Access Control

CSE497b - Spring 2007
Introduction Computer and Network Security
Professor Jaeger
www.cse.psu.edu/~tjaeger/cse497b-s07/
Access Control

• Describe the permissions available to computing processes
  – Originally, all permissions were available

• Clearly, some controls are necessary
  – Prevent bugs in one process from breaking another

• But, what should determine access?
Permissions for Processes

• What permissions should be granted to...
  – An editor process?
  – An editor process that you run?
  – An editor process that someone else runs?
  – An editor process that contains malware?
  – An editor process used to edit a password file?

• Q: How do we determine/describe the permissions available to processes?

• Q: How are they enforced?

• Q: How might they change over time?
Protection System

• Any “system” that provides resources to multiple subjects needs to control access among them
  – Operating system
  – Servers

• Consists of:
  – Protection state
    • Description of permission assignments (i.e., policy)
    • Determines how security goals are met
  – Enforcement mechanism
    • Enforce protection state on “system”
Protection State

- Describes the conditions under which the system is **secure**
  - Secrecy
  - Integrity
  - Availability

- Described in terms of
  - **Subjects:** Users and processes
  - **Objects:** Files and sockets
  - **Operations:** Read and write
Secure Protection State

• Set of all protection states $P$
• Set of secure protection states $Q$
  – Subjects access to objects to perform operations
  – Meets secrecy, integrity, availability goal
• Example: Protect access to your private key file
  – Only protection states in which only you can read the private key file are secure
  – Protection states in which only you may write the public key file are secure
• Not all processes are necessarily secure
  – Recall programs running on your behalf
  • *Hey, even some programs running on your behalf are not to be trusted with the private key!*
Access Matrix

- Subjects
- Objects
- Operations
- Can determine
  - Who can access an object
  - What objects can be accessed by a subject
  - What operations a subject can perform on an object
Access Control

- Suppose the private key file for J is object $O_1$
  - Only J can read
- Suppose the public key file for J is object $O_2$
  - All can read, only J can modify
- Suppose all can read and write from object $O_3$
- What’s the access matrix?
Secrecy

• Does the following protection state ensure the secrecy of J’s private key in O₁?

<table>
<thead>
<tr>
<th></th>
<th>O₁</th>
<th>O₂</th>
<th>O₃</th>
</tr>
</thead>
<tbody>
<tr>
<td>J</td>
<td>R</td>
<td>R</td>
<td>R</td>
</tr>
<tr>
<td></td>
<td>R</td>
<td>W</td>
<td>W</td>
</tr>
<tr>
<td>S₂</td>
<td>N</td>
<td>R</td>
<td>R</td>
</tr>
<tr>
<td></td>
<td>W</td>
<td>W</td>
<td></td>
</tr>
<tr>
<td>S₃</td>
<td>N</td>
<td>R</td>
<td>R</td>
</tr>
<tr>
<td></td>
<td>W</td>
<td></td>
<td>W</td>
</tr>
</tbody>
</table>
Integrity

- Does the following access matrix protect the integrity of J’s public key file $O_2$?

```
<table>
<thead>
<tr>
<th></th>
<th>$O_1$</th>
<th>$O_2$</th>
<th>$O_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>J</td>
<td>R</td>
<td>R</td>
<td>R</td>
</tr>
<tr>
<td></td>
<td>R</td>
<td>W</td>
<td>W</td>
</tr>
<tr>
<td>$S_2$</td>
<td>N</td>
<td>R</td>
<td>R</td>
</tr>
<tr>
<td></td>
<td>W</td>
<td>W</td>
<td></td>
</tr>
<tr>
<td>$S_3$</td>
<td>N</td>
<td>R</td>
<td>R</td>
</tr>
<tr>
<td></td>
<td>W</td>
<td>W</td>
<td></td>
</tr>
</tbody>
</table>
```
Trusted Processes

- Does it matter if we do not trust some of J’s processes?

<table>
<thead>
<tr>
<th></th>
<th>O₁</th>
<th>O₂</th>
<th>O₃</th>
</tr>
</thead>
<tbody>
<tr>
<td>J</td>
<td>R</td>
<td>R</td>
<td>R</td>
</tr>
<tr>
<td>J</td>
<td>R</td>
<td>W</td>
<td>W</td>
</tr>
<tr>
<td>S₂</td>
<td>N</td>
<td>R</td>
<td>R</td>
</tr>
<tr>
<td>S₂</td>
<td>W</td>
<td>W</td>
<td>W</td>
</tr>
<tr>
<td>S₃</td>
<td>N</td>
<td>R</td>
<td>R</td>
</tr>
<tr>
<td>S₃</td>
<td>W</td>
<td>W</td>
<td>W</td>
</tr>
</tbody>
</table>
Protection vs Security

• Protection
  – Security goals met under *trusted* processes
  – Protects against an error by a non-malicious entity

• Security
  – Security goals met under *potentially malicious* processes
  – Protects against any malicious entity

• For J:
  – Non-malicious process shouldn’t leak the private key by writing it to \( O_3 \)
  – A potentially malicious process may contain a Trojan horse that can write the private key to \( O_3 \)
Least Privilege

- Limit permissions to those required and no more
- Consider three processes for user J
  - Restrict privilege of the process J_1 to prevent leaks

<table>
<thead>
<tr>
<th></th>
<th>O_1</th>
<th>O_2</th>
<th>O_3</th>
</tr>
</thead>
<tbody>
<tr>
<td>J_1</td>
<td>R</td>
<td>R</td>
<td>N</td>
</tr>
<tr>
<td>J_2</td>
<td>N</td>
<td>R</td>
<td>R</td>
</tr>
<tr>
<td>J_3</td>
<td>N</td>
<td>R</td>
<td>R</td>
</tr>
</tbody>
</table>
Options for Subjects

• Possible subjects
Role-Based Access Control

• Associate permissions with job functions
  – Each job defines a set of tasks
  – The tasks need permissions
  – The permissions define a role

• Bank Teller
  – Read/Write to client accounts
  – Cannot create new accounts
  – Cannot create a loan
  – Role defines only the permissions allowed for the job

• What kind of jobs can we define permission sets for?
Role-based Access Control

• Model consists of two relationships
  – Role-permission assignments
  – User-role assignments

• Assign permissions to roles
  – These are largely fixed

• Assign a user to the roles they can assume
  – These change with each user
  – Administrators must manage this relationship
Enforcement Mechanism

- Every system needs to enforce its protection state
- \textbf{Q}: What is required of such an enforcement mechanism?
Reference Monitor

• Properties
  – Complete Mediation of all security-sensitive operations
  – Tamperproof
  – Simple enough for verification of correctness

• Reference Monitor Structure
  – Interface
    • Where is it called to mediate (authorize)?
  – Mechanism
    • How are authorization queries processed?
  – Policy
    • How are authorization decisions expressed?
Reference Monitor

Loadable Authorization Module

Authorization Mechanism

Policy Server

User

Trap

Kernel
Protection State Transitions

• Transition
  – From one access matrix state to another
  – Add/delete subject, object, operation assignment

• Transition semantics
  – Owner-driven
  – Delegation
  – Administrator-driven
  – Administrative permissions

• Attenuation of Rights Principle
  – Can’t grant a right that you do not possess
Protection State Transitions

• Owner
  – Implicitly has all rights to owned objects
  – Grants at will
  – Reader can copy object to self-owned object and distribute

• Delegation
  – Copy flag
    • Presence of copy flag permits granting of one’s rights to that object

• Administrators
  – Implicitly have all rights
  – Grant to subjects as necessary (w/i security goals)

• Administrative permissions
  – Permissions to perform administrative operations on objects
  – Distinction between active and administrative rights
Safety Problem

• Is there a general algorithm that enables us to determine whether a permission may be leaked to an unauthorized user from any future protection state?

• Intuition:
  – From a protection state, users can administer permissions for the objects that they own
  – Enable other subjects to access those objects

• For typical access control models (UNIX)
  – Problem is Undecidable
  – Can also extend representation (new users, objects)

• Practice:
  – Check current protection state for “safety”
Take Away

- Access Control is expressed in terms of Protection Systems
  - Protection Systems consist of
    - Protection State representation (e.g., access matrix)
    - Enforcement Mechanisms (e.g., reference monitor)
- Protection States
  - Challenge to choose subjects (RBAC)
  - Must to ensure security goals in spite of state transitions
- Enforcement Mechanism
  - Reference Monitor
  - Ensures protection state is enforced
- Transitions
  - Cannot prove safety for future protection states