CS 3IS3 Fall 2007

05 Information Control Mechanisms

William M. Farmer

Department of Computing and Software
McMaster University

22 November 2007

Identity

- A principal is a unique entity.
- An identity is a definite description of a principal.
  - A definite description has the form

    "the entity x that satisfies the property P".

  - An indefinite description has the form

    "some entity x that satisfies the property P".

- A principle may have more than one identity.
- Identities are used for:
  - Accountability.
  - Access control.

Identity of an Object

- Objects are often identified by an assigned name.
  - An object can be assigned several names (which are sometimes called aliases).
- Example: Unix files have three kinds of names:
  1. The inode uniquely identifies a file and includes information about the file's access control permissions, ownership, disk location, modification time, etc.
  2. The file descriptor is a description of a file's inode.
  3. The path name describes the file by its position in the file hierarchy. Path names may be absolute or relative.
- Example: Uniform resource locators (URL) are names of resources on the Internet consisting of:
  1. A retrieval protocol.
  2. A host name and port number.
  3. A relative path name.

Identity of a User

- Unix systems identify users with two systems:
  1. Integers ≥ 0 called user identification numbers (UIDs).
  2. Text strings called login names.
- A real UID is the user identity at initial login.
- An effective UID is the user identity used for access control. It is changed using the Unix program `su`.
- A Unix process executes on behalf of a subject identified by a UID.
  - Usually the UID belongs to the user that executed the process's program.
  - For a process executing a setuid program, the UID belongs to owner of the program.

Identity of a Group or Role

- A group is a set of principals.
- Groups are used to assign privileges to the members of a group simultaneously.
- Groups can be implemented as either static (principals stay in their groups) or dynamic (principals are allowed to change their group affiliations).
- A role is a group associated with a certain function.
    - The rights needed to perform the role's function are granted by assigning the principal to the role.

Certificates and Identity (1)

- Recall that a certificate binds an identity of a principal to a cryptographic key.
- The identity is given by an identifier that should be a definite description of the principal.
  - In X.509v3 certificates use identifiers called Distinguished Names.
- A certification authority (CA) must verify that the principal requesting a certificate is actually the principal identified by the identifier.
- Every CA has two policies controlling how it issues certificates:
  - A CA authentication policy describes the level of authentication required to identify the principal to whom the certificate is to be issued.
  - A CA issuance policy describes the principals to whom the CA will issue certificates.

Certificates and Identity (2)

- Certificates can be issued to people, organizations, hosts, and roles.
- Someone who uses a certification is trusting that the CA has correctly verified that the identity bound to the key is actually principal who owns the key.
- A persona certificate binds an anonymous identity to a cryptographic key.

Identity of a Host

- Each host on the Internet is identified by an IP address.
  - The IP address is used to route packets to the host.
  - A host usually has one IP addresses, but it can have several.
  - The IP address may be a local address.
- A host registered with the DNS system is also identified by a domain name.
  - The domain name identifies what domain the host belongs to.
  - A host can have several domain names called aliases.
- IP address and domain names are dynamic identifiers—which complicates authentication.
- The DNS system is open to integrity attacks.

Cookies

- A cookie is a token or small message that one process sends to an another.
  - Cookies are often used with client-server applications.
  - To ensure integrity, cookies can be encrypted.
- Uses of cookies:
  - For a process to authenticate itself to another process (e.g., X Windows magic cookies).
  - To identify a particular transaction between processes.
  - To share state information concerning a transaction on a network (e.g., HTTP cookies)
- HTTP cookies are a concern for Internet privacy because they expose a user's web browsing behavior.

Anonymity on the Internet

- Anonymity is a means to protect privacy, but it also allows one to carry out behavior, for good or bad, without being responsible for the behavior.
- Anonymity is hard to achieve when communicating over the Internet.
- Approach 1: Spoof the source address.
  - Very easy to spoof the source address of IP packets.
  - Difficult to receive replies sent to the spoofed address.
- Approach 2: Use an anonymizer.
  - Functions as a trusted proxy server.
  - The communicating parties do not know each other, but the anonymizer knows them.
  - Anonymizers can be chained.
  - Example: `anon.penet.fi`.

Anonymizing Remailers (1)

- A pseudo-anonymous remailer replaces the source address information of an incoming message and then forwards the new message, but keeps the mapping between the source identity and the anonymous identity.
  - Allows the receiver of the message to reply to the original sender.
  - The binding between the real address and the anonymous address of a message must be stored by the remailer.
- A Cyberpunk remailer deletes the header of an incoming message and then forwards the data of the message as a new message.
  - The binding between the real address and the anonymous address of a message is not remembered.
  - The data of the message is encrypted.
  - Cyberpunk remailers are often used in chains.
  - Open to attacks based on traffic monitoring.

Anonymizing Remailers (2)

- A Mixmaster remailer is a Cyberpunk remailer that pads or fragments outgoing messages to fixed size.
  - Are much less vulnerable to attacks based on traffic monitoring.

Access Control Mechanisms

1. Access control matrix (for theoretical purposes).
2. Access control lists.
3. Capabilities.
4. Locks and keys.
5. Ring-based access control.
6. Propagated access control lists.

Weaknesses of the Access Control Matrix

- Weakness 1: An access control matrix is an inefficient data structure.
  - Requires a huge amount of storage space.
  - Much of the matrix is empty or contains redundant information.
- Weakness 2: The management of an access control matrix is complicated.
  - There are many objects and subjects to manage.
  - Objects and subjects are created and deleted on a continuous basis.
- Weakness 3: An access control matrix is a centralized data structure for storing access control information.
- As a result of these weaknesses, the access control matrix is not a practical access control mechanism.

Access Control Lists

- Let S be a set of subjects and R be a set of rights of a system.
- A access control list (ACL) is a set

$$L = \{(s, R') : s \in S \text{ and } R' \subseteq R\}.$$

- Each object o is assigned an access control list $\mathrm{acl}(o)$ such that, if $(s, R') \in \mathrm{acl}(o)$, then the subject s may access o using any right $r \in R'$.
- An access control list corresponds to a single column of an access control matrix.
- For efficiency, an access control list can be written in an abbreviated form.
  - Example: File access control lists in Unix.

Management of Access Control Lists

- Which subjects can modify an object's ACL?
  - A subject that has the own right to a object is usually give the right to modify the object's ACL.
  - The creator of an object is usually given all rights to the object including own.
- Do the ACLs apply to a privileged user?
  - ACLs may not apply to privileged users like the Unix root and Windows administrator.
- Do the ACLs support groups and wildcards?
  - Groups and wildcards are used to make ACLs more compact and easier to manipulate.
- How are conflicts in an access control list handled?
- How are default permissions used with ACLs?
  - Default permissions can be used in place of a nonexistent ACL.
  - Default permissions can be integrated with the ACL permissions.

Capability Lists

- Let O be a set of objects and R be a set of rights of a system.
- A capability list (C-List) is a set

$$L = \{(o, R') : o \in O \text{ and } R' \subseteq R\}.$$

- Each subject s is assigned a capability list $\text{cap}(s)$ such that, if $(o, R') \in \text{cap}(s)$, then the subject s may access o using any right $r \in R'$.
- A capability list corresponds to a single row of an access control matrix.
- Capability lists are the dual of access control lists.

Capabilities

- Each member of a capability list is a capability.
- A capability can be an encapsulation of the identity of an object.
  - For example, the capability can include the location of the object.
- A subject, such as a process, needs to possess an appropriate capability to access an object.
- The integrity of capabilities must be protected for the access control system to work.

Management of Capabilities

- Mechanisms for protecting the integrity of capabilities:
  1. Tagged architecture: Each hardware word has an associated tag with two states, set and unset.
  2. Protected memory: Capabilities are stored in memory that a process can read but not write.
  3. Cryptography: Each capability has an associated cryptographic checksum which is encrypted.
- Copying of capabilities can be controlled with copy flags.
- Sometimes processes need to have their rights to an object temporarily amplified.
- Revocation of a right can be achieved by indirection in which an object is accessed via an entry in a global object table.

Comparison between Capabilities and Access Control Lists

- Two fundamental access control questions:
  1. What objects can a given subject access, and how?
  2. What subjects can access a given object, and how?
- Capabilities are best for answering question 1.
- Access control lists are best for answering question 2.
- Access control lists are more often used than capabilities because question 2 is more often asked than question 1.

Locks and Keys

- Each object has a set of locks.
- Each subject has a set of keys.
- If a subject has a key that fits an object's lock, then the subject can access the object in a particular way.
- Main advantage: Locks and keys can be dynamically changed.

Cryptographic Locks and Keys

- Locks and keys can be implemented crytographically:
  - An object is locked by encryption.
  - A subject unlocks the object by decrypting it.
- Cryptographic locks and keys enable:
  - Or-access: $(E_1(o), \ldots, E_n(o))$.
  - And-access: $E_1(\cdots(E_n(o))\cdots)$.

Type Checking Locks and Keys

- Locks and keys can be implemented with type checking:
    - Each object has one or more types.
    - A subject has one or more types.
    - A subject can access an object if its type matches the type of the object.
- Type checking is powerful method that is used in many areas of computer science, particularly in programming languages and specification languages.

Ring-Based Access Control (1)

- Ring-based access control is a generalization of the user/supervisor protection mechanism.
  - Introduced by the Multics operating system.
- The operating system is organized as a sequence of 64 protection rings, numbered 0–63.
  - The higher the ring, the lower the privileges.
  - The kernel resides in ring 0.
- There are two kinds of memory segments: data and procedure.
  - Segments have the following possible access rights: read (r), write (w), execute (e), append (a)
  - Access to segments is control by ACLs.
- A procedure normally executes in the ring to which it is associated, but a procedure can cross ring boundaries under certain circumstances.

Ring-Based Access Control (2)

- A gate allows access to a procedure segment across ring boundaries.
  - They are like "public" data and procedures.
- Each data and procedure segment has an access bracket (a_1, a_2) of ring numbers $a_1 \leq a_2$, and each procedure segment may have also have a call bracket (c_1, c_2) of ring numbers $c_1 \leq c_2$.
- Suppose a procedure executing in ring r wants to access a data segment with access bracket (a_1, a_2). Then the following MAC rule is used in addition to the data segment's ACL:
  - $r \leq a_1$: access permitted.
  - $a_1 < r \leq a_2$: r access permitted; w and a access denied.
  - $a_2 < r$: access denied.

Ring-Based Access Control (3)

- Suppose a procedure executing in ring r wants to access a procedure segment with access bracket (a_1, a_2) and call bracket (a_2, a_3). Then the following MAC rule used in addition to the procedure segment's ACL:
  - $r < a_1$: access permitted; ring-crossing fault.
  - $a_1 \leq r \leq a_2$: access permitted; no ring-crossing fault.
  - $a_2 < r \leq a_3$: access permitted if through a valid gate.
  - $a_3 < r$: access denied.
- Ring-crossing faults are handled by a kernel mechanism called the Gatekeeper.

Propagated Access Control Lists

- A propagated access control list (PACL) is an ORCON mechanism in which the creator of an object controls access to the object.
  - When a subject creates an object, the subject creates the PACL that is associated with the object.
  - Only the creator of the object can change the object's PACL.
  - When a subject reads an object, the PACL of the object is incorporated into the subject's PACL.
- An object's PACL is copied with the object.
- DACs can augment PACLs to further restrict access.