EBOOK

Contrast SECURITY

Using Security Instrumentation to Analyze and Protect Software

Executive Overview

Traditional application security (AppSec) remains a critical problem at a development operations (DevOps) level—slowing delivery cycles and incurring valuable time identifying and remediating vulnerabilities. In most cases, companies opt to forego robust security testing in order to accelerate time to market—leaving their code riddled with multiple vulnerabilities.

To address these challenges, organizations need to look to an instrumentation-based approach for application and application programming interface (API) testing. An effective AppSec platform that unifies key instrumentation solutions for testing both code and libraries enables continuous security testing for working applications across all critical points in the development process (including preproduction and production). This allows teams to streamline software development and increase the velocity of innovation without compromising security.

> Integrating AppSec into DevOps can be a challenge. But the urgency of going to production without proper AppSec testing practices will only lead to a buildup of defects that will cost more in the long run.[1]

[1] "Nine Best Practices For Integrating Application Security Testing Into DevOps," Forbes, July 5, 2019.

Table of contents

01

Traditional AppSec Testing is Behind the Times

Existing approaches to AppSec testing are not getting the job done. This is evident in the fact that the number of vulnerabilities per application is the same today as it was in 2000—26.7 serious vulnerabilities.[2]

Most current AppSec tools are complex—which slows down development operations (DevOps) processes. By their nature, traditional AppSec tools, such as static analysis and dynamic scanning, require security experts to install, configure, schedule, and manage each of the different tools and solutions. They also require extensive triage as they generate large numbers of false positives. After almost 20 years with these technologies, the consensus is that they haven't worked.

That's because these approaches place developers and security teams at odds with each another when it comes to their objectives. DevOps is tasked with getting a product to market, while security is tasked to find problems. Security teams cannot win this battle—which is why most applications have multiple critical flaws that can lead to breaches. A different approach is needed: Improving AppSec requires finding better ways for developers and security teams to work together.

> *More than half (52%) of all breaches involve hacking, and web applications are by far the most common vector for hacking-based breaches.[3]*

> *55% of security professionals said it is difficult to get development teams to prioritize remediation of vulnerabilities—even if it's a performance metric for developers.[4]*

[2] "Malware and ransomware attack volume down due to more targeted attacks," Help Net Security, February 5, 2020.
[3] "2019 Data Breach Investigations Report," Verizon, April 2019.
[4] "2019 Global Developer Report: DevSecOps," GitLab, July 2019.

Contrast SECURITY

02

Instrumentation Offers a Path Forward

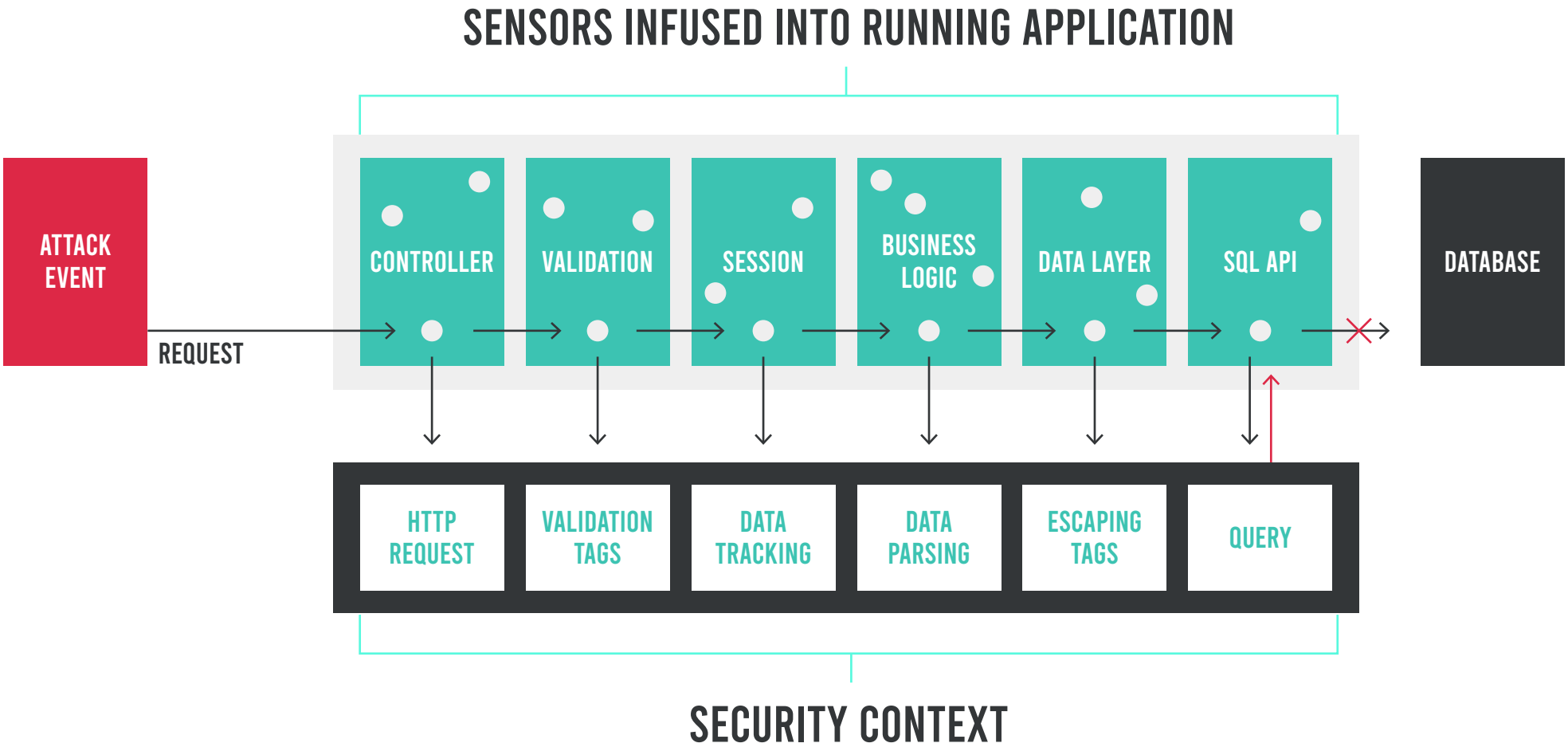
Key to a strong AppSec platform is uniting development and security. An instrumentation-based approach to application testing can help realign development and security objectives while eliminating counterproductive workflows that slow down development and operations. It's similar to the parallel approach taken to solve application performance testing (accomplished by the likes of New Relic and AppDynamics, among others).

Here, dynamic byte code instrumentation helps identify vulnerabilities inside running applications. It can protect applications in the same ways that smoke detectors and fire suppression systems protect buildings. As an application starts up, an agent automatically places instrument sensors at specific locations within the application code (including libraries, frameworks, and application servers) to observe operating conditions and ensure safe operation. If an outside threat seeks to exploit a vulnerability, sensors can detect the activity in real time and respond accordingly.

> *By 2022, 10% of coding vulnerabilities identified by static application security testing (SAST) will be remediated automatically with code suggestions applied from automated solutions, up from less than 1% today.*[5]

[5] "Magic Quadrant for Application Security Testing," Gartner, April 18, 2019.

Contrast SECURITY



Runtime applications protected by security instrumentation.

Following are Some of the Core Capabilities Needed in an AppSec Platform:

AUTOMATION OF APPSEC.

Instead of an asynchronous development and testing loop that impedes time to market, instrumentation-based security becomes a parallel process that automatically launches with applications under real-world operating conditions. This provides direct, real-time vulnerability analysis and threat telemetry to improve application security without slowing down DevOps. This also reduces the demand on overburdened security teams to ensure validation under the pressure of delivery—at a time when the majority (65%) of organizations report a shortage of cybersecurity staff.[6] And at the same time, these capabilities dynamically scale across any number of deployed applications.

DISCOVERY, VISIBILITY, AND VERIFICATION OF VULNERABILITIES.

The old adage, "You can't secure what you can't see," applies here. Organizations need to know where all the software in use is running. Instrumentation supports automatic software inventory to track all applications, APIs, libraries, and frameworks across all environments (e.g., production, test servers, development servers) with 24x7 monitoring and risk management.

[6] "Strategies for Building and Growing Strong Cybersecurity Teams: (ISC)2 Cybersecurity Workforce Study 2019," (ISC)2, November 6, 2019.

POLICY-BASED SECURITY CONTROLS.

Like physical sensors in a factory, instrumentation measures the operating conditions within an application—allowing organizations to then set policies in the event that a potential event is detected. Policy-based control responses range from alerts with detailed contextual analysis or real-time remediation actions to prevent an attack from spreading.

INTEGRATIONS AND REPORTING.

Organizations should treat security vulnerabilities the same way as they do other bugs. The era of PDF-based security reporting is over. Modern application security works through instant notifications and alerts, sending details immediately into the tools that development and operations teams are already using. This enables quick remediation at the lowest cost possible.

> *Over half of cybersecurity professionals indicate their organization is at moderate or extreme risk due to staff shortages, and AppSec is an area where the gaps are the most glaring.*[7]

[7] "Strategies for Building and Growing Strong Cybersecurity Teams," (ISC)2 Cybersecurity Workforce Study 2019, November 6, 2020.

03

Choosing an Instrumentation Platform

As part of a unified platform that can add on capabilities as requirements grow, an effective instrumentation-based solution for AppSec should include three main functions:

- Interactive application security testing (IAST), which is run in preproduction, detects vulnerabilities in both custom code and libraries during normal use by gathering data from running code.
- Software composition analysis (SCA) analyzes libraries to identify potentially vulnerable third-party and open-source components. This is particularly important with 84% of applications consisting of more than half open source code.[8]
- Runtime application self-protection (RASP) is run in production to verify queries and prevent vulnerabilities from being exploited inside the application (both custom code and libraries). Because RASP runs inside the app itself, it provides more effective protection than a web application firewall (WAF), which sits in front of the application.

Instrumentation enables continuous application security (secure, analyze, verify, defend) for working applications across all critical points in the development process (Figure 1).

[8] Stan Wisseman, "Master your code's open source or it will bite you in the app sec," TechBeacon, October 3, 2019.

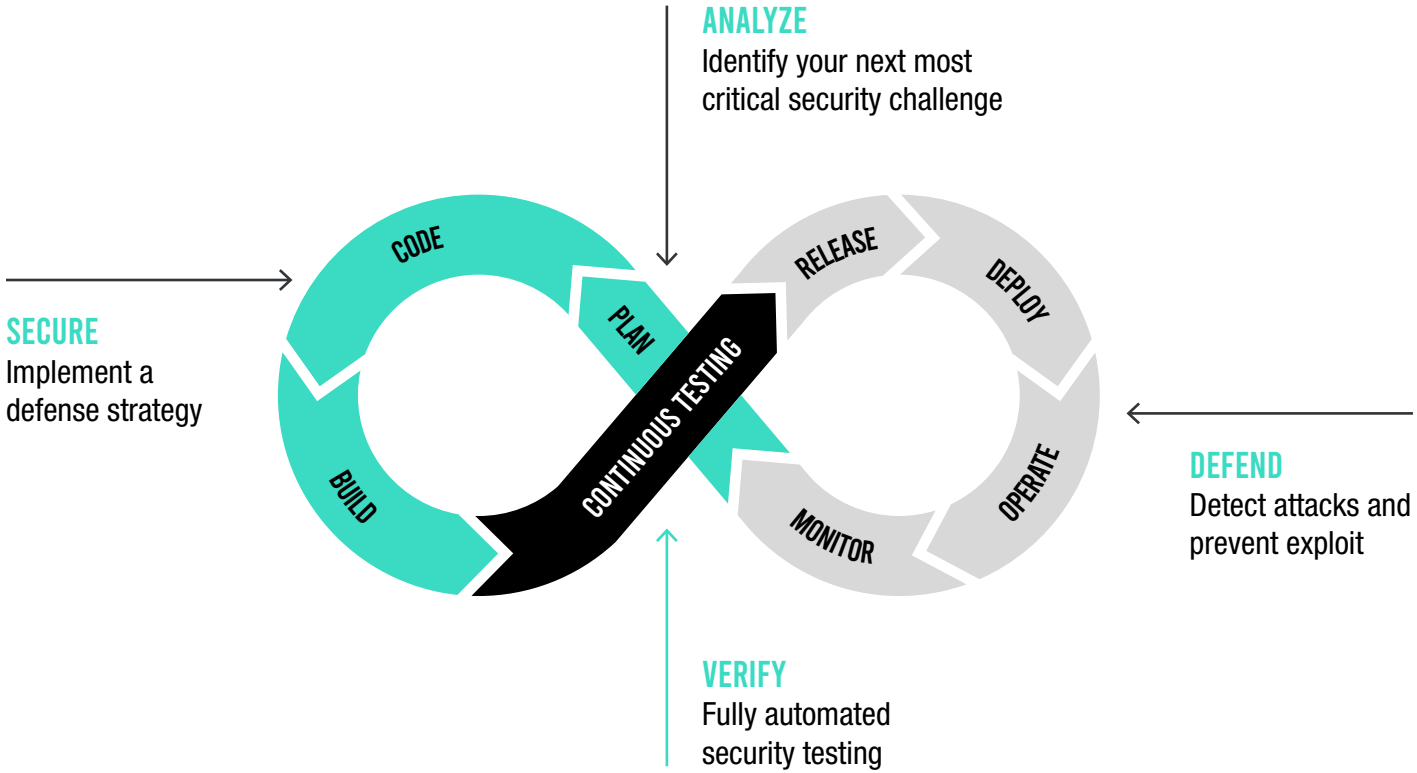


Figure 1: Instrumentation of AppSec involves four key security components: secure, analyze, verify, and defend.

Security teams must also understand privacy ramifications and where compliance plays a role.[9] Instrumentation enable organizations to comply with security and regulatory standards such as Federal Information Security Management Act (FISMA), National Institute of Standards and Technology (NIST), Payment Card Industry Data Security Standard (PCI DSS), and Open Web Application Security Project (OWASP).

[9] "What Happens When You Inject Security into DevOps: DevSecOps," InformationWeek, November 7, 2019.

04

Aligning Development and Security for True AppSec

Instrumentation-based application testing improves security without skilled security staff or the need to change code. It can help developers push code into production much faster than formal processes for testing and approval. Finally, organizations can transcend adversarial relationships and put developers and security on the same operational cycle—all while improving the security of the applications they produce.

To recap, using instrumentation to automate AppSec offers advantages in the following areas:

- Speed. Provides real-time vulnerability analysis and threat telemetry
- Accuracy. Directly measures the running application
- Scale. Runs in parallel across any number of applications
- Process fit. Aligns development and security operations
- Cost. Eases the burden on security staff while reducing operating expenses (OpEx)

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