From Gates to Guardrails

A Practical Guide to Preventing Risk at Scale

paloalto NETWORKS | CORTEX CLOUD™

Laying the Foundation for a Mature AppSec Program

Application security programs are designed to achieve two main goals. The first is to help developers identify issues earlier in the software development lifecycle. The second is to reduce the number of vulnerabilities that make it into production where they can potentially become business-impacting problems.

Application security, however, faces the fundamental challenge of enabling prevention without slowing development.

To avoid friction, most programs focus on detecting issues rather than preventing them. The problem with that approach is that issues still reach production, teams are forced to chase them, backlogs continue to grow, and applications remain at persistent risk levels.

A mature AppSec program requires moving beyond simple identification to preventing risk and turning security from gates to guardrails so teams can stop risk, while letting developers move at full speed.

Moving Beyond the Basics

AppSec programs can follow a structured maturity progression to advance from basic visibility and reactive detection to prevention at scale. The early stages lay the foundation for security by focusing on visibility and integrating security directly into development workflows. Later stages leverage automation and continuous feedback to prevent risk before it reaches production.

In mature organizations, AppSec prevention controls function as guardrails that reduce risk while enabling teams to deliver software quickly and confidently.

STAGE 1 Understanding Your Environment

THE FIRST STEP toward a mature AppSec program is visibility. Without a clear view of the engineering ecosystem, security efforts are blind. Organizations must have comprehensive visibility over all repositories, CI/CD pipelines, contributors, and development tools, ensuring teams operate from a single, consistent foundation.

In this stage, AppSec teams should:

- **Gain visibility** across repositories, version control system (VCS) platforms, CI/CD tools, contributors, and other components of the engineering ecosystem.
- **Standardize development tools** to eliminate blind spots caused by code scattered across multiple GitHub, GitLab, or other instances.
- **Consolidate platforms** into a smaller set of approved systems to improve insight into where code resides, how it's built, and how it's deployed.

Establishing visibility and control sets the foundation for all subsequent AppSec efforts, ensuring that security can be both effective and scalable as programs mature.

40%

of cloud incidents stem from unmonitored cloud assets and shadow IT, making lateral movement easier for attackers.

Source: *Unit 42. 2025 Global Incident Response Report,* Palo Alto Networks Unit 42, February 2025.

STAGE 2 **Empowering Security Teams**

WITH A CLEAR VIEW of the development ecosystem, the focus shifts to enabling teams to work more effectively together. Developers need a secure baseline, while security teams must maintain full coverage and be able to trace issues to their source.

This stage is about embedding security into workflows so it becomes an enabler, not a bottleneck. AppSec teams should:

- **Integrate AppSec tools into development workflows.** Provide security checks where developers already work—inside pull requests, integrated development environments (IDEs), VCS, or CI/CD pipelines—rather than in separate dashboards.
- **Ensure comprehensive security coverage.** Scan all repositories with static application security testing (SAST), software composition analysis (SCA), infrastructure as code (IaC), and secret scanners. Also, maintain awareness of all code assets.
- **Provide a secure baseline** through golden (preconfigured and standardized) IaC templates and container images, enabling developers to build safely from the start.

Stage 2 establishes a productive relationship between security and development, creating a shared foundation that puts teams on the same path so they can advance toward prevention-first AppSec.

Maturity Checkpoint

To complete these stages and move on to stage 3, AppSec teams must:

- ☐ Standardize development tools.
- ☐ Integrate security scanning into developer workflows.
- ☐ Track scanning coverage across all repositories and environments.
- ☐ Provide golden container images and IaC templates.

STAGE 3 **Cutting Through the Noise**

BY THIS STAGE, organizations have achieved visibility, standardized tools, and established a secure coding baseline with golden templates for IaC and container images. A substantial security backlog often remains, despite this progress. Before teams implement prevention, they need a clear process for understanding and fixing existing issues. The challenge now is to assess the current risk landscape and define a strategy that focuses on what matters.

Assess the Current Estate

The first step is to assess your environment and identify which issues pose exploitable risks. Teams should archive or exclude dormant repositories and dead code to avoid wasting resources on findings that cannot be exploited.

Prioritize with Context

Effective risk reduction requires context from across the entire application lifecycle to understand which risks are actually critical, including:

- **Code:** Is the vulnerability reachable? Does a known exploit exist? Is this code actively executed in critical workflows?
- **Cloud:** Could sensitive data be exposed? Could this vulnerability enable lateral movement? Are there misconfigurations that increase risk?
- **Runtime:** Is the affected workload heading for production or a test environment? Which application is impacted? Are multiple instances affected?
- **Business impact:** What would the consequences be if this application were compromised? Could it impact customers, revenue, or regulatory compliance?

Applying these layers of context makes it easier for teams to separate the noise from high-priority risks that require immediate attention, enabling smarter, faster, and more effective remediation.

Eliminate Business-Critical Risks

Mature AppSec programs focus resources on the issues that pose the greatest business threat. Prioritization and accountability ensure critical risks are addressed efficiently and keep less urgent issues from draining time or attention.

STAGE 3 Cutting Through the Noise (continued)

Your AppSec team must:

- **Define SLAs:** Ensure critical and high-level issues are resolved within agreed-upon time frames.
- **Escalate missed policies:** Notify leadership when SLAs are not met.
- **Deprioritize noncritical alerts:** Focus effort on issues that pose real business risk.
- **Align resources to risk:** Allocate effort based on code, cloud, runtime, and business impact.
- **Track outcomes:** Measure remediation effectiveness and continuously refine workflows.

Automate as Much as Possible

Automation reduces manual effort, speeds remediation, and ensures consistency across the environment. With routine tasks taken care of automatically, teams can focus on the highest-priority risks:

- **Autoremediate common issues:** Apply automated fixes for recurring vulnerabilities to reduce developer workload and prevent the same issues from resurfacing.
- **Remediate issues at the source:** Implement changes directly in the code, IaC templates, or container images to prevent vulnerabilities from reappearing in future builds.
- **Measure effectiveness:** Track which issues are resolved automatically, identify gaps where manual intervention is still required, and then refine automated processes accordingly.

By automating repeatable remediation tasks, your application security teams can focus on high-impact vulnerabilities, maintain velocity, and continuously improve their security posture. With clear processes and effective prioritization, teams can target reducing their biggest outstanding risks.

Maturity Checkpoint

To complete this stage and move on to stage 4, application security teams must:

- ☐ Ensure outstanding risks are clearly visible across all environments.
- ☐ Prioritize findings by using code, application infrastructure, cloud, runtime, and business contexts.
- ☐ Establish SLAs to ensure critical and high-level issues are resolved within defined time frames.
- ☐ Provide leadership with visibility when SLAs are missed.

STAGE 4 **Preventing Risk**

BY NOW, TEAMS have visibility, standardized tools, and processes to remediate risk efficiently. But remediating issues in production is a losing battle. Backlogs grow faster than they can be addressed, and despite the best efforts of security teams, 90% of cloud risks go unresolved each month.[1]

Differentiating between new and existing issues enables teams to stop the inflow of new risks while continuing to work through the growing backlog. Prevention that blocks new vulnerabilities before they reach production ensures existing issues do not slow down developers while the organization steadily reduces risk.

Turning on Prevention One Step at a Time

Stage 4 demands the most time and discipline. Gradually implementing and fine-tuning prevention controls, and then scaling them across development teams, ensures a lasting impact without slowing delivery.

The most effective approach is to start small, placing guardrails strategically to block new security issues while teams continue addressing the existing backlog. AppSec teams should scale controls incrementally—one repository, one pipeline, one application, one product at a time—enabling the organization to identify what works best before expanding across business units.

By rolling prevention out incrementally, teams can continue refining policies, ensure security controls are effective, reduce false positives, and maintain developer velocity while steadily addressing the backlog. They can also scale prevention gradually by shifting scans closer to the developer by starting in CI/CD pipelines and then moving to pull requests, and eventually integrating directly into IDEs.

1. *2025 State of Software Security: A New View of Maturity*, Veracode, February 26, 2025.

How to Turn on Prevention

STEP 1
Start with Secrets

Secrets are among the first targets for attackers. They provide high-fidelity findings and face minimal developer resistance—everyone recognizes the risk of exposed credentials.

STEP 2
Tackle IaC

Site reliability engineers and DevOps teams are a smaller subset of the developer community (roughly 1:10), so starting here limits disruption. It also enables teams to gather feedback and refine their approach before a broader rollout.

STEP 3
Secure Containers

Because many vulnerabilities come from base images, securing them can have a major impact. One organization eliminated more than 60,000 runtime vulnerabilities by simply updating the base images with secure versions.[2]

STEP 4
Address Open Source

Apply prevention controls to open-source dependencies, reducing risk from widely used libraries.

STEP 5
Extend Controls to Custom Code

Apply controls to custom application code to ensure full-stack protection across all internal development.

Maturity Checkpoint

To complete this stage and move on to stage 5, application security teams must:

- ☐ Put security policies in place to block new issues from reaching production.
- ☐ Implement prevention guardrails across the entire organization.
- ☐ Make measurable progress in shrinking the AppSec backlogs across teams.
- ☐ Ensure fewer new issues are being created.
- ☐ Reduce mean time to remediate (MTTR).

2. Based on internal customer data gathered by Palo Alto Networks.

STAGE 5 Maintaining Long-Term Success at Scale

BACKLOGS ARE SHRINKING, SLAs are enforced, and guardrails are trusted. The program is strong, but maintaining long-term success requires staying ahead of emerging risks and continuously improving security controls.

Now, the focus is on sustaining and scaling prevention across the organization. Security becomes an integral part of development, with automation and measurement ensuring issues are blocked early, controls remain effective, and teams can continuously improve without slowing delivery.

3 Keys to Prevention at Scale

- **Automate remediation**
  - › Reduce the time developers spend fixing issues by automatically resolving common risks.
  - › Automate pull requests to remediate issues at the source, preventing recurrence.
- **Measure relentlessly**
  - › Track how many issues are blocked, bypassed, or resolved.
  - › Measure false positives and tune guardrails accordingly.
  - › Share metrics transparently with developers and leadership to build trust.
- **Continuously improve**
  - › Feed runtime intelligence back into guardrails and update them to prevent new issues.
  - › Treat policies as living controls, not one-time rules.

By embedding automation, measurement, and continuous improvement into daily workflows, organizations can maintain prevention at scale, reduce risk steadily, and empower development teams to move at full speed.

Success in Action

Palo Alto Networks InfoSec team used the stage-based approach that served as the foundation for the guidance in this e-book to transform its own application security program. The internal results underscore the benefit organizations can see by implementing this prevention-first approach:

- 92% of issues were prevented from reaching production.
- 52% faster remediation.
- 90% reduction in developer time spent fixing issues.

Achieving these results required a deliberate, structured journey. By following these stages, organizations can reduce risk systematically, accelerate delivery, and build an AppSec program that scales with their business.

92% of issues were prevented from reaching production

52% faster remediation

90% reduction in developer time spent fixing issues

Take the Next Step Toward Prevention at Scale

No matter where you are on the path to prevention, Cortex® Cloud™ is built to help you move forward. With Application Security Posture Management (ASPM) and AI-driven prevention, the platform unifies code, cloud, and runtime context to:

- Give complete visibility across your engineering ecosystem.
- Prioritize the risks that matter.
- Automate remediation and scale prevention across every team.

See prevention-first AppSec in action. Request a Cortex Cloud demo.

About Cortex Cloud

Cortex Cloud, the next generation of Prisma® Cloud, merges best-in-class CDR with industry-leading CNAPP for real-time cloud security. Harness the power of AI and automation to prioritize risks with runtime context, enable remediation at scale, and stop attacks as they occur. Bring together your cloud and SOC on the unified Cortex platform to transform end-to-end operations. Experience the future of real-time cloud security at https://www.paloaltonetworks.com/cortex/cloud.

CORTEX CLOUD
BY PALO ALTO NETWORKS

AppSec Program Maturity Checklist

A mature AppSec program requires clear visibility, standardized workflows, and intelligent, targeted risk prevention. This checklist highlights the essential practices that enable teams to reduce risk systematically, scaling prevention-first security across the organization to accelerate secure development.

- ☐ Gain complete visibility into applications, repositories, and cloud environments.
- ☐ Standardize development tools.
- ☐ Embed security into development tools and workflows.
- ☐ Track AppSec scanning coverage across repositories and pipelines.
- ☐ Assess the cloud environment and prioritize issues based on code, cloud, runtime, and business context.
- ☐ Define SLAs to remediate critical and high-severity issues.
- ☐ Give leadership visibility when SLAs are not met.
- ☐ Give developers a secure baseline through golden IaC templates and container images.
- ☐ Apply guardrails to block new issues from reaching production.
- ☐ Scale prevention controls incrementally across applications, pipelines, and teams.
- ☐ Automate remediation and pull requests to fix issues at the source in code.
- ☐ Continuously improve guardrails with feedback loops and runtime intelligence.
- ☐ Track and share metrics that measure blocked issues, false positives, and progress.
- ☐ Regularly review and update prevention policies.

paloalto NETWORKS

3000 Tannery Way
Santa Clara, CA 95054

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

www.paloaltonetworks.com

© 2025 Palo Alto Networks, Inc. A list of our trademarks in the United States and other jurisdictions can be found at https://www.paloaltonetworks.com/company/trademarks.html. All other marks mentioned herein may be trademarks of their respective companies.

cortex_eb_modernizing-financial-services-security_111225