

Trusted Systems



1

Trusted system

A computer system architecture that is characterized by functions that rigorously enforce a multi-level security model.



2

“System High”

P Entire system (all data and processes) protected at the level of the most sensitive object.

P Non-trusted system “wrapped” in external security.



3

Discretionary Access Control (DAC)

P Systems discriminate between ownership of data resources (e.g., files) through a User ID mechanism, and permit a resource owner to grant simple access rights to these resources (typically read access, write access, execute access and delete access).

P Ability to grant access rights

P Rights of the User ID are extended to program processes launched under the computer account associated with a User ID.

P Discretionary separation of users and data

P Minimum form of trusted system

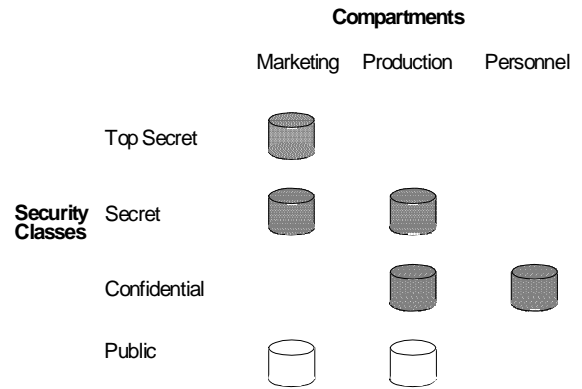
4

Lattice Model Data Classification Systems

| | | Compartments | | |
|------------------|--------------|--------------|------------|-----------|
| | | Marketing | Production | Personnel |
| Security Classes | Top Secret | | | |
| | Secret | | | |
| | Confidential | | | |
| | Public | | | |

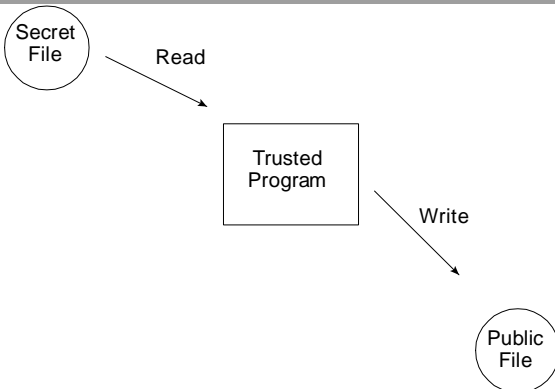
5

Non-symmetric in Practice



6

Mandatory Access Control (MAC)



7

Bell-Lapadula Model

Two characteristic properties of allowable information flow for MAC

Assumes objects (e.g., files) and subjects (e.g., program processes and users) have assigned security classes (lattice levels).



8

The Simple Security Property

P The "no-read-up" rule

P A subject may read an object only if the subject is characterized by a security class that is equal or greater than the object's security class

- $OC \leq SC$
 - where OC is the object security class
 - SC is the subject security class



9

The *-property

P The "star property"

P A subject may have write access to an object only if that object is characterized by a security class that is equal or greater than the highest security class for any objects for which the subject currently has read access

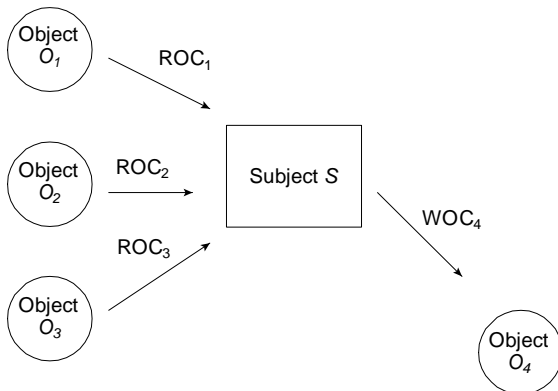
P $\max(ROC_1, ROC_2, \dots, ROC_n) \leq WOC$

- where ROC is the read object security class
- WOC is the write object security class
- s is the subject
- o is the object
- s is the subject



10

The *-property



11

Bell-lapadula Model Access Matrix

| | | Objects | | | | | | | | | | | | |
|----------|-----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|-----------------|-----------------|-----------------|-----------------|
| | | O ₁ | O ₂ | O ₃ | O ₄ | O ₅ | O ₆ | O ₇ | O ₈ | O ₉ | O ₁₀ | O ₁₁ | O ₁₂ | O ₁₃ |
| Subjects | S ₁ | e | r | a | w | e | e | e | e | e | e | w | r | e |
| | S ₂ | e | e | e | w | e | r | e | e | e | e | e | e | w |
| | S ₃ | a | e | e | e | e | r | e | e | e | e | e | w | e |
| | S ₄ | e | e | e | r | w | e | e | e | e | e | w | e | e |
| | S ₅ | e | e | e | e | a | e | r | e | w | e | e | e | e |
| | S ₆ | w | e | e | e | e | e | e | e | r | e | e | e | e |
| | S ₇ | e | r | e | e | e | e | a | w | e | e | e | e | e |
| | S ₈ | e | w | e | e | e | r | r | r | r | e | e | e | e |
| | S ₉ | e | e | e | e | e | w | w | e | e | e | r | e | e |
| | S ₁₀ | e | r | w | e | e | e | e | e | e | e | e | e | e |
| | S ₁₁ | e | e | a | w | e | r | e | e | e | e | e | e | e |

Key
 e - exclude
 a - append
 r - read
 w - read/write

12

Typical Operating System Security

P Single-level of security on objects

- Discretionary access control (DAC) or
- Mandatory access control (MAC)

P Allocation monitor module in the operating system

P Access validated against the access rights

P Blocked (rejected) when violated

P Several independent allocation monitors:

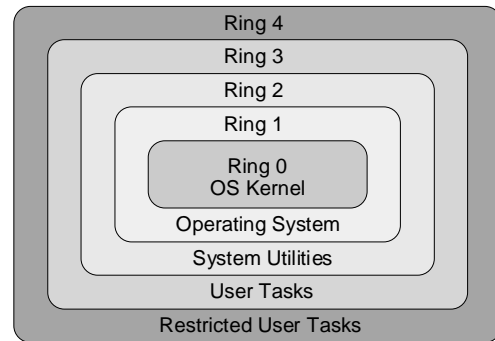
- Storage allocation monitor
- Device allocation monitor
- Memory allocation monitor, etc.



13

Operating System Kernel Architecture

"Privileged" States



14

Security Kernels



P Reference monitor replaces or supplements the access monitors

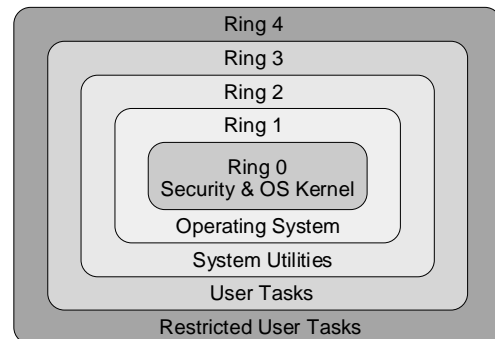
P mediates all activities of subjects (processes) on objects (memory, devices, storage areas, etc.)

P Enforces access matrix for MAC

P Must run in the highest privilege state of the processor

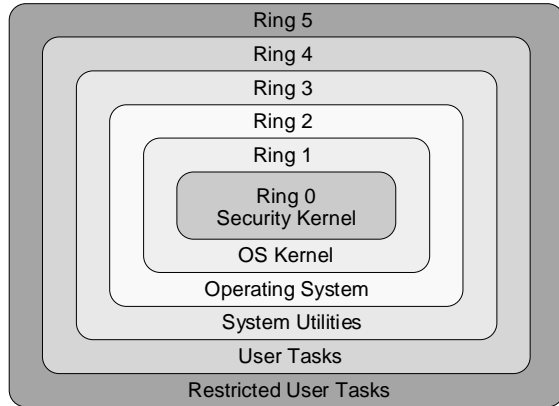
15

Sharing OS Kernel



16

Displacing OS Kernel



17

Data Base Process Granularity

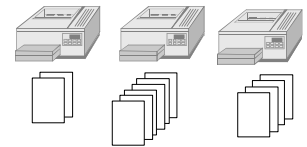
PLevel 1, No Process Distinction.



PLevel 2, Session or Account Distinction.

PLevel 3, Process Distinction.

PLevel 4, Process State Distinction.



18

Data Base Object Granularity

PTable-level granularity.

PRow-level granularity.

PColumn-level granularity.

| | Age | Dept | Ph | Mail |
|------|-----|------|----|------|
| Jon | | | | |
| Sam | | | | |
| Mary | | | | |
| Mike | | | | |
| Fred | | | | |
| Jim | | | | |
| Don | | | | |
| Dick | | | | |

19

Trusted System Certification

PKnown degrees of access enforcement

PMilitary application

PMoving into industrial practice

PAn economic history

▸ Opening military market

PA political history

▸ Opening international military market



20

Division D: *Minimal*

PClass D: Minimal Protection. Meets no other criteria below.



21

Division C: *Discretionary*

PClass C1:

- ▶ Discretionary Security. Self-protection through user authentication (e.g., user login password). Example: most multi-user commercial systems, such as Compaq's Open VMS operating system.

PClass C2:

- ▶ Controlled Access. Encapsulation of important objects in the system. Example: most add-on security packages, such as the IBM RACF package, Trusted Oracle DBMS



22

Division B: *Mandatory*

PClass B1:

- ▶ Labeled Security. Explicit protection model, all objects are protected.

PClass B2:

- ▶ Structured. All security modules in the operating system are identifiable. Example: Honeywell Multics operating system.

PClass B3:

- ▶ Security Domains. Security mechanism is the central kernel, beneath, but isolated from all service modules. Example: GEMSOS iAPX86 operating system.



23

Division A: *Verified*

PClass A1:

- ▶ Verified Security. Verification tools support dynamic security analysis. Example: Honeywell SCOMP military processor.

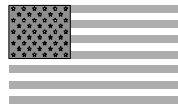


24

Genesis of Criteria

P The “Orange Book” (1985)

- ▶ U.S. Department of Defense & National Bureau of Standards (now NIST),
- ▶ *Trusted Computer System Evaluation Criteria* (TCSEC),



25

Spread of National Criteria

P ITSEC (1990)

- ▶ Commission of European Communities
- ▶ Information Technology Security Evaluation Criteria
- ▶ Harmonize preexisting German, British and French criteria, along with the U.S. Orange Book, and sought to take the best features of each and maintain maximized compatibility with all

P Canadian Trusted Computer Product Evaluation Criteria (1992)

- ▶ The Canadian Security Establishment

P The Federal Criteria for Information Technology Security (1992)

- ▶ U.S. NIST revision of the Orange Book



26

Common Criteria (1996)

P International Common Criteria Editorial Board



27

28