PROACTIVE MOBILE APPSEC: A 2020 GUIDE

datatheorem

Prevent AppSec Data Breaches

Table of Contents

In this white paper, we will discuss the various challenges teams will encounter when trying to start or run an application security program so that budgets, teams, and the software releases are optimized. According to the Verizon data breach report, 56% of data breaches took "months or longer" to discover. With this knowledge, is it possible to create proactive mobile appsec programs that continuously discover potential issues, without extra staff to manage them, and won't slow down software development?

Challenges with Mobile Application Security

The first challenge in working with a security product, vendor or program is to agree on how to prioritize solving the most important issues first. But security and development teams use different vernacular when assessing the risk level of security issues - whether they use a scale, story points, or list of exploits. Establishing a shared standard is the first challenge. To prioritize the highest vulnerabilities, we recommend a non-subjective approach in very clear terms of fixing or non-fixing. For first priority items, we refer to these as P1 vulnerabilities. P1 vulnerabilities are any issue that allows a remote attacker to export data from an app. The key to getting development and security teams to agree on a P1 vulnerability is to anticipate how an attack might come from a remote attacker and what data could be compromised. Then teams would agree to check those P1 vulnerabilities everyday, with every release.

Challenge number two is working with the app store itself. We have a two-party system - Apple and Google. In the United States, they control what is on Google Play and in the App Store and that affects development as well as security. They also have low and inconsistent security requirements that have to be met before they will allow the release of apps or updates. If requirements are not met, they can legally remove apps from the app store and potentially halt your business.

The third challenge is one of the biggest - SDK and open source software. In modern mobile applications, it is common practice to use a sandbox. So when you download an app on your iOS device or your Android device, it is isolated from the other apps as well as its data, memory and all other components. For SDKs and libraries, these are unvetted business partners that have full control over your app, its data, its permissions, its network layer, everything. Your code in the third-party code sits next to each other in the app. That third-party code can do anything your code can do. So those SDKs within the apps must be vetted before they start causing problems or you run the risk that some nefarious character will get 10, 100 or a million developers to use the SDK for free, then they will monetize the data collected. On average, each app will have around 12 to 18 SDKs so the probability of having this data hacked is fairly high. Security will need to monitor any changes regularly.

Challenge number four is completely non-technical, considered the least interesting, and the one most often ignored:

> How do we keep our company name out of the data breach headlines?

Avoiding breaches is the main reason that security leaders spend money on an information security program or the staff to support it. When a data breach is publicized, it hurts the company as a whole. Customers' brands are damaged, jobs are lost, and the stock prices drop. It is also hurtful to employees' reputations after they leave a company, even if their role had zero influence on security.

It is important to stay focused on the first three challenges, but this fourth challenge is what you need to stay ahead of. These four challenges could be considered unit tests that should be considered on every single release that extends to the enterprise as well. The first enterprise challenge is regulatory compliance. Companies need to make sure they are compliant on a regular basis. There are old and new mobile privacy laws - the FTC, GDPR, CCPA, and more privacy legislation has been proposed by other U.S. states. Privacy is nothing new to mobile, but now there are giant compliance issues that did not exist years ago. For example, is your app collecting geolocations? How are you storing it and who are you sharing it with? Let us assume you are using the data for a legitimate business purpose, but you are accidentally storing that data with an analytics SDK that is housed in Florida. Now as a European Union-based app, you are not compliant with GDPR. GDPR guidelines state that the geolocation or trackers cannot be sent to a third-party outside of the EU. So while geolocation data collection is very common and has been around for years, the use of it is now highly regulated.

Another risk in security is data-at-rest and data-in-transit. Some of the biggest attack surfaces for mobile apps are not necessarily data-at-rest, but data-in-transit. At some point your users may end up on a hostile network using your app. What can your app do to be fully secure on an untrusted network? The key thing is making sure things like TLS is not only working, but also enforced. There are obviously tools like SSL pinning. SSL pinning is a disaster on web apps, but SSL pinning is very achievable on mobile apps. One example of this is our open source project called TrustKit. This is a free tool that you can use at any time to enforce TLS and it has assisted in avoiding over 400 million attacks in the past two years.

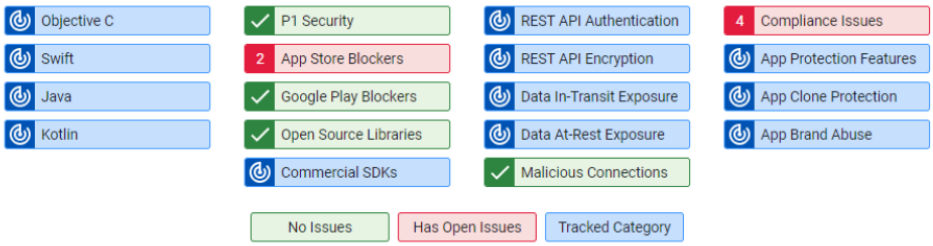
Limitations of Traditional Approaches

Next, we will explore two ways that organizations have managed mobile application security in the past and how these approaches have evolved or are failing. If you are reading this white paper, penetration testing (or pen testing) may already be a strategy of the past for you. But just in case it is not, let us review why this strategy should be abandoned. Pen testing was a reliable strategy to evaluate app security before major releases. But software development today requires updates 30 times a month, week, or day. This is why the development world has now moved on to DevSecOps strategies in which apps are scanned and tested automatically and updates are released non-stop.

A DevSecOps approach gives users the best-performing, most-secure app experience and companies remain compliant. The second way companies have managed mobile app security in the past is by measuring success from a security perspective instead of from a developer perspective. Referring back to unit tests, these are a level of software testing where individual units or components of software are tested. The purpose is to validate that each unit of the software performs as designed. If something fails, that means that the release should not proceed. Security is managed the opposite way. The more issues the better. More issues show progress in quality. The best approach that we recommend to our customers is to avoid a tool that alerts you to code quality issues or nice-to-have issues. What is most critical is to keep quality high by keeping alerts low. You will still have issues to attend to in security, but the most important issues get alerts and those are the issues where you would stop a build from going into production. The best solution for this is to automate so that the prioritization is done for you and you don't fatigue the development team with lower priority issues, allowing them to focus on overall quality and ease of use on the app.

Automate Security Checks with Daily Scans

- Automates Static and Dynamic Analysis
- Discovers Dynamic Run-time Security Flaws
- Alerts on newly discovered P1 issues and Store Blockers
- Provides Secure Code Samples and Recommendations
- Identifies Third-party SDKs Vulnerabilities
- Inspects Open Source Libraries for Insecure Code
- Reports on compliance of PCI, GDPR, HIPAA, FTC, and more
- Protect against SSL/TLS Man-in-the-Middle Attacks
- Protect against 3rd party keyboard loggers
- Encrypt Data Storage for the App
- Help Remove Malware
- Provides up to 25 additional security measures

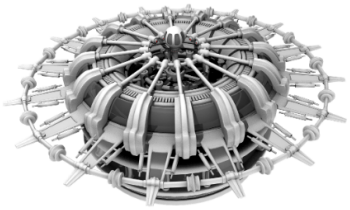


Automation for mobile software development lifecycle (SDLC)

Show in the image above are lists of tests that are ideal for your application security program. This is not exclusive to Data Theorem; most mobile application security companies are scanning for similar vulnerabilities. At a bare minimum, static and dynamic analysis is essential. The key indicators are at the bottom. Ideally, you would have code scanning (which is on the left), a unit test (in the middle), and then you have compliance status, API discovery, and malicious scans (on the right). Overall, the goal for a mobile app is to check its status and see green check marks across the board. These results are what developers are used to reading and security can understand and prioritize. When looking at compliance, you can see we found four alerts in this scan and each would warrant a different approach to resolve them.

Depending on what is important for your app and business, you might not want to block certain updates going into production. For example, if you're not taking credit cards, a PCI (Payment Card Industry) violation is not something you have to consider. Similarly, if you are not a healthcare company, HIPAA (Health Insurance Portability and Accountability Act) is not a guideline that is applicable to you. This framework is something you should have baked into your mobile application security program. You will aspire to have all green levels and remain compliant. But because the work has been automated to prioritize the right alerts, they know it is serious when they see red and will act to remediate.

Your solution to having superior and efficient mobile application security is to automate what machines can do, instead of asking your staff to take on things like unit tests. Use an automated analyzer engine that discovers and conducts continuous security assessments, identifies vulnerabilities, and delivers secure code to fix issues in a language that developers understand. Your staff is valuable and you will want to bring security to their level in order to make it a part of the CI/CD cycle, and to ensure that security scales the way your business does.



Use an automated analyzer engine that discovers and conducts continuous security assessments, identifies vulnerabilities, and delivers secure code to fix issues.

Customer Case Study: Large Global Investment Bank

To understand further how this all works, we would like to share an example of how one customer is leveraging automation to overcome and help scale their limited security team. The first challenge for this organization was to enable the ability to fix any application that had vulnerabilities and be able to continuously identify and address those at the rate in which their development teams were producing code. The second was to make it possible to identify blockers that did not allow them to publish a release in an app store and how they could remediate the blocker quickly. The third challenge was how to unite app teams that were organized across multiple product teams. This is a common obstacle for organizations that have many mobile applications, as opposed to just one or two. These product teams all use different tools, especially when it comes to their development cycles. Their CI/CD tools vary, as well as things like their bug repository.

The security team needed to work with a variety of different integration points and tools to be able to create their audit process and make sure that it is working within those development programs. Lastly, our customer was a financial institution so they had a certain level of compliance and government regulatory agencies, like the FTC to deal with. In the past, they worked with manual audits. As discussed, this is hard to scale, especially when you have to do internal scripting in order to work with different development teams. Eventually the manual processes become a bottleneck for the modern-day release cycle and create tension between security and development teams.

Our customer was faced with all of these challenges and we partnered with them to bring in an automated solution. We had to provide the ability to integrate into the multiple CI/CD pipelines, do a continuous evaluation of each of their different product and app teams, and build indicators that would keep up with their nightly build cycle. To address the compliance issues, we had to help them move away from manual reporting. During our on-boarding, the customer realized that they needed to start providing reporting for their customers to prove compliance and not just industry regulators. As a result, they had to begin scanning all of their apps in pre-production (before they ever make it to a production release). In their first 90 days, they were able to close out over 75 security issues, and identified 51 compliance and regulatory issues. Today, multiple product and security teams within the company use our tools to be proactive against data breaches instead of reacting to issues on a daily basis.

AppSec Program Results:

Percentage of Apps Scanned including Pre-Production

100%

Overall Security Issues Resolved and Closed

75

Regulatory Compliance Issues Identified and Resolved

51

P1 and App Store Blockers that did not make it into production

7

Summary

It is crucial for every company that the top-priority data vulnerabilities and app store blockers can be resolved in a way that is swift and accurate before data is exposed. The best way to avoid this is to provide a simple tool that developers can use to implement an automated strategy into every software release cycle.

Data Theorem offers a tool that security and operations teams can both leverage to find and resolve critical security vulnerabilities across their entire mobile application tech stack by performing continuous dynamic runtime analysis on each release.

Data Theorem is a leading provider of modern application security. Its core mission is to analyze and secure any modern application anytime, anywhere. The Data Theorem Analyzer Engine continuously scans APIs and mobile applications in search of security flaws and data privacy gaps. Data Theorem products help organizations build safer applications that maximize data security and brand protection. The company has detected more than 400 million application eavesdropping incidents and currently secures more than 4,000 modern applications for its Enterprise customers around the world.