IMPERVA®

# Top 5 Database Security Threats

WHITEPAPER

## Introduction to Database Security Threats

Data breaches are a threat to every organization. Breach damage goes beyond the actual loss or disclosure of sensitive, confidential data and brand damage, companies incur significant financial costs associated with remediation and multi-year legal liability claims. Risk sensitive organizations must remain a step ahead in their database security to protect and defend their data from a myriad of external and internal threats.

## What Makes Your Data a Prime Target?

According to the Verizon 2016 DBIR report, hackers are motivated by financial gain, espionage, ideology or grudge, and even fun, though financial gain is the primary object for breaching commercial entities. Most predators attack the weakest prey via the path of least resistance. The good news is that this means your security does not need to be perfect–it just has to be sufficient to deter the predator so they go elsewhere to find easier prey.

> *"You don't have to run faster than the bear to get away. You just have to run faster than the guy next to you."*
>
> JIM BUTCHER

The bad news is many companies struggle to implement a multi-layered security approach that detects, monitors, prevents, and mitigates threats.

In this paper, we'll discuss the top five database security threats to relational databases. We'll also explore the need to secure Big Data which is often the repository of choice for business analytics and customer experience applications that rely on sensitive data.

## What are the Top 5 Database Security Threats?

1. Excessive, inappropriate, and unused privileges
2. Privilege abuse
3. Insufficient web application security
4. Weak audit trails
5. Unsecured storage media

The top two threats can be directly attributed to an increase in insider threats. Typically, the enterprise network is considered fairly protected by next-gen firewalls that protect the perimeter. However, once a bad actor gets past the firewall, there is no protection mechanism in most enterprises that can detect lateral movement and prevent major data breaches. This poses a major hazard to data. Additionally, external threats are constant and insufficient internal processes leave security gaps, which is why today's security best practices dictate that organizations must take a multi-layered, multi-front approach to effectively secure data and prevent breaches.

That said, let's explore the top five threats in detail.

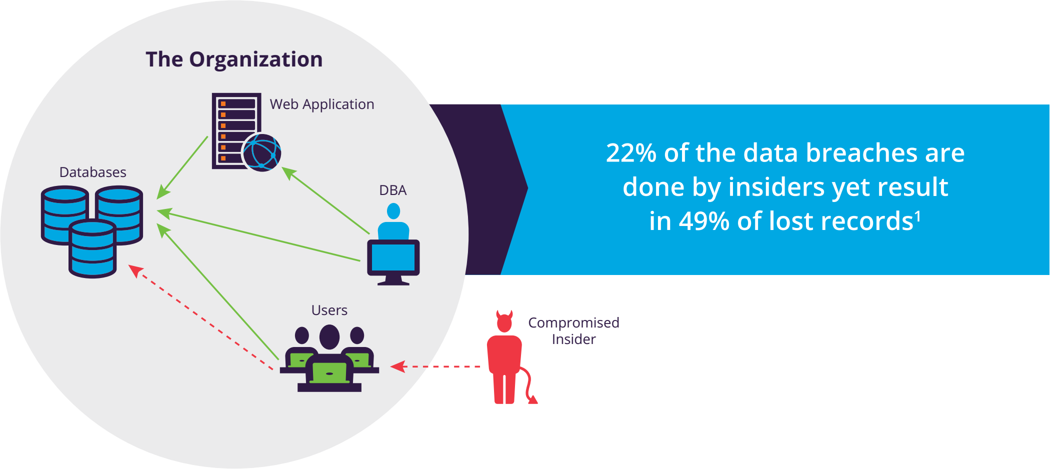
## 1. Excessive, Inappropriate and Unused Privileges

When you grant someone database privileges that exceed the requirements of their job function, these privileges can be abused. For example, a HR employee whose job requires the ability to update employee time-off information may take advantage of excessive database privileges and perform an unauthorized lookup of the salary data of peers or executives. Further, when someone changes roles within an organization, often his or her access rights to the sensitive data are not updated to remove rights no longer necessary for their new role.

**47% of companies report users have excessive rights**

The complexity of applications and the corresponding data structures used mean that administrators are inclined to grant excessive privileges by default just to avoid the risk of application failure due to lack of access privileges. Thus, 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. Usually, enterprises protect or "harden" the devices of employees in high positions (like the CEO, CFO etc.) from external (and internal) attackers to protect the vast access to sensitive data these users require. This hardening facilitates the detection of a compromising situation, termination of access and potential destruction of locally stored data. However, this is not a scalable or viable solution in a BYOD world. When a device of an average user is compromised, it will likely be harder to detect, and if this user has excessive privileges it can create a breach leading to a massive data loss incident.

An efficient method to pinpoint excessive and unused privileges in the organization is to use database assessment and database monitoring with a user rights management option.



**22% of the data breaches are done by insiders yet result in 49% of lost records[1]**

Figure 1: An attacker compromised an employee devise and uses the excessive privileges of this employee to access the enterprise database(s).

[1] 2015 Data Breach Trends, Data Breach Quick View, January 2016

## 2. Privilege Abuse

Imperva research that includes data from multiple organizations over a two-year period indicates that in every organization humans used database service accounts to access databases, and that these users were misusing these privileged service accounts to access sensitive data directly, bypassing the application interface.

In addition, certain "Privileged Users" may abuse legitimate database privileges for unauthorized purposes. Certain user groups in the organization have privileges to access entire databases due to their occupation and activities. The two main categories of privileged users are database systems administrators and developers:

- Database systems administrators (DBAs) have unlimited access to all data in the database. For the best security, DBAs should not access application data in the database (applicative data/tables) directly when they are administrating the database. When a DBA accesses the applicative data directly through the database instead of the application interface, he bypasses the application logging and retrieval limitations and avoids the application permissions and security mechanism.

One Imperva client using the CounterBreach solution was alerted to the fact that a trusted DBA had accessed sensitive data in tables of a PeopleSoft application, directly through the database, and not through the PeopleSoft interface. These tables contained financial information that the DBA should not have been accessing. This discovery clearly illustrates an insider threat risk.

- Developers often have full access to production databases. The QA teams snapshot the databases for testing while engineers may debug live production systems. In both scenarios, sensitive data is vulnerable to privilege abuse.

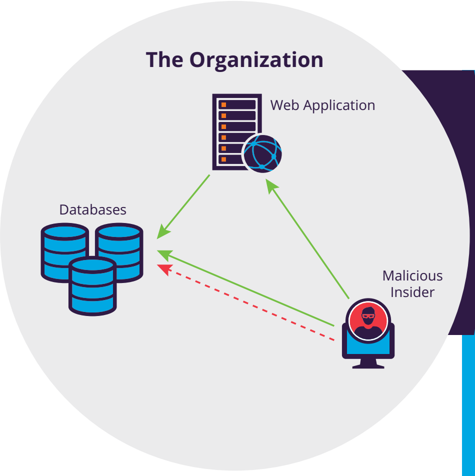


Figure 2: A DBA decides to act as a malicious insider and uses his privileges to access and exfiltrate applicative data directly from the database, overcoming the application permission mechanism.

**Of the insider threats detected in 2016, more than 65% of the data breach losses were attributed to privileges abuse. The challenge of detecting privilege abuse can be monumental without the aid of a coordinated user behavior analysis system and database activity monitoring and blocking solution.**

## What are Insider Threats?

Insider threat can be categorized into three profiles –malicious, negligent, and compromised:

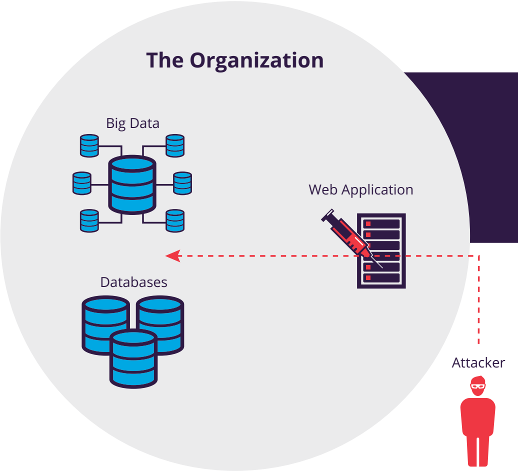
- **Malicious** insider threats come from people within or directly associated with the organization (e.g., employees, former employees, contractors, business associates) who have inside information concerning the organization's security practices, data, and computer systems. The 2016 Insider Threat Spotlight Report presented by Palerra indicates that on average one in every 50 users is a malicious user.
- **Negligent** insiders are people within or directly associated with the organization that don't have malicious intentions, but they expose sensitive data to data breaches, due to careless behavior.
- **Compromised** users fall victim to "outside" malicious attackers that exploit or take over control of systems of the organization. Outside attackers can use a variety of techniques to attack the organization, including using direct attacks, computer viruses, social engineering techniques, phishing, and other evolving techniques. The Verizon DBIR indicates that one in every six users will misuse or expose data.

For more information on the insider threat risk, please review the Imperva HII research report: Insiders: The Threat Is Already Within

## 3. Insufficient Web Application Security

Most organizations rely heavily on apps to interface with customers. There are many types of attacks on applications that can expose data. Two common types of web application attacks that target databases are SQL Injection and Web Shell.

SQL Injection (SQLi) attacks have been a top threat on the Verizon DBIR report for multiple years. SQLi attacks are a result of incomplete or inadequate input validation that allows bad actors to pass SQL commands to the database via the web application in a manner that was never anticipated.



**That SQL Injections target traditional databases is a well understood threat. However, since some Big Data solutions also provide SQL like interfaces, SQL Injection is also a threat for Big Data. Unfortunately, most security professionals are unaware of this threat for Big Data.**

Figure 3: An external attacker or a malicious insider succeeds in accessing the database or Big Data instance using a code injection to the application.

Web Shell attacks are a stealthy method used to gain unauthorized remote access to a server. Web shells are backdoors that utilize the web server core functionality (serving remote clients) to gain persistent remote access and obtain full or limited control over the server through an interface to the server's shell. According the Verizon DBIR the number of web app attack breaches caused by Web Shell backdoors is second only to stolen credentials.

Web shells can use the shell's capabilities to compromise the organization databases and exfiltrate data without detection. The attacker uses the shell's file browsing capabilities to locate and steal the database credentials used by the legitimate application from the application's configuration file. This is made possible by the fact that the shell inherently possesses the OS privileges of the server application/daemon itself. Furthermore, in some applications, the DB credentials (username and password) are stored in plaintext in a documented file location.

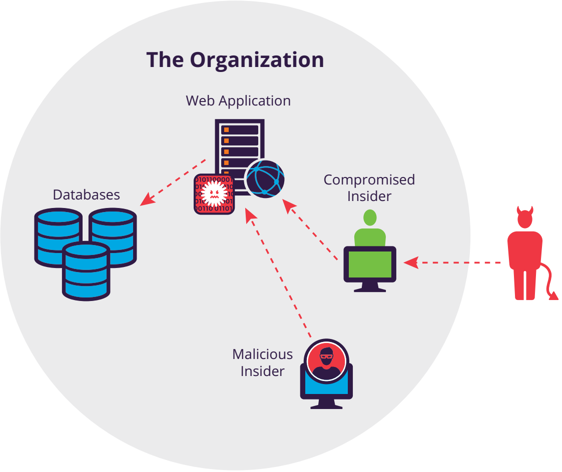


Figure 4: An external attacker or a malicious insider succeeds in accessing the database using a web shell disguised as the web application.

## 4. Weak Audit Trail

Here we move into threats caused by insufficient internal processes or gaps. Monitoring data access across the enterprise should be part of any production database deployment. The failure to monitor for both security and compliance anomalies and collect the appropriate audit details of database activity represents a serious organizational risk at many levels.

Additionally, organizations with weak (or sometimes non-existent) database audit mechanisms also find that they are at odds with industry and government regulatory requirements. Sarbanes-Oxley (SOX), which seeks to prevent accounting errors and fraudulent practices, and the Healthcare Information Portability and Accountability Act (HIPAA) in the healthcare sector, are examples of regulations with clear database audit requirements. The newly introduced General Data Protection Regulation (GDPR) for the European Union (EU) is the first regulation that imposes debilitating fines for enterprise that fail to meet the stringent data protection measures that include a database monitoring capability sufficient to meet the audit and breach notification requirements for all personal data.

### Why are Audit Trails So Challenging?

The first reason is that many enterprises turn to native audit tools provided by their database vendors or rely on ad-hoc and manual solutions and assume these approaches are sufficient. Native audit does not record the contextual details necessary to support security and compliance auditing or to detect attacks, and provide incident forensics. Furthermore, native database audit mechanisms are notorious for erratic and excessive consumption of the database server's CPU and disk resources, forcing many organizations to scale back or eliminate native 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 and reports.

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 be separate from both database administrators and the database server platform to ensure strong separation of duties.

**Only 19% of companies report monitoring database activity/movement[2]**

### Second Challenge: Processing the Audit

Having the correct audit trail is only the first step towards protecting data. The second step is understanding the data activity and access attempts to process that data and determine credible threats. It is difficult to identify the entity that accessed the database and distinguish between DBAs, applications, users, and jobs if you don't have the tools built for this task. You need to understand which accesses to the database are suspicious. For example, failed logins attempts are a regular phenomenon in database access. Users fail to login to a database due to forgotten or mistyped credentials or due to a password change. However, when a user fails to login to a database several times without success and never tries again or when a user tries to access several databases in the organization without success, it is suspicious and may indicate that the user is not authorized to access the application.

[2] Palerra Insider Threat Spotlight Report

In an Imperva research account the Imperva solution identified a user that tried to access a database he never accessed before using four different accounts in less than an hour without success. He succeeded in accessing the database using a fifth account, however the account did not have sufficient privileges to perform any actions on that database.

There are multiple red-flags with this activity:

- A sudden interest in databases the user has never attempted to access
- The use of multiple accounts by a single user
- The account that did access the DB had no rights—which might lead to a conclusion that the account should not have been able to access the DB at all

**2016 US Data Breaches: Over 35 million records reported lost or stolen, 44% of records were medical or healthcare related[3]**

Imperva flagged the activity as high-risk and provided an analysis that this incident was perpetrated by a compromised insider. To identify such incidents, you need to learn which of the users are human users (as opposed to jobs and applications). Then you need to learn the normal behavior of the users—which databases they access, which database accounts they use, when they are used, and more details that ultimately define peer groups and normal behavior. Unfortunately, many of the security tools used today fail to identify data breaches since they fail to separate between suspicious and normal accesses to the database. The tools overcompensate by producing too many vague alerts that required significant investigation to achieve visibility. This overload of generalized alerts is why less than one percent of critical security alerts are investigated.[4]

An example of a peer group anomaly is when one developer accesses an applicative table as part of his development job, and another developer accesses a table to see the personal data of a colleague. The key to determining the risk level is context, specifically understanding what table access is normal for the user and the user's peers. This inability to distinguish context is dangerous since malicious insiders take advantage of their privileges to steal data from the organization (see 'Privileges Abuse' section).

Native audit tools are not capable of discerning appropriate from inappropriate user access and often result in many excess alerts all which must be sifted by your security professionals. SIEM tools can trim this volume down and make it easier to visualize, but they lack the domain expertise and are abstracted from the source data and provide limited actionable options for direct investigation. Imperva CounterBreach behavior analytics and automated database monitoring and detection can provide the intelligence needed to focus on real threats in a contextual and actionable manner.

## 5. Unsecured Storage Media

When was the last time you pondered the vulnerability of your backup storage media? Typically, it's completely unprotected. Numerous security breaches have involved the theft or incidental exposure of database backup disks and tapes. Taking the appropriate measures to protect

[3] Identity Theft Resource Center, December 2016

[4] EMA Report Summary: Achieving High-Fidelity Security

backup copies of sensitive data is not only a data security best practice, but also mandated by many regulations.

In addition, highly privileged users will often have direct physical access to the database servers. This physical proximately means they can insert thumb or USB drives and execute SQL commands directly to the database that will shut off native audit and bypass all protection mechanisms except those deployed at the kernel level of the database server. You want a robust database monitoring and protection tool that will not allow these types of breaches.

### What About a Combination of Threats?

Each of the database threats discussed so far is certainly enough to create a data breach, but the opportunistic bad actor will look for the path of least resistance. Many times, we see a combination of threats that speed the attacker's access to data and simplify their ability to exfiltrate it undetected. Here's some examples:

- SQL Injection or Web Shell expose the database to breaches when the application has excessive privileges
- Privilege abuse is hard to detect due to the weak audit trail
- Privileges abuse is more severe when the user and/or application have excessive privileges

**Topping the risk list: 57% of companies think their databases are the most vulnerable asset to an insider attack**[5]

### Big Data Applications Security Threats Can't Be Ignored

Big Data applications are still in their infancy with few mature commercial solutions that can be deployed without customization to each company's specific requirements. At this stage in the market's development, there remains a lack of experts who understand the Big Data technology and can keep pace with its rapid-fire evolution.

In most situations, in-house developers design, code, test, and deploy Big Data applications and hardware without the benefit of adequate training, requirements definition, time, or resources. It can be mistakenly perceived that Big Data 'open source' packages are a quick win install. In reality, these systems are much more complex. A second issue when building the software is the lack of a viable native security or audit framework that will not impede the customized solutions. The lack of a native model makes the implementation of security a non-trivial process and requires extensive design and on-going maintenance. Thus, security and audit features that need to be considered get deferred repeatedly, leaving your data open to attack.

### Big Data – Security is Not a Priority

Some elements of the Big Data sector recognize the need for a native security and governance capability. There are early stage Apache projects that are seeking to address the requirements. Unfortunately, these projects often carry their own share of security issues which can directly affect the security of the Big Data systems they are attempting to protect.

[5] Palerra Insider Threat Spotlight Report

These issues include:

- Adding authentication processes to the application. This requires more security considerations which make the application much more complex. For example, the application needs to define users and roles. Based on this type of data, the application can decide whether to grant the user access to the system.
- Input validation. Once again we are seeing issues that have haunted RDBMS applications come back and haunt NoSQL databases. OWASP now recommends testing NoSQL databases like MongoDB for SQL injection style attacks.
- Application awareness. In the case where each application needs to manage the security, it must be aware of every other application. This is required to disable access to any non-application data.
- When new data types are added to the data store, the data store administrator must figure out and ensure what application cannot access that specific data.
- Weak code. There are many Big Data projects and products being developed via hyper-agile methodologies that do not allocate the time or resources for security reviews and testing. Bad actors will take advantage of flawed development methodology, exploring for vulnerabilities to exploit.

### Duplicated Data

The power of NoSQL is also its security Achilles Heel. In these systems, the data is not strictly saved in unique tables. Instead, the data is duplicated to many tables to optimize query processing. Thus, it is not possible to classify credit cards according to a particular sensitive table. On the contrary, this type of data can be found in different places: transaction logs, personal account details, specific tables which represents all credit cards, and other locations which may have not even been considered.

### Privacy

Although our focus is on security, privacy concerns cannot be ignored. Take, for example, a healthcare Big Data platform where providers share patient data. A patient might access the system for genetic information, and later access it in respect to drug info. An application which analyzes this data can correlate the information to find purchasing trends relating to genetics and health. The problem is that this type of correlation was not considered when the data was initially inserted. Thus, the data was never anonymized allowing pinpointing specific individuals from within the bigger trends picture. This will violate multiple regulations including HIPAA and GDPR.

**Average Costs in 2016: $158/record—average cost incurred for each lost or stolen record, $4 million consolidated total cost of a data breach[6]**

## How to Create a Comprehensive Data Security Solution

Data security requires an accounting of the data and user activity. This process begins with discovery of the database servers followed by access/activity monitoring. Continuous user rights management is also needed to stop privilege abuse. A robust solution like Imperva

[6] 2016 Ponemon Cost of Data Breach Study

CounterBreach and SecureSphere builds a complete security profile of the users and applications taking into account every instance of data access including that of privileged users. The contextual use of machine learning on database audit logs and deception token technology can accurately identify insider threats and prevent data breaches.

In parallel, it's important to harden the applications accessing the databases. SQLi and Web shells are just two of the threats to come in through your web apps. An advanced Web Application Firewall that can stop SQLI, web shell events, and prevent sophisticated business logic attacks will provide significant protection against unauthorized data access.

## Next Steps

As you have seen, the top five threats to databases require a multi-layered security approach. It's no longer enough to rely on native tools or to ignore security gaps that external and internal attackers can and will exploit. Protecting the data in databases is critical to protecting your customers, your reputation, and your business's viability. Imperva provides the tools needed to create a multi-layered, robust security approach that delivers 24/7 protection to your most important asset: your data.

To get started, contact Imperva today at US: 1-866-926-4678.