SURVEY REPORT

# The Rise of Client-Side Risk and the Trust Gap

New survey from TrendCandy finds that 72% of organizations experienced mobile app security incidents in the past year

THE RISE OF CLIENT-SIDE RISK AND THE TRUST GAP

# Executive Summary

The vast majority of mobile app publishers have experienced at least one mobile app security incident within the past year. The unique "living in the wild" nature of mobile applications has historically presented challenges when it comes to trust-based risks. Once a mobile application is released, it will run on individual devices that the organization does not own, cannot control, and that attackers can manipulate at will. Because mobile apps must operate outside of trusted networks and controls, any and all transactions are inherently untrusted. Accelerated adoption of AI-based coding and testing tools in development pipelines is further widening the mobile app trust gap.

To address this problem, modern mobile app security must evolve from reactive, OS-dependent defenses toward purpose-built security across all stages of the software development lifecycle (SDLC). An approach that combines automated mobile application security testing (MAST) with multiple layers of code protection helps close the trust gap. Capabilities like continuous threat monitoring and server-side protection further support proactive mobile app visibility post-release. In combination, this approach also helps eliminate line-of-business tradeoffs between security, performance, and time to market.

## Contents



## Key Findings

A recent global survey of 1,360 mobile app developers and security leaders revealed:

**72%** experienced mobile security incidents last year.

**65%** reported customer churn/uninstalls due to security issues.

**82%** say exposure to backend/API abuse is increasing.

**79%** cite time-to-market pressure as the top barrier to stronger protection.

**91%** prefer security that covers the entire SDLC.

**96%** of those using multi-layered protection report fewer incidents.

# Introduction: The End of Mobile Tradeoffs

Mobile application developers no longer have to choose between security, release cadence, and performance. But most organizations still operate under outdated assumptions that compromise testing and protection.

This survey, conducted by TrendCandy, reveals the widening client-side trust gap – driven by reverse engineering, tampering, API abuse, and accelerated release cycles fueled by AI-assisted development.

Development teams can now secure mobile apps – without slowing down delivery cycles, hampering performance, or negatively impacting the user's experience – by combining:

1. **Automated testing, multi-layered code hardening, and runtime defenses.**
2. **App attestation to validate every client and protect the backend.**
3. **Continuous threat monitoring, for insights into real-world attack vectors.**

This integrated security model is the emerging standard, and the following survey findings consistently support this shift.

## Who This eBook is For

- Chief Information Security Officers (CISOs)
- Mobile App Security Architects
- Product Development Leaders
- Development / Engineering Managers / Team leads
- Compliance and Risk Officers

## What You Will Learn

- Why outdated perceptions about mobile app security are leaving organizations vulnerable and leading to customer churn.
- The biggest challenges slowing down mobile security adoption — from skills gaps to culture barriers.
- The impact of AI-generated code on vulnerabilities and developer confidence.
- How the combination of integrated testing with multi-layered code hardening and runtime security checks can dramatically reduce security incidents.
- How application attestation ensures the mobile app interacting with your APIs at runtime is authentic, untampered, and safe to trust.
- Practical solutions for applying effective security throughout the mobile app SDLC, without compromising speed, performance, or user experience.

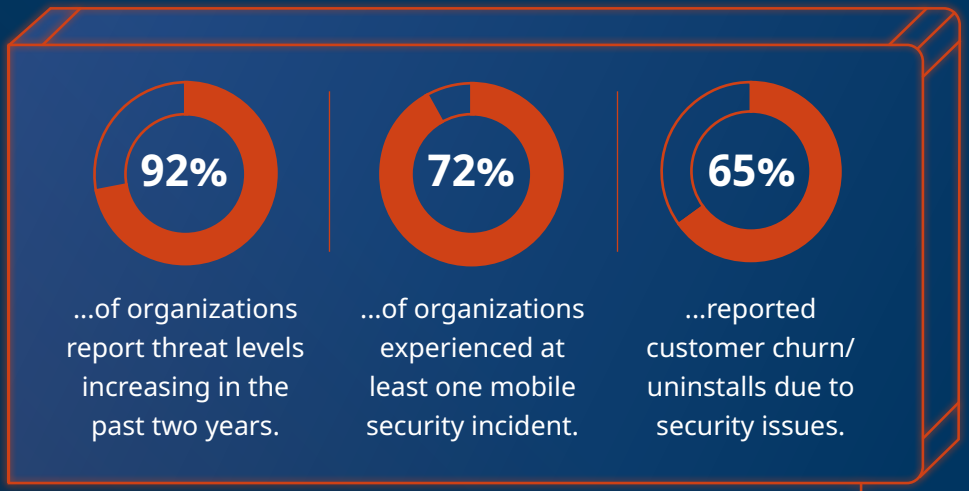
**Survey Methodology**

Guardsquare commissioned independent research firm TrendCandy to survey 1,360 mobile app developers and security leaders globally, across multiple countries on the topic of mobile app security. The margin of error for this study is +/-2% at the 95% confidence level.

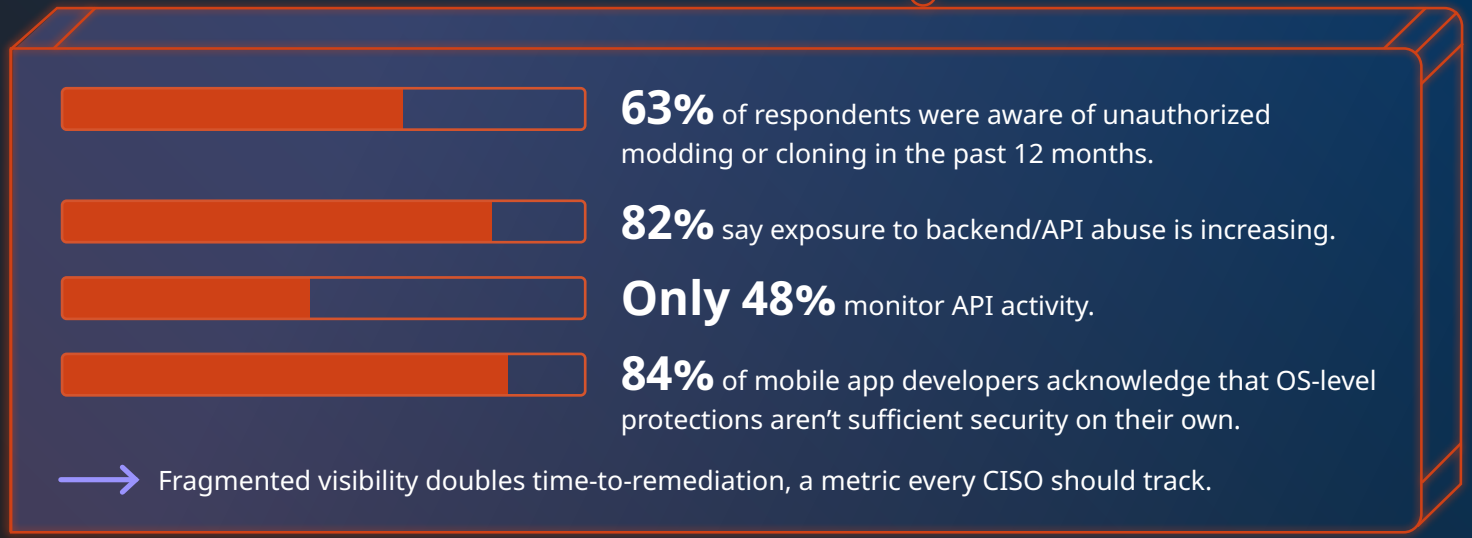
THE RISE OF CLIENT-SIDE RISK AND THE TRUST GAP

# The Cost of Compromise: Incidents, Churn, and the Trust Gap

The observed levels of threats targeting mobile applications are on the rise and nearly three-fourths of organizations have experienced one or more confirmed security incidents over the past year. For about two-thirds of mobile app developers, security issues led to customer churn or uninstalls.




Attackers bypass OS protections by weaponizing the app itself. Insufficient trust-based security is expolited throughout the software development lifecycle to reverse engineer, modify, and repackage maliciously cloned apps – which then target APIs and exploit backend data and services.

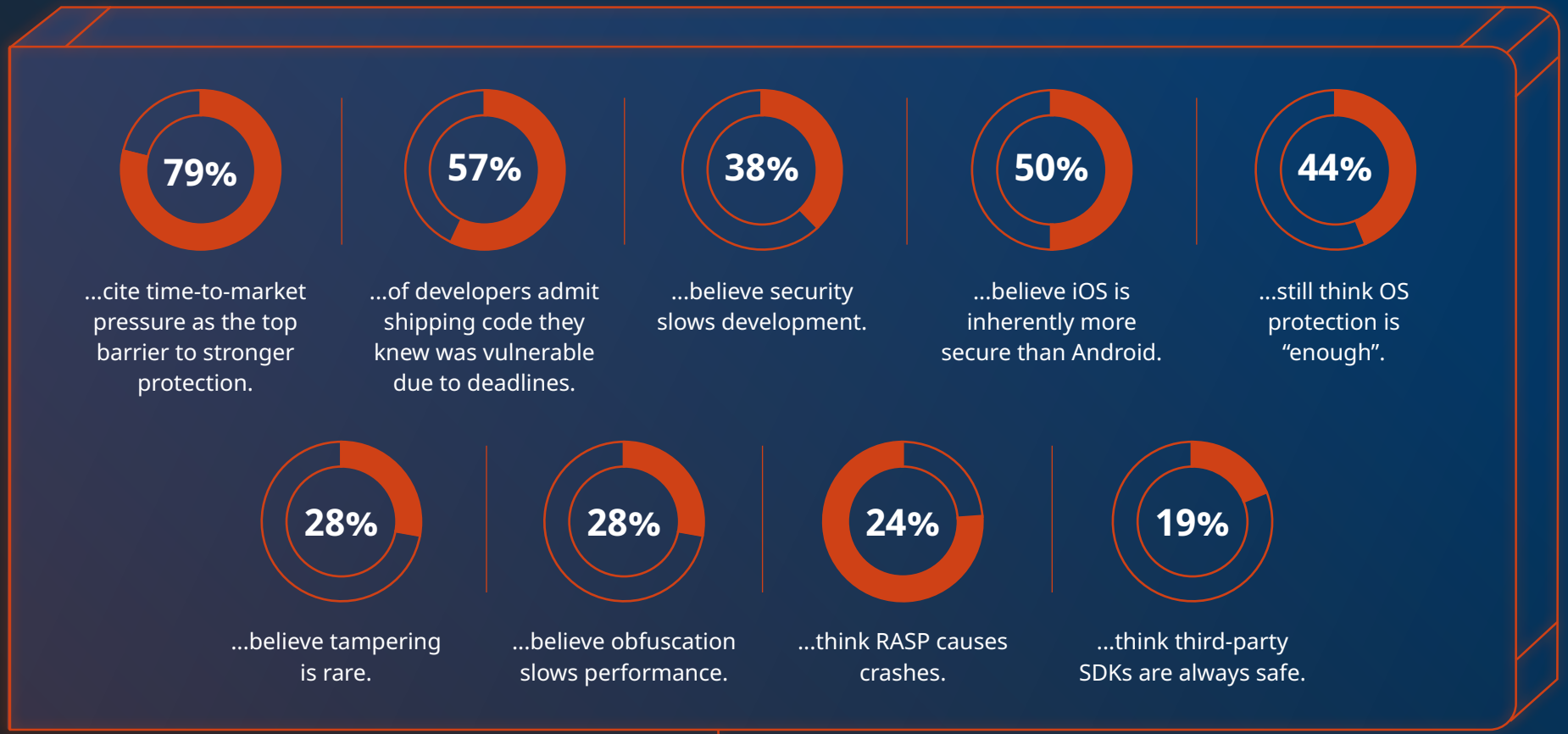



**KEY INSIGHT:**

*Widespread incidents expose the limitations of current approaches to mobile application security. The combination of testing (MAST) plus protection ensures a secure mobile app.*

# Why Compromise Exists: Myths, Misconceptions, and Market Pressure

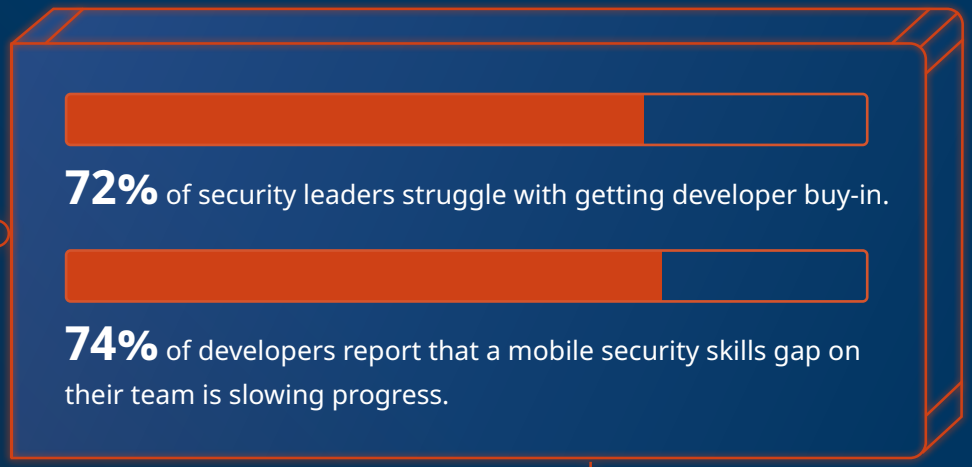
Mobile app developers know that security is important, but speed is perceived to be more important. Dev teams are under intense line-of-business pressure to accelerate delivery cycles. While trading off comprehensive mobile app security (i.e., continuous testing and multi-layered protections) in favor of time-to-market is often a conscious decision, it's often based on outdated myths and misconceptions about security as a pipeline bottle-neck, a drag on application performance, or a disruption to user experience (UX).



## THE RISE OF CLIENT-SIDE RISK AND THE TRUST GAP

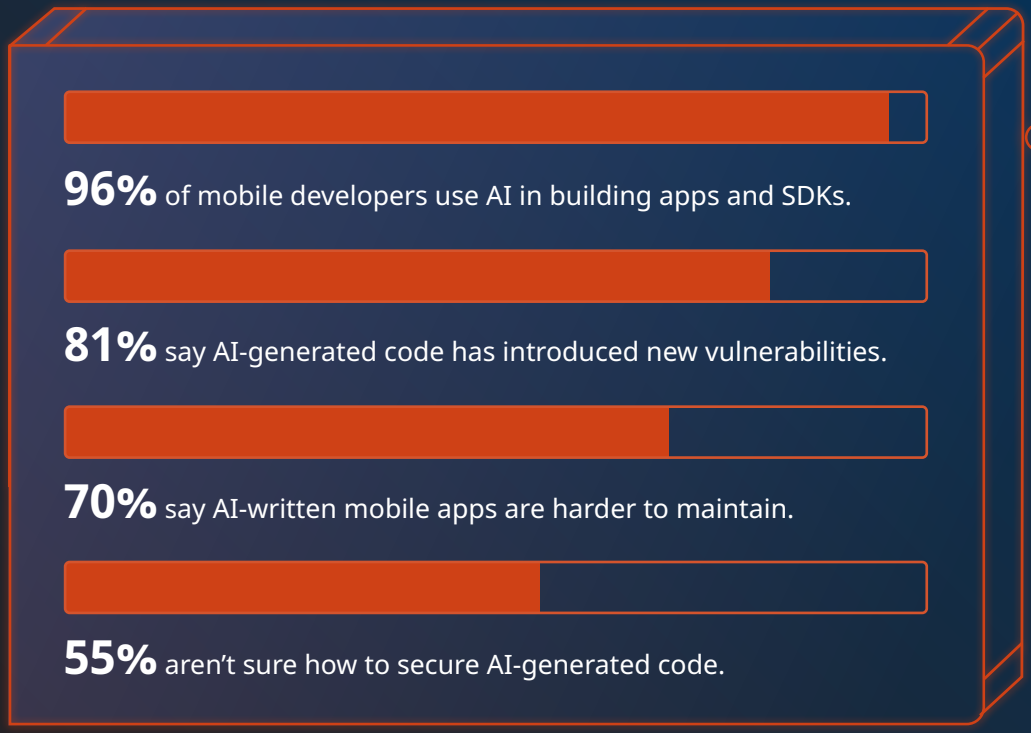
Cultural barriers between developers and security leaders as well as a lack of skilled security professionals within many organizations contribute to outdated awareness of both current mobile app risks and the latest security remedies.

**KEY INSIGHT:** *Opposition of security and development priorities persists within organizations due to real business pressures, false cultural perceptions, limited resources, and outmoded DevOps practices.*



**72%** of security leaders struggle with getting developer buy-in.

**74%** of developers report that a mobile security skills gap on their team is slowing progress.



**96%** of mobile developers use AI in building apps and SDKs.

**81%** say AI-generated code has introduced new vulnerabilities.

**70%** say AI-written mobile apps are harder to maintain.

**55%** aren't sure how to secure AI-generated code.

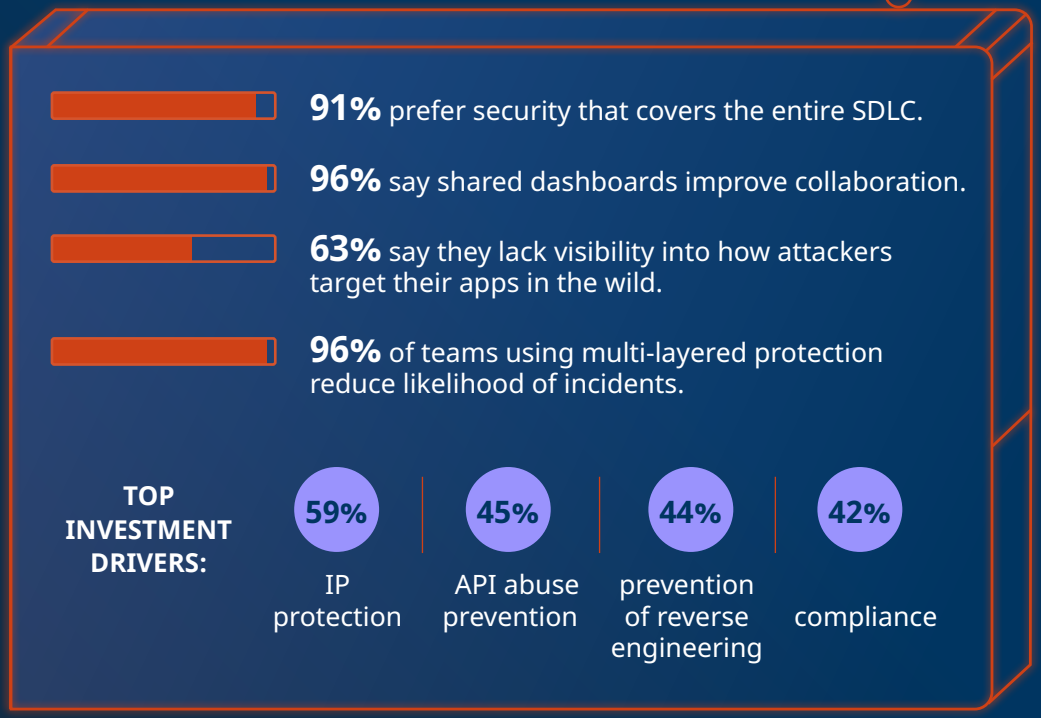
Market pressure has also hastened the widespread adoption of new AI-based tools to further accelerate development cycles. However, the vast majority of respondents report that AI-generated code introduces new vulnerabilities and makes mobile apps harder to maintain. More than half of developers admit that they're not sure how to secure AI-generated code.

**KEY INSIGHT:** *Developers are implicitly trusting AI tools out of necessity. But AI adoption without effective safeguards amplifies existing mobile app security risks.*

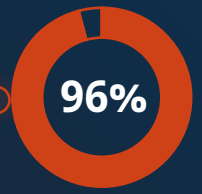
# What Teams Actually Want:
# A New Standard of Testing and Protection

Survey responses from 1,360 mobile application developers and security leaders show a clear and overwhelming desire for integrated security that supports development across all stages of the mobile SDLC, from development and testing through deployment and maintenance. The vast majority of respondents agree that they need tools that help reduce vulnerabilities, facilitate better collaboration between security and development, and provide comprehensive protection against key risks to mobile apps and the broader organization.

**KEY INSIGHT:** *Organizations want comprehensive security (integrated mobile app testing and multiple layers of protection) that fulfills the promises of DevSecOps to proactively avert the risks of IP loss, mobile app tampering/clone attacks, and the consequences of non-compliance with regulations.*

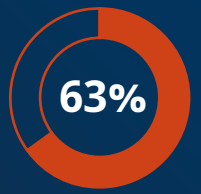



Several survey data points highlight the acknowledged value of multiple layers of mobile app protection to reduce security incidents, as well as a need for greater reinforcement in the areas of reverse engineering and tampering attacks.




...using multi-layered protection report fewer incidents.

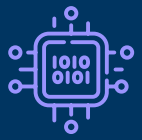

...of SDK devs already apply multi-layered protection.




...noticed modding/ cloning despite partial protection.

## THE RISE OF CLIENT-SIDE RISK AND THE TRUST GAP

Multi-layered, polymorphic protections applied across the mobile SDLC include:



**Code hardening:** Mobile application protection starts with multi-layered code hardening to block static analysis of a reverse engineering attack or attempts by a threat actor seeking to extract secrets or sensitive information related to authentication, transactions, and in-app purchases. An effective solution should use multiple techniques of hiding sensitive code, including name obfuscation, control flow obfuscation, code virtualization, and data encryption.

**Runtime defenses:** Runtime application self-protection (RASP) counters dynamic analyst attacks via built-in security checks inside the mobile app code. RASP monitors the app's behavior in real time and provides automated defensive responses. These include anti-debugging capabilities that respond to an attack by terminating the app or restricting its functionality. It also blocks common tampering attacks, such as function hooks, dynamic instrumentation, or other code changes. RASP can take action against tampered system libraries, emulators, virtual environments, and root privileges. These runtime defenses can also defend against common overlay attacks, a11y abuse, and other sophisticated malware techniques.

**Application attestation:** App attestation can prevent API abuse by verifying that every client-side app is authentic, unmodified, and running on a secure device. This helps to enforce dynamic security policies that automatically block bots and cloned/modified apps from gaining access to backend application resources. Both Google and Apple offer OS-based app attestation services to help developers protect their apps. These can reliably detect tampering with the app package on clean devices, but they fall short against more sophisticated real-time attacks.

**Continuous threat monitoring:** Once a mobile app is released to the public, developers and security teams need real-time visibility to track emerging threats, stay ahead of malicious actors, and proactively limit the impact of an impending attack. An effective threat monitoring solution should help teams visualize real-world telemetry gathered by the application's RASP and app attestation solutions to help reinforce protection via maintenance and version updates. Threat monitoring reports should include detailed reasons why a detection was triggered and a clear explanation of the risk.

**KEY INSIGHT:** *Multi-layered protection is not "extra"—it's the new baseline.*

# The Low-Down on Security (Without the Slow Down)

While 79% of respondents cite accelerated time to market pressure as a barrier to stronger protection, and smaller groups believe obfuscation slows performance (28%) and that RASP causes crashes (24%), these lingering misperceptions aren't aligned with what organizations want or what they currently can access in terms of available solutions.

The vast majority of teams (91%) want automated mobile app security across the entire SDLC with tools to facilitate better visibility and collaboration between security and development disciplines.

The criteria for selecting a "no-compromises" mobile application security solution (one that supports development speed, app performance, UX requirements, and comprehensive security) should include:

**Integrated and automated testing (MAST)** tools that scan code for potential issues without disrupting development workflows. Rigorous testing should happen early and often throughout the development, monitoring, and maintenance phases of the SDLC. The earlier code issues are discovered in the development process, the less time and cost it takes to remediate them.

**Multi-layered, polymorphic protections** that are applied at every build (code hardening + RASP) and that protect every session (app attestation)—*with zero development slowdown.*

**Continuous threat monitoring** that detects suspicious activity in real time to help proactively maintain and adjust defenses against evolving attack strategies.

Most mobile app developers know they're shipping vulnerable code, correlate backseat security to increasing incidents, and even understand what kind of tools they'd need to address the problem. But to reap the full benefits of an accelerated mobile app delivery cycle, organizations need security that can ride up front with time-to-market.

GUARDSQUARE

Mobile application protection

www.guardsquare.com