

7.1.2 Linux Group Overview

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

Linux Group Overview 0:00-0:16

Like most other operating systems, Linux uses the concept of user groups. This is a really good thing, because groups make our lives as system administrators much easier. To understand why, let's take a look at a scenario.

Benefits of Groups 0:17-1:33

Let's suppose that we have seven different user accounts shown here on a Linux system. Now, of these seven users, five of them need basically the same level of access to a particular directory in the Linux file system. In this case it's `/opt/app`. That's not a real directory. It's hypothetical, but let's just suppose that it were there.

Now, if we did not have the concept of user groups, in order to grant these five users access to this particular folder, we would have to go into and assign each individual user the same permissions. In this case we're going to assign each user read, write, and execute permissions to this folder. Basically this means I would have to do the exact same task five different times.

Now, that may not sound too bad. It's just five users, right? Just suppose you had 30 users or 100 users or maybe 1,000 users that all needed the same level of access to this same directory. Assigning permissions one user at a time would be a waste of resources, and it would frankly take forever, and you wouldn't ever want to be a system administrator for the rest of your life.

Users 1:33-3:56

A better way is to implement groups on your Linux system. With groups, you assign permissions to just the group itself. In this case, we've created a group named `Users`. We still have that same directory that we had before, `/opt/app`, whatever happens to be installed in there.

We assigned read, write, and execute permissions to `/opt/app` to the group named `Users`. Then we took every single user up here that needed the same level of access to that directory, and we simply made them a member of the `Users` group. That is a lot easier.

I had to make only one set of permissions assignments. I didn't have to do it five separate times, or 20 different times, or 100 different times, whatever the situation may be. Just once.

Now the other nice thing about this is that it makes things very, very scalable and very, very flexible. Let's suppose for some reason these users no longer required the write permission for this directory. If I had used the old model, I would have had to go through every single user account and manually remove the write permission.

That would A) be a lot of work, and B) it would be prone to error because it's very likely that I'm going to forget somebody, especially if it's been two or three years since I assigned them the write permission to that directory in the first place, which makes for potential security problems.

Instead, using groups, I simply remove the write permission from the `Users` group, and then every single user that is a member of that group automatically receives the change. In addition, it's also really easy to alter the group membership. Let's say that for whatever reason, this user just got a different position; they no longer need access to the `/opt/app` directory. We simply remove them from membership in the `Users` group.

Let's likewise say that this user took over that other user's position in the company and they need access to that directory now, `/opt/app`. So all we have to do is make that user a member of the `Users` group, and then they automatically receive the permissions that are assigned to the `Users` group.

When this user leaves the group, they automatically lose the permissions that are assigned to the `Users` group. This is just a much easier way to administer things.

`/etc/group` 3:55-6:02

If your Linux system has been configured to use local authentication, meaning that you're using files in the `/etc` directory to store your user accounts and passwords, then your group accounts are stored in the `/etc/group` file. As with the password and shadow files, each line in the group file is a single record that represents one particular group.

Each line is a record really that's composed of four fields. First, we have the group name field, which simply identifies the name of the group. In this example, the name of the group is RandD. The next field identifies whether or not a password has been assigned to the group.

Notice here that we have an x. The x simply means that we're not storing passwords in the group file. And just as with the password and shadow files, we don't store the passwords for user accounts in the password file; we store them in the shadow file.

Same holds true for groups. We don't store group passwords in the group file; we store them in the gshadow file. The x tells the system that the passwords aren't here; they're in gshadow instead.

The next field specifies the group ID number, or GID, of the group. In this example, the GID of the RandD group is 1002. As with user accounts, standard group accounts are numbered with a number greater than 1000 on some distributions, and on others it might be greater than 500. You have to look at the individual distribution to see which case it is.

Just as there are system user accounts, there are also system group accounts as well. Just as with user accounts, system group accounts are numbered with lower numbers. Now again, it depends upon the distribution. Usually they're numbered with a number less than either 500 or 1000. Again, you need to look at your particular distribution and see what numbering scheme it uses.

Then the last field specifies which user accounts on the system are members of this group. In this case, you can see that the rtracy and ksanders users are members of the RandD group.

/etc/gshadow 6:01-6:47

Now as we said a minute ago, you can assign passwords to groups if you want to. We don't store them in the group file; we instead store them in the gshadow file. Now, as with the shadow file for user accounts, the gshadow file uses one line to represent a record for one single group.

We have the group name right here. We have the encrypted password for that group right here. There's an optional field here called Group Admins; I don't ever use that. Then we finally have a list of group members over here. To be honest, I don't actually use the gshadow file very much. I typically don't assign passwords to groups, but do be aware that it is an option to use if you want to.

Summary 6:48-6:58

That's it for this lesson. In this lesson we reviewed Linux groups. We discussed the benefits of using groups. We discussed how group accounts are stored on a Linux system. Then we discussed the roles of the group and gshadow files in the /etc directory.

Copyright © 2022 TestOut Corporation All rights reserved.