Database Administration Security

Cybersecurity Department

Course Code: ***CBS 214***

Theoretical Lecture 3: Database Control Frameworks

Halal Abdulrahman Ahmed

Lecture Outlines

- Database Security Control Layers
- Access Control Models
- Authentication & Account Maintenance

Learning Outcomes

By the end of this lecture, students will be able to:

- **Categorize Security Controls:** Differentiate between Structural, Functional, Data, and Procedural layers in a Defense-in-Depth strategy.
- **Evaluate Access Models:** Identify and apply the appropriate access control model (MAC, DAC, RBAC, or RuBAC) for various organizational scenarios.
- **Implement Account Security:** Manage the full life cycle of user credentials, including password policies and lockout procedures.
- **Verify Data Integrity:** Explain how hashing and checksums protect data from unauthorized modification or corruption.

Database Security Control Layers.

They represent a structured framework used to implement a **Defense-in-Depth** strategy. By organizing security measures into these layers, a Database Administrator (DBA) ensures that if one control fails, others are in place to protect the data. It focuses on:

- Structural security
- Functional security
- Data security
- Procedural security

Structural Security

Structural security focuses on the **architecture and environment** where the database resides. It is the "outer shell" and the foundational design of the system.

- **Environment Isolation:** Keeping separate servers for Development, Testing, and Production. This ensures that a bug or a security hole in a test environment cannot be used to access live production data.
- **Schema Hardening:** Designing the database objects to be inherently secure. This includes removing default sample databases (like "Northwind" or "AdventureWorks") and disabling unused services.

Structural Security (cont.)

- **The Use of Views:** Instead of giving users access to raw tables, DBAs create **Views**. A View can hide sensitive columns (like salary or passwords) while still allowing the user to see the information they need.
- **Network Architecture:** Implementing firewalls and **Demilitarized Zones (DMZ)** to ensure that the database server is never directly exposed to the public internet.

Functional Security

Functional security manages **how the database operates** and how users interact with its features. It defines the rules of engagement.

- **Access Control Models: * Role-Based Access Control (RBAC):** Assigning permissions to roles (e.g., "HR_Manager") rather than individual people.
    - **Least Privilege:** Ensuring users have the absolute minimum permissions required to perform their tasks.
- **I-A-A-A Implementation:** Ensuring the database follows the cycle of Identification, Authentication, Authorization, and Accounting (Auditing).

Functional Security (cont.)

- **Integrity Constraints:** Using SQL constraints like PRIMARY KEY, FOREIGN KEY, and CHECK. While often viewed as data tools, they are security features because they prevent "Data Corruption" and unauthorized logic changes.
- **Stored Procedures:** Using procedures to execute commands so that users never have "Direct Table Access."

Data Security

Data security is the layer that protects the **actual information** stored within the rows and columns. This is the "heart" of the system.

- **Encryption at Rest (DAR):** Using tools like Transparent Data Encryption (TDE) to encrypt the physical database files on the hard drive. If a disk is stolen, the data remains unreadable.
- **Encryption in Transit (DIT):** Using SSL/TLS protocols to protect data as it moves across the network between the client and the server.

Data Security (cont.)

- **Data Masking:** Obscuring sensitive data in real-time. For example, showing only the last four digits of a credit card number to a call center agent.
- **Backups and Redundancy:** Ensuring that data is backed up daily and stored in a secure, off-site location to maintain **Availability** in case of an attack.

Procedural Security

Procedural security is the **administrative and human layer**. It involves the policies, documents, and rules that govern human behavior regarding the database.

- **Separation of Duties:** Ensuring that no single person has enough power to compromise the entire system. For example, the person who requests a data change should not be the one who approves and executes it.
- **Change Management:** Requiring a formal process (documentation and approval) before any script or configuration change is applied to a production database.

Procedural Security (cont.)

- **Auditing and Monitoring:** Reviewing log files to see who accessed what data and when. This provides the "Accounting" part of the security model.
- **Incident Response Planning:** Having a written "Playbook" on what to do if a breach is detected, including who to call and how to isolate the database.

Access Control of Data

Access Control in Database Security

- Access control is the process of granting or denying specific requests to obtain and use information. It is one of the most important mechanisms in database security. Its main purpose is to ensure that only authorized users can access, modify, or manage data. Without proper access control, sensitive data can easily be exposed, modified, or destroyed.

Every access decision answers three questions:

- Who is requesting access?
- What resource are they trying to access?
- What actions are allowed?

Access Control Models

Different organizations apply access control in different ways. Each model defines how permissions are assigned and controlled.

The four major models are:

- Mandatory Access Control (MAC)
- Discretionary Access Control (DAC)
- Role-Based Access Control (RBAC)
- Rule-Based Access Control (RuBAC)

Access Control Models (cont.)

- **Mandatory Access Control (MAC):** Decisions are based on security clearances. A central authority (like the military) assigns a clearance level to a user (Subject) and a sensitivity label to data (Object). The user can only access data that matches their clearance.
- **Discretionary Access Control (DAC):** The "owner" of the data decides who can access it. For example, if you create a table in SQL, you can use the GRANT command to give a colleague permission to read it.

Access Control Models (cont.)

- **Role-Based Access Control (RBAC):** Access is based on a person’s job function (Role) rather than their identity. Permissions are assigned to the "Manager" or "Student" role, and users are added to those roles.
- **Rule-Based Access Control (RuBAC):** Access is based on specific conditions or "rules," such as the time of day, the user's IP address, or the location of the request.

Authentication and Credential Management: Password Security

Passwords, Logins, and Maintenance

- **The Weakest Link:** Passwords are the most common but also the weakest form of authentication because they can be guessed or stolen. Password policy maintenance refers to the set of rules and ongoing practices used to ensure that passwords remain secure throughout their lifecycle.

Passwords, Logins, and Maintenance

Best Practice: Do not store passwords in plaintext. Databases must store them as **Hashes**, which are one-way mathematical representations of the password.

"'Woof-woof'? That's your idea of a secure password?"

Passwords, Logins, and Maintenance

Password Policy Maintenance:

- **Complexity:** Passwords should include special characters (@, #, $), numbers, and mixed-case letters. Require users to create passwords with a minimum number of characters (e.g., 8-12 characters or more). Longer passwords increase resistance to brute-force attacks.
- **Password Expiration (Maximum Age):** Passwords must be changed regularly (e.g., every 90 days).
- **History:** The system should prevent users from reusing their last 5 or 10 passwords.

Locking, Unlocking, and Resetting

Administrative procedures to manage user accounts and prevent unauthorized access attempts.

- **Account Locking:** If a user enters the wrong password multiple times (e.g., 3 or 5 attempts), the system should automatically **Lock** the account. This prevents "Brute Force" attacks where a hacker tries thousands of combinations.
- **Unlocking:** Only an authorized administrator or a secure multi-factor process (like a code sent to a phone) should be able to unlock a frozen account.

Locking, Unlocking, and Resetting (cont.)

Secure Resetting:

- **No Plaintext Resets:** Never allow a website to "show" an old password.
- **Secure Links:** Send a one-time, time-limited link to the user's registered email to allow them to set a new password.

"Grandma, I need to use your phone, what's your password?"

Next Lecture

- “Data Refresh, Backup, and Restore”

Research Task

1. Which access control model is best for a university where permissions are based on being a "Teacher" or "Registrar"?
2. If you change a single letter in a 1GB file, what happens to its SHA-512 Hash?
3. Why is "Password History" an important security policy?

Your recipe and deadline both need attention, enjoy your iftar.

References

- **Danturthi, S.** (2019). *Database and application security: A practitioner's guide*. Apress.


Any
Question