Access Control

1 Overview of Access Control

- What is Access Control?
  - The ability to allow only authorized users, programs or processes system or resource access
  - The granting or denying, according to a particular security model, of certain permissions to access a resource
  - An entire set of procedures performed by hardware, software and administrators, to monitor access, identify users requesting access, record access attempts, and grant or deny access based on pre-established rules.
  - Access control is the heart of security

- Examples of Access Control
  - *Social Networks:* In most social networks, such as Facebook and MySpace, some of your personal information can only be accessed by youself, some can be accessed by your friends, and some can be accessed by everybody. The part of system that implements such kind of control is doing access control.
  - *Web Browsers:* When you browse a web site, and run JavaScript code from that web site, the browser has to control what such JavaScript code can access, and what it cannot access. For example, a code from one web site cannot access the cookies from another web site, and it cannot modify the contents from another web site either. These controls are conducted by the browser's access control.
  - *Operating Systems:* In an operating system, one user cannot arbitrarily access another user's files; a normal user cannot kill another user's processes. These are done by operating system access control.
  - *Memory Protection:* In Intel 80x86 architecture, code in one region (for example, in Ring 3), cannot access the data in another more privileged region (e.g. Ring 0). This is done by the access control implemented in the CPU (e.g. 80386 Protection Mode).
  - *Firewalls:* Firewalls inspect every incoming (sometimes outgoing) packet, if a packet matches with certain conditions, it will be dropped by the firewalls, preventing it from accessing the protected networks. This is also access control.

- What should we learn about access control?
  - *Access Control Policy Models:* how access control policies are configured and managed.
    - Discretionary Access Control (DAC)
    - Mandatory Access Control (MAC)
  - *Access Control Mechanism:* how access control is implemented in systems.
    - Access Control Matrices
    - Access Control List

    - Capability
    - Role-Based Access Control
  - *Design Principles:* what are the useful principles that can guide the design and contribute to an implementation that is strong in security. Building a protection system is like building a bridge. We never ask people without civil engineering training to build a bridge for us, because we know that to build a bridge, we need to follow some civil engineering principles.

- DAC: Discretionary Access Control
  - Definition: An individual user can set an access control mechanism to allow or deny access to an object.
  - Relies on the object owner to control access.
  - DAC is widely implemented in most operating systems, and we are quite familiar with it.
  - Strength of DAC: Flexibility: a key reason why it is widely known and implemented in mainstream operating systems.

- MAC: Mandatory Access Control
  - Definition: A system-wide policy decrees who is allowed to have access; individual user cannot alter that access.
  - Relies on the system to control access.
  - Examples: The law allows a court to access driving records without the owners' permission.
  - Traditional MAC mechanisms have been tightly coupled to a few security models.
  - Recently, systems supporting flexible security models start to appear (e.g., SELinux, Trusted Solaris, TrustedBSD, etc.)

2 Access Control Methods

- Access Control Matrices
  - Disadvantage: In a large system, the matrix will be enormous in size and mostly sparse.

- Access Control List
  - The column of access control matrix.
  - Advantage:
    - Easy to determine who can access a given object.
    - Easy to revoke all access to an object
  - Disadvantage:
    - Difficult to know the access right of a given subject.
    - Difficult to revoke a user's right on all objects.
  - Used by most mainstream operating systems.

- Capability List
  - The row of access control matrix.

  - A capability can be thought of as a pair (x, r) where x is the name of an object and r is a set of privileges or rights.
  - Advantage:
    - Easy to know the access right of a given subject.
    - Easy to revoke a users access right on all objects.
  - Disadvantage:
    - Difficult to know who can access a given object.
    - Difficult to revoke all access right to an object.
  - A number of capability-based computer systems were developed, but have not proven to be commercially successful.

3 Access Control List Examples

- UNIX ACL
  - Abbreviations of Access Control Lists:
    - Three classes: owner, group, other users
    - Suffer from a loss of granularity
  - Full Access Control Lists
- Windows NT
  - Generic rights: No access, Read, Change, Full control.
  - Built-in Groups (each has different privileges)
    - Everyone: all users
    - Interactive: users logged on locally
    - Network: users logged on over the network
    - System: the operating system
    - Creator / Owner: creator or owner of a file or a resource
- Social networks
  - Most social networks use ACL as its main access control model. Users can specify who can access their profiles, friend lists, etc.
- How is the ACL implemented in operating systems?
  - Where to store the access control list? (Must be in a safe place)
  - ACL is saved in the i-node data structure.
  - The i-node data structure (see Figure 1).

```
EXTERN struct inode {
  mode_t i_mode;     /* file type, protection, etc. */
  nlink_t i_nlinks;  /* how many links to this file */
  uid_t i_uid;       /* user id of the file's owner */
  gid_t i_gid;       /* group number */
  off_t i_size;      /* current file size in bytes */
  time_t i_atime;    /* time of last access (V2 only) */
  time_t i_mtime;    /* when was file data last changed */
  time_t i_ctime;    /* when was inode itself changed (V2 only)*/
  zone_t i_zone[10]; /* zone numbers for direct, ind, and dbl ind*/

  ...
} inode[NR_INODES];
```

Figure 1: The i-node Data Structure in Minix

4 Design Principles of Access Control

In practice, producing a system that can prevent all attacks has proved to be difficult. However, experience has provided some useful principles that can guide the design and contribute to an implementation without security flaws. Here are eight examples of design principles that apply particularly to protection mechanisms. These principles are summarized and explained by Saltzer and Schroeder in a classical paper, *The Protection of Information in Computer Systems* [1]. We list these principles here, and you can read the detailed explanations from the paper.

1. *Economy of mechanism:* Keep the design as simple and small as possible.

2. *Fail-safe defaults:* Base access decisions on permission rather than exclusion.

3. *Complete mediation:* Every access to every object must be checked for authority.

4. *Open design:* The design should not be secret.

5. *Separation of privilege:* Where feasible, a protection mechanism that requires two keys to unlock it is more robust and flexible than one that allows access to the presenter of only a single key.

6. *Least privilege:* Every program and every user of the system should operate using the least set of privileges necessary to complete the job.

7. *Least common mechanism:* Minimize the amount of mechanism common to more than one user and depended on by all users.

8. *Psychological acceptability:* It is essential that the human interface be designed for ease of use, so that users routinely and automatically apply the protection mechanisms correctly. Also, to the extent that the user's mental image of his protection goals matches the mechanisms he must use, mistakes will be minimized.

These principles do not represent absolute rules– they serve best as warnings. If some part of a design violates a principle, the violation is a symptom of potential trouble, and the design should be carefully reviewed to be sure that the trouble has been accounted for or is unimportant.

5 Reference Monitor

The Reference Monitor concept was introduced in the *Computer Security Technology Planning Study* (Oct, 1972) by James Anderson & Co. This document is widely referred to as the *Anderson Report*. Reference Monitor provides an abstract model of the necessary and sufficient properties that must be achieved by any system claiming to securely enforce access controls. The three properties of Reference Monitor are summarized in the following:

1. The access mediation mechanism is always invoked every access is mediated. If this were not the case, then it would be possible for an entity to bypass the mechanism and violate the policy that must be enforced.

2. The access mediation mechanism is tamperproof. In the model, it is impossible for a penetrator to attack the access mediation mechanism such that the required access checks are not performed and authorizations not enforced.

3. It must be small enough to be subject to analysis and tests, the completeness of which can be assured. This must be the case, since if the mechanism could be demonstrated to be flawed, then it would not enforce the policy.

References

[1] J. H. Saltzer and M. D. Schroeder. *The Protection of Information in Computer Systems*. In Proceedings of the IEEE, Vol. 63, No. 9. (1975), pp. 1278-1308.