



Vulnerability Assessment and Penetration Testing (VAPT) Report

Title

Vulnerability Assessment and Penetration Testing (VAPT) of Metasploitable Using Open-Source Tools

Prepared By

Atharva Dave

Tools Used

- Kali Linux
 - Nmap
 - OpenVAS (Greenbone Vulnerability Manager)
 - Nikto
 - Google Sheets (Risk Tracking)
 - VMware Workstation
-

1. Executive Summary

This report presents the results of a Vulnerability Assessment and Penetration Testing (VAPT) exercise performed on an intentionally vulnerable system (Metasploitable). The objective was to identify security vulnerabilities using free and open-source tools, analyze their risks, and recommend remediation steps.

The assessment identified multiple **Critical and High severity vulnerabilities**, including outdated services, exposed network services, weak configurations, and web application vulnerabilities. Successful exploitation of these vulnerabilities could lead to **remote code execution, unauthorized access, credential theft, and full system compromise**.

2. Scope and Objectives

2.1 Scope

- **Target System:** Metasploitable Virtual Machine
- **Target IP Address:** 192.168.227.129
- **Assessment Type:** Vulnerability Assessment and Limited Penetration Testing

- **Out of Scope:** Denial of Service (DoS) and Social Engineering attacks

2.2 Objectives

- Identify vulnerabilities without using paid tools
 - Perform network and web vulnerability scanning
 - Assess risks using CVSS scoring
 - Document findings and remediation steps
-

3. Methodology (VAPT Approach)

The assessment followed a structured VAPT methodology aligned with **NIST**, **OWASP**, and industry best practices.

3.1 Planning

- Defined scope and rules of engagement
- Selected open-source tools
- Configured an isolated lab environment

3.2 Discovery

- Network and service discovery using **Nmap**
- Identification of exposed ports and running services

3.3 Vulnerability Assessment

- Automated vulnerability scanning using **OpenVAS**
- Web server scanning using **Nikto**

3.4 Risk Assessment

- CVSS-based risk scoring
- Risk prioritization using a 3×3 risk matrix

3.5 Reporting

- Documentation of findings
 - Remediation recommendations
 - Evidence-based reporting
-

4. Lab Environment and Network Setup

4.1 Environment Details

Machine	Role	IP Address
Kali Linux	Attacker	192.168.227.128
Metasploitable	Target	192.168.227.129
OpenVAS	Vulnerability Scanner	192.168.227.131

4.2 Network Configuration

- Hypervisor: VMware Workstation
- Network Type: Host-Only (Isolated Testing Environment)

5. Connectivity Verification

To verify network connectivity between the attacker and target machines, ICMP ping tests were performed.

Ping Test Results

```
ping 192.168.227.128
ping 192.168.227.129
```

Result:

- 0% packet loss
- Low latency observed

```

kali@kali:~$ ping 192.168.227.128
PING 192.168.227.128 (192.168.227.128) 56(84) bytes of data:
64 bytes from 192.168.227.128: icmp_seq=1 ttl=64 time=1.38 ms
64 bytes from 192.168.227.128: icmp_seq=2 ttl=64 time=0.076 ms
64 bytes from 192.168.227.128: icmp_seq=3 ttl=64 time=0.059 ms
^C
--- 192.168.227.128 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2042ms
rtt min/avg/max/mdev = 0.059/0.503/1.375/0.616 ms

kali@kali:~$ ping 192.168.227.129
PING 192.168.227.129 (192.168.227.129) 56(84) bytes of data:
64 bytes from 192.168.227.129: icmp_seq=1 ttl=64 time=5.30 ms
64 bytes from 192.168.227.129: icmp_seq=2 ttl=64 time=0.633 ms
64 bytes from 192.168.227.129: icmp_seq=3 ttl=64 time=0.560 ms
^C
--- 192.168.227.129 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2039ms
rtt min/avg/max/mdev = 0.560/2.162/5.295/2.215 ms

```

- Ping results from Kali to Metasploitable

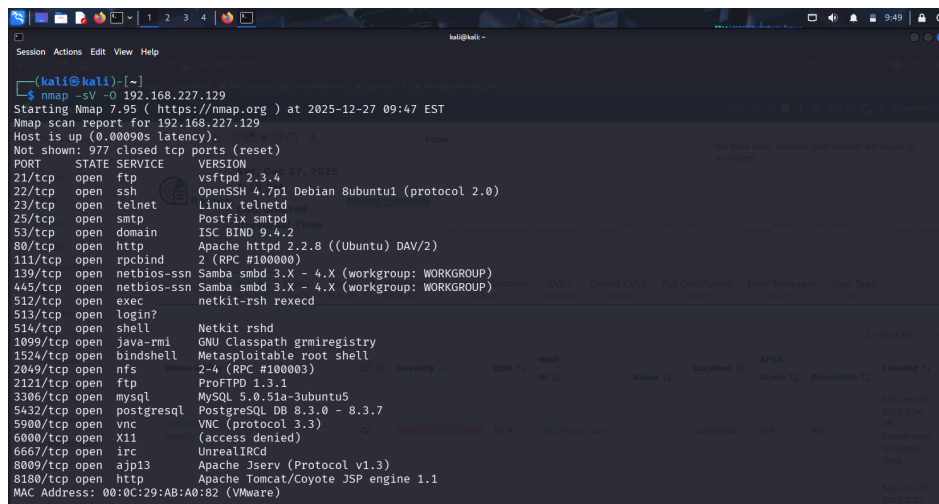
6. Discovery Phase – Nmap Scan

6.1 Nmap Command Used

```
nmap -sV -O 192.168.227.129
```

6.2 Key Findings

- Multiple open ports detected (FTP, SSH, HTTP, SMB, MySQL, Tomcat, IRC)
- Several outdated and vulnerable services running
- Presence of a root bind shell on port 1524



```
(kali@kali) [~]
$ nmap -sV -O 192.168.227.129
Starting Nmap 7.95 ( https://nmap.org ) at 2025-12-27 09:47 EST
Nmap scan report for 192.168.227.129
Host is up (0.00090s latency).
Not shown: 977 closed tcp ports (reset)
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
23/tcp    open  telnet
25/tcp    open  smtp
53/tcp    open  domain
80/tcp    open  http
111/tcp   open  rpcbind
139/tcp   open  netbios-ssn
445/tcp   open  netbios-ssn
512/tcp   open  exec
513/tcp   open  login?
514/tcp   open  shell
1099/tcp  open  java-rmi
1524/tcp  open  bindshell
2049/tcp  open  nfs
2121/tcp  open  ftp
3306/tcp  open  mysql
5432/tcp  open  postgresql
5900/tcp  open  vnc
6000/tcp  open  X11
6667/tcp  open  irc
8009/tcp  open  ajp13
8180/tcp  open  http
MAC Address: 00:0C:29:AB:A0:82 (VMware)
```

- Nmap scan output

7. Web Vulnerability Scanning – Nikto

7.1 Nikto Command Used

```
nikto -h http://192.168.227.129
```

7.2 Key Findings

- Outdated Apache web server
- Missing security headers
- Directory listing enabled
- Exposed phpinfo() page
- phpMyAdmin publicly accessible

```
Session Actions Edit View Help
Nmap done: 1 IP address (1 host up) scanned in 68.26 seconds

(kali@kali)-[~]
$ nikto -h http://192.168.227.129

- Nikto v2.5.0

+ Target IP: 192.168.227.129
+ Target Hostname: 192.168.227.129
+ Target Port: 80
+ Start Time: 2025-12-27 09:49:43 (GMT-5)

+ Server: Apache/2.2.8 (Ubuntu) DAV/2
+ /: Retrieved x-powered-by header: PHP/5.2.4-2ubuntu5.10.
+ /: The anti-clickjacking X-Frame-Options header is not present. See: https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-Frame-Options
+ /: The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in a different fashion to the MIME type. See: https://www.netsparker.com/web-vulnerability-scanner/vulnerabilities/missing-content-type-header/
+ Apache/2.2.8 appears to be outdated (current is at least Apache/2.4.54). Apache 2.2.34 is the EOL for the 2.x branch.
+ /index: Uncommon header 'ton' found, with contents: list.
+ /index: Apache mod_negotiation is enabled with MultiViews, which allows attackers to easily brute force file names. The following alternatives for 'index' were found: index.php. See: http://www.wisec.it/sectou.php?id=4698ebdc59d15,https://exchange.xforce.ibmcloud.com/vulnerabilities/8275
+ /: Web Server returns a valid response with junk HTTP methods which may cause false positives.
+ /: HTTP TRACE method is active which suggests the host is vulnerable to XST. See: https://owasp.org/www-community/attacks/Cross_Site_Tracing
+ /phpinfo.php: Output from the phpinfo() function was found.
+ /doc/: Directory indexing found.
+ /doc/: The /doc/ directory is browsable. This may be /usr/doc. See: http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-1999-0678
+ /?=PHPBB5F2A0-3C92-11d3-A3A9-4C7B08C10000: PHP reveals potentially sensitive information via certain HTTP requests that contain specific QUERY strings. See: OSVDB-12184
+ /?=PHEP9568F36-D428-11d2-A769-00AA001ACF42: PHP reveals potentially sensitive information via certain HTTP requests that contain specific QUERY strings. See: OSVDB-12184

+ 1 host(s) tested
```

```
Session Actions Edit View Help

+ /doc/: Directory indexing found.
+ /doc/: The /doc/ directory is browsable. This may be /usr/doc. See: http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-1999-0678
+ /?=PHPBB5F2A0-3C92-11d3-A3A9-4C7B08C10000: PHP reveals potentially sensitive information via certain HTTP requests that contain specific QUERY strings. See: OSVDB-12184
+ /?=PHEP9568F36-D428-11d2-A769-00AA001ACF42: PHP reveals potentially sensitive information via certain HTTP requests that contain specific QUERY strings. See: OSVDB-12184
+ /?=PHEP9568F34-D428-11d2-A769-00AA001ACF42: PHP reveals potentially sensitive information via certain HTTP requests that contain specific QUERY strings. See: OSVDB-12184
+ /?=PHEP9568F35-D428-11d2-A769-00AA001ACF42: PHP reveals potentially sensitive information via certain HTTP requests that contain specific QUERY strings. See: OSVDB-12184
+ /phpMyAdmin/ChangeLog.php: phpMyAdmin is for managing MySQL databases, and should be protected or limited to authorized hosts.
+ /phpMyAdmin/ChangeLog: Server may leak inodes via ETags, header found with file /phpMyAdmin/ChangeLog, inode: 92462, size: 40540, mtime: Tue Dec 9 12:24:00 2008. See: http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2003-1418
+ /phpMyAdmin/ChangeLog: phpMyAdmin is for managing MySQL databases, and should be protected or limited to authorized hosts.
+ /test/: Directory indexing found.
+ /test/: This might be interesting.
+ /phpinfo.php: PHP is installed, and a test script which runs phpinfo() was found. This gives a lot of system information. See: CWE-552
+ /icons/: Directory indexing found.
+ /icons/README: Apache default file found. See: https://www.vntweb.co.uk/apache-restricting-access-to-iconsreadme/
+ /phpMyAdmin/: phpMyAdmin directory found.
+ /phpMyAdmin/Documentation.html: phpMyAdmin is for managing MySQL databases, and should be protected or limited to authorized hosts.
+ /phpMyAdmin/README: phpMyAdmin is for managing MySQL databases, and should be protected or limited to authorized hosts. See: https://types.org/
+ /#wp-config.php#: #wp-config.php# file found. This file contains the credentials.
+ 8910 requests: 0 error(s) and 27 item(s) reported on remote host
+ End Time: 2025-12-27 09:50:34 (GMT-5) (51 seconds)

+ 1 host(s) tested

(kali@kali)-[~]
```

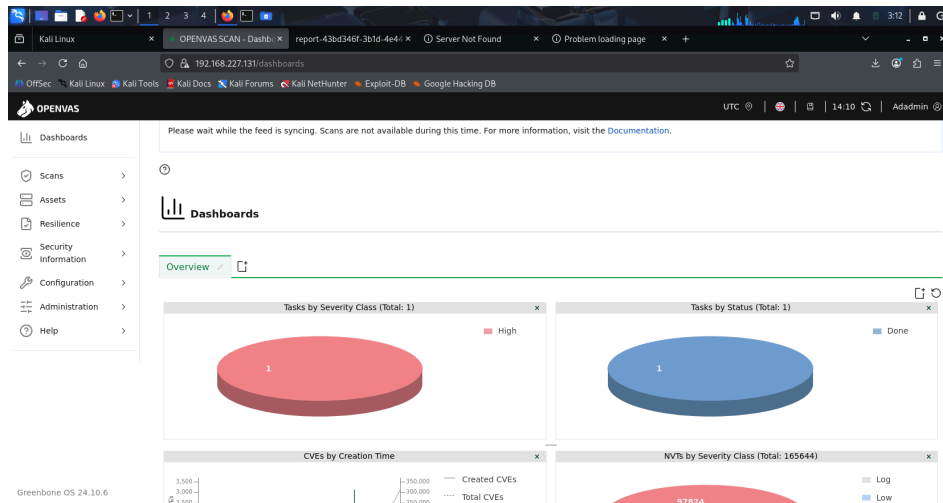
- Nikto scan output

8. Vulnerability Assessment – OpenVAS

8.1 Scan Overview

- Scan Type: Full and Fast
- Tool: OpenVAS (Greenbone)
- Target: Metasploitable

The OpenVAS scan identified multiple vulnerabilities across network services and web applications, including several **Critical** vulnerabilities.



- OpenVAS dashboard showing scan summary

9. Detailed Vulnerability Analysis (Sample)

TWiki Cross-Site Scripting and Command Execution Vulnerability

Vulnerability Name:

TWiki XSS and Command Execution Vulnerabilities

OID:

1.3.6.1.4.1.25623.1.0.800320

Installed Version:

01.Feb.2003

Fixed Version:

4.2.4

CVE IDs:

- CVE-2008-5304
- CVE-2008-5305

Description

The vulnerability exists due to improper input sanitization of `%URLPARAM{}%` and `%SEARCH{}%` variables. The `%SEARCH{}%` variable is directly evaluated using Perl `eval()`, allowing attackers to inject and execute arbitrary code.

Impact

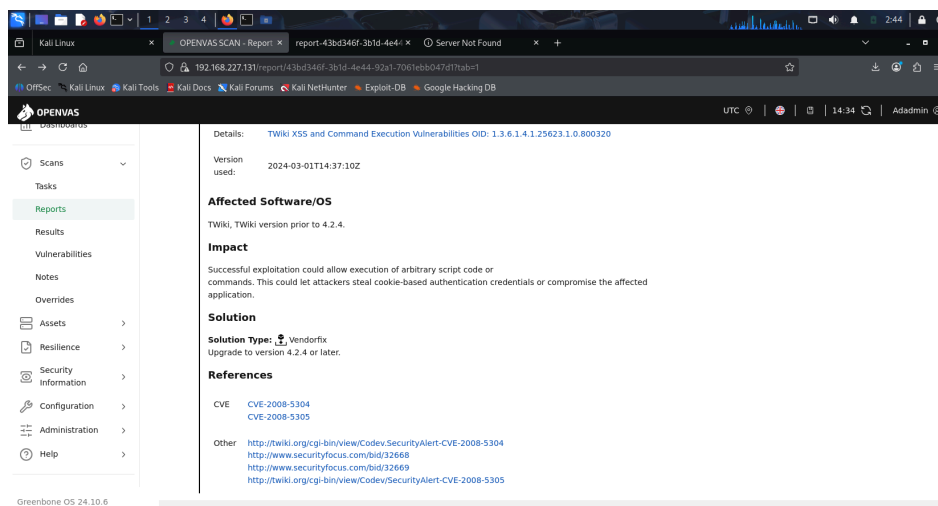
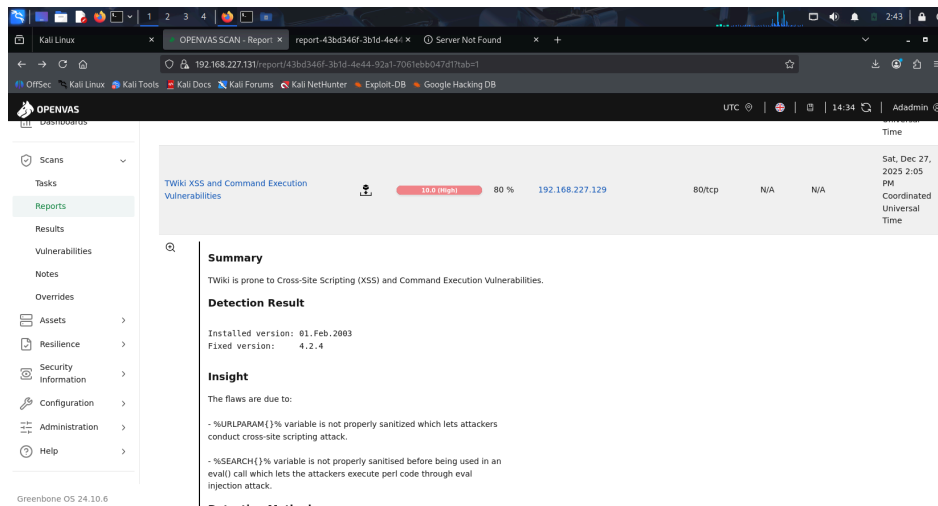
Successful exploitation could allow:

- Cross-Site Scripting (XSS)

- Remote command execution
- Theft of authentication cookies
- Complete application compromise

Remediation

Upgrade TWiki to version **4.2.4 or later**.



- OpenVAS detailed vulnerability page

10. Risk Assessment

10.1 CVSS Severity Classification

Severity	CVSS Score
Critical	9.0 – 10.0

Severity	CVSS Score
High	7.0 – 8.9
Medium	4.0 – 6.9
Low	0.1 – 3.9

10.2 Risk Matrix (3×3)

Likelihood	Impact	Risk
High	High	Critical
Medium	High	High
Low	Medium	Medium

10.3 Google Sheet Link:

https://docs.google.com/spreadsheets/d/1MBNVd9gAjG6ezBX_6rh664ZK71TcjPSRBkHvrHPZ8M/edit?usp=sharing

11. Remediation Recommendations

- Disable unused and insecure services (FTP, Telnet, Bind Shell)
- Upgrade outdated software and services
- Apply CIS Benchmarks for system hardening
- Restrict access to administrative interfaces
- Implement proper input validation for web applications
- Conduct regular vulnerability scans

12. Conclusion

The VAPT exercise demonstrated that the target system is highly vulnerable due to outdated software, insecure configurations, and exposed services. Immediate remediation is required to prevent exploitation. Regular vulnerability assessments, patch management, and secure configuration practices are essential to maintaining a secure environment.

13. References

- NIST SP 800-115
- OWASP Top 10
- Greenbone OpenVAS Documentation
- NVD CVSS Calculator
- CIS Benchmarks