Top Ten Database Security 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. Background information, general risk mitigation strategies, and Imperva's SecureSphere Database Security Gateway protections are provided for each threat.

Introduction

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 database vulnerabilities as identified by Imperva's Application Defense Center. Background information, general risk mitigation strategies, and Imperva's SecureSphere Database Security Gateway protections are provided for each threat.

Top Ten Database Security Threats

1. Excessive Privilege Abuse
2. Legitimate Privilege Abuse
3. Privilege Elevation
4. Database Platform Vulnerabilities
5. SQL Injection
6. Weak Audit
7. Denial of Service
8. Database Protocol Vulnerabilities
9. Weak Authentication
10. Backup Data Exposure

By addressing these top ten threats, organizations will meet the compliance and risk mitigation requirements of the most regulated industries in the world.

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 - Query-Level Access Control

The solution to excessive privileges is query-level access control. 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 – Automated Query Level Access Control

The SecureSphere Database Security Gateway provides 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[1], 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 non-standard 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

[1] Available Q2 2006

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 - Platform Vulnerabilities

Vulnerabilities in underlying operating systems (Windows 2000, UNIX, etc.) and additional services installed on a database server may lead to unauthorized access, data corruption, or denial of service. The Blaster Worm, for example, took advantage of a Windows 2000 vulnerability to create denial of service conditions.

Preventing Platform Attacks - Software Updates and Intrusion Prevention

Protection of database assets from platform attacks requires a combination of regular software updates (patches) and Intrusion Prevention Systems (IPS). Vendor provided updates eliminate

vulnerabilities found in database platform over time. Unfortunately, software updates are provided and implemented by enterprises according to periodic cycles. In between update cycles, databases are not protected. In addition, compatibility problems sometimes prevent software updates altogether. To address these problems, IPS should be implemented. As described previously, IPS inspects database traffic and identifies attacks targeting known vulnerabilities.

SecureSphere Platform Protection - IPS

As described previously (see Privilege Elevation above), SecureSphere integrates advanced IPS for protection against database worms and other platform attacks. Imperva's Application Defense Center research organization delivers unique database specific attack protections that ensure the world's most robust database IPS security. In fact, SecureSphere IPS even includes protections against vulnerabilities that have not been made public by database platform vendors and for which fixes are not available.

Threat 5 - SQL Injection

SQL injection is among the most common attack technique applied by the international criminal computer attackers to access databases information from outside perimeter network defenses. In a SQL injection attack, the perpetrator typically inserts unauthorized database queries into vulnerable Web application input forms. These queries are then passed to the database where they are executed with the privileges of the application. Internal users (inside the perimeter firewall) may also use SQL Injection by inserting unauthorized queries into internal applications or database stored procedures. Using SQL injection, attackers may gain unrestricted access to an entire database.

Preventing SQL Injection

Network perimeter firewalls cannot differentiate between normal application traffic and traffic containing unauthorized SQL queries. IPS may be used to indicate SQL injection, but is not reliable alone since SQL Injection IPS protections are prone to false positives. Query access control (see Excessive Privilege Abuse above) is the key to preventing SQL injection. The enforcement of granular, query-level access control policies can detect unauthorized (in this case – "injected") queries originating from any application or user.

SecureSphere SQL Injection Protection

SecureSphere's Dynamic Profiling (see Excessive Privilege Abuse above) automatically delivers query access control. Queries that do not match the previously established user/application patterns are immediately identified. In the specific case of SQL Injection, unauthorized queries inserted into a Web application would obviously deviate from the normal profile of the application. SecureSphere immediately detects such a deviation.

SecureSphere recently enabled an ecommerce retail organization to detect and prosecute a former software development employee who attempted to use SQL Injection to change product prices. With detailed knowledge of the Web application and database infrastructure, the developer knew of a specific shopping cart page which was vulnerable to a SQL injection by inserting unauthorized "UPDATE" queries to change prices. Since the externally facing ecommerce application would never use an UPDATE Query on a database price column under normal circumstances, the unauthorized query immediately triggered a high priority SecureSphere alert. The forensic information provided by SecureSphere logging mechanisms then enabled the firm to track the attempted transaction to the former employee.

Dynamic Profile violations are also correlated with Imperva's advanced database signatures for enhanced accuracy. For example, the "UNION SELECT" query is a common although non-

definitive SQL injection indicator that can be detected via IPS. However, if a query does not match an application's Dynamic Profile and includes the "UNION SELECT" IPS violation, SecureSphere correlates IPS and profile violations to validate with high accuracy that a SQL Injection attack is in progress.

Threat 6 - Weak Audit

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 in the financial services sector and the Healthcare Information Portability and Accountability Act 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 repair a system in the event that denial of service or corruption has occurred.

Database software platforms typically integrate basic audit capabilities but they suffer from multiple weaknesses that limit or preclude deployment.

- **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.

- **Vulnerability to Privilege Abuse –** Users with administrative access (either legitimately or maliciously obtained – see privilege elevation) can simply turn off auditing to hide an attack.

- **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, 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 all of the weaknesses associated with native audit tools. As database independent network devices, audit appliances can operate at line

speed with zero impact on database performance. They operate independently of all users making them invulnerable to privilege attacks. They offer granular data collection for advanced detection, forensics and recovery. Finally, they enable multi-vendor organizations to implement uniform audit standards and centralize audit processes across the enterprise. Together, these attributes reduce the number of database servers deployed, reduce load-balancing requirements, reduce administrative costs, and deliver better security.

SecureSphere Audit Capabilities

SecureSphere delivers database audit capabilities that range from comprehensive to selective tracking of specific sensitive transaction types. Preconfigured and custom reporting is available via an integrated Crystal Reports™ package or any ODBC-compliant database access tool. As a network device, audit data may be collected with complete independence from users - including administrators and developers. Multi-vendor audit logs can be automatically archived to meet future reporting requirements. SecureSphere appliances can be combined with a lightweight host agent for tracking of direct keyboard/terminal activity. Three general audit categories are available.

- **Activity Auditing** enables comprehensive transaction tracking based upon a combination of attributes (username, table, SQL operation, IP address, etc.). Multiple audit logs can be configured and operate in parallel to achieve both comprehensive and selective audits.

- **Real-Time Alert Auditing** presents a prioritized view of potentially dangerous activity in real time.

- **Dynamic Profile Auditing** provides a powerful tool for understanding normal user behavior and comparing it to best practices or regulatory requirements. Audit reports can include current profile data or changes over time.

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 diverse. 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

- **SecureSphere Connection Control**[1] prevents server resource overload by limiting connection rates, query rates, and other variables for each database user.

- **SecureSphere 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. Please refer to the Privilege Elevation and Protocol Vulnerabilities sections of this document for more complete descriptions of SecureSphere IPS and Protocol Validation technologies

- **Dynamic Profiling** (see Excessive Privilege Abuse above) 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.

- **Response Timing**[2] – 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 Protocol Vulnerabilities

Vulnerabilities in database protocols may allow unauthorized data access, corruption, or denial of service. For example, the SQLSlammer[3] worm took advantage of a flaw in the Microsoft SQL Server protocol to force denial of service conditions.

Preventing Database Protocol Attacks

Database 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 Protocol Validation[1]

SecureSphere compares live database protocols to expected protocol structures. Imperva's unique understanding of proprietary database protocols and potential vulnerabilities that exist within these protocols ensure that SecureSphere users are protected by the best available technology. For example, Imperva's Application Defense Center recently discovered protocol vulnerability in Microsoft SQL Server 2000 (http://support.microsoft.com/default.aspx?scid=kb;en-us;910741) that enables a user to mask their login name from the standard SQL Server audit tools. SecureSphere protocol validation protects databases against attacks targeting this and other vulnerabilities.

[1] Available Q2 2006
[2] Available Q2 2006
[3] The SQL slammer worm caused a denial of service on some Internet hosts and dramatically slowed down general Internet traffic on January 25, 2003. It spread rapidly, infecting most of its 75,000 victims within 10 minutes. Source: http://en.wikipedia.org

Threat 9 - Weak Authentication

Weak authentication schemes allow attackers to assume the identity of legitimate database users by stealing or otherwise obtaining login credentials. An attacker may employ any number of strategies to obtain credentials.

- **Brute Force** - The attacker repeatedly enters username/password combinations until he finds one that works. The brute force process may involve simple guesswork or systematic enumeration of all possible username/password combinations. Often an attacker will use automated programs to accelerate the brute force process.

- **Social Engineering –** A scheme in which the attacker takes advantage the natural human tendency to trust to convince others to provide their login credentials. For example, an attacker may present himself via phone as an IT manager and request login credentials for "system maintenance" purposes.

- **Direct Credential Theft –** An attacker may steal login credentials by copying post-it notes, password files, etc.

Preventing Authentication Attacks

Strong Authentication

The strongest practical authentication technologies and policies should be implemented. Two-factor authentication (tokens, certificates, biometrics, etc.) are preferable whenever possible. Unfortunately, cost and ease of use issues often make two-factor authentication impractical. In such cases, strong username/password policy (minimum length, character diversity, obscurity, etc) should be enforced.

Directory Integration

For scalability and ease of use, strong authentication mechanisms should be integrated with enterprise directory infrastructure. Among other things, a directory infrastructure can enable a user to use a single set of log in credentials for multiple databases and applications. This makes two-factor authentication systems more cost effective and/or makes it much easier for users to memorize regularly change passwords.

SecureSphere Authentication Protections

Unfortunately, despite best efforts at strong authentication, breakdowns occasionally occur - password policies are ignored; a lucky attacker may successfully brute force even a reasonably strong password; a legacy authentication scheme may be required for practical reasons; the list goes on. To deal with these situations, SecureSphere's Dynamic Profiling, Failed Login Detection[1], and Authentication Assessment[1] provide broadly applicable authentication protection.

Dynamic Profiling

Dynamic Profiling (see Excessive Privileges and Privilege Abuse above) automatically tracks a range of user attributes that detect compromised login credentials. These attributes include user IP addresses, hostnames[1], operating system username[1], and client application. For example, the previously described attacker who manages to gain login credentials by posing as an IT administrator would trigger multiple SecureSphere alerts when trying to use stolen credentials. The attacker's hostname, operating system username, and possibly even the IP address would not match the profile of the real owner of the compromised login credentials.

[1] Available Q2, 2006

To further illustrate the power of Dynamic Profiling, assume an attacker somehow manages to compromise a user's database credentials and operating system credentials. Further assume the attacker finds a way to also use the victim's actual computer. SecureSphere is still extremely likely to identify the attack! At least two SecureSphere violations come into play.

- **Unauthorized Query -** Attack activity is likely to violate the compromised user's normal usage profile. The attacker may access an unusual table or use an unusual database operation (UPDATE, DELETE, etc).

- **Time of Day –** To gain access to the compromised user's computer, the attacker is likely use the machine at night or during other off-hours. Since the SecureSphere Dynamic Profile includes a model of normal hours, unusual off-hours access will trigger a Time of Day violation.

Failed Login Detection[1]
SecureSphere's Failed Login Detection optionally enforces a failed database login threshold (count and timeframe) to prevent brute force attacks.

Authentication Assessment[2]
SecureSphere's Authentication Assessment proactively detects weak passwords and authentication mis-configuration. Each user password is evaluated to make sure that it meets minimum standards for length, character diversity, etc.

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.

[1] Available Q2, 2006
[2] Available Q2, 2006

Summary

Although databases information is vulnerable to a host of attacks, it is possible to dramatically reduce risk by focusing on the most critical threats. By addressing the top 10 threats outlined above, organizations will meet the compliance and risk mitigation requirements of the most regulated industries in the world.

US Headquarters
950 Tower Lane
Suite 1710
Foster City, CA 94404
Tel: (650) 345-9000
Fax: (650) 345-9004

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