Techstrong Group
a Futurum company

A DevSecOps Buyer's Guide for Application Security

SPONSORED BY

CONTRAST SECURITY

Application Security Must Meet the Moment

Today, application security (AppSec) faces unprecedented challenges. As software architectures become more complex—spanning enterprise and web applications, application programming interfaces (APIs), microservices, and cloud environments—the attack surface dramatically expands. Cyber threats such as zero-day exploits and supply chain attacks are becoming more sophisticated, rendering traditional security approaches insufficient.

The core problem is a critical mismatch of velocity: Development teams now deploy code at a pace that far outstrips application security teams' ability to mitigate potential vulnerabilities. Traditional security approaches—such as manual code reviews, periodic penetration testing, and tools that generate numerous false positives—have become significant bottlenecks in the software development process. This challenge has been further intensified by the rapid adoption of AI, with 62% of developers currently using AI tools and an additional 14% planning to implement them in the near future.[1]

While AI-powered development tools dramatically accelerate code creation, they also introduce new security considerations, from potential vulnerabilities in AI-generated code to novel attack vectors targeting machine learning components integrated into applications.

As companies intensify their efforts to secure software code, the volume of potentially vulnerable applications continues to grow. In addition, older applications, which often lack comprehensive automated test coverage, are becoming increasingly susceptible

to sophisticated application-layer attacks. Moreover, current testing practices typically fail to thoroughly examine an application's runtime environment—leaving critical vulnerabilities in code, data flows, and application logic undiscovered before production deployment. This creates a challenging trade-off where organizations must balance the competing priorities of rapid innovation and robust security.

The result is mounting "security debt": Applications are hastily deployed with insufficient testing, leaving organizations exposed to increasingly sophisticated cyber threats. In this high-stakes environment, security can no longer be treated as an afterthought. Instead, it must evolve into an integrated, proactive discipline that can keep pace with modern development cycles. This new approach goes beyond traditional traffic blocking, focusing instead on continuously enhancing application resilience throughout the lifecycle.

Security must fundamentally transform. It can no longer function merely as a periodic checkpoint; instead, it must become a continuous, integrated practice that spans both the software development lifecycle and production environments.

What Must Change?

Security must fundamentally transform. It can no longer function merely as a periodic checkpoint; instead, it must become a continuous, integrated practice that spans both the software development lifecycle and production environments. Runtime application security offers the capability to proactively identify vulnerabilities in legacy applications before potential attackers can exploit them. By analyzing production traffic, developers can gain targeted insights that go beyond traditional pre-production testing. These insights include detailed information about specific code lines, actual data flows, precise remediation recommendations, and potential consequences of leaving vulnerabilities unaddressed.

1 StackOverflow 2024 Developer Survey.

When AI-assisted development enters the picture, this need for transformation becomes even more urgent. Modern security approaches must embrace AI tools themselves, utilizing machine learning models to understand patterns in AI-generated code, predict potential vulnerabilities unique to AI development workflows, and adapt protection strategies to the acceleration of development timelines that AI enables.

Organizations can transform security into a core development principle by embedding protective measures throughout the entire software lifecycle. Runtime vulnerability detection involves instrumenting code and monitoring its behavior during live user interactions, providing developers with immediate feedback at any stage of development or deployment. While identifying and addressing potential security issues during design and testing is crucial, the security challenge doesn't end there. Even well-crafted applications can become vulnerable as cyberattacks grow more sophisticated. Recognizing this ongoing threat, an increasing number of organizations are now focusing on continuous security analysis directly within production environments.

This approach of runtime application security creates multiple defensive layers:

1. Runtime observation identifies potentially dangerous functions, enabling early threat modeling and risk assessment by providing concrete insights into application behavior.
2. Comprehensive testing—both manual and automated—uncovers code vulnerabilities across the entire system, from endpoint interactions to individual lines of code, delivering precise and actionable findings without speculation.
3. Runtime analysis provides a comprehensive overview of third-party libraries, revealing their inherent risks and actual usage, which empowers teams to make strategic and efficient remediation decisions.
4. Continuous behavioral analysis in production proactively identifies potential threats before they can be exploited by malicious actors, effectively conducting real-world security testing.
5. AI-powered detection systems can further enhance these defensive layers by learning normal application behavior patterns, identifying anomalies that might indicate exploits targeting AI components, and automatically generating protection policies tailored to AI-accelerated development environments.
6. Runtime application security for applications running in production enables the detection and response to attacks that are actively exploiting application vulnerabilities enabling organizations to respond swiftly. This approach allows teams to either block potential exploits or enhance application resilience by leveraging hyper-relevant, end-to-end information directly from the production environment.

THE RESULT?

A robust security strategy where every team member treats protection as fundamental, ensuring that if one control fails, another stands ready to defend. Development, operational, and security teams can now collaborate seamlessly, utilizing integrated tools that support their unique workflows while contributing to a unified, holistic security framework.

A Cultural Progression and a Shared Vision

A holistic approach to application security transforms traditional team dynamics by breaking down silos and creating shared responsibilities. By integrating security directly into development workflows, organizations achieve a more collaborative and proactive security strategy.

Developers can address vulnerabilities more quickly by leveraging precise identification of affected lines of code, along with contextual insights to ensure accurate fixes. Security teams gain deeper visibility into vulnerabilities and can more accurately identify viable exploits, helping development teams mitigate affected code effectively.

Security Operations Center (SOC) teams gain deep insights into the application layer, enabling them to detect and respond to previously invisible threats with unprecedented speed. By capturing and analyzing detailed threat intelligence, these teams can provide critical feedback to development teams, allowing for continuous improvement of application resilience and creating a dynamic defense against evolving attacks. This synergy creates an environment where teams work together, using integrated tools that align with both developer and production workflows.

For stakeholders, this approach delivers tangible benefits: DevOps teams receive clear security requirements and automated tools, application security teams focus on strategic coaching, SOC teams can protect applications without relying on hotfixes from development, and compliance teams gain easier regulatory documentation. The results are reduced friction, accelerated decision-making, and enhanced security without compromising business agility.

Most importantly, this strategy cultivates a culture of collective security responsibility. Rather than treating security as an isolated function either in development or in production, all teams proactively identify and address risks within their expertise. Through shared metrics, cross-team communication, and joint planning, security becomes a seamless part of the software development lifecycle.

Key Challenges in AppSec

Securing modern applications is a complex game of technological, procedural, and cultural challenges. Organizations struggle with critical vulnerabilities stemming from:

- **Blind spots:** Lacking real-time visibility into application behavior and network interactions, leaving critical weaknesses undetected.
- **Risk management:** Balancing vulnerability mitigation without disrupting development workflows.
- **Alert precision:** Navigating the thin line between aggressive scanning and permissive settings that might miss critical security threats.
- **AI-assisted vulnerability detection:** Leveraging AI capabilities to detect complex vulnerabilities while avoiding false positives that can undermine developer confidence in security tooling.
- **Development velocity:** Integrating security measures that enable faster fixing without hampering innovation and tight project timelines.
- **AI acceleration effects:** Adapting security processes to match the increased speed of development enabled by AI coding assistants without creating new gaps in protection.
- **Vulnerability triage:** Prioritizing and addressing risks across expansive, interconnected application ecosystems with limited resources.
- **Regulatory compliance:** Adapting to rapidly evolving legal and industry security requirements.

Effectively addressing these challenges demands a holistic strategy that synchronizes technology, process optimization, and organizational culture—transforming security from a bottleneck to a strategic enabler of robust software development.

Core Components of AppSec Solutions

Effective AppSec isn't about theoretical defenses; it's about real-world resilience. More than just safeguarding applications, effective and efficient AppSec embeds security into the fabric of software development from inception to deployment. This is the best strategy to ensure resilience against evolving threats.

The Keys to AppSec Success Include:

- **Continuous real-world visibility** that automatically maps and tracks all applications, APIs, and dependencies dynamically. Maintaining a living software bill of materials that tracks internal and third-party components is paramount to providing instant visibility into actual security postures.
- **Integrated security testing** embedding automated security testing into the development pipeline—static application security testing (SAST), dynamic application security testing (DAST), software composition analysis (SCA), interactive application security testing (IAST), and API security testing—that run with every code change and deliver immediate, actionable developer feedback.
- **Application detection and response** (ADR) empowers apps to spot and counter attacks instantly, sometimes using runtime application self-protection (RASP) to actively block exploits. Far from theoretical, ADR and web application firewalls (WAF) are dynamic defenses that tackle real threats—WAFs act as a broad shield, filtering out known attacks with signature-based detection, while ADR zeroes in on the application layer, making smart calls based on how the app behaves internally. Even top-tier security testing can't catch every risk, which is why strong runtime protection is a vital, yet often ignored, piece of DevSecOps. This powerful layer lets developers and DevOps teams prioritize critical apps, confident that solid security is always on guard.
- **AI-powered detection capabilities** that leverage machine learning to establish behavioral baselines for applications, recognize pattern deviations that might indicate novel attacks, and automatically adapt protection rules based on emerging threat intelligence without requiring manual updates.
- **Threat modeling and risk assessments** that identify potential security weaknesses early in the design activity, automatically, at scale, providing meaningful insights for security experts and developers.
- **Intelligent policy enforcement** with dynamic security policies that adapt to evolving threats, automated compliance checking, and risk-based security gates that reflect actual operational contexts. Ensuring policies comply with regulations and standards helps organizations address increasing regulatory pressures.
- **Developer-centric security** that provides contextual, just-in-time security guidance that helps developers understand and immediately remediate vulnerabilities without requiring deep security expertise. Making security checks as seamless as submitting code changes is crucial for enabling and accelerating secure application development, and tools that integrate effortlessly into developers'

daily workflows promote a security-first culture without sacrificing productivity. Tools and processes should clearly show developers what to fix first, explain the risks of not fixing it, and reveal how likely attackers are to exploit the flaw. They should also make the problem easy to understand and recreate with full endpoint and code details, while giving developers the knowledge to turn vulnerability discoveries into an opportunity to improve their skills.

- **AI-assisted remediation support** that provides intelligent recommendations for fixing vulnerabilities based on context, code patterns, and historical fixes, while also educating developers about secure AI implementation practices and helping them write more secure code when working with AI components.
- **Robust reporting and analytics** with customizable dashboards, risk scoring, and the ability to track security improvements over time provides stakeholders at all levels visibility into security metrics, trends, and risks.

By aligning with these elements, organizations can build a robust AppSec program that safeguards applications while supporting innovation and growth. The ultimate goal: A unified, intelligent system that transforms application security from a compliance checklist to a responsive, adaptive defense mechanism that integrates with and accelerates continuous integration (CI) and continuous delivery (CD) pipelines.

Best Practices

The only fruitful, long-lasting approach to application security is to implement comprehensive security measures across the entire software development lifecycle while maintaining developer productivity, enabling developer growth, improving application resilience, and ensuring compliance with regulatory requirements. By adopting a well-rounded security strategy, organizations can better protect their applications from emerging threats while fostering a security-first culture among development teams. Best practices for holistic AppSec include:

Test coverage from design to production: Implementing security controls and checks throughout the entire software development lifecycle, from initial design to the application running in production. With this in place, developers can routinely identify and address vulnerabilities. The security of CI/CD pipelines is particularly crucial as they represent a critical infrastructure component and a potential attack vector if compromised.

Reporting and compliance: Regular monitoring and reporting of security metrics, particularly vulnerability remediation rates and compliance status. This enables organizations to maintain visibility into their security posture and demonstrate adherence to regulatory requirements. It is a data-driven approach that helps AppSec teams prioritize fixes and allocate resources effectively while providing stakeholders with clear insights into security performance.

Integration with developer tools: Security being a seamless part of the developer experience, workflows, and culture. By providing security functionality that works naturally within integrated development environments (IDEs), command-line interfaces (CLIs), and other familiar developer tools, organizations can reduce friction and encourage broader adoption of security practices among development teams.

AI tool integration: Ensuring security tools integrate directly with AI coding assistants and other AI development tools, providing real-time security feedback on AI-generated code and helping developers implement secure patterns when working with AI capabilities.

Vendor Evaluation Checklist

Selecting the right AppSec vendor is critical to safeguard your software development lifecycle and protect your organization from ever-evolving threats. An effective vendor must align with your security goals, address your key AppSec challenges, and provide the critical elements for success; all while supporting AppSec best practices. This checklist will guide you through the key factors to evaluate when assessing AppSec vendors, ensuring you choose a partner that meets your organization's unique needs and scales with your security strategy.

AppSec Tool Requirements

Runtime protection - test and protect applications while running in production

- ✓ Identify vulnerable lines of code, along with details of application logic and data flows
- ✓ Runtime protection against exploits of custom and third-party code vulnerabilities
- ✓ Effective threat detection/prevention
- ✓ OWASP top 10 protection
- ✓ Zero-day protection
- ✓ Unpatched common vulnerabilities and exposures (CVE) protection
- ✓ Comprehensive application attack insights for remediation and forensic analysis

Code assessment

- ✓ Visibility during development, testing and production
- ✓ Multiple test options: Developer sandbox, security sandbox, staging, and runtime environments
- ✓ AI-generated code evaluation capabilities

Software composition analysis (SCA)

- ✓ Visibility into third-party library vulnerabilities
- ✓ Specific remediation guidance
- ✓ Gap analysis/focus on executed code
- ✓ Identification of licensing risks

Security

- ✓ Risk scoring of custom and third-party code
- ✓ Benchmarking across the organization's application portfolio
- ✓ Risk scoring accounting for runtime protection
- ✓ Vulnerability detection
- ✓ Real-time alerting
- ✓ Alert prioritization
- ✓ AI-specific threat intelligence
- ✓ AI vulnerability severity scoring

Reporting

- ✓ Vulnerability open/close rates
- ✓ Vulnerability remediation time
- ✓ Attacks against specific vulnerabilities
- ✓ Consolidation across assessment types, policies, and apps
- ✓ Customizable reporting
- ✓ AI usage and security impact metrics

Compliance

- ✓ Compliance by policies (OWASP, PCI-DSS, etc.)
- ✓ Support for NIST 800-37, NIST 800-53, PCI-DSS Requirement 6

AppSec Platform Requirements

- Unified/integrated management for vulnerability remediation and runtime protection
- Integration with DevOps CI/CD workflow and tools
- Integration with security stacks (security information and event management (SIEM) and security orchestration, automation, and response (SOAR))
- Prioritization via active attack data
- Custom policies per application
- AI development tool integration capabilities

About Contrast Security

Contrast Security is the global leader in Application Detection and Response (ADR), empowering organizations to see and stop attacks on applications and APIs in real time. Contrast embeds patented threat sensors directly into the software, delivering unmatched visibility and protection. With continuous, real-time defense, Contrast uncovers hidden application layer risks that traditional solutions miss. Contrast's powerful Runtime Security technology equips developers, AppSec teams and SecOps with one platform that proactively protects and defends applications and APIs against evolving threats.

How Contrast Security Enables Secure Development

- Empowers developers to implement secure-as-they-code, accelerating release cycles with real-time risk visibility.
- Secures the entire application stack and software supply chain, auto-remediating exploitable vulnerabilities with Contrast AI.
- Enables seamless integration with preferred AI models with Contrast MCP server, automating and simplifying security across the software lifecycle.
- Unifies development, security, and operations with real-time, correlated insights into attacks and live vulnerabilities, enabling rapid resolution through AI-powered remediation guidance.

- **Discover how Contrast Security enables SOC teams to eliminate application-layer blind spots with Contrast Application Detection and Response (ADR).**

- **Start protecting what matters most and fix the vulnerabilities using Contrast Application Security Testing (AST) today!**

Techstrong Group

a Futurum company

SPONSORED BY

CONTRAST SECURITY