SECURITY AND RISK: MANAGEMENT AND CERTIFICATIONS

Simone **Soderi**

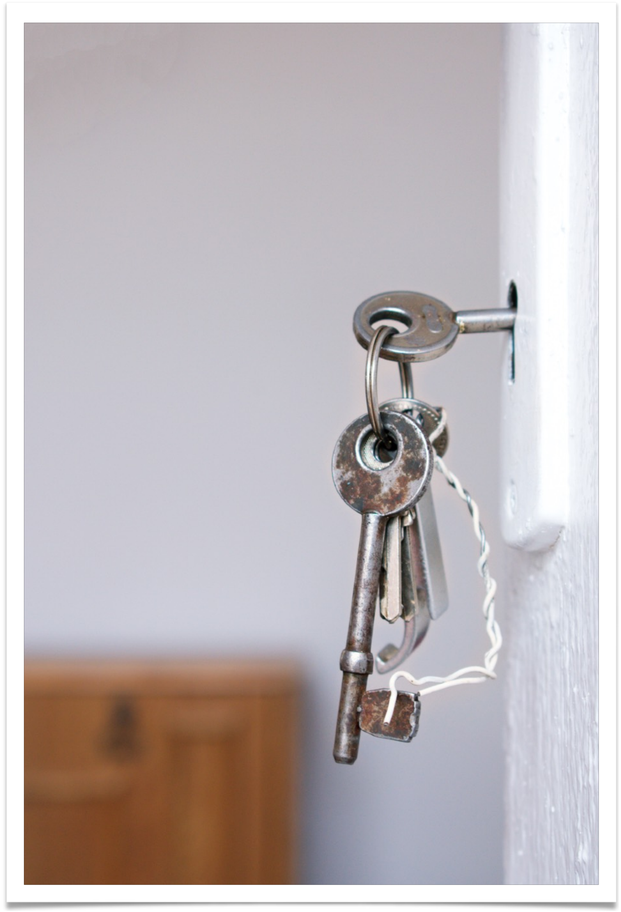
M3.2 - Cybersecurity Operations and Management

Contents (1/2)

Contents (2/2)

Contents

System Access Functions (1/2)



System access is the capability that restricts access to business applications, mobile devices, systems, and networks to authorized individuals for specific business purposes

System access comprises **three distinct functions**:

AUTHENTICATION

- **Verifying the identity** of a user, process, or device, often as a prerequisite to allowing access to resources in an information system
- This function is often referred to as user authentication

AUTHORIZATION

- The **granting of access** or other rights to a user, program, or process to access system resources
- Defines what an individual or program can do after successful authentication

ACCESS CONTROL

- The process of **granting or denying specific requests for accessing and using information** and related information processing services and for entering specific physical facilities
- Ensures that access to assets is **authorized and restricted** based on business and security requirements

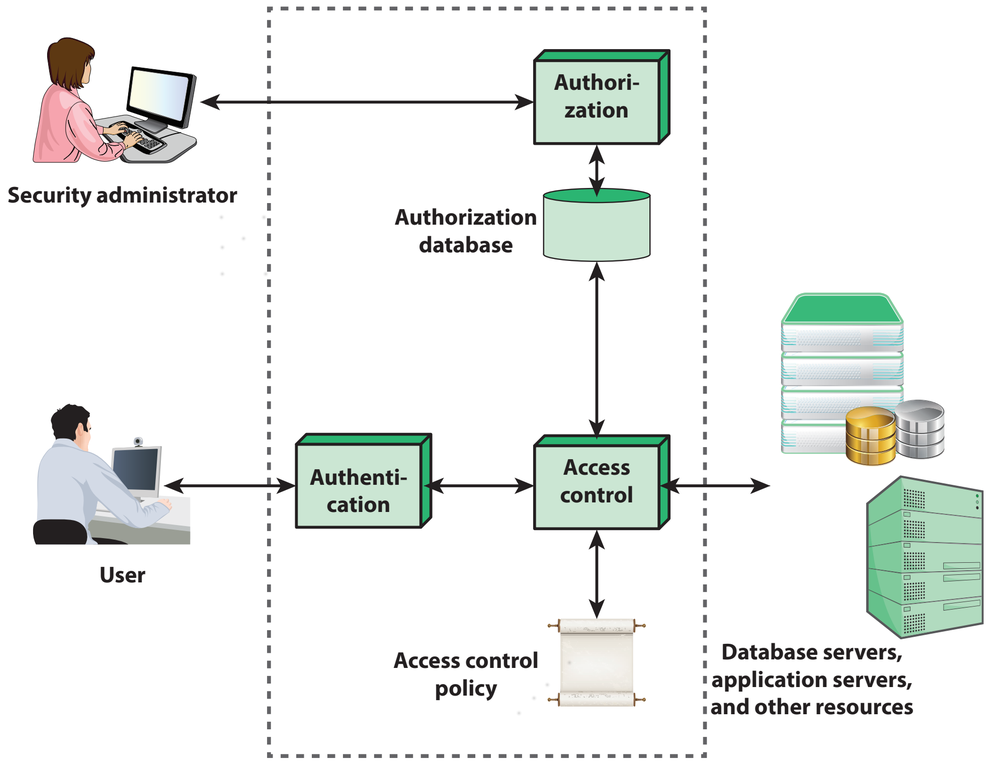
System Access Functions (2/2)

- **System access** is **concerned with denying access to unauthorized users and limiting the activities of legitimate users** to only those actions they are authorized to perform on system resources.

  The **access control function moderates attempted access to an object in the system** by a user or a program executing on behalf of that user.

- The authentication function **establishes the identity** of the user.

- The **authorization** function maintains an authorization **database that defines the access privileges** for each user. The access control function consults the authorization database and **uses an access control policy that specifies how a user's privileges** are to be mapped into allowable actions for specific data items or other resources.



Authorization

- A designated **security administrator** is responsible for creating and maintaining the **authorization database**.
- The administrator sets these **authorizations on the basis of the security policy** and the roles and responsibilities of individual employees.

The **process for authorizing** users should include the following:

- Associating access privileges with **uniquely defined individuals**
- Maintaining a **central record of access rights granted** to a user ID to access information systems and services.
- **Obtaining authorization from the owner** of the information system or service for the use of the information system or service.
- **Applying the principle of least privilege** to give each person the minimum access necessary to do his or her job.
- Assigning individual **access privileges for resources based on information security levels** and classification of information.
- **Specifying the networks and networked services to be accessed**, such as files and databases.
- Defining requirements for **expiration of privileged** access rights.
- Ensuring that **identifiers are not reused**. Deleting authorizations associated with a user ID when the individual changes roles or leaves the organization.

User Authentication

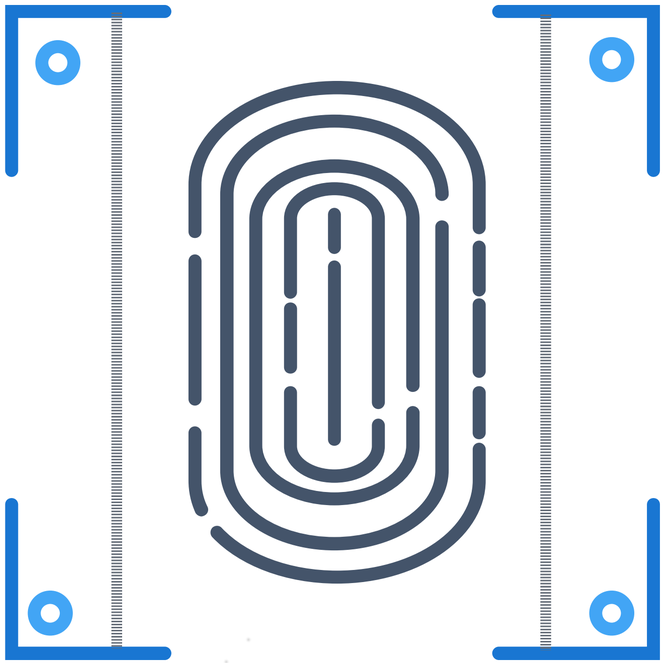
User authentication is one of the most complex and challenging security functions
There are a wide variety of methods of authentication

Three general **authentication factors** are:

- Password
- Hardware token
- Biometric

User authentication:

- Is the **basis for most types of access control** and for user accountability
- Is a fundamental building block and the **primary line of defense**
- Includes two functions:
  - **Identification** step
  - **Verification** step



Authentication Factor

USER AUTHENTICATION

A method of **authentication**, **based on either something the user has** (such as a smart card or dongle), **something the user knows** (such as a password, passphrase, or PIN), or **something the user is or does** (such as fingerprints or other forms of biometrics)

The specific items used during authentication, such as a password or hardware token, are referred to as **authenticators**.

Means of Authentication

USER AUTHENTICATION

There are **three general means** of authenticating a user's identity—called **authentication factors**—which are used alone or in combination:

- **Knowledge factor (something the individual knows)**: The user must demonstrate knowledge of secret information. Knowledge factors, routinely used in single-layer authentication processes, can come in the form of **passwords**, passphrases, personal identification numbers (PINs), or answers to secret questions.

- **Possession factor (something the individual possesses)**: The authorized user must **present a physical entity to connect to the client computer** or portal. This type of authenticator used to be referred to as a token, but that term is now deprecated. The term hardware token is a preferable alternative. Possession factors fall into two categories:

  - **Connected hardware tokens:** Items that connect to a computer logically (e.g., via wireless) or physically in order to authenticate identity. Items such as smart cards, wireless tags, and USB tokens are common connected tokens used to serve as possession factors.

  - **Disconnected hardware tokens:** Items that do not directly connect to the client computer but instead require input from the individual attempting to sign in. Typically, a **disconnected hardware token device uses a built-in screen to display authentication data** that are then utilized by the user to sign in when prompted.

- **Inherence factor (something the individual is or does)**: These are characteristics, called biometrics, that are unique or almost unique to the individual. These include static biometrics, such as fingerprint, retina, and face; and dynamic **biometrics**, such as voice, handwriting, and typing rhythm.

Authenticator

USER AUTHENTICATION

The means used to confirm the identity of a user, **process**, or **device** (for example, user password, hardware token). An authentication factor is based on the use of a particular type of authenticator

Example of Authentication Factor

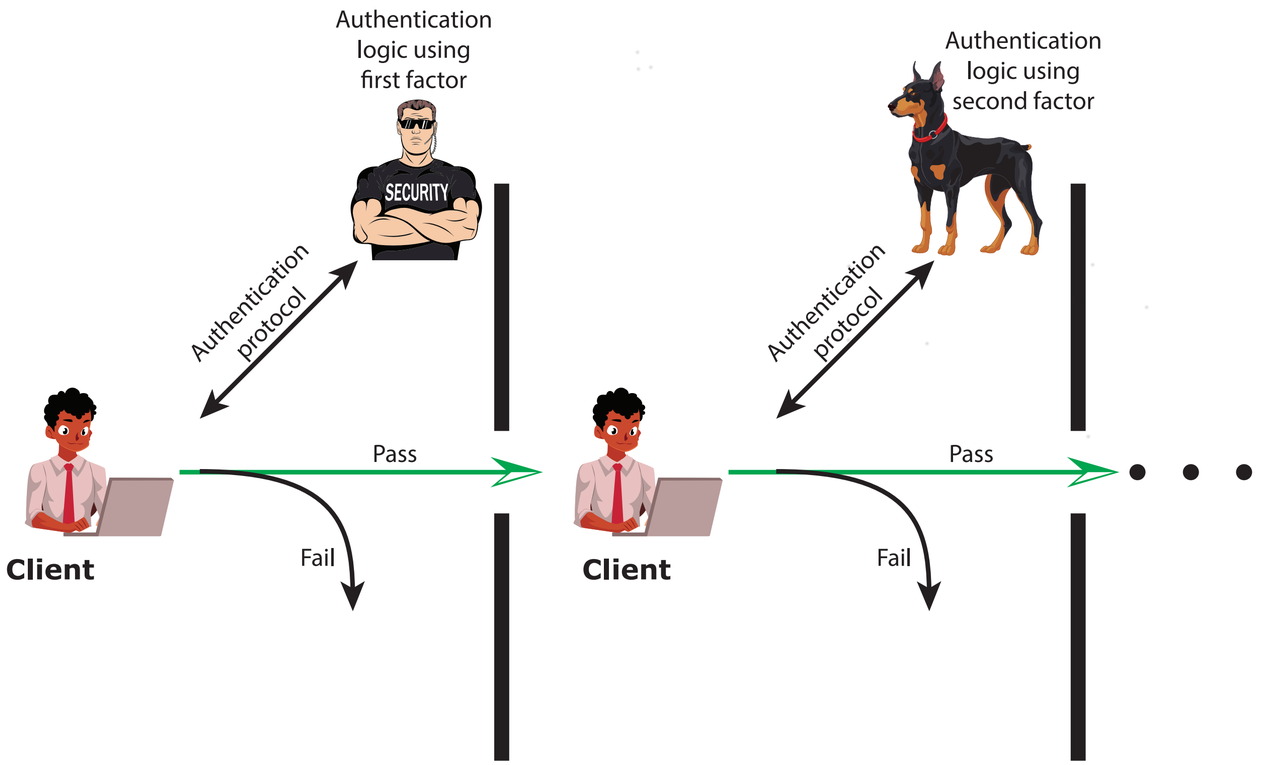
Factor	Exam ples	Properties
Knowledge	User ID Password PIN	Can be shared Many passwords easy to guess Can be forgotten
Possession	Smart Card Electronic Badge Electronic Key	Can be shared Can be duplicated (cloned) Can be lost or stolen
Inherence	Fingerprint Face Iris Voice print	Not possible to share False positives and false negatives possible Forging difficult

Each method has problems.

- An adversary may be able to guess or steal a password.
- With respect to biometric authenticators, including dealing with false positives and false negatives.

MultiFactor Authentication

USER AUTHENTICATION



Multifactor authentication **refers to the use of more than one of the authentication means** in the preceding list.

The **strength** of an authentication system is largely determined **by the number of factors incorporated by the system**. A system that requires two factors is generally stronger than a system requiring a single factor, assuming that the individual factors are reasonably strong.

Password-Based Authentication

USER AUTHENTICATION

- A **widely used line of defense against intruders** is a **password** system
- Virtually all multiuser systems, network-based servers, web-based ecommerce sites, and other similar **services require that a user provide** not only a name or **identifier (ID)** but also a **password**
- The **ID provides security** in the following ways:
  - The ID determines whether the user is **authorized** to gain access to a system
  - The ID determines the **privileges** accorded to the user
  - The ID used in what is referred to as **discretionary access control** (by listing the IDs of the other users, a user may grant permission to them to read files owned by that user)

Vulnerability of Password (1/4)

USER AUTHENTICATION

Typically, a server that uses **password-based authentication maintains a password file indexed by user ID**.

When logging on or making an access request, a user presents both his or her ID and password. The server **looks up the password for that ID in the password file** to determine if there is a **match**.

One security technique that is **typically used is to store not the user's password but a one-way hash function of the password**, as described subsequently.

Vulnerability of Password (2/4)

Attack strategies and countermeasures:

Offline dictionary attack

- Although strong access controls are used to protect a system's password file, determined hackers frequently bypass such controls and gain access to password files. An attacker who obtains a system password file **compares the password hashes against hashes of commonly used passwords and if a match is found**, the attacker gains access by using that ID/password combination
- Countermeasures include controls to prevent unauthorized access to the password file, **intrusion detection measures** to identify a compromise, and **rapid reissuance of passwords** in the event that the password file is compromised

Specific account attack

- In this type of attack, **an attacker targets a specific account and submits password guesses** until the correct password is discovered
- The standard countermeasure is an **account lockout mechanism**, which locks out access to the account after a number of failed login attempts standard countermeasure

Popular password attack

- A variation of the preceding attack is to use a **popular password and try it against a wide range of user IDs**. A user's tendency is to choose a password that is easily remembered; this unfortunately makes the password easy to guess
- Countermeasures include policies to **inhibit the selection by users of common passwords** and scanning the IP addresses of authentication requests and client cookies for submission patterns

Vulnerability of Password (3/4)

USER AUTHENTICATION

Attack strategies and countermeasures:

Password guessing against a single user

- An attacker may attempt to **gain knowledge about an account holder and system password policies** and uses that knowledge to guess the user's password
- Countermeasures include training in and enforcement of password policies that **make passwords difficult to guess**. Such policies address the secrecy, minimum length of the password, character set, prohibition against using well-known user identifiers, and length of time before the password must be changed

Workstation hijacking

- In this type of attack, an attacker waits until a **logged-in workstation is physically left unattended**
- The standard countermeasure is automatically **logging out the workstation after a period of inactivity.** Intrusion detection schemes are used to detect changes in user behavior

Electronic monitoring

- If a **password is communicated across a network to log on to a remote system**, it is vulnerable to eavesdropping. Simple encryption does not fix this problem because the encrypted password is, in effect, the password and can be observed and reused by an adversary

Vulnerability of Password (4/4)

Attack strategies and countermeasures:

Exploiting multiple password use

- Attacks become much more effective or damaging **if different network devices share the same or a similar password** for a given user
- Countermeasures include a policy that **forbids using the same or similar password** on particular network devices

Exploiting user mistakes

- **If the system assigns a password, then the user is more likely to write it down because it is difficult to remember.** This situation creates the potential for an adversary to read the written password. A user may intentionally share a password to enable a colleague to share files, for example. Also, attackers are frequently successful in obtaining passwords by **using social engineering tactics that trick the user or an account manager into revealing a password.** Many **computer systems are shipped with preconfigured passwords** for system administrators. Unless these preconfigured passwords are changed, they are easily guessed
- Countermeasures include user **training**, intrusion detection, and simpler passwords combined with another authentication mechanism

Remarks on Password Vulnerability

Despite the many security vulnerabilities of passwords, **they remain the most commonly used user authentication techniqu**e, and this is unlikely to change in the foreseeable future.

Among the **reasons for the persistent popularity of passwords are** the following:

- Techniques that utilize **client-side hardware**, such as fingerprint scanners and smart card readers, **require the implementation of the appropriate user authentication software** to exploit this hardware on both the client and server systems. Until there is widespread acceptance on one side, there is reluctance to implement on the other side, and the result is a who-goes-first stalemate.

- **Physical tokens**, such as smart cards, are expensive and/or **inconvenient to carry around**, especially if multiple tokens are needed.

- Schemes that rely on a **single sign-on (SSO)** to multiple services, using one of the non- password techniques described in this chapter, create a **single point of security risk**.

- **Automated password managers** that relieve users of the charge of knowing and entering passwords have poor support for roaming and synchronization across multiple client platforms, and their usability had not been adequately researched.

Hashed Password (1/2)

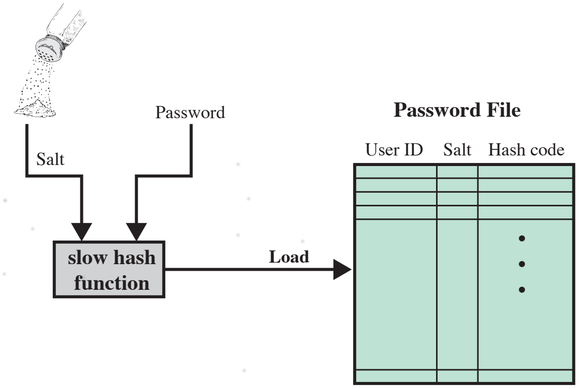
UNIX PASSWORD SCHEME

A **widely used password security technique** is the use of **hashed passwords and a salt value**.

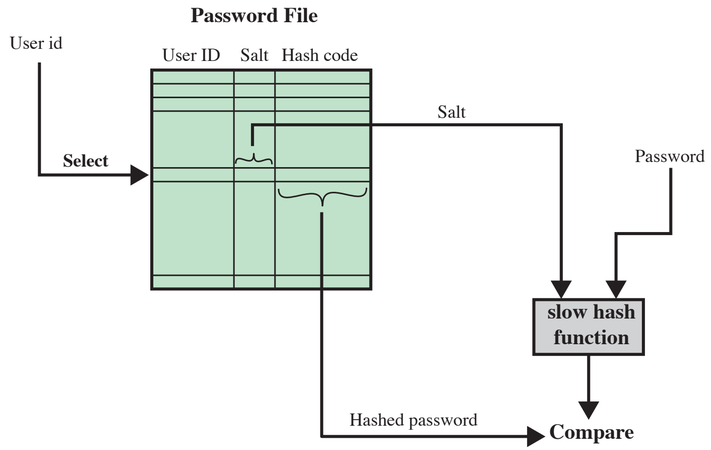
This scheme is found on virtually all **UNIX variants** as well as on a number of other operating systems.

LOADING NEW PASSOWRD

- The system combines the password with a fixed-length salt value (time or pseudorandom number).
- The password and salt serve as inputs to a **hashing** algorithm to produce a fixed-length hash code.
- The hashed password is then stored, together with a plaintext copy of the salt, in the password file for the corresponding user ID.



(a) Loading a new password



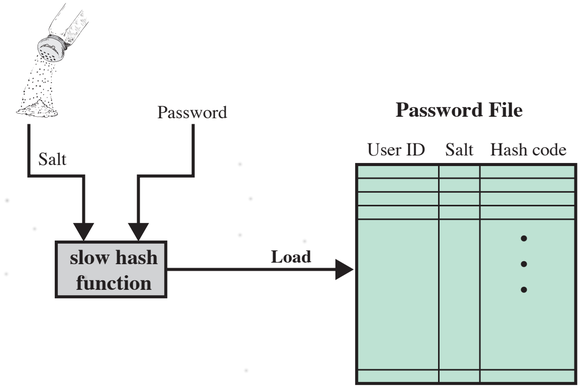
(b) Verifying a password

Hashed Password (2/2)

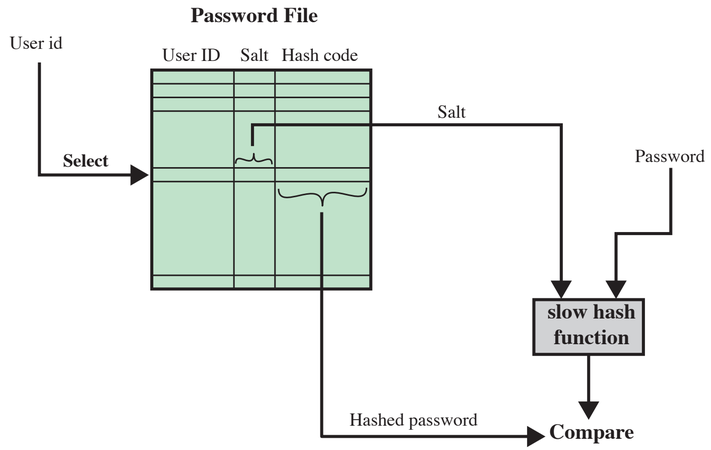
UNIX PASSWORD SCHEME

A **widely used password security technique** is the use of **hashed passwords and a salt value**.

This scheme is found on virtually all **UNIX variants** as well as on a number of other operating systems.



(a) Loading a new password



(b) Verifying a password

VERIFYING A PASSOWRD

- When a user attempts to log on to a UNIX system, the user provides an ID and a password.
- The operating system uses the ID to index into the password file and retrieve the plaintext salt and the hashed password.
- The salt and user-supplied password are used as input to the hash algorithm. If the result matches the stored value, the password is accepted.

Why do we need Salt ?

HASHED PASSWORD

The salt serves **three purposes**:

- It **prevents duplicate passwords** from being visible in the password file. Even if two users choose the same password, those passwords are assigned different salt values. Hence, the hashed passwords of the two users differ.
- It greatly **increases the difficulty of offline dictionary attacks**. For a salt of length b bits, the number of possible passwords is increased by a factor of 2^b, increasing the difficulty of guessing a password in a dictionary attack.
- It becomes nearly impossible to find out whether a person with passwords on two or more systems **used the same password on all of them**.

Mitigating Password Attacks with Salt

To mitigate the damage that a hash table or a dictionary attack could do, we salt the passwords. According to ***OWASP Guidelines***, a salt is a value generated by a cryptographically secure function that is added to the input of hash functions to create unique hashes for every input, regardless of the input not being unique. A **salt makes a hash function look non-deterministic**, which is good as we don't want to reveal duplicate passwords through our hashing.

Password Cracking

GENERAL DEFINITION

Password cracking is the **process of recovering secret passwords stored** in a computer system or transmitted over a network.

Password Cracking of User-Chosen Passwords

Traditional approaches

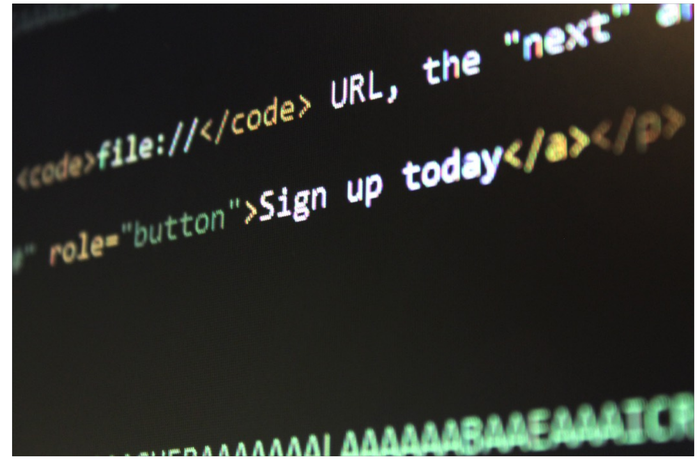
- The traditional approach to password cracking is to **develop a large dictionary of possible passwords** and to try each of them against the password file
- **Each password must be hashed using each available salt** value and then compared to stored hash values
- If no match is found, **the cracking program tries variations on all the words in its dictionary** of likely passwords
- Such variations include **backward spelling of words**, additional numbers or special characters, and a sequence of identical characters

An alternative is to trade off space for time by precomputing potential hash values

- In this approach, the **attacker generates a large dictionary of possible passwords**
- For each password, the attacker **generates the hash values** associated with each possible salt value
- The result is a mammoth table of hash values known as a rainbow table
- This **approach is contrasted by using a sufficiently large salt** value and a sufficiently large hash length
- Both FreeBSD and OpenBSD, which are open source versions of UNIX, use this approach, and so should be secure from this attack for the foreseeable future

Password File Access Control (1/2)

- One way to contrast a password attack is to **deny the attacker access to the password file**
- If the hashed password portion of the **file is accessible only by a privileged user**, then the **attacker cannot read it** without already knowing the password of a privileged user
- Often, the **hashed passwords are kept in a separate file from the user IDs**, referred to as a shadow password file
- Special attention is paid to making the shadow password file protected from unauthorized access



Password File Access Control (2/2)

Although password file protection is certainly useful, there **remain vulnerabilities, including** the following:

- An accident of protection **might render the password file readable**, thus compromising all the accounts
- Some of the users **may have accounts on other machines** in other protection domains, and they **may use the same password of all of them**
- A **lack of or weakness in physical security** may provide opportunities for a hacker
- Instead of capturing the system password file, another approach to **collecting user IDs and passwords is through sniffing network traffic**

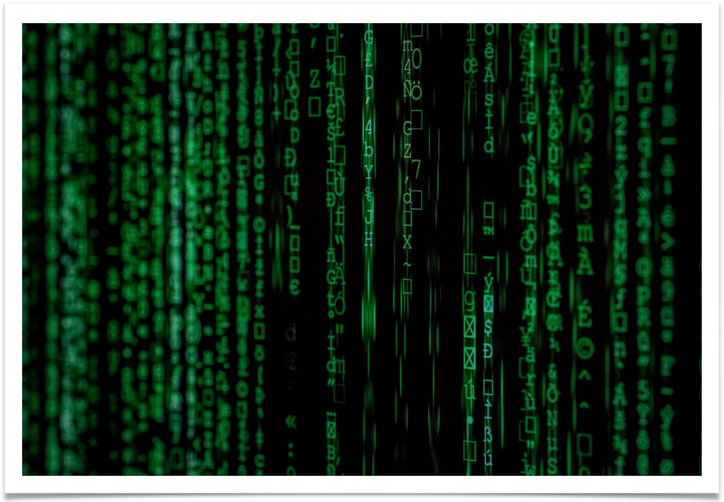
Thus, a **password protection policy must complement access control** measures with techniques **to force users to select passwords that are difficult to guess**

Password Selection

When not constrained, **many users choose a password that is too short or too easy to guess**

At the other extreme, if users are assigned passwords consisting of eight randomly selected printable characters, password cracking is effectively impossible.
But it is almost as impossible for most users to remember such passwords.

The goal is to eliminate guessable passwords while allowing the user to select a password that is memorable



Possession-Based Authentication

USER AUTHENTICATION

Objects that a **user possesses for the purpose of user authentication** are sometimes called **hardware tokens** (to distinguish from a number of software types of tokens).

Card Type	Defining Feature	Exam ple
Embossed	Raised characters only, on front	Traditional credit card
Magnetic stripe	Magnetic bar on back, characters on front	Bank card
Memory	Electronic memory inside	Prepaid phone card
Smart Contact Contactless	Electronic memory and processor inside Electrical contacts exposed on surface Radio antenna embedded inside	Biometric ID card

Memory Cards

USER AUTHENTICATION

- Memory cards **store but do not process data**
  - The most common is a **bank card** with a magnetic stripe on the back (a magnetic stripe stores only a simple security code, which is read by an inexpensive card reader)
  - There are also **memory cards that include an internal electronic memory**

- Memory cards are used **alone for physical access**, such as a hotel room

- For authentication, a **user provides both the memory card and some form of password or PIN** (a typical application is an ATM)
  - The memory card, when combined with a PIN or password, provides significantly greater security than a password alone

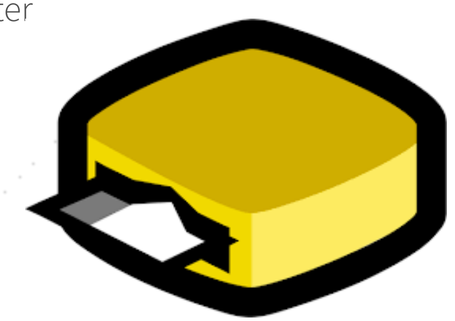
- Among the **potential drawbacks** are:
  - Special reader requirement
  - Hardware token loss
  - User dissatisfaction

Smart Cards (1/2)

USER AUTHENTICATION

Smart tokens are **categorized along four dimensions that are not mutually exclusive**

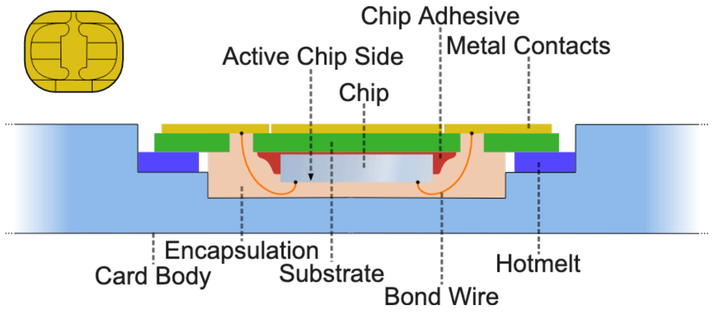
- **Physical characteristics**
  - A smart token that looks like a bank card is called a smart card
  - Other smart tokens look like calculators, keys, or other small portable objects
- **User interface**
  - Manual interfaces include a keypad and display for human/token interaction
- **Electronic interface**
  - A smart card or other token requires an electronic interface to communicate with a compatible reader/writer
  - A card may have a contact or contactless interface (or both)
- **Authentication protocol**
  - A smart token provides a means for user authentication
  - The authentication protocols used with smart tokens are classified into three categories:
    - Static
    - Dynamic password generator
    - Challenge-response

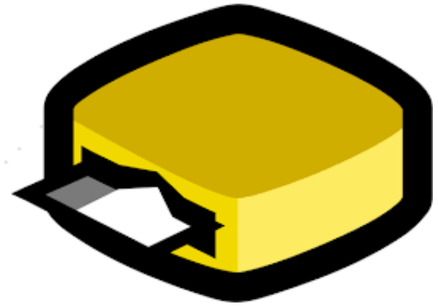


Smart Cards (2/2)

USER AUTHENTICATION

- For user authentication, the most important category of smart token is the **smart card, which has the appearance of a credit card**, has an electronic interface, and may use any of the type of protocols just described
- A smart card contains within it an entire **microprocessor**, including processor, memory, and I/O ports
- Some versions incorporate a special **coprocessing circuit for cryptographic operation** to speed the task of encoding and decoding messages or generating digital signatures to validate the information transferred
- In some cards, the **I/O ports are directly accessible by a compatible reader** by means of exposed electrical contacts
- Other cards **rely instead on an embedded antenna for wireless communication** with the reader





Electronic Identity Cards

USER AUTHENTICATION

An application of increasing importance is the use of smart **cards as national ID cards for citizens**

A national electronic identity (**eID**) card serves the same purposes as other national ID cards (such as driver's licenses)

In addition, an **eID card provides stronger proof of identity** and is used in a wider variety of applications

In effect, an **eID card is a smart card that was verified by the national government** as being valid and authentic

One-Time Password (OTP) Device

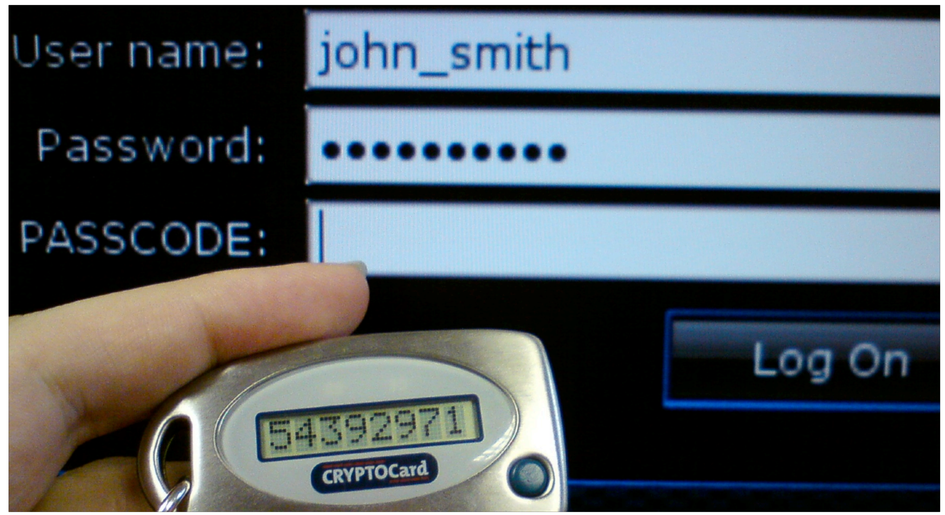
USER AUTHENTICATION

An increasingly widespread device used for authentication is a keychain type of device that **generates one-time passwords**

The use of a **one-time password (OTP)** improves security, **preventing the risk of guessing and reusing a password**, especially if used together with other authentication mechanisms

This device **has an embedded secret that is used as the seed** for generation of one-time passwords and may not require activation through a second factor

Authentication is accomplished by **providing an acceptable one-time password and thereby proving possession and control of the device**



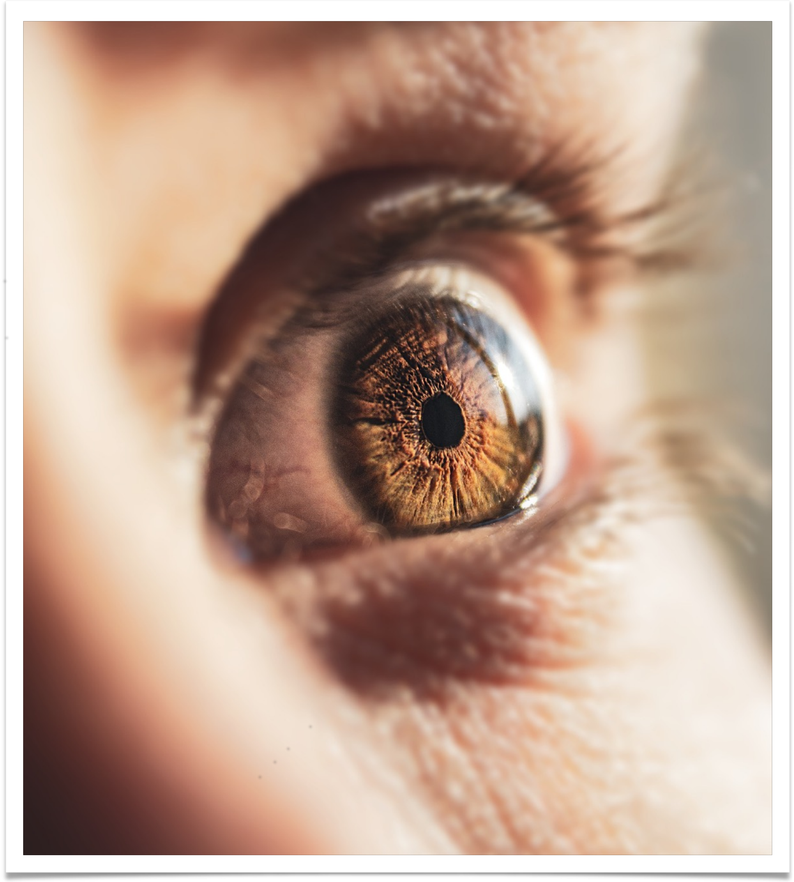
Biometric Authentication

USER AUTHENTICATION

A biometric authentication system attempts to authenticate an **individual based on his or her unique physical characteristics**

These include both **static** characteristics (fingerprints, hand geometry, facial characteristics, retinal and iris patterns) and **dynamic** characteristics (voiceprint, signature)

Compared to passwords and hardware tokens, **biometric authentication is both technically complex and expensive**



Criteria for Biometric Characteristics

USER AUTHENTICATION

In designing a biometric identification system, two sets of criteria must be considered:

- The **nature** of the biometric feature
- **Requirements** for a biometric system

With respect to **biometric features**, such features must have the following properties:

- Universality
- Distinction
- Permanence
- Collectability

Criteria for Biometric Identification

USER AUTHENTICATION

A biometric identification system **should also meet the following system criteria**:

Performance

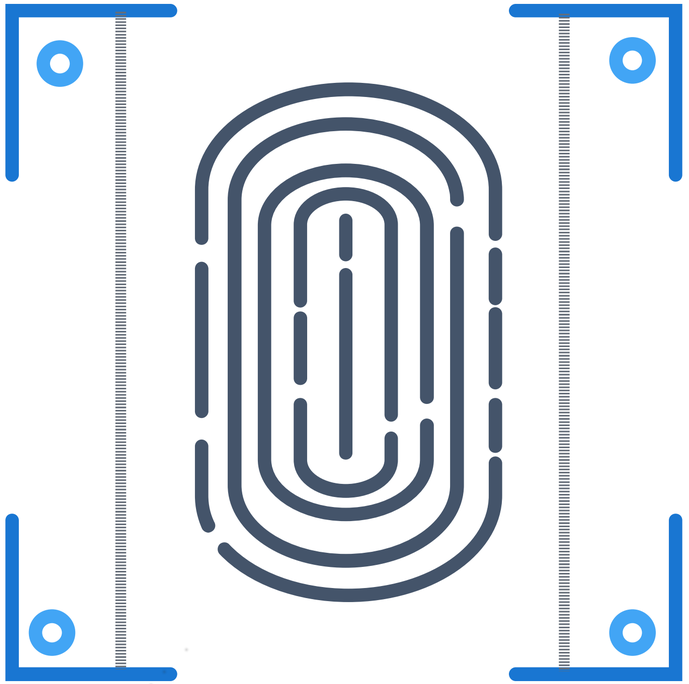
The system must meet a r**equired level of accuracy**, perform properly in the required range of environments, and be cost-effective

Circumvention

The **difficulty of circumventing** the system must meet a required threshold. This is particularly important in an unattended environment, where it is easier to use such countermeasures and a fingerprint prosthetic or a photograph of a face

Acceptability

The system must be generally acceptable to users



Access Control

TERMS DEFINITION

Useful **terms** for access control:

Access

Ability and **means** to **communicate** with or otherwise **interact** with a system, to **use** system resources to handle information, to **gain knowledge** of the information the system contains, or to control system components and functions.

Access Control

The **process of granting or denying specific requests** for obtaining and using information and related information processing services to enter specific physical facilities.

Access Control Mechanism

Security safeguards (that is, hardware and software features, physical controls, operating procedures, management procedures, and various combinations of these) **designed to detect and deny unauthorized access** and permit authorized access to an information system.

Access Control Service

A **security service that protects against** a system entity using a system resource in a way not authorized by the system's security policy.

We define access control

- Lampson in 1992 [*] defined **access control** as consisting of two functions

ACCESS CONTROL = AUTHENTICATION + AUTHORISATION

(*) B. Lampson, M. Abadi, M. Burrows, and E. Wobber, "Authentication in distributed systems: Theory and practice," *ACM Transactions on Computer Systems*, vol. 10, no. 4, pp. 265-310, November 1992.

What is needed for access control

Access control is ***'the process of granting or denying specific requests …'***.

This process needs the following **inputs**

- **Who** issued the request?
- **What** is required?
- **What rules** apply when deciding on the application?

The '**who**' is inaccurate for two reasons:

1. the **source of** a request may be a particular machine, a machine in a particular configuration or a particular programme.
2. On a technical level, requests in a machine are issued by a **process**, not by a person.

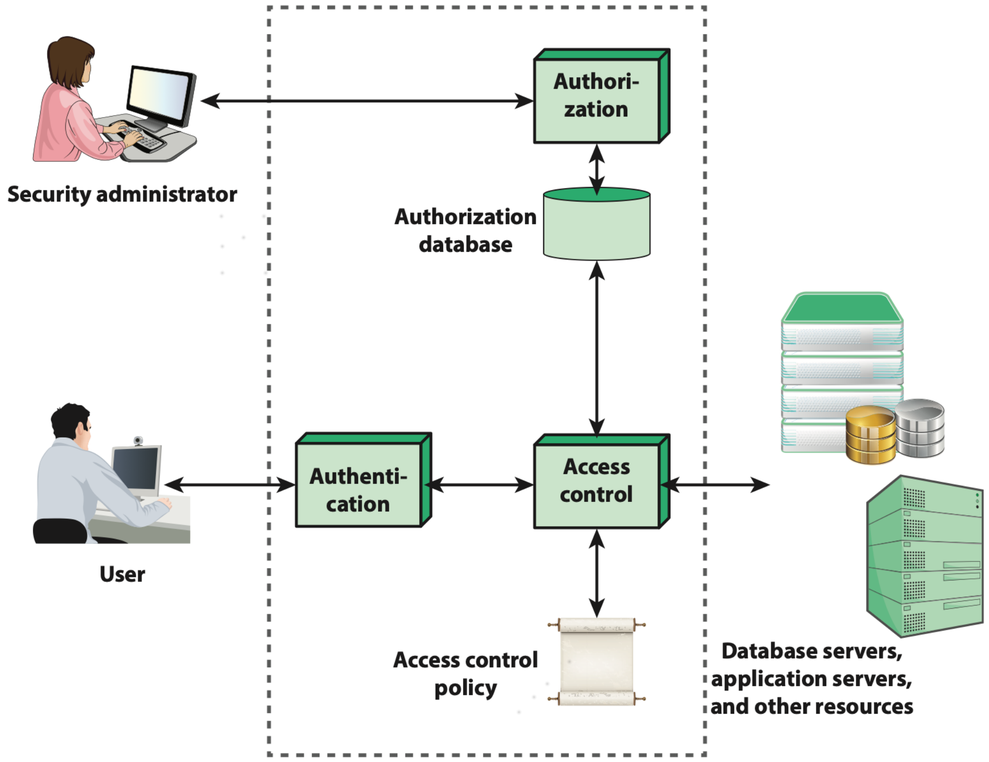
The question then becomes: "**For whom or for what does the process speak when it makes the request?**".

System access functions for the user

- **System access deals with denying access to unauthorised users and limiting the activities of legitimate users** to only those actions they are authorised to perform on system resources. The access control function **moderates** attempts to access a system object by a user or a programme running on behalf of that user.

- The **authentication** function **establishes the identity of the user**.

- The **authorisation function** maintains an **authorisation database that defines access privileges** for each user. The access control function consults the authorisation database and uses **an access control policy that specifies how a user's privileges are** to be mapped into permitted actions for specific data items or other resources.



Access Control Elements

The basic elements of **access control** are three:

- **SUBJECT**
- **OBJECT**
- **ACCESS RIGHTS**

Elements of Access Control (1/3)

SUBJECT:

- A subject is an **entity capable of accessing objects**
- **A subject is typically considered accountable for the actions** he or she has initiated, and an audit trail can be used to record the association of a subject with security-relevant actions performed on an object by the subject
- Basic access control systems typically define **three classes of subject**, with different access rights for each class:
  - **Owner:**
    - This can be the creator of a resource
  - **Group:**
    - Group of users can also be granted access rights
  - **World:**
    - The least amount of access is granted to users who are able to access the system but are not included in the categories owner and group for this resource.

Elements of Access Control (2/3)

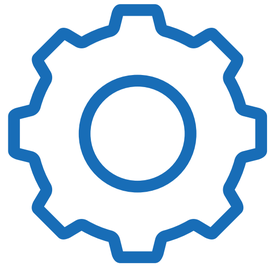
OBJECT:

- An object is a **resource** to which access is controlled
- In general, **an object is an entity used to contain and/or receive information**.
  Examples include records, blocks, pages, segments, files, portions of files, directories, directory trees, mailboxes, messages, and programs
- The number and **types of objects** to be protected by an access control system **depends on the environment** in which access control operates and the desired trade-off between security on the one hand and complexity, processing charge, and ease of use on the other hand

Elements of Access Control (3/3)

ACCESS RIGHT:

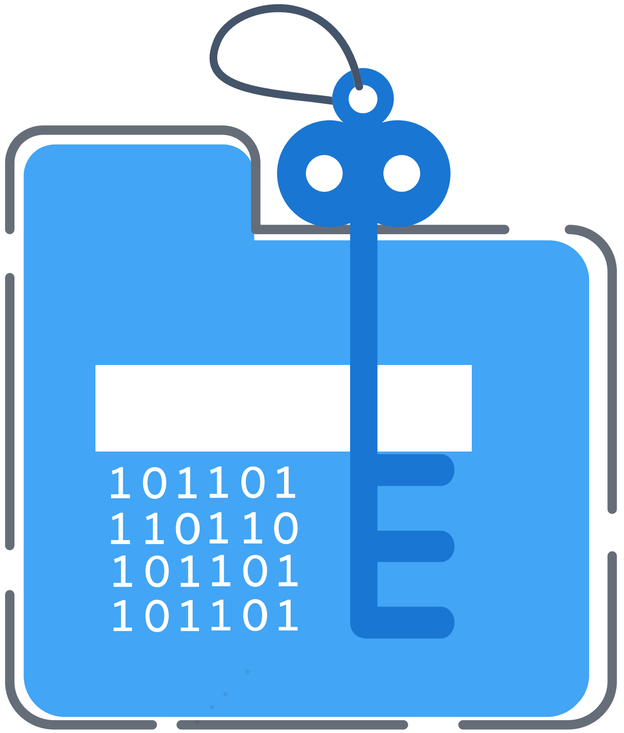
- An access right **describes the way in which a subject may access an object**
- Access rights **include the following**:
  - Read
  - Write
  - Execute
  - Delete
  - Create
  - Search



Access Control Policies (1/5)

An **access control policy dictates what types of access are permitted**, under what circumstances, and by whom. Access control policies are generally grouped into the following **categories**:

- Discretionary access control (**DAC**)
- Mandatory access control (**MAC**)
- Role-based access control (**RBAC**)
- Attribute-based access control (**ABAC**)

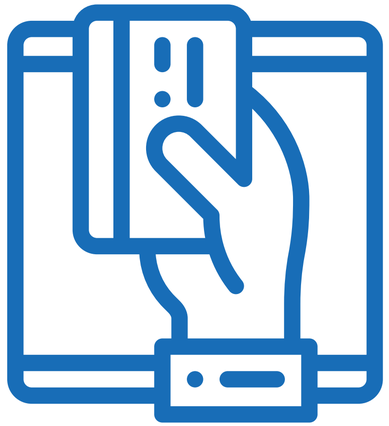


Access Control Policies (2/5)

DISCRETIONARY ACCESS CONTROL (DAC)

- Access control **based on the identity of the requestor and on access rules** (authorizations) stating what requestors are (or are not) allowed to do

- The **controls are discretionary in the sense that a subject with a certain access permission is capable of passing** that permission (perhaps indirectly) on to any other subject



Access Control Policies (3/5)

MANDATORY ACCESS CONTROL (MAC)

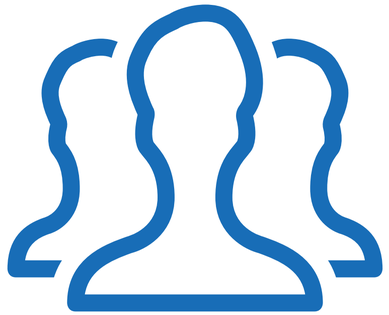
- Access control **based on comparing security labels** (which indicate how sensitive or critical system resources are) **with security clearances** (which indicate system entities are eligible to access certain resources)

- This policy is termed **mandatory** because an entity that has authorisation to access a resource **cannot, just by its own wishes, enable another** entity to access that resource

Access Control Policies (4/5)

ROLE-BASED ACCESS CONTROL (RBAC)

- Access control **based on user roles** (that is, a collection of access authorizations a user receives based on an explicit or implicit assumption of a given role)
- Role **permissions can be inherited through a role hierarchy** and typically reflect the permissions needed to perform defined functions within an organization
- A given role can apply to a **single** individual or to **several** individuals

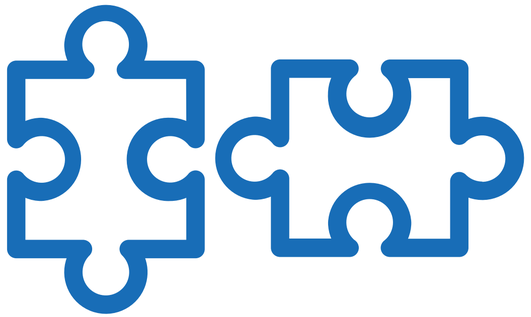


Access Control Policies (5/5)

ATTRIBUTE-BASED ACCESS CONTROL (ABAC)

- Access control **based on attributes associated with and about subjects, objects**, targets, initiators, resources, or the environment

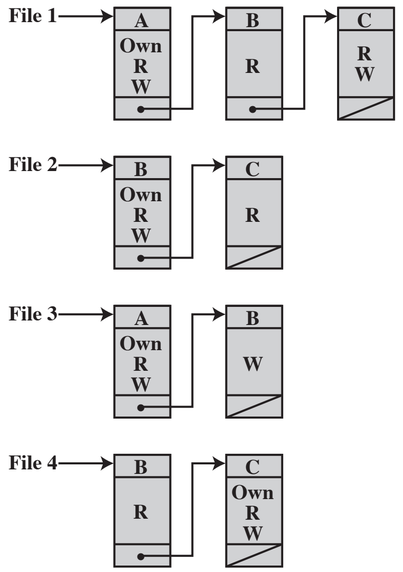
- An access control rule set defines the **combination of attributes under which an access takes place**



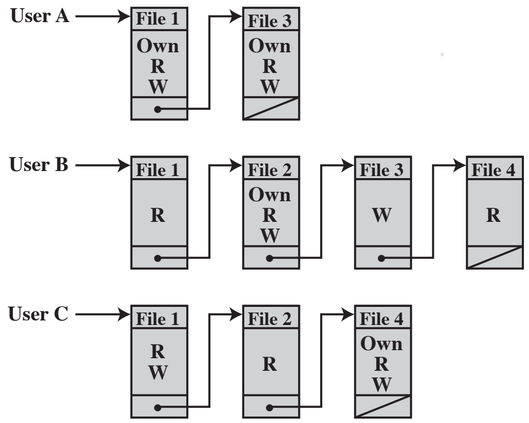
Example of Access Control Structures

		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



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



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

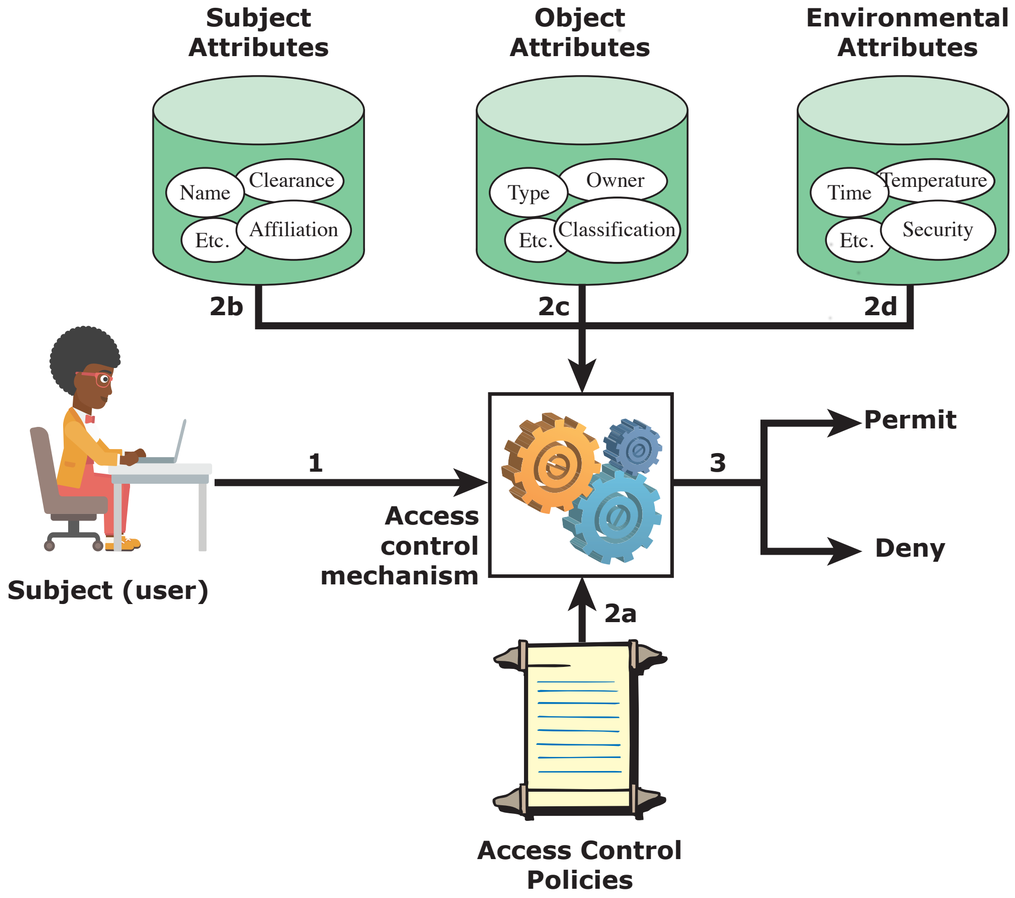
As shown here, user A owns files 1 and 3 and has read and write access rights to those files. User B has read access rights to file 1, and so on.

In practice, an **access matrix** is usually sparse and is implemented by decomposition in one of two ways. The matrix can be decomposed by

- Decomposition by columns, yielding **access control lists (ACLs)**. Lists users and their permitted access rights.

- Decomposition by rows yields **capability tickets**. A capability ticket specifies authorized objects and operations for a particular user

Example of Attribute-Based Access Control



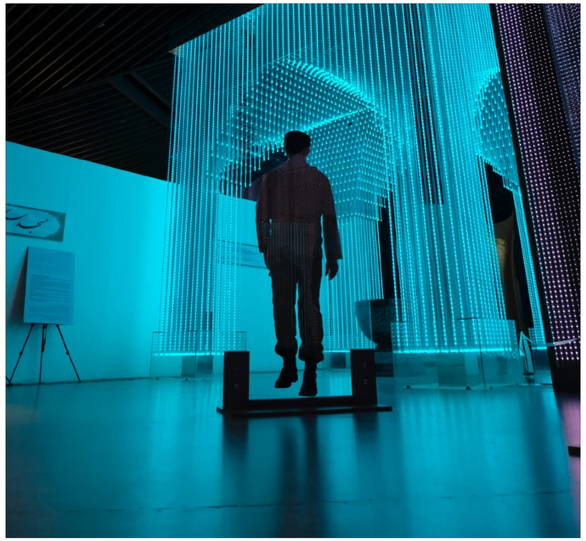
An **access by a subject to an object** proceeds according to the following steps:

- A subject requests access to an object. This request is routed to an access control mechanism.
- The access control mechanism is governed by a set of **rules** (2a) that are defined by a preconfigured access control policy. Based on these rules, the access control mechanism **assesses the attributes of the subject** (2b), **object** (2c), and **current environmental conditions** (2d) to determine **authorization**.
- The **access control mechanism grants the subject access** to the object if access is authorized and denies access if it is not authorized.

Customer Access

Customer access refers to **access to business applications by individuals**:

- Many of the security controls implemented to provide security for employee access to business applications apply to customers as well
- Management needs to determine that in each individual case these controls are applied, all aspects of **customer access to the organization's business applications meet security requirements**
- An individual or a group within the organization should be given responsibility for authorizing each customer access arrangement
- In addition, each **customer needs to be uniquely identified and approved** by the owner of the application, who designates the access privileges for this customer
- If it is appropriate, **provide the customer with awareness training and education** relevant to the threats associated with customer access and the possible consequences in the event access is compromised
- Consult various stakeholders within the organization to determine the **appropriate balance** between providing for **customer satisfaction** and convenience on the one hand and **meeting security requirements on the other**



Customer Connections

- Customer access remotely over the Internet or on a private network, **should be subject to the same types of technical controls** discussed in earlier.
- **Authorize each customer** with the process which includes **defining access privileges** for applications, information, and other resources within the organization.
- Then, as with any user access to an application or other resources, determine the authentication assurance level and **select an appropriate authentication procedure**.

It is absolutely clear to the customer that they cannot connect their own network to the organization's computer network.

SECURITY AND RISK: MANAGEMENT AND CERTIFICATIONS

Simone **Soderi**

simone.soderi@unipd.it

M3.2 - Cybersecurity Operations and Management

Thanks for your attention!