WHITE PAPER

# Software Composition Analysis in the Payment Card Industry

## Protecting cardholder data and meeting PCI Software Security Standards

# Executive Summary

The payment card industry continues to evolve. For the last 30 years function-specific secure hardware has been a dominant component of protecting cardholder data. During the last 11 years there has been a significant shift in consumer behaviour driven by the broadening of capabilities across the Payment Card Industry. Software has always been at the heart of payments, existing within dedicated devices from specialist suppliers. Now the industry is creating payment software for use in a wide range of consumer-based, off-the-shelf products. How software is being developed has also changed. Agile development is the standard development method to achieve the economy, efficiency and effectiveness that developers and consumers require. In response to these changes and the increasing demand, The Payment Card Industry Security Standards Council (PCI SSC) published security standards for the development and management of payment application software that stores, processes or transmits cardholder data[i].

With the increased availability and use of open source components in software development there is also an increase in probability that a developer will use components which have known vulnerabilities. This paper is designed for organizations and teams that rely on the DevOps cycle of software code in the context of creating secure software for payments. We take a look at the standards published by the PCI SSC, the role of Software Composition Analysis (SCA) in complying with the standards for payment systems application development, and the importance of people, processes and technology relative to managing the software supply chain.

> IN 2019, THE AVERAGE COST OF A DATA BREACH WAS **$3.92 MILLION**.
>
> *—THE PONEMON INSTITUTE*

## Understanding the PCI Software Security Framework

The number and cost of data breaches globally across all industries has risen with the explosion of the internet and e-commerce. In the *2019 Cost of a Data Breach Report* from the Ponemon Institute, in 2019 the average cost of a data breach was $3.92 million. At $5.86M, the financial services industry experienced an average total cost of a data breach higher than less regulated industries. According to IronNet Cybersecurity in a 2019 report, 20% of IT security leaders said their organization was attacked six or more times annually, and 80% said they had experienced at least one cybersecurity incident over the last 12 months that was so severe it required a board-level meeting. Breaches and cyberattacks are by no means slowing down.

The software industry has a role to play in the protection of cardholder data, and the PCI Security Standards Council as part of their Software Security Framework has introduced the PCI Secure Software Requirements and Assessment Procedures and the PCI Secure Software Lifecycle (Secure SLC) Requirements and Assessment Procedures. The aim of security standards published by the PCI SSC is to protect payment account data throughout the payment lifecycle and to enable technology solutions that devalue this data and remove the incentive for criminals to steal it. Software specific standards are a recognition of the role that developers have in creating solutions that are secure in their coding and the entire software development lifecycle.

## PCI Standard References

### SECURE SOFTWARE LIFECYCLE AND ASSESSMENT PROCEDURES – SECTION 3.2.A

Control Objective: Threats to the software and weaknesses within its design are continuously identified and assessed.

Section 3.2.a – [documented testing process which]… *accounts for the entire code base, including detecting vulnerabilities in third-party, opensource, or shared components and libraries.*

Section 3.2b – *The vendor* ***monitors vulnerabilities in open-source components throughout their use*** *or inclusion in the vendor's software to determine when new vulnerabilities are identified*.

### SOFTWARE SECURITY FRAMEWORK – SECTION 10 – THREAT AND VULNERABILITY MANAGEMENT

Control Objective: Vulnerabilities in the software and third-party components are tested for and fixed prior to release.

Section 10.2b – [documented testing process which]… *accounts for the entire code base, including detecting vulnerabilities in third-party, opensource, or shared components and libraries.*

The new standard requires organizations to not just "keep an eye on" their open source software (OSS) use, but it requires software companies to continuously identify and assess weaknesses within software applications. This includes the complete software supply chain. Specifically, the PCI Secure Software Requirements and Assessment Procedures provides a baseline of requirements with corresponding assessment procedures and guidance. This includes accounting for the entire codebase, and detecting vulnerabilities in third-party, open source, or shared components and libraries. In addition to this enhanced governance, key security principles addressed in the Secure SLC Standard include threat identification, vulnerability detection and mitigation, security testing, change management, secure software updates, and stakeholder communications.

Vulnerability and Software Composition Analysis solutions are key in addressing open source security, compliance and risk management.

# Modern Software Development and the Role of Software Composition Analysis (SCA)

Modern applications are typically composed of 90% open source components[ii]. Comparing this to legacy applications which are 90% proprietary code gives an indication to the changes that have happened in software development, creating a change in the risk landscape and how software now needs to be managed.

Because of the software development practice of leveraging third-party software for the engineering, production and delivery of applications, it has created what has become known as the software supply chain. Software components within the supply chain include attributes such as a license and version number.

A Gartner technology insight report shows that[iii]:

- 6% of codebases contain at least some open source
- 40% of those components contain at least one high-risk vulnerability
- Concerns over the long-term viability of packages and the presence of security vulnerabilities were cited as the most significant challenges faced in using open source

If the open source software supply chain is not managed through security and compliance processes, attackers can exploit gaps and carry out malicious activities. An example of what can happen is Equifax's massive data breach in 2017. Attackers exploited vulnerabilities in Apache Struts 2, a popular open source framework. The breach is estimated to have cost Equifax $4 Billion.

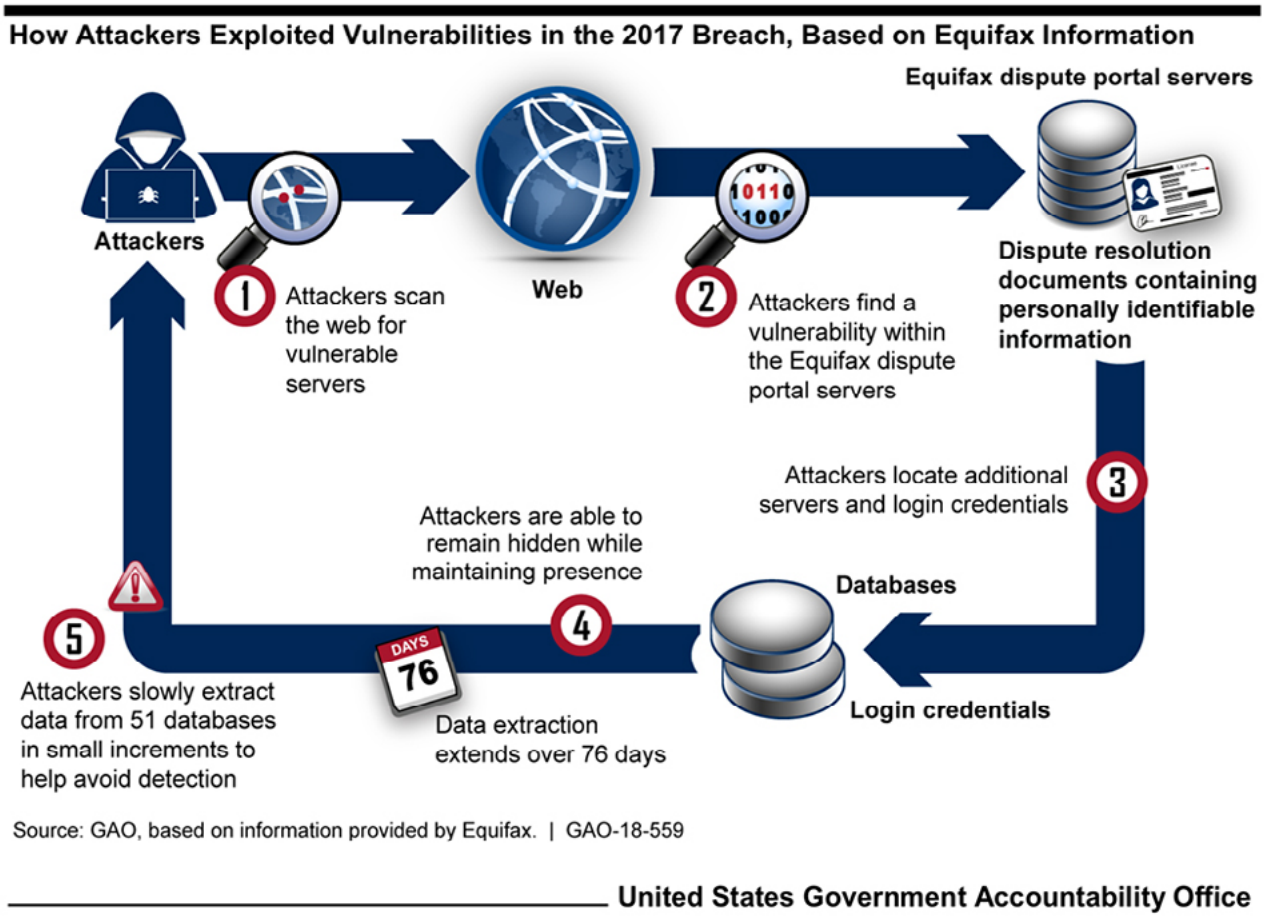


*Figure 1 Extract from United States Government Accountability Office Report GAO – 18 – 559*

According to Graeme Payne, former Equifax SVP and CIO, "At the time that the breach was announced, I wasn't even aware that we were running Apache Struts in the particular environment."

The Equifax breach is just one of many, and because security breaches overall (cross-industry) are on the rise — 4.1 billion records impacted in the first half of 2019 alone — providers of payment applications need to do everything possible to demonstrate to users that their software supply chain is managed[iv]. HeartBleed, ShellShock, and GlibC are just a sample of other well-known software vulnerabilities. Having central control, knowing what is in code, and where it's being utilized should be a top priority for both open source creators and users in the Payment Card Industry.

This has led to the development of tools, processes and evolving best practices and standards to help providers manage and secure their software supply chain by creating an environment where software is secure by design. This engineering process is referred to as DevSecOps, where the collaborative framework of DevOps includes security as a shared responsibility (See Fig 2).

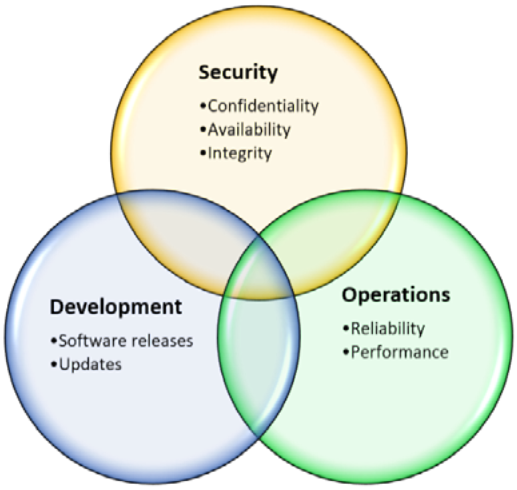


*Figure 2 DevSecOps*

## Defining Software Composition Analysis

Software Composition Analysis (SCA) is the process of automating the visibility into open source software use for the purpose of risk management, security and license compliance. With the rise of open source software use in software development across all industries including the payments industry, the need to track components increases exponentially to protect companies from issues and open source vulnerabilities. Open source manual tracking and ad hoc audits are both difficult and impractical, requiring automation to scan source code, binaries and dependencies. Automating SCA allows for proactive tracking of vulnerabilities early in the development cycle, helping organizations avoid any engineering risk with applications from the start. Alerts about new vulnerabilities in delivered applications also creates a dynamic environment for security risk management which is in agreement with PCI SLC requirements.

SCA helps organizations build an inventory of open source software components and their attributes such as:

- Licensing and copyright information — Are there licensing obligations that create a risk?
- Security vulnerability information — Are there know security vulnerabilities in components?
- Operational information — How well supported are components?

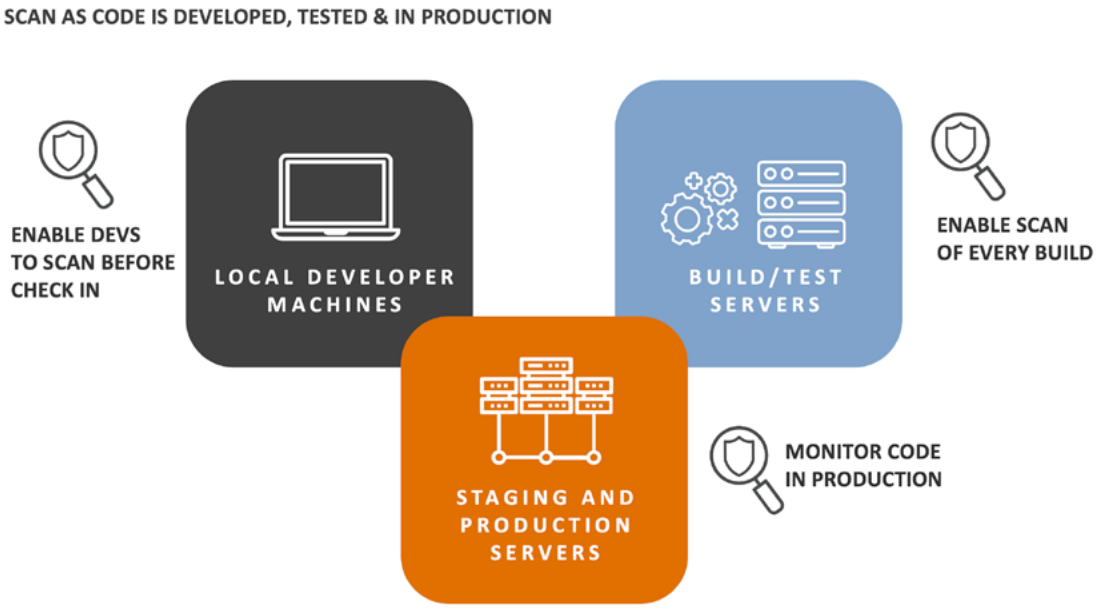


*Figure 3 Continuous SCA Scanning*

This inventory is referred to as a software Bill of Materials (BoM), which is produced by a SCA solution. A good solution would report on known security vulnerabilities and licensing issues but can also help with copyright and quality code management. Applying SCA and continuous open source scanning to payment solutions creates transparency and control, allowing payment providers to validate how they are properly managing the security of payment software throughout the entire software lifecycle and thus, meeting the requirements set forth by the PCI SSC.

| Name | Component | License | Vulnerabilities |
|---|---|---|---|
| Apache Struts 2.3.1 (Apache 2.0) | apache-struts - 2.3.1 | Apache License 2.0 | 73 \| 43 25 5 0 |
| Apache Struts 2.3.12 (Apache 2.0) | apache-struts - 2.3.12 | Apache License 2.0 | 72 \| 43 24 5 0 |
| Apache Struts 2.3.14.1 (Apache 2.0) | apache-struts - 2.3.14.1 | Apache License 2.0 | 70 \| 41 24 5 0 |
| glibc 2.3.7 [Bundled with ossimage 7.4.1e ] (GNU General Public License) | glibc - 2.3.7 | GNU General Public License | 70 \| 20 43 7 0 |
| pcsc-lite 1.5 [Bundled with ossimage 7.4.1e ] (BSD) | pcsc-lite - 1.5 | BSD-Style License | 7 \| 0 5 2 0 |
| bash 2.05 [Bundled with ossimage 7.4.1e ] (GPL-2.0) | bash - 2.05 | GNU General Public License v2.0 | 69 \| 65 1 3 0 |
| Apache Struts 2.3.14.2 (Apache 2.0) | apache-struts - 2.3.14.2 | Apache License 2.0 | 69 \| 40 24 5 0 |
| Apache Struts 2.3.14.3 (Apache 2.0) | apache-struts - 2.3.14.3 | Apache License 2.0 | 67 \| 38 24 5 0 |
| devise 1.5.0 (MIT) | devise - 1.5.0 | MIT-Style License | 6 \| 1 2 3 0 |
| jquery 1.6.2 (MIT or GPL-2.0) | jquery - 1.6.2 | None Selected | 6 \| 0 6 0 0 |

# Automating Software Supply Chain Management

If an organization is new to managing an open source software supply chain, they may start by performing an audit of their software code internally or through a software audit service provider. Issues are discovered and prioritized, and recommended remediation steps identified. This is more of a point-in-time approach and doesn't create an ongoing, continuous and repeatable scenario for payment software companies. Audit services really lend themselves to specific events such as Merger and Acquisition (M&A) and other due diligence efforts where third-party open source validation may be contractual. A more automated approach addresses the ongoing dynamic nature of open source management for the payment card industry. This increases in importance given the number of open source components available to developers is rising, which also leads to an increase in the number of vulnerabilities being discovered and reported. Fortunately, this problem can be managed and automated via a combination of people, processes and technology.

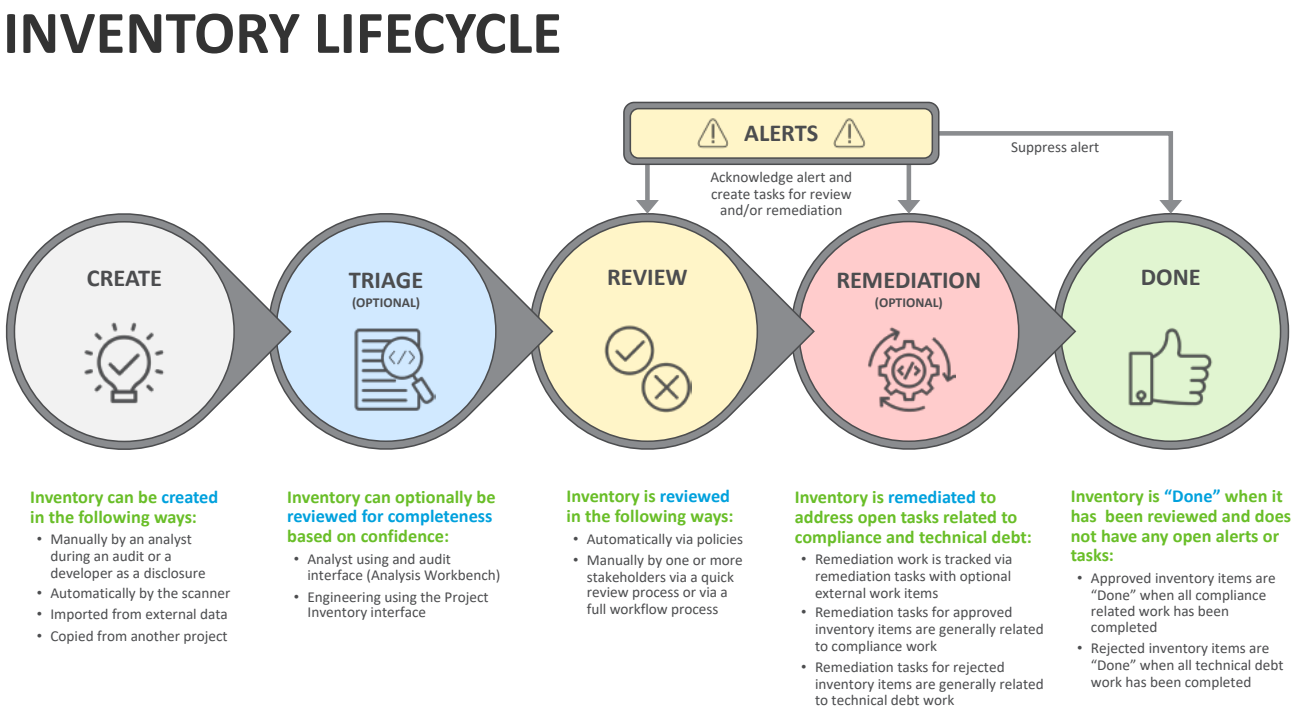


*Figure 5 The Lifecycle of a BOM Item*

For payment software providers the first step is bringing all areas of the organization together — from leadership to engineering to operations — to define a policy and process for managing open source security and license compliance, and then integrating SCA technology into the development environment enabling a complete and wholistic approach to a secure by design principle.

At the start of the software engineering process, unsecure open source components can be identified and therefore avoided earlier in the development process. At the delivery end of the process a Software Bill of Materials (SBOM) can be generated which gives an accurate, complete roll call of the secure components being used. This helps build trust in the software supply chain and is especially important if products are shipped outside the organization to customers.

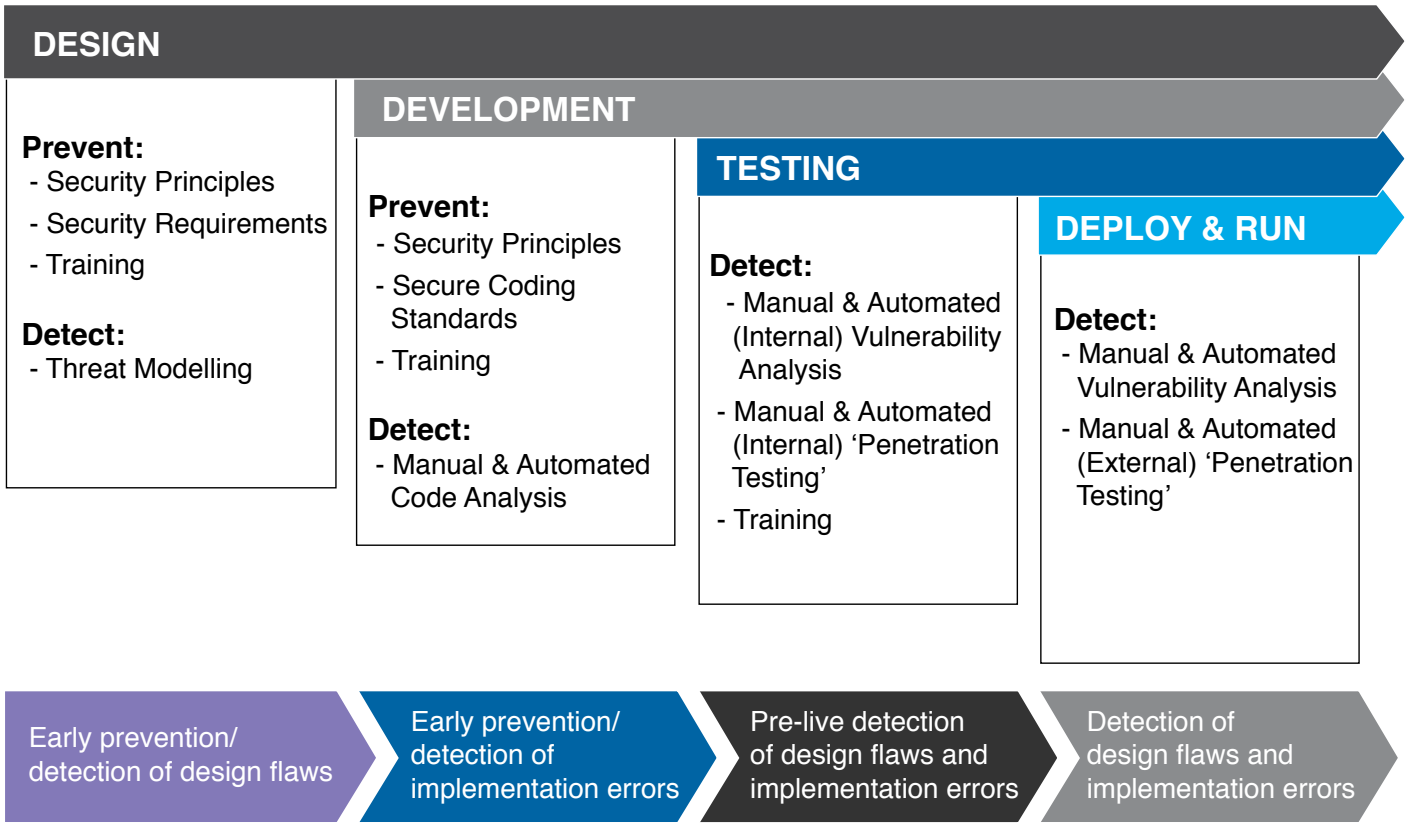


*Figure 6 Design Security into all DevOps phases*

In the age of agile software development combined with the PCI SSC's secure software development and testing standards, payment software suppliers need a strategy that includes an automated, end-to-end open source scanning approach that supports detection of both security and open source software license compliance issues throughout the DevOps lifecycle.

## Key Takeaways

Any entity that stores, processes, and/or transmits cardholder data or has an impact on the security of cardholder data needs to be cognizant that the Payment Card Industry has evolved its security standards for software. New standards were introduced in 2019 and it is our prediction that the PCI SSC will continue to evolve its framework for the development and maintenance of secure software. It is for the software developer and their stakeholders to ensure that all software has been developed in accordance with the PCI software security framework and its associated standards.

The PCI SSC does not and cannot predict the future. The PCI SSC publishes security standards as a response to visible trends. Therefore, while the PCI SSC Software Security Framework and associated standards are relatively new, the underlying trends indicating the need for these changes has been clear and evident for several years. We only scratched the surface with the breach and security statistics provided as evidence in this paper.

Net? For the payment card industry there is no need to wait. Use what is happening now to demonstrate to your customers and users that your software development lifecycle is secure and that the security of cardholder data is deeply embedded within the fabric of your business.

Contractual relationships between organizations and their card payment counterparts means that compliance with the security standards published by the PCI SSC should always be written into the contracts with organizations accepting payments for goods or services. Credit card brands drive the compliance relationship that the acquirer has with their client organizations.

Change can create stress and pain as vendors scramble for compliance to meet industry standards and wider regulations such as the General Data Protection Regulation (GDPR). The PCI SSC does its job to help make the transition for vendors smooth by laying out implementation timelines. For most organizations, however, PCI Software Security Standards are one of many top-of-mind business initiatives, which is why working with experienced partners is a proven method of simplifying the problem of payment security and compliance and taking on a progressive approach to demonstrating proper secure software practices.

**Payment software vendors should embark on an open source compliance and risk management journey by:**

- Creating a consistent, repeatable process for software vulnerability and risk management
- Gaining buy-in from critical stakeholders across the organization, including executives, developers, engineers, and legal
- Setting and enforcing policies for remediation
- Creating a complete and accurate Bill of Materials for all applications to meet the requirement of accounting for the entire codebase
- Making open source scanning an ongoing best practice effort, including integrating an SCA solution into your Engineering process

# About the Authors

**Martin Callinan** has over 20 years' experience in the software industry specializing in software licensing, IT Governance and risk avoidance. He has seen the challenges of risk management related to various aspects of the software ecosystem. Martin is now focused on assisting organizations leveraging the benefits of open source software to create bespoke applications in-house or through third parties while managing the business risks of intellectual property, open source component licensing, copyrights, security vulnerability management, and operational risk.

Martin is actively involved in the open source software industry to help create best practices and industry standards for the adoption and management of open source-based software. He is a member of The Linux Foundation's OpenChain project and Software Freedom Conservancy.

Source Code Control
*Open Source Risk Management Specialists*

**Michael Christodoulides** is a cyber security, risk and assurance professional with extensive experience of managing and delivering risk and assurance assessments across the payment card industry and its stakeholder groups.

Michael believes that businesses which deploy effective security controls in accordance with internationally recognized standards are also enabled to grow. This is because they can focus on doing what they do best without unplanned distractions caused by breaches in data security.

Michael has previously represented payment card industry stakeholders as a representative to the Board of Advisors of the Payment Card Industry Security Standards Council (PCI SSC), the UK Finance Acquirer Working Group, and was previously global Co-chair of the PCI SSC Taskforce responsible for developing secure practices for merchants.

**Kendra Morton** is Senior Product Marketing Manager at Revenera focused on Software Composition Analysis and open source scanning solutions. Kendra has 20+ years of experience leading all areas of technology marketing including software, data and analytics, cloud offerings, and customer management solutions. At Revenera Kendra supports the development of product strategy and positioning that align with the company's business goals and help drive the continued success of Revenera's valued end users.

revenera

[i] PCI Security Standards https://www.pcisecuritystandards.org/

[ii] Fischer, Donald. "How Managed Open Source Boosts Developer Productivity and Saves Money." TheNewsStack, August 2019. https://thenewstack.io/how-managed-open-source-boosts-developer-productivity-and-saves-money/

[iii] Gardner, Dale. "Technology Insight for Software Composition Analysis." Gartner, 1 November 2019. https://www.gartner.com/en/documents/3971011/technology-insight-for-software-composition-analysis

[vi] 2019 Mid-Year Quick View Data Breach Report. https://pages.riskbasedsecurity.com/2019-midyear-data-breach-quickview-report