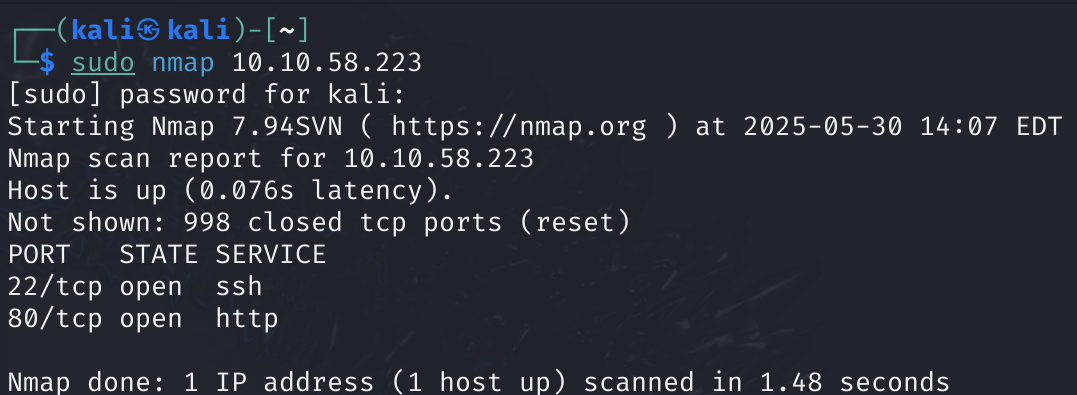
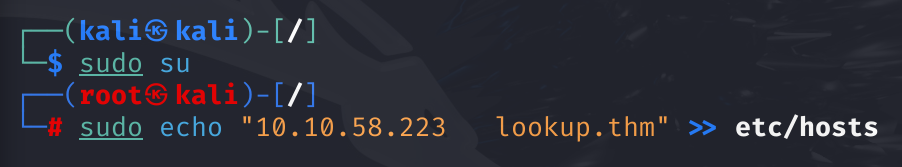
**Lookup\_Documentation**

**Written by: Yuval Quina**

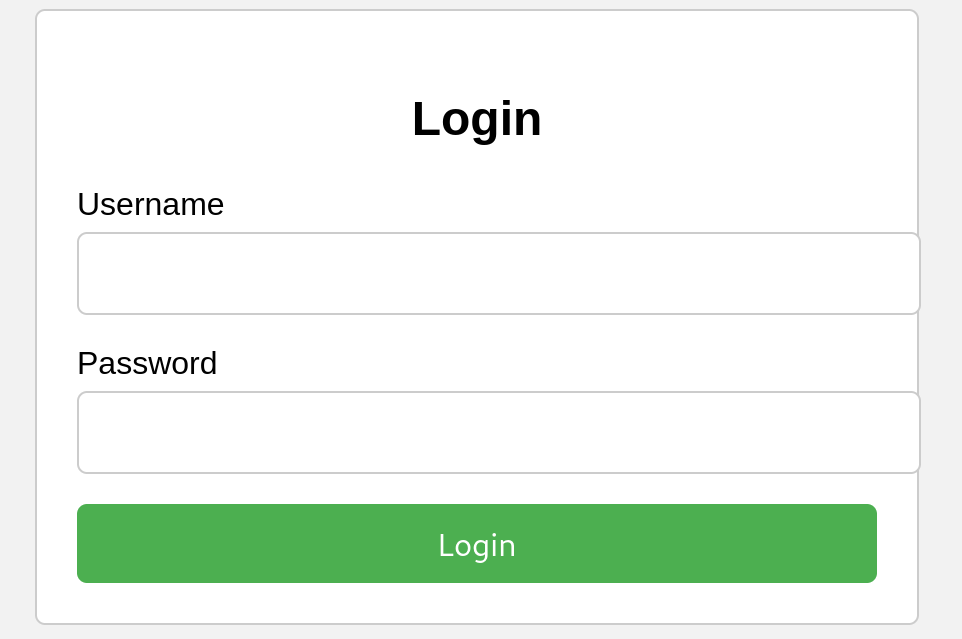
1. At the beginning, I scan the target’s open tcp top 1000 open ports using Nmap:



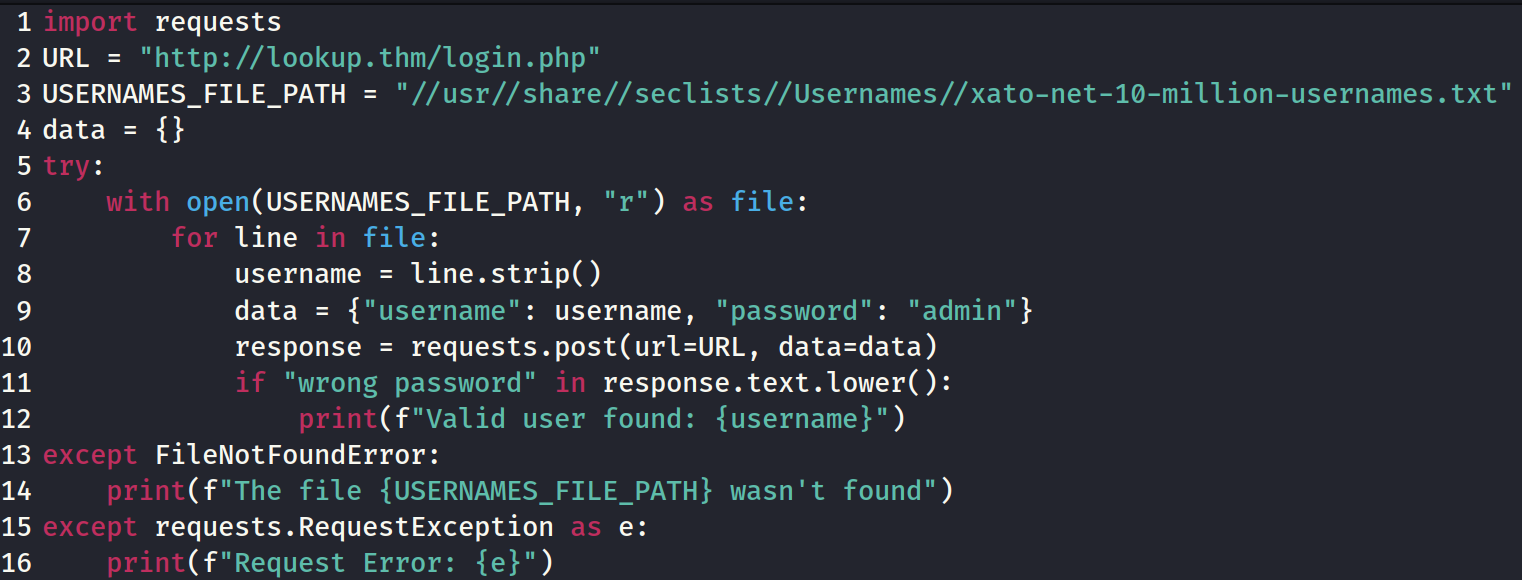
1. I realized that this target has an http open port, so I tried to enter the website. The problem was the domain lookup.thm wasn’t matched to the target’s IP. Therefore, I manually added to my machine’s hosts file the domain with the IP of the target by entering these commands:



1. After doing that, I opened that website again and this is the main page:

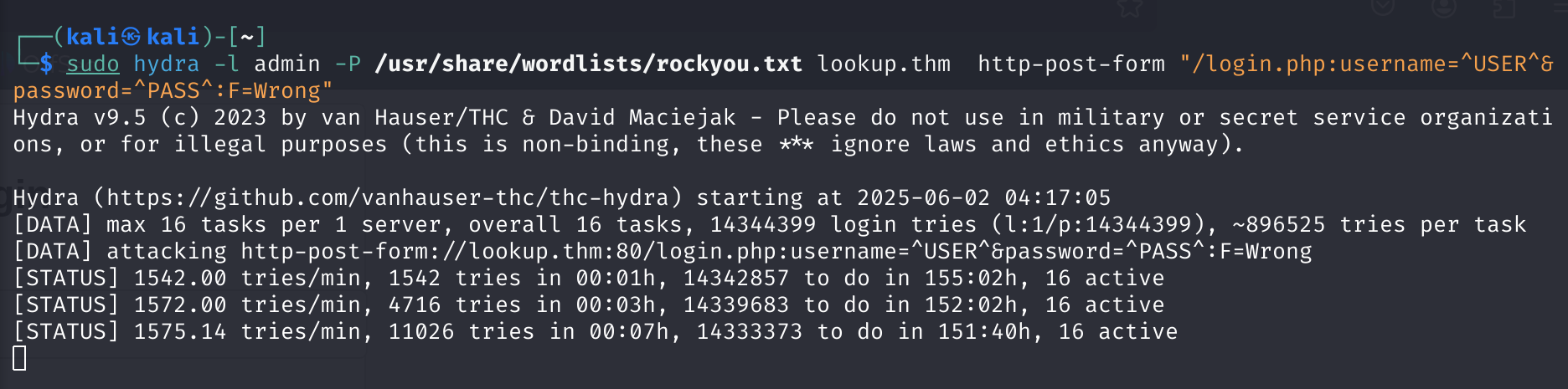


1. After some failed methods(like SQLi and XSS) to find the right credentials, I moved to write this Python script to enumerate usernames over the system:



After running this script, I found the username “admin” and “jose”.

1. Now, I tried to use hydra on both users to discover their passwords:

תמונה שמכילה טקסט, צילום מסך, גופן

תוכן שנוצר על-ידי בינה מלאכותית עשוי להיות שגוי.

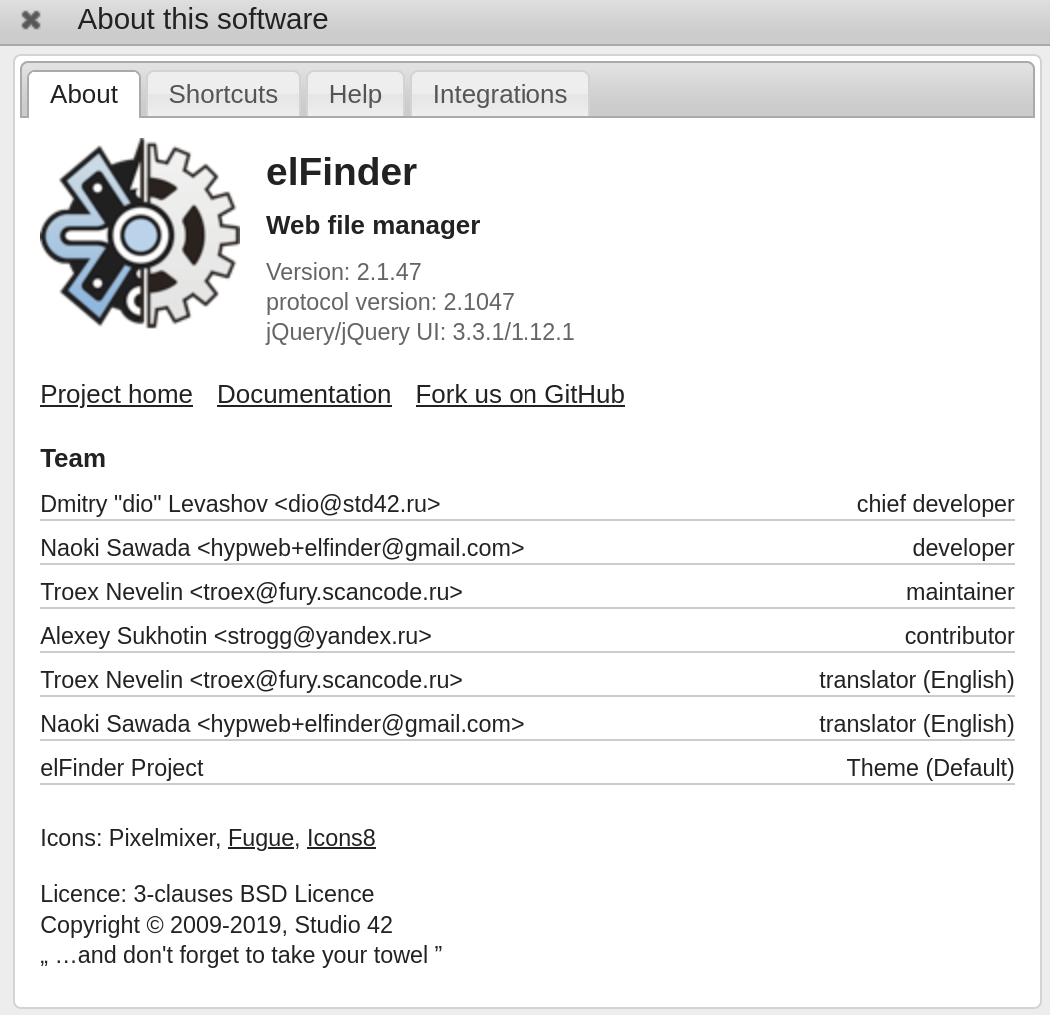
I found that jose’s password is “password123”.

1. I entered the right credentials (jose, password123) and the page wasn’t loaded. I thought a little bit and I realized that I haven’t update the new subdomain “files.lookup.thm” in the hosts file.
2. After connecting again this page appeared:

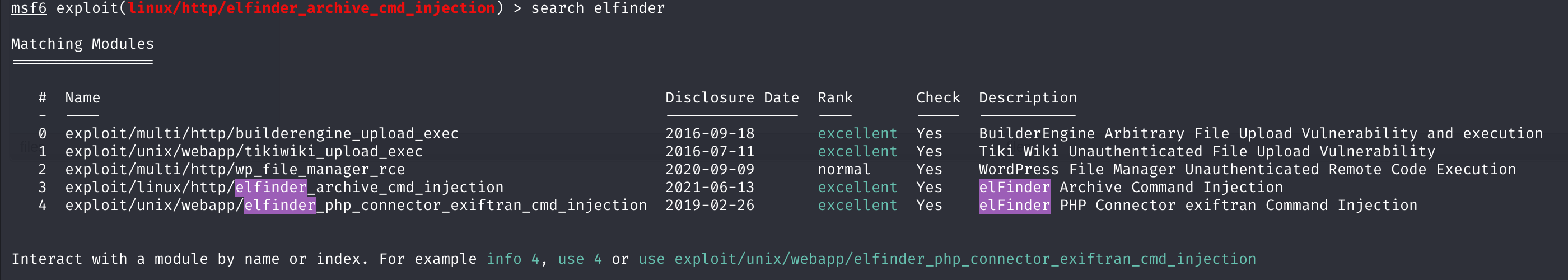
תמונה שמכילה טקסט, תוכנה, סמל מחשב, צילום מסך

תוכן שנוצר על-ידי בינה מלאכותית עשוי להיות שגוי.

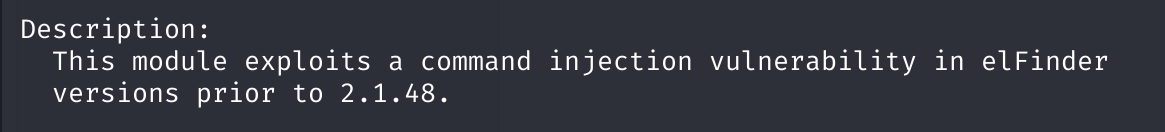
1. I opened all the files and all of them contained passwords, but the “thislogin.txt” and “credentials.txt” contained the usernames and passwords of 2 users. The thislogin.txt contained the credentials of jose and the credentials.txt contained the credentials of the user: “think” with the password: “nopassword”.
2. I tried to login again to the site with the new credentials, but it didn’t work.
3. I tried to find the “elFinder”(the file manager platform of the website) version to maybe exploit it.
4. I clicked the question sign icon on the tools top panel, and I found that the version of the elFinder is “2.1.47”:



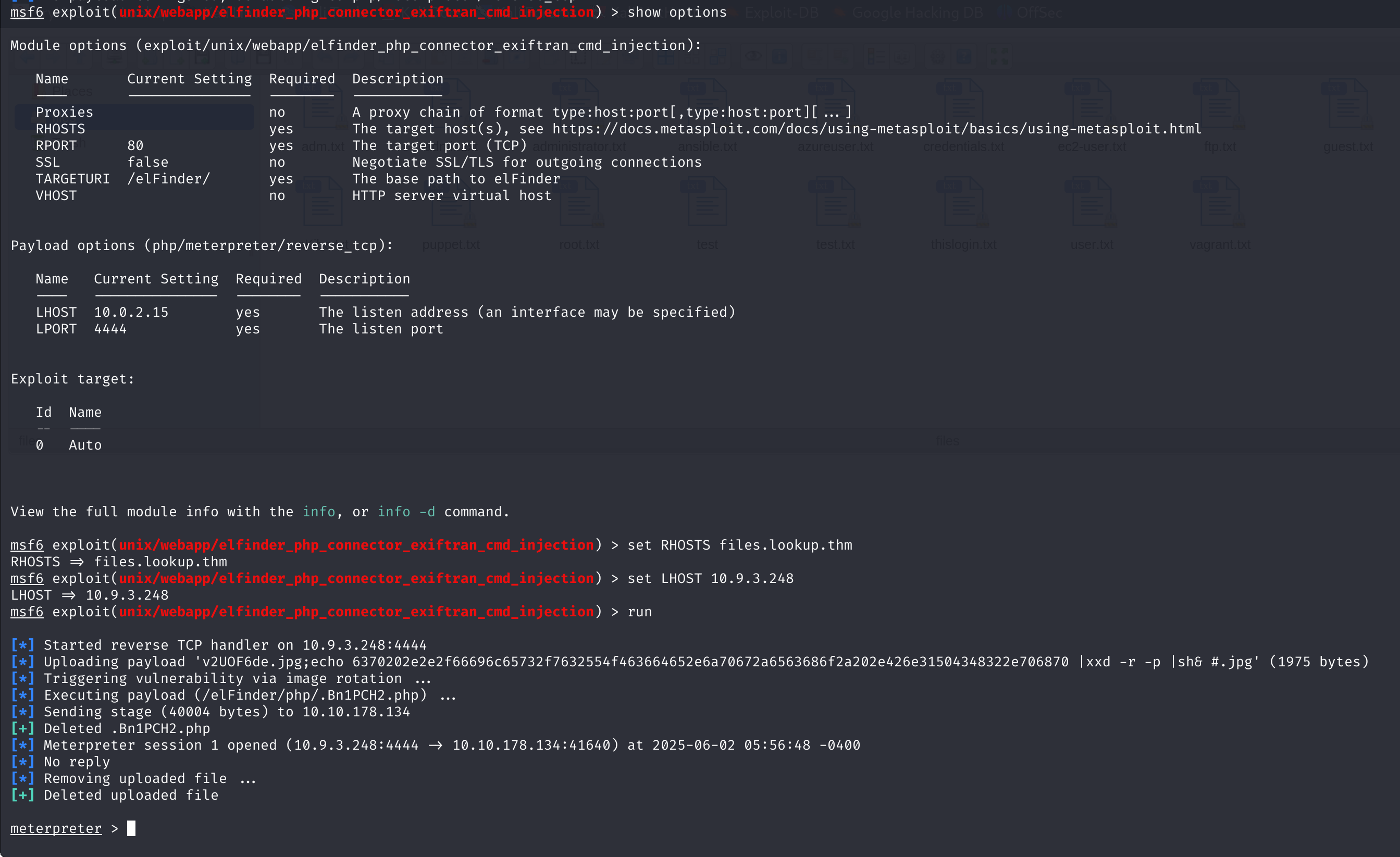
1. After finding the version, I searched the right exploit for this service using Metasploit:



I chose the “elfinder\_php\_connector\_exiftran\_cmd\_injection” exploit (because it works until vesion 2.1.48):



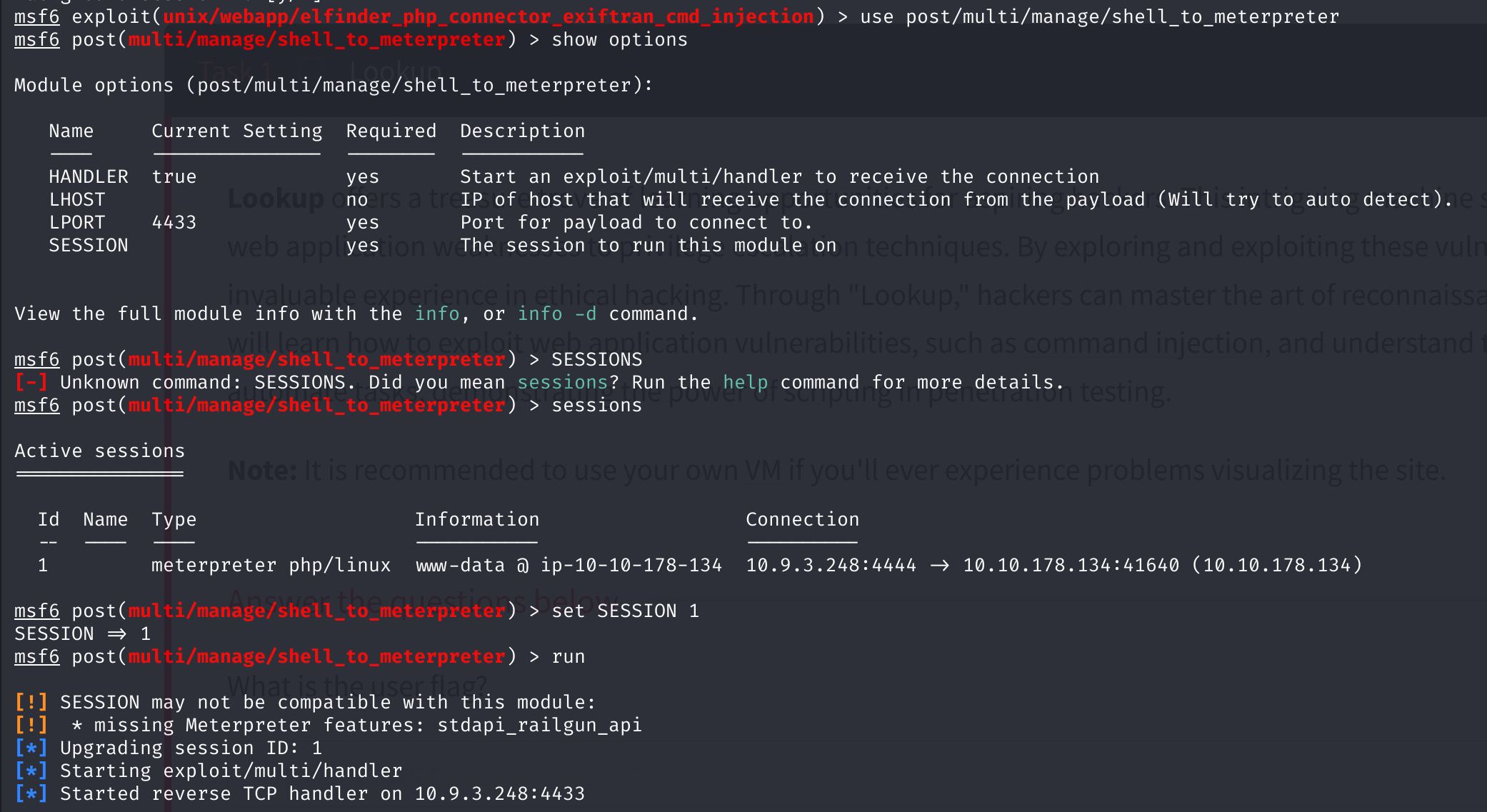
I set the RHOSTS and LHOST values on this exploit and run the exploit:



1. I moved the exploit to the background by pressing ctrl+z.



1. Immediately after moving it to the background, I wanted to convert the reverse shell to meterpeter shell using the post-exploitation module “post/multi/manage/shell\_to\_meterpreter”:

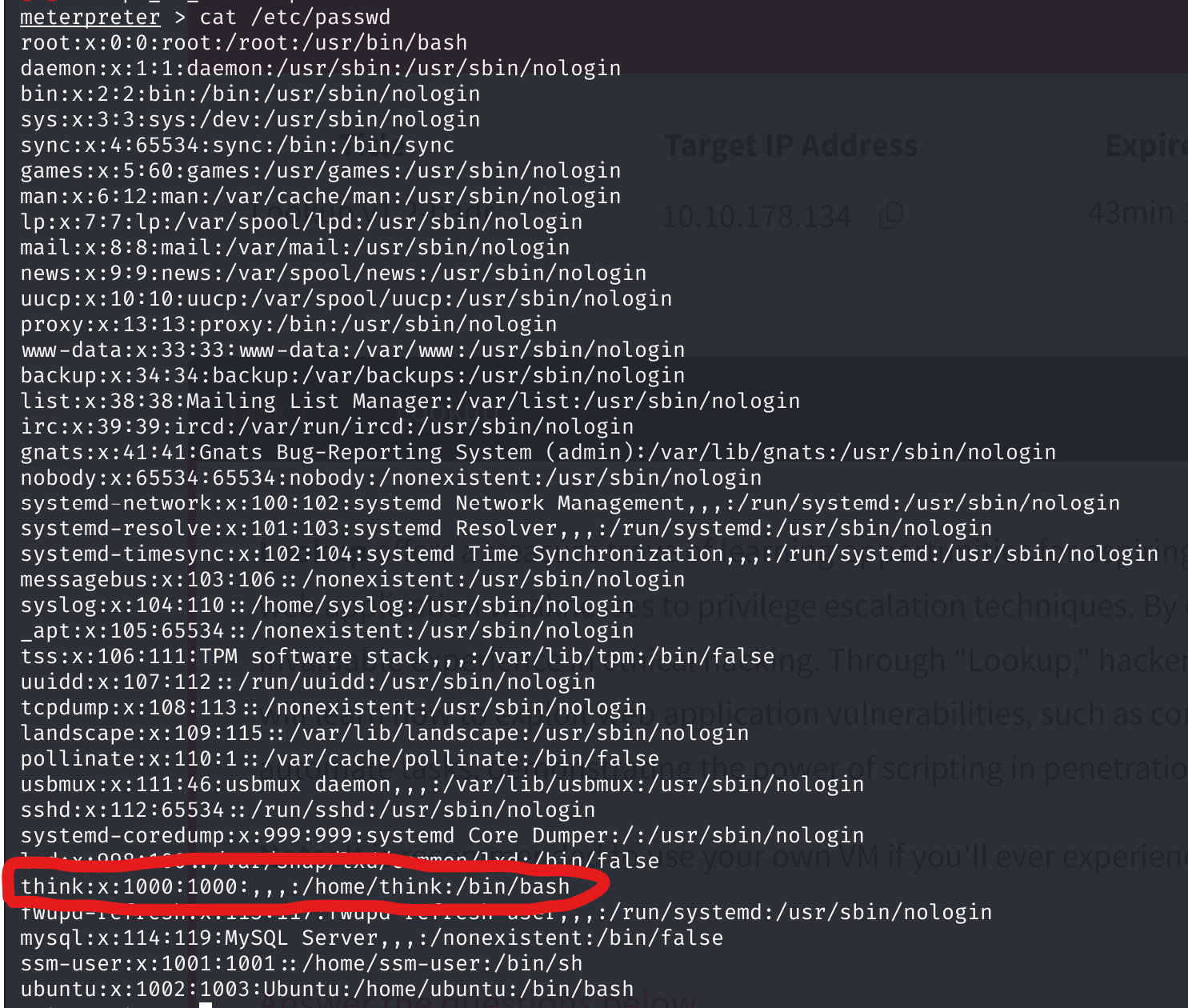


1. After completing the reverse shell, I started navigating the target’s files and finding the flag.
2. I used “getuid” command to see which user I’m currently running on.

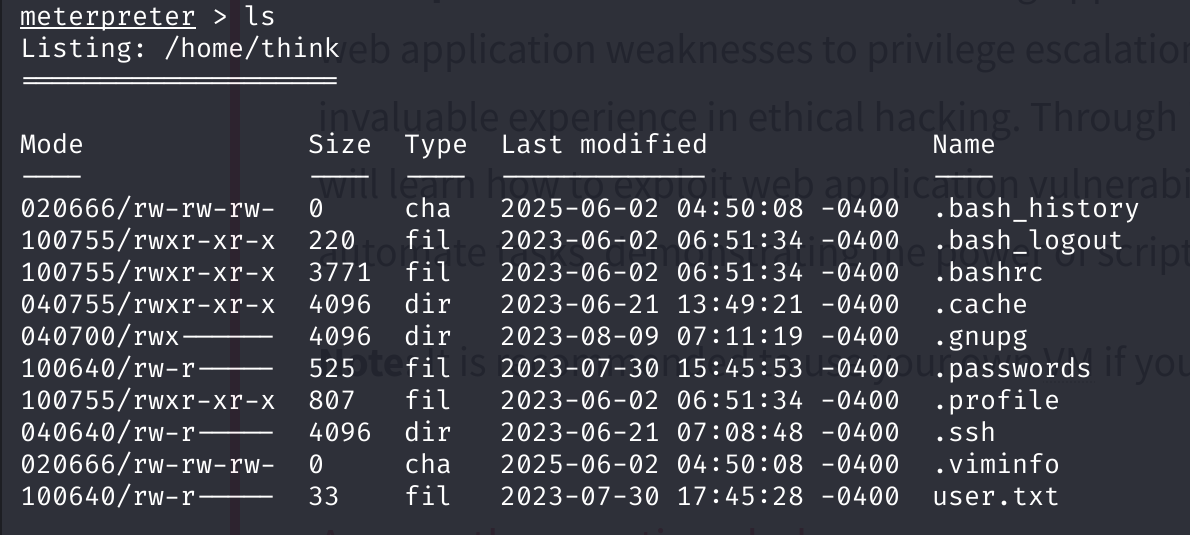
תמונה שמכילה טקסט, גופן, צילום מסך, קו

תוכן שנוצר על-ידי בינה מלאכותית עשוי להיות שגוי.

1. I understood that this is low privilege user, so I need to escalate privileges to get more accessibility
2. I found that “think” is a higher privilege user after opening the passwd file:

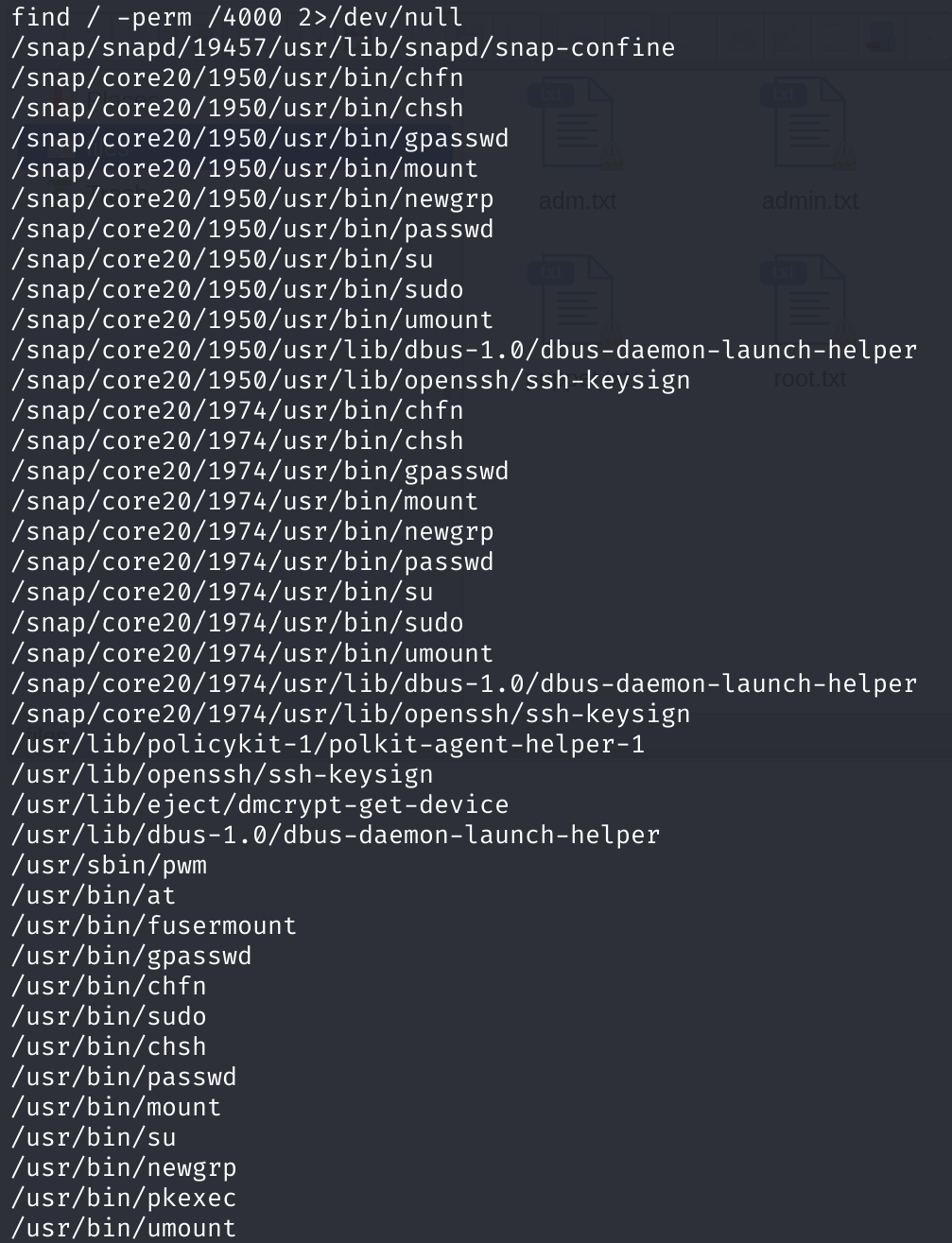


1. Now I navigate to the think’s directory and found “user.txt” and “.passwords” files, which seems relevant to the CTF:

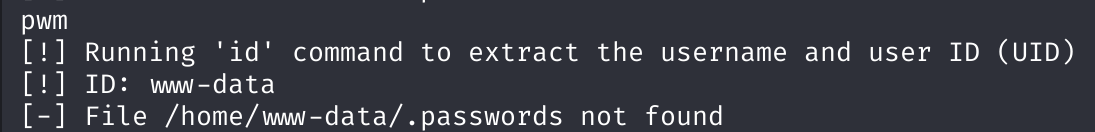


The problem is that while I was logged as “www-data” I didn’t have permissions to open them.

1. Therefore, I searched for SUID files that allow me access these files (Note: The usr/bin folder contains the binary files of the tools/commands of the Linux host):



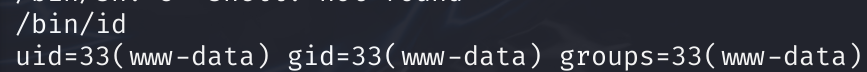
1. I found the “pwm” file which shouldn’t be there by default, so I run it:



It seems like the program tries to use the **id** command (which located in: ”/bin/id”) on the current user (www-data) and use it to open the .passwords file using this path: “/home/(USERNAME)/.passwords”.

This made me think that maybe I will be able to run this command over the think user and by doing so accessing his .password file.

1. I run the id command of the current user:



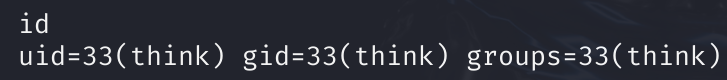
1. I created a temporary id command that will contain the username “think” instead of the current user so the program will use him as the username on the .passwords file path:



תמונה שמכילה טקסט, גופן, צילום מסך

תוכן בינה מלאכותית גנרטיבית עשוי להיות שגוי.



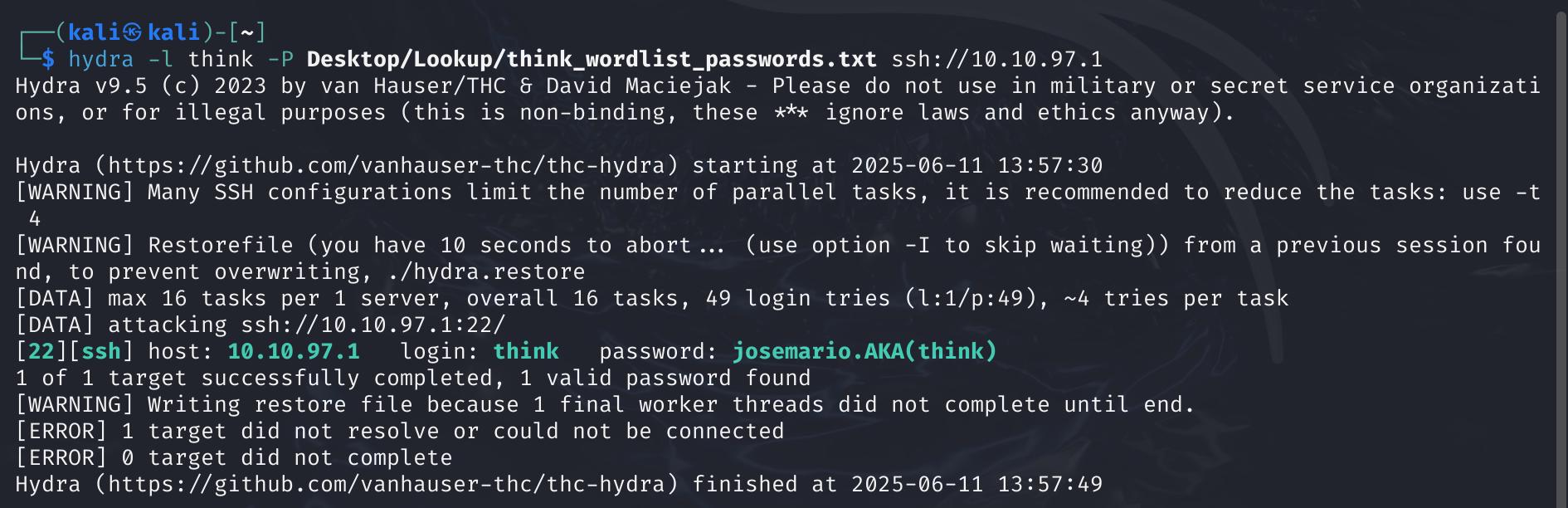


1. Now, after I run the pwm command again, I saw the entire content of the .passwords file of think.

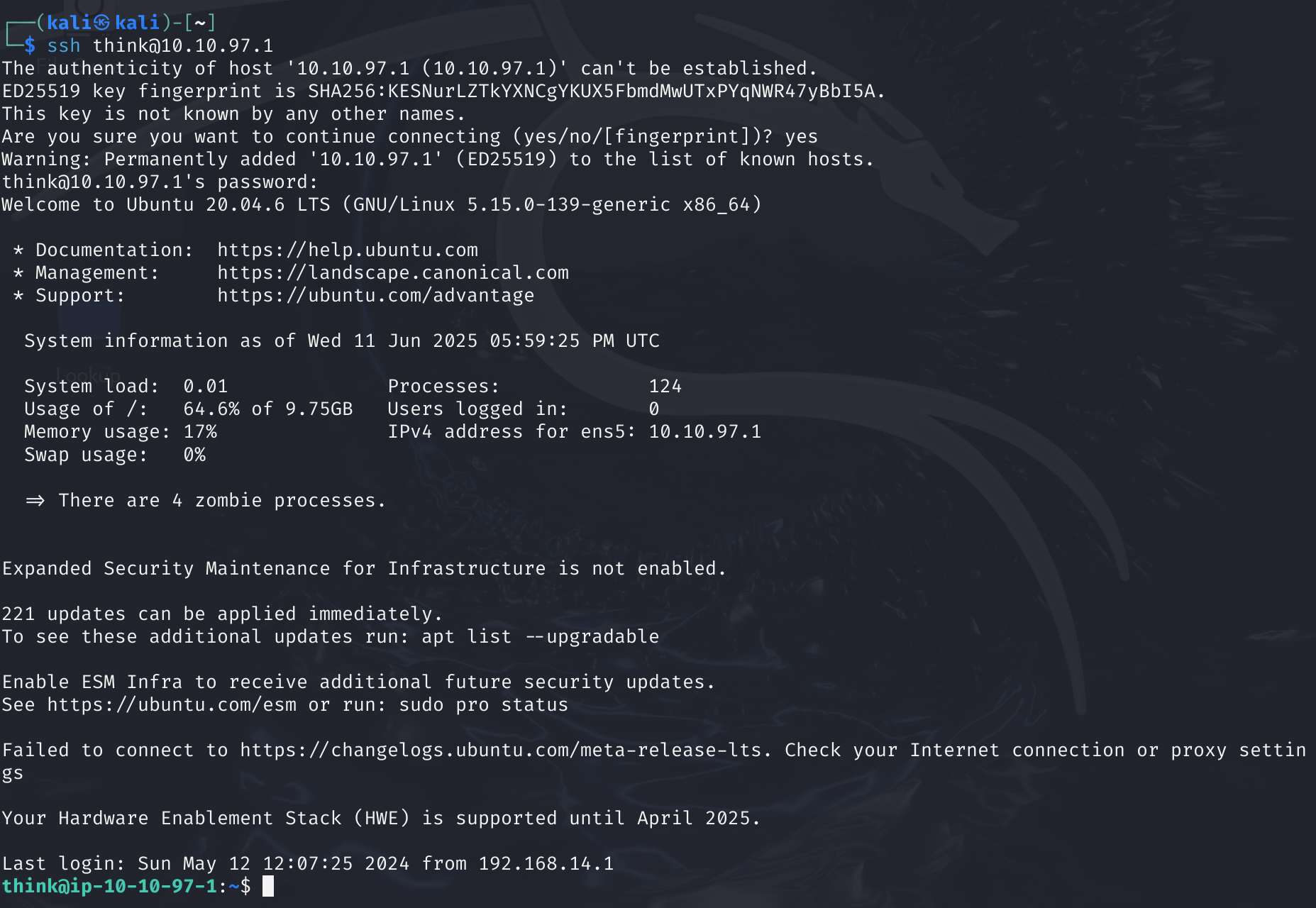
תמונה שמכילה טקסט, צילום מסך

תוכן בינה מלאכותית גנרטיבית עשוי להיות שגוי.

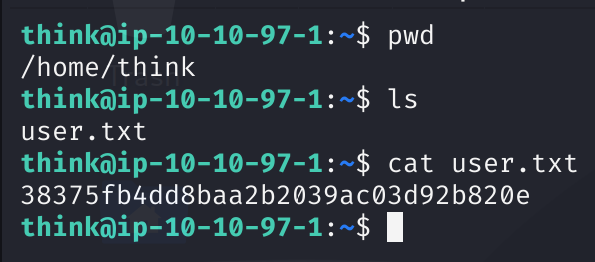
1. After I discovered all the passwords of the user “think”, I run the following hydra command to brute force the entire wordlists of the possible passwords over the open ssh port that he have:



1. Now, I just used the basic ssh tool of Linux to get a shell over the machine over the open ssh port as the user “think”:

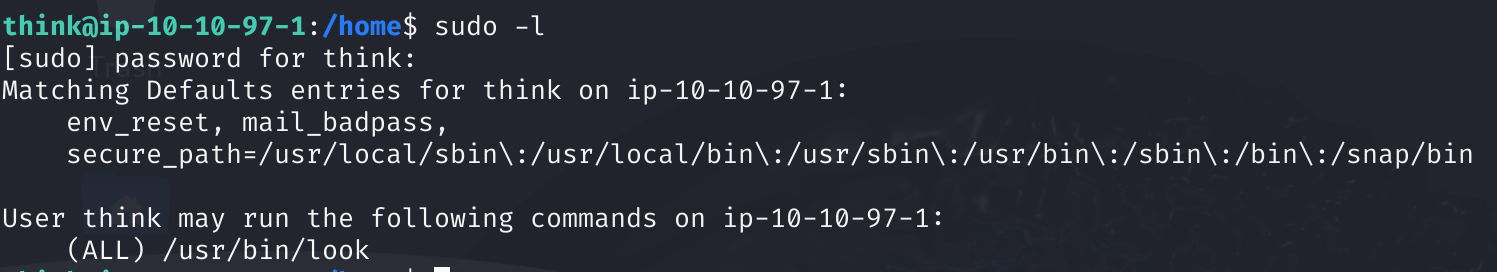


1. I navigated through the files and opened the user.txt file (the same file that I found from section 19):



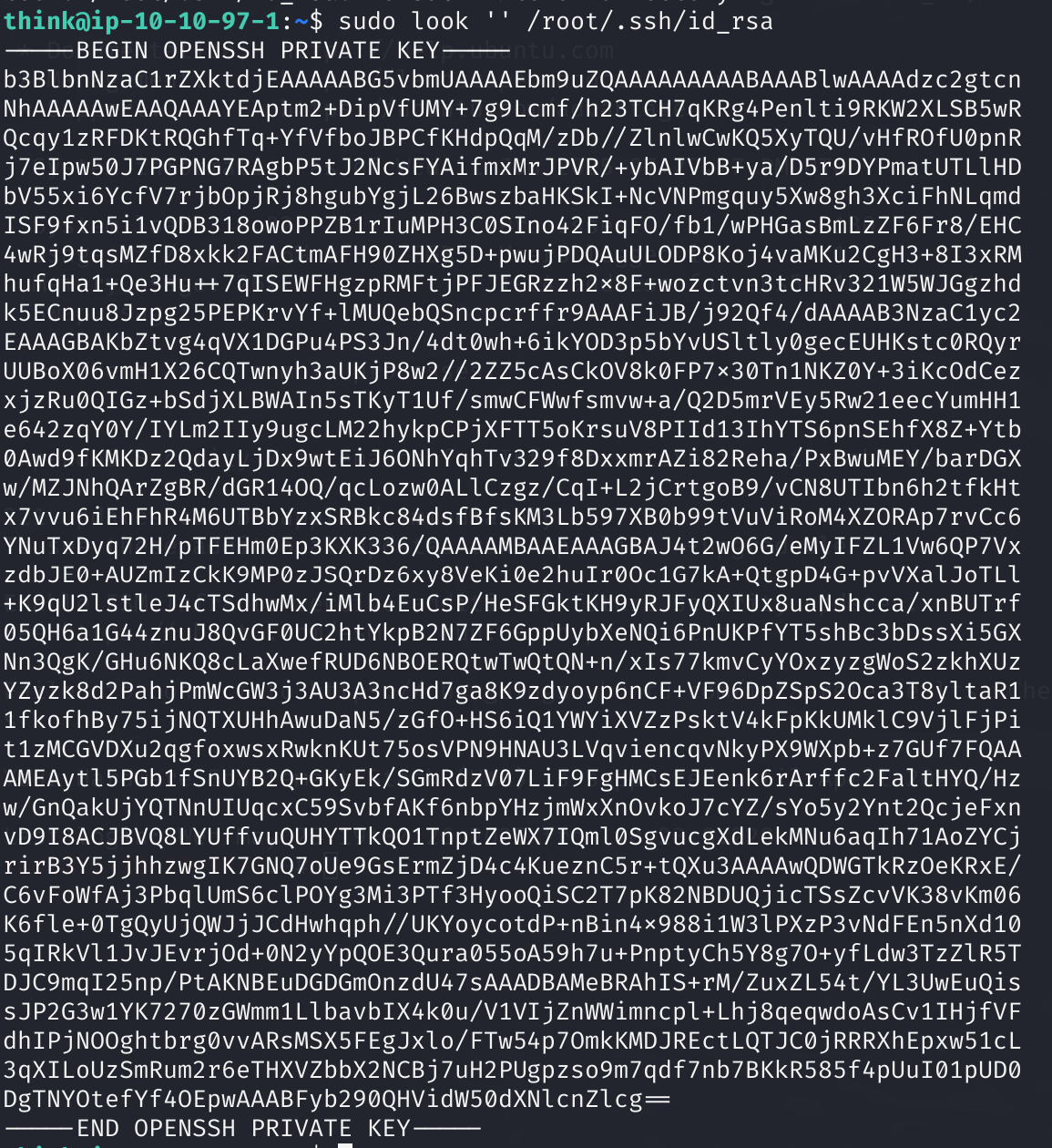
And here it is, my first flag!

1. Now I checked my sudo privileges as the user think:



So, it seems like that I can only run the look command as the root (a command which operate like the default “grep” and “find” commands). This made me feel like I can use this command for searching files that are restricted to the root and access them.

1. I automatically searched for the file that contains the private ssh key of the root:



As soon as it found the key, I copy the entire key to a new .pem file in order to be able to access it as non-root user.

Now, I loaded this private key to this ssh command in order to escalate to the root:

תמונה שמכילה טקסט, צילום מסך, גופן

תוכן בינה מלאכותית גנרטיבית עשוי להיות שגוי.

1. And now after navigating the files I found the second flag:

