Auditing Security and Compliance across Database Environments

Monica Harjani
Indus University

Abstract

Every business has to deal with some or the other type of data. These data is stored, managed and used such that they are accurate, efficient and highly secured. There are so many such Database Management System software today that helps in dealing with these data in a disciplined way. But there are users who are not sure about which system to use when. This paper compares the features of various Database Management Systems in terms of security so that it is helpful for the reader to understand and decide upon the system they need to use based on their requirements. I have provided the comparison between highly used Database Management Software used in market which is again done on various features in terms of security as everyone needs to have secure data. Based on the type of the requirements the Database Management System can be selected from there highly used systems such as Oracle, Microsoft SQL server and MySQL. When a user has Linux or Unix background then Oracle is more preferred. If Windows is the platform then Microsoft SQL server Is more preferred. If system is build using the open source technologies (like php) then MYSQL is more preferred.

1. Introduction:

A Database is a collection of organized data in the form of rows and columns.

A Database Management System is software that is used to manage the data in the database using different languages like DDL, DML and DCL.

Every organization today has database to store their large volume of data and one of the so many database management system software available to mange those data.

How would these organizations decide to which dbms software to select and implement to manage that large volume of data?

The most popular and frequently used software available in the market are Oracle, Microsoft SQL Server and MySQL.

2. Oracle database

Oracle is Object-Relational Database Management System (ORDBMS) produced and marketed by Oracle Corporation. There have been tremendous changes in the development of the software since the first version released in 1980. Oracle Database is generally used when the Os is Unix or Linux. There are various versions released till date.

It provides much secured infrastructure through a vast range of products, processes, and technologies to help prevent unauthorized access to confidential information with reduced cost of managing users and facilitates privacy management.
The security features provided are as below:

1. Non password Credentials (Kerberos and PKI Certificate) Support

   No password credentials for target access are managed which gives advantage of strong authentication to secure target access. Because SSH key-based authentication is supported for host access, Kerberos ticket and PKI certificate are supported for database access.

2. Named Credential Audit Support

   Accountability and traceability of the use of named credentials meets a user's security compliance requirement. The system audits all operations on named credentials, including creating, updating, deleting, associating, and accessing a named credential.

3. Credential Sharing and Recycling Support

Sharing and recycling credentials without disclosing the sensitive content of the credentials which significantly reduces the number of credentials that Enterprise Manager must maintain to access managed targets, reducing the time and effort needed to manage the stored credentials.

4. Centralized and Secure Credential Storage

   All credentials used to access managed targets are named, encrypted, and maintained in a centralized logical credential store. This storage saves time and effort for Enterprise Manager administrators who must manage the credentials with tasks such as keeping credentials synchronized.

5. Fine-Grained Privileges for Access Control of Named Credentials

   It controls access to stored and shared credentials. These fine-grained privileges now keep sensitive credentials protected to meet the security requirement, while simplifying and facilitating the sharing of the credential.

6. Independent Life Cycle of Named Credentials

   A credential can exist without any entity association. The independent life cycle of named credentials simplifies the management of credentials, from creation to deletion, by separating credentials from the life cycles of other entities in Enterprise Manager.

7. Named Credentials Testing and Verification

   This release introduces the ability to test and verify a named credential and provide enough diagnosis information in the log file for analysis. It allows a user to verify the credentials before using them; this prevents use of the wrong credential in jobs or deployment procedures.

3. Microsoft SQL Server

Microsoft SQL server is a relational database (RDBMS) developed by Microsoft. Different versions are available in SQL server to target different types of audience and different types of work from small organization using few data to large organizations handling millions of data. Microsoft SQL Server is generally used when the OS is Windows.

Microsoft SQL Server provides various security features for different versions.

SQL Server 2012 provide Mission Critical Confidence with greater uptime, blazing-fast performance and enhanced security features for mission critical workloads; Breakthrough Insight with managed self-service data exploration and stunning interactive data visualizations capabilities; Cloud On Your Own Terms by enabling the creation and extension of solutions across on-premises and public cloud.
The security features that SQL server 2012 provides are,

- Data Transparent Encryption
- Extensible Key Managements
- Kerberos authentication enhancements
- SQL Server Audit
  - Change Data Capture
  - Audit Resilience
  - T-SQL Stack Info
- Policy-Based Management
- Common Criteria Certification

4. MySQL

MySQL is an open source Relational Database Management System. It is RDBMS being used for web based software development. It is very fast reliable and flexible Database Management system that provides a very high performance and it is multi threaded and multi user Relational Database management system. MySQL is generally used when applications using open source technologies are used.

Security features provided by MySQL are

1. Three-Tier Design

Also referred to as n-tier design, this design incorporates the three layers of a Web application running on different servers, usually set apart by firewalls that have specific rules to only let traffic through to the specific port on a specific server at whichever layer that the user is trying to access:

Internet -> Firewall -> Web -> Firewall -> Application -> Firewall -> Database

Something else that it should demonstrate is that it is very costly to implement such a design because firewalls and servers are not cheap. Oftentimes, a sys admin will choose a compromise, combining the application and database servers. This isn't ideal from a security perspective; nevertheless, it is a vast improvement over leaving a sensitive database facing the Internet directly. The point is that if one of the layers closest to the Internet is compromised, then several more layers still need to be compromised before access to the vital information can be gained.

2. Access Control

Access to information contained in the tables must be properly regulated. This can be done with control over direct access to the tables, and also through views. Views and privileges assigned to the views can be created to limit users to only see specified portions of data contained within a table [2]. Through the use of the selects, projections and joins, existing relations between tables in a relational database, as well as a single table, can be created. Control over the read, insert, update and delete commands must also be assigned appropriately within those views.

3. Roles

Role-based authentication should be considered when adding access to any database. Typical roles for access include administrator, user, programmer and operator. For the first three roles, it is fairly obvious what access should be granted; it is the operator role that can be a sticking point. Operators are expected to play an essential part in the production operation of a system, yet they are often restricted in what type of access they are granted. Segregation of duties should be considered in the operator role, instead of just granting one operator control over an entire process. Operators' roles do need to be carefully defined and kept within the realm of production support as much as possible. Furthermore, all roles should have logging enabled to keep track of what occurs.

4. Integrity

Key ingredient in database design is data integrity, or ensuring that the data that is stored in the database is in fact valid and accurate. It is best to determine very early in the design process that it will be responsible for ensuring the integrity of the database. No matter the sensitivity of the data (credit card information vs. your record collection), if the data isn't right, then what good is the database? When the owner is determined, they should maintain this role and appropriate access only, not attempting to dole this out to others less it become diluted and possibly become corrupt.

A good process for ensuring the integrity of the data includes understanding what is processed and then identifying what can be considered personal, critical, or proprietary. As with any security issue, risk must be assigned according to the likelihood that something could occur to that data and the potential effect of such an occurence. Most of all, accountability must be assigned and designed into the environment where the database resides. Otherwise, the goals of privacy and security cannot be met [3].

5. Encryption

The sensitivity of the data will logically determine the need for the use of encryption. There are a few things to consider when thinking about implementing encryption:

6. Change Control

It is important to remember that changes made to the database, whether structural or to the data itself, must be tracked and regulated by interested parties. Whether formal or informal, the process must be defined and followed by all roles defined in the database structure.

7. Specific MySQL Security Considerations

Many variables that are mentioned in the following discussion are set in the “my.cnf” file. The location of this depends on how the MySQL database is installed. Essentially, you can create the file on your own, or use one of the handy sample files that come with the distribution (see the “support-files” directory). Then, if you would like the parameters to apply all MySQL users, you can place the “my.cnf” file in /etc. If you want the parameters to apply to specific users, then you can set the file in their respective home directory as “.my.cnf”. Make sure that the appropriate permissions are applied to the file wherever it resides, ensuring that the unauthorized users cannot write to it.

A discussion of the basic post-installation configuration of MySQL is beyond the scope of this discussion. For that information, please refer to the MySQL documentation, Post-Installation Set-up and Testing, and Setting Up the Initial MySQL Privileges, as well as Ryan W. Maple's article MySQL Security.

8. The MySQL Permission Model

In order to fully implement a secure MySQL database, it is necessary to learn the MySQL access control system.

5 References

[1] http://www.adp-gmbh.ch/ora/misc/features.html

[2]http://www.oracle.com/us/products/database/security/index.html

[3]http://en.wikipedia.org/wiki/Microsoft_SQL_Server

[4]http://www.microsoft.com/sqlserver/en/us/solutions-technologies/mission-critical-operations/SQL-Server-2012-security-and-compliance.aspx

[5]http://www.databasesecurity.com/dbsec/comparison.pdf

[6]http://docs.oracle.com/cd/E25054_01/doc.1111/e25353/whats_new.htm#CEGCGECE

[7] http://www.symantec.com/connect/articles/secure-mysql-database-design