# Computer Security 1 Lab 1

Joseph Abel

October 27, 2018

Welcome to Computer Security 1! There is a growing need for professionals who understand computer security topics. The first lab will explain about virtual machines, go through the installation process of the Ubuntu Operating System and practice some basic Linux commands work.

## 1 Virtual Machines

What is a Virtual Machine? In order to define this we need to first define the term Virtualization. Virtualization is an abstraction within Information Technology that allows you to run multiple instances of an operating system isolated from your physical machine. This means if you install an operating system on a virtual machine it won't do anything to your primary operating system which is also known as the host operating system. The host operating system is the software which is installed on your main computer that interacts with the hardware. These virtual machine instances that we are describing are what we call guest operating systems or virtual machines. As they are being run in a virtual circuit with virtual resources and no actual underlining hardware.

The first thing one should consider when setting up virtual machines are the hardware requirements. For instance some of the questions we need to ask are the following:

- How many cores do we want the CPU to have?
- How much RAM do we want?
- Will the virtual machine be in host mode or not?

All these questions and many more questions allow us to set up our virtual resources for our virtual machine. Virtual resources are just like physical resources for an operating system except well virtual. Every device connected to a computer system is a resource and these resources have limited availability within the system/virtual machine. These resources which we are describing are managed by the operating system. Virtual resources include files, network connections, memory areas, CPU TIME, memory like RAM and virtual, hard disks, network throughput, battery power and external devices.

The second thing we consider is a hypervisor. A hyper who? A hypervisor is software that manages multiple operating systems (or multiple instances of a virtual machine) in a single computer system. The hypervisor manages the systems resources to allocate what each operating system requires. Hypervisors can also be called virtualization machine monitors meaning that we can run multiple virtual machines and keep track of them all in one place.

At a high level we can depict how virtual machines function we can break this architecture down in a picture like this:

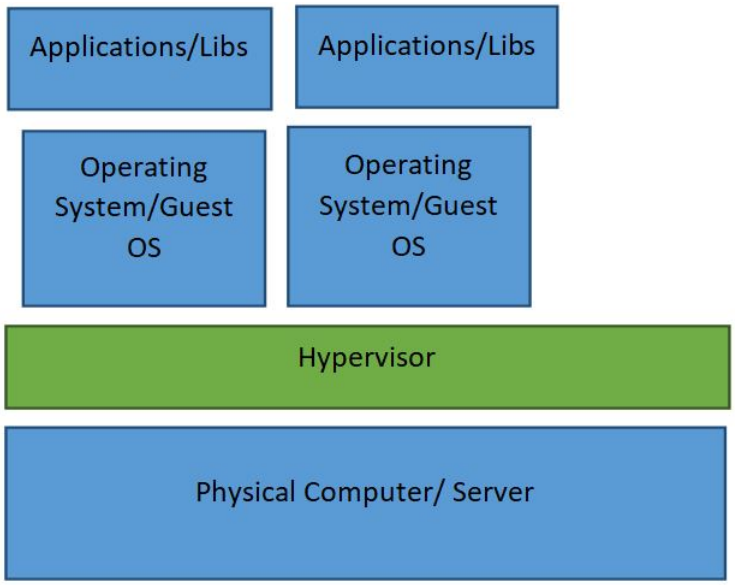


Figure 1: A high level view of how a virtual machine operates

Learning virtual machines can be tricky at first but as you become familiar with them and work with them. **Over time you will appreciate their power and ability to isolate work from your actual physical computer.**

# 2 Ubuntu

In this course we will use virtual machine's to host an operating system called Ubuntu "pronounced: oo-boon-too". Ubuntu is a powerful Linux operating system based on another Linux Operating System called Debian. (Linux is an open source operating system written in the C programming language and Debian is a Unix-like operating system that is composed of entirely free software). Ubuntu is commonly used to host web servers, run cloud infrastructures, run docker containers and run Internet of Things devices.

Behind every website there is typically a Linux server more specifically an Ubuntu Linux web server (it is usually used for performance, security and stability and etc...). For example if you were starting up a business you would run that website on an Ubuntu web server. Any of the big tech companies like Google, Facebook, IBM, Oracle and Amazon have custom web servers that run their technology stacks but these web servers are all likely based off of Linux.

# 3 Installing Ubuntu on a Virtual Machine

Now that we have gone through the basics of how virtual machines function it is time to set up Ubuntu. In order to do this we will be using a piece of software called VMware Workstation Player. VMware Workstation player is virtualization software which enables IT professional and students to run virtual machines and manage them. Since you will be running these labs in class you we will be using VMware Workstation player

**Step 1:** We are going to create a new virtual machine. In order to do this we will open up VMware workstation player. Then navigate to player-¿ File-¿ New Virtual Machine (or Ctrl+N)

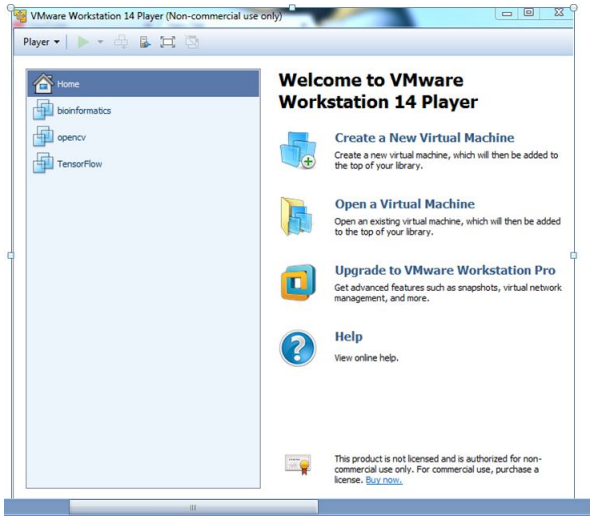


Figure 2: The main menu screen of VMware Player

**Step 2:** In this step we will install a disc image file of Ubuntu on VMware Workstation Player. In order to install this you will have the options to install the disc, install the operating system later, or installer the disc image file (iso). We will click the middle option and then browse and go to our path where the Ubuntu ISO is located.

If you can't find the Ubuntu ISO and it isn't already on your computer go to the Ubuntu website and download the disk image Ubuntu:

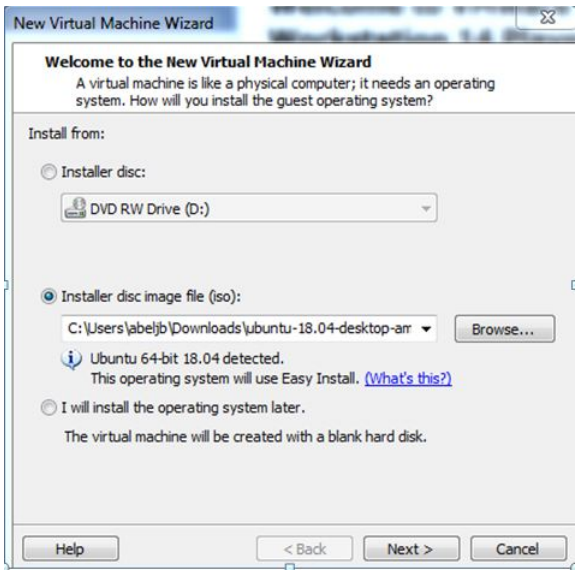


Figure 3: Setting up the path to the Ubuntu ISO file.

**Step 3:** In this step we are going to personalize our virtual machine. VMware Workstation Player will allow us to personalize your machine by adding a username, full name and a password. (make sure the user name and full name is lowercase):

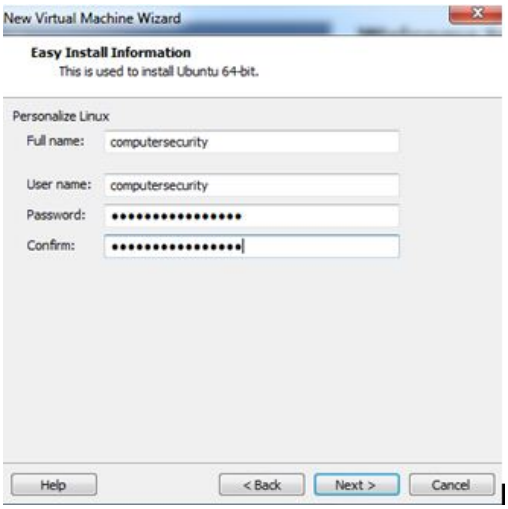


Figure 4: The user's information for the virtual machine which you are trying to use.

**Step 4:** In this step we are going to name the virtual machine. It is good practice to not put any spaces in put underscores in-between words or capitalize the beginning of each new word:

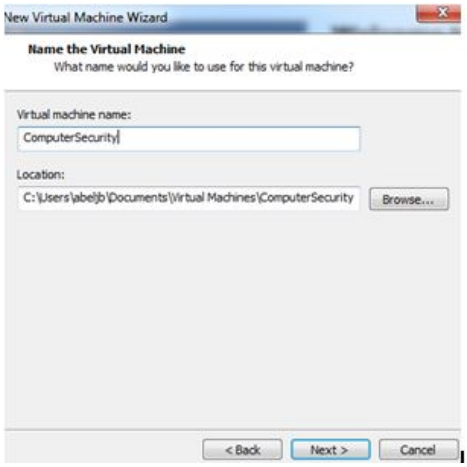


Figure 5: The virtual machine name and location is shown above.

**Step 5:** Here we specify the disk capacity we can keep these settings exactly as they are. Virtual Machine disk capacity is the way the virtual machines save's its data. Think of it has a virtual hard drive for the virtual machine.

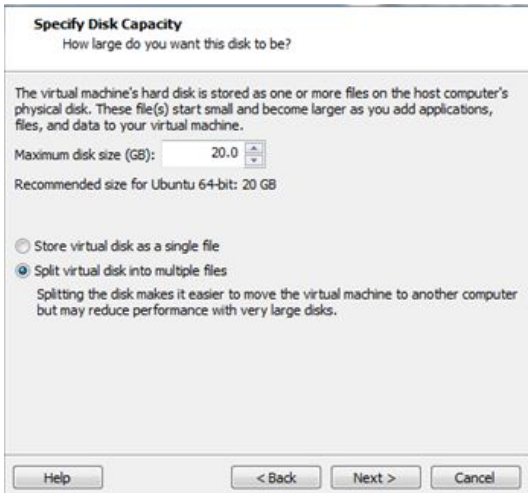


Figure 6: Specify the virtual machine disk capacity.

**Step 6:** Here we can customize the virtual machine resources we are going to customize it so it says 4096 MB for memory which is equal to 4GB and change processors to two. Check out the images below:

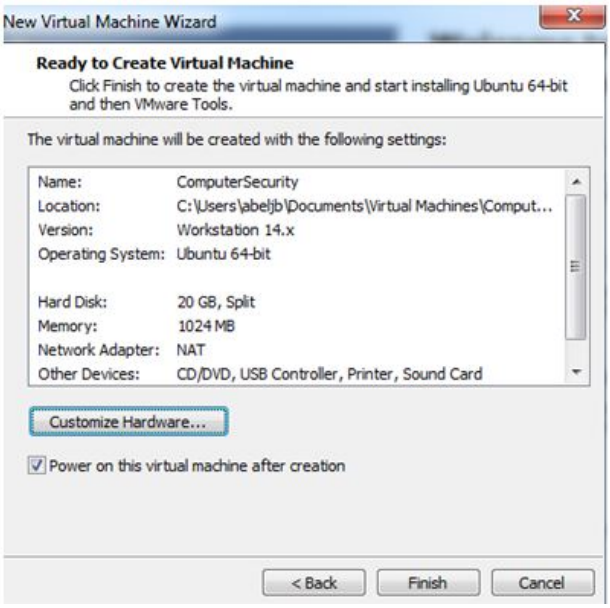


Figure 7: Finalizing the virtual machine settings. This final page also contains the option to "customize hardware". This option will allow us to accomplish changing the memory and processor core allocations.

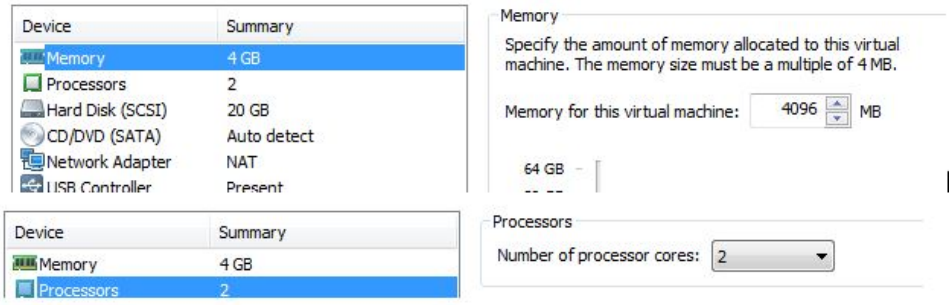


Figure 8: The customization of hardware for the virtual machine which we are making. As seen from step 6.

**Step 7:** the VMware player do its magic. You should see a large screen with this in the middle(if it freezes try again close it power it off and re try make sure the processors are 2 and the memory is 4GB:


Figure 9: This is the Ubuntu loading screen.

**Step 8:** Answer the questions in this part and watch your setup:

**Question 1.1:** What are the components which make up a virtual machine and explain them?

**Question 1.2:** What is an operating system resource name 5?

**Question 1.3:** What is Ubuntu?

**Question 1.4:** What is a virtual machine?

**Step 9:** Once the setup is complete we will be prompted to log on and enter your password:


Figure 10: This is the first user of the virtual machine created in the steps above. The user name is computersecurity.

**Step 10:** Explore the Ubuntu Desktop. In this class and in your career you will be using the command line but becoming familiar with the Graphical User Interface (GUI) of Ubuntu is still worthwhile.

**Step 11:** Click on the main menu icon at the Task Bar. The main menu icon is displayed as the 3x3 dots at the bottom left of the screen:

Figure 11: The main menu icon on the Ubuntu Task bar.

**Step 12:** Next we will open the terminal. In order to open the terminal where it says "type to search" we can type in "Terminal" and click on the appropriate result. The image is shown below:

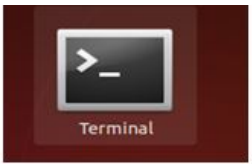


Figure 12: The terminal icon that allows us to access the terminal in Ubuntu

**Step 13:** Right click on the Terminal icon in the task bar and click "add to favorites" . The terminal will now be added to the task bar. Check out the image below:

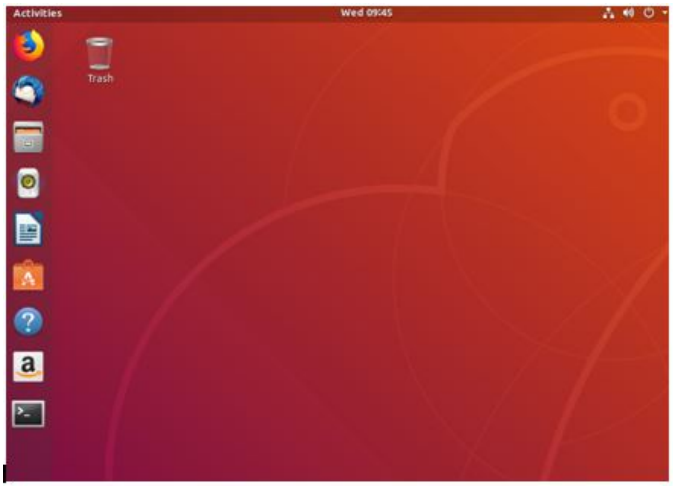


Figure 13: This is the Ubuntu home screen when we do a fresh install of Ubuntu Desktop this should come up.

Congratulations! You just set up an Ubuntu Virtual Machine. Now it is time to familiarize yourself with how Ubuntu Operates in the terminal.

# 4 Linux Permissions

Before we start entering in commands we need to learn one thing called Linux Permissions.

In Linux there are two basic file types normal and special. The file type is indicated by something we call the type field.

As discussed above, normal files/regular files will have a "-"or a hyphen character in the type field for their file type.

Special files can be identified by files that have a non-hyphen character such as a letter in their file type field. The Operating System handles special files. For example we will see there is a letter "d" for directory in the type field later and much more. We will see this in the ls section.

In the "ls" section of this lab we will go through this more. But in the meantime we are going to explain about read/write permissions. These read/write permissions have something we call permission classes. Permission classes in Linux are broken down to user, Group and other

The user is the owner of a file and is identified by a numerical called the UID or unique identification number. There are two types of user the root user and the normal user. A root user can access all the files and modify an account. A normal user has limited access to files.

The group is several users organized together into one Unique Identification number or UID. There are two types of groups: primary group and supplementary group. The primary group is the group applied to you when you log in. The primary group is used by default when creating new files and directories, modifying files or executing commands. The supplementary group(s) is any groups you are a member of beyond your primary group. Supplementary groups are useful when you are part of an organization with multiple teams working on a project. Each user would be assigned to a supplementary group or groups depending on their role which would allow them to access the files they need based upon their membership to each group. This would then restrict access to file not deemed necessary for their groups.

The other is for any user that is not part of the user or group class.

For instance an example we will see below is the following permission:

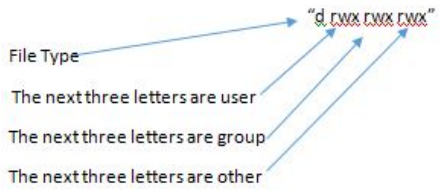


Figure 14: Understanding how permissions work.

# 5 Access Modes

Access modes are the ways in which someone may use a file. There are three access modes: read, write and execute.

The read access mode is denoted is denoted with the character “r” Allowing a user read access for a normal file will allow the user to view the contents of that file. Allowing a user read access for a directory will in addition allow a user to view the names of the file(s) in that directory.

The write access mode is denoted with the character “w” Allowing a user write privileges to a given file will enable the user to modify and delete the file. Mean while with a directory enabling write permissions allows a user to delete the directory and modify its contents.

The execute access mode is denoted with the character “x”. Allowing a user execute access for a normal file will allow the user to execute the file. Allowing a user execute permissions for a directory will allow the user to access or traverse and access metadata about files and directories. Note: the user must also have read permissions to do this.

If there is a “-“hyphen in place of where an access mode should be listed it means that user doesn’t have that permission class.

# 6 Linux Commands and the Terminal

## 6.1 The PWD Command

The first command we are going to learn is the **pwd** command. The pwd command is an acronym for present working directory.

What is a directory? A **directory** is a place for storing files, similar to a folder. The difference between a directory and a folder is the GUI: the contents of folders can be reviewed in a GUI, but the contents of a directory are reviewed on the command line without a GUI. The PWD command once entered into the terminal will output your current directory

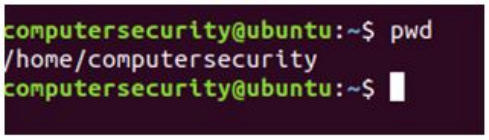


Figure 15: The command above shows us how to check our present working directory.

The output of Figure 1.15 is /home/computer security the home directory for the user computersecurity. This is our main directory. Every time you boot I virtual machine it will go to this folder unless you instruct it to do otherwise.

## 6.2 The CD Command

The **cd** command stands for change directory. This command takes a directory as an argument and changes your directory to the directory specified in the argument. For instance if you type cd Desktop you will maneuver to a directory called Desktop:

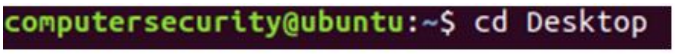


Figure 16: The command above illustrates how we can use the cd command to move from one directory to another.

As you can see right away the directory change is present by adding a blue labeled keyword" /Desktop" This indicates to the terminal and you that you are currently in that Desktop directory don't believe me run the pwd command:

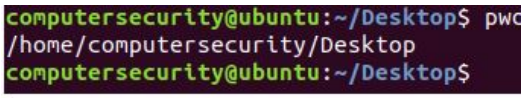


Figure 17: Let's prove the way the cd command works by using the pwd command. The output of the pwd command after entering cd Desktop is the Desktop directory, proving that cd changed the directory we are in.

Note: A helpful tip to keep in mind as you use the cd command is that you can further maneuver using the period character. For example, the command cd.. Would move up one directory. In this case you would maneuver back to the home folder. Alternatively the cd command with no directory as an argument will always return you to the home directory.

**Question 1.5:** Maneuver to the /usr/bin directory and take a screenshot of it. (Inside this bin directory is where many standard commands are installed.)

**Question 1.6:** Run the ls command in the bin directory and take a screenshot of the output is? Describe what the ls command did?

## 6.3 The ls command and the man command

The ls command lists the contents in your current directory. The ls command gives you the opportunity to start learning about flags. Flags are a way of getting a specific output or more detailed output.

To learn more about flags use the man command, this will bring up the manual for that command. To learn more about the ls command, type: "man ls"

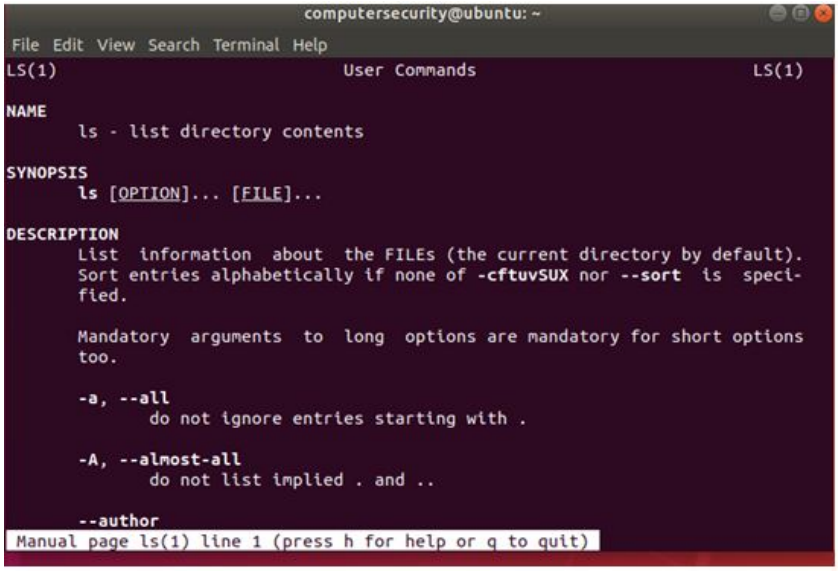


Figure 18: The output of the man ls command, the manual for the ls command. The manual includes information about the flags that can be used with the ls command.

**Question 1.7:** What do the following flags do when used with the ls command?

a. –i

b. –l

c. –r

d. –r

e. –s

f. –a

g. No Flag

## 6.4 An example of the ls command

Maneuver to the /usr/share directory. The share directory allows files to be tested on multiple networks, but don't concern yourself with that now. Once in the share directory we can enter the ls command to list the contents of share.

The share directory contains a lot of subdirectories, which will give you an opportunity to begin observing the color-coding used in Ubuntu.

- **Blue**: Directory
- **Green**: Executable or recognized data file
- **Sky Blue**: Symbolic link file
- **Yellow with black background**: Device
- **Pink**: Graphic image file
- **Red**: Archive file
- **Red with black background**: Broken link

Figure 19: The color schema when describing the different colors and their meanings in Ubuntu.

The listed contents of the share directory might not make much sense at this point unless you have prior experience with Linux. However you can observe the colors and see that all of the listings are directories.

Now enter the " ls – l" inside /usr/share. The output of ls using the -l flag provides more details for each of the directories. What do these details mean? To find out check out below!

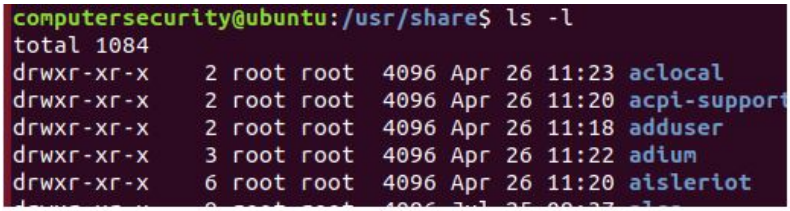


Figure 20: The output of the ls –l command. This is done to show the user permissions field.

The first block describes in this output below describes the file type (if the column is a "d" the item is a directory if it is a "-"it is a normal file the permissions are listed after the type. To understand some of the permissions listed on your screen check out the table below:

| -rw-r—r— | drwxr-xr-x |
|---|---|
| "-" means the file is a regular file | "d" means the file is a directory |
| "rw-" this means the file's owner can read and write from the file | "rwx" means the directory's owner can list its contents and create new files within it and can view the contents. |
| "r—" means members of the fie's group can read from the file | "r-x" means members of the directory's group can list its contents and view it |
| "r—" means other uses can read from the file | "r-x" means other uses can list the directory's contents and view it |

Figure 21: Table of permissions

Let's address figure 1.20 again the "total" value on the top is the block size which states the total disk allocation for all the files in that directory. The first column, as you now know, states the file type and its permissions. The second column indicates the number of hard links (you will learn about hard links in a future lab), the third column states the owner, the fourth column states group owner, the fifth column states the size of the file or directory, the sixth-eighth column states the time at which the file or directory was last modified, and the ninth column states the file or directory name.

**Question 1.8:** Examine the different permissions and tell me the type of file and what permissions the user does and doesn't have:

| Permissions | What type of file is it? | What permissions does the user have | What permissions the user doesn't' have |
|---|---|---|---|
| -rw------- | | | |
| -rwxrwxr-x | | | |
| drw-rw-rw- | | | |
| drwxrwx--- | | | |
| drwxr-x--- | | | |
| -rwxrwxrwx | | | |
| drwxrwxr-x | | | |

Figure 22: A question in regards to permissions. Create a separate table in a word docuemnt to answer it

**Question 1.9:** In the home directory analyze all the directories and files in there and describe the users and permissions for each. Create it all into a nice table.

## 6.5 Superuser Do

In Linux we define someone who has "superuser do" or" root access" privileges as an administrator. The **sudo** command allows a system administrator to delegate authority to certain users or a group of users. If a file is developed

with the sudo command that file can only be altered by the administrator who has those credentials. For every terminal session the administrator must enter in their credentials in order to use the sudo command.

For instance if you want to create a text file called farmingdale.txt using sudo privileges. Since you are using sudo it will prompt you for your password to create the text file. Let's break down an example command:

sudo nano farmingdale.txt

sudo: The administrative command. It is only used by administrators and delegates commands to groups or users. It allows the creation of password protected administrative files.

nano: The nano command specifies the text editor to be used. In this case the text editor is called nano. Two other popular text editors are Emacs and vim.

farmingdale.txt: Specifies the file and file type to be created. Here a file farmingdale is created as a text file (.txt). In summary, you are creating an administrative file using a text editor called farmingdale.txt.

In summary, you are creating an administrative file using a text editor called farmingdale.txt.


Figure 23: Creating an administrative text file. Note that the system prompts for the user's password before the file is created.

Once the text file is created it can be edited. Check out the images below:

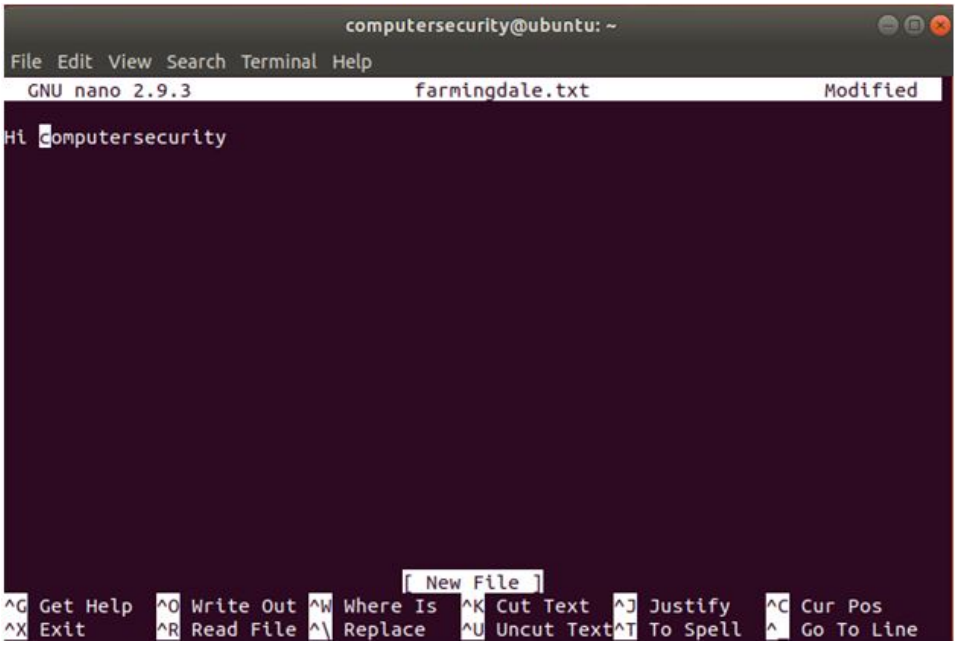


Figure 24: A screenshot of the nano text editor engaged in the editing of the text file farmingdale.txt

Once you have finished editing the text file you will need to exit the text editor. As you can see in the screenshot (figure 6.3), you will type "^, X" to quit. The "^, " character denotes the control key, so the command to exit would be ctrl+x, ctrl+g, ctrl+o, ctrl+r and etc...

In response to your command that the text editor exit you will be prompted to save, or not save, your work. In response type "Y" for yes, "N" for no. If you attempt to exit the program without meaning to, type control C to cancel the exit request and return to editing the file.

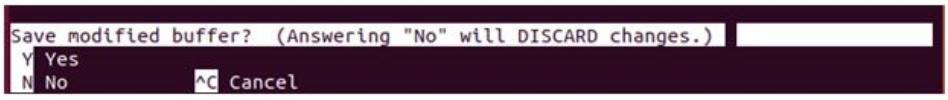


Figure 25: Screenshot of nano prompting the user to save the changes made to the text file file.

This file was created using sudo permissions, or administrative privileges, meaning that if a normal user attempted to edit it they would not be able to do so.

**Final Questions**

**Question 1.9:** Without using the sudo command attempt to edit the file farmingdale.txt with the command nano farmingdale.txt. Then try to save your edits. What error message do you receive? Why do you receive it? (Hint: sudo is an administrator and you are just a user)

**Question 1.10:** What does the pwd command do?

**Question 1.11:** Name the most common text editors that can be used on the command line. Which text editor did you use to complete this lab?

**Question 1.12:** What's the difference between a user and a group?

**Question 1.13:** Identify five flags that can be used with the sudo command and describe their purpose.

**Question 1.14:** What is the difference between UNIX and LINUX?

**Question 1.15:** What is the difference between a Command Line Interface (CLI) and a Graphical User Interface (GUI)?

**Research Question**

What is the purpose of virtual machines? Some outside research will be required. The response should be at least a paragraph.