

27 September 2006

By: Bogdan Radulescu, Editor, Linux Software Reviews



## Managing User Accounts on Linux

*The straightforward guide for the command line*

Most of us need to create users accounts once in a while. System administrators frequently work with user accounts on a daily basis. The following text wants to explain how to work with (create, modify) user accounts and groups in a Linux environment. Most of the people never bothered to understand this or they run into complicated documentation that doesn't clarify things. I will try to make this as comprehensive as possible, but in the end, you'll be the one to judge. Note: I know most of you use the root account for day to day tasks but this is wrong. I know because I do wrong things too :). Now, we'll need to login as root to manage user accounts. The easiest way to add a new user is to use the `useradd` command like this: **`useradd bart`**. We have now created a new user called bart. All basic requirements are met at this point. We would probably want to assign a password for that user and this is done with the command **`passwd bart`**. The last paragraph explained the basics but it doesn't help you manage anything. Next, I'll tell you how things actually work. Everything starts with the `/etc/passwd` file. Here, you can see all the accounts that exist on your Linux system and for each one, there are several fields that describe various stuff. By default, your `/etc/passwd` file has several entries that are actually users for programs that need to control processes or need "special" access to the filesystem. You'll also see there the root account and perhaps some user accounts that were created at installation time or after that. Each entry has the following fields: **`user:password:UID:GID:comment:home:shell`** `user` is the username that is used for logging or by programs. The username is case sensitive on Linux systems and it is recommended to keep the special characters out of it. `password` is the field where the encrypted password is stored in. The `passwd` command encrypts the passwords and stores them in that field. The default encryption algorithm used is considered rather poor today. It is better to choose shadow passwords and, in that case, the field will remain blank and all the passwords will be stored in the `/etc/shadow` file. `UID` is the user ID. It's a numerical value that is bound to a user. For the root user is always 0. The UID has to be unique and should have a value between 0 and 4 mil. Usually, for users, the UID is greater than 100. All the files in a Linux system have an UID. This UID determines the ownership of files and processes. `GID` is the group ID for the primary group of the user. This is also a numerical value and to root, also has the value 0. For every user, there is at least a GID. This field identifies the primary group to which a user belongs. Note that a user can be assigned to several groups. Both the UID and the GID are very important for filesystem security. `comment` is a field that holds text information about a user. Usually, you add here the name of the user but you can also add the phone number, the e-mail address or whatever you like. Where there are many users to manage, the comment field can really come in handy. `home` defines the home directory of that user. This directory is created automatically by the `useradd` command. If you want to change it from here, you should keep in mind that it has to exist. `shell` is the shell that will be used by the user. The default should be more than ok most of the time. Accounts created for people will have assigned the bash shell and the accounts created for programs will have no login which is a nice trick for disallowing logins with that user. As you can see, there are many things stored in a `/etc/passwd` file. You can either enter all this information by hand or count on various commands to aid you in adding, modifying or deleting user accounts. Now, we should take a look in the `/etc/group` file to see how it is structured. The entries in each field are as follows: **`name:password:gid:users`**, I'm not going to dissect those because they are pretty suggestive. I only want to add that in the users' field, the users should be separated by a comma if you want to add more. The whole purpose of the groups is to contain several

users. A group with only one user isn't really a group :). **An example:** We have only a PC in the Springfield town. We want to create an account for the Simpsons. I will add a new group like this: **groupadd simpsons** Now, we'll have the Simpsons group in our /etc/group file. We want to add here the users Homer, Marge, Bart, Lisa, Maggie. We do this using the command: **useradd -c "The head of the Simpson family" -G simpsons Homer**. We've just created the Homer username, we've added a comment for it and we assigned it in the Simpsons group. Now, we should set a password for Homer. We do this with the **passwd Homer** command. We should do this for all the members in the Simpson family and then we'll be good to go. If we made a mistake and want to modify something for a user, we use the command **usermod** similar to how we use **useradd**. The difference is that **useradd** adds users and, if a user already exists, we get an error message. If we want to delete a user we have the command **userdel**. This allows us to delete a username if it isn't logged on. Similarly, we have: **groupmod** and **groupdel** for working with groups. These commands have many functions and things can get really complicated and I don't intend to explain everything. You always have the main file and if you have a general idea, it will be a lot easier for you.