





the second field contains an encrypted password that is prefixed with an exclamation point (!), that user account is locked. Removing the ! will unlock the account.

## Account Administration

Now that we've discussed the files involved with system and user accounts, we can work on adding, removing, and modifying user accounts. Let's start by adding a user to your system using the **useradd** command. We can use the **getent** command to get the new entry we've created from the passwd file:

### INTERACTIVE SESSION:

```
[username@username-m0 ~]$ sudo useradd foobar
[username@username-m0 ~]$ getent passwd foobar
foobar:x:1045:1045::/home/foobar:/bin/bash
```

Your uid and gid will probably be different, but the rest of the entry will be identical. Keep in mind that you need to use super user privileges to add, delete, or modify a user. Look in **/home** and **/var/mail**; **useradd** not only adds an entry to **/etc/passwd**, but it also creates a home directory and a mail spool for the new user (don't worry if you aren't familiar with mail spools for now). If no primary group is specified, **useradd** also creates a group with the same name as the user and sets the user's primary group to this new group. Now, using **sudo**, look for the new user's entry in **/etc/shadow**:

### INTERACTIVE SESSION:

```
[username@username-m0 ~]$ sudo less /etc/shadow
...
foobar:!!!:15523:0:99999:7:::
```

Your new user's account is **locked!** The **useradd** command creates the user but it does not set a password for the account. By default, the account is locked, and it must have a password set before the new user can log in. To set a password for your new user, use the **passwd** command. Your keystrokes will not appear on the screen, but **passwd** requests the new password twice so it can confirm that you typed it correctly:

### INTERACTIVE SESSION:

```
[username@username-m0 ~]$ sudo passwd foobar
Changing password for user foobar.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
```

**Note** If your password is deemed too simple, **passwd** displays a message to convey that, but still prompts you to re-enter it, and allows you to set the password. Search the internet for

suggestions on [password strength](#).

Now try using **ssh** to log into your machine as the new user, using the password you just set. Once you have confirmed that the account is working, you can log out of the ssh session. Using the **groupadd** command, let's create a group and add foobar to it. To specify a group id number for your new group, use the **-g** flag (keep in mind that gid numbers of 999 and below are reserved for system groups, so you should use numbers of 1000 or larger); otherwise a gid will be chosen for you:

#### INTERACTIVE SESSION:

```
[username@username-m0 ~]$ sudo groupadd -g 5555 testgroup
[username@username-m0 ~]$ getent group testgroup
testgroup:x:5555:
```

Your new group has been created, but it's empty for now. Create two more groups named **staff** and **employees**. If you want to specify gid numbers for these groups, remember to use numbers greater than or equal to 1000. After you have created your groups, add your new user to one of them. This is where the **usermod** command comes in handy. Usermod is used to add and remove users from groups, change user information, and lock accounts:

#### INTERACTIVE SESSION:

```
[username@username-m0 ~]$ sudo usermod -G testgroup foobar
[username@username-m0 ~]$ groups foobar
foobar : foobar testgroup
```

The **groups** command lists the groups to which a user belongs. Now let's see what happens when you try to add the user to the other two groups you created:

#### INTERACTIVE SESSION:

```
[username@username-m0 ~]$ sudo usermod -G staff,employees foobar
[username@username-m0 ~]$ groups foobar
foobar : foobar staff employees
```

Where did testgroup go?! Look at the **man** page for **usermod** to see if you can figure out why testgroup disappeared from the list of groups of which foobar was a member. Keep in mind that foobar's primary group membership is with the group "foobar" and all other groups are secondary. Take a look at **/etc/group** to see how it looks after adding groups and then adding a user to those groups. It's possible to edit the **group** file by hand to add a group or to add users to groups if you are more comfortable doing it that way. Feel free to add more users or groups, or add users to groups, until you're comfortable with the process. Also, experiment with the **groupmod** command to modify the names and gid numbers of some of the groups you've created. Be careful not to modify any system groups as there could be adverse consequences!

The last thing we'll cover in this lesson is removing users and groups from the system using the **userdel** and **groupdel** commands. Both are fairly straightforward to use, in fact, **groupdel** takes only one argument: the group to be deleted. Let's see it in action:

#### INTERACTIVE SESSION:

```
[username@username-m0 ~]$ groups foobar
foobar : foobar staff employees
[username@username-m0 ~]$ sudo groupdel employees
[username@username-m0 ~]$ groups foobar
foobar : foobar staff
```

In this example, when a group is removed, any user who is in that group has it removed from his or her list of secondary groups. There's really nothing else to the **groupdel** command. The **userdel** command has a couple of options. Just executing **userdel** and passing it a username will delete that user's account, but it does not remove that user's home directory or mail spool. Although sometimes you'll want to remove the user and all of his or her data. To do that, run **userdel** with the **-r** flag. Let's get rid of the user foobar. (I never liked him anyway):

#### INTERACTIVE SESSION:

```
[username@username-m0 ~]$ sudo userdel -r foobar
```

Afterwards, this is the state of our system:

#### INTERACTIVE SESSION:

```
[username@username-m0 ~]$ ls /home
username
[username@username-m0 ~]$ ls /var/mail
username
[username@username-m0 ~]$ grep foobar /etc/passwd
[username@username-m0 ~]$
```

Foobar and all of his data are gone. Take a moment and think about what you've done. When you use your super user powers, you can do things like remove someone's account and delete all of their data. What if you were removing an account and made a typo and removed the wrong one? It's *extremely* important to understand and respect the power you will have as an administrator with super user privileges. Anything from a mistyped **userdel** to an errant **rm** can destroy data and lead to down-time and loss of productivity. Always think before you type and, perhaps more importantly, keep solid backups!

Source: [http://courses.oreillyschool.com/sysadmin2/user\\_accounts.html](http://courses.oreillyschool.com/sysadmin2/user_accounts.html)