

# Top-10 Guide for Protecting Sensitive Data from Malicious Insiders

By Brian Contos, Chief Security Strategist, Imperva

Insider threats, both careless and malicious, abound. This fact is amplified during difficult economic times. With a plethora of digitized information, and vehicles for turning credit card data, personally identifiable information and intellectual property into cash, goods, and other services, risks have increased. It's no wonder that we're hearing about a growing number of attacks where the target is sensitive data, and the perpetrators are those with evaluated levels of trust and access: insiders. For years, organizations have worked diligently to lock down their perimeters only to find out that the most devastating enemy is already inside.

Today, most organizations would agree that their most valuable IT assets reside within applications and databases. Most would probably also agree that these are areas that have the weakest levels of security, thus making them the prime target for malicious activity from system administrators, DBAs, contractors, consultants, partners, and customers.

According to William (Bill) Crowell – former Deputy Director of the NSA

*“Security is not just the perimeter; layered defenses must be inside of the network and on the applications and databases if we really want to protect information. We haven't done nearly enough to protect applications and databases...and the magnitude of losses around insider threats are underreported.”*

This guide will explore the top ten ways to protect sensitive data from the very people that need access to it. While this is a difficult problem to address, it is not impossible – especially when leveraging the right tools.

In particular this guide will focus on protecting applications and databases with purpose-built solutions designed to mitigate data security risks through:

- Discovery & Classification
- Incident Prevention
- Incident Detection
- Continuous Auditing

## 10 – To Secure It You Have to Know About It

Read any security book written in the last decade and you will find references to the identification of mission-critical assets that warrant specific security controls. Why? Because wholesale security – that is the notion of protecting everything equally, is too expensive and doesn't scale. While everyone seems to agree that this makes perfect sense, the identification of these mission-critical assets is rarely done. This is especially true for databases and the sensitive data they contain. This data might be credit card numbers, PII or Personally Identifiable Information, employee compensation, healthcare records, research data, business plans, or top secret documents. Irrespective, virtually every organization has information that is sensitive and requires protection they however might not know where that information is.

The databases themselves first need to be identified. This may seem like a simple task at first, but it's not just global enterprises and government organizations that have difficulty here. Service Oriented Architectures (SOA) can be vast and complex. Databases might be replicated for testing – still containing sensitive data. Also, rogue or undocumented databases are not uncommon.

Once the databases have been identified, it's vital to classify the sensitive data and identify the objects contained. The classification must also be validated to reduce false positive identification of sensitive data. This type of information has a half-life; the processes identified here must be automated and reoccurring to be effective and support both scalability and accuracy over time.

## 9 – Don't Trust Native Database Tools

In cases of privileged insiders such as DBAs and system administrators, it is impossible to trust security information from the systems that have been attacked. If the malicious insider has access to the database and can possibly manipulate the native database audit logs, then these logs are useless. It's like having the criminal also be the crime scene investigator. This is a textbook example where Separation of Duties (SoD) is a must: Security and operations have to be separated. Audit information residing in or created from a database system under attack cannot be trusted. Additionally there are certain technical issues that native database audit logs introduce. For example:

- In many cases, native database auditing simply isn't enabled.
- Native audit enablement is manual and thus error prone; the DBA may enable insufficient auditing.
- Native auditing is notorious for its significant overhead on audited servers. Enabling native audit may drain the database host resources, impacting the productivity of the system.
- Native auditing can't capture enough information about the source of the activity to derive user accountability in environments where connection-pooling is used – more on that later in the guide under "You Can't Arrest an IP Address).
- Different databases provide different auditing capabilities. Enabling auditing on every database version and type is resource intensive in a heterogeneous environment, and because the output

is different, might be incomplete and hard to interpret by someone that isn't an expert in every database variant.

- Many databases, such as versions of Oracle, don't capture malformed SQL queries; guess what the bad guys use for reconnaissance? And these queries would be 100% undetected by native auditing.

Capturing database audit logs should be done independently of the database tools thus: enforcing SoD, increasing database performance, establishing user accountability, including appropriate "complete" details, and working across heterogeneous environments.

## 8 – Monitoring the Good, the Bad and the Privileged

Accessibility vs. Security: users who need access to sensitive data are also users that pose threats to that data. DMZs, partner networks, business process outsourcing, customer self-service portals, and SOAs are designed to improve access, increase operational efficiencies, facilitate information sharing, and cultivate creativity. Since you can't have a front without a back, all this "good stuff" has a sinister side.

**The Good:** must be monitored because you can't simply limit security controls to watching out for bad guys coming through the perimeter firewalls. Bad guys might disguise themselves as good guys. M&M security (hard on the outside and soft on the inside) has long since been retired. This is the era of the jawbreaker: watching everyone outside and inside – especially the "good guys" that interact with sensitive data.

**The Bad:** must be monitored because multiple risks from un-trusted outsiders still exist, so you can't operate without traditional network security controls. But leveraging them for sensitive data protection is like trying to hold back water with a tennis racquet. Network security is simply not designed to address data security.

**The Privileged:** privileged users must be monitored because they hold the keys to the kingdom. They are not only responsible for operations, but in many cases security. The irony is that they are asked to secure systems against themselves; with malicious privileged users this is like telling the inmates to keep an eye on the front door.

Full protection of sensitive data is required at all times, regardless of trusted, un-trusted, and privileged users. Assume everyone has access to the data. Assume everyone wants to steal it. Secure it accordingly by preventing where you can and monitoring where legitimate access is needed. Since web servers are the typical attack vector for external attackers to steal data, and traditional network security controls are not effective for mitigating data attacks, application firewalls need to be implemented to detect and prevent web-based attacks such as SQL Injection, Cross-site Scripting, Cookie Poisoning, Parameter Tampering, and Session Hijacking. Ensure "data security" controls are monitoring all privileged user activity as well as application users communication. Implemented controls must include incident blocking, alerting, reporting, and auditing. Finally, ensure these security controls are independent of the operational staff to effectively monitor for abuse by privileged users.

## 7 – Profiling Isn't Just for the FBI Anymore

Many people would argue that the focal point for malicious insider activity is where users are able to interact with sensitive data. Traditional network security controls, like firewalls that use signature matching methods, are focused on detecting and blocking specific events and are too binary when dealing with both people and data. If an organization can determine what “normal” activity is, it can begin classifying usage profiles based on the source of the activity (source user, source application, source location), the destination (targeted database and database objects) and the context of the activity (time of day, regular usage, successful vs. failed activities and more). Usage anomalies can then be detected where signatures and binary protection mechanisms fall short. One example that demonstrates the value of profiling is business logic attacks. Business logic attacks are attacks that turn the web application functionalities against the business - breaking the business logic instead of breaking the application.

It is also important to profile application and database interactions. This enables better protection against SQL injections attacks and identifies abnormal database activities that result from application design flaws. Profiling applications in this way will illuminate various, legitimate usages that might seem anomalous if not put in the right context:

- What might first appear as malware propagation where a single device is communicating with many - may actually be a backup server or possibly a proxy.
- A large amount of data transfer during non business hours might first appear to be information theft - while it might be legitimate database replication.

It will also find potentially malicious patterns such as:

- Excessive file downloads
- Activity during questionable times of days or days of the week
- Unauthorized attempts to reach classified data, for example: developers trying to access HR systems
- Suspicious failed activities like such as a high number of invalid login attempts

Profiling isn't like a onetime scan. Because applications and databases are dynamic environments which continuously change, it's important to implement a process of learning and constantly updating application and database usage profiles. Continuous updates to a usage profile are an indispensable part of a mature data security strategy.

## 6 – You Can't Arrest an IP Address

It's all about tracking users and holding them accountable. Identifying the person responsible for the activity can be very difficult. In most modern applications which use connection pooling user sessions are not natively tracked between Web applications and databases. Connection pooling improves the performance of applications by re-using existing open database connections instead of opening a database connection for each user. While the performance of the application is improved, connection

pooling masks the identity of the end user requesting the activity which means that there is really no way to correlate database activity with a specific user. Sarbanes-Oxley is very clear about the need to have user accountability for each change to financial reporting data. PCI DSS also requires the identification of unique users and associated activities.

Organizations can try to re-write custom application and database code to support this. However, this is an expensive and lengthy proposition that might introduce vulnerabilities with the additional code, and it would have to be done for every version of application and database within the organization.

The reconciliation of Web application and database activity should be done outside of the Web application and database and be independent of vendor, version, etc. Tracking user sessions in this way allows for greater control of session tracking without putting additional resource strain on the Web and database applications themselves, or pulling valuable development resources away from other projects. User sessions can be effectively reconciled between the Web application, database, and back again, thus not just capturing queries but the data returned as a result of the query. Perhaps most importantly, organization will be able to determine empirically who the user is.

## **5 – Augmenting Machine-Based Analytics with Human Intuition**

The scalability of preventative security controls is finite, as is the value that can be derived purely from technology without human analysis. Real-time alerts are generally based on a small window of analysis time that is derived based purely on technical analytics. Reporting augments this process in two ways. First, it allows for trends to be detected over longer periods of time, such as a pattern of abuse over months. Second, it leverages human intuition to augment machine-based analytics.

With insiders, it is essential to have an easy and efficient way for humans to derive results. This is because humans can take into consideration a broader set of variables than that which can be captured via application and database analysis alone— i.e. the user is underperforming and has been put on plan, or there have been rumors that the user is leaving the company and going to a competitor. Because IT security may not have the “big picture” for every person in every organization, it’s important for the reports to be useable by various stakeholders such as non-technical managers, human resources, and legal. This combination of real world analysis supported by detailed application and database evidence can yield more accurate results than either by itself.

## **4 –Forensic Crime Scene Investigation through Audit Logs**

Alerts, reports and individual event analysis have their place in finding insiders. But effective insider investigations require even more flexibility. The days of going through multiple heterogeneous databases that may not be auditing, or may not have sufficiently robust audit trails in search of insiders in gone. Organizations once leveraged tools like Perl, grep, SED, and awk to search through audit data for specific, known malicious entries in the past. This isn’t scalable, and it’s useless for finding unknowns.

When it comes to audit logs, a picture is truly worth a million logs. Visually interacting with sets of audit data yields causal relationships that simply aren’t noticeable without visualization tools. For example, an

analyst may start with broad investigatory strokes searching for information based on page hits, users, logins, and monitored assets such as servers. Within these broad searches, they will have access to the underlying event details. This information may cause them to look in new, previously unidentified directions. In most insider threat investigations, once signs of malicious activity are identified, three questions are asked: what else has the insider done, how long has this been going on, and who else might be involved in similar activities.

Consider the following sequence of events within an investigation.

- SQL errors are detected – however, not all are discovered within the production environment
- The analyst quickly and easily can leverage visual tools to filter and deselect sources not in production
- Next they further alter their analysis to refresh the data from the perspectives of users related to the SQL errors
- A particular user has a much higher volume of errors than the rest
- The user is also identified as a developer and the systems accessed are in production – a clear SoD violation by itself
- Drilling down further into the data it is discovered that the information he attempted to access resides in credit card and salary tables
- Other details might illustrate the number of successful and failed access attempts to this data
- Trend reports can be run to see – how long something has been occurring, and who else might be involved in similar activity

Leveraging visual analytics to investigate attacks within complex, heterogeneous audit data via on-the-fly filtering and drill-down can result in flagging malicious activity in minutes as opposed to days, weeks, or never discovering it at all.

### **3 – Sensitive Data Resides in Databases – Protect Them**

Database security is one of the most critical areas for sensitive data protection from insiders because they databases store the information insiders want. They are also highly complex and dynamic, making them difficult to lockdown. Exacerbating the issue is that DBAs are often focused on operational uptime and availability, with security as an afterthought. Historically, this has caused a divide between IT security and DBAs. Two solutions that work well for the needs of DBAs and IT security are Database Firewalls (DBFW) and Database Activity Monitoring (DAM) solutions.

**Database Firewalls:** As the name implies, DBFWs provide preventative controls that block malicious activity directly targeting the database in the forms of both attacks and data leakage just as traditional network firewalls address network-centric attacks. Some of the key capabilities DBFWs provide are:

- Block database attacks and fraudulent activity
- Provide real-time, automated protection of sensitive data
- Enable visibility into how users access data

- Leverage flexible views and audit analytics to further investigate audited events
- Transparently protect databases through virtual patching

With a DBFW, virtual patching can be used to address vulnerabilities discovered through a vulnerability assessment solution, without actually having to re-write code, apply patches, and undergo lengthy and expensive vulnerability mitigation processes. DBFWs can block and or alert on attacks looking to exploit virtually patched vulnerabilities allowing the organization to operate more safely until they have time and resource to directly address the vulnerability.

**Database Activity Monitoring:** While DBFWs provide capabilities at the database layer analogous to traditional network firewalls, DAM provides robust auditing by capturing bidirectional traffic between users and the database independent of the database and of DBAs. DAM helps address five key database audit questions:

1. Is the audit process independent from the database system being audited?  
Don't depend on the database for database audit information.
2. Does the audit trail establish user accountability?  
Specific events need to be associated with specific users for accountability including pooled communication.
3. Does the audit trail include appropriate detail?  
The audit system must collection enough detail to be useful. If the audit system in question is the database's native solution, the answer is typically no; consider the following examples
  - Example 1: JOHN requested DATA from the CUSTOMER database and the database returned DATA
  - Example 2: JOHN requested FIRST NAMES, LAST NAMES, EMAIL ADDRESSES, PHON NUMBERS, and CREDIT CARD NUMBERS for ALL customer from the CUSTOMER database and the database returned 65,000 records
  - Assuming that John is authorized Example 1 is of little use
  - However, since detailed audit logs can overwhelm processors, disks, and I/O resources Example 1 is commonplace
4. Does the audit trail identify material variances from baseline activity?  
It is critical to determine the differences between normal activity and anomalies.
5. Is the scope of the audit trail sufficient?  
The entire database "system" must be monitored and include
  - Database software
  - Operating system software
  - Database Protocols – note that DB protocols don't conform to an open standard and often change – they are common entry points for attackers

Together DBFWs and DAMs provide a combined solution for database protection, monitoring, and auditing that is completely independent from users – privileged and otherwise. They provide audit quality data with the ability to investigate and report on malicious activity targeting the database.

## 2 – Users Get to Databases through Web Applications – Protect Them, Too

While sensitive data resides within the database, most users access that database through a Web application. As such, in addition to layers of database protection, there needs to be protection for the Web applications. Protecting online applications and data against sophisticated application-level attacks requires a two-pronged approach that considers development and Q&A as well as production.

**Development and QA:** When protecting Web applications, several steps can be taken before the application goes into production to improve security. Training application developers about secure coding best practices and leveraging a Security Development Life Cycle (SDLC) can help improve the development of secure code. There are also several models for testing and analyzing code such as, static code testing (reviews and walkthroughs), dynamic code testing (executing programmed code), architectural risk analysis, abuse case testing, black box testing (no internal knowledge), white box testing (internal knowledge of the code), and many others.

Many organizations also use a Web Application Firewall (WAF) during the development and QA phase. In this scenario, a WAF doesn't provide firewalling capability, but monitoring – or more precisely, provides developers a better view into the operations of their applications. This is even effective during beta testing with live customers and customer data. WAFs can provide:

- Visibility into web application (HTTP) and (HTTPS) traffic
- Visibility into how customers are using the web application and how the application is working
- Visibility into expected versus actual user behavior
- Detailed information on the parameters in your web application such as length and type
- Performance results for web connections
- Visibility into database activity (if leveraged with DAM or DBFW)

**Production:** All the secure coding, testing, and best practices are now locked in, and it's just the Web application against the world. Because threats evolve, and code is never perfect or 100% secure, there is a need to enhance Web application security during production. WAFs can provide a number of advantages, not the least of which is the ability to block malicious activity, in addition to monitoring, alerting, and reporting. Another key area is addressing vulnerabilities with the Web application. WAFs can perform virtual patching just as DBFWs do, but at the Web application layer.

Some other WAF capabilities include:

- Operation across multiple application types and versions
- Alerting or blocking on bad events
- Web Application Intrusion detection
- Application hardening
- Defense-in-depth for managing application risk
- Automatic incident documentation

WAFs provide an effective mechanism for increasing the security during development and production phases. By providing incident prevention and detection at the Web application layer attacks on sensitive data – regardless if it's from within or outside of an organization can be mitigated, and sensitive data becomes more secure.

## **1 –Needles Hiding in Stacks of Needles**

A number of approaches and technologies for mitigating malicious insiders have been covered thus far. Yet, there's one last unifying variable required. To bring clarity to data security correlation is necessary. By correlating application and database information, it becomes possible to create a holistic perspective over an organization's entire data security posture while enhancing the ability to prevent, detect, and audit insider activity: finding the needles in the stack of needles.

Today's attack vectors take advantage of weak points across databases and applications. From reconnaissance to actual attacks, it is critical to correlate information across both to truly understand how users are interacting with data, and separate legitimate activity from potentially malicious or known malicious activity.

Insider threat analysis benefits from multiple sources of data-centric information because a single source might not provide the complete story. Discovery and classification should be used to identify critical assets and the information they contain. WAFs should be leveraged to protect applications. DBFWs should be used to protect databases, and DAMs should be used to provide database auditing. Together these solutions can be brought together to provide insider threat mitigation for even the most complex, distributed, and mission-critical environments without impacting web applications and databases operationally.

Imperva is exhibiting at Infosecurity Europe 2010, the No. 1 industry event in Europe held on 27th – 29th April in its new venue Earl's Court, London. The event provides an unrivalled free education programme, exhibitors showcasing new and emerging technologies and offering practical and professional expertise. For further information please visit [www.infosec.co.uk](http://www.infosec.co.uk)

### **About Imperva**

Imperva, the Data Security leader, enables a complete security lifecycle for business databases and the applications that use them. Over 4,500 of the world's leading enterprises, government organizations, and managed service providers rely on Imperva to prevent sensitive data theft, protect against data breaches, secure applications, and ensure data confidentiality. The award-winning Imperva SecureSphere is the only solution that delivers full activity monitoring from the database to the accountable application user and is recognized for its overall ease of management and deployment. For more information, visit [www.imperva.com](http://www.imperva.com).

