The Linux Kernel:

Linux is used traditionally in 96% of the servers around the world because is **Faster** and **More** **secured**. First, Linux is not an OS, Linux is a Kernel, and we build OSs on top of that Kernel. Since Linux is a **Free**, **Distributed** and **Open-Source** Kernel, there are lots of flavors of Linux with different approaches and different functions but every app in every Linux Flavor or Linux Distribution can be used and installed in any other. The most Popular Linux Kernel Based OSs are Ubuntu, Debian, CentOS, **Parrot OS**, and **Kali Linux**, which are used mostly for hacking since they come with lots of hacking tools baked in.

What does the Linux kernel do?

The kernel’s job is to deal with the hardware resources and manage them, like the CPU Hard Drive RAM etc. let’s say for example we want to run nmap. Nmap will talk to the OS, Kali for example, and then Kali will talk to the Kernel, and the kernel will talk to the resources from the hardware and then provide it to the OS, which will provide for the nmap tool.

The Linux GUI and The Command Line:

Yes, most of the Linux distributions come with a GUI (Graphic User Interface) but where is the fun in that! Linux is mostly used from the command line, old school style. The reason why is that most of the time we use it for servers, servers which we control via SSH (Secure SHell) and we can only do this via the command line. Basically, we can do the same things in the Command Line that we do in the GUI, but not the other way around. That’s the other reason we use more the command line because it gives us more power over the device, more options, more functionalities.

**This is how we can Navigate using the Command Line**

New Commands:

**pwd**: This command **prints** the **working** **directory**, meaning that it tells you where you are, like in the GUI.

┌──(alice㉿Alice)-[~]

└─$ pwd

/home/alice

**ls**: This command prints a **list** with the files in the directory you are right now, meaning that you can see not just where you are, but also the file system, like in the GUI.

┌──(alice㉿Alice)-[~]

└─$ ls

Desktop Documents Downloads Music ngrok Pictures Public Templates Videos

**cd**: This command **changes** the working **directory**, like in the GUI when we are moving from one folder to the other.

┌──(alice㉿Alice)-[~]

└─$ cd Desktop/ >>>Changes the directory

┌──(alice㉿Alice)-[~]

└─$ pwd >>>Prints the new directory

/home/alice/Desktop

┌──(alice㉿Alice)-[~/Desktop]

└─$ ls >>>Shows the content of the working directory

Projects

The . and the .. folders: This are folder that are hidden in every directory, They refer to the current directory and the parent directory. This is something very common in all the OSs. So, if we want to come back, we reference

**cd ..**

If we keep using the cd.. command, at some point we will get to what we know as the Root directory, which looks like this:

┌──(alice㉿Alice)-[/home]

└─$ cd ..

┌──(alice㉿Alice)-[/]

└─$

**Using Tab:**

We can use the tab key to autocomplete commands and directories, this way we don’t always have to print a ls in order to know the files in the directory, we can simply use tab repeatedly and go through all the folders and files till we reach the one we are looking for.

New Command:

**whoami**: This is a fun command we can use to verify **who is the logged in user**

┌──(alice㉿Alice)-[/]

└─$ whoami

alice

The File System:

We already saw how to navigate in the File system, we covered the navigation commands like going back, forward, print the current directory, and print all the files in the current directory. We also covered that if we use the command “cd ..” enough times, we can get to what we call the root of the File System, but what is there? Is it Important?

All is a file:

Everything, literally everything in Linux is a file. We can go from the network configuration, going through the Devices connected like Hard drive, and till the command that we use in the command line, they are also files. These commands are the binary representation in a file of what we want the computer to do, and they live in a place called bin (stands for binary) inside the root directory.

The Root:

In this directory we can find a lot of important things we are going to discover over the time. For now, let’s focus on get to know the directories inside “/” and let’s check the binaries.

**Important to mention that this is my Kali Distribution**

┌──(alice㉿Alice)-[/]

└─$ ls

bin etc initrd.img.old lib64 media proc sbin tmp vmlinuz

boot home lib libx32 mnt root srv usr vmlinuz.old

dev initrd.img lib32 lost+found opt run sys var

The Command Binaries:

We can jump into the bin directory and find out that in effect, all the binaries of all the commands, live there:

┌──(alice㉿Alice)-[/bin]

└─$ ls

( . . . )

mp3-decoder zmore

mp64 znew

mpg123 zsh

mpg123-alsa zsh5

mpg123.bin

┌──(alice㉿Alice)-[/bin]

└─$

Since ls is a file, we should be able to look into it and see what’s going on inside. For that we are going to use the command **cat.**

New Commands:

**cat**: This command **concatenates the strings in a file**. Meaning that can parse it and print its content on the command prompt.

┌──(alice㉿Alice)-[/bin]

└─$ cat ls

ELF>�a@h7@8

( . . . )

����$&�� 4��>o

��

▒F���N���o~~[���o��j ▒tB`+`+� ▒~@@y @ @���F���F�F�<����� ��� O �

�

�D �P�P�h�P�X�`3`#x

64H6

┌──(alice㉿Alice)-[/bin]

└─$

What we see here is what we call the binaries of a file. This can be also found in a video, in an image, song or any other file. And they differ from one another because the program that’s used to interpretate each one of them does it differently.

**clear**: This command **clears the command line**, leaving just a line so we can see things more “clear”

**locate**: This command **locates a file** in the whole system

**find**: This command **finds a file or directory** in designated directory.

**grep**: This command **reads text file** and **returns the lines** that contain word we are looking for

grep [word] [file]

Managing Files:

Managing files, as we do in the GUI includes copying, deleting, creating, moving, renaming, and modifying them. We’ll see now what the commands are we use for that.

New Commands:

**cp**: this command allows us to **create a new file with the content from another file**:

cp [source file name or directory] [final file name or directory]

**rm**: this command stands for remove. It **removes a file** from a directory.

rm [file name]

**rmdir**: this command stands for remove. It **removes a directory**.

rmdir [directory]

**mv**: This command is used for **renaming**, and for **moving** **files**. This is because what we are doing is changing the file’s directory in both cases

mv [source file name or directory] [final file name or directory]

**touch**: touch is a command that allow us to **create files**

touch [filename]

**nano**: Nano is a **text editor**, this command is used for both tasks, create new files and mod modify existing ones. If the file exists it opens the file and we can see its content like in cat, but this time we can modify it. If the filename in the directory doesn’t exist, the file is created.

nano [filename]

The Super Binaries:

Inside the root we have another directory called sbin, this stands for super binaries. This are commands that are reserved for administration purposes only. I’m talking about the functions that an admin has to deal with like add a new user, delete users

If we go in our root directory, we’ll notice that in it, we have a user directory, which also contains a bin, and sbin directories. Basically, we have the same commands in the /user/bin directory than we have in /bin, But we have some more in the /bin directory that are not included in the /usr/bin directory.

The Home Directory:

The home directory is located in **/home**, in this directory are the directories corresponding to all the users that exist in your system. If we ls it, we can see that I for example have only one user:

┌──(alice㉿Alice)-[/home]

└─$ ls

alice

If we cd in our user, we will get something like this in the directory **[~]**, this means that you are inside a user’s directory. And if we cd into it, and the ls it:

┌──(alice㉿Alice)-[/home]

└─$ cd alice

┌──(alice㉿Alice)-[~]

└─$ ls

Desktop Documents Downloads Music ngrok Pictures Public Templates Videos

The /root Directory:

Inside “/” we have a root folder that actually contains a user named root. This user is the ine that has total control over the system, even us if we want to do most of the tasks, we must ask for permission using a command named “**sudo**” which means **superuser do**.

The /dev Directory:

As we said before, even the devices are represented as files. This is the directory containing such files. Particularly, the **sda** file is the one representing our hard drive.

The /etc Directory:

Here there are lots of stuff and configuration. As we said before, the configuration that we are using right now, is also a file, and this file lives here. More specifically the network configuration file lives in the network directory.

The Terminal:

There is a technical disruptive here when we refer to the terminal. The terminal, as we know is something physic, like a smartphone, a smartphone is a terminal, right? So, this is something that we got because at the beginning we got used to call the back screen with white letters, the terminal. What we really see in a Computer nowadays, it is really called the **Shell**. The terminals then, are the physical parts pf the device that can interact with us by receiving and getting information, like the mouse, the keyboard, and the screen.

The Terminal, meaning the whole compound, The Computer, Interacts with the shell, and the shell with the OS, and the OS with the Kernel, and the Kernel with the resources.

The Shell (UI):

The shell is the user interface that we use to interact with the OS. The terminal of the Terminal Emulator is what we use to interact with the shell. There are several kinds of shells like for example bash from Linux, or pwsh (power shell) from Microsoft.

Bash(Born Again Shell):

This is the most common shell we’ll see and use while working with Linux. Bash is also a programing language by its own and is pretty extensive in content. Bash is crucial while working with Linux and is super important to know how to use it.

There are way too many commands for us to remember all of them. This is how we get:

HELP

New Commands:

**man:** This command stands for **manual** and is like a manual of each command we use and it helps you by teaching you how is supposed to be used.

man [command]

**-h or --help:** This is not a command, is more of an option that most of commands come with. Stands for **help** and is like a manual of each command we use, and it helps you by teaching you how is supposed to be used.

[command] -h

[command] –help

**apropos:** This command helps you **find** **keywords** in the documentation of a command, or in the command name itself. This command is very useful when we know what we want to do, but we have no idea of what command can we use for it.

SUDO (Super User DO ):

Sudo, or Super user do, is a command that allows you to impersonate the root user for just one command, it gives to a user with administrative rights the ability to use certain commands that normally he is not supposed to. We use sudo as the first word in a command and then the command we want to execute.

┌──(alice㉿Alice)-[/etc]

└─$ sudo su -

[sudo] password for alice:

┌──(root㉿Alice)-[~]

└─#

Sudoers file:

This is a file that contains the information about the permissions. It is a better practice to just use the command **visudo** to edit this file instead of nano. Here we’ll find users with root privilege, loke root, of course, and a group called sudo, which contains all the users that are allowed to use the sudo command. If we want to give to a user root privilege, we only need to copy the root line, paste it bellow and then change the name to the username we want.

If we wanted to add a whole group, we only have to add a “%” in front of the group to specify that’s in fact, a group and not a user.

New Command:

**visudo:** this command **opens the Sudoers file** the nano text editor in a temporary file so is safer in case something breaks.

# User privilege specification

root ALL=(ALL:ALL) ALL

# Allow members of group sudo to execute any command

%sudo ALL=(ALL:ALL) NOPASSWD:ALL

Users:

Every computer system has users. One of the main responsibilities as a Linux OS administrator is to administrate users. That’s why this is a very valuable skill to learn.

As we mentioned before, not all users are the same, or have the same rights. For example, let’s remember that in all Linux based OSs we have a user named **root** who is the super user, not the admin, but the super user. Once you’re logged is as root you don’t need to use sudo for executing **any** command. Other important thing to point out here, is that if we are logged in as a user who doesn’t have administrative rights, this user won’t be able to use the sudo command to impersonate root in any command.

Creating a User:

Whenever we create a new user, we create both a user and a group containing that user. We can also get access to a file with the **cat** command that contains all the information of all the users that are currently in the system. This file is located in **/etc/passw** and contains some important data like the username, UID (User ID), GID (Group ID) , and all the other fields that we have set before.

alice:x:1000:1000:Alice Sintesys Thirty,,,:/home/alice:/usr/bin/zsh

username psw UID The rest of the data home location default shell (zsh or bash)

Here we can see that we have also a passwd field with a letter “X” in it. What this means is that the password, or the password’s hash, is stored somewhere else, more specifically in a file named **shadow**, also located in **/etc**.

Groups:

The same way as we have a file that contains all the user’s information, we also have one that contains all the group’s information. This file is also located in **/ect** and it’s called **group**.

New Commands:

USERS

**adduser:** this command **adds a new user**

adduser [username];

[password]

[Full Name]

Other fields:[Room Number], [Work Phone], [Home Phone], [Other]

Is the information correct? [Y/n]

**useradd:** this command **adds a new user**, but it doesn’t manage the home directory, or password or any of the other fields so we would have to manage those manually

useradd [username]

**su:** this command allows us to **switch** **users**. We can use it to log is as someone else.

su [username];

[password]

**userdel:** this command **deletes** **a** **user**

userdel [username]

**passwd:** this command **changes** or **adds** a **password** to a user

passwd [username];

[password]

**usermod:** this command **changes** or **adds** the characteristics of a **User**

usermod –-help to see all the modifiable fields

usermod [username] –[a…z]

New Commands:

Groups

**groupadd:** this command **creates a new group**

**groups:** this command prints on the screen all the matching results for all the **groups** the logged in **user belongs**.

**usermod:** this command can also be used for **adding members to a group**

usermod –aG [group name] [username]

**dpasswd:** this command, along with “-d” can be used for **deleting members from a group**

dpasswd –d [username] [group name]

**groupdel:** this command is used to **delete a group**

groupdel [group name]