IT Scripting and Automation

System Administration
&
System Administration in a Unix/Linux environment

Lecturer: Art Ó Coileáin

System Administrator

General definition:

- A person who is responsible for setting up and maintaining the system or server is called a **System Administrator**.

- Usually System Administrators are part of the information technology department.

System Admin common duties

- The duties of a System Administration are wide-ranging
- They usually include:
  – Installing (Apps, OS’s, HW etc.),
  – supporting and maintaining server(s) or other computer systems, and
  – planning for and responding to service outages.

System Administrator

- A System Admin is usually responsible for the following tasks:
  - User Administration: setup and maintaining accounts
  - Maintaining systems
  - Verify that peripherals are working properly
  - Quickly arrange repair for hardware in occasion of hardware failure
  - Monitor system performance
  - Create file systems
  - Install software
  - Create a backup and recovery policy
  - Monitor network communication
  - Update systems (OS and applications)
  - Implement policies for the uses of the computer systems and network
  - Setup security policies for users (strong computer security policies)
  - Documentation

System Administrator

- Many of System Admin Tasks can be automated using Perl/Python or Shell Scripts such as:
  - Create new users
  - Resetting user passwords
  - Lock/unlock user accounts
  - Monitor sever security
  - Monitor special services or resources.
- The system administrator account (root in Unix-like systems) has full access (unrestricted).
- System Administrators are not Developers, but they must understand the behaviour of software in order to deploy it and to troubleshoot problems.
- They should be good at several programming/scripting languages used for scripting or automation.

SA Best Practices

- State knowledge is critical:
  - An understanding of the current state of your hardware, OS, and users give you a baseline for problem diagnosis, security management, and planning for growth.
  - Tools: log analysis, system monitors and alarms.
- Communication is important:
  - Between admins and users, between admins and management, with vendors and among fellow admins.
  - Tools: e-mail lists, user education share, policy statements.
- Standardise and automate:
  - Develop habits to handle smaller tasks in more automated fashion.
  - Tools: scripting languages.

SA Best Practices (2)

- Document everything:
  - User documentation leads to more informed and happier users; maintenance documentation leads to more consistently managed systems and more quickly debugged system problems.
  - Tools: weblogs, asset databases, ...
- Software application maintenance:
  - Keep a balance between functionality and stability. Some upgrades may require extensive testing, (and rollback plans).
  - There is no substitute for planning.
- Security:
  - It affects all aspects: hardware, software, network. To develop security polices is critical.
  - Tools: VPN, TCP wrappers, port scanners, etc.

SA Best Practices (3)

- Reliability:
  - A primary goal of many these practices is to ensure the system availability for users, despite an every increasing complexity such as growing number of users, and security threats.
  - **Tools**: redundant hardware, load-balanced and failover systems, data backup equipment, procedures and polices are important, as are the disaster recovery plans. Such plans must be tested and reviewed regularly.

System Administrator Roles

- The goal of effective system administration:
  - To provide a stable ICT environment, enabling users to conduct their business with ease and efficiency; While taking into consideration the demands of:
    - security,
    - other users' needs,
    - the inherent capabilities of the system, and
    - the realities and constraints of the human community in which they reside.

Continue.....

System Administrator Roles

….continue

- Successful System Administration is a combination of careful planning and habit.
  - The key to handling a crisis lies in having foresight, and taking the time to anticipate and plan accordingly for the emergency scenario(s) which arise.

System Administrator Roles

- Many crisis can be prevented by a determined devotion to carry out all the careful procedures you have designed.
- E.g.:
  - Changing the Admin password (root) regularly
  - Faithfully backups (*and test restores*)
  - Close monitoring systems logs
  - Logging out & clearing the terminal screen as a ritual
  - Testing every change several times before letting it loose
  - Sticking to policies you have set for users’ benefit

Basic System Admin Strategy

- Know how things work
- Plan it before you do it
- Make it reversible (backups and rollback plans help)
- Make changes incrementally
- Test, test, test before you unleash it to the world

IT Scripting and Automation

System Administration
in
a Unix/Linux environment

Superuser(Unix-like systems)

- The superuser refers to a privileged account with unrestricted access to all files and commands. The username of this account is root.
- Many administrative tasks and their associated commands require superuser status.
- There are two ways to become superuser:
  - first to log in as root directly and
  - the second way is to execute the command su while logged in to another user account.
- To exit from superuser account use exit or Ctl-D

Superuser(Unix-like systems)

- To set or change the superuser password, become a superuser and execute one of the following commands:
  - **passwd** or **passwd root**

Important recommendations:

- It is recommended to avoid logging in directly as root, instead use **su** command only as necessary.
- **Never** leave any logged-in session unattended

Superuser(Unix-like systems)

Running a command as Root:

- Single command can be run as root. It allows to fix something quickly. E.g.:
  - $ su root -c “command”
  - $ su root -c “vim /etc/hostname”

Superuser: sending Messages

- If you need to send a message to every user on the system. The wall command allows the administrator to send a message to all users simultaneously.
  - **$ wall**
  - Followed by the message, terminated with Ctrl-D
- Message of the Day: Login time is a good time to communicate certain types of information to users.
  - The file /etc/motd is the system’s message of the day.
  - You can use it to display system-wide information such as maintenance schedules, news, announcements or anything else considered important and appropriate to your system.
- The content of the file /etc/issue is displayed immediately before the login on unused terminals.

Root Password

- Most of your administrative team do not need to know the root password.
- It should be something that is secure.

Characteristics of a good password:

- The most important characteristic of a good password is length.
- The most secure type of password consist of a random sequence of letters, punctuation and digits.
- It may not be optimally secure if administrator write it down or type it slowly.
- “Balance” and something that can be remembered.

Root Password

Changing the root password:

- Change root password at least every three months,
- every time someone who knows the password leaves your site or organisation,
- when you think security may have been compromised.

General Ownership Rules in the System

- Objects have owners. Owners have control over their objects.
- You own new objects that you create.
- Only root can act as the owner of any object in the system.
- Only root can perform sensitive administrative operations.

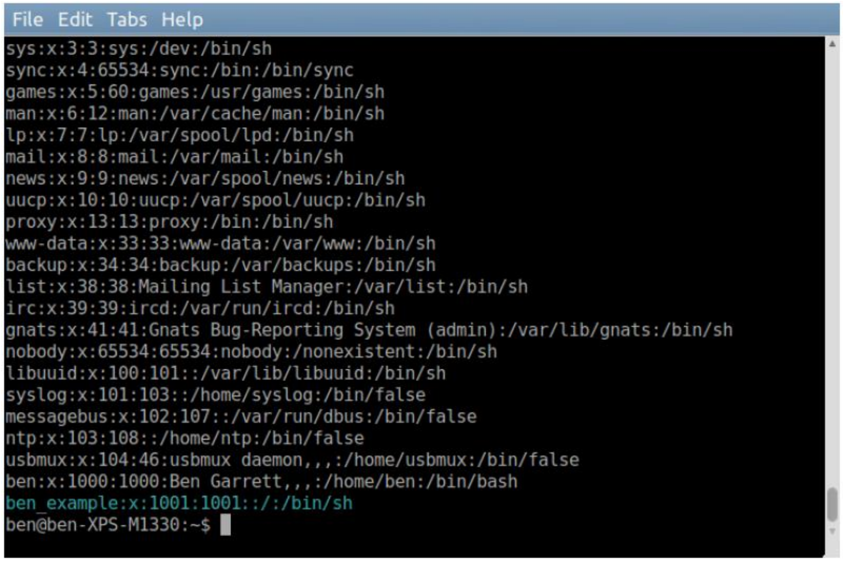
General Ownership Rules in the System

Group(s) and Ownership:

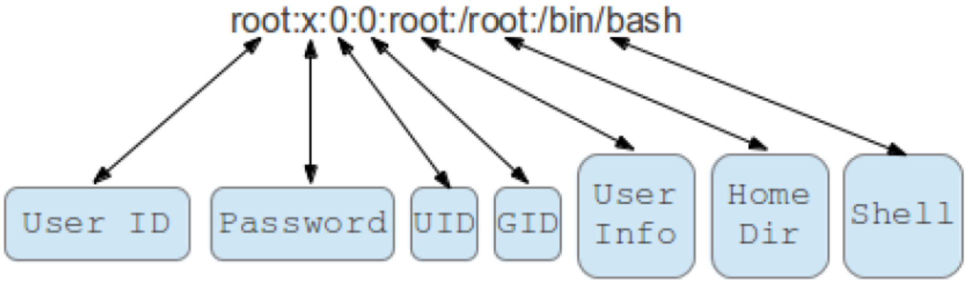
- Although the owner of a file is always a single person, many people can be group owners of the file, as long as they are all part of a single group.
- Groups are traditionally defined in the path /etc/groupfile.
- These days group information is more commonly stored on a NIS or LDAP server on the network.

File: /etc/passwd

- This file stores essential information, which is required during login process, i.e., user account information. It is a text file, that contains a list of system's accounts and useful information like user ID, group ID, home directory, shell, etc. All the fields are separated by a colon (:) and it contains one entry per line.
- It should have general read permissions as many utilities, like **ls** command use it to map user IDs to user names.
- Your encrypted password is NOT stored in this file.

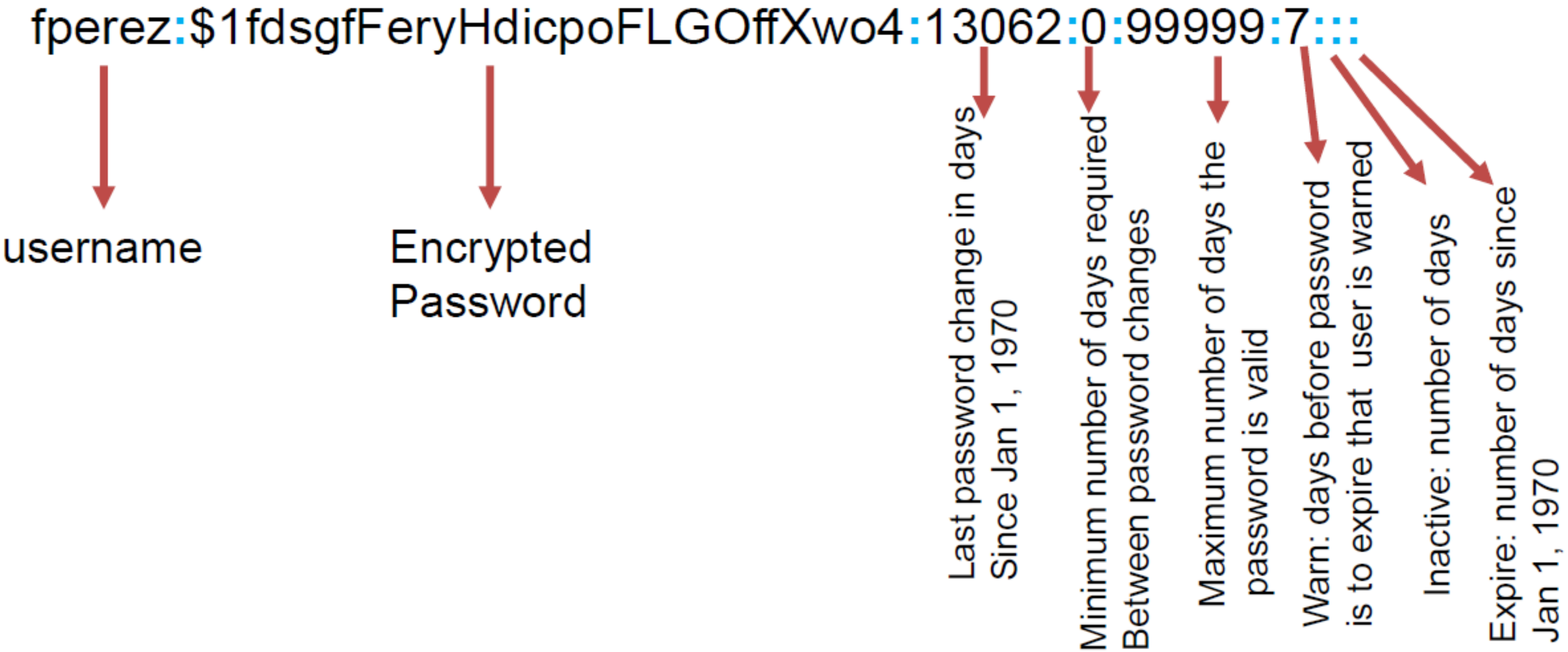


```
sys:x:3:3:sys:/dev:/bin/sh
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/bin/sh
man:x:6:12:man:/var/cache/man:/bin/sh
lp:x:7:7:lp:/var/spool/lpd:/bin/sh
mail:x:8:8:mail:/var/mail:/bin/sh
news:x:9:9:news:/var/spool/news:/bin/sh
uucp:x:10:10:uucp:/var/spool/uucp:/bin/sh
proxy:x:13:13:proxy:/bin:/bin/sh
www-data:x:33:33:www-data:/var/www:/bin/sh
backup:x:34:34:backup:/var/backups:/bin/sh
list:x:38:38:Mailing List Manager:/var/list:/bin/sh
irc:x:39:39:ircd:/var/run/ircd:/bin/sh
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/bin/sh
nobody:x:65534:65534:nobody:/nonexistent:/bin/sh
libuuid:x:100:101::/var/lib/libuuid:/bin/sh
syslog:x:101:103::/home/syslog:/bin/false
messagebus:x:102:107::/var/run/dbus:/bin/false
ntp:x:103:108::/home/ntp:/bin/false
usbmux:x:104:46:usbmux daemon,,,:/home/usbmux:/bin/false
ben:x:1000:1000:Ben Garrett,,,:/home/ben:/bin/bash
ben_example:x:1001:1001::/:/bin/sh
ben@ben-XPS-M1330:~$
```



File: /etc/shadow

- This file stores actual passwords in encrypted format for user’s accounts with additional properties. All the fields are separated by a colon (:). It contains one entry per line for each user listed in **/etc/passwd** file.



```
fperez:$1fdsgfFeryHdicpoFLGOffXwo4:13062:0:99999:7:::
```

username

Encrypted
Password

Last password change in days
Since Jan 1, 1970

Minimum number of days required
Between password changes

Maximum number of days the
password is valid

Warn: days before password
is to expire that user is warned

Inactive: number of days

Expire: number of days since
Jan 1, 1970

Pseudo-Users other than root

- Root is generally the only user with special status in the eyes of the kernel, but several other pseudo-users are defined by the system.
  - You can identify these sham accounts by their low UIDs, usually less than 100. Most often UIDs under 10 are system accounts, and UIDs between 10 and 100 are pseudo-user associated with specific pieces of software.
- It is customary to replace the encrypted password field of these special users in **/etc/shadow** file with a star (*****) so that their accounts cannot be logged in to.
- Set **/etc/passwd** file their shells to **/bin/false** or **/bin/nologin** as well, to protect against remote login exploits that use password alternatives such as SSH key files.