Securing Web Applications and Databases for PCI Compliance

The Most Challenging Aspects of PCI Compliance

White Paper

Web and database security present some of the most complex and costly barriers to compliance with the Payment Card Industry Data Security Standard (PCI DSS). Issues like secure Web application software development, database encryption, and database transaction auditing in high performance production environments cannot be addressed by simply adjusting system configurations or documenting policy for an existing process. These issues have architectural and human resource implications that significantly impact IT budgets.

Of the twelve requirements that comprise the standard, there are three requirements that most strategically impact Web and database security: requirement 3, requirement 6 and requirement 10. This paper focuses on these requirements. Designed for Web and database security professionals, key compliance challenges are analyzed with respect to impact on existing IT infrastructure, staff, and budget. We'll also discuss how Web and database security appliances can be applied to achieve compliance more cost effectively than alternative approaches.

Requirement 6 – Develop and Maintain Secure Systems and Applications

PCI DSS Requirement 6, titled "Develop and Maintain Secure Systems and Applications", focuses on the establishment of controls that minimize the existence of security vulnerabilities in systems and software. It specifies requirements for software patching, vulnerability identification, secure software development, change controls, and attack protection. Although the general scope of requirement 6 extends across the entire cardholder environment including routers, databases, applications, and other network resources, it uniquely isolates Web applications as an area of explicit and detailed focus. Of the six main sub-requirements to section 6, two of them (6.5 and 6.6) are excusive to Web applications. The following paragraphs discuss these very detailed Web application requirements.

Why Focus on Web Applications?

Web applications represent the most obvious entry point for the global cyber-criminal community to gain access to back-end databases containing credit card data. An un-validated input vulnerability, for example, can be exploited to gain access to the entire contents of a back-end database. Attackers are taking advantage. According to an FBI computer crime and security survey, 92% of businesses have suffered Web application attacks. Many of those attacks have been successful, as evidenced by the increasing frequency of public disclosures of online credit card losses[1].

Requirement 6.5 – Secure Web Application Development Practices

Implementation of secure coding guidelines, as recommended by requirement 6.5 and the Open Web Application Security Project, can effectively reduce the volume of vulnerabilities that exist in new Web software as it is developed. Clearly, implementation of a secure coding practices framework, although challenging, is an important aspect of Web application security. However, exclusive reliance on this approach ignores several important facts.

1. **Secure Coding Only Applies to New Software** – What can be done about the many millions of lines of pre-existing custom Web software linked to credit card data around the globe? Should that code be re-written? Not likely.
2. **All Software Has Bugs** – Anyone who has ever written business software knows that bugs exist no matter how rigorous software developers are in their practice of secure coding. Some bugs are just annoying. Some bugs affect reliability. But inevitably, some bugs affect security.
3. **Inconsistent Architecture and New technologies** – Changes made to a new software component may inadvertently expose vulnerabilities in another older component. For example, the growing prevalence of Web services and Web 2.0 leads to exposure of vulnerable objects or methods in older code that may not have originally been designed to these new technologies.
4. **IT Resources Change** – New developers or contractors may not immediately understand secure coding principals despite an organization's best efforts to educate them. There is always a learning period.

The designers of the PCI DSS added section 6.6 to address these problems.

Requirement 6.6 - Protect Web Applications against Known Attacks

Requirement 6.6 provides two options to address the risks described above.

» **Reviewing Web Applications** – By having all Web applications reviewed at least annually and after any changes, organizations can reduce the number of application vulnerabilities and the associated risk of data compromise.

» **Installing a Web Application Firewall** – By installing a Web application firewall in front of Web applications, attacks can be blocked before credit card data is compromised.

Both of these mechanisms simultaneously reduce the risk associated with pre-existing custom Web application software and the inevitable bugs that affect security. Organizations may choose to implement either approach to achieve PCI DSS compliance. But which option makes the most sense?

[1] A reference may be found at http://www.webappsec.org/projects/whid/list.shtml

Application Review versus the Web Application Firewalls

Application Vulnerability Assessments and Code Reviews

Application reviews are theoretically the most straightforward method of mitigating Web application security vulnerabilities. Without vulnerability, there is no hope for an attacker. However, application reviews introduce a number of problems.

» **Cost** – For organizations that opt for source code reviews, the task of implementing code reviews is an extremely labor intensive process that involves line-by-line inspection of code. Together, these factors make code reviews very expensive for applications of any significant size.

» **Cost (again)** – According the PCI DSS test procedures, organizations must engage application security specialists, review applications annually or after any changes, correct discovered vulnerabilities, and retest the applications after corrections have been applied. The bottom line is that application reviews are prohibitively expensive both in terms of security consultant fees as well as internal application development and QA personnel costs.

» **Window of Exposure** – Finding a vulnerability and fixing it are entirely different. Unfortunately, many vulnerabilities are found in applications that are already deployed. Most organizations will work through a set process before a fix can be deployed: analysis of the nature of vulnerability, proposed fix approach, software architecture for the fix, actual development, QA and validation, change management documentation and approval, and finally deployment in a scheduled maintenance window. Critically, applications can remain exposed while organizations work through the weeks and months this process can take.

Web Application Firewalls

Web Application Firewalls automatically detect and block attacks before damage may occur. With a Web application firewall deployed, a given Web application may have a vulnerability, but the risk associated with that vulnerability is mitigated. Web Application Firewalls offer several advantages relative to application review and fix cycles.

» **Cost** – With a limited up-front investment, Web application firewalls scale to automatically protect thousands or millions of lines of code without manual labor. For applications of any significant size, this automation will offer significant cost savings compared to application review and fix cycles.

» **Cost (again)** – Most Web application firewalls require support and maintenance fees at a fraction of their initial price. Again, these costs are significantly less than the costs associated with repeated reviews and fix cycles for applications of any significant size.

» **Continuous Security** – Web application firewalls provide continuous attack protection over time. It is not subject to the varying risk windows that are inherent to the application review and fix cycles as described above. Web application firewalls can deliver critical virtual patching for vulnerabilities as organizations work through their fix process.

How can SecureSphere Help?

Table 1 details how Imperva's SecureSphere Web and Database Security Solutions can be applied to achieve compliance with PCI DSS requirement 6.

Table 1: SecureSphere Applicability to PCI Requirement 6

Req #	Text	SecureSphere Applicability			SecureSphere Applicability Detail
		Direct	Partial	NA	
6.1	Ensure that all system components and software have the latest vendor-supplied security patches installed. Install relevant security patches within one month of release. ***Note:*** *An organization may consider applying a risk-based approach to prioritize their patch installations. For example, by prioritizing critical infrastructure (for example, public-facing devices and systems, databases) higher than less-critical internal devices, to ensure high-priority systems and devices are addressed within one month, and addressing less critical devices and systems within three months.*		X		SecureSphere Database Security Solutions includes assessment capabilities that evaluate configuration of both underlying operating systems and database management system software. Assessments include patch and version information. This is an effective audit mechanism for database update/patch control objectives.
6.2	Establish a process to identify newly discovered security vulnerabilities (for example, subscribe to alert services freely available on the Internet). Update configuration standards as required by PCI DSS Requirement 2.2 to address new vulnerability issues.		X		Subscription-based SecureSphere ADC Insight Services provide regular updates to the assessments of known vulnerabilities in business application (SAP, Oracle EBS, etc.) and database software. This information is integrated in the assessment capability mentioned in section 6.1
6.3	Develop software applications in accordance with PCI DSS (for example, secure authentication and logging) and based on industry best practices, and incorporate information security throughout the software development life cycle. These processes must include the following:			X	
6.3.1	Testing of all security patches, and system and software configuration changes before deployment, including but not limited to the following:			X	
6.3.1.1	Validation of all input (to prevent cross-site scripting, injection flaws, malicious file execution, etc.)			X	
6.3.1.2	Validation of proper error handling			X	
6.3.1.3	Validation of secure cryptographic storage			X	
6.3.1.4	Validation of secure communications			X	
6.3.1.5	Validation of proper role-based access control			X	
6.3.2	Separate development/test, and production environments			X	
6.3.3	Separation of duties between development/ test, and production environments		X		SecureSphere's Dynamic Profiling can enforce and audit separation of duties between different users and/ or roles based upon their normal behavior profiles. For example, by monitoring production environments user access SecureSphere can detect and log unauthorized changes made by developers.

6.3.4	Production data (live PANs) are not used for testing or development			X	
6.3.5	Removal of test data and accounts before production systems become active		X		Sensitive data discovery and user account assessments can audit removal of these accounts from production Web application and database systems
6.3.6	Removal of custom application accounts, usernames, and passwords before applications become active or are released to customers		X		Sensitive data discovery and user account assessments can audit removal of these accounts from production Web application and database systems
6.3.7	Review of custom code prior to release to production or customers in order to identify any potential coding vulnerability ***Note:*** *This requirement for code reviews applies to all custom code (both internal and public-facing), as part of the system development life cycle required by PCI DSS Requirement 6.3. Code reviews can be conducted by knowledgeable internal personnel or third parties. Web applications are also subject to additional controls, if they are public facing, to address ongoing threats and vulnerabilities after implementation, as defined at PCI DSS Requirement 6.6.*			X	
6.4	Follow change control procedures for all changes to system components. The procedures must include the following:		X		SecureSphere Change Control Module can audit on configuration changes to database servers
6.4.1	Documentation of impact			X	
6.4.2	Management sign-off by appropriate parties			X	
6.4.3	Testing of operational functionality			X	
6.4.4	Back-out procedures			X	
6.5	Develop all web applications (internal and external, and including web administrative access to application) based on secure coding guidelines such as the Open Web Application Security Project Guide. Cover prevention of common coding vulnerabilities in software development processes, to include the following: ***Note:*** *The vulnerabilities listed at 6.5.1 through 6.5.10 were current in the OWASP guide when PCI DSS v1.2 was published. However, if and when the OWASP guide is updated, the current version must be used for these requirements.*		X		SecureSphere protects against each of the vulnerabilities identified by requirements 6.5.1 through section 6.5.10 – all of the vulnerabilities enumerated by the Open Web Application Security Project (OWASP) guideline – as well as many advanced vulnerabilities not covered by PCI or the OWASP Top Ten. Since SecureSphere capabilities go above and beyond the latest standards, it may be considered as a compensating control in certain scenarios. It's also worth noting that as threats change over time, SecureSphere is automatically updated by Imperva to provide protection against the latest vulnerabilities and standards. It is much more difficult to identify and deploy secure coding practices to developers as threats change. For more information about how SecureSphere protects against the OWASP Top Ten list of security vulnerabilities, please see the SecureSphere and OWASP Top Ten Technical Brief.
6.5.1	Cross-site scripting (XSS)		X		SecureSphere protects against cross-site scripting.
6.5.2	Injection flaws, particularly SQL injection. Also consider LDAP and Xpath injection flaws as well as other injection flaws.		X		SecureSphere protects against injection flaws, including SQL, LDAP, and Xpath injection flaws.

6.5.3	Malicious file execution		X		SecureSphere protects against malicious file execution.
6.5.4	Insecure direct object references		X		SecureSphere protects against insecure direct object reference.
6.5.5	Cross-site request forgery (CSRF)		X		SecureSphere protects against cross-site request forgery.
6.5.6	Information leakage and improper error handling		X		SecureSphere protects against information leakage and improper error handling.
6.5.7	Broken authentication and session management		X		SecureSphere protects against broken authentication and session management.
6.5.8	Insecure cryptographic storage		X		SecureSphere protects environments that use cryptographic storage.
6.5.9	Insecure communications		X		SecureSphere protects environments that use secure communications. In proxy mode, SecureSphere can secure communications by terminating HTTPS (SSL) connections.
6.5.10	Failure to restrict URL access		X		SecureSphere can restrict URL access to sensitive resources.
6.6	For public-facing web applications, address new threats and vulnerabilities on an ongoing basis and ensure these applications are protected against known attacks by either of the following methods: • Reviewing public-facing web applications via manual or automated application vulnerability security assessment tools or methods, at least annually and after any changes • Installing a web-application firewall in front of public-facing web applications	X			SecureSphere is the market leading Web Application Firewall. It automatically detects and blocks attacks before damage may occur. SecureSphere deployment meets this control requirement more cost effectively and with less security risk than repeated application review and fix cycles (see Application Reviews versus Web Application Firewalls section above).

Requirement 10: Track and Monitor all Access to Network Resources and Cardholder Data

Requirement 10, titled "Track and Monitor all Access to Network Resources and Cardholder Data", focuses on setting controls that establish an audit log of all access to network resources and cardholder data. It includes 25 extremely detailed sub-requirements that delineate not only what needs to be logged but also how those logs are managed. Since databases are the primary repository for cardholder data, each of these requirements apply directly to the database infrastructure. In the following paragraphs, we analyze the most challenging of these sub-requirements in the context of database security.

Database Performance and Storage

Requirement 10.2.1 requires the establishment of an audit trail that logs all individual access to card holder data. Such a broad logging requirement triggers the need for high volume transaction logging. Native database audit mechanisms (those that are "built-in" to the database) must write to local tables resulting in significant performance degradation even under moderate transaction loads. Native audit storage requirements also quickly consume valuable hard disk resources. Therefore, compliance with PCI DSS requirement 10.2.1 via native database audit mechanisms introduces significant performance and storage risk. It is often not simply a matter of "turning on" logging, but of redesigning, adding CPU capacity, adding storage capacity, and purchasing database licenses to account for additional CPUs. The costs of database audit mechanisms mount quickly.

Database Audit Appliances and Database Performance

By monitoring traffic flowing to and from each database, database audit appliances are able to meet the high volume audit requirements implied by PCI DSS requirement 10.2.1 without any risk to performance or any need to add database storage capacity. As independent network security devices, audit appliances also simultaneously meet PCI DSS requirement 10.5 provisions to secure the database audit trail so that it cannot be altered (see Table 2 below).

Database User Accountability for Applications

PCI DSS requirement 10.1 mandates a process for linking all access to system components to an individual user. By forcing users to access systems using unique user accounts and credentials, this requirement is met for most IT infrastructure. Login events and subsequent access behavior is tracked using standard audit log mechanisms.

However, there is one very common access scenario where user access activity is not linked to specific users via most traditional systems. That scenario is perhaps the most critical from a risk management perspective – access to cardholder databases through applications. Many applications, and most Web applications in particular, authenticate users and then pool all user connections through a single (or small number) of database connections to optimize performance. Thousands of different database users may be aggregated into a single connection to a front-end application. Thus, logs recorded natively by the database and those recorded by most audit appliances are not linked to any individual user, but to the application account name.

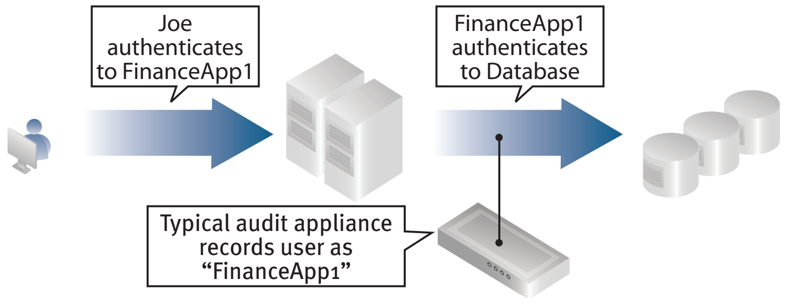

Joe authenticates to FinanceApp1
FinanceApp1 authenticates to Database
Typical audit appliance records user as "FinanceApp1"

Figure 1: Typical audit appliances and native database audit capabilities record application names, not actual user names.

Establishing User Accountability for Credit Card Access

Linking individual Web application users to specific database transactions is no simple task. Organizations essentially have three options: application rewrites, manual audit log correlation, and the Imperva SecureSphere Web and database appliances.

Application Rewrites

All applications accessing credit card data could potentially be rewritten to pass user names to backend databases or establish unique connections for each user. Not only is this an expensive, time consuming process with significant database performance risk, but it may be impossible for commercial applications provided by third party software vendors.

Manual Audit Log Correlation

In the event of a database breach via external Web applications, some organizations may attempt to indirectly identify the perpetrator by correlating database logs with Web logs. This is easier said than done. At any given time, hundreds or thousands of users may be logged into an application. Correlating the timestamps of application and database logs reduces the number of potential perpetrators, but does not usually deliver a definitive result. The auditor must match a specific application request to a corresponding database transaction by interpreting the details of each transaction (Web parameters, query parameters, etc.). The drawbacks to this process are as follows.

» **Expertise** – The extensive application and database design expertise required to carry out the manual correlation described above is not common and very expensive.

» **Cost** – The manual correlation process is extremely laborious and therefore costly even for a single application. Support for this process across many applications and database platforms extend costs further.

SecureSphere and Universal User Tracking

SecureSphere's Universal User Tracking technology makes individual users accountable under any authentication scenario by combining multiple user identification methods.

- **Direct Query** – Account names and source IPs identify DBAs with direct database query access privileges.
- **Shared Root Privilege** – OS hostnames and usernames identify users of shared root privilege accounts.
- **Pooled Application** – The SecureSphere Data Security Suite, which combines the capabilities of the Database Firewall and the Web Application Firewall, can identify the individual application users that perform database queries, even if user connections are pooled into one database session. Web application login and session information is correlated with database transactions to link specific credit card data access events to specific Web application users. Custom algorithms extend this feature to users who authenticate via three-tier (non-Web) business applications such as SAP, and Oracle EBS.

How can SecureSphere Help?

Table 2 details how Imperva's SecureSphere Web and Database Security Solutions can be applied to achieve compliance with PCI DSS requirement 10.

Table 2: SecureSphere Applicability to PCI DSS Requirement 10

Req #	Text	SecureSphere Applicability			SecureSphere Applicability Detail
		Direct	Partial	NA	
10.1	Establish a process for linking all access to system components (especially those with administrative privileges such as root) to each individual user.	X			SecureSphere audits all user access to database information and management functions by administrators. SecureSphere also captures Web application user names and audits all subsequent Web session activity associated with that specific user name. SecureSphere can uniquely combine Web and database security capabilities to link specific SQL queries to specific Web user names even when user information is not passed to the back-end database by the application (e.g. connection pooling).
10.2	Implement automated audit trails for all system components to reconstruct the following events:	X			SecureSphere audits on all database events specified for compliance with requirement 10.2. SecureSphere audits some of the Web application events required for compliance with requirement 10.2. In addition to auditing, SecureSphere can issue real-time alerts in the event of significant policy violations. For example, it may be important to know in real-time when a database administrator uses a decryption stored procedure to access a credit card table.
10.2.1	All individual accesses to cardholder data	X			SecureSphere audits all database access to cardholder data and link that access to specific users (see item 10.1).
10.2.2	All actions taken by any individual with root or administrative privileges	X			SecureSphere audits all database management actions by administrators.
10.2.3	Access to all audit trails	X			SecureSphere audits access to its own Web and database audit trail. SecureSphere can additionally audit access to data that resides within native database logs.
10.2.4	Invalid logical access attempts	X			SecureSphere audits invalid Web and database logical access attempts.

10.2.5	Use of identification and authentication mechanisms		X		SecureSphere audits Web and database authentication events.
10.2.6	Initialization of the audit logs	X			SecureSphere audits initialization of its own Web and database audit logs.
10.2.7	Creation and deletion of system-level objects	X			SecureSphere audits creations and deletions of system-level database objects (schemas, installation dangerous stored procedures, etc.).
10.3	Record at least the following audit trail entries for all system components for each event:	X			SecureSphere Web and database event logs include all of the data specified in requirement 10.3.
10.3.1	User identification	X			SecureSphere Web and database event logs include specific user names (see 10.1).
10.3.2	Type of event	X			SecureSphere Web and database event logs include event types.
10.3.3	Date and time	X			SecureSphere Web and database event logs include the date and time.
10.3.4	Success or failure indication	X			The SecureSphere Web Application Firewall records Web server response codes, indicating server errors. The SecureSphere Database Firewall and Database Activity Monitoring track database errors and login failures.
10.3.5	Origination of event	X			SecureSphere Web and database event logs include detailed event origination information.
10.3.6	Identity or name of affected data, system component or resource	X			SecureSphere Web and database event logs include detailed affected system information.
10.4	Synchronize all critical system clocks and times.		X		SecureSphere can synchronize itself with your network nntp or other time mechanism.
10.5	Secure audit trails so they cannot be altered.	X			SecureSphere audit logging is completely separate from the native database and Web application logging. Only authorized users can access audit logs. Logs can be signed and encrypted
10.5.1	Limit viewing of audit trails to those with a job-related need.	X			Only users with appropriate roles and permissions can access SecureSphere Web and database audit trails. Roles and permissions are customizable. All access to audit trails is itself logged by SecureSphere.
10.5.2	Protect audit trail files from unauthorized modifications.	X			Only users with appropriate roles and permissions can access SecureSphere Web and database audit trails. Roles and permissions are customizable. All access to audit trails is itself logged by SecureSphere.
10.5.3	Promptly back up audit trail files to a centralized log server or media that is difficult to alter.	X			Imperva SecureSphere audit trail files can be backed up via scheduled processes. Logs can be digitally signed and encrypted.
10.5.4	Write logs for external-facing technologies onto a log server on the internal LAN.		X		The SecureSphere Web Application Firewall, typically deployed as an external-facing technology, can write logs to a log server on the internal LAN.

10.5.5	Use file integrity monitoring or change detection software on logs to ensure that existing log data cannot be changed without generating alerts (although new data being added should not cause an alert).		X		SecureSphere Web and database audit logs can be signed and encrypted.
10.6	Review logs for all system components at least daily. Log reviews must include those servers that perform security functions like intrusion-detection system (IDS) and authentication, authorization, and accounting protocol (AAA) servers (for example, RADIUS). ***Note:*** *Log harvesting, parsing, and alerting tools may be used to meet compliance with Requirement 10.6*		X		SecureSphere Web and database audit reports and alerts enable this requirement. Workflow can be defined to track review activity and require sign-off.
10.7	Retain audit trail history for at least one year, with a minimum of three month immediately available for analysis (for example, online, archived, or restorable from back-up).		X		Imperva SecureSphere Web and database audit trail logs can be backed up and stored for a configurable period. Support for SAN, NAS and integration with log management from SenSage, ArcSight, Prism Microsystems, RSA, and others provides multiple options for long-term, high-volume data retention.

Requirement 3: Protect Stored Cardholder Data

Requirement 3, titled "Protect Stored Cardholder Data" focuses upon establishing controls that protect the confidentiality of stored cardholder data (a.k.a. -"data at rest"). It specifies detailed requirements for first minimizing storage of cardholder data and then encrypting the remaining data that must be stored. It also includes provisions for setting "compensating controls" to address technical or business constraints that prevent encryption. Compensating controls provide organizations with an option to implement alternative controls when an explicitly-stated requirement cannot be met due to technical or business constraints. Any compensating control implementation must be over-and-above other PCI DSS control requirements and their justification must be reevaluated on an annual basis.

Requirement 3.4 of the standard is the only requirement that refers to compensating controls as an explicit option. It refers to Appendix B for guidelines to implementing compensating controls for rendering cardholder data unreadable. However, before applying any of the compensating controls in Appendix B, the organization must document a legitimate business or technical constraint that prevents use of encryption. In the following paragraphs, we discuss some of the challenges associated with encrypting cardholder data and suggest several techniques for implementing the compensating controls specified in Appendix B.

Encryption Can Be a Problem

Although encryption is the preferred approach to comply with requirement 3.4, the existence of Appendix B is an acknowledgement by the authors of the PCI standard that many organizations find it difficult, if not impossible to encrypt cardholder data. Some of the typical reasons for not implementing encryption are as follows.

» **Cost** – The incremental hardware and software expense associated with encrypting large quantities of data is significant by itself. However, by far the most expensive aspect is usually in the area of database application development. Front-end applications have to be rewritten and often encryption changes database data types. In the end, the combination of hardware, software and development costs adds up to a massive implementation cost.

» **Performance** – Database encryption often has a major impact on performance. For example, if the card number is used as the primary key, the database must be reconfigured to support joins of encrypted data. In most cases, this dramatically reduces database performance.

» **Non-Existent Encryption Support** – Many legacy systems do not support encrypted databases. There is no way to practically implement encryption in these environments.

For these and other barriers to encryption, the compensating controls specified in Appendix B are a sensible option for achieving compliance with requirement 3.4.

Implementing Appendix B

In PCI DSS version 1.2, the PCI Security Standards Council created a framework for implementing compensating controls rather than delineating specific criteria. According to Appendix B in PCI DSS 1.2, compensating controls must:

1. **Meet the intent and rigor of the original PCI DSS requirement** – The SecureSphere Database Firewall, a dedicated database security and monitoring appliance, can be used as an alternative control for PCI requirements that govern the security, confidentiality, and integrity of cardholder data in transit and at rest. More specifically, SecureSphere may be used as a compensating control for requirement 3.4, rendering cardholder data unreadable. The SecureSphere Database Firewall addresses the underlying goal of requirement 3.4 – protecting stored cardholder data from unauthorized internal users and intruders.

2. **Provide a similar level of defense as the original PCI DSS requirement** – PCI requirement 3.4 is designed to protect cardholder data stored in primary storage such as databases as well as non-primary storage such as backup and audit logs. SecureSphere prevents unauthorized users and intruders from accessing cardholder data stored in databases through its access controls and database attack protection. SecureSphere provides internal network segmentation and IP and MAC address filtering. In addition, it can limit access to database data by user name, data type and source application and it can prevent common database attacks.

   To minimize the risk of database activity logs, SecureSphere masks credit card numbers in audit and security logs. SecureSphere enables payment card processors to meet the auditing requirements designated in PCI DSS section 10 without exposing them to additional data compromise.

3. **Be "above and beyond" other PCI DSS requirements** – The SecureSphere Database Firewall far exceeds the requirements in the PCI Data Security Standard by not only protecting databases from unauthorized access, but by also delivering a wealth of other security, assessment, and monitoring capabilities.

   With SecureSphere, organizations can scan their network and discover all databases, including rogue databases. Then SecureSphere can inspect databases for sensitive data such as credit cards. Locating credit card data on the network is the first step to securing stored cardholder data. In addition, SecureSphere can scan databases for over 500 vulnerabilities and configuration flaws. Coupled with its database security features, SecureSphere goes well "above and beyond" other PCI DSS requirements.

4. **Be commensurate with the additional risk imposed by not adhering to the PCI DSS requirement** – The SecureSphere Database Firewall addresses the overall intent of requirement 3.4 by reducing the overall risk of cardholder data compromise. In fact, SecureSphere can detect and stop unauthorized access to cardholder data that would be allowed by a database encryption solution. For example, many organizations allow internal users to view customer credit card numbers for business purposes. SecureSphere would detect and potentially block irregular behavior, such as queries with unusually large result sets or requests from unauthorized client applications. Organizations that rely solely on database encryption may still allow malicious internal users to access cardholder data.

Note: *All compensating controls must be reviewed and validated for sufficiency by a PCI DSS assessor.*

How Can SecureSphere Help?

Table 3 details how Imperva's SecureSphere Web Application Firewall and database security appliances can be applied to achieve compliance with PCI DSS requirement. Table 4 details how SecureSphere database security appliances can meet the compensating control conditions for database encryption as specified in requirement 3.4 and Appendix B.

Table 3: SecureSphere Applicability to PCI DSS Requirement 3

Req #	Text	SecureSphere Applicability			SecureSphere Applicability Detail
		Direct	Partial	NA	
3.1	Keep cardholder data storage to a minimum. Develop a data retention and disposal policy. Limit storage amount and retention time to that which is required for business, legal, and/or regulatory purposes, as documented in the data retention policy.		X		SecureSphere assessment capabilities can automatically locate cardholder information anywhere within the database infrastructure. This assessment can help in building a storage policy by determining where existing data (both minimally required and excessive) is stored. After policy is set, these assessments serve to document compliance with policy for auditors.
3.2	Do not store sensitive authentication data subsequent to authorization (even if encrypted). Sensitive authentication data includes the data as cited in the following Requirements 3.2.1 through 3.2.3:	X			SecureSphere assessment capabilities can automatically locate authentication information data anywhere within the database infrastructure. This assessment can help in building a storage policy by determining where existing data (both minimally required and excessive) is stored. After policy is set, theses assessments serve to document compliance with policy for auditors.
3.2.1	Do not store the full contents of any track from the magnetic stripe (located on the back of a card, contained in a chip, or elsewhere). This data is alternatively called full track, track, track 1, track 2, and magnetic-stripe data. ***Note:*** *In the normal course of business, the following data elements from the magnetic stripe may need to be retained:* • *The cardholder's name,* • *Primary account number (PAN),* • *Expiration date, and* • *Service code* *To minimize risk, store only these data elements as needed for business.* ***Note:*** *See PCI DSS Glossary, Abbreviations, and Acronyms for additional information.*	X			SecureSphere has dedicated database security rules that flag any inbound or outbound magnetic strip data. This can be used to audit database read and write activity that violate this requirement.
3.2.2	Do not store the card-validation code or value (three-digit or four-digit number printed on the front or back of a payment card) used to verify card-not-present transactions. ***Note:*** *See PCI DSS Glossary, Abbreviations, and Acronyms for additional information.*	X			SecureSphere has dedicated database security rules can flag inbound/ outbound card-validation codes. This can be used to audit database read and write activity that violate this requirement.
3.2.3	Do not store the personal identification number (PIN) or the encrypted PIN block.		X		If stored as part of the raw "Track Data", SecureSphere can flag inbound/ outbound PIN. This can be used to audit database read and write activity that violate this requirement.

3.3	Mask PAN when displayed (the first six and last four digits are the maximum number of digits to be displayed). ***Notes:*** *This requirement does not apply to employees and other parties with a specific business need to see the full PAN.* *This requirement does not supersede stricter requirements in place for displays of cardholder data – for example, for point-of-sale (POS) receipts.*		X		SecureSphere masks PAN within its log files. It does not provide active masking as data is in use by applications.
3.4	Render PAN, at minimum, unreadable anywhere it is stored (including on portable digital media, backup media, in logs) by using any of the following approaches: • *One-way hashes based on strong cryptography* • *Truncation* • *Index tokens and pads (pads must be securely stored)* • *Strong cryptography with associated key management processes and procedures* • *The MINIMUM account information that must be rendered unreadable is the PAN.* ***Note:*** *If for some reason, a company is unable to render the PAN unreadable, refer to Appendix B: Compensating Controls.* ***Note:*** *"Strong cryptography" is defined in the PCI DSS Glossary, Abbreviations, and Acronyms.*		X		SecureSphere masks PAN within its log files. It does not provide active masking as data is in use by applications. SecureSphere completely meets the compensating controls for encryption conditions as specified in Appendix B. For more information in this paper, see Implementing Appendix B in the previous section.
3.4.1	If disk encryption is used (rather than file- or column-level database encryption), logical access must be managed independently of native operating system access control mechanisms (for example, by not using local user account databases). Decryption keys must not be tied to user accounts.			X	
3.5	Protect cryptographic keys used for encryption of cardholder data against both disclosure and misuse:			X	
3.5.1	Restrict access to cryptographic keys to the fewest number of custodians necessary.			X	
3.5.2	Store cryptographic keys securely in the fewest possible locations and forms.			X	
3.6	Fully document and implement all key management processes and procedures for keys used for cryptographic keys used for encryption of cardholder data, including the following:			X	
3.6.1	Generation of strong cryptographic keys			X	
3.6.2	Secure cryptographic key distribution			X	
3.6.3	Secure cryptographic key storage			X	
3.6.4	Periodic cryptographic key changes As deemed necessary and recommended by the associated application (for example, re-keying); preferably automatically At least annually			X	

3.6.5	Retirement or replacement of old or suspected compromised cryptographic keys			X	
3.6.6	Split knowledge and establishment of dual control of cryptographic keys			X	
3.6.7	Prevention of unauthorized substitution of cryptographic keys			X	
3.6.8	Requirement for cryptographic key custodians to sign a form stating that they understand and accept their key-custodian responsibilities			X	

Table 4: SecureSphere Applicability to PCI DSS Appendix B

Req #	Text	SecureSphere Applicability			SecureSphere Applicability Detail
		Direct	Partial	NA	
	Compensating controls must satisfy the following criteria:				
1.	Meet the intent and rigor of the original PCI DSS requirement.	X			The SecureSphere Database Firewall can act as a compensating control for PCI requirements intended to protect, monitor, or assess cardholder data in transit or at rest. SecureSphere can be used as a compensating control for requirement 3.4, rendering PAN (Primary Account Number) unreadable anywhere it is stored. The SecureSphere Database Firewall, deployed to protect sensitive databases, meets the intent and rigor of PCI DSS requirement 3.4.
2.	Provide a similar level of defense as the original PCI DSS requirement, such that the compensating control sufficiently offsets the risk that the original PCI DSS requirement was designed to defend against. (See Navigating PCI DSS for the intent of each PCI DSS requirement.)	X			According to Navigating PCI DSS, requirement 3.4 is designed to protect cardholder data in primary storage such as databases as well as non-primary storage such as audit logs from unauthorized internal users and intruders. Since audit and troubleshooting logs have to be retained, businesses can prevent disclosure of data in log files by masking or removing PANs. SecureSphere prevents unauthorized users and intruders from accessing primary account numbers stored in databases. SecureSphere can provide internal network segmentation and IP and MAC address filtering. In addition, it can restrict access to cardholder data by user name, data type, and application. Furthermore, SecureSphere prevents common application and database attacks with its robust database security engine. SecureSphere also masks primary account numbers stored in its database audit logs. Using SecureSphere, organizations can retain detailed database audit trails without exposing companies to cardholder data compromises.

[2] Payment processors should consult with their Qualified Security Assessors to determine whether compensating controls may be considered.

3.	Be "above and beyond" other PCI DSS requirements. (Simply being in compliance with other PCI DSS requirements is not a compensating control.)	X			The SecureSphere Database Firewall goes above and beyond other PCI DSS requirements by providing a dedicated database security device that polices all database activity. SecureSphere meets the intent of the original requirement by preventing authorized users and intruders from viewing and downloading PANs.
4.	Be commensurate with the additional risk imposed by not adhering to the PCI DSS requirement.	X			SecureSphere mitigates the risk imposed by not rendering stored cardholder data unreadable. By protecting stored cardholder data in databases and by masking primary account numbers stored in audit logs, SecureSphere is commensurate with database encryption or truncation. In addition, SecureSphere provides additional controls beyond encryption by monitoring and controlling access by authorized users. Since many organizations must allow some internal users to view cardholder data, cardholder data can be compromised even if stored data is encrypted. SecureSphere monitors database queries and automatically recognizes aberrant activity such as a request for an unusually large number of records or access from an unusual database application.

SecureSphere

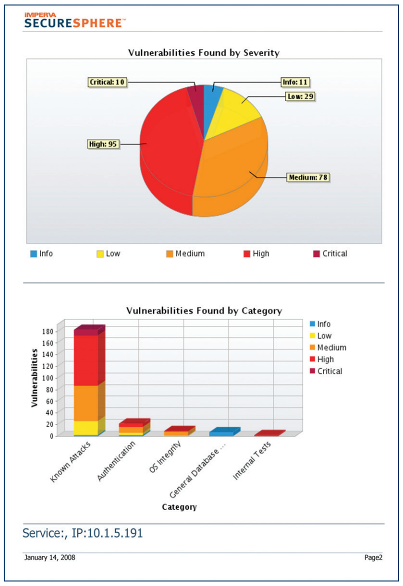
Imperva SecureSphere Web and Databases Security Solutions monitor, audit, and secure business application data. Together SecureSphere Data Security Solutions comprise the most complete, automated family of products that protect information that is stored in databases and accessed through business applications.

SecureSphere is the only solution designed from the ground up to meet the high volume database auditing and advanced security capabilities mandated by PCI. As a result, over half of SecureSphere customer deployments are driven by PCI.

Key Features and Benefits

Transparent Inspection enables deployment in sensitive datacenters without change to the network, applications, or databases. There is also no risk to application or database performance.

Dynamic Profiling automatically builds detailed models of normal application and user behavior. Profiles can
be reviewed, modified, and approved by administrators before being applied for granular access control. The scope of profile granularity extends from user names to cookies, parameters, tables, columns, query text, and much more.



Service:, IP:10.1.5.191

January 14, 2008 Page2

Figure 2: Imperva's extensive PCI experience is reflected in SecureSphere's targeted PCI reports.

Imperva's SecureSphere products are widely used by organizations to monitor, audit, and secure business application data for PCI regulatory requirements. Three appliance families are available: the Web Application Firewall, the Database Security Gateway, and the Database Monitoring Gateway. Together they comprise the most complete, automated platform for protecting information that is stored in databases and accessed through business applications.

SecureSphere is the only solution designed from the ground up to meet the high volume database auditing and advanced security capabilities mandated by PCI. As a result, over half of SecureSphere customer deployments are driven by PCI.

Business Relevant Reporting document security and compliance status with relevance to specific business environments. Out-of-the-box compliance reports accelerate audit against regulations/best practice frameworks including PCI, SOX, and HIPAA. Reports focus not only on regulatory criteria, but on business applications such as SAP and Oracle EBS.

Database Security Assessment identifies key areas of risk and documents compliance with PCI requirements and best practices. Assessments include vulnerabilities, configurations, database discovery, sensitive data discovery (PAN, etc.), and user behavior analysis (shared accounts, etc.).

ADC Insight Services subscription ensures that SecureSphere is updated as new attacks and regulatory requirements emerge. Updates include signatures, assessments, compliance reports and more.

Correlated Attack Validation (CAV) correlates events across multiple SecureSphere security layers (Web, database, IPS, etc.) and over time to identify complex events and attacks with unmatched accuracy.

Real-Time Alerting enables immediate response to significant violations. For example, when a DBA uses a decryption procedure to access a credit card, SecureSphere can alert or block in real-time.

For More Information

For more information on SecureSphere capabilities and related regulatory compliance requirements, visit www.imperva.com.

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.

Imperva
Americas Headquarters
3400 Bridge Parkway
Suite 101
Redwood Shores, CA 94065
Tel: +1-650-345-9000
Fax: +1-650-345-9004

International Headquarters
125 Menachem Begin Street
Tel-Aviv 67010
Israel
Tel: +972-3-6840100
Fax: +972-3-6840200

Toll Free (U.S. only): +1-866-926-4678
www.imperva.com