7.1.3 User and Group File Review

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

User and Group File Review 0:00-0:07

In this demonstration we're going to look at Linux user and group accounts.

/etc Directory List 0:08-0:30

First I need to switch to my root user account with the 'su -' command. The user and group files are stored in the /etc directory. I'll press Enter here and then use the 'ls' command to list all the files here. There are four key files that you need to be familiar with where user and group account information is stored.

---

passwd File 0:31-3:28

The first one is the passwd file. The passwd file stores our Linux user accounts. What it does not store, however, are the passwords for those accounts, which is kind of confusing to a lot of new Linux administrators because the name of the file would imply that there are passwords in there, but there are not.

Actually, the reason it's named passwd is because back in the old days of UNIX (many, many, many, many years ago before the Earth was round, it was still flat), we did actually store user passwords in the passwd file, hence the name passwd. Today, that's not the case.

For security reasons, we actually take the passwords of user accounts out of this file and store them in a second file called shadow, where they're stored in hashed format to protect their security. User accounts are stored in passwd; user account passwords are stored in the shadow file.

Your Linux group accounts are stored in the group file, and if you've assigned any passwords to those group accounts, they are stored in the gshadow file. Let's take a look at each one of these in a little more detail.

Let's use the 'cat' command to view the contents of the passwd file. Notice within the passwd file that each user account is contained on a single line. Basically, the passwd file is kind of like a database of user accounts, and each line within that file is a separate user account record.

Each record is then divided up into several fields. The fields are separated by colons within the single line. The first field is my username line. This is just the username that I supply when I'm logging in to the system. My username is rtracy.

The second field is the password field, and as we said earlier, we don't actually put passwords in the passwd file anymore. We just put an x in the password field, which tells the Linux kernel that we're using the shadow file for passwords and we're not keeping them here in the passwd file.

The next field is my user ID number that's been assigned to my user account. Notice that I have a user ID number of 1000. On a Linux system, the first standard user you create on the system will be assigned a user ID number of 1000, the second one will be 1001, the third one is 1002, and so on.

This numbering system is very important, and we'll talk about it more in just a second. Just remember that the user ID that is assigned to standard user accounts, that users can log in to the system with, start at 1000 and go up.

This field here is the group ID number of the default group assigned to my user account. My user account can be a member of multiple groups at the same time. But in Linux, every user account has one group that is the default group, and that is identified in this field right here, with its group ID number.

The next field is the description field, which we use to store the user's full name. The next field identifies the user's /home directory. My /home directory is /home/rtracy. Then the last field identifies which shell I want to use by default. You can see that I'm using the Borne Again Shell by default.

---

finger Command 3:29-4:11

There are a variety of different utilities that you can use on a Linux system to view information about user accounts without actually having to dig into the passwd file. For example, I can use the finger command to view information about a user on the system. All I have to do is type 'finger' and then the name of the user account that I want to get information for.

Here you can see that my username is rtracy. My full name is Robb Tracy. Here's my /home directory and my default shell. The finger command grabs that data right out of the passwd file right here. However, it also pulls information from other log files on the system.

For example, it tells whether or not I'm currently logged in to the system or not--and if so, how long. In this case, I've been logged in for 2 hours and 36 minutes.

---

id Command 4:12-4:44

The one thing that the finger command does not show is the user ID that's been assigned to my system. To view this information, I use the 'id' command followed by the username that I want to view information about. Here you can see that I have a user ID assigned to my account of 1000, my default group has a group ID number of 1000, and over here is a list of all the other groups that I am currently a member of.

Right now I'm a member of a only single group, so I have only one showing here. But if I had multiple group memberships, they would be listed out on this line.

Shadow File 4:45-8:27

Before we go on, there's one thing that I want to show you. Remember earlier I said that all standard user accounts that you can use to log in to the system will have a user ID of 1000 or greater. That is always true, except for the root account.

If we do an 'id root' command for the root account, you can see that the root account has a user ID assigned to it of 0, and it also has a group ID of 0. The root account always has a user ID number of 0, whereas all standard user accounts, non-superuser accounts, will have a user ID number of 1000 or more.

With that in mind, let's look at our next file, which is the shadow file. Every single account listed in my passwd file will have a corresponding account over here in the shadow file. And you can see here that it's structured in pretty much the same way as the passwd file.

Each line in the file represents a user account record, and that record is divided up into fields. The first field is my username. Obviously this has to match with whatever is in the passwd file.

The second field is my password. What you're seeing here is not actually my real password. To keep prying eyes from being able to just grab the shadow file and learning what my user's password is, the passwords are stored in the shadow file after they have been hashed. What you're seeing here is the hashed equivalent of my actual password.

The next field right here is my last modified field. This field displays the number of days since January 1st, 1970 that my password was last changed. In this example, my password was changed 16,673 days after January 1st, 1970, which was actually just a few minutes ago; it was today.

The next field is the min-days field. This field displays the minimum number of days required before a password can be changed. In this case, it's set to 0, so I can actually change my password anytime I want to.

The next field is the max-days field. The max-days field displays the minimum number of days before a password must be changed. This field specifies the number of days that must pass before you can change the password, and this field specifies how long before you have to change the password.

In this example, it's set to 99,999 days. Basically, this means that a password change is never required, because the system is not going to be around in 99,999 days.

The next field is the days-warned field. This field displays the number of days prior to password expiration that the user will be warned of that pending expiration. In this case, it's set to 7 days. So 7 days before my password expires, I'm going to get a warning telling me I need to change my password.

The next field right here is the disabled-days field. This field displays the number of days to wait after a password is expired to disable the account. Essentially giving you kind of a grace period before your account gets disabled when your password expires if you don't change your password. In this case, it's set to a null value.

Finally we have the expire field. This field displays the number of days since January 1st, 1970 after which the account will be disabled. Again it's set to a null value. It indicates that the account never expires.

I want to point something out to you real quick before we leave the shadow file. Notice that my user account here has a password assigned to it, and if I scroll up we also see that my root user account has a hashed password assigned to it. Notice that there are several other accounts listed in this file that have either an asterisk assigned to it or 2 exclamation points.

Standard Login and System User Accounts 8:28-9:14

You'll also notice that there are an awful lot of user accounts defined both here and in the passwd file. That's because the rtracy user account and the root user account are what we call standard login user accounts. They can be used to actually authenticate through the system.

These other accounts are non-login accounts. We call them system user accounts. System user accounts cannot be used to log in to the system. Instead, they're used by the services, the daemons, running on this system. When one of these services needs to do something--for example, in the Linux file system--it does so as its associated user account from the passwd and shadow files.

---

passwd File 9:15-10:21

Let's go ahead and look at the passwd file again. Notice that the system user accounts have a much lower user ID number assigned to them. For example, the FTP system account has a user ID of 14. Basically, with Linux, any user ID that is less than 1000 is considered to be a system user account and cannot be used for login.

This FTP user account is used by the FTP service running on this system. Suppose I attached to this system from a different computer using an FTP client and I log in as an anonymous user and I upload a file to this system.

Well, that file has to be written to this system's file system somewhere. When the FTP daemon writes that data to the file system here, it writes it using the permissions that are assigned to the FTP user. By configuring the system in this way, you can use permissions to control what a particular service can or can't do.

Shadow File 10:22-12:04

Let's take a look at the shadow file again real quick. You'll notice here that any user account that's a system user account will have a star or 2 exclamation points in the password field in the shadow file. That indicates that this is a system user account.

With that in mind, let's take a look at the groups that have been defined on the system. These are stored in the group file in /etc. Just as with the passwd and shadow files, each line in the group file is a single record that represents one particular group. Each record is composed of four different fields.

For example, down here notice that we have a group named rtracy. Understand that with some distributions, when you create a new user account on the system, it will create that user account and then automatically make that user a member of the user's group. Other distributions do not.

When you create a new user account, it will create an associated group with the same name as your user account. This distribution is one that does just that. I created a user named rtracy; therefore, it also created a group named rtracy and made me a member of it.

The first field in the record specifies the name of the group. In this case, it is rtracy. The next field is the password field, and just like with the passwd file, we don't actually put the group password in the group file. The x tells us that any passwords that have been defined are stored in the gshadow file.

The next field identifies the group ID number. In this case, the group ID number assigned to the rtracy group is 1000. The last field identifies the users that are members of this group. In this case, just the rtracy user is a member of the rtracy group.

Gshadow File 12:05-13:05

As we said just a minute ago, some distributions use an additional group file to store the group passwords. That is the gshadow file. Let's take a look at it. The gshadow file stores group passwords for the groups that are defined in the group file.

No passwords are used by default, but you can add a password to a group if you want to. If you add a password to a group, then whenever you try to add a user to that group, you'll be prompted to supply the group password before you'll be allowed to do so. Where we don't have any passwords defined, we don't have to supply a password to join a group.

Within the gshadow file we have the name of the group from the group file, then we have the password field that contains the hashed version of the password, which we don't have any defined for it on this system. If we did, the hashed password for the group would be added right here.

The next field right here specifies the group administrator account, which none are defined right now. Then the last field defines which users are members of that group.

Summary 13:06-13:17

That's it for this demonstration. In this demo, we reviewed the various files that are used to store group and user account information on a Linux system. We first looked at the passwd file. We then looked at the shadow file. We then looked at the group file, and then we ended this demonstration by looking at the gshadow file.
