

Lab - Getting Familiar with the Linux Shell

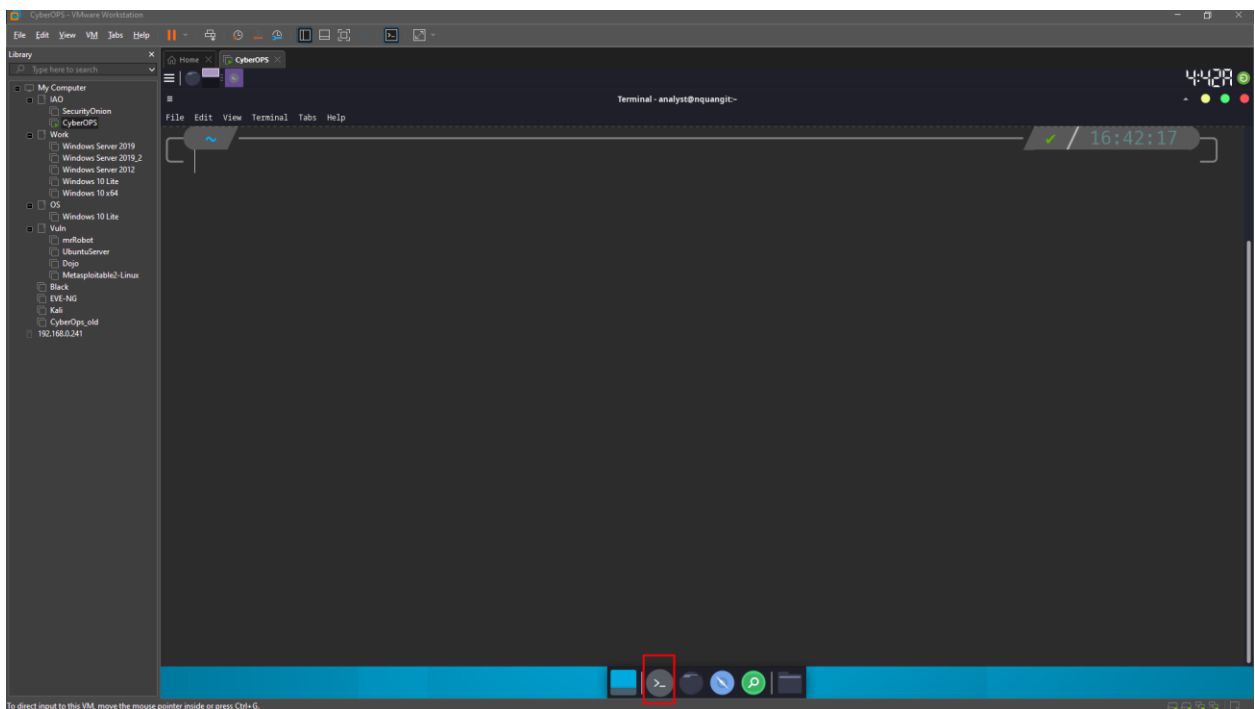
Instructions

Shell Basics

The shell is the term used to refer to the command interpreter in Linux. Also known as Terminal, Command Line and Command Prompt, the shell is very powerful way to interact with a Linux computer.

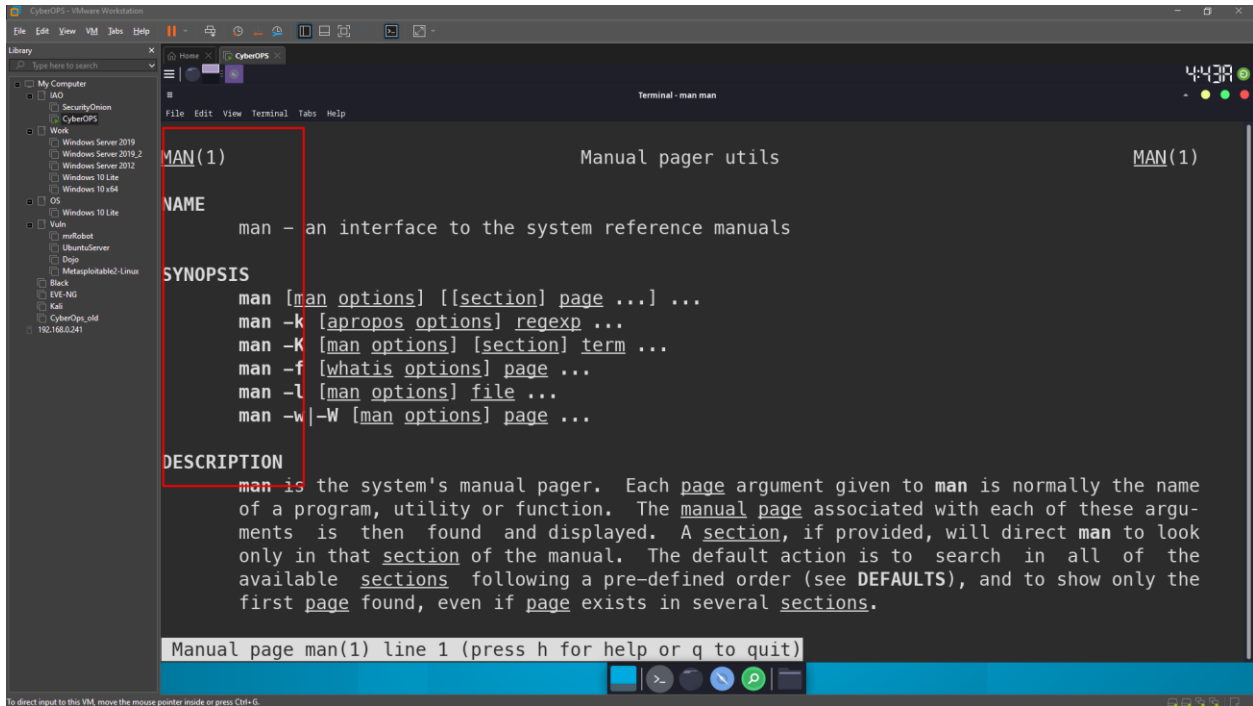
Access the Command Line

- Log on to the CyberOps Workstation VM as the **analyst** using the password **cyberops**. The account **analyst** is used as the example user account throughout this lab.
- To access the command line, click the **terminal** icon located in the Dock, at the bottom of VM screen. The terminal emulator opens.



Display Manual Pages from the command line.

- To learn more about the man page, type:
Name a few sections that are included in a man page.
NAME, SYNOPSIS, DESCRIPTION, ...



```

MAN(1)                                     Manual pager utils                                     MAN(1)

NAME
    man - an interface to the system reference manuals

SYNOPSIS
    man [man options] [[section] page ...] ...
    man -k [apropos options] regexp ...
    man -K [man options] [section] term ...
    man -f [whatis options] page ...
    man -l [man options] file ...
    man -w|-W [man options] page ...

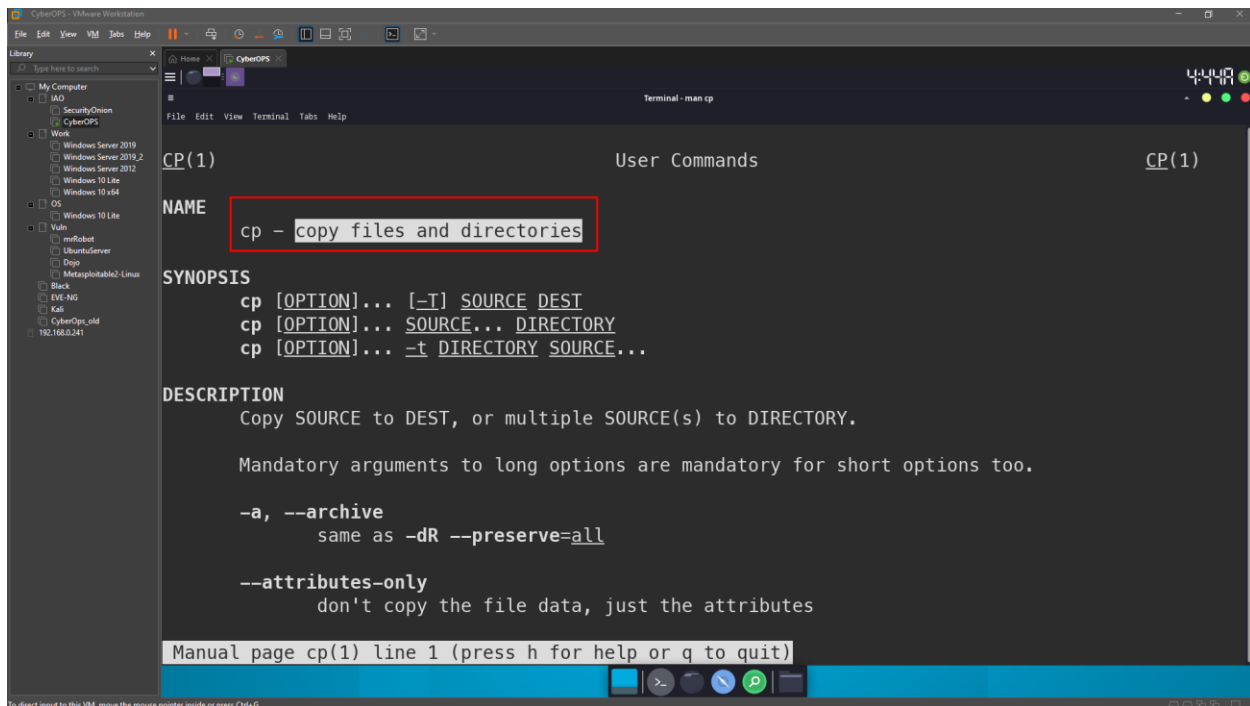
DESCRIPTION
    man is the system's manual pager. Each page argument given to man is normally the name
    of a program, utility or function. The manual page associated with each of these argu-
    ments is then found and displayed. A section, if provided, will direct man to look
    only in that section of the manual. The default action is to search in all of the
    available sections following a pre-defined order (see DEFAULTS), and to show only the
    first page found, even if page exists in several sections.

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

- Type **q** to exit the man page.
- Use the **man** command to learn more about the **cp** command:

What is the function of the **cp** command?

copy files and directories.



```

CP(1)                                     User Commands                                     CP(1)

NAME
    cp - copy files and directories

SYNOPSIS
    cp [OPTION]... [-T] SOURCE DEST
    cp [OPTION]... SOURCE... DIRECTORY
    cp [OPTION]... -t DIRECTORY SOURCE...

DESCRIPTION
    Copy SOURCE to DEST, or multiple SOURCE(s) to DIRECTORY.

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

    -a, --archive
        same as -dR --preserve=all

    --attributes-only
        don't copy the file data, just the attributes

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

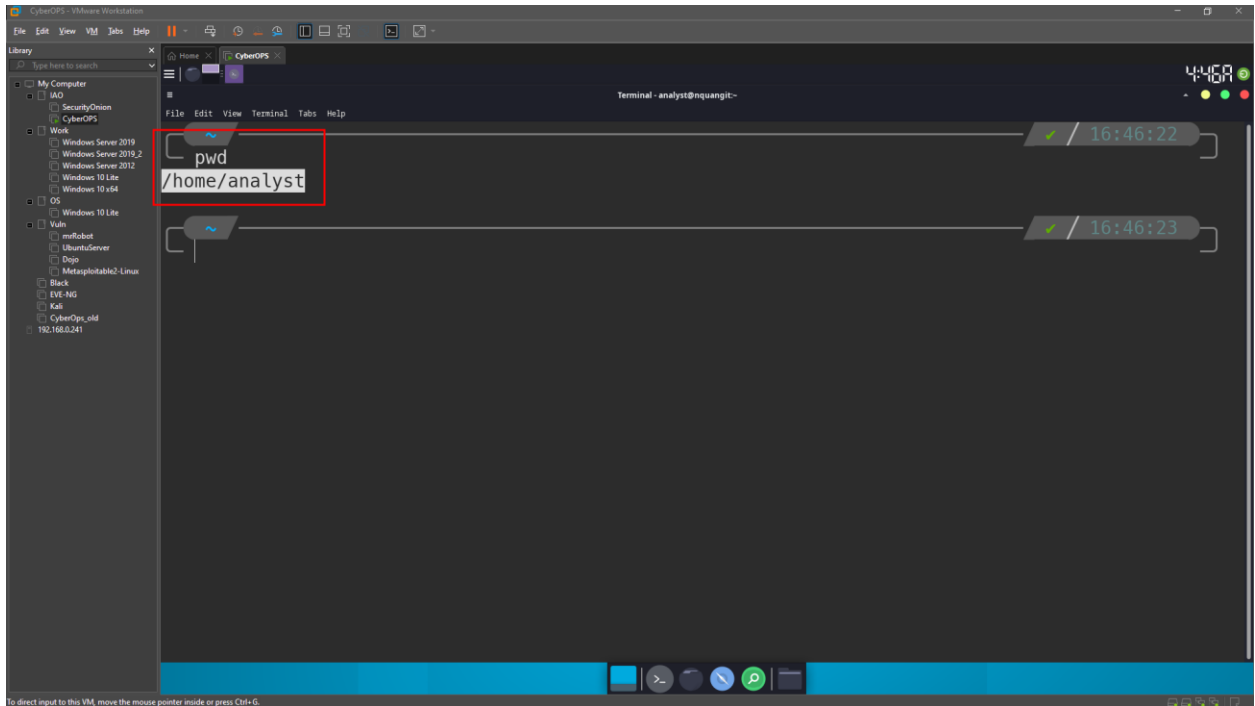
What command would you use to find out more information about the **pwd** command? What is the function of the **pwd** command?

Use “man pwd” command to find out more information.

Function of the **pwd** command: pwd - print name of current/working directory.

Create and change directories.

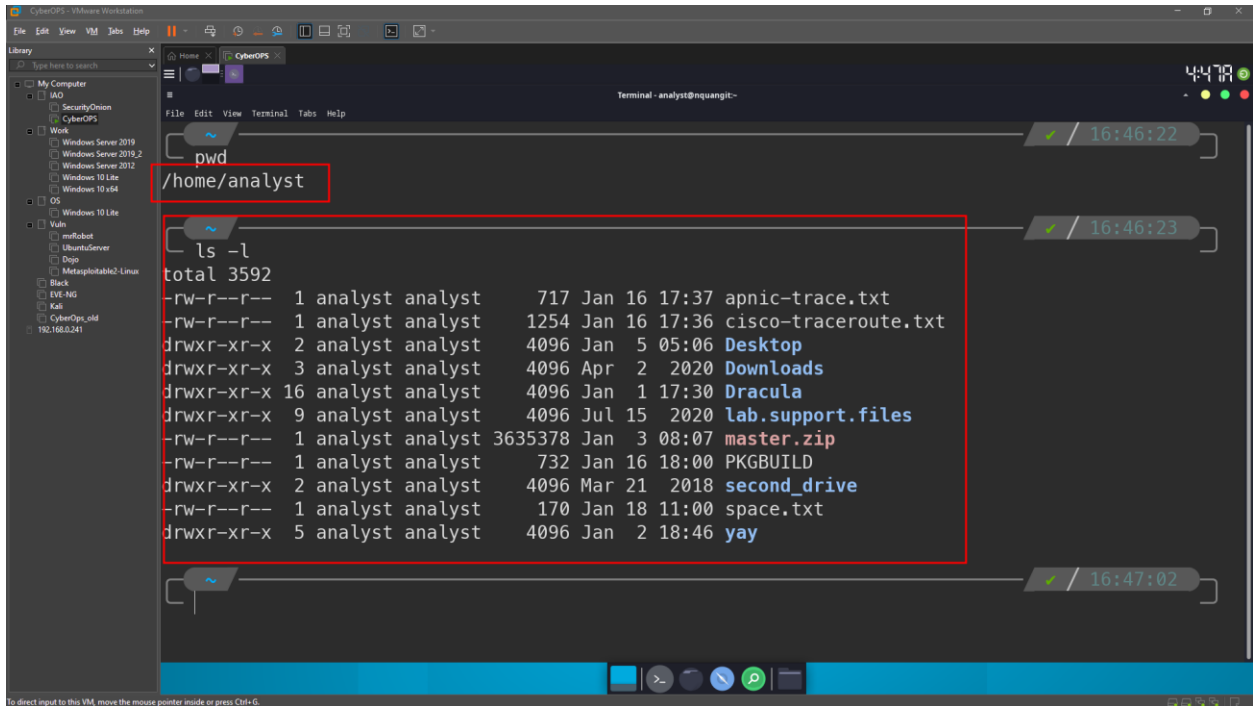
- a. Type **pwd** at the prompt.



What is the current directory?

~ or /home/analyst

- b. Navigate to the **/home/analyst** directory if it is not your current directory. Type **cd /home/analyst**
- c. Type **ls -l** at the command prompt to list the files and folders that are in the current folder. Standing for list, the **-l** option displays file size, permissions, ownership, date of creation and more.

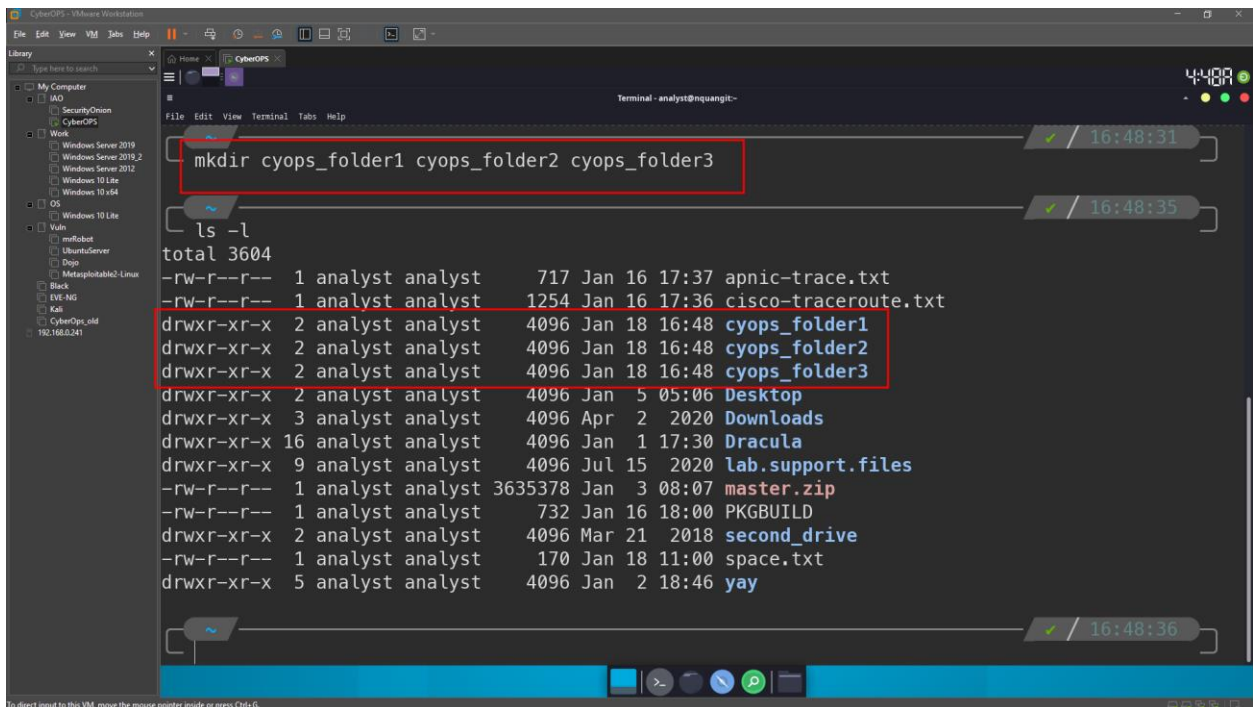


```

Terminal - analyst@quangit-
~
pwd
/home/analyst

ls -l
total 3592
-rw-r--r-- 1 analyst analyst 717 Jan 16 17:37 apnic-trace.txt
-rw-r--r-- 1 analyst analyst 1254 Jan 16 17:36 cisco-traceroute.txt
drwxr-xr-x 2 analyst analyst 4096 Jan 5 05:06 Desktop
drwxr-xr-x 3 analyst analyst 4096 Apr 2 2020 Downloads
drwxr-xr-x 16 analyst analyst 4096 Jan 1 17:30 Dracula
drwxr-xr-x 9 analyst analyst 4096 Jul 15 2020 lab.support.files
-rw-r--r-- 1 analyst analyst 3635378 Jan 3 08:07 master.zip
-rw-r--r-- 1 analyst analyst 732 Jan 16 18:00 PKGBUILD
drwxr-xr-x 2 analyst analyst 4096 Mar 21 2018 second_drive
-rw-r--r-- 1 analyst analyst 170 Jan 18 11:00 space.txt
drwxr-xr-x 5 analyst analyst 4096 Jan 2 18:46 yay
  
```

- d. In the current directory, use the **mkdir** command to create three new folders: **cyops_folder1**, **cyops_folder2**, and **cyops_folder3**. Type **mkdir cyops_folder1** and press **Enter**. Repeat these steps to create **cyops_folder2** and **cyops_folder3**.
- e. Type **ls -l** to verify that the folders have been created:

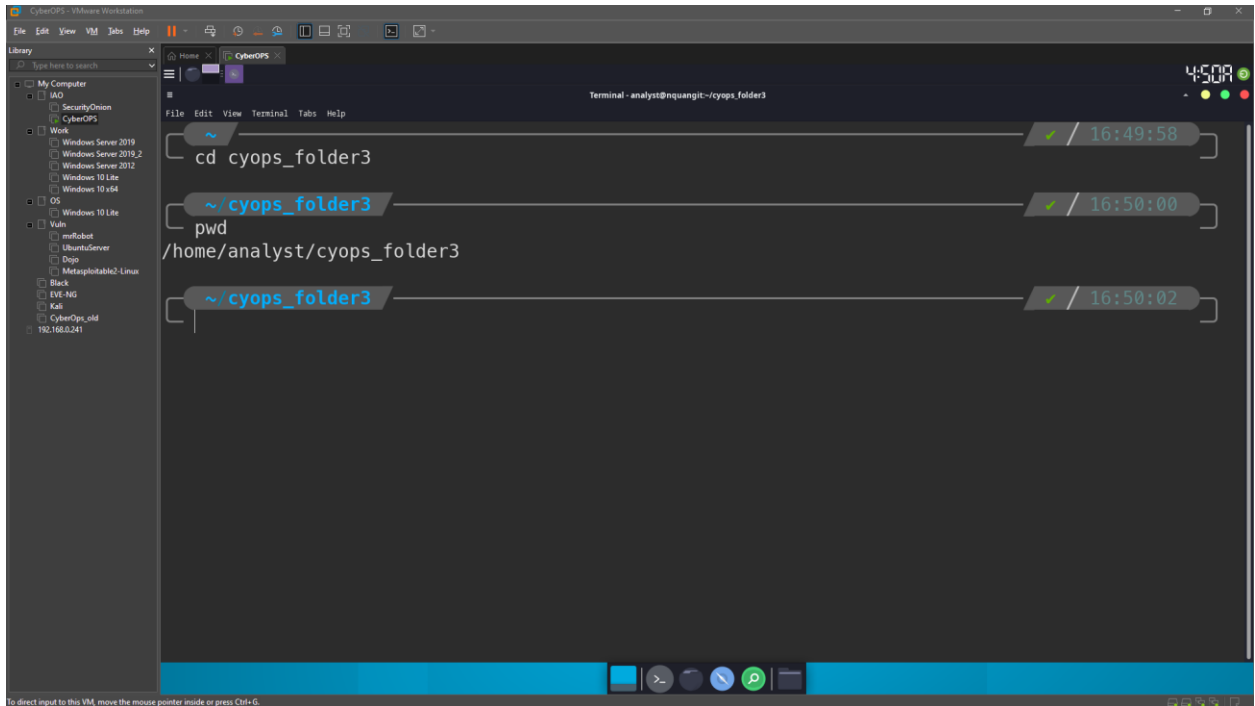


```

Terminal - analyst@quangit-
~
mkdir cyops_folder1 cyops_folder2 cyops_folder3

ls -l
total 3604
-rw-r--r-- 1 analyst analyst 717 Jan 16 17:37 apnic-trace.txt
-rw-r--r-- 1 analyst analyst 1254 Jan 16 17:36 cisco-traceroute.txt
drwxr-xr-x 2 analyst analyst 4096 Jan 18 16:48 cyops_folder1
drwxr-xr-x 2 analyst analyst 4096 Jan 18 16:48 cyops_folder2
drwxr-xr-x 2 analyst analyst 4096 Jan 18 16:48 cyops_folder3
drwxr-xr-x 2 analyst analyst 4096 Jan 5 05:06 Desktop
drwxr-xr-x 3 analyst analyst 4096 Apr 2 2020 Downloads
drwxr-xr-x 16 analyst analyst 4096 Jan 1 17:30 Dracula
drwxr-xr-x 9 analyst analyst 4096 Jul 15 2020 lab.support.files
-rw-r--r-- 1 analyst analyst 3635378 Jan 3 08:07 master.zip
-rw-r--r-- 1 analyst analyst 732 Jan 16 18:00 PKGBUILD
drwxr-xr-x 2 analyst analyst 4096 Mar 21 2018 second_drive
-rw-r--r-- 1 analyst analyst 170 Jan 18 11:00 space.txt
drwxr-xr-x 5 analyst analyst 4096 Jan 2 18:46 yay
  
```

- f. Type **cd /home/analyst/cyops_folder3** at the command prompt and press **Enter**.



Which folder are you in now?

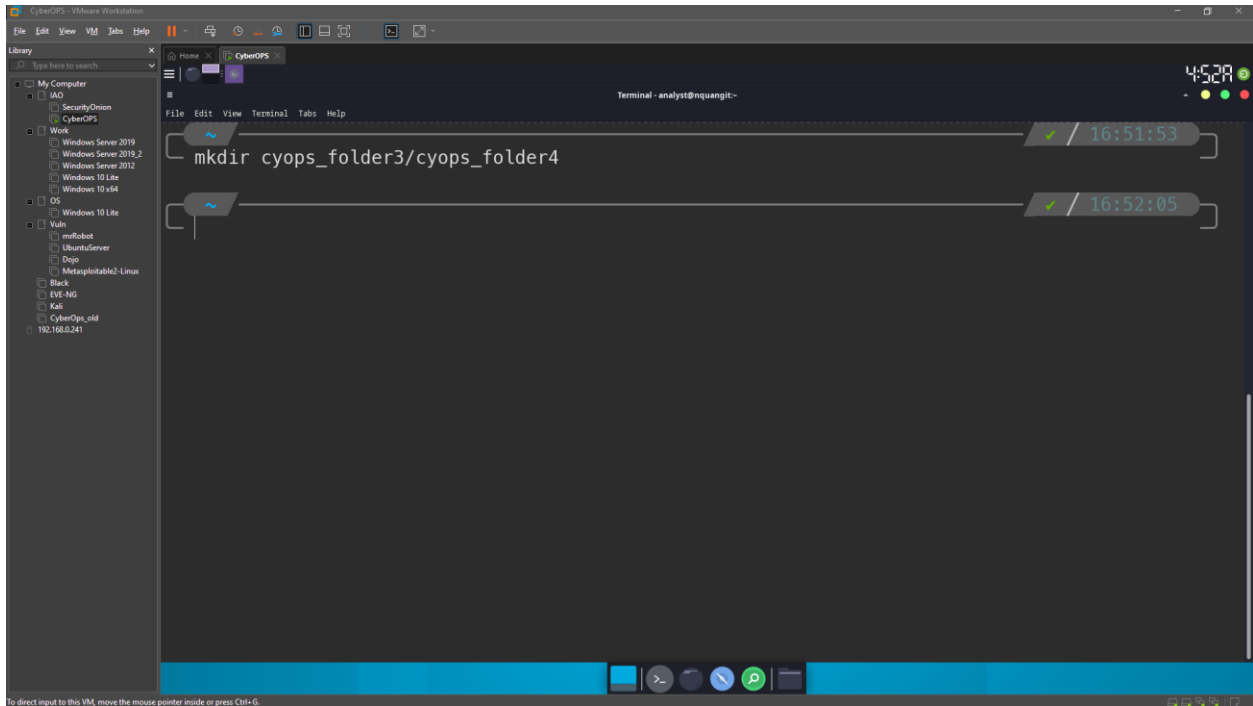
/home/analyst/cyops_folder3 or ~/cyops_folder3

Challenge: Type the command `cd ~` and describe what happens.

Why did this happen?

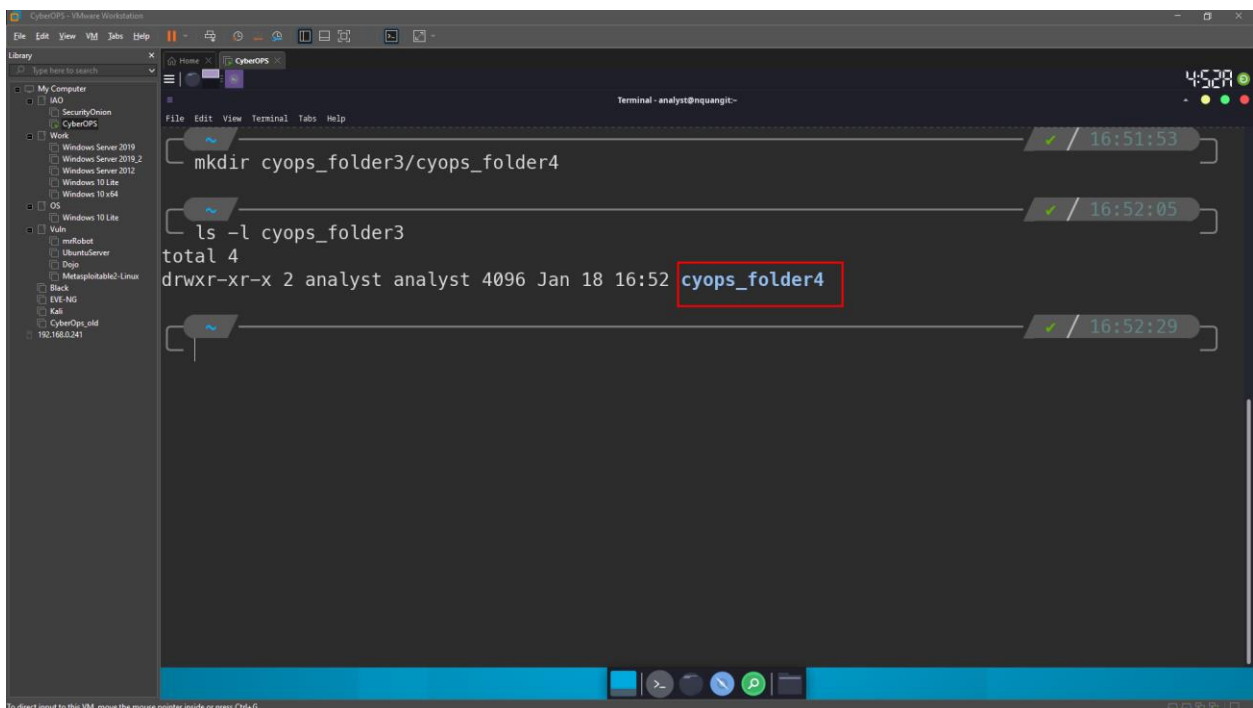
The terminal change the current working folder to the user's home directory (~ or /home/analyst)

- g. Use the **mkdir** command to create a new folder named **cyops_folder4** inside the **cyops_folder3** folder:



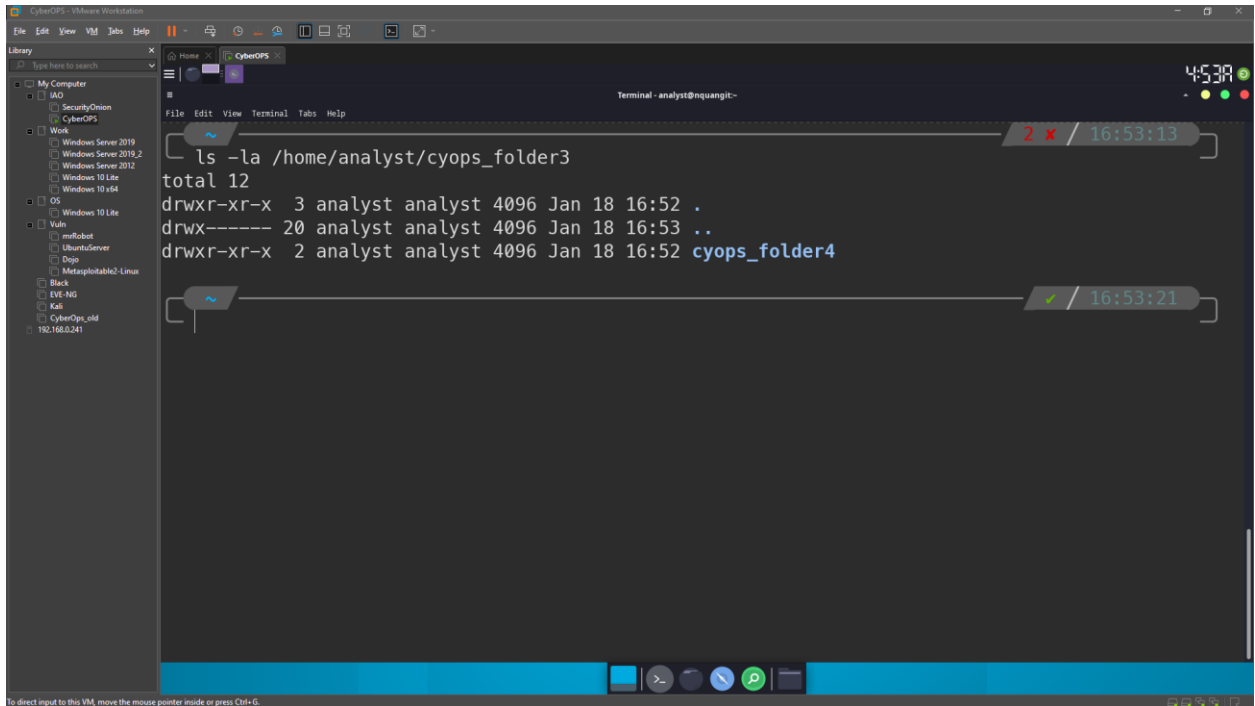
```
Terminal - analyst@nquangit-  
~  
mkdir cyops_folder3/cyops_folder4  
✓ / 16:51:53  
✓ / 16:52:05
```

h. Use the **ls -l** command to verify the folder creation.



```
Terminal - analyst@nquangit-  
~  
mkdir cyops_folder3/cyops_folder4  
✓ / 16:51:53  
✓ / 16:52:05  
ls -l cyops_folder3  
total 4  
drwxr-xr-x 2 analyst analyst 4096 Jan 18 16:52 cyops_folder4  
✓ / 16:52:29
```

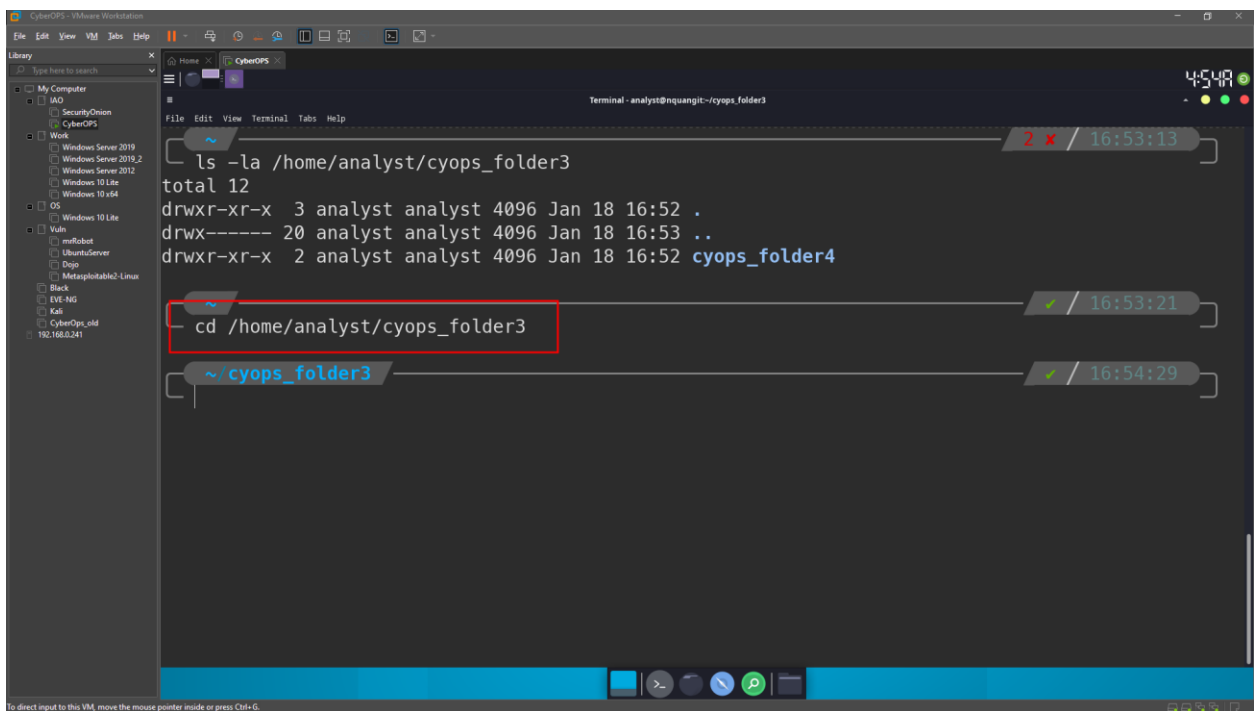
i. Up to this point, we have been using *full or absolute paths*. Absolute path is the term used when referring to paths that always start at the root (/) directory. It is also possible to work with *relative paths*. Relative paths reduce the amount of text to be typed. To understand relative paths, we must understand the . and .. (dot and double dot) directories. From the **cyops_folder3** directory, issue a **ls -la**:



The screenshot shows a terminal window titled 'analyst@nquangit-'. The command `ls -la /home/analyst/cyops_folder3` has been executed, resulting in the following output:

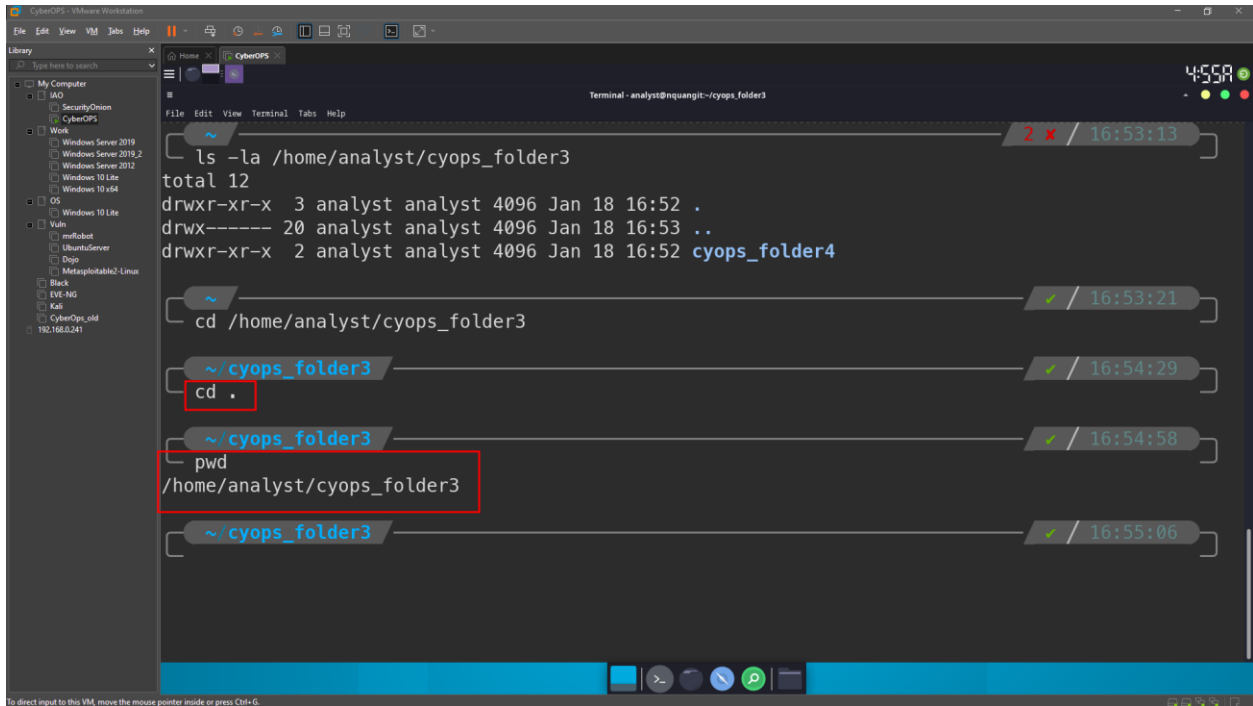
```
total 12
drwxr-xr-x 3 analyst analyst 4096 Jan 18 16:52 .
drwx----- 20 analyst analyst 4096 Jan 18 16:53 ..
drwxr-xr-x 2 analyst analyst 4096 Jan 18 16:52 cyops_folder4
```

j. Change the current directory to `/home/analyst/cyops_folder3`:



The screenshot shows the same terminal window. The command `cd /home/analyst/cyops_folder3` has been entered and is highlighted with a red box. The prompt has changed from `analyst@nquangit-~` to `analyst@nquangit-~/cyops_folder3`.

k. Type `cd .`



```

ls -la /home/analyst/cyops_folder3
total 12
drwxr-xr-x 3 analyst analyst 4096 Jan 18 16:52 .
drwx----- 20 analyst analyst 4096 Jan 18 16:53 ..
drwxr-xr-x 2 analyst analyst 4096 Jan 18 16:52 cyops_folder4

cd /home/analyst/cyops_folder3

~/cyops_folder3
cd .

~/cyops_folder3
pwd
/home/analyst/cyops_folder3

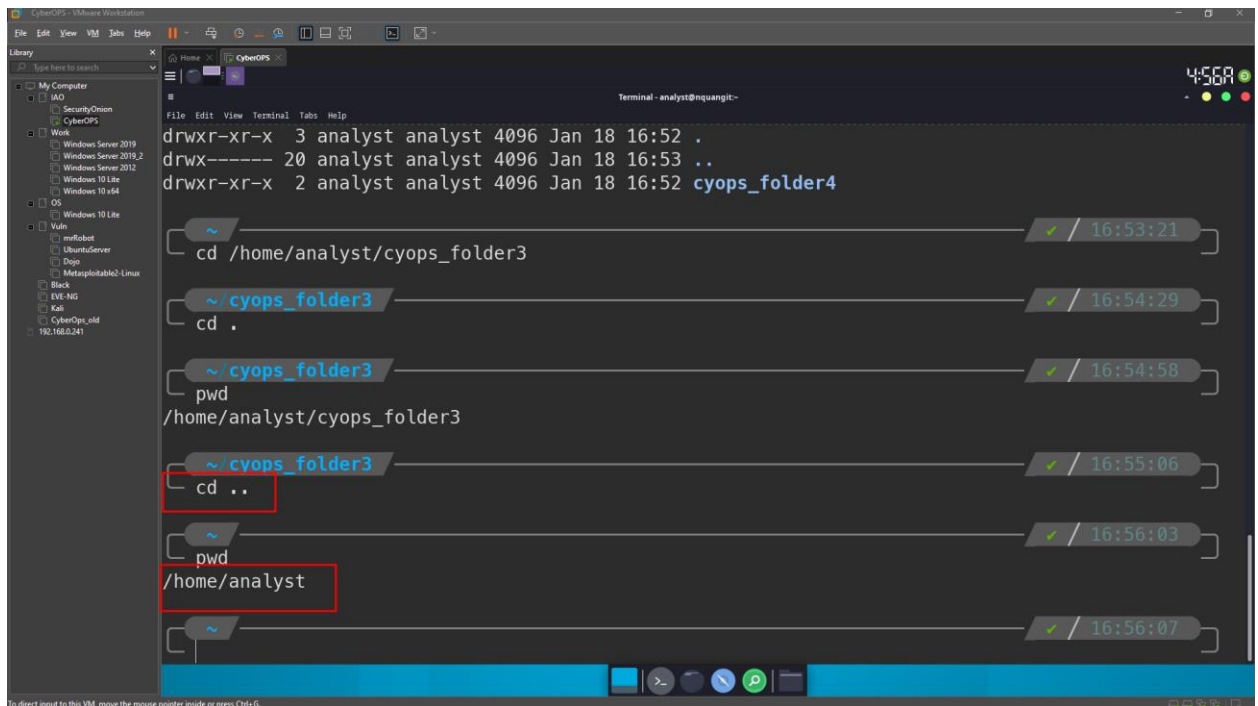
~/cyops_folder3

```

What happens?

It still in the **cyops_folder3** directory and with change to another.

- I. Changing the directory to the **..** directory, will change to the directory that is one level up. This directory is also known as *parent directory*. Type **cd ..**



```

drwxr-xr-x 3 analyst analyst 4096 Jan 18 16:52 .
drwx----- 20 analyst analyst 4096 Jan 18 16:53 ..
drwxr-xr-x 2 analyst analyst 4096 Jan 18 16:52 cyops_folder4

cd /home/analyst/cyops_folder3

~/cyops_folder3
cd .

~/cyops_folder3
pwd
/home/analyst/cyops_folder3

~/cyops_folder3
cd ..

~/
pwd
/home/analyst

~/

```

What happens?

The directory changed to the **cyops_folder3**'s parent: **/home/analyst**

What would be the current directory if you issued the `cd ..` command at `[analyst@secOps ~]$`?

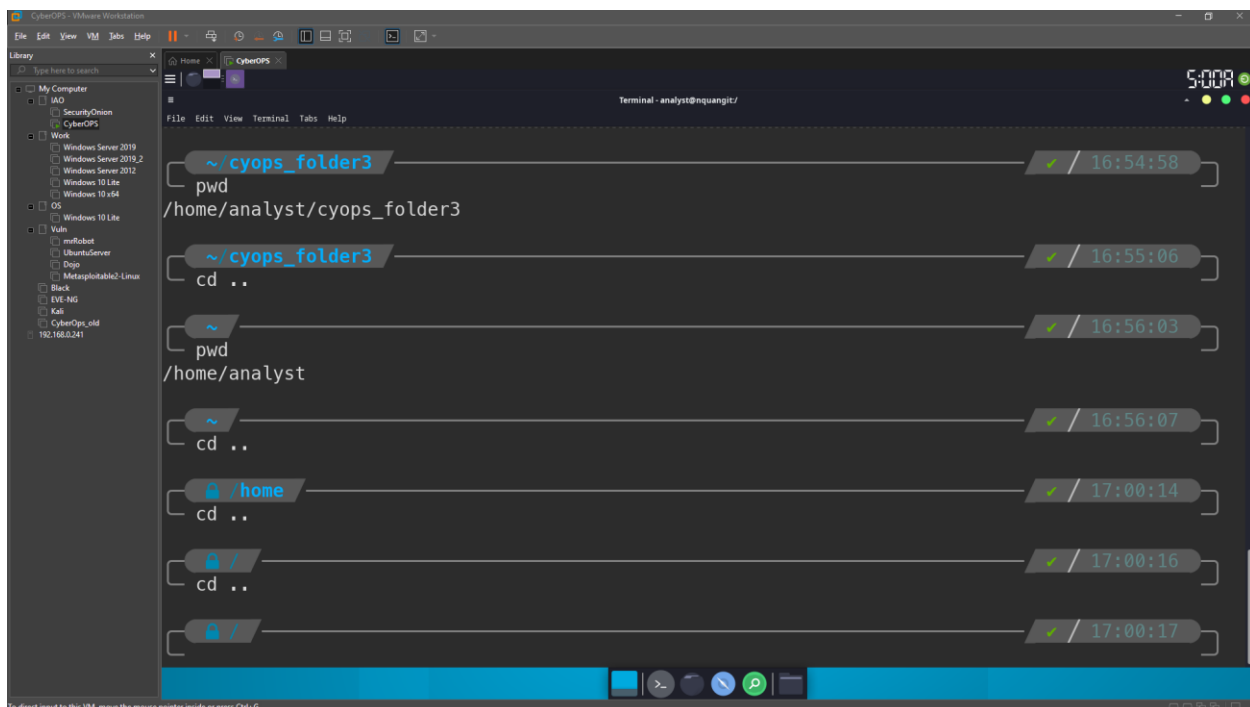
`/home`

What would be the current directory if you issued the `cd ..` command at `[analyst@secOps home]$`?

`/` (the root of the filesystem)

What would be the current directory if you issued the `cd ..` command at `[analyst@secOps /]$`?

`/` (the root of the file system. This is the highest level. It cannot go up)



```
~/cyops_folder3
pwd
/home/analyst/cyops_folder3

~/cyops_folder3
cd ..

~
pwd
/home/analyst

~
cd ..

/home
cd ..

/
cd ..

/
```

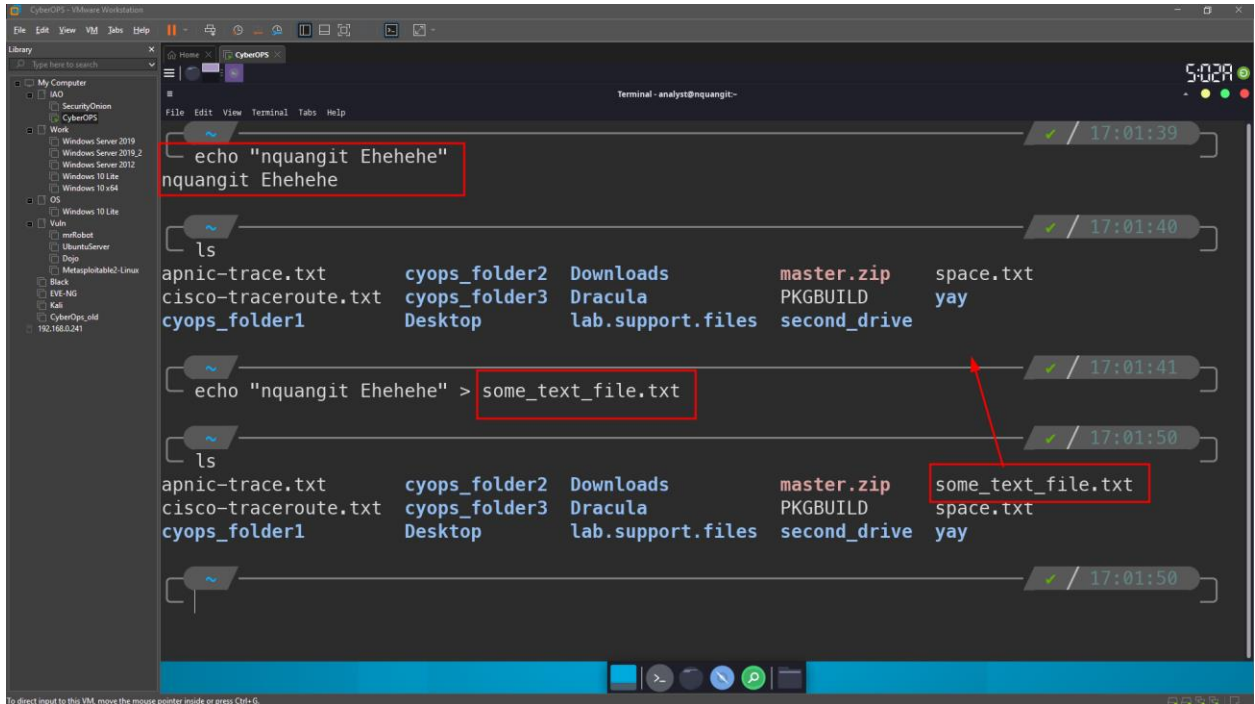
Redirect Outputs.

- Use the **cd** command to change to the **/home/analyst/** (**~**) directory:
- Use the **echo** command to echo a message. Because no output was defined, echo will output to the current terminal window:
- Use the **>** operator to redirect the output of echo to a text file instead of to the screen:

Is that expected? Explain.

Yes. The output was redirected to the `some_text_file.txt` file.

- d. Notice, that even though the **some_text_file.txt** file did not exist, prior to the echo command, it was automatically created to receive the output generated by **echo**. Use the **ls -l** command to verify if the file was really created:

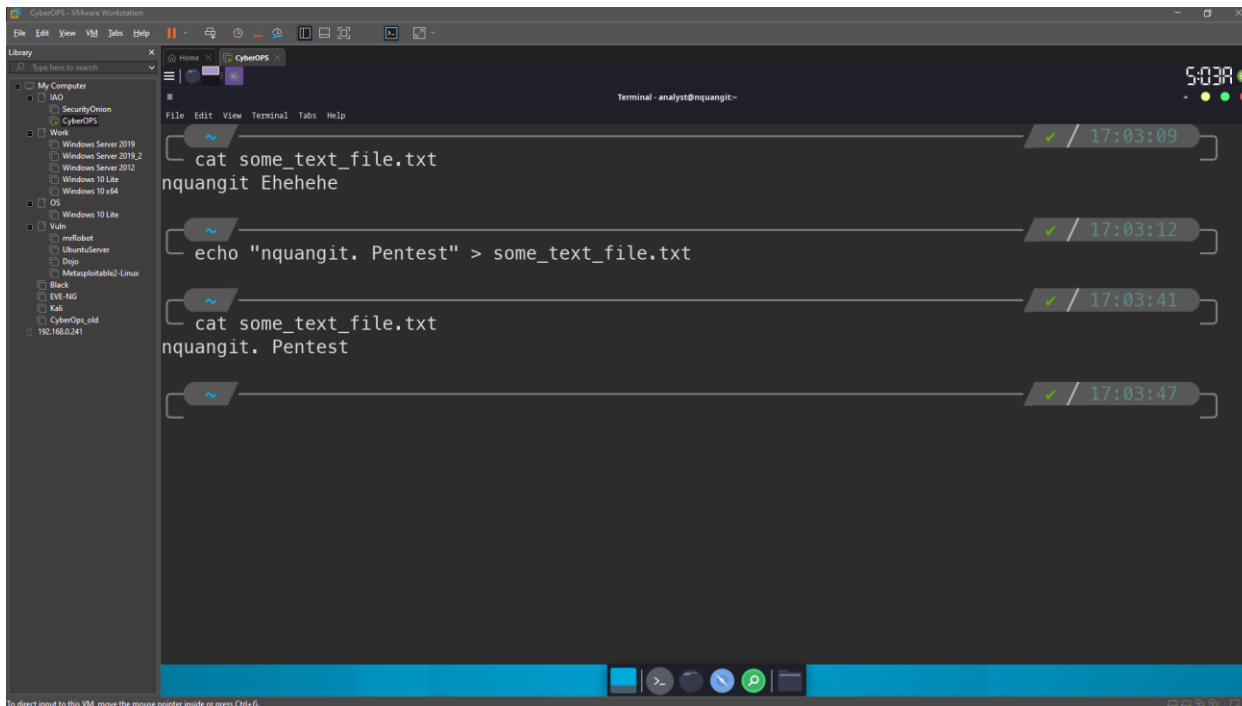


The screenshot shows a CyberOps virtual machine terminal window. The terminal has a dark background with a light blue title bar. On the left, there is a sidebar with a file explorer showing the contents of the virtual machine. The terminal window displays the following commands and output:

```
Terminal - analyst@nquangit-  
File Edit View Terminal Tabs Help  
17:01:39  
echo "nquangit Ehehehe"  
nquangit Ehehehe  
17:01:40  
ls  
apnic-trace.txt      cyops_folder2  Downloads      master.zip      space.txt  
cisco-traceroute.txt cyops_folder3  Dracula        PKGBUILD       yay  
cyops_folder1        Desktop        lab.support.files second_drive  
17:01:41  
echo "nquangit Ehehehe" > some_text_file.txt  
17:01:50  
ls  
apnic-trace.txt      cyops_folder2  Downloads      master.zip      some_text_file.txt  
cisco-traceroute.txt cyops_folder3  Dracula        PKGBUILD       space.txt  
cyops_folder1        Desktop        lab.support.files second_drive    yay  
17:01:50
```

A red box highlights the command `echo "nquangit Ehehehe"` in the first terminal session. Another red box highlights the command `echo "nquangit Ehehehe" > some_text_file.txt` in the second terminal session. A red arrow points from the second box to the file `some_text_file.txt` in the output of the `ls` command in the third terminal session.

- e. Use the **cat** command to display the contents of the **some_text_file.txt** text file:
- f. Use the **>** operator again to redirect a different echo output of echo to the **some_text_file.txt** text file:
- g. Once again, use the **cat** command to display the contents of the **some_text_file.txt** text file:



The screenshot shows a terminal window within a CyberOps virtual machine. The terminal displays the following commands and their outputs:

```

[analyst@secOps ~]$ cat some_text_file.txt
nquangit Ehehehe

[analyst@secOps ~]$ echo "nquangit. Pentest" > some_text_file.txt

[analyst@secOps ~]$ cat some_text_file.txt
nquangit. Pentest
  
```

The terminal output shows that the file `some_text_file.txt` was initially created with the content `nquangit Ehehehe`. After running the command `echo "nquangit. Pentest" > some_text_file.txt`, the file's content was replaced with `nquangit. Pentest`. The terminal also shows the file's content after the second `cat` command.

What happened to the text file? Explain.

The old text in the file was replaced by the new message. The `>` operator remove the file's content before writing the new data.

Redirect and Append to a Text File.

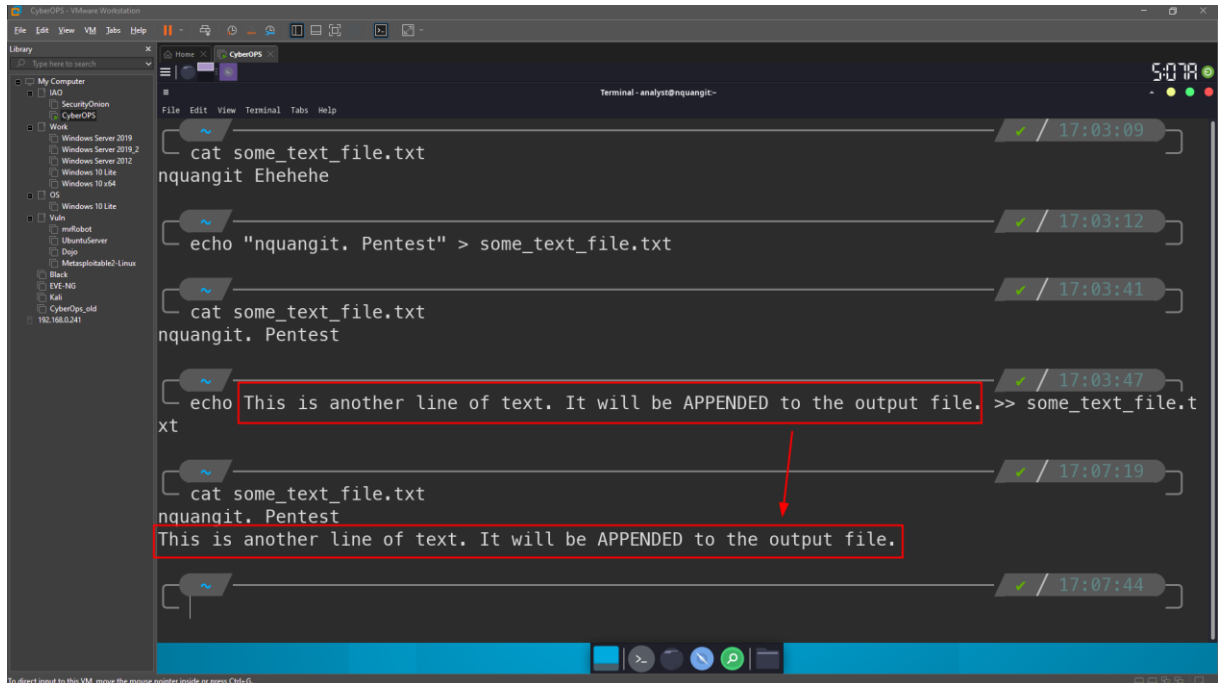
- Similar to the `>` operator, the `>>` operator also allows for redirecting data to files. The difference is that `>>` appends data to the end of the referred file, keeping the current contents intact. To append a message to the `some_text_file.txt`, issue the command below:

```
[analyst@secOps ~]$ echo This is another line of text. It will be APPENDED to the output file. >> some_text_file.txt
```

- Use the `cat` command to display the contents of the `some_text_file.txt` text file yet again:

```
[analyst@secOps ~]$ cat some_text_file.txt
```

This is a DIFFERENT message, once again echoed to the terminal by echo.
This is another line of text. It will be APPENDED to the output file.



```

cat some_text_file.txt
nquangit Ehehehe

echo "nquangit. Pentest" > some_text_file.txt

cat some_text_file.txt
nquangit. Pentest

echo "This is another line of text. It will be APPENDED to the output file." >> some_text_file.txt

cat some_text_file.txt
nquangit. Pentest
This is another line of text. It will be APPENDED to the output file.

```

Question:

What happened to the text file? Explain.

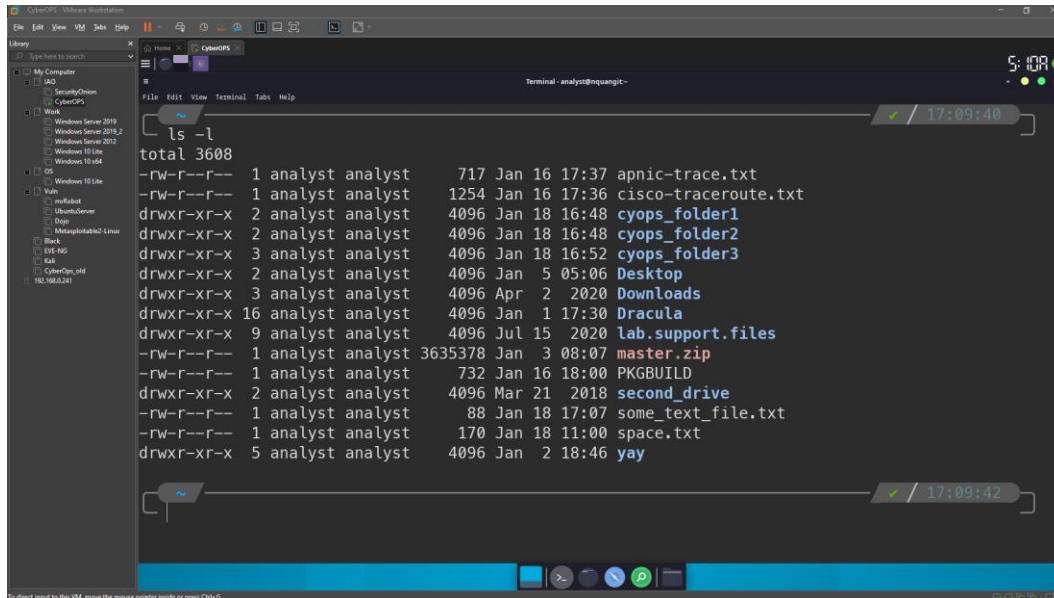
The new text was append to a new line of the end of the text file.

Work with hidden files in Linux.

- In Linux, files with names that begin with a '.' (single dot) are not shown by default. While dot-files have nothing else special about them, they are called hidden files because of this feature. Examples of hidden files are **.file5**, **.file6**, **.file7**.
- Use **ls -l** to display the files stored in the analyst home directory.

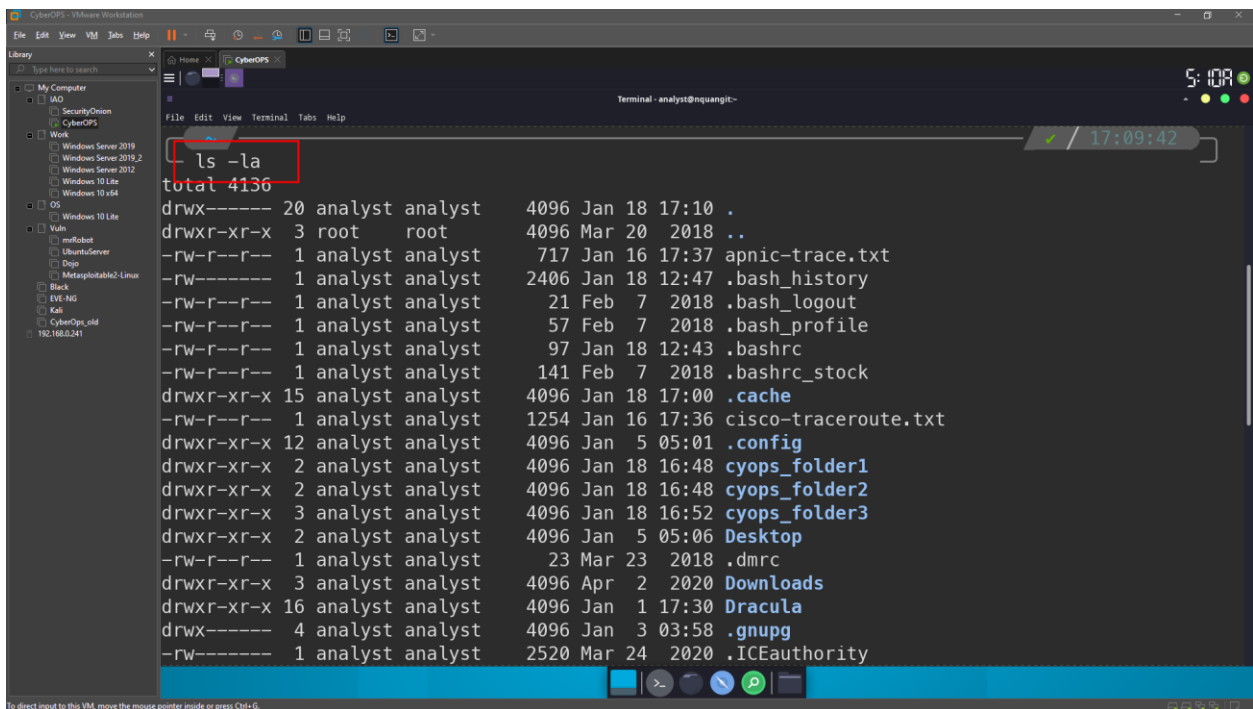
How many files are displayed?

15



```
ls -l
total 3608
-rw-r--r-- 1 analyst analyst 717 Jan 16 17:37 apnic-trace.txt
-rw-r--r-- 1 analyst analyst 1254 Jan 16 17:36 cisco-traceroute.txt
drwxr-xr-x 2 analyst analyst 4096 Jan 18 16:48 cyops_folder1
drwxr-xr-x 2 analyst analyst 4096 Jan 18 16:48 cyops_folder2
drwxr-xr-x 3 analyst analyst 4096 Jan 18 16:52 cyops_folder3
drwxr-xr-x 2 analyst analyst 4096 Jan 5 05:06 Desktop
drwxr-xr-x 3 analyst analyst 4096 Apr 2 2020 Downloads
drwxr-xr-x 16 analyst analyst 4096 Jan 1 17:30 Dracula
drwxr-xr-x 9 analyst analyst 4096 Jul 15 2020 lab.support.files
-rw-r--r-- 1 analyst analyst 3635378 Jan 3 08:07 master.zip
-rw-r--r-- 1 analyst analyst 732 Jan 16 18:00 PKGBUILD
drwxr-xr-x 2 analyst analyst 4096 Mar 21 2018 second_drive
-rw-r--r-- 1 analyst analyst 88 Jan 18 17:07 some_text_file.txt
-rw-r--r-- 1 analyst analyst 170 Jan 18 11:00 space.txt
drwxr-xr-x 5 analyst analyst 4096 Jan 2 18:46 yay
```

- c. Use the **ls -la** command to display all files in the home directory of analyst, including the hidden files.



```
ls -la
total 4136
drwx----- 20 analyst analyst 4096 Jan 18 17:10 .
drwxr-xr-x 3 root root 4096 Mar 20 2018 ..
-rw-r--r-- 1 analyst analyst 717 Jan 16 17:37 apnic-trace.txt
-rw----- 1 analyst analyst 2406 Jan 18 12:47 .bash_history
-rw-r--r-- 1 analyst analyst 21 Feb 7 2018 .bash_logout
-rw-r--r-- 1 analyst analyst 57 Feb 7 2018 .bash_profile
-rw-r--r-- 1 analyst analyst 97 Jan 18 12:43 .bashrc
-rw-r--r-- 1 analyst analyst 141 Feb 7 2018 .bashrc_stock
drwxr-xr-x 15 analyst analyst 4096 Jan 18 17:00 .cache
-rw-r--r-- 1 analyst analyst 1254 Jan 16 17:36 cisco-traceroute.txt
drwxr-xr-x 12 analyst analyst 4096 Jan 5 05:01 .config
drwxr-xr-x 2 analyst analyst 4096 Jan 18 16:48 cyops_folder1
drwxr-xr-x 2 analyst analyst 4096 Jan 18 16:48 cyops_folder2
drwxr-xr-x 3 analyst analyst 4096 Jan 18 16:52 cyops_folder3
drwxr-xr-x 2 analyst analyst 4096 Jan 5 05:06 Desktop
-rw-r--r-- 1 analyst analyst 23 Mar 23 2018 .dmrc
drwxr-xr-x 3 analyst analyst 4096 Apr 2 2020 Downloads
drwxr-xr-x 16 analyst analyst 4096 Jan 1 17:30 Dracula
drwx----- 4 analyst analyst 4096 Jan 3 03:58 .gnupg
-rw----- 1 analyst analyst 2520 Mar 24 2020 .ICEauthority
```

How many more files are displayed than before? Explain.

51. The output included all the files and folder was hidden.

Is it possible to hide entire directories by adding a dot before its name as well? Are there any directories in the output of ls -la above?

Yes, adding a dot before a folder's name will make it hide.

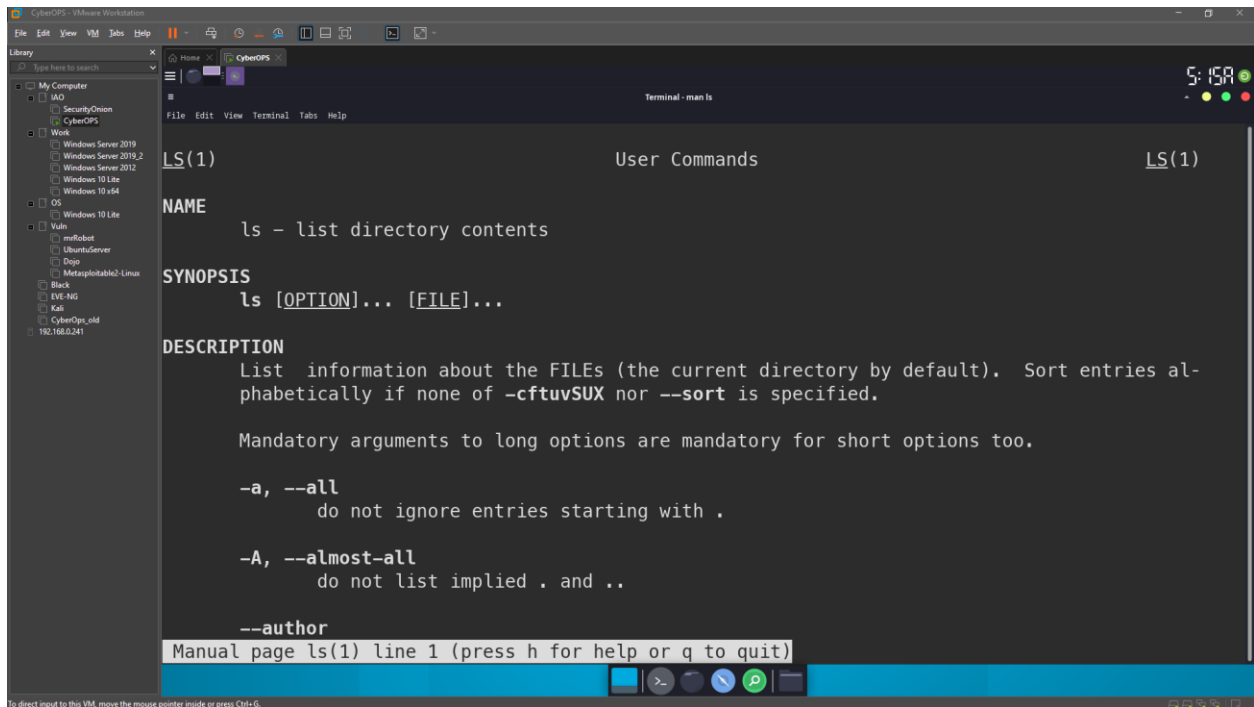
There are many hidden directories in the **ls -la** output

Give three examples of hidden files shown in the output of **ls -la** above.

.bash_history, .bash_profile, .bashrc_stock

d. Type the **man ls** command at the prompt to learn more about the **ls** command.

```
[analyst@secOps ~]$ man 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 al-
phabetically if none of -cftuvSUX nor --sort is specified.

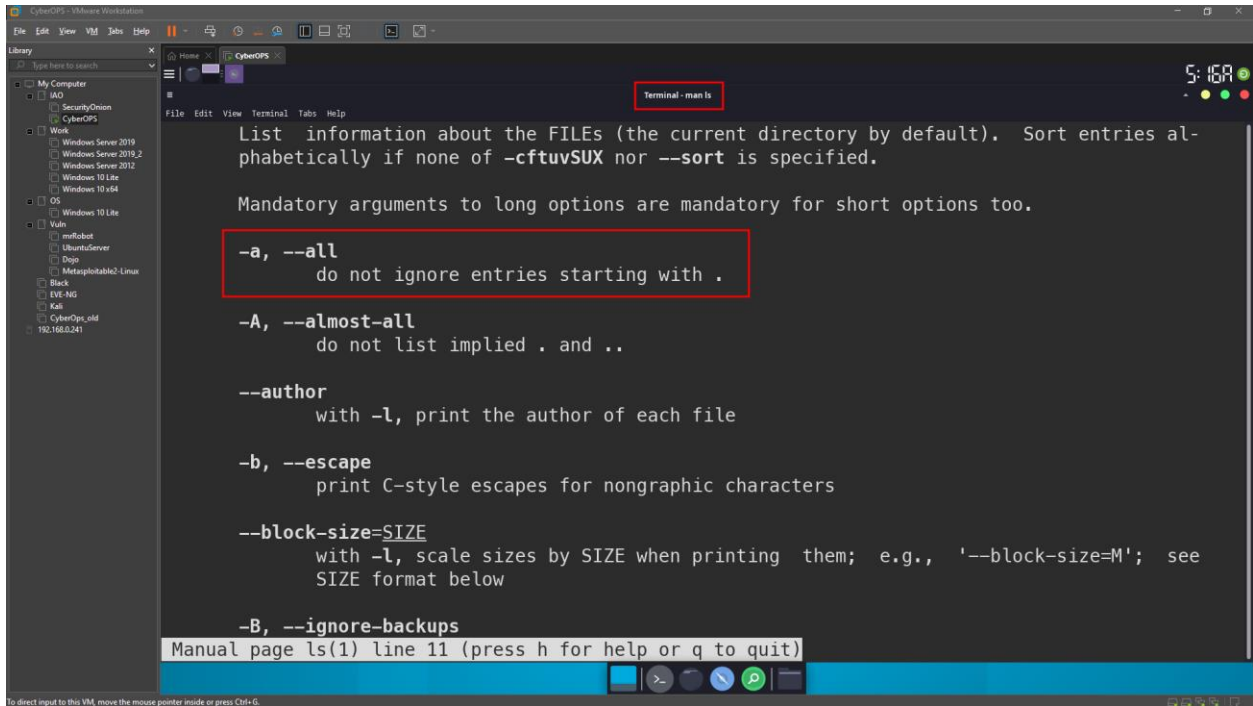
Mandatory arguments to long options are mandatory for short options too.

-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)
```

e. Use the down arrow key (one line at a time) or the space bar (one page at a time) to scroll down the page and locate the **-a** option used above and read its description to familiarize yourself with the **ls -a** command.



```

List information about the FILES (the current directory by default). Sort entries al-
phabetically if none of -cftuvSUX nor --sort is specified.

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
    with -l, print the author of each file

-b, --escape
    print C-style escapes for nongraphic characters

--block-size=SIZE
    with -l, scale sizes by SIZE when printing them; e.g., '--block-size=M'; see
    SIZE format below

-B, --ignore-backups
    do not list implied . and ..

Manual page ls(1) line 11 (press h for help or q to quit)
  
```

Copying, Deleting, and Moving Files

- The **cp** command is used to copy files around the local file system. When using **cp**, a new copy of the file is created and placed in the specified location, leaving the original file intact. The first parameter is the source file and the second is the destination. Issue the command below to copy **some_text_file.txt** from the home directory to the **cyops_folder2** folder:

```
[analyst@secOps ~]$ cp some_text_file.txt cyops_folder2/
```

Identify the parameters in the cp command above.

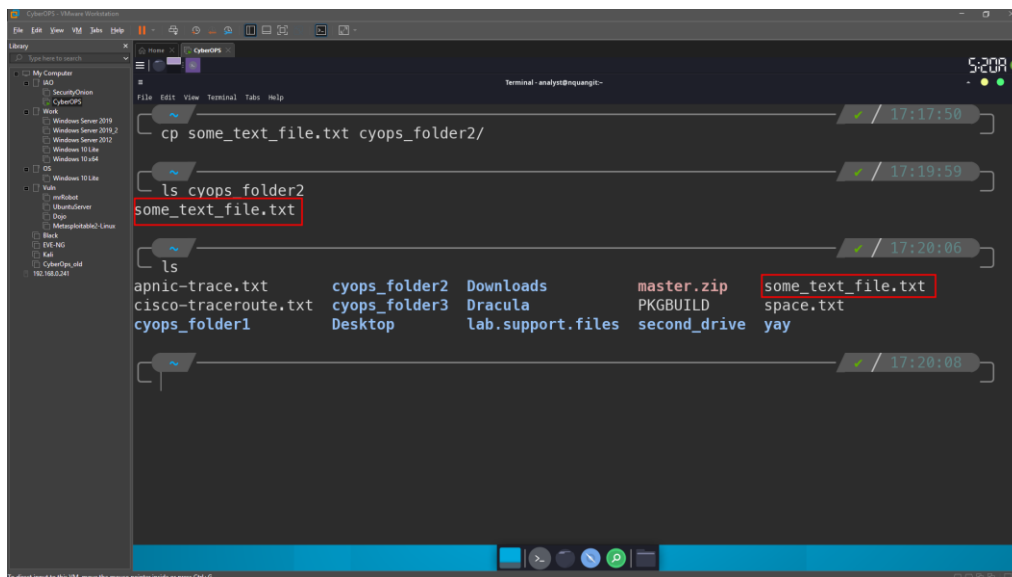
Question:

What are the source and destination files? (use full paths to represent the parameters)

Source: **some_text_file.txt** (absolute path: **/home/analyst/some_text_file.txt**)

Destination: **/home/analyst/cyops_folder2/some_text_file.txt**

- Use the **ls** command to verify that **some_text_file.txt** is now in **cyops_folder2**:
- Use the **ls** command to verify that **some_text_file.txt** is also in the home directory:



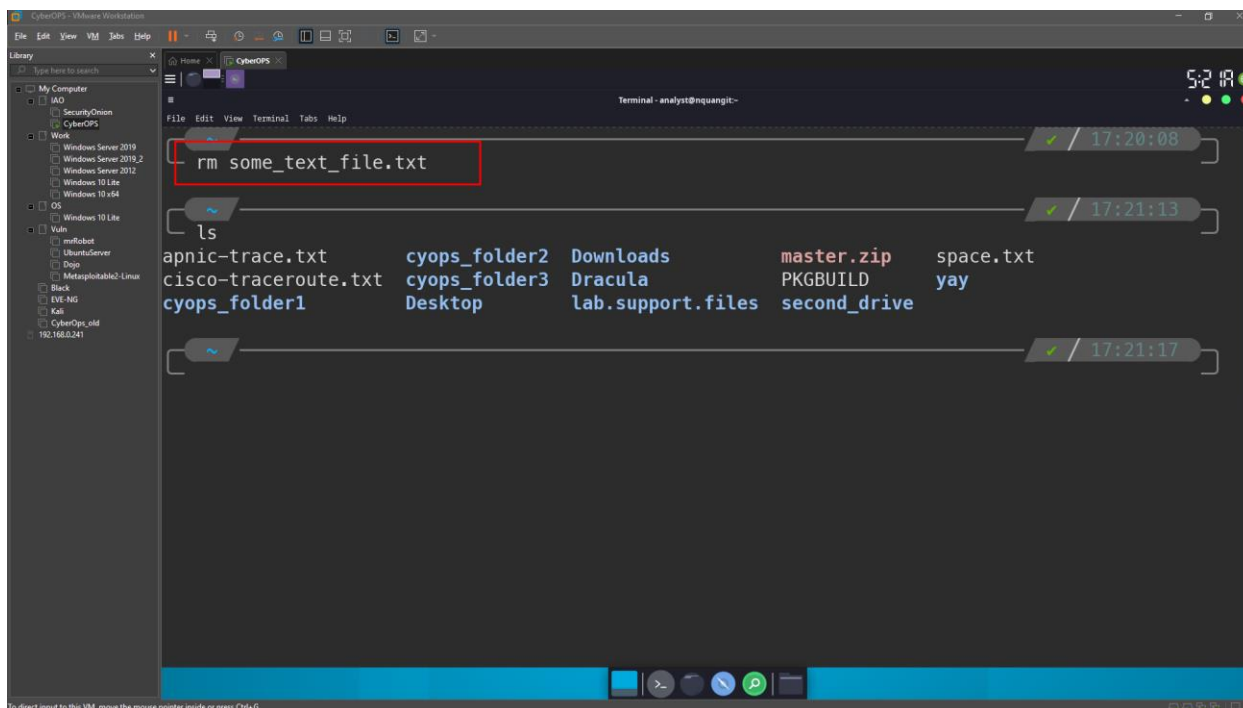
```

Terminal - analyst@nquangit-
17:17:50
cp some_text_file.txt cyops_folder2/
17:19:59
ls cyops_folder2
some_text_file.txt
17:20:06
ls
apnic-trace.txt      cyops_folder2  Downloads      master.zip      some_text_file.txt
cisco-traceroute.txt cyops_folder3  Dracula        PKGBUILD       space.txt
cyops_folder1       Desktop        lab.support.files second_drive    yay
17:20:08

```

Deleting Files and Directories

- Use the **rm** command to remove files. Issue the command below to remove the file **some_text_file.txt** from the home directory. The **ls** command is then used to show that the file **some_text_file.txt** has been removed from the home directory:



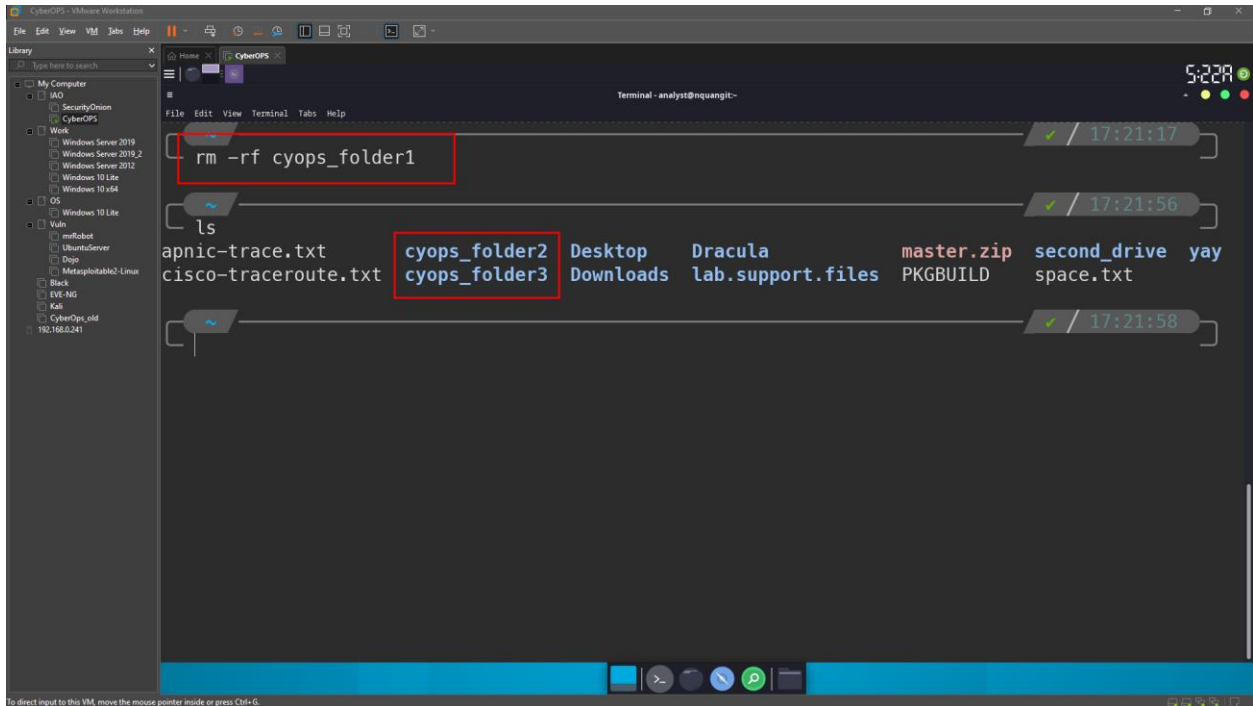
```

Terminal - analyst@nquangit-
17:20:08
rm some_text_file.txt
17:21:13
ls
apnic-trace.txt      cyops_folder2  Downloads      master.zip      space.txt
cisco-traceroute.txt cyops_folder3  Dracula        PKGBUILD       yay
cyops_folder1       Desktop        lab.support.files second_drive
17:21:17

```

- In Linux, directories are seen as a type of file. As such, the **rm** command is also used to delete directories but the **-r** (recursive) option must be used. Notice that all files and other directories

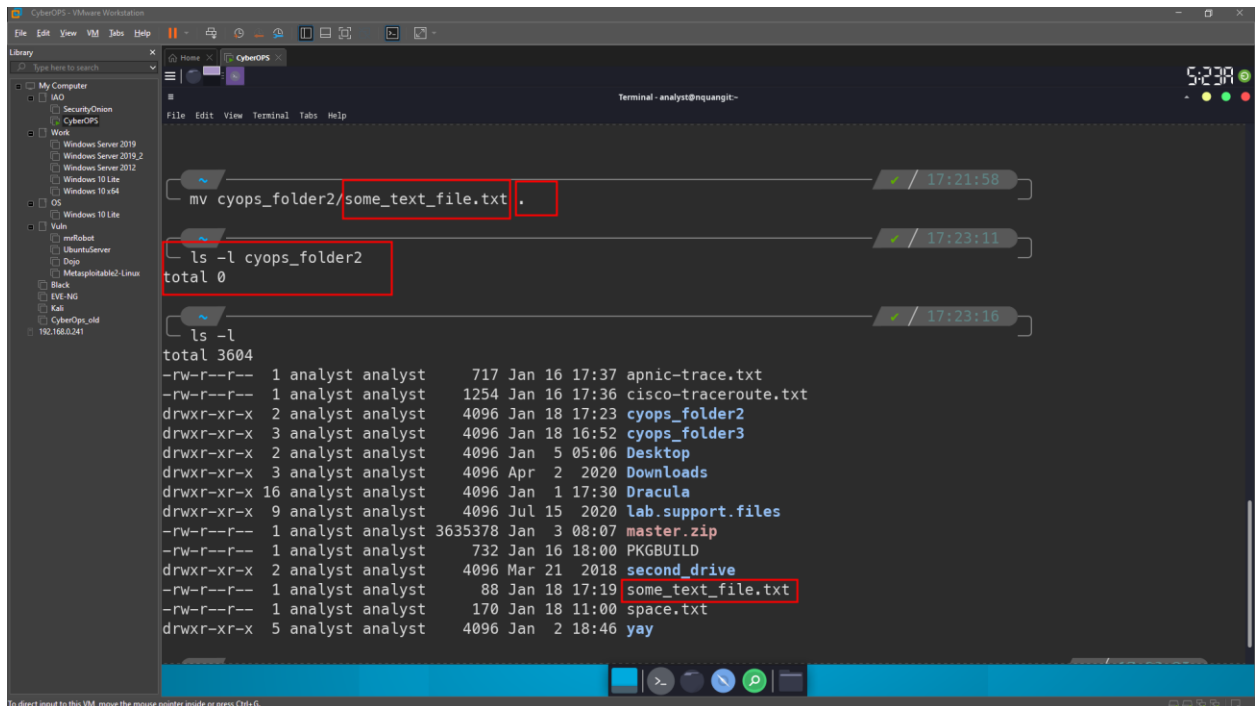
inside a given directory are also deleted when deleting a parent directory with the `-r` option. Issue the command below to delete the **cyops_folder1** folder and its contents:



```
Terminal: analyst@nquangit~  
File Edit View Terminal Tabs Help  
17:21:17 ✓  
rm -rf cyops_folder1  
17:21:56 ✓  
ls  
apnic-trace.txt      cyops_folder2 Desktop  Dracula  master.zip  second_drive  yay  
cisco-traceroute.txt cyops_folder3 Downloads lab.support.files PKGBUILD    space.txt  
17:21:58 ✓
```

Moving Files and Directories

- a. Moving files works similarly to copying files. The difference is that moving a file removes it from its original location. Use the **mv** commands to move files around the local filesystem. Like the **cp** commands, the **mv** command also requires source and destination parameters. Issue the command below to move the **some_text_file.txt** from **/home/analyst/cyops_folder2** back to the home directory:



The screenshot shows a terminal window titled 'Terminal - analyst@nquangit-'. The terminal displays the following commands and output:

```

mv cyops_folder2/some_text_file.txt .
ls -l cyops_folder2
total 0
ls -l
total 3604
-rw-r--r-- 1 analyst analyst 717 Jan 16 17:37 apnic-trace.txt
-rw-r--r-- 1 analyst analyst 1254 Jan 16 17:36 cisco-traceroute.txt
drwxr-xr-x 2 analyst analyst 4096 Jan 18 17:23 cyops_folder2
drwxr-xr-x 3 analyst analyst 4096 Jan 18 16:52 cyops_folder3
drwxr-xr-x 2 analyst analyst 4096 Jan 5 05:06 Desktop
drwxr-xr-x 3 analyst analyst 4096 Apr 2 2020 Downloads
drwxr-xr-x 16 analyst analyst 4096 Jan 1 17:30 Dracula
drwxr-xr-x 9 analyst analyst 4096 Jul 15 2020 lab.support.files
-rw-r--r-- 1 analyst analyst 3635378 Jan 3 08:07 master.zip
-rw-r--r-- 1 analyst analyst 732 Jan 16 18:00 PKGBUILD
drwxr-xr-x 2 analyst analyst 4096 Mar 21 2018 second_drive
-rw-r--r-- 1 analyst analyst 88 Jan 18 17:19 some_text_file.txt
-rw-r--r-- 1 analyst analyst 170 Jan 18 11:00 space.txt
drwxr-xr-x 5 analyst analyst 4096 Jan 2 18:46 yay

```

What command did you use to accomplish the task?

```
mv cyops_folder2/some_text_file.txt .
```

```
rm -r cyops_folder2/
```

Reflection

What are the advantages of using the Linux command line?

Advantages of Linux Command Line:

Efficiency: Faster and more precise execution of tasks.

Scripting: Powerful scripting capabilities for automation.

Resource Efficiency: Lower system resource usage compared to graphical interfaces.

Remote Access: Facilitates remote server management.

Customization: Highly customizable and scriptable for personalized workflows.

Learning Curve: Enhances understanding of system internals and commands.

Server Administration: Ideal for server environments with no graphical interfaces.

Resource Monitoring: Robust tools for monitoring system resources.



Text Processing: Powerful text processing tools for data manipulation.

Script Portability: Scripts are often portable across different Linux distributions.