Authentication, Access Control, Auditing and Non-Repudiation

Principals

- Humans or system components that are registered in and authentic to a distributed system.
- Principal has an identity used for:
  - Making principal accountable for its actions
  - Obtaining access to a protected component
  - Identifying the originator of a message
  - Identifying who to charge for service provision.
**Credentials**

- Information the system has about principals:

  - **Credentials**
    - unauthenticated attributes
    - authenticated attributes
      - identity attributes
      - privilege attributes

**Secure Requests**

- Principals (or objects acting on their behalf) make requests

  - Client
    - request
    - client-side security on invocations
      - security association
      - access control
      - message protection
      - audit
  - Target Object
    - request
    - target-side security on invocations
      - security association
      - access control
      - message protection
      - audit
What’s needed for secure requests?

- Establishing security association between client & server (authentication)
- Deciding whether principal may perform this operation (access control)
- Making the principal accountable for having requested the operation (auditing)
- Protecting request and response from eavesdropping in transit (encryption)

Establishing Security Association

- Involves
  - Establishing trust in one another’s identities
    - Client authenticating server’s identity
    - Server authenticating client’s identity
  - Making client credentials available to server
  - Establishing the security context used for protecting requests and replies in transit (e.g. distributing private keys)
What is Authentication?

- **Authentication**: Proving you are who you claim to be.
- **In centralised systems**: Password check at session start.
- **In distributed systems**:
  - Use of authentication server
  - Usually based on ability to encrypt/decrypt a message (c.f. Needham/Schroeder Protocol)

Message Protection

[Diagram showing the flow of data in a secure communication protocol with annotations explaining each step.]
Access Control

- Object invocation access control
- Application level access control

Object Invocation Access Policies

- Access decision functions enforce object invocation access policies:
  - client-side access decision functions and/or
  - server-side access decision functions

- Decisions are based on
  - operation to be performed
  - privilege attributes of principal
  - control on principal’s privilege attributes (e.g. time valid)
  - server control attributes
Application Access Policies

- In previous case access control is transparent to client and server objects
- In this case client and/or server objects implement access control themselves
- Application access policies
  - can take into account the particular data being accessed
  - can take into account the semantics of request parameters

Access Control Privilege Attributes

- Privilege attributes of principals for access control include:
  - principal’s identity
  - roles (related to the principal’s job functions)
  - groups (related to organizational structure in which principal is embedded)
  - security clearance
  - capabilities of server objects that the principal is allowed to use
  - others...
Server Control Attributes

- Access Control Lists (ACLs) identifying permitted users by
  - name or
  - privilege attributes
- Information for label-based schemes
- Control attributes are generally shared by groups of operations of an object or even by groups of objects

Auditing

- Assists in detection of attempted or actual security breaches
- By recording details of security relevant events
  - Writing event details into a log file
  - Generating a security alert
  - Taking other actions
- Two levels of auditing
  - system-level
  - application-level
Potentially a large number of events could be recorded

Security auditing policies restrict the set of events to those that are critical for the particular environment

System auditing policies log all security relevant events, even from security unaware applications
Non-Repudiation

- Makes principals accountable for their actions
- Irrefutable evidence about events/actions is generated
- Used to settle disputes about the occurrence or non-occurrence of an event
- Example: Electronic commerce

Components of Evidence

- Depend on non-repudiation policy.
- Examples include:
  - Type of action or event
  - A timestamp obtained from a trusted authority
  - Parameters related to action or event
  - Proof of origin of parameters
Common Types of Evidence

- **Proof of creation of a message**
  - Protects against originator’s false denial of having created a message

- **Proof of receipt of a message**
  - Protects against receiver’s false denial of having received a message

© Wolfgang Emmerich, 1997