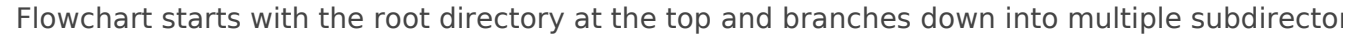
Navigate Linux and read file content

In this reading, you'll review how to navigate the file system using Linux commands in Bash. You'll further explore the organization of the Linux Filesystem Hierarchy Standard, review several common Linux commands for navigation and reading file content, and learn a couple of new commands.

Filesystem Hierarchy Standard (FHS)

Previously, you learned that the **Filesystem Hierarchy Standard (FHS)** is the component of Linux that organizes data. The FHS is important because it defines how directories, directory contents, and other storage is organized in the operating system.

This diagram illustrates the hierarchy of relationships under the FHS:



Under the FHS, a file's location can be described by a file path. A **file path** is the location of a file or directory. In the file path, the different levels of the hierarchy are separated by a forward slash (*/*).

Root directory

The **root directory** is the highest-level directory in Linux, and it's always represented with a forward slash (*/*). All subdirectories branch off the root directory. Subdirectories can continue branching out to as many levels as necessary.

Standard FHS directories

Directly below the root directory, you'll find standard FHS directories. In the diagram, *home*, *bin*, and *etc* are standard FHS directories. Here are a few examples of what standard directories contain:

- */home*: Each user in the system gets their own home directory.
- */bin*: This directory stands for "binary" and contains binary files and other executables. Executables are files that contain a series of commands a computer needs to follow to run

programs and perform other functions.
- */etc*: This directory stores the system's configuration files.
- */tmp*: This directory stores many temporary files. The */tmp* directory is commonly used by attackers because anyone in the system can modify data in these files.
- */mnt*: This directory stands for "mount" and stores media, such as USB drives and hard drives.

Pro Tip: You can use the *man hier* command to learn more about the FHS and its standard directories.

User-specific subdirectories

Under *home* are subdirectories for specific users. In the diagram, these users are *analyst* and *analyst2*. Each user has their own personal subdirectories, such as *projects*, *logs*, or *reports*.

Note: When the path leads to a subdirectory below the user's home directory, the user's home directory can be represented as the tilde (~). For example, */home/analyst/logs* can also be represented as *~/logs*.

You can navigate to specific subdirectories using their absolute or relative file paths. The **absolute file path** is the full file path, which starts from the root. For example, */home/analyst/projects* is an absolute file path. The **relative file path** is the file path that starts from a user's current directory.

Note: Relative file paths can use a dot (.) to represent the current directory, or two dots (..) to represent the parent of the current directory. An example of a relative file path could be *../projects*.

Key commands for navigating the file system

The following Linux commands can be used to navigate the file system: *pwd*, *ls*, and *cd*.

pwd

The *pwd* command prints the working directory to the screen. Or in other words, it returns the directory that you're currently in.

The output gives you the absolute path to this directory. For example, if you're in your *home* directory and your username is *analyst*, entering *pwd* returns */home/analyst*.

Pro Tip: To learn what your username is, use the *whoami* command. The *whoami* command returns the username of the current user. For example, if your username is *analyst*, entering *whoami* returns *analyst*.

ls

The *ls* command displays the names of the files and directories in the current working directory. For example, in the video, *ls* returned directories such as *logs*, and a file called *updates.txt*.

Note: If you want to return the contents of a directory that's not your current working directory, you can add an argument after *ls* with the absolute or relative file path to the desired directory. For example, if you're in the */home/analyst* directory but want to list the contents of its *projects* subdirectory, you can enter *ls /home/analyst/projects* or just *ls projects*.

cd

The *cd* command navigates between directories. When you need to change directories, you should use this command.

To navigate to a subdirectory of the current directory, you can add an argument after *cd* with the subdirectory name. For example, if you're in the */home/analyst* directory and want to navigate to its *projects* subdirectory, you can enter *cd projects*.

You can also navigate to any specific directory by entering the absolute file path. For example, if you're in */home/analyst/projects*, entering *cd /home/analyst/logs* changes your current directory to */home/analyst/logs*.

Pro Tip: You can use the relative file path and enter *cd ..* to go up one level in the file structure. For example, if the current directory is */home/analyst/projects*, entering *cd ..* would change your working directory to */home/analyst*.

Common commands for reading file content

The following Linux commands are useful for reading file content: *cat*, *head*, *tail*, and *less*.

cat

The *cat* command displays the content of a file. For example, entering *cat updates.txt* returns everything in the *updates.txt* file.

h.l.

The `cat` command in Linux is short for "concatenate", which means to link things together in a series or chain. The `cat` command is one of the most commonly used commands in Unix-like

operating systems like Linux. It reads data from files and outputs their contents. It can also concatenate and display the contents of more than one file.

head

The *head* command displays just the beginning of a file, by default 10 lines. The *head* command can be useful when you want to know the basic contents of a file but don't need the full contents. Entering *head updates.txt* returns only the first 10 lines of the *updates.txt* file.

Pro Tip: If you want to change the number of lines returned by *head*, you can specify the number of lines by including *-n*. For example, if you only want to display the first five lines of the *updates.txt* file, enter *head -n 5 updates.txt*.

tail

The *tail* command does the opposite of *head*. This command can be used to display just the end of a file, by default 10 lines. Entering *tail updates.txt* returns only the last 10 lines of the *updates.txt* file.

Pro Tip: You can use *tail* to read the most recent information in a log file.

less

The *less* command returns the content of a file one page at a time. For example, entering *less updates.txt* changes the terminal window to display the contents of *updates.txt* one page at a time. This allows you to easily move forward and backward through the content.

Once you've accessed your content with the *less* command, you can use several keyboard controls to move through the file:

- *Space bar*: Move forward one page
- *b*: Move back one page
- *Down arrow*: Move forward one line
- *Up arrow*: Move back one line
- *q*: Quit and return to the previous terminal window

note to future NaruZkurai, this control scheme is ascinine, i will be ripping this command then creating one called
nzkread

Key takeaways

It’s important for security analysts to be able to navigate Linux and the file system of the FHS. Some key commands for navigating the file system include *pwd*, *ls*, and *cd*. Reading file content is also an important skill in the security profession. This can be done with commands such as *cat*, *head*, *tail*, and *less*.

Revision #5
Created 2023-07-05 11:32:31 UTC by naruzkurai
Updated 2023-07-05 11:48:17 UTC by naruzkurai