User Management

- Need to maintain access by adding and removing users
- Need to decide who should be allowed to access a machine – a management function
- Infrequently used accounts or accounts with easily guessable passwords can be exploited by hackers

`/etc/passwd` file

- Contains list of users allowed to access the system
- Consulted by many applications, including the login daemon to authenticate the user
- Seven fields separated by columns
  1. Login name or username
  2. Encrypted password
  3. UID
  4. Default GID
  5. GECOS information
     - General Electric Comprehensive Operating System
     - An OS developed in 1962-64
  6. Home directory
  7. Default shell
- May be shared among systems using NIS
- Login name or user name
  - Identifies a user to the system
  - Must be unique for the system
  - Usually no more than eight characters; limited to eight characters if you use NIS
    - Usernames on hoare are longer than eight characters
    - Applications like ps only use the first 8 characters of username
  - Traditionally limited to alphanumeric characters but modern OSs allow the use of special characters, except colon and newline
  - Case sensitive, traditionally all lowercase
  - Should be easy to remember, and also easy to guess
    - How do you rank the user names in use on admiral? On hoare?
  - Use mail aliases in file `/etc/mail/aliases` to resolve long names
  - Use uniform names across machines
    - Take proper precautions to secure machines in case two machines have different persons with same login names
    - File `/etc/hosts.equiv` should not have a `+` sign by itself on a line
- Password
  - Kept in encrypted form, using DES
  - Can be copied from another machine if needed

    - That is how your initial password on `hoare` is from `admiral`
  - A `*` in the field disables unauthorized use
    - If you put a `*`, make sure that you disable trusted access through `.rhosts` file, `.ssh` keys, or other means as well
  - Password limited to 8 characters (even if you use more), with the encrypted version coming to 13 characters regardless of the number of characters in the original version
    - Encrypted with random 2-character "salt"
    - Two users using the same password will have different encryption due to salt
  - MD5-based passwords
    - Can be any length; not limited to 8 characters
    - Encrypted version is always 31 characters
    - First three characters in encrypted form are always `$1$`
  - Modern systems hide encrypted passwords by using shadow file
    - Solaris requires passwords to be maintained in `/etc/shadow`
- UID
  - 32-bit integer, but suggested limit to 15-bit unsigned for compatibility, always more than 100 for normal users
  - Linux supports 16-bit unsigned
  - UID of root is 0
  - Do not recycle UIDs
    - Avoids confusion in backup files saved on the basis of UID
  - Should be unique across organization, with each user having uniformly same UID across machines
    - Avoids problems with NFS mounts
    - Technical and political problems can be avoided by creating a central database of UIDs, or by assigning range of UIDs to groups
- Default GID
  - Unique 16/32 bit integer
  - GID 0 is for `root` or `wheel`
  - GID 1 is for `daemon`
  - Groups defined in `/etc/group`
  - User can change group by using `newgrp` command
- GECOS field
  - No well-defined syntax
  - Originally had login information needed to transfer batch jobs from Unix to GECOS
  - Information changed via `chfn` or `passwd -g`
  - Disabled to prevent misuse (like obscene messages)
- Home directory
  - Fixed and can only be changed by root
  - If home directory is mounted via NFS, it could be unavailable due to network or server problems
- Login shell

  - Typically a command interpreter but can be any program
  - Can be changed by `chsh` or `passwd -e`
  - Authorized shells can be specified in `/etc/shells`

`/etc/shadow` **file** on Solaris and RH

- Required under Solaris; used through the `shadow` package on Red Hat
- Readable only by superuser
- One line for each user, with nine fields
  1. Login name
  2. Encrypted password
  3. Date of last password change
  4. Minimum number of days between password changes
  5. Maximum number of days between password changes
  6. Number of days in advance to warn users about password expiration
  7. Number of inactive days before account expiration
  8. Account expiration date
  9. Flags (empty – reserved for future use)

`/etc/group` **file**

- Contains names of groups and list of group members
- Each record has four fields, separated by colons
  1. Name of group
  2. Encrypted password, rarely used – leave empty
  3. GID: Group's unique numerical id within the system
  4. Comma-separated list of users in the group
- Maximum allowed GID is 32-bit but recommended value is below 60000
- Password will restrict entry into the group by using `newgrp` command
- No spaces are allowed in the list of members

Adding users

- Get the user to sign an agreement
- Steps in adding a user
  - Edit `passwd` and `shadow` files to establish account
  - Set an initial password
  - Create the home directory
  - Copy default startup files to user's home

  - Set mail home and establish mail aliases
  - Add user to `/etc/group` if needed
  - Configure disk quotas
  - Verify that the account is correctly set up
- Edit `passwd` and `shadow` files through `vipw`
- Initial password can be set by using the `passwd` command by root
  - Prospective passwords are checked for guessability if you use `npasswd` (freely available) instead of standard `passwd` command
  - Never leave the password field empty for a user
- Create home directory for user
  - Make sure that you change ownership and group ownership of home to user
- Copy default startup files and change their ownership to user as well
  - Maintain default startup files in `/usr/local/lib/skel`
  - Vendor-supplied startup files are in `/etc/skel`
- Set up disk quotas using `edquota`

Removing users

- Checklist
  - Set user disk quotas to zero
  - Remove user from local user databases or phone lists
  - Add a forwarding address for user in `aliases` file
  - Remove user's `crontab` file and any pending jobs
  - Kill any user processes that are still running
  - Remove user's temporary files in `/var/tmp` or `/tmp`
  - Remove user from `passwd` and `group` files
  - Remove user's home directory
  - Remove user's mail spool
- Run `quot` command to verify that all files belonging to user (via UID) have been removed
  - `quot` does not require quotas to be enabled
  - `quot` only works on local file systems
  - Find orphaned files using the command

```
find /home -nouser -print
```

Disabling logins

- Put an asterisk in the encrypted password field
- Users can log in via other means that do not require a password, such as `rlogin` and `ssh`
- You can also replace user's login shell with a pseudo shell

  - Pseudo shell should not be listed in `/etc/shells`, to avoid `ftp` access
  - Mail may not get delivered because of unauthorized shell
    * Can be fixed by adding a fake shell named `SENDMAIL/ANY/SHELL/` to `/etc/shells`
  - Just print a message to convey that account is locked

Vendor-supplied account management utilities

- `admintool`
  - Menu-based sys admin package in Solaris
  - Must be a member of group `sysadmin` to run it
- `useradd`, `usermod` and `userdel`

GUI and menu-based tools

- Advantages
  - Provide a quick start to sys admin
  - Get things done while still learning about the OS
  - Get syntax right for commands with a lot of complex options
  - Make some operations more convenient by hiding the details behind a single menu screen
- Disadvantages
  - Typing is faster than running an administrative GUI
  - Not all commands are available in GUI
  - Can slow down the learning process
  - Cannot be easily automated, unless the admin learns the commands behind the GUI