What Every Good CIO Needs to Know About Mainframe Database Auditing
By Craig S. Mullins

Synopsis: Database auditing can help you achieve regulatory compliance; this article describes what you must understand and consider before moving forward.

Callout: After you generate your compliance roadmap, you'll need to determine what level of data access auditing is required.

Regulatory compliance has become a critical aspect of the IT landscape, and is a big component of every CIO's job. Nowhere is compliance more crucial than in mainframe database management. A growing number of regulations dictate increased efforts be made to better secure and protect the accuracy and privacy of enterprise data. Regulatory compliance requires diligence from CIOs and their team.

The most valuable enterprise data frequently is stored in a mainframe database, so organizations must implement more robust auditing capabilities into their DB2 and IMS environments. CIOs can quickly lose their job, as well as credibility, if they don't take responsibility for protecting and auditing this valuable corporate asset.

The Regulatory Environment

Let's take a moment to review several of the high visibility regulations:

- The goal of the Sarbanes-Oxley Act (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.
- The Health Insurance Portability and Accountability Act (HIPAA) requires healthcare providers to protect an individual's healthcare information; providers must be able to document everyone who even looked at their information. Think about that. Could you produce a list of everyone who looked at a specific set of rows or group of segments in any database under your control?
- The Payment Card Industry Data Security Standard (PCI DSS) was developed by the major credit card companies to help prevent credit card fraud, hacking, and other security issues. A company processing, storing, or transmitting credit card numbers must be PCI DSS-compliant or they risk losing the ability to process credit card transactions. Payment card transaction data is typically stored in an enterprise database such as IMS or DB2.

So CIOs have expanding requirements to be able to prove their databases are protected so only properly authorized entities have access to only the specific data they need to do their jobs.

The ability to track who did what to which piece of data and when is important because there are many threats to the security of your data. External agents trying to compromise your security and access your company data are rightly viewed as a security threat. But industry studies have shown that most security threats are internal. Some studies have shown that internal threats comprise 60 to 80 percent of all security threats. The most typical security threat comes from a disgruntled or malevolent current or ex-employee with valid access to the DBMS. Auditing is crucial because you may need to find an unauthorized access emanating from an authorized user.

Compliance Tactics

How can organizations ensure they're complying with these and other regulations? Data access auditing, sometimes called database auditing, can help you track the use of database resources and authority. When auditing is enabled, each audited database operation produces an audit trail. The audit trail will show which database objects were impacted, what the operation was, who performed it, and when it occurred. This comprehensive audit trail can be maintained over time to allow DBAs and auditors, as well as any authorized personnel, to perform in-depth analysis of access and modification patterns against data in the DBMS.

Data access auditing is promising, but as with any technology, there are multiple considerations to understand and consider before you move forward.

You'll need to classify your data and match each data element against the regulations applicable to your organization. To do so, you'll need input from subject matter experts, legal experts, and IT staff such as data architects and DBAs. The output will differ for every organization because of the many variations in technical database implementations and the different regulations that apply to each business.

After you generate your compliance roadmap, you'll need to determine what level of data access auditing is required. Be sure to check with your internal auditing group to gather their requirements. Be prepared for the typical response: "We want to see what everyone touches." Such grandiose requests aren't feasible and the requirements can be scaled back to a reasonable level. Be careful, too, that the auditors understand mainframe technology and the idea of SYSADM and "on call" IDs. Depending on their age, they may have poor mainframe knowledge, which can impact the requirements gathering phase.

With the combined efforts of IT, compliance, auditing, and your business experts, you should be able to develop a reasonable list of what needs to be audited. You'll then need to compile a list of the types of questions you want your data access auditing solution to be able to answer. A good database access auditing solution should be able to provide answers to at least these questions:

- Who accessed the data?
- At what date and time was the access?
- What program or client software was used to access the data?
- From what location was the request issued?
- What SQL was issued to access the data?
- Was the request successful and if so, how many rows of data were retrieved?
- 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 auditing solution should provide an independent mechanism for the long-term storage and access of audit details. The solution should offer 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.

You'll also need to consider any additional type of database auditing you want to accomplish. A common need is to be able to track privileged user accesses, such as SYSADM for DB2. Users with such authority (usually DBAs) have carte blanche access to the DB2 subsystem and all its

data. DBAs are trusted agents and most won't abuse the overarching security access granted to them. But a data access auditing solution can verify that by tracking all SYSADM accesses. That will make both your DBAs and internal auditors happy. The DBA gets to keep the SYSADM authority needed to perform his or her job; the auditor gets to monitor the DBA's activities to ensure that those actions are appropriate.

Data Access Auditing Techniques

Let's examine how data access auditing solutions produce detailed audit trails. You can use several popular techniques; we'll briefly discuss four of them and highlight their pros and cons.

Trace-based auditing is usually built directly into the native DBMS capabilities. Commands or parameters are set to turn on auditing and the DBMS begins to write trace records when activity occurs against audited objects. Although each DBMS offers different auditing capabilities, some common items DBMS audit facilities can audit include:

- Login and logoff attempts
- Database server restarts
- Commands issued by users with system administrator privileges
- Attempted integrity violations (where changed or inserted data doesn't match a referential, unique, or check constraint)
- Select, insert, update, and delete operations
- Stored procedure executions
- Unsuccessful attempts to access a database or a table (authorization failures)
- Changes to system catalog tables
- Row-level operations.

The biggest problem with this technique is the high potential for performance degradation when audit tracing is enabled. The IBM manuals indicate up to a 10 percent performance hit when DB2 audit traces are started. Additional problems include the need to modify the database schema to turn auditing on and insufficient granularity of audit control, especially for reads.

You also can scan and parse the database transaction logs. Every DBMS uses transaction logs to capture database modifications for recovery purposes. Software exists that can interpret these logs and identify 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.

You also can 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 all 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 Time Sharing Option (TSO) requests, or any other work done right on the mainframe.

Finally, you can employ proactive monitoring of operations at the database server level. This technique captures 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.

Additional Considerations

As you review your requirements, be sure to implement database auditing solutions that deliver separation of duties. For example, it's imperative to ensure that DBAs don't need to implement and maintain your 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?

Finally, look for a solution that delivers canned compliance reports. For example, if you're looking to comply with PCI DSS, a database auditing solution that delivers out-of-the-box PCI reports will shorten your implementation timeline.

Conclusion

It's a daunting task to ensure compliance with today's avalanche of government and industry regulations and protect databases from the increasing online and internal threats to your precious business data. These trends have resulted in mainframe professionals being asked to more closely protect corporate data in their databases and to monitor who does what to which data and when. Data access auditing solutions can help organizations safely and proactively meet these growing requirements.

About the Author MFExec – What Every Good CIO Needs to Know

Craig S. Mullins is a data management strategist for NEON Enterprise Software. He has extensive experience in database management and data architecture, having worked as an application developer, a DBA, and an instructor with multiple database management systems. He has written two books, *DB2 Developer's Guide* and *Database Administration: The Complete Guide to Practices and Procedures*. You can contact him via his Website at www.craigsmullins.com.