

Keeping Data Private and Knowing It

by: Ken Richardson, 11/17/2004

<http://www.securitydocs.com/library/2715>

Who's Looking at Your Data?

So your databases contain sensitive data, and privacy safeguards are in place. But is someone looking at the data anyway? Could a password have gotten into the wrong hands? Or could an *authorized* user be accessing sensitive data from a remote location after hours? What if someone in the IT organization were to use a privileged username to read sensitive data? Would you know?

How can anyone know for sure? For that matter, just how important is this issue anyway?

Data Privacy: More than Just a Good Idea

Obviously we want to protect sensitive data, such as employee salaries and customer lists. But data privacy is often more than just good practice; it's the law. We must be especially concerned with protecting data privacy whenever government regulations require us to do so. This is true for medical records, financial records, credit card numbers and social security numbers, just to name a few. If your organization must comply with regulations such as *HIPAA*, *Sarbanes-Oxley*, or *Gramm-Leach-Bliley*, you may already be familiar with the issues.

Of course, *prevention* is the first step. We must have conventional safeguards in place, such as passwords and other access controls. But assuming they are in place, how can we be alerted when one of those safeguards has been violated? That's not "if," but "when." We must assume it *will* happen, if we are to take data privacy seriously.

Ideally, we would have a surveillance mechanism watching every access to all sensitive data. And assuming most of these accesses are legitimate, we need the mechanism to bring only the "unusual" accesses to our attention. Let's call this mechanism *Data Access Auditing*.

What is Data Access Auditing?

To answer this question, let's define a few terms. Databases generally organize data into *tables* containing *columns*. For example, an "employee" table may contain a "salary" column. Access to the data generally occurs through a language called Structured Query Language, or *SQL*. These are technical details that may not be apparent to the person accessing the database, but they go on under the covers just the same.

In this environment, the perfect data access auditing solution would be able to answer the following questions about *all* SQL accesses to *all* tables and columns:

1. Who accessed the data?
2. When?
3. Using what computer program or client software?
4. From what location on the network?
5. What was the SQL query that accessed the data?
6. Was it successful; and if so, how many rows of data were retrieved?

Assuming these questions can be answered, one would also want:

7. Management by exception: functionality to interpret the audit trail and bring only the "unusual" accesses to our attention.

These seven facets comprise *the gold standard for data access auditing*. While this is a difficult standard to meet with absolute perfection, some solutions come much closer than others.

How Can We Audit Data Access?

To audit data access, we must start with an understanding of how that access occurs.

People access data in databases by using various forms of client software. This software may be provided by software vendors or by in-house developers. It may be special-purpose, accessing pre-determined data in a well-defined manner, or general-purpose, accessing whatever data the user decides in an ad-hoc manner.

The client software requests the data from a Database Management System (DBMS), which manages and protects the data using conventional safeguards. This communication typically occurs across a network, although the client software may also work from the same machine where the database resides.

Based on this overview, we have three possible locations to perform data access auditing, as shown in Figure 1.

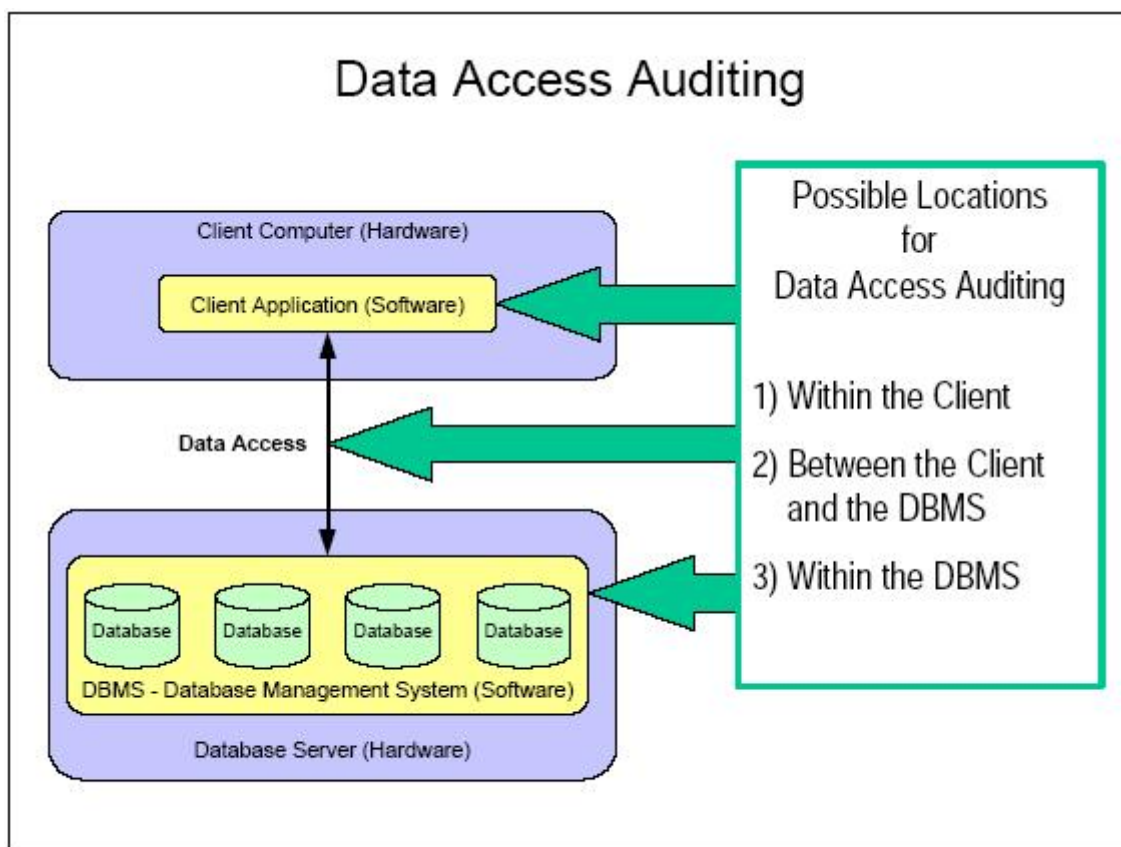


Figure 1

The Worst Choice: Auditing Within the Client

Is it even possible to audit data access from within the client? Yes, sometimes. For example, some database access tools provide the ability to track the end-user activity performed through them.

However, to provide an adequate audit trail, *all* data access would have to occur through these client tools. While it is conceivable that a site would mandate that all data access should occur only through such a tool, the efficacy of this approach is doubtful. How would management know whether anyone is sidestepping the mandate? In reality, it would be practically impossible to ensure that 100% of data access would be through the "authorized" tools.

If we want the audit trail to have any real value, auditing data access from within the client is the worst possible choice.

Second Best: Auditing Within the Database

Since the DBMS is already charged with protecting data, it may seem to be the ideal location for auditing data access. But in actual practice, there are several drawbacks to this approach:

1. **Limited audit functionality** - DBMS vendors offer varying degrees of support for data access auditing. For most DBMS vendors, the audit capabilities provided, if any, will be insufficient to the task.
Because of this, it is usually impossible or at least extremely difficult to meet all seven facets of the gold standard we defined above. For some DBMS types, it is difficult to meet even half of the requirements.
2. **Inconsistency across DBMS types** - As you might expect, the various DBMS vendors take different and incompatible approaches to access auditing. The implementation steps vary from one database to the next, the mechanisms work differently, and even the concepts can differ. In a heterogeneous environment, where more than one DBMS type is in use, this makes data access auditing not only inconsistent, but also unnecessarily complex.
3. **Performance penalty** - If a given DBMS vendor's subset of audit capabilities appears adequate for a particular situation, there is still one more drawback to consider. Most DBMS types incur a huge performance penalty for turning on the auditing mechanism, especially for 24x7x365 monitoring of all accesses to all tables and columns.
This can cause overall database performance to go from good to bad (or from just bearable to absolutely awful). Very costly hardware and software upgrades may be required to regain the pre-auditing level of performance, if it can be regained at all.

Auditing within the database is certainly a better choice than auditing within the client. But with this approach, we are likely to end up with insufficient and inconsistent audit functionality, undesirable complexity in our total data access auditing solution, and a database performance penalty that makes everyone suffer and that takes dollars directly off the bottom line.

The Best Choice: Auditing Between the Client and the Database

This brings us to the best choice: auditing the conversations between the clients and the databases. By listening to all conversations between all clients and all databases, we can achieve a comprehensive data access audit while avoiding all the drawbacks of auditing either within the client or within the database.

The challenge is that the technical details of these conversations vary from one DBMS to the next. In fact, some of these client/server data streams are quite complex. However, the concepts relevant to data access auditing are uniform across all of them.

Therefore, if we can capture and interpret these conversations and abstract them to their uniform concepts, we create a foundation for comprehensive, uniform data access auditing. This foundation paves the way for a single architecture for data access auditing even when many different DBMS types are in use.

Fortunately, various software vendors have worked to implement the capture and abstraction process. However, their implementations differ in several areas:

1. **Supported DBMS Types** - Some specialize on just one or two DBMS types, while others support many more.
2. **Coverage** - Some use periodic sampling of activity that can miss short-duration accesses while others perform comprehensive monitoring of all SQL activity.
3. **Quality** - Some are unable to interpret complex data streams or extremely complex SQL, resulting in untracked data accesses, while others handle these quite well.
4. **Usability of Results** - Some implementations store access audits as text logfiles while others use standard database tables that greatly improve the usability of the results.
5. **Performance** - Some implementations actually slow down the client/server conversations while others offer extremely low overhead or even zero overhead solutions.

Ensuring Data Privacy through Data Access Auditing

The good news is that the gold standard of data access auditing is achievable. And with careful vendor selection, it's possible to enjoy an optimum implementation that includes:

- | support for a large number of DBMS types
- | complete coverage rather than sampling
- | a high quality implementation for complex data streams and complex SQL
- | storage of results in standard database tables
- | high performance with low or zero overhead

And all of this can be implemented on a 24x7x365 basis with absolutely no impact on the databases being monitored.

In summary, whenever there is a concern for data privacy, whether based on good practice or on regulatory compliance, implementing comprehensive data access auditing between the client and the database can enable every organization to *keep data private (and know it)*.

Mr. Richardson is the President and CTO of Ambeo (www.ambeo.com), and has over 23 years of experience in senior management and technical roles in both corporate IT and software engineering environments. This background has equipped him to understand both the need for data access auditing and how best to provide it to Ambeo's customers.

About Ambeo

Since 1997, Ambeo's solutions have provided innovative performance management, data usage tracking, data access auditing and monitoring to Fortune 1000 companies with some of the largest database environments in the world. Ambeo's True Zero Overhead™ solutions provide information about how users actually interact with data that would otherwise be largely unavailable or difficult and costly to obtain. By using this information, IT management gains the visibility and understanding of data usage required to proactively manage data environments, ensure privacy and compliance, maximize database investments, improve performance and ensure end user satisfaction.

For more information contact:

Ambeo

5353 North Union Blvd. Suite 103
 Colorado Springs, CO 80918
 Phone: 719-548-7400
 Fax: 719-548-8982
 e-mail: sales@ambeo.com

