MOBILE & WIRELESS TECHNOLOGIES

Addressing Security and Privacy Risks in Mobile Applications

Anurag Kumar Jain and Devendra Shanbhag, *Tata Consultancy Services*

Applications for mobile platforms are being developed at a tremendous rate, but often without proper security implementation. Insecure mobile applications can cause serious information security and data privacy issues and can have severe repercussions on users and organizations alike.

The era is long gone when mobile devices were just a means of voice communication. They've evolved into a full-fledged computing platform, thanks to an increase in their processing and storage capabilities, a decrease in their hardware costs, and the advent of mobile platforms such as the Android, iOS, Blackberry, and Windows Phone.

Faster Internet connectivity provided by Wi-Fi and cellular data networks (3G, LTE, WiMax) on these devices, with speeds close to that of a PC environment, has put immense capabilities in mobile users' hands. Internet access through mobile devices is slated to outgrow desktop access by 2014.[1] Smartphones and tablet PCs are thus becoming the devices of choice for accessing information and services such as email, social networking, banking and other financial services, healthcare services, corporate applications, and many others. Consequently, mobile applications providing an array of services are being released in a flurry.

But can the mobile environment be considered secure? Past security incidents—including vulnerabilities found in well-known mobile apps[2] and malware attacks on mobile platforms—suggest that the mobile environment is still far from secure.

There's a plethora of security and privacy risks in mobile environments, including malware that steals confidential information from Android devices or sends short message service (SMS) texts to premium numbers; mobile devices that can access corporate applications, systems, and data;

Published by the IEEE Computer Society

and mobile apps that can expose confidential information. Data security and privacy present serious concerns for enterprises and mobile users.

The Need for Mobile Application Security

Security of mobile applications becomes critical for the following reasons.

Storage and Processing of Sensitive Data

Mobile devices are being used to access a range of services, from social networking, banking, ticketing, and shopping to corporate applications such as email, enterprise resource planning (ERP), customer relationship management (CRM), and calendar and address book applications. These mobile applications store and transmit a lot of sensitive personal and corporate information, such as login credentials, credit card details, private contact entries, invoices, and purchase orders. If developed insecurely, these applications could potentially disclose sensitive information.

Nontransparent Use of Mobile Devices

The distinction between corporate and personal use of mobile devices is becoming blurred. With enterprises encouraging bring-your-own-device (BYOD) policies,[3] the number of personal smartphones and tablets being used in the corporate environment is increasing rapidly. As a result, critical and sensitive corporate applications and data are finding their way onto personal devices.

Using personal phones for corporate purposes makes it difficult to enforce corporate policies and restrictions on these devices. Also, an attacker can more easily compromise personal devices than corporate-issued devices, which are locked down using far more draconian measures. Sensitive corporate applications and data on unmanaged personal devices open up security risks, such as exposure of confidential corporate information through lost or stolen phones, data interception and manipulation through Wi-Fi sniffing, and man-in-the-middle attacks at public Wi-Fi hotspots.

Regulatory Requirements

Around the world, countries have their own regulatory requirements for enterprises that manage sensitive and confidential customer data—such as personally identifiable information, personal health information, cardholder information, and financial information. For example, in the US, enterprises that handle such information need to comply with various regulations, such as the Gramm-Leach-Bliley Act,[4] Health Insurance Portability and Privacy Act,[5] and so on.

Also, the Payment Card Industry Data Security Standards Council mandates certain security measures to be taken by organizations handling cardholder information.

Enterprises releasing mobile apps that deal with such confidential information must ensure adherence to security measures mandated by these regulations. Violation of compliance could lead to hefty fines or lawsuits against the organizations.

Mobile Platforms: New Target for Attackers

Because more sensitive data now resides on mobile devices, hackers are gradually shifting their attention toward the mobile environment. We've already witnessed an increase in malware attacks on mobile platforms, particularly Android, targeting sensitive information. According to McAfee, the cumulative malware number exceeded the 70 million mark by the end of 2011.[6] A vulnerable application can become an easy target for a determined attacker.

New Technologies: Newer Risks

Newer technologies—such as near-field communication (NFC) and Quick Response (QR)[7] codes—open up newer attack vectors for malicious users. Eavesdropping, data corruption, data manipulation, and malware on mobile devices are some of the common security concerns with NFC (see www.nearfieldcommunication.org/nfc-security-risks.html). QR codes are also being used as an attack vector, facilitating phishing attacks and redirecting users to malicious websites that host viruses, worms, and Trojans. The emergence of mobile payments using NFC can become an attractive target for attackers. Assuming that mobile communication technologies such as GSM, Wi-Fi, 3G, NFC, and the like are inherently secure can be dangerous.

Threats in a Mobile Environment

Similar to a traditional PC environment, multiple threats in a mobile environment can jeopardize the functioning of applications and devices and can cause misuse of confidential data. The high

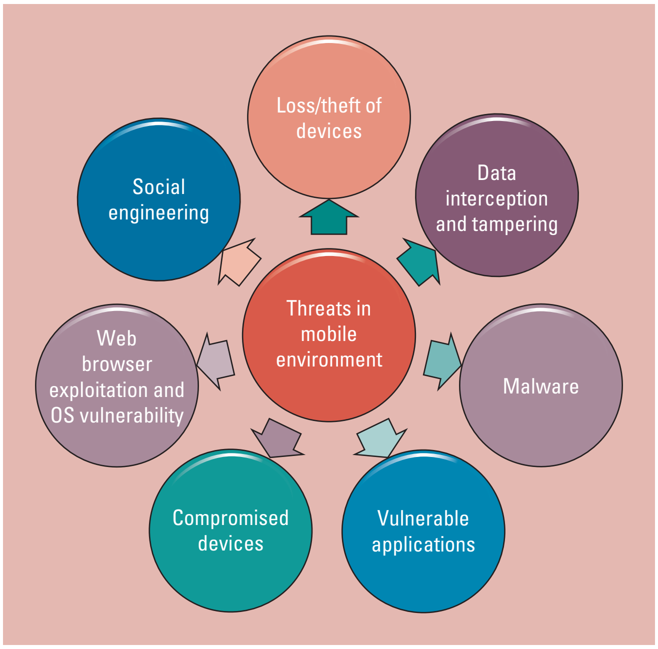
Loss/theft of devices
Social engineering
Data interception and tampering
Threats in mobile environment
Web browser exploitation and OS vulnerability
Malware
Compromised devices
Vulnerable applications

Figure 1. Prominent threats in a mobile environment. Threats such as device loss or theft, data interception and tampering, and malware gain more prominence in a mobile environment.

portability of the mobile devices brings in additional security threats, such as exposure of confidential personal and corporate information stored on these devices. A massive number of mobile devices are reported lost or stolen every year.[8] The cost of a lost or stolen mobile device might be a few hundred dollars, but the value of personal and corporate information stored on the device is much more.

Figure 1 highlights some of the major threats related to a mobile environment. Some of them are similar to those prevalent in a traditional desktop environment; however, threats such as device loss or theft, data interception and tampering, and malware gain more prominence in a mobile environment.

Security Issues with Mobile Applications

Mobile applications can either be *mobile Web* or *native*. Security issues in mobile Web applications closely resemble those of traditional Web applications because of similar underlying development technologies. Native applications inherit a few of the vulnerabilities from the Web applications and bring in additional security risks. Native app development is platform-specific—the code base differs for each platform, which is different from desktop environment. This diversity poses significant challenges for development teams, because they have to start fresh in incorporating security in mobile applications instead of reusing existing secure desktop/Web applications—causing vulnerabilities to creep in. We highlight security issues in both kinds of applications.

Mobile Web Applications

The mobile Web is currently the third most used platform behind the Android and iOS.[9] It commonly employs lightweight pages using Web application development languages and technologies such as XHTML, HTML5, CSS, JavaScript, and so on.

Due to similarities with traditional Web applications, mobile Web applications also suffer from security issues similar to those of Web applications for a desktop PC environment, such as cross-site scripting (XSS), SQL injection, nonSSL login, cross site request forgery, session fixation, and HTTP redirects. The Open Web Application Security Project (Owasp; www.owasp.org) or Web Application Security Consortium (www.webappsec.org) provide more details about these issues.

Native Applications

As of January 2012, there were over 500,000 applications in the Apple App Store and over 400,000 applications in the Android market.

Each mobile platform provides its own security features for device- and application-level security. Mobile operating systems such as the Apple iOS and Android (Honeycomb and later versions) provide device-level encryption. The Android and WP7 provide sandboxing and enforce a white-listed permission model for accessing services and also provide access permissions for resources such as application files and databases. Most mobile operating systems provide APIs for encryption that can be leveraged in the applications, too. However, these platforms rely on the application developers to implement some of these application security features, and that's where security issues can creep into applications.

According to the Owasp Mobile Security Project (www.owasp.org/index.php/OWASP_Mobile_Security_Project), the top risks for mobile applications are as follows.

Insecure data storage. This risk occurs when sensitive data is stored on a device or when cloud-synced data is left unprotected. It's generally the result of not encrypting sensitive data, caching information not intended for long-term storage, allowing global file permissions, or failing to leverage best practices for a particular platform. This in turn leads to the exposure of sensitive information, privacy violations, and noncompliance.

Weak server-side controls. Failure to implement proper security controls such as patches and updates or secure configurations, changing default accounts, or disabling unnecessary backend services can compromise data confidentiality and integrity.

Insufficient transport-layer protection. Mobile applications often use the HTTP protocol for client-server communication, which communicates all information in plain text. Even when they provide transport-layer security through use of the HTTPS protocol, if they ignore certificate validation errors or revert to plain-text communication after a failure, they can jeopardize security by revealing data or facilitating data tampering through man-in-the middle attacks.

Client-side injection. Apart from the known injection attacks such as XSS, HTML injection, and SQL injection applicable to mobile Web and hybrid apps, mobile apps are witnessing newer attacks such as abusing the phone dialer, SMS, and in-app payments.

Poor authorization and authentication. Poor authorization and authentication schemes relying on device identifiers for security—such as the International Mobile Equipment Identifier, International Mobile Subscriber Identity, or Universally Unique ID values—are a recipe for failure. They can lead to broken authentication and privilege-access issues.

Improper session handling. Sessions that have long expiration times or that use device identifiers as the session ID pose security risks, such as privilege escalation and unauthorized access.

Security decisions via untrusted input. If applications make security decisions via user input, then the used inputs can be leveraged by malware or client-side injection attacks for various nefarious purposes, such as consuming paid resources, exfiltrating data, or escalating privileges. For example, attackers have abused URL schemes in the iOS and intents in Androids.

Side-channel data leakage. Programmatic flaws or the failure to disable insecure OS features in applications can result in sensitive data ending up in Web caches, global OS logs, screenshots (an iOS back-grounding issue), and temp directories. The data is then up for grabs for malware or an attacker who steals your phone.

Broken cryptography. This risk emanates from insecure development practices, such as using custom instead of standard cryptographic algorithms, assuming that encoding and obfuscation are equivalent to encryption, or hardcoding cryptographic keys into the application code itself. Such practices can result in a loss of data confidentiality or privilege escalation.

Sensitive information disclosure. Hardcoding sensitive information—such as login credentials, shared secret keys, access tokens, and sensitive business logic—into the application code means that an attacker might be able to access such information through reverse engineering, which is fairly trivial. Once the attacker has this information, it's easy to access more sensitive data. Code obfuscation makes it difficult to comprehend code, but can it completely deter a determined attacker? We don't think so.

Secure Mobile-Application Development

Information security should be an essential part of the mobile application systems-development life cycle from the start. It reduces the cost of security implementation and fixing issues later. The secure development approach described here provides measures to incorporate security during various application development phases to help organizations deliver secure mobile applications (see Figure 2).

Secure Application Design

Before designing an application, it's critical to understand the business requirements, users, data sensitivity, and potential device types—whether managed or unmanaged. Design considerations

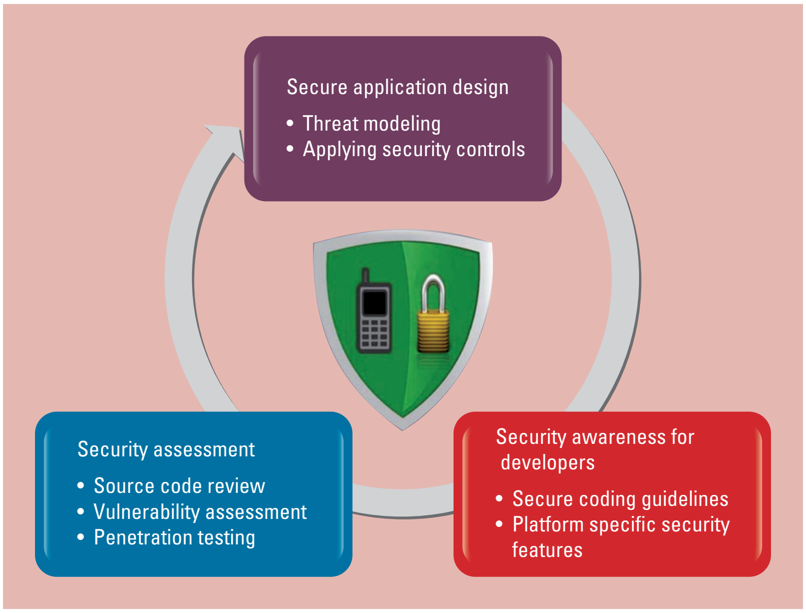
Secure application design
• Threat modeling
• Applying security controls
Security assessment
• Source code review
• Vulnerability assessment
• Penetration testing
Security awareness for developers
• Secure coding guidelines
• Platform specific security features

Figure 2. An overview of a secure mobile development approach. The goal is to incorporate security during various application development phases to help organizations deliver secure mobile applications.

pertaining to access control and privileges, data encryption (in transit and local storage), and strong password and account-lockout policies should be included in any mobile application design. An adequate threat-risk modeling of the application should be considered in this phase to identify any security risks and determine the correct controls and countermeasures to thwart those risks within budget.

Adequate Security Controls

Based on the criticality of the application, appropriate security controls can be incorporated to plug the security gaps identified during threat modeling. Here, we review some security controls to consider. Note that before employing any security control, you must thoroughly evaluate its applicability, platform support, and any additional infrastructural requirements and their costs.

Multifactor authentication schemes. Multifactor authentication schemes address the shortcomings of schemes based on traditional passwords, personal identification numbers, and secret questions—which can be susceptible to guessing, dictionary, and brute-force attacks. They employ two or more independent factors as part of the user credentials and add another layer of security over the existing authentication process, thus increasing their strength. Controls such as one-time passwords, grid-based authentication, and digital-certificate-based authentication schemes can help augment existing security controls in the application.

Digital signature. Mobile applications can leverage a public-key infrastructure to incorporate authentication, integrity, and non-repudiation. Mobile operating systems (such as iOS) provide support for managing digital certificates on mobile devices and provide APIs to generate and verify the digital signature.

Transport-layer and data encryption. Data encryption is a critical requirement for sensitive data at rest and in transit. Applications can encrypt sensitive data in transit end-to-end via SSL/TLS encryption mechanisms. The latest mobile devices, such as the iPhone, the iPad, and some Android tablets, can perform file- and device-level encryption. Mobile operating systems, such iOS, Android, and Blackberry OS, provide encryption libraries that can encrypt data at rest.

Secure Coding Guidelines

Developer training and awareness of secure coding guidelines[10,11] are vital to secure application development. Training must be conducted before the initial coding phase and should cover common programming errors that introduce vulnerabilities. Awareness of Owasp's list of top 10 mobile risks presented earlier is a good starting point. Furthermore, developers should be familiar with the following security guidelines.

Perform secure logging and error handling. Logging to the global log, logging commented code for debugging purposes, and performing poor exception handling can disclose sensitive information in applications.

Follow the principle of least privilege. Correctly implementing the permission model provided by mobile OS and following the principle of least privilege ensures sandboxing and isolation.

Validate input data. It's important to implement input validations correctly and duplicate client-side validations on the server side as well. It's also a good idea to implement security controls for input validations, such as Owasp's Enterprise Security API.

Implement secure data storage. Avoid storing sensitive data on client devices unless absolutely necessary and use standard encryption algorithms with strong key values instead of homegrown ones to encrypt sensitive data residing on devices or at the server backend.

Avoid insecure mobile OS features. Insecure features such as "cut-copy-paste" and "autocompletion" provided by mobile OSs can be exploited to extract sensitive data. Such features should be turned off in the application.

Security Assessments of Mobile Applications

All applications must undergo a thorough security assessment[10,11] before being released into production to confirm that attack surfaces discovered during threat modeling have been addressed. Security assessment can also expose security gaps between project designs and approved corporate policies, which might have evolved during development, or problems resulting from the integration of different modules.

A good approach would be to combine static (secure-code) reviews and dynamic analysis (also known as black box/grey box security assessments) of applications before they go live.

Attackers will continue to target mobile devices and applications. With 650 million mobile devices expected to be shipped in 2012,[12] and e-commerce on mobile devices becoming more mainstream, various manifestations of attacks and malwares against mobile platforms will grow more prominent in the coming years. Securely managing mobile devices and incorporating security into the application development life cycle should be a top agenda item for any organization hoping to protect itself and its consumers from potential attacks. IT

References

1. "Internet Trends," Morgan Stanley, 12 Apr. 2010.
2. R. King, "ViaForencis: Netflix, Foursquare Apps Leave Sensitive Data Vulnerable," ZDNet.com, 10 June 2011; www.zdnet.com/blog/btl/viaforensics-netflix-foursquare-apps-leave-sensitive-data-vulnerable/50330.
3. P. Rubens, "4 Steps to Securing Mobile Devices and Apps in the Workplace," eSecurityPlanet.com, 9 Apr. 2012; www.esecurityplanet.com/mobile-security/4-steps-to-securing-mobile-devices-and-apps-in-the-workplace-mdm-byod.html.
4. "SOX, GLB, SB 1386 and Mobile Devices," white paper, Credent, April 2012.
5. A.H. Greene, "When HIPAA Applies to Mobile Applications," *MobiHealth News*, 16 June 2011; http://mobihealthnews.com/11261/when-hipaa-applies-to-mobile-applications.
6. "McAfee Threats Report: Third Quarter 2011," McAfee Labs, 2011; www.mcafee.com/us/resources/reports/rp-quarterly-threat-q3-2011.pdf.
7. D. Braue, "Malicious QR Codes: A Mobile Security Blind Spot," *PC Advisor*, 3 May 2012; www.pcadvisor.co.uk/news/mobile-phone/3355645/malicious-qr-codes-mobile-security-blind-spot.
8. "Mobile Phone Theft Increasing across the UK," Insure4U.info, 6 Oct. 2010; www.insure4u.info/home-insurance-mobile/mobile-phone-theft-increasing-across-the-uk.html.
9. M. Kapetanakis, "Developer Economics 2011—Winners and Losers in the Platform Race," Vision Mobile blog, June 2011; www.visionmobile.com/blog/2011/06/developer-economics-2011-winners-and-losers-in-the-platform-race.
10. J. Burns, "Developing Secure Mobile Applications for Android," iSEC, Oct. 2008; www.isecpartners.com/files/iSEC_Securing_Android_Apps.pdf.
11. "Introduction to Secure Coding Guide," iOS Developer Library, Feb. 2012; http://developer.apple.com/library/ios/#documentation/Security/Conceptual/SecureCodingGuide/Introduction.html.
12. "Mobile Threat Report," Lookout Mobile Security, Aug. 2011; https://www.mylookout.com/_downloads/lookout-mobile-threat-report-2011.pdf.

Anurag Kumar Jain *is an information security analyst at Tata Consultancy Services. His research interests include information security and penetration testing of Web and mobile applications. Contact him at anuragonl@gmail.com.*

Devendra Shanbhag *is an information security analyst at Tata Consultancy Services. His research interests include Android and iOS native and Web application development and security assessment, penetration testing, and threat modeling. Contact him at devendra.shanbhag@tcs.com.*