Security First

HOW MOBILE APP DEVELOPERS CAN PRIORITIZE SECURITY WHILE RETAINING SPEED TO MARKET

Kobiton

550 Pharr Road NE, Suite 525, Atlanta, GA 30305

Email: info@kobiton.com

Phone: (678) 235 - 4095 | Fax: (770) 359 - 0051

INTRODUCTION

The mobile app marketplace remains highly competitive. Consumer app makers want engaged users, and enterprise app developers are trying to squeeze efficiencies out of inefficient legacy processes. In all cases, mobile app developers feel a growing sense of urgency to put out apps and updates as quickly as possible. But in this haste, do mobile app developers mistakenly believe that only web app security issues result in high profile hacks like Equifax or HBO?

Most mobile app developers assume they've had no security issues to date, so their process must be working. They assume their code is good because they've hired great developers. They assume they've performed at least adequate QA and testing, and everything seems fine. And they assume they can always patch and update later. Their thinking seems sound, but the facts indicate those assumptions don't hold up.

According to the Ponemon Institute, “only 29 percent of mobile applications [...] are tested for vulnerabilities” and “26 percent [of respondents] indicated that their company failed to conduct testing at all.” Not surprisingly, NowSecure “identified a security issue in 26 percent of Android apps tested.” For iOS apps, the same report noted issues ranging from cookies without a secure flag (30 percent of all iOS apps) to unencrypted sensitive data in transit (19 percent of all iOS apps).

Real world examples include

- A coding error by Twilio developers across at least 685 apps led to 180 million phones exposing personal calling and text messaging data. Twilio's stock price dropped 7 percent after the data breach was publicized.

- Hackers stole 34,000 health records from Quest Diagnostics through unauthorized access to the company's MyQuest app.

- Wishbone, a mobile app popular with teenagers, experienced a data breach of 9.4 million records from an API security flaw.

- Iranian hackers exposed 15 million phone numbers from users of an instant messaging service called Telegram through a SMS verification flaw in its mobile app.

- In June 2017, more than 1,300 Google Play store apps were found to contain malware that only activated after a user downloaded the app. Despite built-in security scanning, these apps slipped through Google's quality control. And the malware problems continue - showing that mobile app developers cannot place the responsibility of mobile app security scanning and testing on others.

Obviously, mobile app developers are conscientious and care about security. The issue is time. The steady adoption of DevOps demonstrates the increasing demand to speed up time to market and improve mobile app quality while also securing any critical data handled by the app. To create the best of all worlds, automation and monitoring—including security testing—form an essential part of the continuous DevOps processes that ensure mobile apps are quickly deployed without putting the business at risk.

Trends show that gaming, banking, insurance, and social media mobile apps are highly sought after targets because of the valued information that users share. But as we see with recent breaches, it is clear that any mobile app is a target.

Mobile App Security Issues Rooted in Bad Habits

Despite DevOps adoption, security issues still plague the mobile app development culture, which can be broken down into five problem areas.

1 | No design for data security

When developers design and build, UI and UX get all the attention in order to build a “beautiful” product. But when developers prioritize UI and UX over security, they may create an elegant piece of technology that gives bad guys your users’ private information. When designing an app steer clear of these common mistakes.

- **Not examining code from other developers**

  Open source is easy, cheap, and quick to reuse, but that code can hold security risks and vulnerabilities.

- **Using old libraries**

  Developers who lag or use favorite old libraries risk introducing security vulnerabilities eliminated in more current libraries.

- **Encryption**

  From both a business perspective (protecting intellectual property) and a consumer perspective (preventing security vulnerabilities), source code encryption is important. An easily reverse-engineered app will be ripe for exploitation on the black market.

2 | Failure to protect information in transit

Even if developers build a secure app with sound source code, that app also needs to securely interact with databases, servers, and other services as it exchanges information. Many apps do not properly authenticate or authorize the transmission of user data and cannot detect problems with a malicious user's security posture. Security holes can be left open through problems with server-side controls, cryptography issues, and backdoors (such as through an unsecured API) that give hackers a way inside.

3 | Issues with local data storage

Mobile data breaches are more likely to occur if information is stored locally on a device. Devices may not have adequate antivirus protection, users may fail to update and patch software, and device theft is common. Those conditions expose sensitive information stored directly to the device or cached data that's not cleaned up regularly.

Even if your app doesn't store sensitive data on the device, many apps may unnecessarily ask users for permission to other apps and services on their device like contacts, files, and even the camera or microphone. Those permissions can be leveraged to gain unauthorized access. For example, researchers recently discovered that Uber's app could copy a user's phone screen. These problems are exacerbated with BYOD trends in the workplace that increase the risk of not only compromised user data but also stolen company data.

4 | Lack of thorough testing of mobile apps

Testing not only improves usability but it also highlights security flaws. Without testing, the risk of source code bugs and security

vulnerabilities increase. In addition, a lack of dynamic testing means that high-risk security vulnerabilities related to data input, data transit, and authorized communication with other information sources may be more likely to inhabit the app after it ships. It doesn't take much effort to test, but developers need to go beyond extremely limited testing such as each person on their team only loading the app on their personal device. More rigorous testing is needed to identify previously unidentified issues.

5 | Lack of regular, effective updates and patches

High-risk vulnerabilities often exist for more than six months before they're fixed. That means many developers fail to not only build security into their app but also fail to do a good job of analyzing how their apps are working in the marketplace after they're released. Whether it's not having enough people to fix vulnerabilities, not receiving user feedback effectively, or lacking awareness about security flaws, updates and patches often aren't released regularly enough or they fail to fix the highest-risk flaws until it's too late. As a painful example, the press quickly reported that the flaw that caused the Equifax data breach was published in March but not patched until July. Not surprisingly, the hack is believed to have occurred during that window of susceptibility.

Lack of Mobile App Security Leads to Real Business Impact

Beyond the bad habits that lead to security breaches, two primary repercussions should sober your company's leaders into making the needed investments.

1 | Liability

WhiteHat Security reported in July 2017 that "High risk vulnerabilities took an average of 196 days to fix, up from an average of 171 in 2015." And if you release an app with a high-risk vulnerability, it's costly to fix and exposes you to legal trouble.

For example, data breach notification laws are complicated, costly, and consumer-driven—not business-friendly. If your mobile app becomes the source of a data breach, then your organization will need to report that breach to 48 states, the District of Columbia, Puerto Rico, and the U.S. Virgin Islands. Depending on your industry, you may need to send reports in response to financial services, healthcare, communications, or other industry regulations.

After following the law, additional lawsuits, fines, and penalties may be imposed. Companies are also ill-prepared for the consequences of the EU's General Data Protection Regulations (GDPR)—a set of stringent information privacy regulations to take effect in May 2018. If you violate the privacy of users in the EU through a data breach, your organization may be fined up to 4% of its annual global revenue. And in the wake of the Equifax data breach, the United States appears poised to get stricter—not looser—with its own data breach security and privacy laws.

2 | Brand and revenue impact

Part of growing and sustaining a business is through customer trust. Losing that trust can send your business into a downward spiral from customer loss, lowered shareholder value, and even bankruptcy. True, big companies like Equifax, Target, or Home Depot may survive a major data breach—although they will still lose millions of dollars. But smaller companies are not so lucky. A major breach—which can stem from a simple security vulnerability in a mobile app—can be devastating.

Cloud-Based Services to Start Using Today for Greater App Security

To meet mobile app security best practices while still decreasing your mobile app's time to market, two options can help you automate essential security tasks and processes.

1 Use a Dedicated Mobile Vulnerability Testing Tool

Tools now exist that automate essential security tests so that your team can better focus on mobile app development and deployment. For example, a tool such as App-Ray can do the following for your team

- **Reverse engineer your app** to look at security issues related to the permissions, components, and structure.
- **Perform a static scan (SAST)** by looking at bytecode and structure.
- **Perform a dynamic scan (DAST)** by using the app in a test environment.

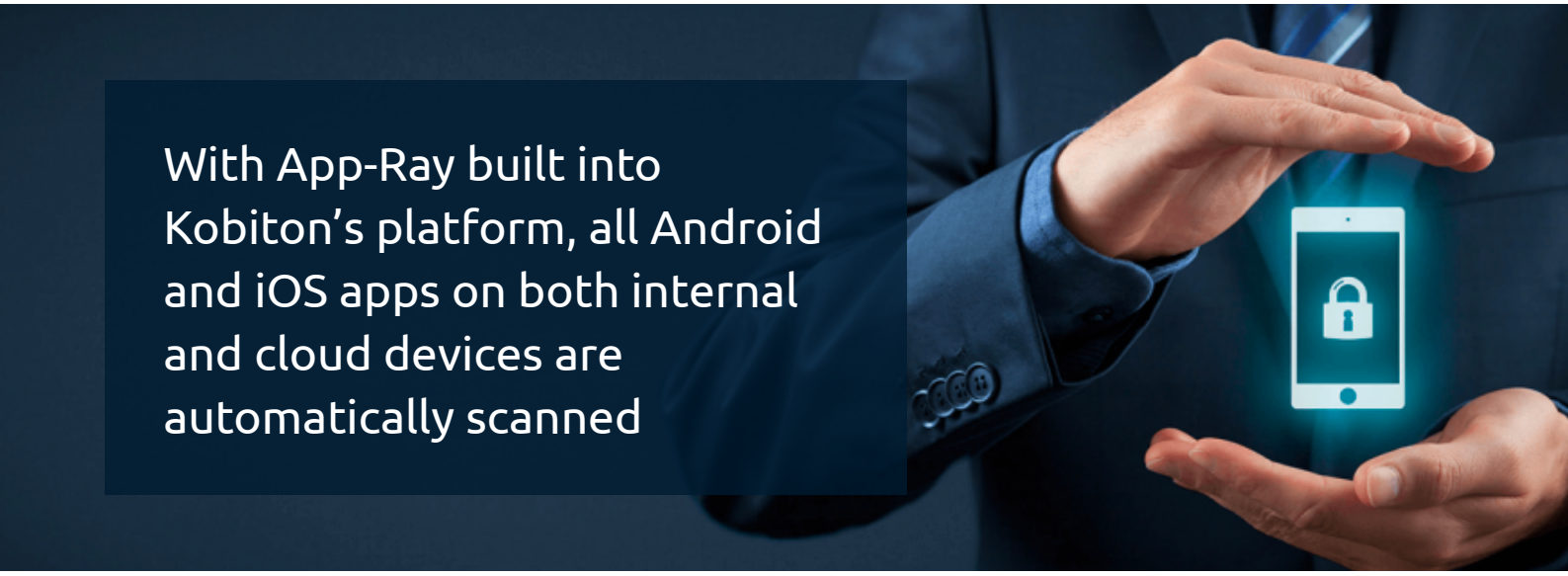
- **Perform a dynamic scan (DAST)** by using the app in a test environment.
- **Perform an interactive scan (IAST)** (if needed) that allows users to manually test certain actions.
- **Produce a security report** and view the results on a dashboard.
- **Provide the raw data** in JSON files for download.

The benefits of such a tool to DevOps teams are numerous

- **Fast, automated scanning** that doesn't slow down your testing processes.
- **Comprehensive tests** that cover a wide range of important security issues.
- **Reduced costs** through the tool's affordability and the elimination of many, many testing hours during your mobile app development lifecycle.
- **Root cause identification and analysis** of security issues.
- **Suggested solutions to security issues** (instead of just noting problems).

Better yet, you can also consider a solution with this tool built into your mobile app testing platform.

2 Use a Cloud Mobile App Testing Platform with Built-In Automated Security Scanning



When multiple users constantly upload mobile apps to test, all it takes is one malicious app to infect an entire device farm. Mobile app developers need to rely on a secure cloud-based device lab that includes a wide range of unobtrusive security testing such as automated security screening for every app and automatically identifying and detecting vulnerabilities

For example, with App-Ray built into Kobiton's platform, all Android and iOS apps on both internal and cloud devices are automatically scanned. In addition to automated security scanning, other important Kobiton security features include:

- Automatically uninstalling any installed apps on a test device when you end a test session.
- Prohibiting the testing of any apps used for hacking or malware distribution.
- Not allowing anyone except for authorized users to access your apps during or immediately after a test session.

Using such a mobile app testing platform can help meet the realities of your business, marketplace pressures, and DevOps objectives by:

- Keeping release cycles short.
- Lowering security costs through automation.
- Reducing liability and risk of security incidents before you ship.

By assessing your current security culture, implementing more mobile app security best practices, and using a cloud platform like Kobiton that automates important security testing, your DevOps team can easily improve its security policies, procedures, and best practices to make sure you're not the next mobile app in the news for all the wrong reasons.

About Kobiton

Kobiton is a powerful mobile device cloud that allows companies to more effectively manage the devices they own and access the real devices they want. Simple to use, easy to access from anywhere and flexible enough to scale capacity across internal or external devices, Kobiton minimizes costs while increasing productivity, helping businesses get their apps to market sooner.

The mobile device cloud platform offers centralized testing history and insights to improve collaboration across teams; access to the most in-demand mobile devices to supplement existing inventory and significant cost savings. Visit kobiton.com to learn more.