Study Guide for Chapter 4 Access Control

March 6, 2025

1 Access Control Principles

1.1 Access Control Context

Access control is a fundamental aspect of computer security, ensuring that only authorized entities can access specific resources. It involves:

- **Authentication**: Verifying the credentials of a user or system entity.
- **Authorization**: Granting rights or permissions to access resources.
- **Audit**: Reviewing system records to ensure compliance and detect security breaches.

1.2 Access Control Policies

Access control policies dictate who or what can access specific resources and the type of access permitted. Policies are categorized into:

- **Discretionary Access Control (DAC)**: Access is based on the identity of the requestor and access rules.
- **Mandatory Access Control (MAC)**: Access is based on security labels and clearances.
- **Role-Based Access Control (RBAC)**: Access is based on user roles within the system.
- **Attribute-Based Access Control (ABAC)**: Access is based on attributes of the user, resource, and environment.

2 Subjects, Objects, and Access Rights

2.1 Subjects

A **subject** is an entity capable of accessing objects, typically equated with a process. Subjects are categorized into:

- **Owner**: Creator or administrator of a resource.
- **Group**: Named group of users with specific access rights.
- **World**: Users not included in owner or group categories.

2.2 Objects

An **object** is a resource to which access is controlled, such as files, directories, or devices.

2.3 Access Rights

Access rights describe how a subject may access an object, including:

- **Read**: View information.
- **Write**: Modify or delete data.
- **Execute**: Run programs.
- **Delete**: Remove resources.
- **Create**: Generate new resources.
- **Search**: List or search directories.

3 Discretionary Access Control (DAC)

3.1 Access Control Model

DAC allows entities to grant access rights to others. It is implemented using:

- **Access Control Lists (ACLs)**: Lists users and their permitted access rights for each object.
- **Capability Tickets**: Specify authorized objects and operations for a user.

3.2 Protection Domains

A **protection domain** is a set of objects with access rights. Domains can be static or dynamic, allowing processes to have different access rights at different times.

4 Example: Unix File Access Control

4.1 Traditional UNIX File Access Control

UNIX uses a hierarchical file system with permissions for owner, group, and others. Special permissions include:

- **SetUID**: Temporarily grants the user the rights of the file owner.
- **SetGID**: Temporarily grants the user the rights of the file group.
- **Sticky Bit**: Restricts file deletion to the owner.

4.2 Access Control Lists in UNIX

Modern UNIX systems support extended ACLs, allowing more flexible access control by assigning permissions to named users and groups.

5 Role-Based Access Control (RBAC)

5.1 RBAC Reference Models

RBAC assigns access rights to roles rather than individual users. The NIST RBAC standard defines four models:

- **RBAC**$_0$: Base model with users, roles, permissions, and sessions.
- **RBAC**$_1$: Adds role hierarchies.
- **RBAC**$_2$: Adds constraints.
- **RBAC**$_3$: Combines RBAC$_1$ and RBAC$_2$.

5.2 Role Hierarchies

Role hierarchies allow roles to inherit permissions from subordinate roles, reflecting organizational structures.

5.3 Constraints

Constraints restrict role assignments and permissions, such as mutually exclusive roles and cardinality limits.

6 Attribute-Based Access Control (ABAC)

6.1 Attributes

ABAC uses attributes of subjects, objects, and the environment to make access control decisions. Attributes include:

- **Subject Attributes**: Characteristics of the user or process.
- **Object Attributes**: Characteristics of the resource.
- **Environment Attributes**: Contextual information like time or location.

6.2 ABAC Logical Architecture

ABAC evaluates access requests based on predefined rules and attributes, providing fine-grained access control.

6.3 ABAC Policies

Policies define rules for access based on attributes, allowing for flexible and dynamic access control.

7 Identity, Credential, and Access Management (ICAM)

7.1 Identity Management

ICAM manages digital identities and attributes, ensuring trustworthy identities across applications.

7.2 Credential Management

Credentials bind identities to tokens, such as smart cards or digital certificates, and are managed throughout their lifecycle.

7.3 Access Management

Access management ensures that entities are granted appropriate access to resources based on their identity and attributes.

7.4 Identity Federation

Identity federation allows organizations to trust digital identities and attributes from external sources, facilitating collaboration.

8 Trust Frameworks

8.1 Traditional Identity Exchange Approach

Traditional identity exchange involves agreements between identity service providers and relying parties, ensuring trust in shared identity information.

8.2 Open Identity Trust Framework (OITF)

OITF provides a standardized approach to identity and attribute exchange, ensuring trust and security in digital transactions.

9 Case Study: RBAC System for a Bank

The Dresdner Bank implemented an RBAC system to manage access to various applications. Roles were defined by job function and position, with access rights assigned based on roles. The system improved security and reduced administrative overhead.

10 Recommended Reading

Key references for further study include:

- [**DOWN85**]: Basic elements of DAC.
- [**SAND96**]: Comprehensive overview of RBAC.
- [**HU13**]: Overview of ABAC models.
- [**CIOC11**]: Introduction to ICAM.
- [**RUND10**]: Overview of OITF.

11 Key Terms

Key terms include:

- **Access Control List (ACL)**: A list of permissions attached to an object.
- **Role-Based Access Control (RBAC)**: Access control based on user roles.
- **Attribute-Based Access Control (ABAC)**: Access control based on attributes.
- **Identity Federation**: Trusting digital identities from external organizations.
- **Protection Domain**: A set of objects with associated access rights.

Multiple Choice Questions

1. Which of the following is NOT a category of access control policies?

   A) Discretionary Access Control (DAC)
   B) Mandatory Access Control (MAC)
   C) Role-Based Access Control (RBAC)
   D) User-Based Access Control (UBAC)

   Answer: D

2. In the context of access control, what is a subject?

   A) A resource to which access is controlled
   B) An entity capable of accessing objects
   C) A set of rules governing access
   D) A type of access right

   Answer: B

3. Which of the following is an example of an access right?

   A) Read
   B) Write
   C) Execute
   D) All of the above

   Answer: D

4. What is the primary characteristic of Discretionary Access Control (DAC)?

   A) Access is based on security labels
   B) Access is based on the identity of the requestor
   C) Access is based on user roles
   D) Access is based on environmental conditions

   Answer: B

5. Which of the following is true about Role-Based Access Control (RBAC)?

   A) Access rights are assigned to individual users
   B) Access rights are assigned to roles
   C) Access rights are based on security labels
   D) Access rights are based on environmental conditions

   Answer: B

6. What is a protection domain in access control?

   A) A set of objects with associated access rights
   B) A list of users and their permissions
   C) A type of access control policy
   D) A security label

   Answer: A

7. In UNIX file access control, what does the SetUID permission do?

   A) Grants the user the rights of the file owner
   B) Grants the user the rights of the file group
   C) Restricts file deletion to the owner
   D) Allows the file to be executed as a program

   Answer: A

8. Which of the following is a key element of Attribute-Based Access Control (ABAC)?

   A) Attributes
   B) Roles
   C) Security labels
   D) Access control lists

   Answer: A

9. What is the purpose of identity federation in access control?

   A) To create digital identities for users
   B) To manage credentials throughout their lifecycle
   C) To trust digital identities from external organizations
   D) To define access control policies

   Answer: C

10. Which of the following is a type of constraint in RBAC?

    A) Mutually exclusive roles
    B) Role hierarchies
    C) Access control lists
    D) Protection domains

    Answer: A

11. What is the primary function of an access control list (ACL)?

    A) To define roles and their permissions
    B) To list users and their permitted access rights for an object
    C) To manage credentials
    D) To define protection domains

    Answer: B

12. Which of the following is true about the NIST RBAC standard?

    A) It defines four models: RBAC_0, RBAC_1, RBAC_2, and RBAC_3
    B) It is based on security labels
    C) It is primarily used in UNIX systems
    D) It does not support role hierarchies

    Answer: A

13. What is the main advantage of Attribute-Based Access Control (ABAC) over Role-Based Access Control (RBAC)?

    A) ABAC is simpler to implement
    B) ABAC provides finer-grained access control
    C) ABAC does not require attributes
    D) ABAC is based on security labels

    Answer: B

14. In the context of access control, what is a credential?

    A) A set of rules governing access
    B) An object that binds an identity to a token
    C) A type of access right
    D) A security label

    Answer: B

15. Which of the following is a key component of Identity, Credential, and Access Management (ICAM)?

    A) Access control lists
    B) Role hierarchies
    C) Credential management
    D) Protection domains

    Answer: C