Whitepaper

Practical Guide to Database Security & Compliance

Including:

- Reconciling Compliance and Security Requirements
- 5 Principles of Protecting the Database
- 5 Practical, Inexpensive Steps to Database Security

Provided by sentrigo

Version 1.1, November 2007

www.sentrigo.com

Why Database Security?

Database security, whether as a topic of discussion or a set of tools, has been around for at least five or six years. It is only in the last couple of years, however, that it is beginning to draw more attention from industry analysts and security and database professionals.

The combination of worsening and highly publicized data breaches on the one hand, and stricter regulatory compliance demands on the other hand are pushing database security to the foreground. There is a still a halo of “black art” around this topic, since many database professionals are not familiar with security aspects of database management, while many security professionals are familiar with network and desktop security, but not with database security. This is beginning to change as the importance of securing databases becomes apparent.

Understanding the Threat

Databases are subject to some unique types of threat that cannot be handled by firewalls, intrusion detection and prevention systems and other perimeter defenses. The threat landscape is constantly evolving and becoming more sophisticated and specialized (e.g. attacking through memory backdoors inside databases).

Who Are The Intruders?

The romantic notion of the lone high-school student, hacking into the Pentagon from his mother’s basement just to prove that he can do it has taken the back seat to armies of professional, organized and highly capable individuals, working in the service of organized crime syndicates with the aim of making a profit.

This has changed the nature of intrusion attempts from ones that try to penetrate, then perhaps deface or wreak havoc, to ones that strive to be stealthy and leave no tracks with the aim of stealing data for financial gain.

Insider Threat, Privileged Users

Concurrently with the change in the nature of the external threat, there is increasing attention being given to the “insider threat”. This umbrella term refers to damage caused by individuals within the organization, either maliciously or accidentally.

Is the insider threat serious? It certainly is. Recent breaches such as the one at Fidelity National Information Services, where a senior DBA sold millions of customer credit card records is proof of that. This does not mean that all insiders are suspects – however it is clear that insiders bent on stealing data have a greater chance of succeeding at it than outside intrusion attempts.

The criminalization of the general threat landscape also has an impact on “crimes of opportunity” committed by insiders. It is often easier and quicker to bribe an insider with access privileges than to attempt hacking from the outside (and certainly easier than holding up a bank).

According to annual research conducted by CERT, up to 50% of breaches are attributed to internal users. The 2006 FBI/CSI report on the insider threat notes that two thirds of surveyed organizations (both commercial and government) reported losses caused by internal breaches, and some attributed as much as 80% of the damage to internal breaches. It was also reported that 57% of implicated insiders had privileged access to data at the time of breach. It is therefore evident that perimeter and network security measures are not enough to stop such breaches.

Many regulatory compliance requirements focus on privileged insiders as well, with special attention given to those whose actions used to go unmonitored in the past.

Vulnerabilities

As database management systems have grown in complexity, they have become more vulnerable to attacks. The nature of these vulnerabilities ranges from relatively benign to ones that allow unauthorized users to own the database through privilege elevation.

Much has been said and written about how DBMS vendors cope with vulnerabilities and how quickly they should patch them. The reality over the past few years shows that the number of reported vulnerabilities is rising, and while vendors are doubling their efforts to patch them, the number is constantly rising.

Additionally, it usually takes the vendor several months or more to distribute a patch, and it takes an additional several months for customer to install the patches, which usually require testing and database downtime. Many customers do not apply the patches at all, and their databases remain vulnerable to severe attacks.

Existing Solutions are Inadequate

Perimeter Security

When it comes to databases, the traditional perimeter defenses such as firewalls and IDS/IPS are grossly inadequate due to the nature of the threats posed to database, and the specific nature of their vulnerabilities.

While many intrusion prevention systems claim to thwart, for example, SQL injection attacks, their capabilities in this area are very weak and are usually based on signatures, which hackers can easily evade. They are certainly incapable of detecting sophisticated attacks that exploit vulnerabilities that are specific to a database vendor, version and platform.

Native DBMS Auditing

Virtually all database management systems have the ability to write audit logs for some or all transactions. However, they are seldom used extensively due to the detrimental effect they have on database performance, and the amount of storage they would need for full auditing.

From a security standpoint, the usage of open logs is inadequate anyway, since the logs can easily be manipulated after the fact, and privileged users can turn the audit function on and off as they please.

Compliance and Security

Compliance Requirements for Databases

There are many different laws and regulations with which businesses need to comply nowadays. This recent development has evolved over the past 5 years and changed the way many IT systems, applications and data are controlled.

Following are brief descriptions of the key regulations and their affect on database management and security practices:

Sarbanes-Oxley

The Sarbanes-Oxley legislation of 2002 (SOX) forced publicly traded companies to be more transparent about their financial data. This is a federal law and does not specify technical measures or tools that enterprises should use, but rather requires them to have “effective controls” in place. Specifically:

- ❖ Section 302 of SOX requires executives to certify the accuracy of financial reports. This entails that company officers must have a handle on how data is flowing to create the reports. They must be certain that the data cannot be seen by unauthorized personnel, altered without authorization, or otherwise tampered with.

- ❖ Section 404 of SOX further requires executives and auditors to confirm the “effectiveness of internal controls”. While the exact nature of internal controls is not specified, auditors commonly put particular emphasis on:

- Ensuring the integrity of sensitive data
- Activity of privileged insiders
- Traceability of data (audit trail)
- Separation of duties (audit independence)

Obviously most of the financial data resides in databases in one form or another, so SOX auditors are increasingly looking for ways to view database activity in a way that can be easily interpreted and acted on.

PCI DSS

The Payment Card Industry's Data Security Standard (PCI DSS) is the result of the joint efforts of the major credit card companies. It is not legislation nor regulation but a standard, and periodically gets updated (so far – once a year). The standard compels merchants and companies who process and store credit card data to comply with a set of technical and procedural requirements, and pass audits. Unlike SOX and HIPAA, PCI gets specific about what measures need to be put in place for protecting credit card data. Inability to comply carries stiff penalties from the credit card companies, and, if not rectified, eventual withdrawal of the right to conduct business with the credit card companies.

Of particular interest is the concept of "compensating controls" (section 3.4), which recognizes that card holder data encryption is sometimes not possible or would take a long time to implement. It allows for a combination of other methods to be used instead, including real-time monitoring of user activity.

CA SB1386 and Similar Privacy Breach Notification Laws

At the time of writing, 39 states in the US have enacted privacy breach notification laws similar to California Senate Bill 1386, and a federal bill may be adopted in 2008. These laws compel organizations to notify the authorities and affected individuals whenever a breach of personal identifiable information (PII) is exposed, such as Social Security numbers.

Privacy breach notification laws do not require a specific set of controls, but rather specify what a company must do when a breach is suspected to have occurred. Since notification is expensive both in direct costs as well as indirect damage to reputation, loss of customer trust etc., it is in the company's interest to take measures to protect PII. Additionally, if a breach occurs, it would be valuable to have the knowledge of exactly what data was compromised. The cost of notification increases almost linearly with the number of affected data records.

HIPAA

The Health Information Portability and Accountability Act is a federal law that was put in place to ensure that the freedom of patients to choose healthcare insurers and providers will not come at the expense of the privacy of their medical records.

The articles relevant to securing database are mainly the following:

- **§ 164.312(a)(1): Access Control**, which requires organizations to *“Implement technical policies and procedures for electronic information systems that maintain electronic protected health information to allow access only to those persons or software programs that have been granted access rights”*
- **§ 164.312(b): Audit Controls,** which requires organizations to *“Implement hardware, software, and/or procedural mechanisms that record and examine activity in information systems that contain or use electronic protected health information.”*
- **§ 164.312(c)(2): Integrity,** which requires organizations to *““Implement electronic mechanisms to corroborate that electronic protected health information has not been altered or destroyed in an unauthorized manner”*

Currently HIPAA is not strongly enforced, but this is expected to change as reforms in the US healthcare system continue.

SAS 70

SAS 70 is an auditing standard for the service industry that allows auditors to certify that a service company has the appropriate controls in place to safeguard customer data. This is not a regulation but a standard, and one that gives organizations who adhere to it a badge of quality, which is important in the service industry.

Like SOX, SAS 70 does not specify what measures need to be in place, but tools and procedures that facilitate the job of the auditors, and can demonstrate who was doing what in the database can automate this process and save costs.

Regulatory Compliance ≠ Security

Many corporate initiatives in recent years that contributed to improvements in security were driven by the need to comply with various laws and regulations. Such projects may lead companies to think that because they are compliant, they are secure, and create a false sense of complacency.

However, compliance does not equate security by default. A company may be compliant with a host of regulatory requirements, but its databases would remain exposed and vulnerable. There are several reasons why this happens:

- Regulations are limited in scope to whatever they are intended to regulate. PCI DSS, for example, is concerned with credit card details, and credit card details only. If someone stole your employee data, for instance, this would be of no concern to the PCI auditors.
- Compliance is often audit focused, and audits are, by nature, activities that take place *after the fact.* So you may discover that you had been breached 4 months earlier, but it will not do you much good in reversing that data theft.
- The requirements posed by some regulations represent a minimum baseline, not a best practice. They are created this way to allow many organizations to comply.
- Regulations are outdated as soon as they are published. The threat landscape changes faster than lawmakers write laws and regulators issue regulations.

Reconciling Compliance and Security Requirements

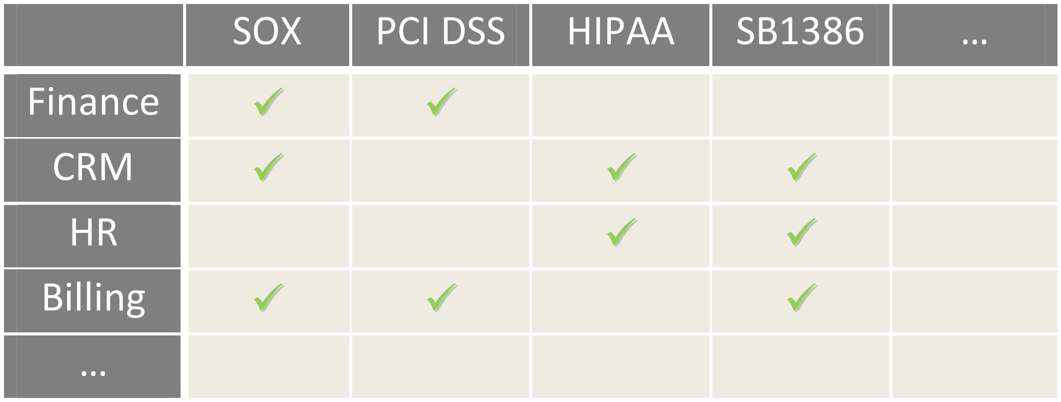
Regulatory compliance can be seen as a burden, or it can be seen as an opportunity to streamline business processes and significantly upgrade security. In order to make the best of this opportunity, enterprises should align their compliance projects with security requirements, to ensure that the required measures are put in place in one fell swoop.

The Big Picture

One can easily be blind sighted by the pressing requirements of regulatory compliance and put aside other considerations until those requirements are met. However, enterprises often have multiple compliance requirements to consider, as well as security considerations, and given the great overlap between the different requirements, it would be a waste of resources to go through separate efforts for each objective.

It is therefore useful to take a step back and look at the big picture. Even if your current initiative does not cover all corporate databases, it makes sense to apply all relevant security improvements to those databases touched on by the next audit or compliance project.

Take, for example, a publicly traded healthcare insurance company. It will need to comply with Sarbanes-Oxley, but also with HIPAA, privacy notification laws such as SB 1386 and possibly PCI DSS if the company takes payment by credit card. Many databases would be relevant to two or more regulatory audits, so it is important to map those as in the following example:



	SOX	PCI DSS	HIPAA	SB1386	…
Finance	✓	✓			
CRM	✓		✓	✓	
HR			✓	✓	
Billing	✓	✓		✓	
…					

The relevant databases can then be mapped for relevant sensitive data, privileged users, and the type and frequency of audit reports that are required. The additional effort is marginal and provides a more efficient and comprehensive view.

Overlapping Requirements

Virtually all compliance requirements adhere to principles that are useful for improving security as a whole, and those can be leveraged for upgrading database security with no additional effort:

- **Controlling Access to Sensitive Data:** Requires identifying sensitive data and using user authentication, auditing and monitoring to enforce access policy to that data
- **Separation of Duties:** Requires that the people in charge of auditing or monitoring the database are not the same people whose actions are being monitored
- **Monitoring Privileged Users:** Due to the level of access given to privileged users, such as DBAs, sys admins and developers, special attention must be given to their activity

Practical Guide to Implementing Database Security

Five Principles of Protecting the Database

I. Think Security in Everything You Do

Constantly examine your actions with security goggles. Starting with application development, through everyday tasks like user management and data management. Do not think of security as something you do once a month, and educate your users to do the same. Most security gaps are there due to ignorance and lack of awareness, more than any other reason.

II. Use the Least Privilege Principle

The least privilege principle calls for users and applications to have the minimal privileges they require to function properly. This entails not only applying restrictions when first granting users access to the database, but also remembering to review those access privileges periodically and changing them. Many organizations grant deep privileges to consultants and developers who work for them on a temporary basis, but then forget to remove or change those privileges when the work is done.

Note that even seemingly innocent VIEW privileges can be used by some attack vectors to gain access privileges through vulnerabilities – carefully consider granting every kind of privilege.

III. Minimize the Attack Surface

It is more difficult to secure a large house with many windows than a small house with few windows. Database systems are the same – the more complex they are, the larger the attack surface.

Strive to reduce the attack surface by eliminating components that are not in use, and avoid installing them in the first place.

IV. Encrypt, but Not as a Panacea

Encryption is often the first thing that comes to mind when thinking of securing data, and is certainly recommended for sensitive data. However, it can be both expensive and difficult to use, and certainly difficult to manage in a way that is secure. Encrypt only sensitive data that requires it, be careful how you manage the encryption/decryption keys, and change them on a regular basis.

It is important to combine encryption with other means and procedures, such as activity monitoring, auditing, periodic vulnerability assessments and user authentication.

V. Development, Testing and Staging Environments

Many organizations invest efforts in securing their production databases, but neglect to do so in development, testing and staging environments. As the staging environment is often copied into production when it is ready, it should obviously be as secure as the production version. Beyond that, it is often the case that real production data is used in non-production environments without any masking, and this poses a serious security risk. It is recommended to treat non-production environments with the same tools and procedures one applies to the production environment.

Five Practical First Steps

The following are steps that anyone can perform on their database without getting into lengthy projects, vulnerability assessments or the use of expensive tools and third party services. All they require is a minimal investment of time and attention. These steps will not make your database 100% secure (nothing can guarantee that), but they will bring you a long way towards a more secure environment.

I. Usernames and Passwords

While on trial for multiple counts of data theft, a hacker recently told the jury in his trial that *80% of his successful break-ins were due to weak username and password combinations or the use of standard passwords*. This demonstrates that duping the user authentication mechanism is still the easiest way of penetrating a database, and nothing makes that easier than sloppy use of default usernames and passwords, weak passwords and shared passwords.

Oracle and other databases come with built-in default usernames and passwords – several hundreds of them – created to make it easier to set up the system. These should be erased after the system has been set up. There are free tools available on http://www.petefinnigan.com that do just that for Oracle specifically.

Weak passwords are passwords that are short (fewer than 6 characters, though we recommend at least 8), based on dictionary words, names and dates. There are tools out there, available for download from the web and not requiring any hacking skills, that can crunch through 400,000 passwords per minute. That means going through all the words in the Oxford dictionary in less than 30 seconds, so even using inflections and combinations of words and numbers, such a tool will break through in a very short time. *It is therefore imperative to use strong passwords* that are not words in the dictionary, names or dates, and contain a combination of letters, numbers and symbols.

II. Remove Unnecessary Components

DBMSs today, especially the enterprise versions, are behemoth applications with many options that most users will seldom use. Certain database vulnerabilities exploit such add-ons and extensions (for example, Oracle's APEX). *This multitude of components creates a very large attack surface* (see above principles) and thus more opportunities to infiltrate the database.

Review your database configuration periodically, and remove components that your users are not using, including various extensions and add-ons. Do not install them with the view for future use – you can always install them when the time comes.

III. Apply Security Patches

New database vulnerabilities are uncovered constantly, and many are patched by the database vendors by issuing updates and patches to the DBMS. It is not always easy to apply those patches, because they require testing and database downtime, but even deciding on a schedule where patches are applied twice a year is better than not applying them at all.

Ironically, it is especially after patches are issued by the DBMS vendor that the systems that remain un-patched are even more vulnerable to attack, since the public announcement of the availability of such patches draws the attention of potential intruders to the existence of vulnerabilities in specific modules.

An alternative and complementary approach is to use virtual patching tools, such as the ones available from Sentrigo, to create an external layer of defense on top of the database that specifically addresses vulnerabilities, and issues alerts or takes action to stop attempts to exploit them.

IV. Secure Coding Practices

Many database vulnerabilities are exposed due to the way applications are coded, and their interaction with the database. Lack of accountability and lack of secure coding practices may open the way to breaches and attacks.

For example, SQL injections in Web applications can be thwarted entirely by binding variables in SQL statements. Unfortunately many developers still do not use the bind variables method when developing applications, leaving the database exposed to SQL injections.

Architecting and designing for security, validating input and sanitizing data sent to other systems are some of the recommended methods used in secure coding. You can visit the CERT website for additional information on coding standards.

V. Monitor, Audit, Monitor and Audit Again…

You don’t know what you can’t see. Seems obvious and yet most DBAs and security professionals have no idea who is doing what in the database.

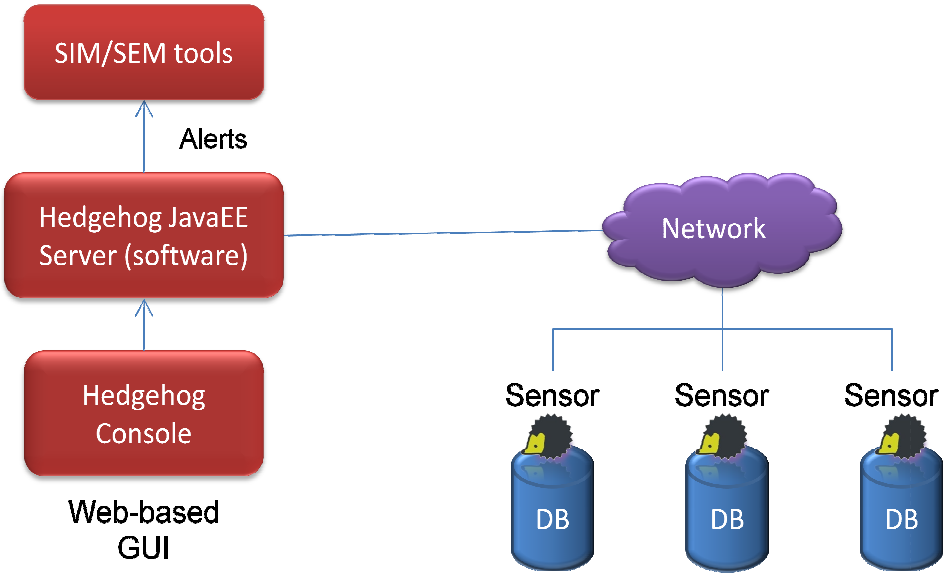
Auditing is an offline endeavor that looks back at the database activity over a period. Full native DBMS auditing is impractical as it significantly slows performance, but selective fine-grained auditing can be used. While certain compliance requirements may force you to audit “everything” and keep an infinite audit trail, it is also impractical – the more benign actions you record, the less likely you are to notice the conspicuous ones. Try and work with the auditors to define the types of activities that are crucial (accessing sensitive data, privileged user access etc.), and record those.

Monitoring occurs in real-time, and is therefore more actionable and useful in security terms. There are mostly expensive tools available for enterprise deployments, some with prevention capabilities. Now there is also a *free* real-time, granular security monitoring tool available from Sentrigo – called Hedgehog Standard – available for download on www.sentrigo.com.

HEDGEHOG ENTERPRISE™ – REAL-TIME ACTIVITY MONITORING, AUDITING AND BREACH PREVENTION

Sentrigo's Hedgehog is an elegant software solution for database security. It is a system that ***monitors all database activity in real-time***, and based on a defined rules and policies issues alerts on suspicious activity and, if necessary, stops it as it happens.

Hedgehog is comprised of a small footprint sensor, a software agent that is installed on the database host server itself and monitors all activity. It is nonintrusive, easy to install and consumes only small amounts of CPU resources (<5% of one single CPU, even on multi-CPU machines). The sensor communicates with the Hedgehog server, which generates alerts according to the defined rules.



The policy rules apply to types of SQL statements, database objects accessed, time of day or day of the month, specific user profiles, IP addresses and the applications used, among other parameters.

The action taken when the conditions of a rule are met can be as simple as logging an event, ***sending an alert*** to a SIM/SEM system, via e-mail or SMS, or ***terminating a user session to prevent malicious activity***. Users can also be quarantined to prevent subsequent attempts at breaching the database. The system comes with predefined rules that prevent known attacks that exploit database vulnerabilities.

A single Hedgehog server can manage and communicate with numerous sensors on different databases, and ***an enterprise installation can scale up*** to encompass hundreds of databases. The server also easily integrates with IT infrastructure to facilitate central IT management and security event management.

Since the Hedgehog sensor is installed on the database machine, it is impossible to bypass and possesses self-defense mechanisms that alert on any tampering attempts. The structure of the system ensures ***separation of duties***, with definable roles and access rights to different users.

Sentrigo employs a “Red Team”, a group of ethical hackers who are on the lookout for new database vulnerabilities. As soon as such vulnerabilities are discovered, the team creates rules that protect against them – essentially ***virtual patches*** that immediately protect the database without the need for a system upgrade or downtime, and leaving the database exposed until DBMS updates are issued by the vendor.

Unique Advantages:

- The only database monitoring solution that monitors all database activities and provides protection against insiders with privileged access
- Granular monitoring of database transactions, queries, objects and stored procedures, with real-time alerts and prevention of breaches
- Flexible rules that allow enforcement of corporate security policy with minimal “false positive” alerts
- Virtual patching of newly discovered database vulnerabilities, providing immediate protection with no DBMS downtime
- Flexible audit and reporting capabilities suitable for PCI DSS, SOX and HIPAA
- An easy-to-deploy and scalable software solution
- Multiple user roles that facilitate separation of duties

Hedgehog Enterprise is available for free evaluation and is downloadable from Sentrigo’s website: www.sentrigo.com

ABOUT SENTRIGO

Sentrigo, Inc. is an innovator in security software that monitors all database activity and protects sensitive information in real time in order to prevent both internal and external data breaches. Sentrigo's Hedgehog software, including a free version, can be downloaded and easily installed to provide immediate protection against breaches, as well as virtual patching against recently discovered threats—with minimal impact on database performance. The product's unparalleled level of protection, coupled with its ease of use, makes it the instant standard for database security and regulatory compliance automation.

Sentrigo was named by Network World magazine as one of the top 10 IT security companies to watch in 2007. For additional information and to download Hedgehog, visit www.sentrigo.com.