COMP09024 Unix System Administration

Lecture 3: User Accounts and Authentication

Duncan Thomson/Hector Marco

UWS

Trimester 1 2020/21

Outline

3.1 Users and Groups

Users, Usernames and UIDs

- Every user in Unix is identified by a *user ID* (UID)
  - Numbers (15, 16, 32b) which identify users in the kernel
  - UID 0 has a username of `root`, and has full administrative privileges
  - Often other low-numbered UIDs (below 100, 500, or 1000) are reserved for other 'system' users
- Users identify themselves using a *username*, used for logging in, user-oriented output of commands (eg `ls -l`, email addresses, etc)
  - Usernames are alphanumeric starting with a letter
  - Depending on Unix version, other characters may be allowed
  - Depending on Unix version, may have maximum length (8)
- File `/etc/passwd` maps between usernames and UIDs

The /etc/passwd File

- The /etc/passwd file contains lines with seven fields, separated by colons (:)
  1. Username
  2. Encrypted password (or x if shadow passwords are being used; * if the account is disabled)
  3. UID — a number
  4. GID (group ID) of user's primary group — another number
  5. User's full name and other details (the GECOS field)
  6. User's home directory (usually in /home)
  7. User's default shell (usually /bin/sh or /bin/bash)
- A typical entry from /etc/passwd might look like:

```
bob:x:1010:1010:Bob Bain,,,:/home/bob:/bin/sh
```

Groups and /etc/group

- Groups are listed in /etc/group, contains four colon-separated fields:
  1. Group name (same limitations as usernames)
  2. Group passwd (often disabled using *)
  3. GID (group ID)
  4. Comma-separated list of usernames who are group members
- In addition to the listed group members, many Unix systems use 'User Private Groups':
  - Each user is a member of a group only containing themselves as the primary member
  - This is designed to ease working in groups

Shadow Passwords and /etc/shadow

- Passwords were originally stored encrypted (with salt) in the (world-readable) /etc/passwd file
- However as systems became more powerful, brute-force password cracking became much more of a threat
- Modern Unix systems usually store the encrypted passwords (usually using a stronger encryption algorithm) in the unreadable /etc/shadow file
- This is a colon-separated 8-field file:
  - First two fields are username and encrypted password
  - Next six fields are used for *password aging*: date of last password change; min password age; max password age; warning period; inactivity period; account expiry date
  - Last field is reserved
- /etc/gshadow can be used for group passwords, if used

3.2 Checking and Changing User Information

Checking Your Identify with `whoami` and `id`

- The `whoami` command prints the username of the current user
- More detailed information comes from the `id` command, which by default gives:
  - Current username and UID
  - Current group name and GID
  - A list of all groups of which the user is a member
- Various flags to `id` can limit the output to specific content only

Changing Roles with `su` and `newgrp`

- `su` allows changing of the current UID (or become superuser)
  - Requires a username as a parameter (defaults to `root`)
  - Asks for the password for that user (unless you're `root`!)
  - `-l` flags creates a full new login environment
- `newgrp` allows changing the current group for a user
  - Requires the group name as a parameter (defaults to group in `/etc/passwd` file)
  - Will ask for a password if the user is not listed as a group member (unless they are `root`)
  - This affects the group owner of files created thereafter

Other Users: who, pinky, write, wall and mesg

- The who command lists the users who are logged in
- Some other less important commands allow finding out more about, and interacting with, other users
- last shows when users were last logged on
- w (watch) shows what commands users are running
- pinky provides info about the people logged
- write allows sending of 'instant messages' to logged in users (end the message with Ctrl-D)
- wall (write all) sends message to all logged in users
- mesg turns on and off such message reception (y or n)
- talk provides an interactive 'chat' session

Changing Passwords with `passwd`

- The `passwd` command is used to change passwords
- This will by default ask for your current password, followed by a new password, twice
- Note that normally nothing is echoed to the screen while passwords are typed
- The `root` user can use `passwd` *`username`* to change other user's passwords
  - Even `root` needs to crack the password to know it.
- There may be system restrictions on password length and contents, and/or password aging policies in effect

chsh and chfn

- As well as passwords, users can change other information held about them in the /etc/passwd file
- The chsh command allows changing the default *shell* — this is the command line interface (usually /bin/sh or /bin/bash) which is used to enter commands
- The chfn command changes a user's 'pinky' entry
  - These are the data held in the 5th field in /etc/passwd
  - Sometimes known as the GECOS field
  - Only root can change the user's full name, but user's can change office number, phone number, and home phone
- Usually a password is asked for before changing anything

3.3 User and Group Management

Adding Users with `useradd`

- `useradd` adds the specified username to the system
- Default operation depends on `/etc/login.defs` settings, and requires options for some information
- `-d` specifies home directory
- `-m` make home directory with default 'skeleton' files
- `-r` create a system user (with a low UID)
- `-u` specify UID explicitly
- `-g` specifies user's login group (in `/etc/passwd`)
- `-U` create a user-private group as login group
- `-G` lists groups of which user is a member
- `-c` comment field (name and contact details)
- `-s` specifies the shell
- Password should be set separately using `passwd`

`usermod` and `userdel`

- `usermod` changes settings for a given user:
  - Many flags are as for `useradd` (eg `-d`, `-c`, `-u`, `-g`, `-s` and so forth)
  - `-l` allows changing of username
  - `-m` moves home directory and contents to new home directory (given with `-d`)
  - `-L` and `-U` lock/unlock account (by putting `!` in front of the encrypted password)
- The `userdel` command deletes a given user account
  - `-f` forces removal even if user is logged in
  - `-r` removes user's home directory and mail files

Managing Groups: `groupadd`, `groupmod` and `groupdel`

- `groupadd` adds a group with the specified name
  - `-g` gives numeric GID
  - `-r` creates a system group (with a low GID)
- `groupmod` modifies the specified group
  - `-g` gives new GID
  - `-n` specifies a new name for the group
- `groupdel` deletes the given group

Password Management with `passwd` and `chage`

- `passwd` changes a user's password (only the new password is required when run as `root`)
- The `chage` command allows management of password aging policies for an account:
  - `-m` and `-M` set min/max time between password changes
  - `-W` and `-I` specify warn/inactive times
  - `-E` allows setting an expiry date on the account
  - `-l` lists current password age status on an account
- Don't expect users to change their passwords too often!
- Educate them about strong passwords instead (software is available to enforce strong passwords)
- In general, long easy-to-remember passwords are better than short complicated ones (<10 characters)

The Debian `adduser` and `addgroup` Commands

- Debian GNU/Linux provides an additional `adduser` command:
  - Syntax is `adduser` *`username`*
  - Interactively prompts for user's name, details and password
  - Home directory with skeleton files is automatically created
  - A range of flags allow other details to be set if required
- It can also be used to add a user to a group: `adduser` *`username groupname`*
- The `addgroup` command similarly adds groups
- Both commands follow Debian's policy on UIDs and GIDs, and can be configured from `/etc/adduser.conf`
- The `--system` flag creates system users and groups

3.4 Pluggable Authentication Modules

PAM Overview

- Authentication in Linux is done by comparing encrypted passwords (`/etc/shadow`)
- Each program (`SU, password, loging, SSH, ...`) requires to implement its own authentication mechanisms.
- PAM mainly purpose is to solve this and have a common and multiple ways to authenticate users.
- PAM was developed by Sun Microsystems developed.
- This has been adopted in most Unix versions
- Different authentication / authorisation methods are implemented as dynamically linked libraries

PAM Groups and Stacks

- There are four Management Groups:
  1. **Authentication**: for credential-based authentication
  2. **Account**: for authorisation linked to account
  3. **Password**: for updating credentials
  4. **Session**: for setting up a user's login session
- In a management group for a service, a number of modules can be 'stacked' together
- Flags then indicate what combination of modules in a stack is required for successful access
- Configuration for individual services are in files in `/etc/pam.d/` (or all in `/etc/pam.conf`)

PAM Modules

- Modules (libraries) are in `/lib/security` or similar
- These include modules designed for authentication...
  - Standard Unix methods using `/etc/passwd`
  - Network security protocols (LDAP, Kerberos, RADIUS, SMB)
  - Database servers
  - Smart cards or USB keys
- ...but also modules for other purposes, eg:
  - Checking time or location of login
  - Changing passwords
  - Checking password strength when changed
  - Mounting directories or setting resource limits on login

3.5 Multiple System Admins

The sudo System

- On large, complex systems, many people may require the root password to manage various subsystems
- An alternative to this is the sudo command
- At its most basic, it allows **specified users** to execute **specified commands** as root using sudo *command*
- Authentication (of the user running sudo) is required
- The /etc/sudoers file contains a list of the user and command combinations which are permitted
- Note that the popular Ubuntu Linux distribution does not have a standard root user — instead, the first 'normal' user created has full administrative privileges via sudo
- sudo -s provides a root shell on such systems

Special Variants of vi

- Since the vi editor is often used to edit essential system files, it is necessary to prevent system administrators from overwriting one another's changes
- A range of special variants of vi exist for this:
  - vim Vi IMproved, it is upwards compatible to Vi.
    - Supports syntax highlighting, code folding, etc.
    - The screen can be split for editing multiple files.
    - Support for plugins, and a long et cetera.
  - vipw is used for manually editing /etc/passwd (or /etc/shadow if the -s flag is used)
  - vigr is used for manually editing /etc/group (or /etc/gshadow if the -s flag is used)
  - visudo is used for manually editing /etc/sudoers

Summary

- UIDs and usernames, GIDs and groups
- `/etc/passwd`, `/etc/group` and `/etc/shadow`
- User/group identity: `whoami`, `id` and `who`
- Changing user/group: `su` and `newgrp`
- Changing user information: `passwd`, `chsh`, `chfn`
- User administration: `useradd`, `usermod`, `userdel`
- Group administration: `groupadd`, `groupmod`, `groupdel`
- Password management with `passwd` and `chage`
- Debian user admin commands: `adduser` and `addgroup`
- PAM: management groups, stacks and modules
- The `sudo` system
- `vi` and its variants