THE BASIC BUILDING BLOCKS OF CYBERSPACE: HARDWARE, SOFTWARE AND NETWORKS – MODIFIED FOR GEN CYBER RASPBERRY PI

HARDWARE AND SOFTWARE

INTRODUCTION TO LINUX LAB

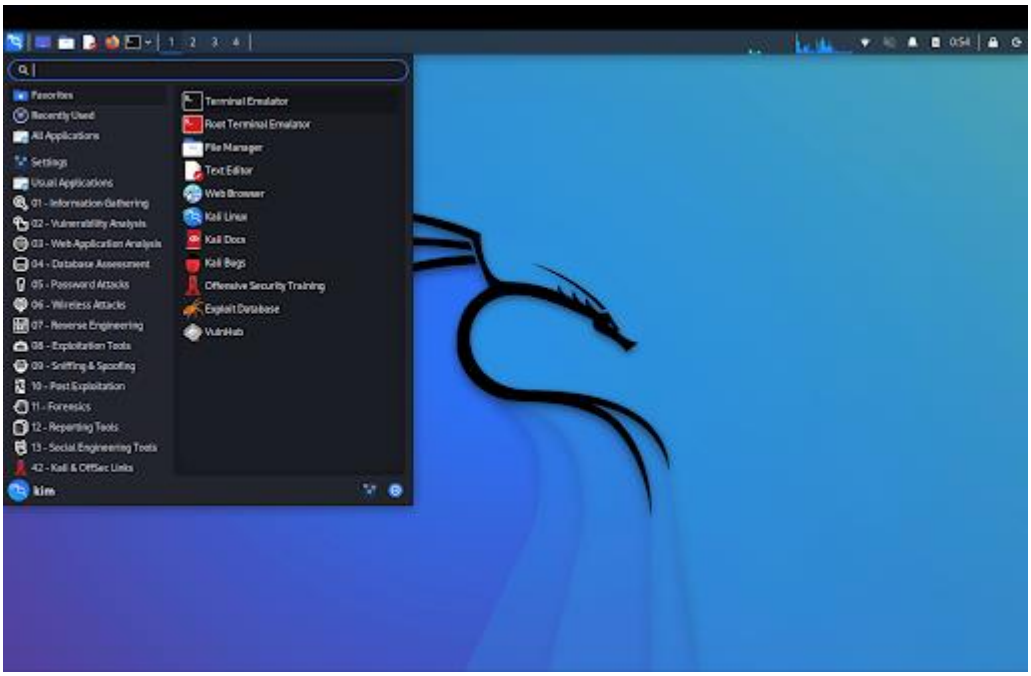
Lab Description: This lab provides a fundamental understanding of the Linux Terminal, Directories, Applications and other tools, Linux User Accounts, Groups and Permissions. There are 14 tasks that you need to complete and three sets of questions. Submit each set of questions before you move onto the next task.

Submission schedule:

- Submission 1 after Task 4
- Submission 2 after Task 8
- Submission 3 after Task 14 (the last task)

Lab Environment:

1) Plug in the mouse, power, and monitor to the Pi 400. Power up.
2) Log onto the Raspberry Pi
   Username: kali
   Password: kali
3) Once you login, you can launch a terminal window either by accessing it through the **Applications** menu at the top right and selecting **Terminal Emulator** or clicking the black terminal icon at the top of the screen. See the image on the next page.



TASK 1: UNDERSTANDING WHAT A DIRECTORY IS.

In your terminal, type "`pwd`" and press the Enter key (Return if you are using a Mac). "`pwd`" is short for "print working directory." This command displays information about the current working directory.

```
$ pwd
/home/kali
```

So, what does this mean? This directory is just like your usual folders and files. "kali" is a file that is located inside of the "home" file, and "home" is inside the file "/," which is a special file known as the root directory. The root directory is the highest file in the Linux filesystem and contains everything else. Please see Figure 1 below, the root directory illustration.

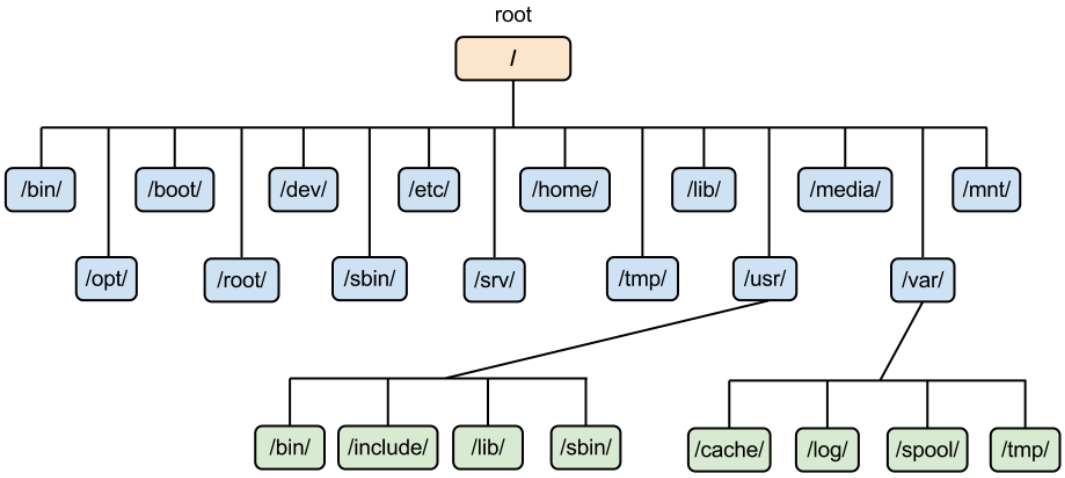


Figure 1: Root directory illustration
(Image source: https://freedompenguin.com/wp-content/uploads/2015/10/linux-filesystem.png)

Note: When you run the `pwd` command it will provide you with the absolute path. The absolute path is the full name of that given directory. This directory will be the full path from the root directory to that folder or file. In Windows this could look like `C:\Users\kali` or in MacOS 10 `/Users/kali`.

Alternatively, a directory can be represented as a relative directory. A relative directory is the path relative to the current working directory. If you are in the "/var/" directory in the picture above, the "/log/" directory has both a relative and absolute path.

Question: Based on the picture above, if `pwd` informs you that your current working directory is "/var/," then what is the relative path of "/log/?" What about its absolute path?ls

Absolute paths begin at the root and work down: `/var/log/`
The relative path is "relative" to your current directory, so just log/.
Command:

- `pwd` (print working directory): Outputs the directory that the terminal is currently in.

TASK 2: LISTING FOLDERS AND FILES CONTAINED WITHIN A DIRECTORY.

In your terminal, type "`ls`" and look at the output. "`ls`" is short for "list;" this command will list all the content of the current working directory.

```
$ ls
```

Most commands in the Linux Terminal have specifications that can be appended to the command called flags. Let's have a look at two very useful flags for the ls command.

```
$ ls -a
$ ls -l
```

As you can see, by running these two commands, your output differs from running `ls` alone. The first, `-a`, will list all files which are hidden. Files can be hidden in the Linux file system if the first character in its name is a period. You will also notice both "." and ".." is listed in your output; this will be explained in more detail later. For now, just understand that these are directories hidden in your home directory.

Your output for `-l` is quite different. This flag allows you to display a *long* listing with much more information about the content, such as the owner and permissions. For this lesson notice that the far left of the output includes a "`d`" which will designate a directory. Everything else is a file, not a directory.

Lastly to note, you can execute a combination of the above two flags into one command!

```
$ ls -la
```

OR

```
$ ls -l -a
```

LINUX TRICK: To view a large listing screen-by-screen, you can pipe another command using the vertical bar "`|`" symbol. Using "`ls -la /etc | more`", use the space to move to the next screen. These commands and output work well in a text-only terminal, which is a command line interface.

Question: Based on Figure 1 – the root directory illustration above, if you were to run the same command above on the "`/usr/`" directory, what are two folders you would expect in the output?

Name any two of the four directories: *`bin`, `include`, `lib`, `sbin`*

Commands:

- `ls`: List all content located in the current working directory.
- `|`: Pipe to add additional instructions or commands to a line
- `more`: allows for pagination (press "Enter" to advance by line, the space bar to advance by a page).

TASK 3: HOW TO CREATE AND CHANGE DIRECTORIES.

Now that you are familiar with printing your current working directory and listing its content, let's have a look at how to create directories and how to change your working directory.

In your terminal type the following commands:

```
$ cd /home/kali/
$ mkdir cyber
$ cd cyber
$ pwd
```

You will notice that your output "`/home/kali/cyber`" is an absolute path to the newly created directory.

The above exercise can be done another way: instead of "`cd cyber`" you could have specified "`cd /home/kali/cyber`" and it would all have worked the same. The former utilizes the relative path and the latter uses the absolute path. It is important you understand how absolute paths work, but as a rule of thumb relative paths exist for convenience.

LINUX TRICK: When typing a command or specifying a path, you can perform auto-completion! If you type the first few characters of a directory or file, you can auto complete it by pressing TAB. This assumes there is no ambiguity between two files: let's look at an example.

Suppose you execute the command `ls` and it lists the two files you see below.

```
$ ls
documents_one       documents_two
```

Then you type:

```
$ cd doc
```

Now from here, if you press TAB, what do you predict the terminal will do?

```
$ cd documents_
```

There is an ambiguity between these two files since they are named similarly. Linux cannot identify which file you wish to access; therefore, you must provide more to complete the command.

LINUX TRICK: `/home/kali/` is the home directory of the kali user. If you run "`cd`" with no specification (no arguments after), the directory will change to the default home directory.

Question to Ponder: In the "`/etc/`" directory, run the "`ls`" command. If you were a hacker, do you see any files that might be of interest to you?

Hint: One file of interest deals with passwords (and the file begins with a 'p').

Commands:

- `cd` (change directory): Change the current working directory to a specified directory.
- `mkdir` (make directory): Create a directory, using either the relative or absolute path.
- *tab* (auto completion of command or filename): Save on typing by using the tab to automatically complete a filename or command.

TASK 4: TRAVERSE TO THE PARENT DIRECTORY.

If your current working directory is "`/home/kali/personal/memelord/bestof2018/allthestuff/`" and you wish to go to the "`bestof2018`," how would you go about it? One way to do it is the following:

```
$ cd /home/kali/personal/memelord/bestof2018
```

That seems to be a lot of typing to move up a single directory level. Luckily enough, there is a way to refer to the directory above the current working directory (called the parent). We do this with the specification "`..`", which simply refers to the parent directory. See for yourself! At the terminal, type the following:

```
$ cd /home/kali/cyber
$ cd ..
$ pwd
```

Notice that you are now one directory level up (`in /home/kali`).

Now let's delete the `cyber` directory that you made earlier. First run a listing, delete the directory, and then confirm it has been removed:

```
$ ls
$ rmdir cyber
$ ls
```

Recall that you can specify the `mkdir` last command with a relative or absolute path, as you saw in Task 3. `rmdir` is similar because you can use the relative or absolute path.

```
$ rmdir cyber
```

OR

```
$ rmdir /home/kali/cyber
```

IMPORTANT NOTE: Any Linux command has an associated manual page you can read for more information in regards to the command, its use, and flags. You can do so with the "`man`" command.

```
$ man ls
$ man cd
$ man rmdir
```

Commands:

- `rmdir` (remove directory): Delete a specified directory given the relative or absolute path.
- `..` (up one directory level): A relative path that refers to one directory level up in the file structure.
- `man` (manual): shows help info about the command given

Submission 1

Submit your answers to these questions:

1. YOU ARE WORKING ON THE LINUX SYSTEM RUN THE FOLLOWING COMMANDS:

```
$ mkdir .new_stuff
$ ls
```

After running these commands, you should notice that you do not see your newly created directory in the output. Why?

2. LIST ALL OF THE SUBDIRECTORIES IN THE /HOME/ DIRECTORY.

3. CREATE A DIRECTORY IN THE /HOME/KALI/ FOLDER NAMED WHATEVER YOU WISH. NOW WHAT ABOUT DELETING THAT DIRECTORY? HOW MIGHT YOU DO THIS?

4. WHAT IS ANOTHER NAME FOR THE LINUX TERMINAL?

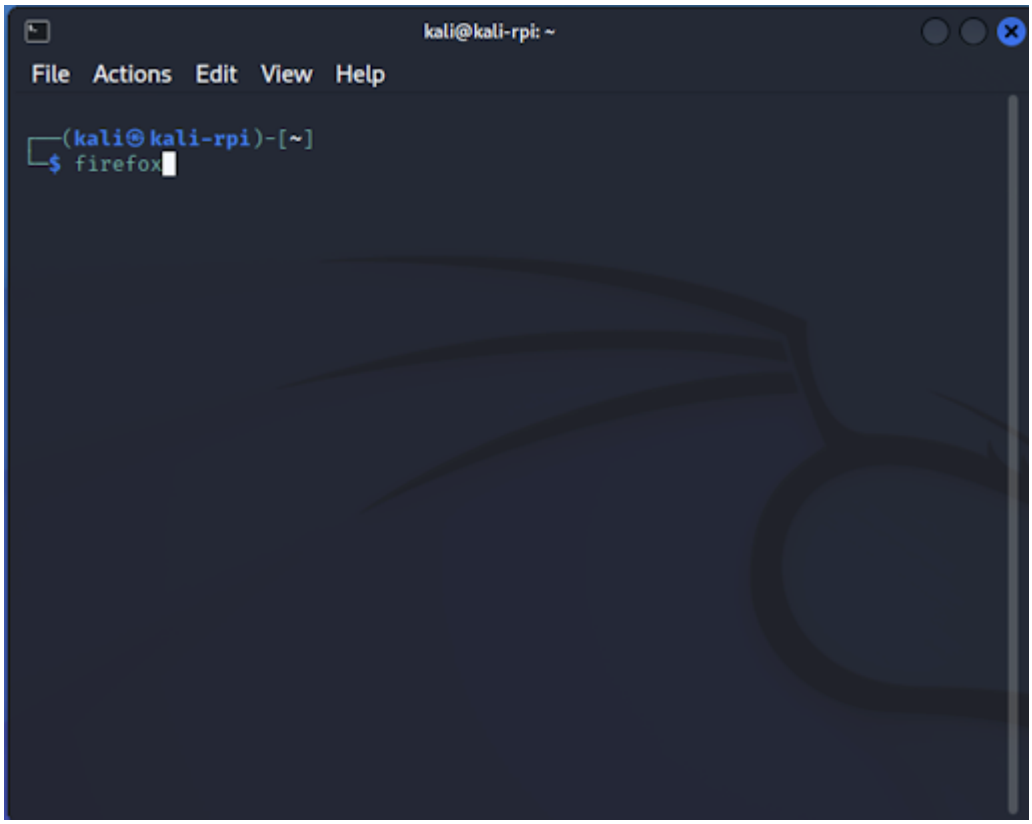
5. WHAT IS THE DIFFERENCE BETWEEN RELATIVE AND ABSOLUTE DIRECTORIES?

TASK 5: RUNNING FIREFOX

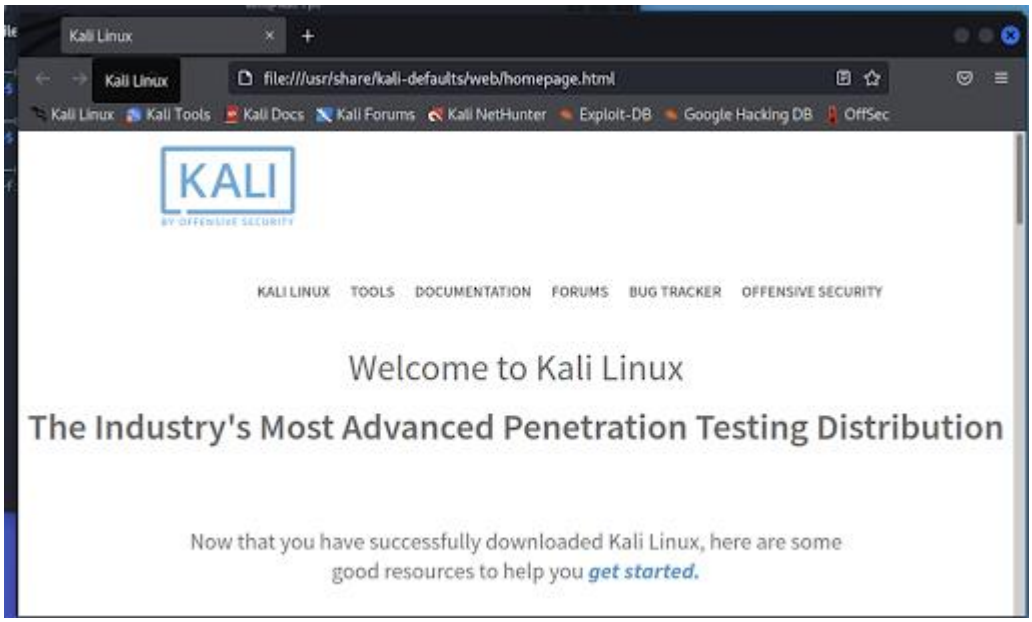
In the Linux Terminal, type the following command to launch the Firefox web browser:

```
$ firefox
```

Your terminal window should look like the following:



Once you run the command (by pressing Enter) it should present you with the following window:



Notice that the terminal runs the process and opens the web browser Firefox window. Now you can browse the internet freely.

You have gone from working in a command line interface (terminal) into a graphical user interface (GUI) with images and being able to use the mouse.

Some important things about running the Firefox program to note:

- If you close the window, the terminal will terminate (exit) the process associated with the Firefox application.
- While running the program, you are unable to run any other commands in the terminal because that session is occupied by Firefox.
- In the terminal window, you can close Firefox by terminating the process by pressing `Ctrl+c`.

LINUX TRICK: If any command is running (i.e. not terminated and you cannot see the usual prompt "kali@kali:[WORKING DIRECTORY]$") and you do not wish to let it finish, you can terminate it with `Ctrl+c`.

Commands:

- `firefox`: Start the Firefox web browser application. This runs as part of the graphical user interface (GUI) versus the command line interface.

TASK 6: OPENING APPLICATIONS AS A BACKGROUND PROCESS

Like many systems, Linux allows you to have programs and processes running in the foreground and in the background. That said, there is a way to start a program in the background so your terminal session can run more than that one process.

```
$ firefox &
```

OR

```
$ firefox&
```

Using the '`&`' (ampersand) character will allow the program to run in the background and allow your terminal session to persist and be usable. This is a convenient way to start applications and perform other tasks all from the terminal interface.

Commands:

- `&`: Placed after the process named, it allows that program to be run in the background and enable the terminal to remain interactive.

TASK 7: TEXT EDITING WITH MOUSEPAD

Kali Linux provides an easy-to-use text editor which allows you to open text files for editing. Let's have a look at an example. From the home directory, run the following commands:

```
$ touch mysecret
$ mousepad mysecret
```

In the window that pops up, type in "password is password" and then save and quit the mousepad window.

At the terminal:

```
$ cat mysecret
```

Your output should be the message from the file you created. What happened here is you used the "`touch`" command to create the file, "`mousepad`" to open the mousepad program to edit the file, and "`cat`" to print the contents of the file.

You can also use the "`more`" command to view the file's contents:

```
$ more mysecret
```

Commands:

- `Touch`: Allows you to update the timestamp of a file or create a new file, if a file with the name provided does not exist.
- `mousepad`: Opens the mousepad application for text editing.
- `cat`: Prints the content of a file to the terminal output.
- `more`: prints out the files to the terminal output, one screen at a time.

TASK 8: IMAGE VIEWER

Kali Linux comes with a pre-installed image viewer to view png files, jpeg files, etc. This image viewer application is called Ristretto.

This program is useful for viewing images. Let's download an image and view it with Ristretto.

First be sure that you are in your home directory.

```
$ cd
```

Then download an image by typing and executing the following command at the prompt:

```
$ wget https://vacr.io/R04fx
```

Launch Ristretto and open the file you just downloaded.

```
$ ristretto R04fx
```

NOTE: A tool was used to shorten the URL: `https://ih1.redbubble.net/image.183022618.2586/sticker,375x360.u10.png` to `https://vacr.io/R04fx`; otherwise, you would need to type the longer URL exactly as it appears after the `wget` command. You may still type the longer URL, but when you use the image viewer `ristretto`, you will need to type the full image name: `sticker,375x360.u10.png`.

You should see an image displayed in the Ristretto window. When you are done with the file, you may delete it by using the "`rm`" command.

```
$ rm R04fx
```

Confirm that you have deleted the file by using the "`ls`" command.

Commands:

- `wget`: Allows you to download files from the internet given a URL.
- `ristretto`: Opens the Ristretto Image Viewer application.
- `rm` (remove): Delete a file.

Submission 2

Submit your answers to these questions:

1. HOW DO YOU START A PROCESS IN THE BACKGROUND?

2. WHAT DOES THE `CAT` COMMAND DO?

3. WHAT IS THE DEFAULT IMAGE VIEWER IN KALI LINUX?

4. WHAT COMMAND WILL DOWNLOAD WEB CONTENT GIVEN A URL IN KALI LINUX?

TASK 9: ADDING A USER

To have additional user accounts, we will need to add a user with the "`adduser`" command. Let's create a user by typing the following:

```
$ sudo adduser john
```

[NOTE: This *may* require you to enter your account password to create the account, but should not in the Cyber Range environment. This is because we are using the `sudo` command. The `sudo` command means "super user do" which allows a user, with the proper permissions, to perform tasks at an elevated level of permission. This command is used to execute higher-privileged tasks and may require a password. If prompted for a password, enter the "kali" account password which is "kali."]

The program will now prompt you for a password for john, simply enter "john" as the password for the account. NOTE: Linux will not display the password or any other indication that you are typing anything. Don't be fooled, this is done for security reasons.

```
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
Changing the user information for john
Enter the new value, or press ENTER for the default
```

Once you get to this point, you will be prompted for the following:

```
        Full Name []:
        Room Number []:
        Work Phone []:
        Home Phone []:
        Other []:
Is the information correct? [Y/n]
```

You do not have to enter any information, simply hit enter until you get back to the command prompt.

You can check to see if a new user directory was created by running a directory listing of `/home`. You should see your kali home directory as well as the new `john` directory.

```
$ ls -l /home
```

Commands:

- `sudo` (super user do): A program to allow certain authorized users to perform sensitive actions.
- `adduser`: Preferred way to create a user account with a password and other information (uses `useradd`).
- `passwd`: Change or set the password of a user.

TASK 10: LOGIN AS DIFFERENT USER

Now for using the new account we just created. Type the following at the command prompt:

```
$ whoami
kali
$ sudo login john
*Enter password for john (john)*
$ whoami
```

Notice that `whoami` provides you with the name of the current logged in account. There are two ways you can logout of an account:

```
$ logout
OR
$ exit
```

NOTE: `exit` will close the login and then you are back to the original user. If you are not logged in as another user, it will close and exit the terminal window.

Commands:

- `whoami` (Who am I?): displays the username of current user when run.
- `login`: log into the system as another user
- `logout`: log out of a login shell
- `exit`: exit the shell where currently running

TASK 11: DELETE USER ACCOUNTS

When it comes to deleting an account, you can do so with the `userdel` command. Type the following:

```
$ sudo userdel john
```

The user should have been removed from the system after executing this command.

Commands:

- `userdel` (user delete): remove a user account from the system
  `userdel -r` (user delete with -r option): remove the user account and their files

TASK 12: UNDERSTANDING GROUPS

The Linux system places every user into groups. These groupings help make logical organization of the users in the system. This separates users for segregation of tasks and permissions. Let's create three accounts. We will use `adduser` and set the passwords equal to the user's name (Bob's password is just bob). Type the following to create the three accounts `bob`, `billy`, and `james` and the `chess_club` group:

```
$ sudo adduser bob
$ sudo adduser billy
$ sudo adduser james
$ sudo groupadd chess_club
$ sudo usermod -a -G chess_club bob
$ sudo usermod -a -G chess_club billy
$ groups billy
$ groups bob
$ groups james
```

You should have observed that `billy` and `bob` were both added to the `chess_club` group we created but `james` was not. This will allow certain content to be created to allow the `billy` and `bob` accounts to access the

`chess_club` data and prevent the account `james`, who is not a member, from viewing the data.

Commands:

- `groupadd`: Create a new group in the system.
- `usermod`: Make a change to a user account. Our example above uses flags to add a user to a specific group.
- `groups`: View the groups a user is a part of.

TASK 13: UNDERSTANDING USER PERMISSIONS

In this task we are going to explore a rather important concept: permissions. Permissions are a very vital part to any Linux systems administration and security. Let's revisit a command we have seen before to explore permissions. Run the following command in the `/home/kali` directory:

```
$ ls -l
OR
$ ls -l /home/kali
```

We have used this command before to distinguish between files and directories, but now we will learn more about the output. You will notice a string of text such as the following at the beginning of each `ls` line:

```
drwx-wx—x
d-wxrwxrwx
d—x—xrwx
```

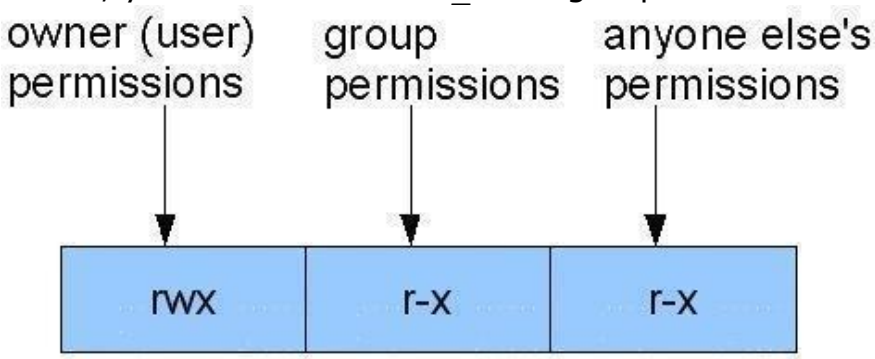
As we know, the d represents a directory, but what about the other text? Simply put…

r = read
w = write
x = execute

There are three groups of three, which are separate permissions for the three groups of users:

1. Owner
2. Groups
3. Other

Above, you created a `chess_club` group.



(Source: https://s11986.pcdn.co/wp-content/uploads/2013/03/linux-permissions.jpg)

Below, we will explore the three groups more thoroughly.

TASK 14: CHANGING PERMISSIONS

Lastly, we are going to take a look at how to change permissions for a specific file or folder. We can do this with the "`chmod`" command. At a command prompt, type the following:

```
$ touch new_file
$ ls -l new_file
$ chmod 722 new_file
$ ls -l new_file
```

To reiterate, the `touch` command will update the timestamp of an existing file or create a new file if a file of that name does not already exist.

You will notice that the permissions have changed. The owner can read, write and execute. Groups and others can only write. Now let's learn the secret behind the number notation of "722."

Each of the three permitted actions (read, write, and execute) are each denoted by their own number or value.

`r (read)` = 4
`w (write)` = 2
`x (execute)` = 1

So, this explains why we ended up with writing permissions for groups and others, but what about the owner having all three? Notice, 4+2+1 = 7; therefore, you are able to perform any combination to yield a certain permission set you wish. For example, if you wanted the owner to have read and write permissions, and give read permissions to groups and others, then r=4, w=2, 4+2=6. Now at the terminal, type the following command.

```
$ chmod 644 new_file
```

What permissions did you just grant the new file? Using 4, 2, and 1, we can check our math.

6 = 4 + 2 = `r + w`
and
4 = 4 = `r`

So, the owner of the new file has read and write privileges, and groups and others only have read permission.

NOTE: It is NEVER recommended that you perform "chmod 777" on ANYTHING. You should never grant permissions to everyone to do everything in a system!

Questions to Ponder: If another user has read, write and execute access (like in the `chmod 777` example) to your personal files, can that user overwrite (replace) or delete your files? *Yes, they can* ☹.

Think about important system files needed for the operating system to run. Would the 777 permissions mode allow you and other users to delete files that disable the operating system? Would that allow others to create or add malware files to the system? *Yes to both.*

Submission 3

Submit your answers to these questions:

1. WHAT DOES `SUDO` TRANSLATE TO? WHY IS IT REQUIRED FOR MOST OF THE COMMANDS WE ARE PERFORMING?

2. LET'S ASSUME YOU HAVE A FOLDER NAMED "STUFF" AND YOU WISH TO MAKE "STUFF" READABLE AND WRITABLE FOR THE OWNER, READABLE BY GROUPS AND OTHERS, HOW WOULD YOU DO THIS?

3. INTERPRET THESE PERMISSIONS: `DR-XRW--W-`

a. Owner:
b. Group:
c. Other: