Database tools for auditing and forensics

Database auditing requires more than just the right tools: those tools also have to be properly configured to offer the information that's needed and database performance that's required.

Sponsored By: IBM
Database tools for auditing and forensics

Table of Contents

Database tools for auditing and forensics

Resources from IBM
Database tools for auditing and forensics

by Adrian Lane, Contributor

Database administrators are tasked with creating audit trails for security and compliance auditing, but those who simply read the manual on how to enable auditing will be disappointed. Database audit tools have their quirks, and each one of them can perform miserably without taking the time to properly plan for the auditing process. It's not uncommon for auditing or tracing to cause more than 50% degradation in database performance. That means seemingly simple auditing will end up slowing down the database, filling up table space, collecting too many events and creating problems for yourself in maintenance and report generation.

Instead, start your process by setting up a test database and run some basic performance tests: Make side-by-side comparisons, first with database auditing off, and then compare different auditing methods, configurations and filtering options to your baseline. This not only provides a great way to understand how each feature effects performance, but also helps identify resource bottlenecks. And trust me, this is information you need to know before reconfiguring production servers. Even if you are using database audit logs to feed SIEM and log management platforms, don't assume those tools will optimize audit settings; setup and maintenance will still be required to enable the audit trails.

Here are some database security best practices for tuning auditing tools:

Auditing methods: All databases offer more than one way to collect audit data, so you should try simple benchmarks to compare your options. For IBM DB2, audit is usually the best bet. But for cases where you only need audit events for a specific user, event monitors may be more efficient. For Oracle databases, fine-grained audit and 'db, extended' offer great flexibility, but also potentially disastrous performance because of the amount of data they produce.

Auditing options: Try some of the different options and run tests. For Oracle databases and DB2, compare database table storage vs. writing to the OS/file system. The former
offers convenience for data retrieval, but the latter performs far better and does not fill up tablespace.

**Resources and management**: By tuning resource allocation to help with data storage, especially when storing audit data within the database, it's easy to overflow tables. Plan on creating scripts to archive audit data and prune the tables of older events. For SQL trace, event streams are stored within separate files, and perform better when placed on dedicated drives. Auditing processes tend to take up quite a bit of memory as well. DB2 Audit Buffers can have a significant negative effect on performance, so plan on increasing memory allocation. Sybase Inc.'s Adaptive Server Enterprise (ASE) requires lots of memory allocation for processing and responds well when the audit queue size is increased. To optimize efficiency of writing to disk, many of the databases allow for data block tuning, which optimizes for write-only and block sizing options that are a multiple of the audit row size.

**Filtering**: Possibly the single most important step is filtering. Work with your security and compliance teams to find out what sort of information they really need, as opposed to what they would like to have. Filtering one or two types of events from the collected data stream can significantly reduce storage and computational overhead. Tools like SQL Server trace can be useful for getting needed data, but don't filter event types well. For the most part, however, all of the databases mentioned offer filtering for user events, administrative events, meta data operations and system-level events. With DB2, for example, it's not uncommon to reduce overhead by 80% with careful filtering. Take the time to understand what you need, and then filter out what you don't.

Don't assume any one form of auditing is the right way until you have verified it through benchmarking. By spending a little time understanding the options, you can save a lot of aggravation.

**About the author**: Adrian Lane is CTO at the security research and advisory firm Securosis.
Resources from IBM

Implementing Database Auditing to Address Security & Compliance Requirements

Oracle Account Security Techniques - Complimentary Chapter Download

Managing the Entire Database Security & Compliance Lifecycle with IBM InfoSphere Guardium

About IBM

At IBM, we strive to lead in the creation, development and manufacture of the industry's most advanced information technologies, including computer systems, software, networking systems, storage devices and microelectronics. We translate these advanced technologies into value for our customers through our professional solutions and services businesses worldwide.