BUYER'S GUIDE

Application Security Posture Management

How to Choose the Right Platform to Prevent Risk from the Start

paloalto NETWORKS | CORTEX® CLOUD

Table of Contents

Application security has reached an inflection point.

Pushing code to production used to take months. Today, most organizations do it weekly,[1] putting unprecedented pressure on dev teams to move fast. AI is accelerating that pace, driving a surge in new code—and with it, new risks.

› **By 2030, AI could produce 95% of all code,[2] with a third potentially introducing security issues.[3]**

The reality is, your dev teams are now creating more risk than you can remediate. And if you're using outdated AppSec approaches that wait until production to detect, prioritize, and remediate security, you won't be able to catch up. These solutions address only a small percentage of vulnerabilities and leave behind a growing backlog of persistent risks.

› **Only 10% of vulnerabilities in production are remediated each month.[4]**

It's also not helpful to simply detect risks during development without stopping them in their tracks. When risks aren't prevented from reaching production, developers spend significantly more time fixing issues than if they'd been addressed in code, which slows development times and leaves dangerous security gaps.

› **Fixing issues once they're deployed is far less efficient than remediating before production.**

The impact is clear.

Outdated AppSec approaches are creating gaps that leave organizations exposed and teams overwhelmed. The effects include:

› **Unresolved risk**

› **Increased exposure in production**

› **Expensive remediation**

› **High likelihood of a breach**

1. "Kevin Scott, CTO @ Microsoft: An Evaluation of Deepseek and How We Underestimate the Chinese," 20VC, 2025.
2. "Kevin Scott, CTO @ Microsoft: An Evaluation of Deepseek and How We Underestimate the Chinese," 20VC, 2025.
3. "Every 1 of 3 AI-Generated Code Is Vulnerable: Exploring Insights with CyberSecEval," SOCRadar, January 2024.
4. *The Fast and the Frivolous: Pacing Remediation of Internet-Facing Vulnerabilities*, Security ScoreCard and Cyentia Institute, 2022.

A NEW APPROACH

Welcome to unified application security posture management.

If you want to prevent breaches before they happen, you need a solution that can contextually prevent new risks without compromising developer speed. You also need to be able to prioritize and remediate your existing backlog of issues—at scale.

Application security posture management (ASPM) centralizes and correlates findings from disparate security scanning tools; integrates seamlessly with developer workflows; and has complete context from code, application infrastructure, and cloud runtime. Armed with prevention-first ASPM, you can address risks in unprecedented ways—crafting more targeted prevention policies, prioritizing risk with greater precision, and automating workflows to reduce time to remediation.

- **Prevent issues instead of chasing them**, reducing application risk by up to 90%.
- **Prioritize business-critical risk** based on likelihood of serious impact.
- **Support developers** by centralizing visibility and improving security posture without disrupting existing development processes.

This guide breaks down the critical evaluation criteria you need to choose an ASPM platform that delivers on these promises.

MUST-HAVES

The top 6 capabilities to look for in an ASPM solution.

From visibility and native code scanning to supply chain and workflow, your solution should address the needs that matter now.

paloalto NETWORKS | CORTEX CLOUD

1 Visibility

- ✓ **Comprehensive asset inventory**
  It's crucial to have an accurate, continuously updated inventory of all the application components in your environment.
- ✓ **Contextualized risk mapping**
  The platform must correlate vulnerabilities and misconfigurations with business impact and application criticality.
- ✓ **Real-time monitoring and alerting**
  Continuous monitoring for changes in the application environment enables you to catch new vulnerabilities and emerging threats.

2 Code scanning (SAST, SCA, secrets, IaC)

- ✓ **Accuracy and fewer false positives**
  Highly precise scanning minimizes false positives, preserving developer trust and reducing noise in remediation workflows.
- ✓ **Native integration in the SDLC**
  Seamless integration into developer tools and processes ensures that vulnerabilities can be caught early.
- ✓ **Comprehensive language and framework support**
  Support for a wide range of programming languages, frameworks, and infrastructure-as-code (IaC) configurations enables broad coverage.

3 Open platform

- ✓ **Native integrations and customizable workflows**
  Look for a solution that seamlessly aligns with your existing environments and adapts to specific security requirements without complex workarounds or rigid constraints.
- ✓ **Extensibility and partner ecosystem**
  The platform should allow for the addition of new capabilities and integrations through a strong partner ecosystem.
- ✓ **Extensive scanner and tool integration**
  The platform should seamlessly ingest and correlate findings from an ecosystem of leading native and third-party AppSec scanners so developers can use their preferred tools.

4 Smart prevention

- ✓ **Predictive risk analysis**
  Using threat intelligence and machine learning, the platform should be able to predict potential future risks based on historical data, code patterns, and emerging threats.
- ✓ **Context-aware security controls**
  The solution should be able to apply security controls based on the specific context of the application and the identified risks. This could include the ability to make dynamic adjustments to WAF rules or runtime protection based on observed threats.
- ✓ **Automated remediation guidance and playbooks**
  When vulnerabilities are identified, the platform should provide clear and actionable remediation guidance and, ideally, integrate with automated remediation workflows or playbooks.

5 Intelligent prioritization

- ✓ **Contextual awareness**
  Seamlessly combine code, pipeline, runtime, and application context to prioritize risk based on the probability of exploitation.
- ✓ **Incorporate business context**
  Use business context to prioritize remediating issues that would have the biggest impact if exploited.
- ✓ **Automated event correlation and remediation**
  Automatically group related issues together to remediate risk at scale.

6 Streamlined workflow between AppSec and developers

- ✓ **Integrated ticketing and collaboration**
  The platform should seamlessly integrate with issue tracking systems (e.g., Jira) and collaboration tools (e.g., Slack, Teams) to facilitate communication and workflow between security and development teams.
- ✓ **Developer-friendly integrations**
  Security findings should be presented to developers in a clear, concise, and actionable format within their native tools.
- ✓ **Automated feedback loops**
  The platform should provide automated feedback to developers on their code security practices, helping them learn and improve over time.

What not to do.

Even the most advanced solutions can fall short if they're selected or implemented without a clear strategy. **Avoid these common pitfalls that can undermine your ASPM investments:**

Focusing only on code scanning

It's not enough to simply focus on security within the pipeline. The entire picture of risk needs to be considered, including the security of the pipeline itself and the environment surrounding it.

Mistaking SBOM for supply chain security

An SBOM tells you what's inside your code, not where the risks are or how to handle them. True supply chain security requires identifying vulnerabilities, assessing their risk, enforcing policies, and securing the entire pipeline—from tools to infrastructure.

Ignoring runtime context in prioritization

If you overlook runtime context—like whether a vulnerable component is actively used or exposed—you risk wasting time on low-impact fixes while critical threats in production go unnoticed.

Settling for "Band-Aid" protections

Preventive measures like WAF are patches that can end up masking the problem instead of solving it. What truly needs to be fixed is the underlying issue in the code.

QUESTIONS TO ASK

A guide for the conversation with your vendor.

Use these questions to challenge your vendor and uncover hidden gaps.

UNIFIED VISIBILITY

- Does your solution provide insights into pipeline tools?
- Does it integrate into VCS and IDE?
- Does the tool track scanning coverage across repositories?
- Does it connect context across cloud infrastructure and runtime?
- Does it track connections to sensitive data?
- Does the solution generate SBOMs?

CODE SCANNING

- Will your solution scan code for CVEs, licenses, operation risk, IaC misconfigurations, code weaknesses, and secrets?
- Does it scan code directly in IDE and VCS? Which ones?

END-TO-END CONTEXT

- Can your solution deliver context across code, runtime, real-world attacks, and the business?
- Does it clearly identify code ownership and responsibility for each application component?
- Can it determine if a CVE is actually exploitable in the environment?
- Does it validate whether exposed secrets are still active and posing a risk?
- Can it determine if vulnerable packages are loaded into memory during runtime?
- Does it identify which applications are running in production environments?
- Can it detect internet-facing applications and services?
- Does it monitor actual application traffic and usage patterns?
- Can it identify applications with access to sensitive data?

QUESTIONS TO ASK

OPEN PLATFORM

- Can your platform integrate with other application security tools? If so, which ones?
- Can it enrich the data coming from the open-source platform?
- Does it support multi-scanner integration?
- Is there an option to achieve unified visibility across all of AppSec?
- Is there an option to apply a single policy across all tools?
- Can your platform integrate with Terraform, GitHub, etc.?

STREAMLINING WORKFLOWS

- Does the solution streamline processes so developers and AppSec can improve collaboration?
- Are remediation workflows automated?
- Beyond basic remediation workflows, are there extensive automation options across the security lifecycle?
- Does your platform offer integrations with Jira, ServiceNow, Slack bots, and security orchestration, automation, and response (SOAR) tools?

COMPLIANCE

- Is the solution aligned with standard regulations?
- Does the solution make it easy to see what your compliance status is?
- How easily can you generate compliance reports?
- Can you track progress against compliance goals?

SMART RISK PREVENTION

- Can the solution prevent risks from reaching production?
- Can targeted prevention policies be created that use cloud and runtime context to only block what's necessary?
- Where does the tool block risk?

INTELLIGENT PRIORITIZATION

- Does the solution offer AI-driven capabilities that combine context from code, pipeline, and cloud runtime to prioritize risk based on the probability of exploitation?
- Does the solution use business context to identify issues that would have the biggest impact?
- Can the solution automatically group related issues together to guide remediation at scale?

Make sure your platform includes the following capabilities.

UNIFIED VISIBILITY

- ☐ Delivers real-time visibility into:
  - All repos, pipelines, and images, as well as the relationships between them
  - All users and contributors, to understand user risk
  - All the sources you have for GitHub, GitLab, etc.
  - Sources that haven't been onboarded and therefore aren't secure
  - All tools within your pipeline, to understand the full scope of your technologies
  - Runtime environments
- ☐ Maintains a complete SBOM across code and pipeline
- ☐ Helps you understand the entire attack surface, including access to data and permissions

CODE SCANNING

- ☐ Scans code at different stages
- ☐ Delivers feedback to developers within native tools
- ☐ Has extensive coverage for different coding languages
- ☐ Enables custom rules for code scanning, secrets, IaC, SCA, and SAST
- ☐ Offers one-click fixes

END-TO-END CONTEXT

- ☐ Understands connected risks across code, cloud infrastructure, and runtime
- ☐ Identifies issues that are both reachable and exploitable
- ☐ Incorporates context to prioritize business-critical applications

OPEN PLATFORM

- ☐ Supports integration with different source control managers, CI/CD, tools, registries, IDs, and application security tools
- ☐ Ingests and normalizes data in a single data lake
- ☐ Integrates with leading AppSec tools

STREAMLINING WORKFLOWS

- ☐ Integrates with ticketing systems such as Jira, ServiceNow, and Slack
- ☐ Provides pull requests to developers with full context on what to fix

COMPLIANCE

- ☐ Provides visibility into compliance status
- ☐ Offers easy-to-generate compliance reports
- ☐ Tracks progress against compliance goals

SMART RISK PREVENTION

- ☐ Blocks issues before they reach production—in the PR, in the CI, etc.
- ☐ Differentiates between new issues and backlog
- ☐ Uses context to prevent issues without getting in the way of development

INTELLIGENT PRIORITIZATION

- ☐ Offers AI-driven capabilities that combine code, pipeline, runtime, and application context to prioritize risk based on the probability of exploitation
- ☐ Leverages business context to prioritize issues that affect critical applications.
- ☐ Automatically groups related issues together to remediate risk at scale

SUPPLY CHAIN SECURITY

- ☐ Secures the code going through the supply chain, the pipelines, the tools being used, and the configurations of the tools
- ☐ Delivers visibility into users access across the pipeline
- ☐ Manages a software bill of materials (SBOM)

PREVENTION-FIRST ASPM

Introducing Cortex® Cloud Application Security Posture Management

Application security posture management in Cortex Cloud provides the unified context you need to fundamentally improve how you secure applications. With a complete view of the development lifecycle—where centralized findings from your existing scanning tools are correlated with deep insights from your code, infrastructure, and cloud runtime—you can shift from a reactive to a prevention-first security model.

Prevent up to 90% of vulnerabilities from reaching production. Proactively block risks without slowing development. Deep insight into potential business impact enables you to create intelligent prevention policies that are easy for developers to work with.

Focus on exploitable risks, not distracting noise. Pinpoint the critical threats that truly matter. Correlate findings from across your security ecosystem—with complete code, cloud, and runtime context—and enable developers to fix what's important without changing their tools.

Automate remediation and eliminate manual work. Escape the endless cycle of manual fixes for both security and development teams. Industry-leading automation works at every stage of the application lifecycle, giving you time back to focus on more urgent security issues.

Curious to learn more about Cortex Cloud application security—and see it in action?

REQUEST SOLUTION BRIEF

SCHEDULE A DEMO

Cortex Cloud ASPM

Find and resolve issues faster, stay ahead of threats, and avoid costly fixes in production with:

- Centralized security insights
- AI-driven risk prioritization
- Developer-first security
- Proactive threat prevention

3000 Tannery Way
Santa Clara, CA 95054

Main: +1.408.753.4000
Sales: +1.866.320.4788
Support: +1.866.898.9087