Users and permissions

instigate-training-center.am

Agenda

- Introduction
- Home directories
- /etc/passwd and /etc/group files
- Creating/deleting users and groups
- File permissions
- Permission types
- Changing file owner and group
- Changing file permissions

Introduction

- Linux is designed to be a multi-user environment
- In an environment with more that one users, it is crucial to have a secure system for deciding which files are yours and who can view or change them
- root user (or superuser) has access to all commands and files on a Linux
- Users currently logged in to the system can be viewed by 'w' or 'who' command
- The current user can be viewed by 'whoami' or 'who i am' commands (also the username is stored in USER environment variable)

Home directories

- Each created account on a system has a separated home directory placed under /home, e.g. /home/john
- The home directory for the superuser is a / directory or /root directory
- Most of the user specific configuration files are stored as a hidden files under home directory

/etc/passwd and /etc/group files

- /etc/passwd file contains various pieces of information for each user account. This is where the users are defined. Contains information such as:
  - Username
  - UID (user ID) and GID (group ID)
  - full name
  - home directory
  - the program to run after login (e.g. the preferred shell)
- /etc/group file contains the information about the existing groups, their IDs and members

Creating users and groups

- **adduser**
  - **adduser [--home DIR] [--shell SHELL] [--uid ID] [-gid ID] user** add a user to the system according to command line options and configuration information in /etc/adduser.conf
  - **adduser user group** - add an existing user to an existing group
- **addgroup group (or adduser –-group group)**
  - add a new group to the system
  - the group is created with no users
- **deluser**
  - **deluser [--remove-home] user** – removes user from the system
  - **deluser user group** – remove a user from a specific group
- **delgroup group** – remove a group from the system

File permissions

- Permissions are defined separately for users, groups, others
  - user – the username of person who owns the file. By default, the user who creates the file become its owner
  - group – the usergroup that owns the file. All users who belong into the group that owns the file will have the same access permissions
  - other – a user who isn't the owner of the file and doens't belong in the same group the file does
- Output of the 'ls -l' command

  -rw-r--r-- 1 student student 12408 2010-25-03 18:44 file

Permission types

- Read permission
  - on a regular file means that the file can be opened and read
  - on a directory means that the contents of the directory can be listed
- Write permissions
  - on a regular file means that the file can be modified (new data can be written to the file)
  - on a directory means that files can be added, removed or renamed in the directory
- Execute permissions
  - on a regular file means that the file can be executed as a program or a shell script
  - on a directory allows to access files in the directory and enter it

Changing file owner and group

- **chown [OWNER][:GROUP] [-R] FILE**
  - e.g. chown student /home/john/file.txt
  - e.g. chown student:student /home/john/file.txt
  - e.g. chown :root /home/student/file.txt
- **chgrp GROUP FILE**
  - e.g. chgrp student /home/john/file.txt
  - equivalent to **chown :GROUP FILE**

Changing file permissions

- Permissions are changed by chmod command
  - **chmod (ugoa) (+-=) (rwx) FILE**
- Add execute permissions for group
  - **chmod g+x file.txt**
- Add both write and execute permissions for the file's owner
  - **chmod u+wx file.txt**
- Remove execute permissions from both owner and group
  - **chmod ug-x file.txt**