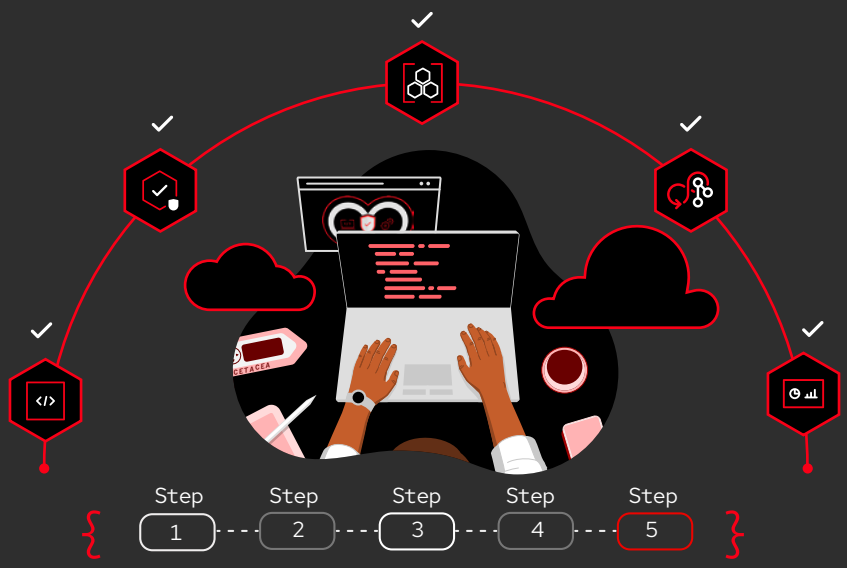
Published by Red Hat Developer

A developer's guide to setting supply chain security in DevSecOps

5 steps to building security early into your software development



Collin Chau, Dash Copeland, and Markus Eisele

Security? Seriously? A developer's view

Navigating software supply chain attacks

Why security and secure supply chains matter even more today

The importance of a DevSecOps strategy

DevSecOps best practices for developers

Set supply chain security in DevSecOps: The promised land

Enabling a successful DevSecOps practice

How Red Hat enables security-focused cloud-native development

For more information

About the authors

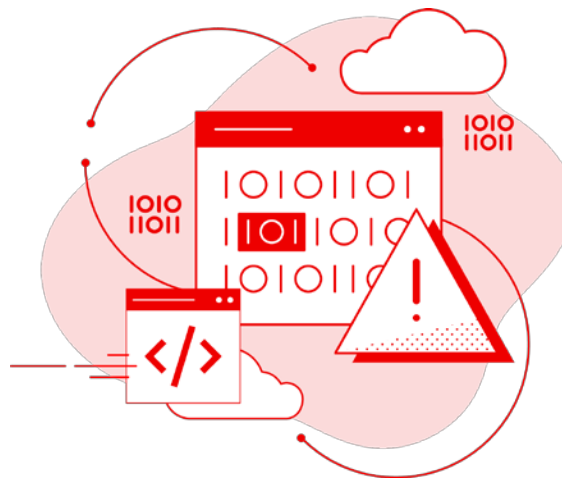
A developer's guide to setting supply chain security in DevSecOps

5 steps to building security early into your software development

DevSecOps is a relatively new term in the world of software development that is quickly gaining popularity as more and more organizations recognize the importance of integrating security practices into their software development processes. DevSecOps combines the principles of DevOps—which emphasizes collaboration and automation between development and operations teams—with security practices to create a culture of security within the software development life cycle.

Developers play a crucial role in implementing DevSecOps practices because they are responsible for writing the code that makes up the software. However, many developers don't have strong security backgrounds and might not be aware of the best practices for building secure software.

This guide provides a developer's introduction to DevSecOps, including the key principles, tools, and techniques you need to know to build secure software as part of a DevSecOps team.



Security? Seriously? A developer's view

Navigating software supply chain attacks

Why security and secure supply chains matter even more today

The importance of a DevSecOps strategy

DevSecOps best practices for developers

Set supply chain security in DevSecOps: The promised land

Enabling a successful DevSecOps practice

How Red Hat enables security-focused cloud-native development

For more information

About the authors

Security? Seriously? A developer's view

Software developers value software security and recognize the importance of having policies that ensure the security of the software they develop. A 2023 Red Hat internal research survey found that software developers' #1 underserved outcome was to "minimize the likelihood of deploying applications that contain security vulnerabilities."

However, ensuring that software is secure and protected against threats adds additional layers of complexity to the development process, which can be a burden for developers already overwhelmed by cognitive load. As a developer, you must constantly be aware of potential vulnerabilities and implement secure coding practices such as input validation, digital signing, data encryption, and access control. This effort is often time-consuming and can slow down the development process, leading to frustration when there's pressure to deliver software quickly.

Many developers do not have strong security backgrounds and might not be familiar with the latest security threats and best practices. This can make it challenging to implement effective security measures and can increase the risk of security breaches.

Furthermore, software security is an ongoing process in the software development life cycle that requires regular updates and maintenance to ensure the software remains secure over time. Such maintenance adds even more work to a developer's busy schedule, which can contribute to burnout and job dissatisfaction.

Software security can be complex and time-consuming, and it can distract from the task of creating functional software that drives business value and delights users. Still it is essential to protect against security threats and ensure that software is secure and reliable for users. And with modern practices, the topic of security does not stop at the source code or non-functional requirements. It goes all the way through the complete software supply chain.

Security? Seriously? A developer's view

Navigating software supply chain attacks

Why security and secure supply chains matter even more today

The importance of a DevSecOps strategy

DevSecOps best practices for developers

Set supply chain security in DevSecOps: The promised land

Enabling a successful DevSecOps practice

How Red Hat enables security-focused cloud-native development

For more information

About the authors

Navigating software supply chain attacks

Software supply chain (SSC) refers to the process of developing, consuming, and distributing software. This can involve numerous parties, software components, and dependencies that make it vulnerable to security risks.

Consuming third-party libraries or distributing your own library with vulnerabilities can have far-reaching consequences, including financial losses, damage to reputation, and legal liabilities that erode customer trust. Simply implementing secure development practices and adhering to established policies to protect against these risks is not enough when we do not directly control, audit, and safeguard the source code and transitive dependencies a project consumes at the onset.

"There could be a huge loss in terms of people's information getting compromised, or just people's trust, from a brand perspective."

Supply chain attacks can go undetected and exponentially impact software producers and consumers. That lack of visibility can lead to significant propagation of compromised code and a wave of vulnerabilities or exploits. Therefore, in addition to securing independent components of their applications, developers should lock and guard all digital entry points into their software factories. Focusing only on one dimension of the software supply chain is not scalable, and also inadequate.

Security? Seriously? A developer's view

Navigating software supply chain attacks

Why security and secure supply chains matter even more today

The importance of a DevSecOps strategy

DevSecOps best practices for developers

Set supply chain security in DevSecOps: The promised land

Enabling a successful DevSecOps practice

How Red Hat enables security-focused cloud-native development

For more information

About the authors

"it should be comprehensive... beyond just the application, and the packages you're using, also checking if you have any ports open that you're not supposed to, or you have something public where it should be private."

Designed to help secure and protect cloud-native applications across development and production, Red Hat's centralized platform of consolidated security and compliance capabilities helps DevOps and security teams work together on DevSecOps requirements.

Drawing on 30 years of building trusted products and packages, safely delivering this into enterprise software that businesses rely on, Red Hat helps developers understand and navigate supply chain security. By making the same software supply chain that we have used to build and deliver trusted content available to developers and security teams, we recognize that it is a shared responsibility across all parties involved in the software development process.

Security? Seriously? A developer's view

Navigating software supply chain attacks

Why security and secure supply chains matter even more today

The importance of a DevSecOps strategy

DevSecOps best practices for developers

Set supply chain security in DevSecOps: The promised land

Enabling a successful DevSecOps practice

How Red Hat enables security-focused cloud-native development

For more information

About the authors

Why security and secure supply chains matter even more today

Digital transformation continues at a relentless pace, putting even greater responsibility on business executives to meet the new demands of a fully digitized customer experience. With every organization now a software-driven company, technology leaders are expected to enable new business outcomes like flexibility and scale from moving to the cloud. However, many struggle to maintain consistent security and performance in their complex, hybrid IT environment, stalling transformation efforts in their software factory.

Organizations are up against active adversaries that are well-funded and eager to exploit an ever-growing threat surface. With payoffs from cyber attacks surging with each passing security incident, this has led to a persistent landscape of distributed denial-of-service attacks, ransomware, zero-day vulnerabilities, and more. Keeping up with security challenges is harder than ever before, as the frequency and sophistication of cyber attacks from bad actors continue to increase dramatically.

Siloed communications, incompatible interfaces, and lack of standardization across too many disparate products have significantly reduced security efficacy across the application development and deployment system. Organizations need integrated security tooling and processes that support DevSecOps practices, driven in part by the following:

- **Elevated complexity for secure development.** While developers understand the importance of writing secure and compliant code, the complexity of addressing these concerns in a modern DevSecOps-driven world is growing exponentially. According to internal Red Hat security outcome research, developers are increasingly unsatisfied with their ability to meet these needs.

  "It's kind of difficult to go to one place and just figure out like, Hey, is everything really up to date and no vulnerabilities? I mean, Google Cloud itself was was pretty large. And so using will these different products and things like that can be hard

Security? Seriously? A developer's view

Navigating software supply chain attacks

Why security and secure supply chains matter even more today

The importance of a DevSecOps strategy

DevSecOps best practices for developers

Set supply chain security in DevSecOps: The promised land

Enabling a successful DevSecOps practice

How Red Hat enables security-focused cloud-native development

For more information

About the authors

to manage. I guess, making sure that every product that we use, there are no vulnerabilities there."

- **The growing use of open source components.** Many organizations use open source components to develop software, and the security of these components can be compromised by attackers seeking to infiltrate the supply chain. Being enterprise, open source-ready is critical. 2 out of 3 organizations surveyed by Red Hat reported currently using open source software (OSS) to augment their internal development when building new applications, with the remaining planning to do so in the future.[1]
- **New attack vectors discovered each day.** Organizations increasingly rely on third-party tools and service dependencies. This inherently introduces security risks and compromises application releases early in the development life cycle if vendors have not properly secured their software upfront. Of the more than 1.2 billion dependencies downloaded each month, a recent study noted that 6 out of 7 project vulnerabilities come from transitive dependencies.[2]

  "All software is built on lots of different packages getting assembled and built. And that's the other place where keeping track of the hundreds of different packages that had downloaded by, and as part of our installation is just you hope there's good protection out there. But it's at the point where you can't just manually go through every single provider and then every single dependency of the provider."

- **Stricter regulatory requirements.**[3,4] Many industries subject to stringent data governance and compliance now require organizations to implement security measures for more robust cyber resiliency in their software supply chain—including tracking all code dependencies in the software. This is coming off the back of an

1 Better Together: DevOps and Open Source Go Hand in Hand, IDC Perspective, 2022.
2 8th Annual State of the Software Supply Chain, Sonatype.
3 White House Executive Order on Improving the Nation's Cybersecurity (EO 14028).
4 European Commission's proposed Cyber Resilience Act (CRA).

Security? Seriously? A developer's view

Navigating software supply chain attacks

Why security and secure supply chains matter even more today

The importance of a DevSecOps strategy

DevSecOps best practices for developers

Set supply chain security in DevSecOps: The promised land

Enabling a successful DevSecOps practice

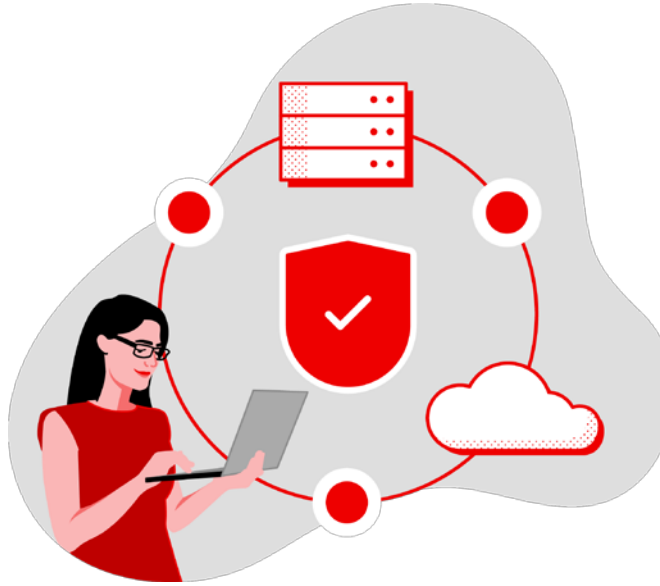
How Red Hat enables security-focused cloud-native development

For more information

About the authors

astonishing 742% average annual increase in software supply chain attacks over the past 3 years.[5] Gartner predicts that by 2025, 45% of organizations worldwide will have experienced attacks on their software supply chains, a three-fold increase from 2021.[6]

By securing the software supply chain, organizations can reduce the risk of cyber attacks, data breaches, and other security incidents that result in financial losses, reputational damage, and harm to end users. Organizations that implement best practices for software supply chain security can improve the overall quality and security of their software and help protect their systems and data from attacks.



5 Sonatype, 8th Annual State of the Software Supply Chain.
6 Gartner's 7 Top Trends in Cybersecurity for 2022.

Security? Seriously? A developer's view

Navigating software supply chain attacks

Why security and secure supply chains matter even more today

The importance of a DevSecOps strategy

DevSecOps best practices for developers

Set supply chain security in DevSecOps: The promised land

Enabling a successful DevSecOps practice

How Red Hat enables security-focused cloud-native development

For more information

About the authors

The importance of a DevSecOps strategy

Threats to software supply chain security have forced a sea change in DevOps practices in favor of a DevSecOps strategy where security is a fundamental and ongoing aspect of the software development life cycle.[7] Software engineering leaders mitigate software supply chain risks by focusing on the security of software components and dependencies early in the software development life cycle. They enforce integrated security gates at every phase for consistent, repeatable, and automated operations in the software factory.

The overall market is moving toward application platforms that can provide for the fast, secure, continuous deployment of great software experiences that companies compete by. But the reality is that enterprises often struggle with running these parallel tasks. Their challenges include the following:

- Maintaining and improving legacy applications and infrastructure is complicated and places strain on already limited IT resources.
- Building and running brand new applications using modern frameworks and cloud-native application architectures increases cognitive load for dev teams.
- Security is often an afterthought that's handled by security and IT operations teams at the end of the application development life cycle, with little to no collaboration with app development and other teams.
- Disparate application security and DevOps tools, practices, and disjointed processes result in tool sprawl; this impedes collaboration, visibility, and productivity and increases the chance of human error.

As a result, organizations often fail to catch security issues early, when they are easier and less expensive to fix. This increases the risk of security breaches and hinders the speed and efficiency of application development and delivery.

7 How Software Engineering Leaders Can Mitigate Software Supply Chain Security Risks, Gartner 2021.

Security? Seriously? A developer's view

Navigating software supply chain attacks

Why security and secure supply chains matter even more today

The importance of a DevSecOps strategy

DevSecOps best practices for developers

Set supply chain security in DevSecOps: The promised land

Enabling a successful DevSecOps practice

How Red Hat enables security-focused cloud-native development

For more information

About the authors

DevSecOps best practices for developers

Software supply chain security and DevSecOps are both important approaches to software development that prioritize security, but they focus on different aspects of the software development process:

- **Secure software supply chain** focuses on the security of the software as it moves through the supply chain.
- **DevSecOps** emphasizes building secure software through secure coding practices and ongoing monitoring and improvement of security.

So, one can not fully strive without the other. For developers, it is essential to embrace DevSecOps practices for the development of the software itself and to base those practices on a secure software supply chain. Let's briefly look at the most important best practices for developers.

Implement security early and often

Security should be integrated into the software development process from the very beginning and considered at every stage of the development life cycle. This includes security testing, vulnerability scanning, and code analysis, as well as secure coding practices.

"There's not a lot of like, automation and checks there for the security parts of it. What I'm interested in is the automation so that we have that manual checks, but anything that can help us find that earlier rather than later is better."

Automate security wherever possible

Automation can streamline the security testing and review process, making it more efficient and effective. Examples of this include automated vulnerability scanning, automated testing, and continuous integration and deployment pipelines.

Security? Seriously? A developer's view

Navigating software supply chain attacks

Why security and secure supply chains matter even more today

The importance of a DevSecOps strategy

DevSecOps best practices for developers

Set supply chain security in DevSecOps: The promised land

Enabling a successful DevSecOps practice

How Red Hat enables security-focused cloud-native development

For more information

About the authors

Emphasize collaboration between development, security, and operations teams

DevSecOps is based on the principles of collaboration and teamwork between different departments within an organization. By working together, teams can identify and address security issues more quickly and effectively.

Use secure coding practices

Secure coding practices such as input validation, data encryption, and access control can help prevent software vulnerabilities. Developers should be trained in secure coding practices and follow established coding standards to ensure the security of the code they write.

Conduct regular security assessments

Regular security assessments help identify potential vulnerabilities and security issues in software. This can include penetration testing, code reviews, and vulnerability scanning, among other techniques.

Continuously monitor and improve security

Security is an ongoing process, and software must be monitored and updated regularly to ensure that it remains secure. This includes monitoring for new security threats, updating software as needed, and addressing security issues as they arise. By continuously monitoring and improving security, organizations can reduce the risk of security breaches and protect their software and data.

Security? Seriously? A developer's view

Navigating software supply chain attacks

Why security and secure supply chains matter even more today

The importance of a DevSecOps strategy

DevSecOps best practices for developers

Set supply chain security in DevSecOps: The promised land

Enabling a successful DevSecOps practice

How Red Hat enables security-focused cloud-native development

For more information

About the authors

Set supply chain security in DevSecOps: The promised land

Software supply chain security is a complex and multifaceted field that requires a comprehensive DevSecOps approach to be effective. At the heart of a successful DevSecOps practice stems from how businesses secure open source software in their software supply chain.

The lack of continuous monitoring and testing of open source code and their third-party dependencies for vulnerabilities early introduces security risks. Development and security teams are unable to track and perform much-needed security maintenance activities to prioritize and prevent service incidents.

Red Hat improves supply chain resiliency that keeps pace with innovation cycles for businesses to release applications faster while meeting security requirements. Available as a cloud service, Red Hat Trusted Software Supply Chain provides a DevSecOps framework from which to consistently code, build, and monitor a trusted software supply chain. In just a few clicks, we secure the use of open source software and third-party dependencies in applications by integrating security guardrails at every phase of the software development life cycle for faster time to value.

Developers care about user trust,[8] and we want to make them successful without all the overhead. Red Hat's solutions help developers shift left with security throughout the entire software development life cycle (Figure 1).

Figure 1: Secure software components and dependencies early in your software development life cycle, from code to production.

8 Developer Security Outcome Discovery Report (Red Hat, Jan 2023).

Security? Seriously? A developer's view

Navigating software supply chain attacks

Why security and secure supply chains matter even more today

The importance of a DevSecOps strategy

DevSecOps best practices for developers

Set supply chain security in DevSecOps: The promised land

Enabling a successful DevSecOps practice

How Red Hat enables security-focused cloud-native development

For more information

About the authors

Enabling a successful DevSecOps practice

Successfully implementing DevSecOps begins well before the application pipeline. As a first step, organizations will want to ensure their underlying infrastructure and application services are running on an enterprise open source foundation pre-hardened with built-in security tools and features. Red Hat's platform is managed by a dedicated product security team that monitors, identifies, and addresses vulnerabilities quickly. Continuous security updates are delivered to security-enhanced Linux containers and distributed through a secured channel.

Developers need security scanning and guidance across all aspects of cloud-based applications. Beyond just the software packages, they need security coverage on tooling, application configurations, and the entire solution architecture, including infrastructure.

Developers also need flexibility to move workloads to any footprint that works best with consumption options to match the organization's needs for an open hybrid cloud. Building on trusted, industry-proven container orchestration platforms adds the advantages of standards and consistency to continue their investments in, for example, a Kubernetes-native Java framework like Quarkus.

You can create applications safely at scale for a modern, cloud-native world through a combination of tools, libraries, and extensions for a frictionless development experience. Teams can contribute code in minutes on Kubernetes with centralized developer workspaces on local machines, based on popular integrated development environment (IDE) tools. Apply the same single sign-on (SSO) tool used across the software factory to secure development workspace and source code access.

1 **Get trusted images and libraries out-of-the-box**
Stay on top of the latest vulnerabilities and security risks by making use of trusted content in the form of libraries from popular application frameworks available including Java, Node.js, Python, Go, and packages from Red Hat Enterprise Linux (RHEL). Standardize content to reduce your team's cognitive load with best practices

Security? Seriously? A developer's view

Navigating software supply chain attacks

Why security and secure supply chains matter even more today

The importance of a DevSecOps strategy

DevSecOps best practices for developers

Set supply chain security in DevSecOps: The promised land

Enabling a successful DevSecOps practice

How Red Hat enables security-focused cloud-native development

For more information

About the authors

that promote shared alignment and easier collaboration across interconnected software factories. Minimize risk profiles by using curated OSS content for the software development life cycle (SDLC) that's already scanned and hardened based on Red Hat's best practices.

When it comes to security, developers want to do the right thing. They care about professional excellence, but even more than that, they care about gaining and keeping their users' trust. Having access to a robust collection of third-party software libraries, frameworks, APIs, and tools that are actively maintained and meet the organization's security standards frees up developers to focus on building new features without the work of evaluating and securing every dependency and tool.[9]

2 Maintain a highly available container registry from which to securely access and incorporate attested, curated packages

Restrict access to the container registry and the images stored within using granular role-based access controls (RBAC) to reduce risk of unauthorized entry. Securely store and manage images that are used to deploy applications and services, ensuring that only trusted images are used in production. Run rootless container images to install packages and run services safely within the container without impacting the host.

Increase transparency and visibility across software factories to build trust between security teams and DevOps teams. Allow image signing for verification and authentication, which helps prevent malicious code from being added to the registry. Verify the authenticity of the software build of materials and prevent tampering to ensure code integrity. Support the use of digital signatures and certificates that attests to the origin of software components as coming from a trusted source. Show software provenance of both build and pipeline using Supply-chain Levels for Software Artifacts (SLSA) standards for a chain of trust that reduces the risk of counterfeits and malicious modification.

9 Developer Security Outcome Discovery Report (Red Hat, Jan 2023).

Security? Seriously? A developer's view

Navigating software supply chain attacks

Why security and secure supply chains matter even more today

The importance of a DevSecOps strategy

DevSecOps best practices for developers

Set supply chain security in DevSecOps: The promised land

Enabling a successful DevSecOps practice

How Red Hat enables security-focused cloud-native development

For more information

About the authors

It is important to realize that developers will occasionally need access to systems to address security issues. In larger organizations, it's important to have trust-gating mechanisms in place such as RBAC and GitOps in order to enforce privileged access. What's more, they need an auditable record of all privileged sessions.[10]

3 Protect source code and dependencies in code management with security best practices

Analyze and detect potential vulnerabilities, malware, or other malicious code before they are consumed across software factories. Make use of automated code analysis to scan for potential security vulnerabilities in images and for other security issues before they're committed to the code repository. You need to carefully manage dependencies, and any libraries or components used in the build process should be regularly audited for vulnerabilities. Component analysis helps organizations identify and assess the risk of third-party components in their software supply chain.

Many developers believe they shouldn't deploy anything that has a severe vulnerability, so it's important to catch severe vulnerabilities early to determine if they will be able to deploy to production. Developers want to be more proactive and find vulnerabilities earlier, but you also need to focus on writing code and delivering fast. Being able to automate security checks and fixes frees you from having to manually track down vulnerabilities and makes it easy to follow security best practices.[11]

Tamper-proof code with ubiquitous cryptographic signing and automatically account for every submission through a public, immutable open source log of all activities. Ensure that code and artifacts are securely stored in a version control system that is protected by strong identity and access management (IAM) policies. Keep source code safely in a repository that's regularly audited for vulnerabilities and secured from external threats by implementing encryption and secure backup systems.

10 Developer Security Outcome Discovery Report (Red Hat, Jan 23).
11 Developer Security Outcome Discovery Report (Red Hat, Jan 23).

Security? Seriously? A developer's view

Navigating software supply chain attacks

Why security and secure supply chains matter even more today

The importance of a DevSecOps strategy

DevSecOps best practices for developers

Set supply chain security in DevSecOps: The promised land

Enabling a successful DevSecOps practice

How Red Hat enables security-focused cloud-native development

For more information

About the authors

4 Strengthen the CI/CD pipeline with an automated chain of trust and approval gates

Control the flow of software dependencies and ensure that only trusted packages are used in builds and deployments to prevent poisoned pipeline execution in the software factory. Manage and secure the use of various software components that make up the build by first auto-generating software bill of materials (SBOMs) with metadata on how each artifact was built. Authenticate provenance to industry standards through version control, auditing, and traceability of all software components used in the development process.

Automate CI/CD pipelines with regular security checks integrated throughout the build process to ensure all inputs and outputs are secure as teams compile code, build images, and run tests. Institute strong protections against tampering through cross-build contamination. Immediately detect and alert on any changes or unauthorized modifications to the source code and OSS dependencies that are impacting build artifacts stored in the repository. Determine which versions of what components were used in any given application and understand the impact of that change to mitigate risks in the SDLC.

Allow organizations to define and enforce enterprise contracts that dictate the acceptable use and behavior of software components within their software supply chain. Continuously deploy to a declarative state with release policies as code that ensure misconfigurations are automatically detected, with deployments halted and rollbacked. Implement security scanning continuously, including binary analysis, to inspect and identify vulnerabilities in compiled code and images for common vulnerabilities and exposures (CVEs) and viruses. Trigger automatic remediation actions to prevent these security gaps from being exploited.

Developers need a comprehensive and accurate inventory that automatically tracks and maintains dependencies to ensure applications are secure. The scale of dependency management makes it infeasible to do manually. An application is composed of hundreds of different packages, and each can introduce security

Security? Seriously? A developer's view

Navigating software supply chain attacks

Why security and secure supply chains matter even more today

The importance of a DevSecOps strategy

DevSecOps best practices for developers

Set supply chain security in DevSecOps: The promised land

Enabling a successful DevSecOps practice

How Red Hat enables security-focused cloud-native development

For more information

About the authors

vulnerabilities through their dependencies. These dependencies can often block developers from installing necessary package updates. Therefore, developers need help tracking dependencies and keeping them updated.[12]

5 Monitor applications at runtime with contextual insights into vulnerabilities and threats to deployed workloads

Ensure that deployment environments are secure at runtime by implementing proper access controls, threat prevention and anomaly detection, network segmentation, and runtime vulnerability detection. Provide complete end-to-end visibility into all components and their respective sources to continuously monitor and proactively identify changes in the risk profile caused by malicious components. Implement monitoring and logging systems that instantly detect, alert, and direct on potential security incidents. This includes detailed reports and analytics to help organizations understand their risk posture and make informed decisions.

Developers need to determine the impact a security vulnerability has based on how the affected package is used. Severity alone is not enough to make a decision. They need contextual information about how the package is used and the application's deployment environment in order to make a determination of its true importance. Developers should have a single place to easily track and manage vulnerabilities throughout their application's entire architecture.[13]

Support compliance and audits that satisfy security assessment and federal government cybersecurity orders for the software factory. Take steps for better API security when connecting services and transferring data to the right users across availability zones. All the while, constantly protect data integrity and confidentiality at rest, in transit, or in runtimes. Use security techniques, such as API management, encrypted communications, and secure authentication to prevent unauthorized access to sensitive data and systems during deployment.

12 Developer Security Outcome Discovery Report (Red Hat, Jan 2023).
13 Developer Security Outcome Discovery Report (Red Hat, Jan 2023).

Security? Seriously? A developer's view

Navigating software supply chain attacks

Why security and secure supply chains matter even more today

The importance of a DevSecOps strategy

DevSecOps best practices for developers

Set supply chain security in DevSecOps: The promised land

Enabling a successful DevSecOps practice

How Red Hat enables security-focused cloud-native development

For more information

About the authors

How Red Hat enables security-focused cloud-native development

Use open source software that supports a software supply chain security strategy either on-premise or across hybrid and multi-cloud environments. Red Hat's DevSecOps approach for a software supply chain security strategy offers the following benefits.

- **Make use of an integrated ecosystem of security tools to identify and assess the potential risks associated with your software supply chain.** Incorporate curated services that are standardized and consumed from an integrated, self-service portal. Red Hat offers security grounded in 30 years of enterprise Linux experience and 7 years of enterprise Kubernetes experience, along with a proactive approach to securing the software supply chain through DevSecOps tools that help you innovate at speed while maintaining operational security requirements.
- **With a secure software supply chain, customers and users can have greater trust in the software they are using.** Reducing the risk of vulnerabilities and threats being introduced into the software builds customer loyalty and brand reputation. You can also release new software features and updates more quickly to keep pace with changing customer preferences.
- **Improve compliance with industry regulations and standards through the implementation of a software supply chain security solution.** Red Hat can help organizations avoid costly fines and penalties for non-compliance, and improve the overall quality of software at the same time. This results in more stable and reliable software, where we catch security issues impacting users before they do.
- **Developers need security scanning and guidance across all aspects of cloud-based applications.** Beyond just the software packages, you need comprehensive security coverage on tooling, application configurations, and across the entire solution architecture, including infrastructure. What's more, you need to have high trust in the accuracy of the data. It should be cross-referenced with multiple security data sources for a holistic security strategy.[14]

14 Developer Security Outcome Discovery Report (Red Hat, Jan 2023).

Security? Seriously? A developer's view

Navigating software supply chain attacks

Why security and secure supply chains matter even more today

The importance of a DevSecOps strategy

DevSecOps best practices for developers

Set supply chain security in DevSecOps: The promised land

Enabling a successful DevSecOps practice

How Red Hat enables security-focused cloud-native development

For more information

About the authors

Together with its partners, Red Hat offers the tools and services to build a comprehensive DevSecOps ecosystem, along with the expertise and ability to deliver a robust portfolio for building, deploying, and running security-focused applications across an open hybrid cloud. This results in improved processes, faster application development without sacrificing security, a culture of collaboration, and reduced risk.

Security? Seriously? A developer's view

Navigating software supply chain attacks

Why security and secure supply chains matter even more today

The importance of a DevSecOps strategy

DevSecOps best practices for developers

Set supply chain security in DevSecOps: The promised land

Enabling a successful DevSecOps practice

How Red Hat enables security-focused cloud-native development

For more information

About the authors

For more information

Explore tutorials, e-books, and other learning resources from Red Hat Developer to support your software supply chain security strategy:

- DevOps
- DevSecOps
- GitOps
- Secure coding
- Software supply chain security
- Developer Sandbox for Red Hat OpenShift

Security? Seriously? A developer's view

Navigating software supply chain attacks

Why security and secure supply chains matter even more today

The importance of a DevSecOps strategy

DevSecOps best practices for developers

Set supply chain security in DevSecOps: The promised land

Enabling a successful DevSecOps practice

How Red Hat enables security-focused cloud-native development

For more information

About the authors

About the authors

Collin Chau enjoys helping agile developers plan, code, test, build, and deploy quality digital experiences. He has scaled DevOps teams for continuous testing and automating application releases in that time—all the while monitoring and troubleshooting IT service health for SREs in production, and enabling ITOps teams to bridge and broker for hybrid cloud services.

Dash Copeland is a user experience designer and researcher who is passionate about building great experiences for technologists. With over 10 years of experience in the enterprise product space, he has designed digital experiences for partners and developers working with Linux, containers, and Kubernetes.

Markus Eisele is a Java Champion, O'Reilly author, founder of German JavaLand conference, reputed speaker at Java conferences around the world, and a very well-known figure in the enterprise Java world.

With more than 16 years of professional experience in the industry, he designed and developed large enterprise grade applications for Fortune 500 companies. As an experienced team lead and architect, he helped implement some of the largest integration projects in automotive, finance and insurance companies.