Chapter 11

Controlling Access and Permissions

Understanding Ownership

Linux uses a three-tiered approach to protecting files and directories:

- **Owner:** In the Linux system, each file and directory is assigned to a single owner. The Linux system administrator can assign the owner specific privileges to the file or directory.
- **Group:** The Linux system also assigns each file and directory to a single group of users. The administrator can then assign that group privileges that are specific to the file or directory and that differ from the owner privileges.
- **Others:** This category of permissions is assigned to any user account that is not the owner or in the assigned user group.
- `ls -l`

Controlling Access Permissions

sudo chown Christine customers.txt

sudo chgrp marketing customers.txt

chmod g-w customers.txt

chmod ug=rwx salesdata.txt

Octal Mode: chmod 664 research.txt

Special Permissions: chmod u+s myapp OR chmod 4750 myapp

chmod g+s /sales OR chmod 2660 /sales

chmod o+t /sales OR chmod 1777 /sales

TABLE 15.2 Results from common **umask** values for **Default Permissions**

Access Control Lists

getfacl test.txt

setfacl -m g:sales:rw test

setfacl -x g:sales test

sudo setfacl -m d:g:sales:rw /shared/sales

EXERCISE 15 . 1: Creating a Shared Directory

Understanding Linux User Types

Root: The root user account is the main administrator user account on the system. It is identified by being assigned the special user ID value of 0. The root user account has permission to access all files and directories on the system, regardless of any permission settings assigned.

Standard: Standard Linux user accounts are used to log into the system and perform standard tasks, such as running desktop applications or shell commands. Standard Linux users normally are assigned a $HOME directory, with permissions to store files and create subdirectories. Standard Linux users cannot access files outside their $HOME directory unless given permission by the file or directory owner. Most Linux distributions assign standard user account user IDs over 1000.

Service: Service Linux user accounts are used for applications that start in the background, such as network services like the Apache web server or MySQL database server. By setting the password value in the shadow file to an asterisk, these user accounts are restricted so that they cannot be used to log into the system. Also, the login shell defined in the /etc/passwd file is set to the nologin value to prevent access to a command shell. Service accounts normally have a user ID less than 1000.

Escalating Privileges

Most Linux administrators use privilege escalation to allow their standard Linux user account to run programs with the root administrator privileges. This is done using three different programs:

su: The su command is short for substitute user. It allows a standard user account to run commands as another user account, including the root user account. To run the su command, the standard user must provide the password for the substitute user account. While this solves the problem of knowing who is performing the administrator task, it doesn't solve the problem of multiple people knowing the root user account password.

sudo: The sudo command is short for substitute user do. It allows a standard user account to run any command as another user account, including the root user account. The sudo command prompts the user for their own password to validate who they are.

sudoedit: The sudoedit command allows a standard user to open a file in a text editor with privileges of another user account, including the root user account. The sudoedit command also prompts the user for their own password to validate who they are.