Mobile App Security Standards | Security Checklist for Mobile Development

Vijay Khatri

2018 has been the year where security proved to be the Achilles heel of many giant companies. From Google G+ disasters to Facebook's congress hearing. It has been a year full of security scandals all around the world.

Security is always a concern when creating an application, but it's often overlooked when developing the application. And what's overlooked in the beginning becomes a dormant vulnerability later on that may threaten your business, but you might not be able to catch it then before something happens.

For that reason, it's often best to account for security from the very start and it's definitely not a time waste.

Mobile App Security Standards/Checklist

There are a few practices that you could follow when creating an application that will help you create more secure applications on the go. Here's the list that you could follow:

1. Securing the source code:

It doesn't make sense to make a powerful app, follow every best practice, and then leave the source code open to anyone. It's like building a high castle and leaving the front gate open. In a mobile application, usually, most of the source code resides on the client side, including the UI and the business logic which presents a threat if this code was accessible to attackers.

Obfuscation is the process of making your code base unclear and confusing, to prevent attackers from understanding or reverse-engineering it. It changes your class, method, attributes names into meaningless letters or characters, making the code un understandable. You can easily obfuscate your code base with Android's built-in Pro-guard and there are many other software that you could use, either for Android or IOS.

2. Securing the files and the database:

It's not enough to secure the code base, you also need to secure the data. You need to store data on the device for all sort of reasons, this data can include critical information such as user credentials or payment info, for that reason you should always make sure that the data you're storing on the user's end is encrypted to prevent its leakage.

3. Securing Communications

Network security in mobile development is not as trivial as it is for web development, and many companies and developers do not opt network security in their development process. It's not enough to secure the data on the generation and storage points only.

Your application's data should also be secured in transit, that means that sending and receiving data inside your application should be via secure mediums, with a VPN tunnel, SSL, TLS or HTTPS communication. This way, if anyone managed to eavesdrop on your network requests, they wouldn't be able to decipher the data out and security will be ensured, otherwise, attacks such as packet-sniffing and man-in-the-middle would be a serious threat to your application. You can find a full guide on mobile network security testing here.

4. Consider Data Portability

Data portability is the practice of using user data across different platforms and services. Like using your Facebook account to sign in other platforms like StackOverflow or GitHub. This allows you to leverage the security of the bigger companies and use it on your side, inside of implementing all the user's authentication and private data all from scratch, it also makes it easier for the user as more people find it plausible to use their old accounts than create new ones.

A popular protocol for that is OAUTH, you can check it out here.

The simple flow of OAuth allows you to access the protected resources a.k.a user data on the other end by just storing the access token, which saves you the hassle of collecting and protecting that data.

5. Brace for Reverse Engineering

This might be more specific to Android applications since Android is an open source platform, which means anyone can look up the source code, make modifications on the OS any way they want. For this reason, you'll need an understanding of the Java-based Android environment as well as of the Linux os kernel to understand the process and understand how you can protect your application against reverse engineering.

6. Perform Input Validation

Input validation is one of the most important practices of taking a user's input, yet it's often disregarded in the development process for the sake of "speed". Input validation allows you to check the data supplied by the user to prevent malformed data. Input validation is very common in most frameworks, both on the web and mobile development and you should make use of it.

7. Use Cryptography wisely

Encrypting your data, or hashing the passwords doesn't necessarily dictate that your application is secure. In fact, broken cryptography is the most common threat to mobile applications. You should avoid weak or broken algorithms and make sure that your program doesn't use them. These algorithms include MD5, MD4, SHA1, BLOWFISH, RC2, and RC4. Cryptography is a strong element of security in a mobile application, and hence, if used correctly it can protect your application and data.

8. Perform Penetration Testing

Penetration testing is one of the most important stages of securing an application as it can scan a wide range of vulnerabilities. It simulates what an attacker can do in various environments and modes of operation. A lot of people confusing regular software testing with penetration testing, but they are really different and serve different purposes, but you need to do both.

Why you should care?

Security is everyone's concern, it's true that most users wouldn't regard the permissions given by the application they're using, and they can't possibly tell if an application is secure or not. But should a leakage happen, it is going to be your responsibility as a developer. There are heavy implications for companies who are not GDPR compliant.

We've all been over flooded with the GDPR emails following the Facebook Congress Hearing, and quite soon, it will be a standard worldwide. It's true that sometimes, it's out of a company or a developer's hand but you always have to be cautious nonetheless.

OWASP (short of Open Web Application Security Project) offers technical guides, checklists, tools and projects for you to use. They also offer a free online project covering many aspects we discussed in this article and even more, it's definitely worth checking out. You can also find the top-rated security courses and resources on Hackr.io, voted by the community if you wanted to take a deeper dive in security whether you want to be a security engineer, or if you want to incorporate security best practices in your development process.

People Also Check:

- **Open Source Security Testing Tools**

Vijay Khatri

Vijay is enthusiastic, passionate and curious, He is always looking for new solutions by continuous learning. Currently, he is working as a Digital Marketing Specialist. View all posts by Vijay Khatri