AUDITING DATABASE ACCESS AND CHANGE:

A NECESSITY MORE THAN A NICETY

BY CRAIG S. MULLINS

The increasing burden of complying with government and industry regulations imposes significant, time-consuming requirements on IT projects and applications. And nowhere is the pressure to comply with regulations greater than on data stored in corporate databases.

Organizations must be hypervigilant as they implement controls to protect and monitor their data. One of the more useful techniques to protect your company's database data is through data access auditing, also known as simply 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 that record can be retrieved and analyzed as needed.

A data access auditing capability enables companies to produce an audit trail of information with regard to their database data. This audit trail should 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 an in-depth analysis of access and modification patterns against data in 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 Requirement

A fine-grained audit trail is necessary to comply with many regulations that apply to organizations of all types.

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 then there is the Sarbanes-Oxley Act (SOX) which has the goal of reducing fraud and conflicts of interest, as well as improving disclosure and financial reporting. Section 404 of the SOX Act specifies that the CFO must guarantee the accuracy of the processes used to add up the numbers; processes 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.

And these are only a few of the pertinent national, international, regional and industry regulations that must be understood and complied with.

Database Access Auditing Techniques

So now that we understand why database access auditing is important, let's look at how it can be accomplished. There are several popular techniques that can be deployed to audit your database structures.

The first technique is trace-based auditing, which is typically built directly into the native capabilities of the DBMS. For example, the audit trace feature of IBM Db2 for z/OS. When an audit trace is started, the DBMS begins to cut trace records when activity occurs against audited objects (selected by DDL option). However, Db2 only captures the first read or write per unit of work (UOW), which will clearly miss activities as most UOWs encompass more than one read or write. Alternately, Db2 audit policies can be created for named tables to capture all activity, which improves the data captured, but can create an excess of audit records that need to be stored in SMF data sets.

So, there are problems with this technique including 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.

Another technique is to scan and parse transaction logs. Every DBMS uses transaction logs to capture every database modification for recovery purposes. If you can read the log and interpret the data (which can be challenging as the data is not simple) it is possible to identify what data was changed and by which users. The biggest drawback to this technique is that database reads are not captured on transaction logs.

Additional issues with relying on log analysis for auditing data access include: it is possible to disable logging such that modification information will not be on the log and therefore not captured; performance issues scanning volumes and volumes of log files looking for only specific information to audit; and the difficulty of retaining logs over long periods for auditing when they were designed for short-term retention for database recovery.

And that brings us to the third, and preferred, method of database auditing for organizations that are serious about regulatory compliance: professional software that proactively monitors and intercepts all SQL requests as they are executed by the DBMS. It is important that all SQL access is audited by monitoring for SQL at the database level, not just by sniffing network calls. This is important because not every SQL request goes over the network, especially for the mainframe platform where much of the activity is centralized and many important business transactions never venture over an IP network (e.g., a CICS or IMS transaction accessing Db2).

Proactive intercept-based database audit monitoring does not require transaction logs, does not require database schema modification, should be highly granular in terms of specifying what to audit and should incur only minimal overhead.

One such product that implements intercept-based auditing for Db2 database access is DBARS, which stands for "Db2 Access Recording Services," available from ESAI Group. http://www.esaigroup.com/products/dbars.htm

Important Features for a Database Auditing Solution

As you investigate the database access auditing requirements for your organization, you should make sure that the solutions you examine support your DBMS using intercept-based auditing, instead of the other methods.

You should also compile a list of the types of questions 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:

- Who accessed the data?
- At what date and time was the data accessed?
- What program or client software was used to access the data?
- For batch mainframe users, what was the z/OS job name?
- From what location was the request issued?
- For distributed Db2 access, what were the names of the external server, application and workstation
- What SQL was issued to access the data?
- Was the request successful; and if so, how many rows of data were accessed or modified?
- 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 that must be investigated for each of these questions. You will want to be able to review recent activities, but you will also want to be able to review actions that happened in the past, so a robust database access auditing solution should provide an independent mechanism for the long-term storage and access of audit details. It should be easy to query the audit trail, perhaps even offering canned queries for the most common types of queries. Nonetheless, the audit information should be accessible using industry standard query tools to make it easier for auditors to customize queries as necessary.

An alerting capability is also desirable, such that when certain SQL activity is intercepted an alert is triggered to take further actions, such as recording an exception, sending information to a log or pinging a DBA or security admin.

Advanced auditing solutions also provide the ability to proactively block suspect access to the database. For example, you may want to stop any attempted access outside of normal, scheduled programs over the weekend. At any rate, it is desirable for an auditing solution to be able to block activities based on parameters such as username, program name, IP address, execution time, type of access and the like. Such a capability is important because preventing fraudulent access is preferable to allowing it and reporting that it happened!

It is also important for a comprehensive database auditing solution to provide a mechanism to audit privileged users, such as DBAs and SYSADMs. Many privileged users have blanket access to all corporate data. Although they can access and modify it at their discretion, 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 with 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.

The Benefits of a Professional Database Auditing Solution

The bottom line is that database auditing should be a crucial component of your organization's data protection strategy. Auditing database activity is a core requirement of compliance with many government and industry regulations, but auditing is also an essential component of securing and protecting the important production data in your database systems.

Be sure to study the auditing and compliance requirements of your organization and to augment your DBMS with the appropriate tools to bolster the auditability of your databases. **ETJ**

Craig S. Mullins is president of Mullins Consulting, Inc., an in-demand analyst, and author of three books on Db2 and DBA. He has more than three decades of experience in all facets of database systems development and has been selected by IBM as a Gold Consultant.
Email: mullc@craigsmullins.com