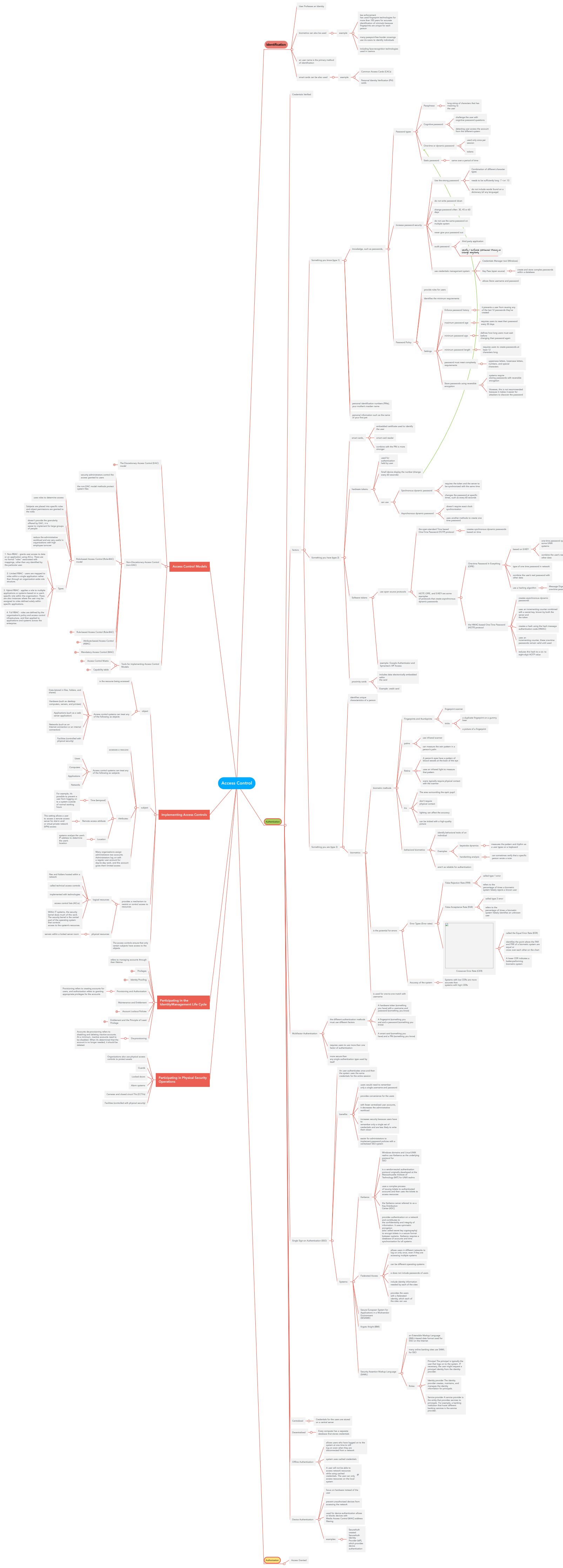

Access Control
Identification
Authentication
Authorization
Access Control Models
Implementing Access Controls
Participating in the IdentityManagement Life Cycle
Participating in Physical Security Operations

Access Control

1. Identification

1.1. User Professes an Identity

1.2. biometrics can also be used

1.2.1. example

1.2.1.1. law enforcement has used fingerprint technologies for more than 100 years for accurate identification of criminals because fingerprints are unique for each person

1.2.1.2. many passport-free border crossings use iris scans to identify individuals

1.2.1.3. including face-recognition technologies used in casinos

1.3. an user name is the primary method of identification

1.4. smart cards can be also used

1.4.1. example

1.4.1.1. Common Access Cards (CACs)

1.4.1.2. Personal Identity Verification (PIV) cards

2. Authentication

2.1. Credentials Verified

2.2. factors

2.2.1. Something you know (type 1)

2.2.1.1. knowledge, such as passwords,

2.2.1.1.1. Password types

2.2.1.1.1.1. Passphrase

2.2.1.1.1.1.1. long string of characters that has meaning to the user

2.2.1.1.1.2. Cognitive password

2.2.1.1.1.2.1. challenge the user with cognitive password questions

2.2.1.1.1.2.2. detecting user access the account from the different system

2.2.1.1.1.3. One-time or dynamic password

2.2.1.1.1.3.1. used only once per session

2.2.1.1.1.3.2. tokens

2.2.1.1.1.4. Static password

2.2.1.1.1.4.1. same over a period of time

2.2.1.1.2. Increase password security

2.2.1.1.2.1. Use the strong password

2.2.1.1.2.1.1. Combination of different character types

2.2.1.1.2.1.2. needs to be sufficiently long: 7 <x< 13

2.2.1.1.2.1.3. do not include words found on a dictionary (of any language)

2.2.1.1.2.2. do not write password down

2.2.1.1.2.3. change password often: 30, 45 or 60 days

2.2.1.1.2.4. do not use the same password on multiple system

2.2.1.1.2.5. never give your password out

2.2.1.1.2.6. audit password

2.2.1.1.2.6.1. third party application

2.2.1.1.2.6.2. verify / enforce password strong or change regularly

2.2.1.1.2.7. use credentials management system

2.2.1.1.2.7.1. Credentials Manager tool (Windows)

2.2.1.1.2.7.2. Key Pass (open source)

2.2.1.1.2.7.2.1. create and store complex passwords within a database

2.2.1.1.2.7.3. allows Store username and password

2.2.1.1.3. Password Policy

2.2.1.1.3.1. provide rules for users

2.2.1.1.3.2. Identifies the minimum requirements

2.2.1.1.3.3. Settings

2.2.1.1.3.3.1. Enforce password history

2.2.1.1.3.3.1.1. it prevents a user from reusing any of the last 12 passwords they've created

2.2.1.1.3.3.2. maximum password age

2.2.1.1.3.3.2.1. requires users to reset their password every 30 days

2.2.1.1.3.3.3. minimum password age

2.2.1.1.3.3.3.1. defines how long users must wait before changing their password again

2.2.1.1.3.3.4. minimum password length

2.2.1.1.3.3.4.1. requires users to create passwords at least 12 characters long

2.2.1.1.3.3.5. password must meet complexity requirements

2.2.1.1.3.3.5.1. uppercase letters, lowercase letters, numbers, and special characters

2.2.1.1.3.3.6. Store passwords using reversible encryption

2.2.1.1.3.3.6.1. systems require storing passwords with reversible encryption

2.2.1.1.3.3.6.2. However, this is not recommended because it makes it easier for attackers to discover the password

2.2.1.2. personal identification numbers (PINs), your mother’s maiden name

2.2.1.3. personal information such as the name of your first pet

2.2.2. Something you have (type 2)

2.2.2.1. smart cards,

2.2.2.1.1. embedded certificate used to identify the user

2.2.2.1.2. smart card reader

2.2.2.1.3. combine with the PIN is more stronger

2.2.2.2. hardware tokens.

2.2.2.2.1. used for authentication held by user

2.2.2.2.2. Small device display the number (change every 60 seconds)

2.2.2.2.3. can use

2.2.2.2.3.1. Synchronous dynamic password

2.2.2.2.3.1.1. requires the token and the server to be synchronized with the same time

2.2.2.2.3.1.2. changes the password at specific times, such as every 60 seconds

2.2.2.2.3.2. Asynchronous dynamic password

2.2.2.2.3.2.1. doesn’t require exact clock synchronization

2.2.2.2.3.2.2. uses another methods to create one time password

2.2.2.3. Software tokens

2.2.2.3.1. use open source protocols

2.2.2.3.1.1. the open-standard Time based One-Time Password (TOTP) protocol

2.2.2.3.1.1.1. creates synchronous dynamic passwords based on time

2.2.2.3.1.2. HOTP, OPIE, and S/KEY are some examples of protocols that create asynchronous dynamic passwords

2.2.2.3.1.2.1. One-time Password In Everything (OPIE)

2.2.2.3.1.2.1.1. based on S/KEY

2.2.2.3.1.2.1.1.1. one-time password system used on some UNIX systems

2.2.2.3.1.2.1.1.2. combine the user’s real password with other data

2.2.2.3.1.2.1.2. type of one time password in network

2.2.2.3.1.2.1.3. combine the user’s real password with other data

2.2.2.3.1.2.1.4. use a hashing algorithm

2.2.2.3.1.2.1.4.1. Message Digest 5 (MD5) to create a one-time password

2.2.2.3.1.2.2. the HMAC-based One-Time Password (HOTP) protocol

2.2.2.3.1.2.2.1. creates asynchronous dynamic passwords

2.2.2.3.1.2.2.2. uses an incrementing counter combined with a secret key, known by both the server and the token

2.2.2.3.1.2.2.3. creates a hash using the hash message authentication code (HMAC)

2.2.2.3.1.2.2.4. uses an incrementing counter, these one-time passwords remain valid until used

2.2.2.3.1.2.2.5. reduces this hash to a six- to eight-digit HOTP value

2.2.2.3.2. example: Google Authenticator and Symantec’s VIP Access

2.2.2.4. proximity cards

2.2.2.4.1. includes data electronically embedded within the card

2.2.2.4.2. Example: credit card

2.2.3. Something you are (type 3)

2.2.3.1. identifies unique characteristics of a person

2.2.3.2. biometrics

2.2.3.2.1. biometric methods

2.2.3.2.1.1. Fingerprints and thumbprints

2.2.3.2.1.1.1. fingerprint scanner

2.2.3.2.1.1.2. tricks

2.2.3.2.1.1.2.1. a duplicate fingerprint on a gummy bear

2.2.3.2.1.1.2.2. a picture of a fingerprint

2.2.3.2.1.2. palms

2.2.3.2.1.2.1. use infrared scanner

2.2.3.2.1.2.2. can measure the vein pattern in a person’s palm

2.2.3.2.1.3. Retina

2.2.3.2.1.3.1. A person’s eyes have a pattern of blood vessels at the back of the eye

2.2.3.2.1.3.2. uses an infrared light to measure that pattern

2.2.3.2.1.3.3. scans typically require physical contact with the scanner

2.2.3.2.1.4. Iris

2.2.3.2.1.4.1. The area surrounding the eye’s pupil

2.2.3.2.1.4.2. don’t require physical contact

2.2.3.2.1.4.3. lighting can affect the accuracy

2.2.3.2.1.4.4. can be tricked with a high-quality picture

2.2.3.2.1.5. behavioral biometrics

2.2.3.2.1.5.1. identify behavioral traits of an individual

2.2.3.2.1.5.2. Examples

2.2.3.2.1.5.2.1. keystroke dynamics

2.2.3.2.1.5.2.1.1. measures the pattern and rhythm as a user types on a keyboard

2.2.3.2.1.5.2.2. handwriting analysis

2.2.3.2.1.5.2.2.1. can sometimes verify that a specific person wrote a note

2.2.3.2.1.5.3. aren’t as reliable for authentication

2.2.3.2.2. is the potential for errors

2.2.3.2.2.1. Error Types (Error rates)

2.2.3.2.2.1.1. False Rejection Rate (FRR)

2.2.3.2.2.1.1.1. called type 1 error

2.2.3.2.2.1.1.2. refers to the percentage of times a biometric system falsely rejects a known user

2.2.3.2.2.1.2. False Acceptance Rate (FAR)

2.2.3.2.2.1.2.1. called type 2 error

2.2.3.2.2.1.2.2. refers to the percentage of times a biometric system falsely identifies an unknown user

2.2.3.2.2.1.3. Crossover Error Rate (CER)

2.2.3.2.2.1.3.1. called the Equal Error Rate (EER)

2.2.3.2.2.1.3.2. identifies the point where the FAR and FRR of a biometric system are equal or cross over each other on the chart

2.2.3.2.2.1.3.3. A lower CER indicates a better-performing biometric system

2.2.3.2.2.2. Accuracy of the system

2.2.3.2.2.2.1. Systems with low CERs are more accurate than systems with high CERs

2.2.3.2.3. is used for one-to-one match with username

2.3. Multifactor Authentication

2.3.1. the different authentication methods must use different factors

2.3.1.1. A hardware token (something you have) with a username and password (something you know)

2.3.1.2. A fingerprint (something you are) and a password (something you know)

2.3.1.3. A smart card (something you have) and a PIN (something you know)

2.3.2. requires users to use more than one factor of authentication

2.3.3. more secure than any single authentication type used by itself

2.4. Single Sign-on Authentication (SSO)

2.4.1. An user authenticates once and then the system uses the same credentials for the entire session

2.4.2. benefits:

2.4.2.1. users would need to remember only a single username and password

2.4.2.2. provides convenience for the users

2.4.2.3. with fewer centralized user accounts, it decreases the administrative workload

2.4.2.4. increases security because users have to remember only a single set of credentials and are less likely to write them down

2.4.2.5. easier for administrators to implement password policies with a centralized SSO system

2.4.3. Systems:

2.4.3.1. Kerberos

2.4.3.1.1. Windows domains and Linux/UNIX realms use Kerberos as the underlying protocol for SSO

2.4.3.1.2. is a vendor-neutral authentication protocol originally developed at the Massachusetts Institute of Technology (MIT) for UNIX realms

2.4.3.1.3. uses a complex process of issuing tickets to authenticated accounts and then uses the tickets to access resources

2.4.3.1.4. the Kerberos server referred to as a Key Distribution Center (KDC)

2.4.3.1.5. provides authentication on a network and contributes to the confidentiality and integrity of information. It uses symmetric encryption (also called secret key cryptography) to encrypt tickets in a secure format between systems. Kerberos requires a database of accounts and time synchronization for all systems

2.4.3.2. Federated Access

2.4.3.2.1. allows users in different networks to log on only once, even if they are accessing multiple systems

2.4.3.2.2. can be different operating systems

2.4.3.2.3. e does not include passwords of users

2.4.3.2.4. include identity information needed by each of the sites

2.4.3.2.5. provides the users with a federated identity, which each of the sites can use

2.4.3.3. Secure European System for Applications in a Multivendor Environment (SESAME)

2.4.3.4. Krypto Knight (IBM)

2.4.3.5. Security Assertion Markup Language (SAML)

2.4.3.5.1. an Extensible Markup Language (XML)–based data format used for SSO on the Internet

2.4.3.5.2. many online banking sites use SAML for SSO

2.4.3.5.3. Roles:

2.4.3.5.3.1. Principal The principal is typically the user that logs on to the system. IF necessary, the user might request a principal identity from the identity provider.

2.4.3.5.3.2. Identity provider The identity provider creates, maintains, and manages the identity information for principals.

2.4.3.5.3.3. Service provider A service provider is the entity that provides services to principals. For example, a banking institution that hosts different banking services is the service provider.

2.5. Centralized

2.5.1. Credentials for the users are stored on a central server

2.6. Decentralized

2.6.1. Every computer has a separate database that stores credentials

2.7. Offline Authentication

2.7.1. allows users who have logged on to the system at one time to still log on even when they are disconnected from a network

2.7.2. system uses cached credentials

2.7.3. A user will not be able to access network resources while using cached credentials. The user can only access resources on the local system

Link: https://www.mindmeister.com/signup/index?f=pricing_map_header

2.8. Device Authentication

2.8.1. focus on hardware instead of the user

2.8.2. prevent unauthorized devices from accessing the network

2.8.3. used for device authentication allows or blocks devices with Media Access Control (MAC) address filtering

2.8.4. examples:

2.8.4.1. SecureAuth created SecureAuth Identity Provider (IdP), which provides device authentication

3. Authorization

3.1. Access Granted

4. Participating in Physical Security Operations

4.1. Organizations also use physical access controls to protect assets

4.2. Guards

4.3. Locked doors

4.4. Alarm systems

4.5. Cameras and closed circuit TVs (CCTVs)

4.6. Facilities (controlled with physical security)

5. Participating in the IdentityManagement Life Cycle

5.1. refers to managing accounts through their lifetime

5.2. Privileges

5.2.1. Rights

5.2.1.1. actions that an account can take on a system, such as backing up files, changing the time, or rebooting the computer

5.2.2. Permissions

5.2.2.1. identify what a user can do with resources, such as reading and writing to a file or printing to a printer

5.3. Identity Proofing

5.3.1. It’s important to verify a person’s identity before creating a user account

5.3.2. Human Resources (HR) departments

5.3.2.1. s require documents

5.3.3. protecting from the fraud

5.3.3.1. a credit card verification value (CVV).

5.3.3.2. Cognitive password

5.4. Provisioning and Authorization

5.4.1. Provisioning refers to creating accounts for users, and authorization refers to granting appropriate privileges for the accounts

5.5. Maintenance and Entitlement

5.6. Account Lockout Policies

5.6.1. locks out an account after too many failed logon attempts. The goal is to prevent attackers from guessing passwords

5.6.2. Settings

5.6.2.1. Threshold

5.6.2.1.1. s identifies how many incorrect passwords are allowed

5.6.2.2. Duration

5.6.2.2.1. identifies how long the account remains locked out

5.7. Entitlement and the Principle of Least Privilege

5.7.1. Entitlement refers to the privileges granted to users, and following the principle of least privilege is an important part of entitlement.

5.7.2. Removing a user from a group immediately removes all the privileges assigned to the user as a member of that group

5.8. De-provisioning

5.8.1. Accounts de-provisioning refers to disabling and deleting inactive accounts. At a minimum, inactive accounts need to be disabled. When it’s determined that the account is no longer needed, it should be deleted.

6. Implementing Access Controls

6.1. object

6.1.1. is the resource being accessed

6.1.2. Access control systems can treat any of the following as objects:

6.1.2.1. Data (stored in files, folders, and shares)

6.1.2.2. Hardware (such as desktop computers, servers, and printers)

6.1.2.3. Applications (such as a web server application)

6.1.2.4. Networks (such as an Internet connection or an internal connection)

6.1.2.5. Facilities (controlled with physical security)

6.2. subject

6.2.1. accesses a resource

6.2.2. Access control systems can treat any of the following as subjects:

6.2.2.1. Users

6.2.2.2. Computers

6.2.2.3. Applications

6.2.2.4. Networks

6.2.3. Attributes:

6.2.3.1. Time (temporal)

6.2.3.1.1. For example, it’s possible to prevent a user from logging on to a system outside of normal working hours

6.2.3.2. Remote access attribute

6.2.3.2.1. This setting allows a user to access a remote access server for dial-in and/ or virtual private network (VPN) access

6.2.3.3. Location

6.2.3.3.1. systems analyze the user’s IP address to determine the user’s location

6.2.4. Many organizations assign administrators two accounts. Administrators log on with a regular user account for day-to-day work, and this account gives them limited access

6.3. provides a mechanism to restrict or control access to resources

6.3.1. logical resources

6.3.1.1. files and folders hosted within a network

6.3.1.2. called technical access controls

6.3.1.3. implemented with technologies

6.3.1.4. access control lists (ACLs).

6.3.1.5. Within IT systems, the security kernel does much of this work. The security kernel is the central part of the operating system that controls access to the system’s resources.

6.3.2. physical resources

6.3.2.1. servers within a locked server room

6.4. The access controls ensure that only certain subjects have access to the objects

7. Access Control Models

7.1. The Discretionary Access Control (DAC) model

7.1.1. provides the most granular level of access control

7.1.2. is an identity-based model and allows data owners to assign permissions to subjects at the most basic level

7.1.3. used by New Technology File System (NTFS, used by Microsoft) and Network File System (NFS, used on UNIX-based systems such as Solaris and Linux)

7.1.4. users have ownership of the data and can exercise full control over it, including assigning permissions to others

7.1.4.1. The CREATOR OWNER group

7.1.4.1.1. is a special group that ensures that users are automatically assigned full control permission to any file or folder they create

7.1.4.2. Administrator

7.1.4.2.1. Can add new user group, it will take all permissions which this group allowed

7.1.4.2.2. remove from the group, it revokes all permissions which that group owned from the user

7.1.4.2.3. place user accounts into groups and assign appropriate privileges (imtiyozlar) to the groups

7.1.5. uses ACLs, or, more specifically, discretionary access control lists (DACLs)

7.1.5.1. Each entry in the DACL includes the SID (security identifier) and the permission assigned to that SID

7.1.5.2. A DACL is composed of several access control entries (ACEs).Individual ACEs Creating a DACL

7.2. Non-Discretionary Access Control (non-DAC)

7.2.1. security administrators control the access granted to users

7.2.2. the non-DAC model methods protect system files

7.2.3. Role-based Access Control (Role-BAC) model

7.2.3.1. uses roles to determine access

7.2.3.2. Subjects are placed into specific roles and object permissions are granted to the roles

7.2.3.3. doesn’t provide the granularity offered by DAC, it is easier to implement for large groups of people

7.2.3.4. reduce the administrative workload and are very useful in organizations with high employee turnover

7.2.3.5. Types

7.2.3.5.1. 1. Non-RBAC - grants user access to data or an application using ACLs. There are no formal “roles” associated with mappings, other than any identified by the particular user.

7.2.3.5.2. 2. Limited RBAC - users are mapped to roles within a single application rather than through an organization-wide role structure.

7.2.3.5.3. 3. Hybrid RBAC - applies a role to multiple applications or systems based on a user’s specific role within the organization. There are also instances where the user may be assigned to roles defined solely within specific applications.

7.2.3.5.4. 4. Full RBAC - roles are defined by the organization’s policy and access control infrastructure, and then applied to applications and systems across the enterprise.

7.2.4. Rule-based Access Control (Rule-BAC)

7.2.4.1. Administrators create rules that determine access to resources.

7.2.4.1.1. routers have rules within an ACL

7.2.4.2. rules identify what traffic the router will pass based on IP addresses, ports, and protocols

7.2.4.3. The last rule in a router is an implicit deny rule. It blocks all traffic that isn't explicitly allowed by previous rules. Permissions assigned in DACLs use a similar concept

7.2.4.3.1. example :you explicitly grant permissions to users for a folder. If you don't assign permissions to a specific user, the system blocks that user from accessing the folder

7.2.5. Attribute-based Access Control (ABAC)

7.2.5.1. a more sophisticated type of Rule-based Access Control

7.2.5.2. evaluates subject and object attributes, and grants access based on the value of these attributes

7.2.5.3. systems implement ABAC with plain-language statements to build POLICIES which work as rules

7.2.5.3.1. Policy statements typically include four elements:

7.2.5.3.1.1. subject

7.2.5.3.1.1.1. The user accessing a resource. You can use any user property as an attribute. This includes roles, group memberships, management level, and assigned department. The preceding example uses the status of the user—logged on

7.2.5.3.1.2. Object

7.2.5.3.1.2.1. The resource that the user is trying to access. It can be a file, a database, or an application. The example uses the YouTube application. Notice that this allows the rule to cover much more than just Internet access.

7.2.5.3.1.3. Action

7.2.5.3.1.3.1. What the user is attempting to do. It could be reading or manipulating a file, accessing specific websites, or accessing website applications. The example allows access to the website via the guest network. This could prevent video streaming from overloading the primary network.

7.2.5.3.1.4. Environment

7.2.5.3.1.4.1. Includes everything outside of the subject and object attributes. This is often referred to as the context of the access request. It can include the time, location, protocols, encryption, devices, and communication method. In the example, it specifies smartphones, tablets, and the guest network as environmental attributes.

7.2.5.3.2. Policy statement is stated in plain language, it would typically be stored in a database using an XML format

7.2.5.3.3. The eXtensible Access Control Markup Language (XACML) is a specialized XML format used specifically for ABAC.

7.2.6. Mandatory Access Control (MAC)

7.2.6.1. uses labels to identify both subjects and objects

7.2.6.2. provides the highest level of security among the models (MAC, DAC, Role-BAC, Rule-BAC, and ABAC)

7.2.6.3. is commonly used by the U.S. military to ensure that data is protected in mission-critical systems

7.2.6.3.1. The U.S. government uses the following classifications for data, from highest to lowest:

7.2.6.3.1.1. • Top Secret • Secret • Confidential • Unclassified

7.2.6.4. uses labels to control access to data. It is the most secure model when compared to other access control models.

7.2.6.5. MAC-based architecture models

7.2.6.5.1. Bell-LaPadula

7.2.6.5.1.1. designed by David Elliott Bell and Leonard J. LaPadula

7.2.6.5.1.2. a primary goal of ensuring confidentiality

7.2.6.5.1.3. enforces security through two primary rules

7.2.6.5.1.3.1. • Simple security property rule—no read up

7.2.6.5.1.3.1.1. Subjects granted access to any security level may not read an object at a higher security level

7.2.6.5.1.3.2. • The * property (read as “star-property”) rule—no write down

7.2.6.5.1.3.2.1. Subjects granted access to any security level may not write to any object at a lower security level

7.2.6.5.1.3.3. These rules compare the subject’s clearance with the object’s classification. Confidentiality ensures that unauthorized personnel cannot access data.

7.2.6.5.2. Biba

7.2.6.5.2.1. enforces integrity

7.2.6.5.2.2. includes two rules that are reversed from the Bell-LaPadula model:

7.2.6.5.2.2.1. • Simple Integrity Axiom—no read down

7.2.6.5.2.2.1.1. Subjects granted access to any security level may not read an object at a lower security level, at least not as the authoritative source.

7.2.6.5.2.2.2. •The * Integrity Axiom (read as “star Integrity Axiom”)—no write up

7.2.6.5.2.2.2.1. Subjects granted access to any security level may not write to any object at a higher security level.

7.2.6.5.2.3. uses the no read down and no write up rules to enforce integrity. Integrity protects against unauthorized data modifications.

7.2.6.5.3. Clark-Wilson

7.2.6.5.3.1. created by David Clark and David Wilson

7.2.6.5.3.2. primary goal is information integrity, although it is more stringent than the Biba model

7.2.6.5.3.3. helps enforce the separation of duties principle

7.2.6.5.3.4. model uses certification rules (identified as C1 through C5) and enforcement rules (identified as E1 through E4) to enforce separation of duties. The certification rules are integrity-monitoring rules, and the enforcement rules are integrity preserving rules

7.2.6.5.3.5. model ensures that different people perform the separate tasks independently of each other

7.2.6.5.4. Chinese Wall

7.2.6.5.4.1. documented by Dr. David F. C. Brewer and Dr. Michael J. Nash (and is sometimes referred to as the Brewer-Nash model)

7.2.6.5.4.2. help prevent a conflict of interest, and it helps enforce the separation of duties principle

7.2.6.5.4.3. Data is classified using different conflict-of-interest classes

7.2.6.5.4.4. provides a barrier between these two groups of employees by classifying data

7.2.6.5.4.5. can be as simple as classifying data as Trader Data and Advisor Data and indicating the two classes have a conflict of interest

7.3. Tools for implementing Access Control Models

7.3.1. Access Control Matrix

7.3.1.1. a list of objects along with the permissions granted for each object

7.3.1.2. a group of ACLs. Each ACL represents a single object and lists all the permissions for that object

7.3.2. Capability table

7.3.2.1. a list of subjects, along with the capabilities granted to the subjects

7.3.2.2. These capabilities include rights and permissions

7.3.2.3. can list several groups such as Project Managers, Project Team Leads, and Project Members. It then lists the rights and permissions granted to each of these groups.