Laboratory #7: Sysadmin tasks

Unix (420-321-VA) - Fall 2021
Teacher: Tassia Camoes Araujo

Goals:

1. Manage sysadmin power
2. Upgrade your system
3. Secure your system

Instructions

Part I: Sysadmin = power user

The *System Administrator* is the person responsible for installing, configuring and maintaining the system in good health. Sysadmin incorporates many of the tasks we are doing since the beginning of the term, such as managing users, installing applications and managing system resources.

The root user of a Linux system has full administrative power, and because of that, you should avoid working as root for long periods of time, to not risk taking undesired actions by mistake. A good practice is to use the command *sudo* to run only the commands you need as root.

1. Check if you have the *sudo* package installed, and if not, install it.

2. Check if your user is part of the sudo group (in some distros, the user you create at installation time will already be part of sudo). If it is, create another user for this exercise. Add the new user to the sudo group.

3. Open a new session as the user you've just added to sudo, so that your group identity is reloaded.

4. Execute a command that needs administrative power, such as creating another user.

5. Let's suppose you forgot the root password. Considering you have another user in the sudo group, what would be an easy way to recover the root password?

6. Now, imagine you have lost the password with no other user in the sudo group. There must be a way to recover your full power! Research online for a procedure to recover the root password, make sure it works, and describe the step-by-step you took in your own words.

Part II: Upgrade your system

One of the main goals of Linux distributions is to provide software packages to facilitate the task of installing, removing and upgrading software for users. Each project or distribution has a different workflow in regards to packaging and packages life cycle. A *stable distribution* is a version of the OS (the set of all packages) that reaches the desired maturity to be published to the general public. Before getting there, there are intermediate stages that individual packages pass through, resulting in multiple *distributions* co-existing at the same time in public repositories.

In Debian, at any given time there is an *oldstable*, a *stable*, a *testing* and an *unstable distribution* in the repositories. Chapters 3 and 6 of the Debian FAQ explain in details what each *distribution* contains, and the evolution of packages among those.

1. In the GNU/Linux distribution you are using, describe the different *distributions* available, or stages that the package pass through before it gets to a stable version.

Considering that CD/DVD installation images are built and made available for a long period of time, one of the first tasks a sysadmin should perform after installation of an OS is the upgrade of the system for the most updated versions of software. In fact, this is a task that should be done on a regular basis, especially for security updates.

In a Debian system, you can use the command *apt* for that.

2. First use the command *apt update* to update the list of packages available for installation in the selected distribution, then run *apt upgrade* to upgrade all installed packages to their most recent version.

3. Check what is the difference from *apt upgrade* to *apt full-upgrade*, and explain when you should use one or the other.

Alternatively, if you do not use a Debian-based system, research about how to upgrade your system. Describe the commands available, and if you have different options of upgrade.

Part III: Basic security tasks

Security might not be in your mind yet, but it is time to start learning a few tricks. Once you are ready to setup a Linux web server on the Internet, securing your system will be essential.

In part I and II you already removed system vulnerabilities, with a better access control and packages upgrades – many updates are security-related: old versions of software might have well-known security flaws.

1. SSH is the network protocol that allows encrypted communication with a remote host. In UNIX, the SSH program implements the protocol, and you can connect to other machines by running “ssh user@machine” (replace “machine” by the ip address or name – a valid Internet address). Once logged in, you have access to a shell to run all the commands you’ve learned to far. Experiment a SSH connection with your own machine, which you can call “localhost”.
2. Exit your SSH session, as you exit from a session in your terminal.
3. Change the default SSH listening port, which is set to 22 by default. Many cracking attempts by robots will target this port. As root user, edit the file /etc/ssh/sshd_config. Find and uncomment the line “Port 22”. To avoid conflict with another port number already in use, choose a number between 49152 and 65535.
4. Restart the service (as root) with “systemctl restart sshd”
5. Try to execute the same command: “ssh user@machine”. What happens now?

6. Now try to connect again, this time using the new port number.
7. Exit your SSH session.
8. Another common security action is to disable SSH access via root user. Find which is the line in /etc/ssh/sshd_config that allows/blocks root login. Experiment changing its value to “yes”, then to “no”, restarting the service, and trying to login as root after each time.

Part V: Deliverables

1. Open LibreOffice to create your lab document.
2. Include a header with course name, section, your student name, the license of your work (suggestion: one of the creative common licenses).
3. Answer all the numbered questions in parts I, II, and III. Make sure to provide enough details that the reader can follow your practice.
4. Export your file as PDF and upload it to Omnivox.

Good luck!