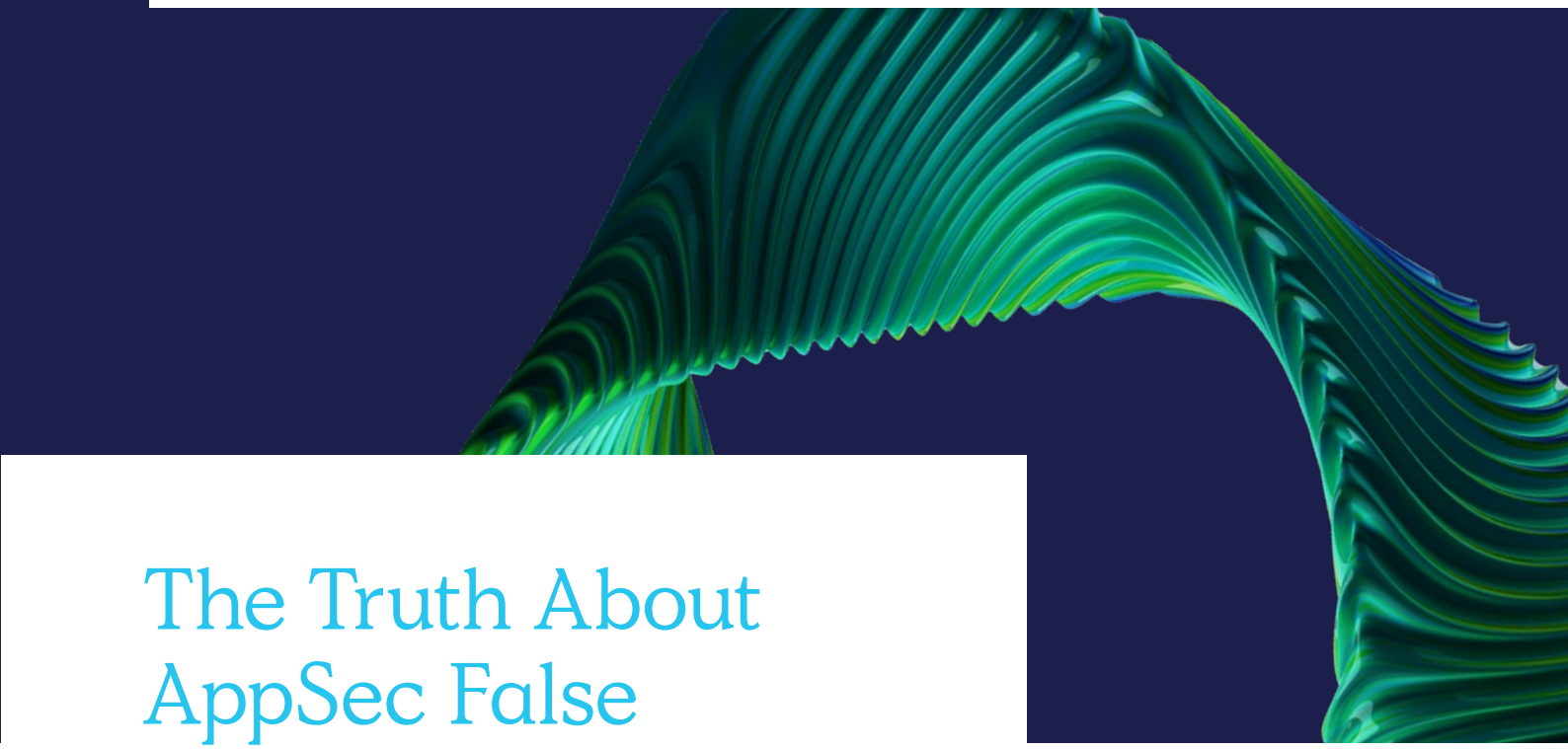
Contrast SECURITY



The Truth About AppSec False Positives

Lack of Accuracy is Burying Organizations with Erroneous Alerts

Executive Overview

Digital transformation is a critical mandate for many enterprises that have turned to DevOps and Agile to speed application development cycles and accelerate the push of new products and services. However, application vulnerabilities are the leading cause of enterprise breaches. Traditional approaches to this problem, such as vulnerability scanners, are too cumbersome and error-prone to be effective in modern high-speed software development environments. They also create a deluge of false positives that create more work and added risk for security teams and their organizations.

False positives impact virtually every area within a security organization. Research conducted by the Ponemon Institute shows that security teams spend 25% of their time chasing false positives.[1] Security teams are overburdened with the time required to manage them, while developers develop alert fatigue and tune out the alerts or ignore them altogether—ratcheting up risk. These challenges, in turn, create friction between security and development teams.

Digital transformation mandates are prompting most enterprises to adopt DevOps and Agile to speed application development cycles and accelerate the push of new products and services. These breakneck speeds are fueled by modern approaches such as Agile and DevOps. However, DevOps and Agile create a huge gap between the demands for faster software development and the challenges introduced by legacy software security tools. Many companies still rely on security tools that are based on decades-old scanning models. This automated testing approach can cause problems such as false positives for security teams.

Understanding the Problem of AppSec False Positives

False positives increase work for already overburdened security teams, as penetration testers must go through every reported vulnerability and manually verify them. And with organizations adding more applications and speeding development cycles, the volume of false positives grows even faster. According to a new report from the Neustar International Security Council (NISC), over one-quarter of security alerts fielded within organizations are false positives.[2] In the report, senior security professionals mention the need for more accurate solutions to alleviate the massive alert volumes bombarding developers and incident response teams.

An overabundance of false positives can put a great burden on security teams. Over one out of five organizations cite false positives as one of their biggest hurdles in maximizing the value of security information and event management (SIEM).[4]

> “More than half of CIOS think legacy applications are delaying digital transformation.[3]

Contrast SECURITY

IMPACT OF USING LEGACY APPLICATION SECURITY TOOLS

Application security (AppSec) specialists use web application security scanners, such as legacy static application security testing (SAST) and dynamic application security testing (DAST) tools, often in conjunction with the penetration testing process. The problem with these AppSec tools is that they are based on nearly two-decades-old technology approaches that were designed for traditional waterfall-based development methods. For high-velocity DevOps and Agile environments, these legacy scanning solutions simply do not scale.

In order to address the deficiencies, organizations often attempt a "tool-swamp" approach where they run a combined group of security approaches, such as legacy SAST and DAST, in the hope that they receive a positive outcome. This hybrid approach to detect vulnerabilities is far from perfect (see sidebar). Legacy AppSec techniques simply do not work at the velocity and scale of modern software. These tools also generate large numbers of false positives, as well as false negatives. As a result, security teams, including quality assurance (QA) and incident response specialists, must perform manual steps to resolve the issues.

This diagnosis and remediation process is incredibly time-consuming and creates a development release bottleneck. Developers get frustrated that they are wasting time chasing vulnerability "mirages." Additionally, measured on velocity of release cycles, they become increasingly frustrated with code commit and release delays. This leads to alert fatigue, where they begin to ignore security alerts that may contain actual vulnerabilities that pose risk.

There is also increased risk in using these legacy scanning tools, which are outdated and have failed, as a category, to deliver on their promises. Worse, the adoption of these scanning tools causes tensions within businesses, and in some cases, cultivates a false sense of security. In general, they lack the visibility and application context needed to produce accurate results. Whether a legacy SAST or DAST tool, each is unnecessarily complex and requires the involvement of experts. In addition, these legacy tools are so unreliable and dependent upon experts that they cannot scale to meet the needs of most organizations.

NOISY RESULTS ARE A COMMON THEME OF LEGACY SAST TOOLS

Legacy SAST tools deliver a distinct advantage for security teams. They identify vulnerabilities early in the development process, so teams can detect and fix them before software is deployed. But the problem is that SAST tools do not execute code. Therefore, they are prone to false positives—identifying perfectly safe code as vulnerable. The OWASP Benchmark Project finds that the overall accuracy score for a legacy SAST solution is just 20%.[5] As a result, it is no surprise that SAST tools are known for "noisy" results that are timeconsuming and difficult to diagnose and trace.

VULNERABILITY INSIGHT IS MISSING WITH DAST TOOLS

DAST runs outside of an application, treating applications as a black box. Because DAST tests against a running application, unlike a legacy SAST tool, it is much better at spotting the runtime vulnerabilities that SAST misses. Because DAST treats the application as a black box, it has no insight into the underlying causes of the vulnerabilities it uncovers. Teams must spend time hunting down the cause of vulnerabilities and correlating them with DAST reports. Further, it is exceptionally difficult to provide the right data to automatically invoke an application programming interface (API) correctly due to many applications using custom, nonstandard protocols, and data structures. This translates into numerous false positives—not to mention false negatives.

40% of organizations do not use automated software code review tools during the software development life cycle primarily due to the overwhelming number of false positives they generate.[6]

Perimeter Defenses Lack Context, Generate Volumes of False Positives

AppSec is not only about finding and fixing vulnerabilities in the software development life cycle (SDLC) but also protecting software in production runtime. Historically, organizations have relied on a perimeter defense approach using network security tools such as web application firewalls (WAFs). WAFs monitor network-level application communications and draw a "best-guess boundary" around an application.

But perimeter defense approaches such as WAFs are plagued with false positives—not to mention false negatives. To begin, a WAF identifies every vulnerability that is found in its known signature engine. With upwards of 96% of attacks never finding an application vulnerability, this generates a huge number of false positives that security teams must spend valuable time diagnosing.[7] Second, as Layer 7 traffic analyzers, WAFs lack the application context to understand what the traffic means and how the data will be used. This results in false positives—and false negatives—and excessive tuning that must be overseen by security specialists.[8]

After two decades of using commercial Web Application Firewall (WAF) products, it is time to acknowledge a hard truth: a traditional WAF no longer provides adequate protection for the modern internet user.[9]

Alert Fatigue and the Mistake of Adding More Security Tools

Alert fatigue is a significant challenge for security teams. The problem arises when the number of non-actionable (informational) alerts far outnumber actual incidents that need action. More than 4 out of 10 organizations experience over 10,000 alerts a day.[10] And some organizations see upwards of an average of 174,000 alerts a week.[11] Estimates show that over a one-year time frame, organizations spend over 21,000 hours investigating false positives.[12]

MEASURING THE IMPACT OF FALSE POSITIVES

A security team overwhelmed by alerts can quickly become inefficient—and ineffective. Remediating this number of false positives consumes a huge amount of time and expertise. Security specialists must be able to pinpoint vulnerabilities that actually pose risk and those that do not and then prioritize remediation. But this is difficult to do due to the high decibel of alert noise.

Unable to cope with the endless deluge of alerts, some security teams might turn to specific alert features to stem the stream of alerts. The idea that more tools means better protection is common in the cybersecurity realm. But these quickly multiply into an AppSec tool swamp that is costly and inefficient to manage. Even worse, by focusing on only certain alert categories and not others, this increases the likelihood that serious vulnerabilities will be missed.[13]

The sheer volume of alerts—real and false—can overwhelm a security team. Many teams struggle with how best to manage the alert overload problem, with significant increases in organizations hiring more analysts or turning off security features, underscoring the market challenge in trying to keep up with the volume of alerts.[14] Overworking already overstressed security teams, combined with the cybersecurity skills shortage, compounds the problem. 80% of organizations indicate their security incident teams experienced 10% turnover in the past year. Nearly half (45%) admit to as much as 25% attrition.[15]

> Security teams can spend upwards of 21,000 hours annually diagnosing and resolving false positives.

FALSE POSITIVES LEAD TO SECURITY-DEVELOPMENT FRICTION

For applications in development, developers are the ones who are faced with deciphering all the alert noise. One result is that a significant percentage admits to ignoring security alerts in order to meet release deadlines. For example, 52% of developers admit to cutting back on security measures to meet a business deadline. Operations teams even do so; 62% push back when asked to implement security measures.[16] This increases friction between security teams and developers—DevOps teams meet their deadlines but at the expense of AppSec.

According to a recent survey, 58% of DevOps professionals report relying on five or more observability tools—which include AppSec tools—to identify the root cause of performance issues, which means for every problem, they must sort through thousands of alerts across multiple locations in order to find the answer.[17]

Legacy Static and Dynamic Fail on Accuracy

The evolution of application vulnerability scanning tools has gone about as far as possible. Velocity demands of digital transformation accelerate the pace of application development—in terms of release cycles—to the point where scanning simply cannot keep up. Legacy static and dynamic security scanning tools as well as perimeter defenses for applications in production lack the accuracy needed by modern software.

Resulting false positives are a critical impediment that consume valuable time for both security teams and developers. Manual diagnosis of false positives equates to hundreds—or even thousands—of hours in wasted time. Plus, built on signature engines that identify known threats but miss unknown vulnerabilities, static and dynamic scanning tools generate significant numbers of false negatives that increase application risk.

Frustrated and measured via how much code they write and release, developers are bypassing the alert noise of false positives. While this enables them to meet their major business objectives (MBOs), doing so ratchets up risk and has an adverse effect on security teams and their MBOs. Security and development teams, as a result, often find themselves juxtaposed and constantly butting heads with each other. In response, a different approach to AppSec is needed—one that analyzes applications for vulnerabilities from within the software.

CONTRIBUTORS TO FALSE-POSITIVE ALERT RATES:

- Security configuration errors
- Inaccuracies in legacy detection tools
- Improperly applied security control algorithms
- Multiple security controls fail to correlate event data
- Tools used operate in separate silos

[1] "Ponemon Institute Reveals Security Teams Spend Approximately 25 Percent of Their Time Chasing False Positives," Exabeam, August 1, 2019.
[2] Michael Hill, "Over a Quarter of Security Alerts Are False Positives," Infosecurity Magazine, March 17, 2020.
[3] Cliff Saran, "Agile IT held back by legacy tech and legacy budgeting," ComputerWeekly.com, September 27, 2018.
[4] "2020 Insider Threat Survey Report," Gurucul, November 2019.
[5] "Accurately Assessing AppSec With the OWASP Benchmark Project," Contrast Security, December 2016.
[6] Dan Harrison, "DevSecOps series: Are your security tools giving you too many false positives?" Capgemini, March 28, 2019.
[7] "Contrast Labs' 2020 Vulnerability and Attack Threat Report," Contrast Security, forthcoming.
[8] "Perimeter Security Noise Leaves Applications Vulnerable to Attacks," Contrast Security, April 2020.
[9] Maksim Blishchik, "A Traditional Web Application Firewall (WAF) Is No Longer Enough," G2 Learning Hub, September 30, 2019.
[10] Patrick Spencer, "Accuracy in AppSec Is Critical to Reducing False Positives," Security Boulevard, April 8, 2020.
[11] "The State of SOAR Report, 2018: The second annual state of incident response report," Demisto, 2018.
[12] David Atkinson, "Rationalizing the Security Stack for More Effective Protection," Infosecurity Magazine, August 21, 2019.
[13] "How Manual Application Vulnerability Management Delays Innovation and Increases Business Risk," Contrast Security White Paper, May 2020.
[14] "Research Report: The Impact of Security Alert Overload," CRITICALSTART, 2019.
[15] Ibid.
[16] "52% of Companies Sacrifice Cybersecurity for Speed—Webinar Recap," Threat Stack, March 13, 2018.
[17] Steven Czerwinski, "Fight Alert Fatigue: How to Wake Up Your Alerts Strategy," The New Stack, July 24, 2018.

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