15.1.4 root User Facts

The root user account is the Linux system superuser and can perform any task. Some Linux commands cannot be run by anyone but the root user. The root account is created during the installation process, and it receives the account number 0 (zero). In contrast, normal (standard) user accounts receive ascending numbers beginning at 500 or 1000, depending on the distribution.

This lesson covers the following topics:

- Security guidelines
- Commands to manage root-level access
- Commands to manage limited root-level access

Security Guidelines

To protect the root user account, use the following guidelines:

- When performing tasks that require the root user account, use the **su -** command to switch to the root user and execute the command and then use the **exit** command to revert back to the regular user account.
- As a general rule, create a user account that gives sufficient permissions to perform most daily tasks. Use this account instead of the root user account when logging in to the system.

Commands to Manage Root-Level Access

The following commands are used to manage root-level access to the system.

Command	Funciton	Examples

su	Switches to the root user account. Be aware of the following **su** options:
	▪ **su -l *user_name*** switches to the specified user in a login shell.
	▪ **su *user_name*** (without the dash, but with the username) switches to the user, but does not load that user's environment variables.
	▪ **su - *user_name*** (with the dash and username) switches to the user and loads the user's environmental variables.
	▪ **su -** (with the dash but no username) switches to the root user and loads the root user's environmental variables.
	▪ **su** (no dash or username) switches to the root user but does not load the root user's environmental variables.
	▪ **-c "*command*"** executes a single command as the root user.
	  ▪ The command is enclosed in either single or double quotation marks.
	  ▪ Include **-l *user*** to execute the command as a user other than root.
	
	su requires the user's password before switching to the account except when switching from root to a normal user.
		su -l jsomes switches to the jsomes user account with jsomes' environment variables.
		su - switches to the root account using root's environment variables.
		su -c "ls /home/rgurate" switches to the root user and executes the **ls** command on the user rgurate's home directory.

exit	Returns to the account from which the **su** command was typed. When no **su** command has been typed, **exit** terminates the shell. When using a computer that uses a shell exclusively, **exit** logs the user out.	[root@ls4 ~]# **exit** logout [jsomes@ls4 ~]$
logout	Logs out of the system, while leaving the system powered on.	**logout** logs the user out of the shell. It is identical to **exit**.

Commands to Manage Limited Root-Level Access

To give standard user accounts the permissions to execute a limited set of commands as the root user, use the **sudo** command coupled with the /etc/sudoers file. Be aware of the following facts about the **sudo** command and the /etc/sudoers file:

- When users need to execute the command, they use the **sudo** command followed by the command they want to execute, such as **sudo nano myfile**. The user is then prompted for their own password prior to running the command, not the root account password.
- Users and the commands they are allowed to execute with elevated privileges are specified in the /etc/sudoers file.
- The /etc/sudoers file must be edited using the **visudo** command using the root account.
- **sudo** logs information about the users and the commands they run, as well as failed attempts to use **sudo** in the /var/log/security log.

The following table describes the sections used to configure the **/etc/sudoers** file.

Section	Description	Examples
User_Alias	Specifies a set of users who are allowed to execute a specific set of commands using the **sudo** command.	**User_Alias INSTALLERS = jsmith, psimms** adds the users jsmith and psimms to the INSTALLERS alias.
Cmnd_Alias	Specifies a set of commands that users can execute using the **sudo** command.	**Cmnd_Alias INSTALL = /bin/rpm, /usr/bin/up2date, /user/bin/yum** assigns the **rpm**, **up2date**, and **yum** commands to the INSTALL alias. Users associated with the INSTALL alias can execute these commands.
Host_Alias	Specifies a list of computers on which **sudo** users can perform commands.	**Host_Alias FILESERVERS = fs1, fs2, fs3** adds the three computers to the alias. **Host_Alias EVERYWHERE = *.mydomain.com** creates an alias for all computers on the mydomain network.
Runas_Alias	Specifies a username that is used when running commands with **sudo**. Usually, this is just root.	**Runas_Alias DATABASE = root** specifies that **sudo** commands are run as the root user.

Each of these aliases are defined independently within the /etc/sudoers file. To grant users elevated access to the system, these aliases need to be associated with each other to define exactly what will happen. The syntax is as follows:

User_Alias Host_Alias* = (*user*) *Cmnd_Alias

For example, the aliases defined in the table above can be associated with each other using the following entry in /etc/sudoers:

INSTALLERS FILESERVERS = (root) INSTALL

Using this entry, the users associated with the INSTALLERS user alias are allowed to run the commands in the INSTALL command alias on the hosts contained in the FILESERVERS host alias as the root user.

If **Runas_Alias** is omitted, the default is to run the commands as the root user.

The following table describes the commands for configuring and using sudo.

Command	Function	Examples
visudo	Opens the **/etc/sudoers** file for editing. The command opens the vi editor and checks the file for sudo syntax errors before saving and exiting.	**visudo** opens the **/etc/sudoers** file in the text editor.
sudo	Executes a command as the root user. To use this command, first type **sudo** and then type the command as you normally would.	**sudo yum install sysstat** installs the sysstat package as the root user.
sudoedit	Allows users to securely edit files. This command is equivalent to running **sudo -e**. When using **sudoedit**, users edit the desired file as themselves, and not the root account. When run, **sudoedit** first creates a temporary copy of the desired file. Changes are then made to that file. When done, the changes made to the temporary files are copied back to their original location, and the temporary versions are removed. To use **sudoedit** to limit users in the managers group to edit a specific file, edit the sudoers file and add a **sudoedit** line similar to the following example: **%managers ALL = sudoedit /*path_to_the_file*.** A common implementation of this is to use the wheel group. Most Linux systems use user groups as a security protocol to control access privileges. The wheel group is a special user group used on some Linux systems that controls access to the **su** or **sudo** command. Therefore, to use **sudoedit** to limit users in the wheel group, add a **sudoedit** line similar to the following example: **%wheel ALL = sudoedit /*path_to_the_file*.**	**sudoedit /etc/hosts.allow** lets those in the group specified in the sudoers file (such as the managers group) to edit the hosts.allow file.