Discretionary Access Control (DAC)

Prof. Ravi Sandhu
Executive Director and Endowed Chair

Lecture 7

ravi.utsa@gmail.com
www.profsandhu.com
Authentication, Authorization, Audit

AAA

Authentication  Authorization  Audit

Who are You?  What are You Allowed to Do?  What Did You Do?

siloed  integrated
Access Control

- Discretionary Access Control (DAC), 1970
- Mandatory Access Control (MAC), 1970
- Role Based Access Control (RBAC), 1995
- Attribute Based Access Control (ABAC), ????

Fixed policy

Flexible policy
Access Matrix Model
## Access Matrix Model

### Objects (and Subjects)

<table>
<thead>
<tr>
<th>Subjects</th>
<th>Objects</th>
<th>Subjects</th>
</tr>
</thead>
<tbody>
<tr>
<td>U</td>
<td>F</td>
<td>G</td>
</tr>
<tr>
<td>V</td>
<td>r w own</td>
<td>r</td>
</tr>
<tr>
<td></td>
<td>r</td>
<td>r w own</td>
</tr>
</tbody>
</table>

**Rights**
Basic Abstractions
- Subjects
- Objects
- Rights

The rights in a cell specify the access of the subject (row) to the object (column)
A subject is a program (application) executing on behalf of a user.

A user may at any time be idle, or have one or more subjects executing on its behalf.

User-subject distinction is important if subject’s rights are different from a user’s rights:
- Usually a subset
- In many systems a subject has all the rights of a user.

A human user may manifest as multiple users (accounts, principals) in the system.
Users and Subjects

- JOE
  - JOE.TOP-SECRET
  - JOE.SECRET
  - JOE.CONFIDENTIAL
  - JOE.UNCLASSIFIED

USER

SUBJECTS
Users and Subjects

JANE

- JANE.CHAIRPERSON
- JANE.FACULTY
- JANE.EMPLOYEE
- JANE.SUPER-USER

USER

SUBJECTS
Objects

- An object is anything on which a subject can perform operations (mediated by rights)

- Usually objects are passive, for example:
  - File
  - Directory (or Folder)
  - Memory segment
  with CRUD operations (create, read, update, delete)

- But, subjects can also be objects, with operations
  - kill
  - suspend
  - resume
# Access Matrix Model

<table>
<thead>
<tr>
<th>Subjects</th>
<th>Objects (and Subjects)</th>
</tr>
</thead>
<tbody>
<tr>
<td>U</td>
<td>F</td>
</tr>
<tr>
<td>W</td>
<td>r w own</td>
</tr>
<tr>
<td>W</td>
<td>r w own</td>
</tr>
<tr>
<td>W</td>
<td></td>
</tr>
</tbody>
</table>
Implementation

- Access Control Lists
- Capabilities
- Relations
Access Control Lists

Each column of the access matrix is stored with the object corresponding to that column.

F
- U:r
- U:w
- U:own

G
- U:r
- V:r
- V:w
- V:own
### Capabilities

<table>
<thead>
<tr>
<th></th>
<th>F/r, F/w, F/own, G/r</th>
</tr>
</thead>
<tbody>
<tr>
<td>U</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>G/r, G/w, G/own</th>
</tr>
</thead>
<tbody>
<tr>
<td>V</td>
<td></td>
</tr>
</tbody>
</table>

Each row of the access matrix is stored with the subject corresponding to that row.
# Relations

<table>
<thead>
<tr>
<th>Subject</th>
<th>Access</th>
<th>Object</th>
</tr>
</thead>
<tbody>
<tr>
<td>U</td>
<td>r</td>
<td>F</td>
</tr>
<tr>
<td>U</td>
<td>w</td>
<td>F</td>
</tr>
<tr>
<td>U</td>
<td>own</td>
<td>F</td>
</tr>
<tr>
<td>U</td>
<td>r</td>
<td>G</td>
</tr>
<tr>
<td>V</td>
<td>r</td>
<td>G</td>
</tr>
<tr>
<td>V</td>
<td>w</td>
<td>G</td>
</tr>
<tr>
<td>V</td>
<td>own</td>
<td>G</td>
</tr>
</tbody>
</table>

commonly used in relational database management systems
ACLs versus Capabilities

- **Authentication**
  - ACL's require authentication of subjects and ACL integrity
  - Capabilities require integrity and propagation control

- **Access review**
  - ACL's are superior on a per-object basis
  - Capabilities are superior on a per-subject basis

- **Revocation**
  - ACL's are superior on a per-object basis
  - Capabilities are superior on a per-subject basis

- **Least privilege**
  - Capabilities provide for finer grained least privilege control with respect to subjects, especially dynamic short-lived subjects created for specific tasks
ACLs versus Capabilities

- **Authentication**
  - ACL's require authentication of subjects and ACL integrity
  - Capabilities require integrity and propagation control

- **Access review**
  - ACL’s are superior on a per-object basis
  - Capabilities are superior on a per-subject basis

- **Revocation**
  - ACL’s are superior on a per-object basis
  - Capabilities are superior on a per-subject basis

- **Least privilege**
  - Capabilities provide for finer grained least privilege control with respect to subjects, especially dynamic short-lived subjects created for specific tasks

Most Operating Systems use ACLs often in abbreviated form: owner, group, world
Content-Dependent Controls

- content dependent controls
  - you can only see salaries less than 50K, or
  - you can only see salaries of employees who report to you

- beyond the scope of Operating Systems and are provided by Database Management Systems
context dependent controls
- cannot access classified information via remote login
- salary information can be updated only at year end
- company's earnings report is confidential until announced at the stockholders meeting

- can be partially provided by the Operating System and partially by the Database Management System
- more sophisticated context dependent controls such as based on past history of accesses definitely require Database support
Information from an object which can be read can be copied to any other object which can be written by a subject.

Suppose our users are trusted not to do this deliberately. It is still possible for Trojan Horses to copy information from one object to another.
User B cannot read file F
User A executes Program Goodies, which contains a Trojan Horse that creates File F and File G.

ACL:
- A:r
- B:r
- A:w

User B can read contents of file F copied to file G.
Read of a digital copy is as good as read of original

Write to a digital copy is not so useful
DAC Subtleties

- Chains of grants and revokes
- Inheritance of permissions
- Negative rights