Blue Team Guides

HADESS

WWW.HADESS.IO

Introduction

Blue Team Guides

In the cohesive world of digital defense, documentation stands as the keystone, ensuring that operations, protocols, and strategies are not only well-devised but also seamlessly communicated and perpetually refined. "Blue Team Guides," an intricate and pivotal component of our blue team operation, is crafted as a comprehensive documentation suite that is meant to navigate through the vast spectrum of defensive cybersecurity.

Overview: The Essence of "Blue Team Guides"

"Blue Team Guides" isn't merely a documentation; it is a meticulously crafted arsenal of knowledge, insights, and guidelines that is shaped to empower organizations in crafting, enhancing, and refining their cybersecurity defenses. It serves as a repository of defensive strategies, operating procedures, tool guides, and case studies, embodying the collective wisdom derived from seasoned cybersecurity professionals and numerous real-world operations.

Brought to you by:

HADESS.IO

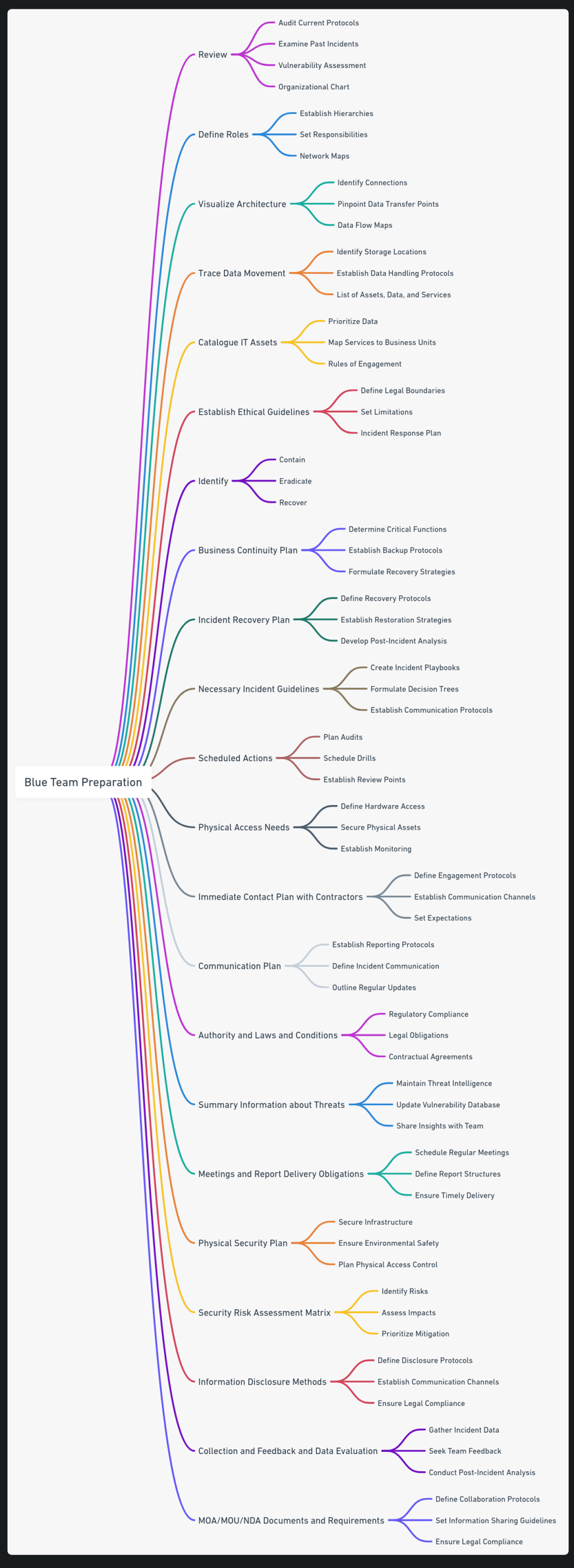
HADESS

HADESS performs offensive cybersecurity services through infrastructures and software that include vulnerability analysis, scenario attack planning, and implementation of custom integrated preventive projects. We organized our activities around the prevention of corporate, industrial, and laboratory cyber threats.

Next
Preparation

Last modified 1h ago

Preparation



1. Review

A rigorous review lays the foundation for an effective blue team operation. Before diving into the technicalities of cybersecurity defense, a comprehensive audit of existing protocols, past incidents, vulnerabilities, and overall cybersecurity posture is vital. This phase ensures that the blue team thoroughly understands the organization's current state, potential weaknesses, and the effectiveness of existing controls, thereby allowing them to identify areas that demand immediate attention and fortification.

2. Organizational Chart

Crafting a transparent and functional organizational chart ensures clear demarcation of roles, responsibilities, and hierarchies within the blue team. It assists in creating a streamlined communication flow, ensuring that every member comprehends their duties, reporting lines, and collaborative structures. Such clarity fosters efficiency during operations and incidents, minimizing confusion and enhancing the team's ability to swiftly and cohesively respond to threats.

3. Network Maps

Developing intricate network maps involves visualizing the organization's network architecture, including all internal and external connections, devices, and data flow. Understanding and visualizing how data traverses through various components of the network enhance the blue team's capability to identify potential vulnerabilities, establish robust monitoring points, and ensures rapid response in isolating incidents and containing threats when they emerge.

4. Data Flow Maps

Data flow maps serve as the blueprint for how information moves within the organization. They guide the blue team in understanding where critical data resides, how it is accessed, and how it moves through various processes. By intimately understanding the data's journey, the blue team can pinpoint critical junctures to monitor, ensuring that anomalies and potential data breaches are swiftly detected and mitigated.

5. List of Assets, Data, and Services

Creating a comprehensive list of assets, data, and services entails documenting every component that plays a role in the organization's information and operational technology environments. This inventory allows the blue team to categorize and prioritize assets based on their criticality to business operations, thereby enabling them to tailor their protective strategies and incident response plans to safeguard vital components effectively.

6. Rules of Engagement and Limitations and Boundaries

Establishing clear rules of engagement, along with defining limitations and boundaries, is pivotal in creating a functional and ethical operating environment for the blue team. These guidelines dictate how the team engages with both internal stakeholders and external entities, ensuring that their actions, especially during incidents, are in compliance with legal, ethical, and organizational norms.

7. Incident Response Plan

The Incident Response Plan (IRP) serves as the blueprint for how the blue team addresses and manages cybersecurity incidents. It outlines the protocols for identifying, containing, eradicating, and recovering from incidents while ensuring that vital business processes are affected minimally. A well-crafted IRP not only mitigates the impact of incidents but also aids in preserving evidence and learning from events to bolster future defenses.

8. Business Continuity Plan

A robust Business Continuity Plan (BCP) ensures that the organization can maintain or swiftly resume critical operations in the face of a cybersecurity incident. The BCP details strategies for preserving the availability, integrity, and confidentiality of critical business processes and data, thereby ensuring that the organization can sustain its vital functions even amidst a cyber crisis.

9. Incident Recovery Plan

The Incident Recovery Plan (IRP) is pivotal in orchestrating the restoration of systems and services following a cybersecurity incident. Focused on timelines, data restoration, service enablement, and minimizing prolonged impacts to business operations, the IRP is integral in ensuring that the organization rebounds with minimized damages and enhanced post-incident postures to thwart future vulnerabilities.

10. Necessary Incident Guidelines

Encompassing strategies, step-by-step actions, and decision-making frameworks, necessary incident guidelines serve as the tactical manual for the blue team. These guidelines offer clear directives during incidents, providing a path to navigate through the chaos and effectively mitigate, manage, and learn from cybersecurity events while ensuring aligned, consistent, and optimal actions amidst crisis situations.

11. Scheduled Actions

Scheduled actions refer to the periodic tasks that the blue team undertakes to ensure continuous robustness of the cybersecurity posture. This includes routine audits, vulnerability scanning, threat hunting exercises, and training drills, which are essential to maintaining a vigilant, prepared, and continuously improving cybersecurity defense mechanism within the organization.

12. Physical Access Needs

Accounting for physical access needs implies ensuring that the blue team can securely access the necessary physical infrastructure when required. Protecting data is not just a virtual endeavor; safeguarding hardware, ensuring secure physical storage, and controlling access to vital IT assets is paramount to ensuring a 360-degree defensive strategy, shielding data from both digital and physical threats.

13. Immediate Contact Plan with Contractors

The immediate contact plan with contractors assures that, in the wake of an incident or a need for specialized intervention, the blue team can swiftly engage with external experts and service providers. This plan will ensure smooth, rapid, and coordinated integration of external entities into the response and recovery operations, enhancing capabilities and resources during crucial moments.

14. Communication Plan

The Communication Plan outlines how information regarding incidents, threats, and cybersecurity postures are communicated within the organization and, when necessary, to external stakeholders. It ensures that accurate, timely, and appropriate information is conveyed, avoiding misinformation while ensuring that relevant parties are informed and aligned with the cybersecurity strategies and incidents.

15. Authority and Laws and Conditions

This component ensures that the blue team's operations are continuously compliant with prevailing legal, regulatory, and contractual obligations. It guides the team in ensuring that defensive strategies, incident responses, and data management practices are in line with legal requirements and organizational policies, safeguarding the organization from potential legal repercussions.

16. Summary Information about Threats

Maintaining a repository of summarized information about threats provides the blue team with a quick reference guide to known threat vectors, vulnerabilities, and indicators of compromise. This facilitates rapid identification, understanding, and response to threats, enhancing the team's capability to recognize and mitigate potential attacks swiftly and effectively.

17. Meetings and Report Delivery Obligations

Structured meetings and adherence to report delivery obligations ensure continuous alignment, information sharing, and strategic planning within the blue team. It supports the orchestration of a unified, informed, and cohesive team, ensuring that insights, threats, and strategies are collectively understood, analyzed, and acted upon.

18. Physical Security Plan

Encompassing strategies to safeguard the physical aspects of the IT environment, the Physical Security Plan addresses measures to protect hardware, data centers, and other physical assets from unauthorized access, theft, and damage. This ensures the integrity and availability of the physical infrastructure that supports the digital realm of the organization.

19. Security Risk Assessment Matrix

The Security Risk Assessment Matrix is a structured tool utilized by the blue team to identify, quantify, and prioritize cybersecurity risks within the organization. It enables the team to systematically evaluate threats, vulnerabilities, and potential impacts, guiding them towards crafting strategic defenses, allocating resources effectively, and ensuring that the highest risk areas are adequately mitigated to safeguard the organization's cyber environment.

20. Information Disclosure Methods

Navigating through information disclosure methods involves determining the strategies and protocols for how and when the organization discloses information related to cybersecurity incidents, vulnerabilities, and defenses. Managing disclosure ensures that information is communicated in a manner that safeguards the organization's reputation, complies with legal obligations, and potentially enables collaborative defense with external entities in the cybersecurity community.

21. Collection and Feedback and Data Evaluation

Engaging in the collection, feedback, and evaluation of data related to cybersecurity incidents, defenses, and overall cyber posture empowers the blue team with insights to perpetually refine their strategies. This iterative process ensures that the defensive approach is continuously enhanced, aligning with the evolving threat landscape and ensuring that past incidents and vulnerabilities translate into future preparedness and resilience.

22. MOA/MOU/NDA Documents and Requirements

Managing Memorandums of Agreement (MOA), Memorandums of Understanding (MOU), and Non-Disclosure Agreements (NDA) curate a legal and collaborative framework that defines how the organization, and by extension, the blue team, interacts, shares information, and collaborates with external entities. These documents ensure that collaborative and external engagements are defined, safeguarding the organization's interests while enabling structured cooperation with external partners, vendors, and cybersecurity entities.

← Previous
Introduction

Next
Identify Scope →

Last modified 3h ago

Identify Scope

Identification (Domain)

- Scan and Vulnerabilities

NMAP Command

- Using Ping sweep for the network:

```
shellCopy code# nmap -sn -PE <IP ADDRESS OR RANGE>
```

- Scan and display open ports:

```
shellCopy code# nmap --open <IP ADDRESS OR RANGE>
```

- Determine open services:

```
shellCopy code# nmap -sV <IP ADDRESS>
```

- Scan http and https (tcp) ports:

```
shellCopy code# nmap -p 80,443 <IP ADDRESS OR RANGE>
```

- Scan DNS (udp):

```
shellCopy code# nmap -sU -p 53 <IP ADDRESS OR RANGE>
```

- Scan UDP and TCP together, be verbose on a single host and include optional skip ping:

```
shellCopy code# nmap -v -Pn -SU -ST -p U:53,111,137,T:21-25,80,139,8080 <IP ADDRESS>
```

NESSUS Command

- Basic Nessus Scan:

```
shellCopy code# nessus -q -x -T html <NESSUS SERVER IP ADDRESS> <NESSUS SERVER PORT 124
# nessus [-vnh] [-c .refile] [-VJ [-T <format>]
```

- Batch-mode Scan:

```
shellCopy code# nessus -q [-pPS] <HOST> <PORT> <USERNAME> <PASSWORD> <targets-file> <res
```

- Get the report:

```
shellCopy code# nessus -i in.[nsrlnbe] -o out.[xmllnsrlnbelhtmlltxt]
```

OPENVAS Command

- Step 1: Install server, client, and plugins:

```
shellCopy code# apt-get install openvas-server openvas-client openvas-plugins-base openv
```

- Step 2: Update the vulnerability database:

```
shellCopy code# openvas-nvt-sync
```

- Step 3: Add a user to the client:

```
shellCopy code# openvas-adduser
```

- Step 4: Log in: sysadm

- Step 5: Authenticate (pass/cert) [pass]: [HIT ENTER]
- Step 6: Enter password: Based on the added user policies
- Step 7: Allow the user to scan networks requiring authentication:

```
shellCopy codeaccept <YOUR IP ADDRESS OR RANGE>
default deny
```

- Step 8: Use Ctrl+D key combination to exit.
- Step 9: Start the server:

```
shellCopy code# service openvas-server start
```

- Step 10: Choose the target for the scan: Create a file containing the targets.

```
shellCopy code# vi scanme.txt
```

- Step 11: Add various hosts on each line:

```
shellCopy code<IP ADDRESS OR RANGE>
```

- Step 12: Begin scan:

```
shellCopy code# openvas-client -q 127.0.0.1 9390 sysadm nsrc+ws scanme.txt openvas-outpu
```

- Step 13: (Optional) Start the scan in HTML format:

```
shellCopy code# openvas-client -q 127.0.0.1 9390 sysadm nsrc+ws scanme.txt openvas-outpu
```

Windows

- Network Identification
- Basic Network Identification:

```
shellCopy codeC:> net view /all
C:> net view \\<HOST NAME>
```

- Using ping to scan and save the result in a file:

```
shellCopy codeC:\> for /L %I in (1,1,254) do ping -w 30 -n 1 192.168.1.%I | find "Reply"
```

```
bashCopy codenbtscan <IP ADDRESS OR RANGE>
```

- Basic nbtstat scan:

```
bashCopy code# find /<PATHNAME TO ENUMERATE> -type f -exec md5sum {} >> md5sums.txt \;
```

- Hashing all executable files in a specific path:

```
bashCopy coderndc querylog
# tail -f /var/log/messages | grep named
```

- DNS reporting start and viewing DNS reports:

```
bashCopy code# cat /var/lib/dhcpd/dhcpd.leases
# grep -Ei 'dhcp' /var/log/syslog.1
```

- View DHCP reports on Red Hat 3 and Ubuntu:

```
bashCopy code# smbtree -b
```

- Network Identification:

Linux

```
batchCopy codeC:\> dsquery ou DC=<DOMAIN>,DC=<DOMAIN EXTENSION>
```

- Commands to list all OUs, workstations, servers, domain controllers, and more:

Active Directory Inventory

```
batchCopy codeC:\> mbsacli.exe /target <TARGET IP ADDRESS> /n os+iis+sql+password
```

- Basic scans for target IP, IP range, domain, and names within a text file:

Microsoft Baseline Security Analyzer (MBSA)

```
batchCopy code:: batch script lines to test usernames and passwords against a target IP
```

- Guess or check password:

Passwords

```
batchCopy codeC:\> for /L %i in (1,1,254) do psloggedon \\192.168.1.%i >> C:\users\_outp
```

- Loop scan script:

```
batchCopy codeC:\> psloggedon \\computername
```

- Display logged-on user:

User Activities

```
batchCopy codeC:\> nbtstat -A <IP ADDRESS>
C:\> for /L %I in (1,1,254) do nbtstat -An 192.168.1.%I
```

- Basic nbtstat scan and loop scan script:

NETBIOS

```
batchCopy codeC:\> Get-FileHash <FILE TO HASH> | Format-List
C:\> certutil -hashfile <FILE TO HASH> SHA1
```

- And other hash, file verification, and checksum operations with commands such as:

```
batchCopy codeC:\> fciv.exe <FILE TO HASH>
C:\> fciv.exe c:\ -r -md5 -xml <FILE NAME>.xml
```

- Using the File Checksum Integrity Verifier (FCIV) software:

Hashing

```
batchCopy codeC:\> DNSCmd <DNS SERVER NAME> /config /LogFilePath <PATH TO LOG FILE>
C:\> DNSCmd <DNS SERVER NAME> /config /logfilemaxsize 0xffffffff
```

- Log path setup, log file size configuration, etc.:

```
batchCopy codeC:\> DNSCmd <DNS SERVER NAME> /config /logLevel 0x8100F331
```

- Enabling DNS Logging:

```
batchCopy codeC:\> %SystemRoot%\System32\Dns
C:\> %SystemRoot%\System32\Winevt\Logs\DNS Server.evtx
C:\> %SystemRoot%\System32\Winevt\Logs\Microsoft-Windows-DNSServer%4Analytical.etl
```

- Default paths for various Windows versions:

DNS

```
batchCopy codeC:> %windir%\System32\Dhcp
```

- Default paths for various Windows versions:

```
batchCopy codeC:\> reg add HKLM\System\CurrentControlSet\Services\DhcpServer\Parameters
```

- Enabling DHCP Reports:

DHCP

- Enabling DHCP Reports:

```
batchCopy codeC:\> reg add HKLM\System\CurrentControlSet\Services\DhcpServer\Parameters
```

- Default paths for various Windows versions:

```
batchCopy codeC:> %windir%\System32\Dhcp
```

DNS

- Default paths for various Windows versions:

```
batchCopy codeC:\> %SystemRoot%\System32\Dns
C:\> %SystemRoot%\System32\Winevt\Logs\DNS Server.evtx
C:\> %SystemRoot%\System32\Winevt\Logs\Microsoft-Windows-DNSServer%4Analytical.etl
```

- Enabling DNS Logging:

```
batchCopy codeC:\> DNSCmd <DNS SERVER NAME> /config /logLevel 0x8100F331
```

- Log path setup, log file size configuration, etc.:

```
batchCopy codeC:\> DNSCmd <DNS SERVER NAME> /config /LogFilePath <PATH TO LOG FILE>
C:\> DNSCmd <DNS SERVER NAME> /config /logfilemaxsize 0xffffffff
```

Hashing

- Using the File Checksum Integrity Verifier (FCIV) software:

```
batchCopy codeC:\> fciv.exe <FILE TO HASH>
C:\> fciv.exe c:\ -r -md5 -xml <FILE NAME>.xml
```

- And other hash, file verification, and checksum operations with commands such as:

```
batchCopy codeC:\> Get-FileHash <FILE TO HASH> | Format-List
C:\> certutil -hashfile <FILE TO HASH> SHA1
```

NETBIOS

- Basic nbtstat scan and loop scan script:

```
batchCopy codeC:\> nbtstat -A <IP ADDRESS>
C:\> for /L %I in (1,1,254) do nbtstat -An 192.168.1.%I
```

User Activities

- Display logged-on user:

```
batchCopy codeC:\> psloggedon \\computername
```

- Loop scan script:

```
batchCopy codeC:\> for /L %i in (1,1,254) do psloggedon \\192.168.1.%i >> C:\users\_outp
```

Passwords

- Guess or check password:

```
batchCopy code:: batch script lines to test usernames and passwords against a target IP
```

Microsoft Baseline Security Analyzer (MBSA)

- Basic scans for target IP, IP range, domain, and names within a text file:

```
batchCopy codeC:\> mbsacli.exe /target <TARGET IP ADDRESS> /n os+iis+sql+password
```

Active Directory Inventory

- Commands to list all OUs, workstations, servers, domain controllers, and more:

```
batchCopy codeC:\> dsquery ou DC=<DOMAIN>,DC=<DOMAIN EXTENSION>
```

Linux

- Network Identification:

```
bashCopy code# smbtree -b
```

- View DHCP reports on Red Hat 3 and Ubuntu:

```
bashCopy code# cat /var/lib/dhcpd/dhcpd.leases
# grep -Ei 'dhcp' /var/log/syslog.1
```

- DNS reporting start and viewing DNS reports:

```
bashCopy coderndc querylog
# tail -f /var/log/messages | grep named
```

- Hashing all executable files in a specific path:

```
bashCopy code# find /<PATHNAME TO ENUMERATE> -type f -exec md5sum {} >> md5sums.txt \;
```

- Basic nbtstat scan:

```
bashCopy codenbtscan <IP ADDRESS OR RANGE>
```

- Guess Passwords:

```
while read line; do username=$line; while read
line; do smbclient -L <TARGET IP ADDRESS> -U
$username%$line -g -d 0; echo $username:$line;
done<<PASSWORDS>.txt;done<<USER NAMES>.txt
```


Previous
Preparation

Next
Protect Defend

Last modified 3h ago

Protect Defend

Protection and Defense Windows

Disabling or Stopping Services

List of stopped or disabled services:

```
C:\> sc query
C:\> sc config "<SERVICE_NAME>" start= disabled
C:\> sc stop "<SERVICE_NAME>"
C:\> wmic service where name='<SERVICE_NAME>' call ChangeStartmode Disabled
```

Host Firewall

View all rules:

```
C:\> netsh advfirewall firewall show rule name=all
```

Enable or disable the firewall:

```
C:\> netsh advfirewall set currentprofile state on
C:\> netsh advfirewall set currentprofile firewallpolicy blockinboundalways,allowoutbour
C:\> netsh advfirewall set publicprofile state on
C:\> netsh advfirewall set privateprofile state on
C:\> netsh advfirewall set domainprofile state on
C:\> netsh advfirewall set allprofile state on
C:\> netsh advfirewall set allprof ile state off
```

Setting a New Rule for the Firewall:

```
C:\> netsh advfirewall firewall add rule name="Open Port 80" dir=in action=allow protoc

C:\> netsh advfirewall firewall add rule name="My Application" dir=in action=allow prog

C:\> netsh advfirewall firewall add rule name="My Application" dir=in action=allow prog

C:\> netsh advfirewall firewall delete rule name=rule name program="C:\MyApp\MyApp.exe"

C:\> netsh advfirewall firewall delete rule name=rule name protocol=udp localport=500

C:\> netsh advfirewall firewall set rule group="remote desktop" new enable=Yes profile=

C:\> netsh advfirewall firewall set rule group="remote desktop" new enable=No profile=pu
```

Setting the Location of Reports:

```
C:\> netsh advfirewall set currentprofile logging C:\<LOCATION>\<FILE_NAME>
```

Setting and Changing the Location of Firewall Reports:

```
C:\> more %systemroot%\system32\LogFiles\Firewall\pfirewall.log

C:\> netsh advfirewall set allprofile logging maxfilesize 4096

C:\> netsh advfirewall set allprofile logging droppedconnections enable

C:\> netsh advfirewall set allprofile logging allowedconnections enable
```

Viewing Firewall Reports:

```
PS C:\> Get-Content $env:systemroot\system32\LogFiles\Firewall\pfirewall.log
```

Passwords

- **Changing the Password:**

```
C:\> net user <USER_NAME> * /domain
C:\> net user <USER_NAME> <NEW_PASSWORD>
```

- **Changing Password Remotely:** Source: Technet Microsoft

```
PS C:\> pspasswd.exe \\<IP_ADDRESS_or_NAME_OF_REMOTE_COMPUTER>
```

Host Files

- **Resetting DNS:**

```
C:\> ipconfig /flushdns
```

- **Resetting NetBios Cache:**

```
C:\> nbtstat -R
```

- **Adding Malicious Domain and Redirecting it to Localhost:**

```
C:\> echo 127.0.0.1 <MALICIOUS_DOMAIN> >> C:\Windows\System32\drivers\etc\hosts
```

- **Checking Host Files by Pinging 127.0.0.1:**

```
C:\> ping <MALICIOUS_DOMAIN> -n 1
```

Whitelist

- **Creating and Using a Proxy Auto Config (PAC) File for Suspicious URLs and IPs:**

```
function FindProxyForURL(url, host) {
    // Send bad DNS name to the proxy
    if (dnsDomainIs(host, ".badsite.com"))
        return "PROXY http://127.0.0.1:8080";
    // Send bad IPs to the proxy
    if (isInNet(myIpAddress(), "222.222.222.222", "255.255.255.0"))
        return "PROXY http://127.0.0.1:8080";
    // All other traffic bypass proxy
    return "DIRECT";
}
```

Application Restrictions

- **Using Applocker - for Server 2008 R2, Windows 7, or higher:**
  - Rules for executable files (.exe, .com)
  - DLL rules (.dll, .ocx)
  - Script rules (.ps1, .bat, .cmd, .vbs, .js)
  - Installation program rules (.msi, .msp, .mst)

Working Steps with Applocker (Requires GUI):

Step 1: Create a new GPO.

Step 2: Right-click on it to edit, then navigate through `Computer Configuration > Policies > Windows Settings > Security Settings > Application Control Policies > Applocker`. Click "Configure Rule Enforcement".

Step 3: Under "Executable Rules", check the "Configured" box and ensure "Enforce Rules" is selected from the drop-down box. Click "OK".

Step 4: In the left pane, click "Executable Rules".

Step 5: Right-click in the right pane and select "Create New Rule".

Step 6: On the "Before You Begin" screen, click "Next".

Step 7: On the "Permissions" screen, click "Next".

Step 8: On the "Conditions" screen, select the "Publisher" condition and click "Next".

Step 9: Click the "Browse" button and navigate to any executable file on your system. It doesn't matter which one.

Step 10: Drag the slider up to "Any Publisher" and then click "Next".

Step 11: Click "Next" on the "Exceptions" screen.

Step 12: Name the policy, for example, "only run executables that are signed" and click "Create".

Step 13: If this is your first time creating an Applocker policy, Windows will prompt you to create a default rule, click "Yes".

Step 14: Ensure "Application Identity Service" is Running.

```
C:\> net start AppIDSvc
C:\> REG add "HKLM\SYSTEM\CurrentControlSet\services\AppIDSvc" /v Start /t REG_DWORD /d
```

Step 15: Changes require a reboot.

```
C:\> shutdown.exe /r
C:\> shutdown.exe /r /m \\<IP ADDRESS OR COMPUTER NAME> /f
```

Using the Applocker Module in PowerShell:

- **Import the Applocker Module:**

```
PS C:\> import-module Applocker
```

- **Display Information about Files and Executables in the Path C:\Windows\System32:**

```
PS C:\> Get-ApplockerFileInformation -Directory C:\Windows\System32\ -Recurse -FileType
```

- **Create an Applocker Policy for All Executable Files in the Path C:\Windows\System32:**

```
PS C:\> Get-ApplockerFileInformation -Directory C:\Windows\System32\ -Recurse -FileType
```

- **Create an Applocker Policy to Allow All Executable Files in the Path C:\Windows\System32:**

```
PS C:\> Get-Childitem C:\Windows\System32\*,exe | Get-ApplockerFileInformation | New-App
```

- **Change Existing Policies Using the File C:\Policy.xml:**

```
PS C:\> Set-AppLockerPolicy -XMLPolicy C:\Policy.xml
```

- **Use Applocker Policies to Allow Running notepad and calc for Users Who are Members of the 'everyone' Group:**

```
PS C:\> Test-AppLockerPolicy -XMLPolicy C:\Policy.xml -Path C:\Windows\System32\calc.exe
```

- **Create a Restriction for the Number of Executions:**

```
PS C:\> Get-ApplockerFileInformation -EventLog -Logname "Microsoft-Windows-Applocker\EXE
```

- **Create a Policy for Applocker from Audited Events for exe and dll Files:**

```
PS C:\> Get-ApplockerFileInformation -EventLog -LogPath "Microsoft-Windows-AppLocker/EXE
```

Extracting All Applocker Policies:

```
PS C:\> Get-AppLockerPolicy -Local | Test-AppLockerPolicy -Path C:\Windows\System32\*.ex
```

Review and Test the Extracted Applocker Policy File:

```
PS C:\> Get-Childitem <DirectoryPathToReview> -Filter <FileExtensionFilter> -Recurse | (
```

Display a GridView List for All Rules:

```
PS C:\> Get-AppLockerPolicy -Local -Xml | Out-GridView
```

IPSEC Commands

Create a Local Security Policy for Applocker for Any Type of Connection and Protocol Using a Preshared Key:

```
C:\> netsh ipsec static add filter filterlist=MyIPsecFilter srcaddr=Any dstaddr=Any prot
C:\> netsh ipsec static add filteraction name=MyIPsecAction action=negotiate
C:\> netsh ipsec static add policy name=MyIPsecPolicy assign=yes
C:\> netsh ipsec static add rule name=MyIPsecRule policy=MyIPsecPolicy filterlist=MyIPse
```

Add a Rule for Allowing Ports 80 and 443 in IPSEC:

```
C:\> netsh ipsec static add filteraction name=Allow action=permit
C:\> netsh ipsec static add filter filterlist=WebFilter srcaddr=Any dstaddr=Any prot
C:\> netsh ipsec static add filter filterlist=WebFilter srcaddr=Any dstaddr=Any prot
```

Display All Local Security Policies in IPSEC Named "MyIPsecPolicy":

```
C:\> netsh ipsec static show policy name=MyIPsecPolicy
```

Stop or Disable Policies in IPSEC:

```
C:\> netsh ipsec static set policy name=MyIPsecPolicy
```

Create a New Policy, Rule, and Preshared Key for Any Type of Connection:

```
C:\> netsh advfirewall consec add rule name="IPSEC" endpoint1=any endpoint2=any action=r
```

Require a Preshared Key for All Outgoing Requests in IPSEC:

```
C:\> netsh advfirewall firewall add rule name="IPSEC_Out" dir=out action=allow enable=ye
```

Create a Rule for Web Browsing:

```
C:\> netsh advfirewall firewall add rule name="Allow Outbound Port 80" dir=out localport
```

Create a Rule for DNS:

```
C:\> netsh advfirewall firewall add rule name="Allow Outbound Port 53" dir=out localport
```

Delete Rule in IPSEC:

```
C:\> netsh advfirewall firewall delete rule name="IPSEC_RULE"
```

ACTIVE DIRECTORY (AD) and GROUP POLICY OBJECT (GPO)

Retrieve and Apply New Policies:

```
C:\> gpupdate /force
C:\> gpupdate /sync
```

Audit Success and Failure for User Bob:

```
C:\> auditpol /set /user:bob /category:"Detailed Tracking" /include /success:enable /fa
```

Create an Organization Unit to Transfer Suspect Users and Computers:

```
C:\> dsadd OU <QUARANTINE BAD OU>
```

Transfer active directory users to a new group "NEW GROUP":

```
PS C:\> Move-ADObject 'CN=<USER NAME>,CN=<OLD USER GROUP>,DC=<OLD DOMAIN>,DC=<OLD EXTENS
```

Similar Method:

```
C:\> dsmove "CN=<USER NAME>,OU=<OLD USER OU>,DC=<OLD DOMAIN>,DC=<OLD EXTENSION>" -newpar
```

System Without ACTIVE DIRECTORY (AD)

Prevent .exe file:

```
C:\> reg add "HKCU\Software\Microsoft\Windows\CurrentVersion\Policies\Explorer" /v Disa
C:\> reg add "HKCU\Software\Microsoft\Windows\CurrentVersion\Policies\Explorer\DisallowR
```

Disable Remote Desktop:

```
C:\> reg add "HKLM\SYSTEM\CurrentControlSet\Control\Terminal Server" /f /v fDenyTSConnec
```

Only send NTLMv2 responses to LM & NTLM: (default in Windows 7)

Limit anonymous access:

```
C:\> reg add HKLM\SYSTEM\CurrentControlSet\Control\Lsa\ /v lmcompatibilitylevel /t REG_D
```

```
C:\> reg add HKLM\SYSTEM\CurrentControlSet\Control\Lsa /v restrictanonymous /t REG_DWORD
```

Do not allow anonymous access to SAM accounts and shares:

```
C:\> reg add HKLM\SYSTEM\CurrentControlSet\Control\Lsa /v restrictanonymoussam /t REG_DW
```

Disable IPV6:

```
C:\> reg add HKLM\SYSTEM\CurrentControlSet\services\TCPIP6\Parameters /v DisabledCompone
```

Disable sticky keys:

```
C:\> reg add "HKCU\Control Panel\Accessibility\StickyKeys" /v Flags /t REG_SZ /d 506 /f
```

Disable toggle keys:

```
C:\> reg add "HKCU\Control Panel\Accessibility\ToggleKeys" /v Flags /t REG_SZ /d 58 /f
```

Disable filter keys:

```
C:\> reg add "HKCU\Control Panel\Accessibility\Keyboard Response" /v Flags /t REG_SZ /d
```

Disable On-screen Keyboard:

```
C:\> reg add HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Authentication\LogonUI /f /v
```

Disable Administrative Shares - Workstations:

```
C:\> reg add HKLM\SYSTEM\CurrentControlSet\Services\LanmanServer\Parameters /f /v AutoSh
```

Disable Administrative Shares - Servers:

```
C:\> reg add HKLM\SYSTEM\CurrentControlSet\Services\LanmanServer\Parameters /f /v AutoSh
```

Delete hashes related to the Pass the Hash attack (requires reboot and password change for old hashes):

```
C:\> reg add HKLM\SYSTEM\CurrentControlSet\Control\Lsa /f /v NoLMHash /t REG_DWORD /d 1
```

Disable Registry Editing: (High Risk)

```
C:\> reg add HKCU\Software\Microsoft\Windows\CurrentVersion\Policies\System /v DisableRe
```

Disable IE Password Cache:

```
C:\> reg add HKCU\Software\Microsoft\Windows\CurrentVersion\Internet Settings /v Disable
```

Disable CMD prompt:

```
C:\> reg add HKCU\Software\Policies\Microsoft\Windows\System /v DisableCMD /t REG_DWORD
```

Disable caching of admin credentials in the host using rdp:

```
C:\> reg add HKLM\System\CurrentControlSet\Control\Lsa /v DisableRestrictedAdmin /t REG_
```

Do not process files that have only been run once:

```
C:\> reg add HKLM\Software\Microsoft\Windows\CurrentVersion\Policies\Explorer /v Disable
C:\> reg add HKCU\Software\Microsoft\Windows\CurrentVersion\Policies\Explorer /v Disable
```

Require User Access Control (UAC):

```
C:\> reg add HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\System /v EnableLU
```

Change the password after logging in again:

```
PS C:\> Set-ADAccountPassword <USER> -NewPassword $newpwd -Reset -PassThru | Set-ADuser
```

PowerShell Script for Windows

Change the password on the next login for the OU Group:

```
PS C:\> Get-ADuser -filter "department -eq '<OU GROUP>' -AND enabled -eq 'True'" | Set-A
```

Enable logging in the firewall:

```
PS C:\> netsh firewall set logging droppedpackets=enable connections=enable
```

Bash Script for Linux

Service Information, List, Start, and Stop services in Ubuntu, and List All Services:

```
Service Information:
service --status-all
ps -ef
ps -aux
# List, Start, and Stop services in Ubuntu:
/etc/init.d/apache2 start
/etc/init.d/apache2 restart
/etc/init.d/apache2 stop # (stops only until reboot)
service mysql start
service mysql restart
service mysql stop # (stops only until reboot)
# List All Boot Up services:
ls /etc/init/*.conf
# Check Boot Up service status:
status ssh
```

Example Firewall (iptables) Commands:

```
Save All Existing iptables Rules:
iptables-save > firewall.out
# Edit File Containing Rules:
vi firewall.out
# Reload iptables Rules:
iptables-restore < firewall.out
# Example iptables Commands to Limit IPs and Ports:
iptables -A INPUT -s 10.10.10.10 -j DROP
iptables -A INPUT -s 10.10.10.0/24 -j DROP
iptables -A INPUT -p tcp --dport ssh -s 10.10.10.10 -j DROP
iptables -A INPUT -p tcp --dport ssh -j DROP
# Block All Connections:
iptables -P INPUT DROP
iptables -P OUTPUT DROP
iptables -P FORWARD DROP
# Logging All Denied Rules in iptables:
iptables -I INPUT 5 -m limit --limit 5/min -j LOG --log-prefix "iptables denied: " --log
```

Example Password Commands:

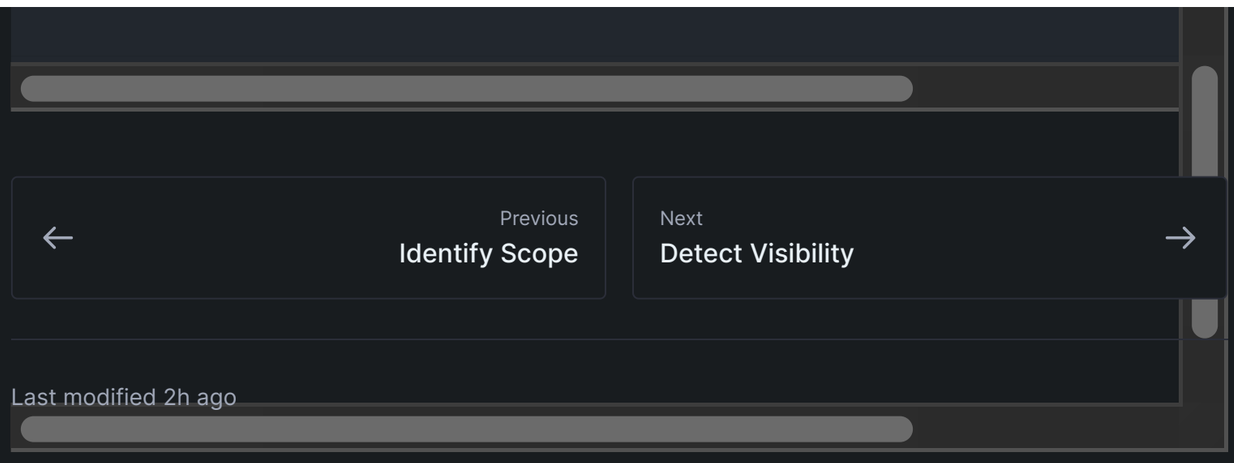
```
Change Password:
passwd # (For current user)
passwd bob # (For user Bob)
sudo su passwd # (For root)
```

Example Host File Commands:

```
Add Malicious Domain and Redirect to localhost:
echo "127.0.0.1 <MALICIOUS DOMAIN>" >> /etc/hosts
# Check Host Files by Pinging 127.0.0.1:
ping -c 1 <MALICIOUS DOMAIN>
# Restart DNS cache in Ubuntu:
/etc/init.d/dns-clean start
```

Example IPSEC Commands:

```
Allow Firewall for IPSEC Traffic:
iptables -A INPUT -p esp -j ACCEPT
iptables -A INPUT -p ah -j ACCEPT
iptables -A INPUT -p udp --dport 500 -j ACCEPT
iptables -A INPUT -p udp --dport 4500 -j ACCEPT
# IPSEC Traffic Pass Setup using Racoon:
# Step 1: Install Racoon on <HOST1 IP ADDRESS> and <HOST2 IP ADDRESS> to enable IPSE
```


Previous
Identify Scope
Next
Detect Visibility
Last modified 2h ago

Detect Visibility

Network Monitoring

TCPDUMP command

Display traffic in ASCII (-A) or HEX (-X):

```
# tcpdump -A
# tcpdump -X
```

Display traffic timestamps and avoid address conversion and be verbose:

```
# tcpdump -tttt -n -vv
```

Identify senders after receiving 1000 packets (possible DDoS attack):

```
# tcpdump -nn -c 1000 | awk '{print $3}' | cut -d. -f1-4 | sort -n | uniq -c | sort -nr
```

Capture all exchanged packets on all host interfaces and port 80, and save them to a file:

```
# tcpdump -w <FILENAME>.pcap -i any dst <TARGET_IP_ADDRESS> and port 80
```

Display traffic between two hosts:

```
# tcpdump host 10.0.0.1 and host 10.0.0.2
```

Display all traffic except for a specified network and host range:

```
# tcpdump not net 10.10.0.0/16 and not host 192.168.1.2
```

Display traffic between Host 1 and other hosts:

```
# tcpdump host 10.10.10.10 and \(host 10.10.10.20 or host 10.10.10.30\)
```

Save a pcap file with a specified size:

```
# tcpdump -n -s65535 -C 1000 -w '%host_%Y-%m-%d_%H:%M:%S.pcap'
```

Save a pcap file on another system:

```
# tcpdump -w - | ssh <REMOTE_HOST_ADDRESS> -p 50005 "cat - > /tmp/remotecapture.pcap"
```

Examine and search traffic for the word 'pass':

```
# tcpdump -n -A -s0 | grep pass
```

Examine and search traffic for clear text protocols:

```
# tcpdump -n -A -s0 port http or port ftp or port smtp or port imap or port pop3 | egrep
```

Check power or throughput:

```
# tcpdump -w - | pv -bert >/dev/null
```

Filter ipv6 traffic:

```
# tcpdump not ip6
```

Filter ipv4 traffic:

```
# tcpdump ip
```

Script to save traffic from multiple interfaces to a file in a timely manner:

```
#!/bin/bash
tcpdump -pni any -s65535 -G 3600 -w 'any%Y-%m-%d_%H:%M:%S.pcap'
```

Script for transferring tcpdump traffic files to other locations:

```
#!/bin/bash
while true; do
  sleep 1;
  rsync -azvr --progress <USER_NAME>@<IP_ADDRESS>:<TRAFFIC_DIRECTORY>/ <DESTINATION_DIR
done
```

Search for self-signed and suspicious certificates:

```
# tcpdump -s 1500 -A '(tcp[((tcp[12:1] & 0xf0) >> 2)+5:1] = 0x01) and (tcp[((tcp[12:1] &
```

Display SSL Certificates:

```
# openssl s_client -connect <URL>:443
# openssl s_client -connect <SITE>:443 </dev/null 2>/dev/null | sed -ne '/-BEGIN CERTIFI
```

Check Self-Signed Certificates:

```
# openssl x509 -text -in <CERT>.pem
# openssl x509 -in <CERT>.pem -noout -issuer -subject -startdate -enddate -fingerprint
# openssl verify <CERT>.pem
```

Extract server name in certificates:

```
# tshark -nr <PCAP FILE NAME> -Y "ssl.handshake.ciphersuites" -Vx | grep "Server Name:"
```

Extract information about certificates:

```
# ssldump -Nr <FILE NAME>.pcap | awk 'BEGIN {c=0;} { if ($0 ~ /Certificate$/) {c=1; prin
```

Check the status of applications and each port usage:

```
netstat -aon | findstr '[port_number]'
tasklist | findstr '[PID]'
tasklist | findstr '[application_name]'
netstat -aon | findstr '[PID]'
```

TSHARK Command Get network interfaces:

```
tshark -D
```

Check several network interfaces:

```
tshark -i eth1 -i eth2 -i eth3
```

Save pcap and disable name resolution:

```
tshark -nn -w <FILE NAME>.pcap
```

... and more commands follow in similar fashion.

Extract POST request values

```
tshark -Y "http.request.method==POST" -T fields -e http.file_data -r keeptryin.pcap
```

Extract DNS response values

```
codetshark -Y "dns.txt" -T fields -e dns.qry.name -n -r keeptryin.pcap
```

SNORT Command Run a test on the snort settings file:

```
# snort -T -c /<PATH TO SNORT>/snort/snort.conf
```

Tools to inspect network traffic or PCAP files

EDITCAP tool Edit pcap files (separate 1000 packets):

```
editcap -F pcap -c 1000 original.pcap out_split.pcap
```

Edit pcap files (separate packets per hour):

```
editcap -F pcap -t+3600 original.pcap out_split.pcap
```

MERGECAP tool To merge several pcap files:

```
mergecap -w merged_cap.pcap cap1.pcap cap2.pcap cap3.pcap
```

Technique: HONEY

Windows

Honey Ports on Windows:

Source: http://securityweekly.com/wp-content/uploads/2013/06/howtogetabetterpentest.pdf

Step 1: Create a firewall rule to identify and deny all connections to port 3333.

```
echo @echo off for /L %%i in (1,1,1) do @for /f "tokens=3" %%j in ('netstat -nao | find
```

Step 2: Execute the batch script.

```
<BATCH_FILE_NAME>.bat
```

... (additional steps for honey hashes and detection methods with PowerShell and batch script)...

Linux

Honey Ports on Linux:

Source: http://securityweekly.com/wp-content/uploads/2013/06/howtogetabetterpentest.pdf

Step 1: Create a loop to reject all requests to port 2222.

```
while [ 1 ]; do IP=$(nc -v -l -p 2222 2>&1 | grep from | cut -d[ -f 3 | cut -d] -f 1); 
```

Honey Port Script on Linux:

Source: https://github.com/gchetrick/honeyports/blob/master/honeyports-0.5.py

Step 1: Download the Python script.

```
wget https://github.com/gchetrick/honeyports/blob/master/honeyports-0.5.py
```

Step 2: Execute the Python script.

```
python honeyports-0.5.py -p <CHOOSE_AN_OPEN_PORT> -h <HOST_IP_ADDRESS>
```

... (additional steps for using netcat, passive DNS monitoring, and log auditing)...

LOG AUDITING METHODS

Windows

Increase Log Size for Better Auditing:

```
reg add HKLM\Software\Policies\Microsoft\Windows\EventLog\Application /v MaxSize /t REG_
reg add HKLM\Software\Policies\Microsoft\Windows\EventLog\Security /v MaxSize /t REG_DW
reg add HKLM\Software\Policies\Microsoft\Windows\EventLog\System /v MaxSize /t REG_DWORD
```

Check Security Log Settings:

```
wevtutil gl Security
```

For Audit Policy Settings:

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

Set Log Auditing (successful or unsuccessful) in All Categories:

```
C: \> auditpol /set /subcategory: "Detailed File
Share" /success:enable /failure:enable
C:\> auditpol /set /subcategory:"File System"
/success:enable /failure:enable
C:\> auditpol /set /subcategory:"Security System
Extension" /success:enable /failure:enable
C:\> auditpol /set /subcategory:"System Integrity"
/success:enable /failure:enable
C:\> auditpol /set /subcategory:"Security State
```

```
Change" /success:enable /failure:enable
C:\> auditpol /set /subcategory:"Other System
Events" /success:enable /failure:enable
C:\> auditpol /set /subcategory:"System Integrity"
/success:enable /failure:enable
C:\> auditpol /set /subcategory:"Logon"
/success:enable /failure:enable
C:\> auditpol /set /subcategory:"Logoff"
/success:enable /failure:enable
C:\> auditpol /set /subcategory:"Account Lockout"
/success:enable /failure:enable
C:\> auditpol /set /subcategory:"Other Logon/Logoff
Events" /success:enable /failure:enable
C:\> auditpol /set /subcategory:"Network Policy
Server" /success:enable /failure:enable
C:\> auditpol /set /subcategory:"Registry"
/success:enable /failure:enable
C:\> auditpol /set /subcategory:"SAM"
/success:enable /failure:enable
C:\> auditpol /set /subcategory:"Certification
Services" /success:enable /failure:enable
C:\> auditpol /set /subcategory:"Application
Generated" /success:enable /failure:enable
C: \> auditpol / set /subcategory: "Handle
Manipulation" /success:enable /failure:enable
C:\> auditpol /set /subcategory:"file Share"
/success:enable /failure:enable
C:\> auditpol /set /subcategory:"filtering Platform
Packet Drop" /success:enable /failure:enable
C:\> auditpol /set /subcategory:"Filtering Platform
Connection" /success:enable /failure:enable
C:\> auditpol /set /subcategory:"Other Object Access
Events" /success:enable /failure:enable
C: \> auditpol /set /subcategory: "Detailed File
Share" /success:enable /failure:enable
C: \> auditpol /set /subcategory: "Sensitive Privilege
Use" /success:enable /failure:enable
C: \> auditpol /set /subcategory: "Non Sensitive
Privilege Use" /success:enable /failure:enable
C: \> auditpol /set /subcategory: "Other Privilege Use
Events" /success:enable /failure:enable
C:\> auditpol /set /subcategory:"Process
Termination" /success:enable /failure:enable
C:\> auditpol /set /subcategory: "DPAPI Activity"
/success:enable /failure:enable
C: \> audit pol /set /subcategory: "RPC Events"
/success:enable /failure:enable
C:\> auditpol /set /subcategory:"Process Creation"
/success:enable /failure:enable
C:\> auditpol /set /subcategory:"Audit Policy
Change" /success:enable /failure:enable
C:\> auditpol /set /subcategory: "Authentication
Policy Change" /success:enable /failure:enable
C:\> auditpol /set /subcategory: "Authorization
Policy Change" /success:enable /failure:enable
C: \> audit pol /set /subcategory: "MPSSVC Rule-Level
Policy Change" /success:enable /failure:enable
C:\> auditpol /set /subcategory:"Filtering Platform
Policy Change" /success:enable /failure:enable
C:\> auditpol /set /subcategory:"Other Policy Change
Events" /success:enable /failure:enable
C:\> auditpol /set /subcategory:"User Account
Management" /success:enable /failure:enable
C:\> auditpol /set /subcategory:"Computer Account
Management" /success:enable /failure:enable
C:\> auditpol /set /subcategory:"Security Group
Management" /success:enable /failure:enable
C:\> auditpol /set /subcategory:"Distribution Group
Management" /success:enable /failure:enable
C:\> auditpol /set /subcategory:"Application Group
Management" /success:enable /failure:enable
C:\> auditpol /set /subcategory:"Other Account
Management Events" /success:enable /failure:enable
C:\> auditpol /set /subcategory:"Directory Service
Changes" /success:enable /failure:enable
C: \> auditpol / set /subcategory: "Directory Service
Replication" /success:enable /failure:enable
C:\> auditpol /set /subcategory:"Detailed Directory
Service Replication" /success:enable /failure:enable
C:\> auditpol /set /subcategory:"Directory Service
Access" /success:enable /failure:enable
C:\> auditpol /set /subcategory:"Kerberos Service
Ticket Operations" /success:enable /failure:enable
C:\> auditpol /set /subcategory:"Other Account Logan
Events" /success:enable /failure:enable
C: \> audit pol /set /subcategory: "Kerberos
Authentication Service" /success:enable
/failure:enable
C:\> auditpol /set /subcategory:"Credential
Validation" /success:enable /failure:enable
```

Available Reports List and Sizes and Allowed:

Available reports list and their sizes and permitted:

```
PS C:\> Get-Eventlog -list
```

Partial List of Security Log Auditing Events Keys:
Partial list of keys for monitoring Security Log Auditing events:

```
PS C:\> Get-Eventlog -newest 5 -logname application | Format-List
```

Display Reports Remotely:

Displaying reports remotely:

```
PS C:\> Show-Eventlog -computername <SERVER NAME>
```

Display Event List Based on Event ID:

Displaying the list of events based on Event ID:

```
PS C:\> Get-Eventlog Security | Where-Object { $_.Eventid -eq 4800}
PS C:\> Get-WinEvent -FilterHashtable @{LogName="Security"; ID=4774}
```

Account Access - Audit Credential Validation for the Last 14 Days:

Logging in - Audit Credential Validation for the last 14 days:

```
PS C:\> Get-Eventlog Security -InstanceId 4768,4771,4772,4769,4770,4649,4778,4779,4800,
```

Account - Login and Logout:

Account - Logins and logouts:

```
PS C:\> Get-Eventlog Security -InstanceId 4625,4634,4647,4624,4625,4648,4675,6272,6273,
```

Account Management - Audit Group Management Programs:

Account management - Managing the group of Audit apps:

```
PS C:\> Get-Eventlog Security -InstanceId 4783,4784,4785,4786,4787,4788,4789,4790,4741,
```

Display Available Event Logs and their Sizes and Quota:

```
PS C:\> Get-Eventlog -list
```

Partial List of Security Log Auditing Events Key Monitoring:

```
PS C:\> Get-Eventlog -newest 5 -logname application | Format-List
```

Display Logs Remotely:

```
PS C:\> Show-Eventlog -computername <SERVER NAME>
```

Display Event List Based on Event ID:

```
PS C:\> Get-Eventlog Security | Where-Object { $_.Eventid -eq 4800}
PS C:\> Get-WinEvent -FilterHashtable @{LogName="Security"; ID=4774}
```

Account Login - Audit Credential Validation for the Last 14 Days:

```
PS C:\> Get-Eventlog Security -InstanceId 4768,4771,4772,4769,4770,4649,4778,4779,4800,
```

Account - Login and Logout:

```
PS C:\> Get-Eventlog Security -InstanceId 4625,4634,4647,4624,4625,4648,4675,6272,6273,
```

Account Management - Audit Group Management Program:

```
PS C:\> Get-Eventlog Security -InstanceId 4783,4784,4785,4786,4787,4788,4789,4790,4741,
```

Fine Tracking - Audit DPAPI Activity, Process Termination, RPC Events:

```
PS C:\> Get-EventLog Security -InstanceId 4692,4693,4694,4695,4689,5712 -after ((get-da
```

Domain Service Access - Audit Access to Directory Service:

```
PS C:\> Get-EventLog Security -InstanceId 4662,5136,5137,5138,5139,5141 -after ((get-da
```

Object Access - Audit File Share, File System, SAM, Registry, Certificates:

```
PS C:\> Get-EventLog Security -InstanceId 4671,4691,4698,4699,4700,4701,4702,5148,5149,5
```

Policy Change - Audit Policy Change, Microsoft Protection Service, Windows Filtering Platform:

```
PS C:\> Get-EventLog Security -InstanceId 4715,4719,4817,4902,4904,4905,4906,4907,4908,4
```

Privilege Use - Audit Sensitive and Non-sensitive Service Privilege Use:

```
PS C:\> Get-EventLog Security -InstanceId 4672,4673,4674 -after ((get-date).addDays(-1))
```

System - Audit Security State Change, Security System Extension, System Integrity, System Events:

```
PS C:\> Get-Eventlog Security -InstanceId 5024,5025,5027,5028,5029,5030,5032,5033,5034,5
```

Add Microsoft IIS Module:

```
PS C:\> add-pssnapin WebAdministration
PS C:\> Import-Module WebAdministration
```

Get Information about IIS:

```
PS C:\> Get-IISSite
```

Get IIS Path Information:

```
PS C:\> (Get-WebConfigurationProperty '/system.applicationHost/sites/siteDefaults' -Name
```

List All Installed Software:

```
PS C:\> Get-WmiObject -Query "SELECT * FROM Win32_Product" | Select-Object Name
```

List Installed Software on Remote Computer:

```
PS C:\> Get-WmiObject -Query "SELECT * FROM Win32_Product" -ComputerName <RemoteComputer
```

Delete/Uninstall Software:

```
PS C:\> Get-WmiObject -Query "SELECT * FROM Win32_Product WHERE Name = '<SoftwareName>'"
```

Query Users Connected to a Domain Controller:

```
PS C:\> Get-WmiObject -Class Win32_ComputerSystem -Property UserName
```

Find Locked Out Accounts:

```
PS C:\> Search-ADAccount -LockedOut | Select-Object UserPrincipalName
```

- Note: Ensure you have the Active Directory module loaded (`Import-Module ActiveDirectory`) before executing.

Unlock User Account:

```
PS C:\> Unlock-ADAccount -Identity <UserName>
```

Check Service Status:

```
PS C:\> Get-Service -Name <ServiceName> | Select-Object Status, Name, DisplayName
```

Start a Service:

```
PS C:\> Start-Service -Name <ServiceName>
```

Stop a Service:

```
PS C:\> Stop-Service -Name <ServiceName>
```

Check Disk Space:

```
```

List All Running Processes:

```
PS C:\> Get-Process | Select-Object ProcessName, Id
```

Kill a Process:

```
PS C:\> Stop-Process -Id <ProcessId>
- or -
PS C:\> Stop-Process -Name <ProcessName>
```

Get All Available Network Adapters:

```
PS C:\> Get-NetAdapter | Select-Object Name, Status, MacAddress
```

Enable Network Adapter:

```
PS C:\> Enable-NetAdapter -Name <AdapterName>
```

Disable Network Adapter:

```
PS C:\> Disable-NetAdapter -Name <AdapterName> -Confirm:$false
```

Get IP Configuration:

```
PS C:\> Get-NetIPConfiguration | Select-Object InterfaceAlias, IPv4Address
```

Set Static IP Address:

```
PS C:\> New-NetIPAddress -InterfaceAlias <AdapterName> -IPAddress <IPAddress> -PrefixLer
```

Set DNS Servers:

```
PS C:\> Set-DnsClientServerAddress -InterfaceAlias <AdapterName> -ServerAddresses <DNSSe
```

Create a New Folder:

```
PS C:\> New-Item -Path <Path> -Name <FolderName> -ItemType Directory
```

Copy a Folder/File:

```
PS C:\> Copy-Item -Path <SourcePath> -Destination <DestinationPath>
```

Move a Folder/File:

```
PS C:\> Move-Item -Path <SourcePath> -Destination <DestinationPath>
```

Delete a Folder/File:

```
PS C:\> Remove-Item -Path <Path> -Recurse -Force
```

Extract a Zip File:

```
PS C:\> Expand-Archive -Path <PathToZip> -DestinationPath <ExtractPath>
```

Compress Files into a Zip:

```
PS C:\> Compress-Archive -Path <PathToFiles> -DestinationPath <PathToZip>
```

Get System Uptime:

```
PS C:\> (Get-Date) - (Get-CimInstance Win32_OperatingSystem).LastBootUpTime
```

Check Memory Usage:

```
PS C:\> Get-WmiObject Win32_OperatingSystem | Select-Object @{Name="FreeMemory(GB)";Expr
```

View Event Logs:

```
PS C:\> Get-EventLog -LogName <LogName> -Newest <NumberOfEvents>
```

Send an Email:

*Note: Use `Get-Credential` to provide username and password for the SMTP server.

Schedule a Task:

```
PS C:\> $Action = New-ScheduledTaskAction -Execute '<PathToExecutable>'
PS C:\> $Trigger = New-ScheduledTaskTrigger -At <StartTime> -RepetitionInterval <Interva
PS C:\> Register-ScheduledTask -Action $Action -Trigger $Trigger -User "<Username>" -Pas
```

Import a CSV File:

```
PS C:\> $Data = Import-Csv -Path <PathToCsv>
```

Export Data to a CSV File:

```
PS C:\> $Data | Export-Csv -Path <PathToCsv> -NoTypeInformation
```

Get a List of User Profiles:

```
PS C:\> Get-WmiObject Win32_UserProfile | Select-Object Special, LocalPath
```

Remove a User Profile:

```
PS C:\> Get-WmiObject Win32_UserProfile | Where-Object { $_.Special -eq $false and $_.Lo
```

Check Firewall Status:

```
PS C:\> Get-NetFirewallProfile | Select-Object Name, Enabled
```


← Previous
Protect Defend

Next
Respond Analysis →

Last modified 2h ago

Respond Analysis

Analysis: LIVE TRIAGE - Windows

System Information:

```
echo %DATE% %TIME%
hostname
systeminfo
systeminfo | findstr /B /C:"OS Name" /C:"OS Version"
wmic csproduct get name
wmic bios get serialnumber
wmic computersystem list brief
```

These commands are used to retrieve information about the system, including the date, time, hostname, detailed system info, operating system name and version, product name, BIOS serial number, and a brief list of computer systems.

Source: https://technet.microsoft.com/en-us/sysinternals/psinfo.aspx

```
psinfo -accepteula -s -h -d
```

This command retrieves detailed system information using the `psinfo` tool from Sysinternals.

User Information:

```
whoami
net users
net localgroup administrators
net group administrators
wmic rdtoggle list
wmic useraccount list
wmic group list
wmic netlogin get name, lastlogon, badpasswordcount
wmic netclient list brief
doskey /history > history.txt
```

These commands gather information regarding the user, like the current user, all user accounts, group and local group administrators, remote desktop settings, and user account details. It also retrieves command line history and saves it to a text file.

Network Information:

```
netstat -e
netstat -naob
netstat -nr
netstat -vb
nbtstat -s
route print
arp -a
ipconfig /displaydns
netsh winhttp show proxy
ipconfig /allcompartments /all
netsh wlan show interfaces
netsh wlan show all
reg query "HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Internet Settings\Connections\
type %SYSTEMROOT%\system32\drivers\etc\hosts
wmic nicconfig get descriptions, IPaddress, MACaddress
wmic netuse get name, username, connectiontype, localname
```

These commands collect various network-related information like network statistics, active connections, routing tables, ARP tables, DNS cache content, proxy settings, interface configurations, and more.

Service Information:

```
at
tasklist
tasklist /SVC
tasklist /SVC /fi "imagename eq svchost.exe"
schtasks
net start
sc query
wmic service list brief | findstr "Running"
wmic service list config
wmic process list brief
wmic process list status
wmic process list memory
wmic job list brief
```

PowerShell commands for service information:

```
Get-Service | Where-Object { $_.Status -eq "running" }
Get-Process | Select-Object Modules | ForEach-Object { $_.Modules }
```

These commands display information related to system tasks, services, and processes that are running, including service configuration and memory usage of processes.

Policy, Patch, and Settings Information:

```
set
gpresult /r
gpresult /z > [OUTPUT FILE NAME].txt
gpresult /H report.html /F
wmic qfe
```

For listing GPO-installed software:

```
reg query "HKLM\Software\Microsoft\Windows\CurrentVersion\Group Policy\AppMgmt"
```

These commands are used to display the environment variable, group policy results, and Quick Fix Engineering (update patches) information.

Autorun and Autoload Information:

```
wmic startup list full
wmic ntdomain list brief
```

Commands to display the content of the startup service path:

```
dir "%SystemDrive%\ProgramData\Microsoft\Windows\Start Menu\Programs\Startup"
dir "%SystemDrive%\Documents and Settings\All Users\Start Menu\Programs\Startup"
dir %userprofile%\Start Menu\Programs\Startup
dir %ProgramFiles%\Startup\
dir C:\Windows\Start Menu\Programs\startup
dir "C:\Users\%username%\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\Startup"
dir "C:\ProgramData\Microsoft\Windows\Start Menu\Programs\Startup"
dir "%APPDATA%\Microsoft\Windows\Start Menu\Programs\Startup"
dir "%ALLUSERSPROFILE%\Microsoft\Windows\Start Menu\Programs\Startup"
dir "%ALLUSERSPROFILE%\Start Menu\Programs\Startup"
type C:\Windows\winstart.bat
type %windir%\wininit.ini
type %windir%\win.ini
```

Showing Microsoft autorun and hidden files **Source:** https://technet.microsoft.com/en-us/sysinternals/bb963902.aspx

```
autorunsc -accepteula -m
type C:\Autoexec.bat
```

Commands to display and save all autorun files in CSV and check them using virustotal:

```
autorunsc.exe -accepteula -a -c -i -e -f -l -m -v
```

Commands querying registry entries:

```
reg query HKCR\Comfile\Shell\Open\Command
reg query HKCR\Batfile\Shell\Open\Command
reg query HKCR\htafile\Shell\Open\Command
reg query HKCR\Exefile\Shell\Open\Command
reg query HKCR\Exefiles\Shell\Open\Command
reg query HKCR\piffile\shell\open\command
```

HKEY_CURRENT_USERS:

Commands for querying various registry keys under the HKEY_CURRENT_USER hive:

```
C:\> reg query HKCU\Control Panel\Desktop
C:\> reg query HKCU\Software\Microsoft\Windows\CurrentVersion\Policies\Explorer\Run
C:\> reg query HKCU\Software\Microsoft\Windows\CurrentVersion\Run
C:\> reg query HKCU\Software\Microsoft\Windows\CurrentVersion\Runonce
C:\> reg query HKCU\Software\Microsoft\Windows\CurrentVersion\RunOnceEx
C:\> reg query HKCU\Software\Microsoft\Windows\CurrentVersion\RunServices
C:\> reg query HKCU\Software\Microsoft\Windows\CurrentVersion\RunServicesOnce
C:\> reg query HKCU\Software\Microsoft\Windows\CurrentVersion\Windows\Run
C:\> reg query HKCU\Software\Microsoft\Windows\CurrentVersion\Windows\Load
... (and so on)
```

HKEY_LOCAL_MACHINE:

Commands querying registry keys under the HKEY_LOCAL_MACHINE hive:

LOGS

Commands related to working with event logs, including exporting logs:

```
C:\> wevtutil epl Security C:\<BACK UP PATH>\mylogs.evtx
C:\> wevtutil epl System C:\<BACK UP PATH>\mylogs.evtx
C:\> wevtutil epl Application C:\<BACK UP PATH>\mylogs.evtx
```

Alternate Data Streams:

https://technet.microsoft.com/en-us/sysinternals/streams.aspx

```
C:\> streams -s <FILE OR DIRECTORY>
```

check malicious file and save in csv

https://technet.microsoft.com/en us/sysinternals/bb897441.aspx .منبع

```
C:\> sigcheck -c -h -s -u -nobanner <FILE OR
DIRECTORY> > <OUTPUT FILENAME>,csv
```

check malicious file

```
C:\> sigcheck -e -u -vr -s C:\
```

DLL Unassigned

https://technet.microsoft.com/en-us/sysinternals/bb896656.aspx

```
C:\> listdlls.exe -u
C:\> listdlls.exe -u <PROCESS NAME OR PID>
```

Windows Defender

http://windows.microsoft.com/en-us/windows/what-is-windows-defender-offline .منبع

```
C:\> MpCmdRun.exe -SignatureUpdate
C:\> MpCmdRun.exe -Scan
```

LIVE TRIAGE - Linux

System Information

```
# uname -a
# uptime
# timedatectl
# mount
```

User Information

- List of users who have logged in:

```
# w
```

- List of users who have logged in remotely:

```
# lastlog
# last
```

- Show unsuccessful logins:

```
# faillog -a
```

- Display local users:

```
# cat /etc/passwd
# cat /etc/shadow
```

- Display local groups:

```
# cat /etc/group
```

- Display sudo access:

```
# cat /etc/sudoers
```

- Display users with UID 0:

```
# awk -F: '($3 == "0") {print}' /etc/passwd
```

```
# egrep ':0+' /etc/passwd
```

- List of valid ssh authentication keys:

```
# cat /root/.ssh/authorized_keys
```

- List files opened by the user:

```
# lsof -u <USER NAME>
```

- Display bash history:

```
# cat /root/.bash_history
```

Network Information

- Display network interfaces:

```
# ifconfig
```

- Display network connections:

```
# netstat -antup
# netstat -plantux
```

- Display listening ports:

```
# netstat -nap
```

- Display routes:

```
# route
```

- Display the ARP table:

```
# arp -a
```

- Display processes and used ports list:

```
# lsof -i
```

Service Information

- List of processes:

```
# ps -aux
```

- List of loaded modules:

```
# lsmod
```

- List of open files:

```
# lsof
```

- List of network-open files:

```
# lsof -nPi | cut -f 1 -d " " | uniq | tail -n +2
```

- List of files opened by a specific process:

```
# lsof -c <SERVICE NAME>
```

- List of all files opened by a specific process:

```
# lsof -p <PID>
```

- List of unlinked processes' keys in execution:

```
# lsof +Ll
```

- Processes of a PID:

```
# ls -al /proc/<PID>/exe
```

- Storing analysis of executable files of malware:

```
# cp /proc/<PID>/exe /<SUSPICIOUS FILE NAME TO SAVE>.elf
```

- Live reports display:

```
# less +F /var/log/messages
```

- List of services:

```
# chkconfig --list
```

Policy, Patch, and Settings Information

- Display files within the pam.d path:

```
# cat /etc/pam.d/common*
```

Autorun and Autoload Information

- List of cron jobs:

```
# crontab -l
```

- List of cron jobs run by root user and UID zero:

```
# crontab -u root -l
```

- Check unusual cron jobs:

```
# cat /etc/crontab
# ls /etc/cron.*
```

Reports

- Check history of executed commands by root user:

```
# cat /root/.*history
```

- Check the last user logged into the system:

```
# last
```

Files, Drivers, and Shared Environment Information

- Display disk usage:

```
# df -ah
```

- Display files in /etc/init.d path:

```
# ls -la /etc/init.d
```

- More information about a file:

```
# stat -x <FILE NAME>
```

- Determine file type:

```
# file <FILE NAME>
```

- Display immutable files:

```
# lsattr -R / | grep -i "-"
```

- List files in /root path:

- Display a list of recently modified files:

```
# ls -alt | head
```

- List writable files:

```
# find / -xdev -type d \( -perm -0002 -a ! -perm -1000 \) -print
```

- List files created since Jan 02, 2017:

```
# find / -newermt 2017-01-02
```

- List all files and their attributes:

```
# find / -printf "%m;%Ax;%AT;%Tx;%TT;%Cx;%CT;%U;%G;%s;%p\n"
```

- List files in a specific path that have a newer timestamp (might be manipulated):

```
# ls -alt /<DIRECTORY> | head
```

- Display file details:

```
# stat /<FILE PATH>/<SUSPICIOUS FILE NAME>
```

- Check file type:

```
# file /<FILE PATH>/<SUSPICIOUS FILE NAME>
```

Run unix-privsec-check tool:

```
# wget https://raw.githubusercontent.com/pentestmonkey/unix-privesc-check/1_x/unix-prive
# ./unix-privesc-check > output.txt
```

Execute chkrootkit:

```
# apt-get install chkrootkit
# chkrootkit
```

Execute rkhunter:

```
# apt-get install rkhunter
# rkhunter --update
# rkhunter --check
```

Execute tiger:

```
# apt-get install tiger
# tiger
# less /var/log/tiger/security.report.*
```

Execute lynis:

```
# apt-get install lynis
# lynis audit system
# more /var/logs/lynis.log
```

Execute Linux Malware Detect (LMD):

```
bashCopy code# wget http://www.rfxn.com/downloads/maldetect-current.tar.gz
# tar xfz maldetect-current.tar.gz
# cd maldetect-*
# ./install.sh
```

Get LMD updates:

```
# maldet -u
```

Run and scan LMD on a specific path:

```
# maldet -a /<DIRECTORY>
```

USB Examination:

Displaying Events using usbrip:

```
usbrip events violations auth.json
```

Git Analysis:

Display history:

```
git log
```

Display commit contents:

```
git checkout <commit> --force
```

MALWARE Analysis:

STATIC ANALYSIS:

Creating Mount live Sysinternals tools drive:

```
\\live.sysinternals.com\tools
```

Checking Signature for dlt and exe files:

Source: http://technet.microsoft.com/en-us/sysinternals/bb897441.aspx

```
C:\> sigcheck.exe -u -e (:\<DIRECTORY>
C:\> sigcheck.exe -vt <SUSPICIOUS FILE NAME>
```

Shell Codes Analysis:

Read More:

Windows PE Analysis:

Display Hex and ASCI of PE files (exe or any file), with switch -n and first 500 bytes:

```
# hexdump -C -n 500 <SUSPICIOUS FILE NAME>
# od -x somefile.exe
# xxd somefile.exe
```

Use debug tool in Windows (for .java files):

```
C:\> debug <SUSPICIOUS FILE NAME>
> -d
> -q
```

Windows PE Analysis:

Script for compile time and date of PE files (Only for Windows). Source: https://www.perl.org/get.html and http://www.perlmonks.org/bare/?node_id=484287

```
C:\> perl.exe <SCRIPT NAME>.pl <SUSPICIOUS FILE NAME>
```

Displaying strings inside PE and string lengths with switch -n:

Using strings in Linux:

```
# strings -n 10 <SUSPICIOUS FILE NAME>
```

Source: https://technet.microsoft.com/en-us/sysinternals/strings.aspx

Using strings in Windows:

```
C:\> strings <SUSPICIOUS FILE NAME>
```

Identify Malware in dumped memory using Volatility and the Windows7SPFix64 profile:

Source: https://github.com/volatilityfoundation/volatility

```
# python vol.py -f <MEMORY DUMP FILE NAME>.raw --profile=Win7SPFix64 malfind -D /<OUTPUT
# python vol.py -f <MEMORY DUMP FILE NAME>.raw --profile=Win7SPFix64 malfind -p <PID #>
# python vol.py -f <MEMORY DUMP FILE NAME>.raw --profile=Win7SPFix64 pslist
# python vol.py -f <MEMORY DUMP FILE NAME>.raw --profile=Win7SPFix64 pstree
# python vol.py -f <MEMORY DUMP FILE NAME>.raw --profile=Win7SPFix64 dlllist
# python vol.py -f <MEMORY DUMP FILE NAME>.raw --profile=Win7SPFix64 dlldump -D /<OUTPUT
```

Process memory output

```
volatility -f flounder-pc-memdump.elf --profile=<PROFILE> memdump -p <PID> -D dump
```

Malware Checking and Identification Tool:

Source: https://github.com/Defense-Cyber-Crime-Center/DC3-MWCP

Installing dc3-mwcp tool:

```
setup.py install
```

Using dc3-mwcp tool to check suspicious files:

```
mwcp-tool.py -p <SUSPICIOUS FILE NAME>
```

MALWARE IDENTIFICATION

- Tool: PROCESS EXPLORER
- Source: YouTube Video

Step 1: Process Listing and Inspecting Suspicious Items:

```
No icon items
- No description or company name items
- Unsigned Microsoft images
- Check hashes in Virus Total
- Suspicious files in Windows directories or user profile
- Purple items (packed or compressed)
- Items with open TCP/IP endpoints
```

Step 2: Checking File Signatures:

```
Use Sigcheck
```

Step 3: Examining Strings:

```
Investigate suspicious URLs in strings.
```

Step 4: Viewing DLLs:

```
Use Ctrl+D to pop open
- Inspect for suspicious DLLs or services
```

Step 5: Stopping and Removing Malware:

```
Suspend and terminate suspicious processes
```

Step 6: Remove suspicious files that run at system startup:

```
Utilize Autoruns
```

Step 7: Process Monitoring

- Source: Process Monitor

Step 8: Repeat Above Steps for Identifying Suspicious Files.

Check File Hashes

```
Using HASH QUERY
```

Utilizing VirusTotal APIs:

- Ref: VirusTotal API Documentation

```
# Submitting suspicious file hashes to virustotal using curl:
curl -v --request POST --url https://www.virustotal.com/vtapi/v2/file/report' -d apikey=

# Submitting suspicious files to virustotal using curl:
curl -v -F 'file=[PATH TO FILE]/[SUSPICIOUS FILE NAME]' -F apikey=[VT API KEY] https://v
```

Utilizing Team Cymru APIs:

- Source: Team Cymru

```
# Check malware hashes using Team Cymru and whois tool:
whois -h hash.cymru.com [SUSPICIOUS FILE HASH]
```

HARD DRIVE AND MEMORY ACQUISITION

- OS: WINDOWS
- Remotely create a memory dump:

```
C:\> psexec.exe \\<HOST NAME OR IP ADDRESS> -u <DOMAIN>\<PRIVILEGED ACCOUNT> -p <PASSWOR
```

- Extract exe and dll files from dumped memory:

```
C:\> volatility dlldump -f memory.dmp -0 dumps/
C:\> volatility procmemdump -f memory.dmp -0 dumps/
```

- OS: LINUX
- Create a memory dump:

```
dd if=/dev/fmem of=/tmp/[MEMORY FILE NAME].dd
```

Investigate Hidden Data in Files and Pictures

- Utilizing various websites and tools like dcode, StegCracker, StegExtract, Sonic Visualizer, spek, etc.

Create a memory dump using LiME tool: Source: https://github.com/504ensicslabs/lime

```
# wget
wget https://github.com/504ensicslabs/LiME/archive/master.zip
unzip master.zip
# cd LiME-master/src
cd LiME-master/src
# make
make
# cp lime-*,ko /media/=/media/ExternalUSBDriveName/
cp lime-*.ko /media/ExternalUSBDriveName/
# insmod lime-3.13.0-79-generic.ko "path=/media/ExternalUSBDriveName/<MEMORY DUMP>, lime
insmod lime-3.13.0-79-generic.ko "path=/media/ExternalUSBDriveName/<MEMORY DUMP>, lime t
```

Create a copy of a suspicious process using process ID:

```
# cp /proc/<SUSPICIOUS PROCESS ID>/exe /<NEW SAVED LOCATION>
cp /proc/<SUSPICIOUS PROCESS ID>/exe /<NEW SAVED LOCATION>
```

More information about the suspicious process in dumped memory:

```
# gcore <PID>
gcore <PID>
```

Using Strings on a file:

```
# strings gcore.*
strings gcore.*
```

Create a copy of a hard drive and partition including tags and hashes:

```
# dd if=<INPUT DEVICE> of=<IMAGE FILE NAME>
dd if=<INPUT DEVICE> of=<IMAGE FILE NAME>
# dc3dd if=/dev/<TARGET DRIVE EXAMPLE SDA OR SDA1> of=/dev/<MOUNTED LOCATION>/<FILE NAME
dc3dd if=/dev/<TARGET DRIVE EXAMPLE SDA OR SDA1> of=/dev/<MOUNTED LOCATION>/<FILE NAME>.
```

Create a hard drive and partition over SSH:

```
# dd if=/dev/<INPUT DEVICE> | ssh <USERNAME>@<DESTINATION IP ADDRESS> "dd of=<DESTINATIO
dd if=/dev/<INPUT DEVICE> | ssh <USERNAME>@<DESTINATION IP ADDRESS> "dd of=<DESTINATION
```

Send a zipped hard drive image over netcat: To send to the host:

```
# bzip2 -c /dev/<INPUT DEVICE> | nc <DESTINATION IP ADDRESS> <PICK A PORT>
bzip2 -c /dev/<INPUT DEVICE> | nc <DESTINATION IP ADDRESS> <PICK A PORT>
```

To receive by the host:

```
# nc -p <PICK SAME PORT> -l | bzip2 -d | dd of=/dev/sdb
nc -p <PICK SAME PORT> -l | bzip2 -d | dd of=/dev/sdb
```

To send to host host:

```
# dd if=/dev/<INPUT DEVICE> bs=16M | nc <PORT>
dd if=/dev/<INPUT DEVICE> bs=16M | nc <PORT>
```

To receive by the host using Pipe Viewer meter:

```
# nc -p <SAME PORT> -l -vv | pv -r | dd of=/dev/<INPUT DEVICE> bs=16M
nc -p <SAME PORT> -l -vv | pv -r | dd of=/dev/<INPUT DEVICE> bs=16M
```

Encryption websites:

- https://www.dcode.fr/

- https://gchq.github.io/CyberChef/
- https://crackstation.net/

Examining hidden data in a file with StegCracker: https://github.com/Paradoxis/StegCracker
Example:

```
stegcracker image.jpg
```

Examining hidden data in a photo with bash script StegExtract:

```
sudo curl https://raw.githubusercontent.com/evyatarmeged/stegextract/master/stegextract
sudo chmod +x /usr/local/bin/stegextract
```

Example:

```
stegextract simple.gif --analysis --string
```

Examining hidden data in a photo with StegSolve:

```
wget http://www.caesum.com/handbook/Stegsolve.jar -O stegsolve.jar
chmod +x stegsolve.jar
java -jar stegsolve.jar
```

Examining hidden data in a file with exiftool:

```
sudo apt-get install libimage-exiftool-perl
```

Example:

```
exiftool poissonrecon.pdf
```

Examining hidden data in music with Sonic Visualizer:
https://www.sonicvisualiser.org/download.html Example: In Sonic Visualizer, select: Pane → Add Spectrogram → Channel 1

Examining hidden data in music with spek:

```
apt-get install spek
```

Previous
Detect Visibility

Next
Recover Remediate

Last modified 1h ago

Recover Remediate

After the Attack

Implementation

Windows

Using a Hotfix Update for Windows 7 or above:

```
C:\> wusa.exe C:\<PATH TO HOTFIX>\Windows6.0-KB934307-x86.msu
```

Using a Hotfix Update for Windows 7 or above with a batch script:

```
@echo off
setlocal
set PATHTOFIXES=E:\hotfix
%PATHTOFIXES%\Q123456_w2k_sp4_x86.exe /2 /M
%PATHTOFIXES%\Ql23321_w2k_sp4_x86.exe /2 /M
%PATHTOFIXES%\Q123789_w2k_sp4_x86.exe /2 /M
```

Checking for Updates in Windows 7 or above:

```
C:\> wuauclt.exe /detectnow /updatenow
```

Linux

Ubuntu Distribution:

- Fetching the update list:

```
# apt-get update
```

- Upgrading current packages:

```
# apt-get upgrade
```

- Installing updates (new):

```
# apt-get dist-upgrade
```

Red Hat Enterprise Linux 2.1, 3, 4:

```
# up2date
# up2date-nox --update
# up2date <PACKAGE NAME>
# up2date -u <PACKAGE NAME>
```

Red Hat Enterprise Linux 5:

```
# pup
```

Red Hat Enterprise Linux 6:

```
# yum update
# yum list installed <PACKAGE NAME>
# yum install <PACKAGE NAME>
# yum update <PACKAGE NAME>
```

Kali Distribution:

```
# apt-get update && apt-get upgrade
```

Backup

Windows

- Backup GPO Audit Policy to a CSV file:

```
C:\> auditpol /backup /file:C\auditpolicy.csv
```

- Restore GPO Audit Policy from a CSV file:

```
C:\> auditpol /restore /file:C:\auditpolicy.csv
```

- Back up all GPOs in the domain and store them in a specified location:

```
PS C:\> Backup-Gpo -All -Path \\<SERVER>\<PATH TO BACKUPS>
```

- Restore backup GPOs in the domain from a specified location:

```
PS C:\> Restore-GPO -All -Domain <INSERT DOMAIN NAME> -Path \\Serverl\GpoBackups
```

- Start the Volume Shadow service:

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

- List all shadow files and storage:

```
C:\> vssadmin List ShadowStorage
```

- List all shadow files:

```
C:\> vssadmin List Shadows
```

- Search Shadow Copy for files and folders:

```
C:\> mklink /d c:\<CREATE FOLDER>\<PROVIDE FOLDER NAME BUT DO NOT CREATE> \\?\GLOBALROOT
```

- Jump to the selected shadow file in Windows Server and Windows 8:

```
C:\> vssadmin revert shadow /shadow={<SHADOW COPY ID>} /ForceDismount
```

- Retrieve the history of previous versions of a file with `volrest.exe`:

```
C:\> "\Program Files (x86)\Windows Resource Kits\Tools\volrest.exe" "\\localhost\c$\<PAT
```

- Jump to a selected version of a file or @GMT using `volrest.exe`:

```
C:\> subst Z: \\localhost\c$\$\<PATH TO FILE>
C:\> "\Program Files (x86)\Windows Resource Kits\Tools\volrest.exe" "\\localhost\c$\<PAT
C:\> subst Z: /0
```

- Jump to another path or sub-path using `volrest.exe`:

```
C:\> "\Program Files (x86)\Windows Resource Kits\Tools\volrest.exe" \\localhost\c$\<PATH
```

- Jump to the selected shadow file in Windows Server, Windows 7, and Windows 10 using `wmic`:

```
C:\> wmic shadowcopy call create Volume='C:\'
```

- Create a shadow copy of volume C on Windows 7 and Windows 10 using PowerShell:

```
PS C:\> (gwmi -list win32_shadowcopy).Create('C:\', 'ClientAccessible')
```

- Create a shadow copy of volume C on Windows Server 2003 and Windows Server 2008:

```
C:\> vssadmin create shadow /for=c:
```

- Create a restore point in Windows:

```
C:\> wmic.exe /Namespace:\\root\default Path SystemRestore Call CreateRestorePoint "%DAT
```

- Recover to a restore point in Windows XP:

```

```

- List recoverable points:

```
PS C:\> Get-ComputerRestorePoint
```

- Recover to a recoverable point:

```
PS C:\> Restore-Computer -RestorePoint <RESTORE POINT#> -Confirm
```

Linux

Resetting the root user password in single-user mode: Step 1: Reboot the system.

```
# reboot -f
```

Step 2: Press the ESC key to enter the GRUB page.

Step 3: Select the default entry and press the e key to edit it.

Step 4: Look for a line that begins with the words linux, linux16, or linuxefi.

Step 5: Add ‘rw init=/bin/bash’ to the end of that line.

Step 6: Press the Ctrl-X key combination to boot.

Step 7: After rebooting, you should enter single-user mode as root and be able to change your password with the following command:

```
# passwd
```

Step 8: Reboot the system again.

```
# reboot -f
```

Reinstalling Packages:

```
# apt-get install --reinstall <COMPROMISED PACKAGE NAME>
```

Reinstall all packages:

```
# apt-get install --reinstall $(dpkg --get-selections | grep -v deinstall)
```

Removing MALWARE Processes

Windows Malware Removal Tool: Source: http://www.gmer.net/

```
C:\> gmer.exe (GUI)
```

Removing a suspicious file that is running:

```
C:\> gmer.exe -killfile C:\WINDOWS\system32\drivers\<MALICIOUS FILENAME>.exe
```

Removing a suspicious running file in PowerShell:

```
PS C:\> Stop-Process -Name <PROCESS NAME>
PS C:\> Stop-Process -ID <PID>
```

Linux Terminate the malware process:

```
# kill <MALICIOUS PID>
```

Disable the malware's executability and change its path:

```
# chmod -x /usr/sbin/<SUSPICIOUS FILE NAME>
# mkdir /home/quarantine/
# mv /usr/sbin/<SUSPICIOUS FILE NAME> /home/quarantine/
```

Terminate the application using a specific port:

```
# fuser -k 80/tcp
```

Previous
Respond Analysis

Next
Tactics Tips And Tricks

Last modified 1h ago

Tactics Tips And Tricks

Operating System Cheat Sheet

Windows

- **Using Pipe for outputs and utilizing in clipboard:**

```
batchCopy codeC:\> some_command.exe | clip
```

- **Retrieving information from the clipboard and saving it to a file:** (Requires PowerShell 5)

```
PowerShellCopy codePS C:\> Get-Clipboard > clip.txt
```

- **Adding timestamps to log files:**

```
batchCopy codeC:\> echo %DATE% %TIME% >> <TXT LOG>.txt
```

- **Remote addition/modification of registry keys:**

```
batchCopy codeC:\> reg add \\<REMOTE COMPUTER NAME>\HKLM\Software\<REG KEY INFO>
```

- **Remote retrieval of registry values:**

```
batchCopy codeC:\> reg query \\<REMOTE COMPUTER NAME>\HKLM\Software\<REG KEY INFO>
```

- **Checking and testing registry paths:**

```
PowerShellCopy codePS C:\> Test-Path "HKCU:\Software\Microsoft\<HIVE>"
```

- **Remote copy of files:**

```
batchCopy codeC:\> robocopy C:\<SOURCE SHARED FOLDER> \\<DESTINATION COMPUTER>\<DESTINA
```

- **Checking various file extensions in a path:**

```
PowerShellCopy codePS C:\> Test-Path C:\Scripts\Archive\* -include *.PS1, *.VBS
```

- **Displaying file contents:**

```
batchCopy codeC:\> type <FILE NAME>
```

- **Merging contents of several files:**

```
batchCopy codeC:\> type <FILE NAME 1> <FILE NAME 2> <FILE NAME 3> > <NEW FILE NAME>
```

Desktops, allowing creation of multiple display pages in Desktop: Source: https://technet.microsoft.com/enus/sysinternals/cc817881

Executing live:

```
batchCopy codeC:\> "%ProgramFiles%\Internet Explorer\iexplore.exe" "https://live.sysinte
```

- **Remote mounting and permitting Read and Read/Write:**

```
batchCopy codeC:\> net share MyShare_R=c:\<READ ONLY FOLDER> /GRANT:EVERYONE,READ
C:\> net share MyShare_RW=c:\<READ/WRITE FOLDER> /GRANT:EVERYONE,FULL
```

- **Executing a task remotely using PSEXEC:** Source: https://technet.microsoft.com/enus/sysinternals/psexec.aspx

```
batchCopy codeC:\> psexec.exe \\<TARGET IP ADDRESS> -u <USER NAME> -p <PASSWORD> /C C:\<
C:\> psexec \\<TARGET IP ADDRESS> -u <ADMIN LEVEL USER NAME> -p <PASSWORD> C:\<PROGRA
```

- **Executing a task and sending its result to a shared environment:**

```
batchCopy codeC:\> wmic /node:ComputerName process call create cmd.exe /c netstat -an >
```

- **Comparing changes between two files:**

```
PowerShellCopy codePS C:\> Compare-Object (Get-Content <LOG FILE NAME 1>.log) -Differenc
```

- **Executing a task remotely using PowerShell:**

```
PowerShellCopy codePS C:\> Invoke-Command -ComputerName <COMPUTER NAME> {<PS COMMAND>}
```

- **PowerShell commands guide:**

```
PowerShellCopy codePS C:\> Get-Help <PS COMMAND> -full
```

Linux

- **Remote traffic inspection and analysis over ssh:**

```
bashCopy code# ssh root@<REMOTE IP ADDRESS OF HOST TO SNIFF> tcpdump -i any -U -s 0 -w -
```

- **Create a note or data entry in syslog:**

```
bashCopy code# logger "Something important to note in Log"
# dmesg | grep <COMMENT>
```

- **Create a read-only mounting:**

```
bashCopy code# mount -o ro /dev/<YOUR FOLDER OR DRIVE> /mnt
```

- **Remote Mounting over SSH:**

```
bashCopy code# apt-get install sshfs
# adduser <USER NAME> fuse
Log out and log back in.
mkdir /<WHERE TO MOUNT LOCALLY>
# sshfs <REMOTE USER NAME>@<REMOTE HOST>:/<REMOTE PATH> /<WHERE TO MOUNT LOCALLY>
```

- **Creating an SMB share in Linux:**

```
bashCopy code# useradd -m <NEW USER>
# passwd <NEW USER>
# smbpasswd -a <NEW USER>
# echo [Share] >> /etc/samba/smb.conf
# echo path = /<PATH OF FOLDER TO SHARE> >> /etc/samba/smb.conf
# echo available = yes >> /etc/samba/smb.conf
# echo valid users = <NEW USER> >> /etc/samba/smb.conf
# echo read only = no >> /etc/samba/smb.conf
# echo browsable = yes >> /etc/samba/smb.conf
# echo public = yes >> /etc/samba/smb.conf
# echo writable = yes >> /etc/samba/smb.conf
# service smbd restart
```

Display Remote System Share:

```
shellCopy code> smb:\\<IP ADDRESS OF LINUX SMB SHARE>
```

Copy File Remotely to Another System:

```
shellCopy code> scp <FILE NAME> <USER NAME>@<DESTINATION IP ADDRESS>:/<REMOTE FOLDER>
```

Create Mount and SMB Shared Environment Remotely in Another System:

```
shellCopy code# mount -t smbfs -o username=<USER NAME> //<SERVER NAME OR IP ADDRESS>/<Sh
```

Monitoring Websites and Files:

```
shellCopy code# while :; do curl -sSr http://<URL> | head -n 1; sleep 60; done
```

Alternative Method (Reference):

```
shellCopy codefor i in `curl -s -L cnn.com | egrep --only-matching "http(s?):\/\/[^ \"\(
do curl -s -I $i 2>/dev/null | head -n 1 | cut -d$' ' -f2; sleep 60; done
```

Decoding

Hex Connection

Convert from hex to decimal in Windows:

```
cmdCopy codeC:\> set /a 0xff
255
PS C:\> 0xff
255
```

Other Mathematical Operations in Windows:

```
cmdCopy codeC:\> set /a 1+2
3
C:\> set /a 3*(9/4)
6
C:\> set /a (2*5)/2
5
C:\> set /a "32>>3"
4
```

Decrypt Base64 Text within a File:

```
cmdCopy codeC:\> certutil -decode <BASE64 ENCODED FILE NAME> <DECODED FILE NAME>
```

XOR Decryption, Search for http: *Source: https://blog.didierstevens.com/programs/xorsearch/*

```
cmdCopy codeC:\> xorsearch.exe -i -s <INPUT FILE NAME> http
```

Convert hex to decimal in Linux:

```
shellCopy code# echo "0xff" | calc -d
= 255
```

Convert decimal to hex in Linux:

```
shellCopy code$ echo "25" | calc -h
= 0xff
```

Decrypt HTML Strings:

```
powershellCopy codePS C:\> Add-Type -AssemblyName System.Web
PS C:\> [System.Uri]::UnescapeDataString("HTTP%3a%2f%2fHello%20World.com")
HTTP://Hello World.com
```

SNORT Tool

SNORT Rules

Snort Rules for Identifying Meterpreter Traffic: *Source: https://blog.didierstevens.com/2015/06/16/metasploit-meterpreter-reverse-https-snort-rule/*

```
alert tcp $HOME_NET any-> $EXTERNAL_NET $HTTP_PORTS
(msg:"Metasploit User Agent String";
flow:to_server,established; content:"User-Agentl3al
Mozilla/4,0 (compatible\; MSIE 6.0\; Windows NT
5.1) l0d 0al"; http_header; classtype:trojanactivity;
reference:url,blog,didierstevens.com/2015/03/16/quic
kpost-metasploit-user-agent-strings/; sid:1618000;
rev:1;)
alert tcp $HOME_NET any-> $EXTERNAL_NET $HTTP_PORTS
( msg: "Metasploit User Agent St ring";
flow:to_server,established; content:"User-Agentl3al
Mozilla/4.0 (compatible\; MSIE 6,1\; Windows NT) l0d
0al"; http_header; classtype:trojan-activity;
reference:url,blog,didierstevens.com/2015/03/16/quic
kpost-metasploit-user-agent-strings/; sid:1618001;
rev: 1;)
alert tcp $HOME_NET any-> $EXTERNAL_NET $HTTP_PORTS
(msg: "Metasploit User Agent String";
flow:to_server,established; content:"User-Agentl3al
Mozilla/4,0 (compatible\; MSIE 7,0\; Windows NT
6.0) l0d 0al"; http_header; classtype:trojanactivity;
reference:url,blog.didierstevens.com/2015/03/16/quic
kpost-metasploit-user-agent-strings/; sid:1618002;
rev: 1;)
alert tcp $HOME_NET any-> $EXTERNAL_NET $HTTP_PORTS
(msg:"Metasploit User Agent String";
flow:to_server,established; content:"User-Agentl3al
Mozilla/4,0 (compatible\; MSIE 7,0\; Windows NT
6,0\; Trident/4,0\; SIMBAR={7DB0F6DE-8DE7-4841-9084-
28FA914B0F2E}\; SLCCl\; ,Nl0d 0al"; http_header;
classtype:trojan-activity;
```

```
reference:url,blog.didierstevens.com/2015/03/16/quic
kpost-metasploit-user-agent-strings/; sid:1618003;
rev: 1;)
alert tcp $HOME_NET any -> $EXTERNAL_NET $HTTP_PORTS
(msg:"Metasploit User Agent String";
flow:to_server,established; content:"User-Agentl3al
Mozilla/4.0 (compatible\; Metasploit RSPEC)l0d 0al";
http_header; classtype:trojan-activity;
reference:url,blog,didierstevens.com/2015/03/16/quic
kpost-metasploit-user-agent-strings/; sid:1618004;
rev: 1;)
alert tcp $HOME_NET any -> $EXTERNAL_NET $HTTP_PORTS
(msg:"Metasploit User Agent String";
flow:to_server,established; content:"User-Agentl3al
Mozilla/5,0 (Windows\; U\; Windows NT 5,1\; en-US)
AppleWebKit/525,13 (KHTML, like Gecko)
Chrome/4.0.221.6 Safari/525,13l0d 0al"; http_header;
classtype:trojan-activity;
reference:url,blog.didierstevens.com/2015/03/16/quic
kpost-metasploit-user-agent-strings/; sid:1618005;
rev: 1;)
alert tcp $HOME_NET any -> $EXTERNAL_NET $HTTP_PORTS
( msg: "Metasploit User Agent St ring";
flow:to_server,established; content:"User-Agentl3al
Mozilla/5.0 (compatible\; Googlebot/2.1\;
flow:to_server,established; content:"User-Agentl3al
Mozilla/5.0 (compatible\; Googlebot/2.1\;
+http://www.google.com/bot.html) l0d 0al";
http_header; classtype:trojan-activity;
reference:url,blog,didierstevens.com/2015/03/16/quic
kpost-metasploit-user-agent-strings/; sid:1618006;
rev: 1;)
alert tcp $HOME_NET any -> $EXTERNAL_NET $HTTP_PORTS
(msg: "Metasploit User Agent St ring";
flow:to_server,established; content:"User-Agentl3al
Mozilla/5,0 (compatible\; MSIE 10,0\; Windows NT
6,1\; Trident/6,0) l0d 0al"; http_header;
classtype:trojan-activity;
reference:url,blog.didierstevens.com/2015/03/16/quic
kpost-metasploit-user-agent-strings/; sid:1618007;
rev: 1;)
```

Snort Rules for Detect PSEXEC:

https://github.com/John-Lin/dockersnort/blob/master/snortrules-snapshot- 2972/rules/policy-other.rules

```
alert tcp $HOME_NET any -> $HOME_NET [139,445]
(msg:"POLICY-OTHER use of psexec remote
admin ist rat ion tool"; flow: to_server, established;
content:" IFFISMB1A2I"; depth:5; offset:4;
content:"ISC
.00 I p I 00 Is I 00 I e I 00 Ix I 00 I e I 00 I c I 00 I s I 00 Iv I 00 I c" ;
nocase; metadata:service netbios-ssn;
reference:url,technet.microsoft.com/enus/
sysinternals/bb897553.aspx; classtype:policyviolation;
sid:24008; rev:1;)
alert tcp $HOME_NET any -> $HOME_NET [139,445]
(msg:"POLICY-OTHER use of psexec remote
administration tool SMBv2";
flow:to_server,established; content:"IFEISMB";
depth:8; nocase; content:"105 001"; within:2;
distance:8;
content:"Pl001Sl00IEl00IXl00IEl00ISl00IVl00ICl00I";
fast_pattern:only; metadata:service netbios-ssn;
reference:url,technet.microsoft,com/enus/
sysinternals/bb897553.aspx[l]; classtype:policyviolation;
sid:30281; rev:1;)
```

Signature of DOS and DDOS Attacks

Methods of DoS and DDoS Attacks: *Source: https://www.trustwave.com/Resources/SpiderLabs-Blog/PCAP-Files-Are-Great-Aren-t-They–/*

Based on Volume: For example, bandwidth usage reaches from 1 GB to 10 GB. *Source: http://freecode.com/projects/iftop*

```
shellCopy code# iftop -n
```

Based on Various Protocols: Using different protocols For example, SYN Flood, ICMP Flood, UDP flood

```
shellCopy code# tshark -r <FILE NAME>.pcap -q -z io,phs
# tshark -c 1000 -q -z io,phs
# tcpdump -tnr $FILE | awk -F '. ' '{print $1","$2"."$3","$4}' | sort | uniq -c | sort -
# tcpdump -qnn "tcp[tcpflags] & (tcp-syn) != 0"
# netstat -s
```

For example, it targets only one protocol

```
shellCopy code# tcpdump -nn not arp and not icmp and not udp
# tcpdump -nn tcp
```

Connection State: For example, the firewall can manage 10,000 concurrent connections, and the

```
shellCopy code# netstat -n | awk '{print $6}' | sort | uniq -c | sort -nr | head
```

Applications: Layer 7 Attacks For example, HTTP GET flood, for high-volume image files.

```
shellCopy code# tshark -c 10000 -T fields -e http.host | uniq -c | sort -r | head -n 10
# tshark -r capture6 -T fields -e http.request.full_uri | sort | uniq -c | sort -r | hea
# tcpdump -n 'tcp[32:4] = 0x47455420' | cut -f 7- -d ":"
```

For example, requests for archive files, GIF, ZIP, JPEG, PDF, PNG are unusual.

```
shellCopy code# tshark -Y "http contains \"ff:d8\"" || "http contains \"GIF89a\"" || "ht
```

For example, pay attention and review the ‘user-agent’ amount in the web request.

```
shellCopy code# tcpdump -c 1000 -Ann | grep -Ei 'user-agent' | sort | uniq -c | sort -nr
```

For example, check the requested source headers.

```
shellCopy code# tcpdump -i en0 -A -s 500 | grep -i refer
```

Review HTTP requests to identify suspicious or dangerous patterns:

```
shellCopy code# tcpdump -s 1024 -l -A dst <EXAMPLE.COM>
```

Poisoning or Poison: Layer 2 Attacks For example, ARP poison, race condition DNS, DHCP

```
shellCopy code# tcpdump 'arp or icmp'
# tcpdump -tnr <SAMPLE TRAFFIC FILE>.pcap ARP | awk -F ',' '{print $1"."$2","$3","$4}'
# tshark -r <SAMPLE TRAFFIC FILE>.pcap -q -z io,phs | grep arp.duplicate-address-detecte
```

Toolset Prepared Machines and Operating Systems

KALI - Open Source Pentesting Distribution *Source: https://www.kali.org*

SIFT - SANS Investigative Forensics Toolkit *Source: http://sift.readthedocs.org/*

REMNUX - A Linux Toolkit for Reverse-Engineering and Analyzing Malware *Source: https://remnux.org*

OPEN VAS - Open Source vulnerability scanner and manager *Source: http://www.openvas.org*

MOLOCH - Large scale IPv4 packet capturing (PCAP), indexing and database system *Source: https://github.com/aol/moloch/wiki*

SECURITY ONION - Linux distro for intrusion detection, network security monitoring, and log management *Source: https://security-onionsolutions.github.io/security-onion/*

NAGIOS - Network Monitoring, Alerting, Response, and Reporting Tool *Source: https://www.nagios.org*

OSSEC - Scalable, multi-platform, open source Host-based Intrusion Detection System *Source: http://ossec.github.io*

← Previous
Recover Remediate

Next
Incident Management Checklist →

Last modified 1h ago

Incident Management Checklist

Identification Tasks

Acquire a copy of Malicious file(s) for analysis?		
Priority: H/M/L	Effort: H/M/L	Open/Closed
Malicious effects on systems list. Acquire an itemized list of all known changes on computer systems, files, settings, registry, services add/modified/deleted or stop/started.		
Priority: H/M/L	Effort: H/M/L	Open/Closed
Which A/V or malware tools can detect and remove malicious threat?		
Priority: H/M/L	Effort: H/M/L	Open/Closed
Where does malware/attacker exit the network?		
Priority: H/M/L	Effort: H/M/L	Open/Closed
Malicious internal/external sites/connections still active?		
Priority: H/M/L	Effort: H/M/L	Open/Closed
Malware listening on any ports?		
Priority: H/M/L	Effort: H/M/L	Open/Closed
Malware method of original infection, and/or weakness?		
Priority: H/M/L	Effort: H/M/L	Open/Closed
Packet capture of Malware trying to infect others?		
Priority: H/M/L	Effort: H/M/L	Open/Closed

Any packet capture of malware trying to communicate out of network and ID method of ports, IPs, DNS, etc?		
Priority: H/M/L	Effort: H/M/L	Open/Closed
Malware pose threat to any sensitive data (Files, credentials, Intellectual Property, PII, etc?		
Priority: H/M/L	Effort: H/M/L	Open/Closed
What are the DNS entries on an infected system?		
Priority: H/M/L	Effort: H/M/L	Open/Closed
Is it possible to detect the first infected system(s)?		
Priority: H/M/L	Effort: H/M/L	Open/Closed
Has the first systems hard drive been preserved?		
Priority: H/M/L	Effort: H/M/L	Open/Closed
Do any scripts need to be ran on live infected systems?		
Priority: H/M/L	Effort: H/M/L	Open/Closed
Does client have desktop management tool? If so, what reports are available to inventory all systems and statuses?		
Priority: H/M/L	Effort: H/M/L	Open/Closed
List of all infected systems?		
Priority: H/M/L	Effort: H/M/L	Open/Closed
Identify any patching missing with current and/or previous vulnerability scan.		
Priority: H/M/L	Effort: H/M/L	Open/Closed
Look for systems that have stopped reporting into Malware servers for updates, or which ones have stopped going to AV vendors for updates.		
Priority: H/M/L	Effort: H/M/L	Open/Closed
Look for systems that have stopped going to Update server or directly to Microsoft for updates.		

Contents of Tasks:

How many systems are still unknown, clear, suspicious, or infected?		
Priority: H/M/L	Effort: H/M/L	Open/Closed
Networking device(s) changes. (Switches, Routers, Firewalls, IPS, NAC, Wi-Fi, etc.).		
Priority: H/M/L	Effort: H/M/L	Open/Closed
Active Directory OU isolation of suspected systems.		
Priority: H/M/L	Effort: H/M/L	Open/Closed
Active Directory - User account restrictions and resets.		
Priority: H/M/L	Effort: H/M/L	Open/Closed
Active Directory policies to prohibit threats from running and/or access.		
Priority: H/M/L	Effort: H/M/L	Open/Closed
Firewall blocks.		
Priority: H/M/L	Effort: H/M/L	Open/Closed
DNS blocks (null route malware site(s).		
Priority: H/M/L	Effort: H/M/L	Open/Closed
Web filtering blocks.		
Priority: H/M/L	Effort: H/M/L	Open/Closed

- Monitor and analyze security alerts
- Validate the incident
- Assign severity to the incident
- Log initial incident details
- Notify the incident response (IR) team

Remediation Tasks

Administrative AD Password Changes.		
Priority: H/M/L	Effort: H/M/L	Open/Closed
Local Administrative Password Changes.		
Are there any applications in use that are facilitating the attack? If so, are there alternatives?		
Priority: H/M/L	Effort: H/M/L	Open/Closed
Is there a baseline system to review for changes?		
Priority: H/M/L	Effort: H/M/L	Open/Closed
Monitor user name variations.		
Priority: H/M/L	Effort: H/M/L	Open/Closed
Managing and monitoring tasks.		
Priority: H/M/L	Effort: H/M/L	Open/Closed
Review border router logs.		
Priority: H/M/L	Effort: H/M/L	Open/Closed
Review VPN (remote access) logs.		
Priority: H/M/L	Effort: H/M/L	Open/Closed
Citrix / VMWare or similar logs.		
Priority: H/M/L	Effort: H/M/L	Open/Closed
Review accounting server(s) logs and trends of users.		
Priority: H/M/L	Effort: H/M/L	Open/Closed
AD server logs.		
Priority: H/M/L	Effort: H/M/L	Open/Closed
Review Anti-Virus (Malicious Code Services) logs.		
Priority: H/M/L	Effort: H/M/L	Open/Closed
Review email abuse notifications and logs.		
Priority: H/M/L	Effort: H/M/L	Open/Closed
Review DNS logs.		
Priority: H/M/L	Effort: H/M/L	Open/Closed
Review account and policy abuse logs.		
Priority: H/M/L	Effort: H/M/L	Open/Closed
Review host firewall logs.		
Priority: H/M/L	Effort: H/M/L	Open/Closed

Contents of Tasks:

- Contain the incident short-term and long-term
- Eradicate the root cause
- Validate system functionality
- Implement system enhancements
- Notify external entities if needed (such as law enforcement or customers)
- Document actions taken and outcomes

Other Matters Regarding Tasks

Rebuild all systems in life cycle rebuild plan.		
Priority: H/M/L	Effort: H/M/L	Open/Closed
Synchronize time services across of systems.		
Priority: H/M/L	Effort: H/M/L	Open/Closed
Create incident data repository.		
Priority: H/M/L	Effort: H/M/L	Open/Closed
Consider host based IPS.		
Priority: H/M/L	Effort: H/M/L	Open/Closed
Consider Network Access Control (NAC).		
Priority: H/M/L	Effort: H/M/L	Open/Closed
3rd Party internal/external security and perimeter security tools and assessment services.		
Priority: H/M/L	Effort: H/M/L	Open/Closed

Contents:

- After-action review: Analyze what happened and why, what was effective, and what can be improved.
- Knowledge sharing: Ensure learnings and insights from the incident are shared with relevant stakeholders.
- Updating protocols: Adjust policies and protocols as necessary to prevent repeat incidents.

Malware Features Checklist

Malware Presence on the System:	
Runs in memory only,	Yes|No|Unknown|N/A
Runs out of registry, or	Yes|No|Unknown|N/A
Artifacts on disk	Yes|No|Unknown|N/A
Disk file presence hidden, stored in unallocated, free/slack space or encrypted.	Yes|No|Unknown|N/A
Has no icon.	Yes|No|Unknown|N/A
Has no description or company name.	Yes|No|Unknown|N/A
Unsigned Microsoft images.	Yes|No|Unknown|N/A
Are packed and likely encrypted.	Yes|No|Unknown|N/A
Suspicious DLLs or services.	Yes|No|Unknown|N/A
Backups and swaps itself in and out in place of real file.	Yes|No|Unknown|N/A
Stays alive working in file pairs	Yes|No|Unknown|N/A
Found in embedded deivces, industrial controls and IOT	Yes|No|Unknown|N/A
Malware Activities	
Downloads new code/functionality.	Yes|No|Unknown|N/A
Leverages pivot system(s) and network path(s) to exit the victim network including VPN/Dial-Up, HTTP/HTTPS, and other standard or non-standard services and ports.	Yes|No|Unknown|N/A
Ability to leverage mobile devices and other removable media.	Yes|No|Unknown|N/A
Ability to detect and utilize authenticated web proxies.	Yes|No|Unknown|N/A
Morphs on victim client system.	Yes|No|Unknown|N/A
Contains red herring (misleading/distracting) features depending on the environment it detects.	Yes|No|Unknown|N/A
Ability to traverse all known operating systems.	Yes|No|Unknown|N/A
Ability to move into embedded devices.	Yes|No|Unknown|N/A

Malware Capabilities	
Ability to conduct most Windows based Active Directory commands.	Yes|No|Unknown|N/A
Ability to upload and download files/payloads.	Yes|No|Unknown|N/A
Can use built-in services or purpose built malware for needed services.	Yes|No|Unknown|N/A
Has several persistent features, making the malware highly resilient to A/v defenses.	Yes|No|Unknown|N/A
Ability to brute force .	Yes|No|Unknown|N/A
Ability to DoS/DDoS tools.	Yes|No|Unknown|N/A
Ability to steal and/or pass the hash.	Yes|No|Unknown|N/A
Ability to conduct credential harvesting.	Yes|No|Unknown|N/A
Privilege escalation capability.	Yes|No|Unknown|N/A
Ransomware or like capability.	Yes|No|Unknown|N/A
Self-Destruct mode, including destructive methods.	Yes|No|Unknown|N/A
Anti memory forensics	Yes|No|Unknown|N/A
Is sandbox aware and virtual machine aware.	Yes|No|Unknown|N/A
Apply software patch to prevent other malware infection	Yes|No|Unknown|N/A
C2 techniques: DNS, HTTP, HTTPS, stegonography, cloud, TOR, online code, etc.	Yes|No|Unknown|N/A
One time install/detonation	Yes|No|Unknown|N/A
Communicates in no predictable patterns including short and long-term sleep techniques.	Yes|No|Unknown|N/A
Makes use of compromised CA, in order to hide communications.	Yes|No|Unknown|N/A
Timezone and IP Geo aware	Yes|No|Unknown|N/A
Makes use of well-established commercial compromised web sites for C2, i.e. Dropbox, Gmail, etc.	Yes|No|Unknown|N/A

network traffic?

6)

, removable drives)?

- Does it exploit any known vulnerabilities?
- Is it propagated via social engineering?

4. **Payload:**
    - Is it ransomware, spyware, a trojan, a worm, or something else?
    - Does it exfiltrate data?
    - What kinds of data does it target (credentials, personal data, etc.)?
    - Does it have any destructive capabilities?
5. **Evasion Techniques:**
    - Does it have anti-analysis capabilities (like sandbox detection)?
    - Does it employ any obfuscation techniques?
    - Does it have rootkit functionalities to hide its presence?
6. **Command and Control (C2):**
    - Does it communicate with a C2 server?
    - What is the IP address/domain of the C2?
    - What protocols does it use to communicate?
7. **Persistence Mechanism:**
    - How does it ensure it remains on the infected system?
    - Does it create or modify registry entries?
    - Does it create or manipulate scheduled tasks?

Previous
Tactics Tips And Tricks

Next
Security Incident-Identification Sche...

Last modified 1h ago

Security Incident-Identification Schema

Identifying Security Incidents Related to Advanced Persistent Threats (APTs)

Group Categorization

- MITRE ATT&CK Groups
- FireEye APT Groups

Group Reviews

- Pauli APT Review
- Peerlyst APT Wiki

Recent Incidents

- Malpedia

General Source: VERIS Community

Identifying Threats Using Patterns:

incident_id
#
security_incident
Confirmed, Suspected, False positive, Near miss, No
confidence
High, Medium, Low, None
victim.employee_count
#
timeline.unit
Unknown, NA, Seconds, Minutes, Hours, Days, Weeks, Months, Years, Never
impact.overall_rating
Unknown, Insignificant, Distracting, Painful, Damaging, Catastrophic
impact.loss.variety
Asset and fraud, Brand damage, Business disruption, Operating costs, Legal and regulatory, Competitive advantage, Response and recovery
impact.loss.rating
Unknown, Major, Moderate, Minor, None

discovery_method
Unknown, Ext - actor disclosure, Ext - fraud detection, Ext - monitoring service, Ext - customer, Ext - unrelated party, Ext - audit, Ext - unknown, Int - antivirus, Int - incident response, Int - financial audit, Int - fraud detection, Int - HIDS, Int - IT audit, Int - log review, Int - NIDS, Ext - law enforcement, Int - security alarm, Int - reported by user, Int - unknown, Other
targeted
Unknown, Opportunistic, Targeted, NA
cost_corrective_action
Unknown, Simple and cheap, Difficult and expensive, Something in-between
country
Unknown, Two Letter, Other
iso_currency_code
AED, AFN, ALL, AMD, ANG, AOA, ARS, AUD, AWG, AZN, BAM, BBD, BDT, BGN, BHD, BIF, BMD, BND, BOB, BRL, BSD, BTN, BWP, BYR, BZD, CAD, CDF, CHF, CLP, CNY, COP, CRC, CUC, CUP, CVE, CZK, DJF, DKK, DOP, DZD, EGP, ERN, ETB, EUR, FJD, FKP, GBP, GEL, GGP, GHS, GIP, GMD, GNF, GTQ, GYD, HKD, HNL, HRK, HTG, HUF, IDR, ILS, IMP, INR, IQD, IRR, ISK, JEP, JMD, JOD, JPY, KES, KGS, KHR, KMF, KPW, KRW, KWD, KYD, KZT, LAK, LBP, LKR, LRD, LSL, LTL, LVL, LYD, MAD, MDL, MGA, MKD, MMK, MNT, MOP, MRO, MUR, MVR, MWK, MXN, MYR, MZN, NAD, NGN, NIO, NOK, NPR, NZD, OMR, PAB, PEN, PGK, PHP, PKR, PLN, PYG, QAR, RON, RSD, RUB, RWF, SAR, SBD, SCR, SDG, SEK, SGD, SHP, SLL, SOS, SPL, SRD, STD, SVC, SYP, SZL, THB, TJS, TMT, TND, TOP, TRY, TTD, TVD, TWD, TZS, UAH, UGX, USD, UYU, UZS, VEF, VND, VUV, WST, XAF, XCD, XDR, XOF, XPF, YER, ZAR, ZMK, ZWD

- Actor: [Individual or entity responsible for the threat]

actor.x.motive
Unknown, NA, Espionage, Fear, Financial, Fun, Grudge, Ideology, Convenience, Other
actor.external.variety
Unknown, Activist, Auditor, Competitor, Customer, Force majeure, Former employee, Nation-state, Organized crime, Acquaintance, State-affiliated, Terrorist, Unaffiliated, Other
actor.internal.variety
Unknown, Auditor, Call center, Cashier, End-user, Executive, Finance, Helpdesk, Human resources, Maintenance, Manager, Guard, Developer, System admin, Other

- Action: [Actions taken or methods used by the threat actor]

action.malware.variety
Unknown, Adware, Backdoor, Brute force, Capture app data, Capture stored data, Client-side attack, Click fraud, C2, Destroy data, Disable controls, DoS, Downloader, Exploit vuln, Export data, Packet sniffer, Password dumper, Ram scraper, Ransomware, Rootkit, Scan network, Spam, Spyware/Keylogger, SQL injection, Adminware, Worm, Other
action.malware.vector
Unknown, Direct install, Download by malware, Email autoexecute, Email link, Email attachment, Instant
action.social.variety
Unknown, Baiting, Bribery, Elicitation, Extortion, Forgery, Influence, Scam, Phishing, Pretexting, Propaganda, Spam, Other
action.social.vector
Unknown, Documents, Email, In-person, IM, Phone, Removable media, SMS, Social media, Software, Website, Other
action.social.target
Unknown, Auditor, Call center, Cashier, Customer, End-user, Executive, Finance, Former employee, Helpdesk, Human resources, Maintenance, Manager, Partner, Guard, Developer, System admin, Other
action.misuse.variety
Unknown, Knowledge abuse, Privilege abuse, Embezzlement, Data mishandling, Email misuse, Net misuse, Illicit content, Unapproved workaround, Unapproved hardware, Unapproved software, Other
action.misuse.vector
Unknown, Physical access, LAN access, Remote access, Non-corporate, Other
action.physical.variety
Unknown, Assault, Sabotage, Snooping, Surveillance, Tampering, Theft, Wiretapping, Connection, Other
action.physical.location
Unknown, Partner facility, Partner vehicle, Personal residence, Personal vehicle, Public facility, Public vehicle, Victim secure area, Victim work area, Victim public area, Victim grounds, Other
action.physical.vector
Unknown, Privileged access, Visitor privileges, Bypassed controls, Disabled controls, Uncontrolled location, Other

action.error.variety
Unknown, Classification error, Data entry error, Disposal error, Gaffe, Loss, Maintenance error, Misconfiguration, Misdelivery, Misinformation, Omission, Physical accidents, Capacity shortage, Programming error, Publishing error, Malfunction, Other
action.error.vector
Unknown, Random error, Carelessness, Inadequate personnel, Inadequate processes, Inadequate technology, Other
action.environmental.variety
Unknown, Deterioration, Earthquake, EMI, ESD, Temperature, Fire, Flood, Hazmat, Humidity, Hurricane, Ice, Landslide, Lightning, Meteorite, Particulates, Pathogen, Power failure, Tornado, Tsunami, Vermin, Volcano, Leak, Wind, Other

- Asset: [Targeted resources or information]

asset.variety
Unknown, S - Authentication, S - Backup, S - Database, S - DHCP, S - Directory, S - DCS, S - DNS, S - File, S - Log, S - Mail, S - Mainframe, S - Payment switch, S - POS controller, S - Print, S - Proxy, S - Remote access, S - SCADA, S - Web application, S - Code repository, S - VM host, S - Other N - Access reader, N - Camera, N - Firewall, N - HSM, N - IDS, N - Broadband, N - PBX, N - Private WAN, N - PLC, N - Public WAN, N - RTU, N - Router or switch, N - SAN, N - Telephone, N - VoIP adapter, N - LAN, N - WLAN, N - Other U - Auth token, U - Desktop, U - Laptop, U - Media, U - Mobile phone, U - Peripheral, U - POS terminal, U - Tablet, U - Telephone, U - VoIP phone, U - Other T - ATM, T - PED pad, T - Gas terminal, T - Kiosk, T - Other M - Tapes, M - Disk media, M - Documents, M - Flash drive, M - Disk drive, M - Smart card, M - Payment card, M - Other P - System admin, P - Auditor, P - Call center, P - Cashier, P - Customer, P - Developer, P - End-user, P - Executive, P - Finance, P - Former employee, P - Guard, P - Helpdesk, P - Human resources, P - Maintenance, P - Manager, P - Partner, P - Other
asset.accessibility
Unknown, External, Internal, Isolated, NA
asset.ownership
Unknown, Victim, Employee, Partner, Customer, NA
asset.management
Unknown, Internal, External, NA
asset.hosting
Unknown, Internal, External shared, External dedicated, External, NA
asset.cloud
Unknown, Hypervisor, Partner application, Hosting governance, Customer attack, Hosting

- Attribute: [Characteristics or properties related to the incident]

attribute.confidentiality.data_disclosure
Unknown, Yes, Potentially, No
attribute.confidentiality.data.variety
Unknown, Credentials, Bank, Classified, Copyrighted, Medical, Payment, Personal, Internal, System, Secrets, Other
attribute.confidentiality.state
Unknown, Stored, Stored encrypted, Stored unencrypted, Transmitted, Transmitted encrypted, Transmitted unencrypted, Processed
attribute.integrity.variety
Unknown, Created account, Hardware tampering, Alter behavior, Fraudulent transaction, Log tampering, Misappropriation, Misrepresentation, Modify configuration, Modify privileges, Modify data, Software installation, Other
attribute.availability.variety
Unknown, Destruction, Loss, Interruption, Degradation, Acceleration, Obscuration, Other

Action Framework Structured Threat Information eXpression (STIX) Source: STIX Project

coa.type
Blocking, Redirecting, Hardening, Patching, Rebuilding, Monitoring, Other
coa.impact
Insignificant, Distracting, Painful, Damaging, Catastrophic, Unknown
coa.efficacy
Not Effective, Somewhat Effective, Mostly Effective, Completely Effective, NA
coa.stage
Prepare, Remedy, Response, Recovered
coa.hosting
Unknown, Internal, External shared, External dedicated, External, NA
coa.objective
Detect, Deny, Disrupt, Degrade, Deceive, Destroy

KILL CHAIN MAPPING Information list for KILL CHAIN MAPPING Source: Lockheed Martin - Intel Driven Defense

Phase	Identified evidence, artifact, info, or intel	Course of Action
Active Reconnaissance		Detect, Deny, Disrupt, Degrade, Deceive, Destroy
Customization		Detect, Deny, Disrupt, Degrade, Deceive, Destroy
Delivery		Detect, Deny, Disrupt, Degrade, Deceive, Destroy
Exploitation		Detect, Deny, Disrupt, Degrade, Deceive, Destroy
		Detect, Deny,

Prioritized Defended Asset List (PDAL) List and prioritize assets to defend Source:

Asset:			
Location:		Criticality:	
Description:		Vulnerability:	
Purpose:		Recoverability:	
Time Prioritized:		Ranking:	
Priority:		I	

Asset:			
Location:		Criticality:	
Description:		Vulnerability:	
Purpose:		Recoverability:	
Time Prioritized:		Ranking:	
Priority:		II	

Asset:			
Location:		Criticality:	
Description:		Vulnerability:	
Purpose:		Recoverability:	
Time Prioritized:		Ranking:	
Priority:		III	

Previous
Incident Management Checklist

Next - Hardening
main

Last modified 1h ago

main

In the cybersecurity ecosystem, the Blue Team is synonymous with defense. Their primary objective is to safeguard an organization's digital infrastructure, data, and networks from cyber threats. One pivotal strategy employed by Blue Teams to enhance cybersecurity defenses is "hardening." This involves implementing measures to secure operating systems (OS), networks, and devices, thereby reducing vulnerabilities and minimizing the attack surface. This article explores hardening in the context of Blue Team operations, providing a table of tools, and offering tips and tricks for hardening various components of an IT environment.

Hardening: A Cornerstone of Cybersecurity

Hardening is the process of securing a system by reducing its surface of vulnerability. It involves configuring the system to minimize the potential for exploitation, implementing protective measures, and conducting regular audits to ensure security. In the context of Blue Team operations, hardening is applied across various domains, including:

- **Operating Systems**: Ensuring that the OS is configured securely and is resilient against threats.
- **Networks**: Protecting the network infrastructure to safeguard data in transit and prevent unauthorized access.
- **Devices**: Securing physical devices to protect data at rest and ensure the integrity of hardware components.

Tools for Hardening in Blue Team Operations

Purpose	Tool Name	Description
OS Hardening	Security Compliance Manager (SCM)	A Microsoft tool that provides ready-to-deploy policies and DCM configuration packs that are tested and fully supported.
Network Hardening	Nmap	A network scanner tool used to discover hosts and services on a computer network and create a "map" of the network.
Device Hardening	BitLocker	A full disk encryption program that protects data from loss, theft, or hackers.
Patch Management	WSUS	Microsoft's Windows Server Update Services allows administrators to manage the distribution of updates released through Microsoft Update to computers.
Configuration Management	Ansible	An open-source software provisioning, configuration management, and application-deployment tool.
Vulnerability Management	OpenVAS	An open-source framework of several services and tools offering a comprehensive vulnerability scanning and management solution.
Firewall Management	Firewalld	A firewall management tool available by default on Ubuntu, CentOS, and Red Hat.

Tips and Tricks for Hardening

Operating System Hardening

- **Patch Regularly**: Ensure that the OS is regularly updated and patched.
- **Least Privilege Principle**: Ensure users and applications operate using the least amount of privilege necessary.
- **Disable Unnecessary Services**: Turn off services and features that are not required to minimize vulnerabilities.
- **Implement Security Policies**: Use Group Policy Objects (GPOs) and Security Templates to enforce security settings.

Network Hardening

- **Network Segmentation**: Divide the network into segments to contain breaches and minimize lateral movement.
- **Implement Firewalls**: Use firewalls to control incoming and outgoing network traffic based on an applied rule set.
- **Use VPNs**: Employ Virtual Private Networks (VPNs) to encrypt data in transit across untrusted networks.
- **Secure Wireless Networks**: Implement WPA3, disable WPS, and use a strong Pre-Shared Key (PSK).

Device Hardening

- **Full Disk Encryption**: Use tools like BitLocker to encrypt the entire disk, protecting data at rest.
- **Physical Security**: Ensure that devices are physically secure to prevent unauthorized access.
- **Secure Boot**: Enable secure boot to ensure that the device only loads trusted software.
- **Device Authentication**: Implement Multi-Factor Authentication (MFA) for accessing devices.

Previous
Security Incident-Identification Sche...

Next - Hardening
SCM

Last modified 1h ago

SCM

Overview

Microsoft's Security Compliance Manager (SCM) is a robust tool designed to help organizations manage and create security baselines for various Microsoft products. It provides ready-to-deploy policies and Desired Configuration Management (DCM) packs that are tested and fully supported.

Key Features of SCM

- **Security Baselines**: Pre-configured security settings for various Microsoft products.
- **Configuration Management**: Manage and customize security baselines.
- **Export Capabilities**: Export security baselines in various formats (GPO backup, SCAP, DCM, etc.)
- **Security Guidance**: Access to security best practices and guidance.

Cheat Sheet

1. **Install SCM**: Ensure that you have the latest version of SCM installed.
2. **Download Baselines**: Download the latest security baselines for the Microsoft products in use.
3. **Import Baselines**: Import security baselines into SCM.
4. **Customize Baselines**: Adjust the settings in the security baselines to meet the specific needs of your organization.
5. **Export Baselines**: Export the customized baselines in the desired format (e.g., GPO backup, Excel, etc.)
6. **Deploy Baselines**: Implement the baselines in your environment using Group Policy or SCCM.
7. **Monitor Compliance**: Regularly check systems for compliance with the applied baselines.
8. **Update Baselines**: Periodically check for and apply updates to security baselines.
9. **Audit and Review**: Conduct audits and review security baselines to ensure they align with organizational security needs.
10. **Document Changes**: Keep a log of all changes made to security baselines and configurations.

Examples for Hardening with SCM

1. Import Windows 10 Baseline

- SCM Home → Import Baseline → Windows 10

2. Customize Windows Server 2019 Baseline

- SCM Home → Windows Server 2019 Baseline → Customize

3. Export Office 365 ProPlus Baseline as GPO

- Customized Office 365 ProPlus Baseline → Export as GPO Backup

4. Deploy Windows 10 Baseline with Group Policy

- Exported Windows 10 GPO Backup → Import in Group Policy Management Console

5. Monitor Compliance for Windows Server 2016

- Deployed Windows Server 2016 Baseline → Monitor using SCCM

6. Update Windows 10 Baseline

- SCM Home → Windows 10 Baseline → Check for Updates

7. Audit SQL Server Configurations

- Deployed SQL Server Baseline → Audit using SCM

8. Document Changes to Exchange Server Baseline

- Customized Exchange Server Baseline → Document Changes

9. Manage Versioning for Windows 10 Baseline

- Documented Windows 10 Baseline → Manage Versioning

10. Validate Compliance for Windows Server 2019

- Deployed Windows Server 2019 Baseline → Validate using SCM

11. Customize and Export Edge Browser Baseline

- SCM Home → Edge Browser Baseline → Customize → Export

12. Deploy Office 2019 Baseline with SCCM

- Exported Office 2019 Baseline → Deploy using SCCM

13. Review and Update Domain Controller Baseline

- SCM Home → Domain Controller Baseline → Review and Update

14. Monitor Compliance for Office 2016 Baseline

- Deployed Office 2016 Baseline → Monitor using SCM

15. Export and Deploy Windows Defender Baseline

- Customized Windows Defender Baseline → Export → Deploy using Group Policy

16. Validate Compliance for SharePoint Server Baseline

- Deployed SharePoint Server Baseline → Validate using SCM

17. Review and Customize Windows Firewall Baseline

- SCM Home → Windows Firewall Baseline → Customize

18. Export and Document Windows 8.1 Baseline

- Customized Windows 8.1 Baseline → Export → Document Changes

19. Deploy and Monitor SQL Server Baseline

- Exported SQL Server Baseline → Deploy using SCCM → Monitor Compliance

20. Audit and Update Windows Server 2012 R2 Baseline

- Deployed Windows Server 2012 R2 Baseline → Audit using SCM → Update Baseline

← Hardening - Previous
main

Next - Hardening
WSUS →

Last modified 1h ago

WSUS

Cheatsheet

1. Install WSUS

- Open Server Manager → Add roles and features → WSUS

2. Configure WSUS

- Open WSUS → Complete the Configuration Wizard

3. Create Computer Groups

- WSUS Console → Computers → Create a Computer Group

4. Approve Updates

- WSUS Console → Updates → Approve Updates

5. Deploy WSUS to Clients

- Group Policy → Configure Update Source → Point to WSUS Server

6. Monitor Update Installations

- WSUS Console → Reports → Update Status

7. Manage WSUS Configurations

- WSUS Console → Options → WSUS Server Configuration Wizard

8. Synchronize Updates

- WSUS Console → Synchronizations → Start Synchronization

9. Cleanup WSUS

- WSUS Console → Options → Server Cleanup Wizard

10. Secure WSUS Communication

```
Configure SSL on WSUS -> Update Group Policy for Secure Communication
```

Examples for Hardening with WSUS

1. Install WSUS Role

```
Install-WindowsFeature -Name UpdateServices -IncludeManagementTools
```

2. Configure WSUS Post-Installation

```
& "$env:ProgramFiles\Update Services\Tools\WsusUtil.exe" postinstall CONTENT_DIR=D:\WSUS
```

3. Create a Computer Group in WSUS

Navigate through WSUS Console → Computers → Add Computer Group → Name: "SecureGroup"

4. Approve Updates for a Group

Navigate through WSUS Console → Updates → Select an Update → Approve → Select "SecureGroup"

5. Configure WSUS on Clients via GPO

- Open Group Policy Management → Create a GPO → Navigate to: Computer Configuration → Policies → Administrative Templates → Windows Components → Windows Update → Configure Automatic Updates & Specify intranet Microsoft update service location → Define WSUS Server

6. Start WSUS Synchronization

```
Get-WsusServer | Get-WsusSubscription | Start-WsusSynchronization
```

7. Retrieve Update Status

Navigate through WSUS Console → Reports → Update Status

8. Configure WSUS to Use SSL

- Configure SSL on WSUS Server → Update Group Policy to use "https://[WSUS_SERVER]"

9. Run WSUS Cleanup

```
Get-WsusServer | Invoke-WsusServerCleanup -CleanupObsoleteComputers -CleanupObsoleteUpda
```

10. Set WSUS to Download from Microsoft Update

Navigate through WSUS Console → Options → Update Source and Proxy Server → Synchronize from Microsoft Update

11. Configure Update Files and Languages

Navigate through WSUS Console → Options → Update Files and Languages → Store update files locally on this server

12. Configure Automatic Approvals

Navigate through WSUS Console → Options → Automatic Approvals → Add Rule

13. Retrieve WSUS Synchronization Status

```
PowerShellCopy codeGet-WsusServer | Get-WsusSubscription | Get-WsusSynchronizationStatus
```

14. Configure WSUS Email Notifications

Navigate through WSUS Console → Options → Email Notifications → Configure SMTP Server and Notification Options

15. Manually Add a Computer to WSUS

```
PowerShellCopy codeAdd-WsusComputer -ComputerToAdd "ComputerName" -TargetGroupName "Secu
```

16. Retrieve WSUS Update Installations

Navigate through WSUS Console → Reports → Update Installations

17. Configure WSUS Reporting Rollup

Navigate through WSUS Console → Options → Reporting Rollup → Enable roll up of update status from replica downstream servers

18. Set WSUS Clients to Download from Peers

- Configure Delivery Optimization on Clients via GPO → Set Download Mode to "LAN" (Value: 1)

19. Retrieve WSUS Computer Status

Navigate through WSUS Console → Computers → Select a Computer Group → Status

20. Configure WSUS Products and Classifications

Navigate through WSUS Console → Options → Products and Classifications → Select Products to Update

← Hardening - Previous
SCM

Next - Hardening
OSSEC →

Last modified 1h ago

OSSEC

Cheatsheet

1. Install OSSEC

- Download OSSEC → Install OSSEC Server/Agent

2. Configure OSSEC

- Edit ossec.conf → Define configurations

3. Manage OSSEC Agents

- Register agents → Manage agent keys

4. Customize OSSEC Rules

- Navigate to rules directory → Customize or add new rules

5. Configure OSSEC Alerts

- Edit ossec.conf → Define email alerts

6. Monitor OSSEC Logs

- Navigate to logs → Monitor ossec.log

7. Upgrade OSSEC

- Download new version → Upgrade OSSEC

8. Integrate OSSEC with SIEM

- Configure OSSEC → Forward logs to SIEM

9. Analyze OSSEC Alerts

- Navigate to alerts directory → Analyze alerts.log

10. Secure OSSEC Communication

```
Configure agent and server -> Validate secure communication
```

Examples for Hardening with OSSEC

1. Install OSSEC Server

```
wget https://github.com/ossec/ossec-hids/archive/[VERSION].tar.gz
tar -zxvf [VERSION].tar.gz
cd ossec-hids-[VERSION]
sudo ./install.sh
```

2. Install OSSEC Agent

```
# Use the same steps as the server but select agent during installation.
```

3. Add an OSSEC Agent

```
sudo /var/ossec/bin/manage_agents
# Follow prompts to add an agent.
```

4. Extract Agent Key

```
sudo /var/ossec/bin/manage_agents
# Follow prompts to extract key.
```

5. Add Agent Key to OSSEC Agent

```
esudo /var/ossec/bin/manage_agents
# Follow prompts to add key.
```

6. Restart OSSEC

```
sudo /var/ossec/bin/ossec-control restart
```

7. Create a Custom OSSEC Rule

- Navigate to `/var/ossec/rules` → Create a custom rule file

8. Configure OSSEC to Send Email Alerts

- Edit `/var/ossec/etc/ossec.conf` → Add email alert settings

9. Check OSSEC Agent Status

```
sudo /var/ossec/bin/agent_control -l
```

10. View OSSEC Logs

```
ecat /var/ossec/logs/ossec.log
```

11. Analyze OSSEC Alerts

```
cat /var/ossec/logs/alerts/alerts.log
```

12. Upgrade OSSEC Server/Agent

- Download new version → Follow upgrade steps

13. Disable an OSSEC Rule

- Navigate to `/var/ossec/etc/rules/local_rules.xml` → Add rule to disable

14. Configure OSSEC Active Response

- Edit `/var/ossec/etc/ossec.conf` → Define active response settings

15. Test OSSEC Rule

```
/var/ossec/bin/ossec-logtest
# Enter log entry to test.
```

16. View OSSEC Agents

```
sudo /var/ossec/bin/agent_control -lc
```

17. Remove OSSEC Agent

```
sudo /var/ossec/bin/manage_agents
# Follow prompts to remove an agent.
```

18. Configure OSSEC Syscheck

- Edit `/var/ossec/etc/ossec.conf` → Define syscheck settings

19. View OSSEC Statistical Information

```
sudo /var/ossec/bin/ossec-logtest -s
```

20. Configure OSSEC to Monitor a File

- Edit `/var/ossec/etc/ossec.conf` → Add file to syscheck

← Hardening - Previous
WSUS

Next - Hardening
Ansible →

Last modified 1h ago

Ansible

Cheatsheet

1. Install Ansible

- Download and install Ansible on the control node.

2. Configure Ansible Hosts

- Define hosts and groups in the Ansible inventory.

3. Write Playbooks

- Create Ansible playbooks to define the desired state of systems.

4. Use Ansible Roles

- Utilize roles for organizing and reusing playbooks.

5. Run Ansible Playbooks

- Execute playbooks to apply configurations to hosts.

6. Use Ansible Galaxy

- Leverage Ansible Galaxy to use pre-built roles.

7. Secure Ansible Vault

- Use Ansible Vault to secure sensitive data.

8. Optimize Ansible Configurations

- Tweak ansible.cfg for performance and behavior.

9. Utilize Ansible Modules

- Use modules to define the desired state in playbooks.

10. Implement Ansible Facts

```
diffCopy code- Use gathered facts for making informed decisions in playbooks.
```

Examples for Hardening with Ansible

1. Install Ansible

```
sudo apt update
sudo apt install ansible
```

2. Add Hosts to Ansible Inventory

```
[webservers]
192.168.1.10
192.168.1.11
```

3. Simple Ansible Playbook to Update Systems

```
---
- hosts: webservers
  become: yes
  tasks:
    - name: Ensure all packages are updated
      apt:
        update_cache: yes
        upgrade: safe
```

4. Run Ansible Playbook

```
ansible-playbook -i hosts update_system.yml
```

5. Use Ansible Role from Galaxy

```
ansible-galaxy install dev-sec.os-hardening
```

6. Use Ansible Vault to Encrypt Data

```
ansible-vault create secret.yml
```

7. Use Encrypted Data in Playbook

```
---
- hosts: webservers
  become: yes
  vars_files:
    - secret.yml
  tasks:
    - name: Add user
      user:
        name: "{{ username }}"
        password: "{{ password }}"
```

8. Run Playbook with Vault Password

```
ansible-playbook --ask-vault-pass -i hosts add_user.yml
```

9. Use Ansible Facts in Playbook

```
---
- hosts: webservers
  tasks:
    - name: Display OS
      debug:
        var: ansible_distribution
```

10. Install and Start Apache using Ansible

```
---
- hosts: webservers
  become: yes
  tasks:
    - name: Ensure Apache is installed
      apt:
        name: apache2
        state: present
    - name: Ensure Apache is running
      service:
        name: apache2
        state: started
```

11. Create a User with Ansible

```
---
- hosts: webservers
  become: yes
  tasks:
    - name: Ensure user 'john' exists
      user:
        name: john
        state: present
```

12. Disable Unused Service

```
---
- hosts: webservers
  become: yes
  tasks:
    - name: Ensure telnet is stopped and disabled
      service:
        name: telnet
        state: stopped
        enabled: no
```

13. Configure SSH Hardening

```
---
- hosts: webservers
  become: yes
  tasks:
    - name: Ensure only SSH protocol 2 is used
      lineinfile:
        path: /etc/ssh/sshd_config
        regex: '^Protocol'
        line: 'Protocol 2'
```

14. Set Up a Firewall Rule

```
---
- hosts: webservers
  become: yes
  tasks:
    - name: Allow only SSH and HTTP through the firewall
      ufw:
        rule: allow
        name: "{{ item }}"
      loop:
        - ssh
```

```
        - http
```

15. Ensure a Package is Removed

```
---
- hosts: webservers
  become: yes
  tasks:
    - name: Ensure 'telnet' is removed
      apt:
        name: telnet
        state: absent
```

16. Configure Password Authentication

```
---
- hosts: webservers
  become: yes
  tasks:
    - name: Disable password authentication
      lineinfile:
        path: /etc/ssh/sshd_config
        regex: '^PasswordAuthentication'
        line: 'PasswordAuthentication no'
```

17. Ensure NTP is Configured

```
---
- hosts: webservers
  become: yes
  tasks:
    - name: Ensure NTP is installed
      apt:
        name: ntp
        state: present
```

18. Configure Kernel Parameters

```
---
- hosts: webservers
  become: yes
  tasks:
    - name: Ensure IP forwarding is disabled
      sysctl:
        name: net.ipv4.ip_forward
        value: '0'
        state: present
```

19. Ensure a Service is Running

```
---
- hosts: webservers
  become: yes
  tasks:
    - name: Ensure Apache is running
      service:
        name: apache2
        state: started
```

20. Apply Security Patches

```
---
- hosts: webservers
  become: yes
  tasks:
    - name: Ensure all packages are updated
      apt:
        upgrade: dist
```


Hardening - Previous
OSSEC

Next - Hardening
Firewalld

Last modified 59m ago

Firewalld

Cheatsheet

1. Install Firewalld

- Ensure Firewalld is installed and running on your system.

2. Manage Firewalld Service

- Start, enable, stop, or disable the Firewalld service.

3. Configure Zones

- Define and manage zones to control the trust level of network connections.

4. Manage Services

- Allow, deny, or customize services in zones.

5. Manage Ports

- Open or close specific ports in zones.

6. Manage Interfaces

- Assign network interfaces to zones.

7. Manage Sources

- Assign specific IP addresses or subnets to zones.

8. Manage ICMP Blocks

- Allow or deny ICMP messages in zones.

9. Manage Masquerading and Port Forwarding

- Configure NAT and port forwarding.

10. Manage Rich Rules

```
Use rich rules for more detailed control over traffic.
```

20 Real Examples for Hardening with Firewalld

1. Install Firewalld

```
sudo yum install firewalld
```

2. Start and Enable Firewalld

```
sudo systemctl start firewalld
sudo systemctl enable firewalld
```

3. Get Active Zone

```
sudo firewall-cmd --get-active-zones
```

4. Change Default Zonesudo firewall-cmd --set-default-zone=home

5. Add Service to Zone

```
sudo firewall-cmd --zone=public --add-service=http --permanent
```

6. Remove Service from Zone

```
sudo firewall-cmd --zone=public --remove-service=http --permanent
```

7. Add Port to Zone

```
sudo firewall-cmd --zone=public --add-port=8080/tcp --permanent
```

8. Remove Port from Zone

```
sudo firewall-cmd --zone=public --remove-port=8080/tcp --permanent
```

9. Reload Firewalld

```
sudo firewall-cmd --reload
```

10. Add Interface to Zone

```
sudo firewall-cmd --zone=public --add-interface=eth0 --permanent
```

11. Add Source to Zone

```
sudo firewall-cmd --zone=public --add-source=192.168.1.0/24 --permanent
```

12. Enable Masquerading

```
sudo firewall-cmd --zone=public --add-masquerade --permanent
```

13. Add Forward Port

```
sudo firewall-cmd --zone=public --add-forward-port=port=80:proto=tcp:toport=8080 --perma
```

14. Add ICMP Block

```
sudo firewall-cmd --zone=public --add-icmp-block=echo-request --permanent
```

15. Create Custom Service

- Define a custom service XML file and place it in `/etc/firewalld/services/`.

16. Add Custom Service to Zone

```
sudo firewall-cmd --zone=public --add-service=custom-service --permanent
```

17. Add Rich Rule

```
sudo firewall-cmd --zone=public --add-rich-rule='rule family="ipv4" source address="192.
```

18. Remove Rich Rule

```
sudo firewall-cmd --zone=public --remove-rich-rule='rule family="ipv4" source address="
```

19. Query Service in Zone

```
sudo firewall-cmd --zone=public --query-service=http
```

20. List All Configurations

```
sudo firewall-cmd --list-all-zones
```

← Hardening - Previous
Ansible

Next - XDR
Wazuh →

Last modified 58m ago

Wazuh

Cheatsheet

Install Wazuh

- Ensure Wazuh Manager and Agent are installed and configured.

2. Configure Wazuh Manager

- Set up manager configurations, including communication and data paths.

3. Register Wazuh Agents

- Add and manage agents to communicate with the Wazuh manager.

4. Implement Wazuh Rules

- Customize and implement rules for log analysis.

5. Implement Wazuh Decoders

- Customize and implement decoders to interpret received logs.

6. Configure Wazuh Policies

- Implement policies for compliance and system checks.

7. Integrate Wazuh with Elastic Stack

- Set up Wazuh-Elastic Stack integration for visualization and analysis.

8. Implement Wazuh File Integrity Monitoring

- Configure syscheck for file integrity monitoring.

9. Configure Wazuh Alerts

- Set up alert levels and actions in rules.

10. Secure Wazuh Manager and Agents

```
Ensure secure communication and access control.
```

Examples for Hardening with Wazuh

1. Install Wazuh Manager

Refer to the Wazuh documentation for detailed installation steps.

2. Register Wazuh Agent

```
/var/ossec/bin/agent-auth -m [MANAGER_IP]
```

3. Start Wazuh Agent

```
systemctl start wazuh-agent
```

4. Create a Custom Wazuh Rule

- Navigate to `/var/ossec/etc/rules` and create a custom rule file (e.g., `1000-my_rules.xml`).

5. Create a Custom Wazuh Decoder

- Navigate to `/var/ossec/etc/decoders` and create a custom decoder file (e.g., `0005-my_decoders.xml`).

6. Restart Wazuh Manager

```
systemctl restart wazuh-manager
```

7. Enable FIM for a Directory

Add the following to your `/var/ossec/etc/ossec.conf`:

8. Configure Wazuh Alert Level

- Edit the rule in `/var/ossec/etc/rules` and set a specific alert level.

9. Configure Wazuh to Monitor a Log File

Add the following to your `/var/ossec/etc/ossec.conf`:

```
<localfile>
    <log_format>syslog</log_format>
    <location>/var/log/my_log.log</location>
</localfile>
```

10. Implement PCI DSS Policy

- Utilize Wazuh's built-in PCI DSS compliance capabilities by enabling relevant rules.

11. Configure Email Alerts

Add the following to your `/var/ossec/etc/ossec.conf`:

```
<global>
    <email_notification>yes</email_notification>
    <email_to>[YOUR_EMAIL]</email_to>
    <smtp_server>smtp.example.com</smtp_server>
    <email_from>ossec@example.com</email_from>
</global>
```

12. Implement GDPR Policy

- Utilize Wazuh's built-in GDPR compliance capabilities by enabling relevant rules.

13. Configure Wazuh for Vulnerability Detection

Add the following to your `/var/ossec/etc/ossec.conf`:

```
<wodle name="vulnerability-detector">
    <enabled>yes</enabled>
    <interval>5h</interval>
    <ignore_time>6h</ignore_time>
    <run_on_start>yes</run_on_start>
    <!-- Add feeds here -->
</wodle>
```

14. Configure Wazuh for Cloud Security Monitoring

- Integrate Wazuh with AWS, Azure, or GCP for cloud security monitoring.

15. Configure Wazuh for Docker Monitoring

Add the following to your `/var/ossec/etc/ossec.conf`:

```
<wodle name="docker-listener">
    <disabled>no</disabled>
    <interval>10m</interval>
    <run_on_start>yes</run_on_start>
</wodle>
```

16. Configure Wazuh for Office 365 Monitoring

- Set up the Office 365 module for monitoring Office 365 activities.

17. Implement HIPAA Policy

- Utilize Wazuh's built-in HIPAA compliance capabilities by enabling relevant rules.

18. Configure Wazuh for Anomaly and Malware Detection

- Implement rules and decoders for detecting anomalies and malware activities.

19. Configure Wazuh for Network IDS

- Integrate Wazuh with Suricata or Zeek for network intrusion detection.

20. Configure Wazuh for Endpoint Detection and Response

- Implement rules, decoders, and policies for monitoring endpoint activities and responding to threats.

← Hardening - Previous
Firewalld

Next - Query Language
KQL →

Last modified 54m ago

KQL

Cheatsheet

1. Basic Query Format

- Start with the table name followed by a series of query operators.

2. Filtering

- Use `where` to filter results based on a condition.

3. Sorting

- Use `order by` to sort results based on a column.

4. Aggregation

- Use `summarize` to aggregate data.

5. Joining Tables

- Use `join` to combine tables based on a related column.

6. Selecting Columns

- Use `project` to select which columns to display.

7. Renaming Columns

- Use `extend` or `project` with `as` to rename columns.

8. Limiting Results

- Use `take` to limit the number of results returned.

9. Calculating Time Difference

- Use `datetime_diff` to calculate the difference between two datetime columns.

10. String Manipulation

```
Use `strcat`, `substring`, etc., for string operations.
```

Examples for Detection Query in KQL

1. Basic Query

```
SecurityEvent
```

2. Filter for a Specific Event ID

```
SecurityEvent
| where EventID == 4624
```

3. Retrieve Specific Columns

```
SecurityEvent
| project TimeGenerated, Computer, EventID
```

4. Count by Event ID

```
SecurityEvent
| summarize count() by EventID
```

5. Filter and Sort by Time

```
SecurityEvent
| where EventID == 4624
| order by TimeGenerated desc
```

6. Join Two Tables

```
SecurityEvent
| join (
    Syslog
SecurityEvent
| take 10
```

8. Calculate Time Difference

```
SecurityEvent
| extend duration = datetime_diff('second', TimeGenerated, TimeGenerated)
```

9. String Concatenation

```
SecurityEvent
| extend info = strcat(Computer, ":", EventID)
```

10. Filter with Multiple Conditions

```
SecurityEvent
| where EventID == 4624 and Computer == "MY-PC"
```

11. Count Events per Computer

```
SecurityEvent
| summarize count() by Computer
```

12. Filter for a Specific Time Range

```
SecurityEvent
| where TimeGenerated between (datetime(2022-01-01) .. datetime(2022-01-31))
```

13. Find Unique Values

```
SecurityEvent
| summarize count() by Account
| project Account
```

14. Calculate Average

```
Perf
| summarize avg(CounterValue) by CounterName
```

15. Group by Time Interval

```
SecurityEvent
| summarize count() by bin(TimeGenerated, 1h)
```

16. Use of Case Statement

```
SecurityEvent
| extend EventType = case(EventID == 4624, "Login", EventID == 4625, "Failed Login", "O
```

17. Filter with String Contains

```
SecurityEvent
| where Computer contains "MY-PC"
```

18. Top N Entities

```
SecurityEvent
| summarize count() by Computer
| top 5 by count_
```

19. Calculate Percentage

```
SecurityEvent
| summarize EventCount = count()
| extend Percentage = EventCount * 100 / toscalar(SecurityEvent | count())
```

20. Filter with Not Equal

```
SecurityEvent
| where Computer != "MY-PC"
```

XDR - Previous
Wazuh

Next - Query Language
EQL

Last modified 52m ago

EQL

Cheatsheet

1. Basic Query Format

- Start with an event type followed by a `where` clause for conditions.

2. Filtering

- Use `where` to filter results based on a condition.

3. Event Joining

- Use `sequence` to correlate events in a sequence.

4. Time Constraints

- Use `until` and `within` to define time constraints between sequence events.

5. Event Type Definition

- Define event types to filter on specific log types.

6. Field Comparisons

- Use field comparisons to correlate fields within and across events.

7. String Functions

- Use string functions like `concat`, `substring`, etc., for string operations.

8. Mathematical Operations

- Use mathematical operations like `+`, `-`, `*`, `/` for calculations.

9. Logical Operators

- Use logical operators like `and`, `or`, `not` for complex conditions.

10. Pipe Operations

```
Use `|` to perform operations like filtering, sorting, and counting on the query results
```

Examples for Detection Query in EQL

1. Basic Query

```
process where process_name == "cmd.exe"
```

2. Event Sequence

```
sequence by host.id
    [process where process_name == "cmd.exe"]
    [network where process_name == "cmd.exe" and port == 80]
```

3. Time Constraint

```
sequence by host.id
    [process where process_name == "cmd.exe"]
    [network where process_name == "cmd.exe" and port == 80] within 1m
```

4. Field Comparison

```
sequence by host.id
    [process where process_name == "cmd.exe"]
    [network where process_name == "cmd.exe" and port == 80 and process.pid == process.
```

5. String Concatenation

```
process where concat(process_name, " ", process.args) == "cmd.exe /c"
```

6. Logical Operator

```
process where process_name == "cmd.exe" and not user.name == "SYSTEM"
```

7. Mathematical Operation

```
file where file.size + 100 > 2000
```

8. Event Type Definition

```
file where opcode == "create" and file.extension == "exe"
```

9. Pipe and Count

```
process where process_name == "cmd.exe"
| count
```

10. Pipe and Unique Count

```
process where true
| unique_count user.name
```

11. Pipe and Sort

```
process where true
| sort process.start_time desc
```

12. Pipe and Filter

```
process where true
| filter process_name == "cmd.exe"
```

13. Pipe and Head

```
process where true
| head 5
```

14. Pipe and Tail

```
process where true
| tail 5
```

15. Subquery

```
process where process_name == "cmd.exe" and
    [file where file_name == "evil.exe"]
```

16. Wildcard Usage

```
process where process_name like "svchost.*"
```

17. Case Insensitive Match

```
process where process_name : "Cmd.ExE"
```

18. Length Function

```
process where length(process_name) > 5
```

19. Number Function

```
process where number(process_name) == 123
```

20. Array Function

```
process where array_length(process.args) > 2
```

Query Language - Previous
KQL

Next - Events
eventvwr

Last modified 50m ago

eventvwr

Cheatsheet

1. Opening Event Viewer

- Use `eventvwr.msc` from the Run dialog or search for "Event Viewer" in the Start menu.

2. Filtering Events

- Use the "Filter Current Log" option to narrow down events based on criteria like Event ID, Keywords, etc.

3. Creating Custom Views

- Use "Create Custom View" to save specific filters for quick access.

4. Exporting Logs

- Use the "Save All Events As" option to export logs in various formats (e.g., CSV, XML).

5. Clearing Logs

- Use the "Clear Log" option to delete all events from a specific log.

6. Attaching Tasks to Events

- Use the "Attach Task To This Event" option to perform specific actions when an event occurs.

7. Using Event Viewer with PowerShell

- Leverage PowerShell cmdlets like `Get-EventLog` and `Get-WinEvent` to query and manage event logs.

8. Understanding Event Levels

- Familiarize yourself with event levels (Information, Warning, Error, etc.) to prioritize investigations.

9. Understanding Event Sources

- Identify the source of events to understand which application or component logged them.

10. Analyzing Event Details

- Dive into the "Details" tab of an event to understand its specifics and troubleshoot effectively.

Event IDs in Microsoft Event Viewer

1. Event ID 4624: Successful Logon

- Indicates a user successfully logged on to a computer.

2. Event ID 4625: Logon Failure

- Indicates a failed logon attempt.

3. Event ID 4634: Logoff

- Indicates a user logoff.

4. Event ID 4648: Explicit Credential Logon

- Indicates a logon using explicit credentials.

5. Event ID 4663: File/Directory Access

- Indicates an attempt to access a file or directory.

6. Event ID 4672: Special Privileges Assigned

- Indicates special privileges assigned to a new logon.

7. Event ID 4688: Process Start

- Indicates a new process creation.

8. Event ID 4689: Process End

- Indicates a process termination.

9. Event ID 4698: Scheduled Task Created

- Indicates a scheduled task was created.

10. Event ID 4700: Scheduled Task Enabled

- Indicates a scheduled task was enabled.

11. Event ID 4719: System Audit Policy Change

- Indicates a change in audit policy.

12. Event ID 4720: User Account Created

- Indicates a user account was created.

13. Event ID 4722: User Account Enabled

- Indicates a user account was enabled.

14. Event ID 4725: User Account Disabled

- Indicates a user account was disabled.

15. Event ID 4738: User Account Changed

- Indicates a user account was changed.

16. Event ID 4740: User Account Locked Out

- Indicates a user account was locked out.

17. Event ID 4776: Credential Validation

- Indicates a domain controller attempted to validate credentials.

18. Event ID 4798: User Account Query

- Indicates a query was issued for a user account.

19. Event ID 4904: Security Auditing Setting Modification

- Indicates an attempt to modify the per-user auditing settings.

20. Event ID 4946: Windows Firewall Rule Added

- Indicates a new Windows Firewall rule was added.

Example PowerShell Commands

Query Specific Event ID

```
Get-WinEvent -FilterHashtable @{LogName='Security'; ID=4624} -MaxEvents 10
```

Query Events within a Date Range

```
Get-WinEvent -FilterHashtable @{LogName='Security'; StartTime='MM/DD/YYYY 00:00:00'; End
```

Query Events from a Specific Log Source

```
Get-WinEvent -FilterHashtable @{LogName='Security'; ProviderName='Microsoft-Windows-Secu
```

Query Language - Previous
EQL

Next - Events
Sysmon

Sysmon

Cheatsheet

1. Install Sysmon

- Ensure Sysmon is installed to monitor and log system activity.

2. Configure Sysmon

- Use an XML configuration file to define what events Sysmon should log.

3. Update Sysmon

- Update Sysmon to the latest version to leverage new features and fixes.

4. Uninstall Sysmon

- Remove Sysmon from the system when it's no longer needed.

5. Sysmon Event Logging

- Understand the different event IDs and what they represent.

6. Sysmon Filtering

- Implement filtering in the configuration to reduce noise.

7. Sysmon with SIEM

- Integrate Sysmon logs with SIEM solutions for analysis and correlation.

8. Sysmon Schema

- Understand the schema of Sysmon logs to create effective queries and alerts.

9. Sysmon and PowerShell

- Leverage PowerShell for Sysmon installation, configuration, and log querying.

45 Real Examples for Sysmon

1-9: Sysmon Event IDs and Their Significance

1. **Event ID 1: Process Creation**
   - Logs when a process is created and includes the command line.
2. **Event ID 2: File creation time**
   - Logs changes in file creation timestamps.
3. **Event ID 3: Network Connection**
   - Logs when a process makes an outbound network connection.
4. **Event ID 4: Sysmon Service State Change**
   - Logs changes in the Sysmon service state.
5. **Event ID 5: Process Termination**
   - Logs when a process terminates.
6. **Event ID 6: Driver Loaded**
   - Logs when a driver is loaded.
7. **Event ID 7: Image Loaded**
   - Logs DLLs and other images loaded into a process.
8. **Event ID 8: CreateRemoteThread**
   - Logs when a thread is created in another process.
9. **Event ID 9: RawAccessRead**
   - Logs when a process reads sectors from disk volume.

10-18: Sysmon Commands and Usage

10. **Install Sysmon**
    ```
    Sysmon.exe -i
    ```
11. **Install with Configuration**
    ```
    Sysmon.exe -i sysmonconfig.xml
    ```
12. **Update Sysmon Configuration**
    ```
    Sysmon.exe -c sysmonconfig.xml
    ```
13. **Uninstall Sysmon**

```
Sysmon.exe -u
```

14. **Dump Sysmon Configuration**

```
shellCopy codeSysmon.exe -c
```

15. **Update Sysmon**

```
Sysmon.exe -u sysmon.exe
```

16. **Check Sysmon Version**

```
Sysmon.exe -v
```

17. **Extract Sysmon Configuration**

```
Sysmon.exe -c config.xml
```

18. **Log to a Different Event Log**

```
Sysmon.exe -i -l <LogName>
```

19-45: Sysmon Configuration Examples

19-45. **Sysmon Configuration Examples** - Below is a sample Sysmon configuration XML snippet. A full configuration would typically contain multiple entries under each event type to define what should be logged and what should be excluded.

```
<Sysmon schemaversion="4.50">
    <!-- Capture all processes -->
    <EventFiltering>
        <ProcessCreate onmatch="exclude">
            <Image condition="is">C:\Windows\System32\svchost.exe</Image>
        </ProcessCreate>
        <!-- Exclude network connections to Microsoft IPs -->
        <NetworkConnect onmatch="exclude">
            <DestinationIp condition="is">13.107.4.50</DestinationIp>
        </NetworkConnect>
        <!-- Log all other network connections -->
        <NetworkConnect onmatch="include" />
        <!-- Log DLLs loaded into lsass.exe -->
        <ImageLoad onmatch="include">
            <Image condition="image">lsass.exe</Image>
        </ImageLoad>
        <!-- Exclude certain drivers -->
        <DriverLoad onmatch="exclude">
            <Signature condition="contains">Microsoft</Signature>
        </DriverLoad>
        <!-- Log other drivers -->
        <DriverLoad onmatch="include" />
    </EventFiltering>
</Sysmon>
```

Top Sysmon Event IDs

1. Event ID 1: Process Creation

- Logs when a process is created and includes the command line.

2. Event ID 2: File Creation Time Changed

- Logs changes in file creation timestamps.

3. Event ID 3: Network Connection

- Logs when a process makes an outbound network connection.

4. Event ID 4: Sysmon Service State Change

- Logs changes in the Sysmon service state.

5. Event ID 5: Process Terminated

- Logs when a process terminates.

6. Event ID 6: Driver Loaded

- Logs when a driver is loaded.

7. Event ID 7: Image Loaded

- Logs DLLs and other images loaded into a process.

8. Event ID 8: CreateRemoteThread

- Logs when a thread is created in another process.

9. Event ID 9: RawAccessRead

- Logs when a process reads sectors from disk volume.

10. Event ID 10: ProcessAccess

- Logs when a process opens another process.

11. Event ID 11: FileCreate

- Logs when a file is created or overwritten.

12. Event ID 12: RegistryEvent (Object create and delete)

- Logs when a registry object is created or deleted.

13. Event ID 13: RegistryEvent (Value Set)

- Logs when a registry value is set.

14. Event ID 14: RegistryEvent (Key and Value Rename)

- Logs when a registry key or value is renamed.

15. Event ID 15: FileCreateStreamHash

- Logs when a named file stream is created.

16. Event ID 16: Sysmon Config State Change

- Logs when the Sysmon configuration is changed.

17. Event ID 17: Pipe Created

- Logs when a named pipe is created.

18. Event ID 18: Pipe Connected

- Logs when a named pipe is connected.

19. Event ID 19: WmiEvent (WmiEventFilter activity detected)

- Logs WMI event filter creation.

20. Event ID 20: WmiEvent (WmiEventConsumer activity detected)

- Logs WMI event consumer creation.

← Events - Previous
eventvwr

Next - Threat Intelligence
Origin →

Last modified 47m ago

Origin

- https://twitter.com/sicehice
- https://twitter.com/realScamSniffer
- https://twitter.com/stealthmole_int
  https://dashboard.tenderly.co/
- https://urlhaus.abuse.ch/browse/
  https://github.com/nu11secur1ty/CVE-mitre/tree/main
- https://cvexploits.io/
- otx
- https://www.cnnvd.org.cn/home/warn
  https://vulmon.com/searchpage?q=&sortby=byactivity
- buaq.net
- https://sec.today/pulses/
- https://0dayfans.com/
- https://notes.netbytesec.com/
- https://speakerdeck.com/
- https://www.cnnvd.org.cn/home/loophole
- https://falconfeeds.io/
- https://opentip.kaspersky.com/

← Events - Previous
Sysmon

Next - Threat Intelligence
IOC →

Last modified 27m ago

IOC

- https://github.com/ThreatMon/ThreatMon-Reports-IOC

← Threat Intelligence - Previous
Origin

Next - CSIRT
Resources →

Last modified 31m ago

Resources

- https://github.com/Spacial/awesome-csirt
- https://github.com/angea/pocorgtfo
- https://lab52.io/blog/2162-2/

Threat Intelligence - Previous
IOC

Next - Digital Forensic
Resources

Last modified 28m ago

Workflow

1. Phishing Email Analysis

- Automate the extraction of indicators of compromise (IOCs) from phishing emails and check them against threat intelligence.

2. Malware Analysis

- Automatically submit suspicious files to a malware analysis sandbox and retrieve the results.

3. Automated Enrichment

- Automatically enrich IOCs with threat intelligence to provide context during analysis.

4. Blocking Malicious IPs

- Automatically block malicious IP addresses at the firewall or other security devices.

5. User Verification

- Automatically verify the status of a user when suspicious activity is detected.

6. Password Reset

- Implement an automated workflow for user password resets following a potential compromise.

7. Disabling User Accounts

- Automatically disable user accounts that are suspected to be compromised.

8. Quarantine Endpoint

- Isolate endpoints that are suspected to be compromised to prevent lateral movement.

9. Data Exfiltration Detection

- Implement workflows to detect and respond to potential data exfiltration.

10. Ransomware Response

```
Automate responses to ransomware, such as isolating affected systems and restoring backu
```

11. Patch Management

```
Automate the detection and deployment of patches for known vulnerabilities.
```

12. Incident Ticket Creation

```
Automatically create incident tickets in the ITSM tool during an incident.
```

13. User Notification

```
Notify users automatically in case of incidents that might affect them.
```

14. Incident Documentation

```
Automatically document all actions taken during an incident for post-mortem analysis.
```

15. Threat Indicator Sharing

```
Share threat indicators with external threat sharing platforms automatically.
```

16. SSL Certificate Renewal

```
Implement workflows to check and renew SSL certificates as needed.
```

17. Backup Verification

```
Automate the verification of backups to ensure they are valid and usable.
```

18. Cloud Security Monitoring

19. VPN Monitoring

```
Monitor VPN logs for abnormal activities and implement automated responses.
```

20. DDoS Mitigation

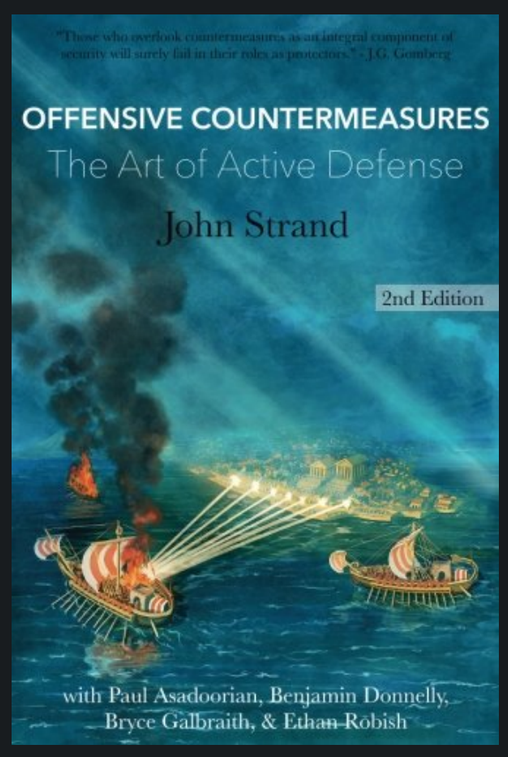
```
Implement workflows to detect and mitigate DDoS attacks, such as adjusting firewall rule
```

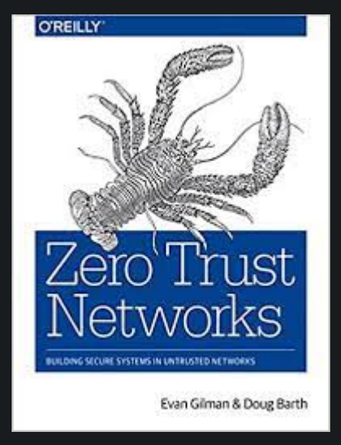
Digital Forensic - Previous
Resources

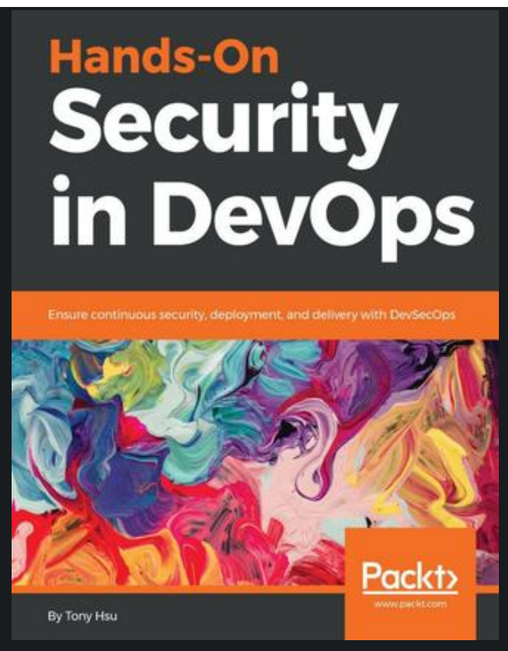
Next - Resources
Book

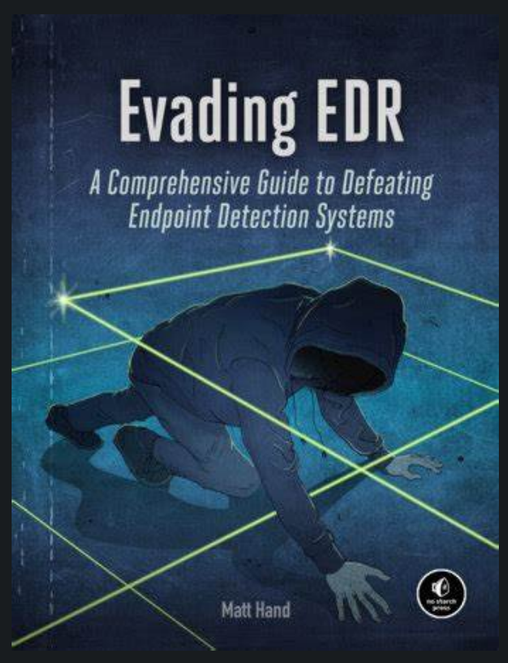
Last modified 16m ago

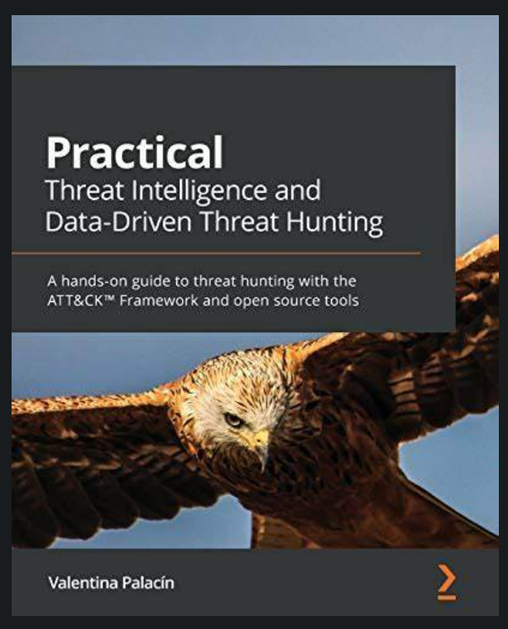
Book











SOAR - Previous
Workflow

Next - Resources
Standards

Last modified 36m ago

Standards

1. NIST Cybersecurity Framework

- **Link**: NIST Cybersecurity Framework
- **Description**: Developed by the National Institute of Standards and Technology, this framework provides a policy for managing and reducing cybersecurity risk.

2. ISO/IEC 27001:2013

- **Link**: ISO/IEC 27001:2013
- **Description**: An international standard that provides the requirements for an information security management system (ISMS).

3. CIS Critical Security Controls

- **Link**: CIS Controls
- **Description**: Developed by the Center for Internet Security, these controls provide a series of cybersecurity actions prioritized to mitigate the most prevalent cyber attacks.

4. MITRE ATT&CK Framework

- **Link**: MITRE ATT&CK
- **Description**: A knowledge base used to describe the actions and behaviors of cyber adversaries, providing a structured understanding of their tactics and techniques.

5. PCI DSS (Payment Card Industry Data Security Standard)

- **Link**: PCI DSS
- **Description**: A set of security standards designed to ensure that all companies that accept, process, store, or transmit credit card information maintain a secure environment.

6. HIPAA (Health Insurance Portability and Accountability Act)

- **Link**: HIPAA
- **Description**: U.S. legislation that provides data privacy and security provisions for safeguarding medical information.

7. GDPR (General Data Protection Regulation)

- **Link**: GDPR
- **Description**: A regulation that demands businesses to protect the personal data and privacy of EU citizens for transactions that occur within EU member states.

8. SOC 2 (Service Organization Control 2)

- **Link**: SOC 2
- **Description**: A framework for managing and securing data that is important to the privacy and confidentiality of an organization's data.

9. COBIT (Control Objectives for Information and Related Technologies)

- **Link**: COBIT
- **Description**: A framework for developing, implementing, monitoring, and improving IT governance and management practices.

10. ITIL (Information Technology Infrastructure Library)

- **Link**: ITIL
- **Description**: A set of practices for IT service management (ITSM) that focuses on aligning IT services with the needs of the business.

← Resources - Previous Book

Next - Resources Blogs →

Last modified 36m ago

Blogs

- https://socradar.io/resource/
- https://www.greynoise.io/blog/
- https://trufflesecurity.com/blog/
- https://konbriefing.com/en-topics/cyber-attacks.html
- https://blog.sinamohebi.com/
- https://threatmon.io/reports/

Resources - Previous
Standards

Next - Resources
Labs

Last modified 30m ago

Labs

TryHackMe

https://tryhackme.com

Blue Team Labs

https://blueteamlabs.online/

Hands-On SOC Analyst Training

https://letsdefend.io

Cybrary

https://cybrary.it

RangeForce

https://www.rangeforce.com/

CyberDefenders

https://cyberdefenders.org/blueteam-ctf-challenges/

Immersive Labs

https://immersivelabs.online/register

LetsDefend

https://letsdefend.io/

← Resources - Previous
Blogs

Next - Resources
Certificate →

Last modified 22m ago

Certificate

1. CompTIA Cybersecurity Analyst (CySA+)

- **Description**: Focuses on behavior analytics to improve the overall state of IT security.
- **Link**: CompTIA CySA+

2. GIAC Certified Incident Handler (GCIH)

- **Description**: Focuses on detecting, responding, and resolving computer security incidents.
- **Link**: GCIH

3. Certified Intrusion Analyst (GCIA)

- **Description**: Focuses on configuring and analyzing network intrusion detection systems.
- **Link**: GCIA

4. Cisco Certified CyberOps Associate

- **Description**: A certification that prepares candidates to begin a career working with associate-level cybersecurity analysts within security operations centers.
- **Link**: Cisco CyberOps Associate

← Resources - Previous
Labs

Last modified 26m ago