SNYK CHEATSHEET

6 BEST PRACTICES FOR DEVELOPER SECURITY IN A FINANCIAL SERVICES INSTITUTION

01

Offer fast, accurate, and relevant security testing.

To keep up with FinTech disruptors, financial institutions must learn to be innovative — and quickly. For application security teams, keeping up with the pace of their developer colleagues is crucial to protecting their bottom line. By embedding security practices into modern development tools and processes, like AI coding assistants and microservices, they can achieve this balance. Here are some effective strategies for securing applications without slowing down development:

- Increase vulnerability scanning coverage for code, dependencies, containers, IaC, and other commonly used software components.
- Use tools that can run tests quickly and accurately. The tools should be able to handle the increased volume of incoming first-party code caused by generative AI.
- Maintain consistent security practices across all your projects by implementing centralized governance and workflows.

Target KPI: Reduced time-per-scan and reduced false positives

02

Embed security directly into existing processes.

To keep up the pace, financial services developers use a single, integrated development environment (IDE) that helps them reach a state of flow as they build applications. Security teams should avoid disrupting this concentration by not forcing developers to switch to a separate security platform. Instead, it's better to integrate security testing directly into the tools and processes they already use. For instance, real-time checks directly within the IDE or at pull requests can enable developers to secure their own code from within their state of flow. This level of security velocity also supports rapid-paced AI code generation by keeping pace with the volume and speed at which AI code gets committed to repositories. By integrating user-friendly security checks this early in the lifecycle, development, and security teams save significant time and resources.

Target KPI: Metrics that point to an increase in developer adoption, such as the number of IDE plugin downloads

03

Simplify the remediation process.

Financial services organizations often face intense pressure to deploy new features quickly, which can lead developers to overlook security alerts to meet deadlines. To prevent this, the security team should focus on removing any obstacles that might make it difficult for developers to address security issues effectively.

One way to do this is by clearly documenting security best practices and expectations and moving vulnerability scanning to an earlier stage in development. Additionally, it's important to use tools that provide actionable guidance. For example, rather than displaying vague error messages or overwhelming developers with long lists of bugs, the right tool should offer clear insights, such as code-specific recommendations or short, educational resources to help developers resolve issues confidently.

Target KPI: Reduced time to remediation for vulnerabilities

snyk

Want to learn how to optimize AppSec for your financial services organization?

Download our eBook to get started.

DOWNLOAD NOW

04

Prioritize fixes based on risk factors.

Many financial institutions have been operating for decades and rely on legacy products that are integral to their operations. However, these products often generate a large number of security alerts, which can be overwhelming. It's crucial to prioritize these alerts effectively so that developers know where to start and understand the importance of each fix.

Security teams should focus on prioritizing issues based on the potential risk to their specific business, rather than relying on generic categories. Consider the following factors when determining which fixes need immediate attention and which can be addressed later:

- The vulnerability's impact on critical business operations.
- The type of vulnerability and its context, informed by trusted security intelligence sources.
- The extent to which the vulnerability appears throughout the application.
- Application-level insights, such as execution data, runtime information, and signals from running containers, etc.

Target KPI: Reduced time-per-scan and reduced false positives

05

Ensure there's an audit trail.

Financial companies must navigate various compliance and regulatory challenges. Keeping track of changes in code, open source licenses, infrastructure, and security processes is crucial for demonstrating compliance with industry standards. To establish a solid audit trail, teams can take the following steps:

- Use security tools that provide detailed audit logs, recording what changes were made, who made them, and the reasons behind them.
- Manage open source licenses to ensure compliance and minimize the risk of violations.
- Utilize a software bill of materials (SBOM) to identify and monitor potential risks in different components, ensuring supply chain security.
- Implement infrastructure as code (IaC) guardrails to increase transparency around infrastructure configuration changes.

Target KPI: The level of visibility for the entire SDLC, such as open source libraries, containers, etc.

06

Facilitate a collaborative relationship between dev and sec teams.

Building strong developer security relies on close collaboration between developers and their security teams. To foster a solid DevSecOps culture, security teams can organize regular meetings that pair developers with their security counterparts. Including security-focused discussions in daily stand-ups and sprint meetings can also be beneficial. To encourage a proactive approach to security, teams can create friendly competition by tracking and sharing security metrics across development teams. Additionally, establishing a security champion program can empower developers with an interest in security, giving them the tools and support they need to raise awareness among their peers.

Target KPI: Measurements related to security education and collaboration, such as training program completion rates and meeting attendance

snyk