International Journal of Innovation in Computer Science and Information Technology
Vol.4, NO1 , P:7 - 13
Received: 01 October 2021
Accepted: 01 December 2021

Data Protection & Security in MS-SQL Server DBMS

Yousef Rahimy Akhondzadeh
Student of Master of Software Engineering, Faculty of Electrical & Computer Engineering, Qazvin Branch, Islamic Azad University, Qazvin, Iran
wkpviana@gmail.com

Abstruct

Data has been, is, and will be an important part of our lives. Today, with the advancement of technology, the amount of data produced and its importance has increased so that a wrong data stream can lead to catastrophic and even irreparable problems. These errors can be made by information thieves and hackers, intentionally, to corrupt or misuse data. In this age when data is a very sensitive commodity, attacking and destroying or losing, or even disclosing it can cause irreparable damage, and in sensitive cases such as military and political data, it can lead to the failure and even overthrow of governments. With regard to such cases, the need for data protection and security and related solutions are raised, such as data encryption and monitoring and control of data access.

Keywords: Database Security, SQL Server, SQL Server Security, Data Security.

Introduction

Given the past efforts to preserve and transmit data to future generations, it is clear that data protection and security have been of great importance since ancient times. In order to protect the data in different eras, humans tried to preserve it in the form of cave inscriptions, stone inscriptions, clay tablets, papyrus scrolls, and so on. In the past, data was also classified according to its sensitivity and importance and sometimes encrypted to prevent the damage and disclosure of sensitive and important data. For example, the ATBASH cryptography used by the Hebrews in writing the Bible of Jeremiah, or the cryptography of Alexander, and so on.

As science and technology progressed, data also grew exponentially in terms of 3v (Volume, Variety, and Velocity), so the amount of data generated between 2010 and 2015 increased from approximately 2 Zeta Bytes to 15.5 Zeta Byte, and in 2020 this amount will reach approximately 64.2 Zeta Byte, and it is predicted that in 2025 this amount will reach more than 3 times the data produced in 2020, ie approximately 181 Zeta Byte. It is natural that because data is an important part of our human life, this amount of data needs to be maintained, managed, and protected.

Since security plays the most important role in the field of data, hacking and publishing data can cause great harm to individuals, organizations, and even governments. A hacker can even destroy or delete data instead of publishing and abusing it. This is why the security of databases and hard-earned data over time is so important.

IT environments and units typically operate in a complete cycle. New hardware and software are always added and old resources are retired. Networks and infrastructure are expanding. The new staff is hired and previous staff is retired. Despite all this, securing IT environments, especially databases, is a vital issue.

Database administrators usually do not have the time, skills, or resources to deal with securing their databases and only perform tasks related to managing and maintaining databases. Therefore, database security is often marginalized and unnecessary tasks.

Although many well-known database management systems offer internal security mechanisms and controls, the information contained in databases is still at great risk. In this article, we present solutions to prevent the risks of unauthorized access to data, especially in the Microsoft SQL Server database management system.

Research method

In this study, several well-known database management systems, including MySQL, Oracle, and MS-SQL Server, were first examined in terms of how to manage, store and access data. Then, the history of attacks on each of them were studied in terms of the type of attack and the reasons for success in infiltration, and the level of vulnerability of each database management system. So the following is a summary of some of them that have been studied by other researchers.

A preview of previous works done

As noted in the article (Ken Munro, 2006), when installing the Oracle database system there are about 14 default passwords that most database administrators leave by default instead of changing them to stronger passwords. For example, a DBSNMP account with the default DBSNMP password used by the Oracle Intelligent Agent, which can be used as a database administrator to log in and take ownership. Also, database administrators usually use the same, easy, and guessable passwords for all parts of the database management system, and by finding one of them, many parts of the database can be accessed. An Oracle database can also be accessed using the Oracle TNS Listener service, for which an effective password is usually and rarely assigned, all of which can be minimized by setting a strong password.

In the MS SQL Server database management system, the Windows user is usually used by default to communicate and log in to the database during installation. While you can use the “sa” user with a strong password and disable or delete the Windows user.

There is also a less common cause in the MySQL database system that is sometimes noticed by hackers. MySQL has a very powerful function called “UDF” (User Defined Function) that allows you to write libraries that MySQL can load into memory. This function can allow hackers to inject SQL code into their libraries and take ownership of the server.

Also (Kev Dunn, 2005) a security expert in his article stated that databases, like any other software product, are prone to important bugs such as buffer overflow. When considering a target, it is easy to see how the buffer overflow bug can provide the attacker with control of the host server and access to information. The most common examples of buffer overflow bugs found in some database management systems are:

- MS SQL UDP Resolution Overflow
- MS-SQL Pre-Authentication Overflow
- Oracle 8/9 Long Username Overflow
- Oracle 8/9 TNS Listener Service Name Overflow

In addition, problems with database engines or communication protocols can lead to similar defects. These can be very dangerous, even without flaws such as “SQL Injection”, some of which include:

- MS-SQL xp_cmdshell / xp_regread (allowing manipulation of the OS)

- Oracle UTL_FILE / UTL_TCP (allowing OS file and socket operations)
- Oracle CTYSYS.DRILOAD (trivial account privilege escalation to DBA in 9i)

In another article (Devanshu Trivedi et. Al, 2016) they state that there are mainly three steps in which information must be secured. The first step is in any application where the data is being processed. The second step is while information is being transmitted through network channels. The third step is in data warehouses where information is stored for future use. Different database management systems offer different controls for securing data after being stored in a single database.

One approach to securing information against unauthorized access is to create “View” on tables. Changes can be saved in the main tables if their operations are found to be legal and valid where users interact with the views. These views can be created by selecting specific columns or rows of tables and permissions that restrict access to information. Database objects can also be kept somewhat hidden using aliases.

Oracle 11g provides a built-in tool called “TDE” (Transparent Data Encryption) for encrypting and decrypting sensitive data that can be applied to all primary keys in various tables. TDE can also be used to secure other critical information.

In another paper (Josh Shaul, 2008), the database lifecycle is described in four simple iterative steps that include Assess, Prioritize, Fix, and Monitor. Security tasks should be performed automatically as a regular part of database maintenance. Security relies so much on regular evaluations and validation that it can quickly become frustrating and overlooked. For example, you need to change the default passwords, use strong passwords, delete or deactivate unused user accounts, and so on.

Also in another article (Roman Ceresnak et. Al, 2021), the reasons for security flaws in databases are stated in the following three cases:

- The majority of the systems for data processing have one security layer.
- None encryption of incorrect data or results/ output data.
- Access to the data of unethical experts in the IT field, which represents the data loss risk.

Only 37% of organizations use a systematic protection system against SQL Injection attacks. Detection of SQL Injection attacks, protection of data against them, mapping of sensitive data, encryption can be among the things that, if used properly, can significantly reduce the security risks associated with databases.

Suggested solutions

In this paper, based on previous research and based on studies, the following solutions are suggested to deal with and reduce the possibility of data corruption due to database security threats, especially in the case of MS-SQL Server database management system:

- **Use of RAID:** RAID (Redundant Array of Independent Disks) technology is one of the tools that can be used to protect data against possible data corruption due to hardware problems.

- **Data distribution:** One of the proposed solutions for data protection is to distribute them on separate servers and even in different geographical locations. This minimizes the possibility of access and damage to other servers if one of the databases is attacked.
- **Changing the default port:** To connect to the MS-SQL Server database, the default port "1433" is used, changing this port can be one of the ways to increase database security.
- **Use of firewalls:** Firewalls can be one of the tools to prevent cyber attacks. They can prevent unauthorized access to data and equipment.
- **Isolating database servers:** Isolating database servers from other network systems, especially the Internet, can greatly prevent the infiltration of databases. In this case, APIs (Application Programming Interface) must be used to access the data, and if it is necessary to access the data from outside the network and through the Internet, this must be done through the controlled Internet web services.
- **Definition of separate users:** It is suggested that instead of using the "sa" user who has full access to all sections of the database, use the division of tasks and assign separate users with more limited and controlled access to each section and task. By doing this, firstly, no user has access to all databases, and secondly, in the event of an attack by that user, the damage to the data is limited.
- **Rename the default "sa" username:** In the MS-SQL Server database management system, during installation, a default user name is created with the name "sa", which has access to all parts of the database. One of the suggested solutions to increase security and reduce the possibility of intrusion through this user is to change its name.
- **Disable the "Windows Authentication" option:** One of the ways to connect to the database is to select the Windows Authentication option, which can be easily accessed through this option if the database server is compromised. Therefore, it is recommended that you disable or delete this option.
- **Disable the "xp_cmdshell":** In MS-SQL Server there is a procedure called "xp_cmdshell" that allows you to access some parts of the operating system and database and, like the Windows Command Prompt, execute some database commands and procedures. One way to increase database security is to disable this routine when we are not using it.
- **Remove access to create a new user:** If you have defined several separate users in databases for separate sections, you must be careful to deny access to create and change user information from them, because despite such access, in the event of attacks by one of these Users, the hacker can easily create a separate user for himself and continue to work secretly in the future.
- **Disable inactive users:** If you are using multiple separate users for your database system, you should be aware that if a user is removed from the organization/company (for example,

retired, transferred, or fired), their user ID must be deactivated or deleted immediately. This is because most attacks and sabotage in organizations/companies are usually carried out by people who have access to the data and have been removed from the organization/company with some dissatisfaction. Also, users who are inactive and have not been in the organization/company for some time (for example, long vacations) should immediately deactivate the user ID associated with that person because unused user IDs can be used by hackers to normalize malicious activity.

- **Use secure passwords:** In selecting passwords for different sections and users of the database, passwords must be set to have maximum security. For example, the minimum length of passwords is 8 characters. The selected password is a combination of uppercase and lowercase letters, numbers, and symbols. The selected password can not be guessed.
- **Breaking large tables (do not keep all eggs in a basket):** One of the techniques to reduce the risk of data loss is to break large tables into several smaller tables. By doing this, we will lose only part of the data if any of the tables are destroyed. This can also affect the speed of data selection.
- **Use of Views:** It is better to use views with limited access levels to perform operations on data because they can keep sensitive data hidden from users, which can play an important role in maintaining the security of the database.
- **Limiting the number of transactions per unit time:** Sometimes hackers send large volumes of the same and similar transactions to the database to disable or disrupt the database engine (causing buffer overflow errors, etc.) in such cases. It is possible by limiting the number of transactions sent per unit time (for example, in the banking system, it can be said that at least 120 seconds must elapse between each transaction from a source account), especially for transactions that perform insertion, editing and deletion operations, it can minimize the success of such attacks.
- **Using “Log Backup”:** Backups that are made at regular intervals based on the schedules and policies of the database administrators are one of the options that can be used to recover and restore most of the data in the event of data corruption. But in addition to the usual backups, it is better to provide a “Log Backup”, because in this type of backup, all operations performed on the data can be monitored and in case of data corruption, they can be repaired as much as possible. This is often overlooked by database administrators because log backups take up a lot of disk space.
- **Do not use “Relation”:** Since “Relation” is often used to communicate between tables based on the “Primary” and “Foreign” keys defined in the tables, attackers can find all the related tables if they have access to one of the tables, and attack

the others. Therefore, instead of using “Relation”, it is better to establish the connection between the tables manually through queries that we design in the form of Functions, Procedures, and Views.

- **Encryption of sensitive data:** In cases where hackers intend to disclose data instead of destroying it, encrypting sensitive data such as credit card numbers, contact numbers, addresses, passwords, etc. can minimize the possibility of their misuse. Also, in cases where you do not use Relation or split large tables into several smaller tables, we can encrypt key relationships and tables as much as possible by encrypting the relationships between the tables.

Conclusion

Because data plays such an important role in today's society, and it takes years and years to collect some of it, it is therefore important to protect and maintain it. Such valuable data, which may be sensitive military, political, medical, etc. data, can also be a good target for hackers and cyber thieves who can not only misuse it but also destroy it and cause irreparable damage. Therefore, despite such threats, the issue of data protection and data storage and management systems becomes more important.

Refrences

Ken Munro. (2006). Database security – an oxymoron? Infosecurity Today. November/December 2006.

Kev Dunn. (2005). Dig yourself out of the data crate – database security isn’t new, so why can’t we get it right? Network Security Journal. October 2005.

Devanshu Trivedi, Pavol Zavarsky, Sergey Butakov. (2016). Enhancing Relational Database Security by Metadata Segregation. The 2nd International Workshop on Future Information Security, Privacy and Forensics for Complex system (FISP-2016).

Josh Shaul. (2008). Implementing database security: using attack analysis to improve your defences. Network Security Journal. July 2008.

Roman Ceresnak, Michal Kvet, Karol Matiasko. (2021). Increasing Security of Database During Car Monitoring. 14th International scientific conference on sustainable, modern and safe transport. 2021.