hakin9.org

Mobile Application Security

Arash Habibi Lashkari

2–3 minutes

As mobile apps become integral to our lives, they also expose users to a growing range of security threats. These threats exploit

system vulnerabilities, impacting both individuals and connected systems. The constrained processing capabilities and streamlined interfaces of mobile devices further obscures these malicious activities.

In this article, we'll peel back the layers of mobile app security, highlighting the must-knows and offering hands-on tips for both users and developers. From understanding common vulnerabilities to implementing effective security measures, we'll guide you through what it takes to keep mobile applications secure.

Application security threats

Mobile apps are developed through four primary methodologies:

1. Native Mobile Applications are built with platform-specific languages and frameworks running directly on the device's OS, like iOS or Android.

2. Cross-platform Native Mobile Applications are developed using various languages but compiled to run natively on the device's OS.

3. Hybrid Mobile Applications are made with standard web technologies like JavaScript, CSS, and HTML5. These apps are packaged for installation but run within a web container; that bridges to native device APIs, typically using Apache Cordova.

4. Progressive Web Applications (PWAs) offer a different take on mobile development. These are web apps that use browser features to deliver an applike; experience, skipping the need for app store distribution and installations.

Mobile platforms, including wearable tech, present unique security challenges not present in traditional PC environments, with vulnerabilities spanning from application to physical layers.

For instance, at the application layer, malicious attacks can manipulate communication links to generate fake messages or data, leading to increased data collisions.

This challenge of mobile security is amplified by the potential for attacks like mobile botnets, leading to data breaches or denial of service. Both technical and non-technical threats, such as environmental and governmental sources, complicate the security landscape. Despite the advantages mobile apps bring to organizational efficiency, their vulnerabilities underscore the need for robust security strategies that evolve alongside technology to protect mission-critical information.

Application vulnerabilities

Vulnerabilities often arise from coding mistakes or design flaws – especially when developers prioritize speed or cost over security. While Android and iOS do attempt to screen for malware, their review processes aren't transparent, leaving users and organizations to manage security risks on their own. Enterprises, government agencies, and industries with strict regulations need to evaluate app security, privacy, and policy requirements before adopting apps.

The risk level associated with these vulnerabilities varies, particularly with apps that handle sensitive data or connect to wireless networks, as these are vulnerable to remote attacks. Even apps that seem low-risk can have serious consequences if exploited. To mitigate these risks, organizations should adopt a thorough app vetting process and strengthen software assurance to ensure apps are secure and perform as intended.

Information sensitivity

The sensitivity of user information varies across different types of apps, from games to finance. Users tend to be more cautious with financial apps, reflecting different privacy concerns depending on the app's purpose. When apps request sensitive information, users' risk perception increases, making them more hesitant to share data.

This aspect of information sensitivity and its impact on privacy concerns is not fully explored in mobile app research. Yet data breaches appear to make users more cautious when apps ask for sensitive information.

Privacy Awareness

Privacy awareness significantly shapes mobile users' concerns and attitudes towards privacy and security. This awareness varies among individuals, often influenced by their exposure to media, news, and discussions within their social circles about privacy practices and breaches. Those who are more informed about unauthorized data collection tend to be more cautious, which affects their perceptions of mobile app security and encourages protective behaviors.

Best Practices for Mobile Users on App Security

To keep your mobile device secure, consider these key practices:

- **Avoid Saving Passwords**: Don’t store your passwords in apps, especially for sensitive activities like online banking. Instead, use your device’s encryption and passcode protections.

- **Steer Clear of Malicious and Fake Apps**: Avoid third-party app stores, as they often host apps that can steal your data or flood you with unwanted ads. Mobile spying ranges from mass data collection for ads to advanced attacks like Pegasus spyware.

- **Be Cautious About Public Wi-Fi Networks**: Public Wi-Fi in cafes or restaurants can be vulnerable to Man-in-the-Middle (MitM) attacks, where attackers intercept your data. Use encrypted connections or switch to mobile data instead.

- **Be Restrictive with App Permissions**: Only grant app permissions necessary for functionality. Be selective about allowing access to your location, camera, microphone, contacts, and IMEI number.

- **Stay Alert to Social Engineering**: Be wary of unsolicited messages asking you to install apps or share login details. Always verify the source before taking any action.

- **Update your Smartphone OS**: Updates improve device performance, keep apps running smoothly, and add security patches to protect you from cyber threats and malware.

- **Turn off Bluetooth:** Disable when not in use, especially in public wireless environments, to reduce the risk of hackers accessing your device.

- **Update your Applications**: Regularly update your apps to the latest versions. Unfortunately, some developers neglect updates, which can leave your device vulnerable to new security threats.

12 Best Practices for Smartphone App Developers

Mobile app security challenges don't end once your app is downloaded. Users can access the source code, potentially exposing weak spots, and jailbroken devices can leak app data, opening the door to other apps or harmful actors. Here are key practices to strengthen your app's security:

1. Protect the App Code Encryption

Encryption converts data from plain text to ciphertext, unreadable without a decryption key. Use symmetric or asymmetric encryption for data at rest and in transit, secured by an SSL or VPN tunnel.

Another important aspect of encryption is proper key management. Keep keys separate from data, define policies for key usage duration, and retire unneeded keys.

2. Code Obfuscation

Make your code harder to decipher without changing how it functions. While encryption locks data, obfuscation scrambles the code itself, making it tough to reverse-engineer. There are a couple of different ways to do this:

- **Name Obfuscation**: Renaming functions, classes, and variables can help, but alone it's often insufficient since hackers can still recognize patterns and deduce their purpose.

- **Control Flow Flattening**: Rearrange the code flow to make the logic less obvious.

- **Arithmetic Obfuscation**: Replace straightforward calculations with more complex ones.

- **Code Virtualization**: Translate your code into bytecode that only a custom virtual machine can interpret.

3. Secure your APIs

While APIs provide enormous convenience, they're also a huge security risk. Use API keys for verification, add identity checks, and watch for abuse patterns. Encrypt the connection between the client and server to protect against man-in-the-middle attacks, and consider obfuscating your code if API keys are stored within the app.

4. Strong Authentication Policies

Choose between a native login flow for a better user experience or a web-based one for more security. Hypermedia authentication APIs are a solution now popping up to bridge this gap and provide the best of both worlds. Hypermedia authentication APIs interact with the authorization server directly without the need for an intermediary like the browser window.

Make sure you enforce strong passwords, consider multifactor authentication, and manage sessions carefully with complex tokens and proper invalidation.

5. Secure the Data-in-Transit

Critical user data sent between client and server must be shielded from theft or privacy leaks. For rigorous protection, an SSL or VPN tunnel is strongly suggested.

6. File-Level and Database Encryption

Encrypt data stored locally on devices using available database encryption modules or file-level encryption methods.

7. Have a Solid API Strategy

APIs facilitate data exchange between apps, users, and cloud services. If your app uses third-party APIs, limit their access to only the necessary parts of your app.

8. Secure the Backend

Most mobile apps rely on a client-server model, so make sure your backend is secure. Don't assume your app is the only one accessing your APIs. Validate all APIs according to your mobile platform, keeping in mind that API authentication and transport can vary across different platforms.

9. Store Private Data within Internal Storage

Keep sensitive user data within the device's protected internal storage. This data is automatically deleted when the app is uninstalled, and no extra permissions are required to access it.

10. Follow Secure Coding Practices

Start with secure coding principles, regularly test your code for vulnerabilities, and keep your designs straightforward to reduce risks. Use both Static (SAST) and Dynamic Application Security Testing (DAST) throughout development and after release.

Some additional secure coding recommendations include:

- Take a default deny approach to data and enforce the principle of least privilege.
- Validate inputs from all untrusted data sources.
- Sanitize data sent to other systems.
- Keep it simple. Complex designs leave more room for vulnerabilities.
- Conduct code audits and tests to see that the app's authentication and authorization procedures have no loopholes.
- Use a gateway to protect your API.
- Use operating system emulators to see how your app would perform in a simulated environment.
- While creating your data storage systems, keep in mind that no sensitive data should be shared with:
    - The application log
    - Third parties
    - The keyboard cache
    - The IPC mechanism
    - The user's device during interaction

- Including multifactor authentication by mandating the usage of a one-time password (OTP) in addition to the normal password.

11. Hire a Mobile App Security Expert

To ensure your app's safety and data integrity, consider hiring a security specialist. They can provide targeted expertise to strengthen your app against specific security threats.

12. Empower your Users

Even the most secure apps depend on user behavior. Educating your users about online safety not only protects them but also strengthens your app's overall security.

Final Thoughts

We've walked through the minefield of mobile security, highlighting vulnerabilities and offering solid strategies for both users and developers. The key takeaway is that a proactive approach – whether by regularly updating your apps, educating users, or adopting robust coding practices – is essential to safeguard against the ever-evolving landscape of mobile security threats.

Author

Canada Research Chair in Cybersecurity
Associate Professor and IEEE Senior Member
Founder and director of the Behavior-Centric Cybersecurity Center (BCCC)
York University | Toronto, ON, Canada