# Security Assessment & VAPT

## Theoretical Knowledge Report

---

### 1. Executive Summary

This report presents a theoretical overview of Security Assessment and Vulnerability Assessment & Penetration Testing (VAPT) using open-source and freely available tools. The objective is to understand how organizations can identify, assess, and prioritize security weaknesses without relying on paid solutions. The report covers assessment methodologies, security standards, risk evaluation techniques, common vulnerabilities, and documentation best practices aligned with industry frameworks such as NIST, OWASP, and ISO 27001.

---

### 2. Understanding Security Assessment

#### 2.1 Objective

To learn how to evaluate systems and applications for security weaknesses using free tools and established frameworks.

#### 2.2 Security Assessment Overview

Security Assessment is a structured process used to identify, analyze, and mitigate vulnerabilities within information systems. It helps organizations understand their security posture and reduce the risk of cyber threats.

#### 2.3 Frameworks Used

- **NIST (National Institute of Standards and Technology)** – Provides guidelines for security and risk management.
- **CIS (Center for Internet Security)** – Offers benchmarks for secure system configuration.
- **OWASP (Open Web Application Security Project)** – Focuses on web application security risks.

#### 2.4 Types of Security Testing

**Vulnerability Assessment**

- Identifies known vulnerabilities in systems.
- Typically automated.
- **Tool Used**: OpenVAS (Open Vulnerability Assessment System)

**Penetration Testing**

- Simulates real-world attacks to validate vulnerabilities.
- Determines the potential impact of exploitation.
- **Tools Used**: Kali Linux, Metasploit Framework, Nmap

**Compliance Testing**

- Verifies alignment with regulatory and security standards.
- Uses checklists and benchmarks.
- **Examples**: CIS Benchmarks, NIST compliance checklists

## 3. VAPT Methodology

### 3.1 Objective

To follow a structured and repeatable approach to vulnerability assessment and penetration testing.

### 3.2 VAPT Phases

#### 3.2.1 Planning

- Define scope, objectives, and rules of engagement.
- Identify target systems and testing limitations.
- **Tool Used**: Dradis CE (for scope and documentation)

#### 3.2.2 Discovery

- Identify active hosts, open ports, services, and applications.
- **Tools Used**:
  - Nmap – Network scanning and service detection
  - OWASP ZAP – Web application vulnerability scanning

#### 3.2.3 Attack (Exploitation)

- Exploit identified vulnerabilities to assess real-world impact.
- **Tool Used**: Metasploit Framework

#### 3.2.4 Reporting

- Document findings, risks, and remediation steps.
- **Resources Used**: Free penetration testing report templates (Pentest-Tools, GitHub)

### 3.3 Learning Methodology

- Practice using the OWASP Web Security Testing Guide (WSTG) to align testing with industry standards.

## 4. Security Standards & Compliance

### 4.1 Objective

To understand how security assessments align with legal and regulatory requirements.

### 4.2 Key Standards

- **GDPR (General Data Protection Regulation)**
  Focuses on data privacy and protection.
- **HIPAA (Health Insurance Portability and Accountability Act)**
  Ensures security of healthcare information.
- **ISO/IEC 27001**
  Defines requirements for an Information Security Management System (ISMS).

### 4.3 Learning Approach

- Map vulnerabilities to OWASP Top 10 categories.
- Understand how vulnerabilities impact compliance requirements.

## 5. Risk Assessment Basics

### 5.1 Objective

To prioritize vulnerabilities based on risk rather than quantity.

### 5.2 CVSS Scoring

- Uses the NVD CVSS Calculator to assign severity scores.
- Evaluates:
  - Attack Vector
  - Attack Complexity
  - Privileges Required
  - Impact on Confidentiality, Integrity, and Availability

### 5.3 Risk Matrix

- Vulnerabilities categorized as High, Medium, or Low risk.
- Likelihood and Impact are evaluated using a 3×3 risk matrix.
- **Tools Used**: Google Sheets / Microsoft Excel

---

## 6. Common Vulnerabilities

### 6.1 Objective

To identify common security flaws through hands-on practice.

### 6.2 Network Vulnerabilities

- Open ports
- Weak services
- Misconfigurations
- **Tool Used**: Nmap

### 6.3 Web Application Vulnerabilities

- SQL Injection (SQLi)
- Cross-Site Scripting (XSS)
- Authentication and session flaws
- **Practice Platform**: OWASP Juice Shop

### 6.4 Learning Labs

- **Metasploitable** – Intentionally vulnerable virtual machine
- **VulnHub** – Collection of vulnerable machines for practice

---

## 7. Documentation Fundamentals

### 7.1 Objective

To produce clear, professional, and actionable security reports.

### 7.2 Documentation Tools

- **Dradis CE** – Centralized and collaborative reporting
- **CherryTree** – Technical note-taking
- Standard reporting tools (Word, LibreOffice, PDF)

### 7.3 Learning Resources

- Free VAPT and security assessment report templates from GitHub
- Sample professional penetration testing reports

---

**8. Conclusion**

This report demonstrates that effective **security assessment and VAPT** can be performed using **free and open-source tools** when combined with industry-recognized frameworks and methodologies. By following structured testing phases, understanding compliance requirements, applying risk assessment techniques, and maintaining professional documentation, organizations and learners can significantly improve their cybersecurity posture.

---

## Practical Application Report

---

**1. Executive Summary**

This report documents the **practical execution of a vulnerability assessment** conducted in a controlled lab environment using **open-source security tools**. The assessment focused on identifying vulnerabilities in an intentionally vulnerable virtual machine (**Metasploitable 3**) using **Kali Linux** as the attacker system. Tools such as **OpenVAS** and **Nikto** were used to discover, analyze, and prioritize vulnerabilities. The findings were evaluated using **CVSS scoring** and a **risk matrix**, followed by remediation recommendations.

---

**2. Testing Environment Setup**

**2.1 Objective**

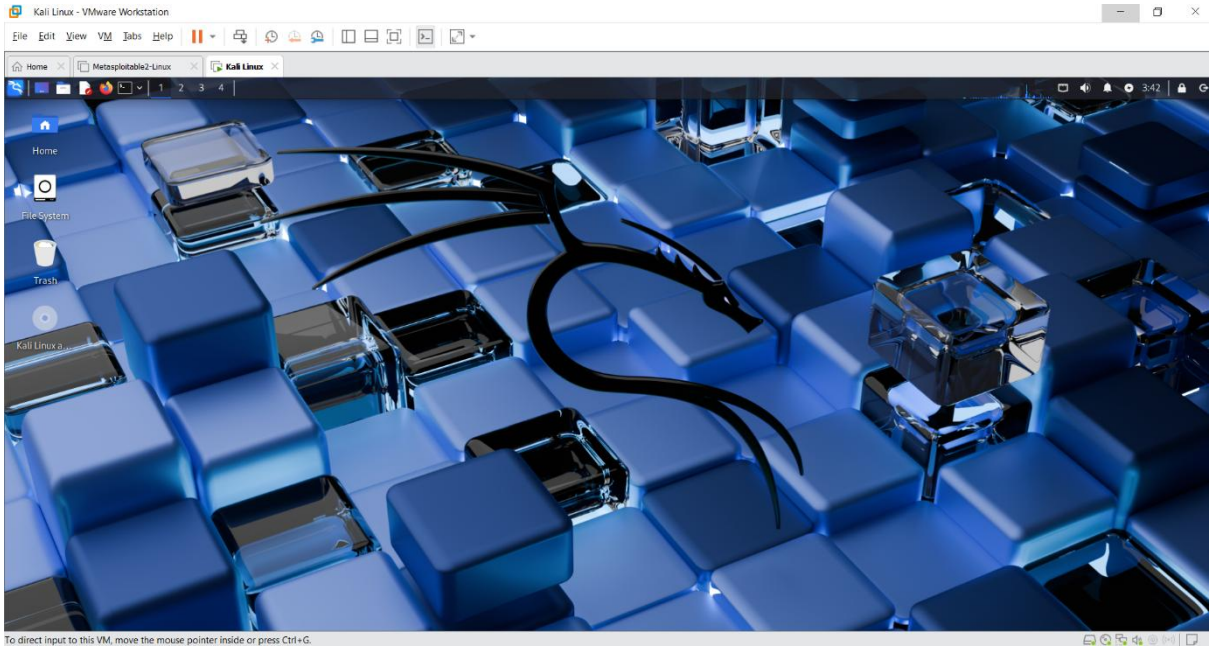
To create a safe and legal lab environment for conducting vulnerability assessments.

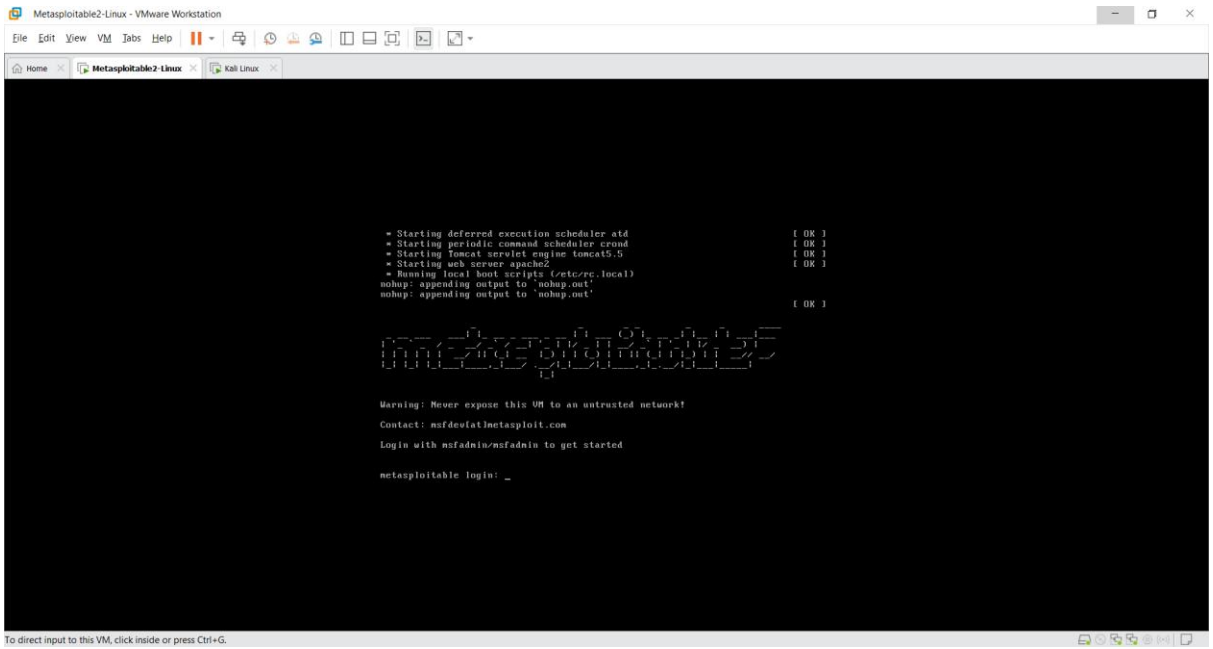
**2.2 Environment Details**

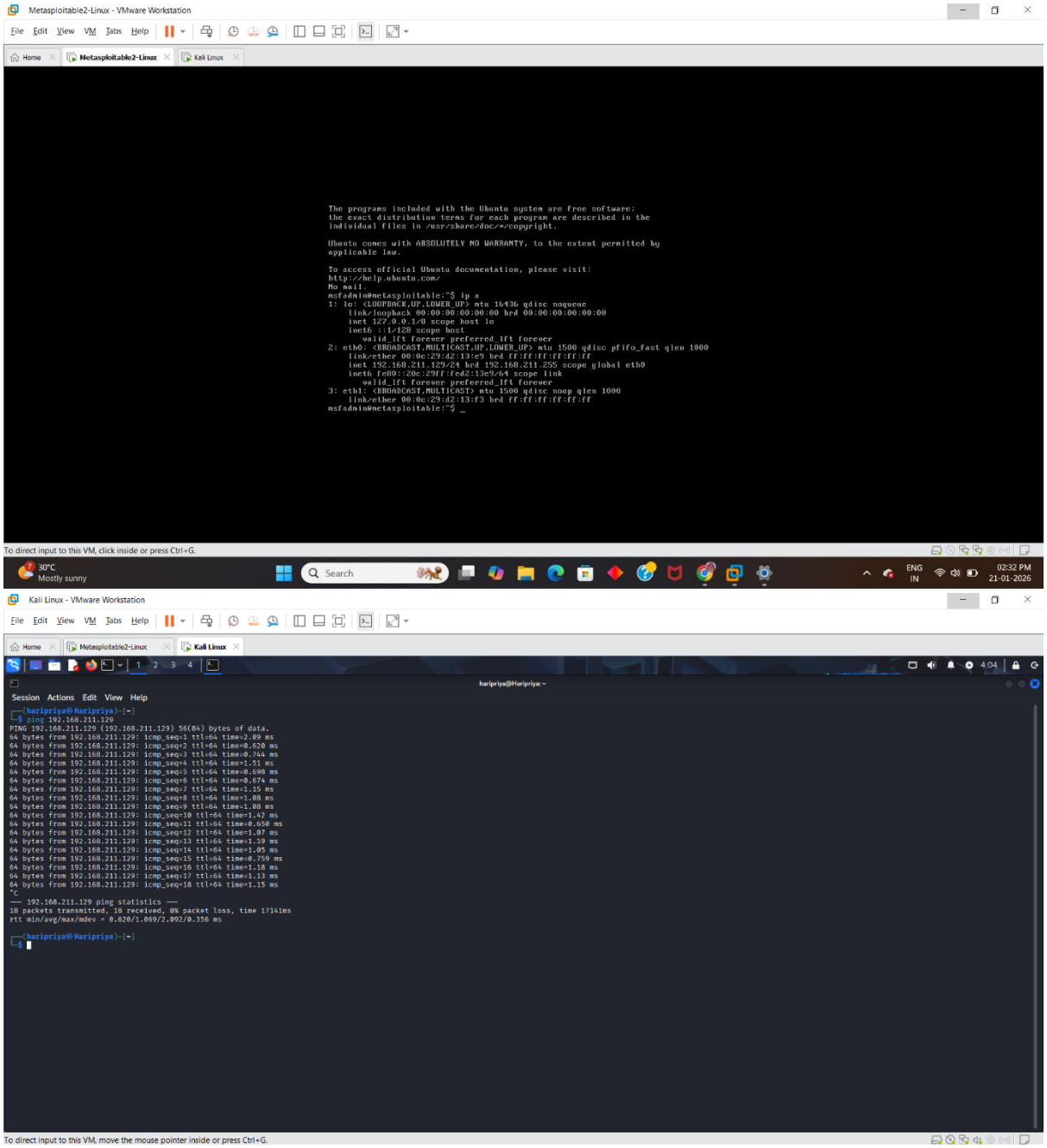
| Component | Description |
|---|---|
| Attacker Machine | Kali Linux |
| Target Machine | Metasploitable 3 |
| Virtualization Tool | Oracle VirtualBox |
| Network Type | Host-Only / NAT Network |

**2.3 Setup Steps**

1. Kali Linux was installed and updated.
2. Metasploitable 3 was downloaded from GitHub.
3. Both virtual machines were configured in VirtualBox.
4. Network connectivity between attacker and target was verified.


Kali Linux - VMware Workstation
File Edit View VM Tabs Help
Home
Metasploitable2-Linux
Kali Linux
Home
File System
Trash
To direct input to this VM, move the mouse pointer inside or press Ctrl+G.


Metasploitable2-Linux - VMware Workstation
File Edit View VM Tabs Help
Home
Metasploitable2-Linux
Kali Linux
* Starting deferred execution scheduler atd [ OK ]
* Starting periodic command scheduler crond [ OK ]
* Starting Tomcat servlet engine tomcat5.5 [ OK ]
* Starting web server apache2 [ OK ]
* Running local boot scripts (/etc/rc.local) [ OK ]
nohup: appending output to `nohup.out'
nohup: appending output to `nohup.out'
Warning: Never expose this VM to an untrusted network!
Contact: msfdev[at]metasploit.com
Login with msfadmin/msfadmin to get started
metasploitable login: _
To direct input to this VM, click inside or press Ctrl+G.




## 3. Vulnerability Scanning

### 3.1 Tools Used

- OpenVAS (Greenbone Vulnerability Manager)
- Nikto Web Server Scanner

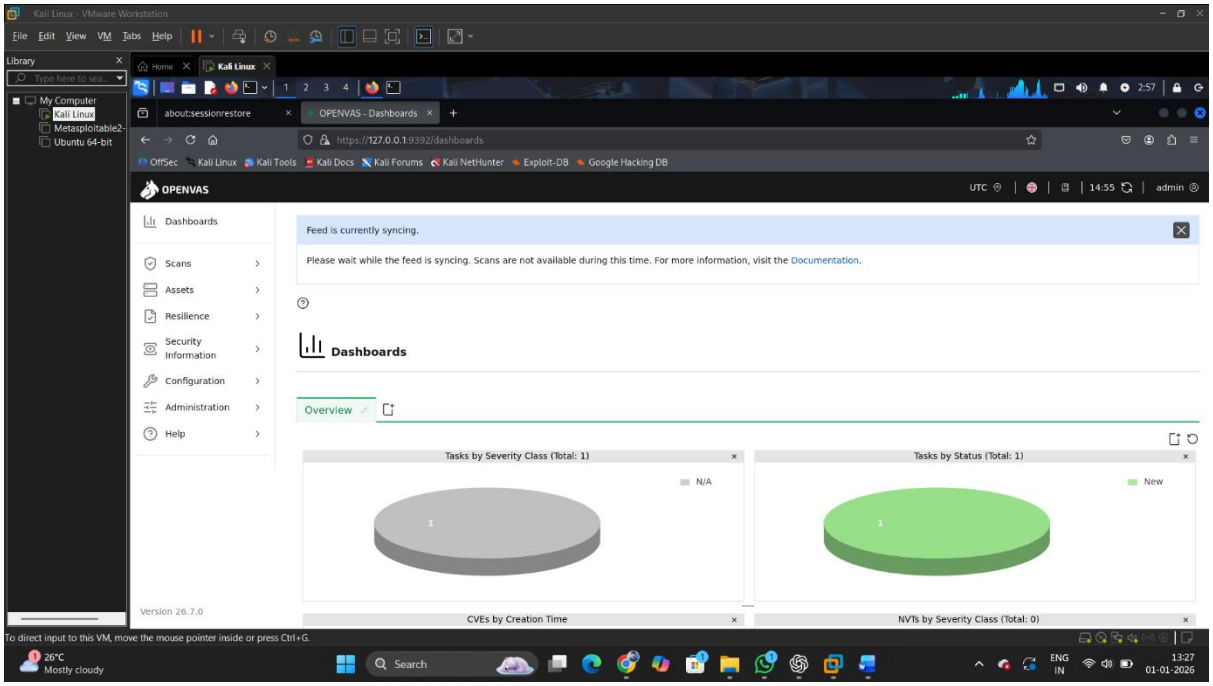
### 3.2 OpenVAS Scan

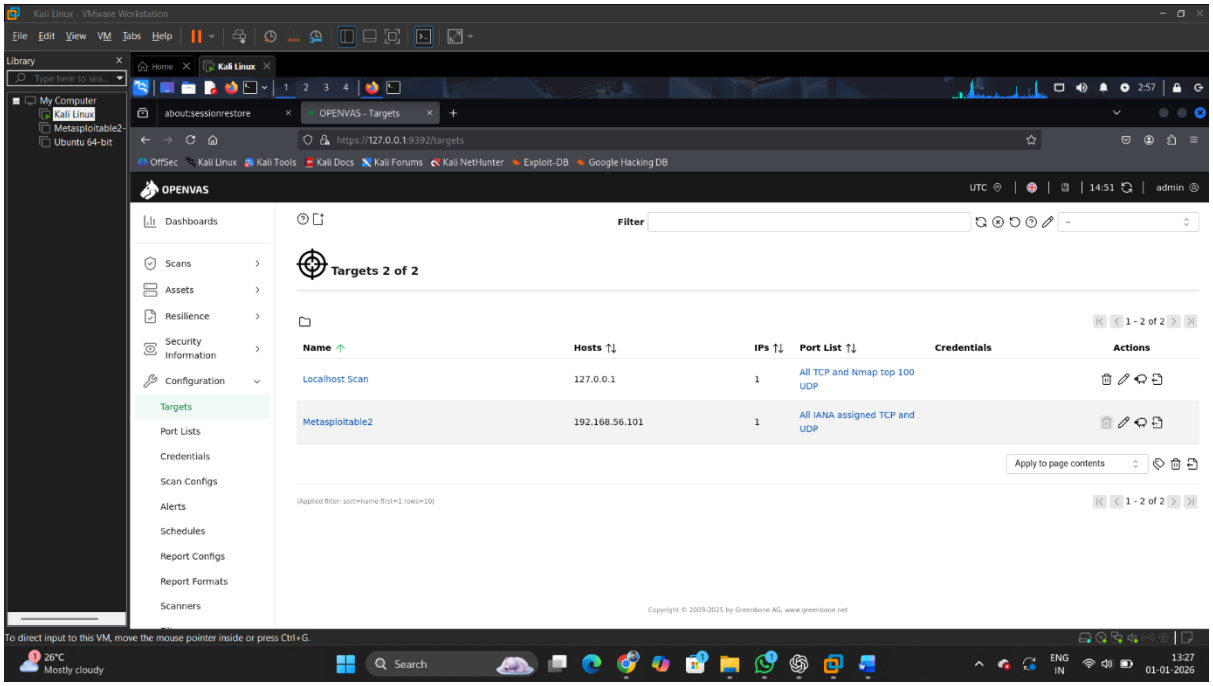
OpenVAS was launched using the following command:
sudo openvas-start

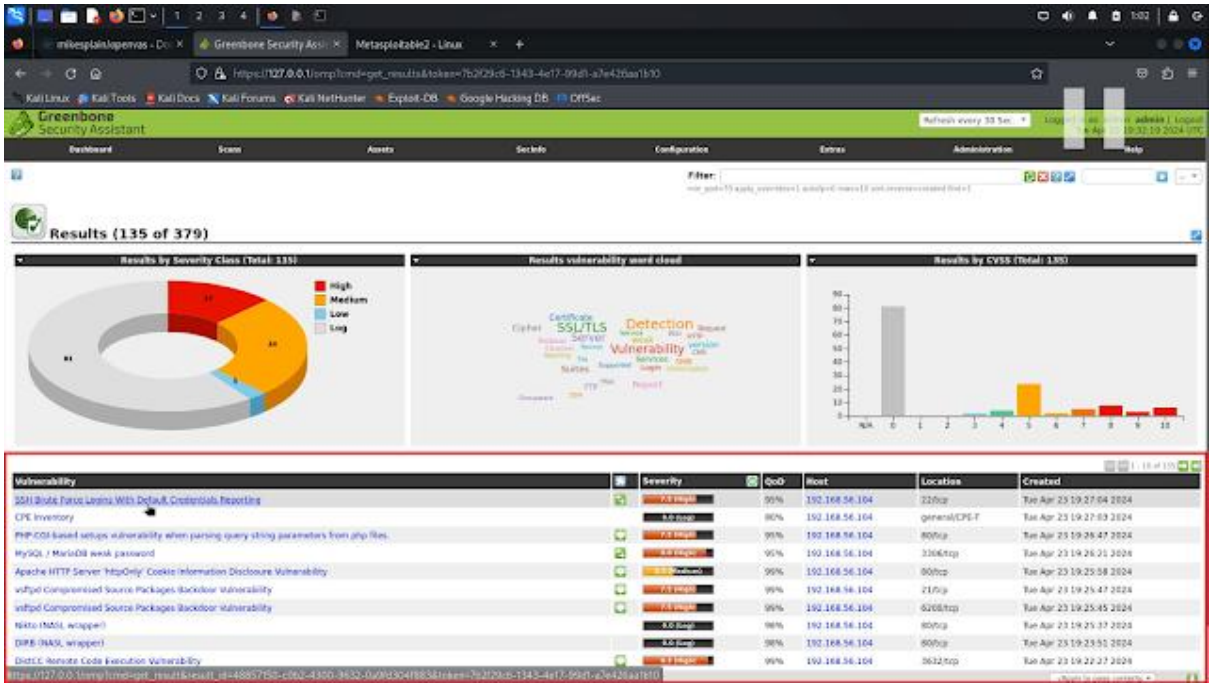
The OpenVAS web interface was accessed through the browser, and a **Full and Fast scan** was performed against the Metasploitable 3 IP address.

**Analysis Performed:**

- Identified critical, high, medium, and low vulnerabilities
- Reviewed CVSS scores and CVE IDs
- Focused on high-risk services

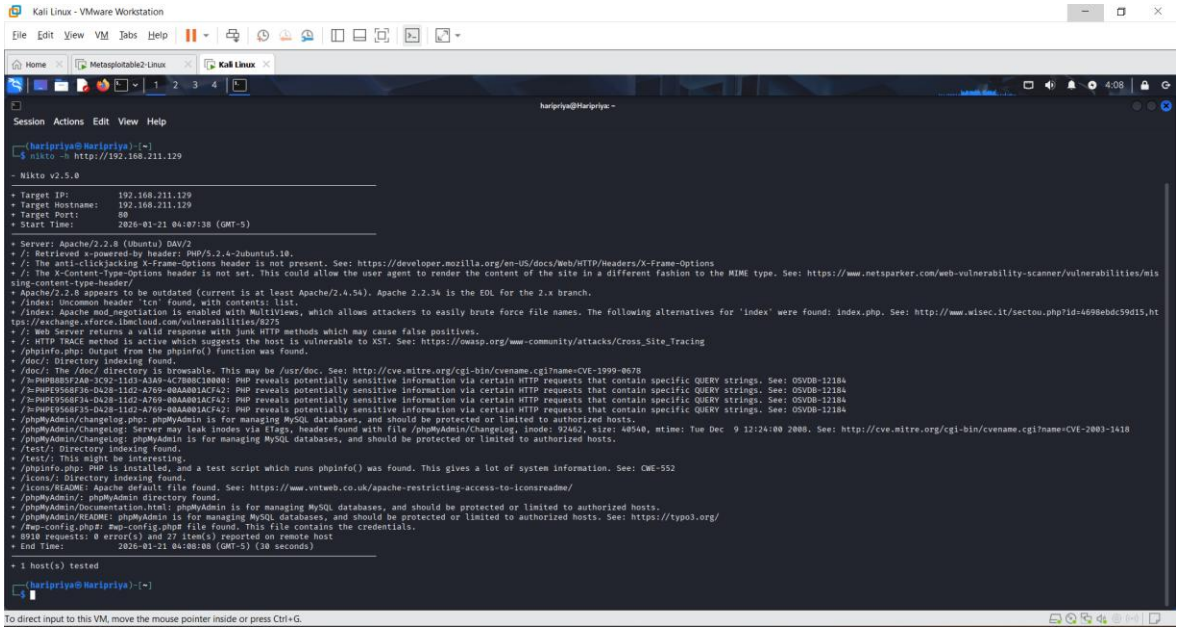





### 3.3 Nikto Web Scan

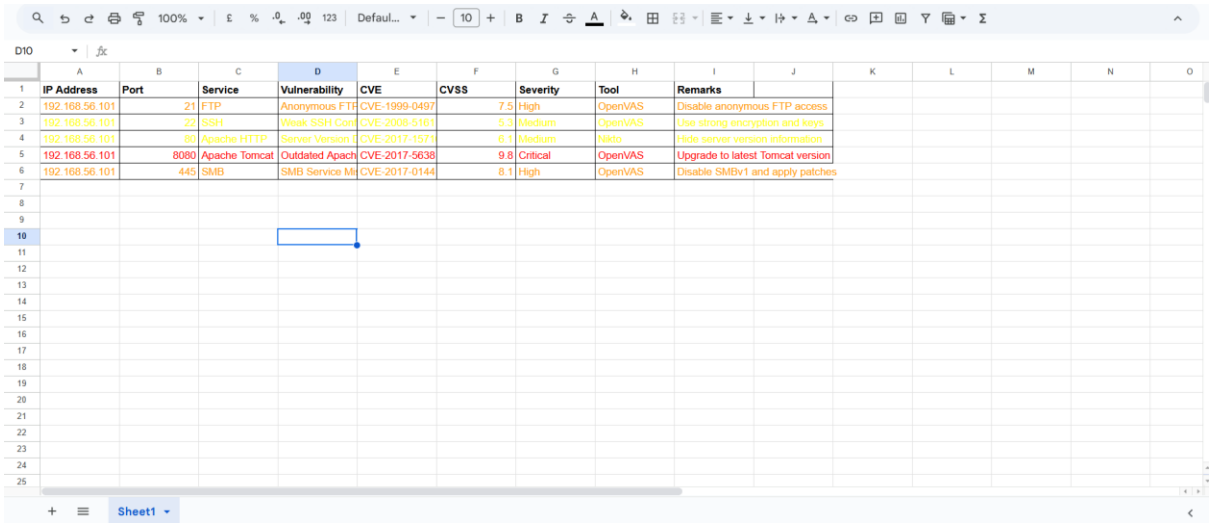
Nikto was used to scan the web server hosted on Metasploitable 3:
nikto -h http://<target-ip>
Nikto identified multiple web-related issues, including outdated server components and insecure HTTP headers.




## 4. Documentation of Findings

### 4.1 Vulnerability Tracking

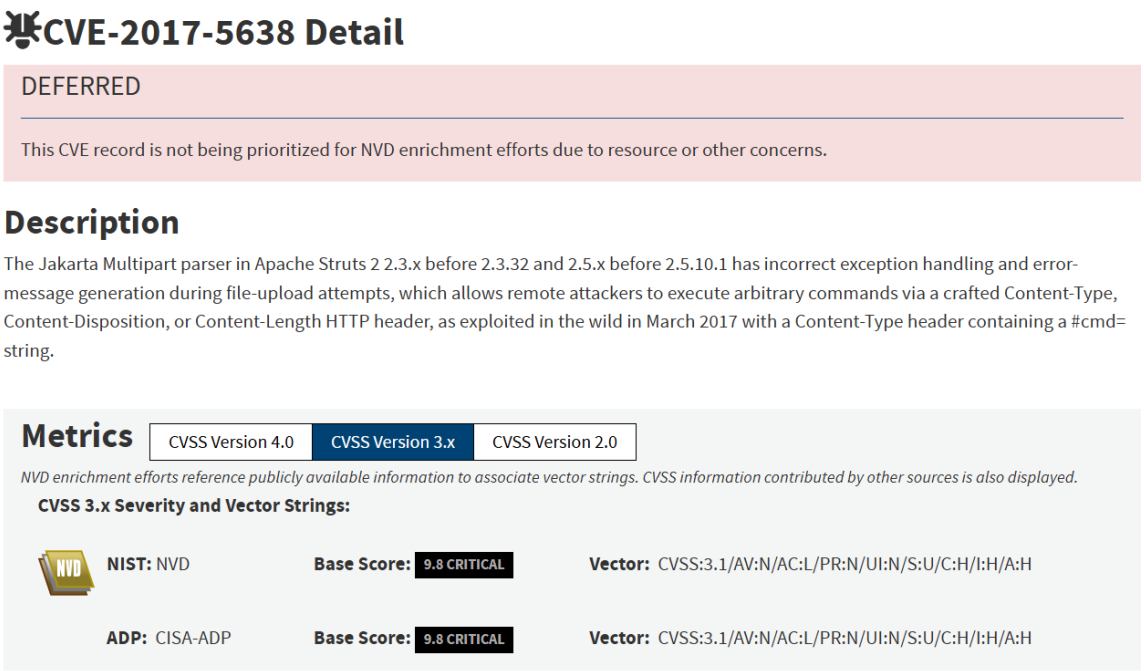
All identified vulnerabilities were recorded in a spreadsheet for tracking and analysis



| IP Address | Port | Service | Vulnerability | CVE | CVSS | Severity | Tool | Remarks |
|---|---|---|---|---|---|---|---|---|
| 192.168.56.101 | 21 | FTP | Anonymous FTP | CVE-1999-0497 | 7.5 | High | OpenVAS | Disable anonymous FTP access |
| 192.168.56.101 | 22 | SSH | Weak SSH Conf | CVE-2008-5161 | 5.3 | Medium | OpenVAS | Use strong encryption and keys |
| 192.168.56.101 | 80 | Apache HTTP | Server Version D | CVE-2017-1570 | 6.1 | Medium | Nikto | Hide server version information |
| 192.168.56.101 | 8080 | Apache Tomcat | Outdated Apach | CVE-2017-5638 | 9.8 | Critical | OpenVAS | Upgrade to latest Tomcat version |
| 192.168.56.101 | 445 | SMB | SMB Service Mi | CVE-2017-0144 | 8.1 | High | OpenVAS | Disable SMBv1 and apply patches |

## 5. Risk Assessment

### 5.1 CVSS Scoring



**CVE-2017-5638 Detail**

DEFERRED

This CVE record is not being prioritized for NVD enrichment efforts due to resource or other concerns.

**Description**

The Jakarta Multipart parser in Apache Struts 2 2.3.x before 2.3.32 and 2.5.x before 2.5.10.1 has incorrect exception handling and error-message generation during file-upload attempts, which allows remote attackers to execute arbitrary commands via a crafted Content-Type, Content-Disposition, or Content-Length HTTP header, as exploited in the wild in March 2017 with a Content-Type header containing a #cmd= string.

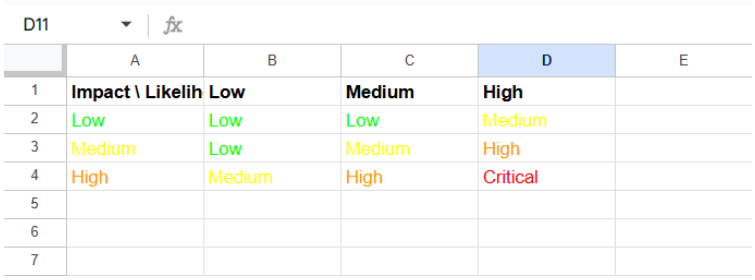
**Metrics** CVSS Version 4.0 | CVSS Version 3.x | CVSS Version 2.0

*NVD enrichment efforts reference publicly available information to associate vector strings. CVSS information contributed by other sources is also displayed.*

**CVSS 3.x Severity and Vector Strings:**

| | | | |
|---|---|---|---|
| NVD | **NIST:** NVD | **Base Score:** 9.8 CRITICAL | **Vector:** CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H |
| | **ADP:** CISA-ADP | **Base Score:** 9.8 CRITICAL | **Vector:** CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H |

### 5.2 Risk Matrix (3×3)



| | A | B | C | D | E |
|---|---|---|---|---|---|
| 1 | **Impact \ Likelih** | **Low** | **Medium** | **High** | |
| 2 | Low | Low | Low | Medium | |
| 3 | Medium | Low | Medium | High | |
| 4 | High | Medium | High | Critical | |
| 5 | | | | | |
| 6 | | | | | |
| 7 | | | | | |

## 6. Remediation Recommendations

### 6.1 General Recommendations

To reduce the overall security risk and improve the system's security posture, the following general remediation measures are recommended:

- Patch outdated software and services regularly to eliminate known vulnerabilities.
- Disable unnecessary open ports and services to reduce the attack surface.
- Apply secure configuration standards, such as CIS Benchmarks, to enforce industry-recognized security best practices.
- Implement regular vulnerability scanning and patch management processes.
- Monitor system logs and enable intrusion detection where applicable.

### 6.2 Remediation

#### 6.2.1 Outdated Apache Tomcat

**Issue:**

The Apache Tomcat service was found to be running an outdated version, which exposes the system to known vulnerabilities.

**Recommended Fix:**

- Upgrade Apache Tomcat to the latest stable version.
- Remove default or unused Tomcat applications.
- Restrict access to the Tomcat Manager interface.

**Patch / Vendor Reference:**

- https://tomcat.apache.org/security.html
- https://tomcat.apache.org/download-10.cgi

#### 6.2.2 Weak HTTP Security Headers

**Issue:**

The web server was missing important HTTP security headers, increasing exposure to attacks such as clickjacking and cross-site scripting (XSS).

**Recommended Fix:**

Configure the following security headers in the Apache configuration file:

Header always set X-Frame-Options "DENY"
Header always set X-Content-Type-Options "nosniff"
Header always set X-XSS-Protection "1; mode=block"
Header always set Content-Security-Policy "default-src 'self'"

**Patch / Best Practice Reference:**

- https://owasp.org/www-project-secure-headers/
- https://httpd.apache.org/docs/current/mod/mod_headers.html

### 6.2.3 Open Unused Ports

**Issue:**

Multiple open ports were detected that are not required for business operations, increasing the attack surface.

**Recommended Fix:**

- Identify unused services and disable them.
- Apply firewall rules to restrict network access.

**Example Configuration (Linux Firewall):**

sudo ufw deny 21
sudo ufw deny 23
sudo ufw enable

**Reference:**

- https://www.cisecurity.org/cis-benchmarks
- https://www.nist.gov/cyberframework

---

## 8. Conclusion

This practical assessment demonstrates how **free and open-source tools** can effectively identify and assess security vulnerabilities. By combining vulnerability scanning, risk analysis, and professional documentation, a comprehensive security assessment can be performed even without commercial tools. This exercise reinforces real-world cybersecurity practices and aligns with industry standards.

---