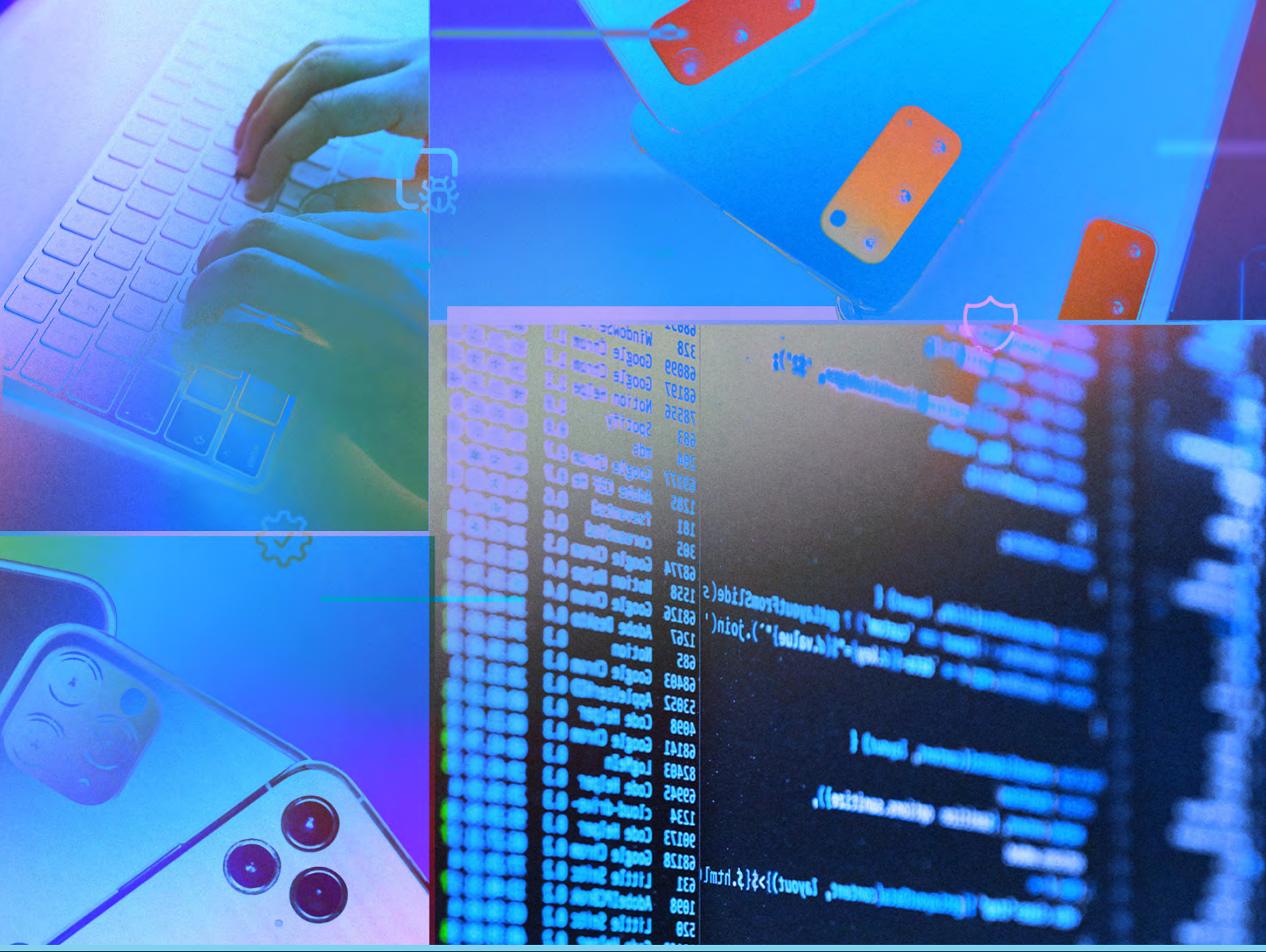


A Comprehensive Guide to **Mobile App Security**

GUARDSQUARE
Mobile application protection

A Comprehensive Guide to **Mobile App Security**

Table of **contents**

Introduction

Mobile apps are integral to our daily lives in today's fast-paced, tech-focused world. Hop on a train or grab a seat in a crowded café, and you'll find yourself surrounded by some of the 3.8 billion global smartphone users, all using mobile apps for business and leisure.

Most of the time we spend with our smartphones today centers on mobile apps. In fact, most of the time we spend consuming any type of digital media happens on mobile apps. And it isn't just for fun: Apps serve as engines for global commerce and business. They make us more productive and keep us connected.

There are 2.87 million apps in the Google Play Store as of this writing, and another 1.96 million in the Apple App Store. The total number of app downloads is set to hit 288 billion in 2024. Alongside usage growth, mobile app revenue is projected to reach $935 billion by the end of 2023.

The businesses that build and maintain mobile apps know just how valuable they are. Many are also all too familiar with the risks that can come with mobile app success.

The responsibility of mobile app security

Indeed, with great success comes great responsibility. The threat landscape for mobile apps is continually evolving, presenting publishers with immediate protection challenges such as app repackaging, code injections, IP and sensitive data theft, fraud, and unauthorized redistribution. There are also the larger security challenges that extend beyond the immediate protection concerns — such as exploitation of vulnerabilities and weaknesses in app design — which must be addressed. Mobile app users also face familiar threats like phishing and malware attacks alongside fresh threats bolstered by malicious actors' "innovative" use of AI as well as the evolution of existing attack methods.

Prioritizing mobile app security is not just about protecting revenue; it's about safeguarding user and customer data and trust, protecting intellectual property, and complying with ever-growing and evolving regulatory requirements.

In this eBook, we'll explore what it takes to secure mobile apps properly, how responsibilities must be divided among mobile app stakeholders, best practices in selecting and implementing mobile app security tools, and the key processes and cultural mindsets that must underpin any strong security program.

By the end of this eBook, you'll have a clearer sense of where to start, how to progress, and how to become more confident that your valuable mobile apps are, indeed, secure.

What is mobile app security?

While anyone working in the tech industry, including the mobile sector, likely has a basic understanding of what "mobile app security" means, it's still worth exploring the concept in some detail.

Mobile app security — done well — requires a multi-faceted approach to protect apps from potential exploitation by threat actors. These threat actors can include cliché "hacker in a hoodie" types looking to make money via app exploitation, nation-states seeking to steal valuable secrets, security researchers showcasing an app's vulnerability for clout, and everyday users who "just" want to score in-app purchases for free.

These threats can lead to severe consequences, as mentioned in the introduction — ranging from loss of revenue to compromised user trust, public reputational damage, regulatory fines, and more.

It's essential for app publishers (and everyone else involved in the mobile app ecosystem) to understand what's at stake. From there, the necessity of adequately securing apps should be abundantly clear.

In a more concrete sense, mobile app security is a holistic strategy that includes protection (specifically against static and dynamic attacks), app testing, and threat monitoring. Most importantly, each portion of the strategy should be applied throughout the software development lifecycle (SDLC).

To achieve comprehensive mobile app security, it's vital to embrace education (internal and external), awareness (of both threat types and security best practices), adherence to evolving security standards and regulations, and the careful selection and implementation of appropriate security tools.

If that sounds overwhelming, this eBook will break it down into manageable pieces so that teams of any size can prioritize and implement mobile app security. The path to mobile app security can seem dark and winding, but we're here to shine a light and guide you through.

Roles and responsibilities in mobile app security

You've likely heard of the "Shared Responsibility Model," initially popularized in cloud security. The model functions similarly in the mobile app security sector, with security professionals embracing distinct roles and responsibilities in securing mobile devices and applications.

Every participant plays a crucial role in ensuring the security of mobile apps.

The shared responsibility model for mobile apps

STAKEHOLDER	SECURITY INTERESTS	RESPONSIBILITIES	BEST PRACTICES & TOOLS
Consumer / end user	Personal data protection	Regularly update apps, use secure connections, report issues	• Stay informed about app permissions • Only use reputable app stores • Do not jailbreak/root or use jailbroken/rooted devices • Adopt secure pass
App developper / publisher	Brand reputation and user trust, revenue, regulatory compliance	Implement robust and appropriate security and protection measures, comply with guidelines and regulations, prevent modded apps, and provide clarity around security practices to users and other stakeholders	• Implement static and dynamic protections • Adhere to OWASP standards • Conduct regular security testing (both MAST and pentesting) • Monitor live apps continuously • Improve security with every new release
App protector vendor	Reputation among app developers and publishers, market competitiveness, comprehensiveness, and quality of solution(s)	Provide robust security solutions, collaborate with developers, provide education and resources to app developers and publishers	• Develop cutting-edge security tools • Appoint dedicated internal security team for threat landscape research
App store (e.g., Apple App Store, Google Play)	User trust, brand integrity	Enfore security standards, review and curate apps	• Conduct thorough app reviews to ensure a safe, trustworth app store • Communicate about security standards and updates with app publishers/developers and end users

Operating systems (e.g., Android, iOS)	Overall system security	Develop secure systems, release timely security updates	• Provide a modern, secure platform for developers • Continue to improve developer security controls
Device manufacturer	Device integrity, user trust	Produce secure devices, support regular updates	• Comply with industry security standards • Collaborate with OS providers • Implement hardware security features (e.g., biometrics, secure hardware-backed storage)
Third parties (e.g., API providers, library providers)	Data integrity	Ensure secure integration, adhere to security protocols	• Conduct security audits • Continuously monitor for threats and vulnerabilities • Provide secure APIs and libraries • Communicate and collaborate with app developers

Adopt a security-first mindset

If mobile app development is a core aspect of your business, adopting a security-first culture is not just a best practice but a necessity. As cyber threats become more sophisticated and user expectations for privacy and data protection rise, app developers and publishers must prioritize security from the outset. Let's delve into the concept of a security-first culture and explore its implications for mobile app development.

Laying a secure foundation

A security-first mindset prioritizes security as a foundational aspect of the entire SDLC, rather than an afterthought or a checkbox to tick at the end. It involves fostering a culture where every member of the mobile app development team — architects, coders, testers, project managers, and security professionals — considers security an integral part of their responsibilities.

We recommend ensuring that even less- or non-technical teams like sales, marketing, and customer service understand the importance of mobile app security and follow company policies so no one becomes a weak link in the chain.

The benefits of a security-first culture

- **Proactive risk mitigation:** Identifying and addressing potential security risks early in the development process can significantly reduce the likelihood of security incidents and vulnerabilities making their way into the final product.
- **Cost savings:** Addressing security issues in the initial stages of development is more cost-effective than dealing with breaches or vulnerabilities after publishing the app. Security-related fixes post-launch are more expensive and can result in reputational damage and loss of user trust.
- **Faster time-to-market:** A security-first mindset doesn't necessarily mean a slower development process. Incorporating security practices early in the app development process helps teams avoid last-minute security-related delays and ensure a smoother, more predictable release schedule.
- **Enhanced customer trust:** A Users are becoming increasingly aware of security and privacy issues, and their trust in a mobile app is closely linked to how well it protects their data. A clear, well-articulated security-first posture helps build and maintain trust, leading to better user retention and more positive reviews.

Critical components of a security-first culture

Here are the primary components to consider when building a security-first culture.

Education and awareness

- **Training:** Ensure all team members receive training on the latest security threats, best practices, and compliance standards.
- **Awareness programs:** Conduct regular awareness programs to inform the team about evolving security challenges and strategies. Leverage Cybersecurity Awareness Month (October) and free resources provided by OWASP to support team-wide security competency.

Collaboration across teams

- **Cross-functional collaboration:** Foster collaboration between development, operations, security, and beyond. Security is a shared responsibility that cuts across various roles.
- **Executive buy-in:** At many companies, security is a C-Suite and board-level concern. If that's not the case at your organization, now's the time to involve those decision-makers in your company's mobile app security strategy.

Security champions

- **Appoint security advocates:** Designate security champions within the development team and other functions across the organization. These advocates can promote mobile app security best practices and act as points of contact for security-related questions or issues.

What goes into a comprehensive mobile app security strategy?

Robust mobile app security begins with a complete and integrated approach. Here's what we mean: Historically, app developers relegated security to the final stages of the mobile app development process, leading to potential delays and compromises.

To avoid such pitfalls, having a comprehensive security plan before development is essential; it aligns security and development teams on priorities from the start.

Mobile security means knowing your risks

Start with identifying your organization's mobile app security needs based on understanding your industry, app types, and security threats.

- **App-specific security needs:** Assess the types of data your app handles (from personally identifiable information (PII) to credit card data to protected health information). Then, evaluate its unique construction and potential vulnerabilities.
- **Industry-specific concerns:** Recognize and address specific threats and compliance regulations within your industry. For example, HIPAA in the healthcare industry and PCI in the mobile payments space.
- **Best practices and standards:** Review and understand established security standards, especially the OWASP Mobile Application Security Verification Standard (MASVS) and any industry-specific regulations that apply to you (see previous point).
- **Thread landscape analysis:** Understand potential vulnerabilities and attacks relevant to your app. For example, mobile game apps face unique threats from attackers who want to distribute cheats or obtain in-app purchases for free. We'll cover this in detail and provide specific threat modeling resources below.

To summarize, a security plan must start with understanding general and specific security best practices as well as any unique considerations that apply to your mobile app(s) as a result of the types of data you collect and store, customers who use your app, or the industry you operate within. No mobile app security plan should look identical because every app and publisher has unique considerations.

Common misconceptions about mobile app security

MYTH	REALITY
The app publisher doesn't need to worry about security if the mobile device and OS are secure.	Protected infrastructure does not equate to a protected app. Secure communication protocols may not cover vulnerabilities within the app's code itself, exposing it to malicious threat actors.
The server is fully protected, so the app publisher doesn't need to worry about security.	A secure server doesn't guard against in-app vulnerabilities. The app may be susceptible to client-side attacks, bypassing server security measures.
The app is "thin" and doesn't require protection.	Even thin apps are vulnerable to code decryption and repackaging attacks.
Incorporating security will compromise the performance of a mobile app.	When done correctly, quality mobile app security solutions seamlessly integrate into your app without significantly compromising performance.
Security is just a series of checkboxes to tick off at the end of the app development process.	Considering security at the end of the development process leaves your mobile app open to serious vulnerabilities, often causing publishing delays and increased development costs.
The app is safe because it's only distributed through verified app stores.	While it's safer for end users to choose apps distributed through a verified app store, the app itself isn't safer from a developer's standpoint. In fact, apps in verified app stores are a bigger target for reverse engineering since threat actors often search these stores for valuable and interesting apps.

The 6 core components of mobile app security

Now that we've defined mobile app security, covered the shared responsibility model, and explored how to build a security culture, let's get more tactical. To implement a comprehensive mobile app security program, follow these five key steps:

1. Risk assessment and threat modeling

You must identify potential weak points before you can fortify your mobile app. Depending on your product development processes, this can be done before you even code the app. If your app is already built, taking a step back and performing threat modeling is still valuable.

So, what is threat modeling?

Threat modeling enables prioritization of the most pressing, relevant threats by helping you think like an attacker would to understand threats better and develop a mitigation strategy. Common frameworks used for threat modeling include STRIDE, PASTA, Trike, and Attack trees.

Choose a threat model framework based on your app's security needs, industry-specific requirements, and considerations outlined in the "Know Thyself" section.

Then, based on the chosen threat model, prioritize threats by importance, likelihood of occurrence, and the current threat landscape.

Some helpful resources for threat modeling include:

- OWASP's Threat Modeling Overview
- CMS's Threat Modeling Handbook

2. Embracing security standards

Think of security standards as the foundation for your security posture. Embracing widely recognized guidelines, such as the ones outlined by OWASP and frequently mentioned throughout this eBook, ensures that you're following industry best practices and the most recent recommendations of mobile app security professionals.

These standards provide a solid foundation for building multi-layered and resilient security practices in your app. As alluded to earlier, there is a difference between security standards like those proposed by OWASP for mobile apps and regulations that must be followed. OWASP has seven security categories, including one for resilience. They also provide app publishers a baseline for maintaining security across app builds.

Again, the unique attributes of your organization and app(s) will determine which regulatory measures you must comply with. The good news is that there is a lot of overlap across various frameworks and regulations, so implementing each doesn't necessarily mean starting from scratch every time.

Embrace security standards early, adhere to them consistently, and stay up to date with any changes to ensure you mitigate both threat-based risks and the consequences of non-compliance.

3. Integrating security into the SDLC

Security should never be an afterthought or secondary consideration; it should be a permanent feature of your development processes. By seamlessly integrating security measures into every phase of the SDLC, you're weaving a safety net that catches potential vulnerabilities before they become major headaches.

This practice has come to be known as DevSecOps. DevSecOps is a holistic approach that combines development, security, and operations. This evolution of DevOps aims to bridge the gap between mobile app development and mobile app security. Through "shifting left," security tools and processes are incorporated into each phase of the SDLC, ensuring a more proactive and comprehensive approach to security.

The DevSecOps lifecycle has five primary phases

1. **Plan:** In this initial phase, feedback from stakeholders is gathered to formulate functional requirements. Security practices involve threat modeling, as described above, anticipating potential vulnerabilities, and mapping data flow.

2. **Build:** Developers write code and integrate third-party dependencies. Security practices include implementing a secure design, adopting coding best practices, and ensuring the protection mechanisms identified in the planning phase are integrated into the code. (See "Hardening Your App" below.)

3. **Test:** Automated testing, including security tests, is conducted to identify vulnerabilities and defects. The most common techniques: Static Application Security Testing (SAST), Dynamic Application Security Testing (DAST), and Interactive Application Security Testing (IAST). These techniques are employed to scan for vulnerabilities in the source code, at runtime, and both, respectively. (See "Thorough Testing" below.)

4. **Release:** Verification before release ensures the bugs and coding errors uncovered during testing have been addressed. Priorities include confirming the proper implementation of protection mechanisms and addressing regulatory requirements.

5. **Operate and monitor:** Verification before release ensures the bugs and coding errors uncovered during testing have been addressed. Priorities include confirming the proper implementation of protection mechanisms and addressing regulatory requirements.

As you can see, DevSecOps is a way of weaving your overall mobile app security plan into the actual development process. By integrating security at every phase of the SDLC, organizations can reduce bottlenecks, optimize security, and deliver secure apps faster.

4. Hardening your app

Alongside the "build" stage of DevSecOps, security is all about hardening your app against potential attacks. Hardening involves implementing security techniques to make an app resilient against reverse engineering and other client-side attacks. Techniques like obfuscation, encryption, layered security, and RASP are vital components of app hardening.

5. Thorough testing

A robust security plan demands rigorous mobile app security testing (MAST) – not just at the end but throughout the development process. Two primary types of testing are necessary. Neither is sufficient on its own, and each serves a specific purpose.

Pentesting and automated security testing

1. **Pentesting:** Penetration testing, or pentesting, involves internal or external security assessments conducted by ethical hackers to evaluate the app's vulnerability. The quality of a pentest depends on the time invested and the expertise of the security professional, and it can be costly and time-consuming.

2. **Automated security testing:** On the other hand, automated security testing using MAST tools offers solutions that empower developers to take ownership of the security testing process. These tools provide resources for quick vulnerability detection, code hardening, and risk mitigation through static and dynamic analysis.

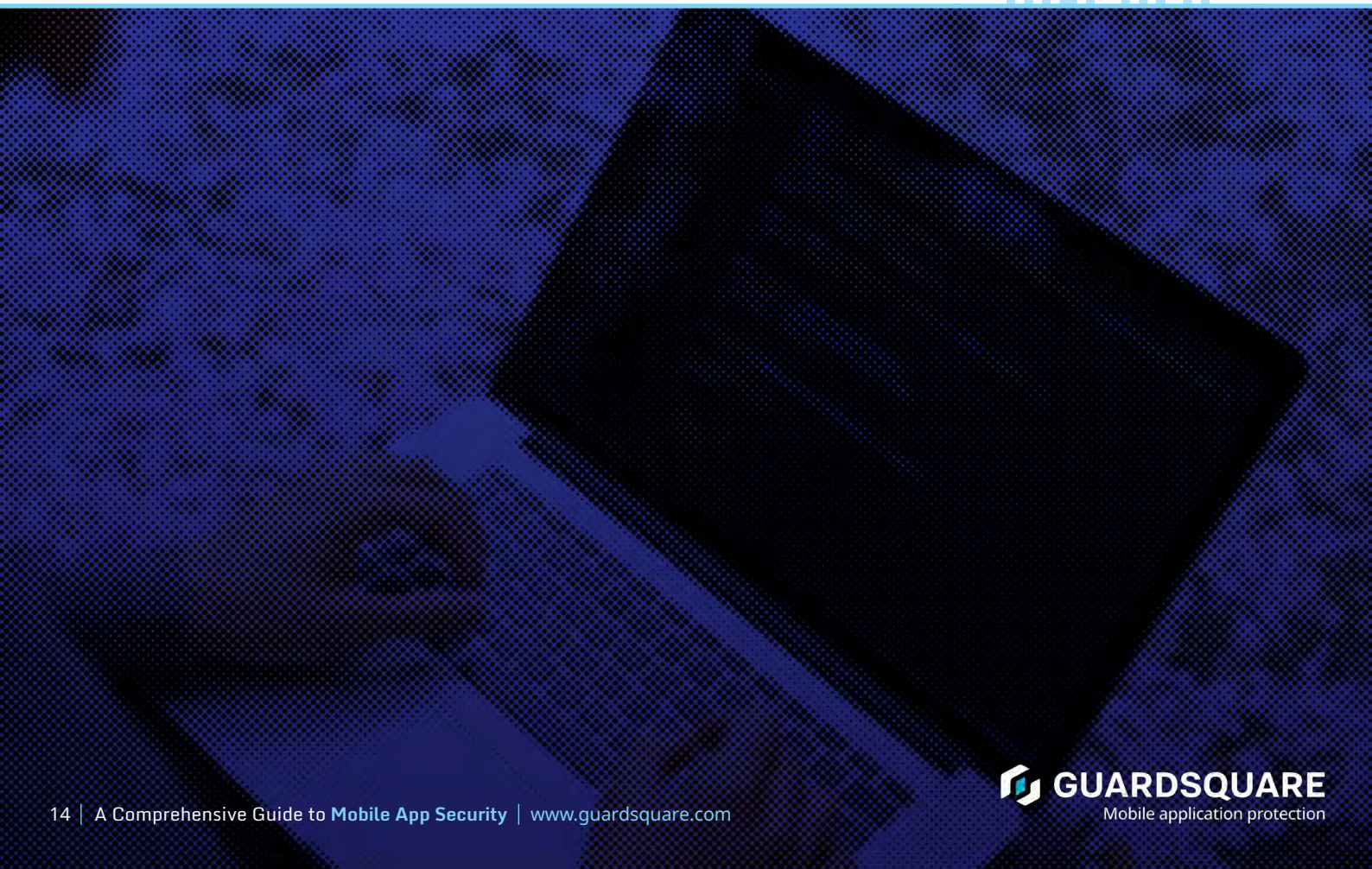
3. **Better together:** Combining pentesting with MAST tools is the most effective and efficient way to enhance mobile app security. While pentesting brings security knowledge and expertise into the process, it can be resource-intensive. Pairing it with an automated MAST tool allows for continuous real-time vulnerability identification — enabling developers to correct vulnerabilities earlier and make their security program more cost-effective and manageable.

6. Vigilant threat monitoring

Once your app is out in the wild, your job as the app developer/publisher is not done. Even after release, mobile apps require surveillance. Threat monitoring is the name for this crucial aspect of security. In the monitoring phase, the ideal tools collect threat data from various sources, such as RASP checks, and translate it into actionable insights that help you better understand the threats facing your app and release a more secure app in subsequent builds. Mobile app security is evolving alongside the threat landscape, so it's essential to stay informed about changes in industry security standards and regulations and regularly test and monitor your apps.

Complete mobile app protection

A complete mobile app security plan combines hardening, testing, and monitoring. Mobile app security is not just a checklist; it's a dynamic, ever-evolving playbook that ensures your app is equipped to handle the unpredictable evolution of vulnerabilities, risks, and threats.



How to select a mobile app security solution

Implementing the six steps covered above would be nearly impossible without the assistance of mobile app security tools. Ongoing IT and security professional shortages, lack of specialized knowledge on teams, and time constraints keep most teams from manually implementing the necessary protections.

Let's explore the options for applying technology to the challenge of mobile app security.

Manual vs. Automated

Manually implementing security protections is insufficient to secure your mobile app adequately. Manual efforts are also time-consuming and challenging to scale. On the other hand, automated protections in mobile app security tools can save you both time and frustration while providing stronger security for your application. For example, a RASP solution can dynamically inject thousands of checks into your app and change their locations in each build, making it significantly harder for threat actors to execute their attacks successfully.

There's no question that a robust mobile app security strategy must rely on automation via security tools.

Build vs. Buy

While some organizations argue that building their custom security solutions is less expensive and equally effective, the costs of maintenance and updating tools are significant. Plus, staying ahead of changes in the threat landscape adds up fast. Building your own solution almost always becomes the more expensive (and less effective) choice.

Finally, buying a solution backed by industry expertise and dedicated security research teams helps stay ahead of the evolving threat landscape and provides crucial support in implementing and maintaining security products. That's one less workstream on your team's plate.

Achieving comprehensive mobile app security with Guardsquare

Guardsquare offers cutting-edge solutions for comprehensive mobile app security. Here are the three major categories of products we offer and how they map to the steps above.

DexGuard/iXGuard

- **What it is:** Powerful code hardening and application protection with multi-layer and polymorphic features
- **How it works:** Protects Android and iOS applications and SDKs against reverse engineering, tampering, and other attacks

AppSweep

- **What it is:** Automated mobile app security testing product that maps to OWASP standards
- **How it works:** Scans your app's code and libraries regularly during development, providing actionable recommendations to enhance your app's security posture

ThreatCast

- **What it is:** A real-time monitoring solution that provides actionable security feedback
- **How it works:** Collects threat data from sources that include RASP checks and accessibility services and translates it into insights to identify which users and devices are targeting your app, how they are attempting to attack your app, and recommended security changes for future releases

Guardsquare's team of developers and security experts is committed to ensuring the security of the mobile landscape. With a decade of experience building and optimizing a set of mobile app security products tailored for diverse industries, from financial services and e-commerce to gaming and media, Guardsquare is here to make your mobile app security journey far less complex and daunting.

Achieving mobile app resilience

A mobile app represents a substantial investment of time and money, often serving as a vital component of an organization's overall business strategy and a significant revenue source.

For these reasons, mobile app security issues can profoundly impact public and customer perception. They can also lead to revenue loss, regulatory consequences, and even loss of licensure in some industries. Without overstating the risks, these valuable assets must be well-protected.

Proactive, security-first apps are more resilient to attacks and less likely to cause production delays at the end of the development lifecycle. With security standard guidelines in place alongside an analysis of your app's unique risk profile, organizations can follow the 6-step process outlined in this eBook to ensure their mobile apps are well protected against current and future threats.

Remember to prioritize automated mobile app security solutions to address the skill gaps in your team, eliminate the frustration of maintaining a custom security solution, and ensure your organization can keep up with the rapidly shifting threat landscape. With these components in your corner, you will be well-equipped to keep your mobile apps secure. ■

GUARDSQUARE
Mobile application protection

PROTECT TEST MONITOR

Ready to take the next step in your
mobile app security journey?

Request a demo

Guardsquare offers the most complete approach to mobile application security on the market. Guardsquare's software integrates seamlessly across the development cycle: from app security testing to code hardening to real-time visibility into the threat landscape, Guardsquare solutions provide enhanced mobile application security from early in the development process through publication.

More than 900 customers worldwide across all major industries rely on Guardsquare to help them identify security risks and protect their mobile applications against reverse engineering and tampering.