Linux Security

C. W. Andreasen – rev 7.01724

Microsoft Windows users are in a constant battle against hackers, virus', adware, ransomware and other security dangers. Windows system need a firewall, anti-virus software, and still the user may click on a bad web site or accidentally install a Trojan that introduces bad software into the computer.

Linux does not have this problem, in fact the average Linux user needs no firewall, although if the user wishes, there is a software firewall built into Linux, but the default setting is disabled, There is no Linux anti-virus program because Linux is immune to virus'. The very nature of Linux makes it nearly impossible to infect anything.

One of the first things that makes it tough for a potential hacker, is essentially no two installations are exactly the same. Programs do not reside in a closed folder or location, different functional parts of the programs are spread throughout the system. Programs can have any name and do not require extensions like ".EXE" or anything else. Some files do use extensions but that is for the programmers or users, Linux does not require it. In Linux a file, folder, program, can have any name and case matters, so "Mydocs" is totally different than "mydocs". Extensions are just for user convenience. The case sensitivity multiplies the difficulty of hacking a password. In MS-Windows "password" and "PASSWORD" are the same however in Linux they are two totally different passwords. The hacker would need to know not only the password, but the case of each character.

In Linux EVERY folder, document or file… anything the user or system writes to the disk, has two types of security. The system absolutely refuses to open any file or folder unless the user either owns the file, or is a member of the assigned group. There is one exception and this is "root". Root is not a user but has the ability to do "anything" without restriction. For this reason one cannot log on as root (root is not a user) and to use root the person must know the password. For this reason root should be assigned a secure password and this password is kept secret and never given out. Users who must occasionally do root functions need to use "sudo" and belong to the sudo group, to get temporary access to root functions.

That being said, one can create a user who can log on, and make that ID a root member who is not the root but has access to all of the powers of the root. This would be like the user pi on the Raspberry Pi computers. Pi is not the root, but is a member of all groups. Normal users should not be members of the root group, but should be members of the sudo group if they will need root power from time to time they can invoke sudo. It is possible, at the console level, to in effect, become root. The command is 'sudo su' which makes the user a "Super User", and the prompt changes from the normal dollar sign ($) to the pound sign (#)… of course you must be logged on on as a regular user who belongs to the sudo group to do this, and you must know the user's password.

What you just read is the foundation of Linux security. The next level is the user. Every user has an ID (login name) and a password. A good password that is kept safe is the first line of defense. You cannot log into my account, as me, unless you know the password. Recall I mentioned that upper and lower case matters so "YoUrnaMe" is much more secure than "yourname".

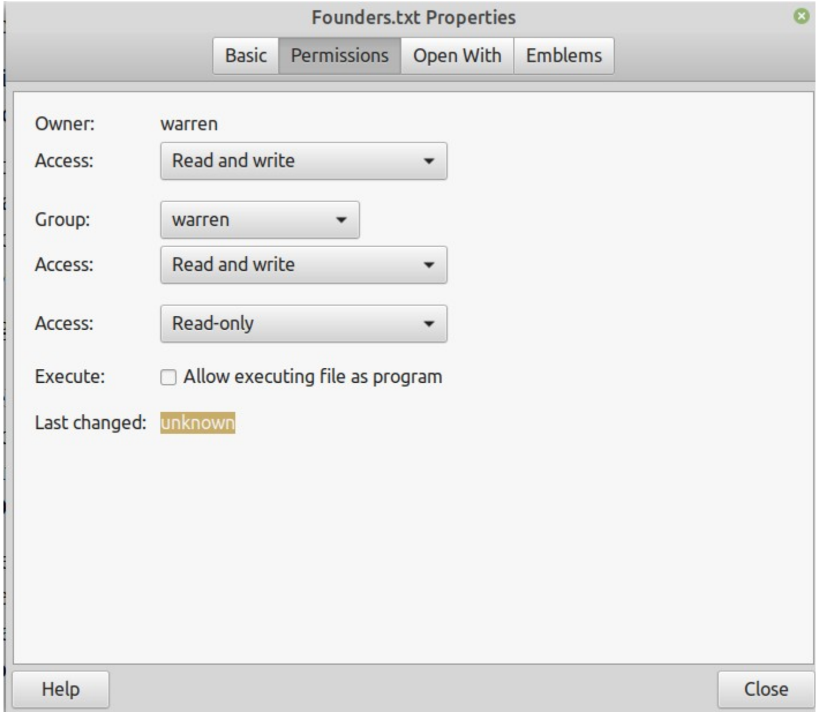
A user is assigned to various groups, depending on the wishes of the system administrator. A user can belong to any number of groups. Everything, including the user, belongs to a group. A user is not required to belong to any group other then his or herself and that is automatic. Such a user could do almost nothing outside of their login account. When a user creates a file or folder, it automatically is owned by that user and only users that belong to that users group would have access to these files. If nobody is a member of your group, ONLY you (and root) can access your files.

To get into the system a hacker would have to know several things, the ID and Password of a user with sudo privileges. "sudo" is a command and a group. If the user belongs to the sudo group, and if the hacker knows the password, then the hacker can get temporary root power to do some things. As you can see, protecting passwords is very important.

If you right-click in any folder or file, and then select file Properties you will get a display similar to **Figure 1.**

Note the Owner and Group are both warren. Warren can read and write and has full access. If I wanted to change this I would have to get root power (sudo) then the owner and groups become pull-down menus and I can select any other user and/or group. As this file stands, warren is the only one who can open, edit, copy, rename, delete the file. Anyhow the point is that each file or directory can be tailored for any desired access.

Remember that almost anyone can look at the Properties, however only a root user can make changes.



Owner: warren
Access: Read and write
Group: warren
Access: Read and write
Access: Read-only
Execute: □ Allow executing file as program
Last changed: unknown

Figure 1

Groups can be a bit confusing, but the system administration should know enough to make the needed selections. It is best explained by saying every program/file belongs to a group and only a user that belongs to that group can execute the program. A program/file can only belong to one group, a user can belong to any number of groups. One command/program is "sudo" and it allows a person to access temporary root status. If a user does not belong to the sudo group, then the command will not work for that user, even if they do know the password.

By now you should be getting the idea that even if a hacker can gain access to the system level, it will do them little good, they can look at a few things but cannot change anything. If the hacker cannot log on with a privileged users ID and password, that person cannot even copy a file

One interesting group is the “www-data” group. It is put on the system when the apache2 server is installed and the system can host web pages. Every file in the /var/www/html/… folders must belong to the www-data group because anybody coming in with their browser are automatically a member of the www-data group (only) and the only files they can access are www-data group files. While on the subject of the web server, anyone who comes in through the web server cannot get into the system in general. Apache will not allow anyone outside of the web page section and it is not possible to back up the directory tree further than the www level. The web server (apache2) is self-contained. Those in the web sites have no indication of what exist outside of their ‘bubble’ and likewise if anybody is using the host computer, there is no clue as to what is going on in the web section, or who or if anyone is even using it.

Groups names are just that, names. They have no power themselves, they are just names. An administrator can easily create more groups. The only power is if a file is assigned to a group, unless you are the file owner, you cannot access the file unless you belong to the assigned group. A root user belongs to all groups and someone invoking ‘sudo’ can override a group name in that sense. For example, I am able to create a group name of “xxx” and it is meaningless. I could create a file and assign it to the xxx group, then only people who have joined the xxx group could access the file.

The user does not need to know much about groups, that is for the system administrator but over the years a lot of groups have been created and they continue to exist, yet the original use has maybe gone away. I think of two, floppy, and tape. Since we no longer use floppy disk or magnetic tape there is nothing assigned to those groups, but they still continue to exist. Some groups are very useful, like adm, backup, dialout as they allow a person to be an administrator, perform backups, send email, and things like that. To remove a group, which is rarely done, remove a group with [sudo] groupdel [group] command <**sudo groupdel floppy**> would remove the floppy group.. There is a lot of ‘deadwood’ I am sure but nobody cares, it is just a few text characters that nobody ever looks at. Since nobody knows what groups are assigned to what, it is best to just leave them alone, but groups are easily added or removed.

In a big system with many users, being the administrator can be a big job, managing the system and users, while on a small system like a Raspberry Pi you might be the only user and just go on as user Pi with root powers (**change the default password**!). If a user tries to open a file but is not authorized, the group name is listed in the properties and once you look and see what you need, you can have the administrator add you to the group.

In real life, I have placed a new MS-Windows installation on line (Internet) and within minutes, I have been probed and attacked. I have been running for years with multiple Linux systems, no firewall, no anti-virus program, and the only attack was while using a browser on the Internet, the browser was hacked, but that could not hurt my system, it just messed up the browser. I uninstalled the browser and deleted the folders it lived in, rebooted and reinstalled the browser and returned to normal.

If you want to see what groups your system has and who is assigned to each, go into the terminal mode (console) and type the following command <cat /etc/group | less> (no brackets) and you can see who belongs to what.

The format is:

groupname: x: number: member(s) separated by commas.
1- The group is the human name,
2- The “x” is in every record but it’s function is hidden, it is tied the shadow password file.
3- The number is the computers identification for the group, used internally.
4- The name(s) following are the members(s) of the group.

Example: **dialout:x:111:joe,sam,tony**
dialout is the group name and joe, sam, and tony are members.

The biggest threat to the security of Linux, is a foolish user that does not protect his login ID and Password. With that information, if the user is a sudo or root user, the system can be totally compromised.

One final reason Linux is so secure; most hackers are MS-Windows people. That is where most of the people and most of the money are. In Linux everything is open source so programs are no mystery and it is so hard to get to the data that it is just not worth the time and effort to try.

It seems ironic but most hackers use special hacking tools created, and running on a Linux computer.

An interesting back-door approach to Assigning a person to one or more groups:

The normal way would be (at the console/terminal level) would be to use the command <sudo adduser [username] [group]. An example: <sudo adduser johndoe www-data> which would add johndoe to the www-data group. If he is already a member, it will say so, or if there is no such group it will tell you that as well.

One ‘can’, from the console level <cd /etc> to change directory to the right place, then invoke the nano text editor <sudo nano group> and you are able to directly edit the file. To add a person to any group just add the logon name to the end. Example: <pulse-access:x:130:warren,k7cwa>. Of course you must be a root user (sudo) user, or nano won’t allow you to save any changes. This process is not the recommended method to manage groups however if there are many changes to be made, it sure makes it a lot easier. As Administrator I made myself a member of ALL groups (135 of them).

The above is all done at the console level (as would be the case on a Raspberry Pi without using a (GUI) graphic user interface). User managing programs are easiest on a normal system.