

# Systems and Information Security SEGSI

Topic 1  
Access Control

# Access Control

- ▶ Access control is the key to security
- ▶ Alternate definitions:
  - ▶ The ability to allow access only to authorized users, programs or system processes and resources
  - ▶ The granting or denial, in accordance with a security model, of access permissions to resources
  - ▶ Procedures performed by hardware, software or administrators to monitor access, validate and grant/deny access requests, based on pre-defined rules

# Security Principles

## ▶ Availability

- ▶ Systems are always usable and productive

## ▶ Integrity

- ▶ Information is not corrupted or modified in unauthorized ways

## ▶ Confidentiality

- ▶ Information is protected from unauthorized views or uses

# Guarantee of security principles

- ▶ Through identification
  - ▶ Process of obtaining an identity
- ▶ Through authentication
  - ▶ Process by which an identity is validated or verified
- ▶ Through authorization
  - ▶ Process of assigning access modes according to identity
- ▶ Through accounting
  - ▶ Collection of access event information

# Implementation

- ▶ Hardware
- ▶ Software
  - ▶ Applications
  - ▶ Standard protocols (Kerberos, IPSec)
- ▶ Physical access
- ▶ Logical implementation (written policies)

# What to protect

- ▶ **Data**

- ▶ Prevent unauthorized modification, copying, or viewing

- ▶ **Systems**

- ▶ Use, unauthorized settings or service unavailability

- ▶ Almost all current operating systems still tend to assume that there is a very secure physical infrastructure (computing, peripherals, network, etc.)

# Pro-active Control

- ▶ Background Checks
- ▶ Separation of tasks/responsibilities
- ▶ Distribution of knowledge
- ▶ Access/use policies and rules
- ▶ Classification of data
- ▶ Compulsory identification
- ▶ Handling procedures
- ▶ Modification of control procedures

# Privacy Issues

- ▶ Access versus privacy control (GDPR)
  - ▶ Typically there is expectation of privacy by users
  - ▶ Policies can not only include privacy requirements
  - ▶ Monitoring activity in access to systems and services collides with privacy
  - ▶ In each service the banners should detail the expectations of privacy and the level of monitoring that will be carried out...



# Physical access control

- ▶ Guards at the door
- ▶ Locks and traps
- ▶ Secure CCTV, IR sensors, alarms
- ▶ Identification devices
- ▶ Biometric Identification
- ▶ Perimeter Barriers
- ▶ Guard dogs
- ▶ Wide range facial recognition
- ▶ Etc.

# Authentication

- ▶ Types of Authentication
  - ▶ Information only you know (KNOW)
    - ▶ Password, PIN, name of dog, some code, etc.
  - ▶ Object only you own (HAVE)
    - ▶ ATM Card, smart card, token, key, ID Card, national Card, passport, etc.
  - ▶ Characteristics of own (BE)
    - ▶ Fingerprint recognition, voice recognition, facial recognition, iris recognition, retinal recognition, body odor, DNA, etc.

# Multi-factor Authentication

## ▶ 2 Factor Authentication

- ▶ To increase security use 2 types of authentication:
  - ▶ ATM Card + PIN
  - ▶ Credit Card + signature
  - ▶ PIN + digital print

## ▶ 3 Factor Authentication

- ▶ For even greater security:
  - ▶ Username + password + digital print
  - ▶ Username + passcode + token SecurID
  - ▶ Credit Card + signature + extra control

# Passwords

- ▶ Problems with passwords
  - ▶ Potentially "unsafe"
    - ▶ Appeal to names of relatives, pets, phone numbers, birthdays, etc.
  - ▶ Decipherable
    - ▶ Computer programs may attempt to decode passwords on almost all operating systems
  - ▶ Inconvenient
    - ▶ They can be difficult and many to memorize
  - ▶ May be repudiated
    - ▶ When a transaction is only authenticated with a password, there is no real proof of identity of the individual who performed the transaction

# Passwords

- ▶ Passwords attacks
  - ▶ Brute force (exhaustive)
    - ▶ Easy to make but take too long
  - ▶ Dictionary of words
    - ▶ Based on word repositories
  - ▶ Dictionary and heuristics
    - ▶ Example: John the Ripper, etc.
  - ▶ Fake login program
    - ▶ The authentication screen can be simulated
    - ▶ Why is used Ctrl+Alt+Del in MS-Windows?

# Biometrics

- ▶ Authentication by human characteristics
  - ▶ Measurable and distinctive personal characteristics are used to prove identity
    - ▶ Fingerprint (is unique except for twins)
    - ▶ The way the signature is made (dynamics)
    - ▶ Iris Patterns
    - ▶ Retinal patterns
    - ▶ The timbre/spectral composition of voice
    - ▶ Face characteristics
    - ▶ DNA
    - ▶ Relative blood composition
    - ▶ Etc.

# Applications of Biometrics

- ▶ Control access to resources and networks
- ▶ Marking on time clocks
- ▶ Authorization of banking transactions
- ▶ In elections / passports / credit cards



# Tokens

- ▶ Aim to facilitate one-time passwords
  - ▶ Physical card
  - ▶ SecurID
  - ▶ S/Key
  - ▶ Smart card





# Access Control

- ▶ Common to all methods previously described, there is a shared secret that must be agreed between the system and the user
  - ▶ Usually, a password
- ▶ Like that, the sequence of operations is create\_user - share\_secret - access
- ▶ The database where the secret is stored varies according to the operating system and role played by it
  - ▶ Linux: /etc/passwd at the least
  - ▶ Windows: SAM on local systems, Active Directory (AD) on domain controllers
- ▶ For organizations, a centralized database of users and passwords is a better option when compared with creating them on every system

# Access Control

- ▶ This might be a problem due to several reasons
  - ▶ If the system that stores the database is inaccessible, how can a user log?
  - ▶ When a user tries to log, how confident can he be that the system is the one that he expects to be?
  - ▶ How will be the organization aware of user's log?
- ▶ Additionally, if a user logs into a network controlled by a server with several other server for different services, how confident can he be that each one of those servers is the expected one?
- ▶ Some centralized databases for user and eventually systems have been developed to mitigate these legitimate interests

# LDAP

- ▶ *Lightweight Directory Access Protocol (LDAP)* is one of those
- ▶ It is a non-relational database that can host entity authentication
- ▶ Its data structure is very flexible and optimized for query operations
- ▶ Its structure is a tree, each branch being an **object** that has **attributes**
- ▶ The top of the structure is called **root**
- ▶ Attributes are indexed in order to optimize the search/query
- ▶ The client (human or system) accesses the global directory (which can contain multiple LDAP servers) through a client or *Directory User Agent (DUA)*
- ▶ This client in turn interacts with one or more servers or *Directory System Agents (DSA)* through the protocol LDAP

# LDAP

- ▶ The LDAP protocol uses TCP as a transport mechanism and is based on *Abstract Syntax Notation One (ASN.1)*
- ▶ All LDAP messages are encapsulated in a format called LDAPMessage, a string in string format that contains a sequence number, an identifier, the intended operation, and a message
- ▶ If a message in which the sequence number is not recognized, the connection is terminated by returning a disconnection indication with an error code (ProtocolError) to the client
- ▶ In all other cases where it is not possible for the client or server to analyze the message, the connection is immediately terminated and may or may not return a message to the other end

# LDAP

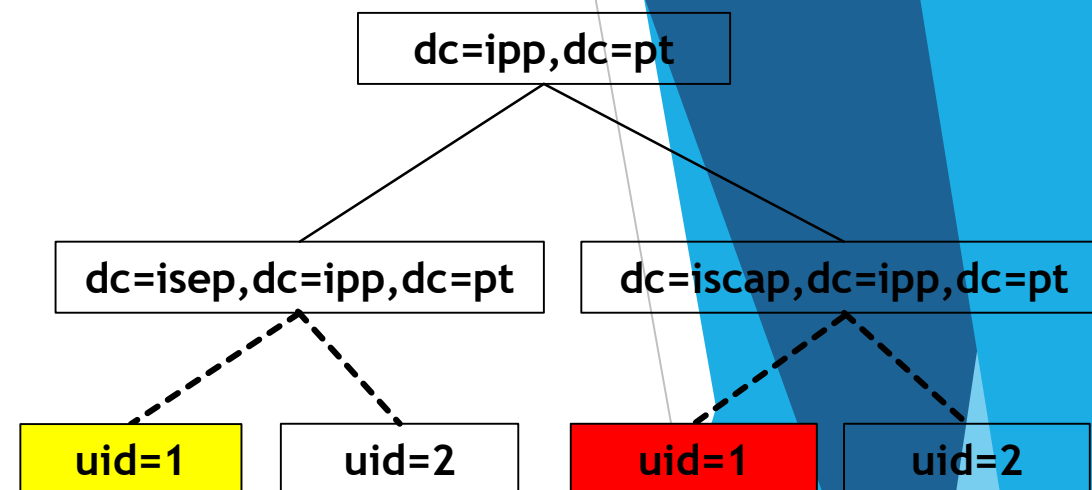
- ▶ The definition of the attributes of objects in LDAP must comply with the definition of the *schema* (defined in RFC 4512) and is therefore not free
- ▶ The *schema* contains the possible classes each containing the attributes that are valid and what kind of use is allowed in each one
- ▶ Some attributes are required, others are optional
- ▶ The definition of classes implements the concept of inheritance, so that when a class is a subclass of another, it incorporates the attributes of the hierarchically superior, and more attributes can be defined in the subclass

# LDAP

- ▶ There are structural and auxiliary classes (however optional)
- ▶ Each object belongs to a single structural class, but can additionally belong to several auxiliary classes
- ▶ Among the mandatory attributes is the **Domain Component (DC)** which can belong to the **dcObject** class (which is an auxiliary) usually combined with the structural class **Organization**
  - ▶ Alternatively you can use the structural class **Domain**
- ▶ Naming based on the DNS domain name has become common, with each of its parts being a **dc**, separated by commas
- ▶ DNS: **isep.ipp.pt**
- ▶ root: **dc=isep,dc=ipp,dc=pt**

# LDAP

- ▶ The name that identifies a single, unique object in the LDAP tree is a **Distinguished Name (DN)**
- ▶ If the DN corresponds to *root*, it is called **base DN** or **root DN**
- ▶ A DN must be unique in LDAP unambiguously identifying the object
- ▶ Attribute names however can be repeated throughout the structure, provided they are on different branches of the tree
- ▶ **Relative Distinguished Names (RDN)** are then distinguished, which, being unique and exclusive in the branch in which they are found, can be repeated in another branch.
- ▶ In the image, objects with RDN 1 have the DN  
(with yellow background) **uid=1,dc=isep,dc=ipp,dc=pt**  
(with red background) **uid=1,dc=iscap,dc=ipp,dc=pt**
- ▶ Under **dc=isep, dc=ipp, dc=pt** there can only be one **uid=1**, being one RDN, yet there may be others **uid <> 1**, as under **dc=iscap, dc=ipp, dc=pt**



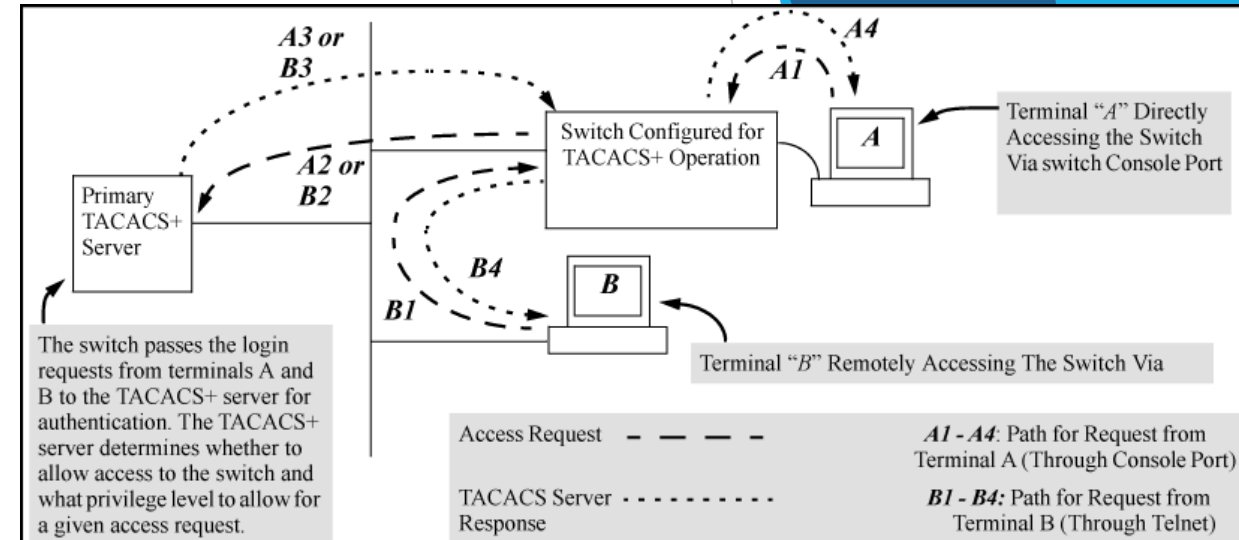
# AAA

- ▶ To allow everyone to be aware (and able to demonstrate) the granted and / or denied access, an **AAA** (***Authentication, Authorization, Accounting***) system is a requisite
- ▶ Protocols have been developed to ensure support for AAA
- ▶ Among these, the ***Remote Authentication Dial In User Service*** (**RADIUS**) (RFC 2865)
- ▶ RADIUS is both a service and a protocol that use UDP to avoid constraints associated with TCP
- ▶ In RADIUS, accounting requests are registered only for the beginning and end of the session
- ▶ It uses 8 bits to set Attribute - Value pairs (AV)
  - ▶ AVPs contain authentication, accounting, authorization, routing, security and configuration information for the request and response



# AAA

- ▶ Another protocol that supports AAA is the **Terminal Access Controller Access-Control System Plus (TACACS+)**
  - ▶ Evolution of TACACS and XTACACS but incompatible with them
- ▶ Unlike RADIUS it uses TCP
- ▶ One significant difference is that TACACS+ isolates the checks (authentication and authorization) and repeats the authorization whenever the execution of a new command is required



Source: techhub.hpe.com

# AAA

- ▶ Later, an evolution based on RADIUS, although not compatible with it, emerged: **Diameter** (RFC 6733 in current version)
- ▶ As most important aspects,
  - ▶ Supports *Extensible Authentication Protocol* (EAP)
  - ▶ Supports *Stream Control Transmission Protocol* (SCTP)
  - ▶ Allows failover mechanisms
  - ▶ It is based on TCP
  - ▶ Uses 32 bits to define Attribute-Value pairs (Attribute-Value Pair AVP) and is therefore more scalable
  - ▶ Supports wired networks, Wi-Fi, 3G, IP Multimedia Systems (IMS), and LTE/4G

# Access Control

- ▶ There is still a problem, the confidence that a user might or can have when accessing a system that it is the correct one
- ▶ This has been solved with **Kerberos**, which is the most common system
- ▶ Kerberos was developed by MIT for internal use and later made available for external use
- ▶ The latest version is V5, described in RFC 4120
- ▶ It offers a ***Single Sign-On (SSO)*** solution to users and provides protection for authentication credentials
- ▶ It is designed to ensure strong authentication for client/server applications through the use of secret key cryptography
- ▶ Through extensions described in RFC 4556, initial authentication can use the ***Public Key Infrastructure (PKI)***

# Kerberos

- ▶ Each of the entities to which Kerberos can distribute keys is called a *principal*
- ▶ Kerberos divides infrastructure into *realms*
- ▶ Each realm must contain the key distribution service called **Key Distribution Center (KDC)**
- ▶ All users and resources (*principals*) are registered on the KDC, which maintains a database of everyone's keys
- ▶ The authentication server performs the functions of the KDC, two distinct and parallel services
  - ▶ *Authentication Service (AS)*
  - ▶ *Ticket Granting Service (TGS)*

# Kerberos

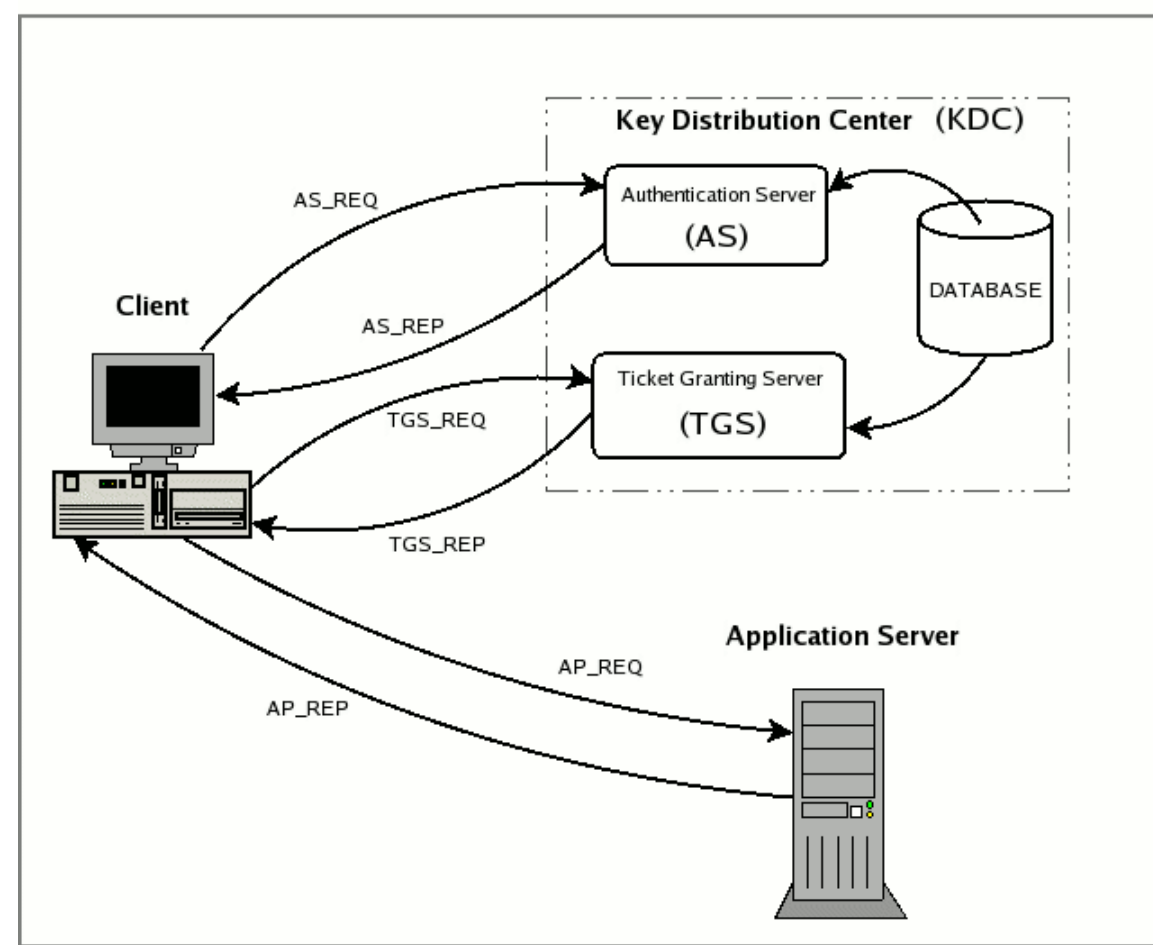
- ▶ **Authentication Server (AS)**
  - ▶ Have a key shared with the principals
  - ▶ Accept or reject the access attempt
  - ▶ Provide principals with a ticket allowing them to prove their identity to TGS
- ▶ **Ticket Granting Service (TGS)**
  - ▶ Provides ***Ticket-Granting tickets*** (TGT), session keys for temporary communication, to principals
  - ▶ Temporary, as TGT has a lifetime
  - ▶ If there is no time synchronization on the principals, access will be denied
  - ▶ TGT proves that the principal has authenticated with the KDC and is authorized to request access to resources

# Kerberos

- ▶ Tickets are encrypted messages that prove authorization to access a resource
- ▶ A user asks for a ticket to access a resource, and if both (user and resource) are authenticated and the user has permission to use the resource, Kerberos (more exactly TGS) provides it
- ▶ In addition to lifetime, Kerberos tickets have specific usage parameters
- ▶ When a ticket expires the customer must request its renewal or a new ticket to continue using the feature
- ▶ Since all principals are authenticated, it ensures the identity and authorization of each other to the entire structure

# Kerberos

- ▶ When a principal connects to the realm it sends an authentication request to the KDC
- ▶ If valid, a Ticket-Granting Ticket (TGT) is returned
- ▶ When the principal accesses after authenticating a resource on the network, the TGT he received is used to request another ticket to access the resource
- ▶ This new request is sent to the KDC which validates it



Source: kerberos.org

# Kerberos

- ▶ Kerberos can manage trust relationships between different realms
- ▶ With this trust relationship, it is possible for a principal of a realm to access resources from a different realm without needing to have credentials in the AS of that other realm
- ▶ This process consists of defining each TGS as a principal in the other realm
- ▶ Regardless of the realm they are in, each principal can get a ticket to access resources from the other realm
- ▶ A realm is considered to be able to communicate with another realm if they both share an inter-realm key
  - ▶ This key will be used to encrypt remote realm resource access request tickets



# Kerberos

- ▶ Kerberos also supports hierarchy between realms
  - ▶ An example is Microsoft's Active Directory domains in the same tree
- ▶ If there is a hierarchy, the trust relationship is transitive, that is, if *realm A* trusts *realm B* and *realm B* trusts *realm C*, then *realm A* trusts *realm C*

# Kerberos

- ▶ As a drawback of Kerberos, note that KDC is a SPOF
- ▶ The keys are based on user passwords, so a "weak" password enhances their capture and subsequent access to the key
- ▶ Kerberos does not provide fault tolerance mechanisms, so it is reasonable to configure more than one KDC in the same realm, however:
  - ▶ The synchronization of databases must be done using external systems (automatic or manual)
  - ▶ Applications should be prepared on a trial-error basis, ie if the main KDC does not respond, try the alternate KDC and so on until success or failure after traversing all KDCs
    - ▶ This functionality is however available in Microsoft's Active Directory
- ▶ If the main KDC is down, Kerberos administration cannot be done until it is restored

# Access Control Models

- ▶ Discretionary Access Control (DAC)
- ▶ Mandatory Access Control (MAC)
- ▶ Role Based Access Control (RBAC)
- ▶ Attribute Based Access Control (ABAC)
- ▶ Access Lists (ACLs)
- ▶ Capabilities
- ▶ Formal Models
  - ▶ Biba
  - ▶ Take/Grant
  - ▶ Clark/Wilson
  - ▶ Bell/LaPadula
    - ▶ Uses Set Theory to define the concept of secure state, access modes and roles in granting accesses

# MAC versus DAC

- ▶ Discretionary Access Control (DAC)
  - ▶ The user of the objects decides how and with whom to share them
  - ▶ The system assumes a secondary role and essentially validates all interactions
- ▶ Imperative or Mandatory Access Control (MAC)
  - ▶ The system decides how objects are shared and validates the interactions
  - ▶ The user has a secondary role

# RBAC versus ABAC

- ▶ Role Based Access Control (RBAC)
  - ▶ Access to objects is given accordingly to the role performed by the user, not the user by itself
  - ▶ The system decides how objects are shared and validates the interactions
- ▶ Attribute Based Access Control (ABAC)
  - ▶ The system decides how objects are shared according to definitions (like location, timeline, function) and validates the interactions

# DAC

- ▶ Relevant aspects
  - ▶ It is the owner of the object who controls the access permissions
  - ▶ The restriction of access is guaranteed by the user's profile of the object
  - ▶ It is used to separate and protect users from unauthorized data
  - ▶ It is used by Unix systems, Windows, Linux, Solaris, FreeBSD and others

# MAC

- ▶ Relevant aspects
  - ▶ Sensitivity levels are used (labels)
  - ▶ Each object is assigned a level of sensitivity and will only be accessible by users with access up to that level
  - ▶ Only system administrators can change the level of objects
  - ▶ It is considered safer than DAC
  - ▶ Used in systems where security is critical

# RBAC

## ► Relevant aspects

- It is a static model based on user roles or profiles within the organization
- Each defined profile has its own permissions to access specific objects (resources, data)
- Allows organizations to eliminate the implementation of individual access controls
- Controls are easy to establish and assign
- Flexibility and simplicity
- Complex in setup and administration on large organizations
- Neglects the principle of least privilege



# ABAC

- ▶ Relevant aspects
  - ▶ Dynamic model that leverages data and attributes unique to the user
  - ▶ Allows the creation of more complex and flexible authorization rules that take into account various attributes in addition to the profile, such as roles, location, time and date, among others
  - ▶ Allows more complex and flexible authorization rules to be defined, consistent security policies to be enforced, and risk-based security policies (e.g., conditional access) to be enforced
  - ▶ Requires little maintenance and is scalable
  - ▶ Initial configuration is complex
  - ▶ Management in large systems can become complicated when it comes to inheriting permissions, the need to grant more specific privileges makes it difficult to handle/create access profiles

# Formal Methods

- ▶ Problems of formal methods
  - ▶ They are based on static infrastructures
  - ▶ They define very brief policies
  - ▶ Do not work in extremely dynamic and reconfigurable enterprise environments
  - ▶ None of the models mentioned deals with
    - ▶ Viruses/active content
    - ▶ Trojan horses and others
    - ▶ Firewalls
  - ▶ Documentation is scarce to assist in the implementation phase of systems

# Access Control Lists (ACL)

- ▶ Relevant aspects
  - ▶ It is a mechanism used by the access control system to determine who can access which programs, by what methods and at what times
  - ▶ There are implementations of access control lists in all modern operating systems
  - ▶ Some types of access
    - ▶ Read/Write/Execute
    - ▶ Create/Modify/Delete/Rename

# Capabilities

- ▶ Similar to ACL but read in reverse
- ▶ Each line contains the user and the accesses he has to the resources

ACL →

Object	r	w	x
--------	---	---	---

Capabilities →

User	Resource 1	Resource 2	Resource N
Bob	r	rw	x

# ACL versus Capabilities

- ▶ ACL
  - ▶ Simple to implement
  - ▶ Data oriented
  - ▶ Less suited to environments where users are constantly changing
- ▶ Capabilities
  - ▶ Runtime security checking is more efficient
  - ▶ Delegation of rights without much difficulty
  - ▶ Change of a file status can suddenly become more tricky as it can be difficult to find out which users have access

# UNIX Permissions

- ▶ Standard UNIX Systems
  - ▶ Permissions are *Read Write eXecute*
  - ▶ User profiles are *User Group Other*
  - ▶ Permissions applied to normal files
    - ▶ How They Work?
  - ▶ Permissions applied to directories
    - ▶ How They Work?
- ▶ Linux systems with file systems Ext2/3/4 and others
  - ▶ Additional permissions to those of rwx
    - ▶ a c d i j s t u A C D S T

# Windows Permissions

- ▶ Windows systems
  - ▶ Permissions are *Read eXecute Write Delete*
  - ▶ The permissions categories are many...
    - ▶ No access / List / Read
    - ▶ Add / Add & Read
    - ▶ Change / Full Control
  - ▶ For each object there is a list of entities that can manipulate this object
    - ▶ More versatile and simple to configure than the standard UNIX access control model

# Access Control

- ▶ The gathering of all these properties (and some more irrelevant here) is the domain of ***Network Access Control (NAC)***
- ▶ In fact, NAC is a concept of controlling access to an infrastructure by implementing security policies
- ▶ Its goals are to
  - ▶ Mitigate and/or prevent zero-day attacks
  - ▶ Apply a security policy across the infrastructure
  - ▶ Use identities to perform access control
- ▶ These objectives can be achieved through various mechanisms (security policies with definition of security control, filtering, prevention, among others)
- ▶ NAC will act as an automatic detection and response system that can react in real time to prevent threats before they do damage



# NAC

- ▶ NAC can be implemented using two criteria separately or a mixture of both
- ▶ **Pre-admission**
  - ▶ Requires a system to meet all security requirements (such as installed updates, updated antivirus) before it can communicate with the network
- ▶ **Post-admission**
  - ▶ Allow or deny access based on user activity, which in turn is based on a previously defined authorization matrix
- ▶ The pre-admission criteria refers mainly to systems and services
- ▶ The post-admission criteria refers mainly to users