snyk

BEYOND THE VULNERABILITY BACKLOG: BUILDING RISK-BASED APPSEC PROGRAMS

EXECUTIVE SUMMARY

Software development is fundamentally transforming, and AppSec programs must change with it. In this eBook, we'll cover the state of application security today, define a risk-based AppSec program according to our core principles, and cover an approachable implementation framework to help you modernize your application security and move beyond the vulnerability backlog.

The state of AppSec today

Application Security teams find themselves at a critical inflection point. While the tools and processes for identifying security vulnerabilities have never been more sophisticated, the ability to effectively prevent, prioritize, and remediate these issues has become increasingly challenging. Overwhelming vulnerability backlogs, disconnected tooling, and a growing divide between security and development teams characterize today's AppSec landscape.

The changing face of development

Software development is undergoing a fundamental transformation. According to our research with ESG, development velocity continues to accelerate, powered by automated testing, streamlined deployment pipelines, and AI-assisted coding in 92% of organizations. What once took months now takes days or even hours. This acceleration is accompanied by increasing application complexity. Modern applications are no longer monolithic systems but intricate ecosystems of microservices, APIs, cloud services, and third-party integrations.

> In 92% of organizations, development velocity continues to accelerate, powered by automated testing, streamlined deployment pipelines, and AI-assisted coding.

This evolution brings unprecedented challenges for application security. Development teams can create and deploy new features faster than ever while applications grow more complex and interconnected. Each microservice, API endpoint, and third-party integration represents potential security risks. The widespread use of open source and third-party dependencies, often automatically suggested and integrated through AI-assisted development, introduces additional attack vectors that traditional security tools struggle to identify. From vulnerable dependencies in the supply chain to insecure API implementations and issues in custom code, the attack surface continues to expand while the speed of development accelerates.

The limitations of traditional approaches

Traditional vulnerability-focused security approaches show their age when stacked against modern development practices. Security teams operate an array of specialized tools — SAST, DAST, SCA, and container scanning — each generating a stream of findings without awareness of what other tools have discovered or the broader application context. This fragmented approach creates multiple challenges.

Security findings exist in silos, disconnected from both development reality and business context. Each tool speaks its own language to a limited audience — SAST findings resonate with application security teams, container vulnerabilities with platform teams, dependency alerts with developers, and security teams must manually correlate findings across different tools to understand the true nature of potential vulnerabilities. Meanwhile, development teams receive security findings as interruptions to their workflow, with little context to help them understand the urgency or impact of reported issues.

The compound nature of modern risk

Today's application security risks are not isolated vulnerabilities but compound threats in dynamic, multi-faceted environments. A seemingly minor vulnerability in a shared service might pose a significant risk when viewed in the context of all dependent applications. API vulnerabilities can have cascading effects across multiple services. Cloud misconfigurations can expose otherwise secure applications to new vulnerabilities, amplifying the potential risk.

This interconnected risk landscape makes traditional approaches to vulnerability assessment and prioritization insufficient. Organizations can no longer evaluate and address security findings in isolation — they must understand how different vulnerabilities interact and their potential combined impact on business operations.

Shifting left is just the first step

While many organizations **(91% in our ESG study) have formal initiatives to implement testing** earlier in the development lifecycle, less than half (46%) have a highly collaborative relationship between security and development. A "shift left" approach alone is not enough to overcome modern security challenges. Early testing is crucial but must be accompanied by a deeper understanding of application risk across various environments.

Shift-left practices often focus on finding vulnerabilities earlier but don't address the fundamental challenges of prioritization, context, and risk assessment. While preventing new vulnerabilities through 'secure by default' practices and pre-deployment checks is valuable (like blocking the introduction of vulnerable dependencies), organizations still face critical prioritization challenges with existing vulnerability backlogs and emerging zero-day threats. The situation is further complicated by disconnected security tools and the growing complexity of modern applications. This makes effective risk-based prioritization essential for managing technical debt and responding quickly to new critical vulnerabilities across the application portfolio.

The cost of inaction

The consequences of maintaining traditional approaches are becoming increasingly more severe. The gap between development and security teams continues to widen. Security teams struggle to keep pace with development velocity, while developers view security requirements as obstacles or distractions rather than safeguards. Without proper context and prioritization, security teams implement broad, heavy-handed gates that needlessly slow down development.

> 83% of developers are willing to secure code if they have the correct tools and training to prevent security tasks from hindering their workflow.

This misalignment has profound implications:

- Security teams spend more time managing tools and triaging findings than addressing real risks. In our research with ESG, 68% of developers felt like security slows them down.
- Developers become frustrated by security requirements that don't reflect their reality.
- Organizations struggle to prioritize or gauge the efficacy of their security efforts effectively.
- Business velocity suffers from unnecessary security friction.
- Teams may miss actual security risks among the noise of less critical findings.

Despite these struggles, 83% of developers are willing to secure code if they have the correct tools and training to prevent security tasks from hindering their workflow. The **status quo is not just ineffective — it's actively hindering organizations' ability to deliver secure applications** at the speed of business. A new approach is needed to align security efforts with modern development practices and business objectives while providing the context necessary for effective risk management.

THE RISK-BASED PARADIGM

While traditional AppSec approaches focus on finding and fixing individual vulnerabilities, a risk-based paradigm fundamentally shifts how we think about and manage application security. This new approach recognizes that vulnerabilities are not merely technical findings to patch but indicators of risk that must be evaluated and managed within a broader application and business context.

Core principles

The transformation to risk-based security is guided by several core principles that fundamentally change how organizations approach application security.

Asset centricity

The risk-based paradigm shifts focus from managing individual vulnerabilities to understanding and securing application assets as complete entities. This **asset-centric approach gives security teams a structured framework for managing their application security program**. By establishing comprehensive visibility into their application assets, teams can better understand what they're protecting, define appropriate security controls, and make informed decisions about resource allocation.

This approach enables security teams to optimize their program's ROI by directing limited resources where they matter most. Instead of spreading effort evenly across all vulnerabilities, teams can prioritize based on the importance of affected assets to the business. This structured approach helps security teams scale their efforts more effectively and demonstrate clear value to stakeholders.

→ *Definition: Application assets are components used to build, deploy, and run applications. For example, code repositories, container images, branches, and third-party dependencies are useful ways to provide a shared language between security and development teams.*

Holistic application risk assessment

Alongside asset centricity, risk-based security requires a comprehensive understanding of application risk. This means building a complete, 360-degree view of applications from build time to runtime. Security teams must understand what vulnerabilities exist and how they interact with application architecture, deployment environment, and existing security controls.

This holistic view combines multiple dimensions of context:

- Application architecture and dependencies
- Build and deployment processes
- Runtime behavior and exposure
- Data flows and access patterns
- Existing security controls
- Vulnerability context and exploitation potential

Business impact integration

With a foundation of asset understanding and holistic risk assessment, organizations can effectively integrate business impact into their security decisions. This means evaluating security issues for their technical severity and potential impact on business operations, revenue, and compliance.

This integration transforms security from a technical exercise into a **business risk management function**. Security teams can better prioritize their efforts based on actual business risk, ensuring their work directly supports business objectives. This alignment helps security teams communicate more effectively with stakeholders and justify security investments in business terms.

Tailored developer guardrails

The combination of asset understanding, holistic risk assessment, and business context enables the creation of intelligent, contextual security guardrails for development teams. These guardrails automatically adjust security requirements based on the actual risk context of code changes, ensuring appropriate controls without unnecessary friction.

Instead of enforcing uniform security requirements across all applications, these guardrails adapt based on factors like:

- The business criticality of affected assets
- Types of changes being made
- Data sensitivity
- Runtime exposure
- Existing controls

This approach helps organizations maintain strong security controls where needed while streamlining processes for lower-risk changes, ultimately enabling faster, more secure development.

Together, these principles create a foundation for more effective application security. They enable organizations to move beyond **reactive vulnerability management to proactive risk management**, ensuring security efforts align with business needs while maintaining development velocity.

Transitioning from vulnerabilities to risk

Risk-based security represents a fundamental shift in how we approach application security management. This transformation affects every aspect of security operations, from assessing vulnerabilities to prioritizing remediation efforts. The change manifests in several key ways across the security program.

Traditional approach	Risk-based approach	Examples
Focus on individual findings	Focus on application asset risk profile	Instead of fixing 50 XSS issues across all apps, prioritizing three critical auth bypasses in a service handling the application's payment system.
Generic severity ratings	Contextualized risk assessment	Upgrading the priority of a "medium" severity vulnerability because it affects HIPAA-regulated data.
Siloed security tools	Integrated security insights	Combining SCA and SAST findings with runtime analysis to identify actual exploitation risks.
Point-in-time security checks	Contextual guardrails	Rather than blocking all pull requests upon a failed security test, implement intelligent rules that only block risky changes.

Key benefits to the organization

Adopting a risk-based security approach delivers transformative benefits that extend far beyond improved security posture. At its core, this approach fundamentally changes how security integrates with and supports business objectives.

Enhanced Dev-Sec collaboration

The most significant benefit is the improved relationship between development and security teams. By providing shared context about application risk, both teams can finally speak the same language. Developers gain a clear understanding of why certain security measures are crucial for their specific applications, while security teams can better articulate their requirements in terms of business impact rather than abstract technical vulnerabilities.

This improved alignment manifests in daily operations. Rather than perceiving security as a bottleneck, it becomes an enabling force that helps development teams understand and manage risk effectively. When security requirements are tied to clear business risks, developers are more likely to embrace them as essential quality attributes rather than arbitrary obstacles. This shared understanding leads to more efficient development processes, with security considerations naturally integrated into development workflows rather than being treated as last-minute checkboxes.

Strategic resource optimization

Risk-based security enables organizations to allocate their limited security resources more effectively. Instead of spreading effort evenly across all applications regardless of their importance, teams can focus their attention where it matters most. This targeted approach reduces risk and demonstrates clear business value from security investments.

The impact on efficiency is substantial. Security teams spend less time investigating and triaging low-impact issues, while development teams face fewer interruptions for non-critical security concerns. This focused approach allows organizations to achieve better security outcomes with the same resources, making security programs more scalable and sustainable.

Accelerated business growth

By aligning security efforts with business priorities, organizations can move faster while managing risk appropriately. Product teams can launch new features more quickly because security requirements are clear and proportional to actual risk. Security teams can provide faster feedback because they understand the business context of their decisions. This alignment enables organizations to seize market opportunities while maintaining appropriate security controls.

Furthermore, the risk-based approach provides clearer visibility into security's impact on business objectives. Organizations can better demonstrate the value of security investments in business terms, making it easier to justify and obtain resources for security initiatives. This improved visibility also helps in communications with stakeholders, from board members to customers, about the organization's security posture.

Long-term program sustainability

Risk-based security creates a foundation for sustainable security growth. As organizations scale their application portfolios and accelerate development, security efforts and patterns can scale proportionally without creating bottlenecks. Focusing on risk rather than raw vulnerability counts helps prevent alert fatigue and maintains program effectiveness even as complexity increases.

This approach also builds organizational resilience. By understanding and focusing on genuine business risks, organizations become better prepared for evolving threats while maintaining development velocity. The improved collaboration between teams creates a culture of shared responsibility for security, making the entire organization more adaptable to new security challenges.

The cumulative effect of these benefits **is a more efficient, collaborative, and resilient organization** that can deliver secure applications at the speed of business. The risk-based approach doesn't just improve security — it transforms how organizations think about and deliver secure software.

IMPLEMENTATION FRAMEWORK

The journey to risk-based security follows a logical progression that builds upon an increasingly sophisticated understanding of your application landscape. Each phase creates the foundation for more advanced capabilities, ultimately enabling intelligent, risk-based decision-making across your security program.

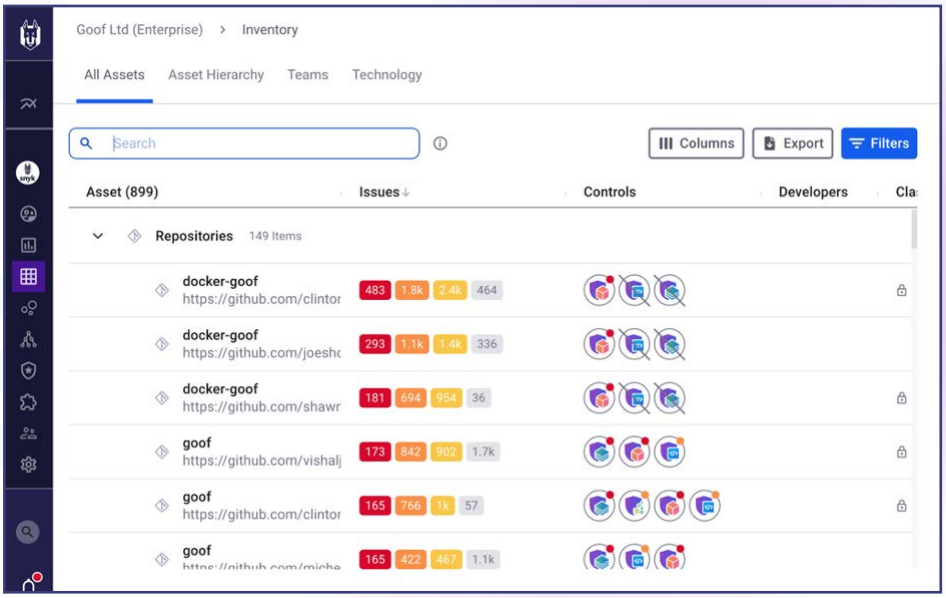
Application asset discovery

The foundation of risk-based security begins with comprehensive visibility into your application landscape. This goes beyond simple application lists to understand the complete technology stack from code to cloud. **Organizations must discover and map all application assets**, including source code repositories, third-party dependencies, container images, development teams, and more.

This discovery process must capture how **these components interact and depend on each other**. Understanding these relationships is crucial — a vulnerability's risk level often depends more on these connections than the vulnerability itself. For example, a seemingly minor Cross-site Scripting vulnerability in a shared authentication service might be classified as medium-severity in isolation but becomes critical when you understand that the service handles authentication for customer-facing financial applications, processes sensitive user data, and could be exploited as an entry point to compromise multiple critical systems.

The scale and complexity of modern application portfolios demand automated and integrated approaches to discovery and mapping. Organizations should implement continuous asset discovery mechanisms that integrate with development pipelines and existing enterprise systems where application information resides. Internal developer portals serve as a crucial source of truth, containing information about application architectures, team ownership, and API relationships. Service catalogs provide additional context about application dependencies and business services.

These mechanisms should automatically detect new repositories and applications, track changes to existing ones, and maintain an up-to-date view of the application landscape as it evolves.



Snyk's asset inventory: Gaining visibility into all your application assets.

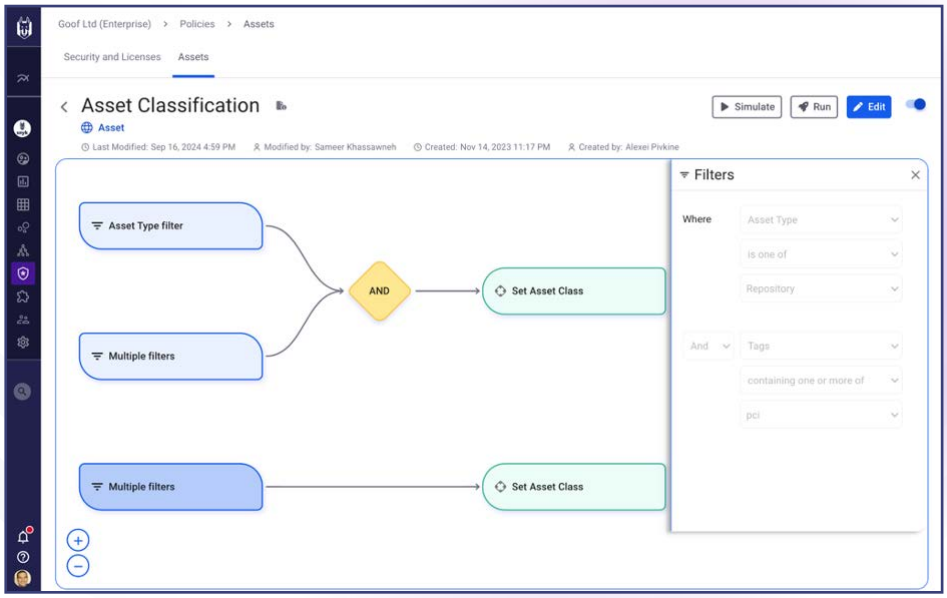
Asset classification

Once you have visibility into your application landscape, the next step is understanding the business context of these assets. This phase involves working with stakeholders to classify applications based on their importance to the business to determine data sensitivity and regulatory requirements.

This classification requires understanding:

- The business processes each application supports.
- The types and sensitivity of data processed.
- The customer impact of security incidents.
- Regulatory and compliance requirements.

This business context enables organizations to make informed decisions about security investments and control requirements. It provides the foundation for risk-based decision-making by establishing the potential impact of security issues on business operations.



Snyk's asset policies: Automatically classify assets based on business context.

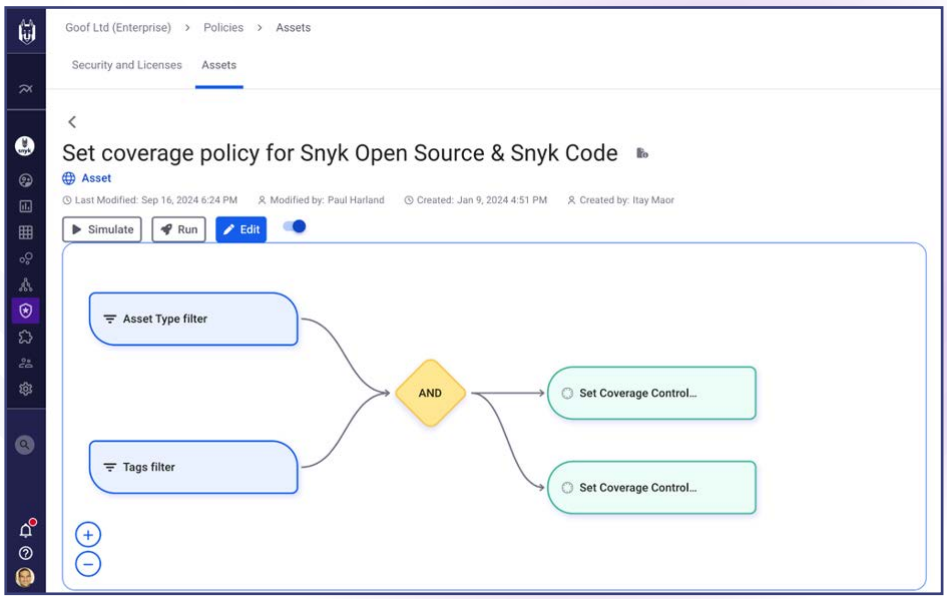
Risk-based coverage strategy

With a clear understanding of application assets and their importance to the business, organizations can develop intelligent coverage strategies that apply appropriate risk-based security controls. This moves beyond one-size-fits-all security requirements to targeted approaches that match security efforts to actual risk.

Coverage strategy development involves:

- Defining security requirements for different risk tiers.
- Mapping appropriate security controls to application types.
- Establishing testing and monitoring requirements.
- Setting validation and acceptance criteria.
- Planning resource allocation.

This risk-based approach ensures that high-risk applications receive comprehensive security coverage while avoiding unnecessary overhead for lower-risk systems. It helps organizations optimize their security investments by focusing resources where they can have the greatest impact on risk reduction.



Snyk's asset policies: Ensure business-critical assets are appropriately secured.

Risk-based prevention, prioritization, and remediation

The combination of application understanding, business context, and runtime behavior enables sophisticated risk-based decision-making. Organizations can now implement intelligent policies that automatically adjust security requirements based on real risk factors.

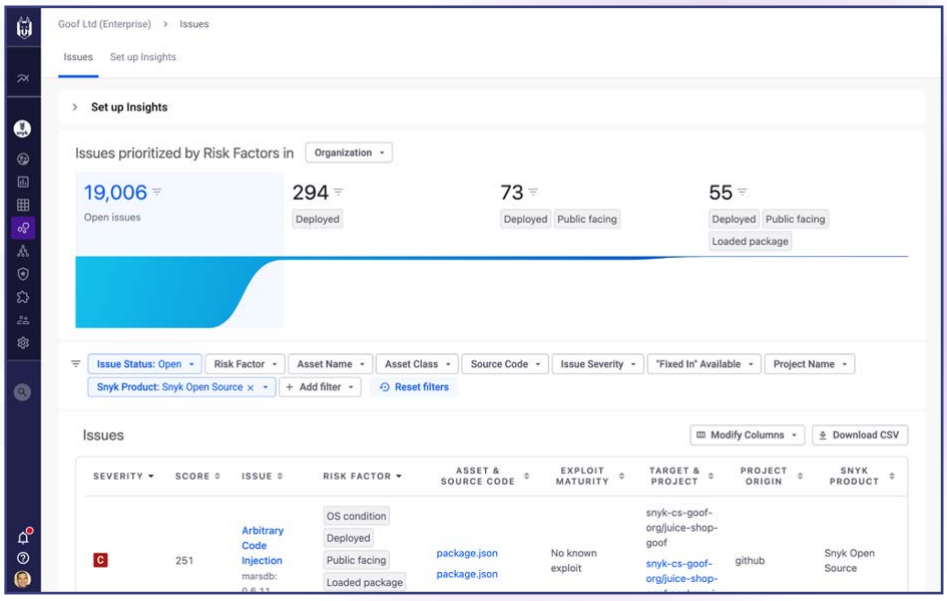
This manifests in two key capabilities:

- Risk-based prioritization that considers technical severity, business impact, and environmental factors when deciding what to fix first.
- Contextual guardrails that automatically adjust security requirements in development workflows based on risk context.

For example, when a developer submits a pull request, **the system automatically determines appropriate security requirements based on:**

- The components that are being modified.
- The business criticality of affected systems.
- Types of data being processed.
- External exposure of the functionality.
- Static reachability analysis showing which other components and data flows might be impacted.
- Existing security controls.

High-risk changes, like modifications to authentication logic in customer-facing applications, automatically trigger enhanced security reviews. Lower-risk changes follow streamlined processes that maintain security without creating unnecessary friction.



Risk-based prioritization in Snyk: Use holistic context to focus on remediating top risks.

Program effectiveness

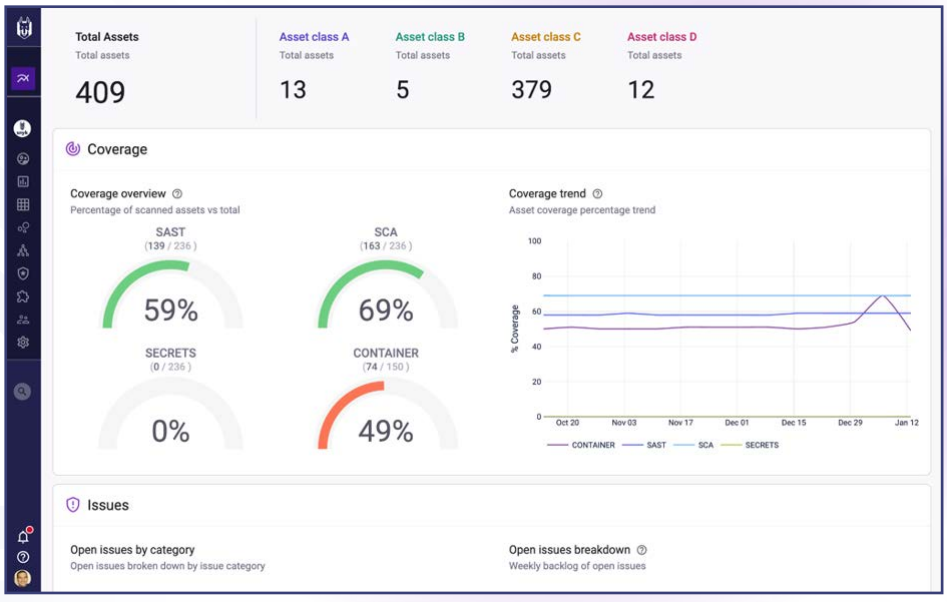
The final phase involves measuring and optimizing program effectiveness. This requires moving beyond traditional security metrics to measures demonstrating actual risk reduction and business value.

Key areas of measurement include:

- Reduction in risk scores across the application portfolio.
- Security coverage effectiveness.
- Developer satisfaction and velocity.
- Resource utilization efficiency.
- Business alignment and impact.

These measurements should drive continuous improvement, helping organizations refine their approach based on actual results and changing business needs.

The success of risk-based security ultimately depends on building these capabilities in sequence, with each phase providing the foundation for more advanced capabilities. Organizations should focus on establishing strong fundamentals in asset discovery and business context before moving to more sophisticated risk-based decision-making.



Tracking coverage effectiveness in Snyk Analytics.

EMBRACING RISK-BASED SECURITY

While adjusting to software's shifting nature is far from new, many organizations are struggling to adapt to the fluid, continuous aspects of modern application security. A risk-based AppSec program can make all the difference. Adopting key principles like asset centricity, holistic application risk assessment, business impact integration, and tailored guardrails represent a fundamental shift in how we approach application security management and allow development and security teams to collaborate more effectively. This change can seem overwhelming, but Snyk is there with you every step of the way.

For easy-to-follow, practical next steps, download our cheat sheet, 5 Steps to Prioritize Based on Risk with Snyk, or book a demo with our security experts to see how Snyk can help you build and enact a custom risk-based prioritization plan.

snyk