

## Linux Fundamentals Part 2

Task 1 Intro

This room is the second part in the Linux Fundamental rooms designed to teach you about various Linux concepts, and in-built tools. This room covers the following topics:

- Linux Operators
- Advanced File Operators

Deploy the machine and SSH into the room as explained in the tasks below using the following credentials (these credentials will be available from the last task of the [Linux Fundamentals Part 1](#) room):

- username: shiba2
- password: pinguftw

Please note, unlike the first Linux room you will need to SSH into the machine (there isn't a browser-based machine). Learn how to do this with task 2 and 3.

**Answer the questions below**

Read the above.

No answer needed

Correct Answer

Deploy the machine attached to this task!

**NOTE:** If you have a machine open in the Welcome room (or any other room) please go to that room and terminate it before deploying the machine attached to this task. These machines are not the same, and only the one attached to this room will work.

Applications Places Sun 17 Sep, 06:18 AttackBox IP: 10.10.98.214

root@ip-10-10-98-214: ~

File Edit View Search Terminal Help

root@ip-10-10-98-214: ~#

THM AttackBox

57m 10s

Task 2 SSH - Intro

SSH is the act of remotely accessing a machine. SSH allows you to run commands interactively on the remote machine. This is done through the use of a program on the target machine, which allows the ssh client to interface with the target host.

While the most common usage of a regular operating system is graphical(allowing you to see pictures, web browsers, file managers etc.) SSH works through a command line, meaning anything done on the target machine will be done through a command prompt similar to this.

```
ssh root@parabox: ~
root@parabox: ~#
```

It may look intimidating at first, but you'll soon find out you can do much of the same functionality that you're able to do using graphical user interfaces!

It is an invaluable tool, and how you will be accessing this machine to learn and to do the challenges. Depending on the operating system there are different ways of SSHing into a machine. This section will purely focus on the windows way(PuTTY), and after we learn more about linux commands, and how they work, we'll return back to this section and learn about the linux method.

**NOTE:** Please do not try to SSH into the VM from the Welcome room. You can only access the content in this room from the VM provided in Task One. If you forgot to terminate any other machines, please do so, then press the green button to deploy the Learn Linux VM provided in

Applications Places Sun 17 Sep, 06:19 AttackBox IP: 10.10.98.214

root@ip-10-10-98-214: ~

File Edit View Search Terminal Help

root@ip-10-10-98-214: ~#

THM AttackBox

56m 11s

Task 3 Putty and SSH

Disclaimer: please do not use putty if you are already on Linux. Use the instructions for the ssh binary down below.

The download for putty can be found [here](#), once you download it go through the install process. Once you've installed it, open it and you should see this screen

PutTY Configuration

Category:

Session

Logging

Terminal

Keyboard

Bell

Features

Window

Appearance

Behaviour

Translation

Selection

Colours

Connection

Data

Proxy

Telnet

Rlogin

SSH

Serial

Basic options for your PuTTY session

Specify the destination you want to connect to

Host Name (or IP address)

Port

22

Connection type:

☐ Rlogin

☐ Telnet

☐ Rlogin

☒ SSH

☐ Serial

Load, save or delete a stored session

Saved Sessions

Default Settings

Load

Save

Delete

Close window on exit:

☐ Always

☐ Never

☒ Only on clean exit

About

Help

Open

Cancel

Applications Places Sun 17 Sep, 06:19 AttackBox IP:10.10.98.214

root@ip-10-10-98-214: ~

File Edit View Search Terminal Help

root@ip-10-10-98-214: ~#

THM AttackBox

55m 32s

Task 4 [Section 4: Linux Operators]: "&&"

&& means as you might expect "and". Meaning && allows you to execute a second command after the first one has executed *successfully*. Meaning `ls && echo hello` will work fine, but `dijahfrsdkjlfhsdhjkjlfdsdhkljfh && echo hello` will fail.

```
shiba2@nootnoot:/tmp/a$ ls && echo hello
file
hello
shiba2@nootnoot:/tmp/a$
shiba2@nootnoot:/tmp/a$ dlofhsdkjfsdhkjlfsdhkljlf && echo hello
dlofhsdkjfsdhkjlfsdhkljlf: command not found
shiba2@nootnoot:/tmp/a$ _
```

Note: Since the second command happens after the first command, you can use something created in the first command in the second command.

```
shiba2@nootnoot:/tmp/a$ echo hello >> file2 && cat file2
hello
shiba2@nootnoot:/tmp/a$ _
```

Answer the questions below

Read the above

No answer needed

Correct Answer

Task 5 [Section 4: Linux Operators]: "&"

Applications Places Sun 17 Sep, 06:20 AttackBox IP:10.10.98.214

root@ip-10-10-98-214: ~

File Edit View Search Terminal Help

root@ip-10-10-98-214: ~#

THM AttackBox

54m 57s

Task 5 [Section 4: Linux Operators]: "&"

Much unlike &&, & has nothing to do with and at all(try saying that 10 times fast). & is a background operator, meaning say you run a command that takes 10 seconds to run, normally you wouldn't be able to run commands during that period; however, with & that command will still execute and you'll be able to run other commands.

```
shiba2@nootnoot:~/tmp/a$ sleep 5
shiba2@nootnoot:~/tmp/a$ sleep 5 &
[2] 8024
shiba2@nootnoot:~/tmp/a$ ls
file file2
shiba2@nootnoot:~/tmp/a$ _
```

Note: I can't exactly show time in an image, but trust me I really did wait the 5 seconds :)

Answer the questions below

Read the above

No answer needed

Correct Answer

Task 6 [Section 4: Linux Operators]: "\$"

Task 7 [Section 4: Linux Operators]: "|"

Task 8 [Section 4: Linux Operators]: "-;"

Applications Places Sun 17 Sep, 06:25AttackBox IP:10.10.98.214

shiba2@nootnoot: -

File Edit View Search Terminal Help

0 updates are security updates.

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.

shiba2@nootnoot:~\$ sudo

usage: sudo -h | -K | -k | -V

usage: sudo -v [-AknS] [-g group] [-h host] [-p prompt] [-u user]

usage: sudo -l [-AknS] [-g group] [-h host] [-p prompt] [-U user] [-u user] [command]

usage: sudo [-AbEHknPS] [-r role] [-t type] [-C num] [-g group] [-h host] [-p prompt] [-T timeout] [-u user] [VAR=value] [-i|-s] [<command>]

usage: sudo -e [-AknS] [-r role] [-t type] [-C num] [-g group] [-h host] [-p prompt] [-T timeout] [-u user] file ...

shiba2@nootnoot:~\$ sleep 10

sleep 10 shiba2@nootnoot:~\$ sleep 10 &

[1] 1614

shiba2@nootnoot:~\$

THM AttackBox 50m 04s

shiba2@nootnoot:~/tmp/a\$ ls

shiba2@nootnoot:~/tmp/a\$ touch \$USER

shiba2@nootnoot:~/tmp/a\$ ls

shiba2

shiba2@nootnoot:~/tmp/a\$ echo hi >> \$USER

shiba2@nootnoot:~/tmp/a\$ ls

shiba2

shiba2@nootnoot:~/tmp/a\$ cat shiba2

hi

shiba2@nootnoot:~/tmp/a\$

Recall that the >> operator appends output to a file.

Environment variables can also be set pretty easily, just running `export <varname>=<value>` will set that as an environment variable

```
root@parabox:~/tmp/a# export nootnoot=test
root@parabox:~/tmp/a# echo $nootnoot
test
root@parabox:~/tmp/a#
```

Answer the questions below

How would you set nootnoot equal to 1111

export nootnoot=1111

Correct Answer

What is the value of the home environment variable

/home/shiba2

Correct Answer

Applications Places Sun 17 Sep, 06:28AttackBox IP:10.10.98.214

shiba2@nootnoot: -

File Edit View Search Terminal Help

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.

shiba2@nootnoot:~\$ sudo

usage: sudo -h | -K | -k | -V

usage: sudo -v [-AknS] [-g group] [-h host] [-p prompt] [-u user]

usage: sudo -l [-AknS] [-g group] [-h host] [-p prompt] [-U user] [-u user] [command]

usage: sudo [-AbEHknPS] [-r role] [-t type] [-C num] [-g group] [-h host] [-p prompt] [-T timeout] [-u user] [VAR=value] [-i|-s] [<command>]

usage: sudo -e [-AknS] [-r role] [-t type] [-C num] [-g group] [-h host] [-p prompt] [-T timeout] [-u user] file ...

shiba2@nootnoot:~\$ sleep 10

sleep 10 shiba2@nootnoot:~\$ sleep 10 &

[1] 1614

shiba2@nootnoot:~\$ echo \$USER

shiba2

[1]+ Done sleep 10

shiba2@nootnoot:~\$

THM AttackBox 47m 11s

Task 7 [Section 4: Linux Operators]: "|"

Continuing with the trend of very special operators, we have the pipe. The pipe is unique because while operators like `>>` allow you to store the output of a command, the `|` operator allows you to take the output of a command and use it as input for a second command.

For example, I can use `cat` to get the output of a file, and pipe that into `grep` to search for a specific string (Note: We will learn more about `grep` later, but for now just know that it's a command used to find specific strings in an input).

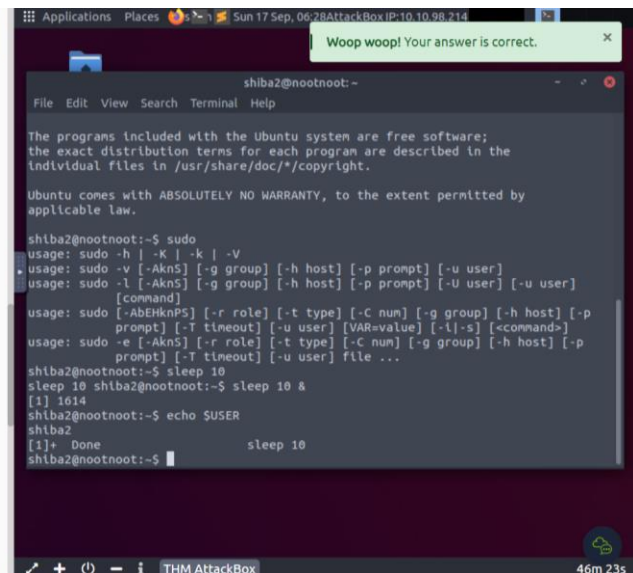
```
shiba2@nootnoot:~/tmp/11$ cat test
12312314
test
noot
shiba2@nootnoot:~/tmp/11$ cat test | grep noot
noot
shiba2@nootnoot:~/tmp/11$ cat test | grep 123
12312314
shiba2@nootnoot:~/tmp/11$ _
```

It is worth noting that not all commands support the pipe, and some that do support it require you to use `-` instead of input, for example `cat -`. So always check to see if the command does support it

Answer the questions below

Read the above!

No answer neededCorrect Answer



Task 8 [Section 4: Linux Operators]: "&&"

The `&&` operator works a lot like `&`, however it does not require the first command to execute successfully. This means that you can do `dkhsgffgsafgfasdfghkgdsgfs; ls` and you would still see the output of `ls`.

```
root@parabox:~/tmp/a# ls
file
root@parabox:~/tmp/a#
root@parabox:~/tmp/a# dkhsgffgsafgfasdfghkgdsgfs; ls
dkhsgffgsafgfasdfghkgdsgfs: command not found
file
root@parabox:~/tmp/a#
```

Answer the questions below

Read the above.

No answer neededCorrect Answer

Task 9 [Section 4: Linux Operators]: ">"

Task 10 [Section 4: Linux Operators]: ">>"

The screenshot shows a terminal window titled "shiba2@nootnoot: ~". A green notification bubble at the top right says "Woop woop! Your answer is correct." The terminal output shows the user running `cat test` and `cat test | grep noot`, which returns `noot`. The user then runs `cat test | grep 123`, which returns `12312314`. The terminal also shows the user running `sleep 10` and `echo $USER`, which returns `shiba2`. The terminal window is part of a THM AttackBox environment.

Task 10 [Section 4: Linux Operators]: ">>"

>> does mainly the same thing as >, with one key difference. >> appends the output of a command to a file, instead of erasing it.

```
shiba2@nootnoot:/tmp/a$ echo hello >> file
shiba2@nootnoot:/tmp/a$ cat file
hello
shiba2@nootnoot:/tmp/a$ echo noot >> file
shiba2@nootnoot:/tmp/a$ cat file
hello
noot
shiba2@nootnoot:/tmp/a$
```

Answer the questions below

Read the above

No answer needed

Correct Answer

Task 11 Binary - shiba2

Task 12 [Section 5 - Advanced File Operations]: Intro

Task 13 [Section 5 - Advanced File Operators]: A bit of background.

Task 14 [Section 5: Advanced File Operations]: chown

Applications Places Sun 17 Sep, 06:30AttackBox IP:10.10.98.214

shiba2@nootnoot: -

File Edit View Search Terminal Help

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.

shiba2@nootnoot:~\$ sudo

usage: sudo -h | -K | -k | -V

usage: sudo -v [-AknS] [-g group] [-h host] [-p prompt] [-u user]

usage: sudo -l [-AknS] [-g group] [-h host] [-p prompt] [-u user] [command]

usage: sudo [-AbEHknPS] [-r role] [-t type] [-C num] [-g group] [-h host] [-p prompt] [-T timeout] [-u user] [VAR=value] [-i|-s] [<command>]

usage: sudo -e [-AknS] [-r role] [-t type] [-C num] [-g group] [-h host] [-p prompt] [-T timeout] [-u user] file ...

shiba2@nootnoot:~\$ sleep 10

sleep 10 shiba2@nootnoot:~\$ sleep 10 &

[1] 1614

shiba2@nootnoot:~\$ echo \$USER

shiba2

[1]+ Done sleep 10

shiba2@nootnoot:~\$

THM AttackBox 44m 29s

tryhackme.com/room/linux2

1916x880 Pause (00:04:34)

Task 10 [Section 4: Linux Operators]: ">>"

Task 11 Binary - shiba2

This challenge is pretty simple. The binary is checking to see if the environment variable "test1234" exists, and if it's set equal to the current \$USER environment variable.

Answer the questions below

What is shiba3's password?

happynootnoises

Correct Answer

Hint

Task 12 [Section 5 - Advanced File Operations]: Intro

Task 13 [Section 5 - Advanced File Operators]: A bit of background.

Task 14 [Section 5: Advanced File Operations]: chown

Task 15 [Section 5: Advanced File Operations]: chmod

Task 16 [Section 5: Advanced File Operations]: rm

Applications Places Sun 17 Sep, 06:32AttackBox IP:10.10.98.214

Woop woop! Your answer is correct.

shiba2@nootnoot: -

File Edit View Search Terminal Help

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

shiba2@nootnoot:~\$ sudo

usage: sudo -h | -K | -k | -V

usage: sudo -v [-AknS] [-g group] [-h host] [-p prompt] [-u user]

usage: sudo -l [-AknS] [-g group] [-h host] [-p prompt] [-u user] [command]

usage: sudo [-AbEHknPS] [-r role] [-t type] [-C num] [-g group] [-h host] [-p prompt] [-T timeout] [-u user] [VAR=value] [-i|-s] [<command>]

usage: sudo -e [-AknS] [-r role] [-t type] [-C num] [-g group] [-h host] [-p prompt] [-T timeout] [-u user] file ...

shiba2@nootnoot:~\$ sleep 10

sleep 10 shiba2@nootnoot:~\$ sleep 10 &

[1] 1614

shiba2@nootnoot:~\$ echo \$USER

shiba2

[1]+ Done sleep 10

shiba2@nootnoot:~\$ export test1234=\$USER

shiba2@nootnoot:~\$ ls

shiba2

shiba2@nootnoot:~\$ ./shiba2

happynootnoises

shiba2@nootnoot:~\$

THM AttackBox 43m 11s

86°F Partly sunny

Search

11:01 AM 9/17/2023



Task 10 [Section 4: Linux Operators]: ">>"

Task 11 Binary - shiba2

Task 12 [Section 5 - Advanced File Operations]: Intro

Much like windows, files have a lot of complexity to them. Multiple different parameters have to be modified to allow certain users to read to files, write to files, and execute certain files. This section will cover modifying these parameters.

Answer the questions below

Read the above.

No answer needed Correct Answer

Task 13 [Section 5 - Advanced File Operators]: A bit of background.

Task 14 [Section 5: Advanced File Operations]: chown

Task 15 [Section 5: Advanced File Operations]: chmod

Woop woopl! Your answer is correct.

```
shiba2@nootnoot:~$ sudo
usage: sudo -h | -K | -k | -V
usage: sudo -v [-AknS] [-g group] [-h host] [-p prompt] [-u user]
usage: sudo -l [-AknS] [-g group] [-h host] [-p prompt] [-U user] [-u user]
[command]
usage: sudo [-AbEHknPS] [-r role] [-t type] [-C num] [-g group] [-h host] [-p
prompt] [-T timeout] [-u user] [VAR=value] [-i|-s] [<command>]
usage: sudo -e [-AknS] [-r role] [-t type] [-C num] [-g group] [-h host] [-p
prompt] [-T timeout] [-u user] file ...

shiba2@nootnoot:~$ sleep 10
sleep 10 shiba2@nootnoot:~$ sleep 10 &
[1] 1614
shiba2@nootnoot:~$ echo $USER
shiba2
[1]+  Done                  sleep 10
shiba2@nootnoot:~$ export test1234=$USER
shiba2@nootnoot:~$ ls
shiba2
shiba2@nootnoot:~$ ./shiba2
happynootnoises
shiba2@nootnoot:~$
```

Task 13 [Section 5 - Advanced File Operators]: A bit of background.

Recall that ls has different flags that allow you to view information about different types of files.

```
root@parabox:/tmp/all $ ls -al
total 24
drwxr-xr-x  2 root root 4096 Feb 19 19:06 .
drwxr-xr-x 13 root root 4096 Feb 19 19:09 ..
-rw-r--r--  1 root root 207 Feb 19 19:05 a.c
-rwxr-xr-x  1 root root 8384 Feb 19 19:05 a.out
```

This image has all of the attributes that will be covered in this section. More specifically we're interested in these three.

```
root@parabox:/tmp/all $ ls -al
total 24
drwxr-xr-x  2 root root 4096 Feb 19 19:06 .
drwxr-xr-x 13 root root 4096 Feb 19 19:09 ..
-rw-r--r--  1 root root 207 Feb 19 19:05 a.c
-rwxr-xr-x  1 root root 8384 Feb 19 19:05 a.out
```

These attributes are (in order) the file permissions, owner of the file, and group that the file is in.

The next few tasks will go over the command to modify these attributes.

Answer the questions below

Read the above!

No answer needed Correct Answer

```
shiba2@nootnoot:~$ sudo
usage: sudo -h | -K | -k | -V
usage: sudo -v [-AknS] [-g group] [-h host] [-p prompt] [-u user]
usage: sudo -l [-AknS] [-g group] [-h host] [-p prompt] [-U user] [-u user]
[command]
usage: sudo [-AbEHknPS] [-r role] [-t type] [-C num] [-g group] [-h host] [-p
prompt] [-T timeout] [-u user] [VAR=value] [-i|-s] [<command>]
usage: sudo -e [-AknS] [-r role] [-t type] [-C num] [-g group] [-h host] [-p
prompt] [-T timeout] [-u user] file ...

shiba2@nootnoot:~$ sleep 10
sleep 10 shiba2@nootnoot:~$ sleep 10 &
[1] 1614
shiba2@nootnoot:~$ echo $USER
shiba2
[1]+  Done                  sleep 10
shiba2@nootnoot:~$ export test1234=$USER
shiba2@nootnoot:~$ ls
shiba2
shiba2@nootnoot:~$ ./shiba2
happynootnoises
shiba2@nootnoot:~$
```

only wanted to change the user but keep the group.

```
root@nootnoot:/tmp/aa# ls -al
total 8
drwxrwxr-x 2 shiba3 shiba3 4096 Feb 20 03:06
drwxrwxrwt 13 root root 4096 Feb 20 03:16
-rw-rw-r-- 1 shiba3 shiba3 0 Feb 20 03:06 file
root@nootnoot:/tmp/aa# chown shiba1 file
root@nootnoot:/tmp/aa# ls -al
total 8
drwxrwxr-x 2 shiba3 shiba3 4096 Feb 20 03:06
drwxrwxrwt 13 root root 4096 Feb 20 03:16
-rw-rw-r-- 1 shiba1 shiba3 0 Feb 20 03:06 file
root@nootnoot:/tmp/aa# _
```

Answer the questions below

How would you change the owner of file to paradox

chown paradox file

Correct Answer

What about the owner and the group of file to paradox

chown paradox:paradox file

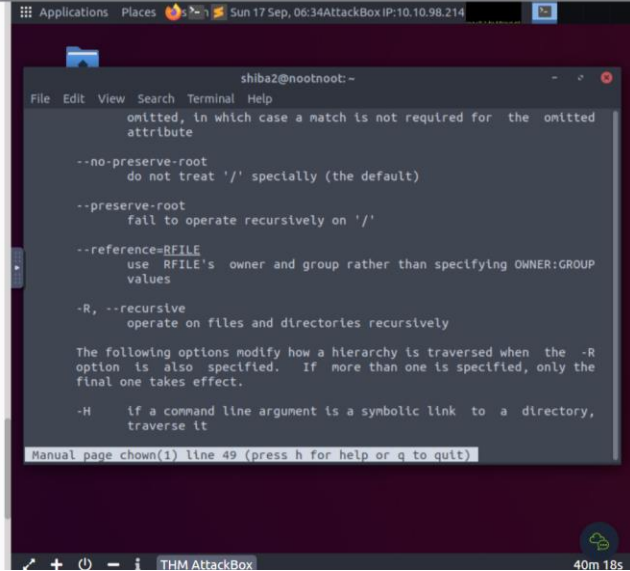
Correct Answer

What flag allows you to operate on every file in the directory at once?

-r

Correct Answer

Task 15 [Section 5: Advanced File Operations]: chmod



```
shiba2@nootnoot:~$ chown --help
omitted, in which case a match is not required for the omitted
attribute

--no-preserve-root
do not treat '/' specially (the default)

--preserve-root
fail to operate recursively on '/'

--reference=RFILE
use RFILE's owner and group rather than specifying OWNER:GROUP
values

-R, --recursive
operate on files and directories recursively

The following options modify how a hierarchy is traversed when the -R
option is also specified. If more than one is specified, only the
final one takes effect.

-H if a command line argument is a symbolic link to a directory,
traverse it

Manual page chown(1) line 49 (press h for help or q to quit)
```

```
shiba3@nootnoot:~$ echo a >> .profile
bash: .profile: Permission denied
shiba3@nootnoot:~$
```

When we try to change the perms to 644 nothing happens because the perms are already 644. The interesting part is while we can write data to .profile with echo while the perms are 644, we can't when we change the perms to 544, because we took away our own write perms. Following that logic, that means we can completely lock ourselves out of writing to a file we already own!

Note: It is possible to give someone no perms to a file, You can just put 0 as the digit. 770 Means that everyone that isn't a part of the user or group can't do anything to the file.

Answer the questions below

What permissions mean the user can read the file, the group can read and write to the file, and no one else can read, write or execute the file?

460

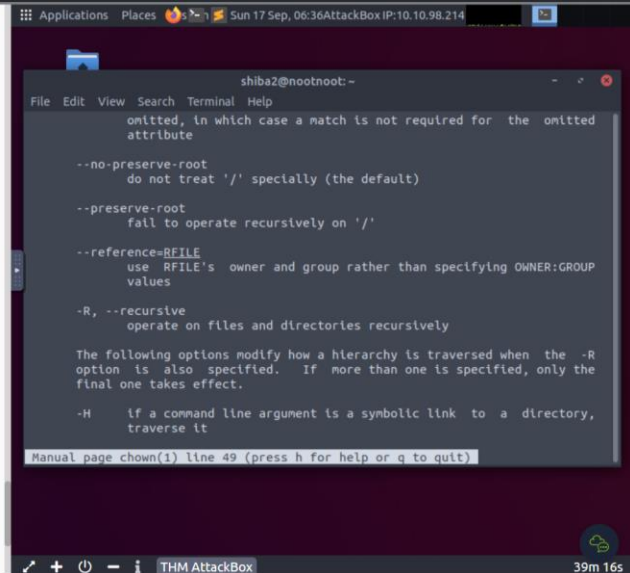
Correct Answer

What permissions mean the user can read, write, and execute the file, the group can read, write, and execute the file, and everyone else can read, write, and execute the file.

777

Correct Answer

Task 16 [Section 5: Advanced File Operations]: rm



```
shiba2@nootnoot:~$ chown --help
omitted, in which case a match is not required for the omitted
attribute

--no-preserve-root
do not treat '/' specially (the default)

--preserve-root
fail to operate recursively on '/'

--reference=RFILE
use RFILE's owner and group rather than specifying OWNER:GROUP
values

-R, --recursive
operate on files and directories recursively

The following options modify how a hierarchy is traversed when the -R
option is also specified. If more than one is specified, only the
final one takes effect.

-H if a command line argument is a symbolic link to a directory,
traverse it

Manual page chown(1) line 49 (press h for help or q to quit)
```

```
root@nootnoot:/tmp/aa$ _
```

As you can imagine this is incredibly dangerous, as you can remove some very important files, and render your system completely unusable. It is not worth noting that you need write permissions to the file to deleted so you cant just delete any file if you're a regular user.

```
shiba3@nootnoot:/tmp/aa$ rm /etc/shadow
rm: remove write-protected regular file '/etc/shadow'? y
rm: cannot remove '/etc/shadow': Permission denied
shiba3@nootnoot:/tmp/aa$ _
```

**Answer the questions below**

What flag deletes every file in a directory

Correct Answer

How do you suppress all warning prompts

Correct Answer

**Task 17** [Section 5: Advanced File Operations]: mv

```
shiba2@nootnoot: -
File Edit View Search Terminal Help
omitted, in which case a match is not required for the omitted attribute

--no-preserve-root
do not treat '/' specially (the default)

--preserve-root
fail to operate recursively on '/'

--reference=RFILE
use RFILE's owner and group rather than specifying OWNER:GROUP values

-R, --recursive
operate on files and directories recursively

The following options modify how a hierarchy is traversed when the -R option is also specified. If more than one is specified, only the final one takes effect.

-H if a command line argument is a symbolic link to a directory, traverse it

Manual page chown(1) line 49 (press h for help or q to quit)
```

THM AttackBox 38m 33s

mv allows you to move files from one place to another. The syntax for the command is `mv <file> <destination>`, so if I wanted to move a file to my home directory I could type `mv file ~`.

```
shiba3@nootnoot:/tmp/aa$ ls
file
shiba3@nootnoot:/tmp/aa$ mv file ~
shiba3@nootnoot:/tmp/aa$ mv file ~
shiba3@nootnoot:/tmp/aa$ ls
file
shiba3@nootnoot:/tmp/aa$
```

Note: You can also use mv to change the name of file, `mv file ~/ghfds` will rename file to ghfds.

```
shiba3@nootnoot:/tmp/aa$ mv file ~/
shiba3@nootnoot:/tmp/aa$ mv file ~/ghfds
shiba3@nootnoot:/tmp/aa$ mv file ~/ghfds
shiba3@nootnoot:/tmp/aa$ ls
file ghfds
shiba3@nootnoot:/tmp/aa$
```

**Answer the questions below**

How would you move file to /tmp

Correct Answer

**Task 18** Linux Fundamentals 3

```
shiba2@nootnoot: -
File Edit View Search Terminal Help
omitted, in which case a match is not required for the omitted attribute

--no-preserve-root
do not treat '/' specially (the default)

--preserve-root
fail to operate recursively on '/'

--reference=RFILE
use RFILE's owner and group rather than specifying OWNER:GROUP values

-R, --recursive
operate on files and directories recursively

The following options modify how a hierarchy is traversed when the -R option is also specified. If more than one is specified, only the final one takes effect.

-H if a command line argument is a symbolic link to a directory, traverse it

Manual page chown(1) line 49 (press h for help or q to quit)
```

THM AttackBox 38m 02s



Task 15 [Section 5: Advanced File Operations]: chmod

Task 16 [Section 5: Advanced File Operations]: rm

Task 17 [Section 5: Advanced File Operations]: mv

Task 18 Linux Fundamentals 3

Now that you have some intermediate knowledge to using Linux, join the [Linux Fundamentals 3](#) room.

*Answer the questions below*

Join the Linux Fundamentals 3 room, and finish learning Linux: <https://tryhackme.com/room/linux3>

No answer needed

Created by [tryhackme](#)


This is a **free** room, which means anyone can deploy virtual machines in the room (without being subscribed): 62743 users are in here and this room is 1292 days old.

Applications Places Sun 17 Sep, 06:17 AttackBox IP: 10.10.98.214

Woop woop! Your answer is correct.

shiba2@nootnoot: ~  
File Edit View Search Terminal Help  
omitted, in which case a match is not required for the omitted attribute  
-o preserve root  
 '/' specially (the default)  
ate recursively on '/'  
owner and group rather than specifying OWNER:GROUP  
iles and directories recursively  
ns modify how a hierarchy is traversed when the -R specified. If more than one is specified, only the  
ect.  
line argument is a symbolic link to a directory, traverse it  
Manual page chown(1) line 49 (press h for help or q to quit)

THM AttackBox 37m 43s



# Learning Linux

## Part 2

Task 1 [Section 1: Linux Fundamentals]: Part 2

Task 2 [Section 1: Linux Fundamentals]: Part 2

Task 3 [Section 1: Linux Fundamentals]: Part 2

Task 4 [Section 1: Linux Fundamentals]: Part 2

Task 5 [Section 1: Linux Fundamentals]: Part 2

Task 6 [Section 1: Linux Fundamentals]: Part 2

Task 7 [Section 1: Linux Fundamentals]: Part 2

Task 8 [Section 1: Linux Fundamentals]: Part 2

Task 9 [Section 1: Linux Fundamentals]: Part 2

Task 10 [Section 1: Linux Fundamentals]: Part 2

Task 11 [Section 1: Linux Fundamentals]: Part 2

Task 12 [Section 1: Linux Fundamentals]: Part 2

Task 13 [Section 1: Linux Fundamentals]: Part 2

Task 14 [Section 1: Linux Fundamentals]: Part 2

Task 15 [Section 1: Linux Fundamentals]: Part 2

Task 16 [Section 1: Linux Fundamentals]: Part 2

Task 17 [Section 1: Linux Fundamentals]: Part 2

Task 18 [Section 1: Linux Fundamentals]: Part 2

Task 19 [Section 1: Linux Fundamentals]: Part 2

Task 20 [Section 1: Linux Fundamentals]: Part 2

Task 21 [Section 1: Linux Fundamentals]: Part 2

Task 22 [Section 1: Linux Fundamentals]: Part 2

Task 23 [Section 1: Linux Fundamentals]: Part 2

Task 24 [Section 1: Linux Fundamentals]: Part 2

Task 25 [Section 1: Linux Fundamentals]: Part 2

Task 26 [Section 1: Linux Fundamentals]: Part 2

Task 27 [Section 1: Linux Fundamentals]: Part 2

Task 28 [Section 1: Linux Fundamentals]: Part 2

Task 29 [Section 1: Linux Fundamentals]: Part 2

Task 30 [Section 1: Linux Fundamentals]: Part 2

Task 31 [Section 1: Linux Fundamentals]: Part 2

Task 32 [Section 1: Linux Fundamentals]: Part 2

Task 33 [Section 1: Linux Fundamentals]: Part 2

Task 34 [Section 1: Linux Fundamentals]: Part 2

Task 35 [Section 1: Linux Fundamentals]: Part 2

Task 36 [Section 1: Linux Fundamentals]: Part 2

Task 37 [Section 1: Linux Fundamentals]: Part 2

Task 38 [Section 1: Linux Fundamentals]: Part 2

Task 39 [Section 1: Linux Fundamentals]: Part 2

Task 40 [Section 1: Linux Fundamentals]: Part 2

Task 41 [Section 1: Linux Fundamentals]: Part 2

Task 42 [Section 1: Linux Fundamentals]: Part 2

Task 43 [Section 1: Linux Fundamentals]: Part 2

Task 44 [Section 1: Linux Fundamentals]: Part 2

Task 45 [Section 1: Linux Fundamentals]: Part 2

Task 46 [Section 1: Linux Fundamentals]: Part 2

Task 47 [Section 1: Linux Fundamentals]: Part 2

Task 48 [Section 1: Linux Fundamentals]: Part 2

Task 49 [Section 1: Linux Fundamentals]: Part 2

Task 50 [Section 1: Linux Fundamentals]: Part 2

Task 51 [Section 1: Linux Fundamentals]: Part 2

Task 52 [Section 1: Linux Fundamentals]: Part 2

Task 53 [Section 1: Linux Fundamentals]: Part 2

Task 54 [Section 1: Linux Fundamentals]: Part 2

Task 55 [Section 1: Linux Fundamentals]: Part 2

Task 56 [Section 1: Linux Fundamentals]: Part 2

Task 57 [Section 1: Linux Fundamentals]: Part 2

Task 58 [Section 1: Linux Fundamentals]: Part 2

Task 59 [Section 1: Linux Fundamentals]: Part 2

Task 60 [Section 1: Linux Fundamentals]: Part 2

Task 61 [Section 1: Linux Fundamentals]: Part 2

Task 62 [Section 1: Linux Fundamentals]: Part 2

Task 63 [Section 1: Linux Fundamentals]: Part 2

Task 64 [Section 1: Linux Fundamentals]: Part 2

Task 65 [Section 1: Linux Fundamentals]: Part 2

Task 66 [Section 1: Linux Fundamentals]: Part 2

Task 67 [Section 1: Linux Fundamentals]: Part 2

Task 68 [Section 1: Linux Fundamentals]: Part 2

Task 69 [Section 1: Linux Fundamentals]: Part 2

Task 70 [Section 1: Linux Fundamentals]: Part 2

Task 71 [Section 1: Linux Fundamentals]: Part 2

Task 72 [Section 1: Linux Fundamentals]: Part 2

Task 73 [Section 1: Linux Fundamentals]: Part 2

Task 74 [Section 1: Linux Fundamentals]: Part 2

Task 75 [Section 1: Linux Fundamentals]: Part 2

Task 76 [Section 1: Linux Fundamentals]: Part 2

Task 77 [Section 1: Linux Fundamentals]: Part 2

Task 78 [Section 1: Linux Fundamentals]: Part 2

Task 79 [Section 1: Linux Fundamentals]: Part 2

Task 80 [Section 1: Linux Fundamentals]: Part 2

Task 81 [Section 1: Linux Fundamentals]: Part 2

Task 82 [Section 1: Linux Fundamentals]: Part 2

Task 83 [Section 1: Linux Fundamentals]: Part 2

Task 84 [Section 1: Linux Fundamentals]: Part 2

Task 85 [Section 1: Linux Fundamentals]: Part 2

Task 86 [Section 1: Linux Fundamentals]: Part 2

Task 87 [Section 1: Linux Fundamentals]: Part 2

Task 88 [Section 1: Linux Fundamentals]: Part 2

Task 89 [Section 1: Linux Fundamentals]: Part 2

Task 90 [Section 1: Linux Fundamentals]: Part 2

Task 91 [Section 1: Linux Fundamentals]: Part 2

Task 92 [Section 1: Linux Fundamentals]: Part 2

Task 93 [Section 1: Linux Fundamentals]: Part 2

Task 94 [Section 1: Linux Fundamentals]: Part 2

Task 95 [Section 1: Linux Fundamentals]: Part 2

Task 96 [Section 1: Linux Fundamentals]: Part 2

Task 97 [Section 1: Linux Fundamentals]: Part 2

Task 98 [Section 1: Linux Fundamentals]: Part 2

Task 99 [Section 1: Linux Fundamentals]: Part 2

Task 100 [Section 1: Linux Fundamentals]: Part 2

Created by [tryhackme](#)

This is a **free** room, which means anyone can deploy virtual machines in the room (without being subscribed): 62743 users are in here and this room is 1292 days old.