PULSEMETER

Techstrong Research

SPONSORED BY Semgrep

Benefits of Secure Guardrails

SECURE GUARDRAILS have emerged as a vital tool for standardizing security measures, enabling security teams to furnish clear guidelines to development and engineering teams across the software development lifecycle (SDLC). By aligning with shift-left practices, guardrails establish specific frameworks and security protocols to safeguard software against vulnerabilities, while also enforcing compliance and policy adherence. This approach not only enhances developer satisfaction and accelerates application delivery but also earns approval from security teams, thereby reducing the risk of catastrophic breaches in production environments.

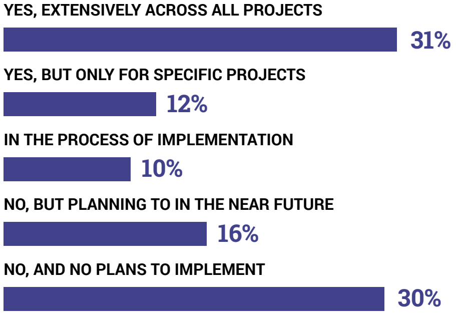
Within the Software Development Life Cycle (SDLC) model, guardrails ensure that every step integrates security, compliance, and policy measures. This involves conducting diverse risk assessments and integrating tools such as Static Application Security Testing (SAS), Software Composition Analysis (SCA), and Dynamic Application Security Testing (DAST) into build pipelines. Additionally, AI-powered guardrail tools offer real-time support to developers by suggesting alternatives when code fails to meet specifications and providing deeper insights into vulnerabilities.

> ***In 2024, Techstrong Research conducted a poll among its community of security, cloud, and DevOps professionals to gauge their perspectives on secure guardrails within software development environments.***

The study aims to assess the effectiveness of automated tools and policies in guiding developers towards secure coding practices, while also examining the balance between automation and developer discretion in mitigating security risks.

Does your organization currently implement secure guardrails in the software development process?

The vast majority either extensively use or are using guardrails across all projects or is in the process of implementing them for certain projects (69%). For those not implementing them yet, many within that group plan to do so in the near future, while the rest are unconvinced, with under a third (30%) not planning to implement guardrails.



The respondents' different roles reflect the breakdown among a typical DevOps organization that has implemented CI/CD, with security engineers (18%), DevOps engineers (19%), platform engineers (6%), and software developers (16%) representing the majority (59%). The different roles reflect how the different stakeholders share security, compliance, and policy concerns that GuardRails are designed to address, aiming for improved business outcomes overall.

TECHSTRONG RESEARCH ANALYST VIEW

The role guardrails play in DevOps teams' DevSecOps and across the SDLC, focusing on enhancing security, compliance, and policy orchestration from the outset of the development cycle, is widely acknowledged. However, despite significant evolution, including advancements in automation and AI, a degree of awareness and challenges persist to achieving consistent and effective implementation.

> ***Guardrail design requires a delicate balance. While they should offer actionable insights and recommendations, they must not overly restrict or impede developers' production process.***

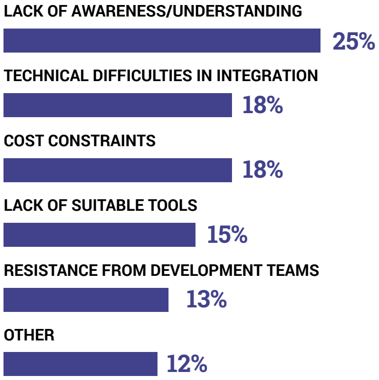
Excessively restrictive guardrails can lead to an abundance of false positives, causing friction for developers and slowing down production due to an overwhelming number of rules and vulnerabilities.

Conversely, ineffective security policies provide little value when critical vulnerabilities emerge in production. Addressing severe vulnerabilities discovered in production often requires more than just patching code; it may necessitate reconfiguring applications without disrupting production — a task that adds extra workload for developers and security teams. Moreover, security gaps in production can be exploited by attackers, posing risks to customer security and damaging the organization's reputation. Fortunately, security guardrail performance is improving.

Guardrails' core principle revolves around empowering engineers to adopt a proactive mindset, emphasizing early intervention in the development process. Using guardrails, software engineers and developers are freer to create code and applications that

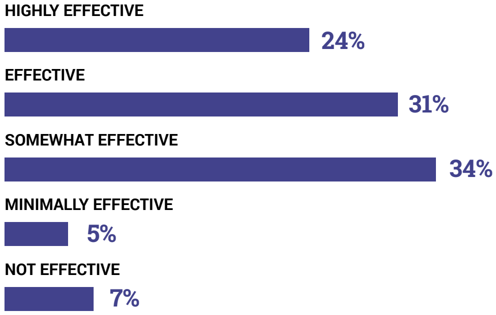
What are the biggest challenges you face in implementing secure guardrails within your development process?

Lack of awareness and understanding, along with technical challenges associated with guardrail adoption, collectively represent the largest challenges, at 25% and 18% respectively. Tied with concerns over technical difficulties are cost constraints.



How effective do you find secure guardrails in preventing security vulnerabilities in your projects?

Guardrails are predominantly accepted as useful for security, as 89% deemed them effective, ranging from 3 to 5 on a scale of 1 to 5. Still, under a quarter of those surveyed gauged guardrails as "highly effective," indicating that work must be done to both improve functionality and educate the community about their value.

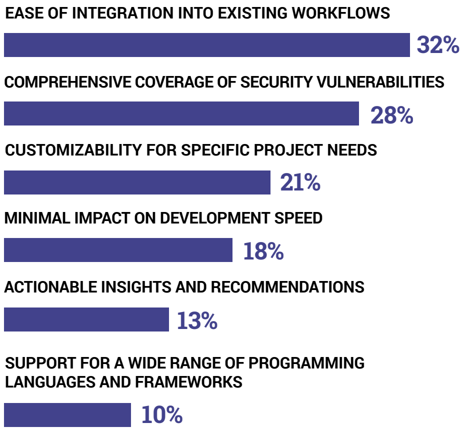


do not impede production cadences by being too restrictive, creating too many false positives, or inundating the developers with too many vulnerabilities to manage. Additionally, security guardrails enable security teams to easily set policies, automate security rules, and monitor and report compliance.

Recent advancements such as secret scanning and AI-enabled assistance offer promising enhancements. Secret scanning addresses concerns about secret disclosure, while AI streamlines vulnerability detection and remediation processes. Instead of receiving a notification when a PR is raised, developers now have the option to engage with AI-enabled assistance.

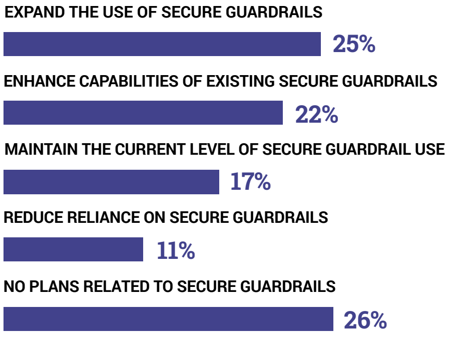
What features do you prioritize in a secure guardrail solution?

Ease of integration and adoption remain the highest priorities cited (32%), reflecting perceived technical challenges in adopting and integrating guardrails. As the breakdown in priorities does not differ significantly, this trend indicates a range of important functions that guardrails offer.



What are your organization's plans regarding adoption or enhancement of secure guardrails within the next 12 months?

The majority plans to continue using or increase their usage of secure guardrails, at 64%. However, a sizable minority, 37%, either plan to reduce their use (11%) or have no plans to adopt secure guardrails. Their hesitancy indicates concerns over their ease of implementation, effectiveness, and other perceived downsides.



Key Takeaways

1. Most organizations consider security, compliance, and policy guardrails as effective ways to implement and orchestrate shift-left practices.

2. Guardrails can scan, analyze and offer fixes for developer code that does not conform to policy, is not compliant, or has critical vulnerabilities.

3. Automation is an essential feature that guardrails must offer. It offers security monitoring and fixes, often with AI for improved and faster remediation, but developers should retain the freedom to reconfigure code manually when required or desired.

4. A guardrail's reach should include comprehensive shift-left capabilities, customization, and easy integration with existing frameworks, libraries, and secrets.

5. Guardrails must offer ease of use and adoption so as not to interfere with developer productivity while operations engineers should be able to seamlessly implement, manage and add new policies, frameworks, and libraries.