Contents

1 Role of the OS

- Compactly combine:
  - Identification
  - Authentication
  - Access control
  - Auditing
- User accounts to store permissions
- Installation and configuration

2 Authentication and Authorisation

- Principal
- Operation
- Reference Monitor
- Object

2.1 Principle vs subject

Principle: An entity that can be **granted access to objects** or can make statements affecting access control decisions. E.g user identity in an OS, used when discussing security policies

Subject An active entity within an IT system. E.g. **process running under a user identity**, used when discussing operational systems enforcing policies.

2.2 Objects

- Object Files or resources
- E.g. Memory, printers, directories
- Two options for focusing control:
  - What a subject is allowed to do. E.g. Word is allowed to write to a file
  - What may be done to an object. E.g. An object cannot be deleted, but can be executed.
- The access control is usually a compromise beween these two.
- In Unix, **everything is a file**
- Files really represent resources
- Organised in a tree structure, with alterations depending on the file system
- Inodes store permission information
- Every resource has an owner and a group
- Every resource has permission bits - held in the inode metadata
- Permissions for the user / group / others

2.2.1 inodes

- Inodes in Unix and Linux store the metadata for files
- Each file name links to an inode, which stores security information
- Directory permissions are slightly different to files:
- r List files within the directory
- w add or remove files
- x traverse directory, open files in the directory

3 General model

- Well settle on some common access on files:
- Read Simply viewing (Confidentiality)
- Write Includes changing, appending, deleting (Integrity)
- Execute Can run a file without knowing its contents

4 Ownership

- **Discretionary**: Owner can be defined **for each resource**. Owner controls who gets access.
- **Mandatory**: There could be a system-wide policy
- Most OSs support the concept of ownership
- Even owner can't pass out a entity, if the receiver doesn't have permission for the file.

5 Unix

- Unix simplifies access control by considering only the user, group and others
- User is the current owner
- Group is a named group entity
- Everyone else
- Unix offers Read, Write and Execute access controls

5.1 Groups

- Users with similar access rights can be collected into groups
- Groups are given permissions to access objects

5.2 UID/GID

- Usernames in Unix / Linux are soft aliases, your UID is what determines permissions
- User identities: UID
- Group identities: GID
- Your IDs are stored in /etc/passwd
- Root has a special UID: 0

6 The Shadow File

- In an attempt to improve password security, we can store password hashes in a shadow file
- Readable only by root users
- /etc/shadow stores the hashed passwords needed to authenticate users

7 SUID

- Set UID: set the effective user to be the file owner when executed
- Necessary to allow non-privileged access to privileged actions e.g. passwords
- Dangerous, but allows specific files to execute as sudo, without requiring it.

Reference section

placeholder