EBOOK

Contrast SECURITY

DevSecOps Buyer's Guide: Application Security

Executive Overview

The software factory is more vulnerable than ever. Traditional approaches to Application Security (AppSec) testing have not kept pace with the scale and complexity of modern applications.

As applications have evolved to become more complex and distributed, the effectiveness of traditional AppSec has fallen behind in several critical areas, including eliminating vulnerabilities during software development, tracking open-source software (OSS) risks and protecting applications post-development release. A recent report by Akamai Industries saw a 257% increase in application programming interface (API) and web application attacks[1]. In addition, we are seeing a surge in attacks on software vendors and the supply chain, as well as a rise in targeted threats on cloud-native application infrastructure.

Choosing an effective AppSec testing solution should be based on the specific requirements of modern software; should work anywhere in the Software Development Life Cycle (SDLC); and must focus on discovering and testing for vulnerabilities in addition to remediating them. Core capabilities include criteria for vulnerability identification, Software Composition Analysis (SCA), runtime protection and compliance, among others.

Organizations must recognize that security is an integral part of the DevOps cycle. While it may be tempting to shift security left into the development process, it is important to understand that security is a fundamental function of the right — i.e., the operations part of the DevOps cycle that ensures performance, resilience and reliability. Prevention and detection must be seen as intertwined and mutually reinforcing. Unfortunately, a shift-left approach to security can fail for a variety of reasons, including lack of security knowledge, overburdening of dev teams, poor integration, misaligned priorities or lack of automation.

In order to meet regulatory and customer demands, organizations must ensure that their software is secure by making security transparent. This involves taking a holistic approach that includes conducting security assessments, implementing secure coding practices, and deploying tools that can accurately identify and mitigate security risks. By taking these steps, organizations can not only meet regulatory and customer demands but can also build trust and confidence in the software they develop.

Use this document as a template for requests for proposals (RFPs) or AppSec vendor-selection projects.

[1] "Enemy at the Gates: Analyzing Attacks on Financial Services," Akamai Technologies, Vol 8, issue 3

Table of contents

01

Limited developer time to remediate vulnerabilities

The first step: recognizing the biggest challenges AppSec teams face today. They are:

LIMITED DEVELOPER TIME TO REMEDIATE VULNERABILITIES

With ever-increasing pressure to accelerate delivery cycles, developers have a limited capacity to find and fix application vulnerabilities. As code volumes increase over time, the backlog of surfaced vulnerabilities also grows. Applications running in production are subsequently exposed, and almost all organizations — more than 99% — report that their applications in production have four or more vulnerabilities[2].

> A concerted effort to remediate the vulnerabilities that put businesses at risk and 'pay down' their security debt is the single most powerful action a company can take to reduce the chance of a breach.

[2] "State of DevSecOps Report," Contrast Security, September 2020

02

Shift smart: The backlash of shifting responsibility

There is a strong backlash against shifting the responsibility too far left[3]. Rather than placing the majority of the burden on developers, we should focus on how to best integrate security practices[4] into the application development process. This will involve collaboration between all parties to ensure that security is considered and implemented throughout the entire process. By taking a collaborative approach to security, we can create applications that are secure and meet the needs of all stakeholders.

Security must be embedded within every phase of DevOps[5], not just as a separate stage. It's not about shifting security left or right; instead, it's about ensuring that security architecture is correctly integrated into the DevOps pipeline. Companies must keep in mind that security is a fundamental function of the right — i.e., the operations part of the DevOps cycle that ensures performance, resilience and reliability — and that not all aspects of security can be shifted left[6]. We must also understand that prevention and detection are not mutually exclusive; rather, they are intertwined and mutually reinforcing. Every failure in prevention must be identified and addressed through detection. In the end, security is an integral part of the DevOps cycle, and it's essential to ensure that it's properly implemented throughout the entire DevOps workflow.

> Security must be embedded within every phase of DevOps, not just as a separate stage. It's not about shifting security left or right; instead, it's about ensuring that security architecture is correctly integrated into the DevOps pipeline.

[3] "Can 'shift left' in DevOps pipelines go too far?" TechTarget, April 4, 2021
[4] "Shift Left is Only Part of Secure Software Delivery," DevOps.com, April 20, 2022
[5] "Shift Left: From Concept to Practice," DarkReading, April 26, 2021
[6] "Shift left: Beyond the cybersecurity buzzword," Security Magazine, Dec. 20, 2022

Contrast SECURITY

03

Developer fatigue

Security teams are under pressure to provide a better way of integrating security[7] into their development process, but it's not easy. Developers need to be provided with a more transparent way of securing code[8] without pushing the line management, project management and security teams to their limits. Some of the factors adding to the problem:

1. **Lack of security knowledge:** A shift-left approach to security relies on developers having the necessary security knowledge to properly identify and address security risks. If developers don't have the required knowledge, they may be unable to adequately secure the code.

2. **Overburdening dev teams:** A shift-left approach can place an increased burden on development teams if the security tasks aren't efficiently implemented. This can cause additional strain on developers and make them less productive.

3. **Poor integration:** If security is not properly integrated into the development process, it can lead to security being treated as a separate, independent process. This can lead to security being overlooked or pushed off until late in the development cycle.

4. **Misaligned priorities:** Security may be seen as a lower priority than other development tasks, leading to security being neglected or inadequately addressed.

5. **Lack of automation:** Without proper automation tools, developers may not be able to quickly and efficiently identify and address security risks. This can lead to security issues going unnoticed or unresolved.

> 68% of incident responders have to defend against two or more attacks simultaneously.[9]

[7] "Cyber responders are outnumbered and under pressure as they defend our modern way of life," Fortune, Oct. 27, 2022
[8] "Key challenges of shift-left tactics," CyberTalk, Nov. 7, 2022
[9] "Cybersecurity Incident Responders Have Strong Sense of Service as Threats Cross Over to Physical World," IBM, Oct. 3, 2022

Contrast SECURITY

04

Tools that cry wolf: Deliver secure code, not false positives

Most AppSec tools create lots of false positives. Security teams need to ensure that their applications are secure, but they also need to ensure that the tools they are using are providing accurate information.

Security is never black and white[10], but the end result of too many false positives is it can lead to a loss of trust in the security tools, as well as confusion about which risks are real and which are false alarms. To ensure secure code, security teams should use tools that are specifically designed to eliminate false positives.

These tools should be easy to use, offer accurate results and provide actionable insights. By implementing tools that accurately identify security risks, security teams can effectively deliver secure code and build trust with developers.

[10] "The Truth About False Positives in Security," The Hacker News, Aug. 9, 2022

05

Increased compliance, trust and transparency for software suppliers

INCREASED COMPLIANCE, TRUST AND TRANSPARENCY FOR SOFTWARE SUPPLIERS

To meet both regulatory and customer demands, organizations must ensure that their software is secure. New laws and regulations require organizations to make security more transparent[11]. In the U.S., federal requirements are demanding software security assurance and Sarbanes-Oxley Act- (SOX-) style attestations, while regulations similar to the General Data Protection Regulation (GDPR) are emerging out of Europe, the Middle East and Africa (EMEA).

Governments expect greater visibility into the software that powers their applications. The Biden administration is set to unveil a national strategy[12] for a comprehensive cybersecurity regulation of the nation's critical infrastructure. The Department of Defense communicated an update to the mandate of their framework to Continuous Authority to Operate (cATO)[13] requiring organizations to take a continuous monitoring approach versus the traditional static approach. This involves taking a holistic approach that includes conducting security assessments, implementing secure coding practices, and deploying tools that can accurately identify and mitigate security risks[14].

> Security must be embedded within every phase of DevOps, not just as a separate stage. It's not about shifting security left or right; instead, it's about ensuring that security architecture is correctly integrated into the DevOps pipeline.

[11] "Cybersecurity Regulations for the Financial Sector," UpGuard, Jan. 10, 2023
[12] "U.S. national cyber strategy to stress Biden push on regulation," The Washington Post, Jan. 5, 2023
[13] "Continuous Authorization To Operate (cATO), Office of the Secretary of Defense," accessed Feb. 7, 2023
[14] "DOD Releases New Continuous ATO Initiative for 'Active' Cybersecurity," Government CIO Media, Feb. 4, 2022

Contrast SECURITY

06

The measure of modern AppSec

An effective AppSec testing solution for modern development environments (e.g., DevOps, Agile) needs to include a few critical benefits to the organization:

- Be optimized for remediation (both in terms of number of vulnerabilities remediated as well as MTTR)
- Measure risk and risk reduction at an application, application portfolio, business unit or company level
- Have completeness of coverage across the entire SDLC
- Be adaptable to both modern and legacy technology
- Be usable and consumable by any technology resource at all phases of the development cycle

As enterprises look to replace outdated AppSec testing tools and begin measuring their programs through the lens of risk reduction rather than coverage, they can evaluate modern AppSec solutions based on criteria in the following categories (see Section 7, Best practices, page 17):

- Vulnerability Identification and Analysis
- Software Composition Analysis (SCA)
- Runtime Protection
- Risk Posture
- Reporting and Compliance
- Enterprise Readiness
- Developer Experience
- Platform Requirements

SECURITY COVERAGE FROM DESIGN TO PRODUCTION

Choosing a modern, effective DevSecOps platform can help organizations unblock their continuous integration/continuous deployment (CI/CD) pipeline by finding more real defects in real time. It can also turn developers into security experts with user-friendly "how-to-fix" guidance and prebuilt command-line interface (CLI) tools. Finally, it can offer protections to ensure that applications are shipped securely, even with open or unknown vulnerabilities in production.

> A comprehensive approach to DevSecOps requires tight integration between application security and DevOps toolsets. But 45% of organizations indicate they struggle to ensure security across their DevOps toolchain.[15]

[15] "Modernize your CI/CD," GitLab, accessed December 21, 2020.

07

Best practices when evaluating a modern DevSecOps platform

For DevSecOps to be effective within today's modern development environments, its coverage must extend across all parts of an application. This includes custom code, open-source libraries, third-party components and application programming interfaces (APIs).[16]

The following checklist can help guide organizations when developing request for proposal (RFP) materials for evaluating and purchasing a DevSecOps solution for their organization.

VULNERABILITY IDENTIFICATION AND ANALYSIS

- Provide visibility into vulnerabilities to developers at the point of development and testing
- Provide options to test on developer desktops, security gates and build stages
- Provide options to test client-side as well as server-side code
- Offer specific guidance to fix vulnerabilities in the SDLC
- Measure the number of routes exercised to identify coverage gaps

SOFTWARE COMPOSITION ANALYSIS (SCA)

- Provide visibility into vulnerabilities present in third-party libraries (i.e., open-source components) and recommendation to fix
- Pinpoint potential third-party library licensing risks and associated vulnerabilities across applications

[16] Derek Rogerson, "What You Need to Know About the New IAST and RASP Guidelines in NIST 800-53," Security Boulevard, March 19, 2020

RUNTIME PROTECTION

- Protect vulnerable applications during runtime for third-party and custom-developed applications (desired) in production
- Effectively prevent/block threats from exploiting any present code vulnerabilities
- Protect applications from Open Web Application Security Project (OWASP) Top 10 vulnerabilities such as cross-site scripting (XSS), SQL injection (SQLi), command injection, XML injection, and other publicly known Common Vulnerabilities and Exposures (CVEs) without patching
- Protect applications from zero-day exploits without remediation
- Includes prebuilt integration capabilities with commonly used monitoring systems — Splunk, Securonix, AppDynamics, etc.
- Provide comprehensive data output (i.e., timestamp, payload, URL, port, source/destination IP, violation/event type, session ID, cookies, stack trace)

RISK POSTURE

- Provide a risk score for an application and track how that score changes over time
- Benchmark the risk score against the rest of the organization's application portfolio
- Break out the risk score of third-party/open-source libraries from the risk score of custom code
- Adjust a risk score if runtime protection is turned on

REPORTING AND COMPLIANCE

- Measure vulnerability open and close rates
- Measure average remediation times
- Measure attacks against specific vulnerabilities
- Identify noncompliant applications by policy — Payment Card Industry (PCI), OWASP, etc.
- Directly address National Institute of Standards and Technology (NIST) Special Publications 800-37 and 800-53 risk management guidelines and PCI DSS Requirement 6 for designing and maintaining secure systems
- Consolidate results across multiple assessment types and policies
- Support customized reporting within the platform

ENTERPRISE READINESS

- Perform AppSec without sending application source code or binary or bytecode to cloud servers to complete the process
- Define custom policies for each application
- Define grace periods within policy (viz., time for teams to fix before failing policy)
- Apply a policy at any level of the enterprise (viz., for a single team or business unit, or across the entire organization)
- Define flaw severities, categories, common weakness entries (CWEs) and standards that comprise the policy
- Include vendor assistance during application onboarding, uploading and results publication phases

Contrast SECURITY

DEVELOPER EXPERIENCE

- Integrate with bug-tracking systems, chat tools, ticketing systems as well as containerization tools (e.g., Docker, Kubernetes)
- Initiate assessments through a CLI
- Scan public or private repositories
- Detect vulnerabilities on a developer desktop (e.g., an integrated developer environment [IDE])

PLATFORM REQUIREMENTS

- Offer a single platform to manage all solutions — including AST (both custom code and open-source libraries) and runtime protection
- Use active attack information to prioritize vulnerability remediation
- Use runtime protection rules to defend against unpatched CVEs
- Use runtime protection rules to defend against open vulnerabilities developers did not fix
- Use runtime protection rules to defend against zero-day vulnerabilities

Contrast SECURITY

Contrast Security provides the industry's most modern and comprehensive Application Security Platform, removing security roadblocks inefficiencies and empowering enterprises to write and release secure application code faster. Embedding code analysis and attack prevention directly into software with instrumentation, the Contrast platform automatically detects vulnerabilities while developers write code, eliminates false positives, and provides context-specific how-to-fix guidance for easy and fast vulnerability remediation. Doing so enables application and development teams to collaborate more effectively and to innovate faster while accelerating digital transformation initiatives. This is why a growing number of the world's largest private and public sector organizations rely on Contrast to secure their applications in development and extend protection in production.

240 3rd Street
2nd Floor
Los Altos, CA 94022
Phone: 888.371.1333
Fax: 650.397.4133

Contrast SECURITY

contrastsecurity.com