PCI-DSS and Application Security

Achieving PCI DSS Compliance with Seeker

ACCURACY

CLARITY

SIMPLICITY

Summary

Abstract

The Payment Card Industry Data Security Standard, commonly referred to as PCI-DSS is a leading standard with which organizations that handle payment data such as credit or debit cards are required to comply.

Defined by the Payment Card Industry Security Standards Council, the standard was created to increase controls around cardholder data to reduce credit card fraud via its exposure. Validation of compliance is done annually by an external Qualified Security Assessor (QSA) that creates a Report on Compliance (ROC) for organizations handling large volumes of transactions, or by Self-Assessment Questionnaire (SAQ) for companies handling smaller volumes.[

This paper discusses PCI DSS and the vital role it plays in building secure software applications. It will focus on specific requirements that deal with the protection and transmission of cardholder data, regular testing of security systems and processes, which are all essential in establishing strong application security.

Seeker maps all critical data in the application, especially payment card information and data related to authentication, and then tracks these data as they traverse the application to ensure the application does not expose it to risks.

The paper will demonstrate how Quotium Technologies' Seeker helps in achieving PCI DSS compliance. It tackles each requirement and explains how Seeker addresses them.

This paper can provide valuable information regarding PCI DSS compliance to:

- Merchants who develop software applications dealing with customer payments

  or

- Software companies who build those applications

PCI DSS – Statistics

The Payment Card Industry Data Security Standards (PCI DSS) apply to organizations or merchants who accept customer payments through credit or debit cards.

Research firm Ponemon Institute has been able to quantify the cost of cyber attacks, although the financial cost is only one of many.

In its “2012 Cost of Cyber Crime Study”, Ponemon found the average annualized cost of cybercrime (per company) in the US to be at $8.9 million per year. That's about 6% more than it was the previous year. Parallel studies conducted in Germany, Japan, Australia, and the United Kingdom revealed average annualized costs (in USD) of approximately $5.9M, $5.2M, $3.4M, and $3.3M, respectively. Although all costs are lower than the US figure, these numbers are still large enough to cause significant damage to any business.

By complying with PCI DSS, you will be able to strengthen your defenses, eliminate vulnerabilities, and significantly reduce the chances of a data breach. In fact, you shouldn't comply with PCI DSS just for the sake of compliance; rather, you should comply because it is critical for your business.

PCI DSS – Application Security

The success and impact of a cyber-attack largely depends on how secure are the organization software applications. When applications have serious vulnerabilities, a cyber-attack will easily succeed and its impact can be considerable.

In the US edition of the Ponemon study mentioned earlier, three of the most expensive types of cyber-attacks, namely malicious insiders, malicious code, and web-based attacks, account for 54% of cyber-crime cost. These three are also the most difficult to resolve, requiring an average of 57.1 days for malicious insiders, 50.3 days for malicious code, and 37.9 days for web-based attacks. The success rate and impact of these particular attacks can be considerably reduced by strong application security.

Because application security plays an important role in countering cyber-attacks, it is given the utmost importance in PCI DSS. Security requirements governing software development are inscribed in major Requirement 6, which charges merchants to “develop and maintain secure systems and applications”.

Other requirements that have implied prescriptions for software development and application security can be found under major Requirements 3, 4, 8, and 11. These requirements specify standards for the protection of stored cardholder data, encryption of cardholder data during transmission, assignment of a unique ID to each person who has computer access, and regular testing of security systems and processes, respectively.

How Seeker Helps You Achieve PCI DSS Compliance

Quotium's Seeker helps you meet PCI DSS requirements with minimal effort. It integrates into the software development lifecycle, identifying vulnerabilities early in the application lifecycle, before they become a liability to your organization.

Seeker does this by conducting simulated attacks and analyzing code as it runs in response to those attacks. At the same time, it closely monitors how the code handles sensitive data as the data flows through all application tiers and components. To eliminate false positives and obtain a more accurate assessment of the potential impact and risk to business, the simulated attacks are based on real world exploits.

Seeker is data centric, meaning all vulnerabilities are assessed in relation to how they affect business critical data. It is therefore the best solution to comply with requirements that concern application data security.

Let us now take a closer look at PCI DSS requirements and discuss how Seeker helps meet them.

Requirement 3: Protect stored cardholder data

Requirement Details	Achieving Compliance with Seeker
3.1 – Keep cardholder data storage to a minimum by implementing data retention and disposal policies, procedures and processes	In order to implement data retention and disposal policies effectively, it is first needed to identify what data needs to be retained or disposed and where these data are located This is no easy task considering the volume of data many organizations handle every day. Seeker automatically identifies payment card information. If further customization is needed, Seeker allows the configuration of user-defined sensitive data. It then uses this knowledge during runtime to monitor the application, seek out the data in question, and track its flow.

	This makes it easier to know where payment card data is stored and whether data retention and disposal policies are violated.
3.2 - Do not store sensitive authentication data after authorization (even if encrypted) **Sensitive authentication data includes:** **Full contents of the magnetic stripe located on the back of the card, which includes the cardholder's name, PAN, expiration date, service code, etc.** **Card verification code or value (CAV2/CVC2/CVV2/ CID)** **Personal identification number (PIN) or PIN block**	Seeker monitors all sensitive data throughout its lifespan in the application to ensure they are never stored. Seeker's unique technology allows the monitoring of web-service, database and file system calls in the path of sensitive data. PCI DSS storage requirements are not limited to primary storage or data repositories such as databases and flat files (e.g. text files and spreadsheets). It also applies to non-primary storage like backups, audit logs, and exception or troubleshooting logs.
3.4 – Render PAN (Primary Account Number) unreadable anywhere it is stored (including on portable digital media, backup media, and in logs)	In addition to being able to identify and monitor PAN and other sensitive authentication data, Seeker can also determine whether they are ever stored unencrypted.

Requirement 4: Encrypt cardholder data over open public networks

PCI-DSS Requirement		Achieving Compliance with Seeker
4.1	**Use strong cryptography and security protocols (e.g. SSL/TLS, IPSEC, SSH, etc.) to safeguard sensitive cardholder data during transmission over open, public networks**	Seeker identifies sensitive data, monitors where it flows, and determines whether it is encrypted or not, regardless whether the data is at rest or in motion. Seeker alerts to the lack of SSL, but it also alerts specifically to payment card information being transmitted insecurely.

Requirement 6: Develop and maintain secure systems and applications

PCI-DSS Requirement		Achieving Compliance with Seeker
6.3	**Develop software applications in accordance with PCI DSS, and based on industry best practices, and incorporate information security throughout the software development life cycle (SDLC). These include:**	Seeker easily integrates throughout the development and testing processes of an SDLC, and ensures that application security is applied throughout the development cycle.
6.3.1	**Removal of custom application accounts, user IDs, and passwords before applications become active or are released to customers**	Seeker alerts the user when it identifies sensitive information such as hardcoded application passwords.
6.3.2	**Review of custom code prior to release in order to identify any potential coding vulnerability.**	Seeker integrates into the development lifecycle, and as part of the SDLC it allows organizations to easily test all new code prior to release.
6.5	**Develop applications based on secure coding guidelines. Prevent common coding vulnerabilities in software**	By analyzing the application from the inside, Seeker gains visibility into the code and internal workings of the application. This

development processes, to include the following:	visibility allows it to provide detailed information regarding the exact location of the vulnerability right within the application code. That's not all. The information is accompanied with context-based recommendations and secure code samples that can guide developers in correcting the vulnerability. Consequently, the developers can learn secure coding best practices as they remediate each vulnerability.
Injection flaws, particularly SQL injection. Also consider **LDAP and XPath injection flaws as well as other injection vulnerabilities.**	Seeker's ability to spot vulnerabilities that might lead to an SQL injection isn't based on theory or speculation. Seeker actually monitors the application during runtime and observes the malicious input as it traverses the application and arrives at the data layer. Seeker sees the internals of the application during run-time for accurate, false positive free detection. This applies also to LDAP queries, LDAP, XPATH and more. Seeker tracks data throughout application components and tiers and monitors these data as they arrive at the database, directory or file repository calls. Seeker then attempts to exploit this access to verify that it could actually be exploited by an attacker.
Insecure cryptographic storage	Seeker monitors data flow during runtime and reports precisely where information is stored unencrypted. This includes databases, file repositories, debug information, and other repositories.
Insecure communications	Seeker can tell whether and where data are transmitted as clear text, so you will know exactly where data-in-motion encryption is needed.
Improper error handling	Applications sometimes inadvertently leak confidential information through error messages. These leakages may include

	security configurations, internal workings, or payment card data. Because Seeker can detect a variety of built-in and user-defined sensitive information types, it can check error messages to see if any sensitive information appears there.
All "High" vulnerabilities identified in the vulnerability identification process As defined in PCI DSS	Seeker accurately assesses the impact and classification of each vulnerability's corresponding risk through simulated exploits and data analysis. This feature makes Seeker an invaluable tool in risk ranking activities and, consequently, in identifying high risk vulnerabilities.
Cross-site scripting (XSS)	XSS allows attackers to execute scripts on a victim's browser and enables them to hijack the user's sessions, alter websites, distribute worms, and perform a host of other malicious activities. Seeker has a unique JavaScript and VBScript analysis engine which identifies cross site scripting and verifies it can be exploited by using simulated attacks. In addition, by analyzing data, Seeker is able to provide unique insight in testing for Persistent Cross Site Scripting.
Improper Access Control (e.g. insecure direct object references, failure to restrict URL access, and directory traversal)	Attackers take advantage of direct object references to gain unauthorized access to other objects. Direct object references are created when developers unwittingly expose references to internal implementation objects such as files, directories, keys, or database records through URLs or form parameters. By identifying and tracking data in the system Seeker identifies whether any references affect data and by modifying them it is possible for an attacker to access privileged information.
Cross-site request forgery (CSRF)	Using CSRF, an attacker can take advantage of a victim's browser by forcing it to

		automatically send a malicious pre-authenticated request to a web application while the legitimate user is logged on. Seeker detects CSRF vulnerabilities which have an actual impact on application operations (for example vulnerabilities which allow file writing operations, or reading from database tables), and only operations which pose a real threat are then reported to the user.
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.**	Web-facing applications are exposed to ongoing threats and can be under attack any time. These attacks often succeed because of insecure coding practices. A regular review on these applications is therefore crucial in preventing attacks from succeeding. Seeker applies a new and highly effective approach of Runtime Code & Data analysis Seeker integrates seamlessly into the software development processes. It becomes part of existing workflow and introduces application security testing as part of ongoing processes. Seeker tracks security flaws at each step of development as well as at every release of the product.

Requirement 8: Assign a unique ID to each person with computer access

PCI-DSS Requirement	Achieving Compliance with Seeker
8.4 Render all passwords unreadable during transmission and storage on all system components using strong cryptography.	User passwords are sometimes stored in a database or transmitted over the network as clear text. In these situations, they can be easily obtained by anyone who can penetrate the database or intercept the transmission. Seeker identifies many kinds of sensitive information, including authentication data like passwords. It can also determine whether the information is encrypted. When passwords are detected, Seeker tracks their flow and maps areas where they fail to be protected by strong encryption.
8.5 Ensure proper user identification and authentication management for non-consumer users and administrators on all system components. **User identification and authentication management best practices involve a number of activities. These include implementing strong password policies, locking out users who fail to meet certain security conditions (e.g. users who make too many failed login attempts), and requiring users to re-authenticate when an idle session limit is reached.**	Seeker verifies whether identification and authentication management best practices are in place. Seeker reports when a weak password policy is being implemented or whether the system accepts weak passwords. Seeker also checks whether the system does not lock user accounts even after a specified number of failed login attempts has been exceeded. It also monitors the length of an idle session and determines whether the idle session limit has been violated.

Requirement 11: Regularly test security systems and processes

PCI-DSS Requirement	Achieving Compliance with Seeker
11.3 Perform external and internal penetration testing at least once a year and after any significant infrastructure or application upgrade or modification.	The purpose of conducting a penetration test on a particular environment is to simulate a real world attack scenario and determine the extent by which the attack can penetrate into that environment. Seeker conducts simulated attacks and then analyzes code during run-time to learn how the application responds to those attacks. But it does not stop there. To avoid false positives, Seeker further tests its own findings to verify whether the detected vulnerabilities are actually exploitable in real world attacks. Furthermore, Seeker integrates into existing R&D methodology, so that every time the code changes, it can move in and carry out the appropriate tests.

Conclusion

PCI DSS offers extensive guidance in achieving strong application security. However, it shouldn't be considered the ultimate yardstick. Meaning, even if you have fully complied with all its requirements, that still wouldn't guarantee a fully impenetrable system. Cyber criminals always come up with new kinds of attacks. Application security undertakings should therefore be an ongoing process.

Seeker can play a vital role in that process. As demonstrated throughout this paper, Seeker possesses the necessary elements for achieving PCI DSS compliance. But more importantly, because of Seeker's versatility and ability to closely scrutinize application code and track data flow through all application tiers and components in real-time, it can discover even the most inconspicuous vulnerabilities and help developers build considerably more secure software applications.

About Quotium Technologies

Quotium Technologies specializes in the development of innovative software solutions to guarantee the security and functionality of business-critical applications throughout their life cycle.

Seeker is the leader of the new generation of application security testing software. Easily integrating with existing software testing processes, Seeker allows developers to efficiently develop secure applications.

For more information www.quotium.com or info@quotium.com