15.1.3 Configure sudo

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

Configure sudo 0:00-0:14

In this demonstration, we're going to look at how to configure sudo. sudo allows standard users on your Linux system to complete certain commands with root level access.

Use sudo 0:15-1:38

Basically, using sudo, you can configure who can do what as root, which is really quite powerful. This distribution is configured by default to ask the user to enter their own password when they run sudo.

Let's just go ahead and confirm that real quick. I'm going to use 'sudo' and then I'm going to enter in the name of the command that I want to run as the root user.

In this case, let's try running 'ps -elf'. Hit Enter, and it prompts me to enter the rtracy user password. I do it and it doesn't work, because as you can see here, my rtracy user account is not in the sudo configuration file called sudoers.

Just be aware that some Linux distributions--and I haven't seen this in quite some time--but there was a time when many Linux distributions came configured such that if you tried to run sudo, you would have to enter in the root password to complete root-level tasks.

Which, if you think about it, doesn't make a lot of sense because if I, as a standard user, already know the root password, why would I bother with sudo? I would just use su to switch to the root user.

With the sudo configured in this manner, it makes a lot more sense because I have to remember only my user's password and I'm still allowed to complete some root-level tasks--not all of them, just those that I'll specifically allow that user to do.

Add Users to the sudoers File 1:39-2:14

In order for sudo to work, we need to add rtracy to the sudoers file. The first thing we need to do it switch to my root user account. You don't edit the sudoers file directly. Instead, you use the visudo command. This will load the sudoers file in the vi editor for you.

The reason you do it this way is because visudo has some error checking built into it. When you are done and you exit out and you save your changes, visudo will actually look at the syntax you used in the file and warn you if you made a mistake, which can be really useful.

Define Aliases 2:15-4:03

You'll notice that this file is broken up into several chunks. We have our host aliases here, we have our user aliases here, and we have our command aliases here.

Let's first define a new user alias called POWERUSERS. Press the Insert key, arrow down, and under User_Alias, I'm going to enter 'User_Aliases' and we will name it 'POWERUSERS'.

I have to specify which users are going to be included in that alias. Let's add 'rtracy' and also the 'ksanders' user on this system.

Now that we've specified who is going to be allowed to run root level commands, we next need to specify which of those commands they are going to be allowed to run as root, because we probably don't want to allow these users to run just any command as root.

They would basically then be root users. What we want to do is say, "You are allowed to run this certain subset of commands--just the ones you need to do your job--as the root user."

Come down here under Command Aliases. Let's define a new command alias called KPROCS that contains the kill and the kill all commands. Enter 'Cmnd_Alias'.

And by the way, I might point out that there are examples of how to use the correct syntax for this file. Already in the file you can see up here some examples for defining a user alias was specified here. Some pre-configured command aliases are listed down here.

We start off with 'Cmnd_Alias', then the name of the alias. We want to call it 'KPROCS' for kill processes, '=', and then we have to list out the commands that we want the user to be able to run. I'll point out that you have to use the full path to these commands.

Use the which Command 4:04-4:47

Let's open up a new window so we can run the which command. Move this over here. The first command we want the user to be able to run as root is the kill command, so let's type 'which kill' and we see that the path is /user/bin/kill, type '/user/bin/kill', and the second command we want them to be able to run is 'killall', which I believe is probably in the same directory. Yep, '/user/bin/killall'.

At this point we've defined our user alias, "who", and then we defined our command alias, which is "what".

Define Where the Root Privileges Can Be Used 4:48-6:06

Now the next thing we need to do is specify "where". In other words, where can these users run these commands as root? This is done using a host alias up here.

For our purposes today, let's just define a simple host alias that allows these users to run these commands just on the local system. The hostname of my system is fs5, so we enter 'Host_Alias', and then we give it a name. Let's call it 'PHOSTS=' and then the hostname of this system, which is 'fs5'.

At this point we have our user alias defined, our command alias defined, and our host alias defined before we can use sudo. However, we have to glue all of these three different entities together. To do this, let's come down towards the end of the file after all of these command alias templates you can use.

By the way, these are really useful. They are not enabled by default, but you can see that we have a networking command alias, software management alias services, database management, storage, delegating permissions, processes--which notice we could have used for what we are doing today--drivers, and so on.

They are already predefined; you just have to un-comment them, and then you can use them.

Let's add a new line right here.

Combine Parameters for sudo 6:07-7:50

Let's glue the three entities we just created together. We do this by first specifying the name of the user alias that we defined that we want to allow root level access to certain commands. We enter 'POWERUSERS'. Then we specify the host alias, in other words, where we want to allow these users to run these root level commands: 'PHOSTS ='. And then we have to specify which user account we want these users to be able to run these commands as.

As you might guess, we want to allow them to run these commands as root. We put '(root)' in parentheses, and then we specify which commands they are going to be able to run as root, which is 'KPROCS', which as you will recall we defined earlier allows us to run /user/bin/kill and /user/bin/killall. Okay, this looks good.

Let's go ahead and save the changes to the file. I'll press Esc and we will enter 'exit' and when we do this, visudo is going to look for any errors we made in the file and notice that I do have a syntax error near line 22. If I press Enter here, it gives me some options--what do I want to do to resolve this?

I'll use 'e' to edit the sudoers file again. Oh, I made the mistake right there. I did User_Aliases, not User_Alias. Notice that when I went in to edit the file again, it took me right to the line where the mistake was--very nice. I get rid of the es that I inadvertently put in there and now let's try it one more time. 'exit' and we are good. Everything is happy. No more errors were displayed.

Test Configuration 7:51-10:26

Let's go ahead and test this configuration. To do this, I'm going to go over here to this window. I'm going to switch to my root user account. What I'm going to do--and this is important--I'm going to run 'gedit' here out of this window and I'm going to run it as root.

Now because I'm going to run this command as root, my rtracy user is not going to be able to kill its process. I'm going to run gedit. The application is displayed, and for convenience's sake let's send the gedit process to the background. I'll press CTRL + Z, and then we will do 'bg1' to send it to the background.

Now let's go back over to my first window and let's 'exit' out of my root account, and now let's try killing the gedit process over here as my rtracy user. Before I can do that, I do need the process ID number of the gedit process.

Let's just enter 'ps' over in this other window, and it tells us here that the PID number is 3223. I'm going to rearrange these so I can see that PID number a little easier. Slide this over, and now we can see the output of the ps command.

The first thing I'm going to try is just killing it without running sudo. In other words, I'm going to try and kill this process as rtracy. I'm going to do 'kill' and I'm going to send it the '-15' termination signal, and then I need to specify the PID number of the process, '3223'.

It says, "Uh, sorry. You're not allowed to do that." Why? Because I'm trying to kill a process owned by root as the rtracy user, and that is not allowed.

Let's try it again, but this time let's run the same command using sudo, which will elevate our privileges to the root level. 'sudo', Space, and then the command that we want to run. We are going to try to 'kill' that same process again.

I'm prompted to enter my rtracy user's password, and notice this time I was able to kill that command. Why? Because rtracy was in the User_Alias and then kill command was in the Command_Alias and fs5 is in the Host_Alias, hence I was allowed to run kill as root on this system without knowing the root user's password.

Summary 10:27-10:37

That's it for this demonstration. In this demo we talked about how to configure sudo to allow users to run commands at the shell prompt with root level privileges, without actually giving them the root password or giving them full root level access to the system.