EBOOK

revenera

Your Code Isn't Static

Evolve Your Open Source Processes

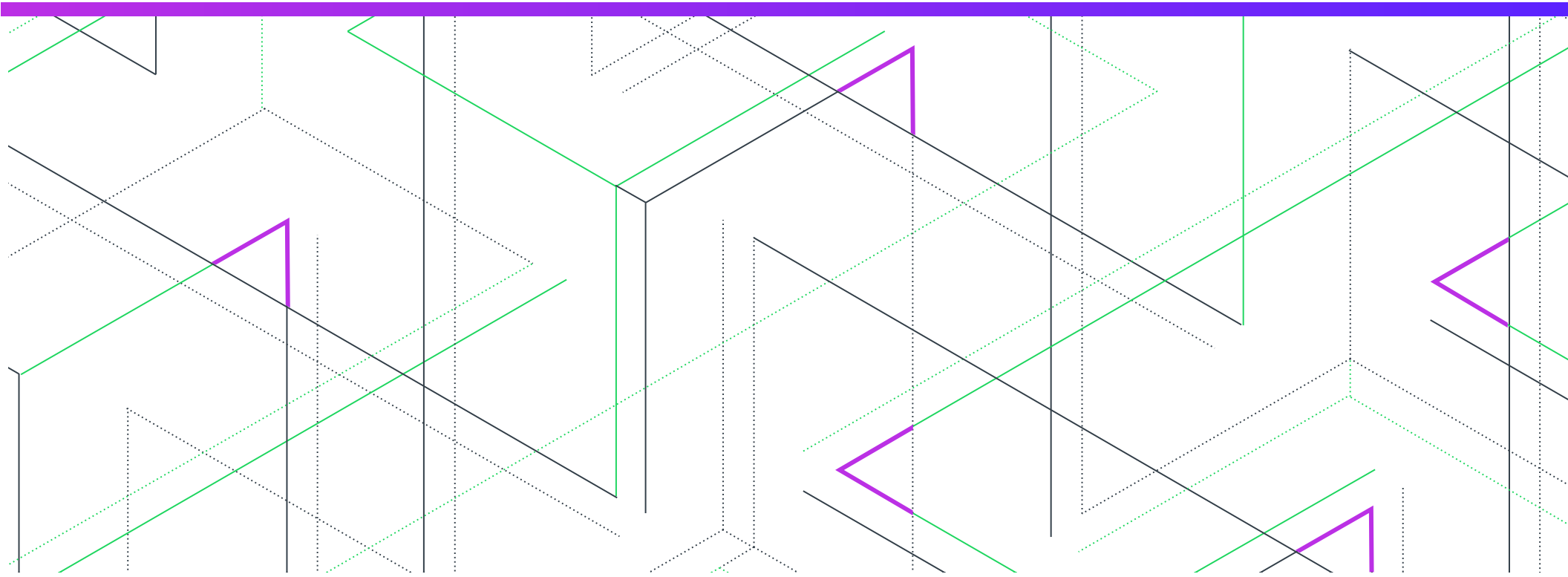


Table of Contents

Tie the DevOps Knot

Marrying DevOps and Software Composition Analysis

Companies have adopted an open source-first mentality for both the software they are acquiring as well as for software they develop and ship to customers. At least 80 percent—and in some cases 90 percent—of software code is open source.[1] Managing inherent license compliance and security risks that come with open source, despite the many benefits it provides, is about marrying the DevOps process with a Software Composition Analysis solution. It's also imperative to make sure processes supporting this environment are dynamic in order to meet the changing needs of both open source license requirements and market shifts.

Create a Dynamic DevOps Environment

1. Automatically, at a moment in time, understand what's in your code
2. Drive agility to reduce time-intensive, complex procedures for developers and engineers
3. Support rapid innovation to meet the changing nature of applications and open source software

[1]Based on 2019 Revenera audit services data

56% of CIOs expect to implement Agile software development, DevOps, or a similar flexible IT delivery model to increase IT responsiveness and help support broader innovation initiatives.[2]

[2]Deloitte Insights: Tech Trends 2020

Providing Clarity

What is Software Composition Analysis?

Open source is pervasive. Developers demand open source software to build applications and to get their jobs done. With the rise in use, the need to track components increases exponentially to protect companies from vulnerabilities.

Software Composition Analysis (SCA) is the process of automating the visibility into open source use for the purpose of license compliance, IP, and security risk management. Because the majority of software creation includes OS, manual tracking is difficult, requiring the need to use automation to scan source code, binaries and dependencies.

A SCA solution allows for the secure risk management of open source use throughout the software supply chain, allowing security and development teams to:

- Create an accurate Software Bill of Materials (SBOM) to give developers insight into potential security and licensing issues.
- Through automation, discover and track all open source used throughout the software supply chain.
- Set and enforce policies for open source use, licenses, and remediation guidance.
- Enable proactive and continuous monitoring in order to create actionable alerts for newly discovered vulnerabilities in both current and shipped products.
- Seamlessly integrate open source code scanning into the build environment to allow for early scanning.

"Expand Left"

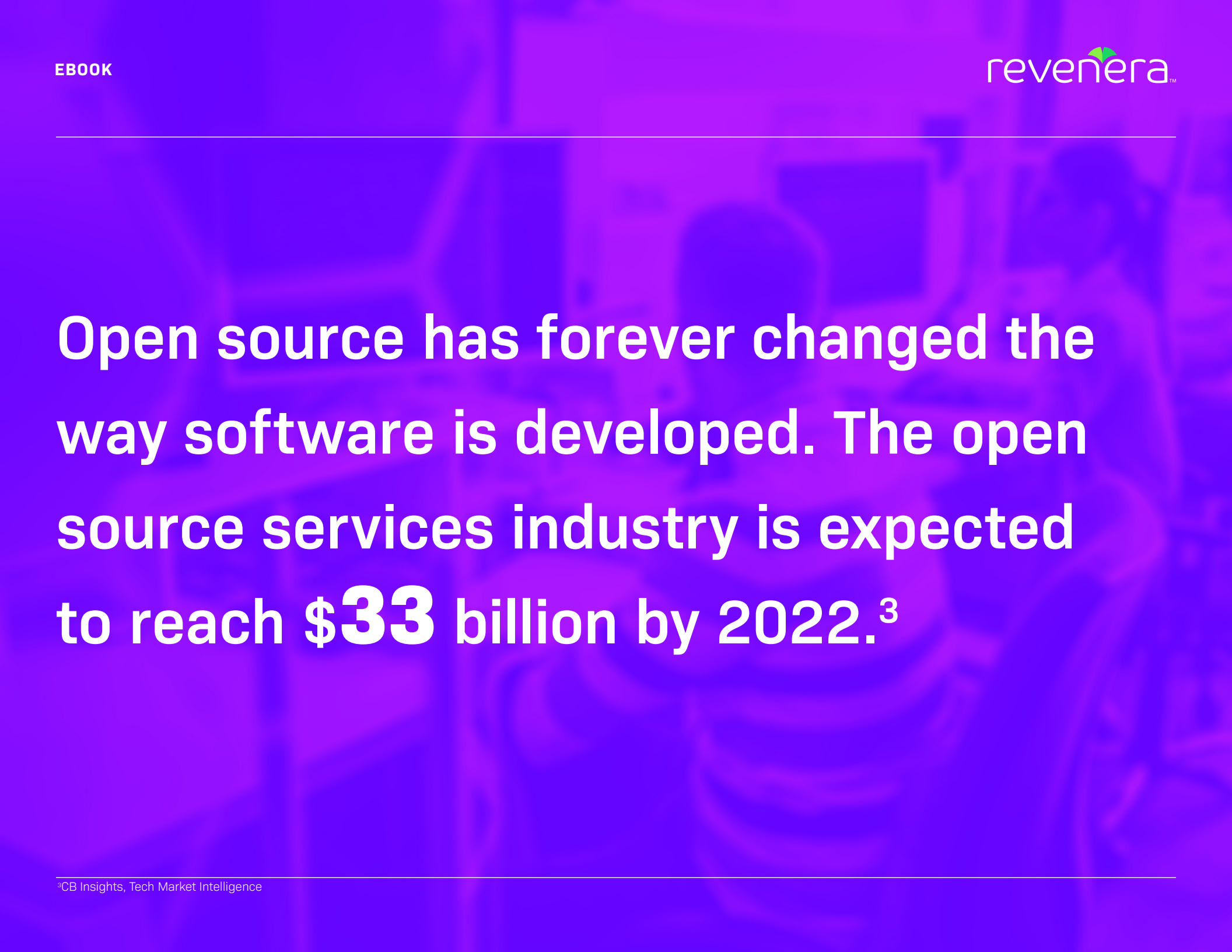
Utilize a Software Composition Analysis solution that allows not just a "shift left" approach to open source scanning, but an "expand left" methodology which highlights the need for scanning early in the DevOps lifecycle, but also throughout.

Understanding History

The Evolution of Open Source Use

To understand where you need to go with your development processes, you first must understand how the use of open source software has fundamentally changed.

	1 The Build	2 Open Source Entry Points	3 Dependencies
THEN	A few decades ago, organizations had complete control of their codebases. Even for work contracted out, companies still had rights to the source code coming back.	Software was a closely held secret. Computers were delivered with operating systems and application software installed in an editable source code form. There were limited avenues of entry where "other" code came from.	The beginnings of the "free software movement" and the freedom to share source code and created licenses, specifically the GNU General Public Licenses.
NOW	More code is leveraged from the open source community with many more hands on it for modifications, fixes, and improvements—diminishing the chain of custody and control.	Software builds have opened to the outside world. There's a vast partner and supplier network to leverage open source projects, to author code, and to interject supplier and commercial code.	Open source drives modern application development, and with increased use engineering teams are unaware of the actual number of open source libraries they are using due to dependencies.



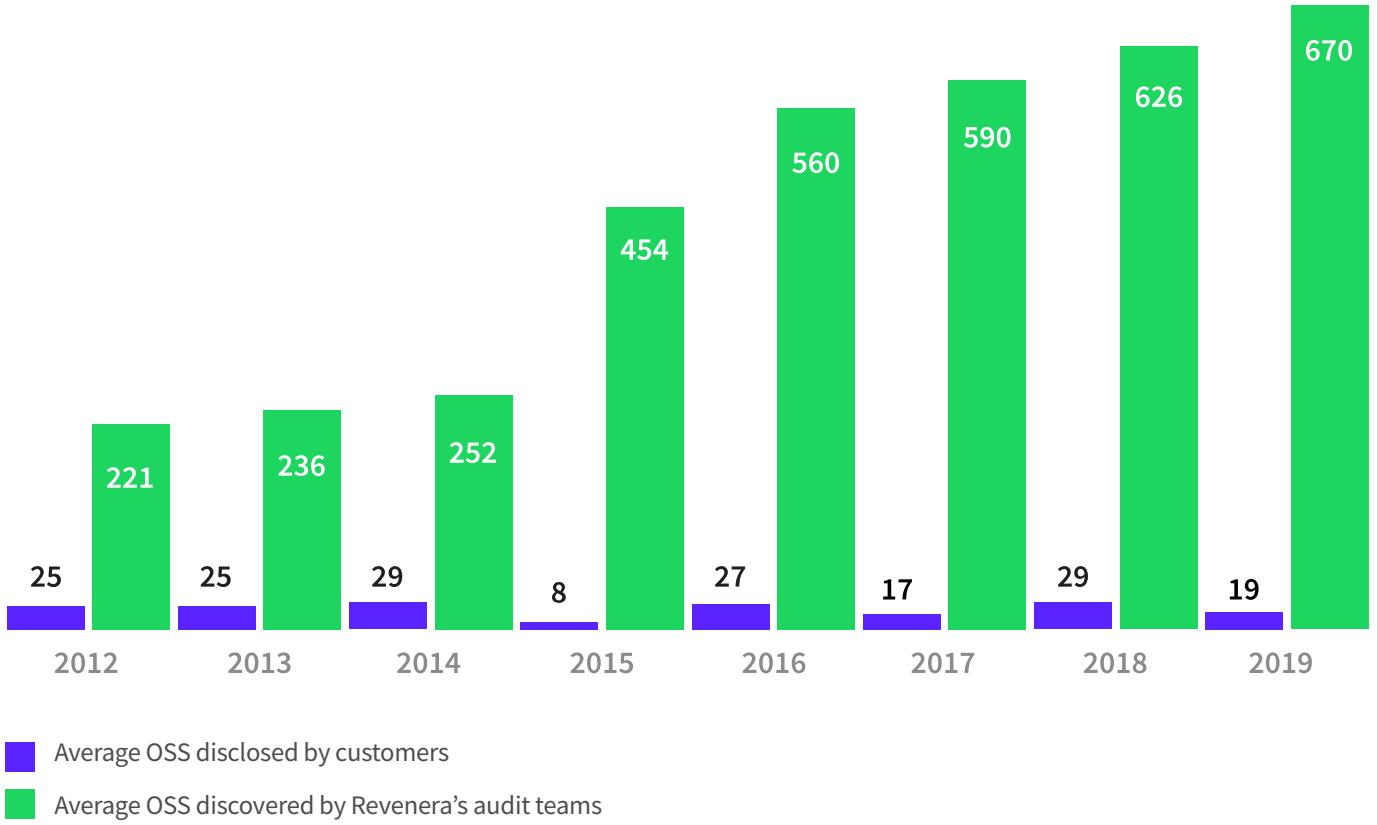
Open source has forever changed the way software is developed. The open source services industry is expected to reach $**33** billion by 2022.[3]

[3]CB Insights, Tech Market Intelligence

The Software Bill of Materials is Growing Due to Dependencies

The Evolution of Open Source Use

To understand where you need to go with your development processes, you first must understand how the use of open source software has fundamentally changed.



Source: Revenera Professional Services Audit Data 2012–2019 (sampled)

Choosing Software Composition Analysis

OS Advancement Leads to Enhanced Management

Looking at even just a brief history of open source provides a glimpse into the change open source has undergone over the decades. There have been pivotal moments and the advancements just keep on coming.

With the changing landscape is it any wonder that processes supporting a strategy of managed open source should not stay static? The need for tools like SCA is such an example. With growing demand, increased use, and competitive pressures, open source use is up. Off-loading the time-consuming and complex tasks of managing the open source supply chain to a continuous, automated solution gives developers the flexibility to focus on projects that drive value.

But are internal demands the only driving force? Not by a longshot:

- Regulated industries such as healthcare, government and finance impose requirements on software suppliers to disclose what's in their code.
- Competitors are automatically supplying an SBOM, creating a best practice service, and giving them a competitive advantage. Why not be proactive in your disclosure processes?
- Contractual requirements with customers demanding not just open source disclosure, but on governance, security, and remediation practices.

Crucial Open Source Advancements

1. Creation of GNU
2. Engineering process innovation
3. Legal requirements such as "copyleft"
4. Software development tooling
5. Open source is the core of web browsers and operating systems
6. Corporate investment in open source

10% of Revenera customers are requiring contractual language supporting open source disclosure

Meeting Industry, Customer and Regulatory Requirements

As Conditions Change, Adapting is Key to Longevity

Maintain an **up to date SBOM** of all open source software components used in your applications

Follow a process to **identify known security vulnerabilities** within open source software components

Monitor existing open source software components for **new security vulnerabilities**

Maintain a **policy and patching process** to remediate impacted open source software components

Organizations Implementing Open Source Requirements

PCI Security Standards Council

GDPR European Commission General Data Protection Regulation

MITRE

NTIA National Telecommunications & Information Administration, U.S. Department of Commerce

NIST

FDA U.S. FOOD & DRUG ADMINISTRATION

National Cyber Security Centre a part of GCHQ

OWASP Open Web Application Security Project

Catalysts for Change Aren't Always Obvious

Other Events Impacting the DevOps Process

Developer Education

The top 40 U.S. and top five international computer science (CS) programs do not include open source licensing and secure coding in their curriculum.[4] Unless a developer happens to sign up for a modern online CS program, chances are they have received no formal education. Responsibility then falls to organizations to fill the gap through onboarding or recertification efforts while keeping engineers focused on their main purpose—developing software to meet the needs of customers.

Changing Software Supply Chain Rules

Once upon a time engineering teams may have made the decision to not take advantage of open source software given, as part of the larger software supply chain, how much open source was coming into the organization was unknown, not to mention there was most likely not a strategy in place for monitoring open source use. Today, however, software suppliers are responsible for all components found in their applications. To remain competitive, companies have evolved to include contractual policies around open source, inbound software reviews, self-reporting, and tooling to certify code before it's accepted for use in product development. All of this with the end goal to increase trust and transparency along the supply chain. The more controls and gates put in place early in the process the easier it is to manage risk in the final product.

Move to the Cloud

Industry, market and environmental pressures force many companies to rethink how they deliver applications to customers. Desktop applications and services are moving to the cloud at an accelerated rate because:

- Users are more distributed than ever before
- Product releases happen more frequently with less time between releases

With code not behind a firewall, not sitting on-premise, and with no controls or lockdown server in place, the more vulnerabilities exist. Identifying and fixing issues is required at a much faster rate in order to mitigate increased risk. Relying solely on human processes is both inefficient and untimely. The answer is automation and fully integrating end-to-end scanning into the engineering toolchain to manage the most egregious issues early and fast.

[4]Show Don't Tell Your Developers How to Write Secure Code, Forrester Research, Inc., April 19, 2019

Evolve Your Engineering Toolchain

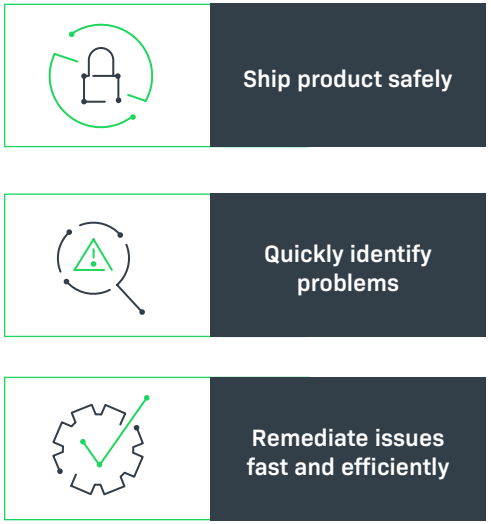
Integrate Continuous SCA for a Proactive Approach

With inevitable change in industry, regulatory, and customer requirements, implementing a continuous SCA solution allows for a proactive strategy to open source code management. Move away from auditing on an infrequent basis to getting ahead of potential issues caused by shifting customer and industry needs.

Put a continuous, automated SCA process in place that allows for finding and fixing security and license compliance issues as early as possible. Integrate SCA into the engineering toolchain with a focus on:

- Artifact repositories
- IDEs
- Source Code Management (SCM)
- CI/CD builds
- Containers
- Issue trackers (Application Lifecycle Management)

Benefits of Continuous SCA



Look for early opportunities to scan and create a culture of continuous monitoring. Today your SBOM is up to date, but for how long? New packages can be pulled into your development environment at any time as a result of application updates and changes. This creates the opportunity to introduce new or unknown vulnerabilities. Identifying risk early and often allows for faster remediation…before products reach your customers' doorstep.

Active monitoring should also include alerts for developers and engineers. Alerting the right people at the right time of non-compliant issues and vulnerabilities is essential to quickly identifying problems, fast remediation, and safe product shipment.

Intersecting SCA and the Engineering Lifecycle

Identify Open Source Scanning Events

What are the various scan events and when does it make sense to initiate an open source scan?	
ARTIFACT REPOSITORY	▪ Scan on the artifact repository server ▪ Identify compliance issues during ingestion of new OS components ▪ Review and define allowed usage once ▪ Quarantine compliance issues and identify exposure across enterprise
IDE	▪ Scan on developer's laptop ▪ Detect compliance issues before code check-in ▪ Smoke test
BUILD	▪ Automatically scan on build server as part of build process ▪ Continuous, nightly, or release team builds ▪ Fail build based on compliance issues
SCAN SERVER	▪ Periodic deep scans on SCA scanning solution ▪ Use SCM plugins to sync codebase to server ▪ Support optional manual analysis
CONTAINERS	▪ Periodic scans of images prior to deployment ▪ Identify environmental dependencies

Be Dynamic in an Active Environment

Leverage SCA for Open Source Management

Gartner predicts that by 2022, 50% of organizations will execute at least one DevOps pipeline relying entirely on open source tools.[5] Along with increased open source growth, development teams should expect to see:

- More requirements around transparency from highly regulated industries
- Market and environmental shifts that necessitate engineers evolve both technology and the processes supporting use
- Customers demanding open source disclosure, governance, security and remediation contractual obligations
- A growing, more complex software supply chain
- Faster transitions to digital transformation
- Increased security, IP, and license compliance risk
- Competitive pressures resulting from companies racing to take full advantage of open source benefits while implementing DevOps toolchain best practices

Your code isn't static. Open source licenses vary and change in complexity. The software supply chain interjects new open source into the pipeline. Packages become outdated. New vulnerabilities are unearthed, and hackers exploit every open door and environmental crisis.

Answers to these challenges require a new approach. It's up to each organization to introduce the right processes and the right tools to manage disruption—both positive and negative.

Continuous Software Composition Analysis supports a managed open source strategy throughout the software supply chain to discover evidence of open source and to find vulnerabilities and licensing issues both early and often. With automated scans, issues are identified and fixed quickly with less effort.

[5]Four Steps to Adopt Open Source Software as Part of the DevOps Toolchain, Gartner, Inc., February 6, 2019

Build Better Products

Invest in Software Intelligence

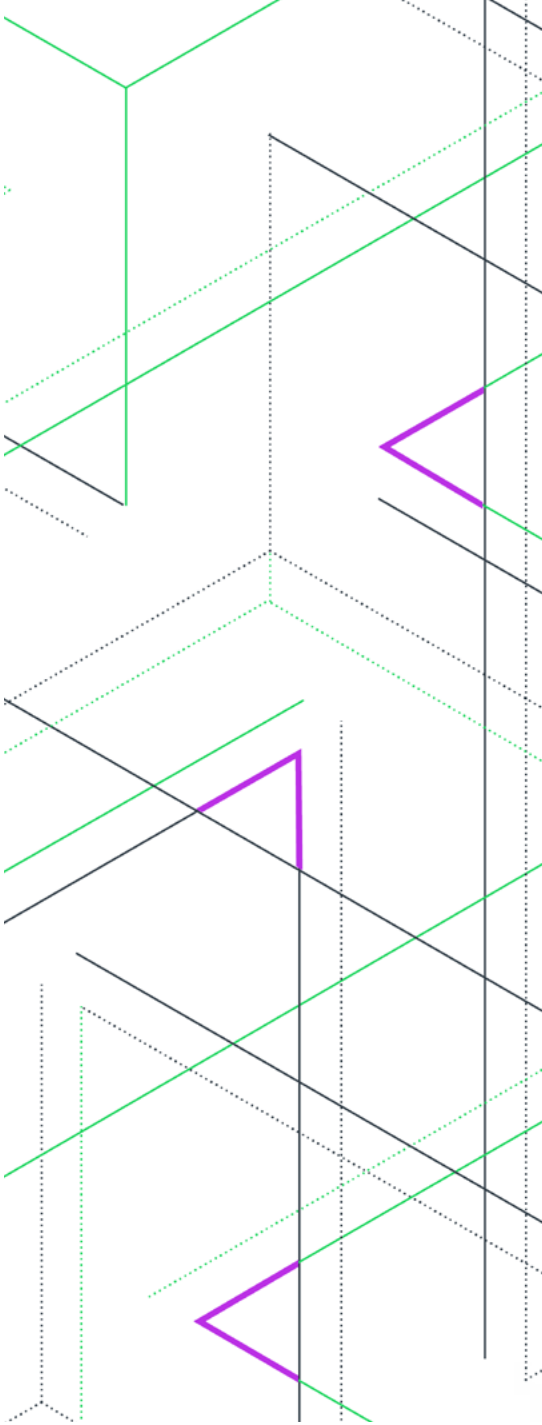
Software Composition Analysis from Revenera unlocks the full potential of open source by enabling your software development teams to innovate faster and more effectively. Revenera provides code scanning intelligence to help you eliminate unknown license compliance and security risks while accelerating your time to market with safer products.

More intelligence and actionable insights. Revenera Software Composition Analysis solutions have the answers.

NEXT STEPS

There's no better time than now. Unlock the full potential of your software engineering processes and take advantage of the many benefits coming from open source.

LEARN MORE >



Revenera provides the enabling technology to take products to market fast, unlock the value of your IP and accelerate revenue growth—from the edge to the cloud. **www.revenera.com**

1.800.809.5659 | +44.870.871.1111 | revenera.com

revenera™