

Data Piracy - The Threat from Within Catching data thieves before it's too late

by: Ken Richardson, 09/30/2004

<http://www.securitydocs.com/library/2621>

What's Going on Here?

Believe it or not, these are actual headlines from recent trade journals. And they contain a pattern - a frightening one:

- *AOL Customer List Stolen, Sold to Spammer*
- *Acxiom Database Hacked Again*
- *IRS Computers Vulnerable, Report Says*
- *Offshore Outsourcing Poses Privacy Perils*

Databases are being stolen. Customer data, credit card data, taxpayer data - they're all vulnerable. Scary? Yes - but wait, there's more. It's not just "their" data that's vulnerable - it's *ours too!*

"Oh, really?" Our first reaction may be skepticism. If so, we may be feeling safe because of our various security infrastructures. Numerous policies, procedures, and technologies may be in place to protect us. We may be spending continuous streams of cold, hard cash on security, so aren't we justified in feeling that our databases are reasonably safe?

Well, no doubt the organizations in the headlines above were spending considerable fortunes on security too, probably much more than most of our organizations spend. But it wasn't enough. Apparently there's a hole in the security cheese that's difficult to fill.

What's actually going on here?

The Hole in the Cheese

Much of the typical security budget protects against intrusion. We're prepared for malicious outsiders breaking in, searching for weaknesses in our perimeter firewalls and our multiple layers of security. But it turns out that the biggest threat to our databases doesn't generally come from that direction.

Of course, it's precisely due to our investment in layers of security that the threat of intrusion *is* so relatively small. The intrusion threat is very real, and if we weren't so well prepared, break-ins would in fact be one of the biggest piracy threats, especially considering the high value of our data to competitors and others. So we definitely need to be spending big to protect against potential unknown intruders.

But assuming we've addressed this area adequately, the threat from outsiders is most likely under reasonable control. So what's left? Where is the danger coming from?

We can find out by reading the stories behind the headlines above. In every case, the threat came from insiders: employees, former employees, individual contractors, employees of corporate contractors, and so forth.

Insiders are the hole in the security cheese.

How Big is the Risk?

It's obvious that we need to protect our sensitive databases, such as customer lists, credit card info, and so forth. We know that there's a competitive threat involved, and also a danger of losing customer confidence when a data loss becomes known. Focusing on security and data privacy is certainly a good practice for all these reasons and more.

But the risk is even bigger. With the never-ending progression of government regulation in the current information age, data privacy is increasingly more than just good practice; it's the law!

Lawmakers seem to produce new laws as fast as hackers produce computer viruses - and some of their recent productivity has been focused on data privacy and auditability. We now enjoy numerous regulations in this area, such as *HIPAA*, *FERPA*, *Sarbanes-Oxley*, *Gramm-Leach-Bliley*, and *California SB 1386*.

HIPAA-Ensure the security of health data stored electronically

FERPA-Protect student data from disclosure

Sarbanes-Oxley-Require provability of data underlying corporate financial results

Gramm-Leach-Bliley-Ensure data privacy for consumers

California SB 1386-Ensure data privacy for customers and require notification of piracy

Of course, these regulations have worthwhile purposes and goals, but they do require work to implement, and they carry stiff penalties for failure to comply. So guess who gets to implement them? We do. They apply to virtually every organization having databases: universities, health providers, financial institutions, and businesses of every sort.

So in addition to the inherent risks involved in the exposure of sensitive data, we also face the risk of breaking the law if we allow data piracy to occur. It's no wonder regulatory compliance is so big on everyone's radar screen these days.

Conventional Safeguards

Because data security and privacy are so important, implementing conventional safeguards has been a standard practice in most companies for a long time. These safeguards include authentication mechanisms such as passwords, smart cards or biometrics, and access controls such as rights lists, group memberships and so forth.

Benefits of Conventional Safeguards

There are many great things about conventional safeguards. They're perfectly suited to policy and procedural standardization. They're generally understandable by ordinary mortals (although complex or nested access controls may occasionally challenge this). They're highly auditable. And they just plain work well. When an outsider doesn't have a valid username and password, he can't log in. When a customer service representative isn't authorized to see payroll data, he won't. And when an accounting clerk has been authorized to update accounts payable records, the bills get paid.

Limitations of Conventional Safeguards

So where's the problem? It lies just beyond the scope of conventional safeguards. It's in the realm of human ingenuity at sidestepping the system, and goes by names such as identity spoofing and privilege abuse.

Conventional safeguards are like locks on a door: they can help to keep the honest people honest, but they'll never make the dishonest ones honest. The dishonest person will steal a copy of the door key; or just find a window to climb through.

So even when conventional safeguards are in place, someone may be looking at the data in unintended ways. A password may have gotten into the wrong hands. An authorized user may be pirating sensitive data from a remote location after hours, through a VPN connection. Someone in the IT organization may be using a privileged username, intended for data backups, to steal sensitive data.

If these things were happening, would you know?

Going beyond Conventional Safeguards

Conventional safeguards are definitely the first step to data privacy. They're necessary - but not sufficient. We must first have conventional safeguards in place, but assuming they *are* in place, we need to be alerted whenever one of those safeguards has been circumvented. That's not "if," but "when." We must assume it *will* happen, if we are to take the possibility of data piracy seriously.

What Else Can We Do?

In an ideal world, we would know comprehensively how all the data has been used and by whom, for virtually all accesses to the data. This would basically be a complete audit trail of all data accesses. Of course, this knowledge would be overwhelming in volume, and since well over 99% of these accesses would be for legitimate purposes, we would really want to have only unusual or suspicious accesses brought to our attention, so we could then examine the audit trail as needed.

How could such knowledge be acquired? Only with some form of surveillance mechanism, one that watches and records every access to all sensitive data, and that brings the suspicious ones to our attention.

Let's call this mechanism *Data Access Auditing*.

What Can We Expect from Data Access Auditing?

To answer this question, let's review a few terms. Most commercial databases organize data into *tables* containing *columns*. For example, a "customer" table may contain a "credit-card-number" 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 such a database 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** - Who accessed the data?
2. **What** - What tables and columns were accessed?
3. **When** - When did they do it?
4. **Where** - From what location on the network?
5. **How** - Using what SQL query, and what computer program or client software?
6. **Result** - Was the query successful; and if so, how many rows of data were retrieved?

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

7. **Exception reporting** - Functionality to interpret the audit trail and bring only the "unusual" or suspicious accesses to our attention.

These seven characteristics comprise *the gold standard for data access auditing*. While this is a high standard, and therefore difficult to meet with absolute perfection, there are effective ways to accomplish data access auditing. As you might expect, some solutions come much closer to meeting the standard than others do.

How Can We Audit Data Access?

To assess the universe of options for auditing data access, it's helpful to start with a mental picture 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 desires to see in an ad-hoc manner.

The client software requests the data from a Database Management System (DBMS), which generally manages and protects the data using the conventional safeguards we discussed earlier. These requests typically occur across a network, although the client software may also work from the same machine where the database resides.

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

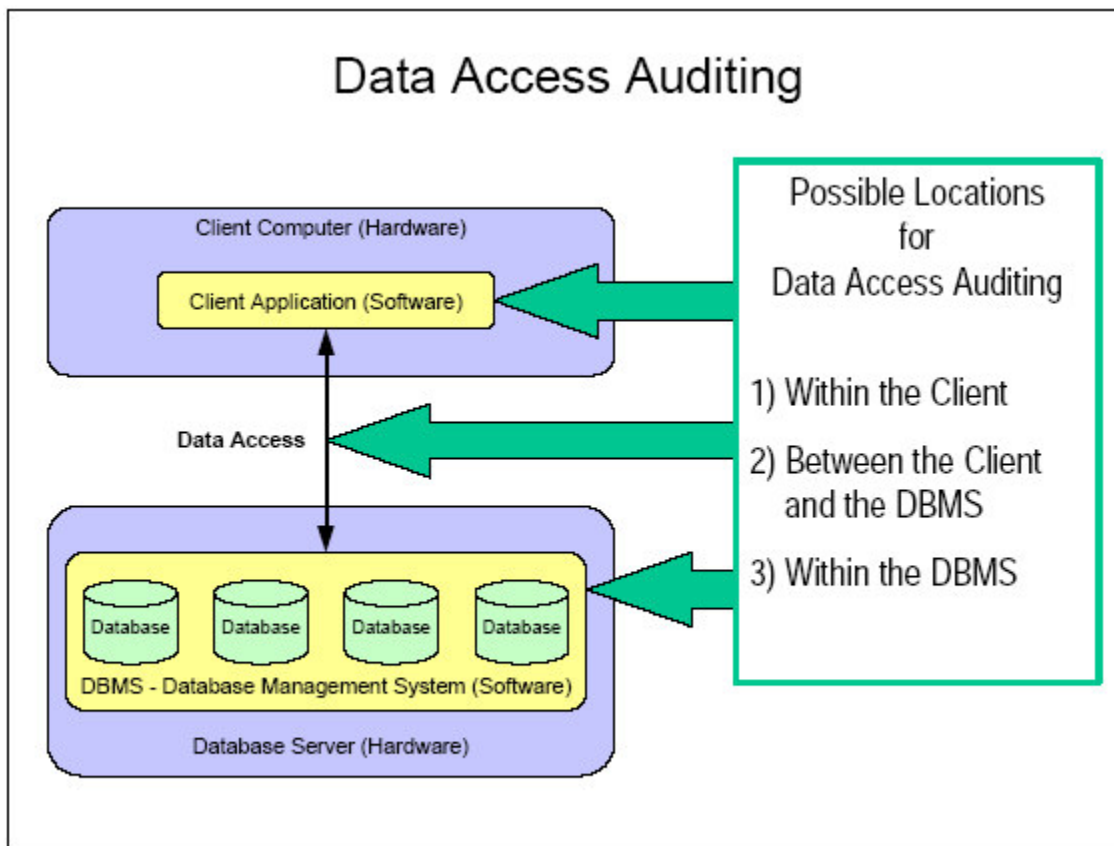


Figure 1

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. In fact, this feature has been showing up recently in an increasing number of database access tools, and is being marketed as an effective solution for data access auditing.

But there's a fly in the ointment. To provide an adequate audit trail, *all* data access would have to occur through these client tools.

While it is theoretically conceivable that a particular organization might mandate that all data access should occur only through such a tool, the efficacy of this approach is doubtful. How would management really know whether anyone were sidestepping the mandate? In reality, it would be practically impossible to ensure that 100% of data access would only be done through the "authorized" tools.

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

Second Best: Auditing Within the Database

At first glance, the DBMS may seem to be the ideal location for auditing data access. After all, every data request goes through the DBMS, and it's already charged with protecting the data anyway.

However, 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. Some are stronger; some are weaker.

Upon close scrutiny, we find that for many DBMS vendors, the audit capabilities provided (if any) are insufficient to the task. 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 (triggers versus logging, for example).

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 to set up and to use.

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 challenges are:

1. **Data capture** - Some mechanism is needed for capturing these conversations, and
2. **Diversity and complexity** - 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.

The concepts relevant to data access auditing are uniform across all DBMS types. Therefore, if we can capture and interpret these conversations and abstract them to their uniform concepts, we can create a foundation for comprehensive, uniform data access auditing. This foundation will be strong enough to support the construction of a single architecture for data access auditing throughout the enterprise, even when many different DBMS types are in use.

What's Actually Available?

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. Clearly the more the better, to support having a single access auditing architecture throughout the enterprise.
2. **Coverage** - Some use periodic sampling of activity that can miss short-duration accesses while others perform comprehensive monitoring of all SQL activity.

In fact, some claim complete coverage when in reality they are just sampling a shared memory area (SGA) multiple times per second. Of course, many SQL accesses may start and finish between the sampling intervals and therefore go unseen.

3. **Quality/Maturity** - Some are unable to interpret complex data streams or extremely complex SQL, resulting in untracked data accesses, while others handle these quite well. Of course, all vendors will claim to do this well. Comparative trials against actual workloads are the only way to absolutely identify which products are truly mature in this area. A reasonable proxy might be the number of years each vendor has actually had their access auditing products deployed into customer sites.
4. **Usability of Results** - Some implementations only store access audits as text logfiles while others can use standard database tables that greatly improve the usability of the results. For those that offer both, some have a weak database structure and others have a robust one. A strong database orientation is generally preferred.
5. **Ease of Implementation** - Some implementations require a lengthy or complex installation process and may even require the installation of software drivers or other changes for every client computer that accesses the DBMS. Others are simple to install and transparent to the end-users.
6. **Performance** - Some implementations actually slow down the client/server conversations while others offer extremely low overhead or even zero overhead solutions.

In particular, solutions that try to combine access auditing and query-blocking generally have very high overhead, since they have to evaluate 100% of the queries against blocking rules before deciding whether to pass them on to the DBMS. This results in slower performance for all queries in order to block a minuscule fraction of them.

The fastest solutions are those that just watch the client/server network traffic without impeding it.

Because of the differences cited above, the success of an organization's data access auditing implementation will depend in great part on the quality of the chosen solution. Vendor selection is clearly an important step in the process.

Obstacles to Data Access Auditing

Whenever one sets out to implement change in an organization, it's obviously wise to consider the obstacles up front rather than just jumping in and waiting for surprises. Overcoming these obstacles will usually take some work, and we should certainly expect this and plan for it when we decide to introduce a data access auditing solution into an organization.

Technical Obstacles

Fortunately the technical obstacles should be relatively few if we have selected a solution that's simple

to install and transparent to end-users.

Some solutions will be largely self-contained, and can be viewed as an *appliance*. Others may reside on one or more existing database servers or gateway servers depending on their software architecture and the architecture of the underlying DBMS. Some include special client software for viewing and monitoring the auditing results. None should require drivers or other changes on all the database client machines.

Disk space may be required for storing the audit trail. Memory and CPU time may be required for running monitoring components. Physical network connections may be required for monitoring client/server database conversations. With the right access auditing solution, these are generally not major obstacles. There's no rocket science here.

Organizational Obstacles

The larger obstacles are generally organizational, not technical. Multiple teams may want or need to be involved, such as database administration, systems administration, network administration, security administration, and so forth. Any endeavor that involves this many people had better be worth the effort! I think it's safe to say that protecting sensitive databases from piracy *is* worth the effort.

But we may encounter resistance from one or more of these groups. Objections or concerns may be raised that appear technical in nature. When this happens, it's worthwhile to dig deeper to see if the objection might really be motivated by fear. There may be fear of change, fear of losing control, fear of "being watched" by the data access auditing solution, and so forth. If some of the players are feeling overloaded already, there may just be a reluctance to take on something new.

It's important to address all legitimate concerns. Protecting data against piracy is a strategic issue, which we can't allow to become a victim of unaddressed concerns or passive resistance.

Once all concerns and objections have been raised and addressed, an action plan can be developed that assigns responsibilities and dates to the players who will install and configure the software and/or hardware involved. Just as important is to understand who will administer the solution once installed, and who will monitor the results. Separation of duties may be appropriate, especially when there is already a distinct security team that can assume the access-auditing role.

The State of the Art

The good news is that the gold standard of data access auditing is available today. We can know who, what, when, where, how, the result, and have exception reporting for all SQL accesses to all tables and columns in our databases. And with wise vendor selection, it's possible to enjoy an optimum implementation that includes:

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

Incredibly, some solutions allow all of this to be implemented on a 24x7x365 basis with absolutely *no impact* on the databases being monitored.

Data Piracy, or Data Privacy?

Databases are being stolen, and the greatest risk is from current or former insiders. Protecting sensitive databases from piracy is both good practice and also necessary for regulatory compliance. Conventional safeguards are necessary but not sufficient. To defend against piracy, we must implement comprehensive data access auditing. The best place to do this is *between* all clients and all databases. If we choose our solution wisely, we'll be able to *catch data thieves before it's too late*.

Mr. Richardson is the President and CTO of Ambeo (www.ambeo.com), and has over 24 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.