IBM Software
Thought Leadership White Paper

March 2013

Ensuring application security in mobile device environments

Detect, analyze and eliminate application security vulnerabilities with IBM Security AppScan

IBM

Contents

2 Mobile application environments

3 Types of mobile applications

4 How mobile application security can be compromised

4 Potential security risks for mobile applications

6 How to prevent vulnerabilities in mobile applications

7 Using IBM Security AppScan to identify vulnerabilities

7 Extending application security intelligence with IBM

7 For more information

7 About IBM Security solutions

In today's business environments, mobile devices such as smartphones and tablets make up the fastest growing segment of computing devices—outpacing desktop and laptop computers. As more employees prefer to use mobile devices in the workplace, organizations are rapidly moving towards a bring-your-own-device (BYOD) model—allowing employees to use their own mobile devices for business purposes. This often leads to employees having a mix of corporate and personal applications on the same device, which gives the security team less control over devices that can access corporate networks.

As a result of the increase in wireless devices in the workforce, organizations are becoming more concerned with mobile security. Many, in fact, see this area as a primary technology challenge to address and a main focus for security initiatives.[1] This is because mobile device applications have the potential to interact with confidential or sensitive information. Hackers have noticed this fact and have started targeting these applications. The resulting attacks, frequently reported by the media, can lead to decreased trust in an application or an organization that uses it. Although some application environments have become increasingly standardized and secure, there is considerable room for concern and significant need to provide improved security for mobile applications.

Mobile application environments

For the current generation of smartphones and tablets, the two most commonly used application environments are iOS and Android. These operating systems support a broad range of applications—from web applications that run within the device's web browser to native applications that run directly on the device's operating system.

iOS

iOS is the operating system developed by Apple that runs on several products including the iPhone, iPod Touch and iPad. Only hardware produced by Apple can run iOS, and Apple controls the native applications that can be installed on iOS-based devices. These applications are distributed through Apple's marketplace, the App Store. When applications are submitted by developers to the App Store, Apple screens them and either accepts or rejects the applications based on results from their analysis.

Android

Android is the mobile device operating system produced by Google. Many hardware manufacturers produce smartphones and tablets that run the Android operating system. Unlike iOS, however, Android is open source, so each hardware manufacturer can provide a custom version of the operating system on its

hardware. Android applications are available through marketplaces similar to the Apple App Store, but there are fewer restrictions on applications that may be distributed. Additionally, users can download Android applications directly from websites to their devices, circumventing marketplaces entirely.

Types of mobile applications

For both iOS and Android environments, there are three types of mobile applications: *web*, *native and hybrid*. The application types differ in how they are developed, what they can do, how they perform and how they are distributed. Each type of application has security vulnerabilities—some unique to each type of application, some common across all types of applications.

Web applications

iOS- and Android-based mobile devices include fully functional web browsers, and any website that can be accessed from a standard computer can be accessed from these devices. Web applications designed for mobile devices use the same components as traditional web applications, and they access the same data through the same servers. The only major difference between web applications designed for standard computers and those designed for mobile devices is how they are rendered.

Native applications

iOS and Android operating systems support native applications that can be downloaded and run on mobile devices. These applications generally have better performance than web applications running on mobile web browsers, and they have tighter integration with available hardware.

Native applications for iOS are usually written in Objective-C, developed in the Xcode integrated development environment (IDE) and then distributed through the Apple App Store. Once they have been installed, iOS applications may access hardware on the mobile device—such as global positioning satellite (GPS) technology. The user is often prompted to verify an application's access to this hardware.

Native applications for Android are typically written in Java and developed in Eclipse, but there are many options for developing them—through different IDEs or even without an IDE. Once an application is built, developers can either upload it to one of several Android markets or have it hosted on a personal or business website for users to download directly. Upon installation on a mobile device, Android applications request user permission to interact with hardware. Once the application is running on the device, it can communicate with other applications running locally on the same device.

Hybrid applications

A third category—hybrid applications—consists of native applications containing web browser components that load and run web applications. A hybrid application is a compromise between a web application and a native application. With hybrid applications, developers can use native application components to customize the look and feel of the application and use web application components to help overcome the update limitations of native applications.

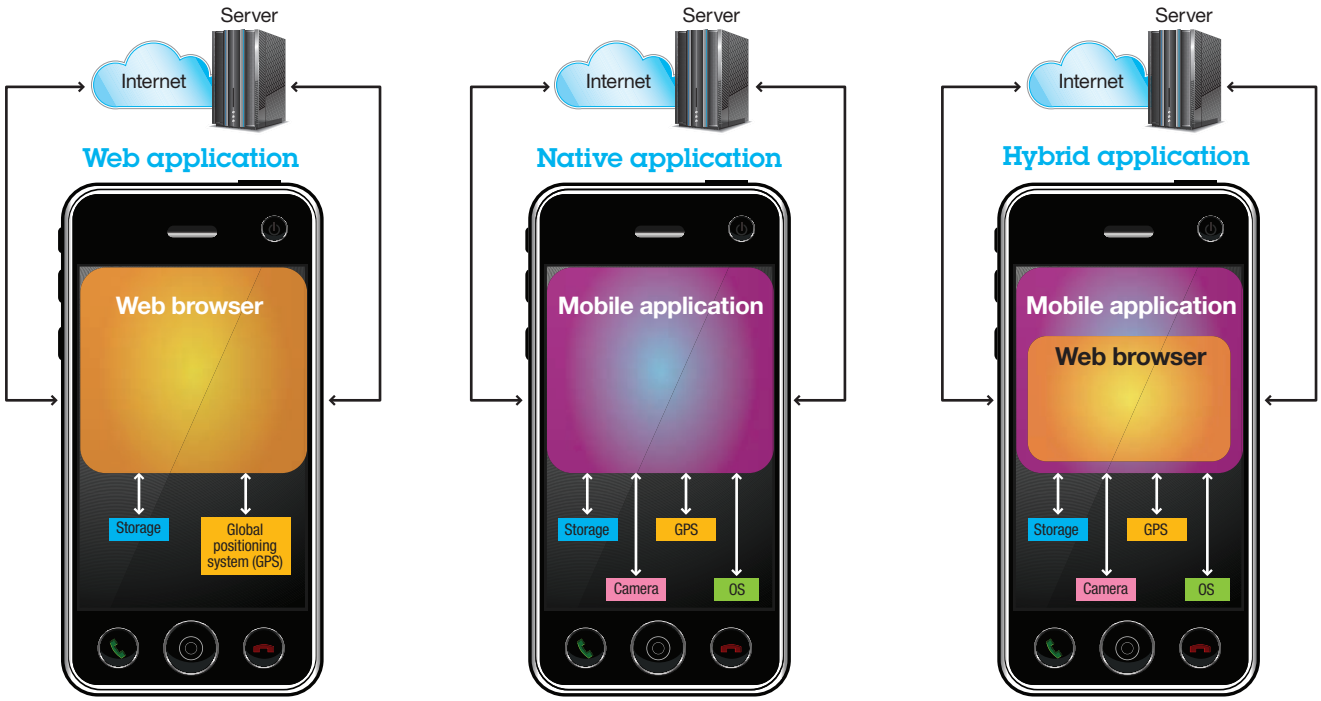

Figure 1. Three types of mobile applications—web, native and hybrid—communicate with mobile device components, web application servers and the Internet. Each of these paths presents a potential vulnerability for attacks.

Each type of mobile application has unique purposes and advantages, but each category is subject to security threats as well. There are several areas of vulnerability for attackers to exploit, which can lead to potential loss or theft of sensitive business or personal information.

How mobile application security can be compromised

Users are capable of installing a variety of applications on their mobile devices. But since users generally have no means of performing a security analysis on them, the applications they install may be malicious or include gaps in security. Even when applications are not intentionally malicious, they can have design flaws that make them insecure. Attackers can do anything from intercepting Internet traffic, to sending crafted data to a user's device, to stealing the actual device to exploit applications.

Potential security risks for mobile applications

Mobile applications have the ability to access security-critical servers, storage and networking systems. An attacker who can exploit an application can access or disrupt these systems as well. In addition to attacking a system, defacing a web page and stealing web-page data, mobile applications are capable of

accessing address books, discovering location information, sending text messages, making calls and accessing internal networks.

The security risks are slightly different for each type of mobile application. Web applications can have client- or server-side vulnerabilities, while native applications can have risks at the application level—where confidential information or the application itself is left open for attack—or the mobile device level. For example, battery life or the ability to make phone calls can be put at risk with native applications. Hybrid applications have the combined security risks of the other two types.

Consider a typical business messaging application that contains login credentials for a private messaging network, contact information for coworkers and message transcripts from past conversations. If this native application is exploitable, an attacker could collect the private contact information, read confidential information in the message transcripts or send out fabricated messages to people in the company—spreading false information and defaming the owner of the mobile device.

Once attackers have access to an exploitable application, they can abuse the application until the user actively stops the attackers or the exploitable application is fixed by developers and updated by the user.

Security risks for native iOS applications

Inter-application communication works differently on native iOS and Android applications, so there are different security vulnerabilities. But at the highest level, native applications on both platforms can inadvertently expose personal information to other parties—and expose applications to third-party/untrusted data.

For native iOS applications, URL schemes are a way to send and receive data between applications, but they are intended for public communication. The openURL method is for sending data to other applications, while the didFinishLaunchingWithOptions method is for receiving data. The latter method has risks in that it can receive untrusted data from another application, and that data can be malicious and cause the application to act suspiciously. When using openURL, there is no authentication that the data it sends has reached the correct application—or whether the data reaches any application at all. That's why sensitive data should not be sent, since it could be potentially intercepted or lost.

In addition, native iOS applications can expose or "leak" confidential data when communicating with a web server, opening web views or creating external notifications.

Security risks for native Android applications

For native Android applications, intents are the preferred mechanism for inter-application communication. And just like on the iOS platform, the sending and receiving of data with intents can create security exposures. Senders of an intent can verify that the recipient has a specific permission, and only applications with that permission will receive the intent. However, the startActivity method for an intent—which launches activities that interact with the user—can receive untrusted data from another activity, and that data can be malicious and cause the application to act suspiciously.

As reported in the news, popular Android applications can leak data when connected to a vulnerable network, such as a Wi-Fi hotspot. This means someone on the network could modify the content in transit. In addition, opening web views with native Android applications can introduce common web security issues, such as cross-site scripting attacks via JavaScript. Loading code from external storage devices can also create security exposures—for example, another application could modify the content on the device.

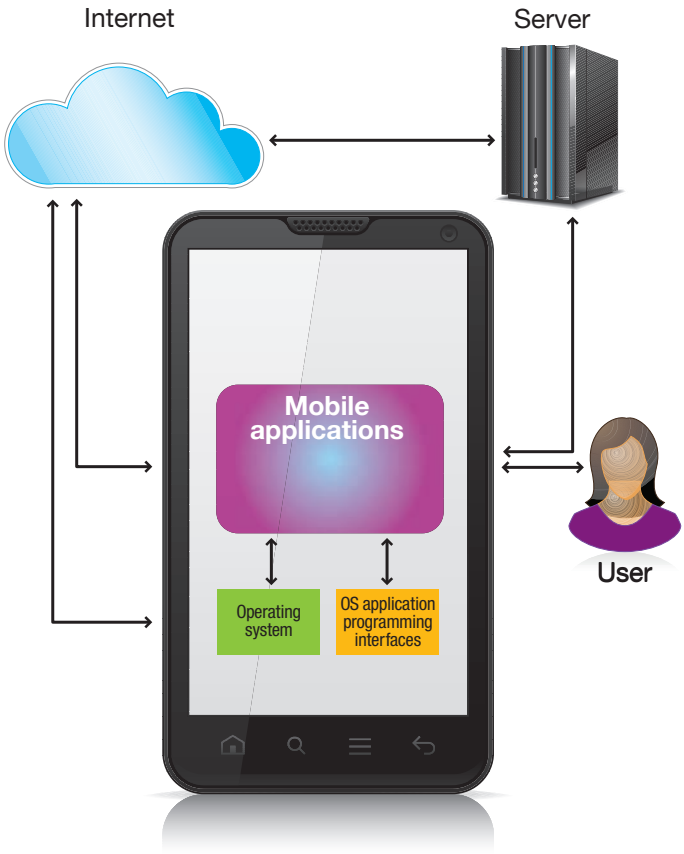

Figure 2. Mobile phone applications can include a number of vulnerabilities that hackers may be able to exploit—vulnerabilities that lie in many possible communication paths.

How to prevent vulnerabilities in mobile applications

Vulnerabilities in mobile applications are becoming more common. In one specific example for an iOS application, a vulnerability was detected in which the application was sending unencrypted data of personal address books to servers belonging to software vendors.[2] In another example involving an Android application, a vulnerability was found that could put personal user information at risk, including account balances, location information and phone numbers.[3] Implementing best practices in application development and analysis can help prevent security issues such as these.

Best practices for writing application code

When creating mobile applications, organizations can benefit from implementing a set of best practices for writing code. Spanning application categories, the following best practices can help organizations prevent and eliminate security vulnerabilities:

- Minimize functionality and make the code as simple as possible
- Minimize permissions that are required or requested
- Validate all data before using it in the application
- Do not store or transmit data unless necessary
- Use encryption to store and transmit data
- Conduct thorough code reviews
- Plan carefully to pick the best type of application to build
- Conduct static analysis to detect problems
- Perform dynamic analysis to detect problems
- Utilize instrumentation to monitor applications
- Conduct testing to verify there is no unintended functionality

Detect attacks using taint analysis

In addition to implementing best practices for creating applications, the practice of *taint analysis* can be useful to prevent vulnerabilities as well. Taint analysis is a specific type of static analysis that is well-suited to detect integrity violations, such as applications using data from untrusted users. It is also helpful to identify confidentiality leaks, such as applications using private user data.

Although using best practices and performing taint analysis can be useful in creating secure applications, having the right tools to identify vulnerabilities can be invaluable to organizations looking to further enhance application security and improve detection and analysis efficiency.

Using IBM Security AppScan to identify vulnerabilities

Designed to identify security vulnerabilities in mobile applications, IBM® Security AppScan® Source is a powerful application security testing solution that can help organizations ensure that both iOS- and Android-based native applications are safe. As part of IBM Integrated Mobile Security Software Solutions, the IBM Security AppScan portfolio uses a combination of static and dynamic analysis to detect potential security issues in applications early in the development cycle—where defects can be fixed quickly with minimal costs and impact to resources.

IBM Security AppScan uses the same techniques to scan web applications for mobile devices that are used to scan web applications for standard computers. This essentially enables organizations to extend their current application security programs to cover their mobile applications as well. IBM Security AppScan also integrates with IBM Rational® application development tools for proactive vulnerability detection, with IBM Security Network Intrusion Prevention System to provide vulnerability data (for active threat protection) and with IBM Security QRadar® SIEM to make application vulnerability information part of the overall security intelligence.

Scanning web, native or hybrid applications is easy using IBM Security AppScan:

- **Web applications:** Simply load the server application or the client web pages into the IBM Security AppScan program and run a scan. IBM Security AppScan can be used to scan web applications designed for any kind of mobile device.
- **Native or hybrid applications:** To analyze an iOS application, import it from xCode into IBM Security AppScan, and then run a scan. To analyze an Android application, import it from Eclipse into IBM Security AppScan, and then run the scan.

IBM Security AppScan can also be used to scan a wide range of server applications—including those that might not be currently running on mobile devices, but may do so at a later date.

Extending application security intelligence with IBM

With an increased wireless workforce in today's BYOD environment, mobile application security is now a top priority for many IT security managers. Compromised application security can cause substantial damage to an organization's sensitive data and public image. Each category of applications for iOS and Android operating systems—web, native and hybrid—has unique security vulnerabilities that need to be addressed. IBM Security AppScan offers a powerful, simplified solution, providing the ability to expand security intelligence required to identify and prevent application vulnerabilities with ease and efficiency.

For more information

To learn more about IBM Security AppScan, contact your IBM representative or IBM Business Partner, or visit: **ibm.com**/software/awdtools/appscan/

About IBM Security solutions

IBM Security offers one of the most advanced and integrated portfolios of enterprise security products and services. The portfolio, supported by world-renowned IBM X-Force® research and development, provides security intelligence to help organizations holistically protect their people, infrastructures, data and applications, offering solutions for identity and access management, database security, application development, risk management, endpoint management, network security and more. These solutions enable organizations to effectively

manage risk and implement integrated security for mobile, cloud, social media and other enterprise business architectures. IBM operates one of the world's broadest security research, development and delivery organizations, monitors 13 billion security events per day in more than 130 countries, and holds more than 3,000 security patents.

Additionally, IBM Global Financing can help you acquire the software capabilities that your business needs in the most cost-effective and strategic way possible. We'll partner with credit-qualified clients to customize a financing solution to suit your business and development goals, enable effective cash management, and improve your total cost of ownership. Fund your critical IT investment and propel your business forward with IBM Global Financing. For more information, visit: **ibm.com**/financing

[1] "Finding a strategic voice: Insights from the 2012 IBM Chief Information Security Officer Assessment." *IBM Center for Applied Insights.* 2012. http://www.ibm.com/common/ssi/cgi-bin/ssialias?subtype=XB&infotype=PM&appname=CHQE_CI_CI_USEN&htmlfid=CIE03117USEN&attachment=CIE03117USEN.PDF

[2] "iOS Social Apps Leak Contact Data." Mathew J. Schwartz. *Information Week.* 2012. http://www.informationweek.com/news/security/privacy/232600490

[3] Bug in Skype for Android Could Expose Your Personal Data." William Fenton. *PC Magazine.* 2011. http://www.pcmag.com/article2/0,2817,2383639,00.asp

IBM Corporation
Software Group
Route 100
Somers, NY 10589

Produced in the United States of America
March 2013

WGW03009-USEN-01