7.2.1 User and Password Management

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

User and Password Management 0:00-0:14

In this lesson, we're going to review the process for managing Linux user accounts and their passwords. Let's begin by discussing how to go about adding a new user account to the system.

Adding a New Account 0:15-3:54

Now, you can do this from the shell prompt using the useradd command. As its name implies, this utility adds user accounts to the Linux system. The syntax is shown here. We run 'useradd', we specify any options we want to use, and then we enter the username for the new account we need to create. And I should point out here that every user account on your system should be given a unique username.

In this example, I'm going to use useradd to create a new account on the system named kmorgan. Now, you'll notice that I omitted any options when I did so. That means the kmorgan user account will be created using default parameters. These default parameters are contained in several configuration files in the /etc directory.

The first one is /etc/default/useradd. And as its name implies, this file contains default values that will be used by the useradd utility. This is an example of the useradd file right here. Notice that this file specifies that the default group for new users will be the group with a group ID number of 100. It also specifies that a home directory will be created for the user in the /home directory. In addition, the inactive account parameter is set to a negative one. And the expired parameter is set to a null value--this basically means that the user account will never expire. And the default shell is set to /bin/bash.

The skeleton directory is set to /etc/scale. And when we create the user account, they will also get a mail spool directory created automatically for that user. Now, this file is just a text file, so if you don't like these values, you can simply edit this file with a text editor and customize it the way you want.

The next file you have to be familiar with is /etc/login.defs. Now, this file contains values that can be used for the group ID and user ID parameters when you're creating a new account with useradd. It also contains default values for creating passwords in the /etc/shadow file.

Notice first of all that in this file we specify the default values for the fields within the shadow file, such as the max days, the min days, the minimum password length, and so on. It also, down here, identifies the user ID number range for both standard user accounts as well as system user accounts. In this example, the first user account created on the system that's, a standard user, will have a number of 1000. The next user account will have a number of 1001. The third user account will have a UID of 1002, and so on up to 60,000. You probably will not have a Linux system with 60,000 users, although I guess it is theoretically possible.

For user accounts on this system, we're going to see that the minimum user ID for a system user account is 201, and it will reach a maximum of 999. Basically, if we see a UID over 999, we automatically know that it's a user account. If we see a user account lower than 999, we know that it's a system user account. We also have parameters down here that define our group ID values. For standard groups, they're going to start at 1000 and end at 60,000. System groups will likewise start at 201 and end at 999. Again, this is a text file, so if you don't like these defaults, you can edit it with your text editor and set it to whatever values you want.

---

skel Directory 3:55-5:24

Next you need to be familiar with the skel directory in /etc. Instead of a configuration file, this is kind of a configuration directory. Basically, what happens is when you create a new user account, the useradd command is going to copy whatever files and whatever directories it finds in the /etc/skel directory to the new user's home directory when it's created. Basically, it makes it so that you can provision a user account when the account is created. If your organization requires them to have a folder named schedules and a folder named projects, as in this example, and maybe a word processing file down here that contains a list of all employees and their phone numbers, instead of going through and manually adding these every single time you create a new user, you simply drop those folders and files in /etc/skel. And then when you create the new user, they're automatically added to that user's home directory, and you didn't have to do anything.

Now, typically, the skel directory contains bash configuration files that are pre-configured, such as your .bashrc and/or .profile files. Here's the key point that I need you to remember. If you don't supply any options when you create a new user account, like we did when we created the kmorgan account a few minutes ago, then all of these defaults from these files will be used. But, you can override these defaults with the useradd command by specifying a list of options in the command line.

---

useradd Command 5:25-9:11

The first one is dash 'c' (-c), which specifies the user's full name. Actually, it fills in the comment field of the user record, but we really don't use it for comments, we use it for full names. The dash 'e' (-e) option specifies the date when the user account will be disabled. The dash 'f' (-f) option specifies the number of days after password expiration before the account ends up getting disabled. The dash 'g' (-g) option specifies the user's default group. That's a lowercase 'g', by the way. An upper case 'G', on the other hand, specifies any additional groups that the user is to be a member of--in addition to the default group. The dash 'M' (-M) option specifies that the user account be created without a home directory.

I should point out that that's a capital 'M'. Because if you specify a lowercase 'm', it allows you to specify, manually, the user's home directory. I don't typically use that because the default useradd configuration files will automatically create the user's home directory in /home. But if there's some reason why you need to use a different path or a different directory name, you would use dash 'm' (-m) to do that.

Now, there's also the dash 'n' (-n) option. Understand that by default, most modern Linux distributions will create a new group whenever you create a new user account, and that new group will be named with the same name as the user account. Understand this is a fairly new functionality. Not everybody likes it. If you want to do things the old way, you use the dash 'n' (-n) option to turn this functionality off. This basically allows you to create one group--say, users--and make it the default group for your user accounts, instead of creating a whole scad of groups named after each individual user.

The dash 'p' (-p) option specifies the user's encrypted password. Now, I've seen this mistake happen many, many times. People use the useradd command, they do dash 'p' (-p), and then they put in the clear text of the user's password. You cannot do that. Instead, what you have to do is run this command first: 'openssl passwd -crypt'. When you do, you'll be prompted to create a user's password. You enter in the password that you want to assign to the user account, and then this command will output that password--but in encrypted form. You can copy that password and then paste it back in here in the useradd command with the dash 'p' option.

The dash 'r' (-r) option is used if you're creating a system user instead of a standard user. The dash 's' (-s) option allows you to specify the default shell for the user. And the dash 'u' (-u) option allows you to specify a custom user ID for the user account. For example, let's suppose I wanted to create a new user account for a user named Erma Mcarthur on a Linux system. I want to use a username of emcarthur, full name of Erma Mcarther, a password of Linux123, and I also want to specify that she have a home directory created.

To do this, I first need to create an encrypted password. I run the command that we talked about on the previous slide. I run 'openssl passwd -crypt'. I provide the password that I want to use, and the output of the command is the encrypted form of the password. Then, I use the 'useradd' command right here. I use the dash 'c' (-c) option to specify the full name. Dash 'm' (-m) to specify that the home directory be created. And then I use the dash 'p' (-p) option and put the encrypted password in quotes. Specify the default shell. And then, finally, I specify the username for the user account.

When I do, the account is added. And if we look at that account in the password file, we see the new record that's been added. Notice that some of these values were defined by the default file--such as user ID, the group ID--but the full name and the default shell were created using the parameters we specified with the useradd command.

---

passwd Utility 9:11-10:37

Next, let's look at the passwd utility. You use passwd to change an existing user's password. Now, at the beginning of this video, we created a new user account called kmorgan using the useradd command, but we didn't specify a password with the command. Therefore, kmorgan has no password and his account is locked. We can fix that by entering 'passwd kmorgan'. We're prompted to provide the password. Notice that when I do, the system will check and see if I have provided a strong password or not. If I have not, it says, "Hey, you know, this password really stinks, you should use something else." But I said, "I don't care, I'm going to use it anyway." And when I do, the password is assigned to the kmorgan account, and then kmorgan can log into the system.

When you're running the passwd command, there's several options you can use.

The dash 'l' (-l) option locks the user's account. If you have a locked account, you can unlock it with the dash 'u' (-u) option. You can use the dash 'd' (-d) option to remove a user's password. You can use the dash 'n' (-n) option to set the minimum number of days required before the user can change their password again. The dash 'x' (-x) option sets the maximum number of days before the password has to be changed. The dash 'w' (-w) option sets the number of days prior to password expiration that the user will be warned of the pending password expiration. And the dash 'i' (-i) option sets the number of days to wait after a password has expired to go ahead and then disable the user account if they don't actually change their password.

Modifying a User Account 10:38-12:10

Now that you know how to create a new user and how to set that user's password, we now need to review how you go about modifying an existing user account. And this can be done from the command line using the usermod command. Now, the cool thing about usermod is that

its syntax is almost identical to that used by useradd. The syntax is to enter 'usermod', the options you want to use, followed by the username that you want to modify.

As you might guess, the dash 'c' (-c) option sets the user's full name, the comment field; 'e' sets the date when the user account will expire; 'f' sets the number of days after password expiration when the account will be disabled. Lowercase 'g' sets the user's default group, while uppercase 'G' specifies additional groups that the user is to be made a member of. Dash 'l' (-l) is used to change the username itself. For example, if a user changes their name for some reason, you can change their username to match using dash 'l' (-l) --lowercase 'l', I should say. Uppercase 'L' (-L) locks the user's account. Dash 'm' (-m) allows you to set the user's home directory. Dash 'p' (-p) is used to set the user's password. And as with the useradd command, the password you specify with dash 'p' has to be submitted in encrypted format, and you should use the 'openssl passwd -crypt' command before you use dash 'p'.

The dash 's' (-s) option is used to set the default shell for the user account. The dash lowercase 'u' sets the user ID for the user account. And dash capitol 'U' (-U) unlocks a user's account that has been locked previously with the dash capital 'L' option.

Deleting a User Account 12:11-13:05

The last topic we need to cover here is that of deleting user accounts. This is done using the userdel command. In order to delete a user account, I just enter 'userdel' followed by the username that I want to get rid of. For example, if we want to get rid of the kmorgan account that we created earlier, we enter 'userdel kmorgan'.

Now, it's important to note that by default, userdel will not remove that user's home directory from the file system when you delete the account. This is actually a good thing, because if that user has proprietary documents in there or any other information in their home directory that's the property of the company they work for. You don't want to delete it. You might need it for the next person who takes that user's place in the organization.

However, if there is a situation when you do want to remove that user's home directory when you delete the account, then you need to include the dash 'r' (-r) option. That will remove the account and that account's home directory.

Summary 13:05-13:23

That's it for this lesson. In this lesson, we discussed how you can manage Linux user accounts and their passwords from the command line. We looked at the commands you can use to add a user account. We looked at the commands you can use to change a user's password. We talked about modifying user accounts. And then we ended this lesson by discussing how you delete user accounts.
