ManageEngine
Log360

Basics of SQL Server log auditing

Introduction

Every database administrator has two major concerns: ensuring that data is secure and the database server is available. As one would expect, databases are prime targets for attacks. Database attacks not only have the potential to cripple business productivity, but they can also expose sensitive data, which often carries legal repercussions. What's worse is that organizations often don't realize they've been attacked until weeks or months after a breach.

Regulations such as the General Data Protection Regulation (GDPR) in Europe and the Notifiable Data Breaches (NDB) scheme in Australia are mandating organizations to report data breaches to authorities within a given time frame. These regulations are holding organizations accountable for the security of the data they store.

Most organizations focus on preventive security measures, such as authentication, encryption, and vulnerability scans; however, these measures don't guarantee data security. There's always going to be an opportunity for a malicious actor—who very well could be an insider—to exploit a security flaw and do real damage. That said, it's vital that security teams are in a position to detect and respond when things go wrong.

The importance of SQL logs

What if something goes wrong right now in one of your SQL servers? Say a regular end user is assigned the database administrator role—will you be instantly notified?

Such an event would be a cause for major concern as it may compromise the security of sensitive data. Scenarios like these can be detected by decrypting logs generated by SQL Server. While maintaining a log management system, security teams often focus disproportionately on a few event sources, such as network devices and domain controllers, and they don't have the same level of visibility on the systems that store sensitive data, such as SQL servers. This results in security teams lacking the information they need to thwart attempts to breach their database at an early stage.

SQL audit logs have a crucial story to tell; in fact, several events, including logon events, only get logged on SQL Server. Organizations must generate and maintain an audit trail to ensure that activities like accesses, changes, and other important database events are being tracked.

The need for a security information and event management (SIEM) solution with SQL auditing capability

The native auditing functionality of SQL Server has some major shortcomings. A lot of valuable information is made available in the logs, but the information isn't actionable. For starters, there's no built-in reporting and alerting available, and it can be time consuming for security teams to sift through all the audit data without specialized tools.

A SIEM solution centralizes audit information from all the SQL servers in your network for effective analysis. This helps security teams discover and respond to security threats swiftly and demonstrate compliance with ease.

Considerations and prerequisites

Before getting started, there are some crucial aspects of SQL auditing that you should think about:

1. **Event volume and performance.** Security teams must identify the number of SQL servers in their environment, as well as the number of database connections that occur during a day. The server's capacity should be analyzed to ensure that the event load can be handled without creating any performance issues.

2. **Audit policy.** The SQL audit policy isn't enabled by default, so it must be configured. Enabling "advanced auditing'" provides more in-depth insights into database activities. Security teams need to identify their security and compliance objectives, and then define an audit policy that generates the log entries they need without compromising performance. The audit policy can then be pushed via a group policy object (GPO).

3. **Log collection mechanism.** Determine how the logs will be collected from SQL Server. Most SIEM solutions in the market give security teams the flexibility to choose either an agent-based or agentless mechanism to collect the logs. Note that an agentless log collection mechanism requires certain ports to be kept open for communication purposes.

4. **Database threats.** Be aware of the common database threats, such as SQL injection and privilege abuse. A good starting point to assess your security posture is evaluating your typical response to top database threats. Gaps can then be filled by defining more security reports and alerts.

Scheduling audit reports

Generating audit reports is important for visualizing and reviewing security events. The basic objective is to track all the changes performed on the database right from the database logon. Reviewing events every 24 hours is a widely accepted best practice. However, more mature organizations must review these events more frequently.

Broadly speaking, database audit reports encompass four categories:

1. **Data manipulation language (DML) activity**
2. **Data definition language (DDL) activity**
3. **User activity**
4. **Server activity**

Configuring security alerts

Setting up alerts for indicators of compromise (IoCs) is vital to discover and respond to potential data breaches quickly. While reports are effective for reviewing events periodically, alerts must be set up for events that require immediate investigation. The idea is to watch out for events that shouldn't happen under normal circumstances. Here are a few examples:

1. **Privilege escalations**
2. **Server shutdowns/restarts**
3. **Repeated failed logons and account lockouts**
4. **Changes made to permissions (roles)**
5. **Modifications made to sensitive columns**
6. **Suspicious backups of data**
7. **Known attack patterns, such as SQL injection**
8. **Changes to audit policies**

Enhanced security posture with event correlation

Organizations that are seeking a more mature security posture must leverage advanced techniques, including event correlation and analytics. Event correlation associates events happening on SQL Server with events happening in other parts of the network. This provides more context about an unusual event to provide a complete picture of the entire attack kill chain.

ManageEngine Log360's SQL auditing capabilities

ManageEngine Log360 is a comprehensive SIEM solution that comes with a pre-built SQL auditing capability that can tackle everything discussed in this e-book and more! Log360 automatically discovers SQL servers in your network; it enables auditing, collects and archives logs, generates exhaustive audit reports, and triggers alerts for potential database threats.

- **In-depth SQL auditing.** The tool comes with pre-built reports and alerts for tracking every important security event occurring on SQL Server. To learn more about all the report and alert profiles that are available, reach out to our support team at log360-support@manageengine.com.

- **Correlation rules to detect attacks.** Log360's powerful correlation engine can correlate SQL audit logs with information collected from other event sources; this way, data breach attempts are detected at an early stage. Examples include the detection of suspicious SQL backups and repeated SQL injection attempts.

About the author

Siddharth Sharath Kumar is an IT security and compliance specialist on ManageEngine's product marketing team. He writes articles and e-books, regularly hosts webinars on key IT security topics, and presents at ManageEngine's conferences and other industry events across the globe.