-character password is unlikely to be cracked. Combine this with constant rotation, and the service account credentials become fairly secure. The downside to MSAs is that they are not supported across multiple systems and do not support common applications like SQL or Exchange. Other limitations include the inability to be used for scheduled tasks.

MSAs require an AD schema of Windows Server 2008 R2 or later and a Windows Server 2008 R2 Domain functional level.

[1] <https://blogs.technet.microsoft.com/askds/2009/09/10/managed-service-accounts-understanding-implementing-best-practices-and-troubleshooting/>

Group Managed Service Accounts (gMSA)¹

MSAs have some significant limitations

- Only work on one system (cannot be shared)
- Does not support scheduled tasks
- Not supported by SQL, Exchange, and many 3rd parties

gMSA solves many of these issues

- Requires Server 2012 or Windows 8 and later
- Allows service account to work on multiple systems
- Works with service accounts and other mainstream services

Group Managed Service Accounts (GMSA)¹

Beginning with Server 2012 and Windows 8 Microsoft operating systems support group managed service accounts (gSMA). A gSMA solves many of the issues associated with MSAs. For example, a gSMA can be used across multiple hosts. This capability alone enables advanced use cases such as with clustering or load balanced farms. Also, gSMAs can be used with scheduled tasks and mainstream applications such as Microsoft SQL 2008 R2 SP1 and later and Exchange 2010 and later. Also, gSMAs work with scheduled tasks.

To create a gSMA, a KDS root key must first be created on a domain controller. To do this, the command `Add-KdsRootKey` needs to be run within PowerShell. After this command, you must wait 10 hours for the key to be replicated to all domain controllers. Then PowerShell can be used to create a managed service account and then PowerShell can be used on each host the managed service account is going to be used on.

The use of gSMA requires an AD schema of Server 2012 or higher, and a domain controller with the Microsoft Key Distribution Service enabled.

[1] <https://blogs.technet.microsoft.com/askpfeplat/2012/12/16/windows-server-2012-group-managed-service-accounts/>

[2] <http://woshub.com/group-managed-service-accounts-in-windows-server-2012/>

Credential Rotation Review

Credential compromise is a reality

- Rotating passwords helps to remove access
- Strong password enforcement necessary to limit risks of password rotation
 - But still increases risk of users writing down passwords
- Also consider implementing password auditing

Multi-factor authentication highly recommended

- Helps to offset the need or rotation period of accounts

Credential Rotation Review

Username and passwords are a major weakness in organizations today. Previously it was thought that passwords would die off by now, but instead, they are in use on more devices and applications than ever. Because of this credential rotation and strong password policies are necessary. In truth, the best solution is still to implement and enforce multi-factor authentication.

Course Roadmap

- Day 1: Defensible Security Architecture
- Day 2: Network Security Architecture
- Day 3: Network-Centric Application Security Architecture
- Day 4: Data-Centric Application Security Architecture
- **Day 5: Zero Trust Architecture**
- Day 6: Capstone: Design, Detect, Defend

CURRENT STATE ASSESSMENT, SOCS, AND SECURITY ARCHITECTURE

1. Zero Trust Architecture
2. Credential Rotation
3. **Securing Traffic**
4. **EXERCISE: Network Isolation and Mutual Authentication**
5. Host-Based Firewalls
6. Network Access Control (NAC)
7. Segmentation Gateways
8. Security Event Information Management (SIEM)
9. **EXERCISE: SIEM Analysis and Tactical Detection**
10. Log Collection
11. Audit Policies
12. Host Hardening
13. Patching
14. Tripwires and Red Herring Defenses
15. **EXERCISE: Advance Defense Strategies**

Course Roadmap

The next section covers the concept of Zero Trust Architecture.

Securing Traffic

Under zero trust all network traffic **MUST** be

- **Authenticated** - Prove user/device is legitimate
- **Encrypted** - Cryptographically prove source and destination while protecting confidentiality

Encryption cannot be perimeter-based

- Requires encryption at either device or application
- Endpoints configured to drop anything not encrypted

Securing Traffic

If the network is not trusted, then measures must be taken to secure network traffic. The two main factors when providing security in a seemingly hostile environment are to enforce authentication and encryption. Encryption provides confidentiality between two systems and authentication allows them to identify themselves before sending data.

Specifically, encryption must be endpoint to endpoint under zero trust. This is because network devices themselves can be compromised or allow sniffing or man-in-the-middle attacks. Full encryption enforcement under zero trust means all traffic must first be authenticated and encrypted. Anything else must be dropped.

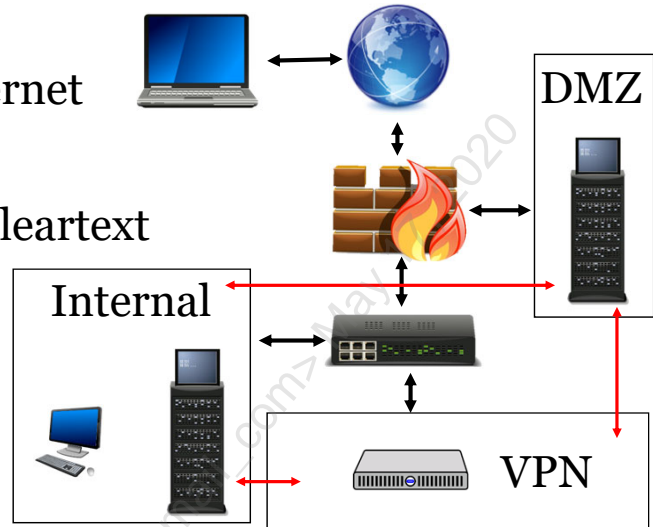
Traditional Communication

Traditional network design

- Encryption used over internet
- Used in DMZ services

Internal communication is cleartext

- Workstations to server
- Server to server
- VPN to server



Traditional Communication

This slide demonstrates a traditional network architecture. In the slide communication from the internet to the VPN concentrator is encrypted. So is communication to the DMZ. However, internal communication such as from a desktop subnet to a server subnet is in cleartext. A laptop that is using a VPN is encrypted up to the concentrator but from the concentrator on it can be cleartext. The problem with this is that it assumes communication on the inside of an organization is trusted.

Methods of Securing Transmission

Multiple methods to secure network transmission

- **TLS** - Transport layer security
- **IPsec** - Kernel-level authentication and encryption
- **802.1X** - Port-based network access control
- **Single Packet Authorization (SPA)**

TLS and IPsec provide authentication and encryption

- 802.1X and SPA only provide authentication
- Zero trust mandates end-to-end encryption

Methods of Securing Transmission

Ideally network traffic is secured usually mutual bidirectional authentication and encryption. Two solutions that allow this are TLS and IPsec. TLS is a transport layer solution, and IPsec is kernel-level. In reality, organizations may opt not to use encryption or may not have the capabilities to do so. In this case solutions like 802.1X or single packet, authorization can be considered to authenticate network access without encrypting all traffic.

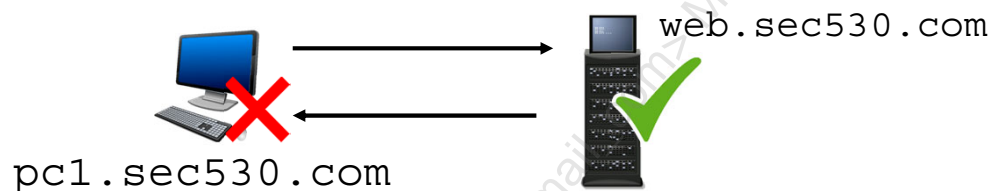
Trust Model Change

Computer authentication is often one-way

- Example: SSL/TLS

Client connects to TLS site and authenticates server

- Client is commonly not verified
- Yet SSL/TLS support mutual authentication

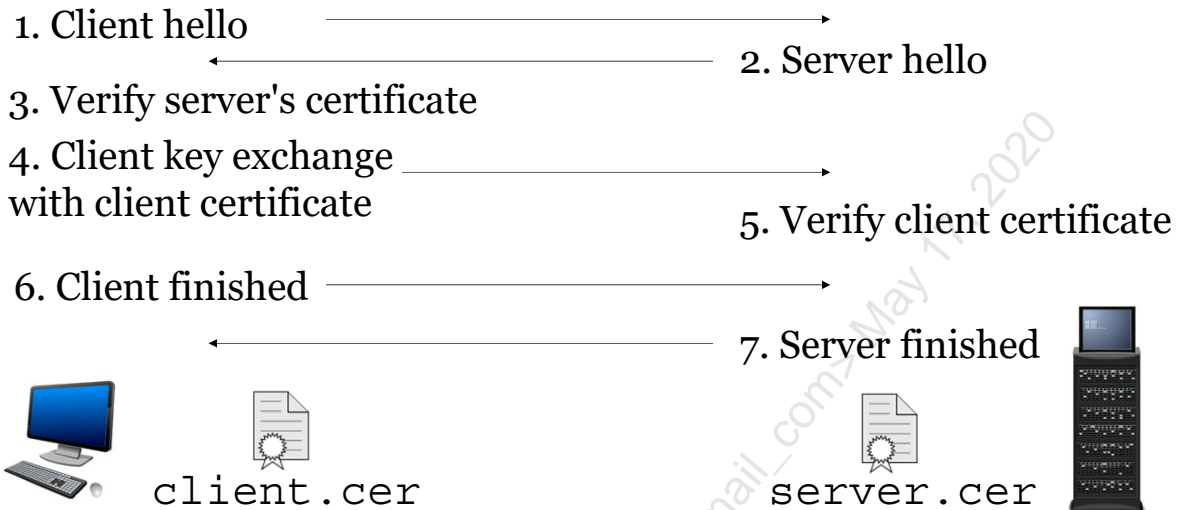


Trust Model Change

Encryption such as TLS often involves one-way trusts. For example, a connection to a website using HTTPS typically involves a client verifying a server is who it claims to be. Yet TLS can also be used to verify a client is who they claim to be. While mutual authentication is supported, it may not be practical for external facing applications that are from unknown clients.

Internally, mutual authentication makes a lot of sense. Client connections should only be from authorized systems. Therefore, why would a connection be allowed to or from untrusted clients? In this respect, systems can be configured to use mutual authentication. This provides enhanced encryption but also minimizes the attack surface. If a client cannot interact with a web server because it cannot pass a client certificate check, then it is significantly less likely to be able to exploit the web server successfully.

Mutual TLS (mTLS)



Mutual TLS (mTLS)

Below is a detailed explanation of the steps in this slide.

1. Client hello contains the TLS versions, cipher suites supported, client's order of preference, a random byte used for subsequent computations, and session-specific data
2. Server sends the client its TLS versions, and cipher suites supported compared to the client's list, a random byte, the server's certificate, and a client certificate request
3. Client verifies the server's certificate is valid
4. Client calculates pre-master key and encrypts it with server's public key and also uses the client's private key to encrypt additional data known by both the server and client. The client also sends the client's digital signature to the server in this step.
5. Server verifies client's certificate

Between step 5 and 6 both the client and server use the pre-master key to generate session keys. These keys are symmetric.

6. The client sends a message to the server that future communication will be encrypted with the session key. Client also sends a separate message that the client handshake is complete
7. The server sends a message to the client that future communication will be encrypted with the session key. Server also sends a separate message that the server handshake is complete

The client certificate request in step two includes a list of supported certificate types and the distinguished names of supported certificate authorities.

[1] <https://support.microsoft.com/en-us/help/257591/description-of-the-secure-sockets-layer-ssl-handshake>

[2] https://www.ibm.com/support/knowledgecenter/en/SSFKSJ_7.1.0/com.ibm.mq.doc/sy10660_.htm

Client Certificates

Enabling mutual authentication requires simple change

- May just require clicking a radio button
- Or adding some text to a configuration file
- **Requires** clients to have **certificates** and a CA

The screenshot shows two parts: on the left, the IIS 'SSL Settings' interface with 'Require SSL' checked and 'Require' selected under 'Client certificates'; on the right, an NGINX configuration snippet with red arrows pointing to 'ssl_client_certificate' and 'ssl_verify_client on;'.

```
server {  
    listen 443 ssl;  
    ssl_certificate /etc/nginx/certs/server/sec530.com.crt;  
    ssl_certificate_key /etc/nginx/certs/server/sec530.com.key;  
    ssl_client_certificate /etc/nginx/certs/ca/ca.crt;  
    ssl_verify_client on;
```

Client Certificates

Configuring an application to require client certificates for mutual authentication is typically easy. For IIS one simply has to enable SSL Settings and then click on Require under Client certificates. For Apache or NGINX, the configuration file needs to be modified to force client verification and to specify which certificate authority the client will be verified against.

The pictures in the slide are from IIS 8 and an NGINX configuration file. While the NGINX configuration has a setting called `ssl_client_certificate` the certificate referenced is actually the public key of an authorized certificate authority. This certificate authority is what is used to verify a client's certificate. The main issue with deploying mutual authentication is not configuring applications but instead getting certificates on both clients and servers.

Public Key Infrastructure (PKI)

Automation is critical to support zero trust

- Private PKI allows automation of certificate deployment
- With support for client and server certificates

Windows Server capable of significant PKI capabilities

- Automatic certificate enrollment via GPO and AD
- Certificate templates and restrictions
- Secure private key archival
- Hierarchical certificate authorities roles and services

Public Key Infrastructure (PKI)

Public key infrastructure is designed to support cryptographic trust and allow encryption via asymmetric and symmetric keys. The problem is PKI is complex and difficult to manage. Even with PKI being out for a long period of time it still is difficult. Fortunately, most organizations own Windows Server licenses. What they do not know is that because of this license they can deploy a Windows PKI system. Windows PKI, in turn, provides an easier PKI implementation that supports automatic certification issuance to Windows systems and can integrate with other systems such as Cisco or Linux.

To be fair, PKI is complex regardless of it being Windows or another solution. The Windows PKI is just positioned well to aid in a zero trust deployment.

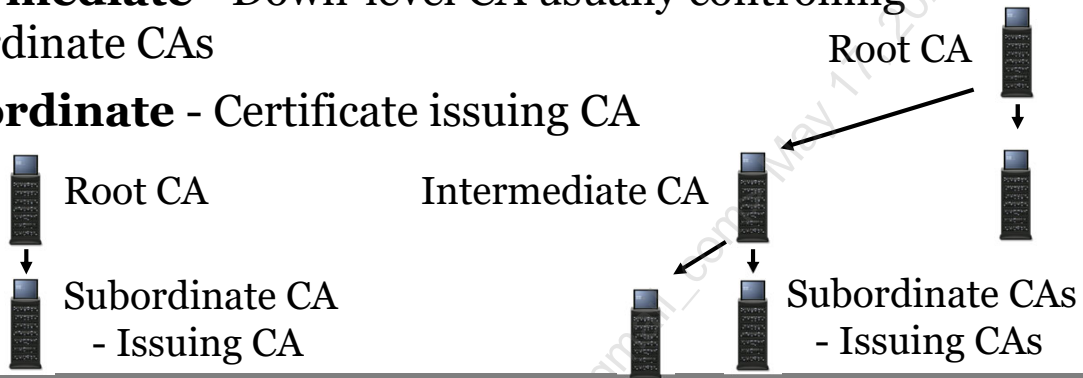
Certificate Authorities (CA)

PKI is composed of one or more certificate authorities

Root - Self-signed CA and most trusted CA

Intermediate - Down-level CA usually controlling subordinate CAs

Subordinate - Certificate issuing CA



Certificate Authorities (CA)

Certificate authorities are a major component of PKI. Certificate authorities are the systems used to verify trust and specifically the systems that issue certificates. Certificate authorities are deployed in a hierarchical fashion. Typically, there are two or more certificate authorities for security purposes.

The initial certificate authority is a root CA. The root CA creates a self-signed certificate as it is the initial chain and beginning of a custom PKI. The CA can be used for issuing certificates but is highly recommended to be only used for issuing or renewing other certificate authorities. The other certificate authorities are used to issue certificates while keeping the root CA secure. How this is done is the root CA is typically offline meaning it is only used during issuance or renewal of a sub-level CA. This protects the private key that has ultimate trust.

Issuing certificate authorities remain online. However, there can be multiple levels of CAs. A basic deployment may just involve the root CA and a single online certificate authority acting as an issuing CA. This down-level CA would be a subordinate CA. But organizations that segment their workforce or assets may need fine-grained trust control. These organizations may implement what is called an intermediate CA. An intermediate CA sits between a root CA and a subordinate CA and can allow granular control such as restricting what subordinate CAs there are and what types of certificates can be deployed.

Certificate Authority Types

Stand-Alone

- Common to small shops
- Manual certificate creation
- Common for Linux shops
- Recommended for root or intermediate CAs
- Should be run off-line
 - Or out-of-band

Enterprise

- Windows specific deployment
- Requires domain membership
- Allows automatic enrollment
- Can be used for smart cards
- Requires AD access
 - Thus, never run off-line
- Contains templates

Certificate Authority Types

Certificate authorities fall into two high-level categories: stand-alone and enterprise. These categories are more of descriptions of how a CA operates and can apply to both Linux, Windows, or appliances. A CA in stand-alone mode is often used for manual certificate issuance and often involves a single CA. In a large PKI deployment, the root CA is almost always a stand-alone CA. Even then the root CA is often shut down or ran out-of-band to protect its private key. In smaller shops, a stand-alone CA may be a root CA and issuing CA. This model does not scale, and certificate creation and issuance are difficult.

An enterprise CA describes a CA that is designed for automatic enrollment and enterprise scalability. These certificate authorities by nature always have to be on and use some form of central configuration and identity management such as Microsoft Active Directory. The window Server operating system can be deployed as a CA and supports certificate templates and automatic enrollment.

Specifically, a Windows Server CA can deploy in two modes called stand-alone or enterprise. The stand-alone mode is meant for a Windows root CA, and enterprise mode is meant for an always-on issuing CA. An enterprise CA uses Active Directory for certificate issuance and potentially for private key archival. Group policy than can be used to configure certificate enrollment for a fully automated distribution platform automatically. Certificates can then be automatically deployed to users and devices.

Automatic Enrollment

Windows PKI allows supports of automatic enrollment of:

- **Device Certificates** - Associated to device activities
- **User Certificates** - Associated to user activities

Capabilities and integration:

- **802.1X**
- **Code Signing**
- **TLS**
- **Smart cards**
- **IPSec**
- **LDAPS**
- **Remote Desktop Protocol**
- **PowerShell Remoting**

Automatic Enrollment

To get enterprise mutual authentication one first needs certificates deployed on both clients and servers. Arguably one of the fastest ways to do this is using a Windows PKI which supports automatic enrollment of both device and user certificates. For this to work a functional enterprise mode certificate authority must be online, certificate templates must be enabled with the automatically enroll privilege enabled for authorized computers and users, and group policy must be configured to allow automatic enrollment.

Once these steps are in place systems and users will automatically begin requesting and renewing certificates. These certificates can then be used for multiple security-related purposes. Specifically, these certificates can be used for IPSec and TLS and provide mutual authentication. Additionally, these certificates can be used for 802.1X port-based authentication which will be discussed shortly.

In a Windows environment, automatic enrollment settings are controlled under the following GPO locations:

Computer Configuration -> Security Settings -> Public Key Policies -> Certificate Services Client - Auto-Enrollment Properties

User Configuration -> Security Settings -> Public Key Policies -> Certificate Services Client - Auto-Enrollment Properties

Computer Configuration -> Security Settings -> Public Key Policies -> Trusted Root Certificate Authorities

IPSec Revisited

Most folks think of IPSec as a VPN protocol

- Works with VPNs but can do so much more

IPSec is a network layer protocol

- Works with application regardless of IPSec awareness
- Works independently of TCP or UDP
- Supports hardware acceleration

Allows transparent encryption and authentication

Application	HTTP
Transport	SSL/TLS
Internet	IPSec

IPSec Revisited

An alternative solution for mutual authentication and encryption is the use of IPSec. While often considered a VPN protocol IPSec can actually be used for a lot of things. For instance, it can be used between two host systems internally to authenticate and encrypt a network session dynamically. In fact, it can be used to hide the fact that a system exists until IPSec is in use.

SSL/TLS operates at the transport layer. Where this comes into play is that an application must be configured to use and accept TLS as part of its supported transport mechanisms. The most common example is the application of HTTP. HTTP is often paired with TLS to form HTTPS. IPSec, on the other hand, is baked into the kernel and is processed by the Internet layer of communication. This allows IPSec to be used regardless of application awareness or without requiring the use of TCP or UDP. Because of this, IPSec is highly flexible and an amazing option for adding authentication and encryption support.

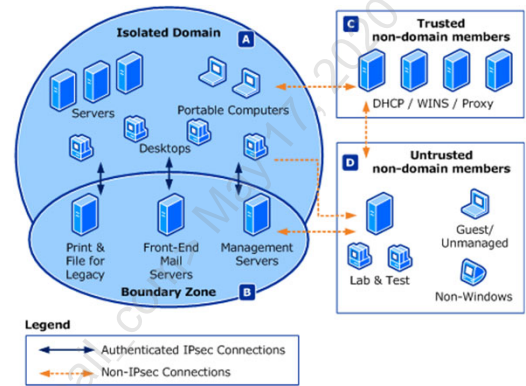
Windows Domain Isolation!

Windows natively supports IPsec for domain isolation

- Blocks unauthorized non-domain access
- Authenticates all traffic
 - Mutually authenticated
- Optionally encrypts traffic

Cannot attack the invisible

- Mitigates man-in-the-middle
- Lowers service exploitation risk



Windows Domain Isolation

Windows has supported the concept of domain isolation for a long time. For example, you can find an article on setting up domain isolation that is from 2007. So, what is domain isolation? Domain isolation is the ability to prevent non-domain joined systems from accessing a Windows environment. Domain isolation uses IPsec built-in to Windows to authenticate, and optionally encrypt traffic. The beauty of the whole concept is that a non-domain joined system cannot see systems that have been isolated. It is as if they do not exist.

Think of this as an invisible force field surrounding an entire city. Enemy forces look towards the city, but all they see is grass and mountains as if the city did not exist. This is exactly the goal of domain isolation. Because IPsec is required to connect to domain systems man-in-the-middle attacks, become mitigated, and the chance of service exploitation drops dramatically. An unmanaged system cannot lob attacks at a domain system's service if it cannot connect to it.

[1] <https://docs.microsoft.com/en-us/windows/security/identity-protection/windows-firewall/domain-isolation-policy-design>

[2] [https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2008-R2-and-2008/cc730709\(v=ws.10\)](https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2008-R2-and-2008/cc730709(v=ws.10))

Windows IPsec

IPsec integration is part of the Windows Firewall

- Change IPsec default options (recommended)
 - Default does not mandate encryption for ESP
- Requires Connection Security Rules



- Granular control via firewall rules

Windows IPsec

Deploying IPsec in a Windows environment is controlled with either PowerShell scripts or group policy as IPsec is integrated into Windows Firewall. The first recommended step is to change the default IPsec Settings. Keep in mind IPsec connections fall into multiple phases and use mutual authentication similar to how a VPN is supposed to work. The default for phase 1 main mode is to use SHA-1 for integrity and either AES-CBC-128 or 3DES for encryption. The key exchange algorithm is set to Diffie-Helman Group 2 which uses a 1024-bit prime number. The default allows for wide support of devices and operating systems. If you are only using Windows Vista or later, main mode can be changed to SHA-384 for integrity, AES-CBC-256 for encryption, and changed to use Elliptic Curve Diffie-Helman P-384. Elliptic Curve is more secure than Diffie-Helman even though it has a smaller bit size.

Phase 2 quick mode is a little different in that it can be set to provide data integrity only using AH or provide data integrity and encryption via ESP or ESP and AH. Because ESP can traverse NAT resolution and a zero trust architecture mandates encryption it is recommended to check "Require encryption for all connection security rules that use these settings." The defaults for phase 2 quick mode allow authentication without encryption and use SHA1 with optional AES-128 or 3DES encryption. Again, if you are using Windows Vista or later, quick mode security can be changed to use a stronger encryption algorithm such as AES-GCM 256 and a stronger integrity algorithm such as AES-GMAC 256. Galois Counter Mode (GCM) and Galois Message Authentication Code (GMAC) requires Vista systems to be service pack 1 or later.

The last setting for IPsec is the authentication method. Windows defaults to Kerberos computer authentication which is a strong default setting. It only authorizes domain-joined systems this way, and Kerberos is a strong authentication protocol. However, alternative options are to use certificates for users or computers, Kerberos for user authentication, a pre-shared key, or in later operating systems NTLM version 2 for users or computers or a health certificate from Microsoft Network Access Protection. IPsec authentication can require both user and computer authentication or both can be optional, or one can be optional. Having a computer and user authentication is necessary if Windows firewall rules may require authentication against computers and/or users.

Windows Ping Example

Without IPsec

No.	Time	Source	Destination	Protocol	Length	Info
9	3.548979	10.0.0.51	192.168.2.101	ICMP	74	Echo (ping) request id=0x0001, seq=17/4352, ttl=128 (reply in 10)
10	3.549891	192.168.2.101	10.0.0.51	ICMP	74	Echo (ping) reply id=0x0001, seq=17/4352, ttl=127 (request in 9)
13	4.555916	10.0.0.51	192.168.2.101	ICMP	74	Echo (ping) request id=0x0001, seq=18/4608, ttl=128 (no response found)
14	4.556596	192.168.2.101	10.0.0.51	ICMP	74	Echo (ping) reply id=0x0001, seq=18/4608, ttl=127 (request in 13)
16	5.563011	10.0.0.51	192.168.2.101	ICMP	74	Echo (ping) request id=0x0001, seq=19/4864, ttl=128 (reply in 17)
17	5.563779	192.168.2.101	10.0.0.51	ICMP	74	Echo (ping) reply id=0x0001, seq=19/4864, ttl=127 (request in 16)
18	6.566996	10.0.0.51	192.168.2.101	ICMP	74	Echo (ping) request id=0x0001, seq=20/5120, ttl=128 (reply in 19)
19	6.567672	192.168.2.101	10.0.0.51	ICMP	74	Echo (ping) reply id=0x0001, seq=20/5120, ttl=127 (request in 18)

With IPsec

No.	Time	Source	Destination	Protocol	Length	Info
29	1.292524	10.0.0.50	192.168.2.108	ESP	90	ESP (SPI=0x8262df4e)
30	1.292525	192.168.2.108	10.0.0.50	ESP	102	ESP (SPI=0x9ff08aa2)
31	1.309018	10.0.0.51	192.168.2.108	ESP	110	ESP (SPI=0x0b77aa7e)
32	1.309787	192.168.2.108	10.0.0.51	ESP	110	ESP (SPI=0x9e0d80f8)
37	2.316280	10.0.0.51	192.168.2.108	ESP	110	ESP (SPI=0x0b77aa7e)
38	2.317000	192.168.2.108	10.0.0.51	ESP	110	ESP (SPI=0x9e0d80f8)
43	3.322412	10.0.0.51	192.168.2.108	ESP	110	ESP (SPI=0x0b77aa7e)
44	3.323136	192.168.2.108	10.0.0.51	ESP	110	ESP (SPI=0x9e0d80f8)

Windows Ping Example

This slide demonstrates the results of IPsec by showing a Windows ping sent against two separate hosts. One host does not use IPsec, and one host does. The system at 192.168.2.101 does not use IPsec, and as a result, the ping packets or ICMP echo request and replies are clearly evident in the packet capture. The second system at 192.168.2.108 uses IPsec, and as a result, you cannot tell what traffic is being sent or any payload information. The only thing you can tell from network sniffing capabilities is that 10.0.0.50 and 192.168.2.108 are communicating.

What is not represented in this slide is that if a non-domain joined system were to ping 192.168.2.108, there would be no response. 192.168.2.108 is domain isolated and thus would drop the ping request from unauthorized systems.

Authenticating Network Access

Zero trust expects mutual authentication and encryption

- May not be possible or practical

Alternative is to authenticate network access

- Achieved with **Network Access Control (NAC)**
- Or **Single Packet Authorization (SPA)**

NAC is more common and is often centralized

- Centralization makes for strong detection capabilities
- What and where are unauthorized devices?

Authenticating Network Access

Mutual authentication protocols such as TLS and IPSec are ideal for achieving both authentication and encryption. However, this is not always possible, and it is not always practical. For example, IPSec can be complex and, in some cases, can have a noticeable performance impact. If IPSec is complex and has overhead and TLS is not supported by all applications, then what next?

The focus may need to shift towards mature technologies that aim to control who can and cannot talk on a network. Technologies have Network Access Control have been around for a long time and had one purpose: authenticate devices before they can talk on the network. This does not offer network encryption of traffic, but it is still significantly better to limit who can be on the network via authentication than to simply let anyone plug in and have access.

Another solution that can be used is Single Packet Authorization. This is not a new technology, but it is not widely used and in many cases is not proven as a mature solution. SPA uses a single network packet to authenticate to an endpoint before gaining traditional access. Again, the focus is on authentication only.

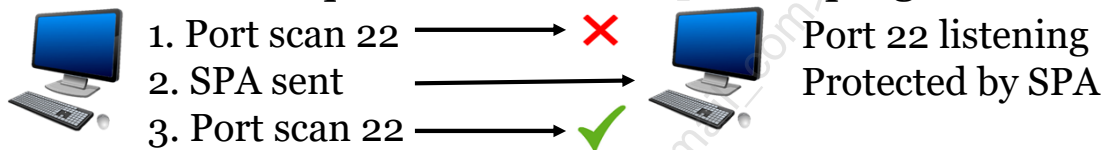
Single Packet Authorization (SPA)

SPA involves blocking connections by default

- Connecting system must first send authentication packet
- Associated as a next-generation port knocking solution

SPA uses asymmetric encryption and HMAC

- Packet is non-replayable due to HMAC and random data
- SPA includes request to authorize port or program



Single Packet Authorization (SPA)

Single packet authorization protects a host from non-authorized connections by blocking connections that have not first authenticated with an SPA packet. Effectively, the host does not appear to exist until passing SPA. This requires software on both the source and destination hosts. An open source example of SPA is fwknop1.

fwknop uses asymmetric keys and an HMAC to transmit a single packet that authenticates a device securely. This packet is composed of the following message composition2:

- 16 bytes of random data
- local username
- local timestamp
- fwknop version
- mode (access or command)
- desired access (or command string)
- MD5 sum

When the packet is received, the host decrypts the data and authenticates the packet. Within the packet is included what the source client wishes to access whether it is a port or a program. If the receiving host authorizes the request, then subsequent requests from the client will be allowed.

[1] <http://www.cipherdyne.org/fwknop/docs/SPA.html>
[2] <http://www.cipherdyne.org/fwknop/>

Securing Traffic Review

Authentication and encryption is mandated under zero trust

- Intent is for endpoint to endpoint traffic

Ideally traffic uses mutual authentication and encryption

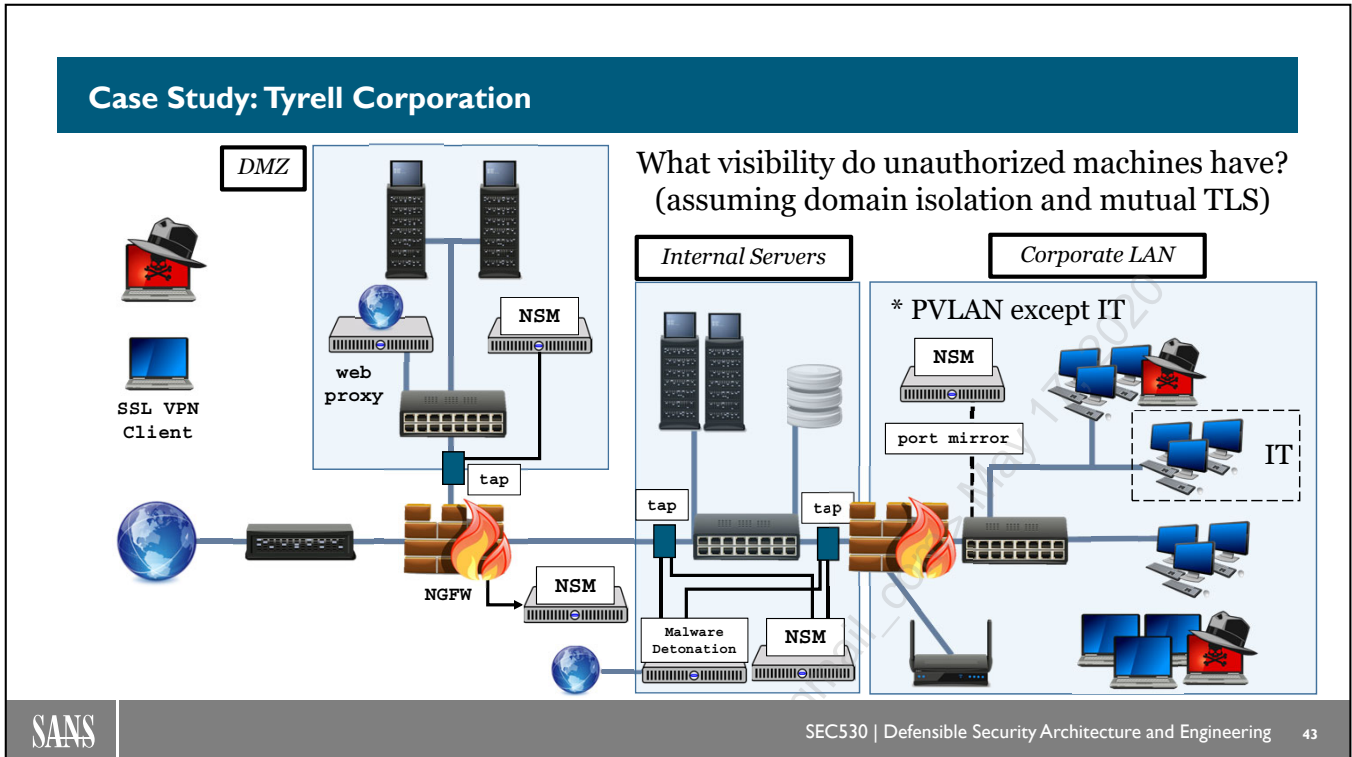
- Easy with Microsoft's IPsec implementation
 - Automation capable with Microsoft PKI deployment
- Or by service using mutual TLS

Above is not always possible and has other implications

Securing Traffic Review

Zero trust heavily relies on securing endpoint to endpoint traffic. By using encryption and mutual authentication only authorized connections can occur. This significantly reduces the attack surface but also deviates heavily from how organizations operate today.

Web servers and other services use TLS today. If they are providing services for organizations directly, it typically is a minor adjustment to switch to mutual TLS. Windows shops are capable of implementing mutual authentication and encryption using the built-in IPsec implementation with Windows Firewall. These options are not always available. The answer is not to give up but to look at other solutions that get to a similar level of only allowing authorized connections to occur.



Case Study: Tyrell Corporation

This diagram represents the Tyrell Corporation's design. Internal to the organization there are servers and workstations. External to the organization are laptops and mobile devices that connect via a SSL VPN to access internal resources.

Licensed To: Martin Brown <hermespaul56@gmail.com>

Course Roadmap

- Day 1: Defensible Security Architecture
- Day 2: Network Security Architecture
- Day 3: Network-Centric Application Security Architecture
- Day 4: Data-Centric Application Security Architecture
- **Day 5: Zero Trust Architecture**
- Day 6: Capstone: Design, Detect, Defend

CURRENT STATE ASSESSMENT, SOCS, AND SECURITY ARCHITECTURE

1. Zero Trust Architecture
2. Credential Rotation
3. Securing Traffic
4. **EXERCISE: Network Isolation and Mutual Authentication**
5. Host-Based Firewalls
6. Network Access Control (NAC)
7. Segmentation Gateways
8. Security Event Information Management (SIEM)
9. **EXERCISE: SIEM Analysis and Tactical Detection**
10. Log Collection
11. Audit Policies
12. Host Hardening
13. Patching
14. Tripwires and Red Herring Defenses
15. **EXERCISE: Advance Defense Strategies**

Course Roadmap

The next section covers the concept of Zero Trust Architecture.

Exercise 5.1: Network Isolation and Mutual Authentication

- Exercise 5.1 is in the digital wiki found in your student VM (recommended)
- Alternatively, you may use your Workbook

This page intentionally left blank.

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

Course Roadmap

- Day 1: Defensible Security Architecture
- Day 2: Network Security Architecture
- Day 3: Network-Centric Application Security Architecture
- Day 4: Data-Centric Application Security Architecture
- **Day 5: Zero Trust Architecture**
- Day 6: Capstone: Design, Detect, Defend

CURRENT STATE ASSESSMENT, SOCS, AND SECURITY ARCHITECTURE

1. Zero Trust Architecture
2. Credential Rotation
3. Securing Traffic
4. EXERCISE: Network Isolation and Mutual Authentication
5. **Host-Based Firewalls**
6. Network Access Control (NAC)
7. Segmentation Gateways
8. Security Event Information Management (SIEM)
9. EXERCISE: SIEM Analysis and Tactical Detection
10. Log Collection
11. Audit Policies
12. Host Hardening
13. Patching
14. Tripwires and Red Herring Defenses
15. EXERCISE: Advance Defense Strategies

Course Roadmap

The next section covers Host-based Firewalls.

Host-Based Firewalls

Under zero trust connections to an endpoint must only involve authorized systems

- Host-based firewalls provide granular controls
- More so than network-based firewalls

Windows comes with **Windows Defender Firewall**

- Previously Windows Firewall with Advanced Security (WFAS)

Linux includes **iptables** and wrappers like **ufw**

Commercial solutions available for Windows, Linux, and Mac

Host-Based Firewalls

Host-based firewalls like Windows Defender Firewall and Linux iptables provide endpoint level network restrictions. These firewalls often come in commercial flavors that either integrate with the native operating system firewalls via built-in APIs or provide their third-party capabilities.

A firewall on an endpoint is capable of more granular control than a network-based firewall. The reason for this is that the process running on the host can see network connections as well as which executables are behind them. Because of this, there are unique capabilities for filtering. On top of this, the endpoint should be the ultimate deciding factor on whether or not a connection is allowed.

[1] <https://opensource.com/article/17/6/4-easy-ways-work-toward-zero-trust-security-model>

Host-Based Firewall Capabilities

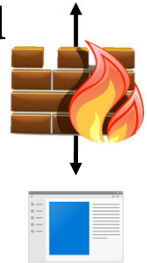
Endpoint firewalls include prevention and auditing

- Allow or deny rules based on ports
- Rules-based on inbound or outbound traffic
- Also supports rules based on executables and file paths

Example: **powershell.exe** denied access **outbound**

Outbound defaults to **allow** and inbound to **deny**

- Assumes connections from endpoint are trusted
- Inbound access is notoriously changed to open



Host-Based Firewall Capabilities

A host-based firewall on Windows is broken down into inbound or outbound rules. Linux iptables defaults to inbound and outbound rules, but because it uses the concept of rule chaining, it is possible to build rule breakouts and simulate zones. Both Windows and Linux default to default inbound deny, and outbound allow. The outbound allow rule is typically left as an allow all rule as systems on the inside are trusted. The concept of trusting a system breaks the law of a zero trust architecture.

The end goal, therefore, is to configure firewall rules that only allow only authorized. All organizations may not be able to reach this goal but should at least make efforts in this direction. One thing that helps make this possible is that host-based firewalls such as Windows firewall can allow or deny based on executables as well as network ports and IP addresses. Rules-based on executables can simply firewall rules for complex applications as well as support more granular filters.

[1] <https://www.howtogeek.com/227093/how-to-block-an-application-from-accessing-the-internet-with-windows-firewall/>

Inbound Access

Organizations should only allow connections to authorized services

- Includes services for workstations and servers

Log inspection can provide list of executables and ports

- Turn these into granular allow rules
 - Lockdown to specific hosts or subnets as necessary
- Then deny anything else

Consider using executables instead of ports to limit access to authorized executables

- Blocks listening on unauthorized applications like Python



Inbound Access

The default rule for inbound access is to deny. Many organizations struggle to manage host-based firewall rules and end up disabling the firewall or switching the default to allow. To implement an inbound deny policy, you first must authorize all ports or executables. The ports and executables necessary to allow a connection, exist in Windows firewall logs. Using scripts or central log collection helps sift through these logs and quickly identifies ports and executables to allow.

Locking down host-based firewalls by executables may be preferable to ports. By allowing access to authorized applications ports will only be reachable when that executable is running. This prevents unauthorized applications from successfully listening on new ports. Regardless, allowing ports or executables is only a first step. Next, these ports and executables need to be locked down to specific hosts or subnets.

[1] <https://security.stackexchange.com/questions/24557/windows-firewall-how-to-block-inbound-for-all-exec-files-in-a-folder>

Outbound Access

Windows and Linux have thousands of binaries per system

- Yet less than a hundred are likely to reach out
- Only authorized binaries should make connections
- Should limit authorized applications to their expected use cases

SMB is necessary but should never happen to internet

powershell.exe should be used for scripting but may not need outbound access at all

Limiting **CustomApp1** to specific hosts/networks helps prevent it from being used as an attack vector upon compromise

Outbound Access

Outbound access defaults to allow and organizations often leave it this way. At a minimum, specific ports or applications should be denied outbound access either to the internal network or the internet. As an example, SMB is necessary internal to select systems but has no business accessing the internet. PowerShell scripting is and should be utilized for automation. Yet it often is called via scheduled tasks or remote calls from specific subnets. Once invoked, powershell.exe on a workstation or server rarely has a business need to make remote connections. Therefore, blocking outbound access from powershell.exe can be beneficial.

This also holds true for custom applications. A custom application may be exploited. When exploitation occurs, the hacker may invoke a new executable or load a malicious payload inside the custom application. If the custom application is limited to only the expected network directionality, then it may be more difficult for an attacker to pivot.

If you stop and think about it the fact that outbound access is allowed by default is crazy. Each system has thousands if not tens of thousands of executables. By having a default allow policy, an organization is stating that every one of these executables is authorized. With log collection or basic PowerShell scripting, an organization can identify on a per system basis what applications are making outbound connections and identify if they are authorized. If they are authorized rules can be created for those executables. Eventually, outbound deny can switch to a default deny.

[1] <https://docs.microsoft.com/en-us/windows/security/identity-protection/windows-firewall/create-an-outbound-program-or-service-rule>

Firewall Logging

Host-Based firewalls provide a gold mine of information

- Monitoring provides the capabilities to implement granular rules
- Blocked events point to unauthorized connections

Logs operationalize endpoints as intrusion detection points

- Notifies of unauthorized inbound and outbound connections
- Allows for early detection and response

Network firewalls can provide high-level central filtering

- Endpoint firewalls provide granular filtering

Firewall Logging

Host-based firewalls are critical for a zero trust network. On Windows, this is doubly true as Windows Firewall makes IPSec domain isolation possible with ease. Host-based firewalls remain important to granular allow or disallow access as close to an endpoint as possible. These firewalls double as zero trust implementations by assuming that network filtering devices may fail.

Also, by using host-based firewalls organizations can weaponize the firewalls for early detection. When an organization has a successful default deny policy for inbound and/or outbound access normal business operations generate zero deny events. Yet if an attacker compromises a single asset, it is highly likely that they will perform a task that will generate a firewall block event. The blocking event then provides early warning.

Course Roadmap

- Day 1: Defensible Security Architecture
- Day 2: Network Security Architecture
- Day 3: Network-Centric Application Security Architecture
- Day 4: Data-Centric Application Security Architecture
- **Day 5: Zero Trust Architecture**
- Day 6: Capstone: Design, Detect, Defend

CURRENT STATE ASSESSMENT, SOCS, AND SECURITY ARCHITECTURE

1. Zero Trust Architecture
2. Credential Rotation
3. Securing Traffic
4. **EXERCISE: Network Isolation and Mutual Authentication**
5. Host-Based Firewalls
6. **Network Access Control (NAC)**
7. Segmentation Gateways
8. Security Event Information Management (SIEM)
9. **EXERCISE: SIEM Analysis and Tactical Detection**
10. Log Collection
11. Audit Policies
12. Host Hardening
13. Patching
14. Tripwires and Red Herring Defenses
15. **EXERCISE: Advance Defense Strategies**

Course Roadmap

The next section covers Network Access Control (NAC).

Network Access Control (NAC)¹

CIS Control 1 - Inventory of Authorized and Unauthorized Devices

- Has always been highest priority control

Step 1 is to inventory all devices

- Step 2 is to only allow authorized devices on the network

NAC provides real-time enforcement of network access

- Thus, it is fundamentally designed to perform both steps
- Also, can change access based on variable conditions

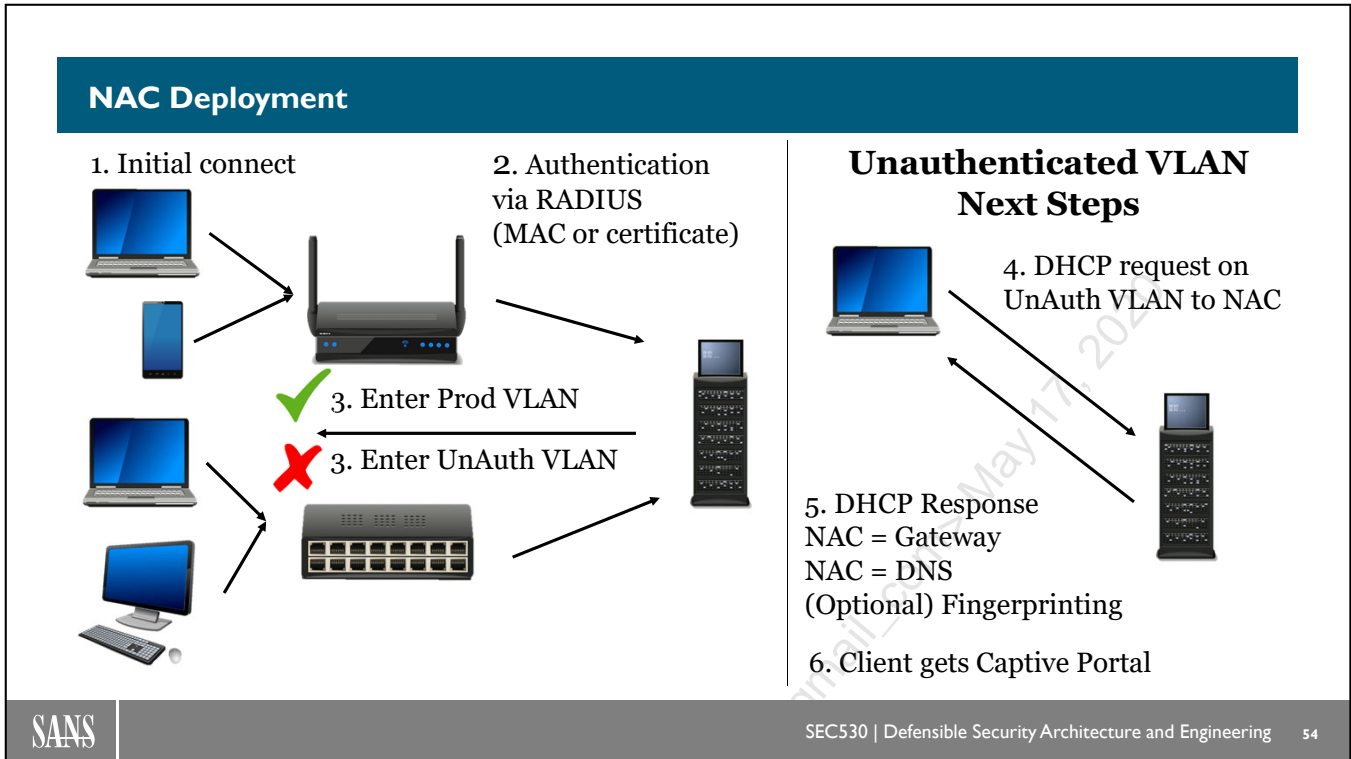
Network Access Control (NAC)¹

The CIS Critical Controls are the top twenty security controls in order of importance that organizations should adopt. The first and highest priority control has always been having an inventory of authorized and unauthorized devices. Yet organizations continue to struggle with having this inventory. Under a zero trust architecture, the internal network is not trusted and thus knowing and controlling network access is critical.

Network Access Control is a solution that provides real-time authorization for network access. NAC functions by integrating with networking gear such as a switch or wireless access point and providing some mechanism for authenticating a device before it has network access. However, the level of network access given can be dynamically controlled by a NAC solution.

Just like zero trust requires, network access can be given, but the level of access can be adjusted based on user or device actions and behaviors. A NAC system controls this by dynamically placing users or systems on specific VLANs or dynamically applying network access control lists.

[1] <https://www.sans.org/reading-room/whitepapers/analyst/membership/35115>



NAC Deployment

This diagram represents some of the capabilities of a modern NAC solution. First, a device wirelessly or physically connects to either a wireless access point or switch. If the device supports 802.1X, the device may attempt to send authentication parameters. If the device does not the MAC address is always available to send in a RADIUS request. The next step is the wireless controller behind the access point, an access point directly, or the switch submits a RADIUS request to the NAC. This radius request includes the MAC address of the device connection and optionally a user or device certificate. At this point, the NAC solution may attempt to authenticate the device using either the certificate or MAC address. If authentication succeeds the NAC solution responds to the network device that the client passed authentication and to place the device on a production VLAN. If the device fails authentication, then the NAC solutions respond to the network device that the client failed authentication and to place the device on an unauthorized VLAN.

Simply because a device gets placed on an unauthorized VLAN does not mean it is game over. It just means the device failed 802.1X authentication and that it is up to a NAC solution to provide alternative controls. One such response can be to have the unauthenticated VLAN use local DHCP or a remote DHCP relay to the NAC system so that it can give the unauthenticated client an IP address. The DHCP response places the NAC as the unauthenticated devices gateway and DNS server. Depending on the NAC solution it is possible that the packets involved during DHCP request and response are captured and analyzed by the NAC solution. The DHCP fingerprint may be used to authenticate a device. More commonly, a captive portal is forced on the unauthenticated client and can be used to provide guest internet or logical steps to become authorized.

Core NAC Capabilities

NAC solutions “authenticate” devices various ways

- 802.1X Port Authentication (CSC 1.5 + 1.6)
- MAC Address OUI (Organizationally Unique Identifier)
- DHCP Fingerprint - analyzes DHCP packets of systems

Compliance checks - Act as augmentation to authentication

- Vulnerability scan
- Intrusion Detection System (IDS) - possibly dangerous...
- Patching/Antivirus/User Agents/Etc.

Core NAC Capabilities

With NAC, each device must meet certain conditions to become authorized. While there are multiple ways of doing this, the two most common are authorization by MAC prefix and 802.1 Port Authentication. The MAC prefix is the first six hexadecimal characters of a MAC address which is associated with the vendor of a specific device. The first six hexadecimal characters are referred to as the OUI or Organizationally Unique Identifier. Port Authentication can be handled multiple ways but is often associated with the use of x509 certificates.

Depending on the NAC solution, other checks can be performed. For instance, a device may require MAC authentication, but since MAC addresses can be spoofed, a post-authentication check, such as a vulnerability scan or SNMP v3 authentication packet, may be used to verify a printer is really a printer. While less common, it is also possible to authorize a device by default but then scan it or attempt to authenticate against it and kick it back off the network should it fail to allow the authentication. All of the options can be evaluated by trying out PacketFence or other commercial NAC solutions. The numerous options and capabilities are why.

MAC Authentication

MAC authentication is not authentication

- MAC addresses can be spoofed
- Yet validating a client based on MAC is better than not
- Beware the perfect solution fallacy

MAC authentication can be combined with 802.1X

- You likely do not use every OUI
- Some MAC addresses are invalid
 - Mainly from MAC tumbling devices or spoofing

```
E0:5F:45 Apple # Apple, Inc.  
E0:5F:B9 Cisco # Cisco Systems, Inc  
E0:60:66 Sercomm # Sercomm Corporation  
E0:64:8B DigiView # DigiView S.r.l.  
E0:66:78 Apple # Apple, Inc.  
E0:67:B3 C-DataTe # C-Data Technology Co., Ltd
```

MAC Authentication

MAC authentication is not a form of authentication. MAC addresses can be spoofed. Authentication normally involves something like my name is Bob and my password is Security555isD@b0mb! (something you know). The something you know could be replaced by something you have (a smart card) or something you are (a fingerprint). Yet MAC authentication effectively says hi my name is Bob. Thus, it is not really a form of authentication even though it is called MAC authentication. This is why NAC solutions started to add capabilities for post-authentication. This allows you to validate a device is really what it says it is.

802.1X is not supported on every device and even if it was it would not always be practical to invest the labor required deploying certificates. While Windows supports automatic certificate enrollment and certain devices like Cisco switches support network enrollment, the truth is many devices do not support automatic certificate deployment. Because of this MAC authentication may be one of the few ways you can enforce NAC.

DHCP Fingerprinting

DHCP request/response can be fingerprinted

- Uses combination of MAC address and option 55
- Option 55 contains the Parameter request list
- How request is performed can be fingerprinted

What order are options?

What order are they requested?

What is the MAC address?

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	0.0.0.0	255.255.255.255	DHCP	342	DHCP Discover
2	0.000972000	10.0.0.2	255.255.255.255	DHCP	352	DHCP Offer
3	0.001732000	0.0.0.0	255.255.255.255	DHCP	365	DHCP Request
4	0.002740000	10.0.0.2	255.255.255.255	DHCP	397	DHCP ACK


```

Option: (60) Vendor class identifier
Option: (55) Parameter Request List
  Length: 13
  Parameter Request List Item: (1) Subnet Mask
  Parameter Request List Item: (3) Router
  Parameter Request List Item: (6) Domain Name Server
  Parameter Request List Item: (15) Domain Name
  Parameter Request List Item: (31) Perform Router Discover
  Parameter Request List Item: (33) Static Route
  Parameter Request List Item: (43) Vendor-Specific Information
  Parameter Request List Item: (44) NetBIOS over TCP/IP Name Server
  Parameter Request List Item: (46) NetBIOS over TCP/IP Node Type
  Parameter Request List Item: (47) NetBIOS over TCP/IP Scope
    
```

DHCP Fingerprinting

DHCP fingerprinting is still not authentication but it ups the difficulty level quite a bit while actually making support for generic devices like printers, scanners, or IoT devices easier. DHCP fingerprinting works by analyzing packets during the DHCP request and response process. Specifically, during the DHCP Discover option 55 contains a list of requested parameters from a DHCP server. The order of this list and the contents of this list tend to be unique to a specific device type. Windows systems will request certain options while a printer would have a very different list. DHCP fingerprinting off option 55 alone is fairly accurate, but the MAC address is also available for use. DHCP option 60 is also used if present. DHCP option 60 is the vendor class identifier often used to identify the operating system in use.

By using a fingerprint profile on DHCP instead of MAC addresses, it is easier to allow network access by device type. For example, MAC authentication would require gathering all the MAC OUI prefixes of printers in an organization. However, you likely purchase multiple printer brands and models, so more than one MAC prefix is necessary. Yet under DHCP fingerprinting, it may be possible just to allow any fingerprint associated with printers.

To be clear, a DHCP fingerprint can be spoofed. Yet it is more difficult than MAC spoofing, and it would require knowledge that DHCP fingerprinting is in use and a system that can craft packets.

[1] <http://lets-start-to-learn.blogspot.com/2015/02/dhcp-fingerprinting.html>

Fingerbank¹

Fingerbank¹ is an online DHCP fingerprint database

- Used by open source NAC Packetfence²
- Used by commercial NAC solutions as well

fingerbank

Contains thousands of DHCP fingerprints

- Supports manual lookups using free API key

```
curl \
  -X GET -H "Content-Type: application/json" \
  'https://api.fingerbank.org/api/v2/combinations/interrogate?key=YOURFINGERBANKAPIKEY' \
  -d '{"dhcp_fingerprint": "1,3,6,15,31,33,43,44,46,47,121,249,252"}'
```

Fingerbank¹

Fingerbank is an online cloud service housing thousands of DHCP fingerprints. Their DHCP fingerprinting solution is integrated and fed by their open source and commercially supported Packetfence² solution. This solution is also implemented in other commercial NAC implementations.

Fortunately, their solution is free, and the fingerprint database supports scripting as well as manual lookups. To perform a lookup a free account needs to be registered and an API key needs to be requested. The API key can then be used with scripts such as bash, PowerShell, or Python to perform fingerprint lookups automatically and to find more information about what DHCP clients are in your environment. This can be beneficial even in a passive capacity to better understand your environment.

[1] <https://fingerbank.org/>

[2] <https://packetfence.org/>

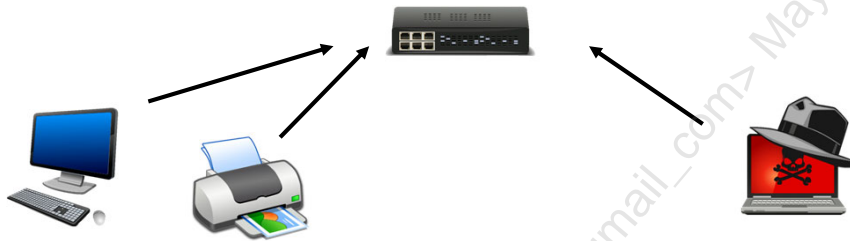
NAC Example

Authorized

- Windows authenticates using computer certificate
- Printer allowed by MAC
- Printer allowed by DHCP

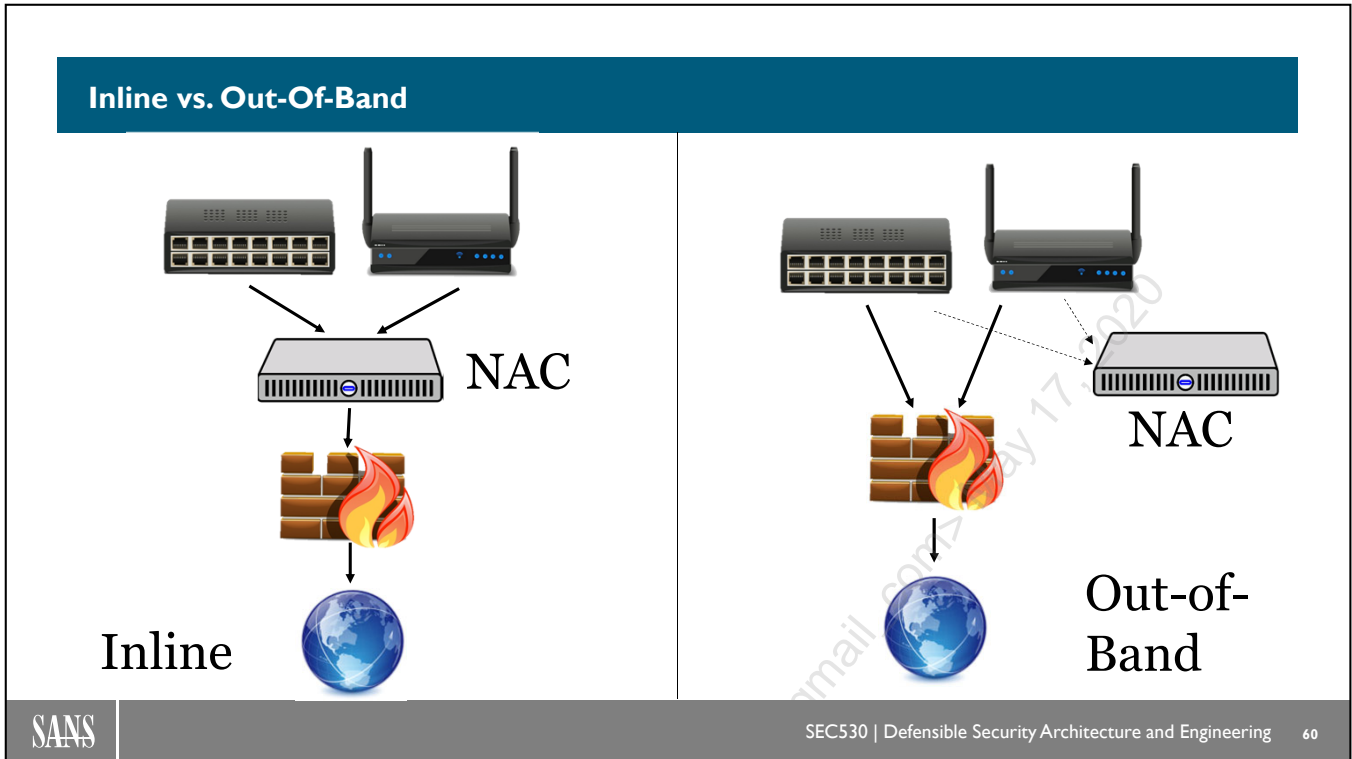
Unauthorized

- Personal device fails
- Misconfigured box fails
- Adversary cloning MAC may succeed



NAC Example

This slide demonstrates why a common NAC deployment would work. Starting on the left, the Windows desktop would be plugged into the switch and then authenticate using an x509 certificate. As long as it passed, it would be granted access and be considered an authorized device. Next, the printer would be plugged in, but since it does not support 802.1x port authentication, it would fall back to MAC authentication. If the MAC prefix or the full MAC address of the printer is allowed, it will be granted access and added as an authorized device. Finally, a hacker plugs his or her laptop into the switch. Likely it supports 802.1x, but the functionality is disabled so a fallback to MAC authentication occurs. However, the MAC address presented is likely not authorized so the device fails and gets added to the list of unauthorized devices. This may be followed up with the attacker looking at the back of the printer, cloning the MAC address and stealing the printer's network jack.



Inline vs. Out-Of-Band

When NAC is deployed inline, it means the NAC solution acts as the gateway for each VLAN. This allows central management and eases network complexity. However, it also introduces a potential point of failure. Inline NAC is only recommended when organizations do not have managed switches with 802.1X support.

Out-of-band deployments involve reconfiguring network devices to use NAC for port authentication. Typically, port authentication is a combination of 802.1X, MAC authentication, and possibly DHCP fingerprinting. With out-of-band, the NAC system can go offline with minimal impact to the network.

Deployment Considerations

Inline

PROs

- Works with any device
- Minimal design changes

CONs

- Potential failure point
- Can slow performance
- Requires more hardware
 - And High-Available server recommended

Out-of-Band

PROs

- Minimal hardware required
- Fail open or fail closed
- Dynamically change ACLs
- Dynamically change VLANs

CONs

- Requires changes to network infrastructure

Deployment Considerations

Both inline and out-of-band have their advantages and disadvantages. Because inline requires all traffic to funnel through the NAC solution, it requires more resources and is highly recommended to have a second or third unit acting as high-available failover units. Because traffic is going through the NAC system, it also has the potential of becoming a performance bottleneck. Realistically, inline deployment modes should only be considered for deploying NAC to legacy devices. Legacy devices are the main reason inline NAC is available. Not all switches or access points support NAC integration and features like 802.1X. These devices can be deployed to an inline NAC solution, and now they can be controlled by the NAC solution. Keep in mind, inline mode places devices in the same layer two networks once authorized.

Out-of-band is the recommended deployment method, but it requires network devices that support NAC integration protocols like 802.1X or SNMP traps. Unfortunately, configuring these protocols can be complex and difficult so not only do you need managed network devices that support the capabilities, but you need network engineers who know how to configure them. Once integrated to an out-of-band NAC solution the NAC solution is able to dynamically configure the switch using real-time VLAN and/or ACL assignment based on device authentication. Also, an out-of-band NAC solution can be configured to fail open or fail closed should the NAC system go offline. This typically is configured at the network device and is one or two commands specifying the expected failure response.

Captive Portal

Ideally device passes initial authentication methods

- Captive portal can handle failed devices or users

Design is flexible and dynamic

- Terms and conditions only
 - Gives guest VLAN access
- AD authentication
 - Provides limited production access

Captive portal could be forced even with authentication

Captive Portal

A NAC's captive portal is generally used to handle unauthenticated devices. This means that a device has been plugged in or wirelessly connected that failed to authenticate via 802.1X, MAC authentication, and DHCP fingerprinting. Rather than just failing a device or user a captive portal can be displayed. Remember, for this to work a NAC solution has to control traffic which usually means it hands out DHCP for the unauthenticated VLAN and sets itself to the default gateway and DNS server. This makes it so that when the end user tries to browse out to a website the captive portal is displayed instead. This is similar to trying to gain internet access at a major hotel.

Generally, the captive portal is considered as a go or no-go decision. Yet it does not have to be. For example, a captive portal can be a multi-option responsive form. If you are not an employee, you could select the option to agree to terms and conditions to gain guest internet access. If you are an employee, you could log in with AD credentials and gain full or limited network access. Part of AD authentication could be to walk the user through a series of steps to have them manually request a user or device certificate and then reboot for 802.1X authentication.

The captive portal could also be required on top of normal authentication measures based on conditions. For example, maybe an organization wants to require that each time a new user and device pair is seen at given locations that the end user must log in with AD credentials and submit an explanation for the new device use. This could be something as simple as "I have received a new work laptop." The point is, a captive portal is dynamic and as flexible as you make it assuming the NAC solution in use allows customizations.

Quarantine

Authorization should not be static

- NAC can dynamically control access

Conditions that may affect access

- Peer-to-peer use
 - Or software installed
- Abnormal connections
- IDS alerts
- Endpoint suite alerts



Quarantine

If a device or user is found performing actions that are not authorized or malicious in nature the system can be quarantined. Note that the quarantine can be automated or manual. Automatic quarantine needs to be done with extreme caution. For example, a NAC solution can be configured to integrate with an IDS or can have an IDS module. This means that alerts could trigger quarantine based on severity. The problem is an attacker could spoof bad traffic to get multiple systems quarantined. Thus, caution is necessary.

Manual quarantine, however, can be built into incident response procedures or even general process workflows. For example, if a machine is confirmed as being infected it can be moved to quarantine. If a machine is infected and the malware may be advanced, it could be moved to a special quarantine. Again, with NAC access can be dynamic. Systems could also be placed in quarantine based on business processes. For example, a desktop or laptop belonging to a user on vacation may be placed in quarantine if it is physically connected to a switch or wireless.

Statement of Health (SoH)

NAC agents are required for real-time health monitoring

- Agentless claims are false or not real-time
- Yet a third-party agent usually is not deployed

Microsoft has built-in **NAC agents** since XP/Server 2003

- Mac and some flavors of Linux have similar abilities
- NAC agent monitors for key changes to system
 - Sends **SoH** on initial connection and over time
 - **SoH** sent on changes such as firewall disabled

Statement of Health (SoH)

Microsoft operating systems since Windows XP and Server 2003 have a built-in network access protection agent. This agent provides the capability to monitor the host operating system for specific things like running antivirus, firewall status, and patch status. This agent also communicates with a NAC service through 802.1X requests to a switch which are then submitted as a RADIUS request to a NAC solution. Previously this went to a NAC solution which would communicate with Microsoft service running the Network Policy Server (NPS) role. However, Microsoft made NPS end of life, and now many NAC solutions handle the policy checks directly.

The fact that Windows systems have a built-in agent means that NAC can respond in real-time to a system's health. If the firewall is disabled, NAC can change the system's access. If antivirus is not running, again, NAC can change the system's network access.

Post Authentication Checks

Post authentication checks are key to dynamic access

- Statement of Health is one form of post check

Other checks can be custom or built-in integrations like:

- Vulnerability scan - Level of risk could change access
- Alternative authentication checks
 - **SNMPv3, SMB, WMI, SSH, HTTPS**
- Port scanning and **fingerprinting**
- Custom script - Choose your destiny

Post Authentication Checks

One of the most frequently overlooked NAC capabilities is post authentication checks. These checks allow a NAC to dynamically change how an endpoint is treated based on the result of a check ran after a client is already authenticated and authorized. Post authentication checks are key for a zero trust architecture. Just because a client passed initial authentication does not mean it should be trusted forever.

Some of the more powerful capabilities come with re-authenticating a client. For example, assume a printer was initially authenticated with MAC authentication or DHCP fingerprinting. Neither one of these is truly a trusted authentication method. Yet printers have web interfaces and could have an HTTPS authentication attempt ran against them to validate it is a corporate device. Other devices can be authenticated with SNMPv3 or SSH. Therefore, the post-authentication check may be a better form of device authentication and can be continuous.

Remember that access should be dynamic. In most cases, failing a post-authentication check does not mean access is completely removed. Instead, a device can accumulate or lose trust. Initially, maybe the device can still access internal resources, but internet access is removed for non-essential business purposes. If the device continues to misbehave access continues to be removed, or a captive portal is displayed.

Electric Fence

Mick Douglas refers to dynamic access as an electric fence

- Behave as normal, and you have full access
- Touch the fence, and a digital shock occurs

Electric shock results in an **automated digital response**

- Quality controls (QoS) slows access
- ACLs remove access
- PCAP recording kicks in
- User is notified of digital shock



Electric Fence

Mick Douglas, a SANS instructor, once was referring to dynamic access as an electric fence. So long as system behave as intended and stay within their expected duties nothing happens. The electric fence is there to protect them just as much as it is to keep them from getting out. However, if the fence is touched, a negative stimulus is triggered. In the real world touching an electric fence would result in pain. In the digital world, this is an automated digital response of some kind.

The digital response can be anything. If a business has a low-risk tolerance, the response may be to start cutting off access to critical systems or the internet. If a business has a high-risk tolerance, they may choose to slow down connections using QoS simply. NAC is able to give access and take it away by either moving systems to different VLANs or applying ACLs. Some actions may actually ask for human interaction. For example, if a user were to make abnormal connections the system may pop up and notify them that unusual activity has been observed and to let them know they are being monitored. By simply providing awareness the user may change their behavior. Of course, this may not be wanted. Again, it is the possibilities that make NAC awesome.

NAC Problems

Organizations are restricted by time and money

- NAC is time-consuming to setup and can be expensive
- As a result, many organizations do not have NAC

Even if deployed it is likely not deployed everywhere

- Virtual environment does not support NAC well
- Switches in data center typically do not need NAC

Other forms of device discovery are necessary

NAC Problems

While NAC seems ideal to discover authorized and unauthorized devices, it has some problems.

1. It is expensive both monetarily as well as operationally—The amount of labor required for ongoing management can be high.
2. NAC does not work in certain segments of the network. For example, it is likely not going to work in a data center.
3. Some devices are not good candidates for NAC. This could be switches in a data center, or it could be a special subnet out on the network. For example, if you have a subnet that only contains phones and a firewall is in place to only allow the phones access to specific services, it may not be worth the time to enable NAC. (Keep in mind, you may be trading off a good detection strategy for the convenience of not implementing something like MAC-based NAC.)

Due to these reasons, it is almost certain that a NAC solution will not contain a full list of authorized and unauthorized devices. Therefore, other forms of discovery are necessary.

Network Access Control Review

NAC is complex but compliments zero trust

- Device/user first must authenticate
- Supports post authentication checks
- Integrates with other systems for system monitoring
- Lots of support for dynamic access conditions

CIS Critical Control 1 has been the same for a long time

- Inventory of Authorized and Unauthorized Devices
- NAC supports passive and active inventory control

Network Access Control Review

NAC may be complex, but it is worth the effort. The first CIS critical control has been inventory authorized and unauthorized devices for a long time. NAC is positioned well do so for both reporting on inventory and actively controlling access to the network. With NAC's capabilities to dynamically change access, pop captive portals, and perform post authentication checks it fits well within a zero trust architecture.

Course Roadmap

- Day 1: Defensible Security Architecture
- Day 2: Network Security Architecture
- Day 3: Network-Centric Application Security Architecture
- Day 4: Data-Centric Application Security Architecture
- **Day 5: Zero Trust Architecture**
- Day 6: Capstone: Design, Detect, Defend

CURRENT STATE ASSESSMENT, SOCS, AND SECURITY ARCHITECTURE

1. Zero Trust Architecture
2. Credential Rotation
3. Securing Traffic
4. EXERCISE: Network Isolation and Mutual Authentication
5. Host-Based Firewalls
6. Network Access Control (NAC)
7. **Segmentation Gateways**
8. Security Event Information Management (SIEM)
9. EXERCISE: SIEM Analysis and Tactical Detection
10. Log Collection
11. Audit Policies
12. Host Hardening
13. Patching
14. Tripwires and Red Herring Defenses
15. EXERCISE: Advance Defense Strategies

Course Roadmap

The next section covers Segmentation Gateways.

Network Agent

Zero trust uses the concept of a network agent for access

- A **network agent** is a user and device combined

The network agent is used to determine authorization

- User + corporate laptop = what access?
- User + personal laptop = what access?
- User + corporate phone = what access?

Access to data should be controlled by network agent

- Rather than traditional internal vs. external perimeter

Network Agent

One response to the lack of protection traditional perimeter protection offers is the use of identity. Some online articles or solutions to perimeter security recommend that identity be used as the new perimeter¹. The concept that identity is the new perimeter is a step in the right direction. This concept of identity is not new. Single-sign-on and federation services such as SAML or OAuth have existed since the early 2000s. Identity management goes a long way in tightening controls and limiting access to data.

The issue with identity management is that most solutions focus entirely on the user. User authentication whether with username and passwords or multifactor authentication is often attributed to the identity. Yet identity should be based on both the user and device being used. The combination of the user and device equates to what the zero trust model refers to as a network agent. Access to data should be controlled based on the combined identity of a network agent.

[1] <https://www.darkreading.com/is-identity-the-new-perimeter/d/d-id/1139110>

Planes of Authorization

Control plane is core of zero trust

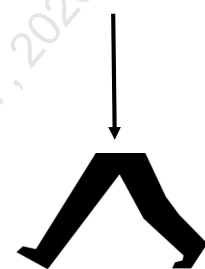
- Handles central authentication and global policy
- Authorizes requests and authorizes access

Data plane handles connections

- Establishes connection mediums
- Provides switching and routing
- But only if control plane continues to authorize access

Ideally is one device but for practical reasons is multiple

Request for
access



Planes of Authorization

Under zero trust, network access is a two-step process. The first step is to have access authenticated and authorized via the control plane. The control plane is the centralized control or brain. This brain makes logical decisions about who and what passes authentication and controls ongoing authorization. Its job is to monitor and continuously renew valid connections. However, applying business logic and calculating access authorization is not a fast process. Therefore, the control plane handles logic and provides access to a data plane. The data plane is the switch and network fabric that handles connections. Think of it as traditional switching.

For zero trust to work, access must be continuously verified. This requires some form of management software to act as a centralized control plane. While it would be ideal to have a single control plane the truth is an organization is likely to have multiple. For example, an NGFW, reverse web proxy, and identity management solutions can all act as a control plane. Each of these then handles data plane access differently or not at all. The concept of control plane and data plane is more geared to zero trust architecture of network control devices like NGFWs and reverse proxies. These devices handle logic access controls and then hand connections off to direct network access via hardware.

Segmentation Gateway

Issues with firewall tiering (firewall sandwich)

- Increases complexity and makes management difficult

Zero trust pushes towards central controls and automation

- Push towards the use of segmentation gateways

NGFW or SDN at the core rather than tiering firewalls

- Requires high-speed links (10 Gb+)
- Focuses on users and endpoints
- Heavy whitelist approach

Segmentation Gateway

Forrester coined the term zero trust architecture as well as the concept of a segmentation gateway. A segmentation gateway is a platform that centrally automates and controls network controls regardless of source device and user. In truth, a segmentation gateway is a concept or application of pre-existing technologies in a different way.

The most common way of implementing segmentation gateways is to use an NGFW solution as a core router. Since all traffic whether internal to external, external to internal, as well as internal to internal must traverse core routing an NGFW at the core provides central control. Historically a firewall was never recommended at the core due to latency and performance reasons. Additionally, costs were astronomical. However, NGFW costs have continued to decline while hardware capabilities and speed have continued to increase.

By having centralized control over network access an organization benefits from:

- Simplified management
- Increased visibility
- Centralized enforcement

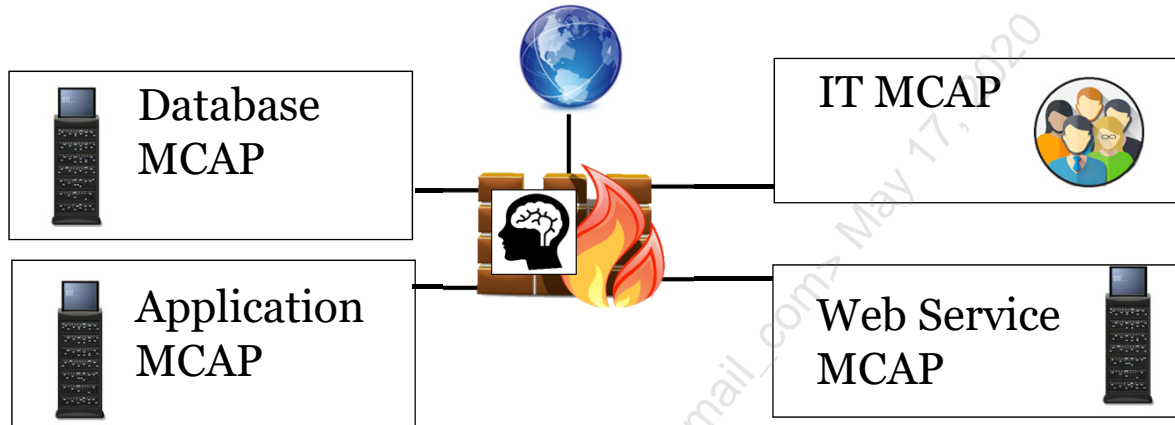
[1] https://www.nist.gov/sites/default/files/documents/2017/06/05/040813_forrester_research.pdf

[2] <https://www.paloaltonetworks.com/resources/videos/zero-trust.html>

Micro Core and Perimeter (MCAP)

MCAP creates logical zones of trust and functionality

- Full design should include intra-zone connections



Micro Core and Perimeter (MCAP)

The use of a segmentation gateway allows the enforcement of micro core and perimeter (MCAP) trust zones. MCAP is the ability to group users and devices of similar trust levels to enforce access controls. MCAP is not bound by VLAN segmentation. For example, it is possible to have users and devices on the same subnet be split into separate MCAP zones. Logical implementations are easier to establish with traditional segmentation such as VLANs, but a router or firewall acting as a segmentation gateway directly handles routing, so it is able to segment as necessary logically.

Proper MCAP groups should be based on grouping based on similar application use and data access requirements. Grouping different levels of trust such as users accessing confidential data with users that access standard data is not recommended due to the chances of accidentally granting access to confidential data. Depending on the security device in use an MCAP may only be able to place logical access constraints to network connections traversing a layer three boundary. This means that host-based firewall filters are still necessary to secure layer two connections or implementing private VLANs.

Different Approach

Segmentation gateways use old technologies

- Just implemented in a different way

Security hardware today supports core backbone speeds

- NGFWs available with 100 Gb interfaces and well over 100 Gb inspection speeds
- SDN or solutions to control switch fabric are maturing
- Reverse proxies and identity management are strong

In many cases, L7 inspection is less important than identity

Different Approach

Today NGFWs from multiple vendors support 100 Gb interfaces with combined inspection speed fabrics upwards of 600 Gbps. This means the technology is there assuming you can afford it. However, the price of even these monster firewalls is often affordable given the security capabilities it allows an organization to enforce.

Keep in mind, while a SDN or NGFW is capable of full layer seven inspections that the implementation as a segmentation gateway focuses on authentication and authorizing. These means NIPS, Antivirus, and other security checks are not the key to enforcement. Those focus on blacklist checks. A segmentation gateway focuses on whitelist checks. All connections should be verified within an allowed set of parameters. This means the focus is on limiting access based on user, device, layer four ports, and application identification.

Having an NGFW at the core would allow deeper security applications for untrusted and outdated systems, but the core capability is logically verifying a connection should be allowed.

MCAP and Network Agent

MCAP and access should be based on network agent

- Developer on corporate desktop has access to source code
- Developer on mobile device does not

Trust is calculated by user, device, and other factors

ID	Name	From	To	Source	Destination	Schedule	Service	Applications	Action
39	Developer Generic	<input type="checkbox"/> any	<input checked="" type="checkbox"/> lan	<div style="border: 1px solid red; padding: 2px;"> all Developers Mobile Assets </div>	Production App Servers	always	ALL	RDP SMB.v3 SSH	DENY
38	Developer Production	<input type="checkbox"/> any	<input checked="" type="checkbox"/> lan	<div style="border: 1px solid red; padding: 2px;"> all Developers Corporate Workstations </div>	Production App Servers	always	ALL	RDP SMB.v3 SSH	ACCEPT

MCAP and Network Agent

The industry push is for any device, anywhere access. This is indefensible because it allows access when it is unnecessary or should not be taking place. As an example, should a developer who has a corporate desktop and mobile device be able to push code into production from both his or her desktop and mobile phone? Likely, the business intent is for them to push data into production only from their authorized desktop. This makes sense as the desktop is not portable and has multiple agents and logging mechanisms in place. The mobile phone is more limited as it does not support. Also, it is not practical to access or push code on a mobile phone, therefore, why should it be allowed?

In this slide, an NGFW is shown with two rules. The first rule blocks developer access to production application servers if it is a mobile asset. The second rule allows developer access to production application servers if it comes from a developer on a corporate workstation. Notice, both of these rules are based on specific applications such as SSH or RDP.

Inventory Automation

Key to MCAP grouping is device and user integration

- Users and groups usually sync with Active Directory
- Device integration requires commercial add-on solutions
 - Or simple scripts that hook REST APIs or SSH

Network agent needs real-time application of user + device

```
import FortigateApi
FG = FortigateApi.Fortigate('10.0.0.1:8443', 'root', 'admin', 'password')
FG.AddFwAddress('testServer', '10.0.0.2/32')
print FG.GetFwAddress('testServer')
```

Inventory Automation

One of the main challenges with implementing a segmentation gateway is that most implementations focus on end user identification and control rather than device and user identification. In fact, many commercial solutions have limited or no capabilities to associate devices with rules. On an NGFW device identification is typically a service that when enabled will passively identify and inventory assets. Passive discovery does not provide accurate means of authenticating a device. Therefore, it does not require the zero trust model of verifying everything.

Instead, an NGFW can be fed accurate inventory information. This slide demonstrates a simple python script that is setting up an address object in a FortiGate firewall. Address objects and groups can be created and manipulated with scripts and with API interfaces for most commercial vendor firewalls. While this task seems daunting, it is a fairly simple script. FQDN objects can also be used but should only be used if DNS is secure such as with Windows secure DNS implementation.

Real-Time Device Inventory

NAC and VPN solutions require authentication before providing network access

- Post-authentication task can feed segmentation gateway
 - Such as running script to update address objects
- Also, can be achieved by using centralized logging
 - Send logs to Security Incident Event Management (SIEM)
 - Use SIEM to react to NAC or VPN logs in near real-time

Real-Time Device Inventory

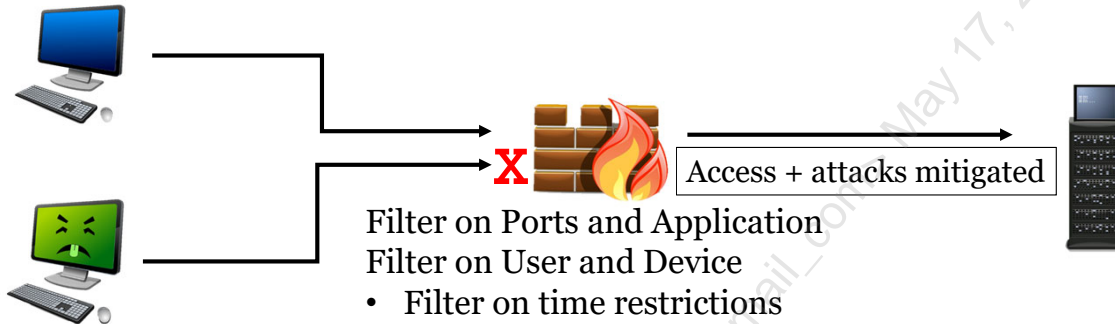
Automation is critical for cyber defense. The previous slide shows an example of using python to create address objects in a firewall. However, for firewall rules to work, they need to be as real-time and accurate as possible. If an organization is using NAC or VPN authentication to authorize access to the network, then scripts can be kicked off as an end result of passing authentication. For example, a VPN solution may support running a post-task when a user connects to the network. This task could be to run the previous python script logic to update an address object. For this to work and be secure, the post-task would need to be supported on the server side of a NAC and VPN solution.

Alternatively, logs from NAC and VPN systems could be sent to a centralized logging system such as a SIEM. Since this happens almost instantly, the logs could be used to trigger a script to be run using the information found in the logs within the SIEM. For multiple solution support, a SIEM is likely the best bet to pull this off.

Centralized Protection

Internal firewalls provide centralized access controls

- Helps push filtering as close to source as possible
- Endpoint firewall is granular but deep within network



Centralized Protection

By implementing a segmentation gateway, a network solution is capable of denying access as close to a source as possible. This is important as it can help mitigate denial of service attacks, data access, and possible service exploits. It also optimizes bandwidth and resources. Host-based firewalls and controls are still necessary as they are likely to be more granular and specific to each system.

Centralized controls help achieve significant filtering against a network agent as well as limit access to whitelisted and known applications. Another capability a segmentation gateway supports is applying time constraints against authorization rules. For example, if certain users do not have remote access and only work 8 AM to 5 PM Monday through Friday then firewall rules can be set to only allow access during this time frame. Then if one of these systems is compromised and attempts to move laterally or phone home it will bump up against a firewall rule preventing the unauthorized activity and providing a red flag for easy detection.

Dynamic Authorization

Abnormal conditions should be monitored and reacted to

- **Temporal** - Access outside normal user window
- **Geographical** - Access from different location
- **Behavioral** - Access to resource user does not normally use
- **Frequency** - Last access or volume of device/user use
 - Or number of requests over time

Deviation from norm may dictate additional checks

- Multifactor authentication
- Approval from manager or administrator

Dynamic Authorization

Both a segmentation gateway and NAC solutions have the capability to alter access dynamically. The questions then become, is modifying access on the fly a good idea and what conditions should affect access. Dynamically altering access should be done but needs to be done with care. Failure to apply care leads to self-triggered denial of service. This is why dynamic access is not intended to be a binary yes and no decision. Instead, it should be risk and reward. A user continuously showing appropriate behavior may be granted additional points and thus make it harder for them to lose access without rapid abnormal behavior. On the contrary, a high-risk user should have to jump through more hoops to verify access and have access to critical data.

Abnormal conditions that affect access can be anything, but common conditions are temporal, geographical, behavioral, and frequency anomalies. A breakdown of these is below:

- Temporal - Based on monitoring user logins and knowing when a login occurs outside normal user logins. This includes a user logging in at 2 AM when they work 8 AM - 5 PM.
- Geographical - Based on monitoring user logins by the source of login. This includes identifying things such as a user logging in from home and then suddenly having a login from another country.
- Behavioral - Based on monitoring what a user does with his or her access. This would include things such as monitoring data assets and flagging access denied errors or accessing a system legitimately that the user has never used before.
- Frequency - Based on monitoring the rate of occurrence of something such as user logins or how many data sources are being accessed at a given time

Segmentation Gateway Review

A segmentation gateway provides centralized:

- Network agent (user + device) access controls
- Time constraints and limitations
- Data-centric port and application controls
- MCAP trust zoning

NGFW can be deployed as a segmentation gateway

- Capabilities increasing and hardware cost decreasing
- Able to automate via robust API support

Segmentation Gateway Review

Segmentation gateways are centralized systems designed to handle authentication and authorization for devices requesting network access. Commonly these devices are NGFWs or SDNs that can control access based on the user and device combination. By combining these into a network agent, specialized pockets of trust can be established, and it becomes easier to implement rules following the principle of least privilege.

A segmentation gateway may need data fed into it from other systems such as NGFW being updated via API calls. This allows automation of dynamic access.

Course Roadmap

- Day 1: Defensible Security Architecture
- Day 2: Network Security Architecture
- Day 3: Network-Centric Application Security Architecture
- Day 4: Data-Centric Application Security Architecture
- **Day 5: Zero Trust Architecture**
- Day 6: Capstone: Design, Detect, Defend

CURRENT STATE ASSESSMENT, SOCS, AND SECURITY ARCHITECTURE

1. Zero Trust Architecture
2. Credential Rotation
3. Securing Traffic
4. **EXERCISE: Network Isolation and Mutual Authentication**
5. Host-Based Firewalls
6. Network Access Control (NAC)
7. Segmentation Gateways
8. **Security Event Information Management (SIEM)**
9. **EXERCISE: SIEM Analysis and Tactical Detection**
10. Log Collection
11. Audit Policies
12. Host Hardening
13. Patching
14. Tripwires and Red Herring Defenses
15. **EXERCISE: Advance Defense Strategies**

Course Roadmap

The next section covers Security Information and Event Management (SIEM).

Solid Detection Required

Scripting or commercial solutions update the control plane

- But dynamic access necessitates custom trust levels
- Cannot be done without low false positive detection

Level of detection maturity and capabilities required

- Integration between disparate solutions necessary

Examples:

- NSM + Sandbox + Flow Data + NGFW + Scripts
- Security Incident Event Management (SIEM) + NAC

Solid Detection Required

The key to implementing strong preventative controls is making sure they get implemented the first time correctly. This requires a few things. First off, it requires sufficient data to make educated decisions. For example, what does normal vs. abnormal look like? What level of deviation should start to affect access? Fidelity of data is key. Also, initial implementations need to be tested without inflicting self-denial of service.

How can this be done? The answer is with solid detection capabilities and reporting. Logs and data from multiple devices need to be collected and analyzed till an organization is confident in their decision making and application. Log analysis is needed even for a single device. Take an NGFW as an example. Firewall rules should not go straight into prevention without testing. Instead, firewall rules should be enabled in allow mode initially but with logging enabled. This provides data for decision making. This type of data analysis is critical for implementing variable trust. While capable of doing this without centralized log collection it is much easier to do when data is stored in a single pot for analysis.

For example, variable trust can be implemented with scripts on disparate solutions such as an NSM or malware sandbox. These scripts could then dynamically change an NGFW or NAC solution through API calls. However, a centralized log system such as a SIEM would provide more data points and higher confidence in decisions. Plus, a SIEM solution usually has an alert engine that can make calls to other solutions such as NAC or NGFW based on the automated analysis.

Security Information and Event Management (SIEM)

What is a SIEM used for?

- Centralized log collection ✓
- Advanced alerting ✓
 - Systems automation ✓
- Analysis system ✓
- Compliance repository ✓
- Big data analytics platform ✓
- Threat intelligence ✓



SIEM != log collection
It can do so much more

Security Information and Event Management (SIEM)

When people think of SIEM, they tend to think of centralized logging. But a SIEM is so much more. In fact, if an organization has a compliance requirement for centralized logging and retention there are significantly cheaper options that can be implemented. A SIEM is an analytical platform built for analysis and automation. Data is consumed in log format, and then mass analysis ensues.

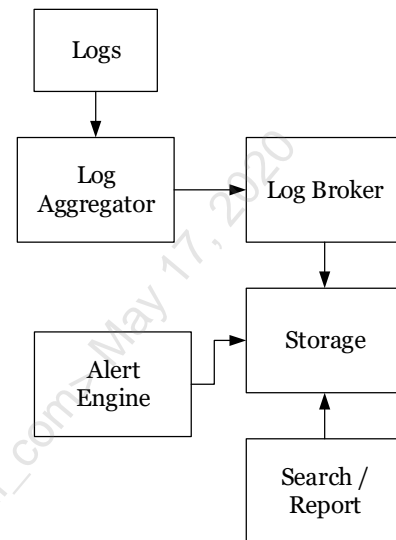
A SIEM should be a tactical weapon yielded by organizations. It supports centralized logging with log parsing, filtering, and enrichment. It normalizes and correlates disparate data. It provides dashboards and visualizations used to help drive analysis. It integrates with threat intelligence services, and it supports generating custom alerts. One of its most underutilized components is its capability around automation.

Multiple times variable trust has been mentioned. Also, the electric fence concept has also been mentioned. Both of these rely on solid detection and automation tools to implement. A SIEM has the capability to reach out to other systems automatically, and a SIEM has high fidelity data to make decisions with.

SIEM Components

A SIEM consists of multiple pieces

- Log collectors (agents, scripts, etc.)
- Log Aggregator
- Log Broker
- Storage
- Search / Report
- Alert Engine



SIEM Components

A SIEM is not complicated; you just need an understanding of what each component is for. Getting each piece to work in synergy with one another is what organizations struggle with.

Log Collector - While not directly part of the SIEM, log collection is a critical piece of the overall SIEM architecture. This can be done many different ways such as through the use of agents, agentless log collection, and scripts.

Log Aggregator - This acts as central collection points of logs. They ingest raw logs and have the capability to parse and add context to the log. A log aggregator can also be used to generate alerts early on in log processing.

Log Broker - A broker is a temporary storage location for logs. Logs go into the broker and are stored until an aggregator can pull them out. Many times, this results in two sets of log aggregators: one to accept logs and put them into a log broker and one to pull logs from a log broker and parse them. A broker's primary purpose is providing redundancy and the ability to handle fluctuations in the log collection process. If processing gets backed up, the logs will not be dropped. They simply stay with the broker until processing catches back up.

Storage - Once logs are finished being processed they end up being stored in a backend storage node. The storage node is responsible for storing logs on disks and retrieval of those logs. How the storage system handles logs varies from solution to solution.

Search / Report - A report node is typically used to search and report on logs that are sitting in the storage node(s).

Alert Engine - An alert engine is used to search for logs in the storage nodes and trigger alerts based on defined workflows.

Log Inspection

All system and network access needs verified

- Ad hoc reporting is inefficient and difficult to do
- Central log collection and scripting is low cost
 - But also suffers from major deficiencies

SIEM log collection and analysis is recommended

- Focus should be on key log sources
- And inspecting them for expected network and system use
- Use log enrichment to enhance analysis

Log Inspection

The core of SIEM revolves around its capability to analyze data. Central logging solutions and ad hoc scripts simply are inefficient. They are inefficient because they require the organization to figure out how to access and use the data in a proper fashion. While it may be possible to establish a few key analysis scripts, the solutions fail at large scale analysis. They also do not provide a GUI for folks who are not as well versed with command line data manipulation.

A SIEM solution is designed not only to collect data but also to analyze it. Analysis typically allows scripting, web GUIs, and potentially other thick-client applications to sift through data. The main thing to keep in mind with a SIEM is that it is only as good as the data that is in it. Many organizations struggle with this as their mentality is to collect everything. This usually starts off well but then due to costs gets limited to everything but only from key servers or services. A well designed SIEM stays tactical by focusing on data sources that matter and applying log enrichment to the logs during data ingestion. This allows the SIEM to focus on inspecting logs to look for out of the ordinary events.

Log Enrichment

query: www.google.com

Enriches to this

query: www.google.com

subdomain: www

parent_domain: google

registered_domain: google.com

creation_date: 1997-09-15

tags: top-1m

geo.asn: Google Inc.

frequency_score: 18.2778256342

parent_domain_length: 6

Log Enrichment

All SIEM solutions have the capabilities to augment and enrich logs either during ingestion or after logs are stored to disk. Enrichment simply means adding additional context to a log. Context is critical to drive analysis as well as to add new detection capabilities. For example, this slide demonstrates taking a single field called query with a value of www.google.com and enriching it to add eight new fields. This is an example taken from a DNS log.

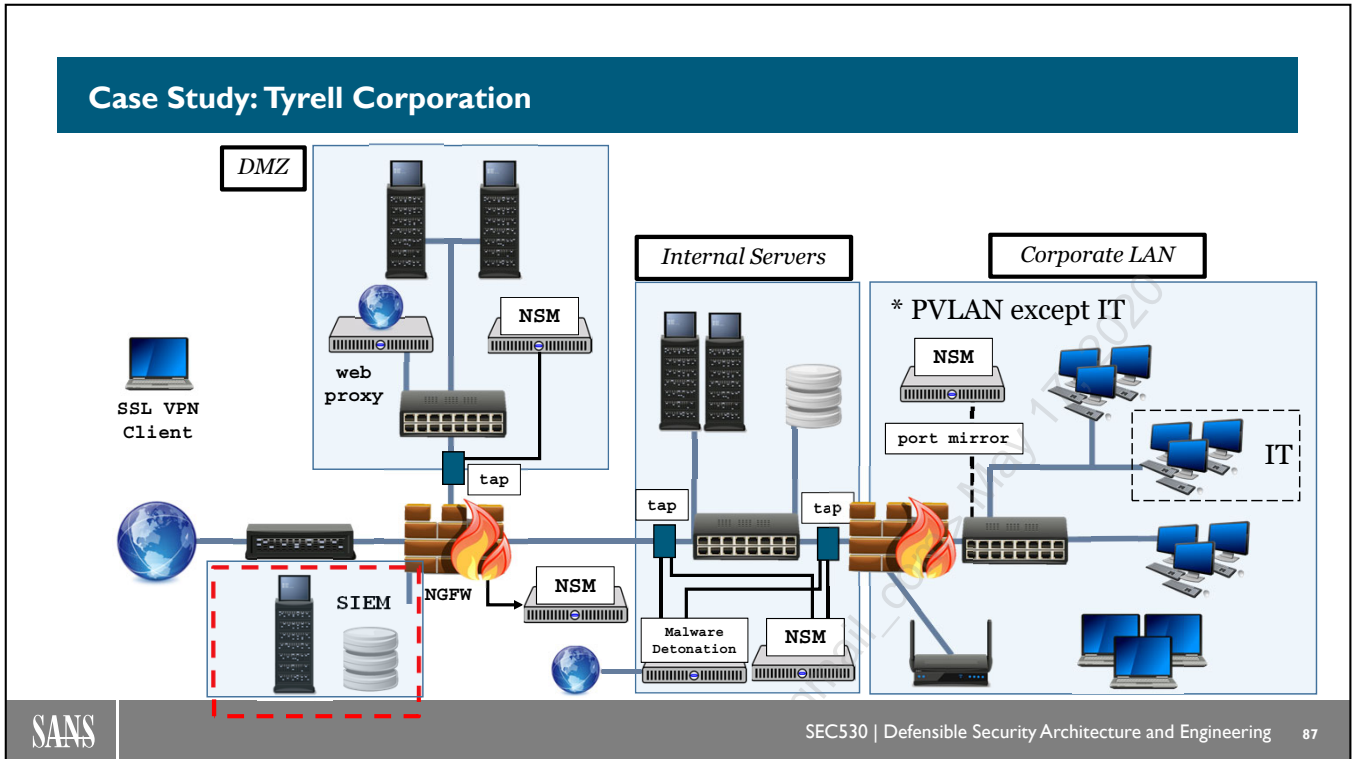
The first couple enrichment fields break www.google.com into pieces. This is necessary as some enrichment techniques only work against parts of a DNS domain. For example, WHOIS1 creation dates and Alexa2 top 1 million lookups should be performed against a registered domain such as google.com. These enrichment technique values are stored in the creation_date and tags fields in this slide. Geographic information can be looked up using IP addresses derived from a DNS entry. In this example, the ASN is gathered which describes the entity that owns the IP address. The frequency_score field is an example of using Mark Baggett's freq_server.py3 solution that performs high-speed natural language processing of a string to see if the string matches the expected character frequencies. This is done to find the existence of randomness or malicious data. The last enrichment field parent_domain_length simply calculates the length of a string.

All of these examples of enrichment show how much context can be added to a log.

[1] <https://whois.icann.org/en>

[2] <https://support.alexacom/hc/en-us/articles/200449834-Does-Alexa-have-a-list-of-its-top-ranked-websites->

[3] https://github.com/MarkBaggett/MarkBaggett/blob/master/freq/freq_server.py



Case Study: Tyrell Corporation

This diagram represents the Tyrell Corporation's design. In it, a SIEM platform has been added and is hanging off the firewall as a dedicated zone. The SIEM platform will be utilized to gather data from endpoints, network devices, and security controls. Once the data is centralized, it will be enriched to add automatic context and then will be reviewed to find unauthorized activity and anomalies.

Course Roadmap

- Day 1: Defensible Security Architecture
- Day 2: Network Security Architecture
- Day 3: Network-Centric Application Security Architecture
- Day 4: Data-Centric Application Security Architecture
- **Day 5: Zero Trust Architecture**
- Day 6: Capstone: Design, Detect, Defend

CURRENT STATE ASSESSMENT, SOCS, AND SECURITY ARCHITECTURE

1. Zero Trust Architecture
2. Credential Rotation
3. Securing Traffic
4. **EXERCISE: Network Isolation and Mutual Authentication**
5. Host-Based Firewalls
6. Network Access Control (NAC)
7. Segmentation Gateways
8. Security Event Information Management (SIEM)
9. **EXERCISE: SIEM Analysis and Tactical Detection**
10. Log Collection
11. Audit Policies
12. Host Hardening
13. Patching
14. Tripwires and Red Herring Defenses
15. **EXERCISE: Advance Defense Strategies**

Course Roadmap

The next section covers the concept of Zero Trust Architecture.

Exercise 5.2: SIEM Analysis and Tactical Detection

- Exercise 5.2 is in the digital wiki found in your student VM (recommended)
- Alternatively, you may use your Workbook

This page intentionally left blank.

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

Course Roadmap

- Day 1: Defensible Security Architecture
- Day 2: Network Security Architecture
- Day 3: Network-Centric Application Security Architecture
- Day 4: Data-Centric Application Security Architecture
- **Day 5: Zero Trust Architecture**
- Day 6: Capstone: Design, Detect, Defend

SIEM Architecture and SOF-ELK

1. Zero Trust Architecture
2. Credential Rotation
3. Securing Traffic
4. **EXERCISE: Network Isolation and Mutual Authentication**
5. Host-Based Firewalls
6. Network Access Control (NAC)
7. Segmentation Gateways
8. Security Event Information Management (SIEM)
9. **EXERCISE: SIEM Analysis and Tactical Detection**
10. **Log Collection**
11. Audit Policies
12. Host Hardening
13. Patching
14. Tripwires and Red Herring Defenses
15. **EXERCISE: Advance Defense Strategies**

Course Roadmap

Our next section focuses on log collection techniques and strategies.

Log Collection

Logs must be collected prior to inspection

Typically done with:

- **Log Agents** – Requires software
- **Agentless** – Requires credentials and scanning of remote systems

Network devices often use syslog (NGFW, Proxy, DLP)

- Other methods exist, such as SNMP traps or APIs

Log Collection

Prior to building out advanced dashboards and having automated alerting, an organization needs logs to process. While the concept is basic in nature, a lot of planning should be done to facilitate this. Multiple methods exist for log collection, but the most common are through the use of log agents, agentless retrieval from a central system, syslog, API calls, and scripts.

Experience shows that many organizations spend little time planning out log collection. When done poorly, this can cause loss of logs, network interruption, and performance issues.

Syslog

Syslog is the most common network protocol for sending logs on the network

- Also, a built-in daemon for network devices and Unix Default is UDP on port 514
- Some systems support TCP but uncommon RFCs support TLS encryption¹
- Yet most systems only support Syslog over UDP without encryption

Syslog

The most common network protocol for transporting logs is probably syslog. Based on RFC 5424¹ syslog supports UDP or TCP as well as optional TLS encryption. However, syslog was adopted during the 1980s where everyone felt security was not needed and was in use long before an RFC was created. In fact, multiple RFCs have been created for syslog with the most notable first being RFC 3164², which is on BSD Syslog, followed by the most current of RFC 5424¹, which is based on the Syslog Protocol.

Due to early adoption by many applications and programmers, syslog messages tend to be inconsistent. Worse yet, initial builds of syslog daemons only supported UDP and, even today, built-in support is limited to UDP without TLS. Linux systems are slowly getting more support for TCP and TLS as newer operating system builds include rsyslog and syslog-ng. These are newer daemons with more built-in capabilities.

[1] <https://tools.ietf.org/html/rfc5424>

[2] <https://www.ietf.org/rfc/rfc3164.txt>

Syslog Devices

- Routers
 - Switches
 - Firewalls
 - Intrusion Detection Systems
 - Application Proxies
 - Wireless Access Points
 - Printers
 - Storage Devices (SAN/NAS)
 - Hypervisors (ESXi/Xen)
 - Unix / Linux
 - Data Loss Prevention
 - Behavior Analytics Solutions
 - Mac
 - Windows* - requires third-party agent
- The list goes on and on...

Syslog Devices

The number of devices supporting syslog is tremendous. Nearly every network, rack-mountable, or appliance-type device supports syslog. Linux and Mac operating systems natively support syslog and even Windows can have a third-party agent installed that takes Windows events and transmits them using syslog.

Because of the widespread use, you should know how to use syslog and, more importantly, how to accept and parse syslog messages.

Traditional Logging - Syslog

```
<81>Jan  4 14:43:13 logparse sudo: jhenderson : 1
incorrect password attempt ; TTY=pts/1 ;
PWD=/var/log ; USER=root ; COMMAND=/bin/su
```

PRI = <81>

Time/date = Jan 4 14:43:13

Source host = logparse

Source process = sudo

Message = jhenderson : 1
incorrect password attempt ;
TTY=pts/1 ; PWD=/var/log ;
USER=root ;
COMMAND=/bin/su

Traditional Logging - Syslog

This slide breaks out a traditional syslog log. It is important to understand syslog logs because they are still the de facto standard today even though better, more modern log formats are available. In this log five fields are shown as being parsed out. They are the PRI, timestamp, source host, source process, and message.

The PRI field is a special field that needs to be dealt with. This field is an integer wrapped within a less than and greater than sign. The integer is a mathematical calculation that stores the syslog facility and severity fields. An integer is used to minimize the amount of data required to be sent over the network. The match behind this field is as follows:

Facility = PRI / 8 and then rounded down to the nearest whole number

Severity = PRI - (8 * facility number)

Example: <189>

Facility = 189 / 8 which results in 23.625. This rounded down results in a facility of 23 which is local7.

Severity = 189 - (8 * 23) = 5. So, the severity is 5 which is notice.

Facility codes are as follows:

Numerical Code Facility

0	kernel messages
1	user-level messages
2	mail system
3	system daemons
4	security/authorization messages
5	messages generated internally by syslogd
6	line printer subsystem
7	network news subsystem
8	UUCP subsystem
9	clock daemon
10	security/authorization messages
11	FTP daemon
12	NTP subsystem
13	log audit
14	log alert
15	clock daemon (note 2)
16	local use 0 (local0)
17	local use 1 (local1)
18	local use 2 (local2)
19	local use 3 (local3)
20	local use 4 (local4)
21	local use 5 (local5)
22	local use 6 (local6)
23	local use 7 (local7)

Severity codes are as follows:

0	Emergency
1	Alert
2	Critical
3	Error
4	Warning
5	Notice
6	Informational
7	Debug

Syslog Message Limitations

Inconsistent

- Message formats and vary dramatically
- Requires additional parsing

Majority of systems are limited in message size

- Syslog over UDP often limited to 1024 bytes (RFC 3164¹)
 - Fragmentation typically not used with UDP
 - Standard MTU is usually 1500 bytes
- Syslog over TCP often limited to 4096 bytes

Syslog Message Limitations

Due to the early adoption and varying implementations of syslog, message consistency is almost non-existent. The message layout used by one vendor or application varies dramatically from the next. This requires per application or device parsing which adds a lot of overhead. As you will find out later, parsing is not difficult, but it is time-consuming.

Another limitation of syslog is the maximum log size supported. In RFC 3164¹, which pertains to BSD syslog, the maximum packet size for syslog over UDP is 1024 bytes. Even though syslog is used on non-BSD systems, many follow suit and will either drop or truncate a log packet over 1024 bytes. The reasoning behind this size limit is that UDP does not keep track of packets and, therefore, the loss of a single packet when using fragmentation would garble a message. Due to this and the fact that the standard MTU size is 1500 bytes, 1024 bytes was selected.

For TCP, this size is often limited to 4096 bytes.

While most logs will fit within the 1024 to 4096-byte range, if you use syslog to transport logs, you must know how large your events are going to be. You also need to know if both your syslog client and syslog server are held to these size limits. Many commercial SIEMs adhere to these size limitations. This is interesting, given that Windows logs can be over 30 K in size, and yet, many agents transmit Windows event logs over syslog.

Field Parsing

Log example:

1000 failed logons against administrator

Regex pattern:

`^[0-9]+ failed logons against [a-zA-Z]{3,}`

- Fields require pre-meditated parsing
- What happens if a new one is added?

Field Parsing

A key drawback to syslog is that custom fields all require manual parsing. This is usually done using regex. Regex stands for regular expressions. Regex is applied by specifying patterns to match against. If a match is found, then the string matching is extracted. In this slide, there is an example log of "1000 failed logons against administrator". The regex pattern used in the slide is "`^[0-9]+ failed logons against [a-zA-Z]{3,}`".

A breakdown of how the regex is processed is below:

- `^` means the regex pattern must start at the beginning of the string
- `[0-9]` means the pattern that is being looked for is a number between 0 and 9
- `+` the plus sign means there must be one or more of the pattern before it which in this case is a number between 0 and 9
- "failed logons against" is a literal string match and would be ignored
- `[a-zA-Z]` matches any English letters that are lowercase or uppercase
- `{3,}` means the pattern preceding must have three or more characters

Windows Events

Microsoft events are stored in proprietary binary format

- Requires Windows Event Viewer or special agent to read

New format is still binary but is XML based

- Supports up to 32,766 bytes (syslog UDP is 1,024)
- Allows for custom parameters in message section

Events are broken up by:

- Channels – A group of logs such as Security or System
- Event IDs – Unique IDs to filter on

Windows Events

Windows Events are stored in channels which are simply groups of logs. The three most common are Application, Security, and System, but there are many more. Within these channels, events are given unique IDs to search and filter on. Event IDs are not unique across channels.

Prior to Vista, Windows events were stored in a proprietary format. This format effectively made it so that Windows Event Viewer was the only method to view the logs. This doesn't mean that vendors and the open source community were not able to develop methods to read this binary format. It just means native support is limited to Windows Event Viewer and a backward compatible PowerShell cmdlet called Get-EventLog. However, pulling event data or user data at a field level is not possible with the old EVT format. For this, EVTX is needed, as well as a tool that can read it. The PowerShell cmdlet for EVTX format logs is Get-WinEvent.

Windows events allow for logs up to 32 KB. However, the actual allowed size limit is 32,766 bytes.

XML Logs

XML is structured

- Allows for automatic field extraction
- Requires agent support

Traditional agent:

100+ parsed fields

Modern agent:

1,000s of extracted fields

```
- <Event xmlns="http://schemas.microsoft.com/win/2004/08/events/event">
- <System>
  <Provider Name="Microsoft-Windows-GroupPolicy" Guid="{AEA1B4FA-97D1-
  <EventID>1500</EventID>
  <Version>0</Version>
  <Level>4</Level>
  <Task>0</Task>
  <Opcode>1</Opcode>
  <Keywords>0x8000000000000000</Keywords>
  <TimeCreated SystemTime="2017-03-18T22:47:22.082837200Z" />
  <EventRecordID>19640</EventRecordID>
  <Correlation ActivityID="{4779A7F8-8DCC-400D-8800-CF7A433EF9EF}" />
  <Execution ProcessID="1240" ThreadID="10680" />
  <Channel>System</Channel>
  <Computer>CIT01LPT.test.int</Computer>
  <Security UserID="S-1-5-18" />
</System>
- <EventData>
  <Data Name="SupportInfo1">1</Data>
  <Data Name="SupportInfo2">4202</Data>
```

XML Logs

Starting with Windows Vista, Windows events are XML based. This means fields are stored in parameters rather than having one large message with a bunch of data. This makes event fields specific and readable. Take, for example, this log below:

Process Create:

UtcTime: 2016-09-10 21:49:31.566

ProcessGuid: {a6b770da-7feb-57d4-0000-0010278c5b34}

ProcessId: 6620

Image: C:\Windows\System32\mmc.exe

CommandLine: "C:\Windows\system32\mmc.exe"

"C:\Windows\system32\eventvwr.msc"

CurrentDirectory: C:\Windows\system32\

User: LIGHTFORGE\jhenderson

LogonGuid: {a6b770da-65fb-57c8-0000-00204d200c00}

LogonId: 0xC204D

TerminalSessionId: 1

IntegrityLevel: High

Hashes: SHA1=6696C2DEF35A1BA4C16C88327DA84F1F3B01E4F9

ParentProcessGuid: {a6b770da-7feb-57d4-0000-00109f835b34}

ParentProcessId: 12228

ParentImage: C:\Windows\System32\eventvwr.exe

ParentCommandLine: "C:\Windows\system32\eventvwr.exe"

If this log was sent via syslog or some raw transport mechanism, it would send this as one large message. However, under the hood, the log actually looks like this:

```
<Event xmlns="http://schemas.microsoft.com/win/2004/08/events/event">
  <System>
    <Provider Name="Microsoft-Windows-Sysmon" Guid="{5770385F-C22A-43E0-BF4C-
06F5698FFBD9}" />
    <EventID>1</EventID>
    <Version>5</Version>
    <Level>4</Level>
    <Task>1</Task>
    <Opcode>0</Opcode>
    <Keywords>0x8000000000000000</Keywords>
    <TimeCreated SystemTime="2016-09-10T21:49:31.575564700Z" />
    <EventRecordID>115</EventRecordID>
    <Correlation />
    <Execution ProcessID="9136" ThreadID="9280" />
    <Channel>Microsoft-Windows-Sysmon/Operational</Channel>
    <Computer>Lightforge</Computer>
    <Security UserID="S-1-5-18" />
  </System>
  <EventData>
    <Data Name="UtcTime">2016-09-10 21:49:31.566</Data>
    <Data Name="ProcessGuid">{A6B770DA-7FEB-57D4-0000-0010278C5B34}</Data>
    <Data Name="ProcessId">6620</Data>
    <Data Name="Image">C:\Windows\System32\mmc.exe</Data>
    <Data Name="CommandLine">"C:\Windows\system32\mmc.exe"
"C:\Windows\system32\eventvwr.msc"</Data>
    <Data Name="CurrentDirectory">C:\Windows\system32</Data>
    <Data Name="User">LIGHTFORGE\jhenderson</Data>
    <Data Name="LogonGuid">{A6B770DA-65FB-57C8-0000-00204D200C00}</Data>
    <Data Name="LogonId">0xc204d</Data>
    <Data Name="TerminalSessionId">1</Data>
    <Data Name="IntegrityLevel">High</Data>
    <Data Name="Hashes">SHA1=6696C2DEF35A1BA4C16C88327DA84F1F3B01E4F9</Data>
    <Data Name="ParentProcessGuid">{A6B770DA-7FEB-57D4-0000-
00109F835B34}</Data>
    <Data Name="ParentProcessId">12228</Data>
    <Data Name="ParentImage">C:\Windows\System32\eventvwr.exe</Data>
    <Data Name="ParentCommandLine">"C:\Windows\system32\eventvwr.exe"</Data>
  </EventData>
</Event>
```

The fact that it is XML means that the parameters are defined and extractable with the right agent. Unfortunately, many native SIEM agents will only pull fields that they want to collect. This means you may only have 50-100 Windows fields when, in fact, there can be more than a thousand in a large environment.

Log Agents

Log agents provide additional functionality

- Auto-parsing
- Log rotation
- Log buffering
- Prioritization
- Filtering



However, all agents are not created equally

- Performance and functionality vary greatly

Log Agents

The capabilities modern log agents have is staggering. A modern agent has many of the capabilities that a log aggregator has and, in some instances, can even act as its own message broker. For example, below is a list of features of modern agents:

Auto-parsing – Automatic parsing of CEF, CSV, XML, KV, LEEF, JSON, GELF, W3C, Syslog, etc.

Data diode support – One-way communication of log traffic

Pre-parsing – Filtering of logs at the endpoint system.

Event rate controls – Can limit the number of logs sent to control bursts

Log rotation – Scheduled purges/moving to archive of local logs

Log buffering – In the event that logs are not being accepted when sent the agent can locally buffer in memory or disk up to a certain size and then resend when communication is reestablished

Server mode – Some agents can also act as a collection server. This can be used to create log relays. For example, if you had a low bandwidth site you could send all logs to an agent in server mode and have that agent send logs on in a highly compressed fashion and you could also apply pre-filtering prior to sending.

Multiple destinations – Logs can be delivered to multiple destinations of which can support various protocols (TCP, UDP, etc.)

Encryption – Logs can be sent over encrypted channels such as TLS

Log integrity – Logs can be sent with hashes or checksums to verify the integrity of the log in transit

Priority routing – Agent can treat certain logs or events as high priority. These receive special treatment such as being routed first. This also can be used in cases where if many logs are getting generated and logs are starting to drop priority logs are not among those to be dropped.

File monitoring – File integrity monitoring of files or directories

Registry monitoring – Registry integrity monitoring of registry keys

NetFlow ingestion – Support for accepting NetFlow and turning it into a log or event

Alerting – Alerts can be triggered based on conditions such as seeing certain patterns or a based on more advanced conditions such as seeing 500 failed logon events.

Remote administration – Some agents require configuration files be pushed through asset management software. Others provide their own central management for remote administration.

Message conversion – Converts log message from one type to another. For example, original message could be syslog and then converted into JSON

Internationalization – Some agents can automatically detect character sets as well as convert to others

Cloud API Integration – Pulls logs from mainstream cloud providers such as Amazon and Google through API calls

Windows Event Forward collection – Some log agents have the capability to act as a Windows Event Collector server

Performance of agents varies greatly and should be tested. The features used can also affect performance. For example, enabling encryption requires CPU for processing on both the agent as well as whatever is receiving the log.

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

Syslog Agents and Windows

Windows events may not fit within the constraints of syslog

- Large events may get truncated
- Or will be “chopped” up

Syslog-based agents may separate logs into smaller pieces

- Newline character used to break up sections
- Last piece without newline distinguishes end of log

Putting the pieces back together adds overhead

- As does monitoring for newlines

Syslog Agents and Windows

Depending on the agent or transport mechanism, Windows logs may be truncated or dropped. For example, the log above is 1,652 characters which means it is 1,652 bytes, not including any header information. If sent over syslog using UDP, it may only store two-thirds the message.

But wait, you might have an agent that sends logs via syslog that exceed these character limitations. How does that work? In most cases, this is done by sending logs and breaking them up into multiple syslog payloads separated by a line break. This means the receiving end not only has to watch for and be aware of broken up logs, but it must also put them back together. This has a significant impact on your EPS rate.

To avoid this, modern agents that are not syslog based, like NXLog1 or fluentd2, can be used.

[1] <https://nxlog.co/>

[2] <https://www.fluentd.org/>

Windows Event Forwarding

Available for Windows XP/2003, built-in to Vista/2008+

- Centrally managed via GPO
- Patched with Windows Updates

Features

- Allows pushing or pulling logs to/from central event collector (uses Windows Remote Management)
- Encryption and Compression
- Basic filtering

Windows Event Forwarding

If you do not have approval to deploy an agent, one thing you may want to try is Windows Event Forwarding. Basically, Windows has its own built-in agent for handling the forwarding of Windows event logs. The best thing is that it is free as it is part of Windows. The next best thing is that operational costs are minimal as it is centrally managed with group policy and is automatically patched as part of Windows Updates.

While it is not feature rich compared to third-party agents, it does include filtering, encryption, compression, event throttling, and the ability to either push logs to a central collector or to have a collector pull the logs.

One of the most complete guides on setting up Windows Event Forwarding can be found using the NSA published PDF called *Spotting the Adversary with Windows Event Log Monitoring*¹. This guide provides step-by-step instructions for setting up Windows Event Forwarding, as well as how to configure it securely. It also demonstrates filtering events. The built-in event filtering is done with XML and is not as easy as filtering with a standard agent.

For Windows Event Forwarding to work, Windows Remote Management and .NET Framework 2.0 SP1 or later are needed. This is installed on Windows 7 and later by default. It also requires allowing TCP port 5985 on host-based firewalls and network firewalls.

Note that unlike a third-party agent, Windows Event Forwarding uses Active Directory authentication to verify that events should be accepted. This layer of authentication can open up systems to man-in-the-middle attacks similar to agentless log collection. However, Microsoft has implemented multiple layers of security around this and has protections in place to prevent this. For example, Kerberos authentication or mutual certificate-based authentication can be used. Also, a setting called channel binding token can be enabled to specifically break authentication requests if man-in-the-middle activity is discovered.

[1] <https://www.iad.gov/iad/customcf/openAttachment.cfm?FilePath=/iad/library/ia-guidance/security-configuration/applications/assets/public/upload/Spotting-the-Adversary-with-Windows-Event-Log-Monitoring.pdf&WpKes=aF6woL7fQp3dJi6c88a6725NB6GHdadkG7evpM>

Windows Event Collector

Push/Pull is set up via subscriptions in group policy

- Events stored on server as standard Windows logs

Searching and viewing logs can be done via Windows Event Viewer and PowerShell

- Not ideal for searching and generating alerts

Can be used with agentless collection or replaced by agents

- Allows for custom architectures to be developed

Windows Event Collector

This server acts as a storage node and smaller organizations may run this as their SIEM. However, logs are stored in the native EVTX format within %SystemRoot%\System32\Winevt\Logs\ForwardedEvents.evtx and are accessible through either Windows Event Viewer or PowerShell. Searching and reporting is severely limited and does not replace the functionality of a full SIEM. Events are collected without alteration or enrichment, and events cannot be sub-parsed. As one might expect, the collector can only accept Windows events.

It is possible to combine forwarded events with Windows Task Scheduler to trigger alerts based on events coming in.

More importantly, it is possible to use a Windows Event Collector with third party agents or even agentless collection. This means that you could roll out Windows event collection using the native Windows Event Forwarding and then pull the logs from the Windows Event Collector into a SIEM solution. This allows better parsing, searching ability, and much more. This may be a compromise for organizations that do not wish to deploy another agent.

This is still not an agent replacement equivalent. The functionality differences are too vast. If you are planning on collecting logs from the central event collector but do not actually need the central event collector, you can replace it directly with a third-party agent.

Third-Party Agents

Many open source and commercial agents available

- Significantly more feature rich than built-in agents

Key areas to focus

- Transport methods (syslog, UDP, TCP, binary, etc.)
- Filtering capabilities
- Special features
- Support

Third-Party Agents

Third-Party Agents offer significantly more capabilities than native agents or syslog. They have more support for transporting logs such as supporting UDP, TCP, binary compression, TLS encryption, mutual authentication, and have page after page of additional value-added feature sets.

The primary difference between open source and commercial agents ends up being support. Many companies shy away from open source agents since it is important to have support available if problems arise.

Open Source Log Agent Capabilities

- Open Source
- Multi-platform
- Capable of 500K+ EPS
- Buffering (disk and/or memory)
- Prioritization
- Log rotation
- Log format support (Syslog, CEF, JSON, XML, Windows, CSV, and more)
- Log format conversions
- Encryption
- Compression
- Internationalization of character sets
- Advanced filtering
- Advanced parsing
- And much more...

Open Source Log Agent Capabilities

This slide demonstrates some of the capabilities of open source agents such as NXLog¹ or fluentd². Clearly, the feature set of open source log agents is enormous. This slide effectively shows that open source agents often are more advanced and feature-rich than native commercial SIEM agents. On top of that most open source agents have commercial support or commercial license offerings with even more features.

[1] <https://nxlog.co/>

[2] <https://www.fluentd.org/>

Agentless Log Collection

Agentless involves a central server to collect logs

Server authenticates to systems over WMI or SSH

- Logs are collected in batches
- Requires admin privileges or additional rights
- Windows 2008+ can use the Event Log Readers group

Main benefit is no additional software

- Do not have to maintain and upgrade agent
- Quicker to deploy and manage

Agentless Log Collection

In order to avoid having to install an agent on systems, some organizations prefer to setup a server to perform agentless log collection. This works by having a server remotely log in to systems often over WMI or SSH and pulling back logs. This works well as long as the server has proper credentials for the remote systems and network firewalls, host-based firewalls, and endpoint security suites are configured to allow this activity.

Because no software needs to be installed on the system, logs that are being collected for agentless deployment allow for a quicker setup and less maintenance. The main advantage is not having to deploy, update, and maintain a log agent.

Keep in mind that agentless servers cannot handle an infinite number of systems. Due to having to authenticate and pull logs, this method may require multiple agentless collectors. This is true even for medium-sized organizations.

Also, agentless collection can introduce a security risk due to constantly logging in over the network. An attacker can attempt to capture these credentials or the security token that is on the system. To minimize this risk, always use the minimum necessary permissions and rights for the service account(s) collecting logs. Always ensure, if you are opening services or ports, that you limit their access to only what is necessary.

Script Collection

Sometimes scripts are the only method to obtain logs

Especially true for:

- Cloud systems and software
- Third party applications

You are responsible for cloud and third-party applications

- Often have APIs for management

Scripts can use APIs and either ship off logs directly or save them to a file for an agent to pick up

Script Collection

Another method of log collection that is important to consider is scripts. There are some instances in which scripts are your only option for collecting logs. More importantly, scripts allow you to create custom logs that add value to your environment.

A great example of this is the continued boom in cloud-based applications and systems. If you are using them, you still have a level of responsibility for the data you put in them. Yet, many do not have logs or if they do you, do not have direct access. A large portion of these cloud providers have APIs that can pull out logs and important events. Scripts can be written to pull this data on an ongoing basis.

Script Use

Scripts are helpful for collecting custom logs

For example, a script could run every night collecting baseline information such as:

- ASEPs (Auto Start Extensibility Points)
- Inventory information

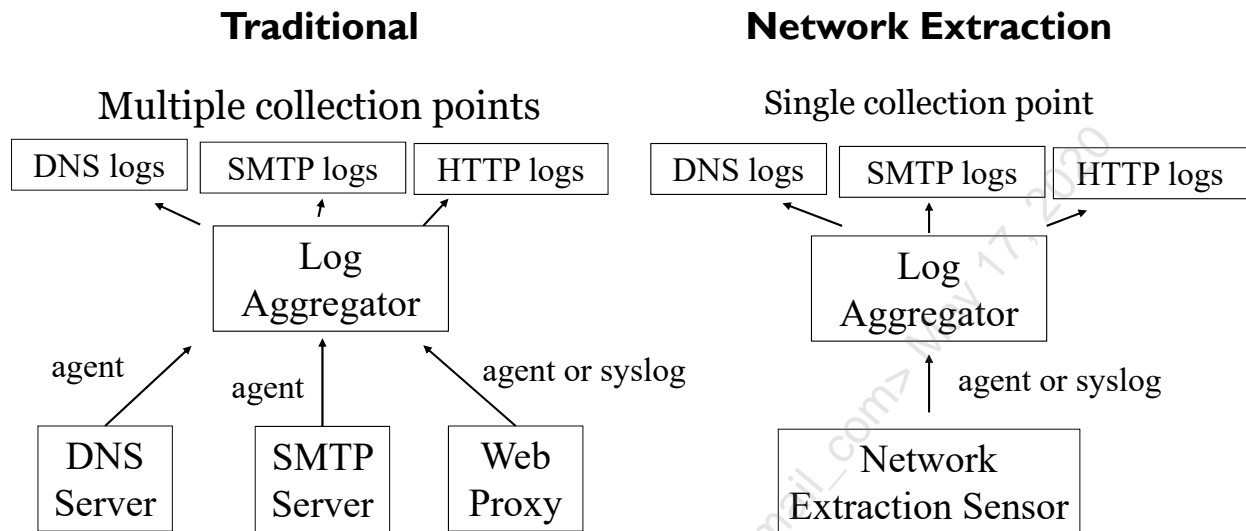
This can then be shipped off to a log aggregator or storage

- Now this data is available for searching and data mining

Script Use

Scripts are incredibly useful for pulling in additional information not found in regular logs. This can include pulling in baseline information, asset inventory, running processes, hashes of files, and more. Running a script to generate this information and then shipping it off to a SIEM provides incredible power for searching and data mining.

Traditional vs. Network Extraction



Traditional vs. Network Extraction

The normal method for collecting service logs is to install agents on all of your servers or enable syslog if you are using an appliance. This makes logical sense as you are going straight to the source of the service. For example, you might install a log agent on a DNS server and then point it to the file location and have it ship off logs as they are generated.

The flip side of this is that you can use a network monitoring system to look at packets and create logs as they are seen on the network. This approach is interesting as instead of having multiple agents on various application servers, a single network monitoring host can often see and generate all the logs from one spot. The main drawback is that you first must give this system network visibility, such as through the use of a tap or port mirror.

SIEM and Log Collection Summary

Multiple ways to send or receive logs

- Agentless
- Native SIEM agent
- Third-party agents
- System built-in agents (syslog or Windows event forwarding)
- Scripts

A strong design likely involves a combination of above

SIEM and Log Collection Summary

A strong log collection implementation most likely involves using more than one method of log collection. This will be affected by corporate policy as well as the limitations of the devices or systems being maintained.

Course Roadmap

- Day 1: Defensible Security Architecture
- Day 2: Network Security Architecture
- Day 3: Network-Centric Application Security Architecture
- Day 4: Data-Centric Application Security Architecture
- **Day 5: Zero Trust Architecture**
- Day 6: Capstone: Design, Detect, Defend

CURRENT STATE ASSESSMENT, SOCS, AND SECURITY ARCHITECTURE

1. Zero Trust Architecture
2. Credential Rotation
3. Securing Traffic
4. EXERCISE: Network Isolation and Mutual Authentication
5. Host-Based Firewalls
6. Network Access Control (NAC)
7. Segmentation Gateways
8. Security Event Information Management (SIEM)
9. EXERCISE: SIEM Analysis and Tactical Detection
10. Log Collection
11. Audit Policies
12. Host Hardening
13. Patching
14. Tripwires and Red Herring Defenses
15. EXERCISE: Advance Defense Strategies

Course Roadmap

The next section covers host hardening.

Audit Policies

Log collection is dependent on proper audit policies

- Logs must be generated before being collected
- Default logging settings are insufficient

Windows utilizes audit policies to enable logging

Linux requires configuration file changes

- Both support custom logs such as with PowerShell
- Or third-party programs to add additional logging

Audit Policies

Zero trust architecture relies on logging and inspecting all activity. The key to that is logging and then reviewing those logs. A SIEM is only as good as the data that goes into it. That means junk in junk out. It also means that if key logs do not exist analysis will fail.

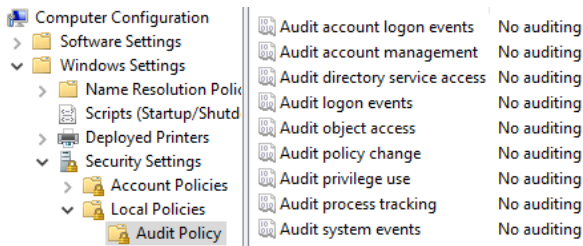
This means that endpoints need to be tuned so that key logs are generated. On Windows, this involves tuning the audit policies. On Linux or Mac, it means making configuration changes. In some cases, this also means installing additional log programs.

Windows Audit Policies

Audit Policy

Basic log settings

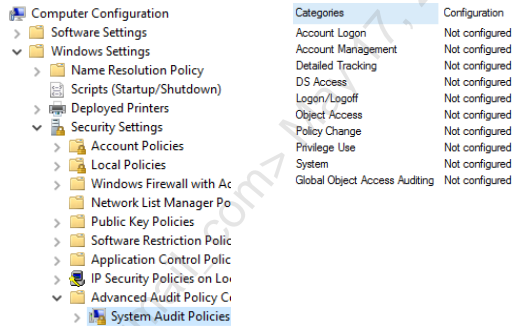
- Available for Windows 2000+



Advanced Audit Policy

Provides granular control of logs

- Requires Server 2008 R2 or Windows 7 and later



Windows Audit Policies

The Windows Audit Policy is available in two formats: the basic audit policy and the advanced audit policy. The basic audit policy was the only option until Windows 7 and Server 2008 R2. These operating systems introduced the Advanced Audit Policy. This allows for a more granular control over what logs are recorded.

Advanced Audit Policy

Default settings prefer Audit Policy rather than Advanced Audit Policy

- If using Advanced Audit Policy, remember to change this Computer Configuration -> Windows Settings -> Security Settings -> Local Policies -> Security Options
- Enable “Audit: Force audit policy subcategory settings



Audit: Force audit policy subcategory settings (Windows Vista or later) to override audit policy category settings

Enabled

Disabled

Advanced Audit Policy

When using the advanced audit policy, it is important to know that the default behavior is for basic audit policy settings to override advanced audit policy settings. Fortunately, Windows warns you of this when you visit the advanced audit policy. It also shows how to change this behavior by setting Audit: Force audit policy subcategory settings to Enabled.

Keep in mind that using advanced audit will also clear the simple policy settings¹.

[1] [https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2008-R2-and-2008/ff182311\(v=ws.10\)#BKMK_3](https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2008-R2-and-2008/ff182311(v=ws.10)#BKMK_3)

auditpol.exe¹

Non-domain joined systems can be configured with auditpol.exe¹

- Can list and set policies

List policy settings:

auditpol /get /category:*

Set policy settings:

auditpol /set /subcategory:"file system" /success:enable /failure:enable

```
auditpol /get /category:*
```

Category/Subcategory	Setting
System	
Security System Extension	Success and Failure
System Integrity	Success and Failure
IPsec Driver	No Auditing
Other System Events	Success and Failure
Security State Change	No Auditing
Logon/Logoff	
Logon	Success and Failure
Logoff	Success and Failure
Account Lockout	Success and Failure

auditpol.exe¹

Not all systems are typically joined to a domain. Some are intended not to be joined for security purposes (such as DMZ systems). In order to effectively maintain log settings, audit policies need to be pushed out. Since group policy is not available, an alternative method such as using auditpol.exe¹ is needed. This command line utility can retrieve and set audit policy settings. Because of this, it is fairly easy to automate and push out a standardized audit policy for non-domain joined systems.

Another command that may be beneficial is the one below. It lists out all the available policy categories.

auditpol /list /subcategory:*

[1] <https://www.ultimatewindowssecurity.com/wiki/page.aspx?spid=SecuritySettings>

Account Management

Used to track changes to groups, users, and computers

- Needed to monitor key groups for modifications
- Such as new members added to Domain Admins

Powerful when combined with change control system

Subcategory	Audit Events
Audit Application Group Management	Not Configured
Audit Computer Account Management	Not Configured
Audit Distribution Group Management	Not Configured
Audit Other Account Management Events	Success and Failure
Audit Security Group Management	Success and Failure
Audit User Account Management	Success and Failure

Account Management

These types of events are useful to keep track of unauthorized changes. For example, it is a best practice to monitor additions to any sensitive security group such as Domain Admins and Enterprise Admins. While out of the gate something such as monitoring Domain Admin additions are high value, more can be achieved if logs from a change control system can be used to verify security group changes.

For example, if employees are added to a group called Sensitive Data access, hopefully, it required someone to authorize the addition. If the authorization record can be used, it can verify that all modifications came with prior consent. Then, should an adversary add an account to this group, it would be red-flagged as no prior authorization was given.

Audit Application Group Management – Monitors changes to application groups. Application groups are used to tie roles using Windows Authorization Manager.

Audit Computer Account Management – Monitors changes to computer accounts.

Audit Distribution Group Management – Monitors changes to Active Directory distribution groups. While this can record details about groups being modified such as Domain Admins it is not needed. The Audit Security Group Management records more information that is helpful and specific.

Audit Other Account Management Events – Monitors special changes such as a password hash of a user account being accessed or changes to the password policy or lockout policy.

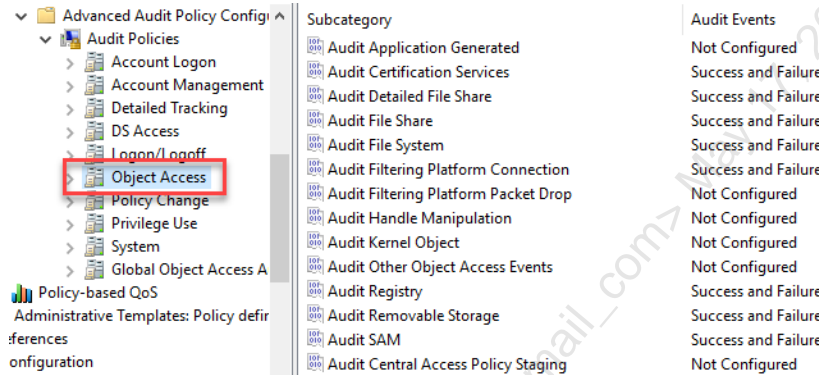
Audit Security Group Management – Monitors changes to standard security groups.

Audit User Account Management – Monitors changes to user accounts.

Object Access

Object access is one of the most misunderstood settings

- Such as audit file system... it does not log all file access



Object Access







Object access controls logging for quite a few things. If you wish to audit files, registry keys, or network shares, you must enable the auditing capability first. For example, after turning on Audit File System, it grants Windows the ability to audit things like files or folders being accessed but only if an ACL is placed on them telling what to audit. In the past, there was a wide misconception that enabling this would generate an event for every file accessed.

This audit policy also controls things such as Windows Firewall logging to a log channel and file access on removable drives. In Active Directory, many things are considered objects (users, groups, files, folders, registry keys, etc.). This policy even controls certificate-related events.

Detailed Tracking

Can generate vast amounts of logs—proceed with caution

- Has built-in monitoring of processes
 - Default behavior does not include command line logging
- Windows 10/Server 2016 also includes plug and play monitoring

Subcategory	Audit Events
 Audit DPAPI Activity	Not Configured
 Audit PNP Activity	Success
 Audit Process Creation	Success and Failure
 Audit Process Termination	Not Configured
 Audit RPC Events	Not Configured
 Audit Token Right Adjusted	Not Configured

Detailed Tracking

Be careful when enabling logs under detailed tracking as they may have performance implications. However, there are a few critical events analysts should regularly monitor. One is process creations which involves logging all new processes being launched. Windows can natively log this if Audit Process Creation is enabled. These types of logs can also be collected from Application Whitelisting suites. It is important to note that the default behavior of Audit Process Creation is to generate an event on each new process being launched but that it does not include any command line parameters involved with the process starting.

To enable auditing process creation and include command line parameters, enable “Audit Process Creation” but also enable the policy “Include command line in process creation events” located at Computer Configuration -> Policies -> Administrative Templates -> System -> Audit Process Creation.

Another thing worth tracking is new devices being plugged in such as plug and play devices. Starting with Windows 10 and Server 2016, Microsoft allows logging of plug and play devices. While this, unfortunately, is not available for older operating systems, it is a step in the right direction.

Sysmon

Free download from Windows Sysinternals

- Written by Mark Russinovich and Thomas Garnier
- Runs as a Windows system service and device driver
- Monitors:
 - Processes
 - Network connections
 - Driver and DLL loading
 - Raw disk access
 - Modifications of file creation times
 - Process access

Provides process hashes and parent processes for analysis

Sysmon

Sysmon was designed to generate logs of interesting activity commonly associated with malicious or anomalous activity. By themselves, the logs are useless; however, when looked at by an analyst, they can be extremely powerful. For example, Sysmon can monitor all processes being launched along with the parent process that spawned the new process and can provide a hash of the new process. On top of this, the log also includes any command line parameters along with other useful information. Collecting this data centrally and evaluating it lets you know exactly what is taking place in your organization.

Similar to NetFlow, Sysmon can optionally log all network connections being made. However, unlike NetFlow, Sysmon records the process that either made the network request or is receiving the network request. This level of detail is great for troubleshooting, incident response, and establishing firewall rules for a default deny policy.

Under the hood, Sysmon utilizes built-in Windows API calls and ETW tracing to generate logs. This allows for minimal performance overhead. Keep in mind it will generate a lot of logs. This author likes to see Sysmon installed on all systems so that the logs are there if needed. However, it may not make sense to try and collect all of these logs centrally unless major filtering is done. Sysmon allows fine-grained filtering on what to log. For example, you can create a configuration rule to log all network connections unless they come from the process iexplore.exe.

Sysmon is built for both desktops and servers. It can be deployed to Windows 7 and newer as well as Server 2012 and newer operating systems. Mark Russinovich wrote a presentation¹ for RSA Conference 2016 which does a great job explaining Sysmon, its use cases, and how to configure it.

[1] https://www.rsaconference.com/writable/presentations/file_upload/hta-w05-tracking_hackers_on_your_network_with_sysinternals_sysmon.pdf

Sysmon Example

Event 1, Sysmon

General	Details		
Process Create: UtcTime: 2017-01-02 16:45:55.172 ProcessGuid: {6a32f033-83c3-586a-0000-0010aa59c4ad} ProcessId: 3048 Image: C:\Windows\System32\timeout.exe CommandLine: timeout 1 CurrentDirectory: C:\Program Files\Intel\SUR\WILLAMETTE\ User: NT AUTHORITY\SYSTEM LogonGuid: {6a32f033-05f4-585e-0000-0020e7030000} LogonId: 0x3E7 TerminalSessionId: 0 IntegrityLevel: System Hashes: SHA1=B55BE2059C5FEE93C803CC853360766707BE8AF4 ParentProcessGuid: {6a32f033-83c0-586a-0000-00105a17c4ad} ParentProcessId: 19216 ParentImage: C:\Windows\System32\cmd.exe ParentCommandLine: C:\Windows\system32\cmd.exe /c ""C:\Program Files\Intel\SUR\WILLAMETTE\svc_install.bat" S > log_start.txt 2> &1"			
Log Name:	Microsoft-Windows-Sysmon/Operational		
Source:	Sysmon	Logged:	1/2/2017 10:45:55 AM
Event ID:	1	Task Category:	Process Create (rule: ProcessCreate)
Level:	Information	Keywords:	
User:	SYSTEM	Computer:	CIT01LPT.test.int
OpCode:	Info		
More Information:	Event Log Online Help		

Sysmon Example

In this slide, the parent process of C:\Windows\System32\cmd.exe is running the batch script at C:\Program Files\Intel\SUR\WILLAMETTE\svc_install.bat. This is being executed by the NT AUTHORITY\SYSTEM account and a hash of the executable being spawned (C:\Windows\System32\timeout.exe) is provided.

Collecting this level of detail allows for studying and understanding what is happening in your environment. For example, if you received this log, would you know what WILLAMETTE is? It sounds odd; possibly it is malicious, but it is in the C:\Program Files\Intel folder so maybe it is not. Investigating this would reveal it is an Intel service for providing power savings and is installed as part of the Intel Driver Update Utility.

Sysmon Configuration

Granular logging available

- Uses XML config
- Can include or exclude on:
 - Path
 - Process/Image
 - Digital Signature
 - Integrity Level

```
<Sysmon schemaversion="3.20">
  <!-- Capture all hashes -->
  <HashAlgorithms>sha1,md5</HashAlgorithms>
  <CheckRevocation></CheckRevocation>
  <EventFiltering>
    <!-- Log all drivers except if the signature -->
    <!-- contains Microsoft or Windows -->
    <DriverLoad onmatch="exclude">
      <Signature condition="contains">microsoft</Signature>
      <Signature condition="contains">windows</Signature>
    </DriverLoad>
    <!-- Log all process creations unless they have a -->
    <!-- integrity level of medium -->
    <ProcessCreate onmatch="exclude">
      <IntegrityLevel>Medium</IntegrityLevel>
    </ProcessCreate>
    <!-- Do not log process termination -->
    <ProcessTerminate onmatch="include" />
    <!-- Log network connection if the process isn't InternetExplorer -->
    <NetworkConnect onmatch="exclude">
      <Image condition="end with">iexplore.exe</Image>
    </NetworkConnect>
  </EventFiltering>
</Sysmon>
```

Sysmon Configuration

Configuring Sysmon to log or not log something can be done with basic command line switches or by using the advanced configuration. The advanced configuration requires the use of XML configuration files. When deploying Sysmon, it is usually recommended to use the advanced configuration files as the level of granularity is often used to include or exclude certain things.

The following commands can be used to find more information on building a Sysmon configuration file:

```
sysmon.exe --help
sysmon.exe -? config
```

The picture in this slide also shows an example of a functional XML configuration file used for Sysmon version 5. The main thing to pay attention to is the use of `onmatch="include"` and `onmatch="exclude"`. In the case of `onmatch="exclude"`, all items are logged except the ones specifically excluded. So, a statement reading `<ProcessCreate onmatch="exclude"></ProcessCreate>` would log all process creations as no exclusions are specified. In contrast, `onmatch="include"` only includes items specified. The configuration of `<ProcessCreate onmatch="include"></ProcessCreate>` would log nothing because nothing is specified to be included for logging.

If attempting to build a new Sysmon configuration, consider using a community provided configuration for a starting point.¹

[1] <https://github.com/SwiftOnSecurity/sysmon-config>

Linux Logs

Syslog is the primary method of logging for Linux/Unix

- Default behavior is to listen using a local socket

Multiple syslog daemons exist

- Default log location is `/var/log/`
- File names such as `auth` or `kern` specify log category
- Log level specified by severity in daemon configuration
- Daemon can be configured to accept and to send logs

Linux Logs

In comparison to Windows logging, Linux logging is substantially different. This adds to the challenge of system administration as differing platforms require different administration. In the case of Linux, syslog is the default logging service, and even then, there are multiple flavors of syslog in use today.

The default behavior of most systems is to log to `/var/log` and file names such as `auth.log` are specified to categorize the log. This is also referred to as the syslog facility in many cases. By default, most syslog services, also called daemons, only log locally. However, they can be modified to ship logs over the network or event accept logs from other network systems.

Syslog Configuration

Linux and Mac come with built-in syslog agents

- Rsyslog and Syslog-NG are most robust and full-featured
 - Common to systems built for functionality (like Ubuntu)
- Syslogd is also common and provides basic functionality
 - Primarily used for systems built around security or designed to limit surface footprint (like CentOS, RHEL, or FreeBSD)
 - Depending on business requirements may need to switch to third-party agent or Rsyslog or Syslog-NG
- Mac uses Apple System Logger or Unified logging¹

Syslog Configuration

Using built-in logging on Linux is most likely to be from either Rsyslog, Syslog-NG, or Syslogd. Many newer operating systems utilize Rsyslog such as CentOS 7, Red Hat Enterprise Linux 7, Kali, and Ubuntu 16.04. Syslog-NG is also still common. For instance, Ubuntu 14.04 systems default to using Syslog-NG.

Occasionally, some systems will use syslogd. This is more common for systems that focus on strict security or smaller service footprints.

[1] <https://developer.apple.com/documentation/os/logging>

Syslog Config Example (Ubuntu 16.04 System)

```

auth,authpriv.*           /var/log/auth.log
authpriv.=warning        @10.5.55.10
*.*;auth,authpriv.none  -/var/log/syslog
cron.warning             /var/log/cron.log
*.!info                 /var/log/verbose.log
kern.*                  -/var/log/kern.log
#lpr.*                  -/var/log/lpr.log
mail.*                   -/var/log/mail.log

```

Syslog Config Example (Ubuntu 16.04 System)

This example is from the default Ubuntu 16.04 rsyslog configuration file. The traditional syslog format is compatible with the newer rsyslog format. This example follows the traditional syslog format for specifying what gets logged and where. The left column specifies the facility and severity levels to log and the right column specifies the destination.

```
auth,authpriv.*           /var/log/auth.log
```

This takes any logs with a facility of auth or authpriv and sends them to /var/log/auth.log. The asterisk referenced means all severities.

```
authpriv.=warning        @10.5.55.10
```

This takes any logs with a facility of authpriv and a severity of warning and sends them over UDP port 514 to 10.5.55.10. The equals sign specifies an exact match on severity so only a severity of warning will be included.

```
cron.info                /var/log/cron.log
```

This takes any logs with the cron facility that have a severity of info or lower and writes to /var/log/cron.log.

```
*.!info                 /var/log/verbose.log
```

This takes any logs with any facility that have a severity of info or lower (which would include info and emergency severities) and writes it to /var/log/verbose.log.

This takes any logs with a facility of cron and a severity of warning or higher (error, critical, alert, or emergency) and sends them to /var/log/cron.log.

```
kern.*                  -/var/log/kern.log
```


This takes any logs with a facility of kern and with any severity level and sends them to /var/log/kern.log. The dash before /var/log/kern.log means that syslog does not have to flush the log to disk after each log. This is often used for log files that receive a lot of logs. Having to confirm the write to disk could cause unacceptable performance issues. However, should a system crash with this setting enabled, it is possible that logs may not have been committed to the log file and thus, can be lost.

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

auditd¹

Provides a customizable Linux Audit system

Monitors:

- File Access
- System calls
- Program execution
- File changes
- Security events (such as failed logins)
- Network access

Granular monitoring allows advanced use cases

- Also, adds complexity and performance overhead

auditd¹

The Linux Audit system, also referred to as auditd, is maintained by Red Hat but is supported and available to install on just about every Linux system such as Ubuntu, Suse, CentOS, etc. Similar to Sysmon, this service provides additional logging such as recording user activity and process activity. To be truly effective, auditd must be configured and told what to log or what not to log. Some Linux systems come with auditd installed but usually with logging disabled. This is because the level of logging can be immense and cause performance issues.

With a little care, auditd can be configured to log only under specific use cases. Unfortunately, the syntax of auditd can be difficult to understand and use. However, because of the granularity allowed, administrators can fine-tune exactly what gets logged.

[1] https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/6/html/security_guide/chap-system_auditing

Audit Example

PID is process id of executable, **PPID** is for parent process

UID is user id of user

AUID is the audit user id (tracks actions against logon even if user changes with su or sudo)

```

type=SYSCALL msg=audit(1483384102.643:128): arch=c000003e syscall=59 success=yes
exit=0 a0=1765008 a1=1758108 a2=18e1008 a3=7ffeb4617ec0 items=2 ppid=3165 pid=4
565 auid=4294967295 uid=1000 gid=1000 euid=1000 suid=1000 fsuid=1000 egid=1000 s
gid=1000 fsgid=1000 tty=pts9 ses=4294967295 comm="ifconfig" exe="/sbin/ifconfig"
key=(null)

type=SYSCALL msg=audit(1483384111.147:144): arch=c000003e syscall=59 success=yes
exit=0 a0=5639061d8e78 a1=5639061def28 a2=5639061d86a0 a3=7ffd5c3bd7e0 items=2
ppid=4566 pid=4567 auid=4294967295 uid=0 gid=0 euid=0 suid=0 fsuid=0 egid=0 s
gid=0 fsgid=0 tty=pts9 ses=4294967295 comm="ifconfig" exe="/sbin/ifconfig" key=(nul
l)

```

Audit Example

This slide shows two logs taken from a Linux system. In these examples, a system administrator ran ifconfig and then changed to the root user using sudo su. Then, they ran ifconfig again, cutting the second log in the slideshow. Normally, when a user uses sudo, su logs cannot be tracked back to the end user as all logs now come from root. However, with auditd logging, the auid field identifies who the user was based on initial login.

Here's a breakdown of the events and logs:

1. The user with a user ID of 1000 logs into the system.
2. This user then runs the command "ifconfig".
3. Auditd generates the first log in this slide. It contains auid=4294967295 uid=1000 gid=1000. The auid is a generated ID associating all session activity back to the original user which, in this case, is the user with the user ID of 1000.
4. The user with a user ID of 1000 then changes user accounts by running sudo su. This makes them root. Root has a user ID of 0.
5. This user then runs the command "ifconfig".
6. Auditd generates the second log in this slide. It contains auid=4294967295 uid=0 gid=0. This shows that the action was performed as root (uid=0); yet, the session and subsequent actions are actually from user ID 1000. This can be traced by looking at the matching auid field.

While ifconfig is recorded as being executed, this log does not contain a hash or digital signature related to the binary. In order to add this, the Linux auditing system would need either patching or an upgrade to include this capability. Alternatively, a log agent would need to augment the log during data collection prior to shipping it off.

audit.rules Example

```
# First rule - delete all
-D
# Increase the buffers to survive stress events.
# Make this bigger for busy systems
-b 320

-a exit,always -F arch=b64 -S sethostname -S setdomainname
-k changenamerule
-w /etc/passwd -p wa -k passwdrule
-w /sbin/ifconfig -p x -k ifconfigrule
```

audit.rules Example

Auditd allows for extremely granular monitoring. As a result, it is usually best to plan exactly what needs logging *before* writing the auditing rules. A general rule of thumb is to try and log only what you intend to actually look at. In the case of auditd, logging everything can have adverse effects on system performance.

The audit.rules file is composed of three types of rules: control rules, filesystem rules, and system call rules. Control rules are used to define configuration settings for the audit system.

Control rules

In this slide, the first rule is -D which clears out all rules at the beginning of a configuration file load. Then, it sets the backlog size limit using -b. In this case, the buffer is set to 320 logs. The limit is likely too small for even moderately accessed systems and may need to be increased to 1024 or higher. The higher this is set the more memory is consumed to handle the buffer. To monitor if your buffer is set high enough, run the command “auditctl -s” and look at the field called lost. If it is not set to zero, then some logs have been lost and you may need to increase the buffer size.

System call rules

System call rules monitor system calls from processes or users. In this slide, the first custom rule is an example of a system call rule. It takes place after the -D and -b control rules.

```
-a exit,always -F arch=b64 -S sethostname -S setdomainname -k system-locale
```

-a stands for append rule and is applied when monitoring system calls. If -A was used instead of -a it would append the rule at the top of the rule list instead of the bottom. Immediately following -a or -A should be one of four possible options: task, exit, user, or exclude. In almost all cases, exit will be used.

-F arch=b64 refers to the 32-bit or 64-bit architecture. In this case, the syscall for 64-bit calls is being monitored. Note that in some cases you may need to monitor both 32-bit and 64-bit syscalls on a 64-bit Linux system.

-S refers to the system call to monitor. In the first rule, both the sethostname and setdomainname system calls are set to be monitored. To see a full list of system calls to monitor, run the command `ausyscall --dump`. Though not recommended, alternatively, all system calls can be monitored by using `-S all`.

-k is used to specify the key name. This is an optional field that might be thought of better as an optional rule name. If you specify -k such as `-k testrule1` and a log gets generated from this rule, then `key=testrule1` will be recorded in the log. This is useful for finding which rules are generating each log. It is a good practice to always specify -k with a unique rule description or name.

File monitor rules

The last two rules specified in this slide are examples of file monitor rules. They are as follows:

```
-w /etc/passwd -p wa -k passwdrule  
-w /sbin/ifconfig -p -x -k ifconfigrule
```

-w specifies that the rule is a watch rule. The file or directory following -w specifies what is being watched.

-p specifies the permissions that are logged. The possible options are r (read), w (write), x (execute), and a (attribute change). In the first example, `-p wa` is used to monitor writes and attribute changes to the `/etc/passwd` file. This would log when new users are added to the system. The next example of `-p x` is used to monitor each time `/sbin/ifconfig` is executed. Both examples use -k to give the rules names that are recorded in the logs they generate.

Many of the rules written within `audit.rules` are file monitor rules. This is because Linux treats almost everything as a file. Even hardware is stored as a special file. Another thing that is significant to know is that rule ordering is important. Rules should be written in a way that the most often hits rules are at the top. Failure to do so causes more performance overhead.

For more details on `audit.rules` try running “`man audit.rules`” on a system with audit installed.

Audit Policies Review

Policies and configuration files control log generation

- Windows uses audit policies
- Linux uses configuration files

Log agents and drivers allow the creation of specialty logs

- Sysmon for Windows
 - Process, network, registry, and file monitoring
- Auditd for Linux
 - Extremely granular logging

Audit Policies Review

Logs are used in a zero trust architecture to drive variable trust as well as detect deviations from the norm. Therefore, it is critical that actionable data is available. Default logs are often insufficient, and thus operating systems need tuning to enable key data sources.

Third-party software such as Sysmon or Auditd provides a significant boon. These solutions provide high-speed, highly actionable logs. Consider deploying these solutions.

Course Roadmap

- Day 1: Defensible Security Architecture
- Day 2: Network Security Architecture
- Day 3: Network-Centric Application Security Architecture
- Day 4: Data-Centric Application Security Architecture
- **Day 5: Zero Trust Architecture**
- Day 6: Capstone: Design, Detect, Defend

CURRENT STATE ASSESSMENT, SOCS, AND SECURITY ARCHITECTURE

1. Zero Trust Architecture
2. Credential Rotation
3. Securing Traffic
4. EXERCISE: Network Isolation and Mutual Authentication
5. Host-Based Firewalls
6. Network Access Control (NAC)
7. Segmentation Gateways
8. Security Event Information Management (SIEM)
9. EXERCISE: SIEM Analysis and Tactical Detection
10. Log Collection
11. Audit Policies
12. Host Hardening
13. Patching
14. Tripwires and Red Herring Defenses
15. EXERCISE: Advance Defense Strategies

Course Roadmap

The next section covers host hardening.

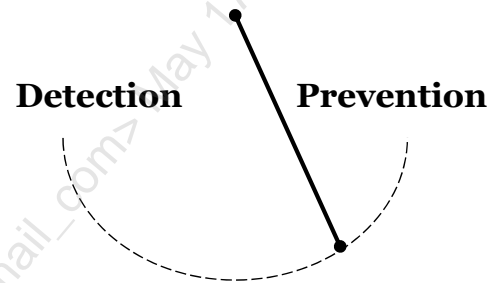
Security Pendulum

Combination of prevention and detection is necessary

- But organizations often lean one way or another
- Current industry focus is heavily on detection
 - Because organizations are prevention-focused

Balance in both is necessary

- Prevention should map to a detection
 - General rule but not always possible
- Detection may enable new prevention capabilities



Security Pendulum

Defensive security falls under two main categories: detection and prevention. These are the two main categories of defenses organizations implement to secure data. For the best security, a mix of both is necessary. Think of this as a recipe. A proper balance of ingredients makes the best tasting food. More often than not, organizations put too much of one ingredient and too little of another. In today's modern world, organizations lean heavily towards preventative technologies.

Because of the heavy prevention focus, cyber security solutions are beginning to lean heavily towards detection. While this is necessary, it is important to remember that a heavy detection-oriented defense with little prevention is not balanced. A balance of detection and prevention is necessary for a good defense.

Host Hardening

Endpoint hardening involves remediating vulnerabilities

- Default system config prioritizes maximum functionality

System changes necessary to limit attack surface

- Remove or disable unnecessary software
- Adjust operating system default settings
- Properly tune system logging
- Install operating system and application patches
- Secure third-party applications
- Install endpoint security software

Host Hardening

Endpoint hardening is thought of as remediating vulnerabilities on systems before going into production. However, hardening is not a one and done process. Hardening needs to take place on systems before they go into production as well as after when a system is in production.

The process of hardening a system involves removing unnecessary software, disabling unused services, enabling key log sources, patching, installing security software solutions, and anything else that decreases the overall risk to an asset.

[1] <https://github.com/PaulSec/awesome-windows-domain-hardening>

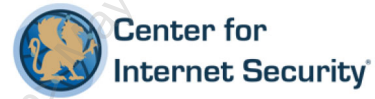
Center for Internet Security (CIS)¹

CIS provides free security benchmarks

- Benchmarks of hardening and best practice guides
- Benchmarks exist for all major operating systems
- As well as major applications like Nginx and IIS

PRO - Benchmarks are heavily detailed

CON - Benchmarks are heavily detailed



Windows 10 benchmark is over **1000 pages**

Center for Internet Security (CIS)¹

Securing systems first requires knowing what to change. Fortunately, there are lots of online hardening guides. The problem is that each guide makes suggestions, but these suggestions do not always come with an explanation of why things matter. For example, many hardening guide recommendations for Windows apply to older operating systems. Modern Windows operating systems already have these settings changed.

What is needed is a hardening guide with explanations. This module will provide such guidance with a focus on key hardening changes. Another source of information that is helpful is the Center for Internet Security¹ or CIS. CIS provides free benchmarks which are hardening guides for major operating systems and applications. CIS benchmarks are highly detailed and explain why a given setting matters and how to set it.

[1] <https://www.cisecurity.org/>

Disable Default Programs / Services

Windows has many defaults that may not be necessary

- **SSDP, LLMNR, Computer Browser, NetBIOS**
- Third-party software installs services like **Dropbox LAN sync**

Commonly results in over 90% of blocked traffic

- And services lead to easy attacks from common tools

Example: LLMNR and NetBIOS are fallback for DNS

- Responder¹ attack tool answers broadcast and multicast requests
- Can lead to credential compromise of screen locked system

Disable Default Programs / Services

Ideally, unnecessary traffic that is constantly being blocked would be cleaned up rather than filtered. As an example, in an enterprise environment, the SSDP service is often not necessary. This service is used for systems to perform network discovery of surrounding devices such as universal plug and play devices. This type of service is helpful for home users, but enterprise environments are typically controlled with asset management tools and group policy. Leaving this service enabled causes multicast traffic that surrounding host-based firewall systems will constantly have to block. Instead of filtering it out, simply disable it. Disabling unnecessary services also reduces the attack surface and any vulnerabilities that may exist should an adversary attack to attack or abuse the service.

Overall, this decreases the amount of traffic on the network, makes it easier to do analysis, and removes the need for filtering large amounts of traffic at the hosts or shipping it off to be filtered at an aggregation unit. There are multiple online sites describing how and why these services should be disabled. One fairly well-documented article is from the University of Iowa.¹ Not every environment will be able to disable all noisy services, but typically most can be disabled.

[1] <https://its.uiowa.edu/support/article/3576>

[2] <https://github.com/Spiderlabs/Responder>

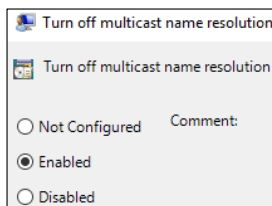
Disabling Software

Group policy is primary method of hardening Windows

- Scripts or asset management also necessary
- Same is true for Linux or Mac

NetBIOS is per NIC and does not have native GPO control

- Use scripts and asset management or GPO



```
$adapters=(gwmi win32_networkadapterconfiguration )
Foreach ($adapter in $adapters){
    Write-Host $adapter
    $adapter.settcpipnetbios(2) # 1 is enable 2 is disable
}
```

Disabling Software

Disabling software on Linux, Mac, and Windows is challenging. For example, Windows predominantly uses group policy or local security policies to harden and adjust settings. However, some settings must be adjusted outside of group policy. NetBIOS as an example is a per-interface setting and must be disabled per-interface. The PowerShell code in the slide is an example of how to loop through each adapter and disable NetBIOS. The picture on the left side of the slide shows group policy disabling LLMNR. So, to adjust everything both a combination of techniques is necessary.

An additional setting worth mentioning for NetBIOS is setting the registry key `HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\NetBT\Parameters\NodeType` to `0x2` which stands for P-node. Setting a machine to P-node makes it so that it will not respond to NetBIOS broadcasts. Why would you do this if you are using the PowerShell script to disable NetBIOS? Setting this is a failsafe in case a machine is not running the PowerShell script. Asset management is difficult and having 100% of assets all configured the same way is near impossible.

Asset management tools come a long way in helping, but regardless scripting is required at some point.

Desired State Configuration (DSC)¹

PowerShell version 4 and later support DSC

- DSC is a configuration management platform
- Pushes and reports on custom configurations
- Assets outside of compliance trigger alerts or remediation

DSC supports configuration pushes and pulls

- Assets reach out to pull servers
- Pull server can be on-premise or in cloud

DSC works for both **Windows** and **Linux** operating systems

Desired State Configuration (DSC)¹

Microsoft PowerShell version 4 and later supports Desired State Configuration or DSC. DSC is a configuration management platform using PowerShell. DSC allows auditing a system's state, reporting on it, and changing settings so that the asset is within compliance. DSC and PowerShell work for both Linux and Windows operating systems making DSC a scripting solution for most major operating systems.

DSC can be used in a push fashion. A push consists of a management system reaching out to machines to apply configurations. The alternative deployment is to establish a pull server. Using a pull server means assets routinely check in with a server to audit or apply all desired state configurations. The pull server can be an on-premise server or a cloud-hosted Windows server. A third option is to use Azure Automation which is a cloud provided DSC pull server. Again, DSC works for both Windows and Linux.

Below is an example of disabling NetBIOS with DSC and Configuration Manager. Notice, DSC checks to see if something is set to a specific value. If it is not DSC can change it. As such the DSC logic is different than traditional asset management.

```
$nics=$null
$nics = (gwmi Win32_NetworkAdapterConfiguration -Filter 'ipenabled = "true"')
foreach ($nic in $nics) {
    If ($nic.TcpipNetbiosOptions -ne 2) {
        $nic.SetTcpipNetbios(2)
    }
}
```

[1] <https://docs.microsoft.com/en-us/powershell/dsc/decisionmaker>

Disable Direct Memory Access (DMA) Devices

System hardening includes securing physical data and access

- Disk encryption secures boot process and data at rest
- Authentication attempts to limit access to data

Physical attacks also possible through DMA devices

- **1394 Firewire** and **Thunderbolt** can access memory directly
- Allows system level access or privilege escalation
- Windows 10 disables DMA when screen is locked

Group policy supports disabling installation of certain devices¹

PCI\CC_0C0A is the device ID type for DMA devices

Disable Direct Memory Access (DMA) Devices

Operating systems sometimes make use of direct memory access (DMA) to provide high-speed interfaces. Thunderbolt connections are an example. By using DMA, a Thunderbolt device can achieve high-speed data transfer. DMA functions by providing input and output directly to memory. DMA even bypasses the CPU to increase the overall speed. The problem with this is that a hardware device effectively has access to memory.

Attack tools allow abusing DMA. One such attack tool is inception². Inception uses DMA to gain system or root access and works on Windows, Linux, and Mac. Worse yet, inception works against systems using full disk encryption. Remember, full disk encryption protects data at rest as well as the boot process. With full disk encryption an attacker can turn on the machine, but he or she gets stuck at the login screen. Inception uses DMA once the machine is active to take over the system thus gaining access regardless of if the system is using full disk encryption.

Windows has a group policy called "Prevent installation of drivers matching these device setup classes." Enabling this and adding the entry d48179be-ec20-11d1-b6b8-00c04fa372a7 helps prevent attacks using Firewire or 1394 interfaces. Another group policy called "Prevent installation of devices that match these device IDs" should be enabled and set to PCI\CC_0C0A. This prevents the installation of plug and play devices that use the Thunderbolt controller.

To secure a Mac device from DMA attacks, you should set an EFI password. Setting an EFI password disables raw DMA access. However, the following command should also be run:

```
sudo pmset -a destroyfvkeyonstandby 1 hibernatemode 25
```

This command makes it so that the encryption key is removed from memory during hibernation events. Removing the key during hibernation helps to secure against attacks that would try to bypass DMA restrictions.

[1] <https://support.microsoft.com/en-us/help/2516445/blocking-the-sbp-2-driver-and-thunderbolt-controllers-to-reduce-1394-d>

[2] <https://github.com/carmaa/inception>

[3] <https://derflounder.wordpress.com/2012/02/05/protecting-yourself-against-firewire-dma-attacks-on-10-7-x/>

Disable Old Protocol Versions

System protocols include old versions for compatibility

- SMB on Windows 10 includes 3.1.1, 3.2.1, 2, and 1
 - SMB 1.0 disabled on Windows 10 version 1709 or later
- SSH on Linux includes version 1 and 2

Older versions come with critical vulnerabilities

- Disabling old versions requires modern operating systems
- Disabling SSH version 1 can be done per system
- Old SMB versions can only be disabled based on OS versions

Disable Old Protocol Versions

Backwards compatibility is the bane of security. While it may be necessary to maintain backward compatibility with systems a version or two prior organizations typically do not need to maintain backward compatibility to systems more than a decade old. Yet default settings often maintain support with ancient operating systems. For example, Windows heavily relies on SMB for communication. SMB 1.0 is supported by Windows Server 2003/XP and earlier. Windows Server 2016 and Windows 10 support SMB 3.1.1. Yet Windows Server 2016 and Windows 10 both come with SMB version 1.0 enabled by default.

Maintaining old protocol versions enables downgrade attacks. If an SSH host supports version 1 and version 2, an attacker can try to downgrade a connection to version 1 and then man-in-the-middle it. Windows hosts supporting SMB 1.0 have had multiple exploits such as Eternal blue. Eternal blue is an example of a remote exploit that allows attackers to compromise a Windows system and immediately gain SYSTEM access easily remotely.

Remediating these vulnerabilities is a simple task. Simply disable or remove old protocols. An example of disabling SMB version 1.0 is below.

```
Disable-WindowsOptionalFeature -Online -FeatureName smb1protocol
```

On Windows Vista, Windows 7, Server 2008, or Server 2008 R2 the command would be as below.

```
Set-ItemProperty -Path "HKLM:\SYSTEM\CurrentControlSet\Services\LanmanServer\Parameters" SMB1 -Type DWORD -Value 0 -Force
```

[1] <https://www.fireeye.com/blog/threat-research/2017/05/smb-exploited-wannacry-use-of-eternalblue.html>

[2] <https://www.giac.org/paper/gsec/2034/conducting-ssh-man-middle-attacks-sshmitm/103515>

SMB Compatibility Matrix

SMB version is negotiated during session negotiation

- Old operating systems cannot use newer SMB versions

OS	Server 2016 Windows 10	Server 2012 Windows 8	Server 2008 R2 Windows 7	Server 2008 Windows Vista	Older versions
Server 2016 Windows 10	SMB 3.1.1	SMB 3.0	SMB 2.1	SMB 2.0	SMB 1.0
Server 2012 Windows 8	SMB 3.0	SMB 3.0	SMB 2.1	SMB 2.0	SMB 1.0
Server 2008 R2 Windows 7	SMB 2.1	SMB 2.1	SMB 2.1	SMB 2.0	SMB 1.0
Server 2008 Windows Vista	SMB 2.0	SMB 2.0	SMB 2.0	SMB 2.0	SMB 1.0
Older versions	SMB 1.0	SMB 1.0	SMB 1.0	SMB 1.0	SMB 1.0

SMB Compatibility Matrix

This slide demonstrates the SMB versions requires between two Windows operating systems. For example, a Windows 10 system talking to a Windows Vista system would negotiate and use SMB 2.0. Basically, an organization can only disable older SMB versions based on the oldest operating system accessing a given asset.

[1] <https://blogs.technet.microsoft.com/josebda/2013/10/02/windows-server-2012-r2-which-version-of-the-smb-protocol-smb-1-0-smb-2-0-smb-2-1-smb-3-0-or-smb-3-02-are-you-using/>

IPv4 and IPv6

Modern systems run dual stacks with IPv4 and IPv6

- If one is not necessary, disable it

Security devices often tuned to IPv4 and not IPv6

- IPv6 is not new, but support for it is not as robust
- Having both confuses analysts

Consider an infection occurring over IPv4 on port 443

- Stage 2 download and C2 may be over IPv6 on port 443
- Then C2 and internal pivoting occurs over IPv6

IPv4 and IPv6

Modern operating systems come with support for both IPv4 and IPv6. This implementation is considered a dual stack. Running both IP versions poses a security risk. The problem is both IP versions have their own IP addresses and routes. Since both are active communication can occur using either IP version. This may not seem like a big deal, but it poses multiple challenges:

- Network security may not handle IPv4 and IPv6 the same way
- Network security may not be properly restricted for both IPv4 and IPv6
- Communication may start over IPv4 then switch to IPv6 to evade controls

Because of these challenges, it is recommended to disable IPv4 or IPv6 depending on what your organization is using. In most cases, IPv6 is not used and can be disabled.

Legacy Windows Settings

Backwards compatibility settings of Windows are insecure

- Network authentication can be LM, NTLM, or Kerberos
- Encryption and digital signing is not required

Recommended policies:

- LAN Manager authentication level (NTLMv2 only - refuse LM and NTLM)
- Digitally encrypt or sign secure channel data (when possible)
- Digitally sign communications (when possible)

Legacy Windows Settings

Windows, in particular, go out of its way to support backward compatibility. Many of the arguments about Windows being insecure stem from supporting backward compatibility. The good news is that many of these settings are easy to change assuming you are using modern versions of Windows. Outside SMB it is important to tune how network authentication and communication are handled.

Specifically, LAN manager authentication should be set to NTLMv2 only - refuse LM and NTLM. NTLMv2 is not new. In fact, it first was supported with Windows NT 4.0 SP4. Therefore, there is no reason not to force NTLMv2 over LM or NTLMv1. Also, the options to digitally encrypt and sign traffic should be set to when possible. These settings are enabled by default for service-side communication but are not enforced client-side. It is best practice to enable the "when possible" settings for both client-side and service-side policies.

[1] <https://docs.microsoft.com/en-us/windows/security/threat-protection/security-policy-settings/network-security-lan-manager-authentication-level>

[2] <https://docs.microsoft.com/en-us/windows/security/threat-protection/security-policy-settings/domain-member-digitally-encrypt-or-sign-secure-channel-data-always>

[3] <https://docs.microsoft.com/en-us/windows/security/threat-protection/security-policy-settings/domain-member-digitally-sign-secure-channel-data-when-possible>

[4] <https://www.itprotoday.com/strategy/nt-gatekeeper-enabling-ntlmv2-windows-nt-40-workstations>

Limit Enumeration

Some settings allow remote enumeration of Windows boxes

- Adjust these settings to limit adversary enumeration
 - **Disable** the **Allow anonymous SID/Name translation** GPO
 - 2003 R2 and earlier DCs have this enabled
 - Set **Named pipes that can be accessed anonymously** to **blank**
 - **Limit** or **disable** the **Remotely accessible registry paths** GPO
 - **Tune** the **Restrict clients allowed to make remote calls to SAM** GPO¹
 - Includes audit only mode and granular access
 - Requires testing to prevent accidental denial of service

Limit Enumeration

Windows also has some default settings that allow an attacker to perform internal reconnaissance against an organization. Many of these apply to older operating systems, but some can persist through upgrades or previous policies. In particular, policies that allow anonymous users to find information about users and their corresponding Security IDs is a bad idea. Group policies like Allow anonymous SID/Name translation can be disabled to help prevent this. Other settings that should be tuned to prevent anonymous network reconnaissance are:

- Do not allow anonymous enumeration of SAM accounts (Recommended to set to enabled)
- Do not allow anonymous enumeration of SAM accounts and shares (Recommended to set to enabled)
- Let Everyone permissions apply to anonymous users (Recommended to set to disabled)

Other settings require careful testing and adjustment. For example, named pipes which are used for inter-process communication may allow anonymous access by default. The policy “Named pipes that can be accessed anonymously” can be enabled and provided either an empty list or a list of named pipes that should allow anonymous access. However, this needs to be done with care as it can break things. Similarly, “Remotely accessible registry paths” can be configured to make certain registry keys available over the network or to make sure certain registry keys are not available over the network.

Another setting is the “Restrict clients allowed to make remote calls to SAM.” This setting is enabled by default in Windows Server 2016 and Windows 10 but can be pushed to older operating systems so long as a patch has been installed supporting the option. Enabling this and tuning it can help prevent credentials from being stolen from the SAM database.

[1] <https://docs.microsoft.com/en-us/windows/security/threat-protection/security-policy-settings/network-access-restrict-clients-allowed-to-make-remote-sam-calls>

Path Vulnerabilities

Attacks against Linux and Windows may abuse paths

- Unauthorized programs can run with unquoted paths
 - Operating system has to interpret and guess path
 - C:\Program Files\a.exe or C:\Program.exe Files\a.exe
- Another issue is improper permissions on critical files
 - Example: Service executable writeable by standard user

```

ServiceName : HTC Account Service
Path        : "C:\Program Files\HTC Account\Htc.Identity.Service.exe"
StartName   : LocalSystem
AbuseFunction : Invoke-ServiceAbuse -Name 'HTC Account Service'
CanRestart  : True
  
```

Path Vulnerabilities

Programs called from services, and scheduled tasks on both Windows and Linux need to be monitored. Improper calls or permissions to programs can lead to system compromise. For example, consider the below paths:

C:\Program Files\a.exe

“C:\Program Files\a.exe”

The first program call is unquoted. Because of this, the operating system attempts to translate the path. Because of space, the operating system may incorrectly interrupt the path such as running C:\Program.exe instead of C:\Program Files\a.exe. For this attack to work the attacker needs write access to the location the operating system attempts to load. In many cases, this requires Administrator access, but in some cases, it does not.

What is more common is for third-party applications to be installed with improper ACLs. The image in this slide shows a third-party program called HTC Account Service. The path to the service is quoted however the executable of Htc.Identity.Service.exe can be overwritten by a standard user. This allows an attacker to overwrite the executable. The next time the service starts the malicious executable is loaded instead. The executable then runs in the context of the service user. Unfortunately, this is often SYSTEM. PowerShell scripts can be used to find unquoted services and user writeable service executables.

[1] <https://www.commonexploits.com/unquoted-service-paths/>

[2] <https://trustfoundry.net/practical-guide-to-exploiting-the-unquoted-service-path-vulnerability-in-windows/>

Protecting BIOS

Hardening needs to include more than just the OS

- What about the BIOS and firmware?

BIOS password should be set on all machines

- Does not require physically touching systems
- Business class systems such as Dell, HP, IBM, support setting BIOS settings and passwords from within OS
- Example: Dell supports multiple ways of changing BIOS settings
 - Dell client configuration toolkit (CCTK) install allows silent switches¹
 - EXE can be built and pushed through asset management software²

Protecting BIOS

When hardening a system, the whole system needs to be considered. This includes the BIOS and firmware. Each system should have a BIOS password to prevent tampering with BIOS settings. Firmware should be consistently upgraded to fix security flaws and potentially increase performance.

The question is how can the BIOS and firmware be controlled? While these are more physical in nature, both can be controlled from within an operating system. All major business class hardware support applications that can change BIOS settings while booted into the operating system. Some hardware supports this in an out-of-band fashion through lights out card such as a Dell DRAC or HP iLO.

Consider a Dell OptiPlex workstation. The BIOS settings can be changed by installing the Dell Open Management Instrumentation or by using the Dell client configuration toolkit (CCTK). Both of these can be pushed through asset management tools or group policy. HP, IBM, and other business class hardware support changing settings in a similar fashion.

[1] <https://www.dell.com/support/article/us/en/19/sln143145/how-to-install-use-dell-client-configuration-toolkit?lang=en>

[2] https://community.spiceworks.com/how_to/123550-managing-the-dell-bios-remotely

Change Default Programs

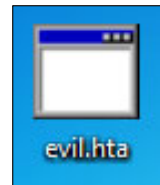
Certain extensions are dangerous and not used by users

- bat, com, hta, jar, js, jse, pif, ps1, vbe, vbs, wsf, wsh
- Consider changing extension to something like **notepad**
- Prevents execution of user double clicking on file

Does not prevent organization use or targeted attacks

- `cscript.exe evil.vbs` still works
- Double click on file will not

Organization can call files from authorized EXEs



Change Default Programs

Securing an operating system against attacks does not require fully removing the attack vector. An example of this is changing the default program for file extensions in Windows. As an example, an organization may use PowerShell or VBS scripts, but these are usually invoked from scheduled tasks, asset management jobs, or group policy scripts. End users outside of IT probably are not intended to run .vbs or .ps1 files directly. Yet malware may attempt to trick an end user into doing just that.

One way to limit the ability for end users to run malicious scripts is to change the default program associated with a dangerous file extension. Now, when an end user double clicks on one the file types it does not run yet, an organization can still invoke scripts in an automated fashion by calling the correct program to invoke the script.

The concept of changing the default program can actually be used to enhance detection. Instead of changing the default program to something benign like notepad.exe the default program could be set to a custom program that gathers information about the system and the file that was attempted to be run by an end user to generate an alert. Thus, executing of a potentially dangerous script or file is recorded as an alert. This alert can help to speed up investigations or to handle false positives.

[1] <https://www.howtogeek.com/137270/50-file-extensions-that-are-potentially-dangerous-on-windows/>

[2] <https://github.com/PaulSec/awesome-windows-domain-hardening>

Network Parameters

Windows

MSS (Legacy) GPO settings¹

IP source routing protection

- Enable - Set to highest protection

Allow ICMP redirects

- Disable

Allow the computer to ignore NetBIOS name release requests

- Enable

Linux

Disable IP forwarding

```
net.ipv4.ip_forward=0
```

Disable IP source routing

```
net.ipv4.conf.all.accept_source_route=0
```

Disable ICMP redirect acceptance

```
net.ipv4.conf.all.accept_redirects=0
```

```
net.ipv4.conf.default.accept_redirects=0
```

Log spoofed packets

```
net.ipv4.conf.all.log_martians=1
```

Network Parameters

System hardening involves more than software. It also involves how the network stack and traffic function. Windows, Linux, and Mac all have different ways of changing core network settings. On Linux settings can be set in `/etc/sysctl.conf`. The examples on the right of the slide are recommended settings to be placed in `/etc/sysctl.conf`.

Windows usually can be set with registry keys or group policy settings. The settings listed on the left require downloading and loading the MSS (Legacy) ADMX template from Microsoft. Once imported the settings can be controlled via group policy.

Many of these settings deal with protection from attacks such as IP source routing or ICMP redirects. IP source routing allows a packet to specify the route it should take. Attackers use this to hide their identity and location. ICMP redirects allow an ICMP packet to tell a destination system how to route traffic through a different gateway. Attackers use this in an attempt to perform a man-in-the-middle attack.

[1] <https://blogs.technet.microsoft.com/secguide/2016/10/02/the-mss-settings/>

[2] <https://www.computerworld.com/article/3144985/linux/linux-hardening-a-15-step-checklist-for-a-secure-linux-server.html>

[3] <https://wiki.ubuntu.com/ImprovedNetworking/KernelSecuritySettings>

Host Hardening Review

Hardening system pre-production is ideal

- Continuous hardening is best

Resources such as CIS benchmarks provide recommendations

- So, do online resources, vulnerability scanners, and government STIGs
 - Security Technical Implementation Guides (STIGs)

Remember to remove or disable unnecessary software and protocols

- And install security software and patch repetitively

Host Hardening Review

Organization assets need protection regardless of location. Whether you are using a laptop in an internet café or sitting at a workstation at a corporate location, your machine has the potential to be attacked. Because of this due diligence needs to be met by reducing the attack surface and adding additional security measures as much as possible.

The best implementations assume failure and continuously evaluate the security of assets. By doing so, you learn new hardening measures that are preventative or for detection. Start with guides found online or from a known organization such as CIS but do not take every recommendation at face value. Also, do not treat external hardening guides as a complete guide.

Course Roadmap

- Day 1: Defensible Security Architecture
- Day 2: Network Security Architecture
- Day 3: Network-Centric Application Security Architecture
- Day 4: Data-Centric Application Security Architecture
- **Day 5: Zero Trust Architecture**
- Day 6: Capstone: Design, Detect, Defend

CURRENT STATE ASSESSMENT, SOCS, AND SECURITY ARCHITECTURE

1. Zero Trust Architecture
2. Credential Rotation
3. Securing Traffic
4. EXERCISE: Network Isolation and Mutual Authentication
5. Host-Based Firewalls
6. Network Access Control (NAC)
7. Segmentation Gateways
8. Security Event Information Management (SIEM)
9. EXERCISE: SIEM Analysis and Tactical Detection
10. Log Collection
11. Audit Policies
12. Host Hardening
13. Patching
14. Tripwires and Red Herring Defenses
15. EXERCISE: Advance Defense Strategies

Course Roadmap

The next section covers host hardening.

Patching

Many vulnerabilities exist due to flaws in code

- Patching is necessary to fix flaws in code
- Should involve operating system and applications

Assumption of compromise is important

- But blatant disregard for patching is asking for trouble



Availability

Security

A patch solution is a tradeoff between up time and security

This page intentionally left blank.

Windows Server Update Services (WSUS)

WSUS is a free patch solution for Microsoft



- Includes patches for Microsoft operating systems
- As well as Microsoft applications (Office, SQL, and more)
- Third-party applications can be added to WSUS
 - Using scripts or more likely by manually addition
- Uses Background Intelligent Transfer Service (BITS)
 - Throttles bandwidth and impact to system
 - Handles connection interruptions

Can chain WSUS servers and supports mutual TLS to everything

Windows Server Update Services (WSUS)

[1] [https://docs.microsoft.com/en-us/previous-versions/tn-archive/cc539281\(v=technet.10\)](https://docs.microsoft.com/en-us/previous-versions/tn-archive/cc539281(v=technet.10))

[2] <https://www.itprotoday.com/management-mobility/publishing-third-party-updates-wsus>

Patch Remediation

How quickly do you need to install patches?

- PCI DSS states within 30 days of patch release
- CIS Critical Controls quick wins states 48 hours

CIS quick wins *"provide solid risk reduction without major procedural, architectural, or technical changes to an environment, or that provide such substantial and immediate risk reduction against very common attacks that most security-aware organizations prioritize these key controls"*

Many organizations struggle with 30 days let alone 48 hours

Patch Remediation

[1] <https://www.optiv.com/blog/pci-dss-the-30-day-patch-rule>

[2] <https://www.sans.org/critical-security-controls/guidelines>

PowerShell Patching

Automation is the key to successful patching

- Asset and patch management allow scheduling
- Scripts can take patching to a new level

**48 to 72 -
hour
patching**

Update-VMs is a PowerShell framework for patching virtual machines

- Supports automatic snapshots and removal
- Supports running optional health check scripts per VM
- Rolls back snapshot on failed reboot or health check
- Has built-in support for beta1, beta2, beta3, production rollouts

PowerShell Patching

[1] <https://github.com/HASecuritySolutions/Update-VMs>

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

Early Patching

Virtual machine templates and workstation images should include patches

- Multiple tools available for early patch integration
 - Microsoft Deployment Toolkit (MDT)
 - Automated Installation Kit (AIK)
 - Windows Deployment Services (WDS)
 - 3rd Party imaging and deployment solutions
- Scripts can auto merge patches with images
 - Still needs testing but testing can also be scripted

Early Patching

[1] <https://docs.microsoft.com/en-us/sccm/mdt/>

[2] <https://docs.microsoft.com/en-us/windows/deployment/windows-deployment-scenarios-and-tools>

[3] [https://docs.microsoft.com/en-us/previous-versions/tn-archive/cc265612\(v=technet.10\)](https://docs.microsoft.com/en-us/previous-versions/tn-archive/cc265612(v=technet.10))

Patching Review

Patching requires automation and process implementation

- WSUS provides a free solution for Microsoft and third-party patches
- Commercial solutions available with enterprise features
- Scripts capable of integrating with WSUS or commercial solution for more granular control and automation

Goal should be to patch quickly after patch release

- Automation is critical to increase system availability

This page intentionally left blank.

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

Course Roadmap

- Day 1: Defensible Security Architecture
- Day 2: Network Security Architecture
- Day 3: Network-Centric Application Security Architecture
- Day 4: Data-Centric Application Security Architecture
- **Day 5: Zero Trust Architecture**
- Day 6: Capstone: Design, Detect, Defend

CURRENT STATE ASSESSMENT, SOCS, AND SECURITY ARCHITECTURE

1. Zero Trust Architecture
2. Credential Rotation
3. Securing Traffic
4. **EXERCISE: Network Isolation and Mutual Authentication**
5. Host-Based Firewalls
6. Network Access Control (NAC)
7. Segmentation Gateways
8. Security Event Information Management (SIEM)
9. **EXERCISE: SIEM Analysis and Tactical Detection**
10. Log Collection
11. Audit Policies
12. Host Hardening
13. Patching
14. **Tripwires and Red Herring Defenses**
15. **EXERCISE: Advance Defense Strategies**

Course Roadmap

The next section covers Tripwires and Red Herring Defenses.

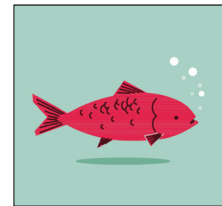
Red Herring¹

In literature "... the **red herring** is a deliberate diversion of attention with the intention of trying to abandon the original argument." ~ logicallyfallacious.com¹

- The concept can be applied as a defensive posture

Red herring defenses beneficial in conjunction with an assumption of compromise

- Redirect adversaries to avoid compromise
- Change behavior of automated and malicious tools
- Increase detection while slowing down attacks



Red Herring¹

Attackers do not play by the rules so why should defenders? Part of the reason is a fear of self-denial-of-service or accidentally breaking things. However, there are multiple techniques defenders can employ that raise the bar for attackers to succeed while simultaneously decreasing the time to detect and allowing new detection capabilities. Some of these techniques can be considered a red herring defense.

A red herring is a diversion to distract from or hide a truth. The concept of a red herring is constantly applied in literature to distract a reader. However, the concept applies well to securing systems and services. If a defender is able to hide the truth about what a system is running or doing without breaking the functionality of the system or service than an attacker is effectively redirected or blinded.

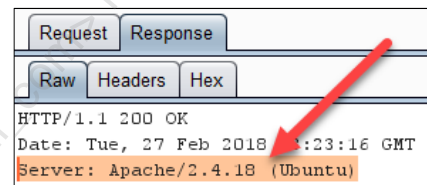
In some circles, this is referred to as deceptive security. However, organizations often have qualms about implementing something that implies deception. Thus, you may have greater success implementing something referred to as a red herring defense rather than asking for approval to implement deceptive security.

[1] <https://www.logicallyfallacious.com/tools/lp/Bo/LogicalFallacies/150/Red-Herring>

Service Banners

Many applications identify their software and version number upon connection

- Attackers and malware use information for exploitation
- Possible to change version, application, or OS identification
 - May be sufficient to break automated attack tools and scans
 - Implemented at the local server or with a reverse proxy
- Alternative is to hide or limit information
 - May break automated attacks
 - But less effective



Service Banners

An example of defenders playing by the rules is how services identify themselves. For example, when connecting to a web server, the default configuration will respond with the web server's application and version number. In some cases, this will even include server-side programming applications and versions. In effect, an organization has told an attacker exactly what they should attempt to exploit.

Instead, defenders can limit or even modify a service banner. By doing so, an attacker may believe a different application is in use. The attacker then may attempt to exploit an application that does not exist thus causing attacks to fail. The extra layer of protection by pretending to be another application is especially helpful against automated attacks and scans.

Tools are often written to identify applications and to exploit them using attributes such as service banners. Simply changing a service banner can be enough to break automated attack tools.

[1] https://www.howtoforge.com/changing-apache-server-name-to-whatever-you-want-with-mod_security-on-debian-6

Minimizing Service Banners

Services may natively support hiding or limiting service information

- **Apache** uses **ServerTokens** directive

`ServerTokens Full` (Full or OS default for Apache)

- Apache 2.4.18 (Ubuntu) PHP/7.2.2

`ServerTokens OS`

- Apache/2.4.18 (Ubuntu)

`ServerTokens ProductOnly` (banner with least info shown)

- Apache

Minimizing Service Banners

This slide demonstrates how to minimize the details presented when using the Apache web service. Inside the main configuration file for Apache is a setting called `ServerTokens`. On older Apache services the setting defaulted to `Full`. As a result, all HTTP responses from the Apache server would include the full Apache versions, operating system, and any additional frameworks such as PHP with a specific version number. Newer Apache services default to `OS` which includes the exact Apache version in use and the high-level operating system in use. This configuration may either be in `httpd.conf` or in newer operating systems can be found in `/etc/apache2/conf-enabled/security`.

Both default settings are horrible for security. By having the specific Apache version, an attacker can automatically or manually identify possible exploits specific to the version of Apache. To better protect the Apache web service the configuration can be changed to `ProductOnly`. This setting will present only Apache as the banner. No version numbers or additional information are presented.

However, it is possible that an application interacting with an Apache server may need the version information. While this is rare, it is possible thus explaining why the default configuration includes the Apache version information.

[1] <https://httpd.apache.org/docs/current/mod/core.html#servertokens>

Changing Service Banners

The alternative is to rewrite or modify the service banner

- Either done with local software or with reverse proxy
- A reverse proxy has extra benefits
 - The real banner can be made visible to select systems/users
 - Centralized control and management

Tools often target system using service information

- Means attack is unlikely to succeed
- And provides early warning of being targeted

Changing Service Banners

An alternative to minimizing a service banner is eliminating or rewriting the service banner to something else. Typically, this requires add-on software or some form of reverse proxy. For example, ModSecurity or a commercial web application firewall is capable of changing a web server's service banner. The change can be anything. For example, an Apache service can be portrayed as a Microsoft IIS service.

Using a reverse proxy has the added benefit of allowing centralized management and control of service banners. A single policy can be applied that identifies all protect web service assets. Doing so also allows detection rules to look for attacks targeting the fake service banner. Having a false service banner in place will lead to attacks to fail and also provide detection that an organization is being targeted. Another benefit of using a reverse proxy is that it can arbitrarily change the service banner depending on the source requesting a connection. For example, a web vulnerability scanner would likely need an exception to properly scan the web service for flaws while desktops and other assets could access the web services content regardless of how it presents itself.

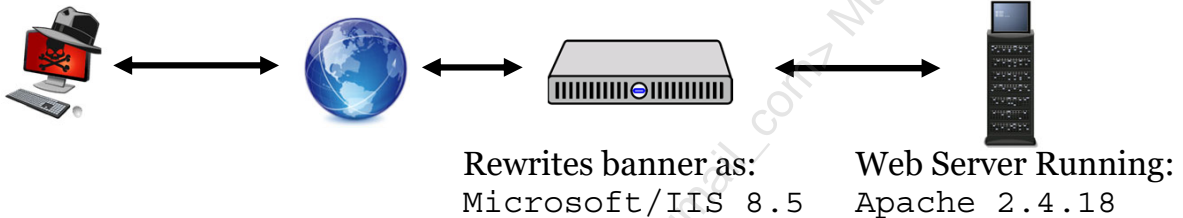
The option of rewriting content on-the-fly with a reverse proxy also allows rewriting other content attackers make use of such as X-Powered-By. The X-Powered-By field contains the programming language in use by a web service. Again, the information provides an attacker directly what they need to target. A similar option is available for IIS using URL Rewrite¹.

[1] <https://docs.microsoft.com/en-us/iis/extensions/url-rewrite-module/modifying-http-response-headers>

Detection Capabilities

Meaning of detection based on attack source and destination

- **External** - Early detection of external threat
 - Expected true positives (may occur so much as to be noise)
- **Internal** - Early detection of insider threat
 - Low false positives and true positives (high fidelity alert)



Detection Capabilities

Changing a service banner adds additional protection and, in some cases, significantly increases detection capabilities. For example, consider an Apache system being protected by being presented as an IIS server. If this service was public facing, then it would routinely have IIS exploits run against it. These would fail as the server is not really running IIS. Every IIS attack launches is a true positive. An attack was identified. However, not much can be done as attacks from the internet are expected.

Now consider the same service is presenting itself as an IIS service internally. If an IIS exploit or attack was launched against it what would it mean? Short of a vulnerability scanner, it would mean an internal system has been compromised and is being used to attack other internal systems. In this use case, identification of an IIS attack is just as important if not more important than preventing the attack. The importance is due to the fact that an internal system should never see an IIS attack or command being run against it if it truly is running Apache.

Modifying User-Agents

Proxy modification can be used to protect clients

- User-Agent describes web client application
- Malware uses User-Agent to identify and deliver malicious payloads
 - Example: **Metasploit Browser Autopwn¹**

Organization may use Google Chrome, but User-agent can be rewritten as Internet Explorer

- Can be applied to any web traffic
- Or possibly only to non-whitelisted websites

Modifying User-Agents

The concept of changing a service banner can also be applied to protect endpoints. When an endpoint reaches out to a web service, it identifies the web client being used via a User-Agent string. An example User-Agent string is as below:

```
Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/59.0.3071.115 Safari/537.36
```

The above User-Agent identifies a system is running a 64-bit copy of Windows 10 that is using Google Chrome version 59.0.3071.115. The Windows version is specified by the number following the string "Windows NT" and Chrome identifies its version number following Chrome. The information in a User-Agent string is verbose and useful to attackers.

An example of how attackers may use this can be found by looking at Metasploit's Browser Autopwn¹ module. This module waits for a client to connect to a malicious web link. When a victim connects to the web service the malicious web service attempts to identify information about the client and sends any possible exploits it has identified to the victim.

A reverse proxy is helpful for changing service banners of a web service, and a forward web proxy can be used to change a web clients User-Agent. Below is an example of using an F5 irule to change a client's User-Agent.

```
when HTTP_REQUEST {  
  if { [string toupper [HTTP::header "User-Agent"]] contains "MSIE 6" or "MSIE 7" or "MSIE 8" or "MSIE 9"  
    or "MSIE 10"} {  
    HTTP::header replace "User-Agent" "Mozilla/5.0 (Windows NT 10.0; WOW64; Trident/7.0; rv:11.0) like  
    Gecko"  
    log local0. "[HTTP::header User-Agent]"  
  }  
}
```

Changing a User-Agent string is best done for targeted sites such as all unknown sites or sites that are not explicitly business associated.

- [1] <https://blog.rapid7.com/2015/07/15/the-new-metasploit-browser-autopwn-strikes-faster-and-smarter-part-1/>
- [2] <https://devcentral.f5.com/questions/irule-to-rewrite-user-agent-header-52517>
- [3] <https://unix.stackexchange.com/questions/89592/how-to-change-user-agent-on-all-http-requests-made-from-my-machine-with-squid-in>

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

Honeypots

A honeypot is a system designed only to be attacked and monitored

- **High interaction** honeypots use real services
 - Often use vulnerable services and hope to be compromised for research
- **Low interaction** honeypots emulate services
 - Useful as early detection devices



Honeypots

Historically, honeypots have been a bad word to say within the security community. Doing so would immediately cause you to be outcasted as a rebel and an idiot. This was because of the controversies that first came with research honeypots. Today, however, there are multiple types of honeypots and they mean and do different things. At a high-level, a detection-based honeypot is a piece of software that is not intended to be accessed, but when it is, it generates alerts.

Multiple honeypots exist, and many are simple to stand up and centrally maintain. Also, the output logs are simple to collect with a SIEM. Unless stated otherwise, the term honeypot is used in this class to reference a virtual honeypot, not a research honeypot. Original honeypots were research honeypots. These were intentionally vulnerable systems that were intended to be compromised. These honeypot systems were full-blown operating systems set up with additional logging. In some cases, even going as far as intentionally applying a benevolent rootkit so that attackers were unaware they were on a honeypot. The main controversy around these systems is that they can be compromised and used to attack other systems. Also, they tended to be set up by defenders who could better spend their time increasing defenses internally. These research honeypots are still used today but primarily by security vendors studying and learning how to prevent or detect new attacks and malware.

A virtual honeypot, also coined as low-interaction honeypots, are software-based programs that emulate connections. For instance, an attacker could port scan a virtual honeypot, and the software would emulate real services to make it look real. These are designed for low interaction, so attackers can compromise the systems and attack others. The reason for some interaction rather than none is that it slows down attackers from targeting real systems while providing early detection.

[1] <http://www.omniseccu.com/security/infrastructure-and-email-security/low-interaction-honeypots-and-high-interaction-honeypots.php>

Low Interaction Honeypots

Multiple community projects offer easy to set up honeypots

- Often support multiple low interaction services
- And centralized management with web management

Two modern honeypot frameworks include:

- **Modern Honey Network (MHN)¹**
 - Provides centralized management, deployment, and collection
- **T-Pot²**
 - Easy deployment of multiple honeypots with advanced reporting

Low Interaction Honeypots

Historically the time commitment to deploy and maintain low interaction honeypots prevented successful implementations. However, today multiple free or open-source solutions exist that provide centralized deployment and management of unlimited low interaction honeypots. Two common honeypot frameworks are Modern Honey Network (MHN)¹ and T-Pot².

MHN focuses on simplifying honeypot deployment and management. It provides a web interface and automated scripts. T-Pot is a more holistic solution but with supported honeypot software and granularity. However, T-Pot installs quickly and uses an Elastic Stack interface to collect and store logs with a full reporting interface.

[1] <https://github.com/threatstream/mhn>

[2] <https://github.com/dtag-dev-sec/tpotce>

Redirecting to Honeypots

An internal only honeypot should see more than scans

- Records should point to honeypot
 - Common DNS records
 - OS shortcut links
 - Web links

```
C:\> nslookup vpn.sec530.com
Server: dns-server1.sec530.com
Address: 10.5.30.2

Name: web.sec530.com (Honeytoken)
Address: 10.5.30.17 (HoneyPot IP)
```

Under normal conditions, a honeypot should not be hit

- Now attacker scans or enumeration identifies them

Redirection to Honeypots

Traditional honeypots aim to detect external threats. However, a higher fidelity use case is to use honeypots to look for insider threats. When deployed internally a honeypot should not be accessed. Thus, no logs should be generated.

The initial thought is that a port scan or even something as simple as an ICMP echo request against an internal honeypot provides early detection. Yet there are other ways to increase detection capabilities. DNS A records that are commonplace but not used by an organization can be created that point to the IP address of a honeypot. For example, an attacker may attempt to enumerate systems using DNS lookups. One common DNS record is web. Yet if an organization does not use that name, the record can be created and reference a honeypot. Now access to the honeypot is more likely, and DNS logs also can be used as an extra layer of detection.

Similar approaches can be found by using operating system shortcut links or web links. The point is that if you are able to create something that should not be accessed and it either directly accesses a honeypot or cuts a log than a tactical detection mechanism is enabled.

[1] <https://tools.kali.org/information-gathering/dnsrecon>

Honeytokens

Fake objects or content is helpful for identifying unauthorized activity

- Fake objects or content is referred to as a **Honeytoken**
 - Sometimes referred to as a **Canarytoken**

Implementation requires data placement and logging

- Credit card excel sheet with file auditing
- Web bug links with web server's logs
- SQL record with stored procedure logging



Honeytokens

Digital content or objects can be implemented as early detection sensors as well. Special content created for the sole purpose of identifying unauthorized activity is referred to as a Honeytoken or in some cases a Canarytoken. A Honeytoken can be implemented on any endpoint or in applications such as databases.

Deploying honeytokens to servers and workstations weaponizes them as detection sensors. Fortunately, honeytokens are simple to deploy and maintain. On Windows systems, a single group policy can deploy multiple honeytokens with ease. Linux systems can use scripts or free asset management tools. The trick to a successful Honeytoken implementation is ease of deployment and some form of logging or action when the Honeytoken is accessed.

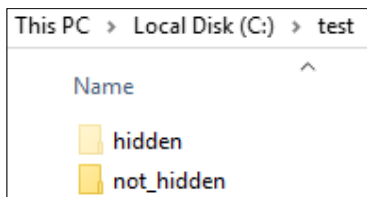
The next couple slides provide examples of honeytokens.

File Auditing

Automated scripts/malware often used to find patterns

- TOP SECRET, social security #, credit card #, etc.
- Operate by enumerating and reading through files
- Often ignores hidden folders

Enable file auditing using Windows GPO or Linux auditd



```
C:\test>dir /s
Directory of C:\test

03/27/2017  10:40 AM    <DIR>          not_hidden
Directory of C:\test\hidden
03/27/2017  10:40 AM                0 passwords.txt
Directory of C:\test\not_hidden
03/27/2017  10:40 AM                0 file.txt
```

File Auditing

One easy-to-implement example is implementing folders or files that are intended to never be accessed. Because most tools crawl hidden folders, it is possible to push out a hidden folder on all systems and audit it for access. Then, when an attacker or malware is scanning the filesystem, it will access the file or folder intended to never be accessed.

Doing so will log an event to Windows which can generate an alert from the SIEM. For this to work, other solutions—such as antivirus—need to be configured not to access these objects; or, the SIEM can be used for filtering.

This slide shows a hidden folder and a viewable folder, on the left. Then, on the right, the command `dir /s` is used to list the contents of each folder. Both the hidden and viewable folders are scanned for files. This shows how easy it is to crawl a hard drive regardless of whether files are hidden.

[1] <https://docs.microsoft.com/en-us/windows-server/identity/solution-guides/deploy-security-auditing-with-central-audit-policies--demonstration-steps->

Security Access Token (SAT) Honeytokens

Lateral movement usually from credential compromise

- Security Access Tokens (SAT) stolen from memory and reused
 - **Mimikatz**¹ is common attack tool to steal credentials
- Possible to place SAT honeytoken on all systems
 - Simple method is to run logon script via group policy
 - **MimikatzHoneyToken** project available on GitHub²
 - Or open source agent/server system available
 - **DCEPT**³ project provide easy to roll out agents and server
 - Creates honeytoken per system to pinpoint compromised asset

Security Access Token (SAT) Honeytokens

Even the fact that attackers and malware steal credentials can be used against them. In early 2015, Mark Baggett wrote an article called “Detecting Mimikatz Use On Your Network”². In it, he explains that the `runas` command in Windows can be used with a `/netonly` switch. When this is used, an arbitrary account and password can be specified to launch processes. Effectively, you can have a fake user account running notepad. This can be used as a honeytoken since the username, domain, and password are then in memory. Attackers stealing credentials may be tricked into grabbing a fake account that has no privileges or permissions.

One solution to quickly deploy a SAT honeytoken across an enterprise is to use **MimikatzHoneyToken**³. **MimikatzHoneyToken** allows for quick enterprise wide deployment of a SAT honeytoken using a logon script via group policy. This script causes a hidden process to be created with a honeytoken on all machines the group policy is enabled on. This script is available on GitHub. Be advised, some antivirus vendors mark it as malicious because it uses **AutoIT**⁴. If you use this script, be sure to customize it to your environment using the wiki found on its GitHub page.

A more sophisticated approach is to deploy **DCEPT**⁵. **DCEPT** is an open source agent/server solution. A server is deployed at a central location, and then a custom agent is deployed to all systems. Each agent obtains a honeytoken credential from a **DCEPT** generation server. This occurs once a day by default and is used to identify which system is compromised uniquely. A **DCEPT** sniffer runs alongside the **DCEPT** server looking for honeytoken credentials. If a honeytoken is identified the server looks up the credential used in the database to identify the specific asset that is compromised.

[1] <https://github.com/gentilkiwi/mimikatz>

[2] <https://www.dshield.org/diary/Detecting%2BMimikatz%2BUse%2BOn%2BYour%2BNetwork/19311>

[3] <https://github.com/SMAPPER/MimikatzHoneyToken>

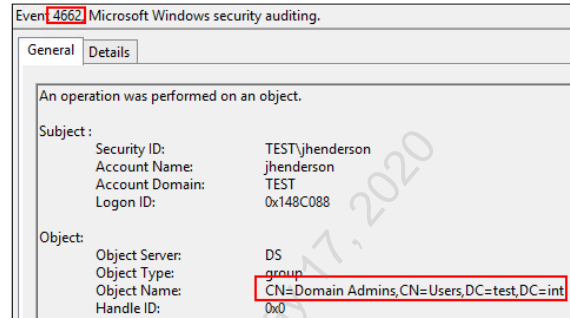
[4] <https://www.autoitscript.com/site/autoit/>

[5] <https://github.com/secureworks/dcept>

Auditing Attacker Reconnaissance

Certain insider threat activities are common

- Such as crawling folders/files
- Or finding who is in the domain administrators group
- Tactical auditing



```
C:\Users\jhenderson>net group "Domain Admins" /domain
The request will be processed at a domain controller for domain
-----
Administrator          jhenderson
SVC Backup              SVC Monitor
```

Auditing Attacker Reconnaissance

In Active Directory, almost every object (user, group, file, folder, printer, etc.) has extensive NTFS permission and auditing capabilities. It is not commonly known that you can audit and log any attempts to enumerate group membership. This is especially useful as, by default, all authenticated users can request to see who is a member of any group, including sensitive groups like Domain Administrators.

Mickey Perre has a blog online that walks through enabling auditing policies and setting up permissions on the Domain Administrators group. Then, when any user uses commands to see its members, the event ID 4662 is generated.

[1] <https://mickeysecurity.blogspot.com/2017/03/acsc-log-source-configuration.html>

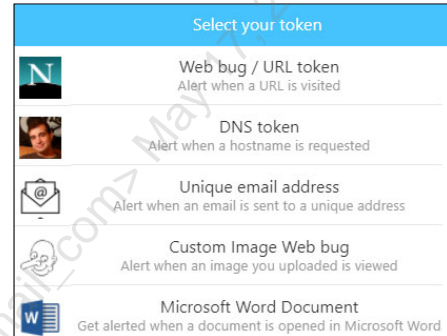
CanaryTokens.org¹

Thinkst² offers a free token service at **canarytokens.org**

- Thinkst also provides commercial honeypot solutions

GUI wizards provides walkthrough

- Takes minutes to deploy a token
- Wizard supports unique monitors
 - Detect website being cloned
 - Alert if AWS key is used
 - Identify if custom binary executed
 - Monitor folder being browsed



CanaryTokens.org¹

The website canarytokens.org is a free honeypot service offering by the commercial organization Thinkst. Canarytokens.org provides a wizard interface for quick deployment of a targeted honeypot. The solution supports many different honeypots. Upon selecting a use case and entering an email address the website generates detailed instructions on how to deploy the honeypot.

For example, a SQL honeypot generates a SQL query that creates a stored procedure and trigger. A Windows folder honeypot generates a desktop.ini file that is used when a folder is browsed. The cloned website honeypot uses JavaScript to identify a site has been cloned. It does so by using JavaScript that does nothing if the domain of the site being loaded matches the JavaScript. If the site is cloned and accessed without removing the JavaScript, then the domain will not match causing the JavaScript to phone home.

These and many more ideas can be implemented through canarytokens.org or by reverse engineering the callback to a system under your organization's control.

[1] <https://blog.thinkst.com/p/canarytokensorg-quick-free-detection.html>

[2] <https://thinkst.com/>

HALO (Honeytokens Against Leveraging OSINT)

Fake users can be created publicly to combat recon

- Could be just in hidden metadata and/or key public sites

Example: Peter Parker(pparker@sec530.com)

- On LinkedIn, Facebook, Adobe, PGP, GitHub, etc.
- Likely to be picked up during OSINT
- May eventually make compromised account lists
- Takes minimal time to set up... can get fairly elaborate

Activity from this account is malicious and provides context

HALO (Honeytokens Against Leveraging OSINT)

Keeping with the common theme of this course, defenders can use attacker techniques against them. In the case of OSINT, fake users can be created on the Internet. This involves creating a valid e-mail account and then signing up for public sites using this account. If you really want to get crazy, give them a full bio and sign up for as many common sites as possible. Focus on key sites and services used with OSINT such as PGP key sites. If you are lucky enough, the accounts will eventually be compromised if one of the public sites it is registered on gets breached. This would provide even higher chances of detecting adversaries trying to break in with the honeytoken account.

In this case, the fake user should never generate logs. For example, logon attempts to public mail access or VPNs using this account should never happen. If they do, it is because someone performed reconnaissance and found the account and is now trying to use it to target your organization.

If you like this technique, consider setting up developer accounts and fake code on forums. This is another common attacker vector—especially for web applications.

Tripwires and Red Herring Review

A well placed virtual tripwire or diversion greatly aid in:

- Early detection
- Gaining time to catch and deal with adversaries

Consider using simple configuration changes or tools such as:

- File, folder, and object auditing
- Honeypots
- Honeytokens

Tripwires and Red Herring Review

Solutions that aim to trick adversaries should focus on ease of deployment and an ability to provide early detection. Honeypots, in particular, can be sophisticated and time-consuming. Instead, emphasis needs to prioritize time and the benefits reaped from the time spent. Internal low interaction honeypots with emphasis on quick deployment and logging with automation or central management minimize effort over external facing or high interaction honeypots that end up being noise.

Through the deployment of simple red herring techniques such as replacing service banners or User-agents or tripwires such as honeytokens, organizations can consume little time yet benefit from significant detection and prevention capabilities. These techniques increase the difficulty level of an attacker to perform automated attacks and thus greatly increase the level of difficulty first to compromise an asset and secondly to do so without being detected.

Course Roadmap

- Day 1: Defensible Security Architecture
- Day 2: Network Security Architecture
- Day 3: Network-Centric Application Security Architecture
- Day 4: Data-Centric Application Security Architecture
- **Day 5: Zero Trust Architecture**
- Day 6: Capstone: Design, Detect, Defend

CURRENT STATE ASSESSMENT, SOCS, AND SECURITY ARCHITECTURE

1. Zero Trust Architecture
2. Credential Rotation
3. Securing Traffic
4. **EXERCISE: Network Isolation and Mutual Authentication**
5. Host-Based Firewalls
6. Network Access Control (NAC)
7. Segmentation Gateways
8. Security Event Information Management (SIEM)
9. **EXERCISE: SIEM Analysis and Tactical Detection**
10. Log Collection
11. Audit Policies
12. Host Hardening
13. Patching
14. Tripwires and Red Herring Defenses
15. **EXERCISE: Advance Defense Strategies**

Course Roadmap

The next section covers the concept of Zero Trust Architecture.

Exercise 5.3: Advanced Defense Strategies

- Exercise 5.3 is in the digital wiki found in your student VM (recommended)
- Alternatively, you may use your Workbook

This page intentionally left blank.

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