GUIDE

Seeker and PCI DSS Compliance

Building real-time security and compliance at the speed of business

Overview

Vulnerabilities can appear anywhere in the card-processing ecosystem: in point-of-sale devices, mobile devices, personal computers, servers, wireless hotspots, and web shopping applications, through the transmission of cardholder data to service providers, and in remote access connections. These weaknesses can also extend to systems operated by service providers and acquirers, the financial institutions that maintain relationships with merchants accepting payment cards.

According to Privacy Rights Clearinghouse, more than 11.5 billion records with sensitive information were breached between January 2005 and February 2019. This statistic proves the need for businesses that store, process, or transmit cardholder data to implement standard security procedures and technologies that prevent the theft of sensitive information. Compliance with the technical and operational requirements set by the Payment Card Industry Data Security Standard (PCI DSS) helps to alleviate vulnerabilities and protect cardholder data.

Seeker, our interactive application security testing solution, is the ideal choice when you must demonstrate PCI DSS and PA DSS compliance. Seeker generates strong documentation that meets security assessors' expectations for quality of evidence—the most effective way to fulfill information assurance requirements.

Benefits

Sensitive-data tracking capability

The primary differentiator for Seeker in PCI DSS environments is its sophisticated sensitive-data leak checker, which we designed specifically to help organizations ensure that they handle cardholder data and personally identifiable information (PII) properly. Seeker helps monitor and track any type of sensitive user data, including national ID, cardholder data, account data, transaction information, medical information, biometric data, and geographical data. Mishandling of this type of information is a significant contributor to information leakage and the most common cause of failed audits.

High accuracy with unique active verification

PCI DSS requires applications to be free of certain defect types. Defect detection is an area of high risk, as an organization that fails to achieve the necessary defect detection rate will have a secondary compliance issue. Seeker's strong runtime verification and analysis algorithms help it yield the highest true-positive defect detection rate. A high true-positive rate minimizes the possibility of defects surfacing later in the software development life cycle (SDLC). Fixing these defects early leads to significant cost savings and risk reduction; it also has a direct benefit in terms of meeting compliance obligations.

Seamless integration into automated development and testing environments

Seeker enables DevOps, development, and QA teams to become security champions by integrating security testing into their existing workflows. Seeker continuously monitors, verifies, and validates all detected vulnerabilities in the background while the application is under test. It prioritizes all identified vulnerabilities in order of severity and reports them in real time. Its seamless integration with automation tools makes it an ideal fit for both manual and automated functional and CI/CD test environments.

Comprehensive vulnerability and security risk reporting

Seeker's dashboard shows applications' compliance to standards such as CWE/SANS Top 25, OWASP Top 10, GDPR, CAPEC, and more. Its vulnerability findings are mapped to the current OWASP Top 10 and other standards. Users can customize, build, and share security risk reports in multiple formats easily and quickly.

How Seeker helps you meet PCI DSS requirements

Seeker is the tool of choice for those needing to meet PCI DSS and other compliance obligations.

PCI DSS requirement	Addressing compliance
6.1: Establish a process to identify security vulnerabilities, using reputable outside sources for security vulnerability information, and assign a risk ranking (for example, as "high," "medium," or "low") to newly discovered security vulnerabilities.	Seeker provides a continuous vulnerability and remediation process for web applications. Risk rankings are based on industry best practices, consideration of CVSS base score, and/or classification and potential impact by Synopsys. When Seeker discovers new security vulnerabilities, it categorizes them by impact/risk (high, medium, or low). Developers can sort and view defects based on impact/risk.
6.3: Develop internal and external software applications (including web-based administrative access to applications) securely, as follows: • In accordance with PCI DSS (for example, secure authentication and logging) • Based on industry standards and/or best practices. • Incorporating information security throughout the software-development life cycle	Based on industry standards and best practices, Seeker enables you to "build security in" throughout the SDLC, whether you're developing internal or external applications. Seeker provides the strongest means of achieving the necessary level of information security assurance. It generates strong documentation to assist with compliance activities in accordance with PCI DSS.
6.4.5.3: Functionality testing to verify that the change does not adversely impact the security of the system.	Seeker continuously monitors, detects, and verifies common application security weaknesses during functional testing to ensure changes do not adversely affect the security of applications.
6.5: Address common coding vulnerabilities in software-development processes as follows: • Train developers at least annually in up-to-date secure coding techniques, including how to avoid common coding vulnerabilities. • Develop applications based on secure coding guidelines.	One of the best ways to teach developers how to address coding vulnerabilities is to explain why a vulnerability was flagged as exploitable (including the source code) and give them information on the flow of tainted data from source to sink. Seeker provides these details, as well as remediation advice based on secure coding guidelines. It also offers contextual learning with built-in eLearning support. Developers can attain in-depth knowledge and stay up-to-date on secure coding practices for many programming languages.
6.5.1: Injection flaws, particularly SQL injection. Also consider OS Command Injection, LDAP and XPath injection flaws as well as other injection flaws.	Seeker checks for these injection flaws (including second-order injection): 6.5.1 SQL INJECTION 6.5.1 REFLECTION INJECTION 6.5.1 LDAP INJECTION 6.5.1 XPATH INJECTION 6.5.1 HIBERNATE INJECTION 6.5.1 NoSQL INJECTION 6.5.1 REMOTE FILE INCLUSION 6.5.1 LOCAL FILE INCLUSION

6.5.1 (cont.)	6.5.1 LOG INJECTION 6.5.1 CONNECTION STRING INJECTION 6.5.1 HTTP HEADER INJECTION 6.5.1 COOKIE INJECTION 6.5.1 EXPRESSION LANGUAGE INJECTION 6.5.1 SERVER SIDE JS INJECTION 6.5.1 XML EXTERNAL ENTITY INJECTION 6.5.1 UNSAFE DESERIALIZATION INJECTION 6.5.1 COMMAND INJECTION
6.5.3: Insecure cryptographic storage	6.5.3 WEAK ENCRYPTION 6.5.3 SENSITIVE INFO SAVED UNENCRYPTED 6.5.3 WEAK HASH 6.5.3 WEAK HASH ALGORITHM USED WITH SENSITIVE DATA
6.5.4: Insecure communications	6.5.4 SERVER SIDE CODE CONSUMES INSECURE WEB SERVICE 6.5.4 WEAK CRYPTO USED IN SSL 6.5.4 INSUFFICIENT SSL ENFORCEMENT 6.5.4 SENSITIVE INFO SENT IN URL 6.5.4 SESSION ID SENT IN URL
6.5.5: Improper error handling	6.5.5 MISSING CUSTOM ERROR PAGE
6.5.6: All “high risk” vulnerabilities identified in the vulnerability identification process (as defined in PCI DSS Requirement 6.1).	Seeker meets this requirement by categorizing certain vulnerabilities as “high impact/risk” as defined by PCI DSS Requirement 6.1. Seeker’s unique engine automatically verifies critical vulnerabilities, submits tickets, and sends alerts to developers for immediate remediation for a broad range of vulnerability classes, including those listed below. For more details, please contact Synopsys. 6.5.6 DIRECTORY TRAVERSAL 6.5.6 DIRECTORY TRAVERSAL (SECOND ORDER)
6.5.7: Cross-site scripting (XSS)	6.5.7 CROSS SITE SCRIPTING 6.5.7 CROSS SITE SCRIPTING (SECOND ORDER)
6.5.8: Improper access control (such as insecure direct object references, failure to restrict URL access, directory traversal, and failure to restrict user access to functions)	This list is not exhaustive. Please contact Synopsys for the latest additions. 6.5.8 CLICKJACKING 6.5.8 INSECURE REDIRECT 6.5.8 SERVER-SIDE REQUEST FORGERY 6.5.8 DOS VIA BLOCKING CALL 6.5.8 SENSITIVE INFO IN BROWSER CACHE

6.5.8 (cont.)	6.5.8 INSECURE HTTP METHOD (TRACE ENABLED) 6.5.8 INSECURE HTTP METHOD (HEAD ENABLED) 6.5.8 USER-DEFINED SENSITIVE INFO STORED IN BROWSER CACHE 6.5.8 REGULAR EXPRESSION DENIAL OF SERVICE (ReDoS) 6.5.8 INSECURE REDIRECT (SECOND ORDER) 6.5.8 SERVER-SIDE REQUEST FORGERY (SECOND ORDER)
6.5.9: Cross-site request forgery (CSRF)	6.5.9 CROSS-SITE REQUEST FORGERY (CSRF)
6.5.10: Broken authentication and session management	6.5.10 NO SESSION EXPIRATION 6.5.10 INSUFFICIENT COOKIE PROTECTION (MISSING 'HTTP ONLY') 6.5.10 INSUFFICIENT COOKIE PROTECTION (MISSING 'SECURE ONLY') 6.5.10 INSECURE AUTHENTICATION MECHANISM IS IN USE (BASIC HTTP AUTH) 6.5.10 IMPROPER LOGOUT 6.5.10 SESSION FIXATION
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 an automated technical solution that detects and prevents web-based attacks (for example, a web-application firewall) in front of public-facing web applications, to continually check all traffic.	Web-based applications suffer constant attacks, which often succeed because of increasingly complex and insecure development practices. Most traditional application security testing tools can't trace this client traffic through back-end pathways without extensive tailoring or simulation of the complex traffic required for detection. Therefore, it's crucial to complement your manual or automated approach with an interactive tool such as IAST to catch runtime vulnerabilities early in the application life cycle. Seeker provides continuous assessment and review of complex applications by simply observing applications and APIs as they run in real time. It can pinpoint identified vulnerabilities down to the line of source code, provide detailed insights into application behavior during runtime, track and assess the dataflow, and recommend remediation where appropriate.
8.2.3: Passwords/passphrases must meet the following: Require a minimum length of at least seven characters. Contain both numeric and alphabetic characters. Alternatively, the passwords/passphrases must have complexity and strength at least equivalent to the parameters specified above.	8.2.3 WEAK PASSWORD POLICY 8.2.3 WEAK PASSWORD DB CONNECTION

Ready to learn more? Contact our Sales team

The Synopsys difference

Synopsys helps development teams build secure, high-quality software, minimizing risks while maximizing speed and productivity. Synopsys, a recognized leader in application security, provides static analysis, software composition analysis, and dynamic analysis solutions that enable teams to quickly find and fix vulnerabilities and defects in proprietary code, open source components, and application behavior. With a combination of industry-leading tools, services, and expertise, only Synopsys helps organizations optimize security and quality in DevSecOps and throughout the software development life cycle.

For more information, go to www.synopsys.com/software.

Synopsys, Inc.
185 Berry Street, Suite 6500
San Francisco, CA 94107 USA

Contact us:
U.S. Sales: 800.873.8193
International Sales: +1 415.321.5237
Email: sig-info@synopsys.com