VERACODE

SECURE THE SDLC IN 6 STEPS

Optimizing DevSecOps Experience through Comprehensive Coverage

INTRODUCTION:

How can you mitigate potential risks as your attack surface expands with more code streams, frequent releases, the rapid adoption of AI generated code, and a wider variety of advanced technologies?

The answer lies in comprehensive security coverage across the Software Development Lifecycle (SDLC). Traditional security models are insufficient, and embedding security assurance into every SDLC stage represents a cultural and procedural transformation. This e-book outlines a six-step framework for securing the SDLC, emphasizing comprehensive coverage across DevSecOps. By adopting these principles, organizations may achieve accurate, more secure software delivery, reduce security debt, and empower development teams.

The Evolving Imperative of DevSecOps

Modern software development is dynamic, with continuous design, writing, building, and testing. Increased code volume, development streams, and environment complexity drive demand for rapid time-to-market, regulatory compliance, and reduced operational risk. This is not just about shifting security left but how comprehensively that shift is integrated throughout the SDLC with speed and accuracy.

However, developers often lack the right resources and tools – and configuration of those resources and tools – to properly secure their code, third-party libraries, microservices, and AI-assisted coding tools. This can inadvertently introduce insecure code, increasing vulnerabilities and security debt.

DevSecOps is the solution: integrating security into the SDLC from the outset and throughout. This proactive approach ensures secure coding, security accuracy, business resilience, and reduces supply chain risk. By embedding security into every DevOps function, DevSecOps becomes an overarching layer, making security an inherent and continuous part of the CI/CD process.

The Opportunity: Securing the SDLC

By adding security into each stage of the SDLC, the most important outputs of the SDLC are secure when deployed and attestable for compliance. Furthermore, integrating security into the SDLC can greatly improve developer experience. With security being a top priority from the beginning of the development process, developers are equipped to focus on building high-quality, functional code. This not only increases efficiency and productivity, but it also allows developers to take pride in their work and feel confident in the security of the end product. Additionally, they gain valuable knowledge and skills that may benefit them in their career growth and make them more valuable to the organization. Let's dive into the six steps that will secure your SDLC.

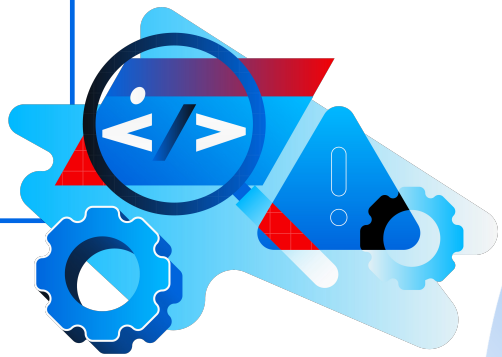
STEP 1:

Discover and Assess Risks

Before adding tooling or recommending integrations, identify all applications and assess their importance to the organization. Focus security efforts on the applications that are critical. Consider asking:

- How many applications do you have?
- What is their relative importance to the organization?
- Where do they reside? Are they in the cloud or on-prem?
- Who owns them?
- Are they still required?
- Do they have open-source dependencies? What are they?
- Is AI being used to generate code? If so, in which apps?

CONCLUSION



Understanding these systems, their owners, and the inventory of applications will allow you to include the appropriate stakeholders and integration points throughout the assessment. It's important to understand how different systems and tools are used for legacy versus cloud-native applications, as well as where outsourced and acquired applications fit into your broader portfolio.

Consider these areas during the discovery process

Different Application Types

Identify the entire portfolio, including micro-services and legacy, cloud-native, web, mobile, outsourced, and acquired apps.

Different Application Elements

Establish a complete inventory of applications, the first-party code, their frameworks, additional third-party components, and their deployment artifacts.

Different Development Tools and Ecosystems

Discover which integrated developer environments (IDE), external dependencies, developer tools (including AI), languages, and frameworks are used or supported. Also, you will want to identify specific CI/CD pipeline systems and cloud environments to map out Code, Build, and Deploy aspects of the SDLC (ex: Github Actions, Gitlab Pipelines, Azure Pipelines, Jenkins Pipelines).

Different Risk Levels

Understand which assets in your inventory carry the most risk (likelihood and impact) and have the most valuable data. Threat modeling balanced with current risk state is a solid consideration at this step.

STEP 2:

Establish Prevention Methods

After taking a thorough inventory of your applications and identifying those most valuable to your organization, the next crucial step is to establish robust prevention methods. This begins by equipping developers with the necessary tools to write secure code from the outset. Fostering a secure coding culture empowers your development teams with the knowledge and best practices to identify and mitigate vulnerabilities early. Simultaneously, embedding security controls directly into your development pipeline ensures that security is an inherent part of the software development lifecycle, preventing flaws from ever reaching production.

These prevention methods are significantly strengthened by integrating specialized security tools:

Veracode Static Analysis (SAST):

Performs static analysis by scanning source or binary code in a preproduction or “static” state to find security flaws or Common Weakness Enumerations (CWEs) early in the SDLC.

Veracode Software Composition Analysis (SCA):

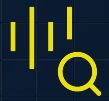
Continuously monitors open-source libraries and third-party components for Common Vulnerabilities and Exposures (CVEs). SCA is also instrumental in creating a Software Bill of Materials (SBOM) and supports standard formats like CycloneDX and SPDX.

Veracode Fix

Guides developers with AI-assisted recommendations and automated code fixes, helping them quickly resolve identified vulnerabilities.

Veracode Package Firewall (VPF)

Proactively detects and blocks malicious packages with up to 60% greater accuracy than competitors, protecting against supply chain attacks.



Veracode Container Security

Provides crucial scanning for cloud-native applications to find container and Infrastructure as Code (IaC) configuration risks, secrets exposure, and misconfigurations that can be remediated before deployment.

STEP 3:

Onboard and Scale Applications

After portfolio understanding and prevention methods, onboard applications with initial scans to establish a security baseline and gain visibility.

Continuous scanning via Veracode SAST and Veracode SCA in the IDE allows developers to maintain velocity without sacrificing security. By integrating directly into the Integrated Development Environment (IDE), these tools provide developers with seamless, real-time feedback. This means fewer vulnerabilities are introduced into the code as they write it, allowing them to focus on core development tasks. For security teams, this continuous scanning across the entire SDLC (for both legacy and cloud-native applications) delivers better accuracy in identifying vulnerabilities and provides consistent visibility into the security posture.

Addressing flaws early is demonstrably more cost-effective, significantly reducing remediation expenses and the potential costs of a security breach, thereby demonstrating a clear Return on Investment (ROI) for DevSecOps initiatives.

STEP 4:

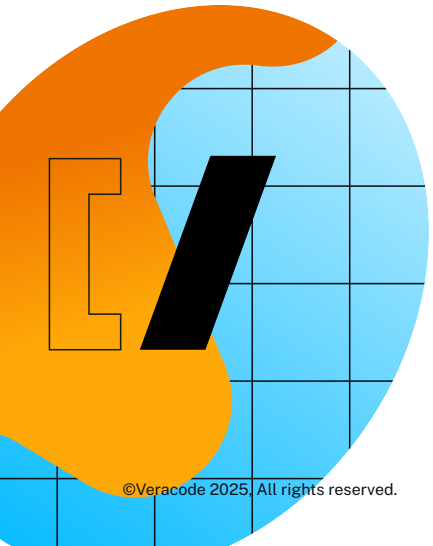
Set Policies

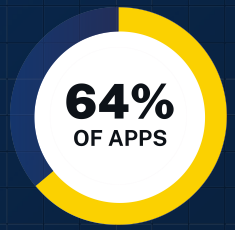
> Define and document clear application security policies, considering risk tolerance, regulatory requirements, and application criticality.

Different applications may have distinct policies. Policy effectively communicates risk and compliance objectives to development teams, establishing clear expectations. Without it, you're unable to manage risk and compliance because there is no signal between the roles.

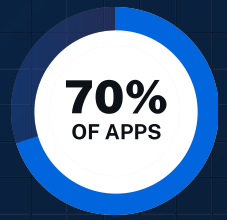
Policy setting requires collaborative input from security, development, and executive sponsors to qualify risk tolerance. Technical controls in the CI/CD process establish a continuous feedback loop for compliance.

Veracode offers configurable policy recommendations (severity, CWE, license risk, CVSS, OWASP, PCI DSS), ensuring effective, comprehensive testing in an auto-scaling cloud environment. Ultimately, policy becomes the shared language bridging security and development, operationalizing risk tolerance, and aligning security decisions with business objectives.





have flaws in first-party code



contain flaws in third-party code.

Approximately 50% of organizations have persistent, high-severity flaws that constitute critical security debt.

STEP 5:

Prioritize and Address Findings

Once applications are onboarded, scanned, and have policies set, it's time to prioritize and address security findings. This involves categorizing and resolving policy violating flaws through remediation or mitigation. Research shows that roughly 64% of applications have flaws in first-party code and 70% contain flaws in third-party code.[1] That's why testing both throughout the SDLC is so critical. More concerning still, approximately 50% of organizations have persistent, high-severity flaws that constitute critical security debt.[2]

If you haven't been integrating security into development thus far, you're likely going to fall into that group of organizations with critical security debt. That's why it's so important to remediate flawed code as early as possible.

Research reinforces the importance of speed in flaw remediation. Development teams that address flaws promptly can reduce critical security debt by a remarkable 75%.[3] By fixing vulnerabilities quickly, these teams build habits and muscle memory around securing code from the start, significantly enhancing improving their security posture and reducing the prevalence of security debt in their applications.

To help development teams fix security flaws and vulnerabilities, security teams must hand off specific, actionable findings with clear directives for remediation. AI-assisted remediation tools can help with addressing findings by rapidly reducing time spent on issue pre-investigation, as long as they're responsible-by-design. Tight collaboration between teams is essential to ensure findings are properly identified, prioritized, and ultimately addressed.

STEP 6:

Leverage Reporting and Analytics

The final step in securing the SDLC involves leveraging reporting and analytics to gain valuable insights into the effectiveness of security measures and to track progress over time. A unified reporting system that consolidates data from diverse security tools, providing a comprehensive, real-time view of your application security program's health such as policy compliance, scanning coverage, open/closed findings, etc. These reporting capabilities help establish a good starting point, pinpoint areas for enhancement, set measurable objectives, and enable continuous monitoring.

Beyond these general reporting insights, a solution like Veracode Risk Manager (VRM) provides a unified view of all application security findings from diverse sources (SAST, DAST, SCA, Container, etc.), automatically prioritizing critical risks, linking issues to their root cause and owner, and delivering Next Best Actions™ to efficiently eliminate security debt.

This combined approach provides essential visibility for demonstrating compliance and assuring stakeholders. General reporting and prioritization, including insights from VRM, help identify trends and patterns in vulnerabilities, enabling proactive security measures and driving continuous improvement within DevSecOps.

For effective security management and board reporting, clear visibility into software security and automated progress is paramount.

Veracode helps HDI Global SE's business initiative for security innovation become a measurable reality.

"From a security management perspective, the visibility into our software and progress being made through automation are paramount to me for reports to the board. Veracode, as a central tool for our visibility and vulnerability management, is very helpful. I use the reports to establish a baseline, identify areas for improvement, set quantitative goals, and track progress against those goals."

Nils Brenneis, Information Security Manager, HDI Global SE

Conclusion

By integrating security into each step of the SDLC, organizations can create a culture of security awareness, reduce risk, and ensure that software is secure from the start.

The six essential steps outlined in this e-book provide a roadmap for optimizing DevSecOps collaboration and integrating security into every SDLC stage. Collaboration between security and development teams is crucial in resolving security flaws promptly and reducing critical security debt. This collaborative approach ensures security accelerates innovation, enabling high-quality, secure digital products.

Securing the SDLC is an ongoing process that requires continuous effort and adaptation. By following these six steps and embracing a culture of security, organizations can build resilience, protect sensitive data, and stay ahead of potential cyber threats.

VERACODE

SCHEDULE A DEMO

A security pro will gladly help you implement these steps in your organization.

Schedule a Demo

1,2. Veracode 2025 State of Software Security Report
3. Veracode 2024 State of Software Security Report