The Linux kernel allows properly authenticated users to access files and applications. While each user is identified by a unique integer (the user id or UID), a separate database associates a username with each UID. Upon account creation, new user information is added to the user database and the user's home directory must be created and populated with some essential files. Command line programs such as **useradd** and **userdel**as well asGUI tools are used for creating and removing accounts.

For each user, the following seven fields are maintained in the **/etc/passwd** file:

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Details** | **Remarks** |
| **Username** | User login name | Should be between 1 and 32 characters long |
| **Password** | User password (or the character **x**if the password is stored in the **/etc/shadow**file) in encrypted format | Is never shown in Linux when it is being typed; this stops prying eyes |
| **User ID (UID)** | Every user must have a user id (UID) | * UID 0 is reserved for root user * UID's ranging from 1-99 are reserved for other predefined accounts * UID's ranging from 100-999 are reserved for system accounts and groups * Normal users have UID's of 1000 or greater |
| **Group ID (GID)** | The primary Group ID (GID); Group Identification Number stored in the **/etc/group** file | Is covered in detail in the chapter on *Processes* |
| **User Info** | This field is optional and allows insertion of extra information about the user such as their name | For example: **Rufus T. Firefly** |
| **Home Directory** | The absolute path location of user's home directory | For example: **/home/rtfirefly** |
| **Shell** | The absolute location of a user's default shell | For example: **/bin/bash** |

Types of Accounts

By default, Linux distinguishes between several account types in order to isolate processes and workloads. Linux has four types of accounts:

* root
* System
* Normal
* Network

For a safe working environment, it is advised to grant the minimum privileges possible and necessary to accounts, and remove inactive accounts. The **last** utility, which shows the last time each user logged into the system, can be used to help identify potentially inactive accounts which are candidates for system removal.

Keep in mind that practices you use on multi-user business systems are more strict than practices you can use on personal desktop systems that only affect the casual user. This is especially true with security. We hope to show you practices applicable to enterprise servers that you can use on all systems, but understand that you may choose to relax these rules on your own personal system.

## Understanding the root Account

**root** is the most privileged account on a Linux/UNIX system. This account has the ability to carry out all facets of system administration, including adding accounts, changing user passwords, examining log files, installing software, etc. Utmost care must be taken when using this account. It has no security restrictions imposed upon it.

When you are signed in as, or acting as root, the shell prompt displays '**#**' (if you are using **bash** and you have not customized the prompt, as we have discussed previously). This convention is intended to serve as a warning to you of the absolute power of this account.

**sudo**has the ability to keep track of unsuccessful attempts at gaining root access. Users' authorization for using **sudo** is based on configuration information stored in the **/etc/sudoers** file and in the **/etc/sudoers.d** directory.

A message such as the following would appear in a system log file (usually **/var/log/secure**) when trying to execute **sudo bash** without successfully authenticating the user:

**authentication failure; logname=op uid=0 euid=0 tty=/dev/pts/6 ruser=op rhost= user=op**  
**conversation failed**  
**auth could not identify password for [op]**  
**op : 1 incorrect password attempt ;**  
**TTY=pts/6 ; PWD=/var/log ; USER=root ; COMMAND=/bin/bash**

## The sudoers File

Whenever **sudo**is invoked, a trigger will look at **/etc/sudoers** and the files in **/etc/sudoers.d** to determine if the user has the right to use **sudo** and what the scope of their privilege is. Unknown user requests and requests to do operations not allowed to the user even with **sudo**are reported. The basic structure of entries in these files is:

**who where = (as\_whom) what**

**/etc/sudoers** contains a lot of documentation in it about how to customize. Most Linux distributions now prefer you add a file in the directory **/etc/sudoers.d**with a name the same as the user. This file contains the individual user's **sudo** configuration, and one should leave the master configuration file untouched except for changes that affect all users.

You should edit any of these configuration files by using **visudo**, which ensures that only one person is editing the file at a time, has the proper permissions, and refuses to write out the file and exit if there are syntax errors in the changes made. The editing can be accomplished by doing a command such as the following ones:

**# visudo /etc/sudoers  
# visudo -f /etc/sudoers.d/student**

The actual specific editor invoked will depend on the setting of your **EDITOR**environment variable.