University of BRISTOL

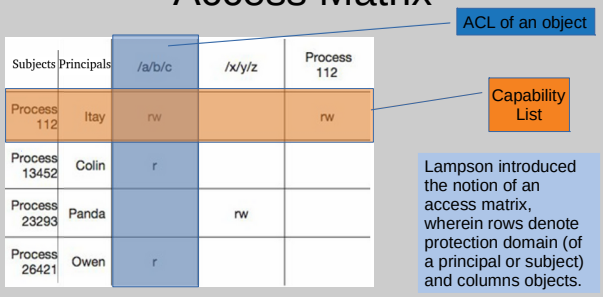
Computer System- B Security

Introduction to OS Security

Access Control Mechanisms (DAC/MAC/RBAC)

Sanjay Rawat
bristol.ac.uk

Access Matrix



Lampson introduced the notion of an access matrix, wherein rows denote protection domain (of a principal or subject) and columns objects.

Multi-* environment

- Multi-user
- Multi-process/tasking
  - Virtualization for separating process address spaces (via page table and MMU)
- Access-control mechanism
  - **who** is allowed to do **what** to the system **objects** we'd like to protect: files, devices, etc.
  - How do we decide these policies?

Discretionary Access Control (DAC)

- The owner of an object controls access to it.
- Practically, a subject's access rights depend on the user, associated to that subject!
- This is the most common case of computing systems we use on daily basis.

Terminology

- Objects (*aka* resources): entities that we want to protect. E.g. files, memory, etc.
- Subjects: entities that performs actions on objects. e.g. processes, threads.
- Note: threads/process also performs actions on other threads!
- We model ***subjects*** as running on behalf of (human) users (*also called as **security principals***!) and of ***objects/resources*** as being owned by such users.
- What subjects are allowed to "do" is called ***access rights***. Collectively, it is called ***protection domain*** of a subject.
- Actions are: **read**, **write** and **execute**.

Access Control List (ACL)

- Owner of an object creates ACL
  - Mainly focuses on **who** can do **what.**
  - **Who:** current owner (user), current group, and others
  - **What: read**, **write** and **execute**.
- Most of our PCs are based on DAC (with some elements of MAC-- the next topic)

Mandatory Access Control (MAC)

- Enforce a system-wide policy on information access, (as opposed to deciding at the discretion of the owner!)
- The idea is to have a classification of objects and access is granted at finer level (e.g. ***need to know***).
- Inspired from Bell–LaPadula model.
- The model adopted the system prevalent in military!
- Used by OSs (SELinux, e.g.)

Bell-LaPadula

- Objects are ***classified*** (security classification) and subjects have *security clearance.*
- *It defines two* **rules** (properties)**:**
  - ***simple security property:*** no subject may read from an object whose classification is higher than the subject's clearance (aka ***no-read-up***).
  - ****-property:*** no subject may write to an object whose classification is lower than the subject's clearance (aka ***no-write-down***).
- Often too theoretical or impractical and therefore has been refined *further.*

Role Based Access Control (RBAC)

- Access under RBAC is based on a user's role within the organization to which the computer system belongs.
  - Permission are based on the roles (often based on the ***principle of the least privilege***)
  - Users are assigned to roles
  - Like MAC, enforcement is based on capabilities of the subjects.