Checkmarx

The world runs on code. We secure it.

WHITEPAPER

Correlation: The Application Security Testing Imperative in Modern Application Development

How to Optimize Application Security with a Holistic View of Vulnerabilities and Risks

Overview

This white paper is designed to help CISOs, application developers, application security (AppSec) teams, and their organizations understand modern application development (MAD), its impact on AppSec and AppSec testing, and how to optimize AppSec testing in this new era. We'll begin with a look at the drivers, benefits, and risks of modern application development. Then we will investigate the challenges of siloed AppSec testing and why an all-in-one platform approach is superior to single best-of-breed products. Lastly, we introduce a way to optimize application security even further through a holistic, correlated view of vulnerabilities and risks that yields prioritized actionable insights.

After reading this white paper, readers will understand why the inherent challenges of vulnerabilities, risks, and exposure through an expanded attack surface continue to rise. This is primarily due to applications being increasingly deployed in the cloud and composed of custom code, microservices, containers, open source code, Infrastructure as Code (IaC), and APIs. You will also learn what organizations can and should do about the challenges of securing the many lines of code that make up modern applications.

Most important, you will learn about the introduction of new innovative technology from Checkmarx that adds a correlation layer to its **Checkmarx One Application Security Platform.** Now available, **Checkmarx Fusion** serves as a powerful tool to identify and prioritize risks within and across componentized application functions in a modern application development environment. This revolutionary context yields actionable insights, visualization, correlation, and prioritization far beyond today's manual or aggregation methods to ensure that organizations never have to worry about releasing vulnerable code again.

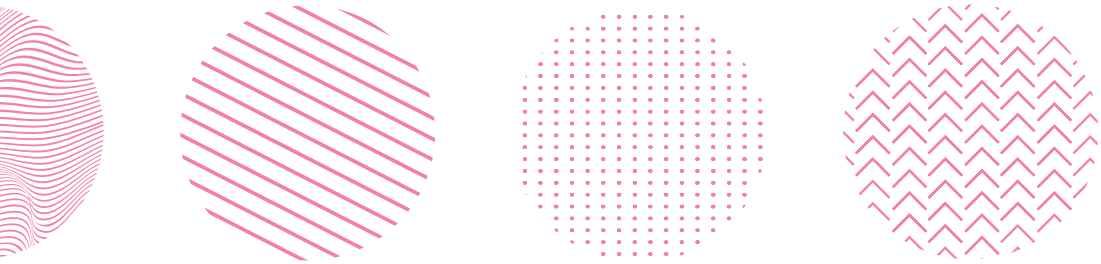


Table of Contents

Introduction

In 2011, when Marc Andreessen wrote that "software is eating the world," little did we realize how true that sentiment would become. Consumers, professionals, and companies have now experienced one of the most significant impacts of digital transformation: the central, uncontested role of software applications in almost every single aspect of work and life.

As a result, increasing demands on businesses to develop smarter and easier-to-use applications that align with fast-changing customer needs and preferences are driving application development, delivery, and deployment at breakneck speeds.

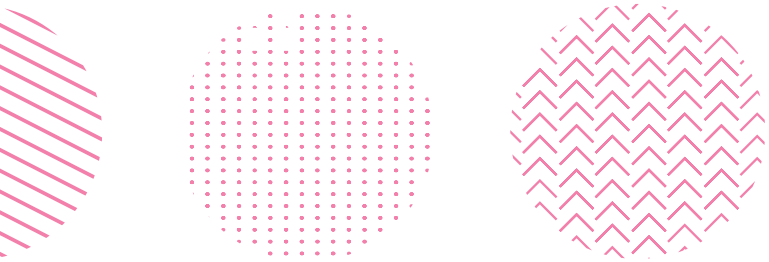
With organizations continuing to move toward shorter development cycles, more frequent releases, DevOps and CI/CD practices, and increasingly complex architectures, the adoption of modern application development (MAD) approaches is critical. And yet modern application development—the very process that enables development teams to accelerate the development and delivery of software solutions—is stretching their ability to keep up with the management of vulnerabilities and security risks within the software development life cycle (SDLC).

In fact, ***survey*** respondents have cited "keeping up" as one of the most prominent challenges in modern application development. In a worldwide survey of more than 1,500 AppSec managers and software developers conducted by Censuswide on behalf of Checkmarx in Q3 of 2021, 54% of respondents said that the shift to the cloud has made them more concerned about secure application development. The biggest challenges they cited were:

- Adopting cloud native security testing (37%)
- Hybrid deployment (36%)
- Upskilling developers (35%)

One of the biggest issues is the inability to address new security risks within each component and across each area—APIs, microservices, open source supply chains, and containers, among others—which impairs an organization's ability to develop secure applications quickly and reliably. That impairment can have a cascading effect on the business, potentially impacting revenue goals, market share, and overall business performance.

To support modern application development environments, AppSec managers and app developers are turning to all-in-one application security testing platforms for one-click scanning, with the ability to correlate results across the entire application, something that hasn't truly been possible until now. But before we look at the imperative and promise of Checkmarx Fusion, let's set the stage by diving more deeply into the modern application development environment itself.



It's Good That It's a MAD World Out There

Imagine developing applications capable of running anywhere and on any commodity infrastructure at scale—on-prem, cloud, hybrid, and multi-cloud. MAD brings this vision into reality through cloud-based native approaches and facilities that disassociate the services software provides from design, development, and deployment practices.

That shift enables teams to operate at greater scale than ever before. Functioning under the notion that **EVERYTHING** is code, MAD assumes nothing is defined by the former operational aspects of past monolithic software builds, in effect, isolating software innovation from operational boundaries. This enables teams to spend more time building innovative features and functions instead of wasting time considering the effects on underlying systems. As such, MAD holds the key to modernization and software-based digital transformation. Because software-reliant organizations (and let's face it, who isn't these days?) can gain tremendous benefits from MAD, it is expected to be adopted by all organizations that develop their own software.

One of the key benefits, says Ori Bendet, Checkmarx VP of Product Management, is the ability for MAD to operate at ridiculous speed. "Because everything is integrated and automated in MAD, delays caused by people are removed or overcome. This radically improves how fast organizations can deliver software at scale. In fact, MAD is key to organizations being able to deliver five years' worth of software development in a single year—that's 400% faster. You can get new builds into production not once a week, a month, or a quarter, but multiple times a day, but a simple mistake can reach production in a matter of minutes. The impact of MAD on development, delivery, and deployment is so enormous that it can be difficult to believe, much less comprehend."

Building Blocks of MAD

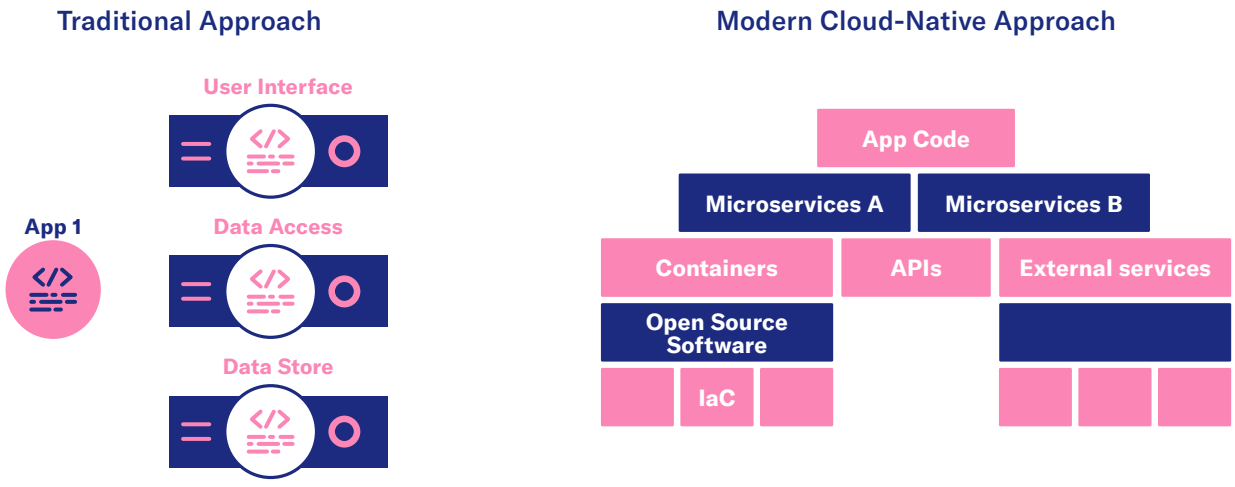


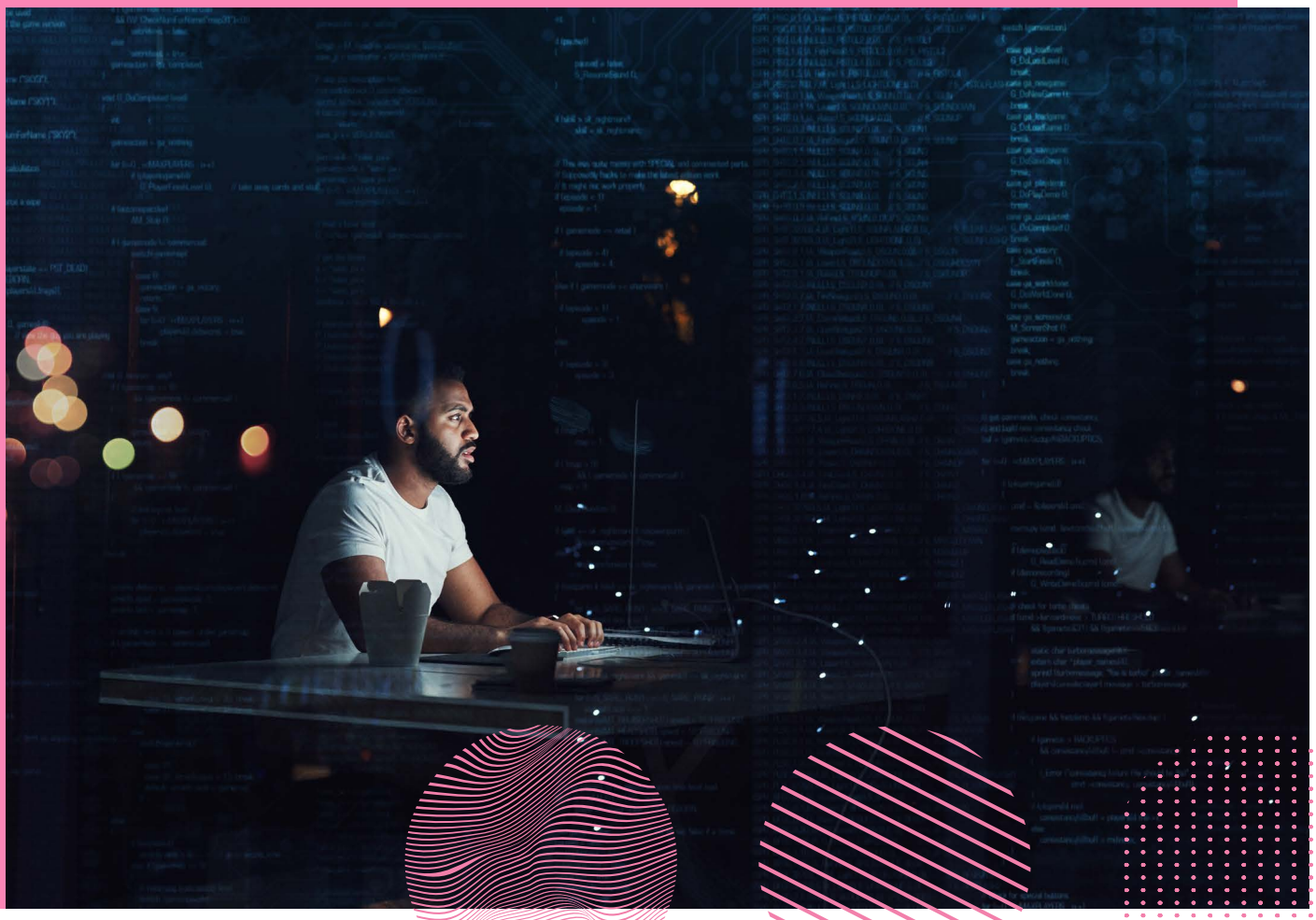
Figure 1: The traditional way of developing apps compared to today's componentized, cloud-native approach in modern application development.

But MAD Is Not Without Its Challenges

But, even with these benefits, this new way of developing software is not without its challenges. As revolutionary as it is, it's still software. Inputs and outputs, networking, storage, databases and so on still exist in MAD outcomes, but as developers become more accustomed to development frameworks, the abstractions increase. For example, instead of a building a data center out of hardware and software, the entire process can now be summarized by calling an infrastructure as a code function. That's great until it breaks or doesn't play well in the sandbox with other functions, when you may not know how to quickly fix it, since you may not fully comprehend how it worked in the past.

Other challenges abound: with MAD, organizations must address how they scale, as it can take longer to provision, support, fine-tune, and troubleshoot in MAD environments. **Observability** is another issue— with hundreds of components, like microservices, running to support a single app—it's imperative to observe them all to ensure they're working properly, no matter where or how they are deployed.

Lastly, MAD will not reduce or eliminate many of the tools already in place to develop or secure code— just the opposite. "However, **tool sprawl** is being addressed and analysts now predict that 70% of companies will consolidate into 2-3 vendors by 2025," Bendet explains.



The Impact of MAD on Application Security

In addition to the challenges cited above, MAD introduces a new set of general risks and specific security risks. For example:

- **Componentization** – Because MAD incorporates hundreds of piece-parts stitched together, it introduces new risks related to the tech stack, architecture, processes, and vulnerabilities.
- **Shifting Left of Security** – With MAD, the responsibility for detecting vulnerabilities early in the SDLC shifts to developers as a part of ongoing development and continuous integration.
- **Continuous Security Assessment** – This is now an integral part of software development and not a separate phase anymore, where we're scanning code once in a while during nightly builds.
- **Additional Security Risks** – The risk landscape expands through the introduction of additional security challenges for every component within a MAD initiative. For example:

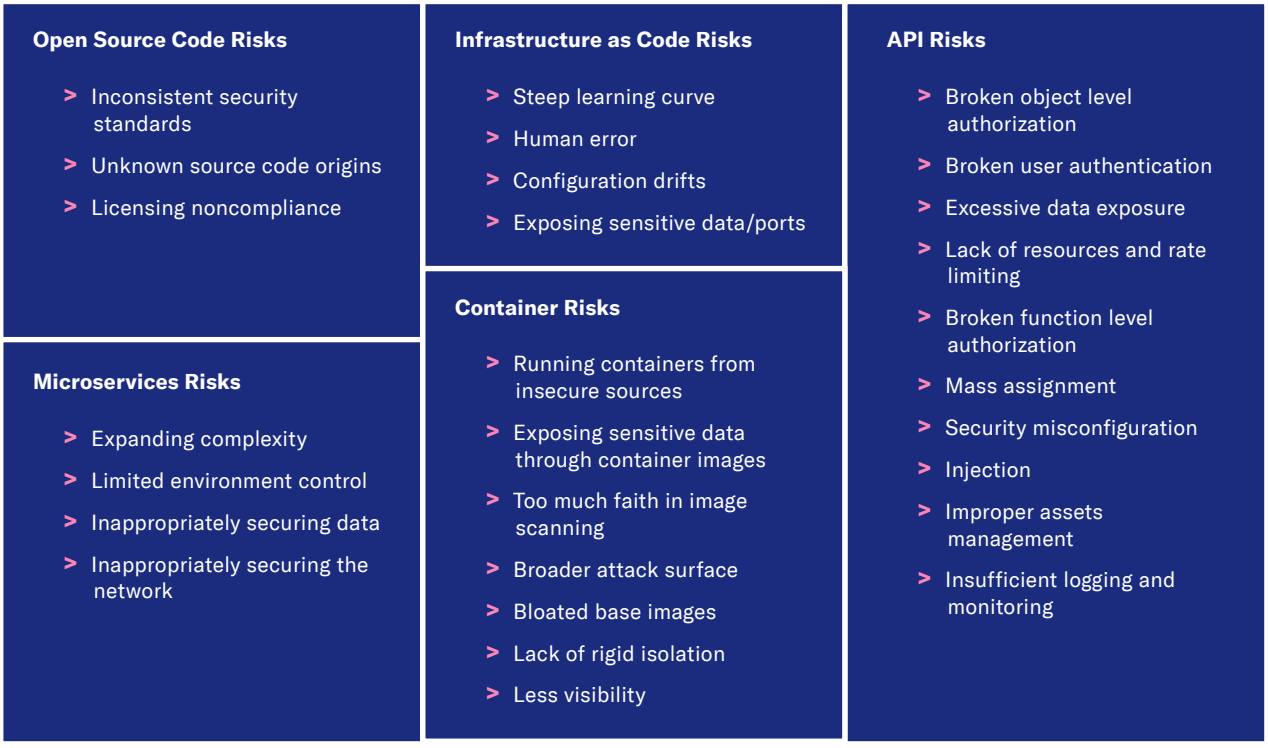


Figure 2: Some of the new security risks involved in modern application development.

The bottom line is that with applications both increasingly deployed in the cloud and composed of custom code, microservices, containers, open source code, and Infrastructure as Code (IaC), the inherent challenges of vulnerabilities, risks, and exposure through an expanded attack surface continue to rise. When asked what areas should be prioritized in AppSec initiatives, it's no surprise then that respondents to the Checkmarx Q3 survey clearly aligned with most of the areas of risk posed above. Their top responses were:

- Containers (31%)
- Serverless technologies (32%)
- Infrastructure as code (28%)
- Hybrid cloud (28%)

Challenges of Siloed Application Security Testing in a MAD World

Unlike monolithic applications of the past, where coding was all in the same language, an average cloud-native application can have anything between 50-5000 different components, any of which can be rife with vulnerabilities that present an attack surface to a malicious hacker. Given that volume and complexity, it's important that modern application development take a different approach to application security testing.

Existing approaches to application security testing, such as Software Composition Analysis (SCA), Static Application Security Testing (SAST), Dynamic Application Security Testing (DAST), Interactive Application Security Testing (IAST), and IaC scanning, look at just one area of the stack and have proven to be less effective in a MAD world due to their siloed nature. That's for a couple of different reasons:

- **Not Integrated:** By themselves, individual software components may be deemed free of vulnerabilities, but once you start to connect the components, new vulnerabilities and risks may be exposed—which single scanners cannot detect. Even though companies may purchase best-of-breed scanners, they individually do not deliver a holistic view and full context of potential vulnerabilities or risks across the MAD environment, which can cause problems.

  Daniela Da Cruz, Checkmarx VP of SAST and Engines Engineering, provides a common scenario to illustrate: "As a developer working on a specific component, I might not even be aware of the components other teams are using. For example, one team might create a custom sanitizer, but when another team doesn't know about it and runs a scan, they wonder why there are so many vulnerabilities, and it's because they haven't accounted for the custom sanitizations the other team has developed. It's only when I can see everything in one place that I can understand the common vulnerabilities and whether they are permissible."

- **Different Timelines:** Each tool type searches for specific vulnerabilities and as such, is deployed in different stages of the SDLC, with different results. Because SAST, which analyzes an applications source code, is used in the programming and testing phases within the SDLC, it can produce many false positives because it does not have the context of the runtime. DAST is used during in staging or production when apps are running, but unfortunately, scanning at the end of development can uncover a multitude of vulnerabilities that have very little time to be fixed.

Because of the overwhelm of false positives, last-minute discovery of vulnerabilities, and lack of correlated data, developers and AppSec teams find themselves in the unfortunate position of manually aggregating and triaging vast amounts of often redundant and uncorrelated findings.

In the end, unable to confidently triage and remediate vulnerabilities and lacking a holistic view of the application security posture, organizations are left with the no-win decision of holding up the release cycle or releasing insecure software—jeopardizing not only their own business, but the businesses of clients and customer data as well.

A Better Way: An Integrated AST Platform Powered by Correlation

With software at the heart of digital transformation, ensuring it remains secure from a developer's first code commit through the push to production is essential. With the limited effectiveness of siloed solutions, AppSec managers are turning to all in-one application security testing platforms placed as close as possible to developers. Ideally, such a platform should scan all the various functions across the modern application environment, including app code, containers, microservices, open source software, APIs, software supply chain, and more with a correlation layer to analyze results across the environment.

What Application Security Testing Approach is Required?

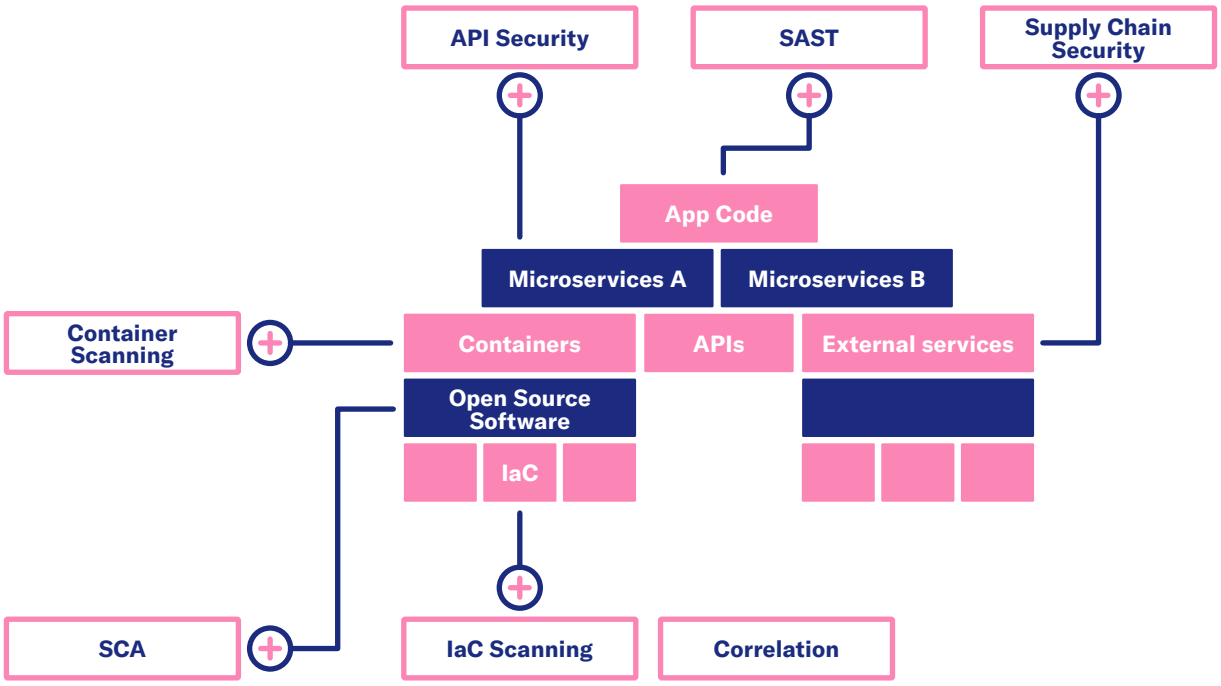


Figure 3: A comprehensive, integrated approach to AST in a MAD world.

When you're looking for AST solutions that fit well into your MAD initiatives, you need to be sure you're enabling digital transformation, not hindering it. Here are seven features we believe modern AST platforms must deliver:

- Be built for cloud development and fully understand modern tech stacks, architectures, processes, and vulnerabilities.
- Be capable of integrating with developer workflows, automating scans across the SDLC, and correlating results to pinpoint risks using unified dashboards and broad reporting capabilities.
- Provide one-click cloud-based scans for a single stakeholder using one process, from one platform, with no installations and no scanning servers required:

What Application Security Testing Approach is Required?

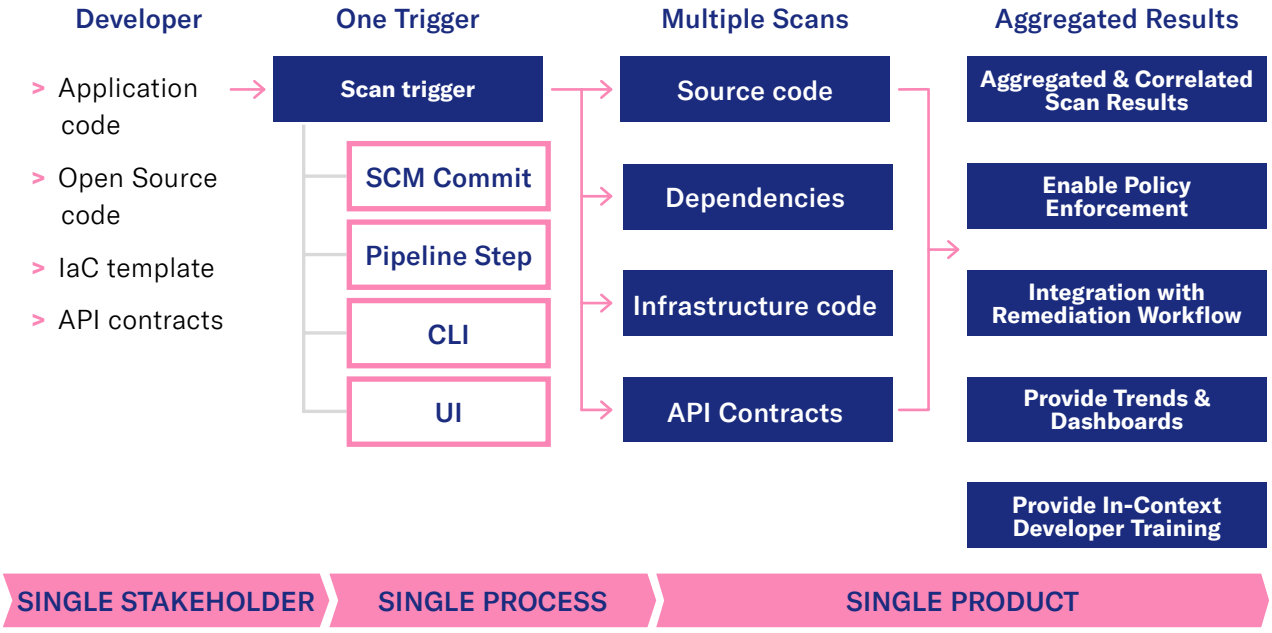


Figure 4: One-click scanning provides an efficient, comprehensive way to understand vulnerabilities and risks across the MAD environment.

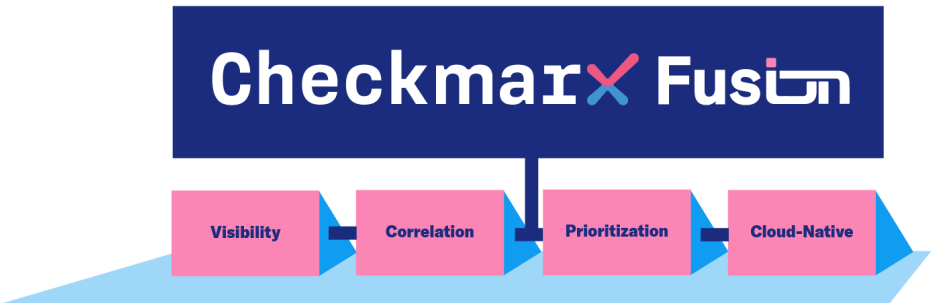
- Include SAST, SCA, container security, IaC security, API security, supply chain security, an orchestration layer, a correlation layer, advanced reporting, and developer training.
- Deliver automated scans with various scan engines and correlate results across your codebase, providing complete and accurate results through a platform-like approach.
- Come with standard support, options for premium support, unlimited scans, concurrent scans, incremental scans, customization, plugins, and add-ons.
- Allow flexible deployments, with identical capacities whether used on-premises, in the cloud, or in hybrid environments.

Most importantly, the availability of a correlation layer can increase the level of confidence in and prioritization of high-risk findings (vulnerabilities detected) from AST scans, especially when organizations have hundreds of applications, and their AST solutions are detecting thousands of potential vulnerabilities. The ability to correlate the same findings from different scanning solutions minimizes false positives, shows where additional unforeseen risk is introduced within the environment, and which risks should be addressed first. Now able to make sense of the large amounts of data from their scan findings, organizations can scale their operations to meet business goals more quickly and easily. That's the power of correlation.

Introducing Checkmarx Fusion™

Recognizing the inefficiency of manual correlation and the deficiency of alternate solutions that merely aggregate results, Checkmarx has developed Fusion, to provide advanced correlation in MAD environments.

Checkmarx Fusion aggregates and correlates findings from the multiple AppSec engines within the Checkmarx One Platform™ to understanding the real impact of existing vulnerabilities and risks that cannot be found by single engines on their own, providing meaningful, prioritized, and actionable insights. This new technology works across the application development and deployment spectrum and across multiple application services and repositories through static and dynamic analysis. By analyzing the holistic impact of all vulnerabilities, Checkmarx Fusion delivers:



- **Visibility**: Provides threat modeling by mapping threats in a visual intuitive graph that contains all software elements, consumed cloud resources, and the relationships between them.
- **Correlation**: Provides context to the siloed scanners by combining and correlating results from static code scans and runtime scans, which helps to eliminate false positives.
- **Prioritization**: Focuses teams on solving the most critical issues that matter most to their business by prioritizing vulnerabilities according to their real risk and potential impact.
- **Cloud-Native**: Covers cloud-native architecture including microservices, cloud resources, containers, and APIs while correlating insights from pre-deployment to runtime.

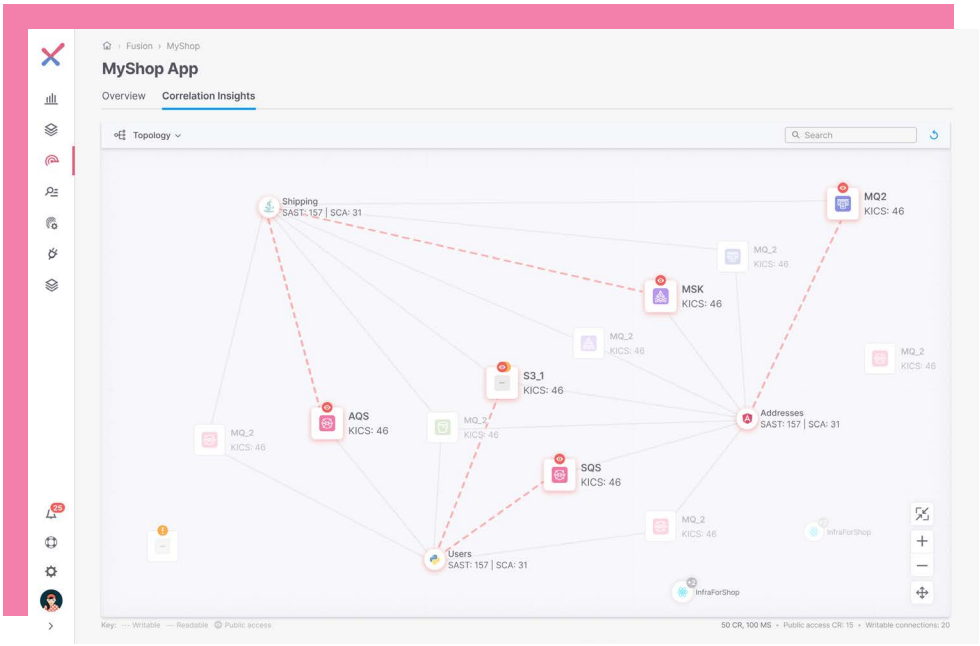


Figure 5: Checkmarx Fusion: Topology View

Checkmarx Fusion maps the results of different engines based on common usage of native cloud inputs and services. Here are just a few examples of correlated findings:

- SAST & KICS* : Sensitive data written to a public S3 bucket.
- SAST & KICS & API Security: API/endpoint found by SAST with missing security configuration found by KICS and enriched with additional information about the same endpoint from API Security.
- SCA & KICS: CVE of code injection that is exploitable only when a specific port is opened.
- SAST & SAST: One microservice sends sensitive data to another microservice, which saves it to a plain text file.

Mark Mishaev, Checkmarx R&D Chief Architect, explains the benefits: “Checkmarx Fusion brings everything together so that all scan results are correlated into a single pane of glass—not just aggregated like alternate solutions. This delivers multiple benefits. First, it helps minimize false positives, because we can now check findings against other engines to verify whether they are real or not. Second, it helps to find results not found by tools independently. By correlating results from different security engines, we’re able to detect vulnerabilities on both the application and deployment sides. Third, we’re able to see trends across different security aspects, such as how long teams take to remediate and whether they’re repeating the same coding mistakes. Fourth, it delivers actionable insights to prioritize findings based on context; for example, proposing the best fix location. All of this is critical for developers and AppSec teams to release secure software.”

Checkmarx Fusion in Action: Main Use Cases

- **Understanding an Application’s Bill of Materials**

  Based on SAST and IaC (KICS) findings, Checkmarx Fusion will provide a Bill of Materials (BoM) of cloud-native application resources such as storage, queuing services, serverless, and databases and will categorize each resource’s services use as public or private. Checkmarx provides a list of all the assets in the cloud resources as well as microservices that use the cloud resource through a textual and visual Bill of Materials.

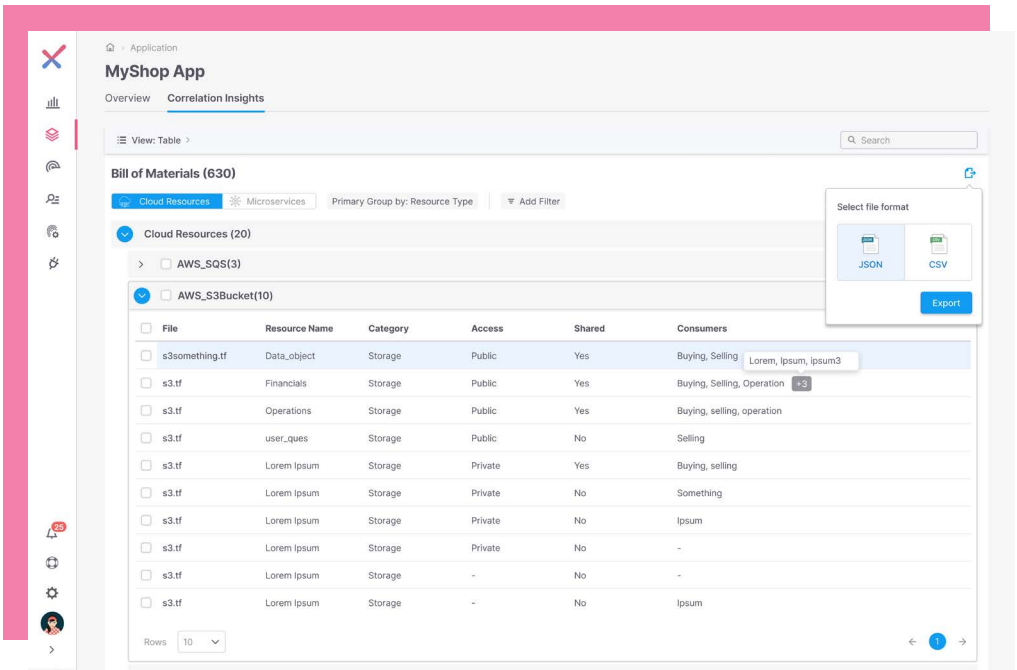


Figure 6: Bill of Materials for cloud resources displayed in the Checkmarx Correlation Engine.

*KICS (Keeping Infrastructure as Code Secure) by Checkmarx is a free, open source solution for static code analysis of IaC.

"Correlating the Bill of Materials in a visual way can help you understand the impact, where there are the most findings and vulnerabilities, and help to prioritize analysis," says Miki Sharon, Checkmarx Senior Product Manager. "Just aggregating the results won't give you that value. You need to go the extra step and actually connect the results."

> Prioritizing SAST Results

Another use case of Checkmarx Fusion is to prioritize SAST results based on whether the cloud resource accessibility is public or private. Miki Sharon explains: "If we had an SQL injection correlated to a privately accessible cloud resource, then it's not an issue like when an SQL injection is correlated to a publicly accessible cloud resource. But the minute you run an IaC scan and correlate the results through Checkmarx Fusion with your SAST findings, you realize it's related to a publicly accessible cloud resource and that it **is** an issue. The correlation helps to weed out false positives, find the real risks, and prioritize the severity of the findings to focus the customer on remediating them. Something like this just hasn't been possible before."

Additional use cases include highlighting shared resources used by multiple microservices in native cloud applications and prioritizing SAST results based on API security information, such as whether an API has private or public accessibility.

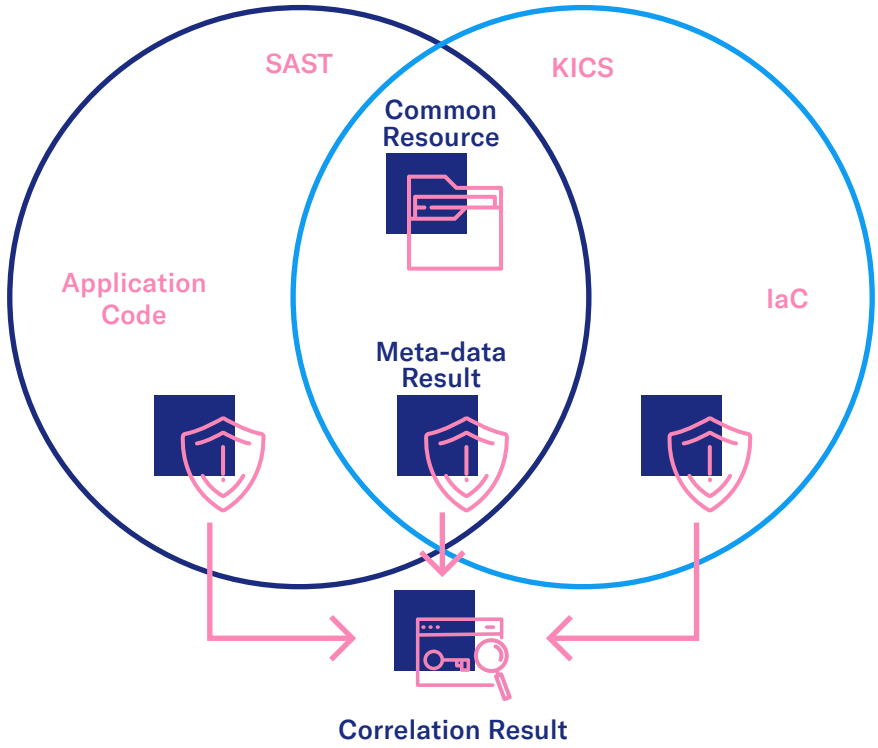


Figure 7: Checkmarx Correlation Engine aggregates and correlates the results of different scanners across the MAD environment.

Daniela Da Cruz notes that, "With the Checkmarx One Platform, we now have a platform that runs multiple engines and technologies under one location. The correlation that sits on top of everything delivers streamlined results from any relevant AST scan to provide additional value to customers. This is a market-changing engine that aggregates **AND** correlates results into a single view—which no one else in the AppSec industry does. The correlation engine will provide advanced AppSec and other insights on customers' cloud-native applications through a visual topology view and improved prioritization."

About the Checkmarx One Application Security Platform™

Now unified through an industry-leading correlation engine, Checkmarx One Platform delivers all the essential application security services from a unified platform. In one scan, it analyzes source code, open source dependencies, and IaC templates; aggregates, correlates, and verifies the results; and augments them with expert remediation advice.

Designed for cloud development and delivered from the cloud, it seamlessly secures your entire development life cycle, and eliminates the overhead of managing the infrastructure while providing continuous service updates as well as functionality enhancements. With the platform, you can trigger a comprehensive code scan from a single event, like a click in the UI or a commit to the source repo.

Checkmarx One Application Security Platform

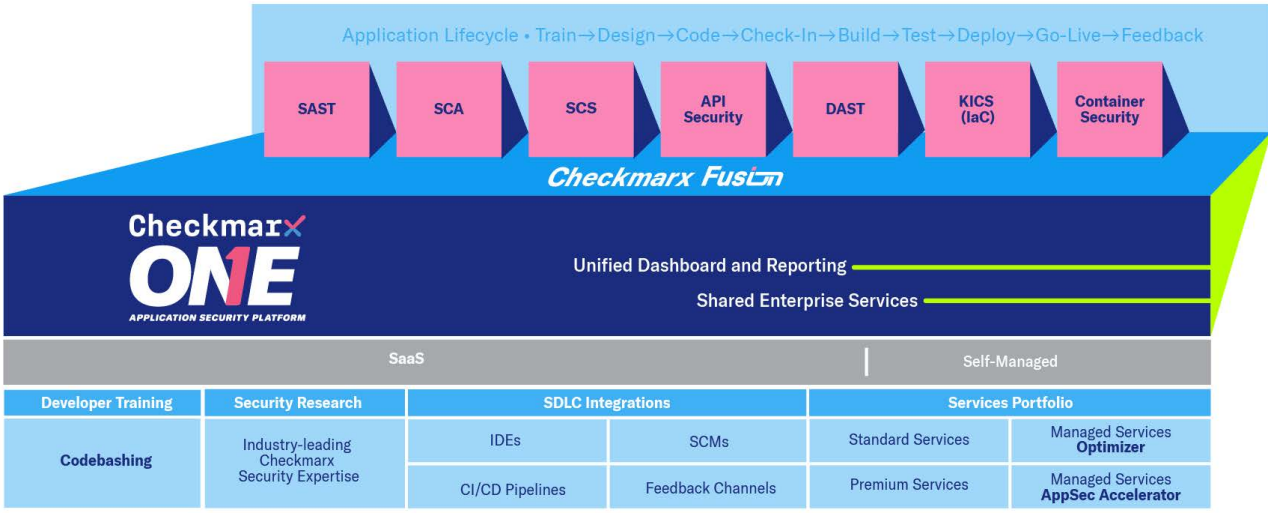


Figure 8: Checkmarx One Platform delivers the widest coverage and best accuracy in the industry.

Let's look at what's included:

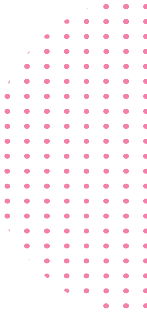
- **Checkmarx SAST™** is an enterprise-grade, flexible, and accurate static code analysis solution that identifies security vulnerabilities in custom code. It allows development, DevOps, and security teams to scan source code earlier in the SDLC, identify vulnerabilities, and provide actionable insights to remediate them sooner.
- **Checkmarx SCA™** leverages our continuously updated open source vulnerability database to empower development, security, and operations teams to find and mitigate security risks from open source code, libraries, and licenses within the software supply chain.
- **KICS by Checkmarx** scans IaC to find security vulnerabilities, compliance issues, and infrastructure misconfigurations. With more than 2,000 predefined queries, KICS can help you quickly find IaC security issues before they make it to deployment.

- Working in concert with Checkmarx SCA, Checkmarx **Supply Chain Security** identifies suspicious and potentially malicious open source packages across the modern application development lifecycle. The result is full-spectrum software supply chain insight and analysis that closes a significant gap in organization's application security.
- **Checkmarx SCA** also covers **containers** in its scan of open source components. Checkmarx SCA combines advanced technology and a dedicated open source research team to produce fewer but more relevant results, while our Exploitable Path feature identifies which vulnerabilities are actually exploitable, helping you prioritize remediation of your real risks.
- **Checkmarx API Security** via SAST enables you to secure APIs against vulnerabilities and any exposed application logic and sensitive data.

Because the Checkmarx One Platform brings together critical services and integrated, proven technology, you benefit from:

- **Correlated scans**: Trigger multiple scan types from a single event and aggregate and correlate the results for a complete, more accurate picture of your code security.
- **Faster time to value**: Kick-start your AST program in hours, not days, with fast onboarding, simple configuration, and automated scan tuning.
- **Speed and scalability**: Leverage secure, cloud-powered scanning at whatever capacity you demand, with no need to manage scanning infrastructure.
- **Lower friction and overhead**: Integrate the platform into your existing software build pipelines and feedback systems, instead of training your teams on whole new toolsets.
- **Wide technology coverage**: Know you're covered across your development portfolio with support for 30+ languages, most popular package managers, and a growing list of IaC and API templates.
- **More accurate findings** through cross-correlation of different engines, minimizing false negatives and false positives.
- An **all-in-one platform that's easy-to-use**, with a fast time to value.





Final Thoughts

We've seen the challenges and benefits of the modern application development environment and looked at how today's approach to application security relies on resource-intensive testing, triage, and vulnerability remediation performed late in the software development life cycle. This results in a backlog of vulnerabilities with too little time to fix them before they're released into production. Moreover, these mechanisms often create too many false positives, overwhelming dev and AppSec teams.

We've also established that organizations face an ever-growing challenge of securing applications and that existing approaches to application security that address specific facets, such as static code analysis, IaC scanning, or software composition analysis prove to be less effective because of their siloed nature.

But, because all of the application security testing scanners are already integrated within the Checkmarx One Platform, we're able to correlate results, which is not possible through external scanners. Checkmarx Fusion combines findings from various security engines such as SAST, KICS, and others to create a comprehensive, holistic, integrated view of application security posture to drive better, faster outcomes integral to DevSecOps and SDLC processes.

The result? Developers can finally leverage security to accelerate their time to market, CISOs reduce risk across the board, and AppSec teams have better visibility into what does and doesn't need to be fixed. "This is light years ahead of what organizations are doing now," concludes Ori Bendet. "We believe this will disrupt the way things have been done in the past, because nothing like Checkmarx Fusion has ever existed. It truly takes AppSec testing in the world of modern application development to a whole new level."

Next Steps

To learn more about Checkmarx's award-winning Application Security Platform and its newly announced Correlation Engine known as Checkmarx Fusion, please visit the following links:

- Learn about Checkmarx's industry-leading *Application Security Platform*.
- Read more about *Checkmarx Fusion*
- *Request a Demo* to see first-hand how powerful it is to scan, correlate, prioritize, and remediate security risks across the entire modern application development environment.

About Checkmarx

Checkmarx is constantly pushing the boundaries of Application Security Testing to make security seamless and simple for the world's developers while giving CISOs the confidence and control they need. As the AppSec testing leader, we provide the industry's most comprehensive solutions, giving development and security teams unparalleled accuracy, coverage, visibility, and guidance to reduce risk across all components of modern software – including proprietary code, open source, APIs, and Infrastructure as code. Over 1,675 customers, including 45% of the Fortune 50, trust our security technology, expert research, and global services to securely optimize development at speed and scale. For more information, visit our website, check out our blog, or follow us on LinkedIn.

Checkmarx

Checkmarx at a Glance

1,675+
Customers in 70 countries

750
Employees in 25 countries

45%
of the Fortune 50 are customers

30+
Languages & frameworks

500k+
KICS downloads in 2021

The world runs on code. We secure it.