CS 3002 Information Security

Fall 2022

1. Explain key concepts of information security such as design principles, cryptography, risk management,(1)

2. Discuss legal, ethical, and professional issues in information security (6)
3. Analyze real world scenarios, model them using security measures, and apply various security and risk management tools for achieving information security and privacy (2)
4. Identify appropriate techniques to tackle and solve problems of real life in the discipline of information security (3)
5. Understand issues related to ethics in the field of information security(8)

ISO/IEC 27001: 2013

Week # 10 – Lecture # 25, 26, 27

28th, 29th, ??nd Rabi ul Awwal, 1444

25th, 26th, 27th October 2022

Dr. Nadeem Kafi Khan

Lecture # 25 - LAB

- SQL Injection Lab (task # 1)
  - Submission on GCR

Lecture # 26

- Discretionary Access Control
  - Access Matric
  - Access Control Lists
  - Capability List
- Virus Propagation Mechanism
- Components of a Virus
- Four phases of the life of a virus

4.3 DISCRETIONARY ACCESS CONTROL

- A discretionary access control scheme is one in which an entity may be granted access rights that permit the entity, by its own will, to enable another entity to access some resource.
- A general approach to DAC, as exercised by an operating system or a database management system, is that of an access matrix.
- Figure shows a simple example of an access matrix. Thus, user A owns files 1 and 3 and has read and write access rights to those files. User B has read access rights to file 1, and so on.

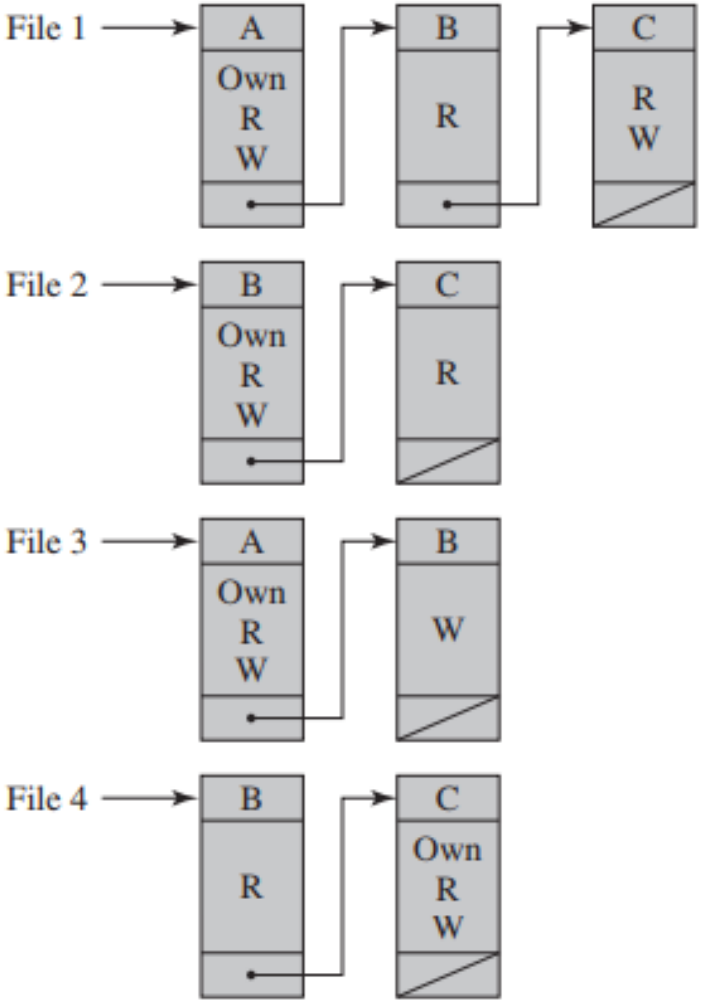
		OBJECTS			
		File 1	File 2	File 3	File 4
SUBJECTS	User A	Own Read Write		Own Read Write	
	User B	Read	Own Read Write	Write	Read
	User C	Read Write	Read		Own Read Write

(a) Access matrix

Figure 4.2 **Example of Access Control Structures**

4.3 DISCRETIONARY ACCESS CONTROL

- The matrix may be decomposed by columns, yielding **access control lists** (ACLs) (see Figure).
  - For each object, an ACL lists users and their permitted access rights. The ACL may contain a default, or public, entry. This allows users that are not explicitly listed as having special rights to have a default set of rights.
  - The default set of rights should always follow the rule of least privilege or read-only access, whichever is applicable.
  - Elements of the list may include individual users as well as groups of users. When it is desired to determine which subjects have which access rights to a particular resource, ACLs are convenient, because each ACL provides the information for a given resource.
  - However, this data structure is not convenient for determining the access rights available to a specific user.

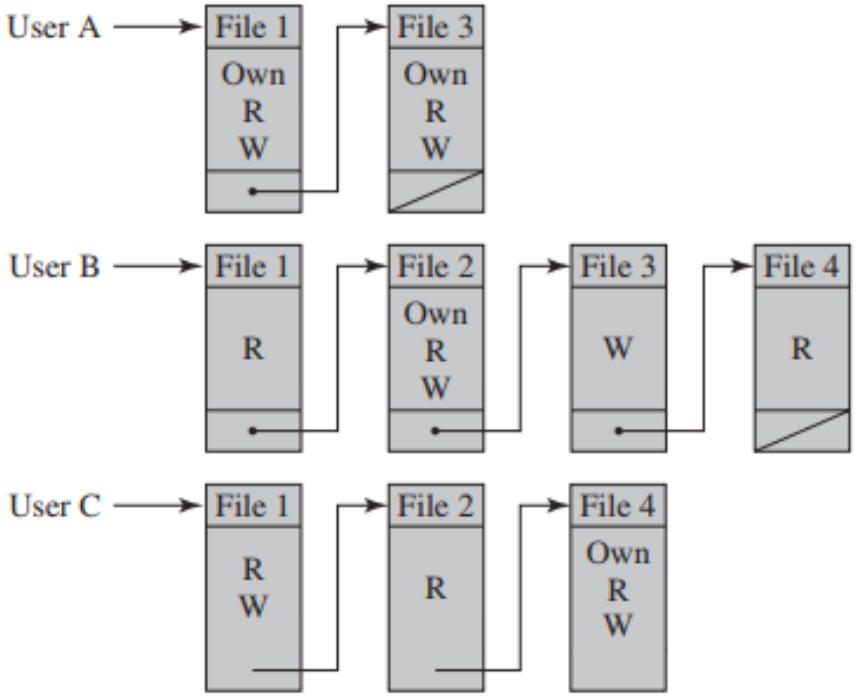


(b) Access control lists for files of part (a)

Figure 4.2 Example of Access Control Structures

4.3 DISCRETIONARY ACCESS CONTROL

- Decomposition by rows yields capability tickets (see Figure).
  - A capability ticket specifies authorized objects and operations for a particular user. Each user has a number of tickets and may be authorized to loan or give them to others.
  - Because tickets may be dispersed around the system, they present a greater security problem than access control lists.
  - The integrity of the ticket must be protected, and guaranteed (usually by the operating system).
    - In particular, the ticket must be unforgeable. One way to accomplish this is to have the operating system hold all tickets on behalf of users. These tickets would have to be held in a region of memory inaccessible to users.
  - The convenient and inconvenient aspects of capability tickets are the opposite of those for ACLs. It is easy to determine the set of access rights that a given user has, but more difficult to determine the list of users with specific access rights for a specific resource.



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

6.3 PROPAGATION—INFECTED CONTENT—VIRUSES

- This category of malware propagation concerns parasitic software fragments that attach themselves to some existing executable content.
  - The fragment may be machine code that infects some existing application, utility, or system program, or even the code used to boot a computer system.
  - Recent addition is some form of scripting code, typically used to support active content within data files such as Microsoft Word documents, Excel spreadsheets, or Adobe PDF documents.
- Computer virus infections formed the majority of malware seen in the early personal computer era.
  - The term "computer virus" is still often used to refer to malware in general, rather than just computer viruses specifically.

6.3 PROPAGATION—INFECTED CONTENT—VIRUSES

- Most viruses carry out their work that is specific to a particular operating system and/or to a particular hardware platform.
  - Thus, they are designed to take advantage of the details and weaknesses of particular systems. Macro viruses however target specific document types, which are often supported on a variety of systems.
- A virus can infect some or all of the other files on that system with executable content when it executes. Depending on the access permissions it has.
  - Thus Viral infection can be completely prevented by blocking the virus from gaining entry in the first place.
- Unfortunately, prevention is extraordinarily difficult because a virus can be part of any program outside a system.
  - Thus, unless one is content to take an absolutely bare piece of iron and write all one's own system and application programs, one is vulnerable.
  - Many forms of infection can also be blocked by denying normal users the right to modify programs on the system.

6.3 PROPAGATION—INFECTED CONTENT—VIRUSES

A computer virus (and many types of current malware) has three components:

- **Infection mechanism:** The means by which a virus spreads or propagates, enabling it to replicate. The mechanism is also referred to as the **infection vector**.
- **Trigger:** The event or condition that determines when the payload is activated or delivered, sometimes known as a **logic bomb**.
- **Payload:** What the virus does, besides spreading. The payload may involve damage or may involve benign but noticeable activity.

6.3 PROPAGATION—INFECTED CONTENT—VIRUSES

A typical virus goes through the following four phases during its lifetime:

- **Dormant phase:** The virus is idle. The virus will eventually be activated by some event, such as a date, the presence of another program or file, or the capacity of the disk exceeding some limit. Not all viruses have this stage.
- **Propagation phase:** The virus places a copy of itself into other programs or into certain system areas on the disk. The copy may not be identical to the propagating version; viruses often morph to evade detection. Each infected program will now contain a clone of the virus, which will itself enter a propagation phase.
- **Triggering phase:** The virus is activated to perform the function for which it was intended. As with the dormant phase, the triggering phase can be caused by a variety of system events, including a count of the number of times that this copy of the virus has made copies of itself.
- **Execution phase:** The function is performed. The function may be harmless, such as a message on the screen, or damaging, such as the destruction of programs and data files.

Lecture # 27

- **Quiz # 2**