Checkmarx

Checklist

The Ultimate AppSec RFP Checklist & Guide

Compare, Evaluate, and Choose the Right AppSec Vendor for Your Organization

With a growing number of AppSec vendors and solutions in the market—and an ever-evolving landscape of development practices, product advancements, and emerging security risks—making a well-informed, structured decision that fully meets your organization's needs is more challenging than ever.

This guide provides a comprehensive, structured, and user-friendly checklist to help AppSec managers and security leaders compare, evaluate, and select the best vendor to meet their needs. Designed to align with standard enterprise procurement processes, this checklist ensures you:

- **Define and prioritize** your requirements based on security, scalability, integrations, and usability.
- **Objectively assess** multiple vendors to compare their capabilities side by side.
- **Justify your recommendation** with a structured evaluation that supports internal approvals.
- **Streamline procurement** by providing security, IT, and finance teams with a clear decision framework.

By using this checklist

you'll ensure a thorough, data-driven evaluation of potential vendors, making it easier to select the right solution while facilitating internal approvals and procurement.

RFP Checklist

Cloud-Native Application Security

Code to Cloud Protection

- [ ] Static Application Security Testing (SAST)
- [ ] Software Composition Analysis (SCA)
- [ ] Software Supply Chain Security (SSCS)
- [ ] API Security
- [ ] Secrets Detection
- [ ] Dynamic Application Security Testing (DAST)
- [ ] Container Security
- [ ] Infrastructure-as-Code (IaC) Security
- [ ] Runtime Insights

Developer Workflow

- [ ] IDE integration
- [ ] SCM integrations with Pull Request (PR) decoration
- [ ] CI/CD and build tool integration
- [ ] Feedback tool integration
- [ ] GenAI plug-ins
- [ ] Command Line Interface (CLI)
- [ ] Comprehensive language support
- [ ] Development framework support
- [ ] Compiler-agnosticism

Accuracy, Correlation, and Prioritization

- [ ] Customizable rules
- [ ] Customizable rulesets
- [ ] AI Query Builder
- [ ] False positive and false negative ratios
- [ ] Best-Fix Location (BFL)
- [ ] Static reachability analysis
- [ ] Runtime reachability analysis
- [ ] Risk management

Developer Enablement

- [ ] Structured learning paths
- [ ] Just-in-time training
- [ ] Remediation guidance
- [ ] AI-guided remediation

Analysis and Triage

- [ ] Data flow visualization
- [ ] Bulk marking results
- [ ] Permanent results modification
- [ ] Vulnerability assignment
- [ ] Deduplication of similar findings
- [ ] Aggregate vulnerability graphs
- [ ] Collaborative auditing

Performance and Scale

- [ ] Supports 1000s of repositories
- [ ] Supports 5 million lines of code
- [ ] Scalability process
- [ ] Incremental scans
- [ ] Concurrent scans
- [ ] Fast scans
- [ ] Deep scans
- [ ] Project exclusion
- [ ] Taxonomy of supported vulnerabilities

RFP Checklist

Cloud-Native Application Security

Reports, Dashboards, And Notifications

- [ ] Unified dashboard
- [ ] Configurable dashboards
- [ ] Export Data to PDF/CSV/JSON
- [ ] Historical changes
- [ ] Executive summaries and dashboards
- [ ] Notification alerts
- [ ] Policy management
- [ ] API support

Professional Services

- [ ] Onboarding
- [ ] AppSec team training
- [ ] Technical training
- [ ] Troubleshooting
- [ ] AppSec program consulting
- [ ] Developer workshops
- [ ] AppSec program management
- [ ] Query tuning and preset optimization
- [ ] Results triage and optimization

Deployment Options and Other

- [ ] Single SaaS cloud platform
- [ ] Private cloud deployment option
- [ ] On-premises deployment option
- [ ] Planned release cycle and documentation
- [ ] Role-Based Access Control
- [ ] Security certifications

General Vendor Requirements

- [ ] Leader in the Gartner MQ or Forrester Wave
- [ ] 10+ years of experience in AppSec market

Download a Workable Checklist Spreadsheet

Where you can rate and compare vendors

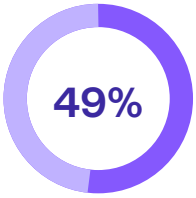
Download Here ↘

The Need for Development-promoting AppSec

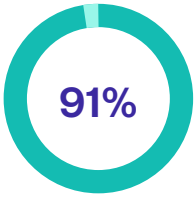
Cloud-native development is shifting how enterprises think about applications: what they are made of, how they're packaged, and the way they're deployed. The result is an increase in developers and distributed teams writing billions of lines of code across hundreds or thousands of apps and microservices, using thousands of pipelines. The rapid nature of this expansion, combined with micro-level needs of teams, has seen companies rely on exponentially more tools with limited visibility into how they all work together as well as security gaps they create.

The stakes here are high

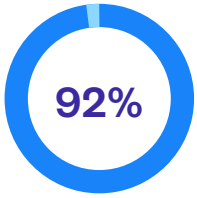
In a recent Checkmarx survey of 200 CISOs



of respondents said that their customers always or very often consider application security in purchase decisions.



of organizations have deployed known vulnerable code in production to meet business or feature deadlines.



of organizations experienced at least one breach in the last year because of a vulnerable application they developed.

Many AppSec teams struggle with managing a fragmented toolset, each producing its own set of vulnerabilities and insights. This siloed approach creates inefficiencies, increased costs, delays in the resolution of critical risks and development bottlenecks. There is a growing need to change the way AppSec practitioners choose their application security solutions.

This guide, together with the checklist, provides the necessary context and background to understand the biggest challenges in modern application security—and how the right platform can seamlessly enhance security, reduce risk, and support development goals simultaneously.

The Three Holy Grails of Modern AppSec

Consolidation

Managing multiple vendors, products, logins, and billing cycles cannot be done effectively without a consolidated platform that provides: the following benefits: reduced total cost of ownership (TCO), simplified vendor management, and a "single pane of glass" for AppSec solutions.

Developer Experience

An AppSec solution must be developer-friendly—this isn't a luxury, it's a necessity. Developers are the ones fixing vulnerabilities, and if security disrupts their workflow, issues won't get fixed, deadlines will slip, or both.

The right platform integrates seamlessly into development environments, providing instant feedback within IDEs, prioritizing vulnerabilities so teams focus on the most critical risks, and automating workflows to reduce friction.

Time to Value

AppSec teams are increasingly asked to do more with less. In large enterprises, they are often outnumbered by developers 150:1. The need to get AppSec solutions onboarded, then tuned properly, all while seamlessly integrating into developer workflows to minimize impact on time to market is a massive challenge. With DevSecOps and continuous secure development as the goal, time to value is essential when considering an AppSec solution.

Code-to-Cloud Protection

Your selected solution must come equipped with a full suite of application security tools to cover the entire SDLC, and assist in the correlation and prioritization of results from different solutions. These are the must-haves:

Static Application Security Testing (SAST)

Conduct fast and accurate scans to identify risk in your custom code. Cover all your applications with both deep scans of critical apps and quick-and-wide scans for the rest of your application footprint.

Software Composition Analysis (SCA)

Identify security and license risks in open-source software used in your applications, while also allowing you to prioritize which vulnerabilities are actually exploitable.

Software Supply Chain Security (SSC)

Proactively identify supply chain attacks and secure developer environments with open-source vulnerability and malicious package detection, SBOMs, and secrets detection in your code and development sources.

API Security

Discover every API in your application and compare the full API inventory against your API documentation to help you eliminate shadow and zombie APIs and mitigate API-specific risks.

Secrets Detection

Automatically scan software repositories for exposed credentials, tokens, and API keys, and flag potential secrets leakage.

Dynamic Application Security Testing (DAST)

Identify vulnerabilities only seen in running applications and assess their behavior by testing against a broad range of web application attacks.

Container Security

Scan static container images, check configurations, and determine what open-source packages are called and identify vulnerabilities pre-production.

Infrastructure-as-Code (IaC) Security

Automatically scan your IaC files to find security vulnerabilities, compliance issues, and infrastructure misconfigurations, using thousands of predefined queries.

Runtime Insights

Extend AppSec into production to correlate pre-production data with runtime insights to better prioritize the most critical vulnerabilities to fix first.

Accuracy, Correlation, and Prioritization

As companies shift to a code-to-cloud mentality, they use a growing number of AppSec tools, further complicating the Appsec process rather than simplifying them. AppSec teams are overwhelmed with disparate tools, incomplete insights, and disconnected workflows.

An effective enterprise AppSec platform consolidates security efforts into a unified solution, integrating security across the development lifecycle. It correlates insights from multiple tools into a single source of truth and enables teams to prioritize remediation based on business risk and exploitability. A unified platform eliminates inefficiencies, reduces operational overhead, and accelerates remediation by improving collaboration between teams. These are the must-haves:

Customizable rules

Augment prebuilt rules and write custom rules based on unique characteristics of your specific applications.

Customizable rulesets

Modify existing rulesets, or build custom rulesets, to make it easier to apply different sets of queries to your applications based on your AppSec goals.

AI Query Builder

Leverage the power of AI to enable teams with little or no AppSec expertise to build custom rules and tailor security coverage to your applications.

False positive and false negative ratios

Providing accurate results is critical to maintaining a positive developer experience. An AppSec platform needs to provide metrics or control measures for false positives and false negatives.

Best-Fix Location (BFL)

Automatically guide developers to the line of code from which to best fix a vulnerability. Using BFL often results in resolving multiple vulnerabilities with one action, saving developers time and effort.

Static reachability analysis

Evaluate vulnerabilities in open-source libraries and analyze whether they are actually called by your application's code. If they are not called, they are not exploitable. Filtering these out will help you focus on vulnerabilities that impact your business.

Runtime reachability analysis

40% of vulnerabilities found by AppSec teams happen once an application is in production. Ensure that your platform can deliver runtime insights to get a full picture of your applications in use. Your platform should connect the dots between pre-production and deployment, giving your team clear visibility into the vulnerabilities that exist in workloads running in production.

Risk management

Your AppSec platform should be able to take its identified list of riskiest apps and present it in an easy-to-read visualization. This visualization should both be easily shareable, and easy to work directly from.

Analysis and Triage

Organizations face an ever-growing number of security findings across their application portfolios, making it crucial to distinguish between critical threats that require immediate attention and less severe issues that can be addressed over time. These are the must-haves:

Data flow visualization

Display data flow (source to sink, across files) to offer optimal triage and remediation of findings.

Bulk marking results

A platform should make it easy to triage large scan volumes by bulk-marking results for simplified management.

Permanent results modification

Support for changes to the state or severity of individual (or bulk) results.

Vulnerability assignment

Vulnerabilities should not just live in a general pool waiting for a developer to remediate. Your platform should allow the specific assignment of vulnerabilities to developers as appropriate.

Deduplication of similar findings

Avoid creating multiple tickets for the same or similar vulnerabilities via correlation of findings and triage status.

Aggregate vulnerability graph

Support visualization of aggregated vulnerabilities to show the same source and destination for optimal triage.

Collaborative auditing

An AppSec platform should support the ability for multiple users and developers to collaborate and comment on audit findings.

Promoting Developer Experience

Addressing application security requires a holistic approach that bridges the gap between AppSec and dev teams. The success of any AppSec program depends on developer buy-in and active participation, achieved by prioritizing developer experience. Developer experience can be improved on two fronts: Streamlined workflows and enablement tools (like education, training, and auto-remediation). These are the must-haves:

↘ Developer Workflow

IDE integration

Import scan results directly into any IDE to maintain developer workflow, so developers don't need to leave their environment. Direct developers automatically to the line of vulnerable code, provide remediation guidance and links to interactive security training. Your platform should integrate with common IDEs like Eclipse, JetBrains, Visual Studio, and VS Code.

SCM integrations with Pull Request (PR) decoration

Integrate directly with the repo to scan uncompiled code, as early as check-in. You should be able to set triggers for scans based on events, such as the PR, allowing your organization to shift further left while staying within developers preferred workflow.

CI/CD and build tool integration

Automate scanning as part of your CI/CD pipeline by integrating with all major tools automating development, deployment, and testing.

Feedback tool integration

Put your vulnerabilities in context for your developers by treating them like any bug. Auto-create tickets in a bug-ticketing system, like JIRA, automatically assign to developers, and automatically close tickets when vulnerabilities are resolved. Vulnerability information, remediation guidance, and knowledge links should all automatically be populated in each ticket.

GenAI plug-ins

70% of developers currently use GenAI to write code. The most advanced AppSec platforms include GenAI plug-ins to scan generated code for vulnerabilities and malicious packages within the GenAI tool before it's exported.

Command Line Interface (CLI)

Provide CLI agents for key platforms (Windows, Linux, etc.) enabling developers to perform the same operations available through the Web UI (creating projects, managing scans, managing triage results, viewing results, etc.)

Development framework support

Support your development teams' work, with broad support across major frameworks, including the latest frameworks like Flutter.

Compiler-agnosticism

Different development teams or business units often use different tools including compilers. An AppSec platform that can support your entire development organization needs to be compiler-agnostic for various development environments.

↘ Developer Enablement

Structured learning paths

Only 25% of developers feel confident writing secure code, largely because top CS programs don't require secure coding courses. Training is essential to closing this gap and strengthening application security. Provide personalized secure code training journeys, designed to equip individual developers with role-specific knowledge.

Just-in-time training

Connect developers to learning modules directly through their vulnerability tickets, to customize their learning path.

Remediation guidance

Close security knowledge and training gaps that hinder developers' ability to fix vulnerabilities, by providing them with actionable and easy-to-understand remediation guidance on every vulnerability.

AI-guided remediation

Utilize AppSec-trained GenAI to suggest remediation steps for identified vulnerabilities, helping developers with less security knowledge reduce the time to identify and fix security flaws.

Cloud-Based Platform

An application security platform must be designed for enterprise-level scalability and performance to effectively secure modern development pipelines.

With multiple development teams simultaneously working across numerous repositories, branches, and pull requests, the platform needs to quickly scan code, identify vulnerabilities, and provide feedback without becoming a bottleneck in the CI/CD pipeline. Performance is crucial because developers require immediate security feedback to maintain productivity and ensure security fixes can be implemented without disrupting development workflows.

A platform that scales poorly or performs slowly will lead to developers bypassing security checks, creating backlogs of security debt, or missing critical vulnerabilities before code moves toward production environments. These are the must-haves:

↘ Performance and Scale

Vulnerability assignment

Large enterprises can have 1000s of developers and hundreds of development teams all working on different applications. An AppSec platform needs to enable a consistent AppSec posture across your entire application footprint, providing the ability to integrate with and scan 1000s of applications and/or software repositories.

Supports 5 million lines of code

Enterprise applications are larger and more complex. While there are many ways to measure scale and complexity, an easy metric is how many millions of lines of code it can cover. Your AppSec platform should be able to scan individual codebases exceeding 5 million lines of code.

Scalability process

Most organizations move to a cloud-based platform to make it easier to scale a solution to meet their needs. Ask every AppSec vendor to describe the process to scale up their environment if your needs grow after you've deployed

Incremental scans

Scanning large codebases is time-consuming. However, most code isn't changed between scans. To reduce disruption, your platform should accelerate scan time by identifying and analyzing only modified code.

Concurrent scans

With many applications being developed simultaneously, an AppSec platform needs the ability to scan many different applications at the same time to keep up with development schedules.

Fast scans

Your platform should give quick scans that return shorter lists of only the most actionable vulnerabilities, with minimal false positives to quickly get a handle and fix the top threats first.

Deep scans

While fast scans make it easy to start and allow you to cover your entire application footprint more easily, you also need run deep scans to cover all the risk in your most critical applications.

Project exclusion

Along with incremental scans, your platform should accelerate scan times by allowing you to exclude projects and folders from specific scans, which results in greater precision and optimization of your time.

Taxonomy of supported vulnerabilities

the platform should provide broad, comprehensive identification of as many vulnerabilities as possible, as well as correlating and prioritization risk within code and containers. Ask the vendor if they can provide a searchable catalogue of the vulnerabilities they detect.

↘ Deployment Options and Other

Single SaaS cloud platform

To properly correlate results across tools, your AppSec platform should be fully integrated into the cloud.

On-premises deployment option

While enterprises are moving to the cloud, there are legitimate reasons for selecting an on-premises solution. Does the AppSec vendor support multiple deployment models, including both cloud-based SaaS and on-premises?

Planned release cycle and documentation

How does the solution manage scheduled release patches and documentation?

Role-Based Access Control

An AppSec platform should support the needs of different teams and stakeholders throughout your organization. Every type of user should have access to the tools and data they need to perform their functions.

Security certifications

Check for compliance standards such as ISO 27001, SOC 2 Type II, and others that are compatible with your organization's requirements.

Reports, Dashboards, And Notifications

AppSec programs live and die by their access to information. The ability to visualize results, correlate findings, and prioritize results – across the entire SDLC from code to cloud - is essential to the functioning of a successful AppSec team. These are the must-haves:

Unified dashboard

One of the primary benefits – and the easiest tests – of a truly consolidated AppSec platform is whether you can see all the vulnerabilities discovered across different tools in the same place. Seeing all your vulnerabilities in one place makes your job of managing, analyzing, and triaging vulnerabilities easier.

Configurable dashboards

Different users or roles within the same organization will want a different view into security data and metrics to make their job of managing, analyzing, or triaging security issues easier. Available dashboards should be configurable to meet the needs of all your users.

Export Data to PDF/CSV/JSON

Data should be easily downloadable whether shared by users or integrated into broader vulnerability management processes and tools.

Historical changes

Compare historical scans and quickly derive the differences both at a high level (new, recurrent, and mitigated vulnerabilities), and in greater detail (location in code).

Executive summaries and dashboards

Your platform should be built to report to management, including the CISO and the Board. It should easily display application rating scores, mitigation and vulnerability trends, and historical data.

Notification alerts

Every team uses different tools to communicate, like email or Slack. Your AppSec platform should be able to send notification alerts through the tools your team uses.

Policy management

An AppSec program is only as strong as its policy. Your AppSec platform should support your program with robust policy management capabilities and notify AppSec teams on significant events.

API support

Your team may already have your own custom-built dashboards or vulnerability management programs. Your AppSec platform should support them with APIs to allow you to funnel data directly into your existing toolset.

Professional Services

Your AppSec platform should provide comprehensive, white-glove support to help you establish, manage and optimize your AppSec program. These are the must-haves:

Onboarding

Your vendor should have experts in place to build a customized onboarding plan, help you understand how long onboarding will take, who needs to be involved, and what the right steps are to proceed.

AppSec team training

If you are building out your first AppSec team, you'll need experts to provide close support and advice. Troubleshooting: Need assistance? Your vendor's services team should be easily available to help you sort through it.

AppSec program consulting

Your platform vendor should have a standardized framework for you to use in building and improving your AppSec program. It should be able to assess your current state and recommend actionable steps for your team to move forward.

Developer workshops

Your platform vendor should offer workshops to help with onboarding, training, and engagement of new and existing developers.

AppSec program management

Your platform vendor should offer a turnkey AppSec service. The managed services team should be able to help set up your program and run it, from configuration and integration to threat modeling, change management, and scanning optimization. It should also be able to plug its own expert AppSec engineers into your existing team to cover gaps in knowledge, skills, or time.

Query tuning and preset optimization

AppSec should be customized to your individual organization. While having a platform work "out-of-the-box" is great, it should offer expert services to help you get the most out of the platform.

General Vendor Requirements

Experience and credibility are paramount especially when it comes to AppSec. While there are many ways to evaluate vendors, here are the best ways to do so:

Leader in the Gartner MQ or Forrester Wave

As with any technology solution, it's often difficult to understand how well a vendor can meet your requirements. A list of promised capabilities on paper can turn out to be different than your experience in practice. A good starting point is an assessment of a leading analyst firm like Gartner and Forrester who has already done the analysis for you.

Years of experience

The technology world is made up of a wide range of companies, from yesterday's startups to veteran vendors. Startups often promise to bring the latest, cutting-edge technologies to market, but suffer in terms of scalability and reliability. A good AppSec platform is built on the combination of depth of knowledge that can only come with experience, combined with continued innovation. You shouldn't take any risks when it comes to AppSec risk management. Look for a vendor that has at least 10 years of experience in securing software development.