CSE2008: Operating Systems

L37: Security & Protection

VIT-AP
UNIVERSITY

Dr. Subrata Tikadar

SCOPE, VIT-AP University

Recap

- Introductory Concepts
- Process Fundamentals
- IPC
- CPU Scheduling Algorithms
- Multithreading Concepts
- Synchronization
- Deadlock
- Memory Management
- Virtual Memory Concepts
- Mass-Storage Management
- File System Management

Outline

- The Security Issue
- Protection Goal and Principle
- Protection Rings
- Domain of Protection
- Access Matrix
- Role-Based and Mandatory Access Control

The Security Issue

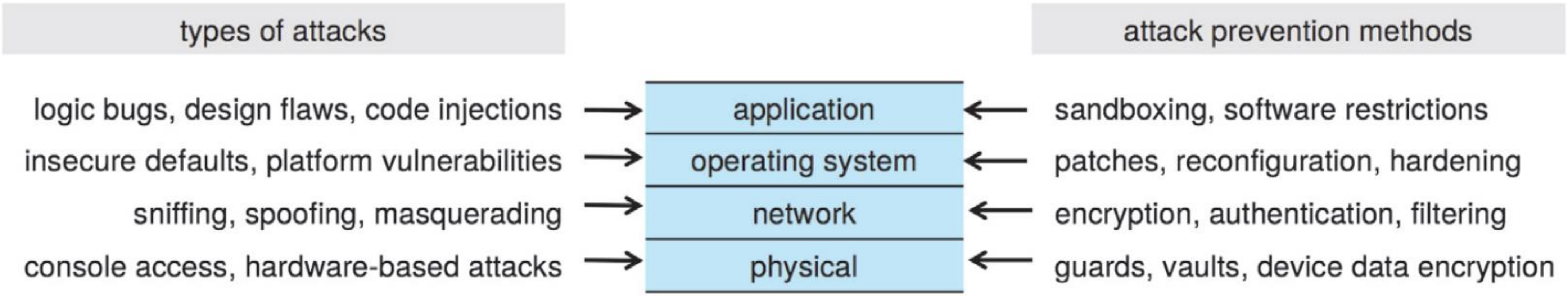
- System violation (misuse) category
  - **Breach of confidentiality.** This type of violation involves unauthorized reading of data (or theft of information). Typically, a breach of confidentiality is the goal of an intruder. Capturing secret data from a system or a data stream, such as credit-card information or identity information for identity theft, or unreleased movies or scripts, can result directly inmoneyfor the intruder and embarrassment for the hacked institution.
  - **Breach of integrity.** This violation involves unauthorized modification of data. Such attacks can, for example, result in passing of liability to an innocent party or modification of the source code of an important commercial or open-source application.
  - **Breach of availability.** This violation involves unauthorized destruction of data. Some attackers would rather wreak havoc and get status or bragging rights than gain financially. Website defacement is a common example of this type of security breach.
  - **Theft of service.** This violation involves unauthorized use of resources. For example, an intruder (or intrusion program) may install a daemon on a system that acts as a file server.
  - **Denial of service.** This violation involves preventing legitimate use of the system. Denial-of-service (DOS) attacks are sometimes accidental. The original Internet worm turned into a DOS attack when a bug failed to delay its rapid spread.

The Security Issue

- Four Levels of Security Measures
  - **Physical.** The site or sites containing the computer systems must be physically secured against entry by intruders. Both the machine rooms and the terminals or computers that have access to the target machines must be secured, for example by limiting access to the building they reside in, or locking them to the desk on which they sit.
  - **Network.** Most contemporary computer systems—from servers to mobile devices to Internet of Things (IoT) devices—are networked. Networking provides a means for the system to access external resources but also provides a potential vector for unauthorized access to the system itself. Further, computer data in modern systems frequently travel over private leased lines, shared lines like the Internet, wireless connections, and dial-up lines. Intercepting these data can be just as harmful as breaking into a computer, and interruption of communications can constitute a remote denial-of-service attack, diminishing users' use of and trust in the system.
  - **Operating system.** The operating system and its built-in set of applications and services comprise a huge code base that may harbor many vulnerabilities. Insecure default settings, misconfigurations, and security bugs are only a few potential problems. Operating systems must thus be kept up to date (via continuous patching) and "hardened"—configured and modified to decrease the attack surface and avoid penetration. The attack surface is the set of points at which an attacker can try to break into the system.
  - **Application.** Third-party applications may also pose risks, especially if they possess significant privileges. Some applications are inherently malicious, but even benign applications may contain security bugs. Due to the vast number of third-party applications and their disparate code bases, it is virtually impossible to ensure that all such applications are secure.

The Security Issue

- The Four-Layered Model of Security



types of attacks				attack prevention methods
logic bugs, design flaws, code injections	→	application	←	sandboxing, software restrictions
insecure defaults, platform vulnerabilities	→	operating system	←	patches, reconfiguration, hardening
sniffing, spoofing, masquerading	→	network	←	encryption, authentication, filtering
console access, hardware-based attacks	→	physical	←	guards, vaults, device data encryption

The Security Issue

- Types of Threats
  - Program Threats
    - Malware
    - Code Injection
    - Viruses and Worms
  - System and Network Threats
    - Attacking Network Traffic
    - Denial of Service
    - Port Scanning

Protection Goals

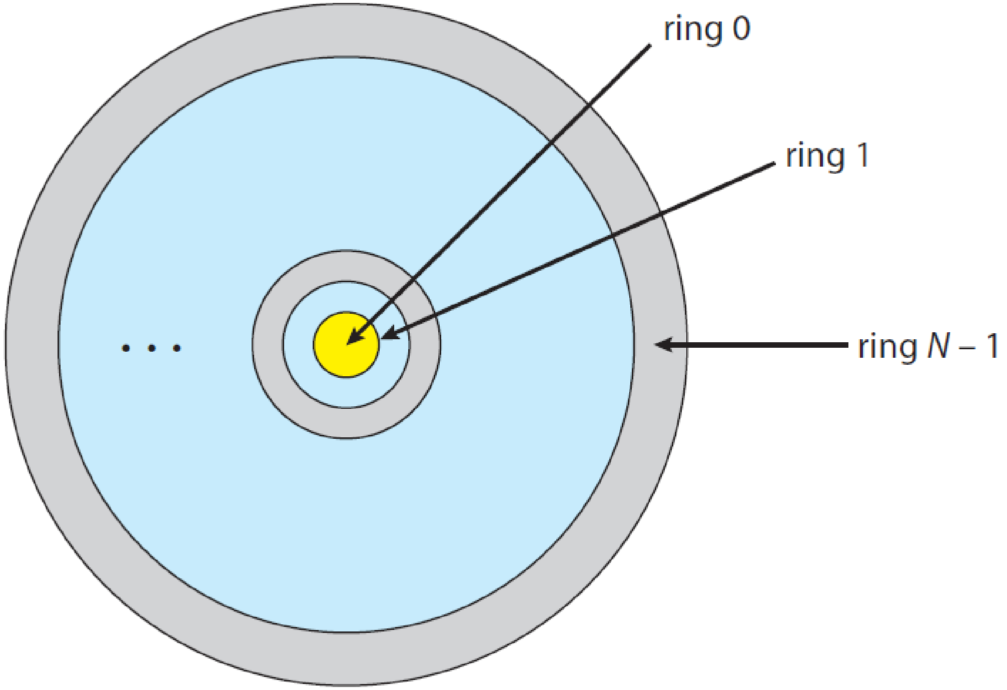
- To safely share a common logical name space, such as a directory of files, or a common physical name space, such as memory
- To increase the reliability of any complex system that makes use of shared resources and is connected to insecure communications platforms such as the Internet

Protection Principles

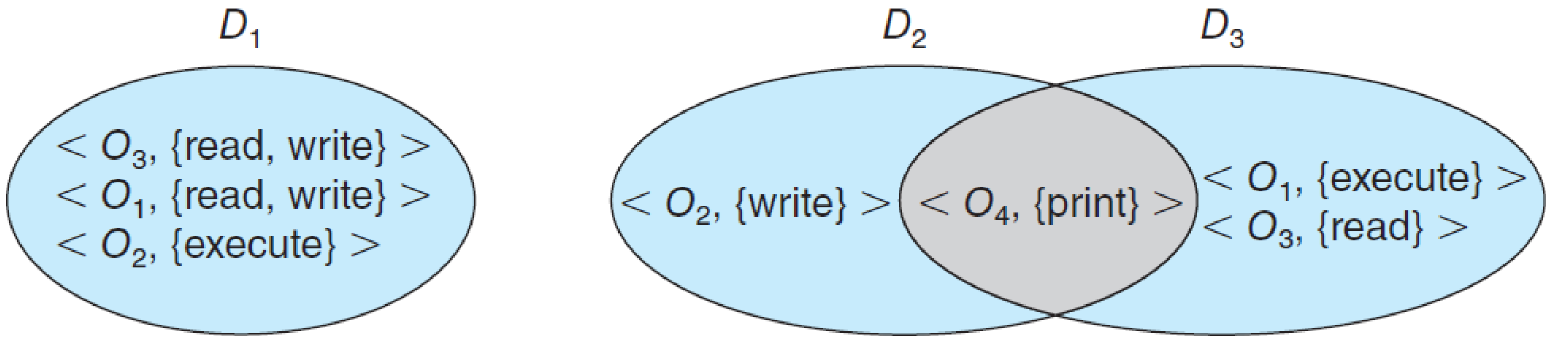
- Principle of least privilege
- Principle of careful use of access restrictions (audit trail)
- Defense in depth (multiple layers of protection - one on top of the other)

Protection Rings

- Protection-ring structure



Domain of Protection



- A domain can be realized in a variety of ways:
  - **Each user may be a domain.** In this case, the set of objects that can be accessed depends on the identity of the user. Domain switching occurs when the user is changed—generally when one user logs out and another user logs in.
  - **Each process may be a domain.** In this case, the set of objects that can be accessed depends on the identity of the process. Domain switching occurs when one process sends a message to another process and then waits for a response.
  - E**ach procedure may be a domain.** In this case, the set of objects that can be accessed corresponds to the local variables defined within the procedure.

Domain of Protection

- Multiple Domains:
  - Example – Multiple Domains in UNIX
    - The root user can execute privileged commands, while other users cannot.
    - However, Restricting certain operations to the root user can impair other users in their everyday operations

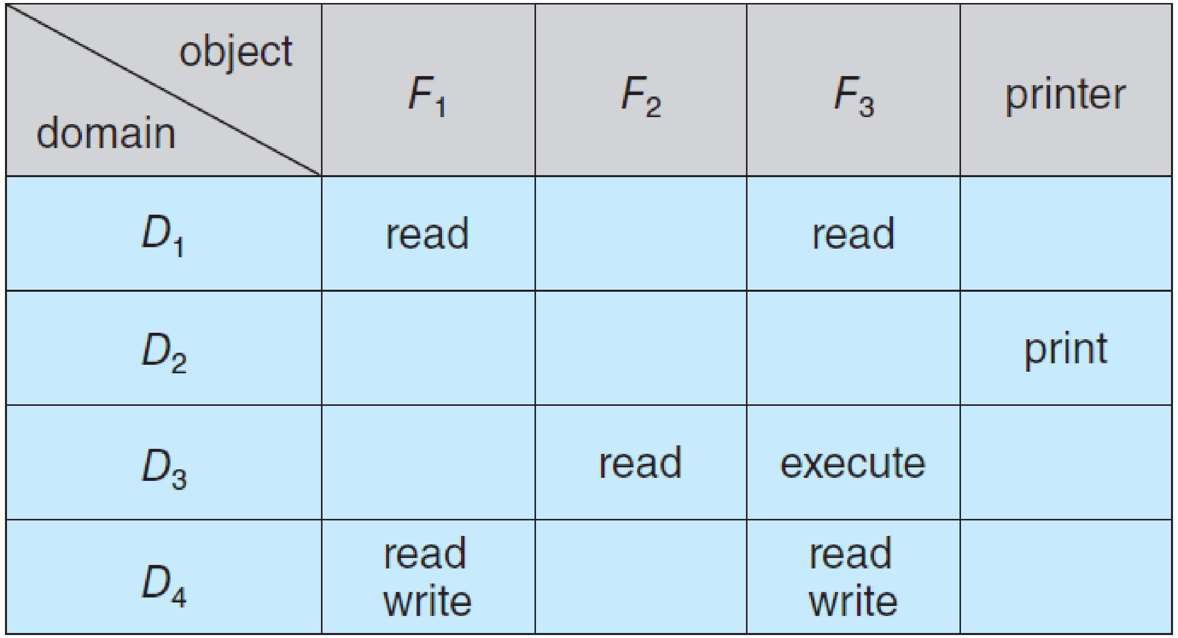
Solution → ***setuid bit*** (enabled through **chmod +s**)

Domain of Protection

- Problems in Multiple Domains:
  - Example – Multiple Domains in Andriod
    - Distinct user IDs are provided on a per-application basis.
    - When an application is installed, the **installd** daemon assigns it a distinct user ID (UID) and group ID (GID), along with a private data directory (/data/data/<appname>) whose ownership is granted to this UID/GID combination alone.

Access Matrix

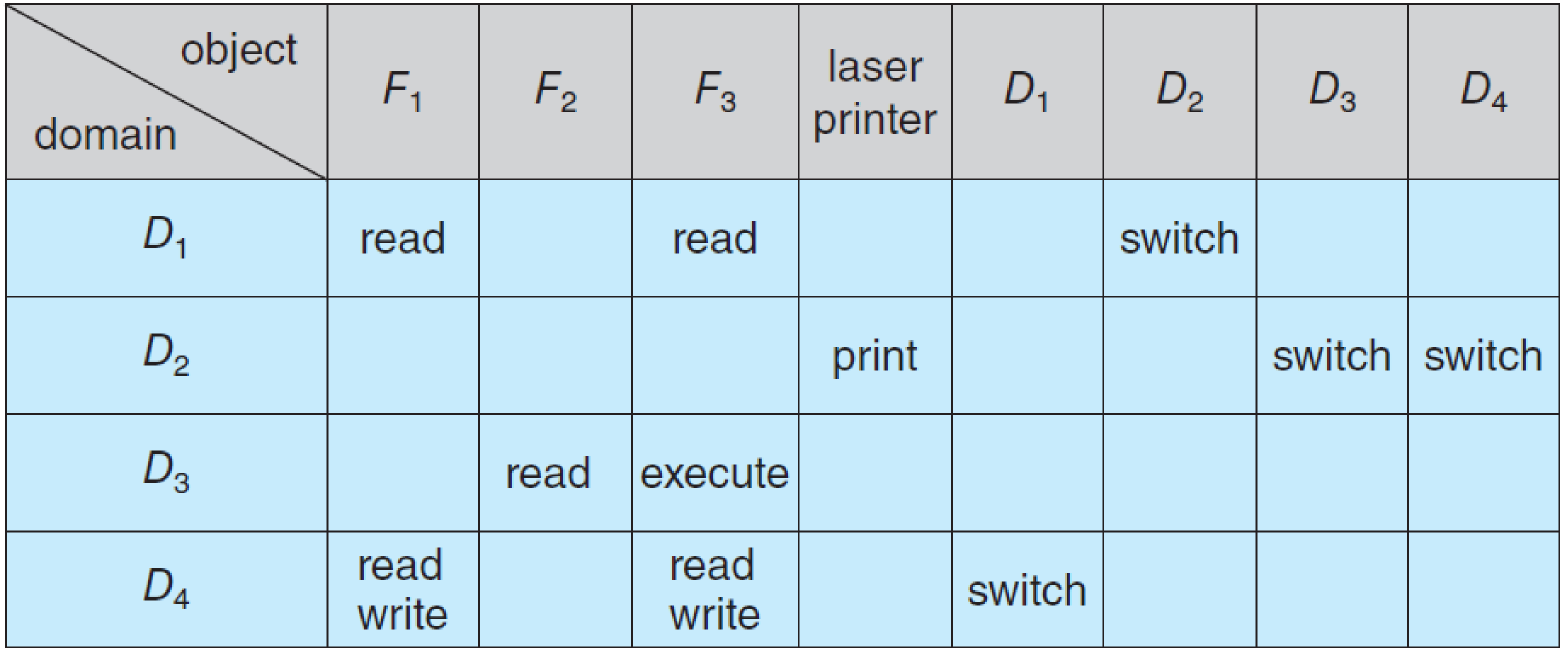
- The general model of protection can be viewed abstractly as a matrix



object domain	F_1	F_2	F_3	printer
D_1	read		read	
D_2				print
D_3		read	execute	
D_4	read write		read write	

Access Matrix

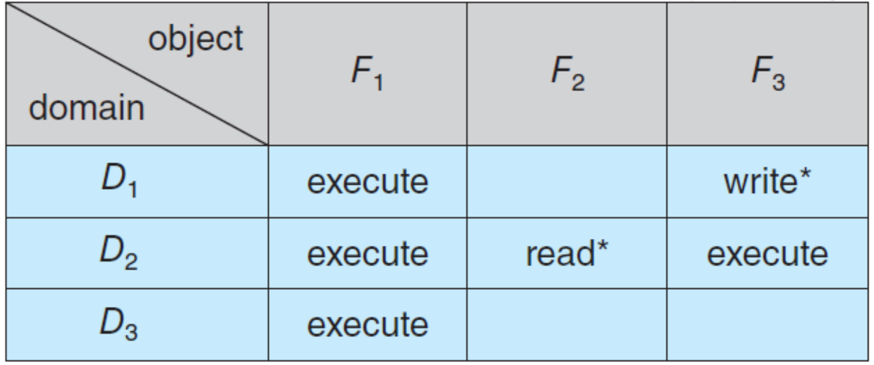
- Modified access matrix with domains as objects



object domain	F_1	F_2	F_3	laser printer	D_1	D_2	D_3	D_4
D_1	read		read			switch		
D_2				print			switch	switch
D_3		read	execute					
D_4	read write		read write		switch			

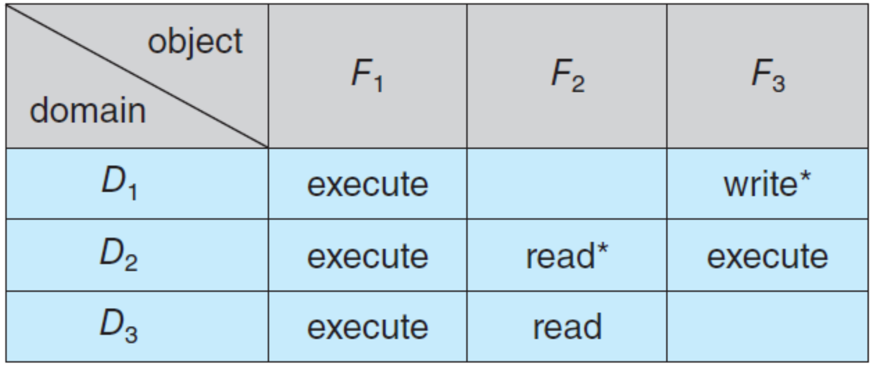
Access Matrix

- Modified access matrix with copy rights



object domain	F_1	F_2	F_3
D_1	execute		write*
D_2	execute	read*	execute
D_3	execute		

(a)



object domain	F_1	F_2	F_3
D_1	execute		write*
D_2	execute	read*	execute
D_3	execute	read	

(b)

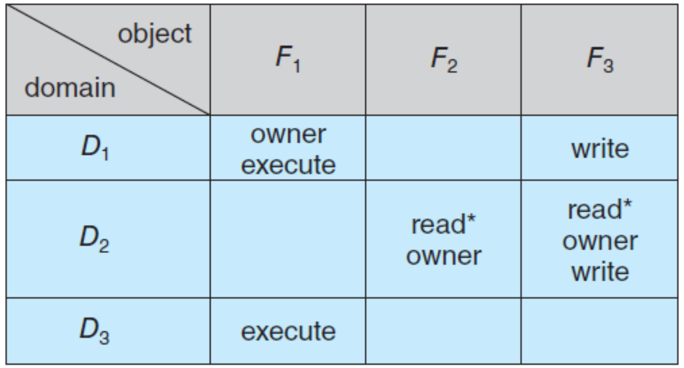
Two additional variants of this scheme:

1. A right is copied from access(i, j) to access(k,j); it is then removed from access(i, j). This action is a transfer of a right, rather than a copy.

2. Propagation of the copy right may be limited. That is, when the right R^* is copied from access(i, j) to access(k, j), only the right R (not R^*) is created. A process executing in domain D_k cannot further copy the right R.

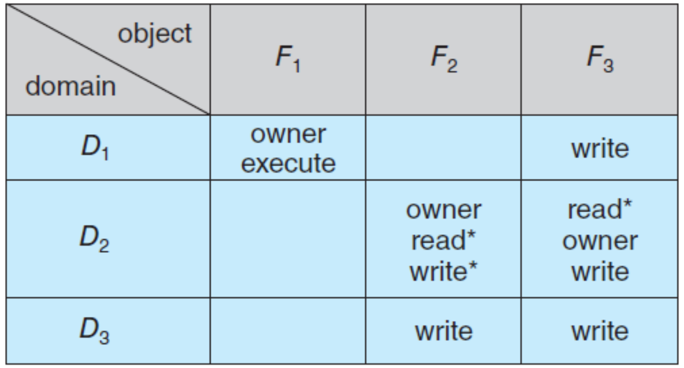
Access Matrix

- Modified access matrix with owner rights



object domain	F_1	F_2	F_3
D_1	owner execute		write
D_2		read* owner	read* owner write
D_3	execute		

(a)

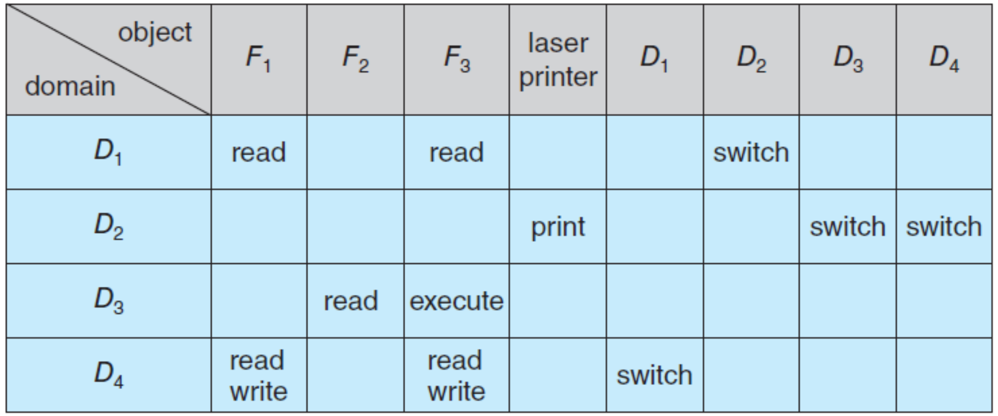


object domain	F_1	F_2	F_3
D_1	owner execute		write
D_2		owner read* write*	read* owner write
D_3		write	write

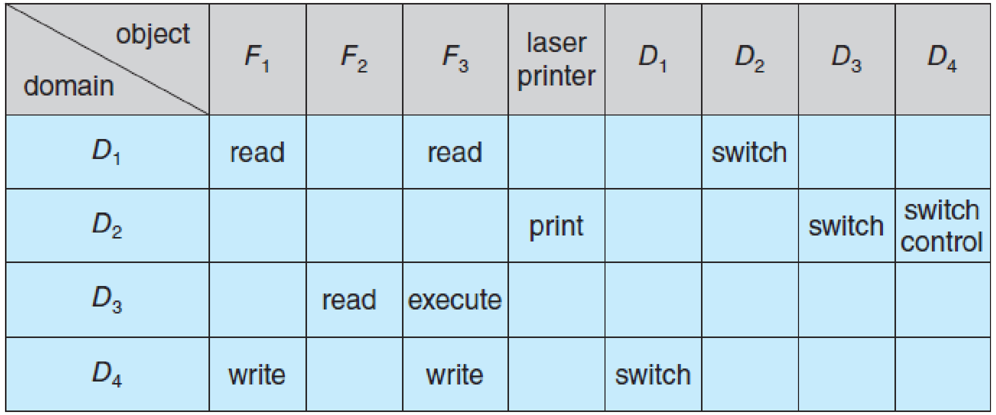
(b)

Access Matrix

- Further Modification of the access matrix



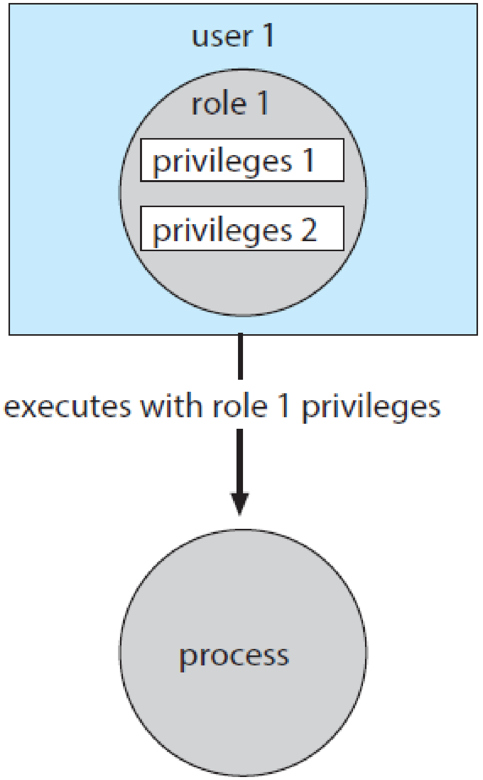
object domain	F_1	F_2	F_3	laser printer	D_1	D_2	D_3	D_4
D_1	read		read			switch		
D_2				print			switch	switch
D_3		read	execute					
D_4	read write		read write		switch			



object domain	F_1	F_2	F_3	laser printer	D_1	D_2	D_3	D_4
D_1	read		read			switch		
D_2				print			switch	switch control
D_3		read	execute					
D_4	write		write		switch			

Role-Based and Mandatory Access Control

- Role-Based and Mandatory Access Control



Reference

- Abraham Silberschatz , Peter B. Galvin, Greg Gagne, “Operating System Concepts”, Addison Wesley, 10th edition, 2018
  - Chapter 16: Section 16.1 – 16.6
  - Chapter 17: Section 17.1 – 17.9

Next

- QUIZ, Assignment, FAT

Thank You