DATA PROTECTION, PRIVACY AND ANONYMITY

Another one.

Fabio Lorenzato

Contents

Chapter 1

Definitions

A fundamental requirement of any information management system is the protection of data and resources from unauthorized disclosure, modification, or service denial, to ensure that the CIA properties are sound. To achieve this, **access control** mechanisms ensure that only authorized accesses occur within a system and to it's resources.

Access control is a major building block of network security, regulating access of legitimate users to resources of a system, but by itself, it cannot prevent the occurrence of cyber-attacks, nor mitigate them, but like a firewall it uses rules to allow or deny access to resources. However, effective protection schemes cannot abstract from the definition of how access to system resources shall take

1.1 Definitions from Standards

The concept of access control is formally defined in several standards:

- **NISTIR 7298**: Access control is ”the process of granting or denying specific requests to obtain and use information and related information processing services, or to enter specific physical facilities (*which we will not deal with in this course*).”

- **RFC 4949**: Access control is ”a process by which use of system resources is regulated according to a security policy and is permitted only by authorized entities (users, programs, processes, or other systems).”

1.2 Access Control Security Requirements

Access control mechanisms should allow specification of:

1. Authorized users and their permissions.

2. The resources they can access.

3. The duration and conditions of access.

4. The specific operations they are permitted to perform.

A comprehensive set of security requirements is outlined in NIST SP 800-171, including:

Basic Security Requirements

1. Restrict system access to authorized users, processes, and devices.
2. Limit user access to only authorized transactions and functions.

Derived Security Requirements

1. Enforcement of information flow restrictions in accordance to the policies.
2. Separation of duties to prevent collusion.
3. Implementation of the principle of least privilege.
4. Prevention of non-privileged users from executing privileged functions.

   The system must perform authorization checks will still ensuring that the privacy of the user is maintained.
5. Limit unsuccessful logon attempts.
6. Use of cryptographic mechanisms for securing remote access.
7. Authorization and encryption for wireless access.
8. Control over external system connections and mobile devices.
9. Terminate a user session after some conditions or period of inactivity.

The NIST mapped all the possible scenarios, the first 3 are the most important, but it's important that as many possible scenarios are mapped to ensure that the system is secure.

1.3 Access Control Principles

Access control is a core aspect of computer security, as such it's important to follow well established guidelines. RFC 4949 defines computer security as ”measures that implement and assure security services in a computer system, particularly those that assure access control service.”

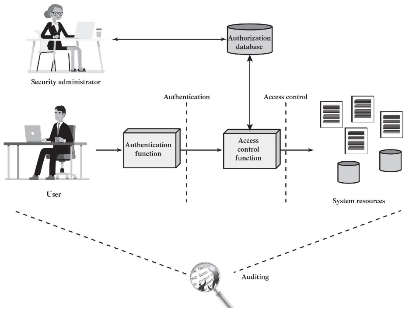


Figure 1.1: Access Control Schema in brief

1.4 Relationship with Other Security Functions

Access control interacts with several other security functions, which occurs at different stages of the security process:

1. **Authentication**: Verifies user credentials before granting access.
2. **Authorization**: Determines permissions based on policies and trust levels. This function determines who is trusted for a given purpose.
3. **Audit**: Verifies that the policies of the company are fully implemented and that the access control mechanisms are working as expected.

In this scenario, there are different actors. The user accesses the systems resources after being authenticated by the authentication function The security administrators manage an authorization database, while authentication functions verify users before granting access. Auditing provides an independent review to maintain security compliance.

1.5 Authentication and Audit

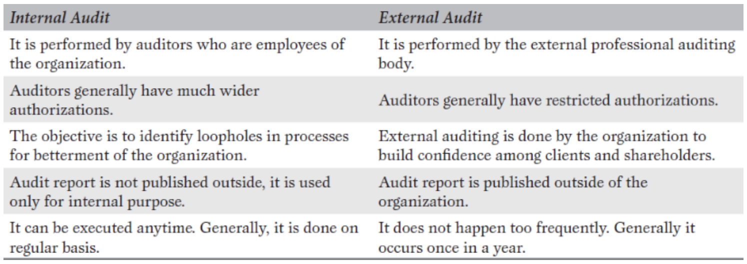
Authentication ensures that a user or process is verified before accessing system resources. Auditing plays a crucial role in maintaining security by monitoring activities.

1.5.1 Internal and External Audit

IT enterprises conduct two types of audits:

Internal Audit is performed within the organization to identify risks related to performance, security, and compliance. Internal auditors monitor mitigation efforts to improve the organization's security posture.

External Audit is conducted by independent professionals, often Certified Public Accountants (CPAs), who assess an organization's security and compliance. External audit reports are vital for stakeholders and clients, and generally comes with some kind of certification granted by a third party.



Internal Audit	*External Audit*
It is performed by auditors who are employees of the organization.	It is performed by the external professional auditing body.
Auditors generally have much wider authorizations.	Auditors generally have restricted authorizations.
The objective is to identify loopholes in processes for betterment of the organization.	External auditing is done by the organization to build confidence among clients and shareholders.
Audit report is not published outside, it is used only for internal purpose.	Audit report is published outside of the organization.
It can be executed anytime. Generally, it is done on regular basis.	It does not happen too frequently. Generally it occurs once in a year.

Figure 1.2: Internal and External Audit

Keep in mind that when we refer to auditing, cyber security auditing, of course, we can not only analyze the access control procedures, but also other aspects, like the security computer, the network security configuration, the software security computation, and so on. All these aspects can provide a feedback to improve the authorization database and the access control mechanisms.

1.6 Access Control Mechanisms

> An access control mechanism **regulates interactions** between users and system resources (applications, files, databases, etc.).

The process of access control begins with user authentication, which verifies the identity of the user requesting access to a specific resource. Once authenticated, the system determines the user's permissions and privileges, taking into account the policies specified by security administrators. This information is stored in an authorization database that maps user identities to resources and services.

The access control function relies on this database to make decisions about granting or denying requests. While this is a straightforward scheme, there are complexities to consider. For instance, coordinating multiple databases and technologies can be challenging, especially for small organizations with limited resources.

In many cases, security administrators must configure each authorization mechanism individually, which can lead to inconsistencies and difficulties in maintaining coherence across the system. Furthermore, as businesses increasingly rely on cloud providers like Amazon, the need for a unified electronic access mechanism (EAM) becomes crucial.

In some scenarios, there may not be a single EAM that can coordinate all authorization mechanisms, leading to fragmentation and potential security risks. However, this is where the coordination with the database comes into play – the security administrator specifies the policy in one unique point, which all other components must adhere to.

Operating systems incorporate built-in access control functions, supplemented by security add-ons and application-specific controls. External devices, such as firewall, can also provide access control services, which is different from the access control on the users.

All those ideas are then conveyed in three main concepts:

- Security mechanism
- Security policy
- Security model

1.6.1 Security Mechanisms

Security mechanisms enforce policies, which are defined trough the policies and formally stated by the model, through software and hardware functions.

1.6.2 Access Control Policies

Access control policies define high level rules for regulating user access. They specify what types of access are permitted, under what conditions, and by whom. These policies can be formalized into **authorization databases**.

Separation between mechanism and policy

As for every high-level/low-level separation, the separation between mechanism and policy allows to define the implementation of the system indipendetly of the protection requirements. It is also possible to compare different access control policies as well as different mechanisms that enforce the same policy.

One of the most important aspects is the possibility to design mechanisms that enforce multiple policies: if a mechanism is designed to enforce a single policy and be tied to it, a change in the policy would require changing the whole access control system.

1.6.3 Access Control Policy Model

It provides a formal representation of the access control security policy and its working. The formalization allows the proof of properties on the security provided by the access control system being designed.

1.7 Subjects, Objects, and Access Rights

Typically, a specific policy is characterized by the concepts of subjects, objects, and access rights, or at least Discretionary Access Control (DAC) and Role-Based Access Control (RBAC) are.

Objects include files, databases, messages, and programs. The number and type of protected objects depend on system complexity and security requirements.

Subjects represent active entities such as users, applications, and processes. Subjects are accountable for their actions, with audit trails recording relevant activities.

Actions refer to access operations on objects form subjects that have access rights, such as Read, Write, Execute, Delete, Create, and Search. Read access includes the ability to copy or print, while write access includes read access.

1.8 Access Control Policy Models

Access control policies are categorized into:

Discretionary Access Control (DAC): Grants access based on user identity and explicit authorization rules. Users can delegate their access privileges to others.

Mandatory Access Control (MAC): Enforces access restrictions based on security labels and clearance levels. Users cannot override these restrictions.

Role-Based Access Control (RBAC): Grants access based on organizational roles rather than individual identities. Users inherit permissions from their assigned roles.

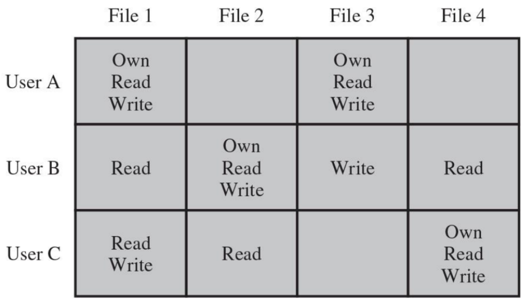
Attribute-Based Access Control (ABAC): Grants access based on attributes of users, objects, and environmental conditions, offering a flexible and dynamic approach.

These four policies are not mutually exclusive. An access control mechanism may employ two or even all three of these policies to cover different classes of system resources.

1.8.1 Discretionary Access Control (DAC)

> Discretionary policies enforce access control on the basis of the **identity** of the requestors and **explicit access rules** that establish who can, or cannot, execute which actions on which resources.

Usually, this kind of policy is expressed by an access matrix, like the one in figure 1.3. The rows represent the subjects, the columns the objects, and the cells the access rights of the subjects on the objects.



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

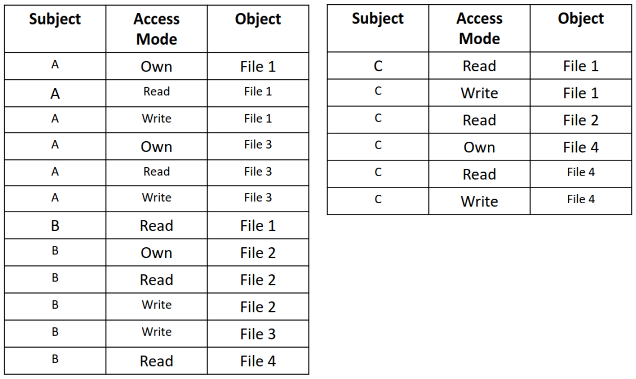
Figure 1.3: Access Matrix

This matrix is a simple way to represent the access control policy, but not a good way to implement it. The matrix is sparse(there could be many empty cells), and it is not efficient to store it in memory, without even considering the management of the redundancy of the information.

For a more practical approch there are 3 different alternatives.

Authorization Tables

Authorization tables are a more efficient way to represent access control policies. They store access rules in a more compact way, without leaving empty cells, like in figure 1.4.



Subject	Access Mode	Object
A	Own	File 1
A	Read	File 1
A	Write	File 1
A	Own	File 3
A	Read	File 3
A	Write	File 3
B	Read	File 1
B	Own	File 2
B	Read	File 2
B	Write	File 2
B	Write	File 3
B	Read	File 4

Subject	Access Mode	Object
C	Read	File 1
C	Write	File 1
C	Read	File 2
C	Own	File 4
C	Read	File 4
C	Write	File 4

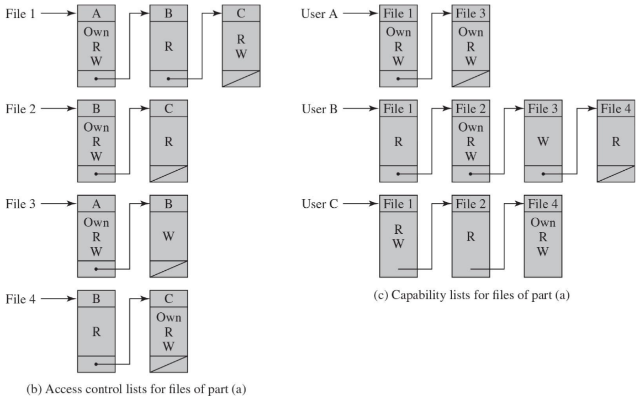
Figure 1.4: Authorization Table

It's possible to further reduce the redundancy of the data by splitting the column in different table and using join operations to reconstruct the original table if needed.

Access Control Lists (ACLs) and Capability Lists

With an Access Control List (ACL), each object has a list of users and their access rights. This list is stored in the object's metadata, and is checked whenever a user requests access to the object.

A similar concept is the Capability List, where each user has a list of objects and their access rights.



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

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

Figure 1.5: Access Control List

1.8.2 Mandatory Access Control (MAC)

Mandatory Access Control (MAC) policies are based on **security labels** (which indicate the sensitivity or criticality of the information/resource) and **clearance levels**, or security permission, which are assigned to subjects and indicate that the subject can access the resources.

Mandatory Access Control was originally introduced in critical domains, like military-grade once, but DAC has become more prevalent nowadays, even though this model is more maintainable. The logic is shown in figure 1.6. The administrator specifies the confidential level of the data, define define which users are allowed to access on specific level of confidentiality and security, and the system will enforce the policy.

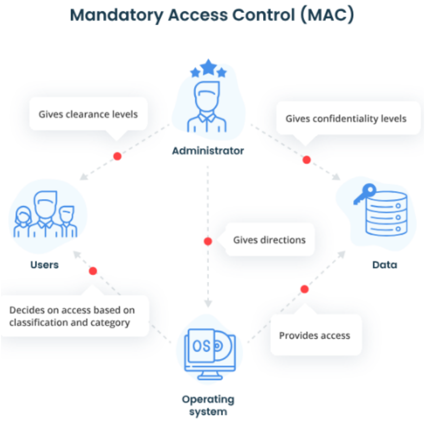

Mandatory Access Control (MAC)
Gives clearance levels
Administrator
Gives confidentiality levels
Users
Gives directions
Data
Decides on access based on classification and category
Provides access
OS
Operating system

Figure 1.6: Mandatory Access Control

1.8.3 Role-Based Access Control (RBAC)

Role-based access control (RBAC) is an alternative to traditional discretionary (DAC) and mandatory access control (MAC) policies that is attracting increasing attention, particularly for commercial applications.

RBAC is designed to enforce security policies that align with an organization's structure. In many business environments, a user's identity is primarily relevant for accountability purposes, while access control decisions are better based on the user's organizational responsibilities rather than their individual identity.

Traditional DAC, which emphasize user ownership of data, often do not fit well in enterprise settings. Similarly, MAC, where users have security clearances and objects have security classifications, can be too rigid. RBAC addresses this gap by combining explicit authorization mechanisms with organizational constraints, providing a balance between flexibility and structured access control.

Unlike DAC systems, which define access rights for individual users and groups, RBAC revolves around roles. A role in RBAC represents a job function within the organization, and access rights are assigned to these roles instead of directly to users. Users are then assigned roles, either statically or dynamically, based on their responsibilities, ensuring a more scalable and manageable approach to access control.

Generally speaking, the set of users, as well as their assignment to roles, can be pretty dynamic, while the set of roles and the permissions associated with them are more static.

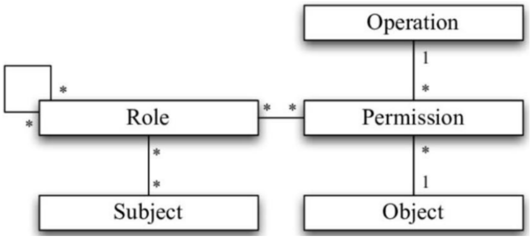


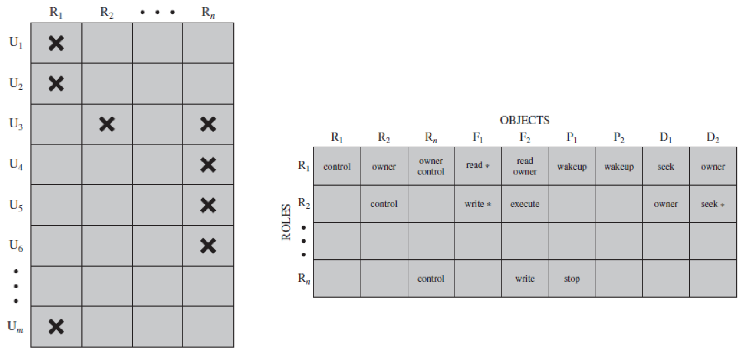
Figure 1.7: RBAC relationship between users, roles and permission

Figure 1.7 shows the relationship between users, roles, and permissions in an RBAC system. Users (subjects) are assigned roles, and roles are assigned permission to do operations on objects. A role can also inherit permissions from other roles, allowing for a hierarchical structure (which is what the recursive relationship in the figure represent).

Access Control Matrix Representation

As you can see in figure 1.8, RBAC can be easily represented by a couple of access control matrix. In the first matrix the rows represent the users, while the columns represent the roles.

The second one associates each role to the permission that it has on the objects.



	R_1	R_2	• • •	R_n
U_1	✕			
U_2	✕			
U_3		✕		✕
U_4				✕
U_5				✕
U_6				✕
• • •				
U_m	✕			

		OBJECTS								
		R_1	R_2	R_n	F_1	F_2	P_1	P_2	D_1	D_2
ROLES	R_1	control	owner	owner control	read *	read owner	wakeup	wakeup	seek	owner
	R_2		control		write *	execute			owner	seek *
	• • •									
	R_n			control		write	stop			

Figure 1.8: RBAC Access Control Matrix

Role Hierarchies

As previously mentioned, roles can inherit permissions from other roles. This is called role hierarchies, and it allows for a more flexible and scalable access control system. A subordinate job function may have a subset of the access rights of the superior job function, meaning that the user can perform all the operations that the superior role can perform.

An example of role hierarchy is shown in figure 1.9. In this example, the role ”Engineer” inherits the permissions of the role ”Engineer dept”, while the Directior has the highest level of permissions.

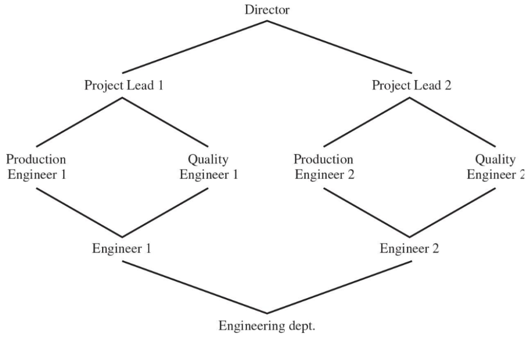


Figure 1.9: Role Hierarchy

In this context, the principles of separation of privilege and least privilege are very important. The first one uggests to break a single privilege among multiple independent subjects (component or user) so that more than one authorizations are required to perform an action, while the second one suggests that an entity should be given the minimum privileges and resources for the minimum period of time required to complete a task.

Constraints in RBAC

Constraints define conditions such as:

- Mutually exclusive roles (a user can hold only one role in a set).
- Cardinality limits (restricting the number of users per role).
- Prerequisite roles (requiring users to hold a specific role before obtaining another).

1.8.4 Attribute-Based Access Control (ABAC)

Attribute-Based Access Control (ABAC) is a policy-driven access control mechanism that grants access based on attributes of users, objects, and the environment.

An ABAC model can define authorizations that express conditions on properties of both the resource and the subject, allowing to define policies for any desired combination of attributes that are more flexible than the ones defined in DAC, MAC, and RBAC. This come at the cost of a more complexity, both in terms of implementation and performance, while also allowing to express very complex policies.

Subject attributes

A subject is an active entity (e.g., a user, an application, a process, or a device) that causes information to flow among objects or changes the system state. Each subject has associated attributes that define the identity and characteristics of the subject. Such attributes may include the subject's identifier, name, organization, job title, and so on. A subject's role can also be viewed as an attribute.

Object attributes

An object, also referred to as a resource , is a passive (in the context of the given request) information system-related entity (e.g., devices, files, records, tables, processes, programs, networks, domains) containing or receiving information.

As with subjects, objects have attributes that can be leveraged to make access control decisions. A Microsoft Word document, for example, may have attributes such as title, subject, date, and author. Object attributes can often be extracted from the metadata of the object.

Environment attributes

They describe the operational, technical, and even situational environment or context in which the information access occurs. For example, attributes, such as current date and time, the current virus/hacker activities, and the network's security level (e.g., Internet vs. intranet), are not associated with a particular subject nor a resource, but may nonetheless be relevant in applying an access control policy.