What Auditors Want - Database Auditing

5 Key Questions Auditors Ask During a Database Compliance Audit

Regulatory legislation is increasingly driving the expansion of formal enterprise audit processes to include information technology (IT) assets. In particular, auditors are looking at regulated data residing in databases connected to enterprise applications such as SAP, Oracle E-Business Suite, PeopleSoft, and other Web Applications. Sarbanes Oxley (SOX), the Health Insurance Portability and Accountability Act (HIPAA) the Payment Card Industry (PCI) standard and other regulatory measures require best practice controls to protect sensitive data.

To verify regulatory compliance, auditors look at multiple aspects of a database environment including: user management, authentication, separation of duties, access control, and audit trail. This paper focuses on the audit trail. The audit trail must meet specific requirements to meet the demands of auditors. In the following, we present five key questions posed by auditors to confirm compliance with best practice and regulatory mandates.

What Questions do Database Auditors Ask?

This paper presents five key questions that IT professionals must answer during a database audit to achieve compliance. These questions are as follows.

1. Is the audit process independent from the database system being audited?
2. Does the audit trail establish user accountability?
3. Does the audit trail include appropriate detail?
4. Does the audit trail identify material variances from baseline activity?
5. Is the scope of the audit trail sufficient?

The answers to these questions vary depending upon the audit mechanism employed. Unfortunately, many database audit mechanisms were not designed to meet the requirements of regulatory auditors and therefore do not adequately address these questions. This paper examines the strengths and weaknesses of alternative audit mechanisms relative to these questions. The goal is to provide the reader with information necessary to make informed choices about which audit mechanisms to deploy to satisfy regulatory compliance audits.

1) Is the Audit Independent?

To ensure audit integrity, the entire process must be independent of the database server and database administrators being audited. Since database administrators and servers are both part of the system being audited, they should not be put in a position of auditing themselves. A rogue administrator, for example, with access to audit records may easily tamper with those records to cover his tracks. Similarly, a non-administrator may exploit a database vulnerability to elevate privileges and tamper with the audit trail. The requirement for independence has three immediate implications for the design of the audit system.

1. **Audit duties should be separate from database administration.** Database administrators may participate in the audit to help interpret events, but their participation should be controlled.
2. **Audit data collection should be independent of native database software capabilities.** Otherwise, a database administrator or non-administrator may tamper with the native audit trail as described above.
3. **External audit solutions may provide independence, but it's important to make sure that it does not rely upon any native database software capabilities.** Some external solutions query native audit mechanisms to collect audit data. As indicated in item 2 above, native audit capabilities are vulnerable to tampering.

SecureSphere - Independent Audit

SecureSphere is a network-based database audit appliance. It automatically creates a detailed audit trail for MS-SQL, Oracle, DB2, and Sybase environments. SecureSphere offers several advantages versus native database audit capabilities including: separation of duties, improved database performance, operational automation, unified audit for heterogeneous database environments, and Web application user audit accountability.

Separation of Duties

SecureSphere maintains complete audit independence both in terms of administrative duties and audit data collection. SecureSphere may be managed by audit staff with complete separation from database administrators. The SecureSphere interface is easily interpreted by auditors with limited database expertise, although role-based administration enables read-only database administrator participation if desired.

Network-Based Audit Data Collection

SecureSphere extract detailed audit information from network traffic traveling to and from a database. It operates in stealth mode (no IP address, etc.) and remains completely invisible to perpetrators. All network activity is tracked and audit records cannot be tampered with regardless of database vulnerability or rogue administrator.

Host-Based Audit Data Collection

The SecureSphere DBA Monitor Security Agent tracks all local database activity on the database server. This includes the database administrator working at the database server console or using ssh (secure shell) or other tool to remotely initiate a user session on the database server. Like the SecureSphere gateway, the SecureSphere agent is independent of native audit capabilities. Installed on the database server as a lightweight host agent, it directly intercepts local activity and forwards a record of that activity to a gateway. Since host-based activity records are securely stored on the gateway, they cannot be tampered with regardless of database vulnerability or rogue administrator. Finally, if the host agent is stopped for any reason, the gateway immediately issues a high priority alert to appropriate staff.

2) Who is Accountable?

The database audit trail must attribute each audited database transaction to specific users. For example, a SOX compliant audit mechanism must log each change to financial reporting data along with the name of the user making the change. However, when users access the database via Web applications (such as SAP, Oracle E-Business Suite, or PeopleSoft), native database software audit logs have no awareness of specific user identities. Therefore, when native audit logs reveal fraudulent database transactions, there is no link to the responsible user.

The problem with auditing Web application access is that the user never directly interacts with the database. Instead, Web applications apply a mechanism known as "pooled connections" to access the database on behalf of the user (
Figure 1). Using pooled connections, the Web application independently authenticates each user and then aggregates all user traffic within a few database connections identified only by the Web application account name. A unique connection for each user is never established and the database receives no information regarding the identity of the actual user.

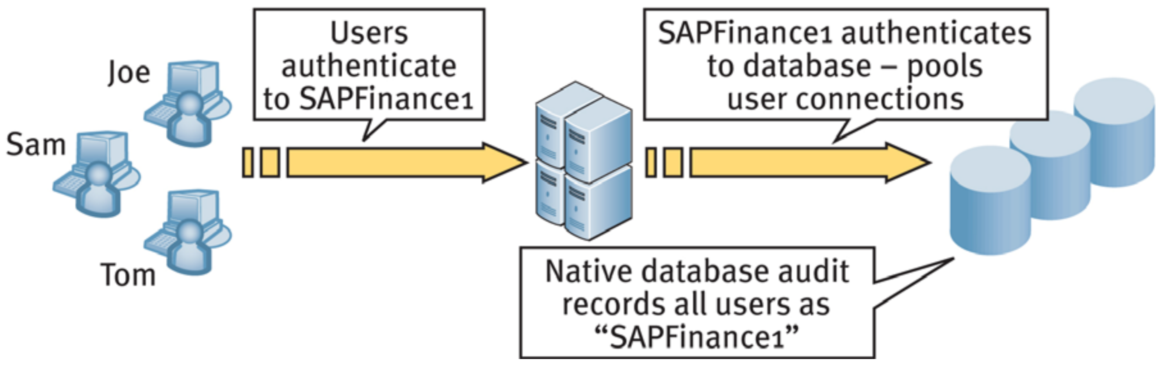
Joe
Sam
Tom
Users authenticate to SAPFinance1
SAPFinance1 authenticates to database – pools user connections
Native database audit records all users as "SAPFinance1"

Figure 1: Native database audit capabilities record application names, not actual user names.

By avoiding the need to authenticate and open unique connections for each user, the Web application realizes significant performance advantage. However, the resulting database audit trail associates all activity with the Web application account name – a clear violation of the auditor's requirement for accountability. What can be done to eliminate this blind spot? There are four options to consider: application rewrites, proprietary database audit mechanisms, Web application audit data, and external database audit devices.

Rewriting the Applications

The most obvious, but perhaps most unrealistic solution to the user accountability problem is to rewrite Web application and database software. There are two general approaches to rewriting applications that might be pursued: unique accounts and embedded references.

Unique Accounts

The first approach we consider involves rewriting Web application software so that they access a unique database account for each user. Doing so enables all standard native database auditing tools to record each transaction with the appropriate user name. There are several problems with this approach.

- Performance - The additional processing overhead required to maintain unique connections for each user dramatically reduces Web application performance. This is the main reason for using pooled connections.
- User Management – Unique connections shifts user management responsibilities to the database administrator. This requires a significant change to organizational responsibilities. Such a shift would require additional database administrators, additional training and new management tools.
- Third-Party Software Limitations - Commercial software packages (SAP, PeopleSoft, etc.), while often customized by the enterprise, rarely have their basic architecture modified to the degree required for this approach. In most cases, the best that can be done is to request that the software vendor add database user audit capabilities in a future release.
- Cost - Code changes and user management changes such as those described above consume significant resources. A significant budget must be allocated to such an undertaking for each application.
- Time - Extensive code and user management changes such as those described above take months to accomplish for a single application. For an organization with tens or hundreds of applications, the transition takes years.
- Risk - Extensive code changes, particularly in the area of user login, introduce significant risk to application availability. Mission critical application changes should be tested extensively before production deployment and monitored closely after deployment.

Embedded References

Clearly, establishing unique connections for each database user is an unattractive option. An alternative approach is to continue to use pooled connections, but to embed a user identity reference within each database request. The database must then be additionally reprogrammed to incorporate the embedded reference information into the audit trail.

The "embedded reference" approach has less impact upon Web application performance and it avoids any shift in user management responsibilities. On the other hand, it requires a potentially more complex rewrite that extends to both Web and database server code. Therefore it retains all other rewrite-related drawbacks described above including: third-party software limitations, cost, time, and risk.

Proprietary Database Solutions

Some database platforms include built-in mechanisms to track application user identities. Unfortunately, these mechanisms still require rewrite of the Web application to include special database requests in all code modules. Therefore, this approach suffers most of the rewrite drawbacks described previously: third-party software limitations, cost, schedule, and risk. In addition, proprietary solutions are (by definition) vendor specific. Therefore, organizations with multiple database vendors will not be able to solve the problem organization-wide, or they will be forced to engineer and support multiple solutions.

Web Application Audit Data

Some organizations may attempt to indirectly satisfy database audit accountability requirements by supplementing database audit data with Web application audit data. For example, if a suspicious database transaction originates from the “application account”, audit personnel may attempt to inspect Web application audit data with similar timestamps to identify the perpetrator. This sounds good, but it’s not easy.

At any given time, hundreds or thousands of users may be logged into an application. Correlating the timestamps of application and database audit data reduces the number of potential perpetrators, but does not usually deliver a definitive result. The auditor must match a specific Web request to a corresponding database transaction by interpreting the details of each transaction (Web parameters, query parameters, etc.). The drawbacks to this process are as follows.

- **Expertise –** The extensive Web application and database design expertise required to carry out the manual correlation described above is not common and very expensive.
- **Cost –** The manual correlation process is extremely laborious and therefore costly even for a single application. Support for this process across many applications and database platforms extend costs further.
- **Uncertain Compliance** - The auditor receives database audit data that is not linked to user identities. The promise that it may be possible to establish identity in the event of suspicious activity may not satisfy auditors.

SecureSphere – Universal User Tracking

SecureSphere’s Universal User Tracking technology delivers audit accountability by simultaneously tracking Web application and database requests. This allows SecureSphere to develop a profile of all database transactions and their corresponding web transactions, essentially reverse engineering the web application. Then in real-time the Web application tracking mechanism captures user login information and tracks that user’s web transactions. This allows SecureSphere to connect every database requests to the associated web transaction and the accountable web user in real-time. Even if there are multiple web users initiating the same web transaction simultaneously SecureSphere can use the details of the transaction (Web parameters, query parameters, etc.) to definitively associate the database transaction with the correct web user.

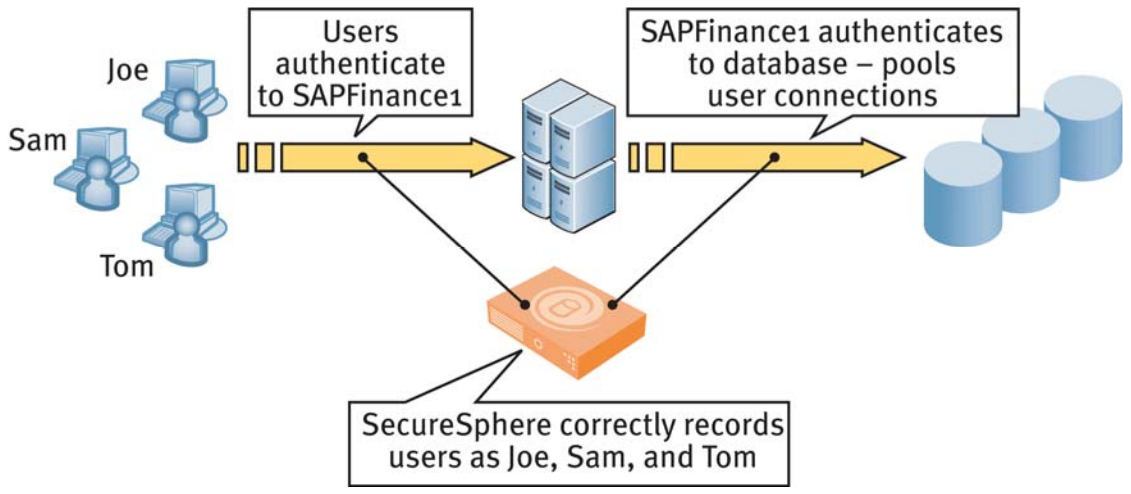
Joe
Sam
Tom
Users authenticate to SAPFinance1
SAPFinance1 authenticates to database – pools user connections
SecureSphere correctly records users as Joe, Sam, and Tom

Figure 2: SecureSphere's Universal User Tracking technology correlates Web application identity information with database activity

The correlation process carried out by SecureSphere is similar to an automated version of the manual postmortem log analysis process described in the previous section (Web Application Audit Information). However, since SecureSphere identifies users in real-time, advanced audit capabilities are enabled that are not possible via postmortem analysis. For example, audit rules may be specified to track access from specific users or groups. Similarly, access to sensitive tables without proper authentication may be detected and investigated.

SecureSphere suffers from none of the drawbacks associated with alternative user accountability approaches described above. It requires no application rewrites, does not impact performance, has no impact on user management, and supports heterogeneous database environments. As a result, SecureSphere can be deployed quickly, at reasonable cost, and without any risk to application availability.

	Performance	User Mgmt.	Cost	Third Party Software Limitations	Risk	Time	Real Time
External Audit Devices (SecureSphere)	No Impact	No Impact	Low	None	None	Short	Yes
App Rewrite - Unique Connections	Negative Impact	Shift to Database Admin	High	Severe	High	Long	Yes
App Rewrite - Embedded Reference	Minimal Impact	No Impact	High	Severe	High	Long	Yes
Proprietary Database Solutions	Minimal Impact	No Impact	High	Severe	High	Long	Yes
Web App Audit Data	No Impact	No Impact	High	None	None	None	No

Table 1: A comparison of alternative approaches to establishing a database audit trail with Web application user accountability

3) Do Audit Records Include Enough Detail?

To effectively reconstruct past database events, auditors require a detailed audit trail that extends to the level of the exact query and response attributes. Consider the following alternative hypothetical audit records for a call center customer service agent named "JOHN".

A. JOHN requested DATA from the CUSTOMER database and the database returned DATA
B. JOHN requested FIRST NAMES, LAST NAMES, EMAIL ADDRESSES, PHONE NUMBERS, and CREDIT CARD NUMBERS for ALL customers from the CUSTOMER database and the database returned 634577 records

Assuming that John is authorized to access individual customer records during the normal course of his work, the first less detailed audit trail **(example A)** does not reveal any unusual activity. However, the second more detailed audit trail **(example B)** makes it clear that a suspicious event has taken place. There is no reason to access the personal information (including credit card numbers) of 634,577 customers. To fully understand the transaction, the audit trail requires complete detail.

Unfortunately, detailed transaction logging can quickly overwhelm processor, disk, and I/O resources available to any audit system. Indeed, the disk space required for native database auditing can easily exceed the space required for actual data storage. Most external audit solutions face the same scalability challenges. To deal with these challenges, many audit systems record necessary detail only in small scale environments. In medium or large scale environments, they record only basic transaction attributes. Even worse, some systems do not record independent events; instead recording only an aggregation of events (e.g. John accessed the CUSTOMER database 10 times). This approach is blind to critical detail. The audit system must record detailed database activity regardless of scale.

SecureSphere – Distributed Audit Architecture

SecureSphere's Distributed Audit Architecture **(Figure 3)** enables detailed data collection while retaining the ability to scale across large data centers. There are three elements to the architecture.

- Multiple SecureSphere Gateways are deployed to scale raw audit computing and storage capacity to required levels.
- The SecureSphere MX Management Server coordinates activity across distributed gateways so that audit staff is presented with a unified view of the data center. The management server effectively enables many gateways to be managed as if they were a single gateway.
- External Storage Systems enables storage volumes exceeding the capacity of the gateways.

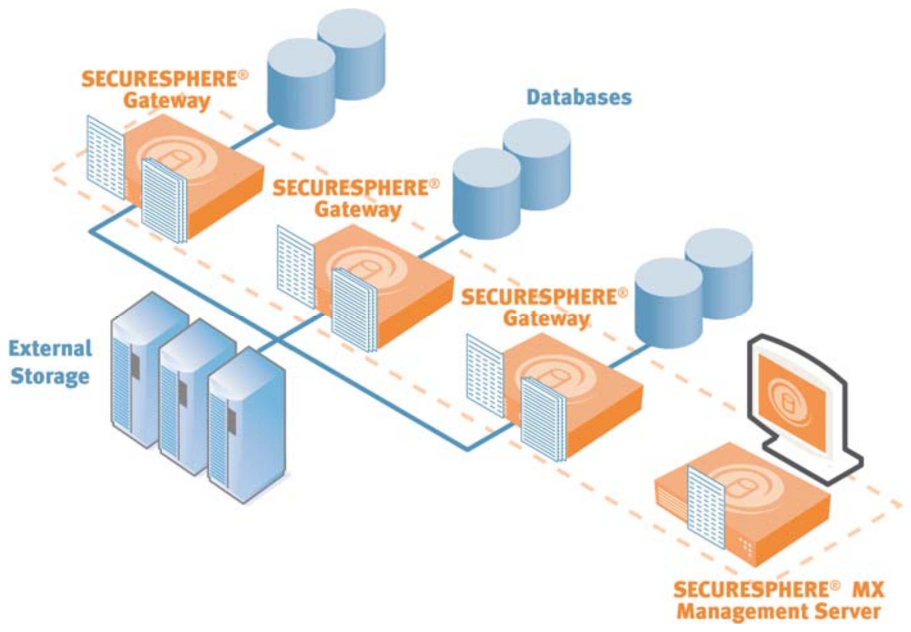

Figure 3: SecureSphere's Distributed Audit Architecture enables detailed data collection while retaining the ability to scale across large data centers.

The Distributed Audit Architecture delivers significant advantages versus alternative approaches.

Versus Native Database Audit Capabilities

Reliance upon native database audit capabilities significantly reduces database performance and useable storage capacity as audit requirements scale. SecureSphere offloads all audit processes from the host database server – actually improving database performance and increasing storage capacity.

Versus other External Audit Solutions

Alternative external audit solutions either recommend limited data collection or require independent management of distributed audit devices for large scale deployments. Limited data collection does not meet audit needs and independent device management dramatically increases administrative cost and reduces overall audit effectiveness. SecureSphere's management server unifies multi-gateway management in large scale deployments.

4) Does the Audit System Identify Material Variances?

It's not enough for the audit system to simply provide a chronological listing of all database transactions. The volume of information generated in most database environments renders such a system useless as a tool for identifying fraudulent activity. An effective audit system should deliver prioritized views of events that separate material variances from legitimate or "baseline" user activity. However, most native and external audit approaches provide un-prioritized views, forcing staff into a costly manual log inspection process.

SecureSphere - Dynamic Profiling

SecureSphere's Dynamic Profiling technology monitors live database traffic to create verified baseline profiles representing each user's normal behavior. Then, by comparing the profile to observed behavior, SecureSphere identifies the material variances from the profile. For example, SecureSphere raises a flag when a DBA with no "business need to know', suddenly retrieves 10,000 customer records.

At the heart of Dynamic Profiling are statistical learning algorithms that filter random events from the profile and enable the system to continuously adapt to legitimate changes over time. Alternative audit systems claim to "learn", but actually provide only a flat recording of all activity over a specified learning period. There are two problems with this simplistic approach used by competitive products.

1. Since competitive products include ALL activity in the baseline profile during a flat recording process, random and possibly deviant events become part of the baseline. SecureSphere learning algorithms, on the other hand, filter random events from the profile. SecureSphere knowledge of database vulnerabilities allows it to filter out deviant events attempting to exploit those vulnerabilities.

2. Once the competitive product specified flat recording period ends, compliance staff must manually and continually update the baseline to reflect changing database activity. If a new baseline is automatically learned at any point, competitive products lose all manual changes that have been made to the baseline. Conversely, SecureSphere learning algorithms never stop functioning. Each profile continuously adapts to behavior changes over time and manual profile modifications can be made at any time.

Since SecureSphere's Dynamic Profiling technology operates continuously, variances may be configured to trigger real-time alerts in addition to the standard audit logging. With real-time alerts, SecureSphere administrators are positioned to immediately respond to serious events when necessary.

5) Is the Scope of the Audit Sufficient?

The scope of the database audit trail should be broad enough to identify any attempt to exploit vulnerability in database platform software (application, operating system, etc.) or protocol implementations. SQL Slammer, Windows RPC vulnerabilities are two examples of the many such vulnerabilities that attackers have exploited to inflict serious damage upon database infrastructure around the world. Dedicated intrusion prevention systems (IPS) and protocol validation solutions are needed to identify such attacks. Therefore, to provide auditors with a complete picture of database activity, it's necessary to integrate data collected from these sources into the audit trail.

SecureSphere – Intrusion Prevention System and Database Protocol Validation

In addition to the detailed transaction logging capabilities described previously, SecureSphere integrates advanced database IPS and protocol validation. Together these technologies provide a comprehensive picture of all database activity.

Database IPS

SecureSphere IPS provides signature-based identification of attacks targeting known vulnerabilities in database platform software. SecureSphere's unique IPS technology is tuned for database deployments by combining a Snort®-compatible signature database with proprietary database-specific signatures developed by Imperva's international security research organization, the Application Defense Center (ADC). The ADC also enhances each signature with contextual attributes such as affected systems, risk, accuracy and attack frequency. These attributes enable users to customize IPS policies to match their specific environment. Finally, Imperva's Data Center Security Update Service provides automatic weekly signature updates ensuring that the most current protections are continuously enforced.

Database Protocol Validation

Attackers may exploit database protocol vulnerabilities to achieve any number of fraudulent objectives: unauthenticated data access, native audit log evasion, etc. To detect this activity, SecureSphere employs the industry's only database protocol validation technology. It incorporates a unique model of database protocol structure, developed by Imperva's Application Defense Center (ADC), which is used as a baseline for comparing ongoing protocol messages to expectations. Any deviation from expectations is logged and presented to auditors in a preconfigured report.

Database protocol validation is unique in the industry for several reasons.

- Database protocols do not conform to any open standard. They are proprietary to each database vendor and their documentation is difficult, if not impossible to obtain. SecureSphere protocol models were obtained only through extensive primary research

- Database protocols are often modified with new database software versions and without public notification. To provide SecureSphere customers with continuous support for all protocols, the Application Defense Center (ADC) continuously tracks and tests each protocol. Any required changes to protocol validation models are updated (along with other monitoring capabilities) as part of the ADC's weekly automated updates delivered to all SecureSphere customers.

As a result of these challenges, no other database monitoring vendor is able to meet the technical and operational challenges associated with protocol validation.

What Does IT Need? – Zero Impact on Data Center

Audit solutions not only need to meet the compliance requirements of auditors, but they must also meet the deployment and resource requirements of IT staff. The ideal audit solution would function with complete transparency to existing IT resources. This concept of transparency is an overriding SecureSphere design criteria. It means that deploying SecureSphere requires no changes to existing data center infrastructure. Moreover, SecureSphere has no impact on database, application or network performance, availability, or reliability. And finally, SecureSphere requires no on-going manual tuning.

Conclusion

Regulatory compliance requirements are expanding formal enterprise audit processes to include IT assets. Auditors are asking IT to prove that sensitive database systems meet best practice management and usage requirements. This paper outlines five critical questions asked of IT during a regulatory compliance audit of sensitive database systems. SecureSphere enables IT organizations to immediately address each of these critical audit questions while minimizing deployment costs or having no impact on mission critical infrastructure. .

iMPERVA®
Data Security for the Data Center™

US Headquarters
950 Tower Lane
Suite 1550
Foster City, CA 94404
Tel: (650) 345-9000
Fax: (650) 345-9004
www.imperva.com

International Headquarters
12 Hachilazon Street
Ramat-Gan 52522
Israel
Tel: +972-3-6120133
Fax: +972-3-7511133