Database Auditing and Compliance in a Mainframe Environment

Craig S. Mullins,
Corporate Technologist, *NEON Enterprise Software, Inc.*

white paper

NEON
Enterprise Software, Inc.

Table of Contents

white paper

Introduction

We are all surely aware of the increasing burden of complying with government regulations. And nowhere is the pressure to comply greater than on data stored in corporate databases. Organization must be hyper-vigilant as they implement controls to protect and monitor their data, as well as the accesses to that data.

One of the more useful techniques to protect your company's database data is through data access auditing, which is a facility for tracking the use of, and modifications made to, database resources and authority. A data access auditing capability enables companies to produce an audit trail of information with regard to their database data. This audit trail can contain information such as what database objects were impacted, who performed the operations, and when the activity occurred. A comprehensive audit trail of database operations, coupled with an analysis engine to review and analyze the audit trail allows data and security professionals as well as IT auditors to perform in-depth analysis of access and modification patterns against data in all of your database systems. Only when armed with such details is it possible to comply with regulations, pass security audits, and drill-down into the details to review potential vulnerabilities for effective issue resolution.

A Look at the Regulations and Requirements

Why would you need a fine-grained audit trail? Well, let's consider a few of the regulations that can require organizations to create database audit trails.

Many of the PCI Data Security Standard requirements emphasize the importance of real time monitoring and tracking of access to cardholder data, as well as continuous assessment of database security health status.

HIPAA, the Health Insurance Portability and Accountability Act, directs health care providers to protect individual's health care information going so far as to state that the provider must be able to deliver a list of everyone who even so much as looked at their patient's information. Could you produce a list of everyone who looked at a specific row or set of rows in any database you manage?

And what about the Sarbanes-Oxley Act (SOX)? The goal of SOX is to reduce fraud and conflicts of interest, to improve disclosure and financial reporting, and strengthen confidence in public accounting. Section 404 specifies that the CFO must guarantee the accuracy of the processes used to add up the numbers. Those processes are typically guided by computer programs that access and manipulate data in a database system. As such, it is important to be able to track who changed database schemata and database data for SOX compliance.

white paper

Figure 1. Industry Regulation Database Audit Requirements

Audit Requirements	SOX (CobiT)	PCI DSS	HIPAA	CMS ARS	GLBA	ISO 17799 (Basell II)	NERC	NIST 800-53 (FISMA)
Access to Sensitive Data *(Successful/Failed SELECTs)*		✓	✓	✓	✓	✓		✓
Schema Changes (DDL) *(Create/Drop/Alter Tables, etc.)*	✓	✓	✓		✓	✓	✓	✓
Data Changes (DML) *(Insert, Update, Delete)*	✓			✓		✓		
Security Exceptions *(Failed logins, SQL errors, etc.)*	✓	✓	✓	✓	✓	✓	✓	✓
Accounts, Roles & Permissions (DCL) *(GRANT, REVOKE)*	✓	✓	✓	✓	✓	✓	✓	✓

DDL = Data Definition Language (schema changes)
DML = Data Manipulation Language (data value changes)
DCL = Data Control Language (authorization changes)

Source: Guardium, Inc.

The chart in Figure 1 summarizes the type of database audit requirements imposed on IT by several of the predominant industry regulations.

Privileged Users

A comprehensive database auditing solution should provide a mechanism to audit privileged users, such as DBAs and SYSADMs. Many privileged users have blanket access to all corporate data. But you can't just revoke the authority of super users because the nature of their jobs requires such overarching privileges.

Because privileged users can access and modify database resources at their discretion, an organizational approach to auditing their every action is prudent. Privileged users have broad authority, but they should not be accessing and modifying production data without due cause.

A database auditing solution enables organizations to implement a "Trust, But Verify" policy for their privileged users. This allows the administrators to retain the authority they need to be able to do their jobs, while at the same time giving the organization the peace of mind that everything the privileged users are doing is tracked for security and compliance purposes.

Without a database auditing solution in place, privileged users are a potential compliance problem lurking within every database implementation.

white paper

Internal and External Threats

Database auditing is important because there are many threats to the security of your data. External agents trying to compromise security and access your company's data are rightly viewed as a security threat. But industry studies show that the majority of security threats are internal – within your organization. Indeed, internal threats can comprise 60% to 80% of all security threats. The most typical security threat comes from a disgruntled current or ex-employee that has valid access to the DBMS. Auditing is crucial because you may need to find unauthorized access emanating from an authorized user.

A robust auditing facility permits auditing at different levels within the database system, for example, at the database, database object, and user levels. One of the biggest problems with existing internal DBMS audit facilities is performance degradation. The audit trails that are produced must be detailed enough to capture sufficient information for each auditable action. But capturing so much information, particularly in a busy system, can cause performance to suffer. Furthermore, the audit trail must be stored somewhere, which can be problematic when a massive number of changes occur. Therefore, a useful auditing facility must allow for the selective creation of audit records to minimize performance and storage problems.

So, What is Database Auditing?

Database auditing is the process of monitoring access to, and modification of, selected database objects and resources within operational databases and retaining a detailed record of the access where said record can be used to proactively trigger actions and can be retrieved and analyzed as needed.

Database Auditing Techniques

There are several popular techniques that can be deployed to audit your database structures. The first technique is trace-based auditing, which is usually built directly into the native DBMS, such as with DB2 for z/OS. Commands are set to turn on auditing and the DBMS begins to cut trace records when activity occurs against audited objects. The problems with this technique include a high potential for performance degradation when audit tracing is enabled, a high probability that the database schema will need to be modified, and insufficient granularity of audit control, especially for reads.

The biggest problem with this technique is the high potential for performance degradation when audit tracing is enabled. Indeed, the IBM DB2 manuals indicate up to a 10 percent performance hit when DB2 audit traces are started.

Another technique that can be used is to scan and parse the database transaction logs. Every DBMS uses transaction logs to capture every database modification for recovery purposes. Software exists that interprets these logs and identifies what data was changed and by which users.

The drawbacks to this technique include:

- No capture of read-only accesses (because reads aren't captured on transaction logs)
- The fact that there are ways to disable logging that will cause modifications to be lost
- Performance issues scanning volumes and volumes of log files looking for only specific information to audit
- The difficulty of retaining logs over long periods for auditing when they were designed for short-term retention for database recovery.

The third database auditing technique used by many distributed database solutions is to sniff packets for database requests as they cross the network. Capturing the SQL statements as they cross the network can generate an audit trail of the database requests that go over the network. The problem is that not every request goes across the network. This is especially the case for mainframe transactions. For example, a DB2 CICS application, where all the work is mainframe-resident, doesn't require TCP/IP. So this work can't be captured by packet sniffing. The same applies to IMS/TM and TSO requests, or any other work done directly on the mainframe.

white paper

The final, and recommended approach, it to employ proactive monitoring of operations at the database server level. This technique deploys a software tap to capture all database requests as they occur. It's important that all database access can be audited, not just network calls. This is the only technique that works well for mainframe auditing because most mainframe database requests don't go out over the network. Proactive audit monitoring doesn't require transaction logs, doesn't require database schema modification, and will be highly granular in terms of specifying what to audit.

Database auditing can be a crucial component of database security and compliance with government regulations. Be sure to study the auditing capabilities of your DBMS and to augment these capabilities with additional tools to bolster the auditability of your databases.

Satisfying All Stakeholders

As you investigate your database auditing requirements and the solutions available, keep in mind the needs of each of the key stakeholders involved in the auditing of your database systems: security and operations, compliance officers and auditors, and DBAs and applications developers.

The needs of these stakeholders are outlined in the diagram in Figure 2.

Security and operations will require a secure audit trail. Auditing is no help at all if the audit trail can be surreptitiously accessed and modified. Security personnel will also be interested in the ability to review, analyze and report on the audit trail to ensure the enforcement of organization security policies and to determine if policy adjustments are needed.

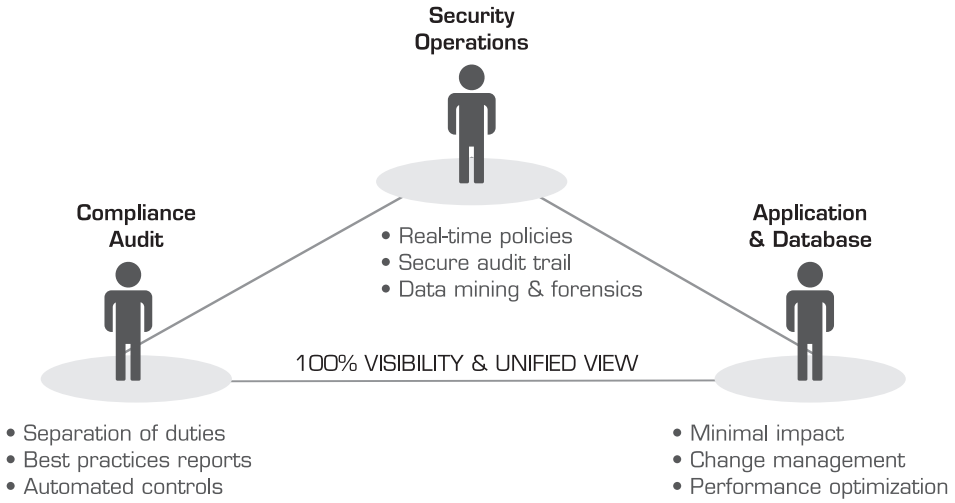

Figure 2. Key Stakeholder Requirements.

Compliance officers and auditor will demand separation of duties. For example, it is imperative that DBAs are not required to implement and maintain any database auditing solution. If you're auditing privileged users, you don't want to rely on those same users to start and stop auditing traces, do you?

For regulatory compliance efforts a database auditing solution that ships with best practices reports for the regulations you are enforcing can significantly reduce the workload of a project and therefore minimize the timeframe needed for your organization to achieve compliance.

And do not forget the requirements of your ***application programmers and database administrators.*** A database auditing solution should not require extra work to implement, such as requiring application program changes or database schema modifications. Furthermore, the solution should not incur a heavy performance penalty. The biggest concern of the DBA staff will be ensuring that the overhead required to audit data access is minimal and does not "dim the lights"

white paper

The Questions That Must be Answerable

As you investigate the database access auditing requirements for your organization, you should compile a list of the types of questions that you want your solution to be able to answer. A good database access auditing solution should be able to provide answers to at least the following questions:

1. Who accessed the data?
2. At what date and time was the access?
3. What program or client software was used to access the data?
4. From what location was the request issued?
5. What SQL was issued to access the data?
6. Was the request successful; and if so, how many rows of data were retrieved?
7. If the request was a modification, what data was changed? (A before and after image of the change should be accessible).

Of course, there are numerous details behind each of these questions. A robust database access auditing solution should provide an independent mechanism for the long-term storage and access of audit details. The solution should offer the canned queries for the most common types of queries, but the audit information should be accessible using industry-standard query tools to make it easier for auditors to customize queries as necessary.

Database auditing can be a crucial component of database security and compliance with government regulations. Be sure to study the auditing and compliance requirements of your organization and to augment your DBMS with third-party tools to bolster the auditability of your databases.

About NEON Enterprise Software

NEON Enterprise Software is the technology leader in enterprise data management software and services. As the rules of business change, our solutions let you efficiently control, protect, retain and manage data to comply with today's business and legal requirements. Founded in 1995, NEON Enterprise Software serves customers worldwide with its dedicated team of industry experts. For more information about NEON Enterprise Software, visit www.neonesoft.com or call 281.491.6366 or 888.338.6366.

white paper