COMPUTER SECURITY ACCESS CONTROL

CHAPTER 6: ACCESS CONTROL

BY: DR. ALA BERZINJI

OBJECTIVES

- Introduce the fundamental model of access control.
- Look at a few sets of access operations.
- Present essential access control structures.

ACCESS CONTROL

After identification and authentication, what can the user do?

ACCESS CONTROL

You have now logged on to the system. You create new files and want to protect your files. Some of your files may be public, some only intended for a restricted audience and some may be private.

You need:

- **A language for expressing your intended access control policy and**
- **You need mechanisms that enforce access control.**

POLICIES AND MECHANISMS

access control mechanisms

- how do we control access (*implementation)*

access control policies(security policies)

- what should be allowed or not (*specification); the security we expect the system to enforce*

security models

- formalization of a security policy

BACKGROUND

Before immersing your self in the details of access control, consider the way computer systems and the use of computer systems have developed over the last decades.

- **Computer systems manipulate data and mediate access to shared resources like memory, printers etc.**
- **They have to provide access control to data and resources, although primarily for reasons of integrity and not so much for confidentiality.**

BACKGROUND

- **Traditional multiuser operating systems offer generic services to a considerable variety of users.**
- **These operating systems have simple and generic access operations and are not concerned with the meaning of the files they handle.**

BACKGROUND

- Modern desktop operating systems support individual users in performing their job.
- You find complex access operations witch are very much application specific.
- You are witnessing the transition from general purpose computer systems to (flexible) special purpose computer systems.

AUTHENTICATION AND AUTHORIZATION

To access control, we first have to develop a suitable terminology:

- The very nature of ‘access’ suggests that there is:
- An active entity,
- a subject or a principle,
- accessing a passive object with some specific access operation,

while a reference monitor grants or denies access.

ACCESS CONTROL MECHANISMS

Subject

- **Active part, users, processes, methods, domains, ...**

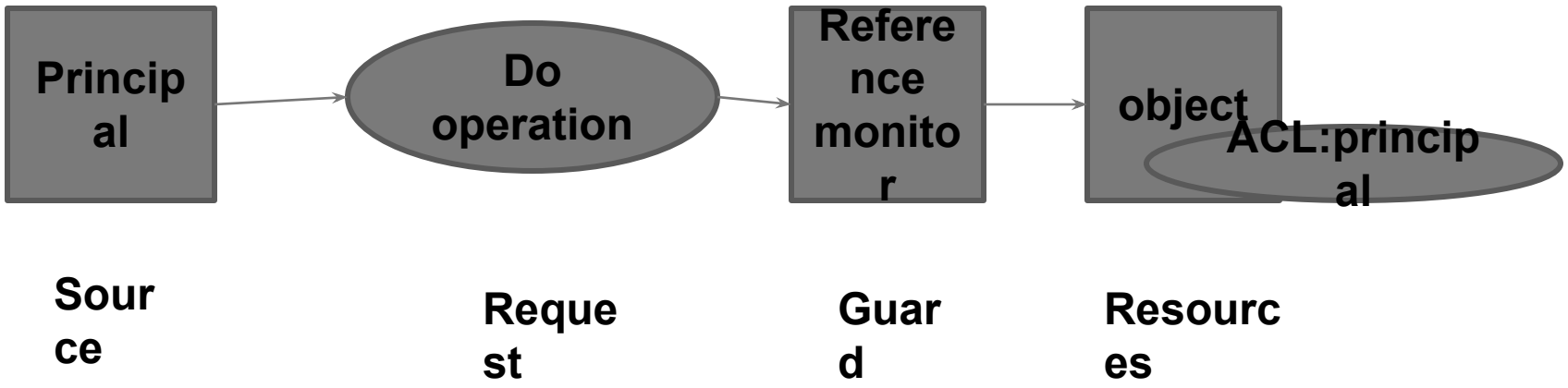
Object

- **Passive part, memory, device, file, process, method, ...**

Access operation

- **Action, read, write, execute, ...**

AUTHENTICATION AND AUTHORIZATION



The Fundamental Model of Access Control

AUTHENTICATION AND AUTHORIZATION

For the purpose of access decisions, subjects have to be bound to principals. When a subject requests access to a protected object the reference monitor checks whether the principal bound to the subject has the right to access the object.

AUTHENTICATION AND AUTHORIZATION

A typical example of a principal in an operating system is

- A user identity

The principals permitted to access a given object could be stored in an access control list ACL attached to the object.

AUTHENTICATION AND AUTHORIZATION

A typical example of a subject is

- **A process running under a user identity**

Principals need not represent human users or attributes of human users.

AUTHENTICATION AND AUTHORIZATION

Typical objects are:

-Files or

-Resources

like memory, printers, or nodes in a computer network.

There is not mean to be a clear distinction between subjects and objects in the sense that every entity in the system has to be either a subject or an object.

AUTHENTICATION AND AUTHORIZATION

An entity can be a subject in one access request and an object in another.

The terms subject and object merely distinguish between the active and passive party in an access request.

Subjects and objects presents two options for focusing control. You can either specify:

- What a subject allowed to do, or
- What may be done with an object.

ACCESS OPERATIONS

Depending on how you look at a computer system, access operations vary from reading and writing to physical memory to method calls in an object-oriented system.

Comparable systems may use different access operations and, even worse, attach different meanings to operations which appear to be the same.

ACCESS OPERATIONS

We will examine some typical sets of access operations taken from important early contributions in this area:

- Access Modes
- Access rights of the Bell-LaPdula Model
- Current Operating systems

ACCESS MODES

One the most elementary level, a subject may

- **observe an object or**
- **alter an object.**

We therefore define the two access modes.

Observe: look at the contents of an object.

Alter: change the contents of an object.

ACCESS RIGHTS OF THE BELL-LAPADULA MODEL

The Bell-LaPadula model has four access rights

- Execute
- Read
- Append(sometimes also referred to as blind write)
- Write

ACCESS RIGHTS OF THE BELL-LAPADULA MODEL

	execute	append	read	write
observe			X	X
alter		X		X

Access Rights in the Bell-LaPadula Model

ACCESS RIGHTS OF THE BELL-LAPADULA MODEL

To understand the rationale for this definitions, consider how a multiuser operating system controls access files.

Files can be opened for read access or for write access. In this way the operating system can avoid potential conflicts like two users simultaneously writing to the same file.

ACCESS RIGHTS OF THE BELL-LAPADULA MODEL

For reasons of efficiency, write access usually includes read access.

For example, a user editing a file should not be asked to open it twice, once for read and once for write.

Hence its meaningful to define the write right so that it includes the observe and alter modes.

ACCESS RIGHTS OF THE BELL-LAPADULA MODEL

Operating systems can use files, e.g. programs, without opening these files at all. Hence the introduction of the execution right which includes neither observe nor alter mode.

CURRENT OPERATING SYSTEMS

A current example is the Unix operating system in which access control policies are expressed in terms of three operations. There are:

- Read: reading from a file
- Write: writing to a file
- Execute: executing a (program) file.

CURRENT OPERATING SYSTEMS

These operations differ from those of the Bell-LaPadula model. For example, in Unix write access does not imply read access. When applied to a dictionary, the access operations take the following meanings:

- Read: list dictionary contents
- Write: create or rename a file in the directory
- Execute: search the directory.

CURRENT OPERATING SYSTEMS

Unix controls who can create and delete files by controlling write access to the file's directory.

The access rights specified for a file are changed by modifying the file's entry in its directory.

OWNERSHIP

When discussing access control, we also must state who is in charge of setting security policies. There are two fundamental options:

- We can define an owner for each resource and let the owner decree who is allowed to have access.
- A system-wide policy decrees who is allowed to have access. For obvious reasons, such a policy may be called mandatory.

OWNERSHIP

Most operating systems support the concept of ownership of a resource and consider ownership when making access control decisions. They may include operations that redefine the ownership of a resource.

ACCESS CONTROL STRUCTURES

We have to state which access operations are permitted. We have to decide on the structures to use for capturing security policies, while facing two competing requirements.

ACCESS CONTROL STRUCTURES

The access control structure should help to express your desired access control policy.

You should be able to check that your policy has been captured correctly.

Some access control structures are:

- Access control matrix
- Capabilities
- Access control list

ACCESS CONTROL MATRIX

Access control matrix also referred as access permission matrix.

Access rights can be defined individually for each combination of subject and object quite simply in the form of an access control matrix.

ACCESS CONTROL MATRIX

	bill.doc	edit.exe	fun.com
Alice	-	{execute}	{execute, read}
Bill	{read, write}	{execute}	{execute, read, write}

An Access Control Matrix

ACCESS CONTROL MATRIX

شرح للجدول
في السلايد سابق

Bill.doc may be read and written by Bill while Alice has no access at all.

Edit.exe can be executed both by Alice and Bill but otherwise they have no access.

Fun.com can be executed and read by both users, but only Bill can write to the file.

ACCESS CONTROL MATRIX

Implementation

- Row-wise: capability lists(CL)
- Column-wise: Access Control List(ACL)

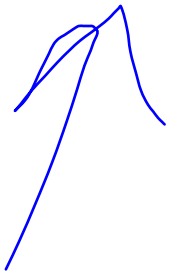
CAPABILITIES

You would hardly implement an access control matrix directly. There is a choice between two obvious options. Access rights can be kept with the subjects or with the object.

In the first case every subject is given a capability, an forgeable token that specifies this subject's access rights.

This capability corresponds to the subject's row in the access control matrix.

CAPABILITIES



The access right of our previous example given as capabilities are:

- Alice's capability: edit.exe:execution;fun.com:execute,read.
- Bill's capability: bill.doc:read,write; edit.exe:execute;fun.com:execute,read,write.

CAPABILITIES

Capabilities are associated with discretionary access control. When a subject creates a new object, it can give other subjects access to this object by granting them the appropriate capabilities.

Also, when a subject(process)calls another subject, it can pass on its capability, or parts thereof, to the invoked subject.

CAPABILITIES

Capabilities are by no means a new concept but up to now they have not become a widely used security mechanism. This is mainly due to the complexity of security management and to the traditional orientation of operating systems towards managing objects.

CAPABILITIES

It is difficult to get an overview of who has permission to access a given object.

It is very difficult to revoke a capability

CAPABILITIES

When you decide to employ capabilities, you also have to give some consideration to their protection. Where do you store capabilities?

- If on a single computer system, then it is feasible to rely only on integrity protection by the Operating system.
- If capabilities travel over a network, you also need cryptographic protection.

ACCESS CONTROL LIST

Stores the access rights to an object with the object itself.

The ACL are a typical security feature of commercial operating systems. The access rights of our previous example, given in the form of ACLs, are:

- ACL for bill.doc: Bill: read, write;
- ACL for edit.exe: Alice: execute; Bill: execute;
- ACL for fun.com: Alice: execute, read; Bill: execute, read, write;

ACCESS CONTROL LIST

Management of access rights based only on individual subjects can be rather cumbersome.

It is therefore common to place users in groups and to drive access rights also from a user's group.