CERT-In Security Guideline CISG-2003-07

CERT-In

Indian Computer Emergency Response Team

Handling Computer Security Incidents

Database Server Security Guidelines

Department of Information Technology
Ministry of Communications and Information Technology
Govt. of India

Issue Date: October 21, 2003

Table of Contents

1. Introduction 3

1.1 Database Vulnerabilities ……………………………….3

1.2 Database Security ……………………………………..3

2. Planning ……………………………………………….4

2.1 Type of Server Required ……………………………….4

2.2 Server Security ………………………………………….4

2.3 Database Connections …………………………………..4

2.4 Table Access Control ……………………………………4

2.5 Physical Location of Server ……………………………..4

2.6 Separate Storage Area …………………………………..4

2.7 Identify Users & Their Needs ……………………………5

2.8 Security Policy …………………………………………..5

2.9 Guidelines for Each User ………………………………..5

3. Installation & Configuration 5

3.1 Check the License ………………………………………..5

3.2 Check for Appropriate Version …………………………..5

3.3 Type of Installation ……………………………………….5

3.4 Change Default Passwords ………………………………..6

3.5 Disable/Remove Unnecessary Accounts …………………6

3.6 Remove Unnecessary Scripts ……………………………..6

3.7 Verify the Features Installed ……………………………..6

3.8 View Error Log …………………………………………..6

3.9 Calculate Checksum ……………………………………...6

3.10 Install All the Patches/Hot-Fixes ……………………….6

3.11 Change Port number …………………………………….6

3.12 Implement Auditing Policy ……………………………..6

3.13 Create an Extra Administrator Account …………………6

3.14 Create an Account for Back-up …………………………..6

3.15 Create Sufficient Tablespace ……………………………..6

4. Operations & Maintenance 7

4.1 User & Application Accounts …………………………….7

4.2 Control the Distribution of Database Name ……………….7

4.3 Encrypt the Contents ………………………………………7

4.4 Effective Auditing …………………………………………7

4.5 Make Password Changes Mandatory ……………………….7

4.6 Isolate Production Database ……………………………….7

4.7 Dormant Accounts …………………………………………..7

4.8 Privileged Accounts ………………………………………..7

4.9 Test Security Patches ……………………………………….8

4.10 Display Warning Banners …………………………………8

4.11 Hide Vendor & Software Version …………………………8

4.12 Login Restrictions …………………………………………..8

4.13 Remedial Action ……………………………………………8

4.14 Re-Evaluate the Security Policy ……………………………8

4.15 Backup & Recovery ………………………………………..8

5. Web Based Databases.......................................10

5.1 Configuration ……………………………………………….10

5.2 Security Threats …………………………………………….10

6. Incident Response .... ………………..10

7. Security Checklist for DBA 11

8. Reference 14

1. Introduction

Database servers are the foundation of e-commerce, e-business and e-governance systems. They should be subjected to the same level of security scrutiny as operating systems and networks. Data integrity and security of a database server can be compromised by:

- Complexity
- Insecure password usage
- Misconfigurations
- Unrecognized system backdoors

It is, therefore, imperative that an adaptive database security policy is formulated and used regularly.

Recent examples where database servers were compromised include the Slammer worm which crashed machines hosting a popular RDBMS and criminals accessing over 8 million credit card numbers from a company's database server.

1.1 *Database Vulnerabilities*

A thorough security analysis of a database server must be much broader, assessing potential vulnerabilities in all possible areas like :

- *Risks associated with vendor-supplied software*
    - Bugs
    - Missing operating system patches
    - Vulnerable services
    - Insecure choices for default implementations and configurations.
- *Risks associated with administration*
    - Security options available but not used correctly
    - Risky default settings
    - Improper granting of excessive privileges to users
    - Unauthorized changes to the system configuration
- *Risks associated with user activity*
    - Insufficient password strength
    - Inappropriate access to critical data
    - Malicious activities such as stealing contents of databases

1.2 *Database Security*

Database security can be divided into the following key points:

- Server Security
- Database Connections
- Table Access Control
- Restricting Database Access

All of these vulnerabilities need to be considered when securing database servers.

2. Planning

While planning a database server for the organization, the DBA should consider the following issues:

2.1 *Type of Server Required*

Depending upon the requirements, the DBA should choose from one of the following types of server types:

- Standalone server
- Client-Server Model
- Clustering Model

2.2 *Server Security*

Server security is the process of limiting actual access to the database server itself. It is the most important aspect of security and should be carefully planned.

- The database server should not be visible to the world.
- There should be no anonymous connection.
- A database server supplying information to a dynamic website should never be on the same machine as the web server.
- If a database server is supplying information to a web server then it should be configured to allow connections only from that web server.
- Every server should be configured to allow only trusted IP addresses.
- A database server supplying information to a homegrown application running on the internal network should only answer to addresses from within the internal network.

2.3 *Database Connections*

- All updates to a database via a web page should be validated.
- No data should be allowed to be submitted if a normal user can’t input data.
- A super-user account like "sa" should not be used for every connection and data source on the server.
- Only the minimum privileges required by a user to connect with a database should be provided.

2.4 *Table Access Control*

Table access control is one of the most overlooked forms of database security because of the inherent difficulty in applying it. Properly using table access control will require the collaboration of System Administrator, Database Administrator and Database Developer.

2.5 *Physical location of server*

Physical protection should be provided to the server depending upon the importance of data being stored in it.

2.6 *Separate storage area*

A separate storage area for keeping the backup of the database and archive should be decided in advance.

2.7 Identify Users and Their Needs

Identify the types of users and grant them minimum access permissions to database depending upon their needs.

2.8 *Security Policy*

A security policy consisting of the procedures and regulations needed to maintain a desired level of system security should be based on:

- ***Identification of Security Requirements***
  - ➢ Identify the business importance of the data and the associated processing system.
  - ➢ Assign a security priority to the data, based on the business case evaluation
  - ➢ Identify the classes of users requiring access to Database Server and the data that it controls
  - ➢ Identify the system resources that require protection to ensure continued availability data to all valid users.
- ***Identification of Security Levels***
  - ➢ Minimal Security : Users have unrestricted access to all database server resources. No one performs security related auditing and no formal security policy exists.
  - ➢ Moderate Security : A small privileged subgroup has unlimited access. The DBA performs only occasional auditing of security-related events, and no formal security policy exists for the users.
  - ➢ High Security : The DBA is the only user whom database server permits to perform the following security-related actions:
    - o Define username/password combinations to whom database server will grant access.
    - o Define and control the auditing of security-related events.
    - o Review the results of security-related audits.

2.9 *Guidelines for Each User*

Each user should receive a document that states the security policy, explains the importance of security, outlines the role of the user in supporting that policy, and defines the guidelines for protecting passwords and data.

3. Installation & Configuration

A DBA should keep in mind the requirements and applications of the database server before starting the installation. The DBA, in consultation with management and Network Administrator, should :

3.1 *Check the License of the Database Server Software*

Ensure that the instance being installed is legal and properly licensed.

3.2 *Check for Appropriate Version*

Ensure that the instance to be installed matches with the hardware and software already present in the organisation.

3.3 *Type of Installation*

Choose custom mode of installation to change the default values and avoid known vulnerabilities of the database server.

3.4 *Change default passwords*

No default passwords should be kept for the database server. Secure passwords should be assigned to all the accounts and objects as defined in the password security policy of the organisation.

3.5 *Disable/Remove unnecessary accounts*

Any account created while setting up the server should be disabled or deleted if not required. If the account has to be kept then the password should be changed.

3.6 *Remove Unnecessary Scripts*

Any script installed or copied during installation of the server should be deleted as soon as possible to secure the database.

3.7 *Verify the Features Installed*

After the completion of installation, check to ensure that all the required features have been installed and no required feature is missing.

3.8 *View Error Log*

After completion of the installation, the error log should be reviewed to ensure that there was no error in the installation.

3.9 *Calculate Checksum*

Checksum of the files installed should be performed to ensure that all the required files have been installed and there has been no error in the installation.

3.10 *Install All the Patches/Hot-Fixes/Service Packs*

Install all the patches available to strengthen the database server. Any hot-fixes and service packs provided by the vendor should be installed immediately. This has been covered in detail in *'System Security Guidelines'* issued by CERT-In.

3.11 *Change Port Number*

Change the default ports used by the database in consultation with the Network Administrator.

3.12 *Implement Auditing Policy*

Implement the auditing policy of the organization.

3.13 *Create an Extra Administrator Account*

Create an extra account with Administrator privileges to recover from any situation in which the database server or administrator account is compromised. It should be kept confidential and not disclosed to any person.

3.14 *Create an Account for Back-up & Archiving*

Create a separate account for backing up the database and archiving it. This account should be different from the administrator account.

3.15 *Create Sufficient Tablespace*

Ensure that sufficient tablespace has been provided to all the applications so that no application comes in conflict with system tables for space and resources.

4. Operations & Maintenance

4.1 *User and Application Accounts*

- During installation, some default accounts are setup. Keep an inventory of all accounts and disable or remove the unnecessary ones.
- Assign privileges to application-owner account as per their roles. Make a policy for assigning roles and privileges and follow that when opening new user accounts.
- It is advisable to secure RMAN account properly, because anyone who can access that account can alter backup schedule and destinations.
- Make sure that the passwords are not visible by file searches (such as use of the UNIX grep command).

4.2 *Control the Distribution of Database Name*

Service names and aliases should be used to mask the physical location and name of every database in the system.

4.3 *Encrypt the Contents*

Enable encryption of stored data on a high risk database environment. Any user trying to access the data should need the right password as well as the encryption key.

4.4 *Effective Auditing*

Logs should include the time and date of activities, the user ID, commands (and command arguments) executed, ID of either the local terminal or remote computer initiating the connection, associated system job or process number, and error conditions (failed/rejected attempts, failures in consistency checks, etc.)

4.5 *Make Password Changes Mandatory*

Users should be required to change their passwords frequently. Force passwords to expire and prevent the reuse of old passwords.

4.6 *Isolate Production Database*

A Production Database should be kept separate from development database.

- Revoke operating-system-level access for developers on the production server and implement a standardized change-control process.
- Never publicize the name of the database and server supporting the production application.
- Forbid the use of the production database for development or testing.

4.7 *Dormant Accounts*

Accounts must be regularly reviewed for inactivity, and any dormant accounts should be suspended.

4.8 *Privileged Accounts*

Passwords for privileged accounts should be given only to people with a need for privileged access. The passwords for these accounts must be encrypted when network is used to access them.

4.9 *Test Security Patches*

Vendor or author provided security patches must be evaluated for compatibility, and installed.

4.10 *Display Warning Banner*

Wherever feasible, a login banner, stating that the system is for authorized use only, should be displayed for anyone attempting to connect to the system.

4.11 *Hide Vendor & Software Information*

Wherever feasible, all operating system, version/release numbers, and vendor information provided in login/sign-on banners should be limited or disabled.

4.12 *Login Restrictions*

Wherever feasible, login restrictions (by time of day, by system address, etc.) should be implemented.

4.13 *Remedial Action*

If any unauthorized or undesirable activity is noticed, one of the following remedial actions should be taken to address the problem:

- Change compromised passwords.
- Change access rights.
- Audit intensively all actions of particular users.
- Deny the offending users any access to database.
- Change the security policy.

4.14 *Re-evaluating the Security Policy*

A system security policy should not remain static. The following factors make a review of the security policy necessary:

- Changes in the profiles of users who access the system.
- Changes in business needs that raise or lower the value of the data being protected.
- New releases of database server software that might introduce new security features.
- Discovery of security violations, potential violations, or attempted violations.

4.15 *Backup & Recovery*

Databases should be protected from accidental data loss. A general backup and recovery strategy must be designed depending on various factors, such as database size, volume of changes, and resources available. Attention must be paid when choosing the backup type (incremental, full) and testing the whole set of procedures to recover the system in case of disaster, and in a timely manner.

- **Backup**

Backing up databases should protect against accidental loss of data, database corruption, hardware failures, and even natural disasters

- A database backup records the complete state of the data in the database at the time the backup operation completes.
- A transaction log backup records the state of the transaction log at the time the backup operation starts.

Depending upon the requirements, one of the following ways to backup the database should be selected:

- *Complete database backups*
  Perform a full backup of the database, objects, system tables, and data.
- *Differential backups*
  Back up data that has changed since the last complete backup.
- *Transaction log backups*
  Back up all database modifications transaction logs.
- *File and filegroup backups*
  Back up database files and filegroups rather than the entire database.

- **Recovery**

A backup is only as good as the recovery it can provide. A DBA may experience one or more of the following database integrity problems and will be required to recover the lost data.

Invalid Data	This is the smallest, but most common database problem. It occurs when a finite number of invalid entries find their way into the data.
Corrupted Database Object	The next level of database problems includes situations in which a single or limited number of database objects have become corrupted or invalid.
Full Database Corruption	At this level, the scope of the problem is so significant that the database is no longer operational and a full database recovery must be performed.
Multiple Database Corruption	The largest levels of database problems occur when multiple databases within the enterprise have been corrupted and must be recovered as a set.

- *Transaction Recovery*

Transaction recovery, also known as data-level recovery, allows DBAs to precisely identify and correct the invalid data. The DBA should select and examine each of the changes that were applied to the database by using selection and filtering capabilities.

- *Database Object Recovery*

Database object recovery allows DBAs to identify and recover only the missing or damaged objects. DBA should use tools available for Object recovery contain built-in database intelligence to identify all of the objects making up the database from information captured when the backup was taken. This information can be then matched against the existing database environment. Missing or invalid objects can then be automatically recovered from the physical backup of the database, while valid objects remain unaffected.

- *Full Database Recovery*

The DBA may need to recover entire database. This requires the database to be closed. During this time, users will not be able to access important business-critical applications.

- *Multiple Database Recovery*

  The DBA should select tools that combine an enterprise-wide view of the organization with maximum database recovery capabilities. This enterprise-wide recovery management console allows consistent, reliable backup and recovery plans to be established and automated.

5. Web Based Databases

Access to a web based database server is via network connections such as SQL/net. Authentication is often an automated or scripted task, or the network access is via a single username as far as the operating system on the server is concerned.

5.1 *Configuration for Web-Based Database Server*

It is recommended that in a web-based application, a typical configuration should keep the database with the sensitive information behind a firewall. It will be accessed from an application-server also located behind a second firewall, which will receive the web server requests. This three-tier design isolates the Web-server from the database, isolating the database server from the outside users by two dedicated private networks. Only the Web server can communicate through the firewall with the application-server, and only this can communicate with the database. This configuration is relatively secure and special attention must be paid on securing the information sent to the client from the Web server, the Web-server itself, and the database/application-server system. The application-server will incorporate the event logging and the security analyzer that recognizes unauthorized attempts to log into an account.

5.2 *Security Threats to Web Based Database Servers*

All web-based database servers have ports that they listen to. Most intruders do a simple 'port scan' to look for ports that are open that popular database systems use by default.

For web security, the following three primary areas must be addressed:

- Server security : Ensure security for the actual data or private HTML files stored on the server.
- User-authentication security : Ensure login security to prevent unauthorized access to information.
- Session security : Ensure that data is not intercepted as it is broadcast over the Internet or Intranet.

6. Incident Response

In case the database server security is compromised or the intruder gains unauthorized access to the database server or the data therein, the DBA should act in accordance with the organizations security policy and check to see if following points have been taken care of :

- Isolate compromised system(s) or take steps to contain attack.
- Consult with management, Network Administrator, legal counsel and law enforcement expeditiously and consult the organization's security policy.
- Report incident to CERT-In.

- Investigate to determine if the attacker also has compromised other systems also.
- Analyze the intrusion, including:
  - Modifications made to the system's software and configuration
  - Modifications made to the data
  - Tools or data left behind by intruder
  - Review system logs, intrusion detection, and firewall log files.
  - Restore the system
- Restore from backups if data has been modified or tempered.
- Disable unnecessary services
- Apply all patches/service packs required to strengthen the database server security.
- Document lessons learned.
- Analyze the intrusion and modify the security policy if required, in consultation with the management and Network Administrator.

7. Security Checklist for a Database Administrator

- *Ensure that the database RDBMS version is a vendor supported product version.*

- *Monitor the RDBMS software on a regular basis to detect unauthorized modifications.*

- *Ensure that all directories and file permissions created by the installation of a RDBMS are protected in accordance with security evaluation specifications if available or, if not, vendor recommendations.*

- *Ensure that end user accounts are not granted permissions to change directory or file permissions associated with the database software.*

- *Ensure that all default installation passwords will not remain on DBA database accounts.*

- *Change all default database account passwords after the application installation and disable default application accounts that are not required.*

- *Ensure that the following password management rules are enforced:*
  - Configure all database accounts to be protected by a password, certificate, or approved network-based authentication.
  - Assign a temporary password at account creation.
  - Store all passwords in an encrypted format.
  - No database account name and password should be visible to the host operating system.
  - Passwords should be alphanumeric characters and should include at least one numeric character.
  - Passwords should not contain consecutively repeating characters.

- *Restrict access to files containing logon credentials and encryption keys to SAs and DBAs.*

- *Ensure that RDBMS installation default object privileges are not granted to PUBLIC except for those object privileges whose removal is not supported by the RDBMS vendor.*

- *Ensure that all user accounts are granted roles containing the minimum set of privileges required for the application.*

- *In a shared production/development environment, ensure that no application developer account is given permission to create, alter, or drop schema objects.*

- *Ensure that application developer accounts on shared production/development systems are at no time given DBA roles within the database or on the operating system.*

- *Ensure that all database actions are traceable to an individual user logon.*

- *All database objects should be owned by the database system, database administrators, or by an account created especially for application object ownership.*

- *Ensure that a tested and verifiable backup strategy is implemented on all RDBMS databases*.

- *Ensure that roles or application object privileges are not granted to PUBLIC.*

- *Ensure that the DBA role is restricted to authorized DBA accounts in a production environment.*

- *Ensure that the DBA role is restricted to DBA accounts and authorized application developer accounts in a development environment.*

- *Restrict assignment of* ***alter****,* ***index****, and* ***references*** *object privileges to DBAs, object owners and predefined roles.*

- *Restrict the assignment of the grant option of any object privilege to DBAs.*

- *Restrict access to the AUD$ table to DBAs and/or security auditors.*

- *Do not include a version number, vendor name or any identity thereof in production database instance names.*

- *Protect the environment variable identifying the location of the password file.*

- *Configure an idle time limit for all database accounts through the use of profiles.*

- *Deny Everyone group any permissions on any database files or directories.*

- *Restrict* ***write*** *permissions to database registry keys to the Database Administrators and System Administrators.*

- *Refer CERT-In website for regular updates on latest database vulnerabilities and security advisories.*

8. References

- Oracle Technology Network
  http://technet.oracle.com/

- University of Pennsylvania, Information Systems and Computing
  www.upenn.edu/computing/policy/acsp.html

- www.governmentsecurity.org/

- www.pentasafe.com/whitepapers/pentaSafe_wp_common_vulnerabilities.pdf

- http://education.protegrity.com/downloads/SecureData_IBMv1.pdf

- www.dbazine.com/

- Database Security Technical Implementation Guide,
  Defense Information Systems Agency, United States of America.
  www.disa.mil