Latio

Application Security Market Report

2026

Table of Contents

Executive Summary

Whether you call it application security, product security, or DevSecOps, securing software is complicated. Today, practitioners are expected to manage a growing set of scanners, reduce large vulnerability backlogs, coordinate remediation across teams, and participate in architecture and threat modeling, often with limited headcount and little tolerance for noise.

AI is adding to this complexity, amplifying both the risks and opportunities in application security. AI assisted coding is reshaping how applications are built, deployed, and maintained. In parallel, the capabilities of platforms themselves are evolving with AI: features from autofix workflows, to false positive analysis, to scanning itself, are all radically changing product expectations.

This report is designed to help practitioners and buyers navigate the current application security landscape. It covers the transitions in application security over time, from waterfall development to DevOps to emerging AI code generation workflows. The report then breaks down every subcategory of scanner, the development of modern features, as well as how AI capabilities are changing functionalities we use today. We conclude with actionable buyer guidance that spans across SMB, mid-market, and enterprise environments.

Key Takeaways

- **Application security has largely consolidated into platform players.** The capability differences have more to do with user, integration and developer experiences than pure scanning functionalities.
- **AI-native static analysis and business logic detection are the most immediately meaningful changes in Application Security detection capabilities.** These new scanners are capable of detecting entirely new categories of vulnerabilities which have traditionally been reserved for manual review.
- **Application security evaluations should focus on usability and backlog reduction more than specific scanner functionalities.** Tool evaluations should be guided by the time to fix an issue, rather than the number of issues detected.
- **ASPM as "management without scanning" has largely collapsed into broader vulnerability management and exposure programs.** ASPM is shifting into continuous threat exposure management, or universal vulnerability management.
- **Securing AI-generated code is still an open market with unclear best practices.** General approaches involve giving organizational context to agents, and having secure code reviews in pipelines, but this field is rapidly changing.
- **Supply chain security is expanding towards malware, package health, and secure-by-default consumption patterns.** CVE detection alone is not enough for modern supply chain security.

2026 APPLICATION SECURITY SURVEY RESULTS

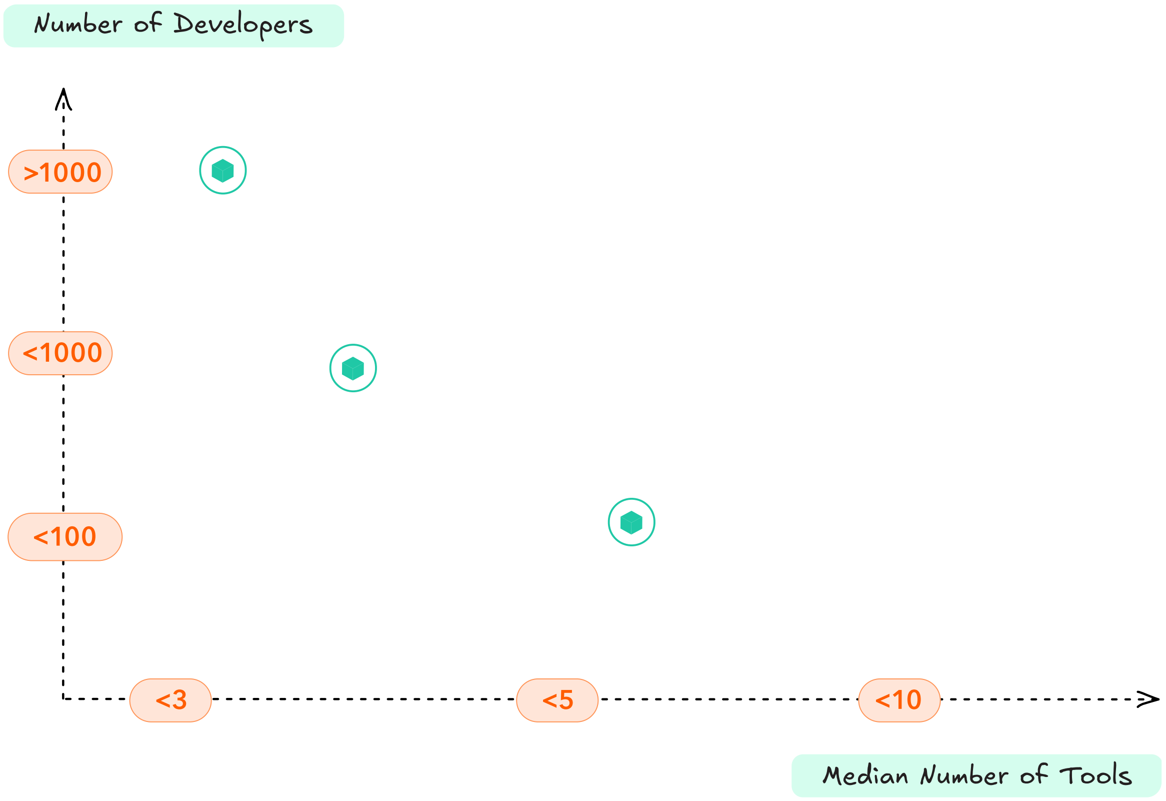
Survey Results

The application security survey spanned organizations from tens of developers to thousands. In many ways, they reflected the reality of application security being a discipline in crisis - as teams were split in their priorities, what they wanted out of a tool, and where they saw the industry going.

The key results were:

- Developer experience is the most important deciding factor for new tools, followed by false positive rates
- AI pentesting is the most desired emerging capability
- The top concern for 2026 is speed and security of AI generated code

Enterprise Doesn't Always Use More Tools



Contrary to popular belief, the survey data indicated that smaller teams can actually use more tools than enterprises, largely because they lean more heavily into open source offerings that they orchestrate themselves. Enterprises, by contrast, tend to manage fewer tools overall, but at a much larger scale. They are also more likely to rely on a smaller set of paid tools for specific purposes, such as supply chain security, SAST, and DAST.

This data helps explain why ASPM, as a standalone management category, struggled to survive. Enterprises with distributed scanning tools and capabilities never intended to centralize everything under a single ASPM layer, and third-party integrations often served as a stopgap rather than a durable, long-term strategy for tool orchestration. As we argued in our Cloud report, ASPM is now evolving into the broader CTEM category, which approaches vulnerability management in a more holistic way.

Developer Experience Matters

Ranked choice voting was completed for several key application security features. The results are listed below in order of how practitioners voted.

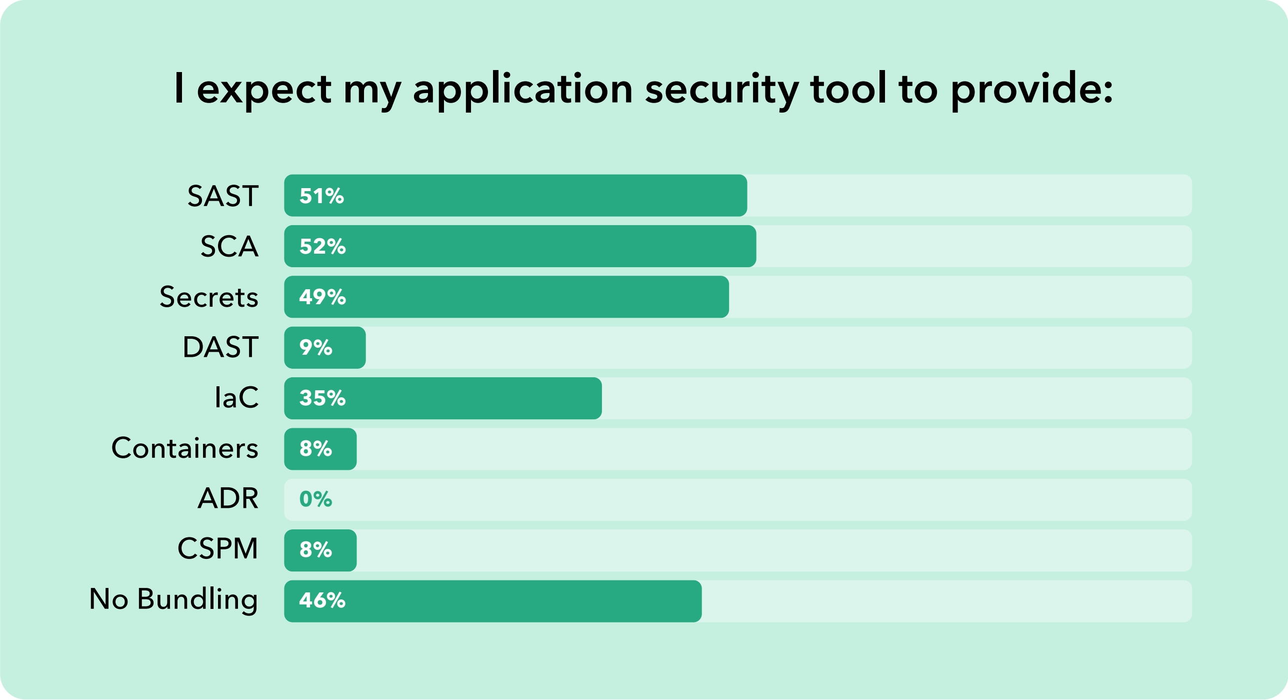
The top 6 application security features:

1. Developer Experience
2. Least False Positives
3. Integrations
4. Most True Positives
5. Remediation Guidance
6. Reporting

The survey results are clear: practitioners are looking for tools that create the least friction with their development teams. Poor developer experiences and high false positive rates are what create friction with other teams, and are the top priorities teams have when assessing tools. This is where user experience matters more than individual scanner finding quality.

Ultimately, practitioners want their security tools to feel invisible to the developer, having them only be helpful nudges in the right direction rather than trying to find as many potential issues as possible. This is why reachability analysis has become a critical component of application security tools.

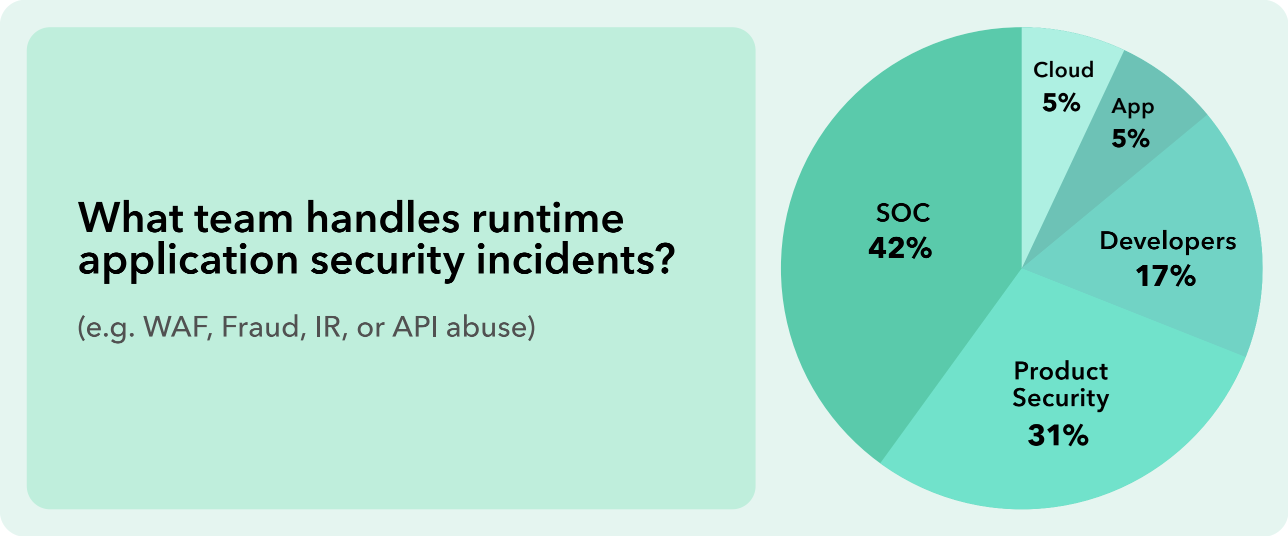
Core Application Security is SCA + SAST + Secrets



As much as security platforms, finance teams, and analysts alike see the value of bundled application security offerings, the majority of practitioners still view only SCA, SAST, and Secrets as part of a core application security offering. The other half of practitioners sometimes included IaC scanning, DAST, and containers, or preferred to have no bundling at all.

While there are clear benefits from implementation and budgeting perspectives for having an all-in-one solution, the core application security stack remain SAST, SCA, and Secrets scanning.

The SOC is Handling Runtime and WAF

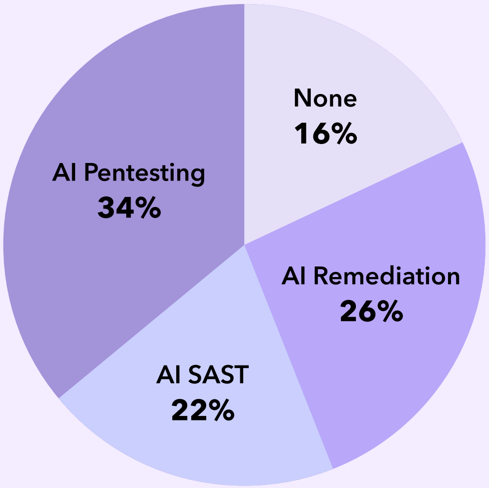


Increasingly, security operations teams are being tasked with handling runtime application security alerts, though ownership remains mixed. Cloud and product security teams are also responsible for these alerts in many organizations, and in some cases developers are involved as well. It is worth highlighting that these management functions no longer sit with traditional network security teams, and have instead shifted toward broader security operations and cloud teams.

This speaks to the growing importance of application-layer visibility for security operations teams, as they are increasingly expected to handle more contextual and technical alerts.

The AI Features That Matter

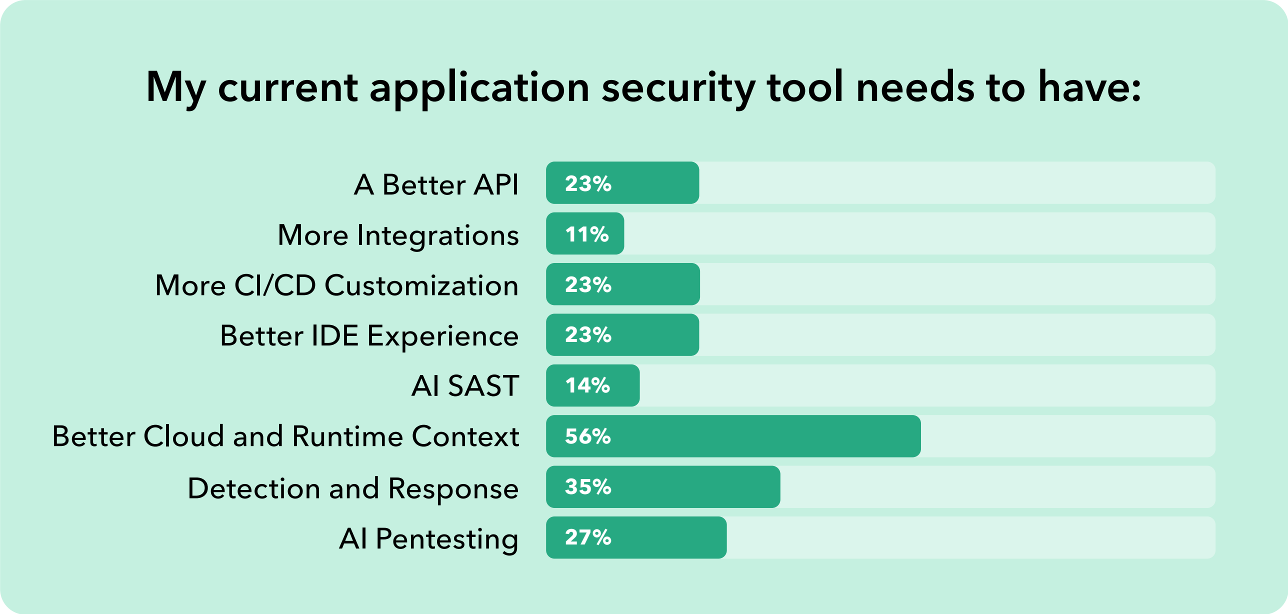
Which AI feature are you the most excited about?



Application security practitioners are ready to adopt the latest AI features - from prioritization to static code analysis. The relatively low number of “none” responses is surprising, because it suggests that practitioners in this area are less skeptical about AI’s ability to transform day to day workflows than in other categories. Seeing the power of AI code generation directly makes practitioners in this category more open to AI adoption.

While most AI features garnered similar levels of excitement, AI pentesting pulled a slight lead, indicating that teams are excited about being able to make continuous pentesting a reality. Overall it's clear that AI is transforming the capabilities of existing testing methodologies and teams are ready for it.

The Gap is Runtime Context

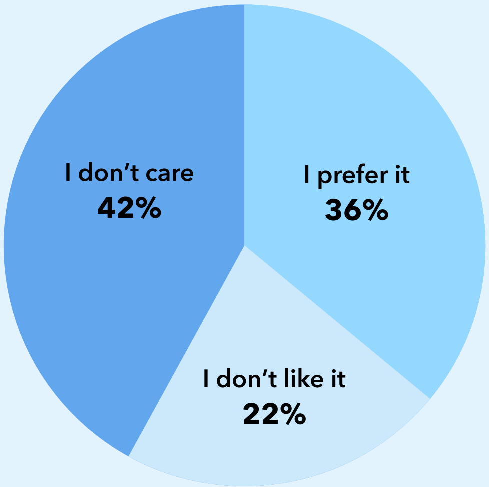

My current application security tool needs to have:

AI may be driving industry excitement, but usability remains the primary focus for immediate feature enhancements. Specifically, better APIs, IDE state tracking, and integration experiences were highly requested from practitioners. The most requested feature by far was better cloud and runtime context, because teams want to better prioritize and determine the truth of a particular finding.

When reviewing the results alongside the preference to separate runtime and code experiences, this finding highlights the need for strong integration between tools managed by different teams to support vulnerability triage, false positive analysis, and delivering fixes.

Open Source or Not - Results Matter More

How do you feel about your vendor using open source scanners?

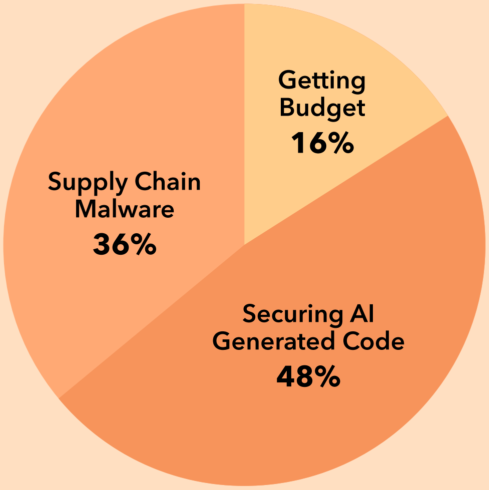


In 2026, practitioners care about end results more than a vendor's underlying scanning engine. Historically, application security practitioners were split about the value of open source scanning engines. About half of practitioners thought they were better than closed source, because passionate developers for every coding language could add to the rule sets, keeping them modern and useful. On the other hand, many other practitioners committed to closed source engines, counting on vendors to do a more thorough job than open source communities.

In the time since, practitioners have come to better appreciate open source software, as well as understand that the underlying scanning engine doesn’t matter as much as the rules that are being applied to it. Most of the time, having customized and editable scanning rules matters more than the engine that is searching the code for the patterns.

Application Security Priorities

What’s your top 2026 application security concern?

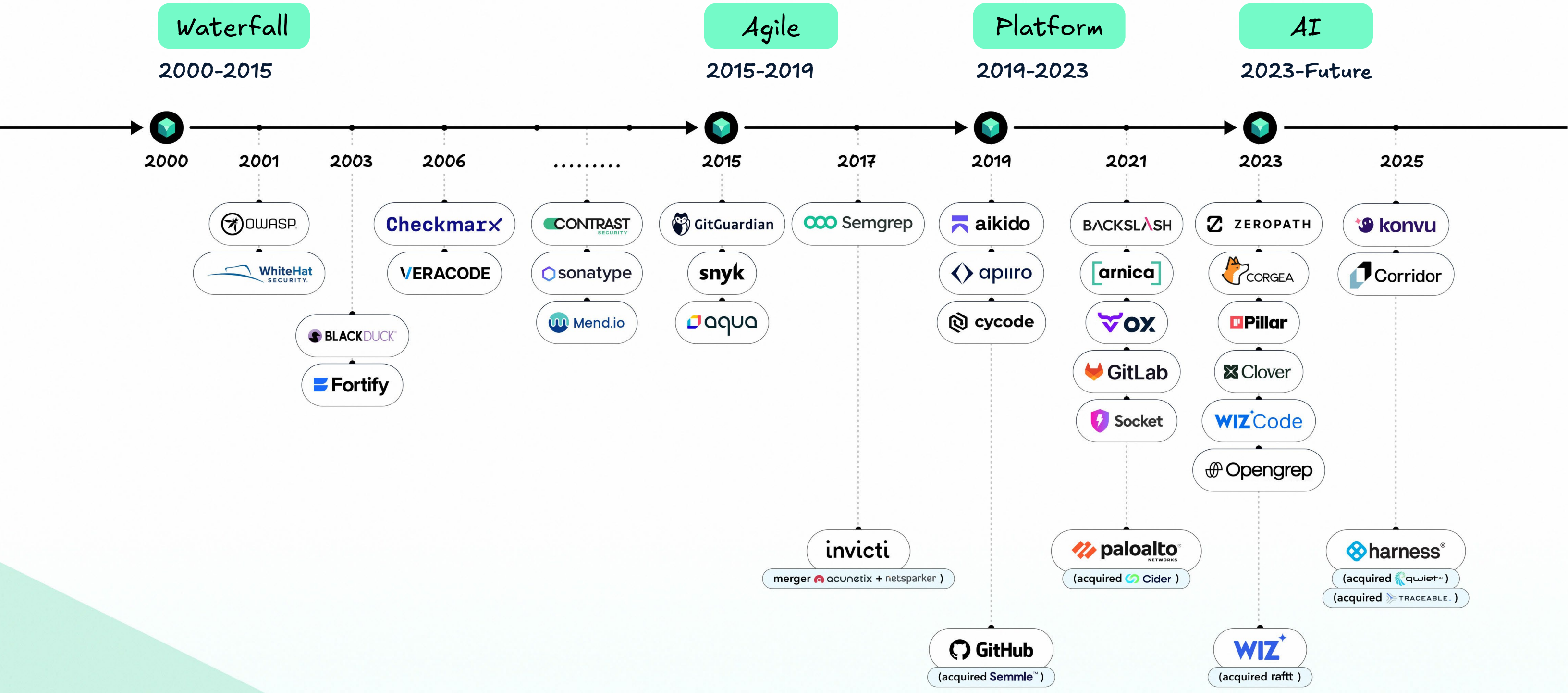


Three concerns stood out most for 2026 priorities: Securing AI generated code, supply chain malware, and getting budget. With a combined 84% of responses, AI generated code and supply chain malware remain top of mind for security teams going into next year. These categories have increased in priority in light of a steady stream of supply chain attacks, combined with the rapid adoption of AI coding tools.

While teams recognize the need to move quickly, they increasingly require tools that enable them to trust the results they’re seeing.

APPLICATION SECURITY TIMELINE

Application security can be divided into four eras: waterfall, agile, platform, and AI. Each era was introduced to address distinct concerns, many of which persist today, even as application security has rapidly matured over the past decade.



Early Application Security

Before 2010, application security was a very different practice in terms of scope, market size, and day to day operations. For the most part, before the proliferation of cloud, DevOps, and B2B SaaS, application security was meaningfully practiced only by the largest software distributors. Companies like Microsoft, Cisco, and IBM were the largest consumers and builders of application security services, as they had real financial obligations to protect the consumers of their software. During this period, OWASP served as a volunteer-led consortium, guiding and developing best practices in real-time.

One primary challenge was developing scaling scanning solutions, primarily for static languages like C and Java. Early solutions were built to support waterfall development workflows, where annual software updates were compiled, uploaded to a platform, and a multi-hour (or multi-day) scan would be kicked off. Then, security teams would work with developers to remediate any discoveries before releasing major software versions.

While the workflow is not as common, the capabilities of these older scanners still serve a key function for enterprises supporting legacy software. Older scanning engines also tend to be more mature, offering styles of analysis and customization that many newer companies have ignored in favor of targeting more cloud native development languages and deployment styles. Especially for static languages, these years of research and development can’t be cloned overnight.

The Shift to Agile

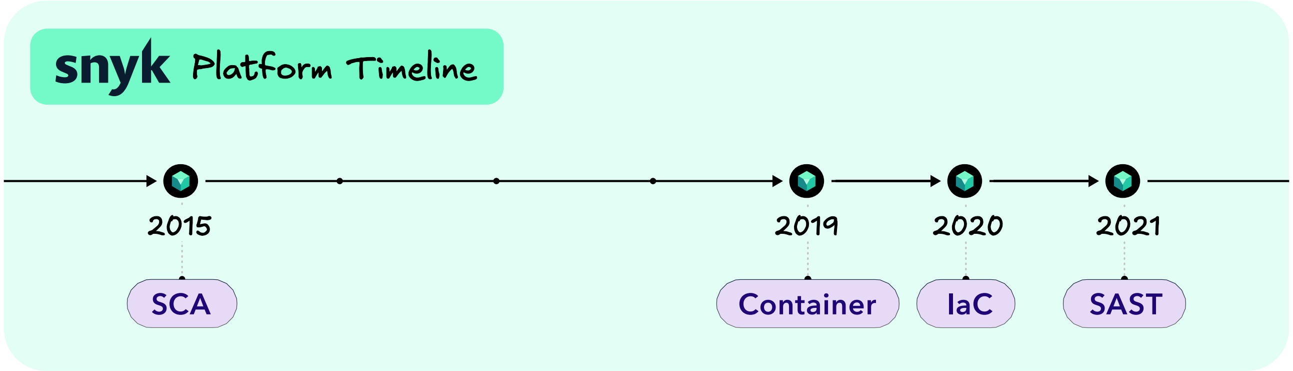
The 2010s saw a massive change in how software was delivered:

- Dynamic languages like Javascript and Python massively increased in adoption
- The adoption of cloud hosted Git and CI/CD pipelines, alongside DevOps practices
- The shift to agile development practices
- Massive adoption of open source packages in software development
- Shift to microservice driven development, APIs and containerization

These changes introduced a new wave of development and observability tools, from GitHub to DataDog. At the same time, solutions like WhiteSource (now Mend), Sonatype, and Snyk emerged, addressing both the growing importance of open source vulnerabilities and the need for faster scan times.

These Software Composition Analysis (SCA) tools saw rapid adoption because existing platforms did not yet support SCA capabilities, and didn’t integrate natively with cloud hosted source code management software. Alongside this shift, the ability to scan code in pipeline created the fundamental promise of the "shift left" movement, an attempt to find and fix vulnerabilities in code before they were deployed.

Shifting left brought a utopian vision of application security - developers organically fixing every vulnerability out of the overflow of their hearts. On the plus side, potential vulnerabilities were identified earlier in development, and teams got new insights into what actually made up their software. On the negative side, developers were bombarded with confusing false positives, left spinning their wheels with pipelines blocked over nothing. Nonetheless, the idea of getting findings directly to developers as they're coding became the driving force of application security.

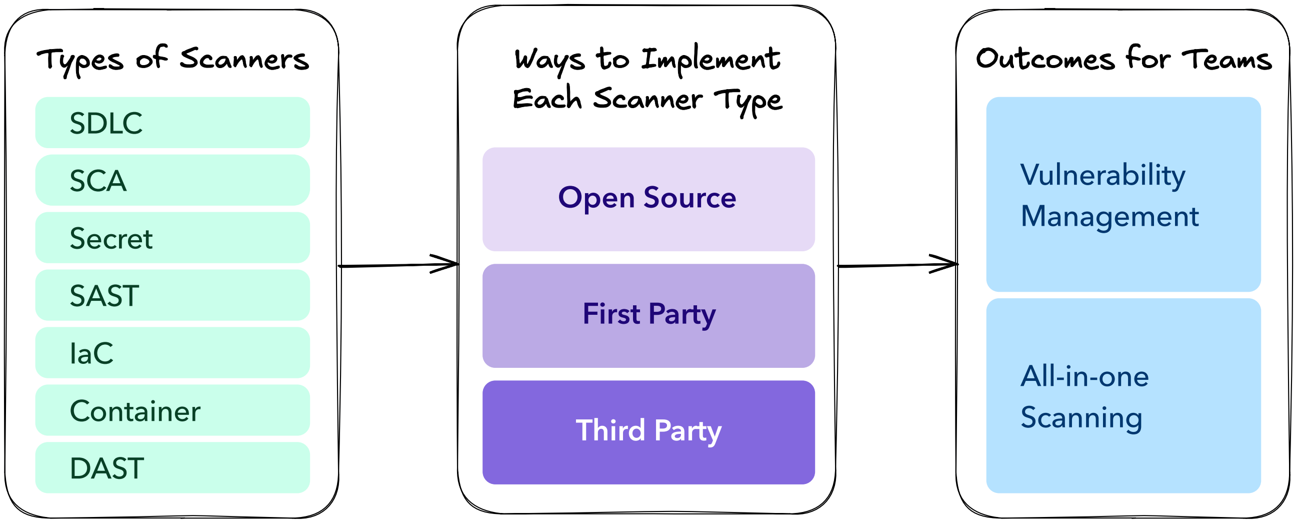


The Market Transition to Platforms

As application security tools became more integrated with developer workflows, it was unrealistic to have developers managing numerous scanners. Snyk was the first incumbent player to define the modern application security platform as they expanded from in pipeline SCA scanning to containers, IaC, and SAST. These scanners existed in other tools, but surfaced results in ways that were detached from the developer experience. Snyk treated everything in a repository as code, surfacing results directly to developers in near real-time, and into the workflows they were used to.

Snyk's expansion into a platform was part of a broader shift in application security. Enterprise incumbents like Checkmarx and Veracode began speeding up their scanning engines for in pipeline scans, and building unified cloud hosted offerings. While these established players were consolidating their capabilities, many startups emerged to disrupt the market.

Several application platform startups like Aikido and Arnica focused on orchestrating and consolidating different tools, often using a combination of open and closed source scanners. Another set of startups like Apiiro, Cycode, and Legit started from the opposite direction - giving teams the ability to manage their different scanners and get visibility into their pipelines. The end vision for most of these tools however was the same: creating an all in one application security platform.



Why Traditional ASPM Failed

The application security platform space has grown particularly confused around the term Application Security Posture Management (ASPM) and the capabilities a platform should deliver. Existing definitions from other analyst firms often fall short, typically describing ASPM as little more than a tool for managing and orchestrating scan results.

However, most products that manage third party results also introduced their own scanners over time. Because these scanners often used powerful open source foundations, it has become rare for teams to run thorough detection bake-offs between scanners themselves, such as trying to compare the capabilities of 8 different scanners directly within a one month trial period. To add to the confusion, numerous platforms launched “ASPM” solutions that were little more than dashboards for managing their own tools.

In our last report, we explained that pure play ASPM solutions are better relegated to the realm of CTEM, as larger vendors went on acquisition sprees to create vulnerability management platforms for every kind of vulnerability. The desire to create a code to cloud picture of a vulnerability has made application security a clear requirement of larger vulnerability management providers. Meanwhile, the advent of AI coding has completely transformed the needs of application security platforms.

As early as last year, code to cloud correlation was the defining feature of application security platforms, as the capability is essential for accurately prioritizing and assigning vulnerabilities in modern systems. However, tools for securing AI assisted development have since become the critical feature for application security buyers. The challenge for buyers and vendors alike is that there’s no defined best practices for AI security development. While some vendors are offering helpful tools, like defining security practices in rules files, we are a long way away from understanding the right kinds of scanners and workflows that will be used at scale.

Moving Towards an AI Future

Securing AI generated code has created a totally new set of technical requirements - MCP servers, rule file management, context management - all data that sits outside of the repository itself. Combined with rapid developments in AI scanning capabilities, it's no longer true that the scanners are commoditized.

> 💡 While application and cloud security will unify towards product security, the rapid adoption of AI code generation tools has steamrolled this simplified vision for the future.

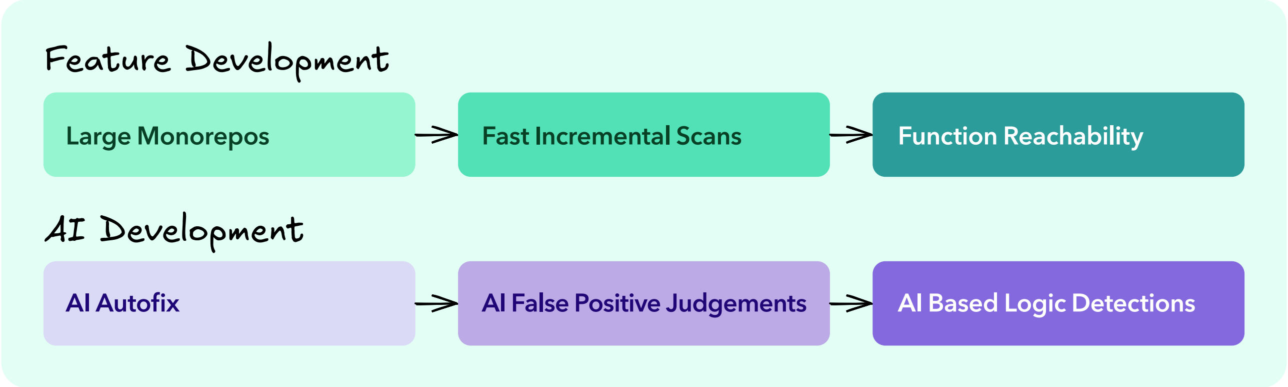
There's never been a more exciting time to be an application security practitioner, and a more uncertain time to be a vendor. New use cases are emerging everyday, requiring rapid pivoting of underlying capabilities, such as focusing on developer endpoints rather than pipeline based code scanning. The most promising approaches bring organizational context directly into the agent's context window, enabling developers to avoid preventable mistakes, like building a custom encryption engine or dealing with complicated library version choices.

The risk AI code generation presents for vendors also creates an opportunity. Just as Snyk's growth was due to rapidly accommodating new workflows, AI code generation is transforming how code is built and deployed. A new paradigm for secure code generation is needed, and someone's going to win this next wave of the market. The challenge for emerging vendors is that OpenAI and Anthropic have a good chance to capitalize on the opportunity themselves; however, this is what people assumed would happen with cloud providers, who failed to deploy security features fast enough to win adoption.

Our prediction is that the new AI driven SDLC will create another application security unicorn, the company that will adapt most quickly to the new workflows of software development.

THE CURRENT STATE OF APPLICATION SECURITY

SAST



Feature Development

Large Monorepos → Fast Incremental Scans → Function Reachability

AI Development

AI Autofix → AI False Positive Judgements → AI Based Logic Detections

Despite the commoditization of core scanners, they still generate more noise than value. On the one hand, Static Application Security Testing (SAST) is a check the box feature, but on the other, scaling a SAST program requires a lot of customization and false positive management. As with code generation, AI is fundamentally changing the benefits of SAST scanning, making results more novel and actionable.

Before exploring the exciting developments of AI SAST, it's important to understand that for large enterprises, legacy scanners are still used due to their support of:

- Large compiled binaries
- Complex static languages with substantial technical debt
- Large monorepos for core services

The shift to rapid in-pipeline scanning was essential for supporting flexible microservice driven development, but is why some large enterprises still use multiple SAST scanners - some for their modern services, and others for their older systems.

Over the last 5 years, the debate around SAST mostly centered on if a company was "just an open source wrapper" or not. It's becoming increasingly clear that buyers should prioritize the quality of the output, rather than the details of the scan engine.

To understand how to evaluate competing SAST vendors, it's important to know what SAST actually consists of. In practice, SAST is a combination of two core functionalities:

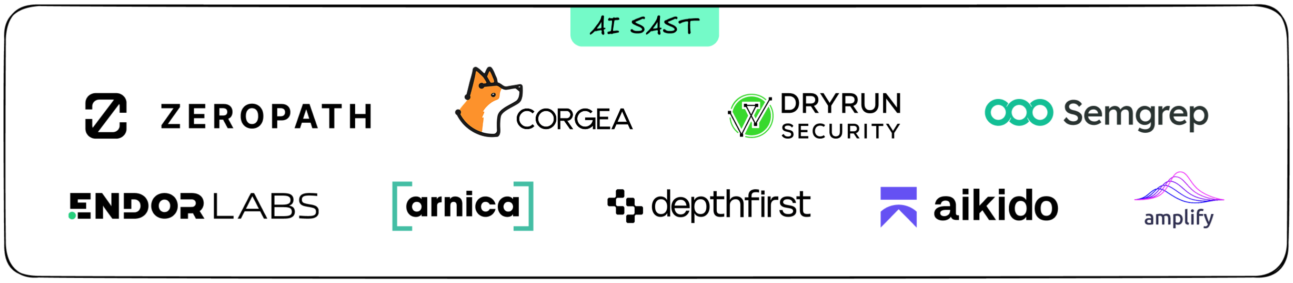
- An Abstract Syntax Tree (AST) - a way to query specific patterns in code.
- Rules that use the AST to look for potential security vulnerabilities.

In other words, it's not the engine that matters, but how it's used, and how customizable it is. Companies that take open source rules without enriching them will have noisy and inefficient scanners, while those who properly customize will have the best.

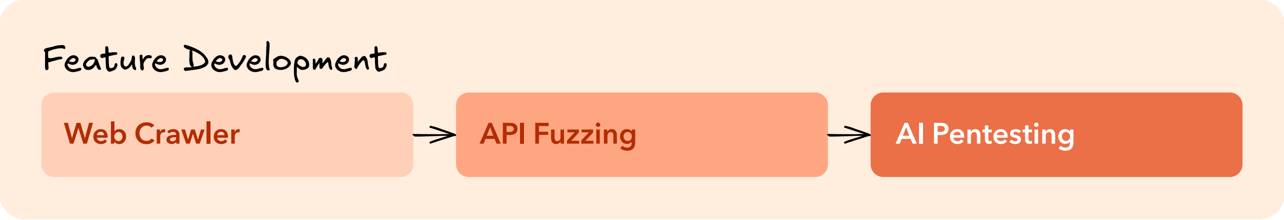
The emergence of AI-driven SAST has offered practitioners a massive advancement in scanning and prioritization capabilities. Early advances in AI SAST help rule out false positives and discover logic and misconfiguration vulnerabilities, opening new possibilities for static code analysis. These tools surface the kinds of issues traditionally found only through manual reviews or pentests. At the same time, they introduce new challenges around de-duplicating findings, scan performance, and incorporating organizational context.

Despite these new challenges, AI based static analysis is the 10x differentiator the static analysis category has needed - the results are a generational improvement that cannot be ignored. That being said, the user experience of SAST tools is also changing, as in-pipeline scanning becomes more about AI code reviews than pure scanning.

Organizations should prioritize modernizing their in-pipeline code review process to rely more on AI based analysis, while investing in the business logic capabilities of AI based scanners. Post-AI startups, like Corgea and Zeropath, are the leaders in these capabilities, while many incumbents are rolling out their own versions of AI scanning.



DAST and API Security



The first wave of dynamic application testing solutions focused primarily on crawling an application's webpages, injecting various payloads, and reading responses to see if they were successful. These early solutions offered a scalable way to test applications after they were deployed, and were also adopted by pentesters and red teamers as part of their toolkit.

Agent Based Detection & Response

- impart
- Operant
- upwind
- oligo
- DATADOG
- MIGGO
- WIZ
- dynatrace
- sweet.
- RoonCyber
- CONTRAST SECURITY

DAST as Part of AppSec

- invicti
- aikido
- snyk
- GitLab
- Checkmarx
- SOOS
- harness
- fluidattacks
- BLACKDUCK
- VERACODE
- CONTRAST SECURITY
- ox

API-First Testing

- escape
- invicti
- STACKHAWK
- 42crunch
- Bright
- pynt

Primarily Network Based

- harness
- Akamai
- akto
- CEQUENCE
- LEVO
- wallarm
- AppSentinels
- SALT
- Operant
- invicti
- MIGGO

DAST as Part of CTEM

- WIZ
- upwind
- PHOENIX SECURITY
- Qualys
- sweet.
- tenable
- RAPID7

AI Pentesting

- XBOW
- aikido
- WIZ

AI Incorporation

- escape
- STACKHAWK
- invicti

With microservice development, a new line of API first testing solutions emerged. These solutions were split between "API Security" solutions, which functioned more as runtime based API specific WAFs, and API testing solutions that excelled at fuzzing API endpoints directly when provided API spec files. These solutions have continued to mature, generating specs themselves, and improving their scanning performance. Solutions like Escape and Stackhawk are especially strong at running meaningful testing of API endpoints, and focused providers in this area are worth exploring for teams with modern API first architectures.

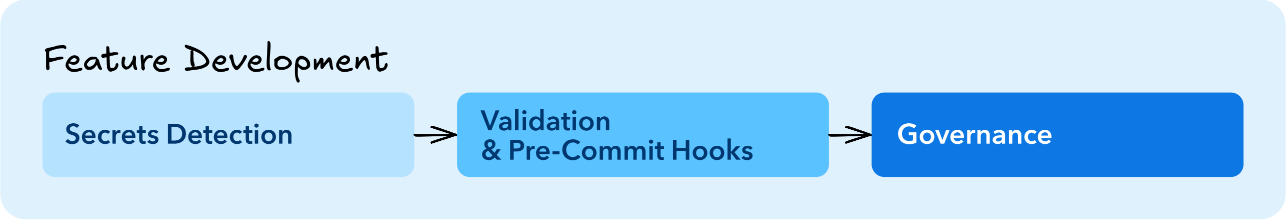
Like it has for SAST, AI is rapidly transforming the nature of pentesting, Dynamic Application Security Testing (DAST), and bug bounty programs, offering autonomous crawling and assessments of public facing endpoints. To give two examples of AI Pentesting, one is Wiz's dynamic payloads for testing public exposures. This is a lightweight solution that tries to test for exploitability from outside the environment - much like a traditional DAST, but with more flexibility. A second example is Aikido's approach, where you provide the platform with different user accounts in order to validate authorization issues in runtime, and watch as the agents attempt different attacks.

AI DAST vs. AI Pentesting

- AI Pentesting: gives agents a variety of tools to enumerate endpoints
- AI DAST: runs application contextual payloads to look for business logic flaws
- Both AI Pentesting and AI DAST offer similar results with different implementations
- Both discover an organization's public endpoints, moving towards Attack Surface Management

Choosing an approach in this category ultimately depends on the outcomes an organization is seeking. DAST generally delivers more consistent results with greater clarity, while AI-based pentesting tends to uncover more unique vulnerabilities, often at the expense of consistency. Over time, these categories will converge and the distinction made in our report is intended to clarify differences in approach rather than to imply a hard boundary. This is why in our images we highlight both the AI pentesting approach, and those who have built substantial AI capabilities into their tooling.

Secrets



Of all the tools in an application security arsenal, secret scanning remains the most binary - it's either a top priority, or a very low one. On the one hand, detecting secrets has always been a relatively commoditized offering, as strong open source options have existed, as well as the more obvious regex use cases for implementation. Some tools like Snyk combined secret detection with their SAST engine, never marketing it as a separate capability; others like GitGuardian invested heavily in the capabilities, building a platform around them, alongside building robust libraries of secret types.

In 2018, GitHub entered the secrets scanning market, and did so in a way that makes it difficult for competitors to provide as simple a deployment. Once a secret is pushed to a repository, dealing with it becomes a major pain. For scenarios where a team can't seamlessly rotate a secret, they need to take the risky action of removing the secret from the repo, as well as the commit history, and force pushing their branch upstream - overwriting the entire history of the repository.

This is why pre-commit hooks, wherein a secret can be detected and blocked before it's sent to the upstream repository, is the essential feature of a secret scanner. The challenge is in deploying these endpoint capabilities across developer laptops. While many companies offer pre-commit hooks, GitHub's advantage is baking these capabilities directly into their own hooks, allowing them to be deployed without additional efforts.

Dedicated platforms like GitGuardian and Trufflehog have innovated by expanding their validation and detection capabilities outside of developer commits in source code. These additional capabilities take their secret scanning offerings across other platforms like Slack or Google Drive, and do more granular validation of the permissions of the leaked keys. These developments are turning the platforms into more non-human identity security providers than traditional scanners.

Software Supply Chain

Static Function Reachability

Most vendors support some amount of static function level reachability, here are some technical considerations:

- Without a CLI, doing this with static languages is incomplete
- Doing this analysis means an increased scan time, it's used for prioritization, not pipeline
- Expect different results based on language, CVE coverage and provider

0-Day Malware Detection

- aikido
- Socket
- WIZ
- DATADOG
- KOI
- Safety
- GitHub
- VERACODE
- Chainguard
- Checkmarx
- xygeni.
- boost security.io
- ENDOR LABS
- paloalto NETWORKS

Patch Assistance

- backline
- FOSSA
- Moderne
- konvu
- HEELER
- ENDOR LABS
- Hopper
- maze
- AISLE
- PHOENIX SECURITY
- snyk
- Checkmarx
- aikido
- xygeni.
- VERACODE
- Mend.io

Runtime Function Reachability

- Kodem
- RAVEN
- oligo
- MIGGO
- DATADOG
- upwind
- konvu
- dynatrace
- CONTRAST SECURITY
- WIZ
- sweet.

SBOM Compliance Management

- FOSSA
- LINEAJE
- Scribe
- SOOS
- HEELER
- VERACODE
- ENDOR LABS
- Hopper
- Mend.io
- Socket
- snyk
- NETRISE

Patch Backporting

- root
- SEAL SECURITY
- Socket
- Chainguard
- aikido
- ENDOR LABS

The proliferation of open source software combined with an executive order mandating Software Bill of Materials (SBOMs) led to an explosion of supply chain security vendors. This has caused software supply chain security to become one of the most crowded markets in all of application security. Early on, differentiation largely came down to how tools performed scans, and what languages they supported. Some vendors, like Snyk, focused on code-level analysis through package files, while others relied on binary scanning, giving more depth insights but at the cost of speed and ease of adoption.

The code scanning approach proved to be the better user experience, but numerous other possibilities exist in supply chain security more broadly. First, the categories of malware detection and package health detection have always lingered as niche use cases (despite their risk being so large), with most startups focusing on these areas facing acquisitions over the last ten years. Only this year, with attacks like Shai Hulud having a massive impact, malware detection has become a mainstream necessity of Software Composition Analysis (SCA) tooling.

Reachability has proven to be an essential feature of SCA platforms, helping reduce massive vulnerability backlogs by providing evidence of exploitability. In brief, reachability analysis attempts to determine whether a vulnerability is actually exploitable within a given environment. At a basic level, this is commonly done by checking whether the vulnerable code is used by the application or whether it is accessible to attackers. You can read more in depth about the types of reachability analysis in our series on reachability.

While several vendors offer strong reachability analysis capabilities, the feature itself is difficult for organizations to evaluate because vendor databases and methodologies are proprietary. Dedicated SCA providers tend to offer stronger support but effectiveness depends heavily on the languages and architectures they support.

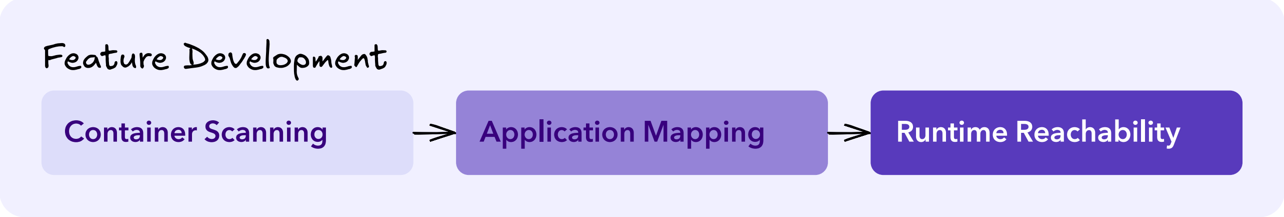
The core problem of SCA has always been that updating open source software is hard. If patching software was easy and automatic, vulnerability scanning wouldn't need to exist at all. Today, many vendors understand the core problem, and are building the solutions to help.

With the rise of AI, new approaches have entered the market, particularly breaking change analysis and backporting patches. Breaking change analysis is one of the most promising approaches to aiding the patching process - it utilizes function level reachability to lookup function changes between versions, and suggests the required changes needed to patch.

Even as AI makes software changes easier, backporting patches will continue to have a role in large enterprises, where patching certain systems brings unacceptable risk. The trade-off is that it has the potential to create divergent supply chain problems, and may introduce additional risk if the vendor patch wasn't completed correctly

Ultimately, AI coding tools are also developing to assist with the patching process, helping developers use the latest versions of software packages by looking up the versions before using them. Similarly, they're getting better at looking up the relevant changes to migrate between versions. We're finally moving towards a future where patching becomes easy enough to actually help drive vulnerability backlogs down.

Cloud and Infrastructure



Feature Development

Container Scanning → Application Mapping → Runtime Reachability

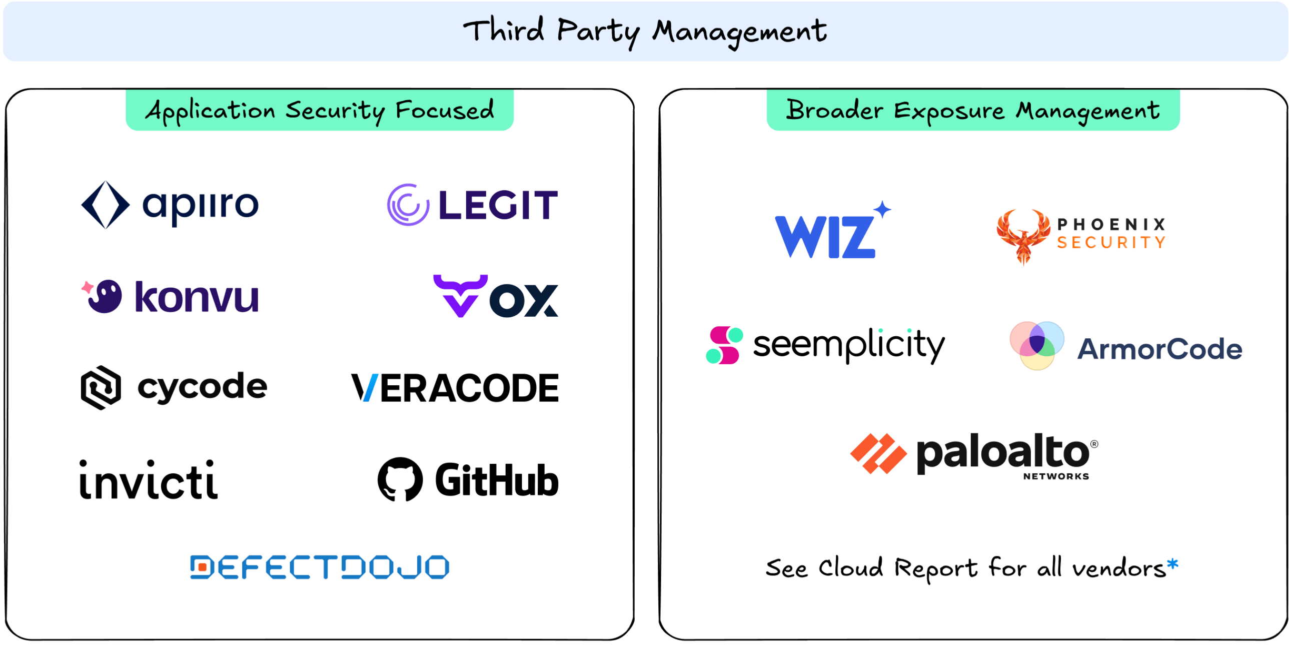
When tools start scanning containers, they open Pandora's box of complexity. Once you scan a container, you're quickly involved in its entire lifecycle - scanning the Dockerfile, the build process, the container registry, and the running container. Additionally, many teams don't realize they're getting duplicative results from their container scanner and their SCA scanner, adding another layer of confusion.

The responsibility of scanning containers has landed squarely on the cloud security side of the fence; however, typically application security tools do a better job of delivering the results to the teams that can actually do the fixing. This tension has led most large application security providers to provide container scanning in some form or another as part of their platform.

Infrastructure as code has the same problem, landing as an infrastructure team responsibility, but being more native to an application security integration workflow. This functionality is generally the most underbaked, because while basic scanning is widely accessible in open source tooling, making it work for your environment requires a high degree of customization. Scaling IaC analysis quickly leads to high levels of complexity, as building drift detection and support for custom modules become must-have capabilities. Other tools also include basic CSPM scanning as part of their platform in order to function as an all in one platform for smaller companies.

All in all, cloud security capabilities have long faced an awkward overlap with application security. On one hand, for organizations that are heavily invested in infrastructure as code and have strong deployment guardrails, an application-security-first approach works well. However, many companies operate in far more cobbled-together cloud environments, consisting of a mix of manually deployed and infrastructure-as-code-managed assets. These organizations tend to value the runtime monitoring capabilities of cloud security providers more than treating everything as code. Nonetheless, having these capabilities closely tied together creates many benefits to prioritization, drift detection, and contextualizing findings.

Management Tools



Third Party Management

Application Security Focused

Broader Exposure Management

See Cloud Report for all vendors*

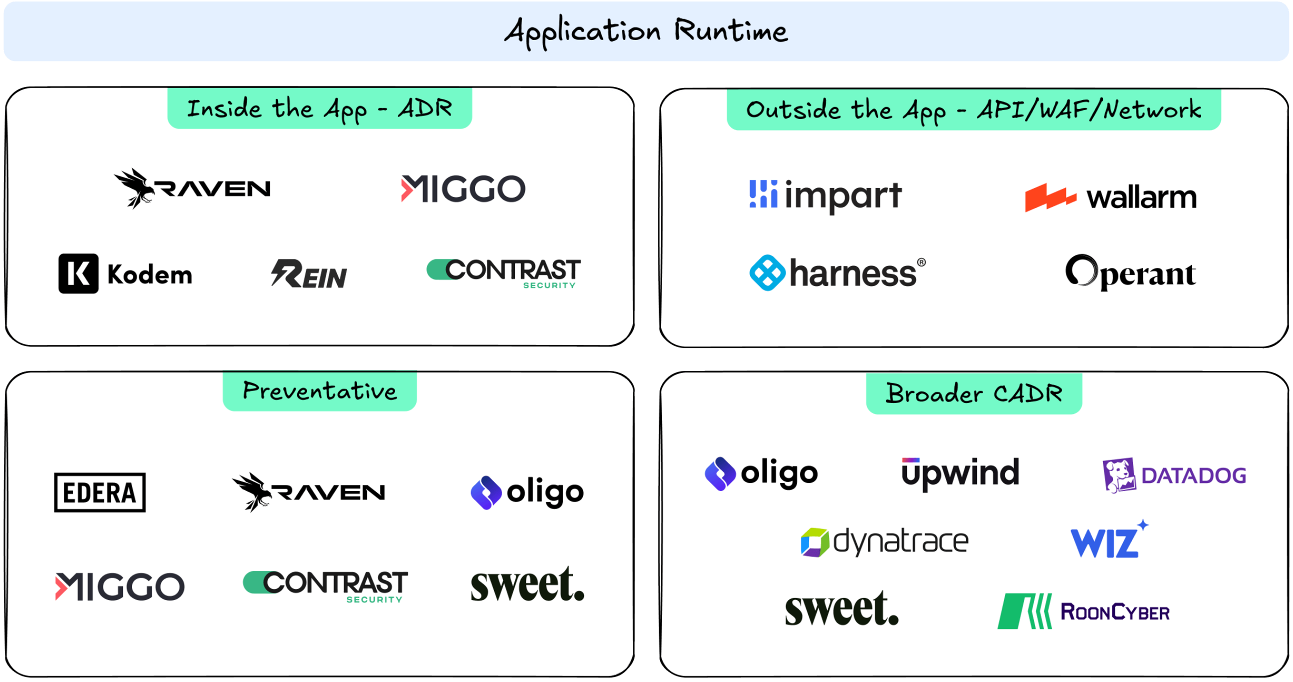
In our cloud security report, we made it clear that vulnerability management solutions are consolidating under CTEM functionalities. ASPM vendors, as traditional analyst firms described them, were a transitional tool for consolidating the findings of different application scanners. Consolidating findings across containers and SCA scanners was important, but the broader de-duplication and contextual project has more to do with infrastructure than code.

The threat exposure management category (vulnerability management 2.0) remains top of mind for security leaders entering 2026, but they don't necessarily expect their application security scanner to be that same tool. Developer experience and finding quality remain paramount, with broader vulnerability management goals typically owned by different teams.

Even as the market shifts toward more unified exposure and vulnerability management tools, there can be meaningful benefits having an orchestration layer on top of your application security scanners. For organizations with tens of thousands of GitHub repositories, monitoring pipeline coverage, understanding which applications map to which teams, and orchestrating scanning tools are massive challenges.

Tools in this category take different approaches to solving these enterprise problems. Some, like Apiiro, focus on a more application-centric model. Others, such as Phoenix Security, lean more heavily into cloud and container-centric views. Legit Security emphasizes CI/CD coverage, while Palo Alto Networks brings a cloud-centric perspective through its broader platform. Each of these approaches deliver value for large organizations that need clear visibility into security coverage across sprawling development environments.

Runtime Technologies



The shift left and secure by design movements promised perfect software, but the reality was large backlogs of vulnerabilities that get explained away rather than fixed. While prioritization and code scanning techniques have improved, runtime remains the source of truth for assessing exploitability and implementing protection. Runtime oriented solutions provide the best possible prioritization and protection organizations can get.

For reachability analysis, ADR providers have built extensive libraries of vulnerable function executions that they monitor for at runtime. If a vulnerable function is executing, teams can have high confidence that an exploit is possible. While the functionality might not seem that different at first glance, runtime function level reachability is a necessary improvement for modern programs. When every finding is interesting, and developers can see exactly how the vulnerability is working, it creates a level of trust and interoperability that older tools haven't been able to achieve. Based on the usage of tools such as Oligo, Raven, and Kodem, these capabilities can meaningfully improve prioritization across both code and operating system packages.

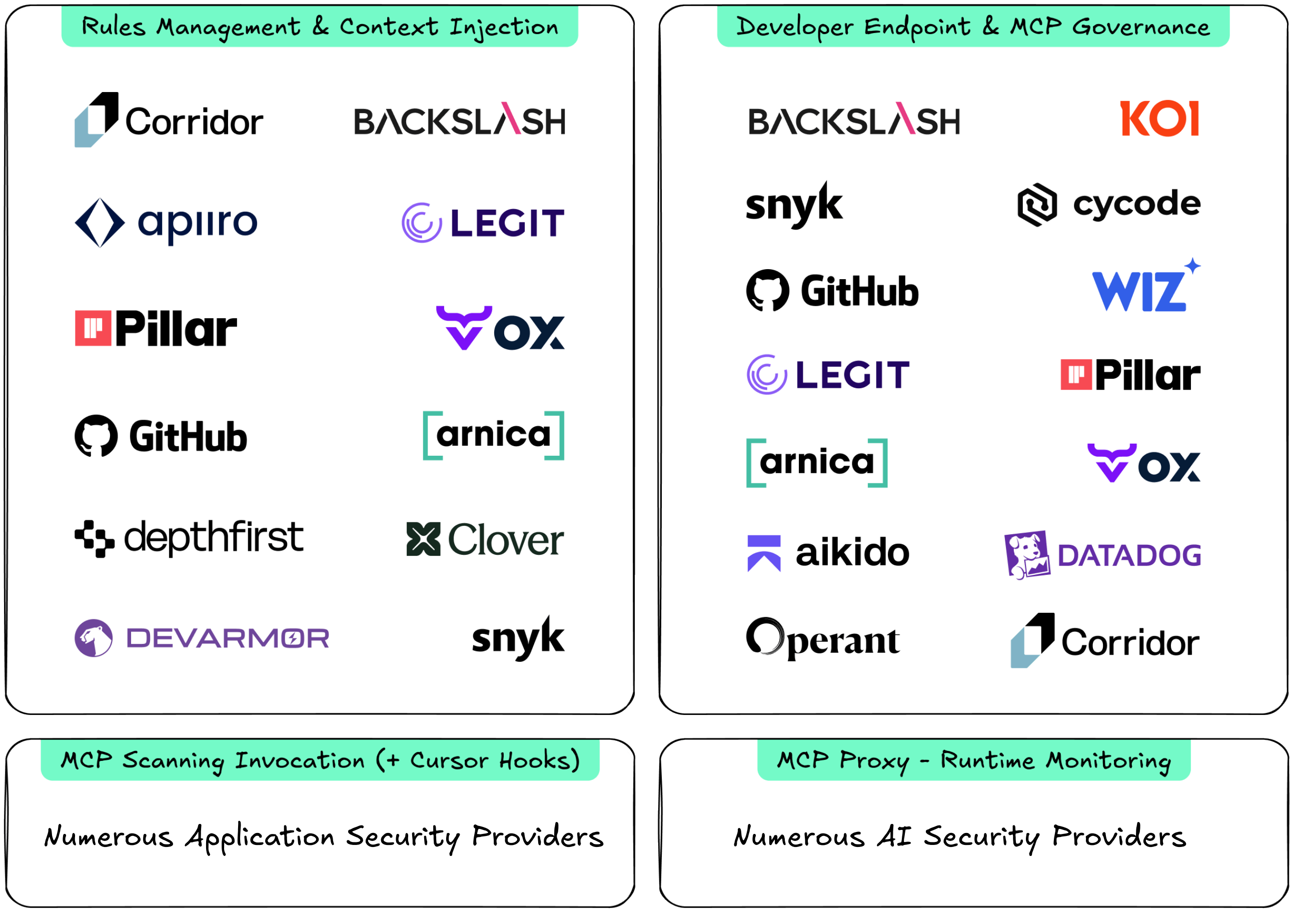
As much as prioritization is helpful, organizations will never hit zero vulnerabilities, or be perfectly protected against zero days. React2Shell was a great example of this, as vendors and attackers raced to play WAF wack-a-mole with each new exploit. This is the true purpose of ADR - helping to protect against ongoing attacks in real time, without needing to sit by and pray that you, or the open source community, get a patch out in time.

It’s also worth discussing the impact runtime technologies are having on improving on the concept of WAF more broadly. There are several approaches out there, each with pros and cons, but each valuable additions to a WAF centric approach. One approach pioneered by Impart Security focuses on providing an eBPF-based WAF, which can be easier to manage and enables more contextual rule creation than traditional WAFs. Another approach comes from Miggo, which applies its function-level reachability capabilities to generate more precise WAF rules. The company has published extensively on this approach in the context of the React-to-Shell vulnerability. A third approach from those like Raven and Oligo utilize preventative security capabilities inside applications themselves to stop novel attacks.

This report focuses specifically on function-level capabilities for prioritization at runtime. For a broader discussion on the importance of CADR and how these capabilities fit into achieving best-in-class cloud runtime detection, see our Cloud Security report.

EMERGING CATEGORIES

AI Code Guardrails



Rules Management & Context Injection

Corridor, BACKSLASH, apiiro, LEGIT, Pillar, OX, GitHub, arnica, depthfirst, Clover, DEVARMOR, snyk

Developer Endpoint & MCP Governance

BACKSLASH, KOI, snyk, cycode, GitHub, WIZ, LEGIT, Pillar, arnica, OX, aikido, DATADOG, Operant, Corridor

MCP Scanning Invocation (+ Cursor Hooks)

Numerous Application Security Providers

MCP Proxy - Runtime Monitoring

Numerous AI Security Providers

It’s tempting for some to picture an AI utopia, where all generated code is perfectly crafted and tested, but the fact of the matter is that (for now) AI generated code is definitively insecure. Whether it’s the numerous research papers demonstrating it, or anecdotal evidence, AI can be too aggressive at delivering what developers ask for, no matter the security concerns.

The foundation model companies, from OpenAI to Anthropic, have strong incentives to make their code generation process as secure and effective as possible, but the reality just isn’t there yet. Combined with the accelerated risk of using more open source AI plugins in the form of MCP servers and IDE extensions, developer endpoints have become more exposed than ever.

In response to the rapid adoption of AI generated code, application security companies have taken different approaches to securing the process. First, I've seen some strong initial results from providers like Backslash and Corridor, building in security rules to add to the agent's context window. These providers allow security teams to monitor for malicious rules files, while also deploying rules files enforcing best practices, or secure architecture considerations specific to the repository. This approach is aimed at making AI generated code more secure by default.

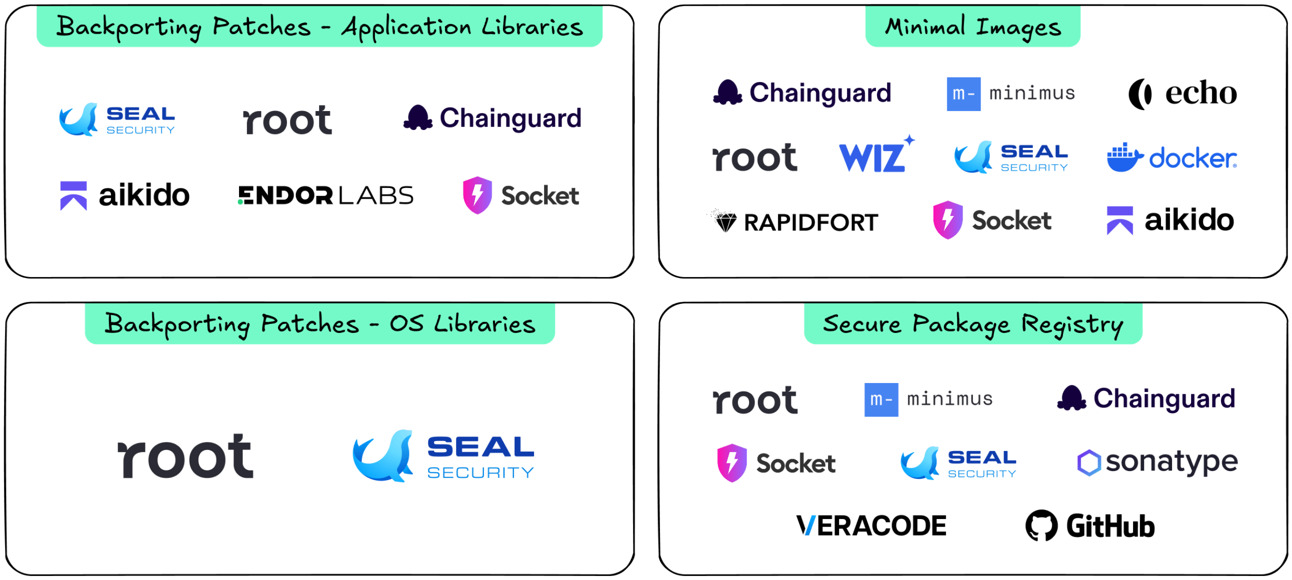
Another approach taken by some providers with a lot of cloud context, such as OX and Apiiro, is to give coding agents access to broader application context via MCP. This goes beyond the scanner invocation some teams are doing with MCP by instead giving agents helpful context about the overall environment. This helps agents automatically understand coding practices that are adopted across the organization, while sneaking in patches for older vulnerabilities along the way.

Finally, insights into developer endpoints are more important than ever as CISOs seek to enforce policies instituted by AI governance committees. This includes initiatives to control what models are approved, alongside AI code assistance tools. These solutions give teams the ability to control what MCPs and other tools are allowed in the organization. Snyk has recently moved in this direction, alongside providers like Koi which we wrote about here.

A sign of how rapidly this space is developing is that when we completed an in-depth report of AI autofixing last year, there were only a few startups in the space of creating AI generated fixes. Since then, many companies now offer and focus on AI autofixing. However, this entire approach has become antiquated by code assistants with MCP workflows generating much better fixes than any company's "proprietary fix engine."

These guardrail approaches offer promising initial results, but the future remains unclear of the best approach to securing AI generated code. With many vendors focusing on runtime testing instead of the guardrail approaches, it's too early to tell which comes out on top.

Secure Supply Chain



An emerging category of Secure Supply Chain tools focuses on securing the supply chain at the source by providing mechanisms to make open source libraries safer before they are imported into your environment. There are three main approaches to creating a secure by default open source consumption model:

- Reducing the attack surface of container images by minimizing them.
- Backporting patches for newly discovered vulnerabilities into older versions, allowing teams to avoid the disruption of major upgrades.
- Hosting a package registry where libraries are scanned before they're imported to reduce exposure to supply chain takeovers.

First, minimal container images are a strong defense-in-depth initiative for teams looking to reduce their attack surface. However, they can be challenging for developers to adopt. Even in relatively small applications, migrating to minimal images can introduce unexpected issues, as many applications implicitly rely on packages that are no longer present.

Vendors have tried to address these challenges with minimal image adoption in several ways:

- Custom image builders
- Debian-based images
- Minimizing existing images instead of requiring a full transition

If you're considering moving to minimal images, first be sure to rebuild your images nightly. Rebuilding images will apply more security benefits than any major migration to minimal images, and even if teams move to minimal images, they'll need to rebuild them regularly for patching.

Second, backporting patches can help in large enterprise environments because major version upgrades have the potential to introduce more risk than they eliminate. These upgrades often involve major architectural changes. To mitigate the risk associated with upgrading, some vendors backport security patches to older versions of software. This approach can meaningfully reduce risk with minimal operational disruption but scalability and long-term support will remain as challenges.

Third, secure package registries hosted by vendors offer mixed benefits. Although scanning packages for security issues before approving new versions can be helpful, there are simpler solutions, such as introducing a new version cooldown or version pinning. At the same time, popular ecosystems like npm are gradually introducing stronger authentication mechanisms to reduce the risk of maintainer account takeovers.

Secure Supply Chain tools are most effective in highly regulated environments with legacy systems that are difficult to patch, or where compliance requirements place heavy emphasis on vulnerability counts. Regardless of the approach used, it is always worth noting that regularly rebuilding and redeploying your software remains the only reliable way to maintain a low-CVE environment.

Developer PAM

P0 SECURITY   Formal   Lumeus   strongdm   Teleport

Governing developer access to infrastructure has long been overlooked: how their credentials are rotated, how just-in-time privilege workflows are enforced, and how NHI's are managed. Many legacy Privileged Access Management (PAM) tools were designed around Windows-based infrastructure, focusing on laptop access, remote desktop, and database connectivity. Modern development environments, however, require access to Kubernetes clusters, cloud environments, and more recently, AI agent tooling. This shift has expanded the attack surface of developer accounts that are often poorly monitored and insufficiently controlled.

The tools highlighted above focus on providing developers with secure access to a wide range of resources while delivering several additional benefits, from just-in-time access to real-time visibility. In many organizations, platform teams have built distributed homegrown systems to manage access, which leads to sprawling permission models that are difficult for security teams to audit and understand. PAM tools excel at restoring control and consistent visibility without disrupting developer workflows.

From a developer perspective, tools in the PAM category make access management far more streamlined than maintaining a collection of long-lived SSH keys or custom tooling.

Code Quality

Sonar DATADOG GitHub Codacy
aikido CORGEA Clover

Code quality and code security often cause confusion, as they should address closely related concerns. In theory, checking code for security issues is very similar to checking code for quality issues. But these tools have historically been built for different personas, leading to different noise-to-signal ratios across each category and disconnected workflows.

Unifying code quality and code security can consolidate tooling and integration needs. The primary benefits are the ability to manage a single scanner across development pipelines, and having a single source of truth for developers to track rules, findings, and remediation priorities. This consolidation reduces operational overhead while making it clearer what needs to be fixed and when.

The most important capability of a code quality tool is the ability to measure code coverage, specifically the percentage of code functions that have tests written for them. Like security posture, code coverage tends to drift over time, with developers struggling to consistently meet quality standards in the same way they can fall short on security requirements. This struggle highlights the shared challenge both code quality and code security tools are intended to address.

Many tools in the code quality and code security categories are now blending AI-driven code review rules across both quality and security, giving teams a unified AI pull request reviewer. From the experience of using AI code quality reviews from both Aikido and Corgea, their ability to catch common issues in AI-generated applications, including authentication flaws and always-true conditional statements is impressive.

AI Prioritization and Remediation

Last year, we released a report that objectively tested different approaches to AI remediation for static code analysis. Since then, autofix capabilities have become widespread across many code security platforms. However, the capability itself has largely been replaced by frontier models and tools like Claude Code, driven by the superiority of dedicated AI agents, rules, and workflows that can deploy large-scale, contextual fixes across entire codebases.

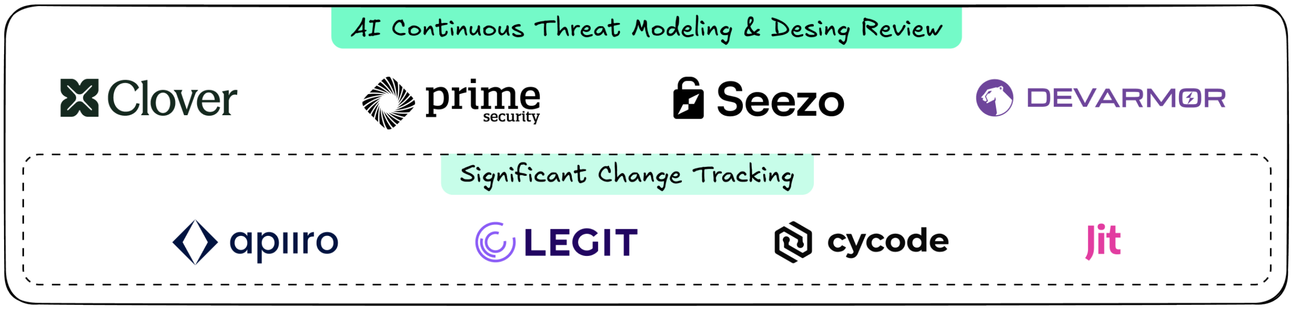
Although many providers now offer both AI-driven prioritization and remediation, the sophistication of these capabilities varies significantly, resulting in a major impact on backlog reduction and remediation outcomes. Based on our testing with more AI-native tools such as Zeropath and Corgea, a substantial number of false positives were tuned out across both SCA and SAST findings.

On the remediation side, the expansion of these capabilities into SCA is exciting. While prioritization alone is already a challenge in SCA scanning, remediation has always been the true bottleneck in the category. If everything is fully patched, prioritization efforts become far less critical.

Newer startups like Aisle, Backline, and Konvu have made meaningful progress generating end to end patches for SCA vulnerabilities. These tools go well beyond the basic approach of opening pull requests that simply bump dependency versions, instead completing major framework and library upgrades directly for users, while also analyzing existing findings for true positives.

AI prioritization and remediation continues to be one of the fastest-moving segments in application security. As frontier models improve at generating and fixing code on their own, security tools are increasingly evolving into contextual layers that guide those models toward higher-quality, safer code changes.

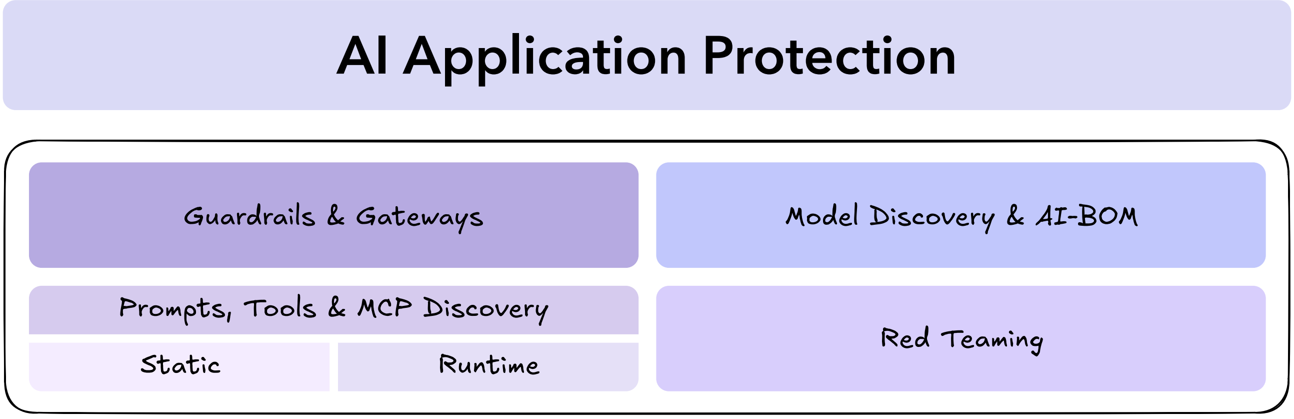
Threat Modeling and Design Review


AI Continuous Threat Modeling & Desing Review
Clover
prime security
Seezo
DEVARMOR
Significant Change Tracking
apiiro
LEGIT
cycode
Jit

Of all the emerging tools in application security, continuous threat modeling and design review is the most exciting area, both for the immediate value they provide and for their long-term impact on AI-driven code generation. These tools work typically by integrating with existing knowledge bases and source code to build an overall threat model of an application. From there, they continuously monitor changes to deliver ongoing design reviews and updated threat models as new systems are deployed.

Having a clear architectural view gives these tools a strong advantage when providing context to AI agents, which often struggle to understand an organization's broader application architecture. These guardrails can then be applied directly to new code generation, allowing teams to enforce organizational standards consistently across their codebase.

Another capability that's particularly valuable is significant change tracking ability. This enables teams to gain visibility into major architectural or code changes that often happen without security being aware. Combining significant changes tracking with organizational context gives security teams real insight into the riskiest changes happening in their applications.


AI Application Protection

It is still too early to tell whether a dedicated vendor will be required to do AI application security, but dedicated startups are offering capabilities that go beyond larger platforms. Several of the tools covered in this report offer baseline AI Application Protection capabilities such as model discovery, basic AI red teaming, and other posture-oriented methods for identifying how an organization is using AI.

As a team prioritizes runtime protection, conversation guardrails, and MCP gateways, the likelihood of needing a dedicated AI security provider increases. This is where tools like Pillar and Operant are clearly differentiated, as they provide both runtime protection and granular visibility into AI-driven applications, alongside a more robust mapping of AI application architectures.

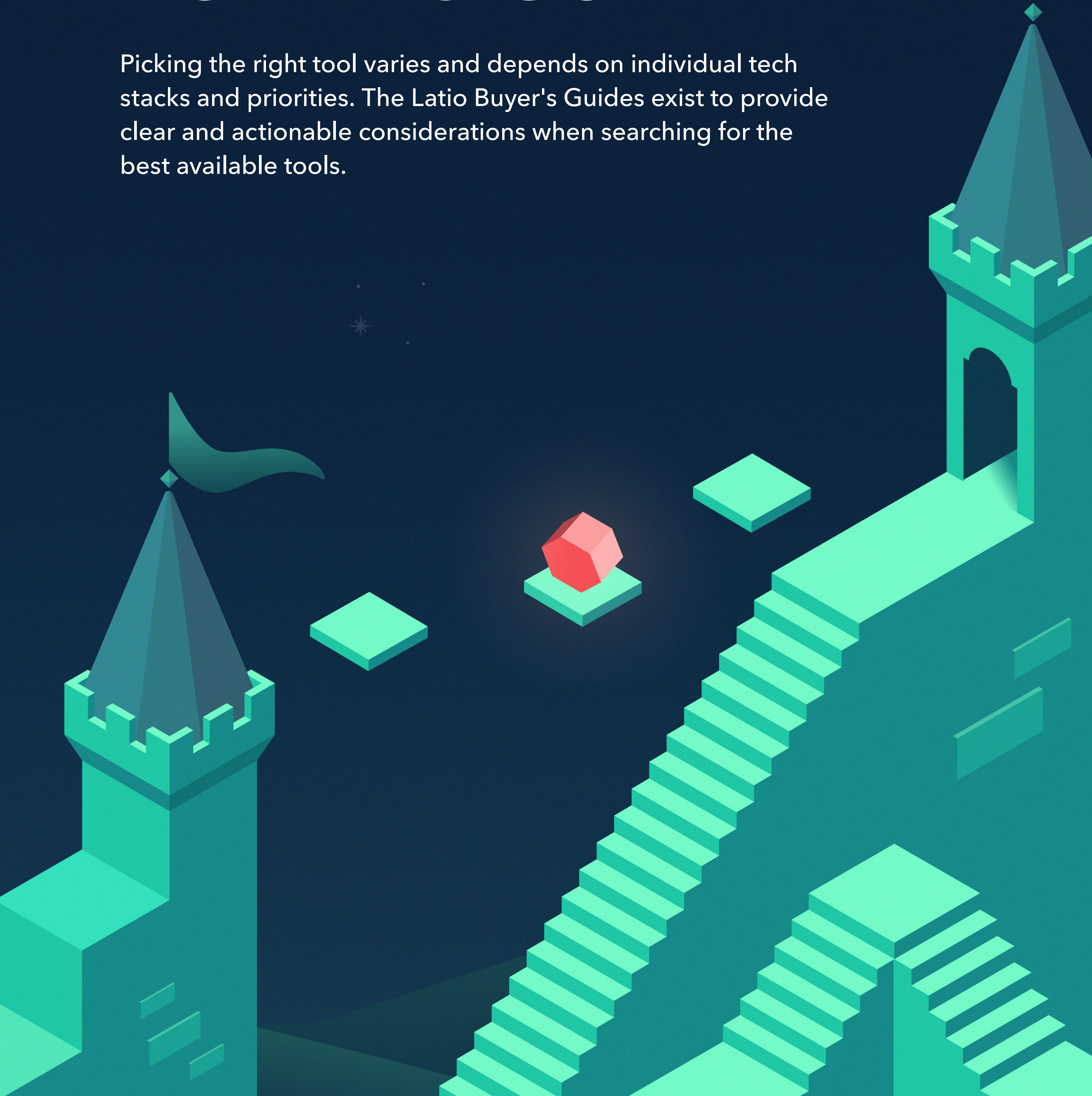
Setting aside the hype, AI red teaming is largely another form of DAST, where inputs are injected into a different type of system and outputs are evaluated, often by another LLM, for inappropriate or unsafe behavior. Similarly, AI BOMs and model discovery often resemble extensions of SCA scanning, as models are frequently packaged as Python or JavaScript libraries. That said, the maturity of these capabilities varies widely across providers.

One area that is especially compelling about ADR and CADR-style providers is their ability to secure AI applications from inside the running application itself. While many tools rely on observing inputs and outputs to infer how a model reached a particular outcome, ADR operates within the application, allowing it to observe chains of thought and tool calls far more natively.

We will cover the AI Application Security market in more detail later this year, but more details are available in the 2025 AI Security report.

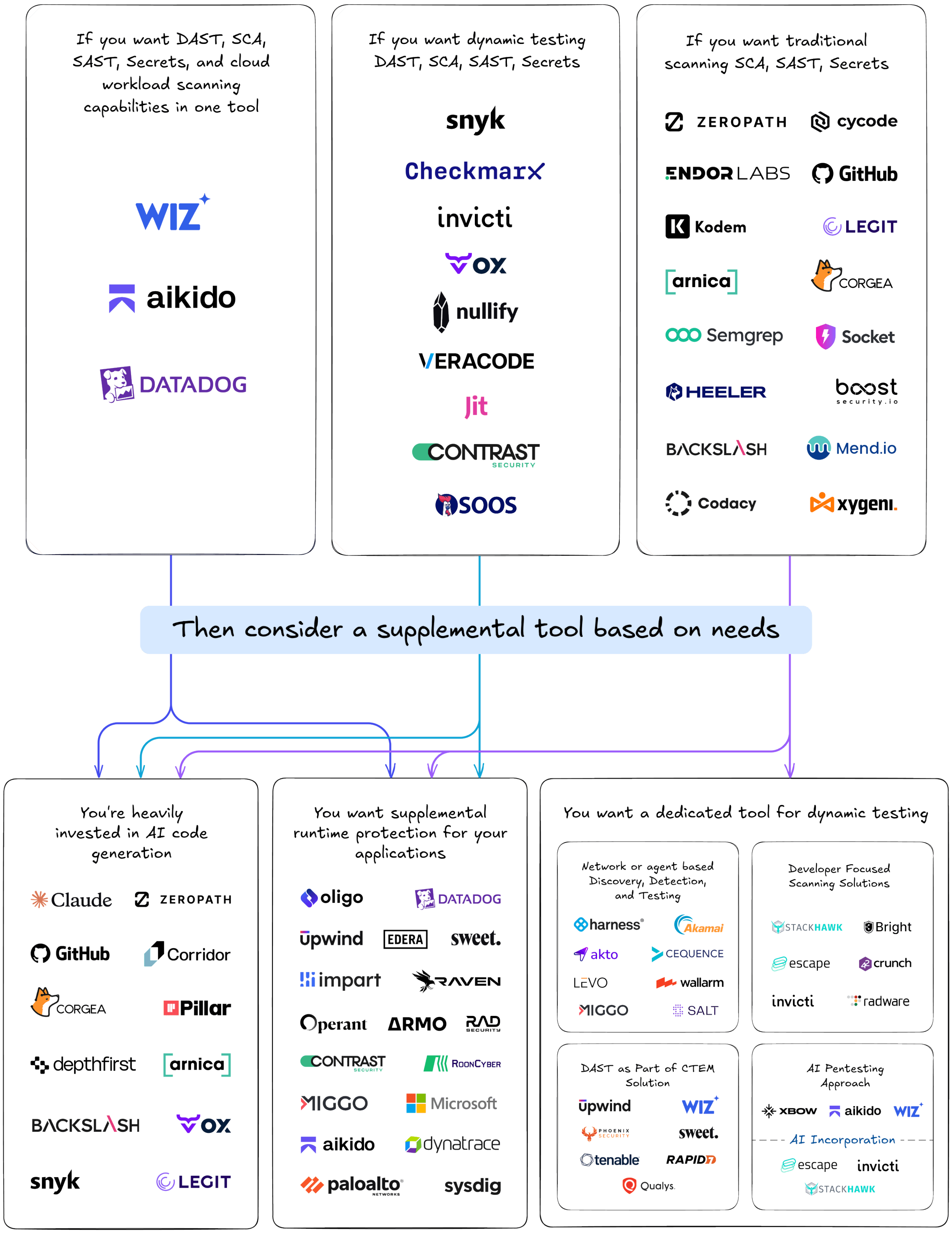
BUYER’S GUIDE

Picking the right tool varies and depends on individual tech stacks and priorities. The Latio Buyer's Guides exist to provide clear and actionable considerations when searching for the best available tools.



SMB and Mid-market Buyer's Guide

Most teams will start with one of the below platforms



If you want DAST, SCA, SAST, Secrets, and cloud workload scanning capabilities in one tool

WIZ
aikido
DATADOG

If you want dynamic testing DAST, SCA, SAST, Secrets

snyk
Checkmarx
invicti
OX
nullify
VERACODE
Jit
CONTRAST SECURITY
SOOS

If you want traditional scanning SCA, SAST, Secrets

ZEROPATH
cycode
ENDOR LABS
GitHub
Kodem
LEGIT
[arnica]
CORGEA
Semgrep
Socket
HEELER
boost security.io
BACKSLASH
Mend.io
Codacy
xygeni.

Then consider a supplemental tool based on needs

You're heavily invested in AI code generation

Claude
ZEROPATH
GitHub
Corridor
CORGEA
Pillar
depthfirst
[arnica]
BACKSLASH
OX
snyk
LEGIT

You want supplemental runtime protection for your applications

oligo
DATADOG
upwind
EDERA
sweet.
impart
RAVEN
Operant
ARMO
RAD SECURITY
CONTRAST SECURITY
RoonCyber
MIGGO
Microsoft
aikido
dynatrace
paloalto NETWORKS
sysdig

You want a dedicated tool for dynamic testing

Network or agent based Discovery, Detection, and Testing

harness
Akamai
akto
CEQUENCE
LEVO
wallarm
MIGGO
SALT

Developer Focused Scanning Solutions

STACKHAWK
Bright
escape
crunch
invicti
radware

DAST as Part of CTEM Solution

upwind
WIZ
PHOENIX SECURITY
sweet.
tenable
RAPID7
Qualys.

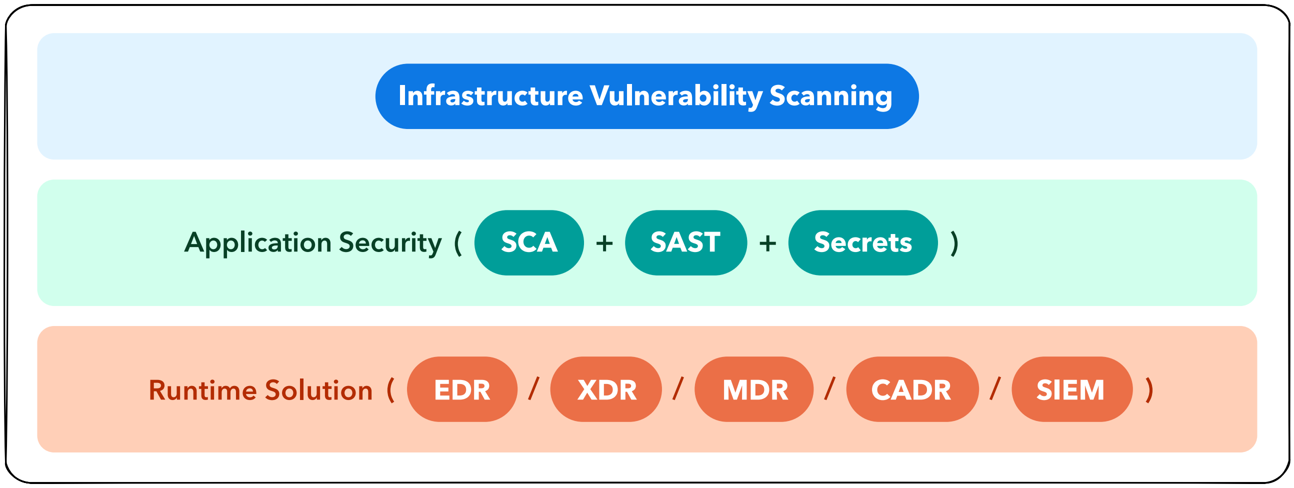
AI Pentesting Approach

XBOW
aikido
WIZ

AI Incorporation

escape
invicti
STACKHAWK

SMB and Mid-market



Infrastructure Vulnerability Scanning

Application Security (SCA + SAST + Secrets)

Runtime Solution (EDR / XDR / MDR / CADR / SIEM)

Mid-market organizations require certain core scanning capabilities from their application security tooling due to how their environments are structured.

Typically, mid-market organizations have one tool in place for infrastructure, one for application security, and one for runtime protection. Beyond these core tools, mid-market organizations may consider additional capabilities like:

- Having a separate MDM solution for device management
- A compliance automation solution for achieving SOC 2
- Depending on the type of business and infrastructure, they may prioritize only infrastructure or only application scanning.

If you're a company looking for a scalable application security tool, you probably care about:

- **Reducing developer work as much as possible**
  - Not wasting time on false positives
  - Having a workflow that's consolidated around cloud hosted products
  - Not using too many tools
- **Spending the least amount of money as possible**

Companies in this position will want to consolidate as many features into one application security tool as possible to save on management overhead.

The two key questions to answer in order to confidently guide your tool choice are:

- **Do you want a separate platform for cloud security?** While most of our survey respondents wanted separate tools, they're not always needed right away, or at all, depending on your specific architecture. There are benefits to consolidating these tools, but they're often separate disciplines.
- **Do you want DAST included?** DAST is one of the easiest compliance checkbox tools in your arsenal, so it's suggested, but many organizations make this a later priority in favor of the immediate value provided by static analysis.

Of these companies, it's worth noting Aikido, JIT, Arnica, and Socket have free account offerings and friendly pricing structures. Also worth noting are Semgrep's open source tooling, and GitHub's value through included or open source offerings.

Once core application security scanners are acquired, teams often consider DAST. It's important to understand that vendor approaches to DAST vary because:

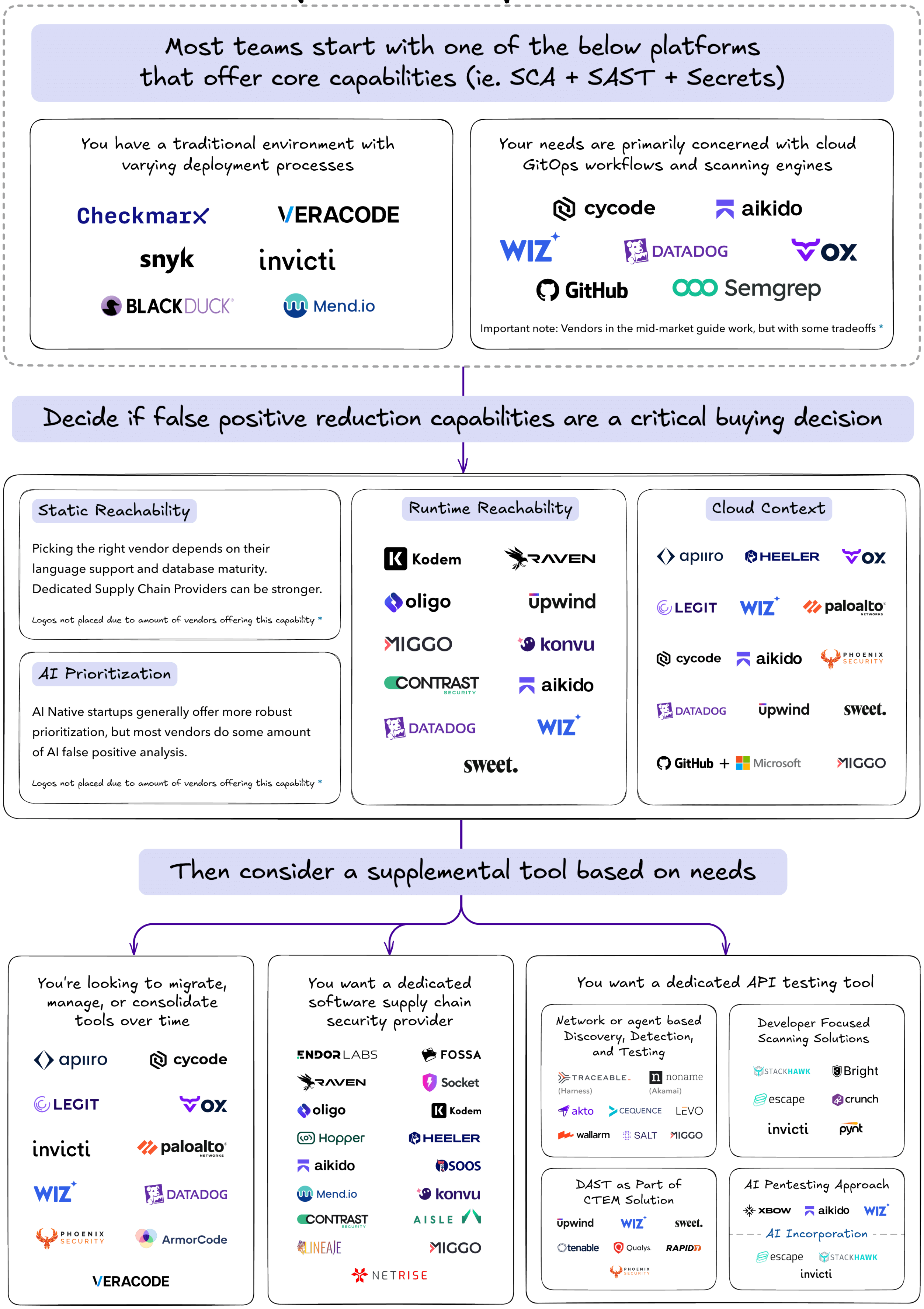
- DAST is included in many infrastructure security tools at a very low cost
- DAST's value has been de-emphasized by "DAST is dead" - old school scanners that lack API context
- Valuable API testing is difficult to build and maintain

While traditional DAST scanning has lost its value, meaningful API testing is an important part of a mature security program. We recommend that teams introduce a DAST that is API driven to support modern architectures, as older DAST scanners will provide little to no additional value in these environments.

There are two key trends that SMB and Mid-Market companies should be conscious of when thinking about tech stack structuring. The first is the shift from traditional in-pipeline scanning to in-pipeline code review, whether through AI code quality or code security tools. This approach can satisfy a large number of compliance requirements, while also giving developers a more unified experience that goes beyond traditional vulnerability management models

The second trend is pairing strong static scanning with an equally strong runtime offering, which is where the CADR category becomes particularly relevant. By combining an all-in-one application security testing platform with robust runtime protection, mid-market companies can achieve an extremely strong security posture. An all-in-one application security scanner with strong AI detection capabilities, combined with a meaningful application runtime solution is our mid-market security stack of choice.

Enterprise Buyer's Guide



Most teams start with one of the below platforms that offer core capabilities (ie. SCA + SAST + Secrets)

You have a traditional environment with varying deployment processes

Checkmarx, VERACODE, snyk, invicti, BLACKDUCK, Mend.io

Your needs are primarily concerned with cloud GitOps workflows and scanning engines

cycode, aikido, WIZ, DATADOG, OX, GitHub, Semgrep

Important note: Vendors in the mid-market guide work, but with some tradeoffs *

Decide if false positive reduction capabilities are a critical buying decision

Static Reachability

Picking the right vendor depends on their language support and database maturity. Dedicated Supply Chain Providers can be stronger.

Logos not placed due to amount of vendors offering this capability *

AI Prioritization

AI Native startups generally offer more robust prioritization, but most vendors do some amount of AI false positive analysis.

Logos not placed due to amount of vendors offering this capability *

Runtime Reachability

Kodem, RAVEN, oligo, upwind, MIGGO, konvu, CONTRAST SECURITY, aikido, DATADOG, WIZ, sweet.

Cloud Context

apiiro, HEELER, OX, LEGIT, WIZ, paloalto NETWORKS, cycode, aikido, PHOENIX SECURITY, DATADOG, upwind, sweet., GitHub + Microsoft, MIGGO

Then consider a supplemental tool based on needs

You're looking to migrate, manage, or consolidate tools over time

apiiro, cycode, LEGIT, OX, invicti, paloalto NETWORKS, WIZ, DATADOG, PHOENIX SECURITY, ArmorCode, VERACODE

You want a dedicated software supply chain security provider

ENDOR LABS, FOSSA, RAVEN, Socket, oligo, Kodem, Hopper, HEELER, aikido, SOOS, Mend.io, konvu, CONTRAST SECURITY, AISLE, LINEAJE, MIGGO, NETRISE

You want a dedicated API testing tool

Network or agent based Discovery, Detection, and Testing

TRACEABLE (Harness), noname (Akamai), akto, CEQUENCE, LEVO, wallarm, SALT, MIGGO

Developer Focused Scanning Solutions

STACKHAWK, Bright, escape, crunch, invicti, pynt

DAST as Part of CTEM Solution

upwind, WIZ, sweet., tenable, Qualys, RAPID7, PHOENIX SECURITY

AI Pentesting Approach

XBOW, aikido, WIZ

AI Incorporation

escape, STACKHAWK, invicti

Additional Enterprise Coverage Guide

Emerging enterprise capabilities that address modern security priorities

Upstream Malware Detection

- aikido
- Socket
- Safety
- Chainguard
- WIZ
- DATADOG
- VERACODE
- KOI
- Checkmarx
- xygeni
- ENDOR LABS
- paloalto NETWORKS

Business Logic Detections

AI DAST

- XBOW
- aikido
- escape
- STACKHAWK
- invicti

AI SAST

- ZEROPATH
- CORGEA
- arnica
- DRYRUN SECURITY
- amplify
- Semgrep
- aikido
- ENDOR LABS
- depthfirst

SCA Complete Patch

- backline
- FOSSA
- HEELER
- konvu
- AISLE
- Moderne
- VERACODE

AI Coding Guardrails

- BACKSLASH
- snyk
- OX
- Corridor
- LEGIT
- GitHub
- arnica
- Pillar
- apiiro
- KOI

Runtime Detection

- oligo
- impart
- RAVEN
- MIGGO
- upwind
- Kodem
- WIZ
- DATADOG
- sweet.
- CONTRAST SECURITY
- sysdig
- RoonCyber
- ARMO
- RAD SECURITY
- paloalto NETWORKS
- Operant

Secrets Detection Outside of Code

- GitGuardian
- TRUFFLE SECURITY CO.
- cycode

or any NHI Provider *

Secure Package Registry

- Chainguard
- root
- echo
- minimus
- WIZ
- RAPIDFORT
- docker
- SEAL SECURITY
- ENDOR LABS
- Socket
- aikido
- VERACODE

AI Continuous Threat Modeling & Design Review

- Clover
- DEVARMOR
- Seezo
- prime security

Enterprise

Choosing a Platform

Application security in large enterprises looks significantly different due to the high number of legacy applications, diverse coding languages, deployment models, and development teams that have built and managed their own pipeline solutions over time. Organizations with these distributed environments often also have strict compliance requirements and long-standing vendor relationships, which require customizations of SBOMs and multiple scanning types as binaries are built and deployed.

These enterprises differ fundamentally from cloud-native startups and SaaS companies, where standardized developer platforms are core to the organization. They typically operate on more unified cloud-native architectures, with deployment processes built from the ground up or fully migrated using consistent, standardized approaches.

The first step of our buyer's guide is intended to highlight this distinction between sprawling developer environments and fully modernized ones. On one side, there are vendors that support a wide range of testing methods and workflow requirements, designed for large enterprises and their often complex compliance needs. On the other side are vendors that function well as all-in-one application security solutions, either through deep customization, strong extensibility into existing platforms, or broad coverage of the many use cases enterprises require.

Additionally, the vendors listed in the cloud native architecture section are not meant to be exhaustive. Nearly every company shown in the mid-market diagram can fit into an enterprise stack, with different trade-offs depending on organizational priorities.

Approaches to False Positive Reduction

Vulnerability prioritization and reduction is a massive challenge on its own for large enterprises. For this reason, static function-level reachability and AI-driven prioritization are important differentiators, but their maturity varies widely by vendor and language, and often depends on proprietary vulnerability databases that are difficult to assess externally. In general, static reachability requires longer CLI-based scans to produce strong results, particularly for statically typed languages.

Vendors have also taken different approaches to AI prioritization. Some build deep, proprietary indexes of customer codebases and run sophisticated prioritization logic, while others rely on far more superficial techniques, such as prompting general-purpose LLMs with a finding and its surrounding code.

The most effective reductions in false positives continue to come from combining runtime function-level reachability with cloud context. We've written previously about the potential unlocked by this combination, but it is difficult to fully appreciate without hands-on experience. When using these tools, every vulnerability investigation becomes genuinely interesting, reshaping how teams think about risk after years of being overwhelmed by noise.

Best in Class Solutions

Once a broad approach to vulnerability discovery and prioritization is in place, several additional options for expanding your security program are possible. The first is the traditional ASPM category, which focuses primarily on vulnerability management. These tools work well for organizations with large, distributed teams that need visibility into what is and is not covered across their existing scanner ecosystem.

The second option is adopting a dedicated software supply chain security provider. These tools typically offer more advanced reachability and license management, at the cost of introducing an additional vendor. In many cases, however, they deliver enough additional value, such as runtime protection, SBOM management, or developer autofix capabilities, to justify their place as standalone offerings.

The final option is investing in a dedicated dynamic scanning solution. While many application security platforms offer some dynamic scanning capabilities, these offerings are maturing due to acquisitions in the space. For teams that prioritize API-first scanning and modern web architectures, a standalone dynamic scanner is often still worth the investment.

Investing in Emerging Capabilities

For some teams, investing in point solutions that address emerging capabilities, or selecting a platform based on strong support for a specific capability, can be sound decisions, depending on organizational focuses. For example, open source supply chain attacks disproportionately target crypto businesses, making upstream malware detection and secure package registries especially important selection criteria for teams operating in that space.

Beyond supply chain risks, investments in application security scanning are introducing new considerations for how enterprises evaluate tools. A key consideration is business logic detections, one of the most impactful application security scanning developments available today. These tools unlock entirely new categories of findings, and typically provide strong false positive reduction methodologies as well.

At the same time, developing AI coding guardrails are also a major concern for enterprises, as they aggressively encourage developers to adopt AI-assisted coding. The vendors highlighted in this category have all taken meaningful steps toward providing management capabilities and guardrails for AI coding agents. That said, it is still early for this category, as frontier AI solutions are evolving so quickly that it can be difficult to determine what best-practice security architectures should look like.

On top of those points, continuous threat modeling and design review remain significant challenges for both security and compliance teams, making this a smart investment for organizations looking to reduce friction in these processes. While many teams attempt to build these capabilities in-house, vendors typically have an easier time integrating with the third-party tooling required to operate at scale.

Finally, robust runtime protection in cloud environments continues to be a major gap for many enterprises, as legacy EDR solutions provide little meaningful coverage in containerized environments. By extending protection into the application layer, newer providers have made substantial progress in delivering effective runtime detection and protection for enterprise applications. When combined with strong false-positive reduction for vulnerabilities, CADR remains a clear choice for modern enterprise cloud workloads.

Concluding Thoughts

Application security is a discipline in crisis. Developer workflows are changing rapidly and the reality is that we are in a period of transition. Traditional scanning methodologies still have a place, but things are quickly evolving.

There are bold promises from nearly every vendor around securing AI-generated code, but in practice much of the heavy lifting is still being done by frontier models. In the meantime, there are also exciting developments that are helping teams reduce backlogs and uncover more meaningful vulnerabilities.

For all the approaches to securing AI-generated code, the real security benefits are still emerging. Looking ahead to practitioner tool roadmaps for the coming year, robust runtime protection remains a key priority, as the AI-driven SDLC is still taking shape. By this time next year, it should be much clearer what this new SDLC looks like, and which vendors are doing the best job securing it.

Application Security Vendor Map

The below map indicates the vendors focused on specific categories of scanning, versus which provide bundled options. This is designed to help organizations think through coverage, but it doesn't indicate scanner maturity.

SCA + SAST + Secrets + Containers + 3rd Party Management + DAST

- aikido
- snyk
- WIZ
- Checkmarx
- invicti
- ox
- fluidattacks
- nullify
- soos
- jit
- harness
- GitLab
- BLACKDUCK
- VERACODE
- dynatrace

SCA + SAST + Secrets+ Containers + 3rd Party Management

- cycode
- apiiro
- LEGIT
- DATADOG
- paloalto NETWORKS
- boost security.io

SCA + SAST + Secrets + Containers

- ZEROPATH
- ENDOR LABS
- [arnica]
- Kodem
- CORGEA
- Socket
- FOSSA
- orca SECURITY
- Mend.io
- xygeni.
- START LEFT
- RAINFOREST.TECH

SCA + SAST + Secrets

- Semgrep
- Codacy
- GitHub
- HEELER
- BACKSLASH
- CONTRAST SECURITY
- Sonar

SCA

- Hopper
- LINEAJE
- sonatype
- oligo
- MIGGO
- RAVEN

SAST

- DRYRUN SECURITY
- amplify
- opentext

Secrets

- TRUFFLEHOG
- GitGuardian

Containers

CNAPP & Supply Chain tools

Third Party Management

- konvu
- CROWDSTRIKE
- PHOENIX SECURITY
- ArmorCode

CTEM

DAST and API Security

- LEVO
- AppSentinels
- akto
- PortSwigger
- escape
- STACKHAWK
- Qualys.
- tenable
- Bright
- crunch
- intruder
- pynt
- ZAP by Checkmarx

ADR

- RAVEN
- MIGGO
- Kodem
- REIN
- CONTRAST SECURITY
- oligo
- upwind
- dynatrace
- DATADOG

VENDOR SPOTLIGHTS

All vendors were given the opportunity to spotlight their product by having the Latio team author a dedicated page explaining why they were awarded in this report.

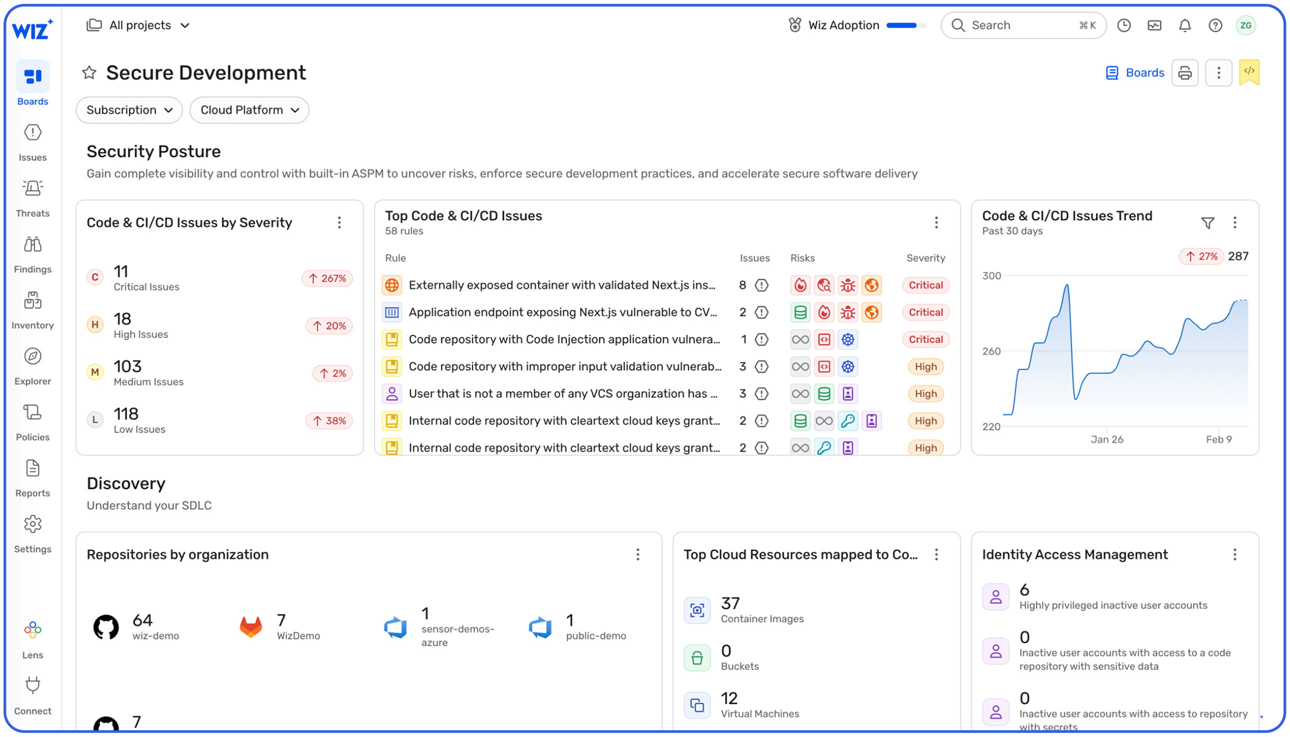
Latio APPLICATION SECURITY PLATFORM LEADER 2026

Latio APPLICATION SECURITY MANAGEMENT LEADER 2026

Latio APPLICATION SECURITY AI PENTESTING INNOVATOR 2026

Latio APPLICATION SECURITY RUNTIME INNOVATOR 2026

Wiz has quickly evolved beyond cloud security to become a leader in the application security market as well. While its core code-to-cloud capabilities remain as strong as ever, the ability to ingest findings from other tools has also made Wiz a centralized vulnerability management platform. At the same time, they have rapidly built and expanded their in-house scanning capabilities to a level that now rivals many dedicated application security providers.



Secure Development

Security Posture

Gain complete visibility and control with built-in ASPM to uncover risks, enforce secure development practices, and accelerate secure software delivery

Code & CI/CD Issues by Severity

C	11 Critical Issues	↑ 267%
H	18 High Issues	↑ 20%
M	103 Medium Issues	↑ 2%
L	118 Low Issues	↑ 38%

Top Code & CI/CD Issues

58 rules

Rule	Issues	Risks	Severity
Externally exposed container with validated Next.js ins...	8		Critical
Application endpoint exposing Next.js vulnerable to CV...	2		Critical
Code repository with Code Injection application vulnera...	1		Critical
Code repository with improper input validation vulnerab...	3		High
User that is not a member of any VCS organization has ...	3		High
Internal code repository with cleartext cloud keys grant...	2		High
Internal code repository with cleartext cloud keys grant...	2		High

Code & CI/CD Issues Trend

Past 30 days

↑ 27% 287

Discovery

Understand your SDLC

Repositories by organization

64 wiz-demo

7 WizDemo

1 sensor-demos-azure

1 public-demo

7

Top Cloud Resources mapped to Co...

37 Container Images

0 Buckets

12 Virtual Machines

Identity Access Management

6 Highly privileged inactive user accounts

0 Inactive user accounts with access to a code repository with sensitive data

0 Inactive user accounts with access to repository with secrets

Beyond the scanning capabilities themselves, Wiz continues to lead in contextualizing and prioritizing findings across the SDLC. Whether issues are tied to infrastructure resources or source code repositories, the Wiz graph provides an elegant way to visualize relationships and drive remediation workflows. This ensures the most relevant issues are routed to the right place with context on what needs to be fixed and why.

On the runtime side, Wiz has significantly matured its attack surface management capabilities to include AI-driven penetration testing, along with API discovery and runtime function-level reachability. Bringing all of this data together in a single platform positions Wiz as the most complete code-to-cloud offering on the market, giving both security and engineering teams a single place to reduce risk and prioritize.

The Benefits of Wiz

Centralized Visibility

Manage application and cloud security findings across code, infrastructure, and third-party tools.

Developer Context

Provide actionable context through graph-based relationships that link vulnerabilities to real-world impact.

End-To-End Coverage

Combine static, runtime, and attack surface insights for comprehensive code-to-cloud visibility.

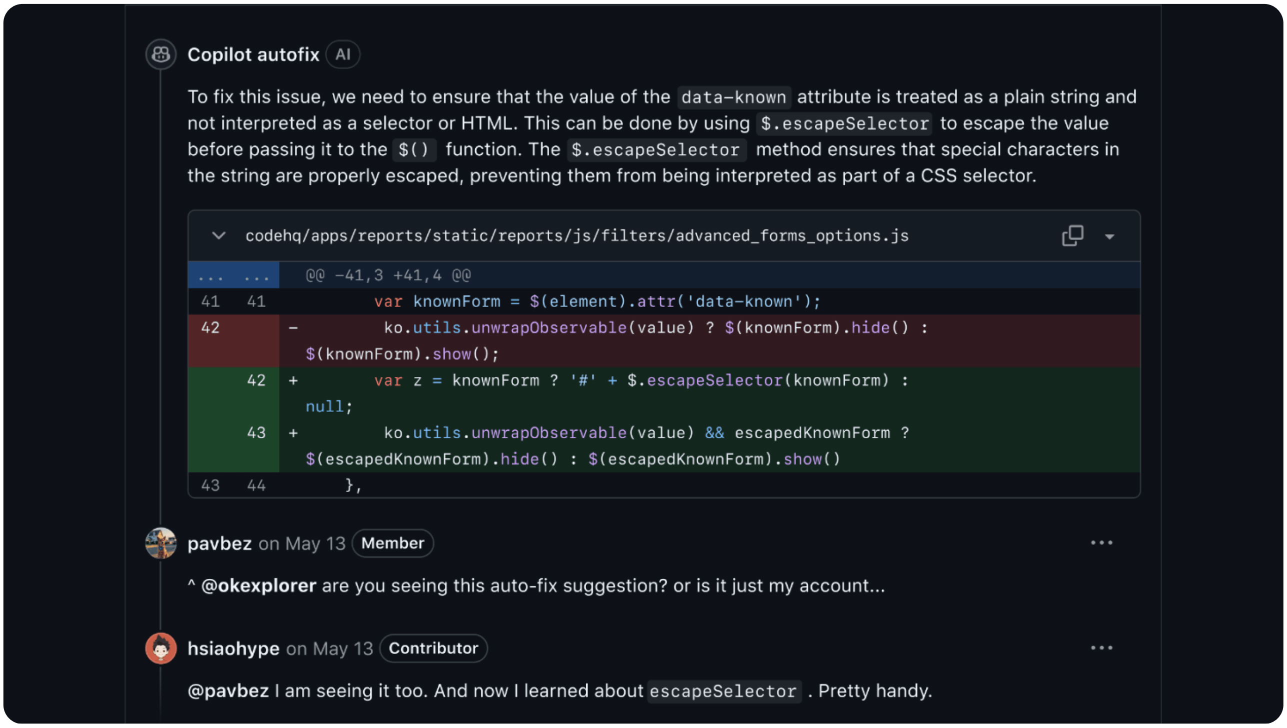
GitHub

Latio
APPLICATION SECURITY
PLATFORM LEADER
2026

Latio
APPLICATION SECURITY
AI CODE INNOVATOR
2026

GitHub Advanced Security has quickly become a strong competitor across traditional application security categories, particularly secret scanning, SAST, and SCA. While many tools attempt to be "developer-friendly," GitHub is the only platform developers actually love, adopt, and use on their own! This organic expansion into security significantly increases developer adoption, and empowers them to directly customize workflows without friction. The result: tools are actually operationalized, preventing leaks and security incidents, and reducing risk across application portfolios of all sizes.

On the scanning side, GitHub has implemented leading features such as incremental scans, code quality scanning and out of the box push protection for secrets. GitHub has also invested in scalable remediation workflows – combining security campaigns with GitHub Copilot to make the remediation process as seamless and automated as possible. Copilot is one of the few AI coding tools broadly trusted for enterprise development, and its ability to generate large volumes of remediation pull requests materially accelerates vulnerability fixes.



To fix this issue, we need to ensure that the value of the `data-known` attribute is treated as a plain string and not interpreted as a selector or HTML. This can be done by using `$.escapeSelector` to escape the value before passing it to the `$()` function. The `$.escapeSelector` method ensures that special characters in the string are properly escaped, preventing them from being interpreted as part of a CSS selector.

codehq/apps/reports/static/reports/js/filters/advanced_forms_options.js

```diff
@@ -41,3 +41,4 @@
         var knownForm = $(element).attr('data-known');
-         ko.utils.unwrapObservable(value) ? $(knownForm).hide() : $(knownForm).show();
+        var z = knownForm ? '#' + $.escapeSelector(knownForm) : null;
+         ko.utils.unwrapObservable(value) && escapedKnownForm ? $(escapedKnownForm).hide() : $(escapedKnownForm).show()
      },
```

pavbez on May 13 Member

^ **@okexplorer** are you seeing this auto-fix suggestion? or is it just my account...

hsiaohype on May 13 Contributor

@pavbez I am seeing it too. And now I learned about `escapeSelector` . Pretty handy.

GitHub is also embedding security directly into the code generation process by scanning code as it is produced by AI agents. This adds an additional layer of protection during AI adoption itself. Combined with GitHub's close integration with Microsoft Defender for Cloud to analyze and establish runtime risk, GitHub stands out as a strong option for enterprises looking to scale AI-driven development without compromising security.

The Benefits of GitHub

Native Developer Experience

Built directly into developer workflows developers already trust and use daily.

Deep Rule Customization

Enables teams to write, extend, and maintain custom security logic with CodeQL.

AI-Driven Remediation

Uses Copilot agents and campaigns to accelerate fixes at enterprise scale.

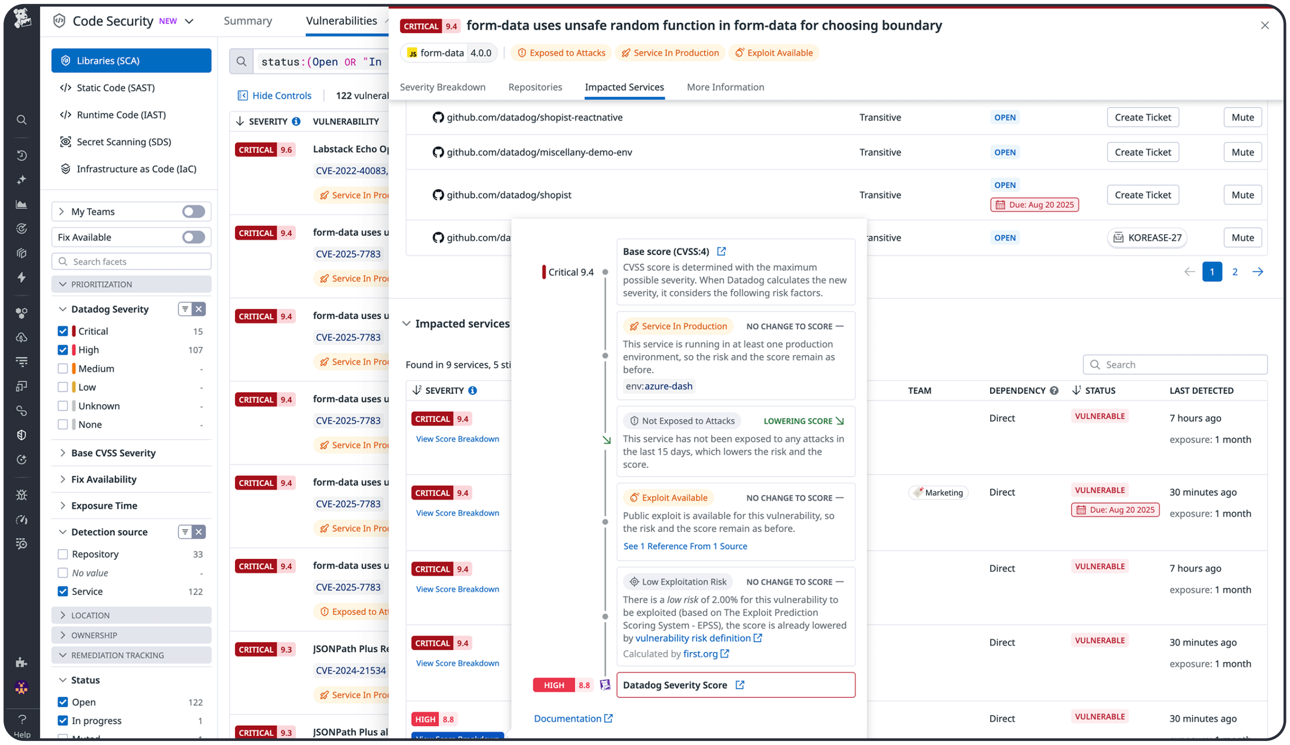
DATADOG

Latio APPLICATION SECURITY PLATFORM LEADER 2026

Latio APPLICATION SECURITY RUNTIME INNOVATOR 2026

Datadog's Application Security offering covers the full set of scanning capabilities teams expect, alongside leading runtime protection features. On the static analysis side, Datadog supports multiple implementation models, ranging from IDE through to runtime scanning. After integrating source code repositories, teams can track and manage SAST, SCA, Code Quality, Secrets, and IaC findings across their environments, with flexibility in how scans are configured and maintained.

Datadog's leading agentic capabilities deliver benefits for application security workflows by giving teams accurate remediation guidance and prioritization. These features pair well with Datadog's broader Bits AI Agents offering, helping to prioritize and remediate security issues.



status:(Open OR "In

SEVERITY	VULNERABILITY
CRITICAL 9.6	Labstack Echo O CVE-2022-40083, Service In Pro
CRITICAL 9.4	form-data uses u CVE-2025-7783 Service In Pro
CRITICAL 9.4	form-data uses u CVE-2025-7783 Service In Pro
CRITICAL 9.4	form-data uses u CVE-2025-7783 Service In Pro
CRITICAL 9.4	form-data uses u CVE-2025-7783 Service In Pro
CRITICAL 9.4	form-data uses u CVE-2025-7783 Exposed to At
CRITICAL 9.3	JSONPath Plus Re CVE-2024-21534 Service In Pro
CRITICAL 9.3	JSONPath Plus al

CRITICAL 9.4 form-data uses unsafe random function in form-data for choosing boundary

form-data 4.0.0 | Exposed to Attacks | Service In Production | Exploit Available

github.com/datadog/shopist-reactnative	Transitive	OPEN	Create Ticket	Mute
github.com/datadog/miscellany-demo-env	Transitive	OPEN	Create Ticket	Mute
github.com/datadog/shopist	Transitive	OPEN Due: Aug 20 2025	Create Ticket	Mute
github.com/da	ansitive	OPEN	KOREASE-27	Mute

SEVERITY	TEAM	DEPENDENCY	STATUS	LAST DETECTED
CRITICAL 9.4 View Score Breakdown		Direct	VULNERABLE	7 hours ago exposure: 1 month
CRITICAL 9.4 View Score Breakdown	Marketing	Direct	VULNERABLE Due: Aug 20 2025	30 minutes ago exposure: 1 month
CRITICAL 9.4 View Score Breakdown		Direct	VULNERABLE	7 hours ago exposure: 1 month
CRITICAL 9.4 View Score Breakdown		Direct	VULNERABLE	30 minutes ago exposure: 1 month
HIGH 8.8		Direct	VULNERABLE	30 minutes ago

Critical 9.4

Base score (CVSS:4)
CVSS score is determined with the maximum possible severity. When Datadog calculates the new severity, it considers the following risk factors.

Service In Production — NO CHANGE TO SCORE —
This service is running in at least one production environment, so the risk and the score remain as before.
env:azure-dash

Not Exposed to Attacks — LOWERING SCORE
This service has not been exposed to any attacks in the last 15 days, which lowers the risk and the score.

Exploit Available — NO CHANGE TO SCORE —
Public exploit is available for this vulnerability, so the risk and the score remain as before.
See 1 Reference From 1 Source

Low Exploitation Risk — NO CHANGE TO SCORE —
There is a *low risk* of 2.00% for this vulnerability to be exploited (based on The Exploit Prediction Scoring System - EPSS), the score is already lowered by vulnerability risk definition
Calculated by first.org

HIGH 8.8 Datadog Severity Score

The value of their scanning capabilities is amplified by Datadog's access to runtime telemetry. By correlating static findings with runtime behavior and threat detection signals, teams gain deeper visibility into how applications actually operate in production. Beyond scanning and detection, Datadog has contributed to research focused on areas such as malicious IDE extensions and software supply chain threats. From leading threat research to scanning capabilities, Datadog is a complete application security platform.

The Benefits of Datadog

Work With Developers

Let developers work with the tools they're already the most comfortable with.

Runtime Context

Correlate security findings with real-world application behavior using native telemetry.

Detection Engineering

Benefit from ongoing detection research across cloud and application attack surfaces.

aikido

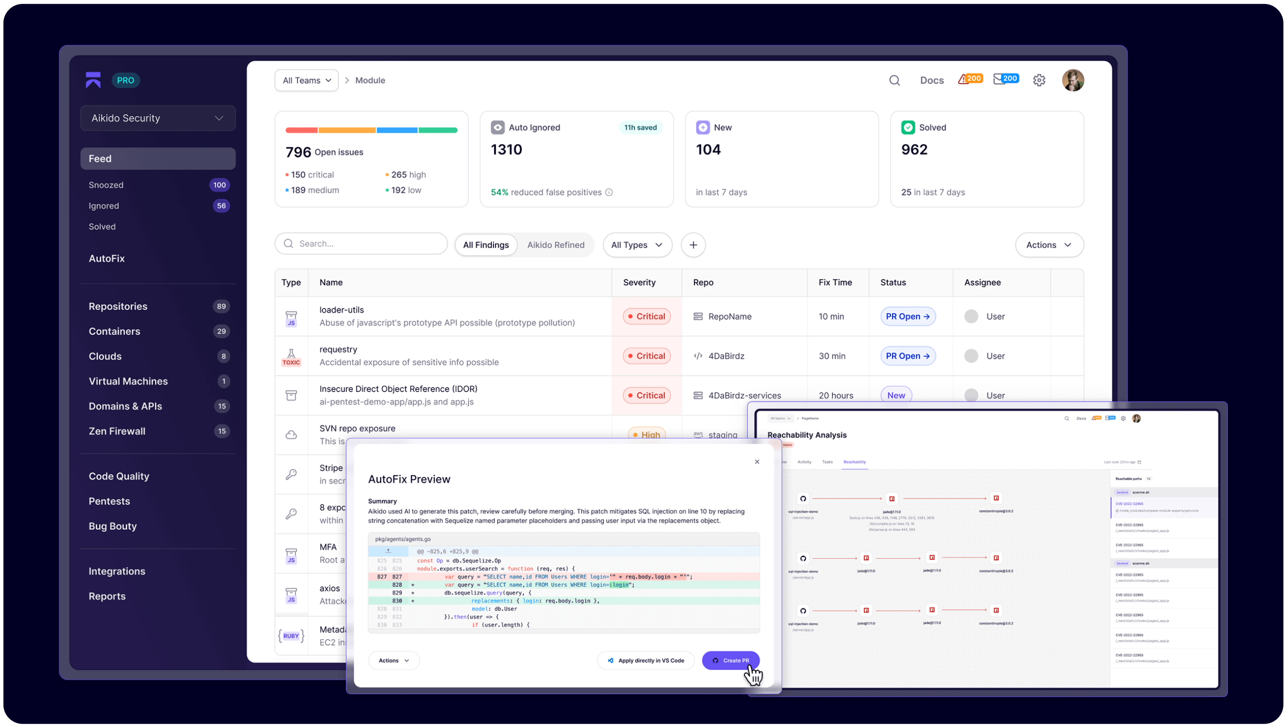
Latio APPLICATION SECURITY PLATFORM LEADER 2026

Latio APPLICATION SECURITY SUPPLY CHAIN INNOVATOR 2026

Latio APPLICATION SECURITY AI PENTESTING INNOVATOR 2026

Since their founding in 2022, Aikido has evolved from a simple bundled application security offering into a platform with leading capabilities across several categories. From robust SAST customization, to open-source malware research, to leading innovations in AI Pentesting, Aikido focuses on delivering features that matter most, improving developer experience while also strengthening real-world security outcomes.

From a scanning perspective, Aikido provides strong reachability analysis, flexible SAST customization, and a highly robust autofix architecture. These capabilities are designed to reduce developer noise through a large set of handcrafted and sensible mitigations for some of the most common vulnerability classes. When combined with the Zen runtime module and cloud capabilities, Aikido is able to add meaningful organizational and runtime context, helping teams significantly reduce false positives - expanding even to proactive protection.



796 Open issues

150 critical · 265 high · 189 medium · 192 low

Auto Ignored 11h saved: 1310 — 54% reduced false positives

New: 104 in last 7 days

Solved: 962 — 25 in last 7 days

Type	Name	Severity	Repo	Fix Time	Status	Assignee
JS	loader-utils Abuse of javascript's prototype API possible (prototype pollution)	Critical	RepoName	10 min	PR Open →	User
TOXIC	requestry Accidental exposure of sensitive info possible	Critical	4DaBirdz	30 min	PR Open →	User
	Insecure Direct Object Reference (IDOR) ai-pentest-demo-app/app.js and app.js	Critical	4DaBirdz-services	20 hours	New	User
	SVN repo exposure This is	High	staging			
	Stripe in secr					
	8 expo within					
JS	MFA Root a					
JS	axios Attack					
RUBY	Metada EC2 in					

AutoFix Preview

Summary

Aikido used AI to generate this patch, review carefully before merging. This patch mitigates SQL injection on line 10 by replacing string concatenation with Sequelize named parameter placeholders and passing user input via the replacements object.

pkg/agents/agents.go

```
@@ -825,6 +825,9 @@
  const Op = db.Sequelize.Op
  module.exports.userSearch = function (req, res) {
          var query = "SELECT name,id FROM Users WHERE login='" + req.body.login + "'";
+         var query = "SELECT name,id FROM Users WHERE login=:login";
+         db.sequelize.query(query, {
+                 replacements: { login: req.body.login },
                  model: db.User
          }).then(user => {
                  if (user.length) {
```

Aikido has also been at the forefront of several emerging capabilities: AI penetration testing, attack surface management, code quality analysis, and AI code security. The platform is a strong offering for companies of any size looking to improve their developer experience, false positive rates, and adoption of the latest technologies.

The Benefits of Aikido

Scalable Pricing

Aikido provides clear pricing and bundling options for companies of every size.

Contextual Prioritization

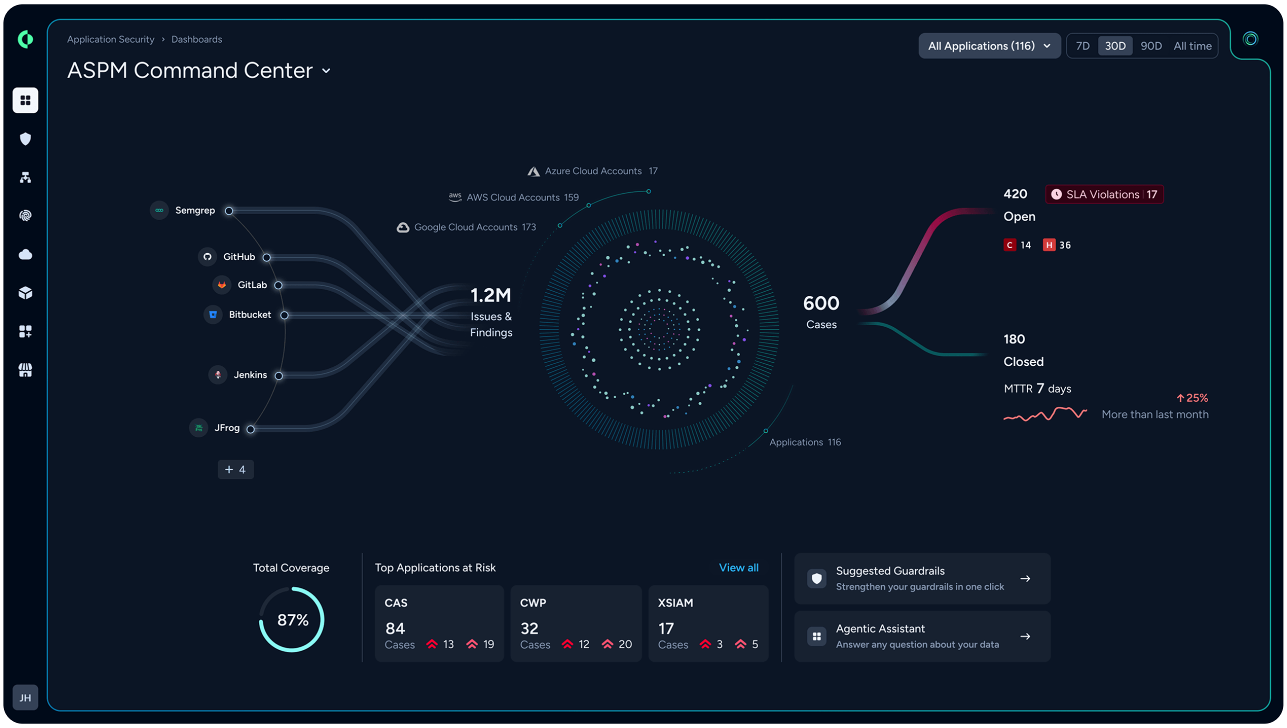
Improve prioritization with runtime and organizational context, alongside strong reachability capabilities

AI-Driven Coverage

Apply AI-based scanning for both security and code quality use cases.

Palo Alto Networks Cortex Cloud is a strong fit for security teams looking to extract more value from their existing tools while also adding robust ASPM, software supply chain and cloud security capabilities. With Cortex Cloud, teams can build a clearer picture of how vulnerabilities enter their environment and where exploitable paths could lead to real-world impact.

Cortex Cloud natively supports IaC, SCA and secret scanning, alongside the broader set of ASPM, software supply chain and cloud security capabilities expected from a CNAPP platform. In addition, it can ingest results through native integrations with a wide range of third-party scanners, allowing organizations to improve prioritization without disrupting their existing development tools and processes.



As with other areas of Cortex Cloud, its primary strength lies in the Command Center making findings actionable. Teams can also assess security coverage across their codebase, gaining visibility into any gaps. The AI Guardrails automatically suggest policies tailored to your environment to stop new risks before they reach production.

The flexibility and configurability of Palo Alto Networks application security capabilities make Cortex Cloud a compelling option for AppSec teams seeking preventing vulnerabilities getting to production, better prioritization and supply chain.

The Benefits of Palo Alto Networks Cortex Cloud:

Prevention-First

Enforce AI generated policies that target and block risks based on your overall infrastructure context.

Unified Visibility

Centralize findings from native and third-party scanners to improve prioritization, analyze coverage gaps, and accelerate remediation from a single interface.

AI Prioritization and Remediation

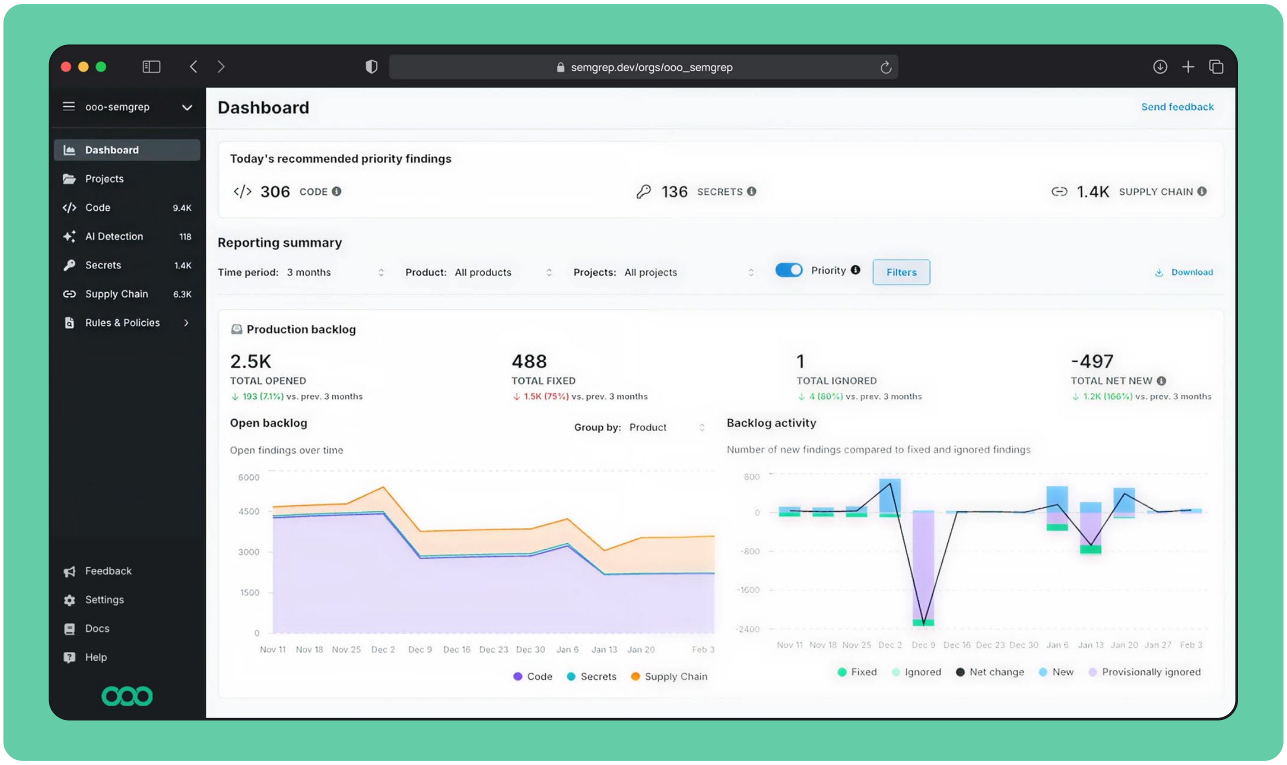
Enables prioritization, coverage analysis, and remediation from a single interface.

Semgrep

Latio
APPLICATION SECURITY
TESTING
LEADER
2026

Semgrep is one of the most developer-oriented platforms on the market. They pioneered making security scanning accessible with their open source engine, which remains highly distributed and relied on by organizations of all sizes. Semgrep has expanded to offer the cloud-native integrations along with robust SCA and secret detection capabilities that practitioners expect from a modern application security platform.

Semgrep has also architected AI-driven detection, prioritization and remediation in ways that work consistently for enterprise environments. The engineering thoughtfulness behind these features delivers a set of AI capabilities that teams can adopt with confidence - whether detecting business logic flaws with AI SAST, or driving remediation with autofixes. Beyond AI, Semgrep's secret detection remains a standout feature, particularly due to its semantic analysis approach, which goes well beyond traditional pattern searching methods.



Dashboard

Today's recommended priority findings

306 CODE · 136 SECRETS · 1.4K SUPPLY CHAIN

Reporting summary

Time period: 3 months · Product: All products · Projects: All projects · Priority

Production backlog

2.5K TOTAL OPENED ↓ 193 (7.1%) vs. prev. 3 months

488 TOTAL FIXED ↓ 1.5K (75%) vs. prev. 3 months

1 TOTAL IGNORED ↓ 4 (80%) vs. prev. 3 months

-497 TOTAL NET NEW ↓ 1.2K (166%) vs. prev. 3 months

Open backlog

Group by: Product

Open findings over time

Backlog activity

Number of new findings compared to fixed and ignored findings

Semgrep is especially well suited for developer-first organizations that encourage teams to actively optimize and maintain the scanning engine they're using. The maturity of the scanning engine, combined with the flexibility of its rule customization, makes Semgrep an enterprise ready application testing solution. At the same time, its distributed and open source roots also make it a strong fit for mid-market organizations that often begin their journey with the open source offering before expanding further.

The Benefits of Semgrep

Developer Native

Deliver a truly developer-first security experience with fast, local, and CI-friendly scanning.

Customize Everything

Reduce noise with semantic analysis and highly customizable rule sets.

Enterprise-Ready AI Remediation

Utilize AI-driven prioritization and remediation with confidence in large environments.

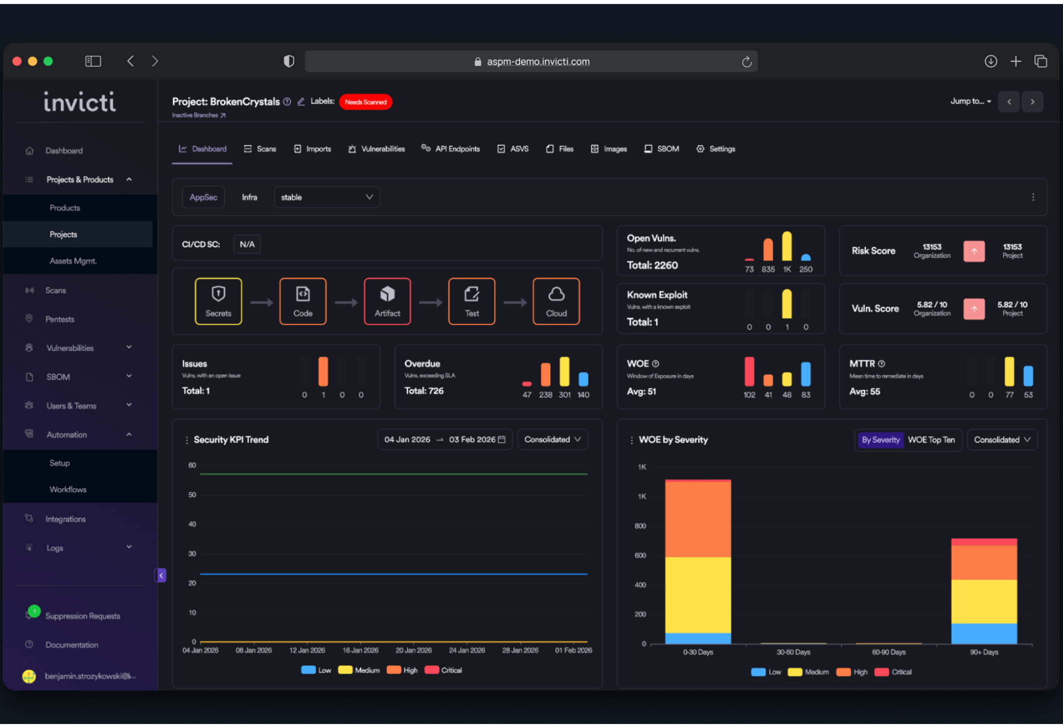
invicti

Latio APPLICATION SECURITY TESTING LEADER 2026

Latio APPLICATION SECURITY DAST INNOVATOR 2026

Invicti has long been known for its robust DAST solutions, which have enabled organizations of all sizes to perform in-depth testing of running applications. The platform has remained competitive in the dynamic testing space by expanding AI capabilities from their Acunetix and Netsparker lineage and has more recently expanded into a broader AppSec platform strategy with its acquisition of ASPM from Kondukto.

Today, Invicti provides a comprehensive set of dynamic testing features, alongside API and LLM discovery, LLM integration testing, and developer-oriented API testing and scanning. Recent enhancements include AI-assisted testing aimed at identifying business logic issues and improving scan context. These additions to the platform expand coverage and reduce false positives, improving everything from initial integration to scan quality.


aspm-demo.invicti.com
invicti
Project: BrokenCrystals
Labels: Needs Scanned
Inactive Branches
Jump to...
Dashboard
Projects & Products
Products
Projects
Assets Mgmt.
Scans
Pentests
Vulnerabilities
SBOM
Users & Teams
Automation
Setup
Workflows
Integrations
Logs
Suppression Requests
Documentation
Dashboard · Scans · Imports · Vulnerabilities · API Endpoints · ASVS · Files · Images · SBOM · Settings
AppSec · Infra · stable
CI/CD SC: N/A
Secrets → Code → Artifact → Test → Cloud
Open Vulns. No. of new and recurrent vulns. Total: 2260 — 73 835 1K 250
Known Exploit Vulns. with a known exploit Total: 1 — 0 0 1 0
Risk Score 13153 Organization 13153 Project
Vuln. Score 5.82 / 10 Organization 5.82 / 10 Project
Issues Vulns. with an open issue Total: 1 — 0 1 0 0
Overdue Vulns. exceeding SLA Total: 726 — 47 238 301 140
WOE Window of Exposure in days Avg: 51 — 102 41 48 83
MTTR Mean time to remediate in days Avg: 55 — 0 0 77 53
Security KPI Trend
04 Jan 2026 → 03 Feb 2026
Consolidated
60 50 40 30 20 10 0
04 Jan 2026 08 Jan 2026 12 Jan 2026 16 Jan 2026 20 Jan 2026 24 Jan 2026 28 Jan 2026 01 Feb 2026
Low Medium High Critical
WOE by Severity
By Severity · WOE Top Ten · Consolidated
1K 1K 800 600 400 200 0
0-30 Days 30-60 Days 60-90 Days 90+ Days
Low Medium High Critical

With the integration of its orchestration capabilities, Invicti can now coordinate existing scanners, deploy open-source scanners, or Invicti-supplied static AST tools across development environments. This allows the platform to extend beyond standalone DAST into broader application security workflows. As a result, Invicti is well suited for mid size enterprises looking for all-in-one AppSec platforms, or for enterprises that prioritize a dedicated DAST solution and want flexibility in how they manage and evolve their existing application security tooling.

The Benefits of Invicti

Discover Undocumented API and LLM

Multi-layered approach to discovery, including during scans, source code analysis, gateway integrations, and network traffic analysis.

API and AI-Assisted Testing

Test APIs and leverage AI-based techniques to improve context and identify complex issues.

Tool Orchestration

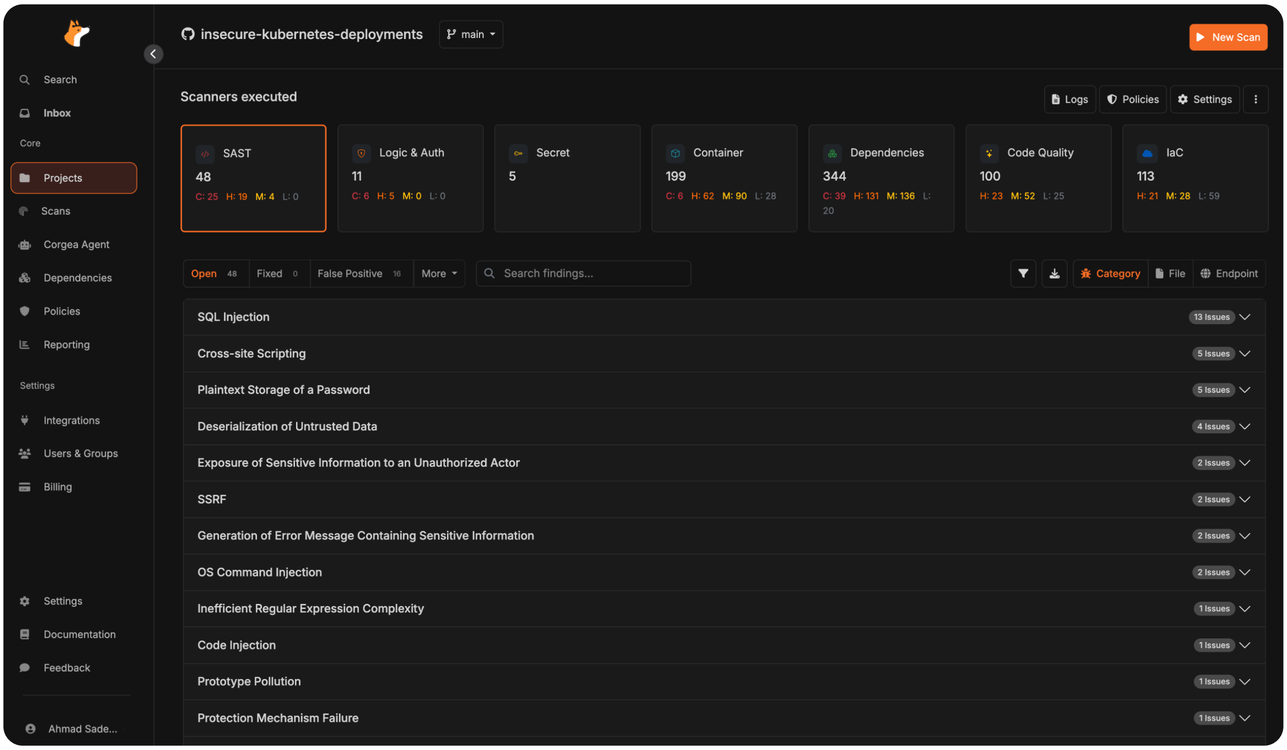
Coordinate existing and open-source scanners to support broader application security workflows.

CORGEA

Latio
APPLICATION SECURITY
AI PLATFORM INNOVATOR
2026

Corgea has built a strong AI-first approach to application security, using AI across vulnerability discovery, prioritization, and remediation. While many vendors claim to use AI for better prioritization, Corgea has delivered real differentiation in how effective these capabilities are across each of those areas.

As AI coding becomes more common, developer expectations for security tools are rising, both in terms of the relevance of findings and how teams interact with them. Corgea's AI-driven SAST engine consistently surfaces findings that are practical and meaningful. I have seen firsthand the value of these discoveries, particularly in identifying business logic flaws that lead to real-world attacks, rather than the typical volume of low-value false positives produced by traditional tools.



insecure-kubernetes-deployments

Scanners executed

SAST	48	C: 25 H: 19 M: 4 L: 0
Logic & Auth	11	C: 6 H: 5 M: 0 L: 0
Secret	5	
Container	199	C: 6 H: 62 M: 90 L: 28
Dependencies	344	C: 39 H: 131 M: 136 L: 20
Code Quality	100	H: 23 M: 52 L: 25
IaC	113	H: 21 M: 28 L: 59

SQL Injection	13 Issues
Cross-site Scripting	5 Issues
Plaintext Storage of a Password	5 Issues
Deserialization of Untrusted Data	4 Issues
Exposure of Sensitive Information to an Unauthorized Actor	2 Issues
SSRF	2 Issues
Generation of Error Message Containing Sensitive Information	2 Issues
OS Command Injection	2 Issues
Inefficient Regular Expression Complexity	1 Issues
Code Injection	1 Issues
Prototype Pollution	1 Issues
Protection Mechanism Failure	1 Issues

On the workflow side, the Corgea team has invested heavily in modern, AI-first developer experiences. Developers can interact directly with pull request comments through chat-based workflows that feel natural and intuitive. This creates reviews that resemble thoughtful, informed security feedback rather than rigid pipeline blocks that disrupt development.

Overall, Corgea is one of the more exciting companies in application security as AI reshapes what is possible across detection, prioritization, and remediation.

The Benefits of Corgea:

AI-Driven Discovery

Identifies meaningful vulnerabilities, including business logic flaws, that traditional scanners often miss.

Relevant Prioritization

Uses AI context to reduce noise and surface issues that pose real security risk.

Developer-Native Reviews

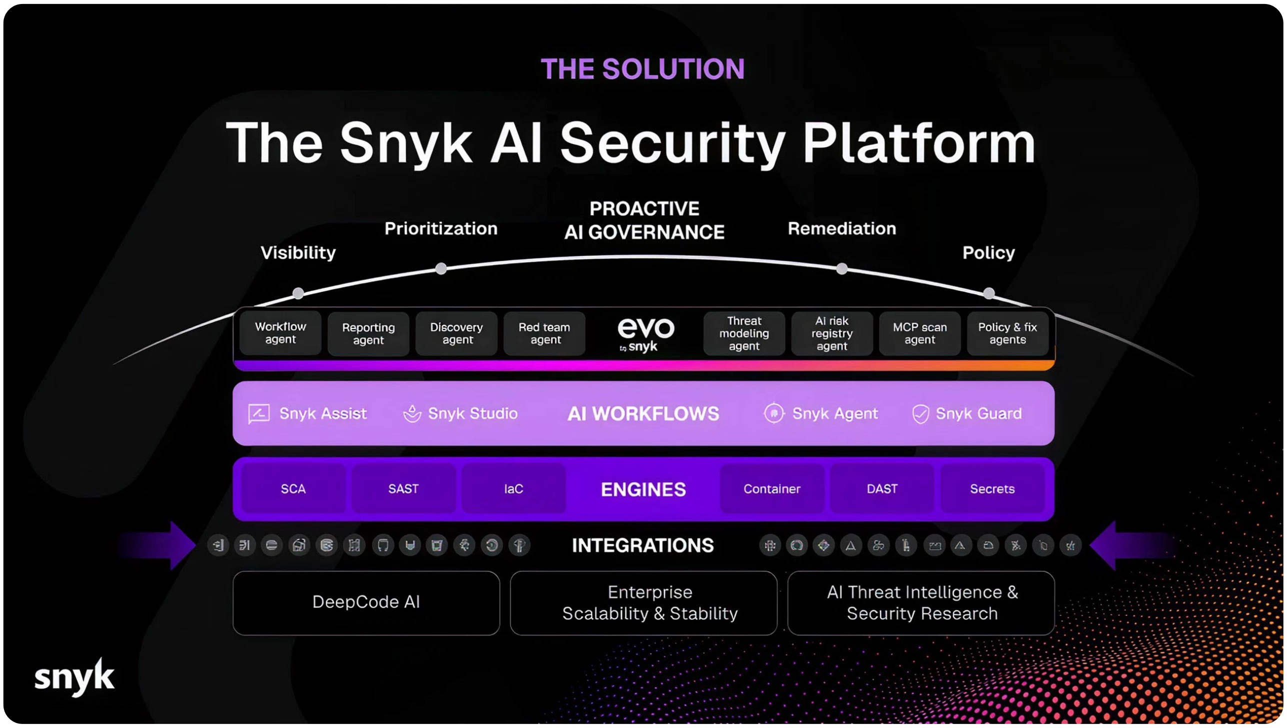
Delivers conversational, PR-based feedback that aligns with modern developer workflows.

Latio APPLICATION SECURITY TESTING LEADER 2026

Latio APPLICATION SECURITY AI CODE INNOVATOR 2026

Snyk pioneered the modern DevSecOps experience by giving developers immediate access to security findings across their code base. Most recently, Evo by Snyk brings many of the capabilities of dedicated AI security tools into the context of Snyk's broader ecosystem, creating a developer first approach to securing AI-Native applications, while giving security engineers visibility into the new wave of AI tools.

Snyk's newest capabilities give teams visibility into developer endpoints, including which MCPs and coding agents are in use, and the ability to track and enforce standards across those endpoints. This visibility forms the foundation for enforcing guardrails on AI generated code. Using Snyk's MCP for agents, coding tools can retrieve additional security context to generate higher-quality code by default and to produce more effective remediations.


THE SOLUTION

The Snyk AI Security Platform

The broader Snyk platform remains a flexible option for organizations managing a wide range of languages, frameworks, and deployment models at enterprise scale. Across its product lines, Snyk has built a robust and continually evolving knowledge base of vulnerabilities and findings that support consistent security outcomes across diverse environments. These capabilities combined with AI capabilities help enterprises looking to safely adopt AI tooling.

Overall, Snyk continues to be a strong choice for enterprises seeking a mature DevSecOps platform, while actively preparing for widespread adoption of AI-assisted code generation.

The Benefits of Snyk

AI-Aware Guardrails

Applies security standards directly to AI-driven code generation workflows.

Developer Endpoint Visibility

Tracks agents, tools, and usage patterns to enforce consistent security controls.

Enterprise Platform

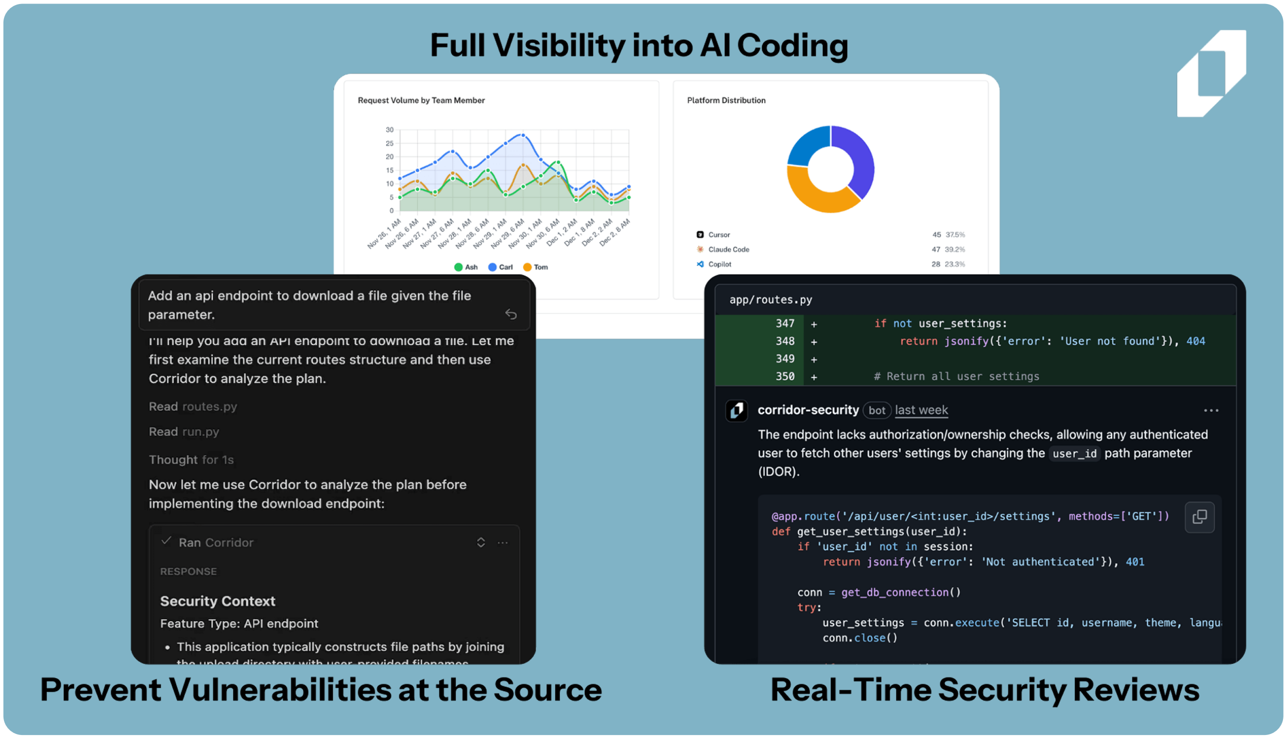
Supports broad language coverage and scalable security management across teams.

Corridor

Latio
APPLICATION SECURITY
AI CODE INNOVATOR
2026

It is increasingly clear that AI coding will play a major role in the future of software development. While AI-generated code introduces new risks, it also introduces new opportunities to secure code by default, in ways that shift-left always promised but rarely delivered. Corridor is a leader in improving the quality of AI-generated code by giving security teams visibility, governance, and guardrails before and after code is created.

Corridor begins by learning an organization's codebase and environment to understand which security controls should be applied alongside existing standards. It uses AI agents to generate security context and guardrails automatically, drawing upon both pre-generated packs as well as building custom context based upon a customer's codebase and documentation. It then integrates directly into the most popular AI coding tools at both the planning and code generation stages.



Full Visibility into AI Coding

Cursor	45	37.5%
Claude Code	47	39.2%
Copilot	28	23.3%

Add an api endpoint to download a file given the file parameter.

I'll help you add an API endpoint to download a file. Let me first examine the current routes structure and then use Corridor to analyze the plan.

Read routes.py

Read run.py

Thought for 1s

Now let me use Corridor to analyze the plan before implementing the download endpoint:

✓ Ran Corridor

RESPONSE

Security Context

Feature Type: API endpoint

- This application typically constructs file paths by joining the upload directory with user-provided filenames

Prevent Vulnerabilities at the Source

app/routes.py

```
+         if not user_settings:
+             return jsonify({'error': 'User not found'}), 404
+
+         # Return all user settings
```

corridor-security bot last week

The endpoint lacks authorization/ownership checks, allowing any authenticated user to fetch other users' settings by changing the `user_id` path parameter (IDOR).

```
@app.route('/api/user/<int:user_id>/settings', methods=['GET'])
def get_user_settings(user_id):
    if 'user_id' not in session:
        return jsonify({'error': 'Not authenticated'}), 401

    conn = get_db_connection()
    try:
        user_settings = conn.execute('SELECT id, username, theme, langu
        conn.close()
```

Real-Time Security Reviews

Corridor enables agents to make better decisions around libraries, frameworks, and coding patterns that align with the organization's security postures and scans code as it is created to prevent flaws at the source. Corridor also provides a pull-request reviewer as a final check to make sure code is secure and compliant before it is merged.

When these capabilities are unified, the platform delivers an end-to-end improvement over traditional scanning approaches. Organizational context is used both to secure code as it is generated and to provide developers with more meaningful, actionable feedback.

The Benefits of Corridor

Secure-By-Default Generation

Applies organizational security standards directly within AI code generation workflows.

Contextual Code Reviews

Replaces noisy findings with AI-driven reviews that understand code intent and usage.

Deep AI Tooling Integration

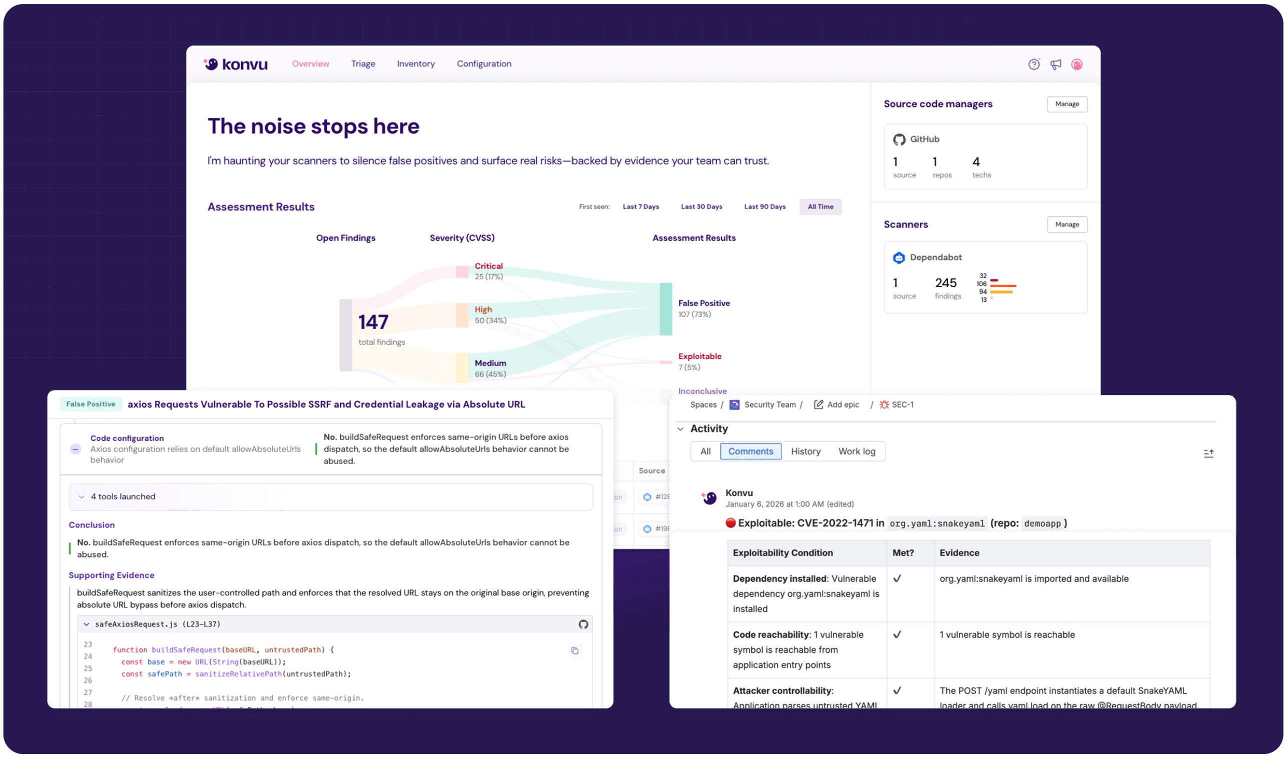
Native tool integration allows developers to scan code and fix vulns from their coding agent.

konvu

Latio
APPLICATION SECURITY
SUPPLY CHAIN INNOVATOR
2026

While most vendors are just beginning to apply AI to enterprise-scale security problems, Konvu was built from the ground up to address SCA backlogs for security teams. The platform delivers two key outcomes: vulnerability prioritization and remediation, and is well positioned to solve both effectively.

Security teams have long struggled to make SCA backlogs shrink instead of grow. One common approach is reachability analysis - there are multiple types of reachability, and most vendors support only a subset of them. Another approach is AI-driven code analysis which introduces another powerful layer of false-positive reduction. Konvu stands out by combining all aspects of reachability with AI-based prioritization, resulting in some of the most robust false-positive reduction on the market.



The noise stops here

I'm haunting your scanners to silence false positives and surface real risks—backed by evidence your team can trust.

False Positive axios Requests Vulnerable To Possible SSRF and Credential Leakage via Absolute URL

Code configuration
Axios configuration relies on default allowAbsoluteUrls behavior

No. buildSafeRequest enforces same-origin URLs before axios dispatch, so the default allowAbsoluteUrls behavior cannot be abused.

Conclusion

No. buildSafeRequest enforces same-origin URLs before axios dispatch, so the default allowAbsoluteUrls behavior cannot be abused.

Supporting Evidence

buildSafeRequest sanitizes the user-controlled path and enforces that the resolved URL stays on the original base origin, preventing absolute URL bypass before axios dispatch.

safeAxiosRequest.js (L23-L37)

```
function buildSafeRequest(baseURL, untrustedPath) {
  const base = new URL(String(baseURL));
  const safePath = sanitizeRelativePath(untrustedPath);

  // Resolve *after* sanitization and enforce same-origin.
```

Konvu
January 6, 2026 at 1:00 AM (edited)

🔴 Exploitable: CVE-2022-1471 in `org.yaml:snakeyaml` (repo: `demoapp`)

Exploitability Condition	Met?	Evidence
Dependency installed: Vulnerable dependency org.yaml:snakeyaml is installed	✓	org.yaml:snakeyaml is imported and available
Code reachability: 1 vulnerable symbol is reachable from application entry points	✓	1 vulnerable symbol is reachable
Attacker controllability: Application parses untrusted YAML	✓	The POST /yaml endpoint instantiates a default SnakeYAML loader and calls yaml.load on the raw @RequestBody payload.

Beyond prioritization, remediation is a critical part of the SCA challenge. Many security engineers have experienced the false promise of automated pull requests that simply bump dependency versions without addressing real migration complexity. Konvu goes significantly further by delivering complete, end-to-end patches for major version upgrades, making complex dependency migrations far more achievable for engineering teams.

The Benefits of Konvu

No Rip and Replace

Konvu plugs into your existing scanners and pushes decisions back where teams already work, no new dashboard needed.

Agentic Triage

Konvu's agents run exploitability analysis in the customer's context, executing a CVE specific investigation plan with reproducible checks.

Evidence Backed Decisions

Every outcome comes with clear reasoning and supporting evidence so AppSec and developers can trust the results, and keep an audit trail.

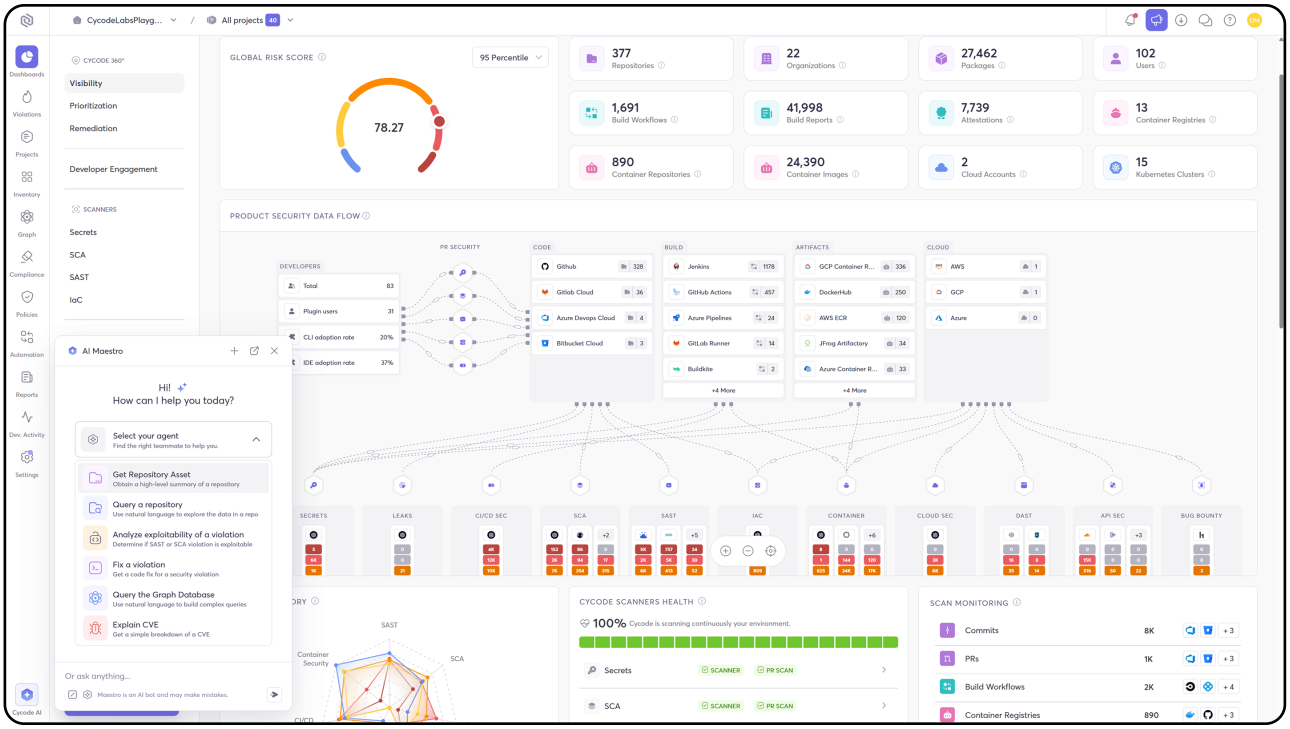
cycode

Latio
APPLICATION SECURITY
PLATFORM LEADER
2026

Latio
APPLICATION SECURITY
MANAGEMENT LEADER
2026

Cycode is one of the most comprehensive application security platforms on the market, spanning software supply chain security, application security testing, and posture management. Cycode can function as an all-in-one application security solution, or as a unifying risk-based platform for vulnerabilities across Cycode and third-party scanners, depending on an organization's needs.

Cycode’s core strength has long been its context intelligence graph - combining data across code, infrastructure, and security findings. This graph allows teams to quickly correlate signals to prioritize exploitable vulnerabilities based on risk and codify why, how, and by whom security decisions and actions are made throughout the vulnerability lifecycle.



Hi!
How can I help you today?

Select your agent
Find the right teammate to help you.

Get Repository Asset
Obtain a high-level summary of a repository

Query a repository
Use natural language to explore the data in a repo

Analyze exploitability of a violation
Determine if SAST or SCA violation is exploitable

Fix a violation
Get a code fix for a security violation

Query the Graph Database
Use natural language to build complex queries

Explain CVE
Get a simple breakdown of a CVE

Or ask anything...

Maestro is an AI bot and may make mistakes.

100% Cycode is scanning continuously your environment.

Secrets ☑ SCANNER ☑ PR SCAN

SCA ☑ SCANNER ☑ PR SCAN

More recently, the team has focused on extending the context intelligence graph to orchestrate AI agents. By providing agents with contextual intelligence, Cycode enables them to emulate security decisions, such as contextual blocking and targeted remediation campaigns. This approach equips security teams to scale alongside developer AI usage.

Whether an organization is looking to replace individual application security testing tools or orchestrate and manage a diverse set of existing scanners, Cycode remains a leading option in the market.

The Benefits of Cycode:

Comprehensive Coverage

Spans supply chain, testing, and posture management within a single platform.

Contextual Risk Graph

Maps findings to ownership, exploitability, and impact across the codebase.

AI-Enhanced Orchestration

Uses rich context to drive smarter blocking, prioritization, and remediation campaigns.

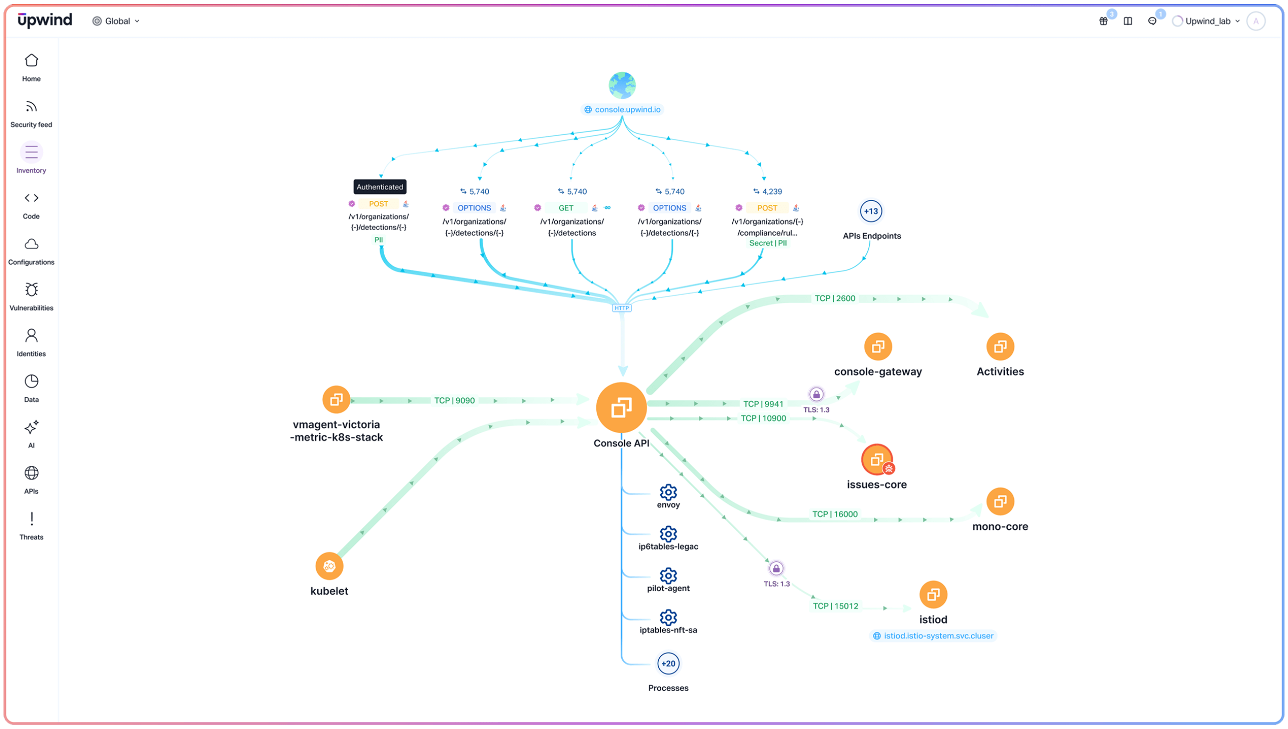
upwind

Latio APPLICATION SECURITY RUNTIME INNOVATOR 2026

Latio APPLICATION SECURITY API SECURITY INNOVATOR 2026

Upwind Security was the first platform to meaningfully combine API security capabilities with those of a traditional CNAPP. By leveraging deep runtime, application, and layer 7 data, Upwind delivers API discovery, dynamic testing of APIs and AI/LLM risks, vulnerability prioritization, and can even serve as a replacement for standalone API security tools. Upwind is unique in pairing these features with the standard suite of CNAPP capabilities, especially agentless scanning, CSPM, and vulnerability management.

Upwind's dynamic testing capabilities are especially strong and exceed what most traditional vulnerability solutions offer today. By collecting network logs directly, the platform deploys contextual tests that reflect real application and network flows, alongside comprehensive API fuzzing and API spec generation. This approach gives teams visibility into how applications actually behave in production, rather than relying on static assumptions.



Among CNAPP providers, Upwind has invested the most heavily in runtime function-level reachability. This capability enables more accurate prioritization, and significant false-positive reduction. These function-level insights also extend into Upwind's Cloud Application Detection & Response (CADR) capabilities, creating a cross-layer runtime protection platform that combines cloud, application, and network security into single detection events.

Overall, Upwind remains a deeply runtime-focused CNAPP, built from the ground up to deliver traditional CSPM capabilities alongside advanced runtime security technologies. These capabilities make it a great selection for teams looking to get the most value out of their cloud security solution.

The Benefits of Upwind:

Better Detections, Faster Response

Combine cloud, workload, and API data to fully detect threats from start to finish, and give teams the context they need to respond.

Improved Dynamic Testing

Use real network and application data to drive more accurate API testing that's contextual to your environment.

Function-Level Prioritization

Reduce alert volume by prioritizing only the vulnerabilities that matter most, focusing on what's executing in your environment.

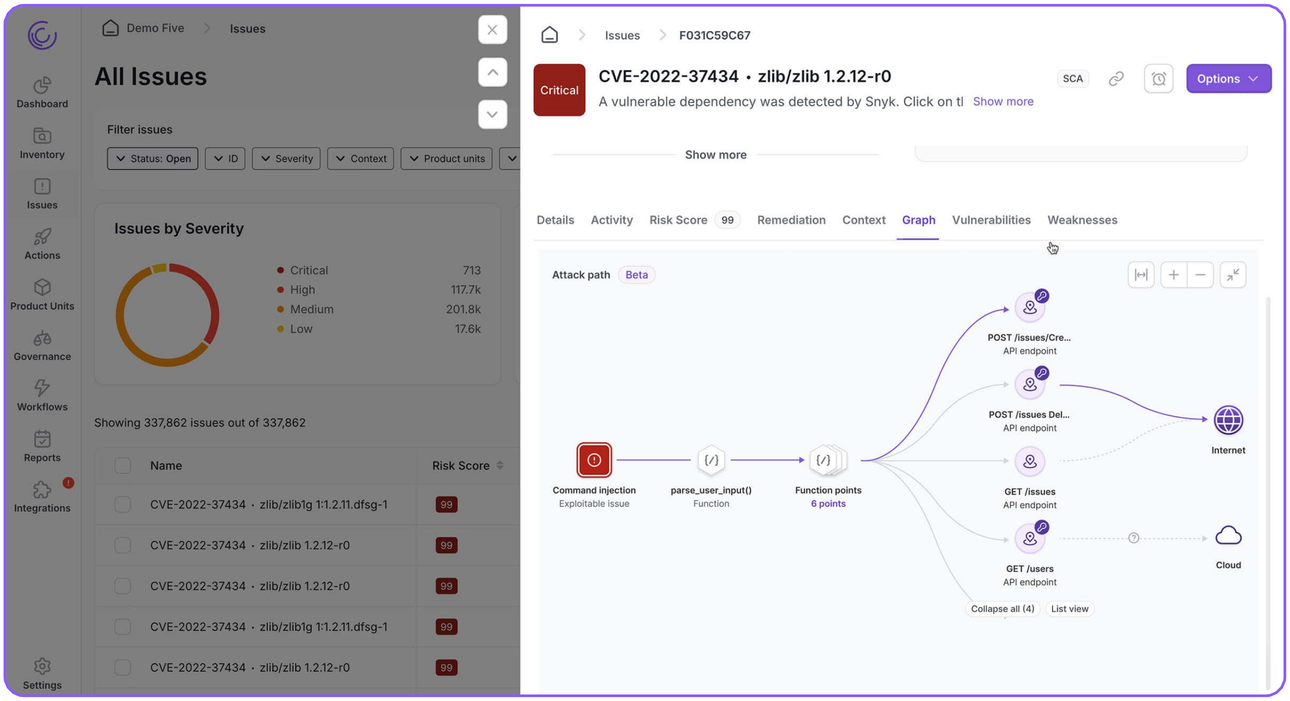
LEGIT

Latio APPLICATION SECURITY MANAGEMENT LEADER 2026

Latio APPLICATION SECURITY AI CODE INNOVATOR 2026

Legit Security is frequently a reference point for enterprises looking for an application security posture management tool. They were the first to make me aware of the ASPM category, and when I saw the product, the solution immediately made sense for those dealing with the "too many scanners" problem. Development teams at large enterprises frequently manage their own pipelines and tools, leading to sprawling and uncertain scanning coverage. Legit steps in to provide teams with a strong mapping of their deployment pipelines, cloud-based or on-premise, and assigns risk and coverage scores as code moves through to deployment.

Within the last few years, Legit has expanded well beyond providing a pure management layer. The platform now includes scanning capabilities like SAST and SCA, API reachability, and significant change tracking. For management and consolidation use cases, these additions have meaningfully increased the overall value of the solution.



CVE-2022-37434 • zlib/zlib 1.2.12-r0

A vulnerable dependency was detected by Snyk. Click on tl Show more

Showing 337,862 issues out of 337,862

Name	Risk Score
CVE-2022-37434 · zlib/zlib1g 1:1.2.11.dfsg-1	99
CVE-2022-37434 · zlib/zlib 1.2.12-r0	99
CVE-2022-37434 · zlib/zlib 1.2.12-r0	99
CVE-2022-37434 · zlib/zlib1g 1:1.2.11.dfsg-1	99
CVE-2022-37434 · zlib/zlib 1.2.12-r0	99

With the recent launch of VibeGuard, Legit Security is on the front lines of addressing challenges with AI code generation. VibeGuard covers critical capabilities, starting with securing AI code generation tools - IDE's, MCPs, and rules - preventing attacks like prompt injection and unapproved secret access. It then helps to secure code as it's generated by fetching organizational and security context, allowing teams to enforce security standards on AI generated code.

Together, these capabilities form a holistic, modern platform that is particularly well suited for enterprise environments.

The Benefits of Legit Security

Gain Visibility Across Deployment Pipelines

Track and monitor workflows to maintain oversight across dispersed development teams, gaining a complete coverage map of your SDLC.

Enforce Governance for AI-Generated Code

Embed AI Governance into development flows, securing end to end developer usage from code generation to ensure secure MCP usage.

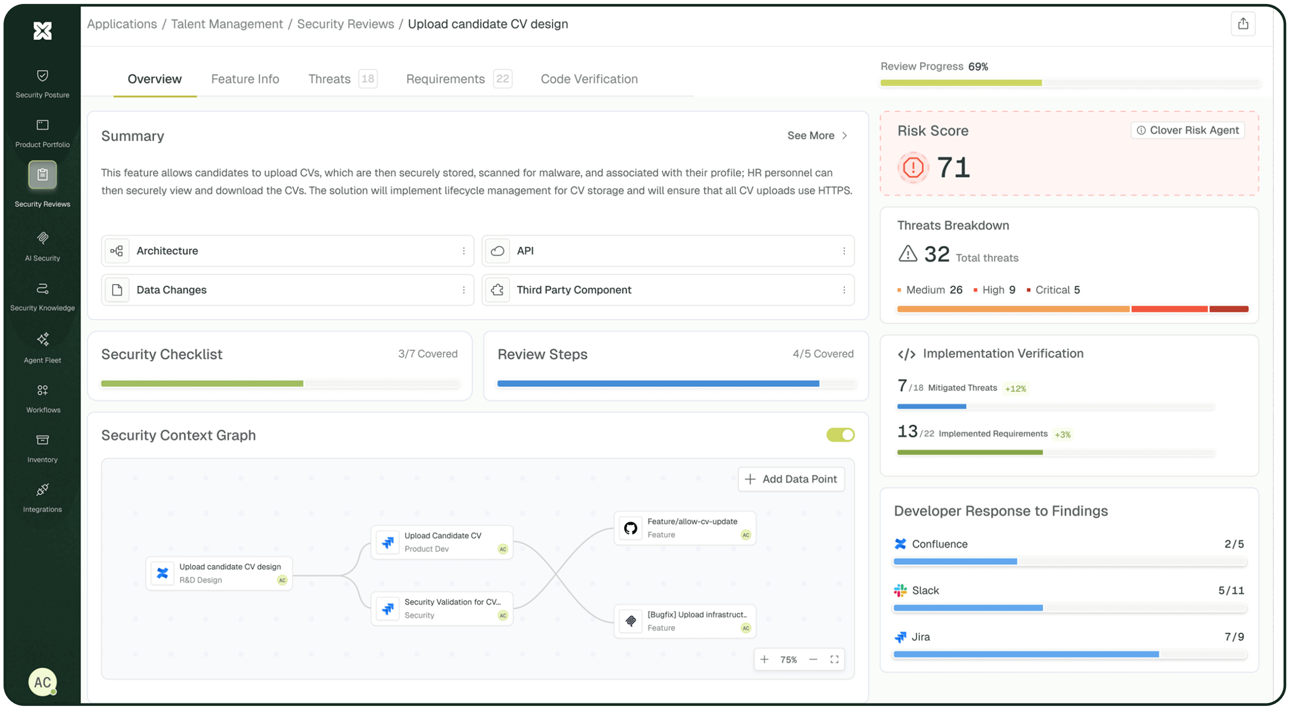
Track Changes Across Enterprise Environments

Teams can assess overall coverage and security posture across their tooling, while using Legit to close gaps across traditional scanners and emerging AI-driven coding workflows.

Clover

Clover Security is a leader in the emerging category of AI threat modeling and design review, which has the potential to reshape the entire application security industry. Clover's ability to generate ongoing security design reviews unlocks two key capabilities. First, it significantly accelerates and improves the quality of threat modeling across an organization. Second, it enables the enforcement of security controls from AI code generation, to pull requests, to deployment.

First, Clover integrates with existing knowledge bases and developer productivity platforms to build baseline contextual awareness of an organization's environment. This allows the platform to prioritize incoming projects while gathering the information needed to support effective threat modeling and design review. Clover then uses this context to help security teams accelerate the often overwhelming and time-intensive processes of conducting design reviews for new systems.



Summary

This feature allows candidates to upload CVs, which are then securely stored, scanned for malware, and associated with their profile; HR personnel can then securely view and download the CVs. The solution will implement lifecycle management for CV storage and will ensure that all CV uploads use HTTPS.

Security Checklist 3/7 Covered

Review Steps 4/5 Covered

Review Progress 69%

Risk Score 71

Threats Breakdown

32 Total threats

Medium 26 · High 9 · Critical 5

Implementation Verification

7 /18 Mitigated Threats +12%

13 /22 Implemented Requirements +3%

Developer Response to Findings

Confluence	2/5
Slack	5/11
Jira	7/9

Second, Clover acts as a continuous enforcement layer for the decisions made during design review. By supplying developers and AI agents with organizational security context, it enables secure-by-default code generation and ensures that security intent is carried through to implementation. Clover brings your organizational security policies, such as architecture and authentication requirements, into every prompt or pull request, enabling catching issues like business logic flaws as early as possible.

By combining these capabilities together, Clover provides a new kind of end to end application security platform - one that helps you design an effective security program, and implement it across the SDLC.

The Benefits of Clover Security

Continuous Threat Modeling

Improves coverage and consistency by making threat modeling an ongoing process.

Context-Aware Design and Code Reviews

Uses organizational knowledge to deliver more accurate and relevant security guidance.

Secure-By-Default Enforcement

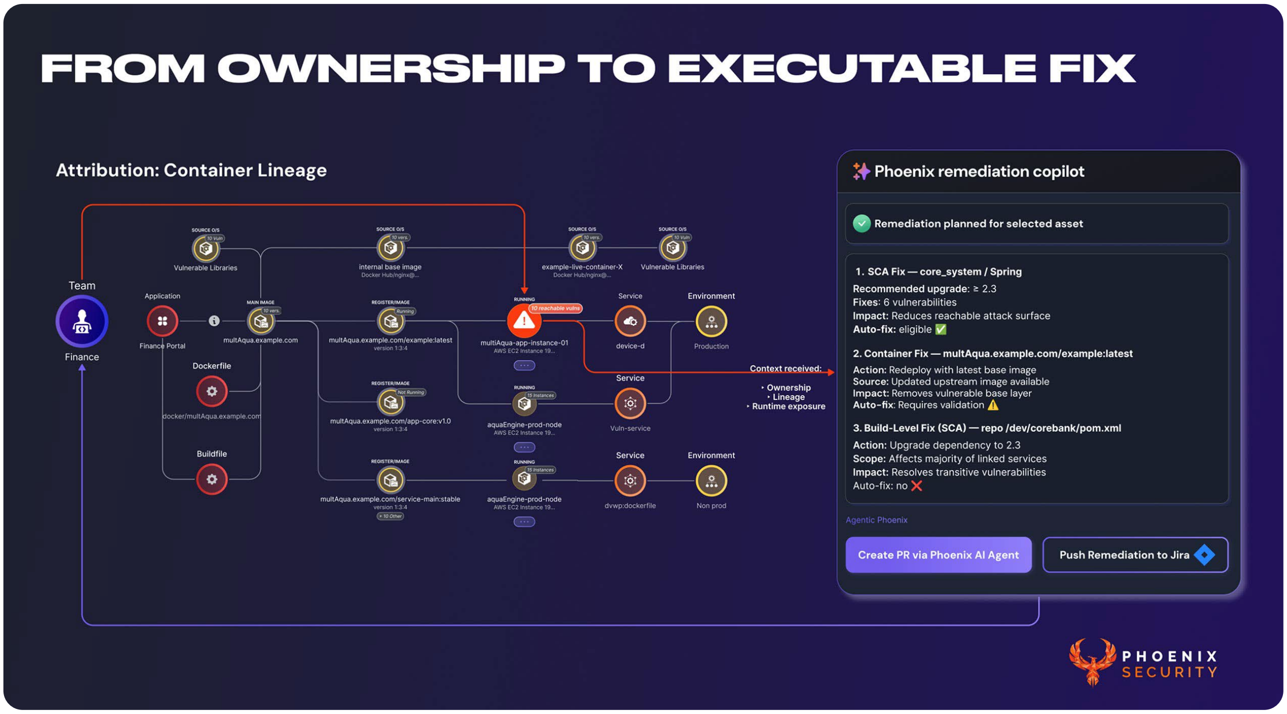
Carries security intent from AI coding through deployment without manual intervention.

PHOENIX SECURITY

Latio
APPLICATION SECURITY
MANAGEMENT
LEADER
2026

Enterprise vulnerability management programs fail because scanners rarely answer the three questions that drive remediation: who owns it, where is it running, and what is the fastest, lowest-impact fix. For organizations that prioritize actionable attribution and operate in regulated industries, many platforms overlook the customization details required to operationalize remediation. Phoenix Security has consistently stood out for its attention to enterprise-level details that make vulnerability management work at scale.

One of Phoenix's primary differentiators is its ability to align asset attribution, code-to-cloud correlation, and reachability analysis with business goals, regardless of which scanning tools feed into the platform. This allows teams to maximize the value of their existing scanners rather than being forced to adopt entirely new ones. These benefits extend to providing AI prioritization and remediation.



FROM OWNERSHIP TO EXECUTABLE FIX

Phoenix remediation copilot

✓ Remediation planned for selected asset

1. **SCA Fix — core_system / Spring**
Recommended upgrade: ≥ 2.3
Fixes: 6 vulnerabilities
Impact: Reduces reachable attack surface
Auto-fix: eligible ✅

2. **Container Fix — multAqua.example.com/example:latest**
Action: Redeploy with latest base image
Source: Updated upstream image available
Impact: Removes vulnerable base layer
Auto-fix: Requires validation ⚠️

3. **Build-Level Fix (SCA) — repo /dev/corebank/pom.xml**
Action: Upgrade dependency to 2.3
Scope: Affects majority of linked services
Impact: Resolves transitive vulnerabilities
Auto-fix: no ❌

Phoenix's attribution model is designed for distributed enterprises where ownership changes, services are ephemeral, and data resides across multiple systems. Phoenix supports PYRUS CMDB-as-code patterns and integrates with enterprise sources of truth to ensure accurate ownership. Phoenix also uses AI across the platform, from vulnerability enrichment to remediation. These features make Phoenix a strong option for enterprises seeking a scalable, configurable vulnerability management solution.

The Benefits of Phoenix Security:

Attribution at Enterprise Scale

Phoenix's attribution CMDB enables teams to manage asset ownership across their entire lifecycle programmatically.

Tool-Agnostic Reachability

Reduce false positives by applying native provenance, lineage, and reachability analysis on top of existing scanners, drastically reducing vulnerability counts.

AI Prioritization and Remediation

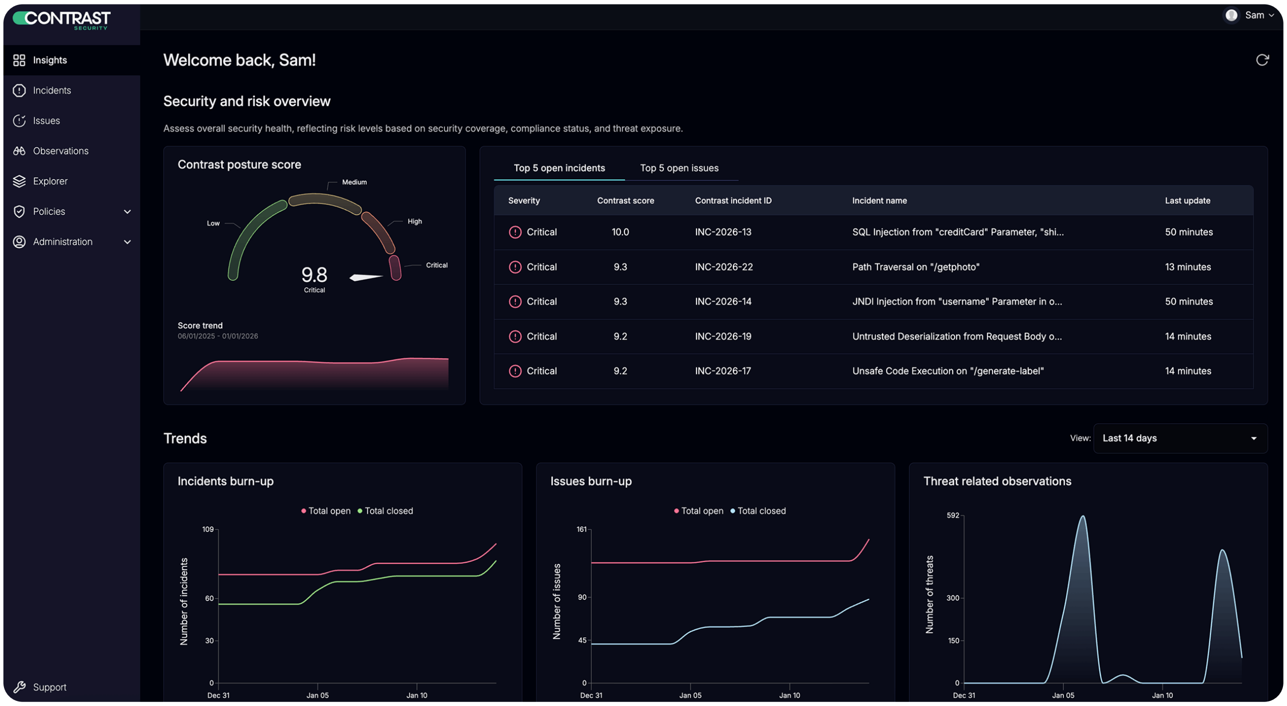
Phoenix's AI systems analyze threat intelligence data to predict the threat types most likely to lead to exploitation, and provide precise remediation guidance.

CONTRAST SECURITY

Latio
APPLICATION SECURITY
RUNTIME INNOVATOR
2026

Contrast Security has long delivered some of the most robust application security protections available. The company pioneered a runtime-oriented approach to application security through Interactive Application Security Testing (IAST) and has since expanded into Application Detection and Response (ADR), bringing the power of its runtime engine directly to security operations teams.

As application attacks continue to increase year over year, public-facing applications have become a primary attack surface. Security operations teams have struggled to act on application-level alerts due to limited visibility and a lack of understanding of how applications actually behave. Contrast addresses this challenge by providing deep, runtime-level insight into application behavior.



Welcome back, Sam!

Security and risk overview

Assess overall security health, reflecting risk levels based on security coverage, compliance status, and threat exposure.

Top 5 open incidents | Top 5 open issues

Severity	Contrast score	Contrast incident ID	Incident name	Last update
Critical	10.0	INC-2026-13	SQL Injection from "creditCard" Parameter, "shi...	50 minutes
Critical	9.3	INC-2026-22	Path Traversal on "/getphoto"	13 minutes
Critical	9.3	INC-2026-14	JNDI Injection from "username" Parameter in o...	50 minutes
Critical	9.2	INC-2026-19	Untrusted Deserialization from Request Body o...	14 minutes
Critical	9.2	INC-2026-17	Unsafe Code Execution on "/generate-label"	14 minutes

Trends

View: Last 14 days

Beyond visibility, Contrast delivers detection, testing, and response capabilities directly at runtime. Through deep application instrumentation, the platform observes payloads, attack paths, and execution behavior in production, correlating this telemetry in the Contrast Graph to give security teams the context they need to respond to attacks with confidence.

Contrast provides testing and remediation by determining which vulnerabilities are truly reachable and uncovering novel exploits in live applications, rather than relying solely on historical CVE data. For teams prioritizing a runtime-first approach across application security and SecOps, Contrast is a compelling option.

The Benefits of Contrast Security

Runtime Application Visibility

Provides deep insight into how applications behave in real production environments.

High-Fidelity Detection and Protection

Surfaces precise payloads and attack paths to support confident investigation and response.

Reachability-Driven SCA

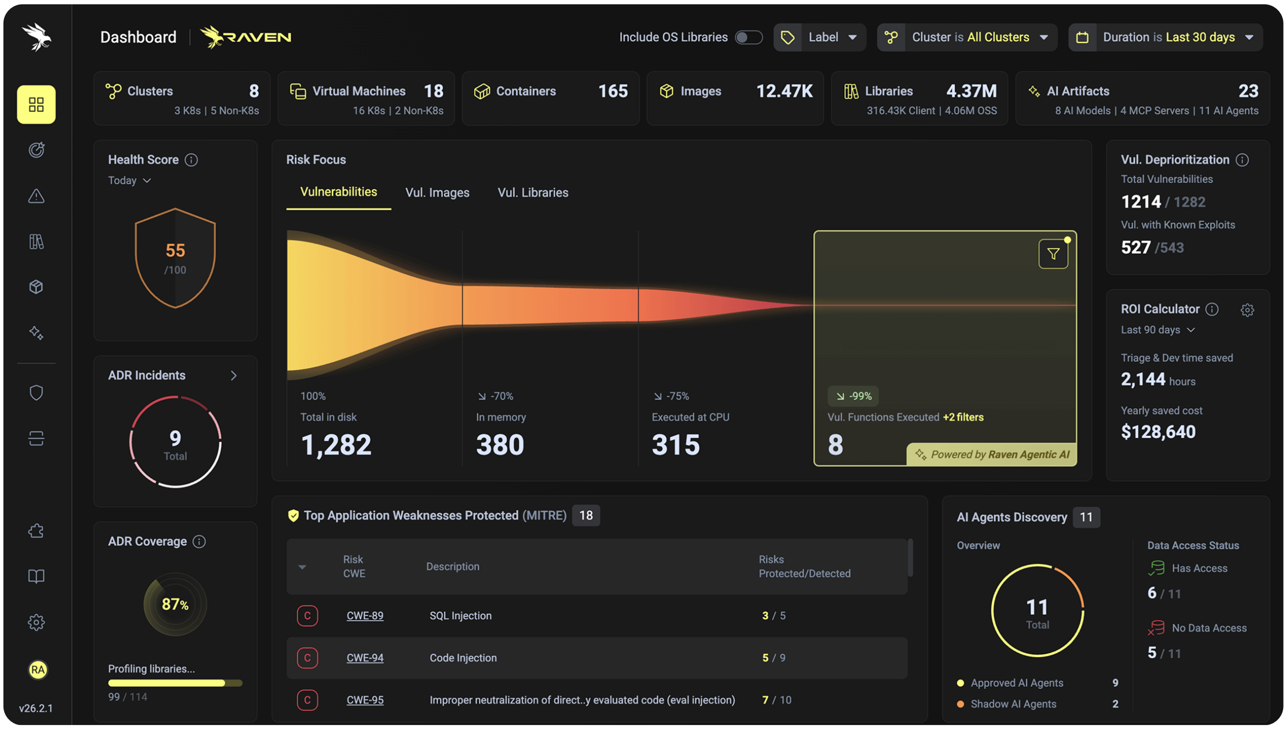
Identifies exploitable vulnerabilities across custom code and open-source dependencies based on runtime reachability.

RAVEN

Latio
APPLICATION SECURITY
RUNTIME INNOVATOR
2026

Raven has built a leading Application Detection Response (ADR) solution for companies to protect critical applications. Raven unlocks the missing piece of workload protection for software teams by preventing malicious deviations, whether a CVE exists or not. In an age where attackers are leveraging AI to automate exploitation, having a prevention solution for CVE-less threats is more valuable than ever. Using deep application inspection at runtime, Raven is able to tie functions that execute back to the original code that produced them, and prevent libraries from executing attacker manipulated paths.

With ADR, developers get the insights they need to separate action from noise by contextualizing alerts to an application's environment, and security is able to prevent attacks, known and unknown, so patching isn't done under fire. Under the barrage of open source and AI assisted exploits, teams need better ways to protect themselves besides waiting for a patch to be released.


Dashboard | RAVEN

Include OS Libraries | Label | Cluster is All Clusters | Duration is Last 30 days

Clusters	Virtual Machines	Containers	Images	Libraries	AI Artifacts
8	18	165	12.47K	4.37M	23
3 K8s | 5 Non-K8s	16 K8s | 2 Non-K8s			316.43K Client | 4.06M OSS	8 AI Models | 4 MCP Servers | 11 AI Agents

Health Score
Today
55 /100

Risk Focus
Vulnerabilities | Vul. Images | Vul. Libraries

100% Total in disk 1,282
-70% In memory 380
-75% Executed at CPU 315
-99% Vul. Functions Executed +2 filters 8
Powered by Raven Agentic AI

Vul. Deprioritization
Total Vulnerabilities
1214 / 1282
Vul. with Known Exploits
527 /543

ROI Calculator
Last 90 days
Triage & Dev time saved
2,144 hours
Yearly saved cost
$128,640

ADR Incidents
9 Total

ADR Coverage
87%
Profiling libraries...
99 / 114

Top Application Weaknesses Protected (MITRE) 18

Risk	CWE	Description	Risks Protected/Detected
C	CWE-89	SQL Injection	3 / 5
C	CWE-94	Code Injection	5 / 9
C	CWE-95	Improper neutralization of direct..y evaluated code (eval injection)	7 / 10

AI Agents Discovery 11
Overview
11 Total
● Approved AI Agents 9
● Shadow AI Agents 2

Data Access Status
Has Access 6 / 11
No Data Access 5 / 11

v26.2.1

Raven helps with incident response and vulnerability management by offering teams the critical application layer insights they've been missing. They do this with low overhead, offering out of the box performance dashboarding with common developer tools. Beyond insights alone, Raven even offers an elegant proactive protection solution by permissioning libraries into known categories of system calls, enabling true zero day protection.

For enterprises looking to bring their applications into their security program, and mitigate advanced attacks, Raven is a great solution.

The Benefits of Raven

Prevent Attacks

As application layer attacks continue to increase, prevent the latest attacks while giving your team time to patch, while keeping your application running.

Eliminate Vulnerabilities

Respond only to known vulnerable function executions, prioritizing what matters.

Simple Deployment

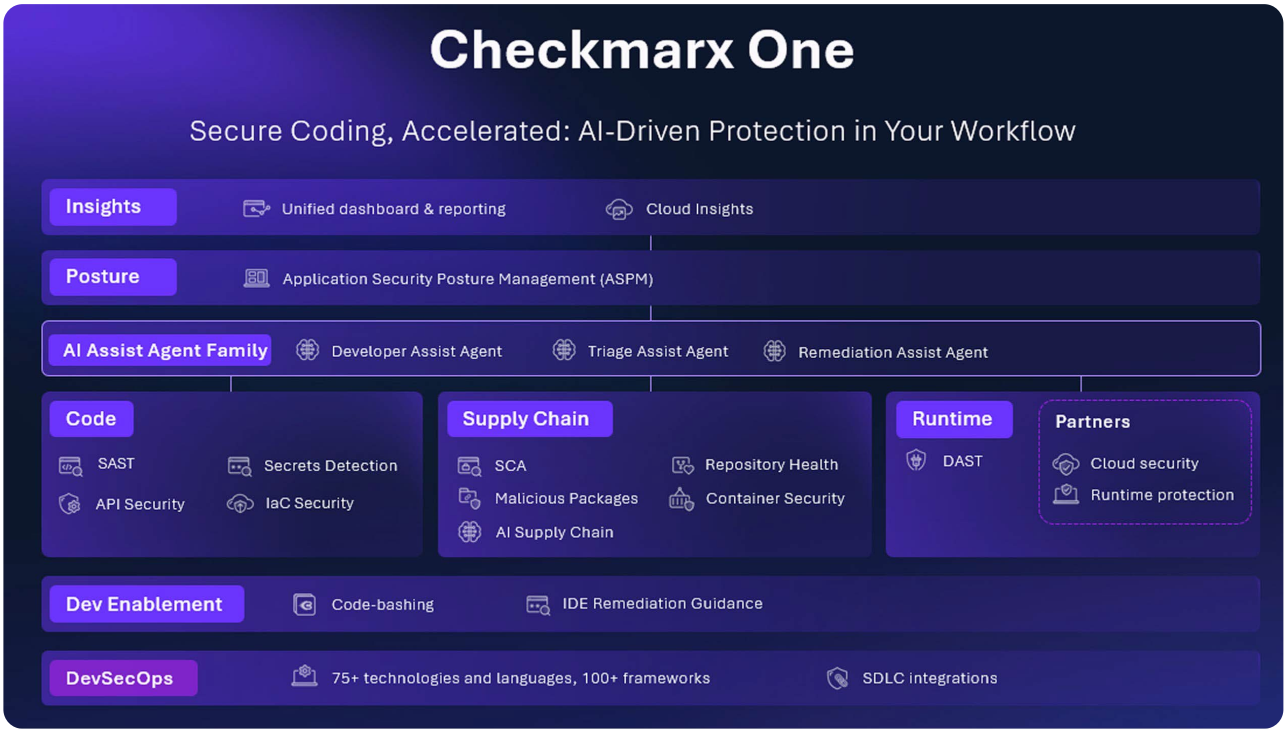
Deploy an extremely efficient sensor in minutes to immediately reduce your vulnerability counts by 99% and stop malicious code before it executes.

Checkmarx

Latio
APPLICATION SECURITY
TESTING
LEADER
2026

Checkmarx has done an impressive job delivering the power of a modern, cloud-driven, AI-focused DevSecOps platform without sacrificing the depth or precision of its historically robust scanning engines. For organizations looking to modernize application security at scale, the Checkmarx One platform offers strong coverage across scanning categories without compromising core capabilities.

Checkmarx has been steadily raising the bar on platform depth and precision by including features such as container image layer analysis, detailed customization of SAST findings, custom query development, and an IDE experience for agentic development lifecycles. These capabilities have been rolled out across a broad set of languages at enterprise scale, giving customers access to modern application security workflows without losing depth.



Checkmarx One

Secure Coding, Accelerated: AI-Driven Protection in Your Workflow

Insights: Unified dashboard & reporting; Cloud Insights

Posture: Application Security Posture Management (ASPM)

AI Assist Agent Family: Developer Assist Agent; Triage Assist Agent; Remediation Assist Agent

Code: SAST; Secrets Detection; API Security; IaC Security

Supply Chain: SCA; Repository Health; Malicious Packages; Container Security; AI Supply Chain

Runtime: DAST

Partners: Cloud security; Runtime protection

Dev Enablement: Code-bashing; IDE Remediation Guidance

DevSecOps: 75+ technologies and languages, 100+ frameworks; SDLC integrations

One of Checkmarx's long-standing strengths has been the level of customization it provides for enterprise application security teams. Its highly configurable scanning engines allow teams to enforce organization-specific requirements across large and complex codebases, aligning security controls with existing standards and threat models.

Checkmarx also delivers a strong management layer for application security teams, enabling centralized rule and risk management across diverse applications. This approach allows organizations to enforce consistent security standards while still accounting for the unique needs of individual applications.

The Benefits of Checkmarx:

Enterprise-Grade Customization

Enforces organization-specific security standards across large and complex codebases.

Modern DevSecOps Platform

Delivers cloud-native AI-assisted workflows without sacrificing scanning depth or precision.

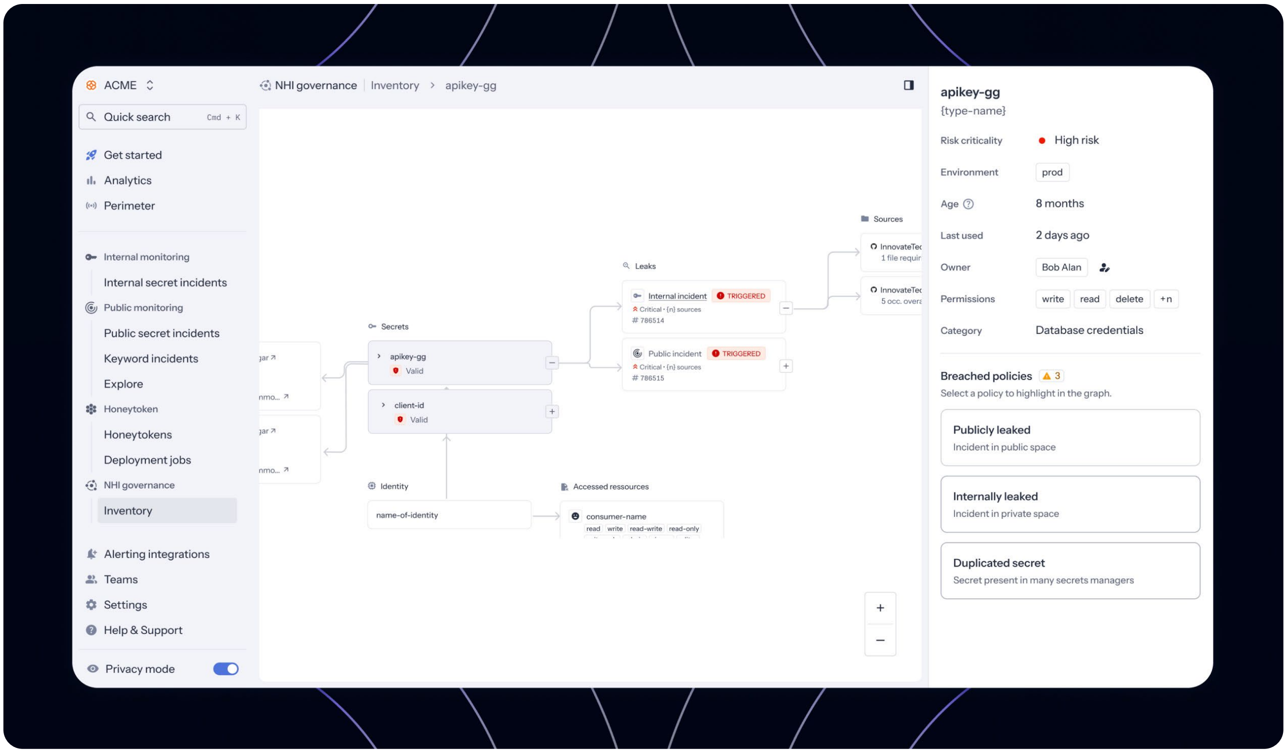
Centralized Risk Management

Manages rules and risk consistently across applications and teams.

GitGuardian

GitGuardian has matured into far more than a secret scanning tool. The detection engine is built and optimized for scanning at internet scale. It leverages a deterministic rules engine to identify secrets, paired with AI to filter false positives and increase context. This distinction matters because understanding whether a secret has been leaked publicly or exposed elsewhere is what determines risk and allows teams to prioritize.

By expanding detection beyond source code, GitGuardian delivers additional value through identifying secrets across other platforms such as Slack and workforce tooling. This broader visibility enables teams to take a more comprehensive data loss prevention (DLP) approach to secrets management, instead of narrowly focusing on code repositories.



apikey-gg
{type-name}

Risk criticality: ● High risk
Environment: prod
Age: 8 months
Last used: 2 days ago
Owner: Bob Alan
Permissions: write read delete +n
Category: Database credentials

Breached policies ⚠ 3
Select a policy to highlight in the graph.

Publicly leaked
Incident in public space

Internally leaked
Incident in private space

Duplicated secret
Secret present in many secrets managers

Risk assessment in GitGuardian tests the validity of credentials while simultaneously connecting with IdPs and applications to enumerate identities and their permissions. By cross-referencing these with identified secrets, the platform provides a precise understanding of the potential blast radius. It also identifies which workloads actively consume them, minimizing the risk of breaking production applications.

These capabilities make GitGuardian a more holistic non-human identity security solution, providing both proactive and reactive identity security controls across the organization.

The Benefits of GitGuardian

Broader Secrets Detection

Identify exposed secrets across code repositories, collaboration tools, and public sources, taking advantage of a highly tuned hybrid detection engine.

Remediation Context

Assess potential impact by understanding permissions, active usage and exposure, allowing teams to remediate based on severity.

Non-Human Identity Coverage

Support proactive and reactive controls for secrets and machine identities across the organization, including detection and response with HoneyTokens.

Latio

APPLICATION SECURITY

PLATFORM LEADER

2026

WIZ GitHub DATADOG

cycode aikido

apiiro OX

Latio

APPLICATION SECURITY

TESTING LEADER

2026

invicti Checkmarx snyk

Semgrep VERACODE

Latio

APPLICATION SECURITY

MANAGEMENT LEADER

2026

apiiro WIZ LEGIT

cycode PHOENIX SECURITY ArmorCode

Latio

APPLICATION SECURITY

INNOVATOR

2026

Category Innovators

AI Code

BACKSLASH

Corridor

OX

Pillar

snyk

GitHub

LEGIT

[arnica]

Supply Chain

ENDOR LABS

Socket

FOSSA

aikido

konvu

HEELER

Runtime

oligo

RAVEN

Kodem

MIGGO

upwind

CONTRAST SECURITY

DATADOG

sweet.

WIZ

dynatrace

AI Platform

ZEROPATH

CORGEA

AI Pentesting

aikido

XBOW

WIZ

Secrets Security

GitGuardian

DAST

escape

invicti

STACKHAWK

API Security

upwind

Developer Experience

[arnica]

Definitions

Platform Leader

Leaders in this category are built to be the only application security platform teams need. This means including all core application security scanning capabilities alongside cloud and runtime context features for prioritization and team context. Leaders in this category have made large investments in the latest scanning features, without sacrificing the overall platform.

Testing Leader

This category represents the companies who meet the diverse requirements of enterprise use cases - meaning support for the wide variety of languages, scan types, and reporting that these companies need. They also have robust hosting, customization, and tertiary support services.

Management Leader

Leaders in this category are built for integrating with numerous scanners to drive workflows with rich application context, creating an orchestration platform for remediating vulnerabilities. While they often provide their own scanning tools as well, the strengths of these platforms are in their ability to consolidate data across massive environments, creating unified remediation workflows.

AI Code Innovator

This award is for companies investing in new technology for securing AI generated code by both securing employee workstations against MCP supply chain and rule injection attacks, and giving AI coding agents the context they need to deploy secure code. These emerging tools are designed to work with AI coding development tools to create secure code by default.

Supply Chain Innovator

Companies in this category have innovated in specific ways for teams with supply chain security concerns. Vendors in this category are highlighted for their investments in legal analysis, malware detection, package health, prioritization, or autopatching capabilities.

Runtime Innovator

Innovators in this category excel at protecting applications against runtime threats. This category is about representing tools that don't merely detect ongoing application attacks, but also offer various threat mitigation capabilities, allowing enterprises to respond in seconds to the latest application attacks.

Category Innovator

These awards acknowledge companies who are especially innovating within either a specific category, emerging capability, or set of features.

Latio

Ever wonder: Am I using the right security tools for my business, or am I building the right product for the market?

Everyday companies are making decisions based on the information that is available to them, which is often incomplete and based on vibes rather than usage.

That's where Latio comes in.

Founded in 2023 by James Berthoty, Latio was built to solve a critical problem James was facing: there was no reliable, credible way to evaluate a vendor's capabilities until after an agreement was signed. Latio exists to make the buying and building processes better by getting accurate information to the most relevant teams.

We focus on the product, the practitioner, and the market rather than slides and hype cycles. We believe the greatest predictor of a great security tool and program is finding the right product fit for both vendors and buyers.

We are creating a future where every decision is based on tests, market insights, experience, and hard work, where it's easy to find the right product you're looking for.

Our mission is to help every team find the right security product. So we test every product, to make it easier for you to pick the right one.

A special thank you to everyone who has supported this mission, without you, none of this would be possible.

Learn more:

🌐 latio.com	📅 Schedule a product briefing
📅 Schedule a security program sync	in Follow us

Latio

The only analyst firm that tests products,
so you can find the right one.