Database Security Guideline

Version 2.0
February 1, 2009
Database Security Consortium
Security Guideline WG

Table of Contents

Chapter 1 Introduction

1.1 Objective

In recent years, security incidents involving information leaks have occurred more frequently in society. Much of this critical information is stored in databases, which adds to the importance of implementing database security controls. Thus there is a need for a technical and procedural standard for the protection of database systems, which lies at the heart of information systems. Such a standard shall serve as a guide to the setting up and operation of; systems that provide and maintain a safe and secure environment. Said standard will eventually help in the establishment of an advanced information and telecommunications network society.

In light of the need for security measures that encompass the broad fields of database and security, a guideline that defines the policies and requirements of database security, has been lacking in Japan.

The objective of this Guideline, which describes the necessity and effectiveness of various database security controls, is to provide a set of guidelines for corporate entities and other organizations to use when implementing said controls.

1.2 Prerequisites of this Guideline

Take into account the following prerequisites when using this Guideline to consider what database security controls to implement.

The security controls described in this Guideline are limited to database controls. Users of this Guideline should refer to other established guidelines for information regarding networking and other security controls.

This Guideline does not describe risk assessment, merely its necessity in considering database security controls to implement. Users of this Guideline should refer to other guidelines for information regarding risk assessment.

The database described in this guideline refers to relational database, the most commonly used database type today.

Certain security controls that must be implemented in order that database security is effectively maintained, such as application authentication, are deemed prerequisite controls that are beyond the scope of this Guideline.

This Guideline has been drafted for use by database administrators and designers.

1.3 Notice

Before using this Guideline, read the following notice.

-Copyright

The copyright of this Guideline belongs exclusively to the Database Security Consortium (DBSC).

-Restrictions on Use

This Guideline may not be sold for commercial purposes. Otherwise, there is no restriction to providing any service that is based on the contents of this Guideline.

-Reference Citation

When referring to parts or the entire Guideline, always include the citation “Database Security Guideline,” regardless of whether the use is for commercial or non-commercial purposes.

1) When referring to parts or the entire Guideline:

Source: “Database Security Guideline (Version 2.0)”
Database Security Consortium (DBSC)
http://www.db-security.org/

2) When parts of the Guideline had been modified for use:

Reference Material: “Database Security Guideline (Version 2.0)”
Database Security Consortium (DBSC)
http://www.db-security.org/

-Disclaimer

DBSC shall not be responsible nor shall it be held liable for any financial loss or damages resulting from the use of this Guideline.

-Publicizing

When using this Guideline for publicizing in the news or other media, contact the DBSC secretariat at info@db-security.org.

1.4 Revising this Guideline

Requirements for security controls (i.e. mandatory or recommended) based upon the importance of information assets has been reviewed and revised. See the following annex for the results. (*1)

Annex 1: “Table of Information Asset Value and Security Control Level”

The content of the Guideline has been mapped to the requirements of major security standards. See the following annex for the results. (*2)

Annex 2: “Table of Database Security Guideline and Security Requirements of Major Security Standards”

*1 Security control requirements “mandatory” and “recommended” are defined as follows:

- Mandatory: a serious security problem shall arise in the database system if the control is not implemented
- Recommended: implementation of the control shall be determined after an assessment is performed and the control is deemed necessary

*2 The controls described in this Guideline were matched to corresponding controls described in FISC Security Guidelines, System Management Guidelines, Standards for Information Security Measures for the Central Government Computer Systems, and ISO/IEC 27001. As for PCI DSS, matching was done at section level, and then at the item level if matched section is in the PCI DSS guideline.

Chapter 2 Database Security within the Context of Information Security

2.1 System Model

This Guideline assumes a three-tier system model that is accessed from both the Internet and the Intranet.

In this model, the network is divided into segments. The database server cannot be accessed directly from the Internet or Intranet. The database server can only be accessed directly from the “Operations/Management Zone” (Administrator LAN) via a firewall.

This system model is shown below in Figure 1.

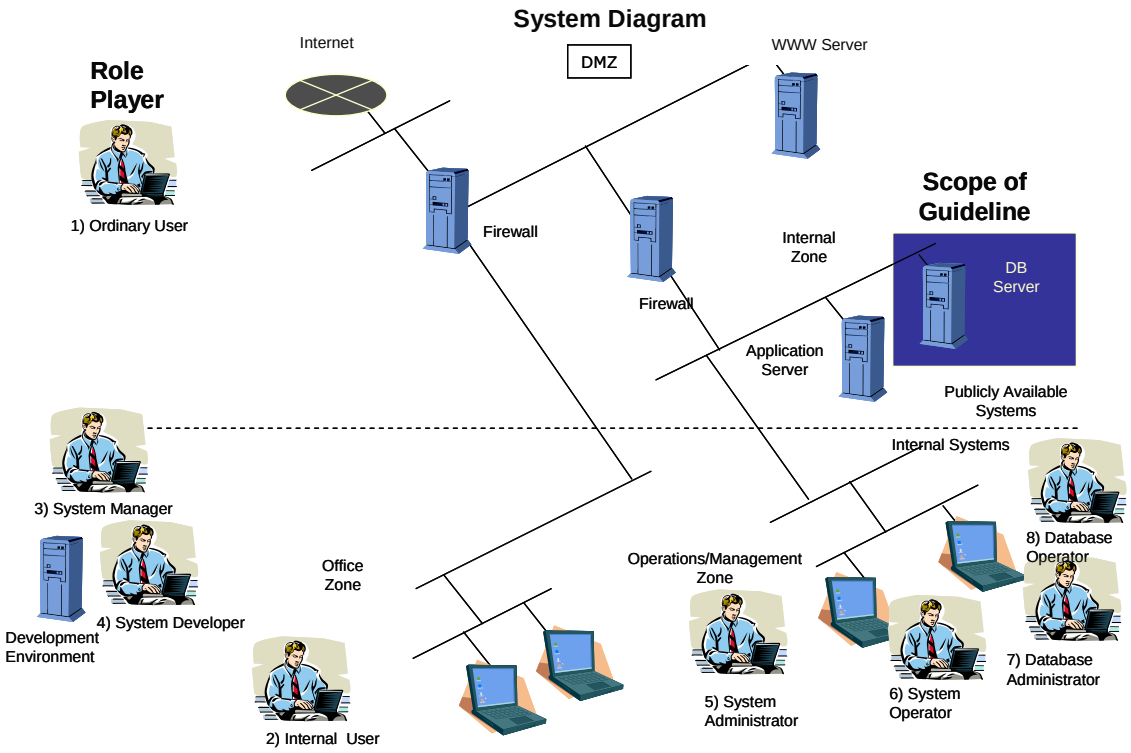

Figure 1. System Model

Note: AP refers to application (the same hereinafter)

2.2 Summary of Security Controls

The summary of the security controls applicable to the system model is shown below.

The security controls can be divided into the following categories:

1. Network controls
2. Web controls
3. Application auditing
4. Database controls
5. Terminal controls

This guideline shall focus on "4. Database Controls" and the details shall be described in Chapter 4 "Database Security Controls."

The applicable security controls for the system model is shown below in Figure 2.

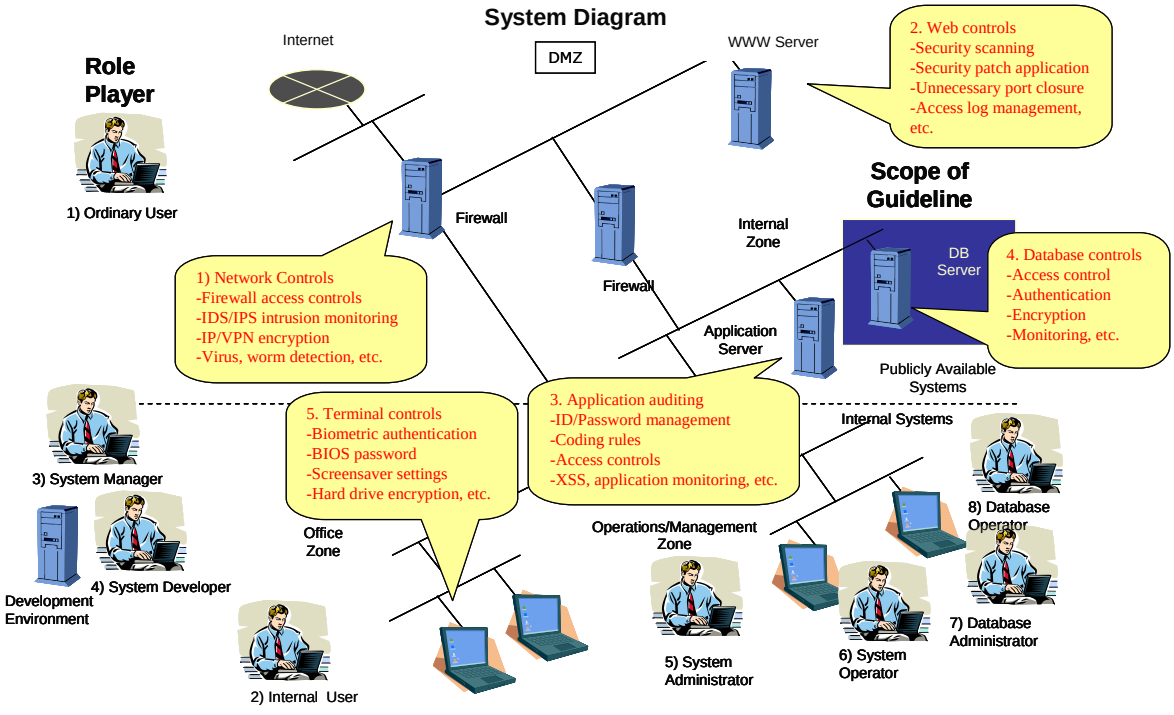

Figure 2. Summary of Security Controls

2.3 Definition and Scope of Database Security

In this Guideline, database security is defined as protecting the information stored in the database. The database server and network around this server comprises the scope of database security.

The definition and scope of database security is shown below in Figure 3.

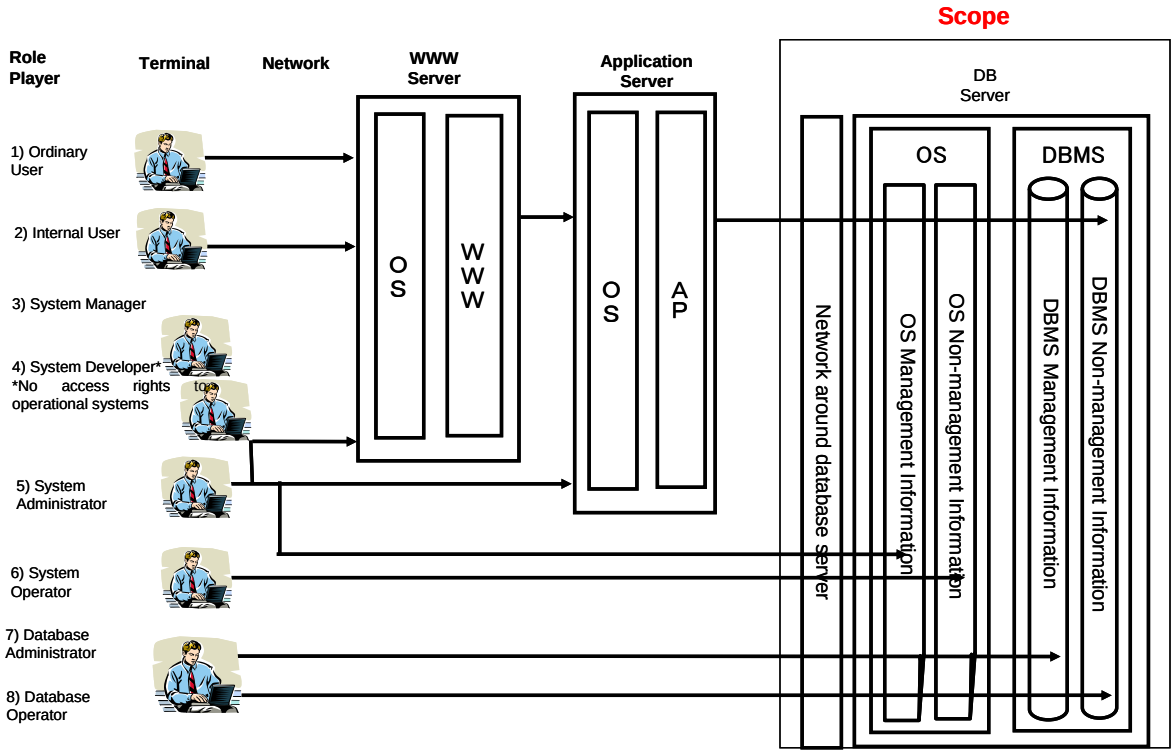

Figure 3. Definition and Scope of Database Security

Note: The arrows represent authorized access

2.4 Definition of Elements Relating to Database Security

2.4.1 Defining Threat

In considering database security, this Guideline takes into account the identification of applicable threats and controls necessary to prevent their occurrence. A threat is defined as any event that may compromise the confidentiality, integrity, and/or availability of information assets (defined in section 2.4.3). Threat consists of a combination of the following four elements: role player that is the source of threat, the information assets that must be protected from the threat, the method of unauthorized access, and the resulting unauthorized action.

(Example: Illegal access of database management information by ordinary users exploiting the vulnerability of the database.)

Threats are defined as follows:

- Threats compromise the confidentiality, integrity, and availability of information assets.
- Threats considered here consist of technical threats related to database access, not physical ones, such as damage by fire, etc.

2.4.2 Defining Role Players

In this guideline, the persons involved in database operations, their respective roles, and the role player who may become the source of threat should be defined. Once this is considered, the identification of role players and the potential threat may be performed. A definition of role players is shown below in Table 1.

Table 1. Definition of Role Players

Role Players	Definition and Roles
1) Ordinary User	End user who does not belong to the organization
2) Internal User	End user who belongs to the organization
3) System Manager	Manages and authorizes activities of 4) through 8)
4) System Developer	-WWW and application development -Building and configuring WWW and application server (including OS and middleware) -Building and configuring the network around the database server -Building and configuring database server (including OS and middleware)
5) System Administrator	-Operating and maintaining WWW and application server (including OS and middleware) -Database server operation (including OS and middleware) -Network (around the database server) equipment operation
6) System Operator	-WWW and application operation -Network (around the database server) operation
7) Database Administrator	-Building and configuring DBMS

	-DBMS operation
8) Database Operator	-Business operations (including tables, data, etc.)

2.4.3 Defining Information Assets Related to Database

In this Guideline, information assets that must be protected are defined as information stored in a database server. Information, as stored in a database server is described below.

1. DBMS management information
   - -Database configuration information (dictionary, user ID/password, etc.)
   - -Database log (access log, etc.)
2. DBMS non-management information
   - -Business data
   - -Business application (stored procedure, etc.)
3. OS administrator information
   - -OS configuration information
   - -OS log (trace log, alert log, etc.)
4. OS information
   - -Database related files (definition file, physical file, log, backup, etc.)

2.4.4 Defining Information Asset Value

In this Guideline, information assets are valued according to the impact incurred upon the loss of confidentiality and/or unauthorized use, as shown below.

Information Asset	Value of Information Asset		
	High	Medium	Low
Personal Information	External personal information (personal information that is gathered from outside the organization)	Internal personal information (personal information that is gathered from inside the organization)	No personal information contained

Company Information	Confidential information (Information whose access is restricted to authorized personnel only)	Internal Use Only (Information whose access is restricted to company personnel only)	Publicly Available Information

Note: Mandatory and Recommended indications for each control described in this Guideline are for assets with “Medium” value. The indications for “High” and “Low” assets are mapped in “Annex 1: Table of Information Asset Value and Security Control Level.”

2.4.5 Defining Means

Means (method) is the way that unauthorized activity is perpetrated by the role player upon information assets. Knowing the means will lead to an understanding of effective controls to mitigate the threat.

Below is a list of some potential means.

- -packet eavesdropping
- -password dictionary attack
- -stealing ID/password via social engineering
- -unauthorized access of DBMS information by exploiting errors in settings
- -unauthorized access of DBMS information by exploiting DBMS vulnerability
- -unauthorized access of DBMS information by modifying database files
- -Illegal access to ID/password by using DB management information
- -creating backdoors
- -unauthorized access of DBMS information by creating unauthorized DBMS administrator or operator account
- -unauthorized removal of information through an unauthorized route
- -unauthorized access of DBMS information by modifying management information
- -SQL execution with the intent of disrupting service
- -unauthorized removal of information through an authorized route

2.4.6 Defining Unauthorized Action

Unauthorized action refers to the illegal activity performed by the role player on the information assets.

The means described in Section 2.4.5 correspond to the means in which the unauthorized action is perpetrated.

Below is a list of predictable unauthorized actions.

-Unauthorized use of information (by removal)

-Unauthorized modification or destruction of information

-Service disruption (by resource exhaustion)

-Unauthorized modification or destruction of DBMS information

-Unauthorized access of DBMS information

Note: This Guideline distinguishes between “Unauthorized Action” and “Unauthorized Access.” A definition of unauthorized access is given in Chapter 3 “Writing a Security Policy.”

2.4.7 List of Threats

A threat consists of role players, information assets, means, and unauthorized action.

Below is a list of threats.

Table 2. Threat Table

Who? (Role Players)	**What? (Information Assets)**	**How? (Means)**	**Action? (Unauthorized Action)**	**Authorized or Unauthorized Access**
1) Ordinary User 2) Internal User 3) System Manager 4) System Developer 5) System	DBMS Management Information	packet eavesdropping	-Unauthorized modification or destruction of DBMS information -Unauthorized access of DBMS information	Unauthorized
		password dictionary attack		
		stealing ID/password via social engineering		
		unauthorized access of DBMS information by exploiting errors in settings		

Administrator 6) System Operator		unauthorized access of DBMS information by exploiting DBMS vulnerability		
		unauthorized removal of information through an unauthorized route		
1) Ordinary User	DBMS Non-management Information	unauthorized removal of information through an authorized route	-Unauthorized use of information (by removal)	Authorized
		SQL execution with the intent of disrupting service	-Service disruption (by resource exhaustion)	
2) Internal User	DBMS Non-management Information	unauthorized removal of information through an authorized route	-Unauthorized use of information (by removal)	Authorized
		SQL execution with the intent of disrupting service	-Service disruption (by resource exhaustion)	
1) Ordinary User 2) Internal User	DBMS Non-management Information	packet eavesdropping	-Unauthorized modification or destruction of DBMS information	Unauthorized
		password dictionary attack		
		stealing ID/password via social engineering		
		unauthorized access of DBMS information by exploiting errors in settings		
		unauthorized access of DBMS information by exploiting DBMS vulnerability		

		unauthorized removal of information through an unauthorized route		
3) System Manager 4) System Developer 5) System Administrator 6) System Operator	DBMS Non-management Information	packet eavesdropping	-Unauthorized modification or destruction of DBMS information -Unauthorized access of DBMS information	Unauthorized
		password dictionary attack		
		stealing ID/password via social engineering		
		unauthorized access of DBMS information by exploiting errors in settings		
		unauthorized access of DBMS information by exploiting DBMS vulnerability		
		unauthorized removal of information through an unauthorized route		
4) System Developer	DBMS Management Information	creating backdoors	-Unauthorized modification or destruction of DBMS information -Unauthorized access of DBMS information	Unauthorized
	DBMS Non-management Information	creating backdoors	-Unauthorized modification or destruction of DBMS information -Unauthorized access of DBMS information	

3) System Manager 5) System Administrator	DBMS Management Information	unauthorized access of DBMS information by creating unauthorized DBMS administrator or operator account	-Unauthorized modification or destruction of DBMS information -Unauthorized access of DBMS information	Unauthorized
	DBMS Non-management Information	unauthorized access of DBMS information by creating unauthorized DBMS administrator or operator account	-Unauthorized modification or destruction of DBMS information -Unauthorized access of DBMS information	
3) System Manager 6) System Operator	DBMS Management Information	unauthorized access of DBMS information by modifying database files	-Unauthorized modification or destruction of DBMS information -Unauthorized access of DBMS information	Unauthorized
	DBMS Non-management Information	unauthorized access of DBMS information by modifying database files	-Unauthorized modification or destruction of DBMS information -Unauthorized access of DBMS information	
7) Database Administrator	DBMS Management Information	unauthorized removal of information through an authorized route	-Unauthorized use of information (by removal)	Authorized
		using management information to access ID/password	-Unauthorized modification or destruction of information	
		unauthorized access of DBMS information by modifying management information		

		SQL execution with the intent of disrupting service	-Service disruption (by resource exhaustion)	
	DBMS Non-management Information	packet eavesdropping	-Unauthorized modification or destruction of DBMS information -Unauthorized access of DBMS information	Unauthorized
		unauthorized removal of information through an unauthorized route		
8) Database Operator	DBMS Management Information	packet eavesdropping	-Unauthorized modification or destruction of DBMS information -Unauthorized access of DBMS information	Unauthorized
		password dictionary attack		
		stealing ID/password via social engineering		
		unauthorized access of DBMS information by exploiting errors in settings		
		unauthorized access of DBMS information by exploiting DBMS vulnerability		
		unauthorized removal of information through an unauthorized route		
	DBMS Non-management Information	unauthorized removal of information after obtaining from an authorized route	-Unauthorized use of information (by removal)	Authorized
		SQL execution with the intent of disrupting service	-Service disruption (by resource exhaustion)	

Chapter 3 Writing a Security Policy

The basis of information security consists of “who will be given access to what and in what way.” As long as this is not clarified, it will be impossible to distinguish between authorized and unauthorized access, and effective security will be difficult to implement.

In this Chapter, we shall describe the prerequisite concepts to implementing effective database security. These concepts shall be the basis for implementing the controls listed in “Chapter 4 Database Security Controls.”

3.1 Writing a Database Security Policy

In order to implement database security controls, the roles of each user and the information assets they are given access to must be determined beforehand. Determining the users and the information assets they must access will lead to an understanding of the controls that must be implemented, as well as functionalities that must be considered in order that these controls may be effective.

Here, we shall present the policies that must be established as a prerequisite to implementing technical security controls such as access controls and audits.

3.1.1 Defining Important Information

In order to apply security controls effectively, the following controls must be implemented.

-Identifying information that must be protected. (Mandatory)

Example:

-DBMS management information, DBMS non-management information

-Classifying information according to their importance. (Mandatory)

Example:

-Personal information, confidential information

3.1.2 Risk Assessment

In order to apply security controls effectively, the following controls must be implemented.

-Identify threats and perform risk assessment. (Mandatory)

-Implement necessary database security controls based upon the importance of assets and the results of risk assessment. (Mandatory)

3.1.3 Account Management Policy

(1) Defining User Roles

In order to apply security controls effectively, the following controls must be implemented.

-Define the roles applicable to the database users. (Mandatory)

-Determine the role of each user based upon the roles defined above. (Mandatory)

(2) Defining User Accounts

In order to apply security controls effectively, the following controls must be implemented.

-Create accounts and allocate access rights based upon users and users' roles (Mandatory)

Example:

- Ordinary User Account (Ordinary Account)
  -Application user account (create for each application)
  -Allocate necessary access rights
- System Administrator (No account)
- System Operator (No account)
- Database Administrator (Database Administrator Account)
  -Create for each database administrator a database administrator account
  -Allocate necessary access rights to DBMS management information
- Database Operator (Database Administrator Account)
  - Create for each database operator a database operator

account
-Allocate necessary access rights to DBMS (non-management) information

(3) Reviewing Account Management Policy

In order to apply security controls effectively, the following controls must be implemented.

-The defined accounts and allocated access rights must be reviewed periodically to check they are still appropriate. (Mandatory)

-The defined accounts and allocated access rights must be reviewed after changes to the system or operation. (Mandatory)

-The defined accounts and allocated access rights must be reviewed whenever inappropriately allocated access rights and /or accounts are found. (Mandatory)

3.1.4 Logging Policy

(1) Defining Logging Objectives

In order that logging activities are effective, the following controls must be implemented.

-The objective of recording logs must be defined. (Mandatory)

Example:

- To obtain forensic evidence in the event of unauthorized access, or to submit as evidence to relevant authorities

(2) Determining Obtainable Audit Logs

In order to obtain appropriate logs, the following controls must be implemented.

-The kind of logs that can be obtained for each system must be defined. (Mandatory)

Example:

- Operating system logs, software logs
- Application logs
- Database audit logs

(3) Determining What Access Must be Logged

In order to determine what access must be logged, the following controls must be implemented.

-The type of access related to important information must be determined. (Mandatory)

Example:

- Access to important information, such as personal information and confidential information
- Access to important information, such as database information
- Access after regular business hours
- Logins
- Other forms of access, such as via SQL

-The conditions in which access may be considered unauthorized must be determined. (Mandatory)

Example:

- Large volumes of database queries
- Access from an unauthorized location
- Access after regular business hours

-All database access must be logged. (Recommended)

(4) Determining What Information Must be Included

In order that logs obtained can be used effectively, the following controls must be implemented.

-Determine the information that must be recorded in the log. (Mandatory)

Example:

- Date and time (When)
- Database/Application account ID (Who)
- Object ID, table name (What)
- Host (PC) name and IP address (Where)

- SQL type and query body/sentence (How)
- Success or failure of access attempt (Result)

(5) Log Retention

In order to ensure the validity of access logs, the following controls must be implemented.

-The retention policy of access logs must be defined. (Mandatory)

Example:

- Storage location
- Storage media
- Retention time
- Access controls to be implemented

3.2 Personnel Controls

Implementing technical controls to a system does not always guarantee that the system is protected against information security threats. This is because it is people who use, operate, and maintain technical controls. In order that system security controls are effectively utilized, personnel controls must be implemented. This is especially true because a database tends to be accessed by a considerable number of internal users, many of whom are allocated administrator rights.

Here, we shall present personnel controls, such as writing security rules and conducting personnel training.

3.2.1 Rules and Training

(1) Drafting Rules

In order to determine the basis for setting the security level of the system, the following controls must be implemented.

-Draft security rules for the system. (Mandatory)

Example:

- Require users to sign written agreements (mandating compliance with security rules)
- Prohibit users from obtaining database information without

authorization
- Prohibit users from storing database information into anything other than authorized media
- Prohibit users from writing down ID/passwords and leaving them visible to the public/others
- Prohibit anyone from becoming both database administrator and system operator

-Draft disciplinary rules for violators. (Mandatory)

Example:
- Set down disciplinary action in company rules
- Set down fines for violators

(2) Conducting Training

In order to raise security awareness and prevent information leaks and human error leading to unauthorized access, the following controls must be implemented.

-Conduct training for all employees and partners. (Mandatory)

Example:
- Communicating information security policy
- Setting down training schedules
- Preparing training materials

-Follow-up on these trainings and obtain feedback. (Recommended)

Example:
- Regularly evaluate the effectiveness of these trainings

(3) Checking Database Management Activities

In order to prevent human error and unauthorized access by system and database administrators, the following controls must be implemented.

-Monitor for the occurrence of security incidents involving the latest database and technical vulnerability bulletins. (Mandatory)

-Administrative activities must be authorized beforehand. (Mandatory)
-Administrative activities must be logged. (Mandatory)
-Activities involving the use of administrator accounts must involve two or more individuals. (Recommended)
-Divide a password between two system administrators to access a database. (Recommended)
  Example:
    - Dividing password
-Rotate system administrators. (Recommended)

Chapter 4 Database Security Controls

In this Chapter, we shall describe implementing the actual security controls conceptualized in Chapter 3 “Writing a Security Policy.”

Database security controls can be classified into “preventive,” “detective,” and “forensic” controls against unauthorized access. Database security also involves considering performance, budget, and other constraints before selection of controls can be made. “Mandatory” or “Recommended” is indicated for each security control in order to facilitate selection.

4.1 Preventive Security Controls

Recently, systems are rapidly becoming more advanced and complex. For this reason, security risks to database systems continue to grow, necessitating the implementation of preventive controls. In this Section, we shall describe security controls that prevent unauthorized access efficiently, and at the same time minimizing the impact to authorized access.

4.1.1 Initial Configuration

DBMS products that have not been updated, or running with unnecessary functionalities may possess multiple vulnerabilities. Furthermore, use of default ports or disabling passwords for network access increases the possibility of unauthorized access.

Here, we shall describe initial configuration settings that are designed to reduce vulnerabilities and minimize the risk of unauthorized access.

4.1.1.1 Installation

(1) Using the Latest Version

In order that the latest security controls are available, the following controls must be implemented.

-Install the latest security patches. (Mandatory)

-Determine and use the latest version of DBMS that are available. (Recommended)

(2) Installing only the Minimum Functionality

In order to prevent unauthorized use, as well as wasteful use of resources, the following controls must be implemented.

-Select and install only the required functionalities. (Mandatory)

-Functionalities that are not needed but installed as defaults must be deleted or disabled. (Mandatory)

(3) Changing Port Settings

In order to prevent unauthorized use, the following controls must be implemented.

-Change port numbers that are widely used or set by default during installation. (Mandatory)

(4) Restrict Network Access

In order to prevent unauthorized use of networking functionalities, the following controls must be implemented.

-Restrict access to DBMS's network access functions. (Mandatory)

Example:

- Set password to Oracle listener

4.1.2 Authentication

Current database authentication systems use password authentication during logins. To prevent unauthorized users from using authorized accounts to leak or modify information, account information must be tightly managed. Furthermore, as DBMS administrator accounts enable users to do everything, their account information must be managed more tightly than ordinary user accounts.

Here, we shall describe account management controls needed to maintain a high level of security.

4.1.2.1 Account Management

(1) Creating Necessary Accounts

In order to prevent impersonation by unauthorized use of account information, the following controls must be implemented.

- -Create only necessary accounts. (Mandatory)
- -Set access rights for each account. (Mandatory)
- -Create separate user and administrator accounts with corresponding access rights. (Mandatory)

(2) Deleting Unnecessary Accounts

In order to prevent impersonation by unauthorized use of account information, the following controls must be implemented.

- -Delete accounts that are not in use. (Mandatory)

  Example:

  - Deleting accounts of employees who have been terminated or have been transferred
- -Delete default accounts that are not used. (Mandatory)

(3) Locking Unused User Accounts

In order to prevent impersonation by unauthorized use of account information, the following controls must be implemented.

- -Lock user accounts that have not been used for a specified period of time. (Mandatory)

  Example:

  - Locking user accounts that have been used only once in a year
- -Confirm the number of times an account has been used regularly. (Mandatory)

(4) Locking Account due to Failed Login

In order to prevent impersonation by unauthorized use of account information, the following controls must be implemented.

- -Set the number of times a login attempt may fail before locking the account. (Recommended)

(5) Database Administrator Account Management

In order to minimize operational errors and unauthorized use by the database administrator, the following controls must be implemented.

-Database administrator accounts shall only be allocated to necessary personnel. (Mandatory)

-Database administrator accounts shall not be used for activities that do not require administrator rights. (Mandatory)

-A unique database administrator account shall be allocated to each administrator. (Mandatory)

-Database administrator accounts shall only be used on specific (PC) terminals. (Recommended)

(6) ID/Password of the Development and Operational System

In order to prevent unauthorized use of the development system accounts, the following controls must be implemented.

-Even if the same account ID is used for a development and an operational system, use a different password for each account. (Mandatory)

-Use a different ID/Password for a given user of a development and operational system account. (Recommended)

(7) Temporary Accounts

In order to prevent unauthorized use by temporary users, the following controls must be implemented.

-Either provide them with a public/anonymous ID whose password is changed every time it is allocated or provide them with a temporary account. (Mandatory)

4.1.2.2 Password Management

(1) Password Complexity

In order to prevent passwords from being guessed, the following controls must be implemented.

-Do not allow the use of easily guessable passwords. (Mandatory)

Example:

- When ID and password are the same
- Easily guessed passwords, such as 1234
- Default passwords that are set automatically during installation

(2) Periodically Changing Password

In the event passwords are leaked, and/or in order to prevent passwords from being used for unauthorized access, the following controls must be implemented.

-Periodically change the database administrator password. (Mandatory)

-Always change initial password. (Mandatory)

-Prevent passwords from being recycled. (Recommended)

-Periodically change user password. (Recommended)

(3) Setting Password Expiration Date

In order to ensure that users and administrators periodically change passwords, the following controls must be implemented.

-Set expiration date for database administrator password. (Mandatory)

-Set expiration date for user password. (Recommended)

4.1.3 Access Control

Even if a database server is secured against external threats, if access controls are not properly implemented, the server becomes vulnerable to internal threats, such as disgruntled personnel who leak information after obtaining authorized personnel account information. To be enforceable, access controls must be based upon firm rules.

Here, we shall describe security controls needed to allocate appropriate access rights to users.

4.1.3.1 Setting Access Rights

(1) Determining Database Access Requirements

In order that access controls are allocated appropriately, the following controls must be implemented.

-Classify accounts based upon their purposes and uses. (Mandatory)

Example:

- For database administrators, object administrators, data access, etc.

-Define the access rights needed for each account classification. (Mandatory)

Example:

- Functionality, access connection path, object access, etc.

-Divide accounts according to access rights. (Mandatory)

-For each of these divided accounts, determine the minimum range of data needed to be accessed and the minimum rights (read, write, create, delete) that need to be set and determine the database access requirements. (Mandatory)

(2) Setting Access Rights

In order to restrict unnecessary data access, the following controls must be implemented.

-Allocate the minimum access rights needed for each account defined in the database access requirements. (Mandatory)

-Administrator rights shall be allocated to only a limited number of accounts. (Mandatory)

(3) Determining Database Access Requirements and Pointers for Setting Access Rights

Keep in mind the following pointers when determining database access requirements.

-Never create user accounts that give access rights to all users unconditionally. (Mandatory)

-Never create user accounts that allow ordinary users to allocate access rights to ordinary users. (Mandatory)

-Lock down and disallow direct access using object owner accounts for purposes other than creating, deleting, and maintaining objects. (Recommended)

(4) Reviewing User Accounts

In order that changes to system access requirements are applied to access right settings, the following controls must be implemented.

-Periodically check to ensure that unnecessary access rights are not set. (Mandatory)

-When system changes have been applied, check to ensure that unnecessary access rights have not been set. (Mandatory)

-When unnecessary access rights are found, make the necessary changes. (Mandatory)

4.1.4 Encryption

Information is compromised more often by internal causes, such as theft by internal personnel or loss of laptops being used outside office premises, rather than by external attacks. Important data must be protected from compromise.

Here, we shall present encryption controls needed to protect database data from external as well as internal threats.

4.1.4.1 Encrypting Transmission

In order to protect transmissions between database servers and clients from eavesdropping, the following controls must be implemented.

-Encrypt transmissions using encryption functions or tools. (Recommended)

4.1.4.2 Encrypting Data

(1) Encryption Controls for DBMS Data

In order that data stored in DBMS is protected against theft, the following controls must be implemented.

-Encrypt stored data using encryption functions or tools. (Recommended)

(2) Encryption of Physical Files

In order that physical files are protected against theft, the following controls must be implemented.

-Encrypt physical files using encryption functions or tools. (Recommended)

(3) Encryption of Backup Data

In order that backup data is protected against theft, the following controls must be implemented.

-Encrypt backup data using encryption functions or tools. (Recommended)

4.1.4.3 Encrypting Procedures

In order that procedures are protected against theft, the following controls must be implemented.

-Encrypt procedures using encryption functions or tools. (Recommended)

4.1.4.4 Managing Encryption Keys

In order that encrypted data is protected against unauthorized use, the following controls must be implemented.

-Implement an encryption key management scheme. (Recommended)

4.1.5 Restricting Removable Media Use

In order to prevent information leaks, connection of removable media devices to database servers, such as floppy disks, CD-R/RW, DVD-R/RW/ROM, USB memory, etc. must be restricted. Also, printer connections and use must also be appropriately restricted.

Here, we shall present controls applied to removable media and terminals connected to database servers in order to prevent information leaks arising from bringing data out of company premises.

4.1.5.1 Restricting Removable Media Connections to Database Server

(1) Restricting Removable Media Connections

In order to ensure that removable media are not connected to database servers, the following controls must be implemented.

- -All unnecessary removable media must be removed. (Mandatory)
- -All unnecessary printers must be disconnected. (Mandatory)

(2) Restricting Removable Media Use

In order to ensure that confidential information is not transferred to removable media, the following controls must be implemented.

- -Access to removable media must be controlled. (Mandatory)
- -Connection to printers must be controlled. (Mandatory)
- -Connection to removable media must be controlled. (Recommended)

(3) Logging Connections to Removable Media

In order to provide deterrence against information leaks and to obtain forensic data in the event of an accident, the following controls must be implemented.

- -Connection to removable media must be logged. (Recommended)
- -Access to removable media must be logged. (Recommended)
- -User access to removable media must be logged. (Recommended)

4.1.5.2 Restricting Database Server Access from a Terminal

In order to prevent information from being transferred from terminals (PCs), the following controls must be implemented.

- -Access to removable media (from terminals) must be controlled.

(Mandatory)

- Database server connection terminals (PCs) must be hardened before access is granted. (Mandatory)
- Terminal (PC) users must be authenticated before access is granted. (Mandatory)
- Software installed on terminals (PCs) and their use must be monitored. (Mandatory)
- Access to printers (from terminals (PCs)) must be controlled. (Mandatory)

4.1.6 Others

We have been presenting preventive controls involving DBMS settings and configurations. DBMS security levels can be increased further by implementing server access and resource controls.

Here, we shall present controls to reduce the possibility of unauthorized access to database server and prevent theft of massive amounts of data.

4.1.6.1 Restricting Terminal Access (by IP Address)

In order to restrict database access, the following controls must be implemented.

- Use a firewall to protect database servers from unauthorized terminal access. (Mandatory)
- Database server access must be restricted to terminals authorized to connect to the server or terminals belonging to the same network segment. (Mandatory)

  Example:

  - IP address filtering using routers
  - Using DBMS network access restriction features

4.1.6.2 Resource Access Restriction on a per User Basis

In order to prevent service interference and theft of massive amounts of data, the following controls must be implemented.

- Prevent CPU time from being excessively used by ordinary users.

(Recommended)

4.1.6.3 Vulnerability Scanning

(1) Vulnerability Scanning prior to the Operational Stage

In order to ensure that security controls have been implemented in compliance with the database security policy, the following controls must be implemented.

-Conduct a vulnerability scan prior to transferring to the operational stage. (Mandatory)

(2) Periodic Vulnerability Scanning

In order to maintain the effectiveness of security controls, the following controls must be implemented.

-Periodically conduct a vulnerability scan, taking into account the latest threats. (Mandatory)

4.1.6.4 Protection Against Backdoor

(1) Hardening Servers

In order to prevent unauthorized access and destruction of database servers, the following controls must be implemented.

-Apply secure OS and network settings on DBMS servers. (Mandatory)

Example:

- Restricting access ports

(2) Protection against Covert Channels

In order to prevent unauthorized access into systems via covert channels introduced into the source code, the following controls must be implemented.

-Assign a person in charge of creating and modifying the source code and also check/test to ensure there are no unauthorized modifications to the code. (Mandatory)

4.1.6.5 Limiting File Access Paths

(1) Limiting Access to Database Configuration Files

In order to prevent destruction of the database, the following controls must be implemented.

-Only administrators can have access to database configuration files. (Mandatory)

-Only administrators can have access to script files. (Mandatory)

-Access rights to database configuration files must be reviewed periodically. (Mandatory)

(2) Limiting Connection Paths

In order to prevent operational errors and unauthorized use of the database, the following controls must be implemented.

-Applications (including management tools) used to access the database must be installed only on the PCs of users that are authorized to access the database. (Mandatory)

-Restrict the network connection paths that may be used to access the database/server. (Recommended)

4.2 Database Detection and Forensic Security Controls

Logs containing the appropriate information must be produced to ensure that they are available for monitoring and forensic analysis to quickly resolve security problems. In this Section, we shall describe controls for effectively detecting and tracing security problems.

4.2.1 Log Management

When DBMS does not produce logs, investigating and tracing information leaks, as well as unauthorized access shall become, at the very least, difficult, and sometimes impossible. Furthermore, in order to retain their validity and reliability, logs must be managed properly.

Here, we shall describe management controls needed to ensure that logs are retained for monitoring and forensic purposes in the event of information leaks and/or unauthorized access.

4.2.1.1 Logging

(1) Obtain Login Logs

In order to monitor logins, the following controls must be

implemented.

-Produce login logs. (Mandatory)

(2) Obtain DBMS Information Access Logs

In order to monitor access to personal, confidential, and other important information, the following controls must be implemented.

-Produce logs relating to access/changes to personal, confidential, and other important information. (Mandatory)

(3) Obtain DBMS Management Information Access Logs

In order to monitor access to DBMS management information, the following controls must be implemented.

-Produce logs relating to access/changes to DBMS management information. (Mandatory)

(4) Obtain Logs to Database Object Changes

In order to monitor changes to database objects, the following controls must be implemented.

-Produce audit logs of database object (e.g. database accounts, tables, views, etc.) creation and changes. (Mandatory)

4.2.1.2 Log Protection

(1) Log Retention

In order that logs can be retrieved when required, the following controls must be implemented.

-Copy logs to external removable media. (Mandatory)

Example:

-Tapes, LTO device, disks, external servers, external database, etc.

Note: 'External' refers to a system or device other than the database system or the third party system in which the log data is stored

-Store the external removable media in a safe location. (Mandatory)

Example:

-Store the external removable media in a lockable place.

-Maintain a written log of all removals of media (from its storage location).

-Apply access controls to the logs. (Mandatory)

Example:

-Logs can only be accessed by the security administrator.

-Logs cannot be modified.

(2) Preventing Log Modification

-Prevent unauthorized modification of logs. (Mandatory)

Example:

- Retaining multiple copies of logs
- Retaining logs in a read-only storage media
- Applying digital signatures using timestamps

(3) Encrypting Logs

-Apply encryption to logs. (Recommended)

Example:

- Encrypting logs

4.2.2 Detecting Unauthorized Access

Even if a database is protected against unauthorized access, there is always the possibility of this being attempted. Therefore, it is necessary to have in place a mechanism to detect such attempts.

Here, we shall describe the controls needed in order to detect unauthorized access attempts.

4.2.2.1 Detective Mechanisms

In order to detect unauthorized access, the following controls must be implemented.

-Provide for a mechanism to notify of an unauthorized access attempt. (Mandatory)

Example:

- Notify, by email, the system manager/administrator/operator that an account lockout has occurred (due to exceeding the

maximum number of failed login attempts)

4.2.2.2 Checking Access Time

(1) Detecting Access to DBMS Management Information

In order to detect suspicious access to DBMS management information during and outside working time, the following controls must be implemented.

-Monitor and detect and log access that is outside authorized time. (Mandatory)

-Check that the work done is as specified by comparing the log and the work application form. (Mandatory)

(2) Detecting Access to Database Information

In order to detect suspicious access to database information during and outside working time, the following controls must be implemented.

-Define for each application user the working time in which he or she is authorized to access the DBMS. (Mandatory)

-Monitor session log and detect any access that is outside the authorized working time. (Mandatory)

4.2.2.3 Detecting Access from Unauthorized Terminals (Using IP Address, etc.)

(1) Detecting Access from Unauthorized Terminals

In order to detect access from unauthorized terminals, the following controls must be implemented.

-Define the terminals that are authorized to access the DBMS. (Mandatory)

-Detect any access from unauthorized terminals. (Mandatory)

(2) Detecting Administrator Account's Access

In order to detect access using an administrator account, the following controls must be implemented.

-Define the terminal/OS account/DB account combination that is authorized to access the DBMS using an administrator account. (Mandatory)

-Detect any administrator access that is not from the authorized combination. (Mandatory)

(3) Detecting Ordinary User Account's Access

In order to detect access using ordinary user accounts, the following controls must be implemented.

- Define the access pattern (the combination of terminal/ OS account / DB account) of ordinary users. (Mandatory)
-Detect any ordinary user's access that is not from the authorized combination. (Mandatory)

4.2.2.4 Checking for Other Unauthorized Access

In order to detect unauthorized access other than those already mentioned the following controls must be implemented.

-Detection of dictionary attacks by checking that failed login attempts are less than a predetermined number for a given period of time. (Recommended)
-Monitor logs and detect SQL execution. (Recommended)
-Monitor logs and detect database object creation and changes. (Recommended)

4.2.2.5 Terminating Unauthorized Access

In order to minimize damage resulting from detected unauthorized access, the following controls must be implemented.

-Provide for a mechanism to terminate or interrupt unauthorized access. (Recommended)

4.2.3 Analyzing Logs

In order to determine that a particular event is a security breach, logs must be analyzed from various angles. Also, previous recorded logs must also be analyzed.

Here, we shall describe controls needed to efficiently detect security breaches from audit logs.

4.2.3.1 Log Analysis Mechanisms

In order to determine the cause of unauthorized access and detect security breaches, the following controls must be implemented.

-Provide for a mechanism to analyze logs. (Mandatory)

Example:

- Installing a log analysis tool

4.2.3.2 Periodic Analysis of Logs

(1) Periodic Analysis of Session Information

In order to detect logins, the following controls must be implemented.

-Analyze session information. (Mandatory)

Example:

- Doing a trend analysis on sessions containing a [high] number of failed login attempts
- Doing a trend analysis on abnormally long login sessions
- Doing a trend analysis on sessions that consume large amounts of resources

(2) Periodic Analysis of Database Access Information

In order to detect database access, the following controls must be implemented.

-Analyze SQL information. (Mandatory)

Example:

- Doing a trend analysis on abnormally long SQL executions
- Doing a trend analysis on SQL commands that consume large amounts of resources