IT Scripting and Automation

Users
Linux/Unix Systems

Lecturer: Art Ó Coileáin

Users

- Adding or removing users is a routine chore on most systems and these tasks may be boring; most administrators tweak the tools provided with the OS to automate the process (using scripts) so they can delegate the actual work.

- Today's enterprise environments need not just a tool for adding users but also a tool for managing users and passwords across the entire computing environment. Some widespread Directory services used are:
  - Microsoft's Active Directory
  - OpenLDAP
  - "389 Directory Server" previously "Fedora Directory Server"
  - Red Hat Directory Server

Users

- Account hygiene is a key determinant of system security. Infrequently used accounts are prime targets for attackers.
- Login names must be unique and depending on the OS may have length and character set restrictions.
- Login names should be easy to remember, so random sequences of letters do not make good login names. Avoid nicknames, even if your organisation is informal.

Rules for forming login names

System	Len	Character set	First	Special rules
Linux	32[a]	a-z0-9_-	a-z_	Some distros are more generous
Solaris	8[b]	A-Za-z0-9+.-_	A-Za-z	At least one lowercase letter
HP-UX	8	A-Za-z0-9_	A-Za-z	
AIX	8[c]	POSIX; no spaces, quotes, or #,=/?\	not -@~	Not all uppercase letters Not "default" or "ALL"

a. Although Linux allows 32 characters, legacy software (e.g., **top** and **rsh**) expects 8 or fewer.
b. Is being increased.
c. Can be changed in AIX 5.3 and later, see opposite page.

Encrypted Password

- Modern systems put a placeholder for the encrypted password in the **/etc/passwd** file and then prompt the user for a real password on first login.
- **chfn** command is used to record personal information about each user.
  - Full name
  - Office or room number
  - Work phone
  - Home phone
- **chfn** command understands only /etc/passwd file, so if you use LDAP or other directory service for login information **chfn** may not work.

/etc/group File

- The **/etc/group** file contains the names of Unix groups and list of each group's members.
- In some cases, group names should be limited to eight characters for compatibility.
- Only Linux has real support for group passwords. The encrypted form is stored in the **/etc/gshadow** file.

/etc/group

```
system:!:0:root,pconsole,esaadmin
staff:!:1:ipsec,esaadmin,trent,ben,garth,evi
bin:!:2:root,bin
sys:!:3:root,bin,sys
adm:!:4:bin,adm
nobody:!:4294967294:nobody,lpd
```

Groups

Usage:

- To add a new group the command **groupadd** is used:
  - **$ groupadd *groupname***
- To add an existing user to an existing group, the command **usermod** is used:
- **$ usermod –G *groupname username***
  - This command will add the user to a secondary group. Notice: **G** is uppercase.
- **$ usermod –g *groupname username***
  - The parameter -g(lowercase)means: Use this groupnameas the primary group.

Adding Users: The Basics

- Before creating an account, it is important that the user sign and date a copy of your local user agreement and policy statement.
- The process of adding a new user consists of several steps required by the system, two steps that establish a useful environment for the new user, and several extra steps for your own convenience as an administrator:
- Necessary steps to add a new user:
  - Have the new user sign your policy agreement
  - Edit the **passwd** and **shadow** files to define the user's account
  - Add the user to the **/etc/group** file (sometimes not necessary)
  - Set an initial password
  - Create (**chown**, and **chmod**) the user's home directory
  - Configure roles and permissions (if needed)

Adding Users: The Basics (2)

- For the user:
  - Copy default startup files to the user's home directory
  - Set the user's mail home and establish mal aliases (if needed)
- For the Administrator:
  - Verify that the account is set up correctly
  - Add the user's contact information and account status to your database
- This task list cries out for a script or tool. Fortunately the **adduser** command (*which is a Perl script*) provides this functionality in Linux.
  - **$ adduser username**
- You must be root to add a user.
- To change the password, the command **passwd** is used.
  - **$ su –c passwd username**

Adding Users: The Basics (3)

- To add a user using **useradd** command:
  - $ **useradd ben**
- This command creates the following entry in /etc/passwd:

```
john:x:1001:1001:John Smith,,083123456,:/home/john:/bin/bash
sean:x:1002:1003:Sean ONeill,,089123456,:/home/sean:/bin/bash
ben:x:1003:1005::/home/ben:/bin/sh
```

- **useradd** disables the account by putting an x in the password field. You must assign a password to make the account usable.
- A more realistic example is shown below:

$ useradd –c "Ben Kenny" –d /home/ben –g students2015 –m –s /bin/bash ben

-c -> comment **-d** -> home directory **-g** -> primary group

-m -> creates home directory if it does not exit

-s -> defines shell

Adding Users: The Basics (4)

```
john:x:1001:1001:John Smith,,083123456,:/home/john:/bin/bash
sean:x:1002:1003:Sean ONeill,,089123456,:/home/sean:/bin/bash
ben:x:1003:1002:Ben Kenny:/home/ben:/bin/bash
```

- The previous command created the password entry, assigned UID and the corresponding entry in **/etc/shadow**
- **useradd** command also added **ben** to the appropriate groups in **/etc/group**, creates the directory **/home/ben**, and populates it from the **/etc/skel** directory

Removing Users

- When a user leaves an organisation, that user's login account and files should be removed from the system. This procedure involves the removal of all references to the login name that were added. If it is removed by hand use the following checklist:
  - Remove the user from any local user database or phone list
  - Remove the user from the aliases file or add a forwarding address
  - Remove the user's crontab file and any pending at jobs or print jobs
  - Kill any of the user's processes that are still running
  - Remove the user from the passwd, shadow, group, and gshadowfiles
  - Remove the user's home directory
  - Remove the user's mail spool (if mail account)
  - Clean up entries on shared calendars, reservation systems, etc.
  - Delete or transfer ownership of any mailing list run by the deleted user

Removing Users (2)

- Most of the systems have a **userdel** command that automates the process of removing a user. By default, it may not do quite as thorough as job as you might like (*User-related information can be stored quite a number of places*).
- Linux distributions (*Debian-like*) provide a Perl script that calls the usual **userdel** and undoes all the things that **adduser** does. This script is called **deluser**.
- These scripts are located in: **/usr/sbin/** directory

Disabling Logins

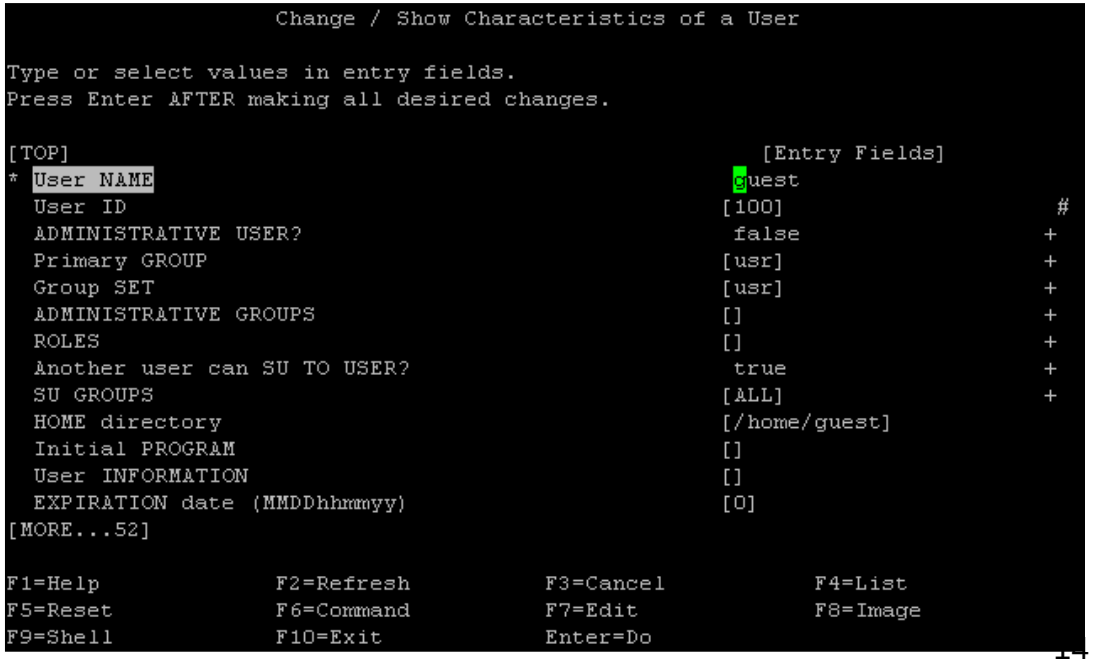
- If a user's login must be temporarily disable. A straight forward way to do this is to put a star or some other character in front of the user's encrypted password in the **/etc/shadow** file. This measure prevents most types of password-regulated access because the password no longer decrypts to anything sensible. However there may be commands that do not necessarily check the system password.
- On Linux distributions, the use of **usermod -L *username*** and **usermod -U username** command provide an easy way to lock and unlock passwords. They will put or remove and **!** In front of the encrypted password in the **/etc/shadowfile**.
- On some Unix systems such as Solaris the command will be **passwd –l *loginame* or passwd –u *loginame***

Managing Users with Tools

- Unix systems such as HP-UX and AIX provide a comprehensive system administration tools that knows how to manage users:
- In AIX it's SMIT, the System Management Interface Tool, and
- In HP-UX it's now called SMH, the System Management Home (previously called SAM-System Administration Manager in earlier HP-UX releases).
- These tools has screens for adding and managing users, either with a windows-based GUI or with a terminal interface.

SMIT -IBM AIX

(terminal interface)



Reducing Risk with PAM

- Pluggable Authentication Modules (PAM). They centralise the management of the system's authentication facilities through standard library routines so that programs like: **login**, **sudo**, **passwd**, and **su** do not have to supply their own tricky authentication code.

- PAM reduces the risk inherent in writing secured software, allows administrators to set site-wide security polices, and defines an easy way to add new authentication methods to the system.