8.11.1 Special Permissions

Click one of the buttons to take you to that part of the video.

Special Permissions 0:00-0:44

In this lesson we're going to discuss special permissions.

Now, honestly, most of the time when you're working with permissions on a Linux system, you're going to be working with the standard read, write, and execute permissions. But be aware that there are other special permissions that are available that you can assign to a file or directory in the Linux file system if you need to.

Be aware that you probably won't actually use these permissions very often. They're designed for very specific purposes. Because of that, if you use them incorrectly, they can actually represent a grave security risk to your system, so make sure you know what you're doing before you go about setting them.

User ID Permission 0:45-1:27

These special permissions are listed here. The first one we want to look at is SUID, which stands for set user ID. The set user ID permission can be applied only to a binary executable file in the file system. It can't be applied to a directory. It can't be applied to a word processing file. And it cannot be applied to a shell script.

If you do assign the set user ID permission to an executable file, and then a user runs that file, then the user who ran the file temporarily becomes that file's owner. As we said, you can't actually apply this permission to a directory.

Group ID Permission 1:28-2:34

There's another special permission you need to be familiar with. The SGID permission, which stands for set group ID. Just like the set user ID permission, the set group ID permission is applied to binary executable files or directories in the file system.

If you assign the set group ID permission to an executable file and then a user runs that executable file, the user who ran the file temporarily becomes a member of the file's owning group.

Unlike set user ID, you can assign the set group ID permission to a directory and when you do something special happens. If a user then creates a file in that directory that has the set group ID permission set, then the file's owner is set to the user's account, which is the normal behavior.

The owning group assigned to the new file is not set to the user's primary group, but is instead set to the owning group of the parent directory.

Sticky bit 2:35-3:12

The last special permission we want to look at here is sticky bit. Sticky bit is assigned only to directories. It is not assigned to files. When the sticky bit permission is assigned to a directory, users can delete only files within the directory for which they are the owner of the file or the parent directory itself.

This is very important because it basically negates the effect of having the write permission to a directory. If the user has the write permission to a directory, then that user could potentially delete files in that directory that they don't actually own. If you want to prevent that from happening, then assign the sticky bit permission to that directory.

Numeric Values 3:13-7:06

Be aware that these special permissions are referenced as an extra digit that gets added to the beginning of the file or directory's mode. Just like with regular permissions, each of these special permissions also has a numeric value assigned to it. Set user ID has a numeric value of 4, set group ID has a numeric value of 2, while sticky bit has a value of 1.

For example, let's suppose we want add the SUID, set user ID, and the set group ID permissions to a file in my home directory. It's an executable file named runme.

In addition to running these permissions, I also want my file's owner to have read and execute permissions to the file, so it gets a value of 5. I want group to have read and execute permissions to the file, so it gets a 5, and I want other authenticated users just to have read permission to the file, so they get a 4--they won't be allowed to run it.

Remember that SUID has a numeric value of 4, and SGID has a numeric value of 2. Therefore, to assign both set user ID and set group ID to the same file, I would combine these two to get a value of 6 and specify it as the first number in the mode. In this case, I would enter chmod 6554 runme at the shell prompt.

Because I assigned these permissions to the runme file, the user who ends up running this file will temporarily become the runme file's owner, and likewise, because we set the set group ID permission on it, that user will also become temporarily a member of that file's owning group.

Notice in the output of the command we have an s now where we should have an x for both owner and for group. If you see s for owner, you know that we have the set user ID permission set. If you see s for group, you know that you have the set group ID permission set.

Let's look at another example. Suppose we granted owner and group read, write, and execute permissions to the RandD directory that you see right here. We would use 7 for owner and 7 for group. Read is 4, write is 2, execute is 1, so we have a sum of 7 for both owner and for group. If we do this, then having the write permission to the directory allows anyone in the owning group to go ahead and delete any file that they want to in that directory.

We don't want this to happen, so we want to reconfigure the directory so that the users in the owning group can still create files in the directory, but we want to block them from deleting files that they don't actually own. To do this we assign the sticky bit permission to the RandD directory.

Remember, sticky bit has a value of 1, so the first number in the mode is a 1 followed by read, write, and execute for user; and read, write, and execute for group. And we don't want to let others have any access to the directory at all.

It's a Research and Development directory. We don't want prying eyes in it, so we're going to actually assign others 0. We'll take away all permissions for others to the RandD directory.

As a result, you'll notice that the mode now has read, write, and execute for user; read, write, and execute for group; and no permissions for others. And we have a T at the end of the mode. When you see a T as the last digit of the mode, it indicates that the sticky bit has been set.

Summary 7:07-7:22

That's it for this lesson. In this lesson we talked about special permissions and how they're used in the Linux file system. We looked at the set user ID permission. The set group ID permission. We looked at the sticky bit permission, and then we ended this lesson by talking about how you assign these permissions to files and directories using the chmod command.
