Hello everyone! In this video, you will learn about file permissions in Linux OS.

Each file in Linux has some permissions associated with reading, writing, and executing for the owners, groups, and other users of the system. When you execute an “ls -l” command,” for the “long listing” option, each file will be listed on a separate line in long format, as shown here.

Let’s understand the output of ls –l command.
The first character of the output will almost always be either a ‘- ‘, representing a file, or a ‘d’, which means it’s a directory.
The next nine characters (rw-r–r–) show the permissions; we’ll talk about them later.
The next column shows the owner of the file. In this case userID is “cdac”.
The next column shows the group owner of the file. In this case “cdac” group have special access to these files.
The next column shows the size of the file in bytes.
And then the next column shows the date and time the file was last modified.
And, of course, the final column gives the filename.

Let's understand the security permissions now. The files can have r, w and x permissions, associated with them where 'r' means "reading" the file's contents. 'w' means "writing" or modifying the file's contents and x means "executing" the file. This execute permission is given only if the file is a program.
If any of the "r w x" characters are replaced by a '- ', in the output, it means that permission has been revoked.

The first set of permissions in output represent the user permissions which are applicable only to the owner of the file or directory, they will not impact the actions of other users.
The second set represents the group permissions which apply only to the group that has been assigned to the file or directory, they will not affect the actions of other users.
The third set represents other permissions which apply to all other users on the system, this is the permission group that you want to watch the most as ethical hacker.

You can see that the user's permissions for some files is "rw-" as the first three characters. This means that the owner of the file ("cdac here") can "read" it and "write" it (modify its contents) but he cannot execute it because it is not a program; it is a text file.

The second set “r--” of characters means that the members of the group “cdac” can only read the files.
The final three characters show the permissions allowed to anyone who has a UserID on this Linux system. This shows that permission (“r– “). Allows anyone in Linux machine to read, but they cannot modify the contents of the files or execute it.

The security permissions of the files can be modified also in Linux OS. command you use to change the security permissions on files is called “chmod”, which stands for “change mode”, because the nine security characters are collectively called the security “mode” of the file.
The first argument you give to the “chmod” command is ‘u’, ‘g’, or ‘o’. Representing user, group and others respectively. you can also use a combination of them (u, g, o). After specifying u, g or O you can use +, - or = for adding, removing or assigning permissions.
In the snapshot, the permissions for file5.txt has been changed for other users.

Rather than using r, w, x notations, Linux also allows to use octal numbers to represent the permissions. In the table you can see that. Different permissions have different numbers associated with them such as to provide full

permissions of read, write and execute for any file the octal number 7 may be used. In the figure two different instances of chmod commands are used using r, w and x characters and octal numbers respectively.
Both of them provides full read write and execute permission (code=7) to all the group.

Thank You…