FORTRA®

GUIDE *(Digital Defense)*

The Complete Guide to Application Security for PCI-DSS

Nothing can be more devastating to a business than losing the ability to take credit cards as payments from customers. Businesses and consumers alike prefer them, with over 55% of payments coming from credit or debit cards. However, if the appropriate security steps to protect the personal information that comes with handling credit card transactions aren't taken, organizations can lose the ability to process card payments. Maintaining Payment Card Information Data Security Standard (PCI DSS) compliance is key to avoiding this financially devastating consequence.

Meeting compliance requirements outlined in the PCI-DSS is about more than checking off compliance boxes. The rules in PCI-DSS are designed to help businesses improve their security game and legitimately protect the data they collect from customers. These rules encompass technology controls, processes, and methodologies for protecting the entire infrastructure dealing with payment cards.

In this ebook, we focus on PCI areas related to protecting software that processes or handles payment cards and related information. Defending development and infrastructure systems helps form the foundation of security to meet PCI-DSS standards.

PCI Requirements

When creating any software for use with payment processing, PCI-DSS has particular requirements and controls that must be in place to protect data and keep the software and infrastructure safe. The approach is holistic and covers every aspect, from how the data is collected, handled, and stored to the infrastructure and processes that support it.

The Latest PCI DSS

PCI-DSS has long been the gold standard for financial and retail industries for defining the security of sensitive cardholder data. Rather than simply being a checkbox of what controls must be implemented, PCI-DSS v4.0.1, shifts to a newer outcome-based set of requirements. PCI-DSS was updated in 2023 to include two new requirements: 6.4.3: Manage Payment Page Scripts to Prevent Skimming and 11.6.1: Deploy a Mechanism to Detect Skimming. The new approach lists the intent of the requirement and focuses less on the specific way in which a control is met and more on the outcome and whether it meets the expected criteria.

This opens up more options for the solutions organizations can use to implement security. Rather than looking for solutions that solely meet PCI requirements, they can instead investigate new solutions that meet organizational pain points and align with the intent of the PCI requirements. A full list of PCI-DSS updates and requirements are kept in the PCI Security Standard Library.

Software Security Testing

According to PCI-DSS 6.3, any software used in relation to payment cards, whether developed in-house or externally, must be developed securely and with industry standards and best practices. The goal is to proactively identify potential attack vectors in the software product and harden it in a way that is more challenging to attack. Part of this requires building security into the entire software development lifecycle to identify and address security concerns from the design phase through development and release to production.

The evaluation phase of the software development lifecycle includes evaluating the software for known vulnerabilities and attacks (PCI-DSS 6.6). Using static application security testing (SAST), dynamic application security testing (DAST), and software composition analysis (SCA) tools, developers can discover flaws in the code used. With this information, these flaws can be prioritized by their potential risk and, if significant, remediated.

Managing Vulnerabilities

Vulnerabilities in software not only come from the application itself, but also from the infrastructure that it runs on. The operating system and other applications on host machines can have configuration issues and vulnerabilities that allow attackers to circumvent security controls to gain access. In the highest impact cases, attackers can gain complete control over a host, with full access to its data along with all processes and memory. Even if the host does not retain and store data, it may reside in memory, allowing attackers to steal payment card information before it is transmitted.

Managing vulnerabilities under PCI-DSS 11.2 requires scanning solutions to evaluate the security posture of endpoints and network devices. Scanners evaluate installed software, open ports, and available services to determine if they contain known vulnerabilities. This information is used by security teams to prioritize and plan remediation efforts.

Implementation of Processes

Maintaining PCI-DSS compliance requires more than one-time assessments of software and infrastructure. It necessitates a set of processes to ensure that these efforts and the remediation of results are carried out on a known cadence.

When new vulnerabilities are identified, there need to be existing workflows to ensure that the vulnerabilities are prioritized by risk (PCI-DSS 6.1) and then dealt with promptly but following a set change control process (PCI-DSS 6.4). In the case of installed applications and vulnerable system components, identified vulnerabilities must be resolved within a month (PCI-DSS 6.2) of a vendor-supplied patch release. To avoid being caught off guard by these requirements, organizations need scheduled processes to ensure timely evaluation and remediation.

" ...identified vulnerabilities must be resolved within a month of a vendor supplied patch release. To avoid being caught off guard ...organizations need scheduled processes to ensure timely evaluation and remediation ."

Code Analysis

Code analysis is crucial for identifying security vulnerabilities, design defects, logical errors, and implementation flaws, as PCI-DSS requires. All of these might be considered bugs and can lead to situations where information gathered or processed by an application could potentially be disclosed. With modern codebases' current size and scale, a manual investigation is no longer sufficient to catch everything. Instead, automated tools are necessary to efficiently and effectively test the code and its implementation.

Static Analysis

SAST is one of the most common forms of testing for developers. It analyzes the code itself for common programming errors that could lead to vulnerabilities. It cross-references existing code against industry standards such as:

- Common Weakness Enumeration (CWE)
- SANS TOP 25
- OWASP TOP 10
- CERT Secure Coding Guidelines

SAST should be implemented early in the development process and throughout the development lifecycle as is required by PCI-DSS 6.3. This approach provides feedback, allowing developers to resolve issues long before production. Early remediation of issues also decreases the overall cost of resolution, as resolution costs dramatically increase after code goes into production.

Early identification of vulnerabilities is also vital to cost-effectively meet PCI-DSS requirements. This is especially important regarding PCI-DSS, which requires vulnerabilities in production to be quickly and efficiently remedied after they are identified. When code is developed in-house, it is the company's responsibility to prioritize remediation.

Dynamic Analysis

DAST has a different approach to testing than SAST. Rather than focusing on the code itself, DAST focuses on implementing the application. This does not mean that the application needs to be developed entirely before DAST can be utilized, but it does need to have some basic functionality that can be tested.

DAST works by using a variety of inputs to identify areas where applications behave unexpectedly. Developers coding an application generally only define a handful of expected input ranges and build with these ranges in mind. Using DAST, comprehensive testing occurs, where the large varieties of inputs are systematically entered into the application, thus testing more broadly than standard QA. Two major advantages to using DAST: It produces virtually no false positives and does not require access to source code. Black Box Fuzzing, a subcategory of DAST, provides even more insight into security weaknesses as it uncovers unknown or unpublished vulnerabilities. DAST is considered an industry best practice as it is one of the requirements for meeting PCI-DSS 6.3. Using DAST against web-based applications is a portion of a robust vulnerability assessment required by PCI-DSS 6.6.

Vulnerability Management

No matter how much effort is put into implementing the proper controls and hardening systems, it can all be undone by the right vulnerability. This is highlighted by recent vulnerabilities discovered in Log4J, a standard library for logging that had long been considered safe. The Log4J vulnerability allowed remote code execution (RCE). Using this RCE, attackers were able to compromise the endpoints. Vulnerabilities such as this are precisely why PCI-DSS requires organizations to have entire processes to analyze, identify, and remediate vulnerabilities as they are discovered.

Analyze Systems

Analyzing systems to identify vulnerabilities is the first step in a comprehensive vulnerability management program and vital to meeting PCI-DSS 6.1. Using tools that analyze endpoints and devices for known vulnerabilities, reports are generated that list the vulnerabilities discovered. Not every exposure has an available method of exploitation, so discovered vulnerabilities need to be assessed and organized by the potential risk. Risk calculations take into account the potential impact of the vulnerability being exploited and its likelihood of exploitation.

It is important to note that just because a vulnerability does not currently have a known exploit does not mean it will always lack an exploit. New attack methods are constantly being developed that are targeted toward known vulnerabilities. It is vital that the vulnerability assessment software leverages known vulnerability databases to stay on top of new developments in the exploitability of known vulnerabilities and newly discovered vulnerabilities.

Automate Processes

Vulnerability analysis is not a one-time assessment of your infrastructure. New vulnerabilities are constantly being discovered; it requires consistent re-evaluation that checks against up-to-date information.

Automation is an integral part of the continuous re-evaluation of vulnerabilities and is essential for meeting PCI-DSS 6.2. Assessment tools that include automation take the guesswork out of maintaining the vulnerability management process. Automation ensures that these steps happen on a scheduled basis rather than relying on employees to remember to update vulnerability signatures and run the scanners. This generates consistent results for evaluation, empowering your organization to catch newly discovered vulnerabilities before attackers can use them.

Plan to Remediate

A crucial part of vulnerability management is eliminating prioritized vulnerabilities in a consistent and controlled process. Generally, remediating a vulnerability requires applying patches to the application with the vulnerability. Unfortunately, this can come with unintended consequences that may include the patch causing the application to no longer work, creating a conflict with other running applications, or even creating a different, new vulnerability. This is why testing all patches in a test environment that mirrors production is strongly advised.

With vulnerability management, it is not a matter of if you will have to patch but when. Establishing a process for validating the deployment of patches is a requirement of PCI-DSS 6.4 and a best practice for avoiding unintended outages of service. By creating an entire remediation process, systems can be thoroughly tested after patch deployment, ensuring that no new vulnerabilities are created from the patching process.

Doing it Right

Meeting PCI-DSS requires more than a single solution. You need a combination of solutions and processes that combine to improve your organization's security posture. Implementing consistent processes into the development cycle will help produce more stable and secure code.

With Fortra VM you get a complete industry-leading vulnerability management solution that is both accurate and easy to use. Fortra VM allows you to perform comprehensive security assessments. Our built-in tracking system helps you manage and prioritize your results, making remediation more efficient and effective, which is essential to maintaining PCI-DSS compliance.

FORTRA

Fortra.com

About Fortra

Fortra provides advanced offensive and defensive security solutions that deliver comprehensive protection across the cyber kill chain. With complete visibility across the attack chain, access to threat intelligence spanning the globe, and flexible solution delivery, Fortra customers can anticipate criminal behavior and strengthen their defenses in real time. Break the chain at fortra.com.