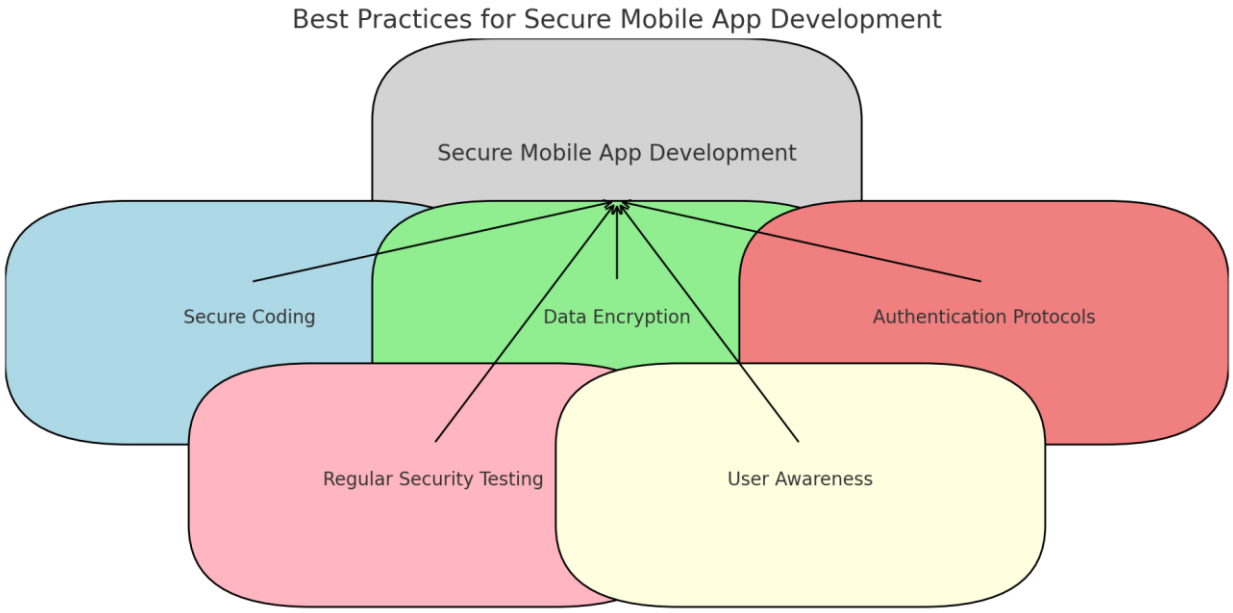
International Journal of Enhanced Research in Science, Technology & Engineering
ISSN: 2319-7463, Vol. 10 Issue 2, February-2021, Impact Factor: 7.957

Addressing Security Challenges in Mobile Application Development

Nakul Pandey

ABSTRACT

The rise of mobile applications across various sectors has heightened the necessity for robust security measures. This paper explores the critical security challenges faced by mobile app developers, including data breaches, insecure data storage, and inadequate authentication mechanisms. It offers best practices for secure coding, effective encryption methods, and implementing strong authentication protocols. Through case studies of recent security incidents, the paper illustrates the severe repercussions of poor security practices. The findings underscore the need for a comprehensive security approach throughout the mobile app development process to safeguard user data and uphold user trust.



Keywords: Mobile App Security, Secure Coding, Data Encryption, Authentication Protocols, Vulnerabilities, Data Breaches, Security Best Practices, Mobile App Development Lifecycle.

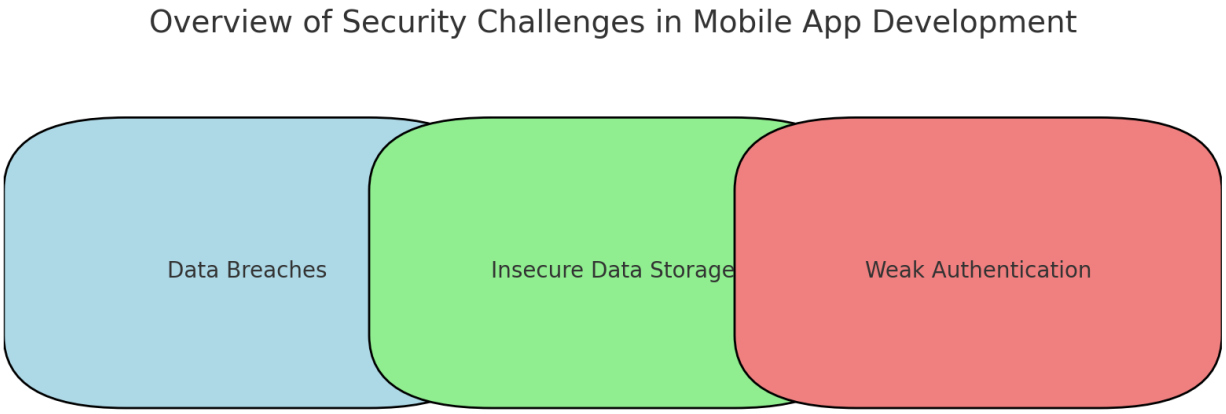
INTRODUCTION

Smartphones and tablets are powerful and popular. More than thousand new mobile apps hitting the market every day. In this fast-moving technological era, is security keeping up?

Apps and mobile devices often rely on consumer data — including contact information, photos, and location to name a few — and can be vulnerable to digital snoops, data breaches, and real-world thieves.

The recent security reports from around the world has expressed their serious concern about mobile device apps. According to malware analyst, while there have been no major targeted attacks on mobile devices – as has been the case with desktop platforms in recent years – it is clear that cybercriminals are focusing their attention on smartphones. It's also clear that the cybercriminals are using social networks to get an 'in' on to users' smartphones. Google, has done a lot to make Android more secure than earlier versions of the OS, but there is much more to be done. Users should also consider using encryption

for their data, and only store the data that they really need to access on the smartphone or tablet itself. In this paper we are discussing some tips and measures which can improve the security of mobile apps development.



2. Challenges of Secure Mobiles Apps Development

Developing a secure mobile application for smartphones or tablets is like running through a technology minefield. If developers are not careful, something is likely to blow up. The devicesare inherently vulnerable to hacking, theft or loss and can serve as on-ramps into enterprise and/or home networks. Wireless network connections may not be encrypted properly. The use of non-standard API libraries, patches and shared code found on developer forums, GitHub, Stack Overflow or from third-party companies may introduce vulnerabilities and privacy issues into your app. Cross-platform development tools can themselves introduce security problems[1][2].

Given this backdrop, the stakes have never been higher for developers to know how to build security into their applications from the very start. There is no checklist for securing all apps[4]. Different apps have different security needs. For example, an alarm clock app that collects little or no data will likely raise fewer security considerations than a location- based social network.

Apps that are more complex may rely on remote servers for storing and manipulating users' data, meaning that developers must be familiar with securing software, securing transmissions of data, and securing servers. Some of the issues which should be critically observed and handled for mobile apps development are as follows:-

i. Secure Data Storage

Insecure data storage is one risk area in mobile security. When sensitive data stored locally or on the cloud are left unprotected, i.e. unencrypted, short- term cached, or with weak permission, confidentiality of data is lost. For some organizations, this also means non-compliance and privacy violations. Preventive measures of insecure data storage include refraining from using public storage areas, leveraging file encryption APIs provided by secure containers and platforms, and not granting files world readable or writeable permissions.

ii. Secure Session Handling

To achieve convenience and usability, mobile app sessions are generally maintained longer via HTTP cookies, OAuth tokens and SSO authentication services. When a mobile app uses the device identifier as the session token, it could expose to risks such as unauthorized access, privilege escalation, and circumvent licensing and payments. Recommended practices include ensuring tokens can be revoked quickly in an event of a lost or stolen device, and utilizing high entropy and tested token generation resources.

iii. Side Channel Data Leakage

Side channel data leakage is usually caused by not properly disabling platform features as well as programmatic flaws. Sensitive data, as a result, ends up in unintended places such as web caches, keystroke logging, screenshots (e.g. iOS back grounding), system logs and temp directories. In order to minimize this risk, mobile apps developers are advised adopt good practices such as 1) never log credentials, PII, or other sensitive data to system logs, 2) remove sensitive data before screenshots are taken,3) disable keystroke logging per field, and utilize anti-caching directives for web content, 4) debug apps before releasing them to observe files created, 5) review third party libraries introduced and the data they consume, and 6) test applications across as many platform versions as possible.

iv. Security Testing for Mobile Apps
In order to assist mobile application developers and security testers to detect and mitigate risks associated with mobile apps, OWASP has developed a ―Mobile Application Testing Guide‖. The Guide covers Static Analysis, which involves analyzing raw mobile source code, as well as Dynamic Analysis, which involves executing an application either on the device itself or within a simulator/emulator and interacting with the remote services with which the application communicates.

• **Static Analysis: -** The primary goal of a Static Analysis is to identify programmatic examples of security flaws by analyzing the source code. OWASP recommends having access to either a development or production instance of web services, so to include both source code and a working test environment to perform the assessment within in order to expedite understanding of the code.

• **Dynamic Analysis: -** Dynamic Analysis includes assessing the mobile application's local inter-process communication surface, performing forensic analysis of the local file system, and assessing remote service dependencies. Dynamic Analysis is conducted against the backend services and APIs. The type of tests varies depending on mobile application type.

v. Understand differences between mobile platforms
Research the mobile platforms you work with. Each mobile operating system uses different application programming interface (APIs), provides you with different security-related features, and handles permissions its own way. Don't expect that one platform works exactly like another. Do your research and adapt your code accordingly.

vi. Don't rely on a platform alone to protect your users
- Mobile platforms often provide helpful security features. But it's your job to understand those features (and their limitations), implement them properly, and take other measures necessary to protect your users. In addition, while platform-based permissions might be helpful in conveying security information to your customers, they're no substitute for your own effective communication. Talk to your users in your own words.

vii. Generate credentials securely
If you create credentials for your users (like usernames and passwords), create them securely. For example, a short number string might be an appropriate token for authenticating a user on a game score board, but the same credential wouldn't be appropriate for a social networking app.

viii. Use transit encryption for usernames, passwords, and other important data: -
Anytime your app transmits usernames, passwords, API keys, or other types of important data, use transit encryption. Mobile devices commonly rely on unsecure Wi-Fi access points at coffee shops, airports, and the like — and it's easy for troublemakers to snoop and intercept connections. To protect users, developers often deploy SSL/TLS in the form of HTTPS.

Consider using HTTPS or another industry-standard method. There's no need to reinvent the wheel. If you use HTTPS, use a digital certificate and ensure your app checks it properly. A no-frills digital certificate from a reputable vendor is inexpensive and helps your customers ensure they're communicating with your servers, and not someone else's. But standards change, so keep an eye on current technologies, and make sure you're using the latest and greatest security features. ix. Use due diligence on libraries and other third- party code Before using someone else's code to build or augment your app, do your research. Does this library or SDK have known security vulnerabilities? Has it been tested in real-world settings?

Have other developers reported problems? Third-party libraries can save time, but make sure you stay accountable for your app.

x. Protect your servers
If you maintain a server that communicates with your app, take appropriate security measures to protect it. If you rely on a commercial cloud provider, understand the divisions of responsibility for securing and updating software on the server. While some commercial services will monitor and update your servers' security, others leave you in control. Server
security is its own complex topic, so do some research. Take steps to protect yourself from common vulnerabilities, including injection attacks, cross-site scripting, and other threat

xi. Stay aware and communicate with users

Even after you ship your app, stay involved. New vulnerabilities arise daily, and even the most reputable software libraries require security updates. Follow general and library-specific mailing lists and have a plan for shipping security updates if needed. Check your inbox, too.

User feedback can help you spot and fix security vulnerabilities. When they discover vulnerabilities, researchers often try to resolve the issue with developers before publishing their findings. It's best to be part of that discussion early on.

If you're dealing with financial data, health data, or kids' data, make sure you understand applicable standards and regulations: - If your app deals with kids' data, health data, or financial data, ensure you're complying with relevant rules and regulations, which are more complex.

As the above mentioned points are important for secure apps development there is also need of proper approach for secure apps development. In the next section we are discussing those stages which can be considered as standard approach for a secure apps development.

3. Stages of Secure Mobile Apps Development

i. Preliminary Analysis

Most app security flaws can be prevented by seamlessly integrating security processes right from the earliest stages of app development. Planning out your initial app design strategy, keeping security in mind all the time, will far reduce the chances of security risks cropping up during the later stages of app development. Incorporating the right security measures earlier on, hence, saves you much time, money and effort, which you may have to invest later. In case you are designing an app for a particular company, you additionally need to take into account several other aspects such as the company's privacy policy, the industry policy (as and when applicable), regulatory requirements, confidentiality and so on[4][5].

ii. App Design Stage

The next step, the app design stage, can give rise to multiple security issues as well. Of course, these issues can also be dealt with relatively easily, when they are caught early enough. The actual problem, though, arises during the implementation of the app design. Security issues arising during this phase are the ones that are the most difficult to spot and resolve. The best way to minimize the risk factor here would be to create a list of all the potential traps, well in advance, also planning your course of action to avoid each of them. This is followed by performing a detailed security design review, which is usually handled by a security expert, authorized to carry out this particular check.

iii. App Development Stage

It is vital to ensure maximum possible app security during this particular phase. Of course, you have readymade, automated tools, to help you fish out issues within the source code. The major issue cropping up at this time would be finding and fixing bugs and tracking other security vulnerabilities. While these tools are effective to tackle common security issues, they maysometimes not be able to detect more complicated issues. This is where a peer review can come of use to you. You could ask a fellow developer to review your code and provide feedback on your app. Approaching a third party helps, as they may be able to find and fix some flaws which you left out during any of the above stages.

iv. App Testing and Deployment

Next, you need to test your app thoroughly, to ensure that it is completely free of security and other issues. Neatly document all processes and build security test cases, prior to testing the app. A professional test team uses these test cases to create a systematic analysis of your app. The last stage involves deployment of the app, wherein it is finally installed, configured and made available for users.

CONCLUSION

While it has never been overtly stated that app developers should have the necessary training in maintaining app security, it is only fair that developers achieve a basic level of knowledge in the field of mobile app security. Developers who are part of companies should receive mandatory security training, so that they can understand and follow the best practices for developing quality apps. In general, app developers should ideally have a grasp on the basic terminology, security processes and the knowledge of implementing appropriate strategies to effectively tackle issues relating to app security. The bottom line is that you cannot trust that all eyes have vetted all security issues in the open source software your application is built on. Both applications and their components need to be tested by the developers for security flaws. Open source software design choices may have been made that are not secure enough for your environment, and the security tradeoffs of these

choices may not even be mentioned in the project's documentation. A huge benefit of open source software is that you can review the security of the code itself; take advantage of that opportunity to ensure you understand the risks that are present in your code.

REFERENCES

[1]. Android Security Overview, http://source.android.com/devices/tech/security/
[2]. Anthony I. Wasserman, Software engineering issues for mobile application development. In Proceedings of the FSE/SDP workshop on Future of software engineering research (FoSER '10). ACM, New York, NY, USA, 397-400. DOI=10.1145/1882362.1882443 http://doi.acm.org/10.1145/1882362.1882443
[3]. Premkumar T. Devanbu and Stuart Stubblebine. 2000. Software engineering for security: a roadmap. In Proceedings of the Conference on The Future of Software Engineering (ICSE '00). ACM, New York, NY, USA, 227-239. DOI=10.1145/336512.336559 http://doi.acm.org/10.1145/336512.336559.
[4]. Priya Viswanathan, Software Security: Creating a Secure Mobile App, Steps to Maintain Security during Mobile App Development,http://mobiledevices.about.com/od/Mobi le-Security/tp/Software-Security-Creating-A-Secure-Mobile-App.htm
[5]. Security Concern of Mobile Applications, Article Published by Hong Kong Software Testing and Certification Centre (HKSTCC), 9th May 2013.