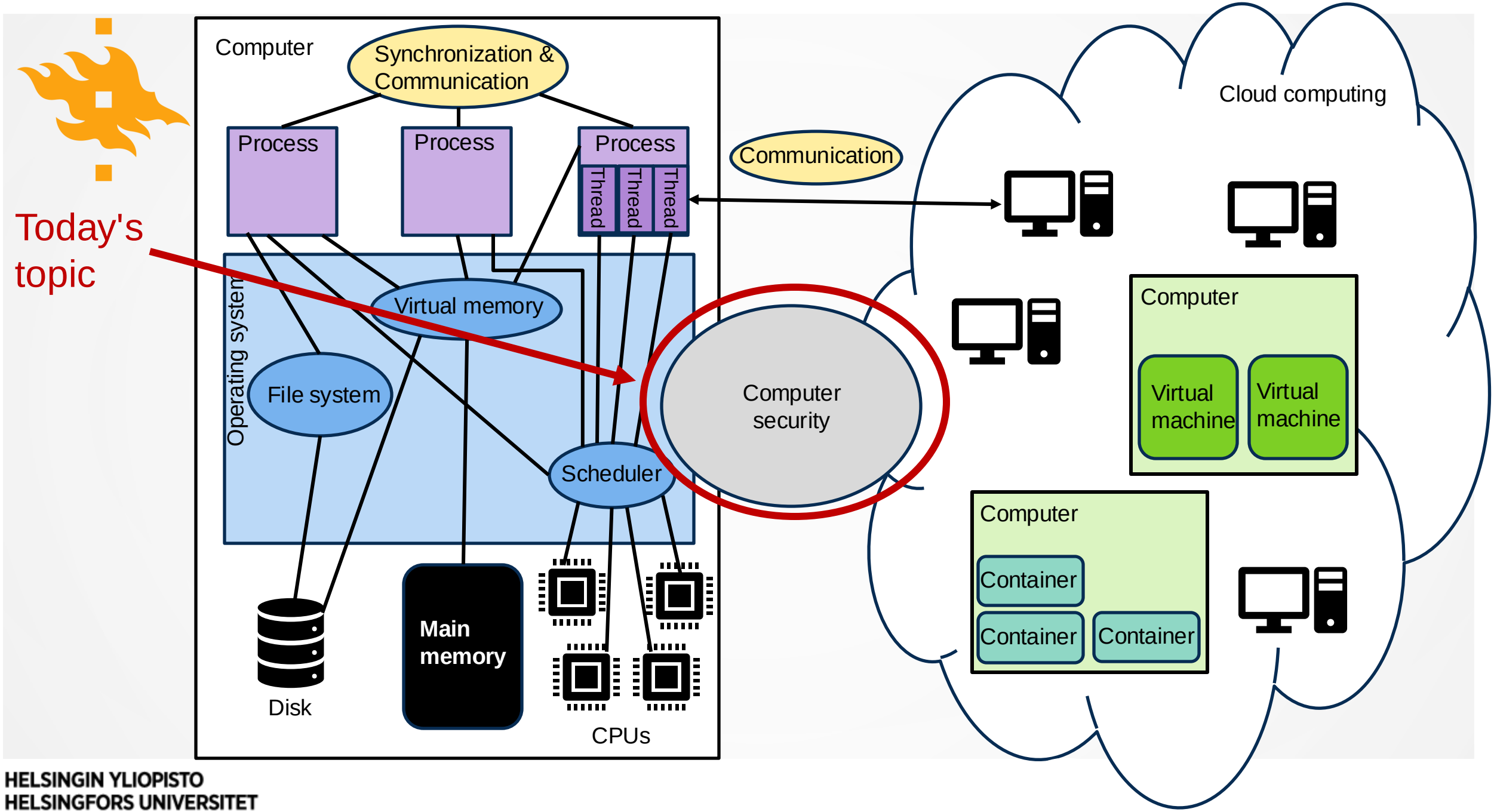
COMPUTING PLATFORMS

Security


Today's topic
Computer
Synchronization & Communication
Process
Process
Process
Thread
Thread
Thread
Operating system
Virtual memory
File system
Scheduler
Disk
Main memory
CPUs
Communication
Computer security
Cloud computing
Computer
Virtual machine
Virtual machine
Computer
Container
Container
Container

LEARNING OUTCOMES

- After today's lecture you
  - Know key issues of OS and cloud security
  - Can distinguish various types of intruder behavior patterns and understand different types of intrusion techniques used to breach computer security
  - Know different kinds of countermeasures for computer security threats
  - Understand design issues for file system security
  - Are able to describe and contrast two methods of access control

WHY SECURITY IS IMPORTANT IN THE CONTEXT OF OS AND CLOUD?

- Think about what OS / Cloud operator can do...
  - ...examine or alter any process's memory
  - ...read, write, delete, or corrupt data on any writeable persistent storage (hard disk, flash drives, cloud storage...)
  - ...change the scheduling or halt execution of a process / container / virtual machine
  - …send any message to anywhere, including altered versions of messages
  - ...enable or disable a peripheral device
  - …give any process access to any other process's resources
  - ...take away any resource a process controls
  - …respond to any system call with a maximally harmful lie

LAWS, REGULATIONS, AND ETHICS

- Laws governing data trespass and data communication crimes; criminal law (Finland)
  - "Intruding ... into a system … fine or prison max 2 years"
  - "… in a particularly methodical manner … fine or prison max 5 years"
  - "Just trying to intrude is punishable..."
- Policies, regulations, standards: GDPR, HIPAA, PCI-DSS,…
- Usage and privacy policies of companies / universities / …
  UH IT usage and privacy policy:
  - "Unauthorized acquisition or attempts to acquire data contained in the information systems is prohibited"
  - "User accounts shall not be used for the identification of security vulnerabilities, for unauthorized decryption or communications interception or distortion, or for invading any other systems, directories or services"
- Ethics for IT professionals (https://tivia.fi/toimiala/etiikan-ohjeet/)

KEY OBJECTIVES OF COMPUTER SECURITY

- **Confidentiality**
  - **Data confidentiality** assures that private and confidential information is not made available or disclosed to unauthorized individuals
  - **Privacy** assures that individuals control or influence what information related to them may be collected and stored and by whom and to whom that information may be disclosed
- **Integrity**
  - **Data integrity** assures that information and programs are changed only in a specified and authorized manner
  - **System integrity** assures that a system performs its intended functions in an unimpaired manner, free from deliberate or inadvertent unauthorized manipulation of the system
- **Availability**
  - Assures that systems work promptly and service to authorized users is not denied

SECURITY THREATS

- **Intruders**
  - An individual who has unauthorized access to a system or who misuses their access to a system
- **Malicious software**
  - Programs that exploit vulnerabilities in systems
- **Vulnerabilities**
  - Bugs (or sometimes features) in software that make attacks to a system possible
- **Service failure**
  - Denial of service (DoS) attacks, distributed denial of service (DDoS) attacks

INTRUDERS

- **Masquerader**
  - Individual not authorized to use the system
  - Penetrates a system's access control to exploit a legitimate user's account
- **Misfeasor**
  - Legitimate user who accesses data, programs, or resources for which such access is not authorized, or who is authorized for such access but misuses their privileges
- **Clandestine user**
  - Individual who seizes supervisory control of the system (in secret) and uses this control to evade auditing and access control or to suppress audit collection

MALICIOUS SOFTWARE (MALWARE)

- Programs that exploit vulnerabilities in computing systems
- Can be divided into two categories:
  - **Parasitic**: fragments of programs that cannot exist independently of some actual application program, utility, or system service. Examples include viruses and logic bombs
  - **Independent**: self-contained programs that can be scheduled and run by the operating system. Examples include worms and bots.

PARASITIC MALWARE

- **Viruses**: software that "infects" other programs by modifying them
  - Carries instructional code to self duplicate
  - Becomes embedded in a program
  - When an infected system is in contact with an uninfected piece of software, a fresh copy of the virus can pass into the new program
  - Infection can spread e.g. by swapping USB sticks
- **Logic bombs**: code embedded in some legitimate program that is set to "explode" when certain conditions are met
  - Once triggered a logic bomb may alter or delete data or entire files, cause a machine halt, or do some other damage

INDEPENDENT MALWARE

- **Worms**: program that can replicate itself and send copies from computer to computer over a network
  - Upon arrival, work may replicate and propagate again
  - Usually also performs some unwanted action
  - Actively seeks out more machines to infect and each infected machine acts as a launching pad for attacks on other machines
- **Bots**: program that secretly takes over another internet-attached computer and then uses that computer to launch attacks that are difficult to trace to the bot's creator
  - Typically planted on hundreds or thousands of computers belonging to unsuspected third parties
  - Collection of bots acting in a coordinated manner is called a botnet
  - Can be used, e.g., to launch denial-of-service attacks or send spam

VULNERABILITIES

- **Backdoor:** Secret entry point into a program that allows someone to gain access without going through the usual security access procedures
  - Maintenance hook is a backdoor used by programmers to debug and test programs
  - Becomes a threat when unscrupulous programmers use them to gain unauthorized access
- **Buffer overflow:** Can occur when a process attempts to store data beyond the limits of a fixed-sized buffer overwriting some other data
  - One of the most prevalent and dangerous types of security attacks

BUFFER OVERFLOW: EXAMPLE

```
char A[8] = "";
unsigned short B = 1979;

strcpy(A, "excessive");
```

- Program has two variables that are stored **adjacent in memory:** an 8-byte-long string buffer A and a two-byte integer B
- Initially, A contains nothing, and B contains the number 1979
- Next, the program attempts to store null-terminated string "excessive" with ASCII encoding in the buffer A
- "excessive" is 9 characters long and encodes to 10 bytes including the terminating null symbol.
- A can only take 8 bytes, strcpy does not check the lengths of the strings, and also the value of B is overwritten!

Example from https://en.wikipedia.org/wiki/Buffer_overflow

EXPLOITING BUFFER OVERFLOW

- Attacker needs to...
  - … **identify a buffer overflow vulnerability** in some program that can be triggered using externally sourced data that the attacker can control
  - … **understand how that buffer will be stored in the process memory**, and hence the potential for corrupting adjacent memory locations and potentially altering the flow of execution of a program

DEFENDING AGAINST BUFFER OVERFLOW

- Prevention
- Detection and aborting
- Countermeasure categories:
  - **Compile-time defense**: aim to harden programs to resist attacks in new programs
    Choice of programming language, safe coding techniques, safe libraries, stack protection mechanisms
  - **Run-time defense**: aim to detect and abort attacks in existing programs
    Executable address space protection, address space randomization, guard pages

COUNTERMEASURES

- Firewalls
- Intrusion detection systems
- Proper authentication systems and access control
- Regular maintenance
- Employee training
- Physical security management
- ...

FIREWALLS

- **All traffic from inside to outside, and vice versa, must pass through firewall**
  - Achieved by physically blocking all access to local network except via the firewall
- **Only authorized traffic**, as defined by the local security policy, will be **allowed to pass**
- Firewall is **immune to penetration**
  - Use of a **hardened system** with a secured operating system
  - **Trusted computer systems** are suitable for hosting a firewall and often required in government applications

INTRUSION DETECTION SYSTEMS

- RFC 4949 (Internet Security Glossary) define **intrusion detection** as a **security service that monitors and analyzes system events** for the purpose of finding and providing real-time or near real-time warning of attempts to access system resources in an unauthorized manner
- Intrusion detection systems (IDS) can be classified as
  - Host-based IDS (monitor the characteristics of a single host and the events occurring within that host for suspicious activity)
  - Network-based IDS (monitor network traffic for particular network segments or devices and analyzes network, transport, and application protocols to identify suspicious activity)

INTRUSION DETECTION: BASIC PRINCIPLES

- Intrusion detection is based on the assumption that the **behavior of an intruder differs from that of a legitimate user in ways that can be quantified**
- If intrusion detected quickly enough, intruder can be identified and ejected from the system before any damage is done or any data is compromised
- Intrusion detection also enables the collection of information about intrusion techniques that can be used to strengthen intrusion prevention measures

IDS COMPONENTS

- **Sensors**
  - Responsible for collecting data
  - Input for sensor may be any part of a system that can contain evidence of an intrusion
  - Types of input include, e.g., network packets, log files, system call traces
- **Analyzers**
  - Receive input from one or more sensors or from other analyzer
  - Responsible for determining if an intrusion has occurred
  - Provides guidance about what actions to take as a result of an intrusion
- **User interface**
  - Enables a user to view output or control the behavior of the system

AUTHENTICATION

- In most computer security contexts, **user authentication is the fundamental building block and the primary line of defense**
  - RFC 4949 defines user authentication as the process of verifying an identity claimed by or for a system entity
- Authentication process consists of two steps
  - **Identification**: presenting an identifier to the security system
  - **Verification**: presenting or generating authentication information that corroborates the binding between the entity and the identifier. I.e., proving entity is who it claims to be

MEANS OF AUTHENTICATION

- Something an individual …
  - … **knows** (password, personal identification number (PIN), or answers to a prearranged set of questions)
  - … **possesses** (electronic keycards, smart cards, and physical keys; referred to as **token**)
  - … **is** (recognition by fingerprint, retina, and face; **static biometric authentication**)
  - … **does** (recognition by voice pattern, handwriting characteristics, and typing rhythm; **dynamic biometrics**)
- All means have weaknesses -> **multi-factor authentication** (MFA)

ACCESS CONTROL

- Implements **security policy that specifies who or what may have access to each specific system resource** and the type of access that is permitted in each instance
- Mediates between a user and system resources, such as applications, OS, firewall, routers, files, databases
- Security administrator maintains **authorization database** specifying what type of access to which resources is allowed by a user
  - **Access control function** consults this database to determine whether to grant access
- **Auditing function** monitors and keeps a record of user accesses to system resources

ACCESS CONTROL: FILE SYSTEMS

- Identifies a user to the system
- Each user has a profile that specifies permissible operations and file accesses
- OS can then enforce rules based on the user profile

FILE SYSTEMS: ACCESS RIGHTS

- **None**: User may not even learn of the existence of the file, much less access it (user not allowed to read the directory that includes this file)
- **Knowledge**: User can determine that the file exists and who owns it (can then ask the owner for additional access rights)
- **Execution**: User can load and execute a program but cannot copy it (proprietary programs)
- **Reading**: User can read the file for any purpose, including copying
- **Appending**: User can add data to the file (often only at the end) but cannot modify or delete any of the file's contents
- **Updating**: User can modify, delete, and add to the file's data (writing the file initially, rewriting it completely or in part, and removing all or a portion of the data)
- **Changing protection**: User can change the access rights granted to other users
- **Deletion**: User can delete the file from the file system

USER ACCESS RIGHTS

- Owner
  - Usually initial creator of the file
  - Has full rights
  - May grant rights to others
- Specific users
  - Individual users who are designated by user ID
- User groups
  - Set of users who are not individually defined
- All
  - All users who have access to the system
  - Public files

FILE SYSTEM SECURITY WITH ACCESS MATRIX

- Basic elements are
  - **Subject:** an entity capable of accessing objects
  - **Object:** anything to which access is controlled
  - **Access rights:** the way in which an object is accessed by a subject

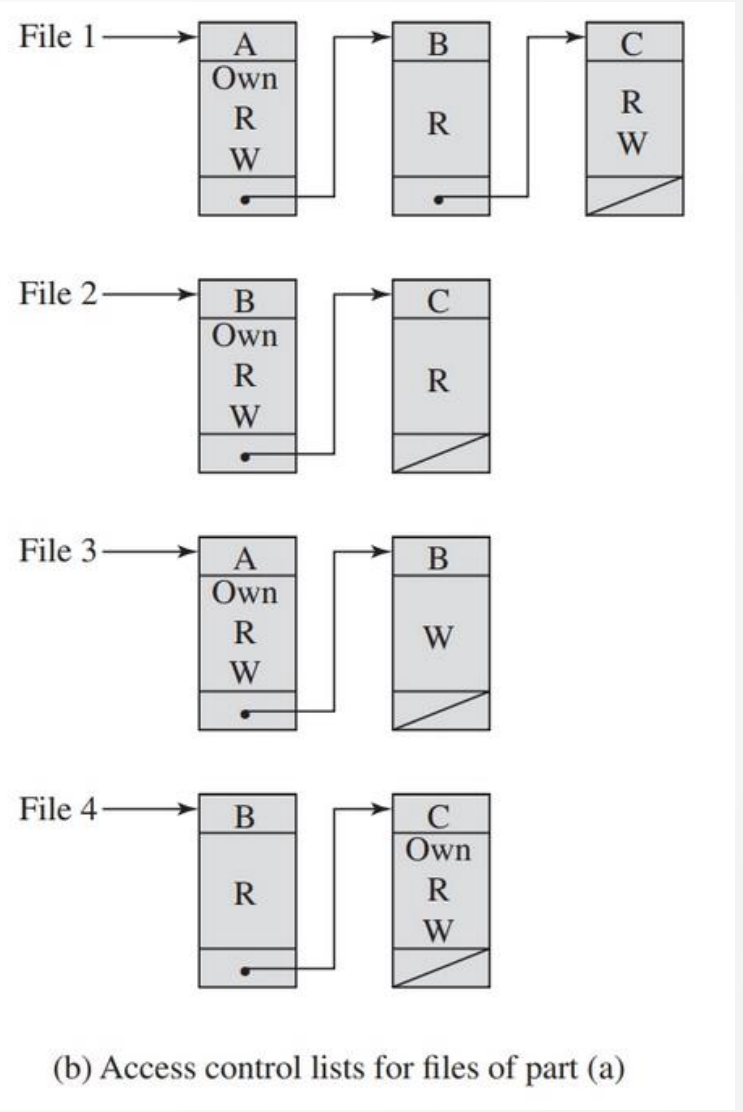
		Objects			
		File 1	File 2	File 3	File 4
Subjects	User A	Own Read Write		Own Read Write	
	User B	Read	Own Read Write	Write	Read
	User C	Read Write	Read		Own Read Write

(a) Access matrix

Figure 15.3 from [Stallings, Operating systems: Internals and design principles, 9th ed]

ACCESS CONTROL LISTS

- For objects
- Matrix may be decomposed by access matrix columns, yielding access control lists
- Access control list lists users and their permitted access rights

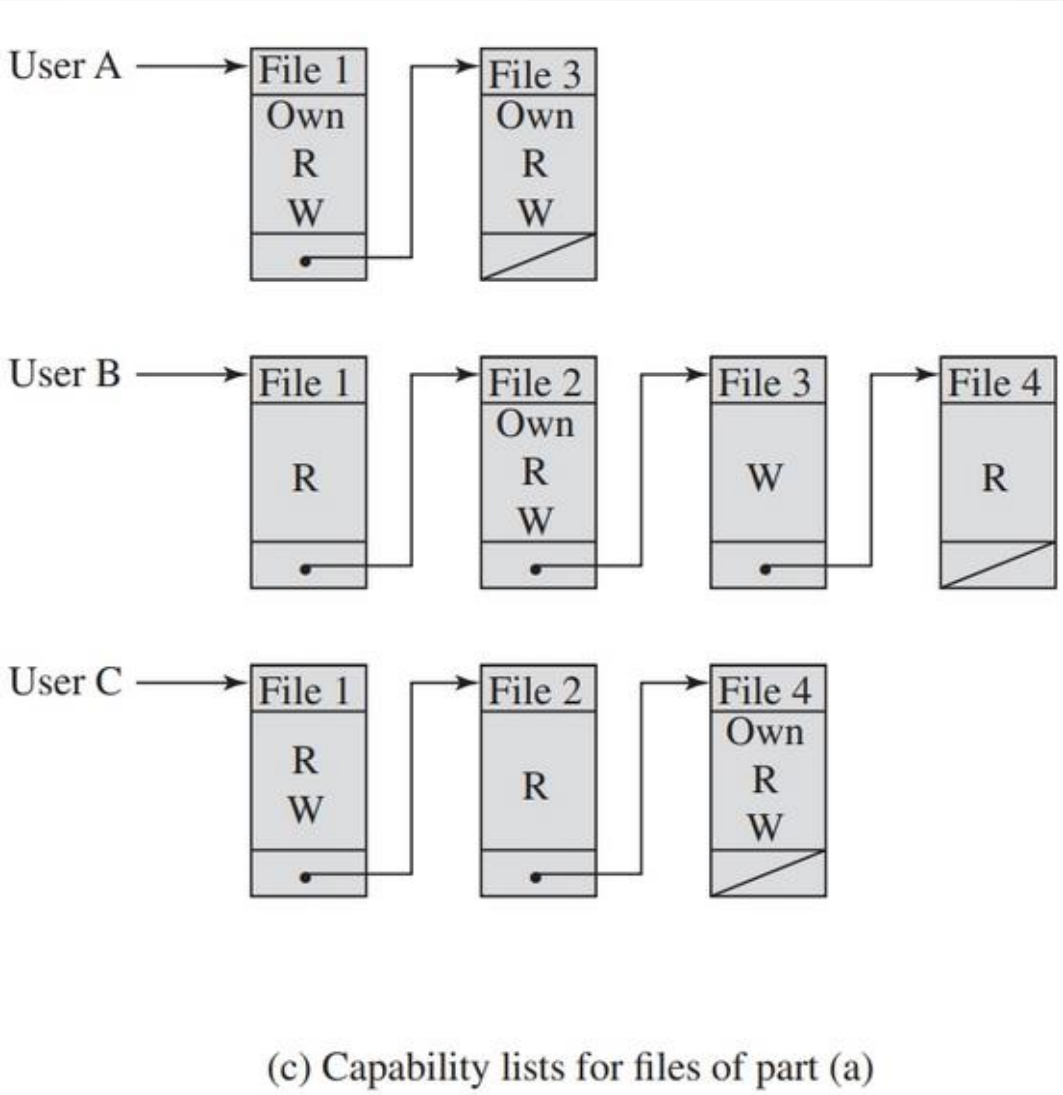


(b) Access control lists for files of part (a)

Figure 15.3 from [Stallings, Operating systems: Internals and design principles, 9th ed]

CAPABILITY LISTS

- Per each user
- Decomposition by rows yields **capability tickets**
- Capability ticket specifies authorized objects and operations for a user
- Need to be unforgeable
  - Encryption
  - System memory



(c) Capability lists for files of part (a)

Figure 15.3 from [Stallings, Operating systems: Internals and design principles, 9th ed]

ACCESS CONTROL POLICIES

- What kind of access is permitted, under what circumstances, and by whom
- **Discretionary access control (DAC)**: Based on the identity of the requestor and on access rules stating what requestors are allowed to do
- **Role-based access control (RBAC)**: Based on the roles that users have within the system, and on rules stating what accesses are allowed to users in given roles
- Attribute-based access control: Based on attributes of the user, the resource to be accessed, and current environmental conditions

DAC: EXTENDED ACCESS CONTROL MATRIX

		Objects								
		Subjects			Files		Processes		Disk drives	
		S_1	S_2	S_3	F_1	F_2	P_1	P_2	D_1	D_2
Subjects	S_1	Control	Owner	Owner control	Read *	Read owner	Wakeup	Wakeup	Seek	Owner
	S_2		Control		Write *	Execute			Owner	Seek *
	S_3			Control		Write	Stop			

* — Copy flag set

Figure 15.4 **Extended Access Control Matrix**

Figure from [Stallings, Operating systems: Internals and design principles, 9th ed]

ROLE BASED ACCESS CONTROL (RBAC)

- **Based on the roles** that users assume in a system **rather than user's identity**
- Models define a role as a **job function within an organization**
- System assigns access rights to roles instead of individual users
- Users are assigned to different roles, either statically or dynamically, according to their responsibilities

USERS, ROLES, AND RESOURCES

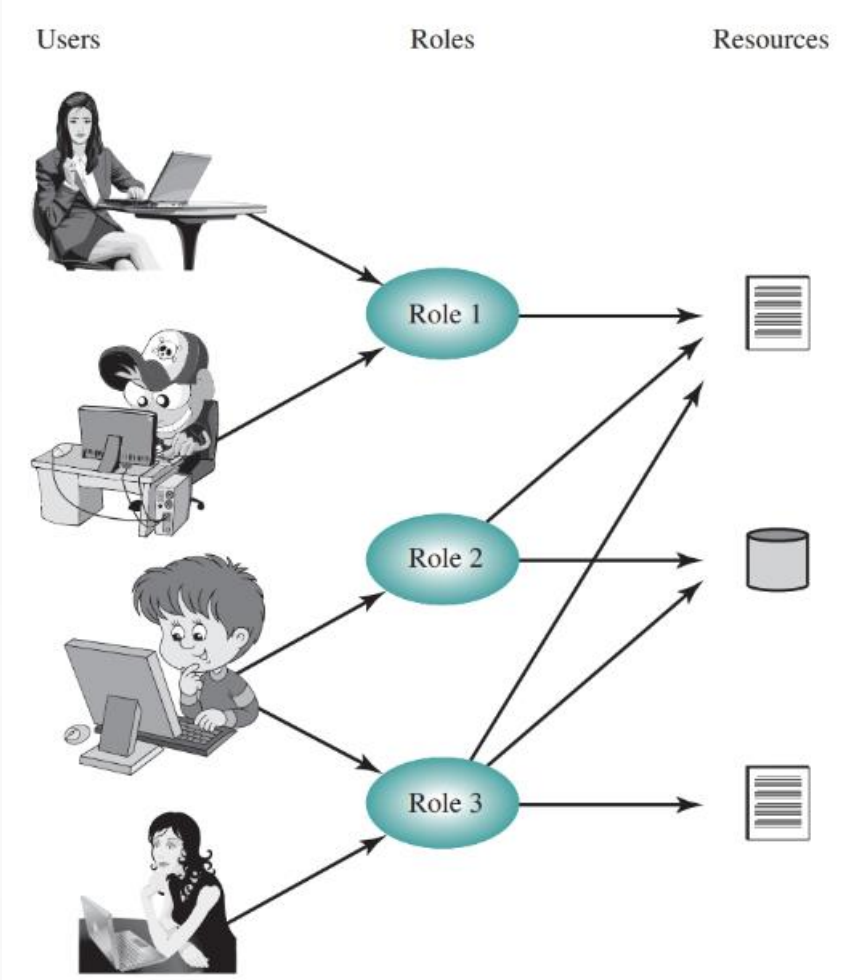
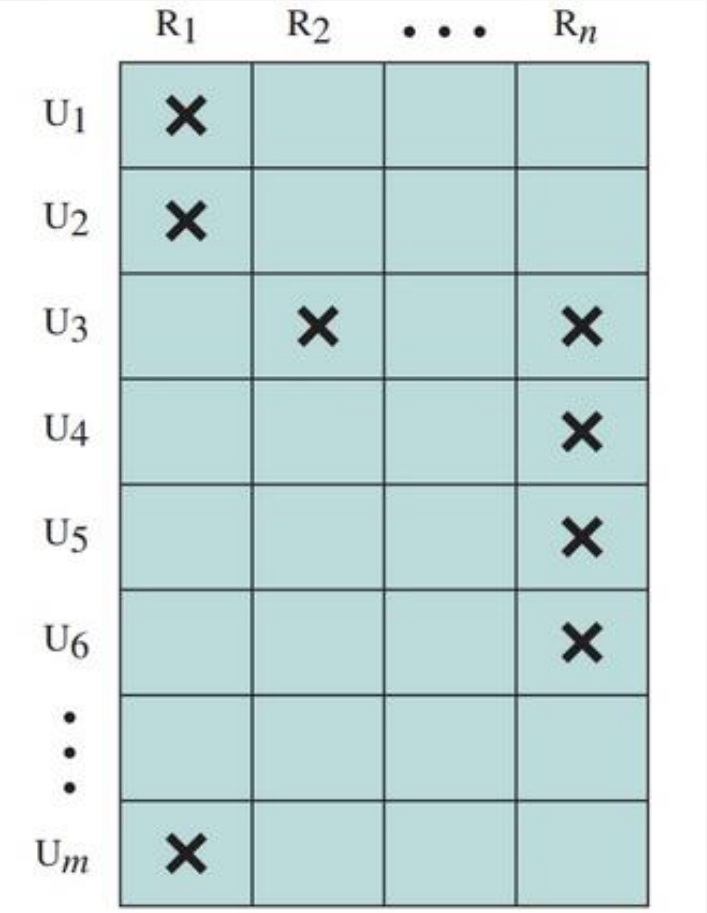


Figure 15.6 Users, Roles, and Resources

Figure from [Stallings, Operating systems: Internals and design principles, 9th ed]

ACCESS CONTROL MATRIX REPRESENTATION OF RBAC



Objects

Roles	R1	R2	Rn	F1	F1	P1	P2	D1	D2
R1	Control	Owner	Owner control	Read *	Read owner	Wakeup	Wakeup	Seek	Owner
R2		Control		Write *	Execute			Owner	Seek *
• • •									
Rn			Control		Write	Stop			

Figure 15.7 Access Control Matrix Representation of RBAC

Figure from [Stallings, Operating systems: Internals and design principles, 9th ed]

SETUP AND MAINTENANCE

- Operating system hardening
- Regular maintenance
- Data backups and archive

OPERATING SYSTEM HARDENING

- Install and patch the OS
- Harden and configure the OS to adequately address the identified security needs of the system by
  - Removing unnecessary services, applications, and protocols
  - Configuring users, groups, and permissions
  - Configuring resource controls
- Install and configure additional security controls, such as antivirus, host-based firewalls, and intrusion detection systems if needed
- Test the security of the operating system to ensure that the steps taken adequately address its security needs

OS INSTALLATION: INITIAL SETUP AND PATCHING

- System security begins with the installation of the OS
- Ideally, new systems should be constructed on a **protected network**
- Initial installation should comprise the **minimum necessary for the desired system**, with additional software packages included only if they are required for the function of the system
- Careful selection when installing any additional device driver code (executes with full kernel privileges but is often supplied by a third party)

SECURITY MAINTENANCE

- **Monitoring and analyzing logging** information
- Performing **regular backups**
- Recovering from **security compromises**
- Regularly **testing system security**
- Using appropriate **software maintenance processes to patch and update all critical software**, and to monitor and revise configuration as needed

DATA BACKUP AND ARCHIVE

- Performing **regular backups** of data on a system is another critical control assisting with **maintaining the integrity of the system and user data**
- Needs and policy related to backups should be determined during the system planning stage
  - Should copies be kept online or offline?
  - Should copies be stored locally or transported to a remote site?
- **Backup**: process of making copies of data at **regular intervals**, allowing the recovery of lost or corrupted data over relatively short time periods of a few hours or some weeks
- **Archive**: Process of r**etaining copies of data over extended periods of time** (months or years) in order to meet legal and operational requirements to access past data

SUMMARY

- Threats: Intruders, malicious software, vulnerabilities, service availability
- Countermeasures: Firewalls, intrusion detection systems, authentication,…
- Access control:
  - File system access control
  - Access control policies
- Security maintenance