MACHINE: 'MIRAI' – HTB

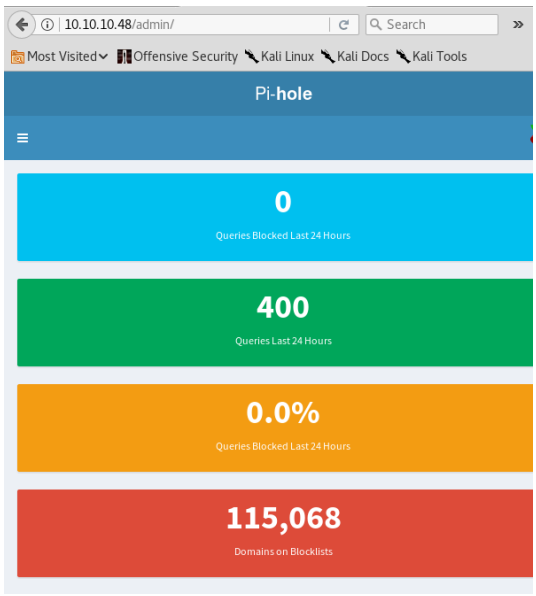
As always, I started with a nmap scan like: nmap -A 10.10.10.48 with the following result

```
root@kali:~# nmap -A 10.10.10.48

Starting Nmap 7.60 ( https://nmap.org ) at 2017-11-11 02:17 CET
Nmap scan report for 10.10.10.48
Host is up (0.070s latency).
Not shown: 997 closed ports
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 6.7p1 Debian 5+deb8u3 (protocol 2.0)
| ssh-hostkey:
|   1024 aa:ef:5c:e0:8e:86:97:82:47:ff:4a:e5:40:18:90:c5 (DSA)
|   2048 e8:c1:9d:c5:43:ab:fe:61:23:3b:d7:e4:af:9b:74:18 (RSA)
|   256 b6:a0:78:38:d0:c8:10:94:8b:44:b2:ea:a0:17:42:2b (ECDSA)
|_  256 4d:68:40:f7:20:c4:e5:52:80:7a:44:38:b8:a2:a7:52 (EdDSA)
53/tcp open  domain  dnsmasq 2.76
| dns-nsid:
|_  bind.version: dnsmasq-2.76
80/tcp open  http    lighttpd 1.4.35
|_http-server-header: lighttpd/1.4.35
|_http-title: Site doesn't have a title (text/html; charset=UTF-8).
No exact OS matches for host (If you know what OS is running on it, see https
```

Both, 22 and 80 ports are very interesting, so I started researching around port 80 in order to find out some hint that let me open a ssh conection in port 22.

Bearing in mind what Mirai is and how it Works, I tried to browse the http server with default credentials as 10.10.10.48/root, 10.10.10.48/Mirai or 10.10.10.48/user for example, and it worked with 10.10.10.48/admin, finding that page:

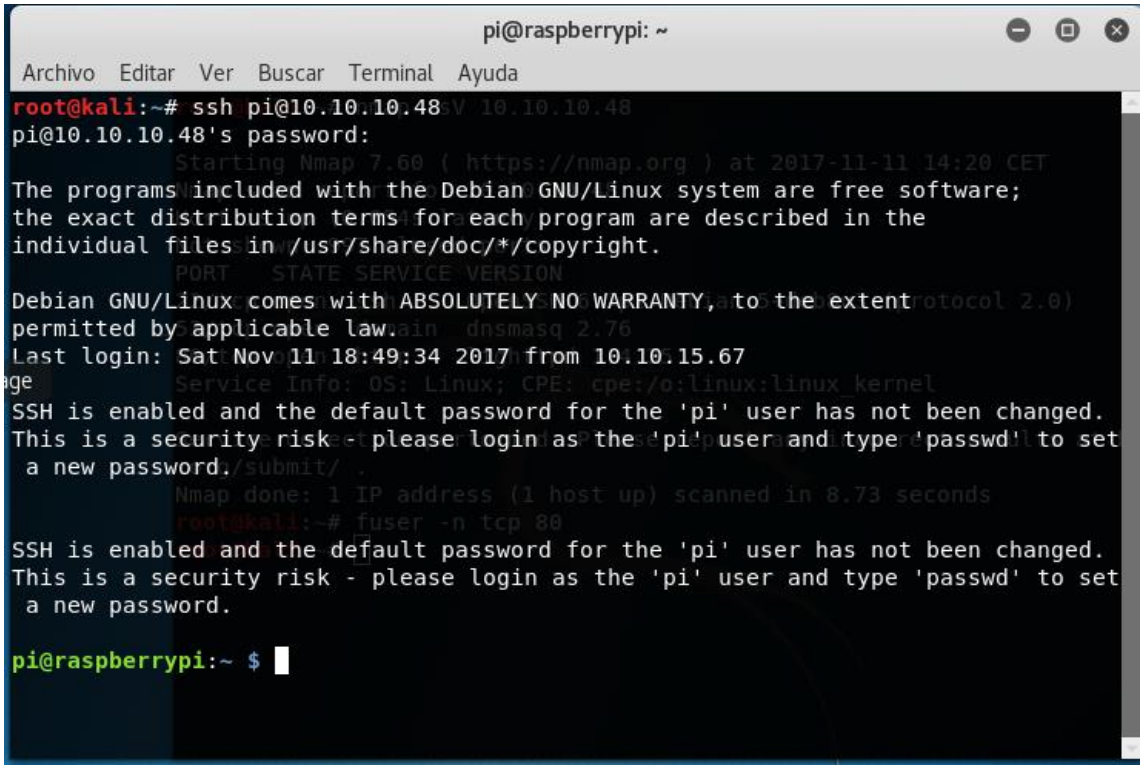


At this point, having in one hand how Mirai Works and in the other hand a Pi-hole application, I did the following Google search: “default credentials of pi-hole”, obtaining this result:

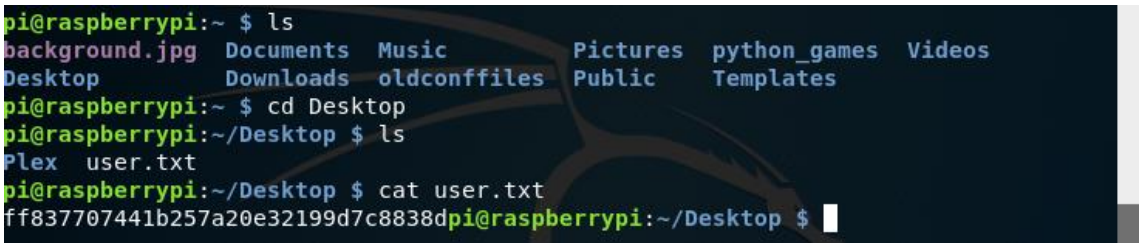
User:pi

Password: raspberry

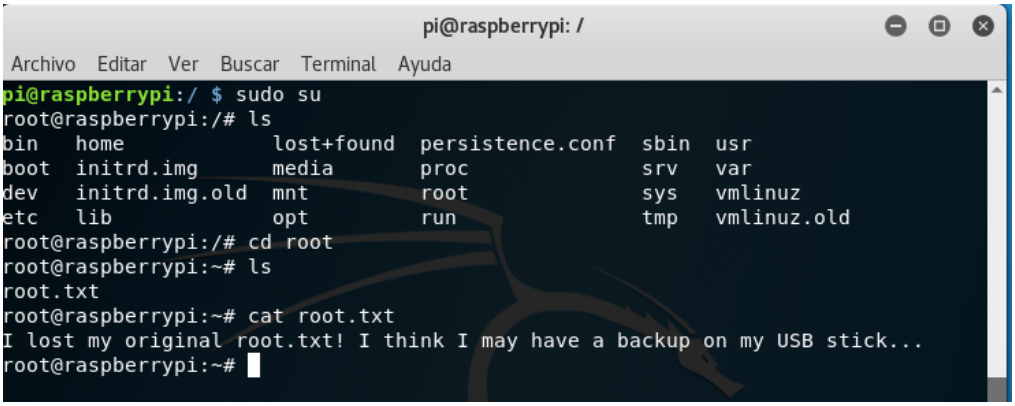
So I tried to open a ssh conection to de target with -> ssh pi@10.10.10.48 with raspberry as password, and I succeeded:



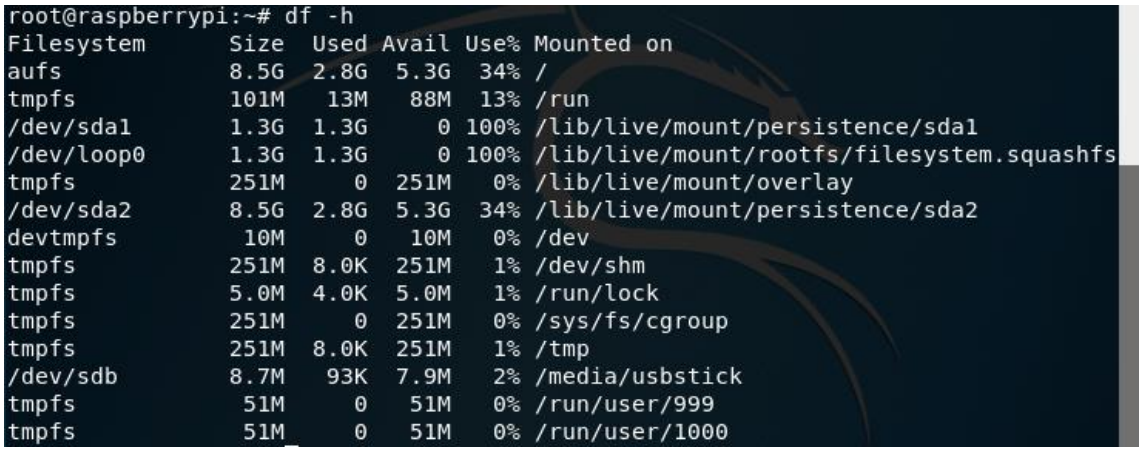
Now, find the user flag was just to:



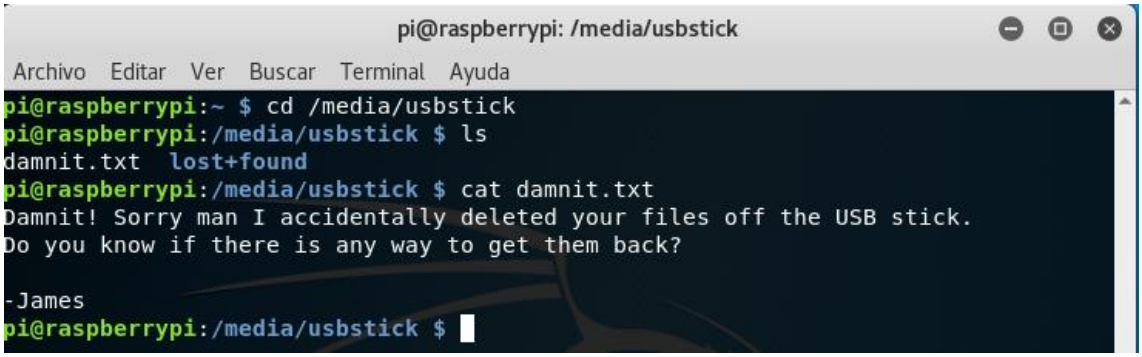
Root’s flag was a Little bit more difficult, because after change to super user, and find the root.txt file, I got that message:



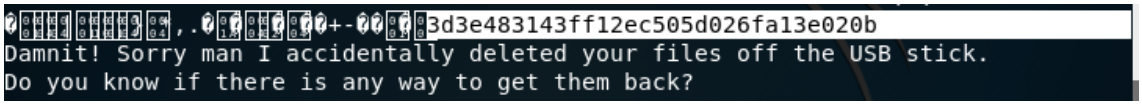
So I needed to see all the hard drives connected to the system in order to know their routes:



Okay, here was the usb stick mentioned in the clue, but, surprisingly this is what I found:



At this point, I broke my mind overthinking how could I read the memory space the flag had occupied. But after a few research, I realized that is possible to read a hard drive as we can read a simple file, so I used a simple ‘cat /dev/sdb’ command, and here is the result:



That’s it !

Pitenager