The Protection of Information in Computer Systems

Jerome H. Saltzer and Michael D. Schroeder
Presented by Kang-Hsien Chou and Sharanya Eswaran
Outline

- Separation of address and protection
- Capability system and access control list system
- Protected subsystem
- Role based access control (other mechanism)
- Research directions
Descriptor-based system

- Capability system and access control list system
- Early feature is to prevent from writing in other users’ allocated memory. (IBM System 360)
- “fetch protect” was than considered (IBM System/370)
- The scheme is supported by many OS including Honeywell, Hydra, StarOS …
Separation of Address and Protection

- All memory accesses need to go through two levels of descriptor.
- Each segment must be a distinct addressing descriptor.
  - A unique segment identifier
  - An offset address
- The protection descriptor itself no longer contains a base and bound.
Capability system

- Who can load values into the protection descriptor registers?
  - Privileged state bit
  - To allow any program to load the protection descriptor registers, but only from locations in memory that previously have been certified to contain acceptable protection descriptor value.

- Two kinds of objects stored in the memory.
  - Protection descriptor values
  - Ordinary data value
  - Tagged bit
Access control list system

- Storage area for data and access controller
  - Address descriptor for the associate segment
  - Access control list
- Authorization check
  - When a virtual processor attempts to refer to the segment associated with the access controller, the memory system looks up the principal identifier in the access control list.
Protected subsystem

- Why protected subsystem?
  - Only those access restrictions provided by the standard system facilities can be enforced.
  - Execution of a borrowed program in the borrower’s domain can present a real danger to the borrower.
- A protected subsystem is a collection of program and data segment that is “encapsulated”
  - Care taker program
Role based access control

- Role based access control is formalized in 1992 by David Ferraiolo and Rick Kuhn.
- Why RBAC?
  - DAC allows users to grant or revoke access to any of the objects under their control.
  - However, if the end uses do not “own” the information for which they are allowed access?
- RBAC is a nondiscretionary access control mechanism which allows and prompts the central administration of an organization specific security policy.
- Provide means to Naming, as well as relationships between individuals and rights
- The user can not pass access permissions on to others at their discretion.
Research directions

- Certification of the correctness of protection system designs and implementations
  - A precise model (complete isolation?, sharing of information?)
  - Verify if the presented implementation actually does what it claims.
  - Friendly user interface
  - System software runs without protection constraints
- Invulnerability of protection system design
  - Undetected disabling of the protection mechanism
- Constraints on use of information after release
Research directions (cont’d)

- Encipherment of information with secret key
  - how to communicate the keys to authorized users
  - Schemes for protecting the keys
  - Development of encipherment techniques

- Improved authentication mechanisms
  - Personal attributes
  - Bioinfo
  - Rhythm of keyboard typing