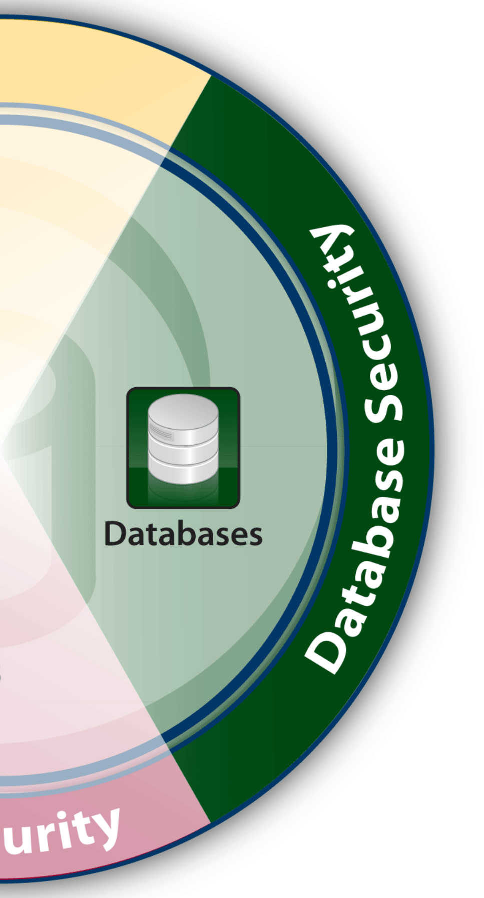
White Paper

Top Ten Database Threats

How to Mitigate the Most Significant Database Vulnerabilities



The enterprise database infrastructure is subject to an overwhelming range of threats. This document is intended to help organizations deal with the most critical of those threats by providing a list of the top ten as identified by Imperva's Application Defense Center. For each threat, the paper describes background information, general risk mitigation strategies, and the database protection provided by Imperva SecureSphere Database Security Solutions.

Top Ten Database Security Threats

1. *Excessive Privilege Abuse*
2. *Legitimate Privilege Abuse*
3. *Privilege Elevation*
4. *Exploitation of vulnerable, mis-configured databases*
5. *SQL Injection*
6. *Weak Audit Trail*
7. *Denial of Service*
8. *Database Communication Protocol Vulnerabilities*
9. *Unauthorized copies of sensitive data*
10. *Backup Data Exposure*

By addressing these top ten threats, organizations will meet global compliance requirements and industry best practices related to data protection and risk mitigation.

IMPERVA®
Protecting the Data That Drives Business®

Threat 1 – Excessive Privilege Abuse

When users (or applications) are granted database access privileges that exceed the requirements of their job function, these privileges may be abused for malicious purpose. For example, a university administrator whose job requires only the ability to change student contact information may take advantage of excessive database update privileges to change grades.

A given database user ends up with excessive privileges for the simple reason that database administrators do not have the time to define and update granular access privilege control mechanisms for each user. As a result, all users or large groups of users are granted generic default access privileges that far exceed specific job requirements.

Preventing Excessive Privilege Abuse – Elimination of Excessive Rights and Enforcement through Query-Level Access Control

The solution to the threat posed by excessive privileges is elimination of any excessive rights. This requires the ability to identify excessive rights – rights that aren't required for the user to perform his job. This is done through extraction of rights form the databases, correlation of rights with the business user and analysis of these rights. This is a daunting process and if it is done manually it requires both time and resources. An automated solution can greatly reduce the time and resource needed and shorten the analysis process.

To better enforce access rights granular query-level access controls are also needed. Query-level access control refers to a mechanism that restricts database privileges to minimum-required SQL operations (SELECT, UPDATE, etc.) and data. The granularity of data access control must extend beyond the table to specific rows and columns within a table. A sufficiently granular query-level access control mechanism would allow the rogue university administrator described previously to update contact information, but issue an alert if he attempts to changes grades. Query-level access control is useful not only for detecting excessive privilege abuse by malicious employees, but also for preventing most of the other top ten threats described herein. Most database software implementations integrate some level of query-level access control (triggers, row level security, etc), but the manual nature of these "built-in" features make them impractical for all but the most limited deployments. The process of manually defining a query-level access control policy for all users across database rows, columns and operations is simply too time consuming. To make matters worse, as user roles change over time, query policies must be updated to reflect those new roles! Most database administrators would have a hard time defining a useful query policy for a handful of users at a single point in time, much less hundreds of users over time. As a result, most organizations provide users with a generic set of excessive access privileges that work for a large number of users. Automated tools are necessary to make real query-level access control a reality.

SecureSphere Dynamic Profiling – User Rights Management and Automated Query Level Access Control

SecureSphere User Rights Management for Databases (URMD) enables automatic aggregation and review of user rights, focused analysis of rights to sensitive data and the identification of excessive rights and dormant users based on organizational context and actual usage.

Access to sensitive objects needs to be granted based on 'Need-To-Know' and is typically defined by the users' organizational context. By adding details such as the user role and department, reviewers have full visibility into the user job function and the type of data he/she can access. URMD's analytical views provide reviewers with the ability to determine if the user access rights are appropriately defined and enable the removal of excessive rights that are not required for the users to do their job.

Using URMD, organizations can demonstrate compliance with regulations such as SOX, PCI 7, and PCI 8.5 and reduce the risk of data breach. URMD is an add-on option to Imperva's Database Security Products.

Imperva SecureSphere Database Security Solutions also provide an automated mechanism for defining and enforcing query-level access control policies. SecureSphere's Dynamic Profiling technology applies automated learning algorithms to create query-level usage profiles for each user and application accessing the database. Each profile extends from general usage patterns to each individual query and stored procedure. SecureSphere's learning algorithms continuously update the profile over time to eliminate manual tuning as user roles change. If any user initiates an action that does not fit their profile, SecureSphere logs the event, issues an alert, and may optionally block the action depending upon severity. The grade-changing university administrator mentioned previously would be easily detected with Dynamic Profiling. The administrator's profile would include a set of queries that reflect normal modifications to specific student contact information and perhaps read-only access to grades. However, a sudden attempt to change grades would trigger an alert.

Threat 2 – Legitimate Privilege Abuse

Users may also abuse legitimate database privileges for unauthorized purposes. Consider a hypothetical rogue healthcare worker with privileges to view individual patient records via a custom Web application. The structure of the Web application normally limits users to viewing an individual patient's healthcare history – multiple records cannot be viewed simultaneously and electronic copies are not allowed. However, the rogue worker may circumvent these limitations by connecting to the database using an alternative client such as MS-Excel. Using MS-Excel and his legitimate login credentials, the worker may retrieve and save all patient records. It is unlikely that such personal copies of patient record databases comply with any healthcare organization's patient data protection policies. There are two risks to consider. The first is the rogue worker who is willing to trade patient records for money. The second (and perhaps more common) is the negligent employee that retrieves and stores large amounts of information to their client machine for legitimate work purposes. Once the data exists on an endpoint machine, it becomes vulnerable to, Trojans, laptop theft, etc.

Preventing Legitimate Privilege Abuse – Understanding the Context of Database Access

The solution to legitimate privilege abuse is database access control that applies not only to specific queries as described above, but to the context surrounding database access. By enforcing policy for client applications, time of day, location, etc., it's possible to identify users who are using legitimate database access privileges in a suspicious manner.

SecureSphere Dynamic Profiling – Context-Based Access Control

In addition to query information (see Excessive Privileges above) SecureSphere's Dynamic Profiling technology automatically creates a model of the context surrounding normal database interactions. Specific contextual information stored in the profile includes time of day, source IP address, volume of data retrieved, application client, etc. Any connection whose context does not match the information stored in the user's profile triggers an alert. For example, the rogue healthcare worker described previously is detected by SecureSphere due not only to nonstandard use of an MS-Excel client, but also due to the volume of data retrieved in a single session. In this specific case, deviations in the structure of the non-standard MS-Excel query would also trigger a query-level violation (see Excessive Privilege abuse above).

Threat 3 – Privilege Elevation

Attackers may take advantage of database platform software vulnerabilities to convert access privileges from those of an ordinary user to those of an administrator. Vulnerabilities may be found in stored procedures, built-in functions, protocol implementations, and even SQL statements. For example, a software developer at a financial institution might take advantage of a vulnerable function to gain the database administrative privilege. With administrative privilege, the rogue developer may turn off audit mechanisms, create bogus accounts, transfer funds, etc.

Preventing Privilege Elevation – IPS and Query Level Access Control

Privilege elevation exploits can be prevented with a combination of traditional intrusion prevention systems (IPS) and query-level access control (see Excessive Privileges above). IPS inspects database traffic to identify patterns which correspond to known vulnerabilities. For example, if a given function is known to be vulnerable, then an IPS may either block all access to the vulnerable procedure, or (if possible) block only those procedures with embedded attacks. Unfortunately, accurately targeting only those database requests with attacks can be difficult using IPS alone. Many vulnerable database functions are commonly used for legitimate purposes. Therefore, blocking all occurrences of such functions is not an option. The IPS must accurately separate legitimate functions from those with embedded attacks. In many cases, the infinite variations in attacks make this impossible. In such cases, IPS systems can be used in alert mode only (no blocking) since false positives are likely. To improve accuracy, IPS may be combined with alternative attack indicators such as query access control. IPS may be used to check whether or not a database request accesses a vulnerable function while query access control detects whether or not the request matches normal user behavior. If a single request indicates access to a vulnerable function and unusual behavior, then an attack is almost certainly in progress.

SecureSphere Privilege Elevation – Integrated IPS and Dynamic Profiling

SecureSphere integrates advanced IPS and Dynamic Profiling for query access control (see Excessive Privileges above). Together, these technologies provide extremely accurate privilege elevation protection. SecureSphere IPS delivers protection against attacks targeting known vulnerabilities with Snort®-compatible signature dictionaries for all protocols. In addition, Imperva's international security research organization, the Application Defense Center, provides proprietary SQL-specific protections to ensure that SecureSphere represents the world's leading database IPS security. The SecureSphere Security Update Service automatically updates all signature dictionaries to ensure that the most current protections are continuously enforced. SecureSphere IPS blocks certain easily identifiable attacks inline without requiring any additional attack confirmations. However, if a given request can be classified as suspicious-only, then SecureSphere correlates the request with related Dynamic Profile violations to validate an attack. To illustrate how SecureSphere integrates IPS and Dynamic Profiling, let's return to the rogue financial services software developer described earlier. Imagine that the developer attempts to take advantage of a known buffer overflow in a database function to insert malicious code to elevate his privileges to those of a database administrator. In this case, SecureSphere identifies two simultaneous violations. First, any query which attempts to access a known vulnerable function triggers an IPS violation. Second, the unusual query triggers a profile violation. By correlating two violations in a single database request from the same user, an attack is validated with extreme accuracy and a high priority alert or blocking action may be issued.

Threat 4 – Exploitation of Vulnerable, Mis-configured Databases

It is common it is to find vulnerable, unpatched databases, or databases that still have default accounts and configuration parameters. An attacker attempting to exploit the database will typically test the systems against these vulnerabilities which may lead to a security breach. As vendors develop a release to patch the systems against a certain vulnerability, the organization's database remains open to exploit. Once a patch is released, it is not readily available. There are different aspects to take into consideration when patching a database. First, the organization must first assess the process of patching the system with the specific patch, understanding how the patch would affect the system. At times a patch may be contradictory to an already existing code, or it may open some work-around. Second, the system suffers from downtime where the database server cannot provide service to users in order to patch it. Finally, large enterprises with dozens and even hundreds of databases must include a patching timeline, prioritizing the databases in the order they should be patched. Thus, it does not come as a surprise that for many organizations, the process of patching lasts a few months – typically between 6-9 months (based on research conducted by the Independent Oracle User Group – IOUG[1]). DBAs, system and IT admins, developers – all these play a role in the patching process. As resources and time are constrained, servers are left vulnerable for months after the release of a patch.

Default accounts and configuration parameters left on a production database might be exploited by an attacker. An attacker can attempt to gain access to the database using a default account. A weakened audit parameter may allow an attacker to bypass audit controls or remove traces of his activities. Weak authentication schemes allow attackers to assume the identity of legitimate database users by stealing or otherwise obtaining login credentials.

Prevention – Vulnerability Assessment and Patching

In order to mitigate the threat of unpatched and vulnerable databases it is first required to assess the security posture of the databases and to close all known vulnerabilities and security gaps. Organizations should scan databases periodically to discover any vulnerabilities and missing patches. Configuration assessments should provide a clear picture of the current configuration state of the data systems. These assessments should also identify databases which aren't compliant with the defined configuration policies. Any missing security patches should be deployed as soon as possible. If a vulnerability is discovered and the patch isn't available yet, either because it hasn't been released by the vendor or because it was not yet deployed, a virtual patching solution should be set. Such a solution blocks attempts to exploit these vulnerabilities. Hence minimizing the window of exposure with virtual patching will protect the database from exploit attempts until the patch is deployed.

SecureSphere Vulnerability Assessments and Virtual Patching

SecureSphere includes a comprehensive vulnerability and configuration assessment solution that lets users schedule periodic scans to find known vulnerabilities, missing patches and mis-configurations. The solution is routinely updated through the automated ADC Updates mechanism with the assessments policies and tests to discover the latest vulnerabilities based on research conducted by Imperva's research center – the Application Defense Center (ADC). SecureSphere also enables users to use virtual patching to block any attempts to exploit vulnerabilities until the patch is deployed. According to a research by the Independent Oracle User Group (IOUG)[1] organizations are typically 6 to 9 months behind on patch deployment. SecureSphere can minimize the risk of exposure during the time required to deploy the patch.

[1] http://ioug.itconvergence.com/pls/apex/f?p=201:1:4201959220925808

Threat 5 – SQL Injection

In a SQL injection attack, a perpetrator typically inserts (or "injects") unauthorized database statements into a vulnerable SQL data channel. Typically targeted data channels include stored procedures and Web application input parameters. These injected statements are then passed to the database where they are executed. Using SQL injection, attackers may gain unrestricted access to an entire database.

Preventing SQL Injection Three techniques can be combined to effectively combat SQL injection: intrusion prevention (IPS), querylevel access control (see Excessive Privilege Abuse), and event correlation. IPS can identify vulnerable stored procedures or SQL injection strings. However, IPS alone is not reliable since SQL injection strings are prone to false positives. Security managers who rely on IPS alone would be bombarded with "possible" SQL injection alerts. However, by correlating a SQL injection signature with another violation such as a query-level access control violation, a real attack can be identified with extreme accuracy. It's unlikely that a SQL injection signature and another violation would appear in the same request during normal business operation.

SecureSphere SQL Injection Protection

SecureSphere integrates Dynamic Profiling, IPS, and Correlated Attack Validation technologies to identify SQL injection with unmatched accuracy.

- Dynamic Profiling delivers query-level access control by automatically creating profiles of each user and application's normal query patterns. Any query (such as a SQL injection attack query) that does not match previously established user or application patterns are immediately identified.
- SecureSphere IPS includes unique database signature dictionaries designed specifically to identify vulnerable stored procedures and SQL injection strings.
- Correlated Attack Validation correlates security violations originating from multiple SecureSphere detection layers. By correlating multiple violations from the same user, SecureSphere is able to detect SQL injection with a degree of accuracy that is not possible using any single detection layer. Consider the stored procedure SQL injection attack shown below.

```
exec ctxsys.driload.validate_stmt (‘grant dba to scott’)
```

In this attack, the attacker (scott) is attempting to grant himself database administrator privileges by embedding a "grant" operation into a vulnerable stored procedure. SecureSphere would handle this attack with one of two processes depending whether or not the stored procedure is part of a required business function.

Vulnerable Stored Procedure Not Required

Ideally vulnerable stored procedures are not used by any users or applications. If this is the case, SecureSphere IPS is sufficient to accurately identify and optionally block all instances of this attack. Normal business activities will not match such a complex character string (…ctxsys.driload…).

Vulnerable Stored Procedure Required

In some cases, a vulnerable stored procedure is part of a required business function. For example, it may be part of a legacy application that cannot be changed. In this case, SecureSphere will first alert security managers to the use of this function. Then, Correlated Attack Validation can be optionally applied to correlate occurrences of this signature with a list of users and applications that are approved to use the procedure. If any unapproved user attempts to use the procedure, SecureSphere can issue an alert or optionally block the request.

Threat 6 – Weak Audit Trail

Automated recording of all sensitive and/or unusual database transactions should be part of the foundation underlying any database deployment. Weak database audit policy represents a serious organizational risk on many levels.

- Regulatory Risk – Organizations with weak (or sometimes non-existent) database audit mechanisms will increasingly find that they are at odds with government regulatory requirements. Sarbanes-Oxley (SOX) in the financial services sector and the Healthcare Information Portability and Accountability Act (HIPAA) in the healthcare sector are just two examples of government regulation with clear database audit requirements.
- Deterrence – Like video cameras recording the faces of individuals entering a bank, database audit mechanisms serves to deter attackers who know that database audit tracking provide investigators with forensics link intruders to a crime.
- Detection and Recovery – Audit mechanisms represent the last line of database defense. If an attacker manages to circumvent other defenses, audit data can identify the existence of a violation after the fact. Audit data may then be used to link a violation to a particular user and/or repair the system. Database software platforms typically integrate basic audit capabilities but they suffer from multiple weaknesses that limit or preclude deployment.
- Lack of User Accountability – When users access the database via Web applications (such as SAP, Oracle E-Business Suite, or PeopleSoft), native audit mechanisms have no awareness of specific user identities. In this case, all user activity is associated with the Web application account name. Therefore, when native audit logs reveal fraudulent database transactions, there is no link to the responsible user.
- Performance Degradation – Native database audit mechanisms are notorious for consuming CPU and disk resources. The performance decline experienced when audit features are enabled forces many organizations to scale back or altogether eliminate auditing.
- Separation of Duties – Users with administrative access (either legitimately or maliciously obtained – see privilege elevation) to the database server can simply turn off auditing to hide fraudulent activity. Audit duties should ideally be separate from both database administrators and the database server platform.
- Limited Granularity – Many native audit mechanisms do not record details necessary to support attack detection, forensics and recovery. For example, database client application, source IP addresses, query response attributes, and failed queries (an important attack reconnaissance indicator) are not recorded by many native mechanisms.
- Proprietary – Audit mechanisms are unique to database server platform – Oracle logs are different from MS-SQL, MS-SQL logs are different form Sybase, etc. For organizations with mixed database environments, this virtually eliminates implementation of uniform, scalable audit processes across the enterprise.

Preventing Weak Audit

Quality network-based audit appliances address most of the weaknesses associated with native audit tools.

- High Performance – Network-based audit appliances can operate at line speed with zero impact on database performance. In fact, by offloading audit processes to network appliances, organizations can expect to improve database performance.
- Separation of Duties – Network-based audit appliances may operate independently of database administers making it possible to separate audit duties from administrative duties as appropriate. In addition, since network devices are independent of the server itself, they are also invulnerable to privilege elevation attacks carried out by non-administrators.
- Cross-Platform Auditing – Network audit appliances typically support all leading database platforms enabling uniform standards and centralized audit operations across large heterogeneous database environments. Together, these attributes reduce database server costs, load-balancing requirements, and administrative costs. They also deliver better security.

SecureSphere Audit Capabilities

In addition to the general advantages associated with network-base audit appliances described above, SecureSphere delivers a series of unique audit capabilities that set it apart for alternative approaches.

- Universal User Tracking makes individual users accountable for their actions – even when they access the database via commercial (Oracle, SAP, PeopleSoft, etc) or custom Web applications. To identify Web application user names, a dedicated SecureSphere interface captures application login information, tracks subsequent Web user sessions, and correlates those with database transactions. The resulting audit logs include unique Web application user names.
- Granular Transaction Tracking supports advanced fraud detection, forensics, and recovery. Log details include details such as source application name, complete query text, query response attributes, source OS, source host name, and much more.
- Distributed Audit Architecture enables granular transaction tracking (see above) while retaining the ability to scale across large data centers. The architecture distributes necessary storage and computing resources across distributed SecureSphere Gateway appliances. The SecureSphere Management Server present audit staff 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 from the perspective of audit staff. Alternative approaches either recommend restricted transaction logging or force administrators to manage many distributed devices independently.
- External Data Archival capabilities automate long term data archival processes. SecureSphere may be configured to periodically archive data to external mass storage systems. Data may be optionally compressed, encrypted, signed prior to archival.
- Integrated Graphical Reporting provides administrators with a flexible and easy-to-understand mechanism for analyzing the audit trail. It includes preconfigured reports that answer common audit questions, while allowing for the creation of customized reports to meet enterprise-specific requirements. Alternatively, any ODBC compliant external reporting package may be used to analyze SecureSphere audit data.
- Local Console Activity Auditing is provided through the SecureSphere Database Agent. The SecureSphere Database Agent is a lightweight host agent installed on the database server to monitor local database administrator activity. Together, the SecureSphere Database Agent and SecureSphere Gateways provide a comprehensive audit trail with negligible impact, or in some cases improving database performance.

Threat 7 – Denial of Service

Denial of Service (DOS) is a general attack category in which access to network applications or data is denied to intended users. Denial of service (DOS) conditions may be created via many techniques – many of which are related to previously mentioned vulnerabilities. For example, DOS may be achieved by taking advantage of a database platform vulnerability to crash a server. Other common DOS techniques include data corruption, network flooding, and server resource overload (memory, CPU, etc.). Resource overload is particularly common in database environments. The motivations behind DOS are similarly divers. DOS attacks are often linked to extortion scams in which a remote attacker will repeatedly crash servers until the victim deposits funds to an international bank account. Alternatively, DOS may be traced to a worm infection. Whatever the source, DOS represents a serious threat for many organizations.

Preventing Denial of Service

DOS prevention requires protections at multiple levels. Network, application, and database level protections are all necessary. This document focuses on database-specific protections. In this database-specific context, deployment of connection rate control, IPS, query access control, and response timing control are recommended.

SecureSphere DOS Protections

- Connection Controls prevents server resource overload by limiting connection rates, query rates, and other variables for each database user.
- IPS and Protocol Validation prevent attackers from exploiting known software vulnerabilities to create DOS. Buffer overflow, for example, is a common platform vulnerability that may be exploited to crash database servers. Refer to the Privilege Elevation and Database Communications Protocol Vulnerabilities sections of this document for more complete descriptions of SecureSphere IPS and Database Communications Protocol Validation technologies.
- Dynamic Profiling automatically provides query access control to detect any unauthorized queries that may lead to DOS. DOS attacks targeting platform vulnerabilities, for example, would be likely to trigger both IPS and Dynamic Profile violations. By correlating these violations, SecureSphere can achieve unmatched accuracy. Refer to the Excessive Privilege Abuse section of this paper for a more complete description of Dynamic Profiling.
- Response Timing – Database DOS attacks designed to overload server resources lead to delayed database responses. SecureSphere's Response Timing feature detects delays in both individual query responses and the overall system.

Threat 8 – Database Communications Protocol Vulnerabilities

A growing number of security vulnerabilities are being identified in the database communication protocols of all database vendors. Four out of seven security fixes in the two most recent IBM DB2 FixPacks address protocol vulnerabilities1. Similarly, 11 out of 23 database vulnerabilities fixed in the most recent Oracle quarterly patch relate to protocols. Fraudulent activity targeting these vulnerabilities can range from unauthorized data access, to data corruption, to denial of service. The SQL Slammer2 worm, for example, took advantage of a flaw in the Microsoft SQL Server protocol to force denial of service. To make matters worse, no record of these fraud vectors will exist in the native audit trail since protocol operations are not covered by native database audit mechanisms. Preventing Database Communication Protocol Attacks Database communication protocol attacks can be defeated with technology commonly referred to as protocol validation. Protocol validation technology essentially parses (disassembles) database traffic and compares it to expectations. In the event that live traffic does not match expectations, alerts or blocking actions may be taken.

SecureSphere Database Communication Protocol Validation

SecureSphere's Database Communication Protocol Validation audits and protects against protocol threats by comparing live database communications protocols to expected protocol structures. No other database security or audit solution provides this capability. It is derived through the Imperva Application Defense Center's (ADC) ongoing research into proprietary database communication protocols and vulnerabilities. Database and application vendors including Oracle, Microsoft, and IBM have credited the ADC with the discovery of serious vulnerabilities and mitigation techniques that have led to increased security in their products. Based upon this research, Imperva is able to incorporate unmatched protocol knowledge into SecureSphere.

Threat 9 – Unauthorized Copies of Sensitive Data

Many companies struggle to locate and accurately maintain an inventory of all their databases. New databases may be created without the security team being aware of them and sensitive data copied into these databases might be exposed if required controls aren't implemented. These "hidden" databases may contain potentially sensitive information such as transaction, customer and employee details but without the responsible people knowing about the content of the databases it is very difficult to ensure that proper controls were implemented. Whether intentionally or not, the fact remains that now sensitive data may be illegally accessed by employees or hackers. Another example is old databases which were forgotten and left out of scope of the assessment. Without anyone managing these databases, data is left unattended to prying eyes that should not be accessing this data.

Prevention – Unauthorized copies of sensitive data

In order to maintain an accurate inventory of databases and location of sensitive data organizations should identify all databases on the network that contain sensitive data. The second step is to find which types of sensitive/classified data are contained within the database objects. Two key challenges to classifying the data are first, finding the sensitive information within the large amount and sizes of tables. Second, finding combinations of data that within itself is considered innocuous, but when combined with other data the combination forms information which is considered sensitive. In order to accurately sensitive information. Once an accurate inventory of databases and location of sensitive data is available, the correct controls should be set in accordance to the data access policies of the organization.

SecureSphere Discovery and Classification

SecureSphere enables users to schedule automated network scans which provide a complete inventory of all databases. It also identifies new or changed databases this is helpful in surfacing any "rogue" databases. Users can then request to scan the content of the databases to identify objects holding sensitive data. SecureSphere is aware out of the box of data types such as credit cards and social security numbers (users can add custom data types as well). In order to reduce false positives SecureSphere uses validation algorithms. It will also highlight any new instances of sensitive/classified data.

Threat 10 – Backup Data Exposure

Backup database storage media is often completely unprotected from attack. As a result, several high profile security breaches have involved theft of database backup tapes and hard disks.

Preventing Backup Data Exposure

All database backups should be encrypted. In fact, some vendors have suggested that future DBMS products may not support the creation of unencrypted backups. Encryption of on-line production database information is often suggested, but performance and cryptographic key management drawbacks often make this impractical and are generally acknowledged to be a poor substitute for granular privilege controls described above.

About Imperva

Imperva is the global leader in data security. Our customers include leading enterprises, government organizations, and managed service providers who rely on Imperva to prevent sensitive data theft by hackers and insiders. The award-winning Imperva SecureSphere is the only solution that delivers full activity monitoring for databases, Web applications and file systems.

To learn more about Imperva's solution visit http://www.imperva.com.

Imperva
Headquarters
3400 Bridge Parkway, Suite 200
Redwood Shores, CA 94065
Tel: +1-650-345-9000
Fax: +1-650-345-9004

Toll Free (U.S. only): +1-866-926-4678
www.imperva.com