Top Ten Database Security Threats

The Most Significant Risks of 2015 and How to Mitigate Them

Introduction to Database Security Threats

Databases are one of the most compromised assets according to the 2014 Verizon Data Breach Report. The reason databases are targeted so often is quite simple: they are at the heart of any organization, storing customer records and other confidential business data. But why are databases so vulnerable to breaches? One reason is that organizations are not protecting these crucial assets well enough. According to IDC, less than 5% of the $27 billion spent on security products directly addressed data center security[1].

Red Flag

Less than 5% of the $27 billion spent on security products directly addressed data center security.

When hackers and malicious insiders gain access to sensitive data, they can quickly extract value, inflict damage, or impact business operations. In addition to financial loss or reputation damage, breaches can result in regulatory violations, fines, and legal fees. However, the good news is that the vast majority of incidents – more than 97% according to the Online Trust Alliance (OTA) in 2013 – could have been prevented by implementing simple steps and following best practices and internal controls.

The top ten threats outlined in this whitepaper apply not only to traditional databases, but also to Big Data technologies. While Big Data's NoSQL technology is different from SQL, the same injection points – such as input fields – still exist for Big Data. These injection points provide an avenue for attackers to access Big Data components. The Input Injection section in this white paper describes the fundamentals for this type of Big Data attack.

[1] Worldwide Security Products 2011–2014 Forecast (IDC—February 2011)

Top Ten Database Security Threats: 2013 vs. 2015

This white paper highlights the ten most critical database threats as identified by the Imperva Application Defense Center. The same threats we saw in 2013 continue to plague businesses. A notable change is renaming the "SQL Injection" threat to "Input Injection" to reflect its relevance to Big Data technology.

Ranking	2015 Top Threats	2013 Top Threats
1	Excessive and Unused Privileges	Excessive and Unused Privileges
2	Privilege Abuse	Privilege Abuse
3	Input Injection	SQL Injection
4	Malware	Malware
5	Weak Audit Trail	Weak Audit Trail
6	Storage Media Exposure	Storage Media Exposure
7	Exploitation of Vulnerabilities and Misconfigured Databases	Exploitation of Vulnerabilities and Misconfigured Databases
8	Unmanaged Sensitive Data	Unmanaged Sensitive Data
9	Denial of Service	Denial of Service
10	Limited Security Expertise and Education	Limited Security Expertise and Education

By addressing these top ten threats, organizations can meet global compliance requirements and industry best practices related to data protection and risk mitigation. The top nine threats can be addressed by using an automated Database Auditing and Protection (DAP) platform, an approach that improves security, simplifies compliance, and increases operational efficiency.

Top Ten Database Security Threats

1. Excessive and Unused Privileges

When someone is granted database privileges that exceed the requirements of their job function, these privileges can be abused. For example, a bank employee whose job requires the ability to change only account holder contact information may take advantage of excessive database privileges and increase the account balance of a colleague's savings account. Further, when someone changes roles within an organization or leaves it altogether, often his or her access rights to sensitive data do not change. In the latter case, if these workers depart on bad terms, they can use their old privileges to steal high value data or inflict damage.

How do users end up with excessive privileges? Usually, it's because privilege control mechanisms for job roles have not been well defined or maintained. As a result, users may be granted generic or default access privileges that far exceed their specific job requirements, or they may simply accumulate such privileges over time. This creates unnecessary risk.

2. Privilege Abuse

Users may abuse legitimate database privileges for unauthorized purposes. Consider an internal healthcare application used to view individual patient records via a custom web interface. The web application normally limits users to viewing an individual patient's healthcare history – multiple patient records cannot be viewed simultaneously and electronic copies are not allowed. However, a rogue user might be able to circumvent these restrictions by connecting to the database using an alternative client such as MS-Excel. Using Excel and their legitimate login credentials, the user could retrieve and save all patient records to their laptop. Once patient records reach a client machine, the data then becomes susceptible to a wide variety of possible breach scenarios.

3. Input Injection (Formerly SQL Injection)

There are two major types of database injection attacks: 1) SQL Injection that targets traditional database systems and 2) NoSQL Injection that targets Big Data platforms. SQL Injection attacks usually involve inserting (or "injecting") unauthorized or malicious statements into the input fields of web applications. On the other hand, NoSQL injection attacks involve inserting malicious statements into Big Data components (e.g., Hive or MapReduce). In both types, a successful Input Injection attack can give an attacker unrestricted access to an entire database.

A crucial point to realize here, is that although it is technically true that Big Data solutions are impervious to SQL Injection attacks – because they don't actually use any SQL-based technology – they are, in fact, still susceptible to the same fundamental class of attack (i.e., Input Injection).

4. Malware

Cybercriminals, state-sponsored hackers, and spies use advanced attacks that blend multiple tactics – such as spear phishing emails and malware – to penetrate organizations and steal sensitive data. Unaware that malware has infected their device, legitimate users become a conduit for these groups to access your networks and sensitive data.

5. Weak Audit Trail

Automated recording of database transactions involving sensitive data should be part of any database deployment. Failure to collect detailed audit records of database activity represents a serious organizational risk on many levels.

Organizations with weak (or sometimes non-existent) database audit mechanisms will increasingly find that they are at odds with industry and government regulatory requirements. For example, Sarbanes-Oxley (SOX), which protects against accounting errors and fraudulent practices, and the Healthcare Information Portability and Accountability Act (HIPAA) in the healthcare sector, are just two examples of regulations with clear database audit requirements.

Many enterprises will turn to native audit tools provided by their database vendors or rely on ad-hoc and manual solutions. These approaches do not record details necessary to support auditing, attack detection, and forensics. Furthermore, native database audit mechanisms are notorious for consuming CPU and disk resources forcing many organizations to scale back or eliminate auditing altogether. Finally, most native audit mechanisms are unique to a database server platform. For example, Oracle logs are different from MS-SQL, and MS-SQL logs are different form DB2. For organizations with heterogeneous database environments, this imposes a significant obstacle to implementing uniform, scalable audit processes.

When users access the database via enterprise web applications (such as SAP, Oracle E-Business Suite, or PeopleSoft) it can be challenging to understand which database access activity relates to a specific user. Most audit mechanisms have no awareness of who the end user is because all activity is associated with the web application account name. Reporting, visibility, and forensic analysis are hampered because there is no link to the responsible user.

Finally, users with administrative access to the database, either legitimately or maliciously obtained, can turn off native database auditing to hide fraudulent activity. Audit capabilities and responsibilities should ideally be separate from both database administrators and the database server platform to ensure strong separation of duties policies.

6. Storage Media Exposure

Backup storage media is often completely unprotected from attack. As a result, numerous security breaches have involved the theft of database backup disks and tapes. Furthermore, failure to audit and monitor the activities of administrators who have low-level access to sensitive information can put your data at risk. Taking the appropriate measures to protect backup copies of sensitive data and monitor your most highly privileged users is not only a data security best practice, but also mandated by many regulations.

7. Exploitation of Vulnerable, Misconfigured Databases

It is common to find vulnerable and un-patched databases, or discover databases that still have default accounts and configuration parameters. Attackers know how to exploit these vulnerabilities to launch attacks against your organization. Unfortunately, organizations often struggle to stay on top of maintaining database configurations even when patches are available. Typical issues include high workloads and mounting backlogs for the associated database administrators, complex and time-consuming requirements for testing patches, and the challenge of finding a maintenance window to take down and work on what is often classified as a business-critical system. The net result is that it generally takes organizations months to patch databases, during which time they remain vulnerable.

According to the 2014 Independent Oracle User Group (IOUG) Enterprise Data Security Survey, 36 percent of Oracle users take more than six months to apply a Critical Patch Update, while another 8 percent have never applied one.[2]

8. Unmanaged Sensitive Data

Many companies struggle to maintain an accurate inventory of their databases and the critical data objects contained within them. Forgotten databases may contain sensitive information, and new databases can emerge – e.g., in application testing environments – without visibility to the security team. Sensitive data in these databases will be exposed to threats if the required controls and permissions are not implemented.

9. 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. DoS conditions can be created via many techniques. The most common technique used in database environments is to overload server resources such as memory and CPU either by flooding them with an excessive number of queries, or with a smaller volume of well-crafted queries that consume a disproportionate amount of system resources (e.g., because they lead to recursive look-ups or table operations). The result in either case is the same; the resource-starved servers become unresponsive and, in some instances, even crash. The motivations behind DoS attacks are often linked to extortion scams in which a remote attacker will repeatedly crash servers until the victim meets their demands. Whatever the source, DoS represents a serious threat for many organizations.

10. Limited Security Expertise and Education

Internal security controls are not keeping pace with data growth and many organizations are ill-equipped to deal with a security breach. Often this is due to the lack of expertise required to implement security controls, enforce policies, or conduct incident response processes.

According to the Ponemon Institute 2014 Cost of Data Breach Study, for 30 percent of data breach incidents, the main root cause was classified as the "human factor" – in other words, a negligent employee or contractor.[3]

[2]2014 IOUG Enterprise Data Security Survey, Unisphere Research, a Division of Information Today, Inc. October 2014
[3] 2014 Cost of Dat Breach Study: Global Analysis, Ponemon Institute, May 2014

Multi-layered Database Security Defense Strategy

As mentioned in the first section of this paper, the top ten database security threats can be prevented by implementing simple steps and following best practices and internal controls. Because there are many different attack vectors associated with each threat, a multi-layered defensive strategy is needed to properly protect databases. The matrix below identifies solutions for each of the top ten database threats. Detailed solution descriptions are presented following the matrix.

	Threat / Solution	Excessive and Unused Privileges	Privilege Abuse	Input Injection	Malware	Weak Audit Trail	Storage Exposure	Vulnerability Exploitation	Unmanaged Sensitive Data	Denial of Service	Limited Security Knowledge
Discovery and Assessment	Scan for Vulnerabilities			•	•					•	
	Calculate Risk Scores			•				•			
	Mitigate Vulnerabilities			•				•		•	
	Identify Compromised Endpoints				•			•			
	Analyze Risk and Prioritize Remediation Efforts							•			
	Discover Database Servers								•		
	Analyze Discovery Results								•		
	Identify and Classify Sensitive Data	•							•		
User Rights Management	Aggregate Access Rights	•	•								
	Enrich Access Rights Information	•	•								
	Identify and Remove Excessive Rights	•	•		•						
	Review and Approve/Reject Individual User Rights	•									
	Extract "Real" User Identity					•					
Monitoring and Blocking	Real-Time Alerting and Blocking	•	•	•						•	
	Detect Unusual Access Activity	•	•	•	•					•	
	Block Malicious Web Requests			•							
	Monitor Local Database Activity					•					
	Impose Connection Controls									•	
	Validate Database Protocols									•	
	Response Timing									•	
Auditing	Automate Auditing with a DAP Platform					•					
	Capture Detailed Transactions					•					
	Generate Reports for Compliance and Forensics					•					
Data Protection	Archive External Data					•					
	Encrypt Databases						•				
Non-Tech. Security	Cultivate Experienced Security Professionals										•
	Educate Your Workforce										•

Database Security Solutions Defined

There are six different categories of solutions in the matrix above that align with your organization's compliance and security objectives.

- **Discovery and Assessment** locate where database vulnerabilities and critical data reside.
- **User Rights Management** identifies excessive rights over sensitive data.
- **Monitoring and Blocking** protect databases from attacks, unauthorized access, and theft of data.
- **Auditing** helps demonstrate compliance with industry regulations.
- **Data Protection** ensures data integrity and confidentiality.
- **Non-Technical Security** instills and reinforces a culture of security awareness and preparedness.

Discovery and Assessment

Scan for Vulnerabilities: Understanding vulnerabilities that expose databases to input injection is essential. Malware may be looking to exploit known database vulnerabilities, making un-patched databases an easy target. Weak authentication rules can enable an application-layer DoS attack by granting access to a database without needing a password. Use vulnerability assessment tools to detect security vulnerabilities, misconfigurations, and missing vendor patches. Assessments should use industry best practices for database security, such as DISA STIG and CIS benchmarks.

Calculate Risk Scores: Score risks based on the severity of vulnerabilities and the sensitivity of the data. Severity values should be based on known systems such as the Common Vulnerability Scoring System (CVSS). Risk scores help prioritize risk, manage, and research vulnerabilities. In this case, higher risk scores would relate to input injection.

Mitigate Vulnerabilities: If vulnerability is discovered and the database vendor hasn't released a patch, a virtual patching solution should be used. Applying virtual patches will block attempts to exploit vulnerabilities without requiring actual patches or changes to the current configuration of the server. Virtual patching will protect the database from exploit attempts until the patch is deployed. Again, focus on patching high-risk vulnerabilities that can facilitate a DoS or input injection attack.

Identify Compromised Endpoints: Identify malware-infected hosts so that you can prevent these devices from accessing sensitive information in databases as well as unstructured data stores. Once you identify compromised devices, you should apply controls to sensitive data to restrict those devices from accessing and exfiltrating data.

Analyze Risk and Prioritize Remediation Efforts: Use reports and analytical tools to understand risks and help prioritize remediation efforts.

Discover Database Servers: In order to build and maintain an inventory of databases and isolate sensitive data contained within them, organizations should first catalog all databases in their data centers. Leverage discovery tools that scan enterprise networks and identify active database services. Look for solutions that can reduce scan duration by filtering on IP addresses and ranges and by specific database services (e.g. Oracle, Microsoft SQL, IBM DB2, etc.). Periodically re-run discovery scans to identify new or changed databases.

Analyze Discovery Results: Review database discovery and classification results to determine which databases that store sensitive data need to be monitored.

Identify and Classify Sensitive Data: Once you have constructed a catalog of databases, it is critical to understand which databases contain sensitive data. Scan the objects, rows, and columns of databases to pinpoint sensitive data. Use data classification solutions that are aware of data types such as credit cards, email addresses, and national identity numbers, and which enable users to add custom data types as well. Classification results should include the IP address and host name of the asset, and indicate the existence of sensitive data on that server. Automatically identifying sensitive data and personally identifiable information helps narrow the scope of security and compliance efforts.

User Rights Management

Aggregate Access Rights: Scan databases for both granted and privileged user rights and extract details such as the actual access right (e.g. SELECT, DELETE, CONNECT, etc), who granted them, who received those rights, and objects to which rights have been granted. Aggregating user rights into a single repository helps streamline the reporting and analysis of user access to sensitive data.

Enrich Access Rights Information with User Details and Data Sensitivity: Adding information related to user roles and their database behavior adds considerable value to user rights analysis and helps zero-in on the abuse of privileges. Collect and append contextual details to user rights information including the user name, department, database object sensitivity, and last time accessed. This allows you to focus your analysis on the access rights that represent the highest business risk.

Identify and Remove Excessive Rights and Dormant Users: Identify users that have too many privileges and users who don't use their privileges. This helps determine if user access rights are appropriately defined, find separation of duties issues, and remove excessive rights that are not required for users to do their job. Hackers use access rights to impersonate users and go after sensitive data stores. Therefore, reducing excessive rights helps protect against malware compromise and targeted attacks.

Review and Approve/Reject Individual User Rights: Perform an organized review of user rights to determine if they are appropriate. Reviewers should approve or reject rights, or assign them to another for review, and administrators can report on the review process. Conducting organized user rights reviews meets regulatory requirements and reduces risk by ensuring that user privileges are granted on a need-to-know basis.

Extract "Real" User Identity: Leverage solutions that correlate user information with database transactions, also known as Universal User Tracking, or UUT. The resulting audit logs can then include unique application user names.

Monitoring and Blocking

Real-Time Alerting and Blocking: Monitor all database access activity and usage patterns in real time to detect data leakage, unauthorized SQL and Big Data transactions, and protocol and system attacks. When attempts to access unauthorized data occur, generate alerts or terminate the user session. Use a solution that leverages policies – both pre-defined and custom – that inspect database traffic to identify patterns that correspond to known attacks, such as DoS attacks, and unauthorized activities. Security policies are useful for not only detecting excessive privilege abuse by malicious, compromised, or dormant users, but also for preventing most of the other top ten database threats.

Detect Unusual Access Activity: Establish a comprehensive profile of each database user's normal activity. Monitoring for deviations from these baselines enables detection of DoS, malware, input injection, and anomalous activities. If any user initiates an action that does not fit their profile, log the event, generate an alert or block the user. Creating activity-based user profiles increases the likelihood of detecting inappropriate access to sensitive data.

Block Malicious Web Requests: Because web applications are the most common vector for initiating an input injection attack, another important line of defense will be your Web Application Firewall (WAF). A WAF will recognize and block input injection attack patterns that originate from web applications.

To protect against Input Injection attacks, a WAF should:

- Inspect HTTP parameter values for special characters like apostrophes and brackets and know whether these characters are expected or indicative of an attack.
- Use application signatures and policies of known input injection patterns to alert and block.

Monitor Local Database Activity: DAP solutions can audit and monitor the activities of your most highly privileged users – database and system administrators. These users have been granted the highest levels of access to your databases and, therefore, require close attention. Should they abuse their privileges or become compromised by malware, the risk of data theft and damage to your organization increases.

Impose Connection Controls: Prevent server resource overload by limiting connection rates, query rates, and other variables for each database user.

Validate Database Protocols: Leverage database activity monitoring solutions that can analyze the protocol and isolate anomalous communications. When atypical communication events are detected, the solution should trigger an alert or block the transaction.

Response Timing: Database DoS attacks designed to overload server resources lead to delayed database responses. This includes delays in both individual query responses and the overall system. Use solutions that monitor response timing and generate alerts when response delays or system sluggishness is observed.

Auditing

Automate Auditing with a DAP Platform: Implement a DAP solution that delivers the performance, scalability, and flexibility to meet the needs of the most demanding environments. A DAP solution can address most of the weaknesses associated with native audit tools:

- **Separation of Duties:** DAP solutions operate independently of database administrators, making it possible to separate audit duties from routine system administration. In addition, they operate independently of the database server and are invulnerable to privilege elevation attacks carried out by non-administrators.
- **Cross-Platform Auditing:** DAP solutions support database platforms from multiple vendors enabling uniform standards and centralized audit operations across large and distributed heterogeneous database environments.
- **Performance:** Leading DAP solutions can leverage high performance appliances that have zero impact on database performance. In fact, by offloading audit processes to network appliances rather than using native auditing, organizations can expect to improve database performance.

Capture Detailed Transactions: To support regulatory compliance requirements, advanced fraud detection, and forensic analysis, DAP solutions can capture audit logs that include details such as source application name, complete query text, query response attributes, source OS, source host name, and more. Use auditing rules to collect the required information needed for regulatory compliance (e.g. SOX, PCI DSS, and HIPPA) or to meet internal audit requirements.

Generate Reports for Compliance and Forensics: Summarize and format database activity details into reports that help meet compliance requirements, conduct forensic investigations, communicate vital database activity statistics, and monitor system performance. Leverage DAP solutions that include reports for industry and government regulations which can be customized to meet business needs.

Data Protection

Archive External Data: Automate the long-term data archival processes. Use solutions that can be configured to periodically archive data to external mass storage systems. Data should be optionally compressed, encrypted, and signed prior to archival.

Encrypt Databases: Encrypt sensitive data across heterogeneous database environments. This allows you to secure both production and backup copies of databases, then audit the activity of and control access to sensitive data from users who access databases at the operating system and storage tiers. By leveraging database auditing along with encryption, organizations can monitor and control users both inside and outside of the database.

Non-Technical Security

Cultivate Experienced Security Professionals: To defend against a growing array of internal and external threats, hire information security personnel that are well versed in IT Security and have experience implementing, administering, and monitoring security solutions. Ongoing education and training are also important for growing deeper security knowledge and skills. Consider outside IT security and specialists to help with implementation, conduct security assessments and penetration tests, and provide training and support for your administrators.

Educate Your Workforce: Train your workforce on risk mitigation techniques including how to recognize common cyberthreats (e.g. a spear-phishing attack), best practices around Internet and email usage, and password management. Failure to enforce training and create a "security conscious" work culture increases the chances of a security breach. The end result is well-informed users who are trained to securely function when connected to key systems.

Video [2:28]

Data Theft Prevention Case Study

An electronic payment processor was auditing their database for PCI compliance and discovered that ATM and PIN numbers were being stolen. Detailed logs from Imperva SecureSphere Database Activity Monitoring helped track down and apprehend the criminals. The company now generates alerts on suspicious database access to sensitive data.

View Video

Summary

Failing to safeguard databases that store sensitive data can cripple your operations, result in regulatory violations, and destroy your brand. Understanding the top database threats and implementing the solutions outlined in this paper will enable you to recognize when you're vulnerable or being attacked, maintain security best practices, and ensure that your most valuable assets are protected.

About Imperva

Imperva® (NYSE: IMPV), is a leading provider of cyber security solutions that protect business-critical data and applications. The company's SecureSphere™, Incapsula™ and Skyfence™ product lines enable organizations to discover assets and vulnerabilities, protect information wherever it lives - on-premises and in the cloud - and comply with regulations. The Imperva Application Defense Center, a research team comprised of some of the world's leading experts in data and application security, continually enhances Imperva products with up-to-the minute threat intelligence, and publish reports that provide insight and guidance on the latest threats and how to mitigate them. Imperva is headquartered in Redwood Shores, California.

IMPERVA®

www.imperva.com

 WP-TOP10-DATABASE-THREATS-0415rev4