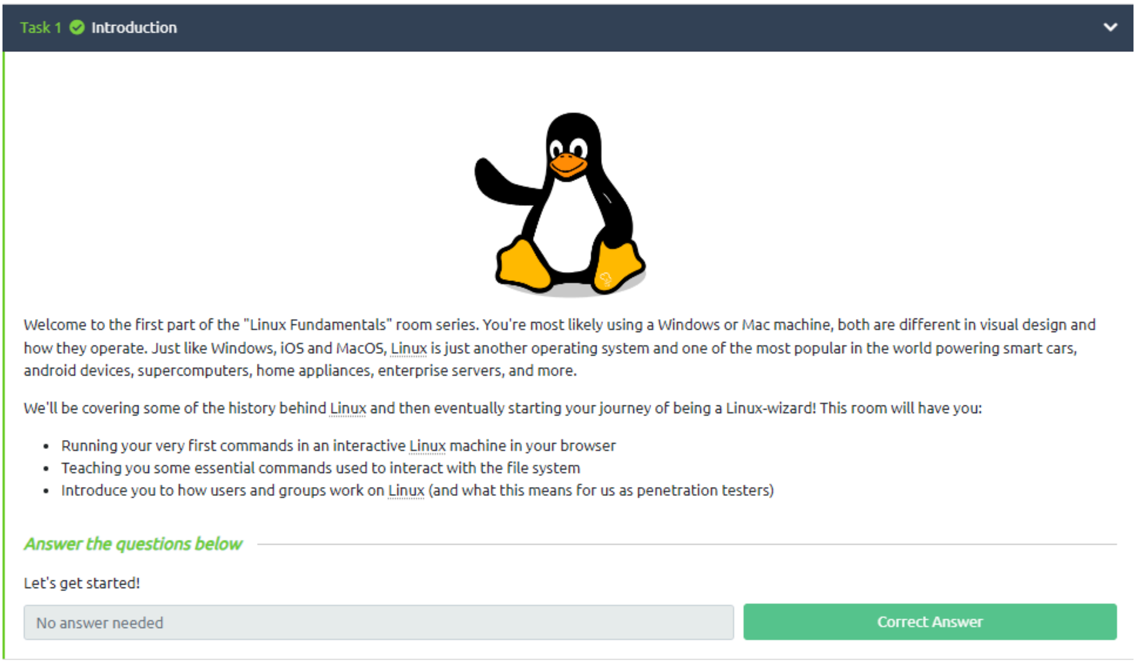
LINUX FUNDAMENTALS

- Sarthak Parashetti

Linux Fundamentals Part 1 :

Task 1 : Introduction



Task 1 ✓ Introduction

Welcome to the first part of the "Linux Fundamentals" room series. You're most likely using a Windows or Mac machine, both are different in visual design and how they operate. Just like Windows, iOS and MacOS, Linux is just another operating system and one of the most popular in the world powering smart cars, android devices, supercomputers, home appliances, enterprise servers, and more.

We'll be covering some of the history behind Linux and then eventually starting your journey of being a Linux-wizard! This room will have you:

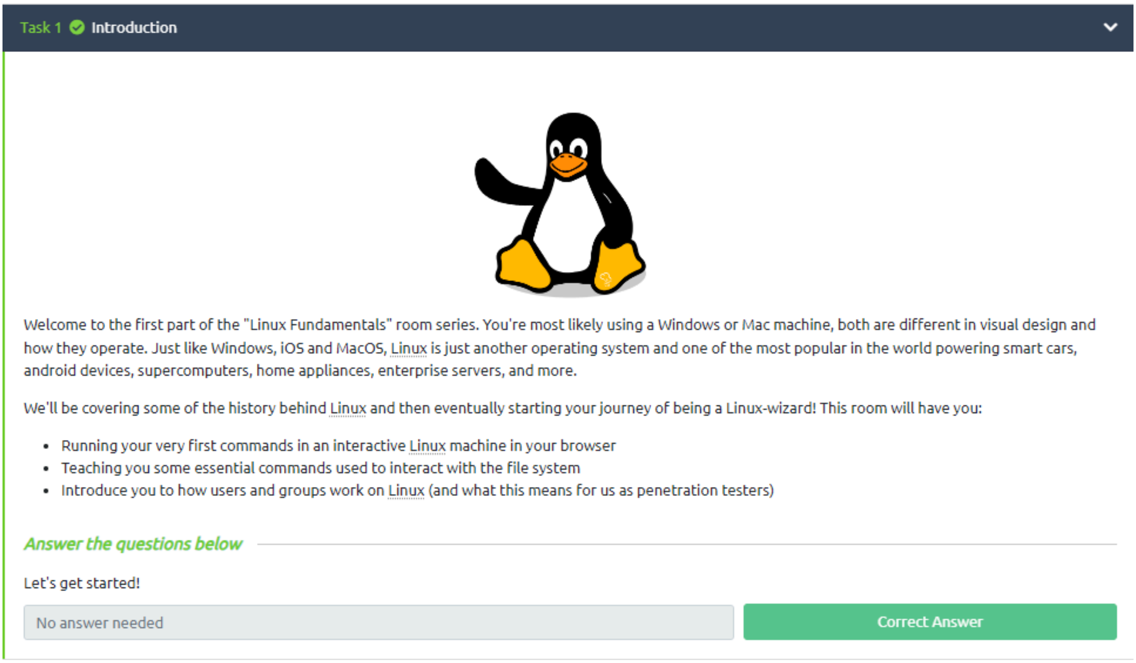
- Running your very first commands in an interactive Linux machine in your browser
- Teaching you some essential commands used to interact with the file system
- Introduce you to how users and groups work on Linux (and what this means for us as penetration testers)

Answer the questions below

Let's get started!

No answer needed

Task 2 : A bit of Background on Linux



Task 1 ✓ Introduction

Welcome to the first part of the "Linux Fundamentals" room series. You're most likely using a Windows or Mac machine, both are different in visual design and how they operate. Just like Windows, iOS and MacOS, Linux is just another operating system and one of the most popular in the world powering smart cars, android devices, supercomputers, home appliances, enterprise servers, and more.

We'll be covering some of the history behind Linux and then eventually starting your journey of being a Linux-wizard! This room will have you:

- Running your very first commands in an interactive Linux machine in your browser
- Teaching you some essential commands used to interact with the file system
- Introduce you to how users and groups work on Linux (and what this means for us as penetration testers)

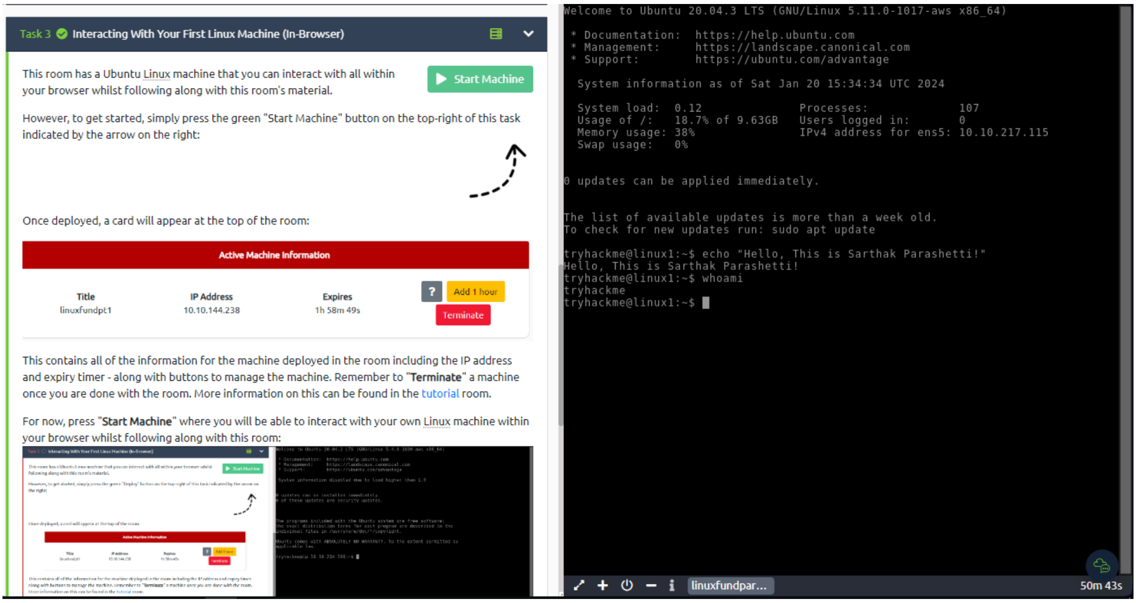
Answer the questions below

Let's get started!

No answer needed

Correct Answer

Task 3 : Interacting with your First Linux Machine(In-Browser)



Task 3 ✓ Interacting With Your First Linux Machine (In-Browser)

This room has a Ubuntu Linux machine that you can interact with all within your browser whilst following along with this room's material.

However, to get started, simply press the green "Start Machine" button on the top-right of this task indicated by the arrow on the right:

Once deployed, a card will appear at the top of the room:

Active Machine Information

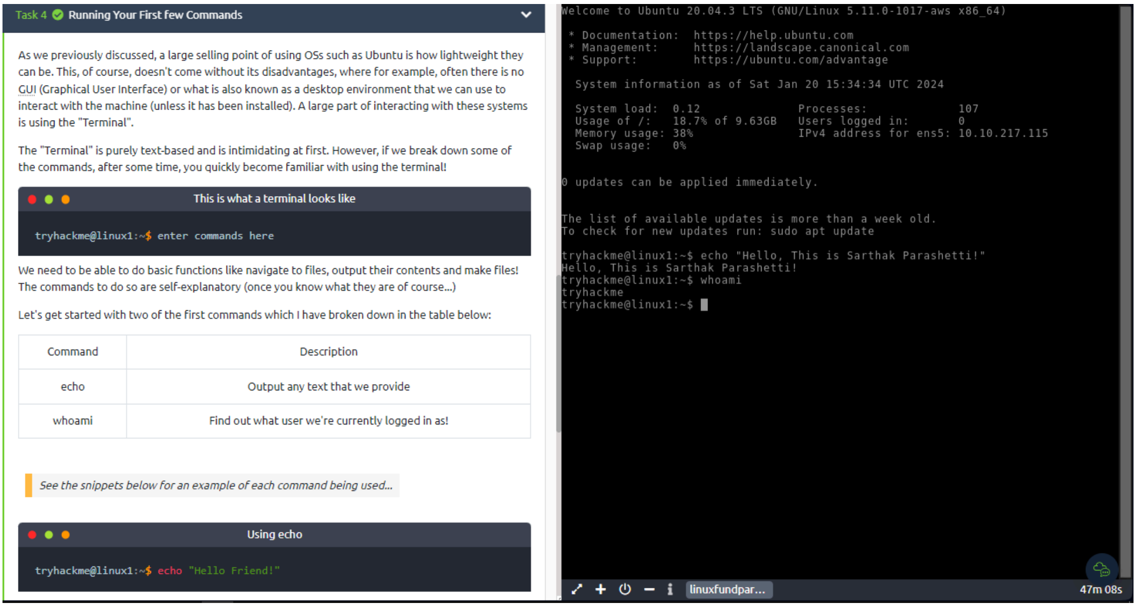
Title	IP Address	Expires
linuxfundpt1	10.10.144.238	1h 58m 49s

This contains all of the information for the machine deployed in the room including the IP address and expiry timer - along with buttons to manage the machine. Remember to "**Terminate**" a machine once you are done with the room. More information on this can be found in the tutorial room.

For now, press "**Start Machine**" where you will be able to interact with your own Linux machine within your browser whilst following along with this room:

```
Welcome to Ubuntu 20.04.3 LTS (GNU/Linux 5.11.0-1017-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Sat Jan 20 15:34:34 UTC 2024

  System load:  0.12              Processes:             107
  Usage of /:   18.7% of 9.63GB   Users logged in:       0
  Memory usage: 38%               IPv4 address for ens5: 10.10.217.115
  Swap usage:   0%

0 updates can be applied immediately.

The list of available updates is more than a week old.
To check for new updates run: sudo apt update

tryhackme@linux1:~$ echo "Hello, This is Sarthak Parashetti!"
Hello, This is Sarthak Parashetti!
tryhackme@linux1:~$ whoami
tryhackme
tryhackme@linux1:~$ 
```

Task 4 : Running your First few Commands



As we previously discussed, a large selling point of using OSs such as Ubuntu is how lightweight they can be. This, of course, doesn't come without its disadvantages, where for example, often there is no GUI (Graphical User Interface) or what is also known as a desktop environment that we can use to interact with the machine (unless it has been installed). A large part of interacting with these systems is using the "Terminal".

The "Terminal" is purely text-based and is intimidating at first. However, if we break down some of the commands, after some time, you quickly become familiar with using the terminal!

```
tryhackme@linux1:~$ enter commands here
```

We need to be able to do basic functions like navigate to files, output their contents and make files! The commands to do so are self-explanatory (once you know what they are of course...)

Let's get started with two of the first commands which I have broken down in the table below:

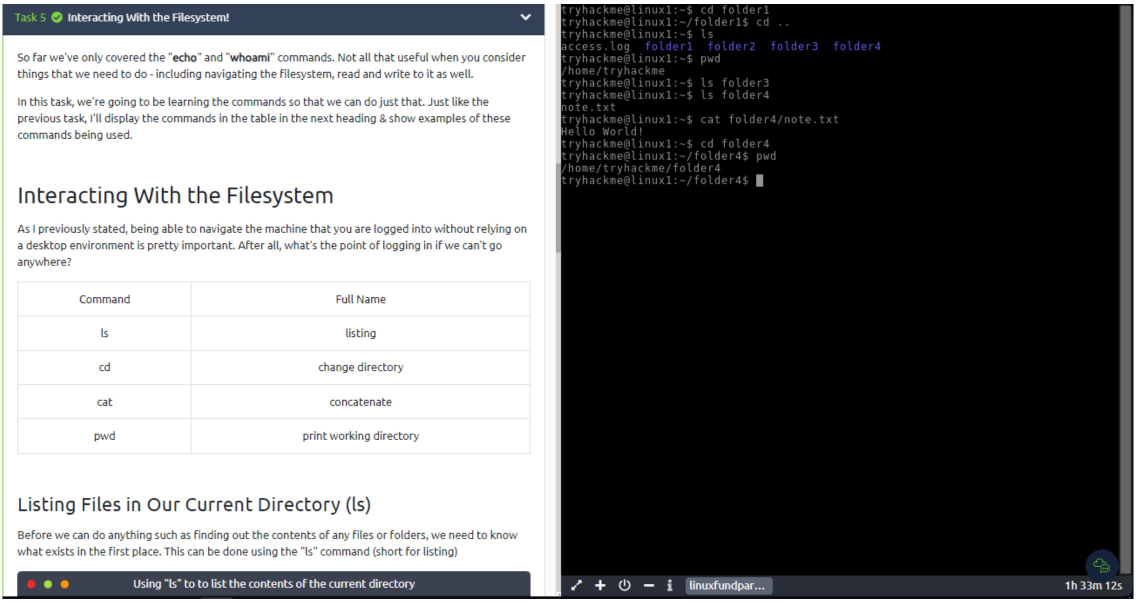
Command	Description
echo	Output any text that we provide
whoami	Find out what user we're currently logged in as!

See the snippets below for an example of each command being used...

```
tryhackme@linux1:~$ echo "Hello Friend!"
```

```
Welcome to Ubuntu 20.04.3 LTS (GNU/Linux 5.11.0-1017-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Sat Jan 20 15:34:34 UTC 2024

  System load:  0.12              Processes:             107
  Usage of /:   18.7% of 9.63GB   Users logged in:       0
  Memory usage: 38%               IPv4 address for ens5: 10.10.217.115
  Swap usage:   0%

0 updates can be applied immediately.

The list of available updates is more than a week old.
To check for new updates run: sudo apt update

tryhackme@linux1:~$ echo "Hello, This is Sarthak Parashetti!"
Hello, This is Sarthak Parashetti!
tryhackme@linux1:~$ whoami
tryhackme
tryhackme@linux1:~$ 
```

Task 5 : Interacting with the Filesystem!



Task 5 Interacting With the Filesystem!

So far we've only covered the "**echo**" and "**whoami**" commands. Not all that useful when you consider things that we need to do - including navigating the filesystem, read and write to it as well.

In this task, we're going to be learning the commands so that we can do just that. Just like the previous task, I'll display the commands in the table in the next heading & show examples of these commands being used.

Interacting With the Filesystem

As I previously stated, being able to navigate the machine that you are logged into without relying on a desktop environment is pretty important. After all, what's the point of logging in if we can't go anywhere?

Command	Full Name
ls	listing
cd	change directory
cat	concatenate
pwd	print working directory

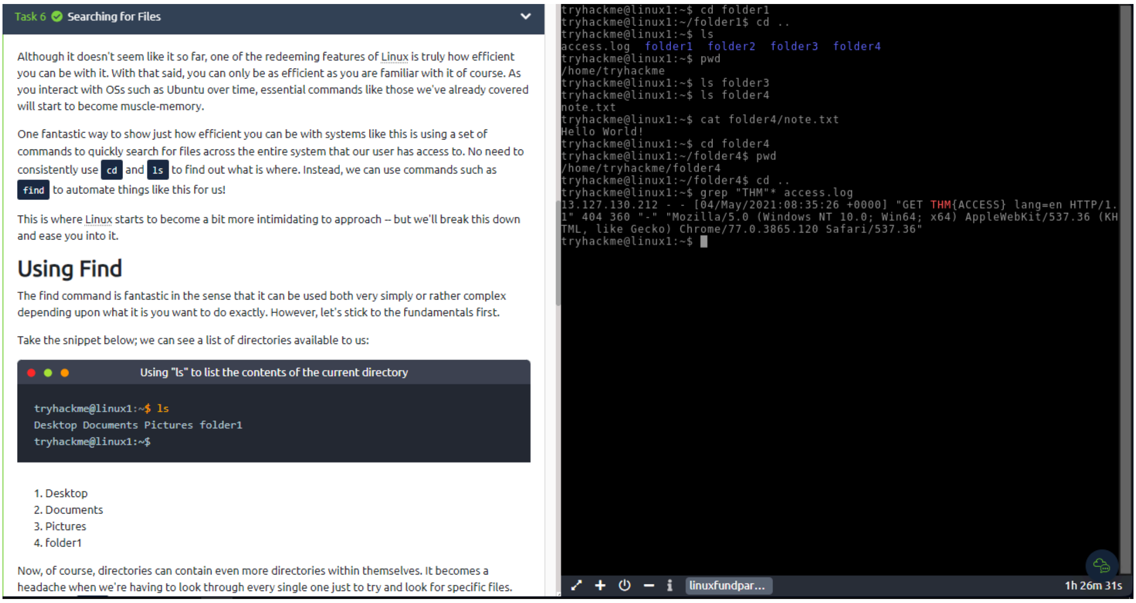
Listing Files in Our Current Directory (ls)

Before we can do anything such as finding out the contents of any files or folders, we need to know what exists in the first place. This can be done using the "ls" command (short for listing)

Using "ls" to to list the contents of the current directory

```
tryhackme@linux1:~$ cd folder1
tryhackme@linux1:~/folder1$ cd ..
tryhackme@linux1:~$ ls
access.log  folder1  folder2  folder3  folder4
tryhackme@linux1:~$ pwd
/home/tryhackme
tryhackme@linux1:~$ ls folder3
tryhackme@linux1:~$ ls folder4
note.txt
tryhackme@linux1:~$ cat folder4/note.txt
Hello World!
tryhackme@linux1:~$ cd folder4
tryhackme@linux1:~/folder4$ pwd
/home/tryhackme/folder4
tryhackme@linux1:~/folder4$ 
```

Task 6 : Searching for Files



Task 6 Searching for Files

Although it doesn't seem like it so far, one of the redeeming features of Linux is truly how efficient you can be with it. With that said, you can only be as efficient as you are familiar with it of course. As you interact with OSs such as Ubuntu over time, essential commands like those we've already covered will start to become muscle-memory.

One fantastic way to show just how efficient you can be with systems like this is using a set of commands to quickly search for files across the entire system that our user has access to. No need to consistently use `cd` and `ls` to find out what is where. Instead, we can use commands such as `find` to automate things like this for us!

This is where Linux starts to become a bit more intimidating to approach -- but we'll break this down and ease you into it.

Using Find

The find command is fantastic in the sense that it can be used both very simply or rather complex depending upon what it is you want to do exactly. However, let's stick to the fundamentals first.

Take the snippet below; we can see a list of directories available to us:

Using "ls" to list the contents of the current directory

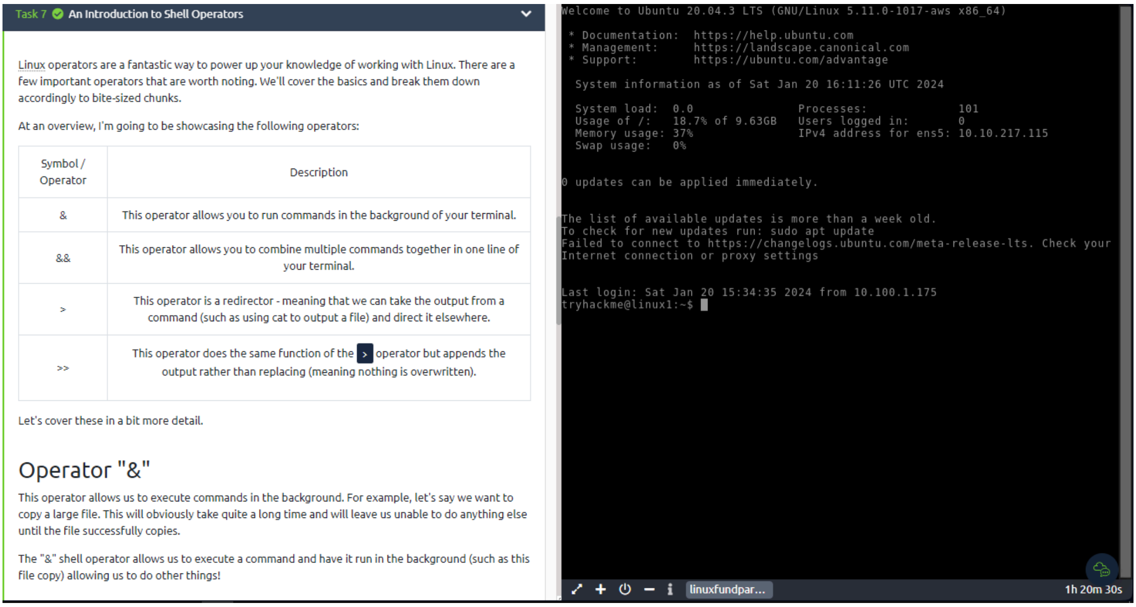
```
tryhackme@linux1:~$ ls
Desktop Documents Pictures folder1
tryhackme@linux1:~$
```

1. Desktop
2. Documents
3. Pictures
4. folder1

Now, of course, directories can contain even more directories within themselves. It becomes a headache when we're having to look through every single one just to try and look for specific files.

```
tryhackme@linux1:~$ cd folder1
tryhackme@linux1:~/folder1$ cd ..
tryhackme@linux1:~$ ls
access.log  folder1  folder2  folder3  folder4
tryhackme@linux1:~$ pwd
/home/tryhackme
tryhackme@linux1:~$ ls folder3
tryhackme@linux1:~$ ls folder4
note.txt
tryhackme@linux1:~$ cat folder4/note.txt
Hello World!
tryhackme@linux1:~$ cd folder4
tryhackme@linux1:~/folder4$ pwd
/home/tryhackme/folder4
tryhackme@linux1:~/folder4$ cd ..
tryhackme@linux1:~$ grep "THM"* access.log
13.127.130.212 - - [04/May/2021:08:35:26 +0000] "GET THM{ACCESS} lang=en HTTP/1.
1" 404 360 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KH
TML, like Gecko) Chrome/77.0.3865.120 Safari/537.36"
tryhackme@linux1:~$ 
```

Task 7 : An Introduction to Shell Operators



Task 7 ✓ An Introduction to Shell Operators

Linux operators are a fantastic way to power up your knowledge of working with Linux. There are a few important operators that are worth noting. We'll cover the basics and break them down accordingly to bite-sized chunks.

At an overview, I'm going to be showcasing the following operators:

Symbol / Operator	Description
&	This operator allows you to run commands in the background of your terminal.
&&	This operator allows you to combine multiple commands together in one line of your terminal.
>	This operator is a redirector - meaning that we can take the output from a command (such as using cat to output a file) and direct it elsewhere.
>>	This operator does the same function of the > operator but appends the output rather than replacing (meaning nothing is overwritten).

Let's cover these in a bit more detail.

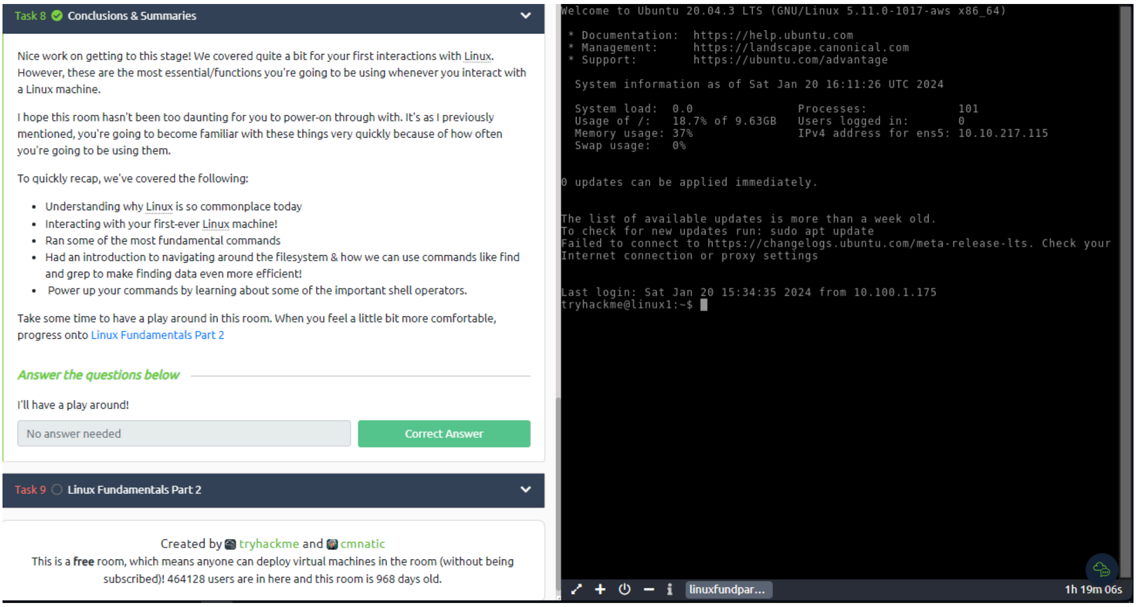
Operator "&"

This operator allows us to execute commands in the background. For example, let's say we want to copy a large file. This will obviously take quite a long time and will leave us unable to do anything else until the file successfully copies.

The "&" shell operator allows us to execute a command and have it run in the background (such as this file copy) allowing us to do other things!

```
Welcome to Ubuntu 20.04.3 LTS (GNU/Linux 5.11.0-1017-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Sat Jan 20 16:11:26 UTC 2024

  System load:  0.0               Processes:             101
  Usage of /:   18.7% of 9.63GB   Users logged in:       0
  Memory usage: 37%               IPv4 address for ens5: 10.10.217.115
  Swap usage:   0%


0 updates can be applied immediately.


The list of available updates is more than a week old.
To check for new updates run: sudo apt update
Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your
Internet connection or proxy settings


Last login: Sat Jan 20 15:34:35 2024 from 10.100.1.175
tryhackme@linux1:~$ 
```

Task 8 : Conclusions & Summaries



Task 8 ✓ Conclusions & Summaries

Nice work on getting to this stage! We covered quite a bit for your first interactions with Linux. However, these are the most essential/functions you're going to be using whenever you interact with a Linux machine.

I hope this room hasn't been too daunting for you to power-on through with. It's as I previously mentioned, you're going to become familiar with these things very quickly because of how often you're going to be using them.

To quickly recap, we've covered the following:

- Understanding why Linux is so commonplace today
- Interacting with your first-ever Linux machine!
- Ran some of the most fundamental commands
- Had an introduction to navigating around the filesystem & how we can use commands like find and grep to make finding data even more efficient!
- Power up your commands by learning about some of the important shell operators.

Take some time to have a play around in this room. When you feel a little bit more comfortable, progress onto Linux Fundamentals Part 2

Answer the questions below

I'll have a play around!

No answer needed

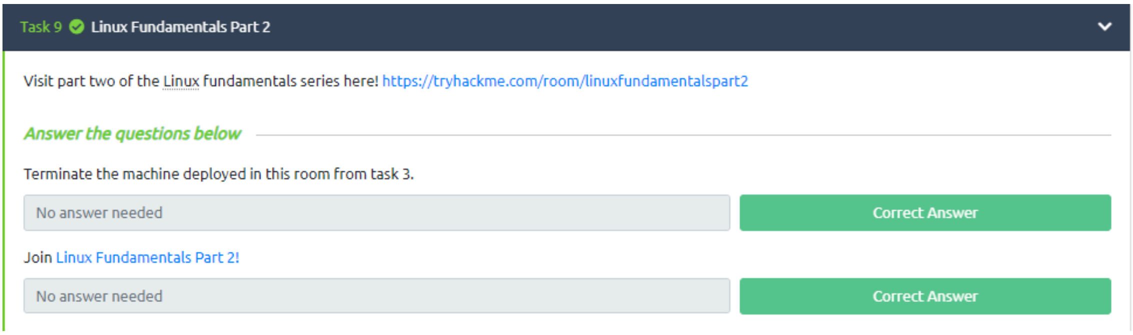
Correct Answer

Task 9 ○ Linux Fundamentals Part 2

Created by tryhackme and cmnatic
This is a **free** room, which means anyone can deploy virtual machines in the room (without being subscribed)! 464128 users are in here and this room is 968 days old.

```
Welcome to Ubuntu 20.04.3 LTS (GNU/Linux 5.11.0-1017-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Sat Jan 20 16:11:26 UTC 2024

  System load:  0.0               Processes:             101
  Usage of /:   18.7% of 9.63GB   Users logged in:       0
  Memory usage: 37%               IPv4 address for ens5: 10.10.217.115
  Swap usage:   0%


0 updates can be applied immediately.


The list of available updates is more than a week old.
To check for new updates run: sudo apt update
Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your
Internet connection or proxy settings


Last login: Sat Jan 20 15:34:35 2024 from 10.100.1.175
tryhackme@linux1:~$ 
```

Task 9 : Linux Fundamentals Part 2



Task 9 ✓ Linux Fundamentals Part 2

Visit part two of the Linux fundamentals series here! https://tryhackme.com/room/linuxfundamentalspart2

Answer the questions below

Terminate the machine deployed in this room from task 3.

No answer needed

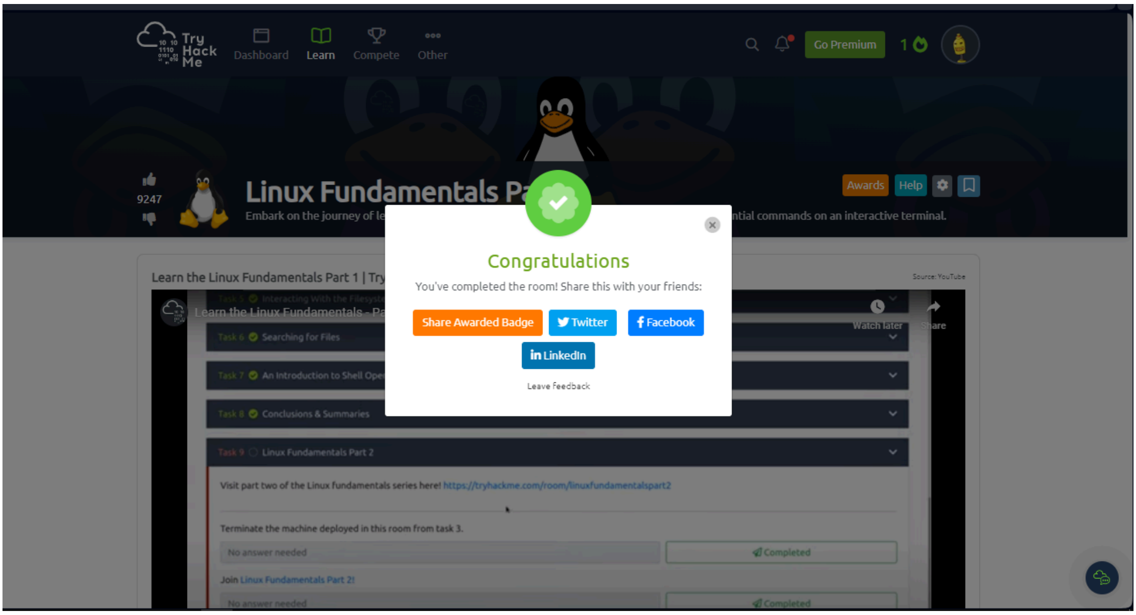
Correct Answer

Join Linux Fundamentals Part 2!

No answer needed

Correct Answer

Part 1 Completed :



Try Hack Me
Dashboard
Learn
Compete
Other
Go Premium
1
9247
Linux Fundamentals Pa
Embark on the journey of le
ntial commands on an interactive terminal.
Awards
Help
Learn the Linux Fundamentals Part 1 | Try
Source: YouTube
Learn the Linux Fundamentals - Pa
Watch later
Share

Congratulations

You've completed the room! Share this with your friends:

Share Awarded Badge

Twitter

Facebook

LinkedIn

Leave feedback

Task 6 ✓ Searching for Files

Task 7 ✓ An Introduction to Shell Ope

Task 8 ✓ Conclusions & Summaries

Task 9 ○ Linux Fundamentals Part 2

Visit part two of the Linux fundamentals series here! https://tryhackme.com/room/linuxfundamentalspart2

Terminate the machine deployed in this room from task 3.

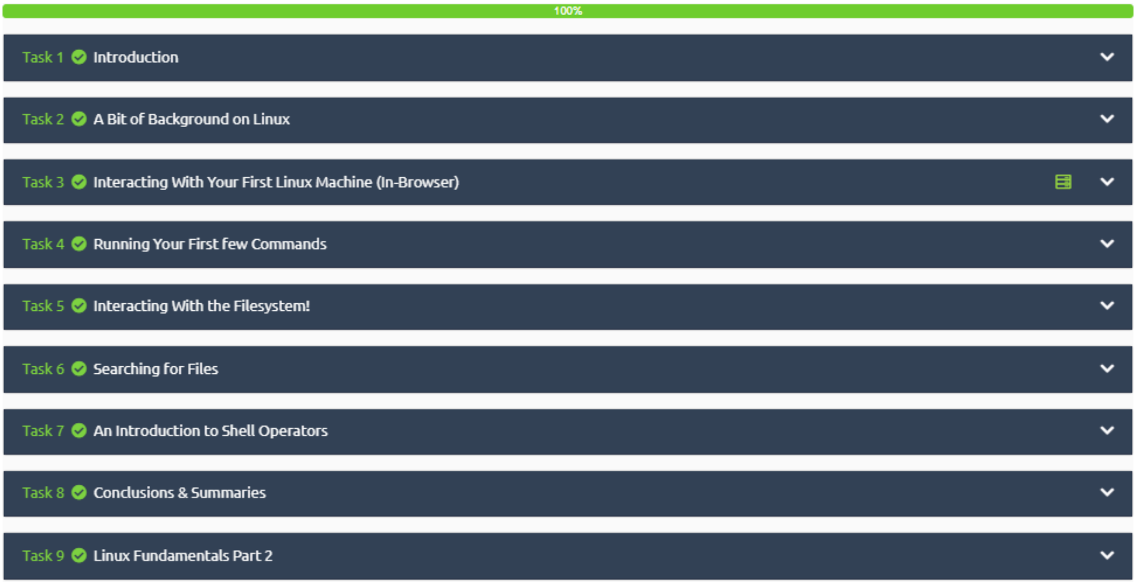
No answer needed

Completed

Join Linux Fundamentals Part 2!

No answer needed

Completed

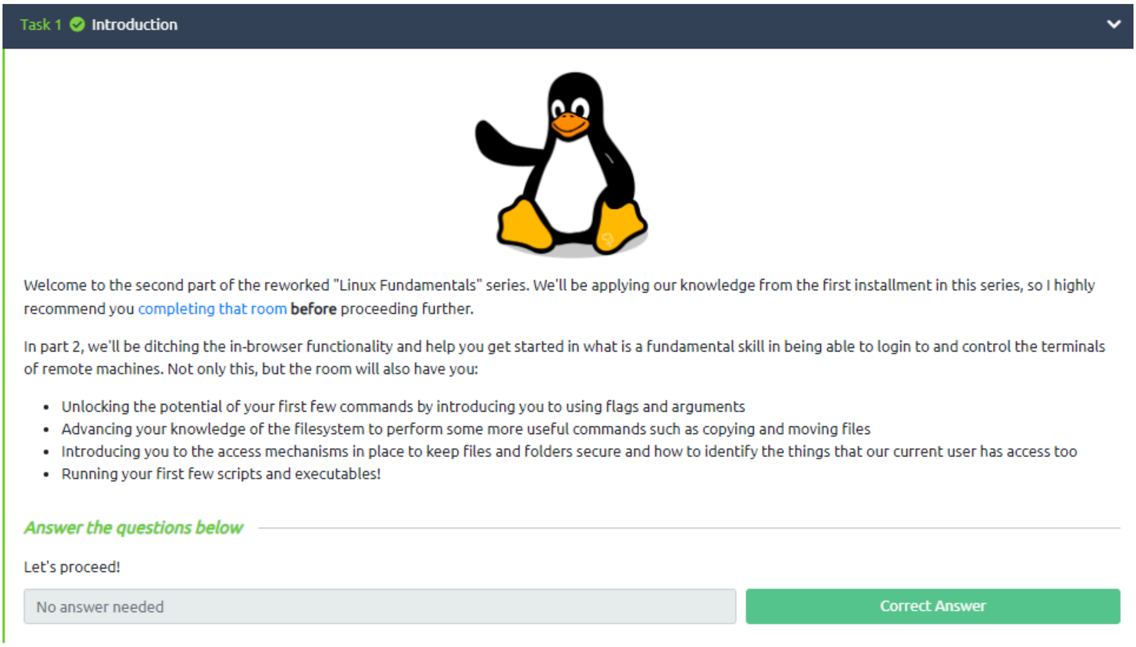


100%

- Task 1 ✓ Introduction
- Task 2 ✓ A Bit of Background on Linux
- Task 3 ✓ Interacting With Your First Linux Machine (In-Browser)
- Task 4 ✓ Running Your First few Commands
- Task 5 ✓ Interacting With the Filesystem!
- Task 6 ✓ Searching for Files
- Task 7 ✓ An Introduction to Shell Operators
- Task 8 ✓ Conclusions & Summaries
- Task 9 ✓ Linux Fundamentals Part 2

Linux Fundamentals Part 2 :

Task 1 : Introduction



Task 1 ✓ Introduction

Welcome to the second part of the reworked "Linux Fundamentals" series. We'll be applying our knowledge from the first installment in this series, so I highly recommend you completing that room **before** proceeding further.

In part 2, we'll be ditching the in-browser functionality and help you get started in what is a fundamental skill in being able to login to and control the terminals of remote machines. Not only this, but the room will also have you:

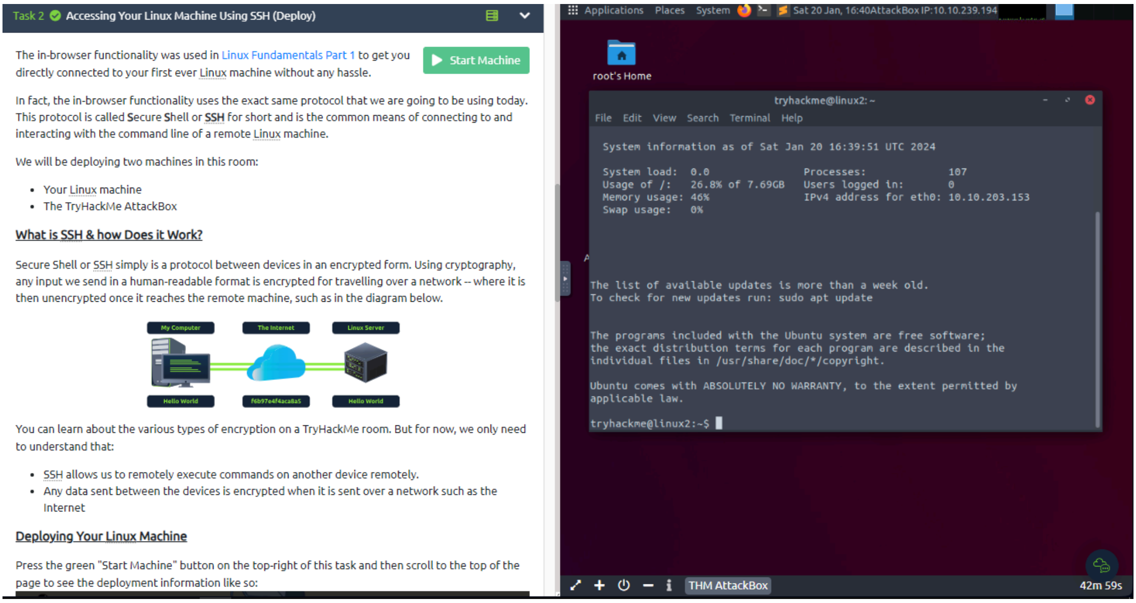
- Unlocking the potential of your first few commands by introducing you to using flags and arguments
- Advancing your knowledge of the filesystem to perform some more useful commands such as copying and moving files
- Introducing you to the access mechanisms in place to keep files and folders secure and how to identify the things that our current user has access too
- Running your first few scripts and executables!

Answer the questions below

Let's proceed!

No answer needed

Task 2 : Accessing your Linux Machine Using SSH (Deploy)



Task 2 ✓ Accessing Your Linux Machine Using SSH (Deploy)

The in-browser functionality was used in Linux Fundamentals Part 1 to get you directly connected to your first ever Linux machine without any hassle.

In fact, the in-browser functionality uses the exact same protocol that we are going to be using today. This protocol is called **S**ecure **S**hell or **SSH** for short and is the common means of connecting to and interacting with the command line of a remote Linux machine.

We will be deploying two machines in this room:

- Your Linux machine
- The TryHackMe AttackBox

What is SSH & how Does it Work?

Secure Shell or SSH simply is a protocol between devices in an encrypted form. Using cryptography, any input we send in a human-readable format is encrypted for travelling over a network -- where it is then unencrypted once it reaches the remote machine, such as in the diagram below.

You can learn about the various types of encryption on a TryHackMe room. But for now, we only need to understand that:

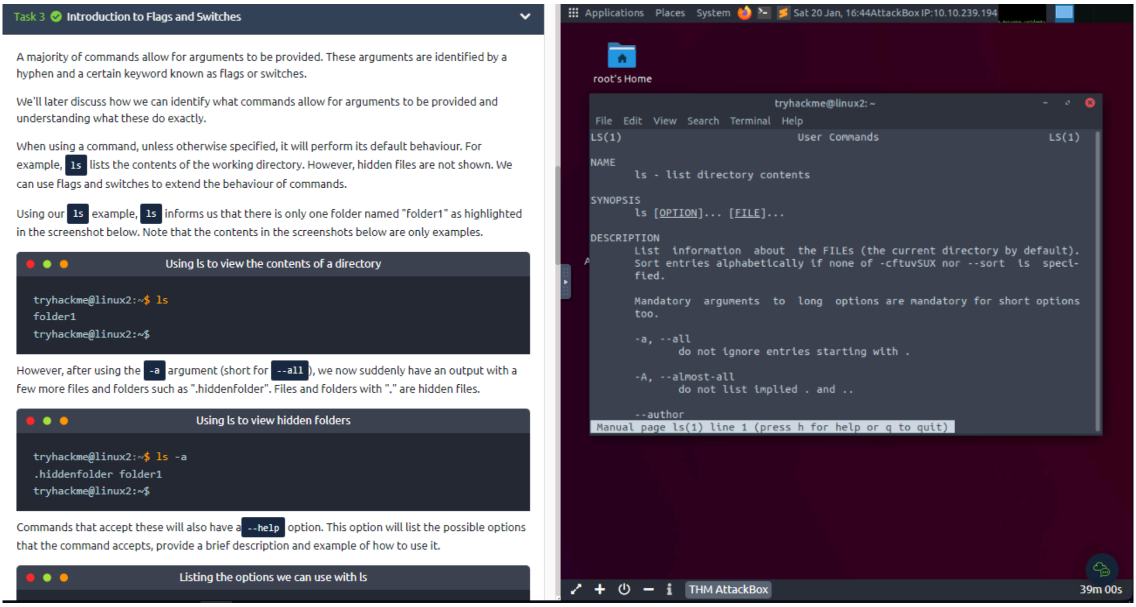
- SSH allows us to remotely execute commands on another device remotely.
- Any data sent between the devices is encrypted when it is sent over a network such as the Internet

Deploying Your Linux Machine

Press the green "Start Machine" button on the top-right of this task and then scroll to the top of the page to see the deployment information like so:

```
  System information as of Sat Jan 20 16:39:51 UTC 2024

  System load:  0.0               Processes:              107
  Usage of /:   26.8% of 7.69GB   Users logged in:        0
  Memory usage: 46%               IPv4 address for eth0: 10.10.203.153
  Swap usage:   0%



The list of available updates is more than a week old.
To check for new updates run: sudo apt update


The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

tryhackme@linux2:~$ 
```

Task 3 : Introduction to Flags and Switches



Task 3 Introduction to Flags and Switches
Applications Places System
Sat 20 Jan, 16:44AttackBox IP:10.10.239.194
root's Home
tryhackme@linux2: ~
File Edit View Search Terminal Help
THM AttackBox
39m 00s

A majority of commands allow for arguments to be provided. These arguments are identified by a hyphen and a certain keyword known as flags or switches.

We'll later discuss how we can identify what commands allow for arguments to be provided and understanding what these do exactly.

When using a command, unless otherwise specified, it will perform its default behaviour. For example, `ls` lists the contents of the working directory. However, hidden files are not shown. We can use flags and switches to extend the behaviour of commands.

Using our `ls` example, `ls` informs us that there is only one folder named "folder1" as highlighted in the screenshot below. Note that the contents in the screenshots below are only examples.

Using ls to view the contents of a directory

```
tryhackme@linux2:~$ ls
folder1
tryhackme@linux2:~$
```

However, after using the `-a` argument (short for `--all`), we now suddenly have an output with a few more files and folders such as ".hiddenfolder". Files and folders with "." are hidden files.

Using ls to view hidden folders

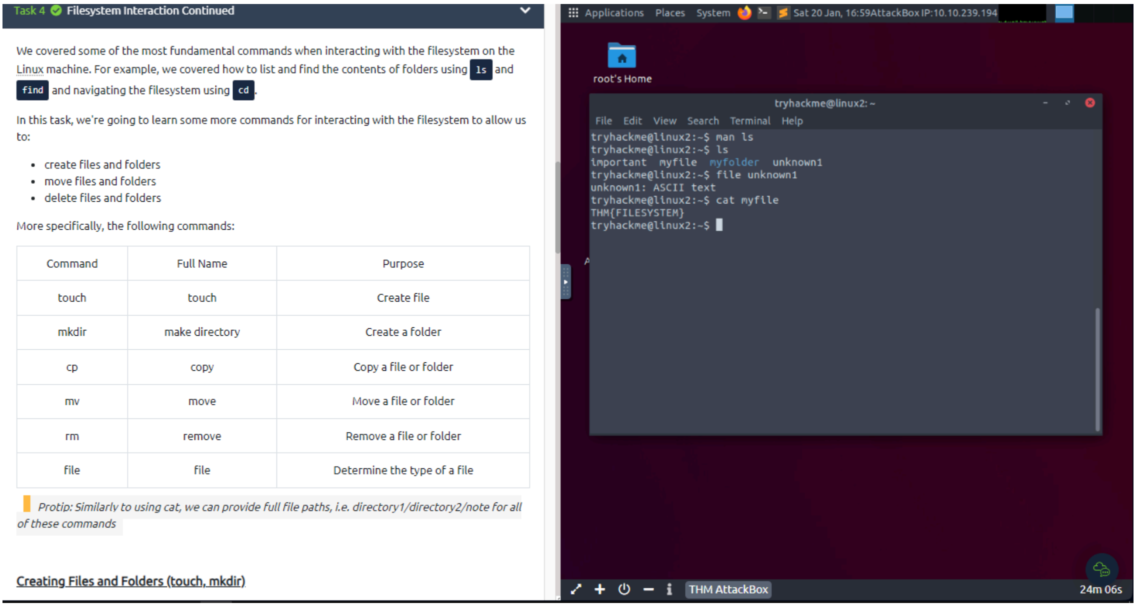
```
tryhackme@linux2:~$ ls -a
.hiddenfolder folder1
tryhackme@linux2:~$
```

Commands that accept these will also have a `--help` option. This option will list the possible options that the command accepts, provide a brief description and example of how to use it.

Listing the options we can use with ls

```
LS(1)                          User Commands                          LS(1)

NAME
       ls - list directory contents

SYNOPSIS
       ls [OPTION]... [FILE]...

DESCRIPTION
       List  information  about  the FILEs (the current directory by default).
       Sort entries alphabetically if none of -cftuvSUX nor --sort  is  speci-
       fied.

       Mandatory  arguments  to  long  options are mandatory for short options
       too.

       -a, --all
              do not ignore entries starting with .

       -A, --almost-all
              do not list implied . and ..

       --author
 Manual page ls(1) line 1 (press h for help or q to quit)
```

Task 4 : Filesystem Interaction Continued



Task 4 Filesystem Interaction Continued
Applications Places System
Sat 20 Jan, 16:59AttackBox IP:10.10.239.194
root's Home
tryhackme@linux2: ~
File Edit View Search Terminal Help
THM AttackBox
24m 06s

We covered some of the most fundamental commands when interacting with the filesystem on the Linux machine. For example, we covered how to list and find the contents of folders using `ls` and `find` and navigating the filesystem using `cd`.

In this task, we're going to learn some more commands for interacting with the filesystem to allow us to:

- create files and folders
- move files and folders
- delete files and folders

More specifically, the following commands:

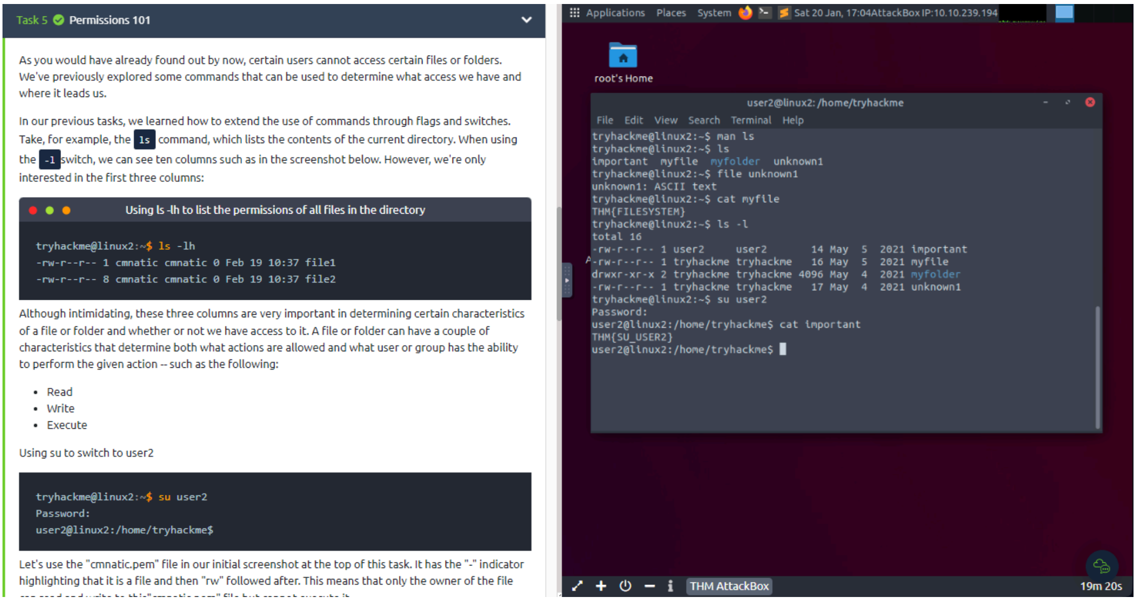
Command	Full Name	Purpose
touch	touch	Create file
mkdir	make directory	Create a folder
cp	copy	Copy a file or folder
mv	move	Move a file or folder
rm	remove	Remove a file or folder
file	file	Determine the type of a file

Protip: Similarly to using cat, we can provide full file paths, i.e. directory1/directory2/note for all of these commands

Creating Files and Folders (touch, mkdir)

```
tryhackme@linux2:~$ man ls
tryhackme@linux2:~$ ls
important  myfile  myfolder  unknown1
tryhackme@linux2:~$ file unknown1
unknown1: ASCII text
tryhackme@linux2:~$ cat myfile
THM{FILESYSTEM}
tryhackme@linux2:~$
```

Task 5 : Permissions 101



As you would have already found out by now, certain users cannot access certain files or folders. We've previously explored some commands that can be used to determine what access we have and where it leads us.

In our previous tasks, we learned how to extend the use of commands through flags and switches. Take, for example, the `ls` command, which lists the contents of the current directory. When using the `-l` switch, we can see ten columns such as in the screenshot below. However, we're only interested in the first three columns:

Using ls -lh to list the permissions of all files in the directory

```
tryhackme@linux2:~$ ls -lh
-rw-r--r-- 1 cmnatic cmnatic 0 Feb 19 10:37 file1
-rw-r--r-- 8 cmnatic cmnatic 0 Feb 19 10:37 file2
```

Although intimidating, these three columns are very important in determining certain characteristics of a file or folder and whether or not we have access to it. A file or folder can have a couple of characteristics that determine both what actions are allowed and what user or group has the ability to perform the given action -- such as the following:

- Read
- Write
- Execute

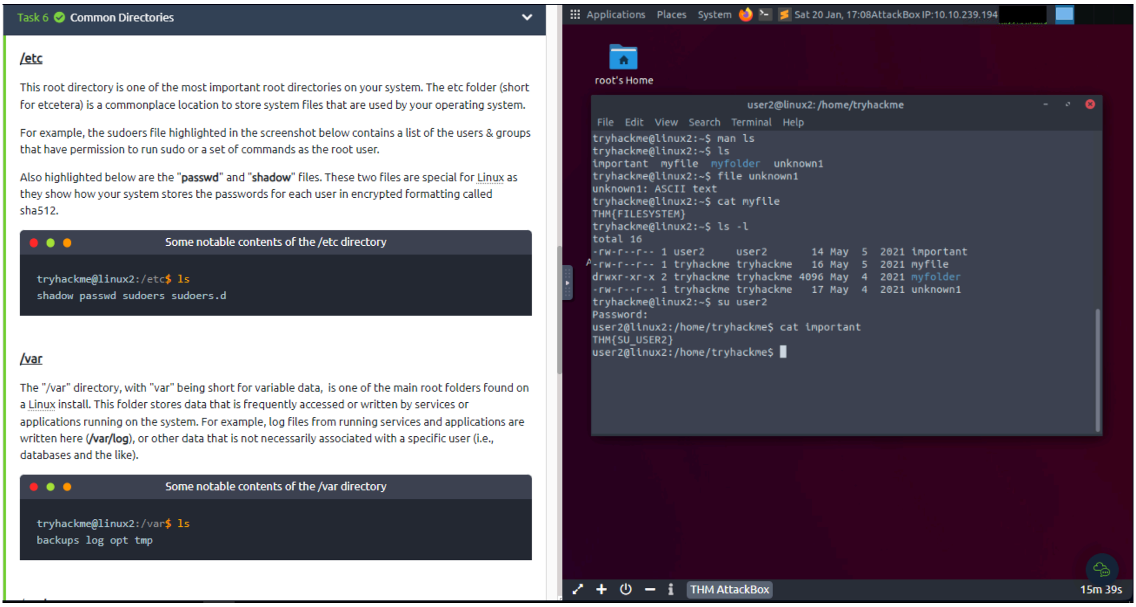
Using su to switch to user2

```
tryhackme@linux2:~$ su user2
Password:
user2@linux2:/home/tryhackme$
```

Let's use the "cmnatic.pem" file in our initial screenshot at the top of this task. It has the "-" indicator highlighting that it is a file and then "rw" followed after. This means that only the owner of the file can read and write to this"cmnatic.pem" file but cannot execute it.

```
tryhackme@linux2:~$ man ls
tryhackme@linux2:~$ ls
important  myfile  myfolder  unknown1
tryhackme@linux2:~$ file unknown1
unknown1: ASCII text
tryhackme@linux2:~$ cat myfile
THM{FILESYSTEM}
tryhackme@linux2:~$ ls -l
total 16
-rw-r--r-- 1 user2     user2       14 May  5  2021 important
-rw-r--r-- 1 tryhackme tryhackme   16 May  5  2021 myfile
drwxr-xr-x 2 tryhackme tryhackme 4096 May  4  2021 myfolder
-rw-r--r-- 1 tryhackme tryhackme   17 May  4  2021 unknown1
tryhackme@linux2:~$ su user2
Password:
user2@linux2:/home/tryhackme$ cat important
THM{SU_USER2}
user2@linux2:/home/tryhackme$ 
```

Task 6 : Common Directories



/etc

This root directory is one of the most important root directories on your system. The etc folder (short for etcetera) is a commonplace location to store system files that are used by your operating system.

For example, the sudoers file highlighted in the screenshot below contains a list of the users & groups that have permission to run sudo or a set of commands as the root user.

Also highlighted below are the "**passwd**" and "**shadow**" files. These two files are special for Linux as they show how your system stores the passwords for each user in encrypted formatting called sha512.

Some notable contents of the /etc directory

```
tryhackme@linux2:/etc$ ls
shadow passwd sudoers sudoers.d
```

/var

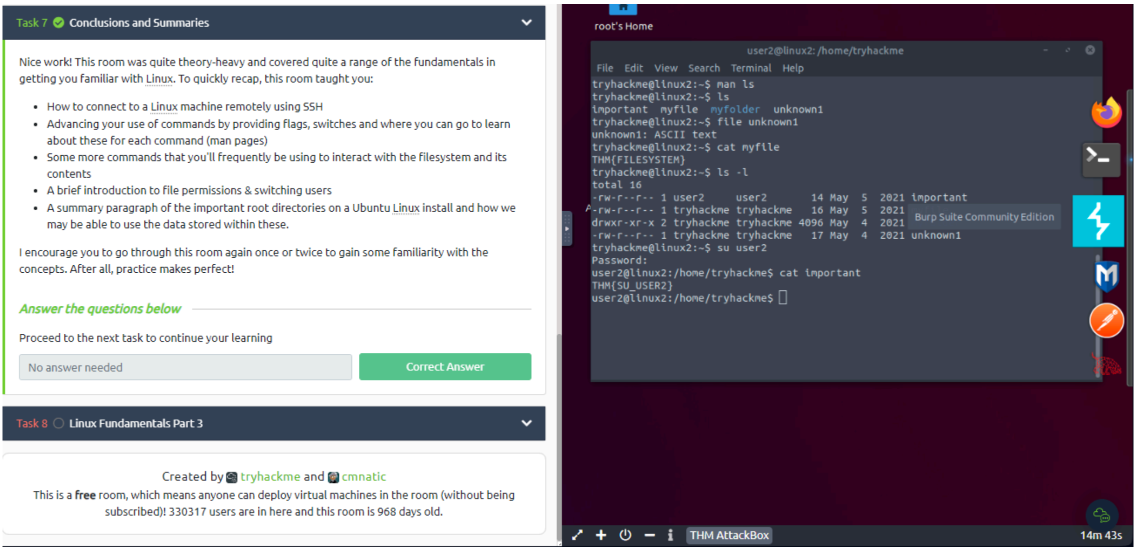
The "/var" directory, with "var" being short for variable data, is one of the main root folders found on a Linux install. This folder stores data that is frequently accessed or written by services or applications running on the system. For example, log files from running services and applications are written here (**/var/log**), or other data that is not necessarily associated with a specific user (i.e., databases and the like).

Some notable contents of the /var directory

```
tryhackme@linux2:/var$ ls
backups log opt tmp
```

```
tryhackme@linux2:~$ man ls
tryhackme@linux2:~$ ls
important  myfile  myfolder  unknown1
tryhackme@linux2:~$ file unknown1
unknown1: ASCII text
tryhackme@linux2:~$ cat myfile
THM{FILESYSTEM}
tryhackme@linux2:~$ ls -l
total 16
-rw-r--r-- 1 user2     user2       14 May  5  2021 important
-rw-r--r-- 1 tryhackme tryhackme   16 May  5  2021 myfile
drwxr-xr-x 2 tryhackme tryhackme 4096 May  4  2021 myfolder
-rw-r--r-- 1 tryhackme tryhackme   17 May  4  2021 unknown1
tryhackme@linux2:~$ su user2
Password:
user2@linux2:/home/tryhackme$ cat important
THM{SU_USER2}
user2@linux2:/home/tryhackme$ 
```

Task 7 : Conclusion and Summaries



Task 7 ✓ Conclusions and Summaries

Nice work! This room was quite theory-heavy and covered quite a range of the fundamentals in getting you familiar with Linux. To quickly recap, this room taught you:

- How to connect to a Linux machine remotely using SSH
- Advancing your use of commands by providing flags, switches and where you can go to learn about these for each command (man pages)
- Some more commands that you'll frequently be using to interact with the filesystem and its contents
- A brief introduction to file permissions & switching users
- A summary paragraph of the important root directories on a Ubuntu Linux install and how we may be able to use the data stored within these.

I encourage you to go through this room again once or twice to gain some familiarity with the concepts. After all, practice makes perfect!

Answer the questions below

Proceed to the next task to continue your learning

No answer needed

Correct Answer

Task 8 ○ Linux Fundamentals Part 3

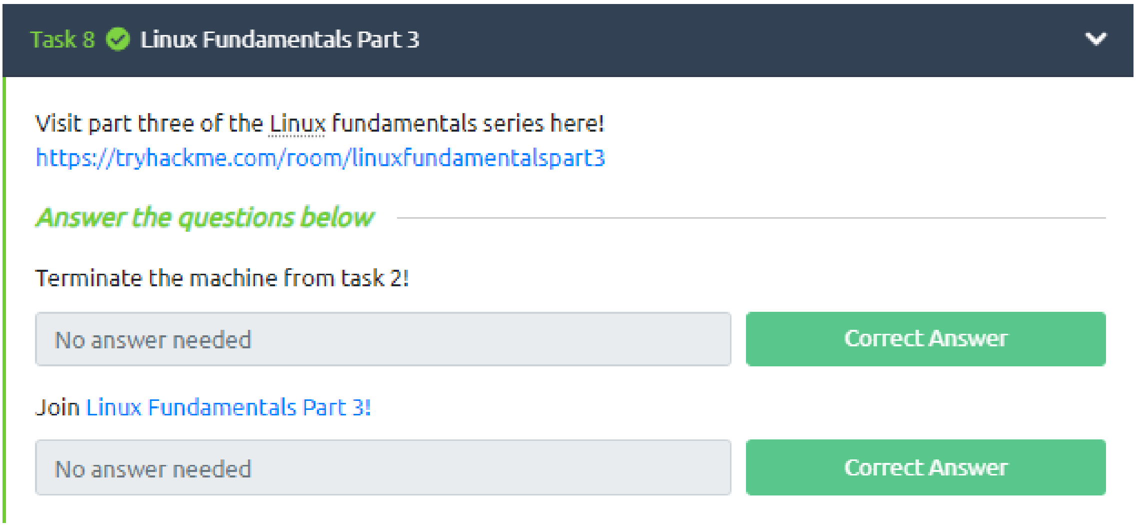
Created by tryhackme and cmnatic

This is a **free** room, which means anyone can deploy virtual machines in the room (without being subscribed)! 330317 users are in here and this room is 968 days old.

```
tryhackme@linux2:~$ man ls
tryhackme@linux2:~$ ls
important  myfile  myfolder  unknown1
tryhackme@linux2:~$ file unknown1
unknown1: ASCII text
tryhackme@linux2:~$ cat myfile
THM{FILESYSTEM}
tryhackme@linux2:~$ ls -l
total 16
-rw-r--r-- 1 user2     user2       14 May  5  2021 important
-rw-r--r-- 1 tryhackme tryhackme   16 May  5  2021
drwxr-xr-x 2 tryhackme tryhackme 4096 May  4  2021
-rw-r--r-- 1 tryhackme tryhackme   17 May  4  2021 unknown1
tryhackme@linux2:~$ su user2
Password:
user2@linux2:/home/tryhackme$ cat important
THM{SU_USER2}
user2@linux2:/home/tryhackme$ 
```

root's Home
user2@linux2: /home/tryhackme
File Edit View Search Terminal Help
Burp Suite Community Edition
THM AttackBox
14m 43s

Task 8 : Linux Fundamental Part 3



Task 8 ✓ Linux Fundamentals Part 3

Visit part three of the Linux fundamentals series here!
https://tryhackme.com/room/linuxfundamentalspart3

Answer the questions below

Terminate the machine from task 2!

No answer needed

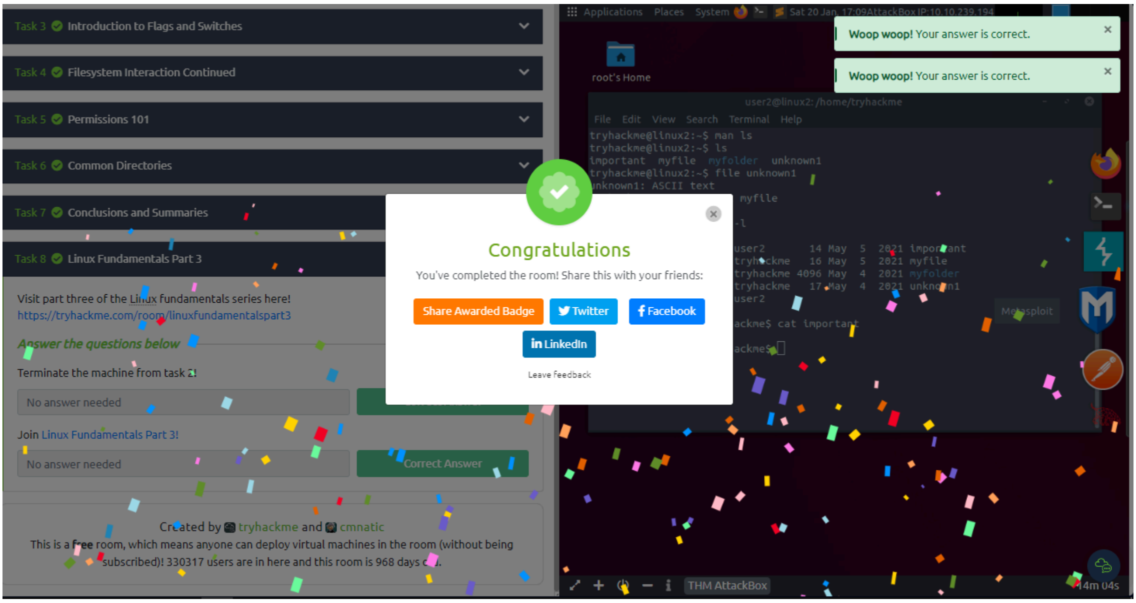
Correct Answer

Join Linux Fundamentals Part 3!

No answer needed

Correct Answer

Part 2 Completed :



Task 3 Introduction to Flags and Switches
Task 4 Filesystem Interaction Continued
Task 5 Permissions 101
Task 6 Common Directories
Task 7 Conclusions and Summaries
Task 8 Linux Fundamentals Part 3
Applications
Places
System
root's Home
File
Edit
View
Search
Terminal
Help
Metasploit
THM AttackBox
14m 04s

Sat 20 Jan, 17:09AttackBox IP:10.10.239.194

Visit part three of the Linux fundamentals series here!
https://tryhackme.com/room/linuxfundamentalspart3

Answer the questions below

Terminate the machine from task 2!

No answer needed

Join Linux Fundamentals Part 3!

No answer needed

Correct Answer

Created by tryhackme and cmnatic

This is a **free** room, which means anyone can deploy virtual machines in the room (without being subscribed)! 330317 users are in here and this room is 968 days old.

Congratulations

You've completed the room! Share this with your friends:

Share Awarded Badge Twitter Facebook

LinkedIn

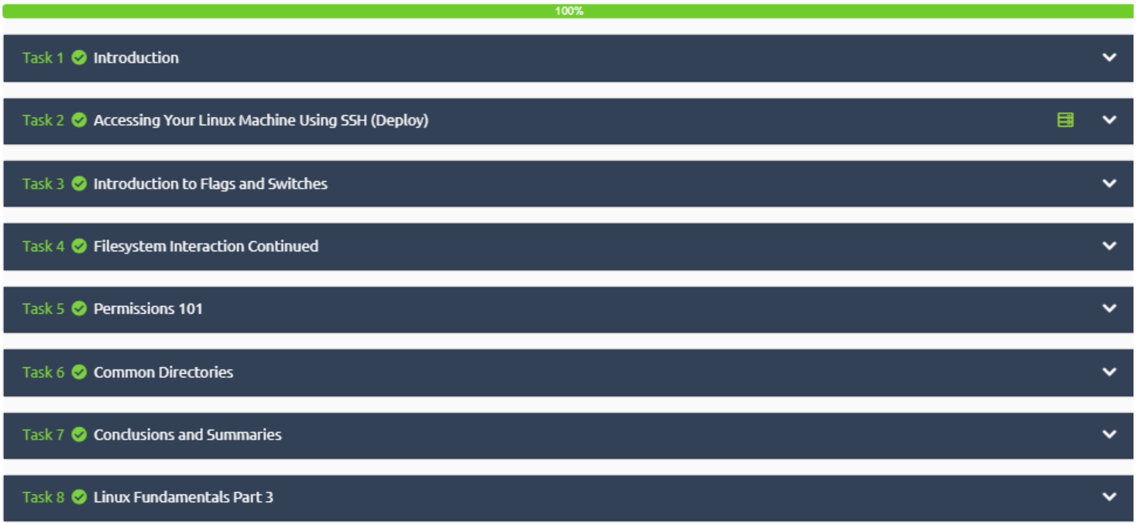
Leave feedback

Woop woop! Your answer is correct.

Woop woop! Your answer is correct.

user2@linux2: /home/tryhackme

```
tryhackme@linux2:~$ man ls
tryhackme@linux2:~$ ls
important  myfile  myfolder  unknown1
tryhackme@linux2:~$ file unknown1
unknown1: ASCII text
                                myfile

                              -l

                            user2        14 May  5  2021 important
                            tryhackme    16 May  5  2021 myfile
                            tryhackme  4096 May  4  2021 myfolder
                            tryhackme    17 May  4  2021 unknown1
                            user2

                            ackme$ cat important

                            ackme$
```



100%

Task 1 Introduction

Task 2 Accessing Your Linux Machine Using SSH (Deploy)

Task 3 Introduction to Flags and Switches

Task 4 Filesystem Interaction Continued

Task 5 Permissions 101

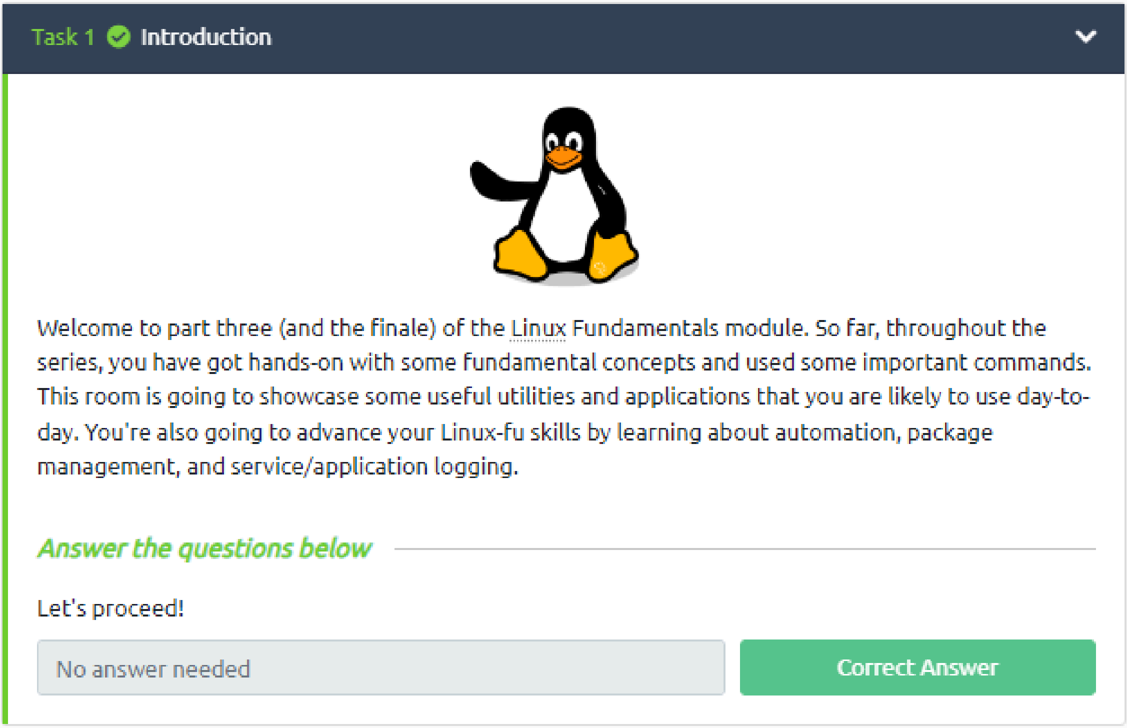
Task 6 Common Directories

Task 7 Conclusions and Summaries

Task 8 Linux Fundamentals Part 3

Linux Fundamentals Part 3 :

Task 1 : Introduction



Task 1 ✓ Introduction

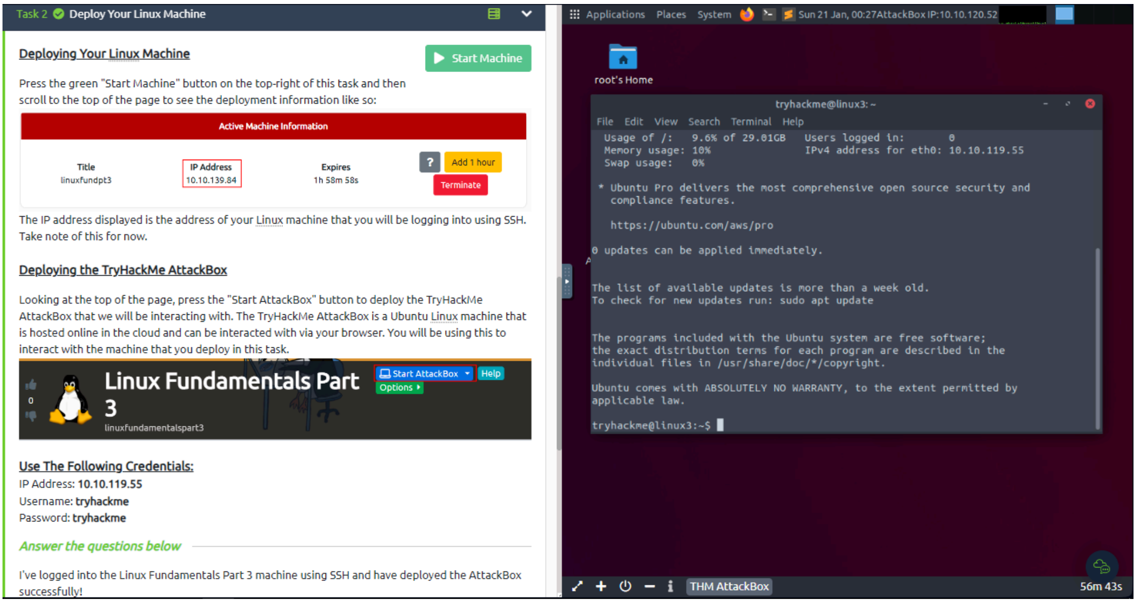
Welcome to part three (and the finale) of the Linux Fundamentals module. So far, throughout the series, you have got hands-on with some fundamental concepts and used some important commands. This room is going to showcase some useful utilities and applications that you are likely to use day-to-day. You're also going to advance your Linux-fu skills by learning about automation, package management, and service/application logging.

Answer the questions below

Let's proceed!

No answer needed

Task 2 : Deploy your Linux Machine



Task 2 ✓ Deploy Your Linux Machine

Deploying Your Linux Machine

Press the green "Start Machine" button on the top-right of this task and then scroll to the top of the page to see the deployment information like so:

Active Machine Information

Title	IP Address	Expires
linuxfundpt3	10.10.139.84	1h 58m 58s

The IP address displayed is the address of your Linux machine that you will be logging into using SSH. Take note of this for now.

Deploying the TryHackMe AttackBox

Looking at the top of the page, press the "Start AttackBox" button to deploy the TryHackMe AttackBox that we will be interacting with. The TryHackMe AttackBox is a Ubuntu Linux machine that is hosted online in the cloud and can be interacted with via your browser. You will be using this to interact with the machine that you deploy in this task.

Use The Following Credentials:
IP Address: **10.10.119.55**
Username: **tryhackme**
Password: **tryhackme**

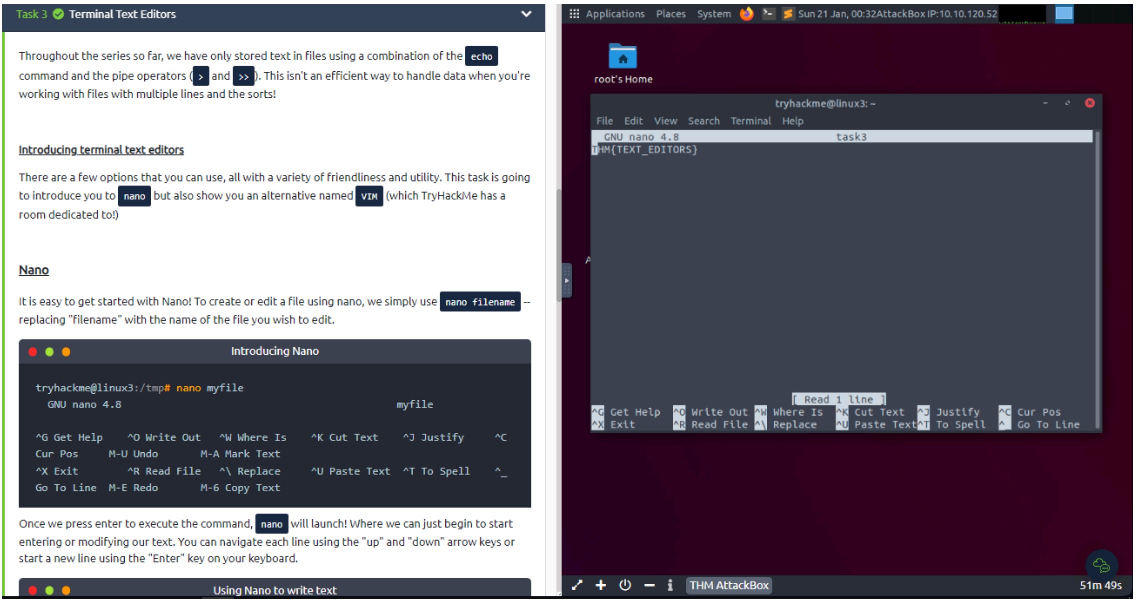
Answer the questions below

I've logged into the Linux Fundamentals Part 3 machine using SSH and have deployed the AttackBox successfully!

Applications
Places
System
Sun 21 Jan, 00:27AttackBox IP:10.10.120.52
root's Home
tryhackme@linux3: ~
File Edit View Search Terminal Help
THM AttackBox
56m 43s

```
  Usage of /:   9.6% of 29.01GB   Users logged in:       0
  Memory usage: 10%               IPv4 address for eth0: 10.10.119.55
  Swap usage:   0%

 * Ubuntu Pro delivers the most comprehensive open source security and
   compliance features.

   https://ubuntu.com/aws/pro

0 updates can be applied immediately.


The list of available updates is more than a week old.
To check for new updates run: sudo apt update


The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

tryhackme@linux3:~$
```

Task 3 : Terminal Text Editors



Task 3 Terminal Text Editors

Throughout the series so far, we have only stored text in files using a combination of the `echo` command and the pipe operators (`>` and `>>`). This isn't an efficient way to handle data when you're working with files with multiple lines and the sorts!

Introducing terminal text editors

There are a few options that you can use, all with a variety of friendliness and utility. This task is going to introduce you to `nano` but also show you an alternative named `VIM` (which TryHackMe has a room dedicated to!)

Nano

It is easy to get started with Nano! To create or edit a file using nano, we simply use `nano filename` -- replacing "filename" with the name of the file you wish to edit.

Introducing Nano

```
tryhackme@linux3:/tmp# nano myfile
  GNU nano 4.8                                  myfile

^G Get Help    ^O Write Out   ^W Where Is    ^K Cut Text    ^J Justify     ^C
Cur Pos      M-U Undo        M-A Mark Text
^X Exit        ^R Read File   ^\ Replace     ^U Paste Text  ^T To Spell    ^_
Go To Line   M-E Redo        M-6 Copy Text
```

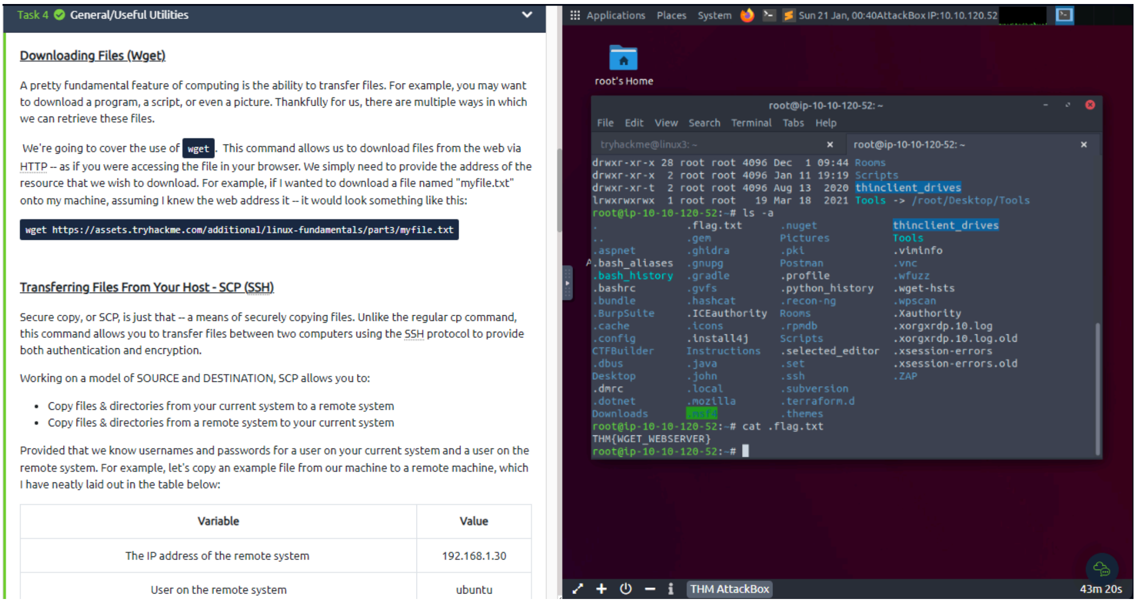
Once we press enter to execute the command, `nano` will launch! Where we can just begin to start entering or modifying our text. You can navigate each line using the "up" and "down" arrow keys or start a new line using the "Enter" key on your keyboard.

Using Nano to write text

Applications Places System
Sun 21 Jan, 00:32AttackBox IP:10.10.120.52
root's Home
tryhackme@linux3: ~
File Edit View Search Terminal Help

```
  GNU nano 4.8                          task3
THM{TEXT_EDITORS}




                          [ Read 1 line ]
^G Get Help  ^O Write Out ^W Where Is  ^K Cut Text  ^J Justify   ^C Cur Pos
^X Exit      ^R Read File ^\ Replace   ^U Paste Text^T To Spell  ^_ Go To Line
```

Task 4 : General / Useful Utilities



Task 4 General/Useful Utilities

Downloading Files (Wget)

A pretty fundamental feature of computing is the ability to transfer files. For example, you may want to download a program, a script, or even a picture. Thankfully for us, there are multiple ways in which we can retrieve these files.

We're going to cover the use of `wget`. This command allows us to download files from the web via HTTP -- as if you were accessing the file in your browser. We simply need to provide the address of the resource that we wish to download. For example, if I wanted to download a file named "myfile.txt" onto my machine, assuming I knew the web address it -- it would look something like this:

```
wget https://assets.tryhackme.com/additional/linux-fundamentals/part3/myfile.txt
```

Transferring Files From Your Host - SCP (SSH)

Secure copy, or SCP, is just that -- a means of securely copying files. Unlike the regular cp command, this command allows you to transfer files between two computers using the SSH protocol to provide both authentication and encryption.

Working on a model of SOURCE and DESTINATION, SCP allows you to:

- Copy files & directories from your current system to a remote system
- Copy files & directories from a remote system to your current system

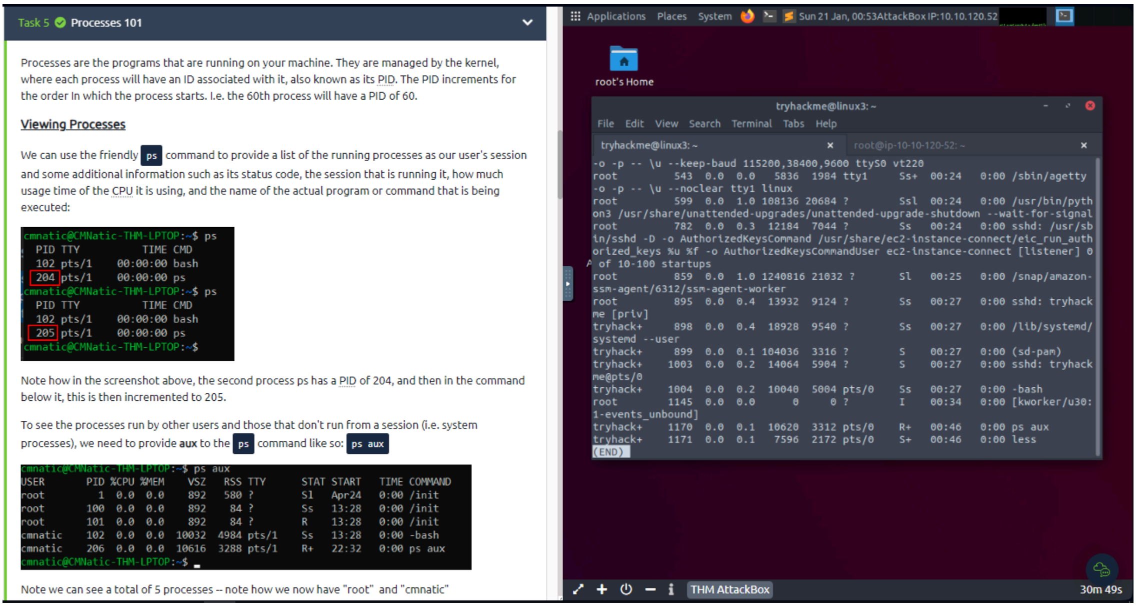
Provided that we know usernames and passwords for a user on your current system and a user on the remote system. For example, let's copy an example file from our machine to a remote machine, which I have neatly laid out in the table below:

Variable	Value
The IP address of the remote system	192.168.1.30
User on the remote system	ubuntu

Applications Places System
Sun 21 Jan, 00:40AttackBox IP:10.10.120.52
root's Home
root@ip-10-10-120-52: ~
File Edit View Search Terminal Tabs Help
tryhackme@linux3: ~
root@ip-10-10-120-52: ~

```
drwxr-xr-x 28 root root 4096 Dec  1 09:44 Rooms
drwxr-xr-x  2 root root 4096 Jan 11 19:19 Scripts
drwxr-xr-t  2 root root 4096 Aug 13  2020 thinclient_drives
lrwxrwxrwx  1 root root   19 Mar 18  2021 Tools -> /root/Desktop/Tools
root@ip-10-10-120-52:~# ls -a
.               .flag.txt       .nuget             thinclient_drives
..              .gem            Pictures           Tools
.aspnet         .ghidra         .pki               .viminfo
.bash_aliases   .gnupg          Postman            .vnc
.bash_history   .gradle         .profile           .wfuzz
.bashrc         .gvfs           .python_history    .wget-hsts
.bundle         .hashcat        .recon-ng          .wpscan
.BurpSuite      .ICEauthority   Rooms              .Xauthority
.cache          .icons          .rpmdb             .xorgxrdp.10.log
.config         .install4j      Scripts            .xorgxrdp.10.log.old
CTFBuilder      Instructions    .selected_editor   .xsession-errors
.dbus           .java           .set               .xsession-errors.old
Desktop         .john           .ssh               .ZAP
.dmrc           .local          .subversion
.dotnet         .mozilla        .terraform.d
Downloads       .msf4           .themes
root@ip-10-10-120-52:~# cat .flag.txt
THM{WGET_WEBSERVER}
root@ip-10-10-120-52:~# 
```

Task 5 : Processes 101



Processes are the programs that are running on your machine. They are managed by the kernel, where each process will have an ID associated with it, also known as its PID. The PID increments for the order In which the process starts. I.e. the 60th process will have a PID of 60.

Viewing Processes

We can use the friendly `ps` command to provide a list of the running processes as our user's session and some additional information such as its status code, the session that is running it, how much usage time of the CPU it is using, and the name of the actual program or command that is being executed:

```
cmnatic@CMNatic-THM-LPTOP:~$ ps
  PID TTY          TIME CMD
  102 pts/1    00:00:00 bash
  204 pts/1    00:00:00 ps
cmnatic@CMNatic-THM-LPTOP:~$ ps
  PID TTY          TIME CMD
  102 pts/1    00:00:00 bash
  205 pts/1    00:00:00 ps
cmnatic@CMNatic-THM-LPTOP:~$
```

Note how in the screenshot above, the second process ps has a PID of 204, and then in the command below it, this is then incremented to 205.

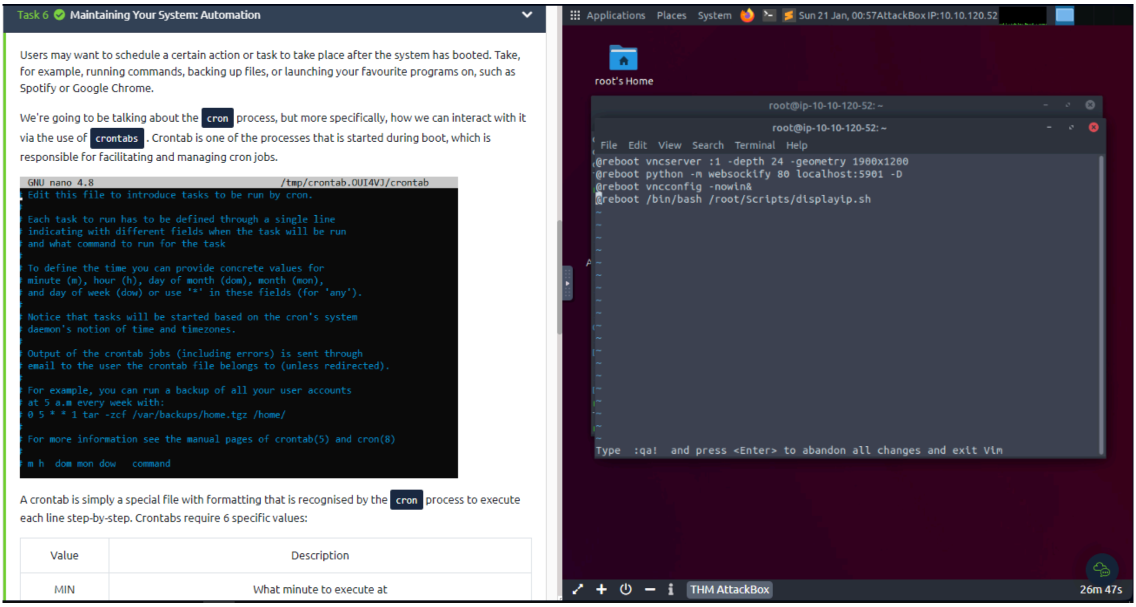
To see the processes run by other users and those that don't run from a session (i.e. system processes), we need to provide **aux** to the `ps` command like so: `ps aux`

```
cmnatic@CMNatic-THM-LPTOP:~$ ps aux
USER       PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root         1  0.0  0.0    892   580 ?        Sl   Apr24   0:00 /init
root       100  0.0  0.0    892    84 ?        Ss   13:28   0:00 /init
root       101  0.0  0.0    892    84 ?        R    13:28   0:00 /init
cmnatic    102  0.0  0.0  10032  4984 pts/1    Ss   13:28   0:00 -bash
cmnatic    206  0.0  0.0  10616  3288 pts/1    R+   22:32   0:00 ps aux
cmnatic@CMNatic-THM-LPTOP:~$ _
```

Note we can see a total of 5 processes -- note how we now have "root" and "cmnatic"

```
-o -p -- \u --keep-baud 115200,38400,9600 ttyS0 vt220
root         543  0.0  0.0   5836  1984 tty1     Ss+  00:24   0:00 /sbin/agetty
-o -p -- \u --noclear tty1 linux
root         599  0.0  1.0 108136 20684 ?        Ssl  00:24   0:00 /usr/bin/pyth
on3 /usr/share/unattended-upgrades/unattended-upgrade-shutdown --wait-for-signal
root         782  0.0  0.3  12184  7044 ?        Ss   00:24   0:00 sshd: /usr/sb
in/sshd -D -o AuthorizedKeysCommand /usr/share/ec2-instance-connect/eic_run_auth
orized_keys %u %f -o AuthorizedKeysCommandUser ec2-instance-connect [listener] 0
 of 10-100 startups
root         859  0.0  1.0 1240816 21032 ?       Sl   00:25   0:00 /snap/amazon-
ssm-agent/6312/ssm-agent-worker
root         895  0.0  0.4  13932  9124 ?        Ss   00:27   0:00 sshd: tryhack
me [priv]
tryhack+     898  0.0  0.4  18928  9540 ?        Ss   00:27   0:00 /lib/systemd/
systemd --user
tryhack+     899  0.0  0.1 104036  3316 ?        S    00:27   0:00 (sd-pam)
tryhack+    1003  0.0  0.2  14064  5904 ?        S    00:27   0:00 sshd: tryhack
me@pts/0
tryhack+    1004  0.0  0.2  10040  5004 pts/0    Ss   00:27   0:00 -bash
root        1145  0.0  0.0      0     0 ?        I    00:34   0:00 [kworker/u30:
1-events_unbound]
tryhack+    1170  0.0  0.1  10620  3312 pts/0    R+   00:46   0:00 ps aux
tryhack+    1171  0.0  0.1   7596  2172 pts/0    S+   00:46   0:00 less
(END)
```

Task 6 : Maintaining your system : Automation



Users may want to schedule a certain action or task to take place after the system has booted. Take, for example, running commands, backing up files, or launching your favourite programs on, such as Spotify or Google Chrome.

We're going to be talking about the `cron` process, but more specifically, how we can interact with it via the use of `crontabs`. Crontab is one of the processes that is started during boot, which is responsible for facilitating and managing cron jobs.

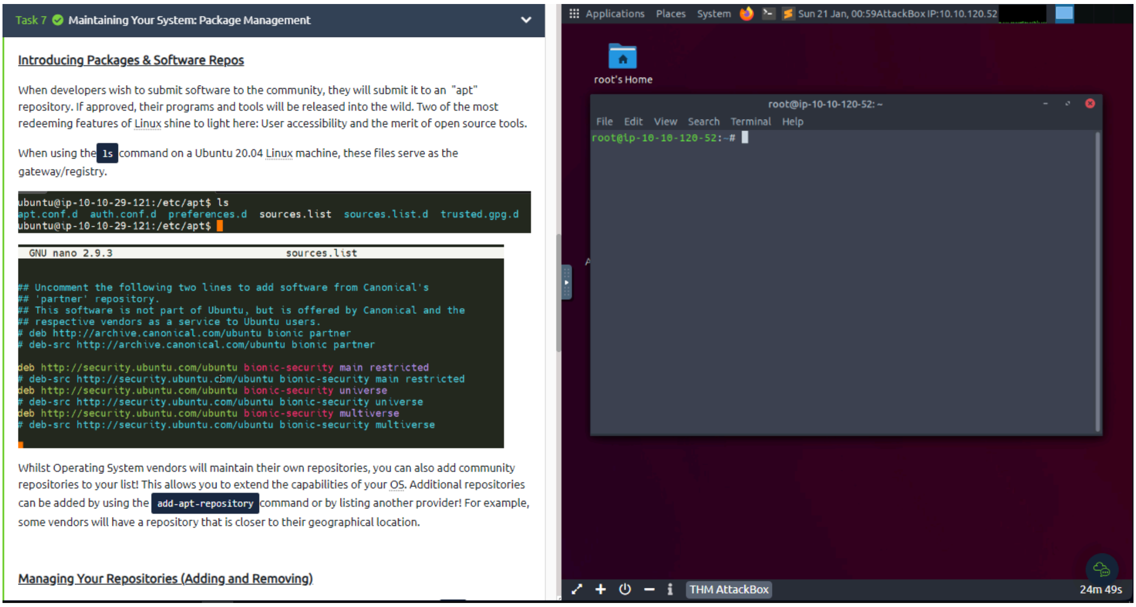
```
  GNU nano 4.8                              /tmp/crontab.OUI4VJ/crontab
# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').
#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h  dom mon dow   command
```

A crontab is simply a special file with formatting that is recognised by the `cron` process to execute each line step-by-step. Crontabs require 6 specific values:

Value	Description
MIN	What minute to execute at

```
@reboot vncserver :1 -depth 24 -geometry 1900x1200
@reboot python -m websockify 80 localhost:5901 -D
@reboot vncconfig -nowin&
@reboot /bin/bash /root/Scripts/displayip.sh
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
Type  :qa!  and press <Enter> to abandon all changes and exit Vim
```

Task 7 : Maintaining your system : Package Management



Introducing Packages & Software Repos

When developers wish to submit software to the community, they will submit it to an "apt" repository. If approved, their programs and tools will be released into the wild. Two of the most redeeming features of Linux shine to light here: User accessibility and the merit of open source tools.

When using the `ls` command on a Ubuntu 20.04 Linux machine, these files serve as the gateway/registry.

```
ubuntu@ip-10-10-29-121:/etc/apt$ ls
apt.conf.d  auth.conf.d  preferences.d  sources.list  sources.list.d  trusted.gpg.d
ubuntu@ip-10-10-29-121:/etc/apt$
```

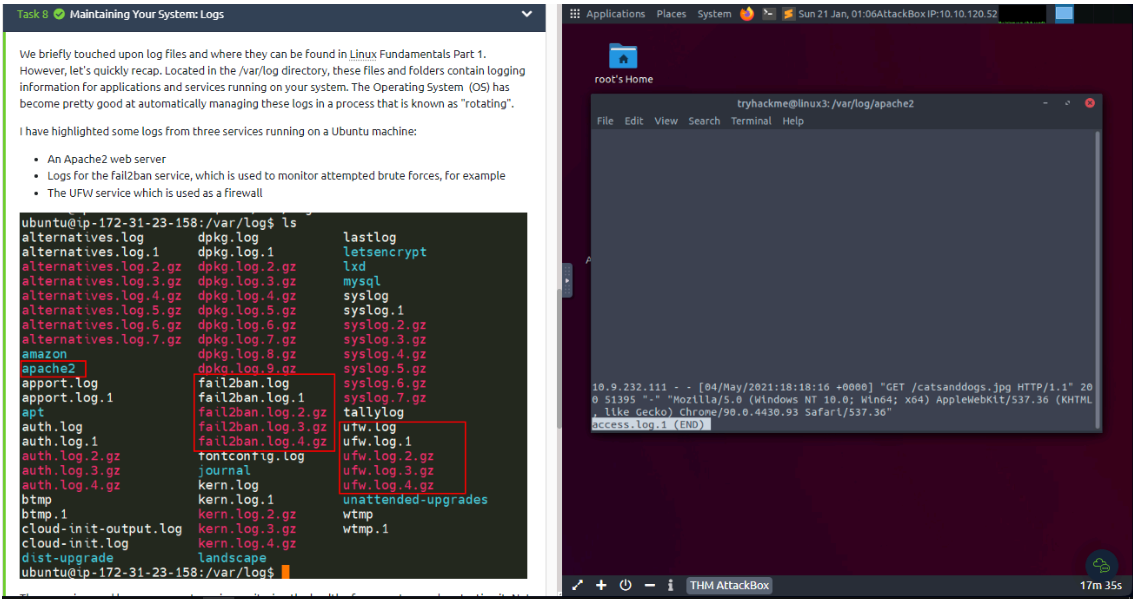
```
  GNU nano 2.9.3                              sources.list


## Uncomment the following two lines to add software from Canonical's
## 'partner' repository.
## This software is not part of Ubuntu, but is offered by Canonical and the
## respective vendors as a service to Ubuntu users.
# deb http://archive.canonical.com/ubuntu bionic partner
# deb-src http://archive.canonical.com/ubuntu bionic partner

deb http://security.ubuntu.com/ubuntu bionic-security main restricted
# deb-src http://security.ubuntu.com/ubuntu bionic-security main restricted
deb http://security.ubuntu.com/ubuntu bionic-security universe
# deb-src http://security.ubuntu.com/ubuntu bionic-security universe
deb http://security.ubuntu.com/ubuntu bionic-security multiverse
# deb-src http://security.ubuntu.com/ubuntu bionic-security multiverse
```

Whilst Operating System vendors will maintain their own repositories, you can also add community repositories to your list! This allows you to extend the capabilities of your OS. Additional repositories can be added by using the `add-apt-repository` command or by listing another provider! For example, some vendors will have a repository that is closer to their geographical location.

Managing Your Repositories (Adding and Removing)

```
root@ip-10-10-120-52:~#
```

Task 8 : Maintaining your system : Logs


Task 8
Maintaining Your System: Logs
Applications
Places
System
Sun 21 Jan, 01:06AttackBox IP:10.10.120.52
root's Home
tryhackme@linux3: /var/log/apache2
File
Edit
View
Search
Terminal
Help
THM AttackBox
17m 35s

We briefly touched upon log files and where they can be found in Linux Fundamentals Part 1. However, let's quickly recap. Located in the /var/log directory, these files and folders contain logging information for applications and services running on your system. The Operating System (OS) has become pretty good at automatically managing these logs in a process that is known as "rotating".

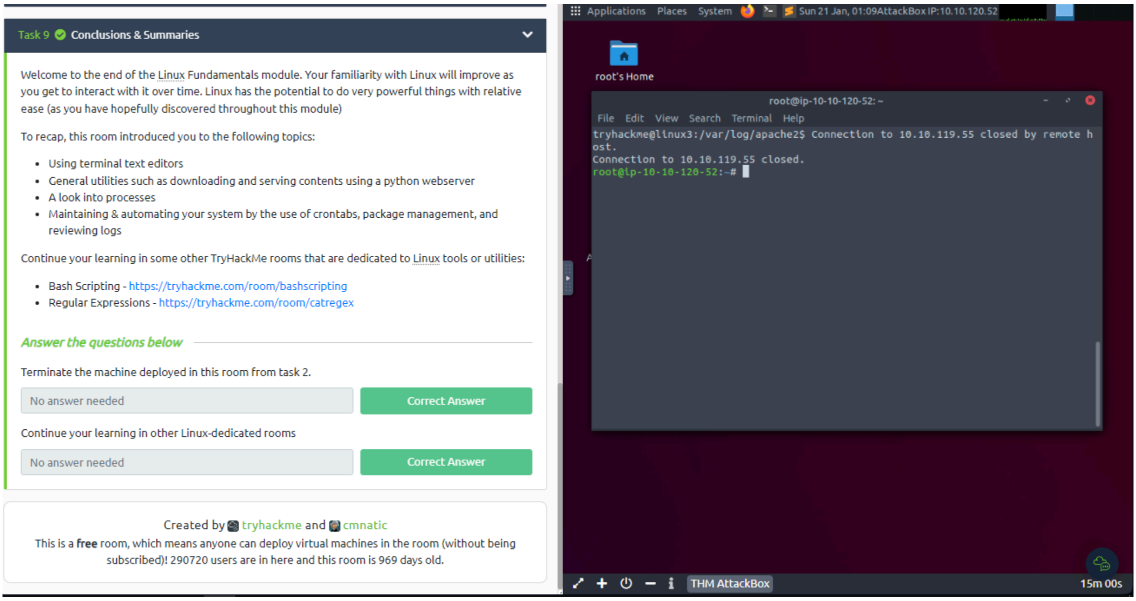
I have highlighted some logs from three services running on a Ubuntu machine:

- An Apache2 web server
- Logs for the fail2ban service, which is used to monitor attempted brute forces, for example
- The UFW service which is used as a firewall

```
ubuntu@ip-172-31-23-158:/var/log$ ls
alternatives.log        dpkg.log          lastlog
alternatives.log.1      dpkg.log.1        letsencrypt
alternatives.log.2.gz   dpkg.log.2.gz     lxd
alternatives.log.3.gz   dpkg.log.3.gz     mysql
alternatives.log.4.gz   dpkg.log.4.gz     syslog
alternatives.log.5.gz   dpkg.log.5.gz     syslog.1
alternatives.log.6.gz   dpkg.log.6.gz     syslog.2.gz
alternatives.log.7.gz   dpkg.log.7.gz     syslog.3.gz
amazon                  dpkg.log.8.gz     syslog.4.gz
apache2                 dpkg.log.9.gz     syslog.5.gz
apport.log              fail2ban.log      syslog.6.gz
apport.log.1            fail2ban.log.1    syslog.7.gz
apt                     fail2ban.log.2.gz tallylog
auth.log                fail2ban.log.3.gz ufw.log
auth.log.1              fail2ban.log.4.gz ufw.log.1
auth.log.2.gz           fontconfig.log    ufw.log.2.gz
auth.log.3.gz           journal           ufw.log.3.gz
auth.log.4.gz           kern.log          ufw.log.4.gz
btmp                    kern.log.1        unattended-upgrades
btmp.1                  kern.log.2.gz     wtmp
cloud-init-output.log   kern.log.3.gz     wtmp.1
cloud-init.log          kern.log.4.gz
dist-upgrade            landscape
ubuntu@ip-172-31-23-158:/var/log$
```

```
10.9.232.111 - - [04/May/2021:18:18:16 +0000] "GET /catsanddogs.jpg HTTP/1.1" 20
0 51395 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML
, like Gecko) Chrome/90.0.4430.93 Safari/537.36"
access.log.1 (END)
```

Task 9 : Conclusions and Summaries :



Task 9 Conclusions & Summaries

Welcome to the end of the Linux Fundamentals module. Your familiarity with Linux will improve as you get to interact with it over time. Linux has the potential to do very powerful things with relative ease (as you have hopefully discovered throughout this module)

To recap, this room introduced you to the following topics:

- Using terminal text editors
- General utilities such as downloading and serving contents using a python webserver
- A look into processes
- Maintaining & automating your system by the use of crontabs, package management, and reviewing logs

Continue your learning in some other TryHackMe rooms that are dedicated to Linux tools or utilities:

- Bash Scripting - https://tryhackme.com/room/bashscripting
- Regular Expressions - https://tryhackme.com/room/catregex

Answer the questions below

Terminate the machine deployed in this room from task 2.

No answer needed

Correct Answer

Continue your learning in other Linux-dedicated rooms

No answer needed

Correct Answer

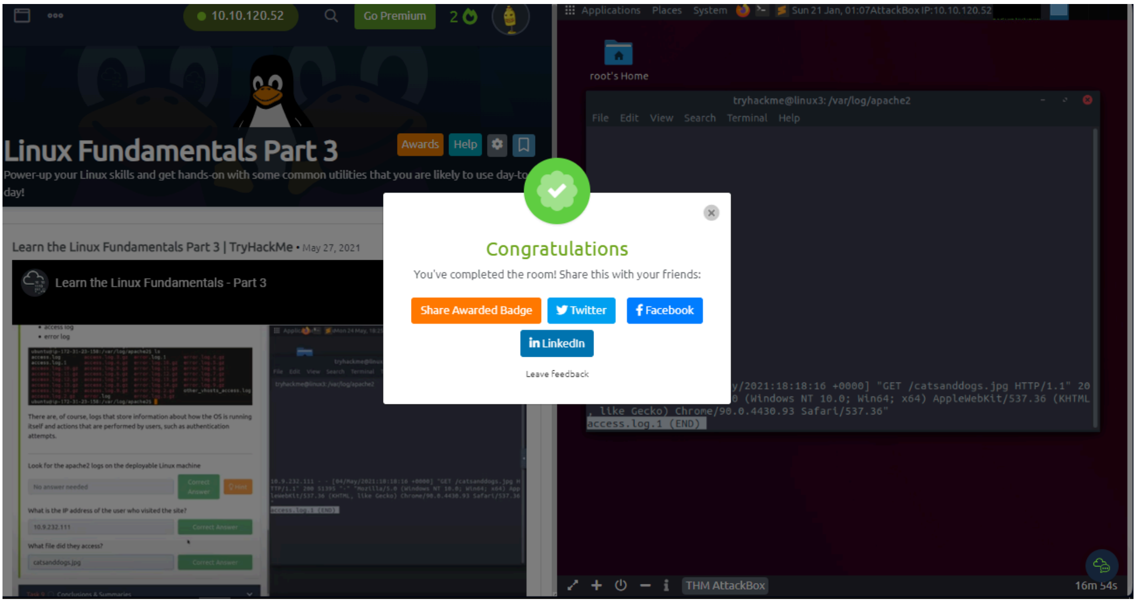
Created by tryhackme and cmnatic

This is a **free** room, which means anyone can deploy virtual machines in the room (without being subscribed)! 290720 users are in here and this room is 969 days old.

Applications Places System
Sun 21 Jan, 01:09AttackBox IP:10.10.120.52
root's Home
root@ip-10-10-120-52: ~
File Edit View Search Terminal Help
THM AttackBox
15m 00s

```
tryhackme@linux3:/var/log/apache2$ Connection to 10.10.119.55 closed by remote h
ost.
Connection to 10.10.119.55 closed.
root@ip-10-10-120-52:~# 
```

Part 3 Completed :



10.10.120.52
Go Premium
2
Awards
Help
Applications Places System
Sun 21 Jan, 01:07AttackBox IP:10.10.120.52
root's Home
tryhackme@linux3: /var/log/apache2
File Edit View Search Terminal Help
THM AttackBox
16m 54s

Linux Fundamentals Part 3

Power-up your Linux skills and get hands-on with some common utilities that you are likely to use day-to-day!

Learn the Linux Fundamentals Part 3 | TryHackMe • May 27, 2021

Learn the Linux Fundamentals - Part 3

- access log
- error log

There are, of course, logs that store information about how the OS is running itself and actions that are performed by users, such as authentication attempts.

Look for the apache2 logs on the deployable Linux machine

No answer needed

Correct Answer

Hint

What is the IP address of the user who visited the site?

10.9.232.111

Correct Answer

What file did they access?

catsanddogs.jpg

Correct Answer

Task 9 Conclusions & Summaries

Congratulations

You've completed the room! Share this with your friends:

Share Awarded Badge Twitter Facebook

LinkedIn

Leave Feedback

```
y/2021:18:18:16 +0000] "GET /catsanddogs.jpg HTTP/1.1" 20
0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML
, like Gecko) Chrome/90.0.4430.93 Safari/537.36"
access.log.1 (END)
```

100%

Task 1 ✓ Introduction

Task 2 ✓ Deploy Your Linux Machine

Task 3 ✓ Terminal Text Editors

Task 4 ✓ General/Useful Utilities

Task 5 ✓ Processes 101

Task 6 ✓ Maintaining Your System: Automation

Task 7 ✓ Maintaining Your System: Package Management

Task 8 ✓ Maintaining Your System: Logs

Task 9 ✓ Conclusions & Summaries