Ensuring application security in mobile device environments

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

2 Mobile application environments
3 Types of mobile applications
4 How mobile application security can be compromised
5 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 Systems software

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. 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 them. 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.
Ensuring application security in mobile device environments

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. Attackers can use these malicious native applications to exploit the user or to exploit other applications on the device. Attackers can also send payloads to web or hybrid applications to use them to exploit mobile devices.

The results of these attacks can range in purpose and severity, and there are a number of potential security risks that organizations should consider to sufficiently secure their mobile environments.

![Diagram of mobile application types and interactions](image)

*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.
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. Each type of mobile application has a slightly different set of risks because each has a different design and set of capabilities.

Security risks for web applications

Web applications involve two main components—the server and the client. Server-side vulnerabilities—such as insufficient screening of client data—may be present in the part of the application that runs on the server. Vulnerabilities on the client side can potentially be exploited inside the web page when it is rendered and executed inside a web browser.

On the server side, the server may accept data from untrusted clients and process the data to return a response to the client. This untrusted data might be used to access a database, a file system or other sources of security-critical information. If the server does not properly sanitize the untrusted data, it could cause corruption in the database, expose confidential files or open the door to other forms of damage.

On the client side, executing a web page sent from the server typically involves rendering the page and executing the JavaScript. The web browser contains a basic security mechanism called Same Origin Policy (SOP)—a policy that basically states that only pages from a specific origin can access stored data and execute scripts. This is necessary because some websites may store private information, such as login credentials. The ability of malicious pages to directly access and modify trusted pages is therefore a major security concern. SOP, however, prevents two pages from different origins from directly interacting with each other. Similarly, the client executes all code in the context of a specific origin—so if untrusted data from an attacker is somehow executed, the attacker has full power to access and modify the page. This means attackers can capture keystrokes, steal entered data, deface the page or execute a convincing phishing attack.

Security risks for native applications

Native applications have their own set of security concerns, which generally fall into two categories—risks to the application and risks to the mobile device. A risk to the application is anything that can endanger confidential information or the application itself. A risk to the mobile device is anything that can occur outside the application, such as sending text messages, draining the battery or making phone calls.

Risks to native applications can be demonstrated through a scenario with a typical business messaging application. In this example, an application contains credentials to log the user into the private messaging network, contact information for people in the company and message transcripts from past conversations. If this 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.
Ensuring application security in mobile device environments

Security risks for hybrid applications
Since hybrid applications are part native application and part web application, they have the combined security risks of the other two application types. Determining exactly where the security risks are located depends on how the hybrid application was built.

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. 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. 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

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.
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 Android-based native mobile 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 (IPS) to provide vulnerability data (for active threat protection) and with QRadar SIEM from Q1 Labs (an IBM company) to make application vulnerability information part of the overall security intelligence.

Scanning web, native, hybrid or even server 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**: Load the Android application into Eclipse software, import the application from Eclipse to IBM Security AppScan, and then run a scan.
- **Server applications**: To scan a server application, simply load it into IBM Security AppScan and run a scan. IBM Security AppScan can be used to scan server applications independently of the application on the mobile device.

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 Systems software
The IBM security portfolio provides the security intelligence to help organizations holistically protect their people, infrastructure, data and applications. IBM offers solutions for identity and access management, database security, application development,
risk management, endpoint management, network security and more. IBM operates the world’s broadest security research, development and delivery organization. This comprises nine security operations centers, nine IBM Research centers, 11 software security development labs and the IBM Institute for Advanced Security with chapters in the United States, Europe and Asia Pacific. IBM 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](http://ibm.com/financing)


2 For more information on SOP, see the [Browser Security Handbook](http://code.google.com/p/browsersec/wiki/Part2#Same-origin_policy).
