2021

Database Security and Backup Plan

ATENA VAHEDIAN (TEAM LEAD)
ADEYEMI OWOSENI
ASHLEY HUSCROFT
LEYSAN GILFANOVA
MAHMOUDREZA KARIMKHANINEJAD
MOHAMMAD MOSIHUZZAMAN

Objectives

The purpose of the database security and backup plan for the “WeVote” industrial project, is to detail the Development teams’ recommendations to safeguard the database against potential threats to its integrity. The need to create robust security and backup plan cannot be overemphasized as every database will encounter threats in both the design stage and after its creation.

Overview

Database in a general context is an organized collection of data that are stored and assessed electronically from a computer system. This collection of data can be of great asset to the organization, managing the database. Hence, a plan is required to protect these assets from known and unknown threats.

The security plan for the “WeVote” industrial project will focus more on protecting database objects such as tables, views, stored procedures, and constraints, with respect to how they can be accessed and who can have access to them. The sensitivity of the data is subjective in this project; therefore, the various levels of protection will depend on how sensitive the data in question is categorized.

Threats to “WeVote” Database

An important aspect to consider in this plan are the potential threats. Threats tend to put the database at risk. A typical example is a power failure.

It is important to acknowledge that threats are inevitable, and they vary from one database to another depending on the sensitivity of the data it contains. Potential threats include unauthorized modification, unauthorized disclosure, denial of service. Other examples of specific regulatory threats are computer misuse, commercial sensitivity, personal privacy and data protection, and audit requirement. Only threats associated with the “WeVote” industrial project will be addressed here.

Unauthorized Modification

These are targeted towards database objects, which in return will disrupt the accuracy of the information contained in the objects. For example, an attempt by an unauthorized user to sabotage the election results in a particular ward. If the right information is interfered with, it could send a wrong interpretation to other candidates using the ward details in their campaign strategy.

Denial of Service

Loss of availability is also a potential threat to this database. It is also known as a denial of service. It may occur when an attacker gets a database function to misbehave. Database commands are altered in a way that confuses the database, the query parser, or a sub-function enough to crash.

Misuse of Computer

In an organizational setting, legislation on the misuse of computers is always made available to database users. Computer misuse simply addresses the violation of access control and users attempt to create damage to the database by changing the state, introducing worms and viruses to interfere with standard operation.

"WeVote" Security Model

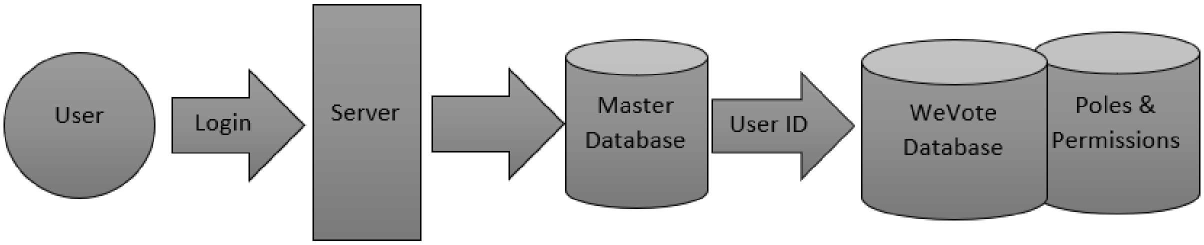

Figure 1.1 Proposed security model for "WeVote" project.

As shown in the proposed security model, from left to right users can login to the database server and then via the database server which keep information about who can access "WeVote" databases. A user is given an ID and is allowed access to certain database objects, what they are allowed to do in the database is governed by the roles they may have or specific permission they may have.

Server Level security

This is where the user gets access first of all to the server. As we already know, users can have server-level security via three methods which are windows user login, membership of a window group, or SQL- Server-specific login.

Windows Authentication

"WeVote" project will be incorporating windows authentication. This authentication will be used because It is the most secured authentication available because it uses a certificate-based security mechanism. Management of passwords and accounts are centralized when using domain accounts. This enables the domain administrator to manage all logins that are used within the organization hence the database administrator does not need to manage separate accounts.

The login is managed by user information in this case a username and a password. Once logged in, users would be identified by their login (SQL Server) where users' rights are determined by fixed server roles.

It is however important to implement user defined role instead of assigning fixed server roles. The fixed server roles cannot be modified, and you do not want to grant more access than required by the user in question. SSMS graphic user interface will be used to create and drop logins, grant deny and revoke permissions throughout the project instead of TSQL commands.

Database Security Login

Once the individual has gained access to the server, access can now be granted to the "WeVote" database by adding them to the database. If there is a need to set up a guest database user in the future, a guest database user can be created for users without their own access. A user's database allocation can be done under the user mapping section at the server level. New users can be created at the database level by setting up a new login and assign a database role to the user. It is important to know the types of fixed database roles available under SQL and what they are permitted to do. The owner of WeVote database will be Kourtney Branagan. This is a special role that has all permissions in the database.

Object Ownership and Security

Ownership is a critical aspect of security in SQL Server. The object includes tables, views, stored procedures, constraints that are contained by a schema. A schema is owned and everything within the schema has the same owner. It should be noted the fact that a user has access to the database does not automatic them permissions by default. What they can do within the database solely depends on the type of permission assigned to the particular user. Permissions may be granted directly to the user by assigning the permission to the role and assigning the user to the role. This is generally regarded as best practice. Users may be assigned to multiple roles which imply that a user can have multiple permission parts. Object permission assigned within "WeVote" database will be assigned by utilizing GRANT, REVOKE, and DENY. It should be noted that REVOKE removes the permission assigned, DENY overrules GRANT. In this project, we will be using the SSMS GUI instead of T-SQL commands.

Database Backup Plan

A typical database is prone to different types of failures. The "WeVote" database is not exempted from such. To mitigate these risks, we recommend implementing proactive measures to minimize failures by creating a regularly scheduled, automated backup plan and restore strategy which is outlined in further detail below. . Some of the failures a database may encounter are hardware failure (network failure, SQL Server instance failure, media, and disk failure), application failure, system failure, and user failure. It is not guaranteed that any of the failures listed will not happen. It is recommended the “WeVote” database institute a full back up, differential back up, and transaction log back up plan.

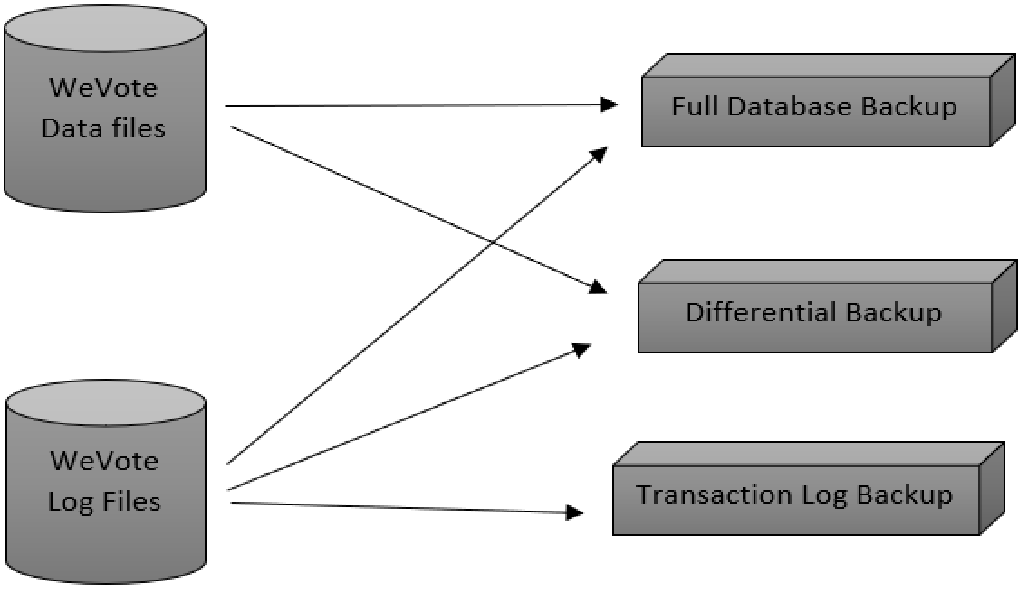

Figure1.2 SQL Backup types used in “WeVote” database.

Full Database Backup

The full back up is the foundational backup for “WeVote” database as it should be done before any other backup within the database. It stores all objects of the database which includes tables stored procedures, functions, views, indexes, etc. the full backup enables restoration of a database in the exact form it was at the time of the backup. A full database backup creates a complete backup of the database as well as part of the transaction log so that database can be recovered. Full backup for “WeVote” project will be created using SSMS graphical user interface and will be scheduled on weekly basis outside of regular business hours (For example, Friday night at 7pm or Saturday morning) based on best practice. If changes are not made frequently within the database for example updating tables or assigning roles and permissions, a bi-weekly full backup might be a better approach. It is however important to understand the risk tolerance is higher when using biweekly backup.

Differential Database Backup

The differential contains all the updates and changes that have been made since the last full backup. It varies in size depending on the number of transactions that have happened since the last backup. Differential backup usually runs faster than the full backup, because it captures the state of the change extent since the last backup was created. As the differential backup increases in size, the restoration time might also increase significantly. At this point, it is recommended that a full backup is taken at this point and set the interval to establish a new differential base. The differential backup generally saves storage space and time required to back up, therefore, it is advisable to have relatively frequent differential back in order to not lose the speed. "WeVote" differential backup will be done daily for the recommendations provided under the full backup i.e., weekly basis which is the best practice and biweekly backup which is an alternative for less frequently used database. SSMS graphical user interface is the preferred method to create the differential backup.

Transaction Log Backup

Transaction log stores a series of log files which provides the history of every modification of data in a database. It contains all log records that have not been included in the last transaction log backup and allows the database to recover to a specific point in time. This implies that transaction log backups are incremental while the differential backups are cumulative. In other to recover "WeVote" database to a specific time before an assumed failure that led to data loss, we have to recover the full database, the most recent differential back up, and all the corresponding transaction log backups. Modifications are contained and maintained using the log sequence number in the log chain (unbroken series of logs that contain all the transaction logs necessary to recover "WeVote" database to any point in time).

This backup only works with a full or bulk-logged recovery model. For this reason, the recovery model for "WeVote" database will be the full recovery model which is the default model. To set up a transaction log, we will use the SSMS graphical user interface.