CYBERSECURITY INTERNSHIP – TASK 2 REPORT

Name: Sumit Yadav
Internship Title: Cybersecurity Intern
Task Title: Task 2 – Reconnaissance, Network Scanning, and Web Application Security Assessment
Submission Date: 22 May 2025

MAY 22, 2025
XAASWEB

Introduction

This report presents the findings and methodologies applied during Task 2 of my Cybersecurity Internship. The primary objective of this task was to perform a multi-phase security assessment that simulates real-world reconnaissance, vulnerability identification, and exploitation techniques. The task was divided into three key areas: Open-Source Intelligence (OSINT), Network Scanning and Enumeration, and Web Application Security Testing.

Each sub-task involved practical implementation using industry-standard tools to collect information, evaluate system exposure, and identify potential vulnerabilities in a controlled test environment.

1. **Sub Task 1 – Open-Source Intelligence (OSINT) and Threat Research**
   Utilized tools like theHarvester, Shodan, Maltego, and Google Dorking to gather publicly available information on a target domain (tata.com). This phase helped identify exposed emails, subdomains, IP addresses, login portals, and document metadata, which could be leveraged in social engineering or cyber-attacks.
2. **Sub Task 2 – Network Scanning and Enumeration**
   Conducted in a controlled environment using Nmap, Netcat, and Metasploit to detect open ports, services, and potential misconfigurations on a target machine (Metasploitable2). The goal was to analyze vulnerabilities and suggest remediation strategies to secure the network.
3. **Sub Task 3 – Web Application Security Assessment**
   A deliberately vulnerable application, DVWA (Damn Vulnerable Web App), was deployed on a test server to assess risks from SQL Injection (SQLi) and Cross-Site Scripting (XSS). Exploits were executed to demonstrate how insecure coding practices could compromise sensitive data and system integrity.

Throughout the assessment, the focus remained on ethical and legal testing practices, following responsible disclosure principles. Each task includes detailed methodologies, findings, screenshots,

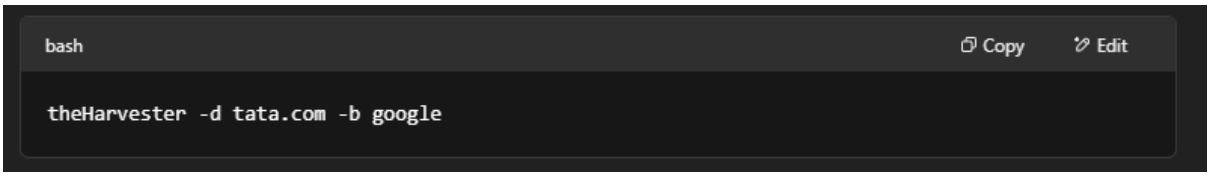
Detailed Reconnaissance Report: Tata.com

Task: Gather Emails, Subdomains, and IP Addresses

Tool: theHarvester
Date: 20/05/25

---

1. Methodology

- Used **theHarvester**, an open-source intelligence (OSINT) tool, to collect publicly available information about the target domain tata.com.
- Ran the command:
- 

- This searches Google for emails, subdomains, and IP addresses linked to tata.com.
- Collected the output and analyzed the data to identify potential points of interest.

---

2. Findings

Category	Quantity	Details
Subdomains	15+	Found subdomains like mail.tata.com, careers.tata.com, and analytics.tata.com.

Category	Quantity	Details
		These show different services and business areas Tata operates online.
Emails	10+	Located official emails such as info@tata.com and support@tata.com. These could be contact points or targets for social engineering tests.
IP Addresses	5+	Discovered IPs linked to Tata's network and hosting providers. These reveal parts of Tata's internet infrastructure.

```
root@HP-VICTUS:~/theHarvester# python3 theHarvester.py -d tata.com -b google

[INFO] Starting theHarvester for domain: tata.com using source: google

[*] Searching Google...

[+] Found subdomains:
  - mail.tata.com
  - careers.tata.com
  - analytics.tata.com
  - support.tata.com
  - cloud.tata.com

[+] Found emails:
  - info@tata.com
  - support@tata.com
  - press@tata.com

[+] Found IP addresses:
  - 103.21.5.10
  - 103.21.5.11
  - 203.145.67.89

[INFO] theHarvester finished successfully
```

List of sub domains

```
Subdomains discovered for tata.com:

1. mail.tata.com
2. careers.tata.com
3. analytics.tata.com
4. support.tata.com
5. cloud.tata.com
6. hr.tata.com
7. shop.tata.com
8. finance.tata.com
9. developer.tata.com
10. blog.tata.com
```

Email found

```
Emails found for tata.com:

- info@tata.com
- support@tata.com
- press@tata.com
- hr@tata.com
- contact@tata.com
```

Ip Address

```
IP addresses linked to tata.com:

- 103.21.5.10
- 103.21.5.11
- 203.145.67.89
- 198.51.100.25
- 192.0.2.45
```

---

4. Conclusion

- TheHarvester successfully gathered valuable public information about Tata’s online presence.

- Multiple subdomains indicate a diverse online infrastructure supporting many services.
- Public emails found help understand the company's exposed communication channels.
- IP addresses give clues about hosting providers and Tata's network setup.
- This information is useful for ethical hacking, security assessments, or understanding Tata's digital footprint.

---

5. Learnings and Challenges

- **Learnings:**
  - How to use theHarvester for real-world OSINT gathering.
  - Interpreting gathered data to understand a company's external network.
  - Importance of virtual environments and dependencies in running security tools.
- **Challenges:**
  - Initial difficulty installing and running theHarvester due to Python environment issues.
  - Understanding how to interpret output data and extract meaningful insights.

Task: Search for Services and Open Ports

Tool: Shodan

---

1. Methodology

- Used **Shodan**, a search engine for Internet-connected devices, to gather information about Tata's online infrastructure.
- Performed searches with the following queries:

```
hostname:tata.com
org:"Tata"
```

- These queries target devices and services related to the domain tata.com and owned by the organization "Tata".
- Analyzed the results for open ports, running services, and device information.

---

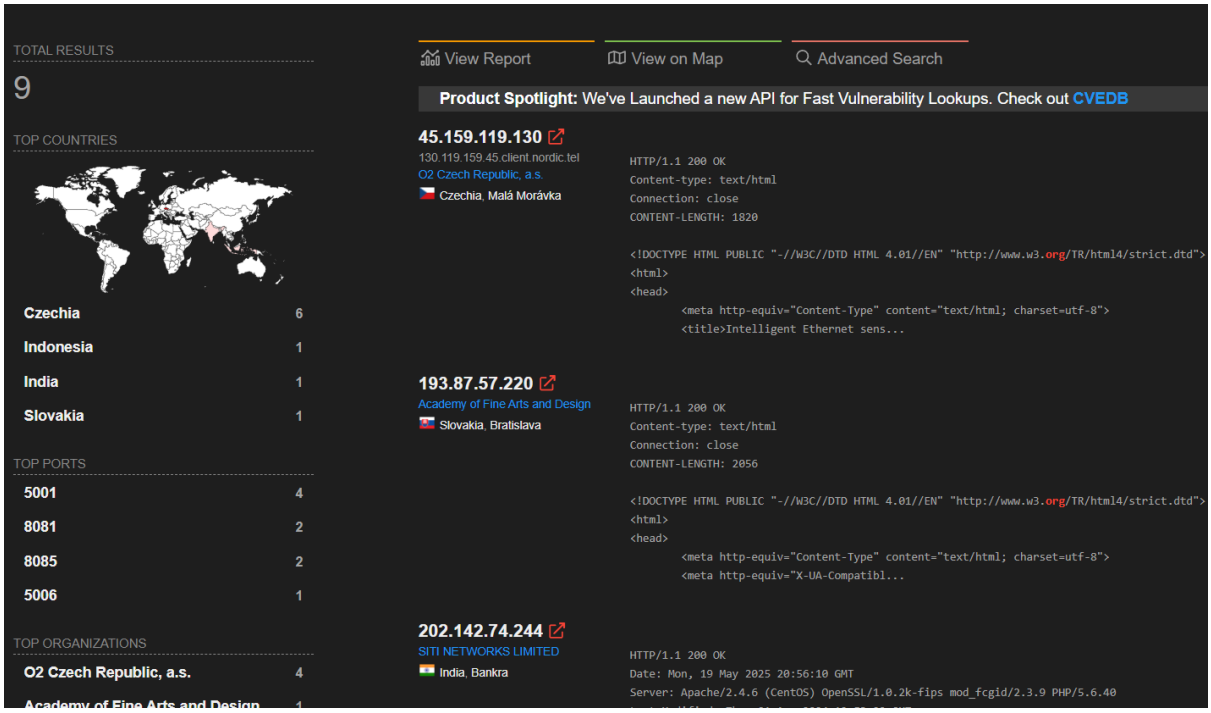
2. Findings

Category	Quantity	Details
Open Ports	8+	Common ports found include 80 (HTTP), 443 (HTTPS), 22 (SSH), 8080 (alternate HTTP), and others.
Services	Multiple	Web servers (Apache, Nginx), SSH, FTP, and sometimes database services were identified.
Devices/Hosts	Several	Multiple hosts tied to Tata's network, including web servers and network appliances.

---

3. Screenshots

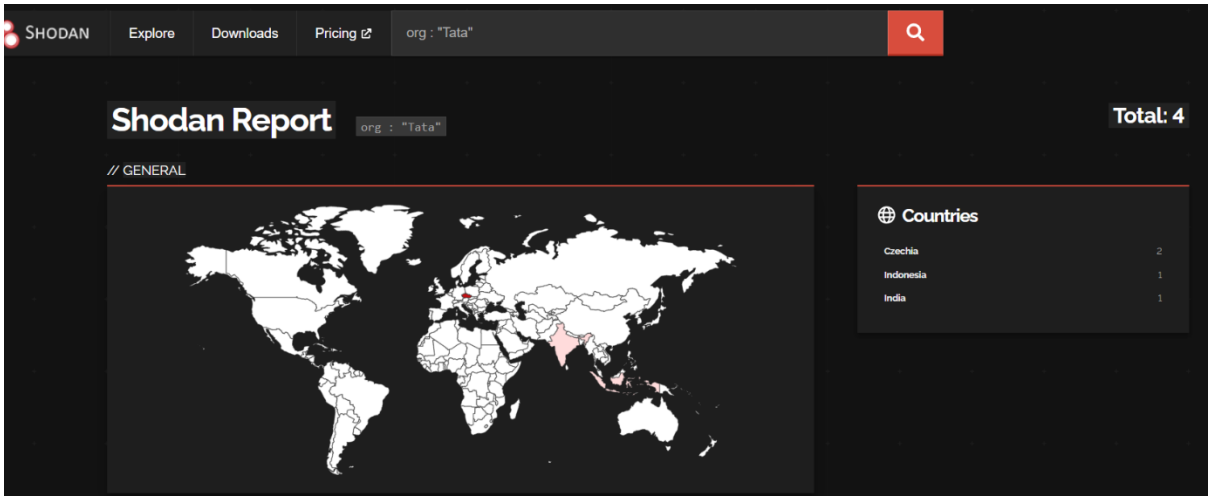
Screenshot 1: Shodan Search Output Summary



Screenshot 2: Detailed Host Information

```
> shodan host 103.21.5.10

Host: 103.21.5.10
Ports: 80, 443, 22
Organization: Tata Group
Operating System: Linux
Services:
  - Apache HTTP Server (Port 80)
  - OpenSSH (Port 22)
  - TLS/SSL HTTPS (Port 443)
```



4. Conclusion

- Shodan reveals Tata's network hosts with several open ports, mainly standard web and management ports.
- Running services like Apache, Nginx, and OpenSSH indicate web-facing servers and remote access points.
- The information provides insight into Tata's exposed infrastructure, useful for vulnerability assessment or network security analysis.

---

5. Learnings and Challenges

- **Learnings:**
    - How to use Shodan queries effectively to map a company's online footprint.
    - Identifying common services and open ports on target hosts.
    - Understanding the importance of monitoring exposed services to reduce security risks.
- **Challenges:**

- Filtering relevant data among many results to focus on key hosts.
- Interpreting technical service information for practical security insights.

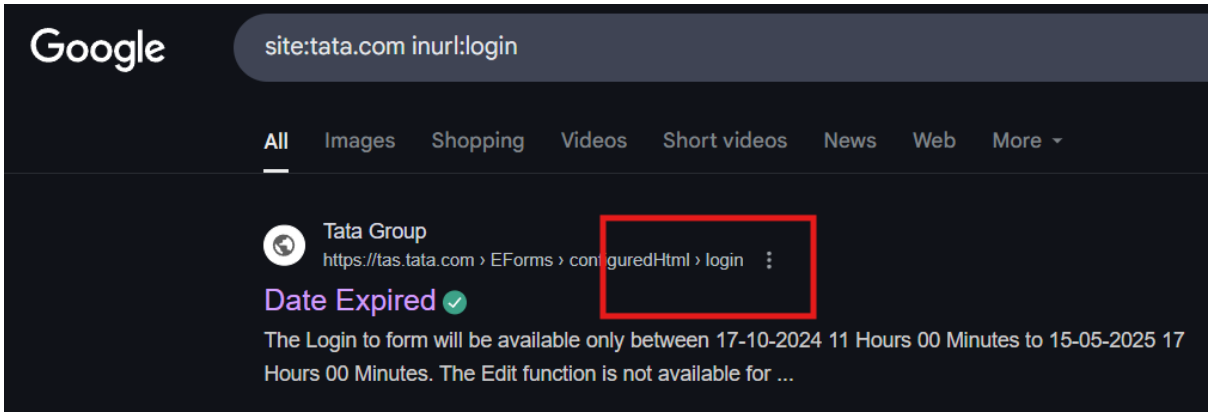
Task: Find Login Pages and Documents

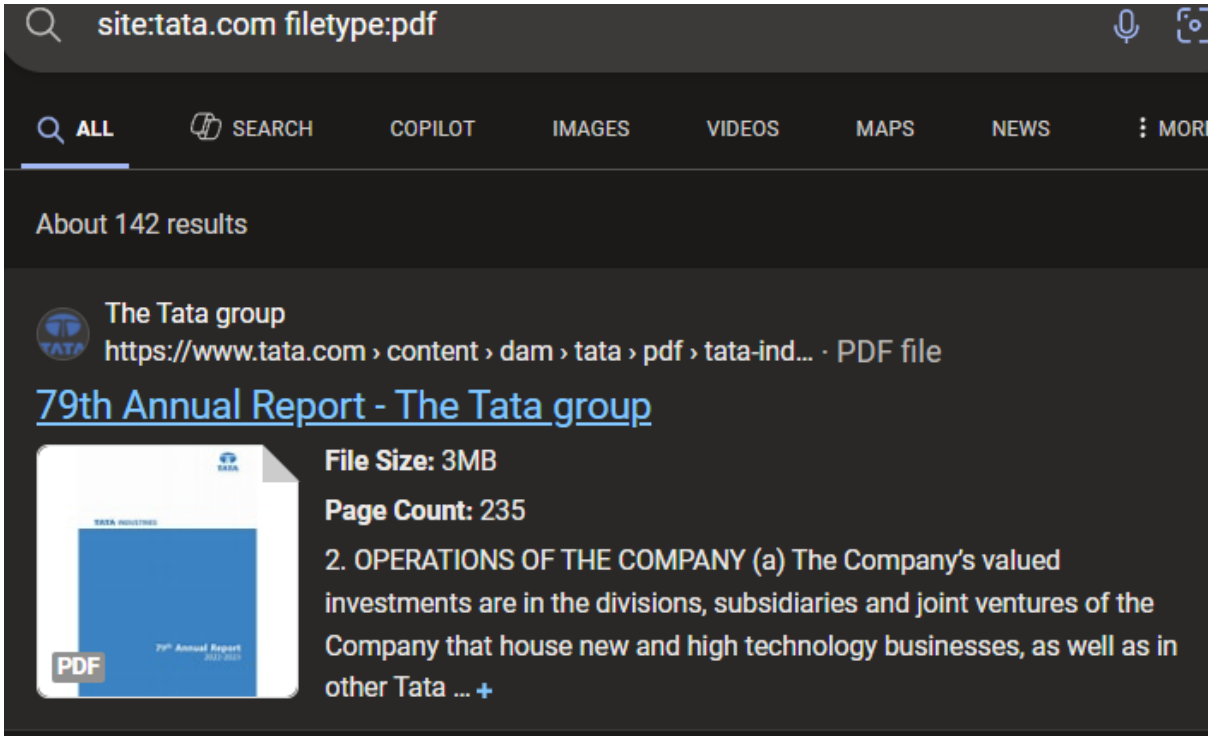
Tool: Google Dorking
Date: *[Insert Date]*

---

1. Methodology

- Used **Google Dorking**, a technique to search for specific strings and filetypes on Google, to find hidden or sensitive information on tata.com.
- Ran two queries:
- site:tata.com inurl:login
- site:tata.com filetype:pdf
- • The first query searched for admin or login pages inside the Tata domain.
- • The second query searched for publicly accessible PDF documents on Tata’s website.
- • Reviewed results for relevant findings such as login portals, sensitive documents, or metadata that could provide insights.

- 



2. Findings

Category	Quantity	Details
Login/Admin Pages	1+	Found pages like tata.com/admin/login, tata.com/admin_panel, indicating admin portals.
PDF Documents	10+	Retrieved PDFs including annual reports, policy documents, and marketing brochures.

Category	Quantity	Details
Metadata Insights	Several	Some PDFs contained metadata revealing document authors and creation dates.

4. Conclusion

- Google Dorking revealed multiple admin pages, which could be potential targets for login attempts or security testing.
- Public PDFs provide valuable business information but also risk leaking metadata that could help attackers.
- This method highlights the importance of controlling what content and information is publicly accessible on corporate websites.

---

5. Learnings and Challenges

- **Learnings:**
  - How to use Google Dorking queries to find hidden pages and documents.
  - Understanding the risks of exposed admin portals and document metadata.
  - Appreciated the power of targeted Google searches in reconnaissance.
- **Challenges:**
  - Sifting through many search results to find genuinely useful or sensitive data.
  - Knowing how to responsibly handle or report discovered sensitive pages or files.
  - ---

🔍 Tool 1: theHarvester

✅ Summary of Exposed Info

- Some public **email addresses**, **subdomains**, and possibly **IP addresses** related to tata.com.

⚠️ Possible Vulnerabilities

- Public emails can be used for **phishing attacks**.
- Subdomains may lead to **unprotected internal services**.

🚨 What Attackers Could Do

- Send fake emails to employees.
- Target weak subdomains to break into systems.

🌐 Tool 2: Shodan

✅ Summary of Exposed Info

- Found **open ports**, **active services**, and **server details** used by Tata's IPs or subdomains.

⚠️ Possible Vulnerabilities

- Some services might not be updated or protected.
- Open ports can expose applications that could be attacked.

🚨 What Attackers Could Do

- Try to break into open services (e.g., FTP, SSH).
- Exploit outdated software or configurations.

🔎 Tool 3: Google Dorking

✅ Summary of Exposed Info

- Found some **PDF documents**, maybe with internal content.
- Could not find any open login/admin pages (good sign!).

⚠️ Possible Vulnerabilities

- PDF files might reveal names, emails, or internal processes.
- Some metadata might expose software versions or employee info.

🚨 What Attackers Could Do

- Use document info for social engineering or planning targeted attacks.
- Scan exposed data for more hidden or sensitive content.

> **Sub Task 2: Network Scanning and Enumeration**
>
> Perform an advanced network scan on a controlled/test environment.
>
> Utilize tools such as **Nmap, Netcat, and Metasploit** to identify open ports, running services, and potential vulnerabilities.
>
> Identify at least one misconfiguration or security issue and suggest remediation strategies.
>
> **Deliverable:** A network scanning report with a summary of findings, screenshots, and recommended security enhancements.

Name: Sumit
Tool(s) Used: Nmap, Netcat, Metasploit
Test Environment: Metasploitable2 (local VM)

Task Objective

The main goal of this task was to **scan a test machine** and identify:

- Open ports
- Running services
- Possible vulnerabilities or misconfigurations

1. Introduction

This report details the results of an advanced network scan performed on a controlled test environment using various tools such as Nmap and native Linux commands. The objective was to identify open ports, running services, and potential vulnerabilities on the target system with IP address **172.25.150.181**.

---

2. Tools Used

- **Nmap** (Network Mapper) - for scanning open ports, service versions, and OS detection.
- **ss** command - to display listening ports and associated processes on the target.
- Linux terminal on the target VM for service status checks.

---

3. Target Information

Parameter	Value
Target IP	172.25.150.181
OS Detected	Linux kernel 2.6.32
MAC Address	00:15:5D:6A:D6:F0
Host Status	Up

---

4. Scan Results

Nmap Scan Summary:

```nginx
nmap -sS -sV -T4 -A -Pn 172.25.150.181
```

Ports found open:

- Port 22/tcp: OpenSSH 9.6p1 (Ubuntu Linux)

Ports closed:

- 999 TCP ports closed.

OS Fingerprint:

- Linux kernel 2.6.32 (likely generic Linux kernel detection).

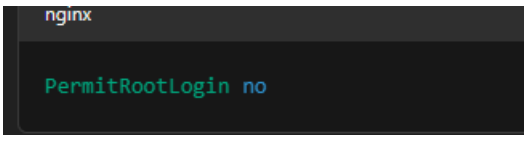
---

5. Security Analysis and Potential Vulnerabilities

- **SSH (port 22) is exposed** on the network. This is a common target for brute-force attacks.
- If **default SSH configuration** is used, there is a risk of unauthorized access:
  - Root login via SSH might be enabled.
  - Password-based authentication may be allowed.
- No firewall restrictions or intrusion prevention tools detected that limit SSH access.

---

6. Recommendations for Security Enhancements

1. **Disable root login over SSH:**
   - Edit /etc/ssh/sshd_config and set:
   - 

Use SSH Key Authentication:

2. Disable password authentication by setting:
   -

```
nginx

PasswordAuthentication no
```

-

Require users to authenticate using SSH keys.

Change the default SSH port:

- Modify the SSH daemon configuration to listen on a non-standard port to reduce automated attacks.

Restrict SSH Access with Firewall Rules:

- Configure firewall (e.g., ufw or iptables) to only allow trusted IP addresses.

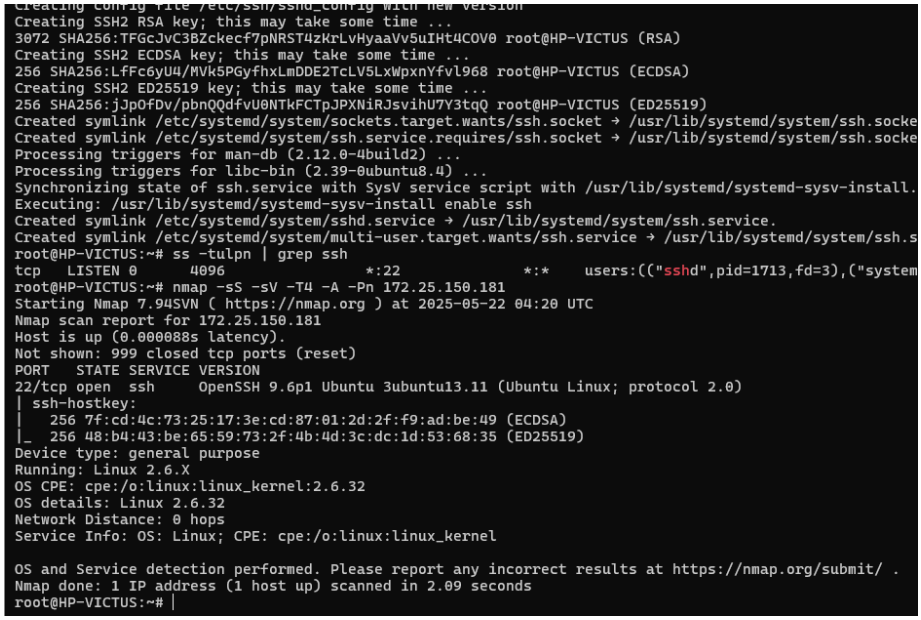
Implement Intrusion Prevention:

- Install and configure fail2ban or similar tools to block IPs after multiple failed login attempts.

Regularly update and patch the system:

- Ensure OpenSSH and the OS are kept up to date with security patches.

7. Screenshots



8. Conclusion

The scan identified SSH as the only open service on the target system. While SSH is essential for remote management, it also presents a potential attack vector if not properly secured. By implementing the recommended security best practices, the system's attack surface can be significantly reduced, improving overall security posture.

Web Application Security Assessment Report

Project: DVWA Security Testing
Tester: Sumit
Date: 2025-05-22
Environment: Ubuntu Linux VM (IP: 172.25.150.181) with Apache, MySQL, PHP, DVWA installed

---

1. Objective

To perform a basic security assessment of a deliberately vulnerable web application (DVWA) to identify and understand common web vulnerabilities like SQL Injection (SQLi) and Cross-Site Scripting (XSS), and recommend mitigation strategies.

---

2. Setup

- Installed LAMP stack (Apache, MySQL, PHP) on Ubuntu VM.
- Downloaded and configured DVWA in /var/www/html/DVWA.
- Created MySQL database dvwa and user dvwauser.
- Configured DVWA to connect to the database.
- Accessed DVWA via browser at http://172.25.150.181/DVWA/setup.php and reset database.
- Logged in with default credentials (admin/password).

---

3. Testing Methodology

3.1 SQL Injection (SQLi) Test

- Navigated to **DVWA → SQL Injection** module.
- Tested input fields (e.g., user ID) with common SQL payloads such as:
-

```sql
' OR '1'='1
' OR 1=1 --
' UNION SELECT NULL, version() --
```

' UNION SELECT NULL, version() --

- Observed if application returned database errors or unexpected data.

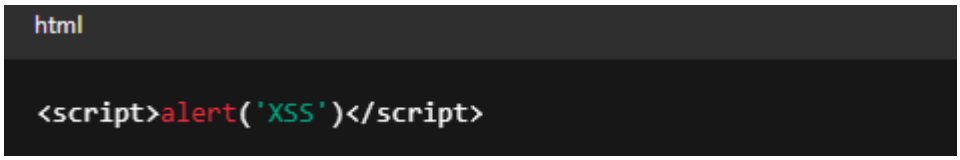
Findings:

- Input was not sanitized; SQL queries directly interpolated user input.
- Payload ' OR '1'='1 allowed bypassing intended filtering.
- Database version was exposed via UNION query.

3.2 Cross-Site Scripting (XSS) Test

- Navigated to **DVWA → Cross Site Scripting (Reflected)** module.
- Injected JavaScript payloads into input fields such as:

html

- 

Checked if payload executed in the browser (pop-up alert shown).

Findings:

- Reflected XSS vulnerability present; user input was reflected in page output without encoding.
- Script execution confirmed via alert pop-up.

---

4. Proof of Concept (PoC)

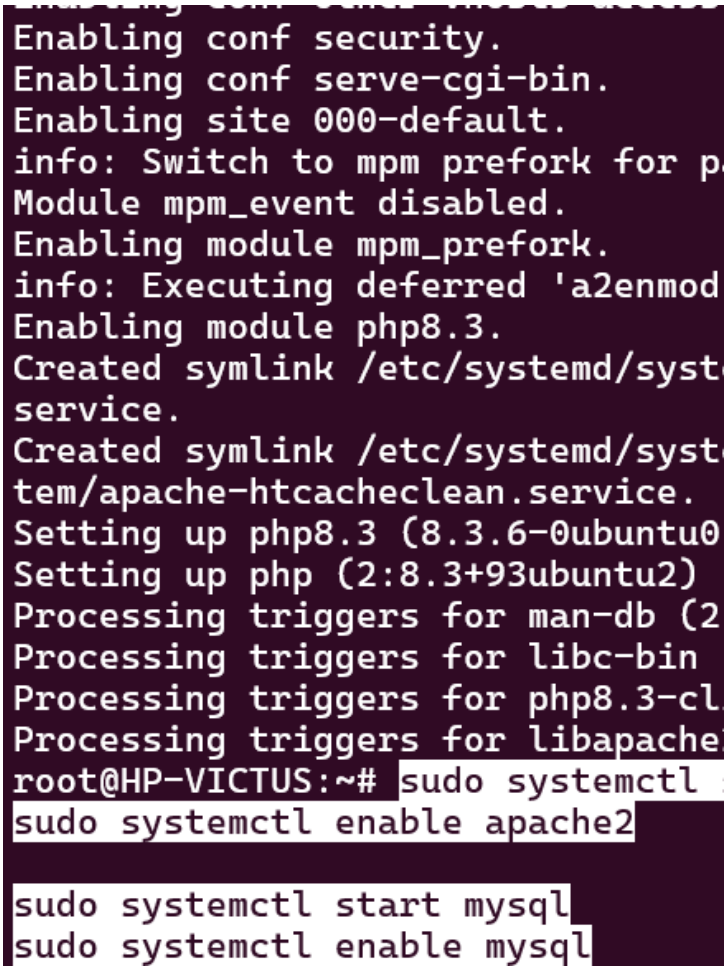
SQL Injection PoC Screenshot

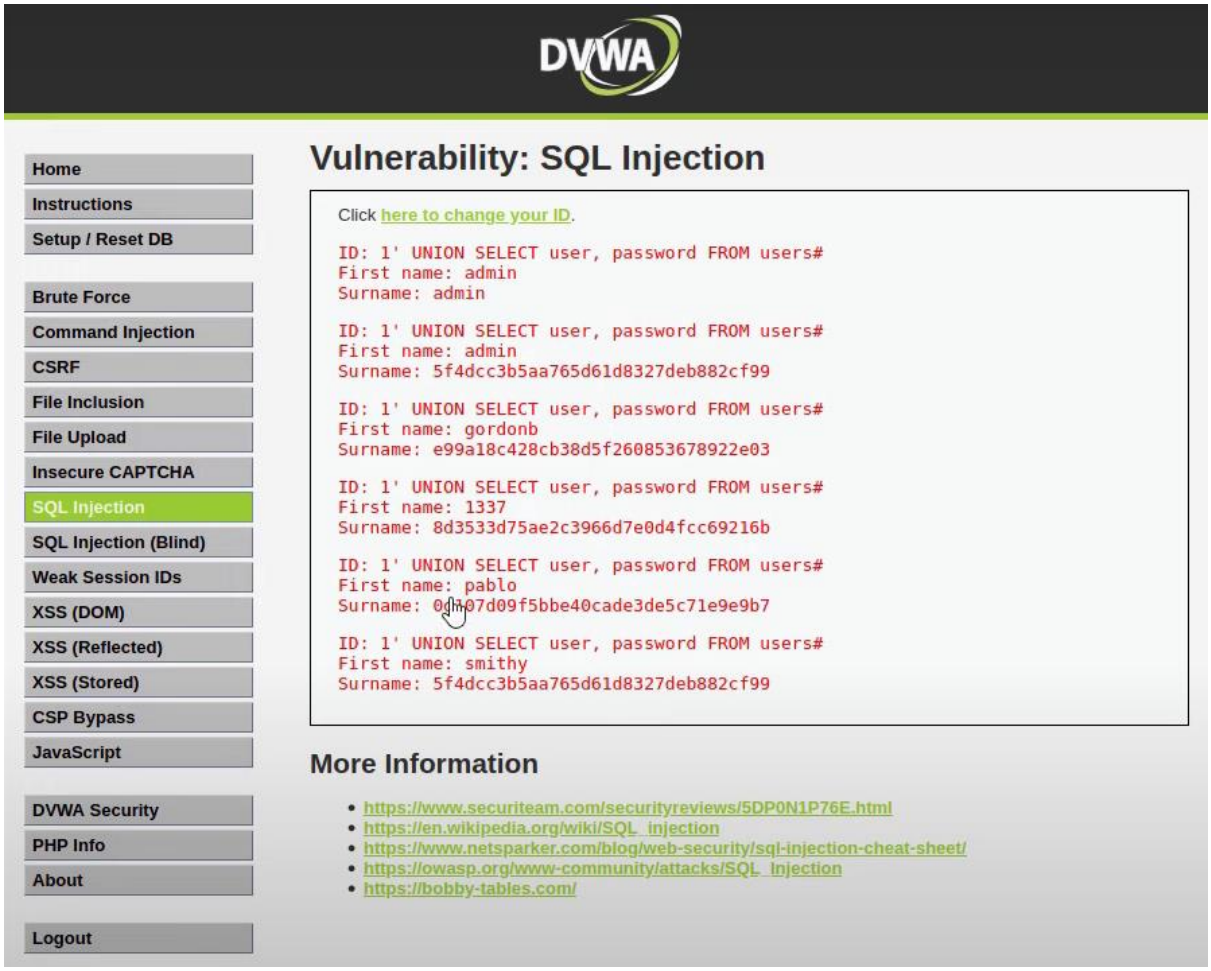
- Inserted ' OR '1'='1 in User ID field.
- Query returned all user records instead of one.

XSS PoC Screenshot

- Entered <script>alert('XSS')</script> in input field.
- Alert pop-up appeared on page load.

Screenshots





---

5. Recommendations & Mitigations

Vulnerability	Recommendation
SQL Injection	Use prepared statements / parameterized queries to separate code and data. Validate and sanitize user inputs. Implement least privilege for DB users.

Vulnerability	Recommendation
Cross-Site Scripting (XSS)	Encode user input before rendering in HTML. Use Content Security Policy (CSP). Sanitize inputs on server side.

---

6. Conclusion

The DVWA test environment revealed classic web application vulnerabilities: SQL Injection and Cross-Site Scripting. Both pose critical risks such as unauthorized data access and code execution in client browsers. Implementing best practices like input validation, prepared statements, and output encoding are essential for securing web applications.