Security Issues in Next Generation Android Mobiles

Deepak Dixit  BE Final Year Assistant Professor  Department of I.T  SRCEM, Banmore

Mr. Koushel Agarwal  Department of I.T  SRCEM, Banmore

Email: dpkdixit9@gmail.com  Email: koushel_agarwal@yahoo.co.in

Abstract—The use of mobile devices has changed since the advent of digital technologies such as GSM. With modern smart phones, users are able to browse the Internet and obtain the service such as e-banking, navigation, social networking. The Android operating system is widely used within several types of embedded & mobile platforms, including mobile phones and tablets, and the industry is exploring the Ability of Android within other embedded platforms. In this paper, I review significant threats to security of android based mobile phones. I also propose novel solution directions in order to tackle some of these challenges in wireless networks and mobile ad hoc networks.

1. Introduction

The Android OS is an operating system primarily designed for mobile platforms by Google. It is an open source OS based on LINUX kernel (version 2.6). Android is finding widespread acceptance in the mobile and portable computing market, and this study examines, for the first time, its performance & reliability in more demanding embedded real-time applications. In addition to the Linux based kernel various libraries were added to the platform in order to support higher functionality. Many of these libraries originate from open source projects. They also developed their own Java runtime engine, optimized for the limited resources available on mobile platform called the "Dalvik Virtual Machine". The Android Runtime System utilizes the Dalvik virtual machine, which allows multiple applications to be run concurrently as each application is its own separate VM. As of July 2010, the latest version of Android available was v2.2 (Froyo) and v3.0 (Gingerbread) is expected before the end year. The analysis described below was performed during the fall of 2009 on a Sprint HTC Hero running Android v1.5 (aka Cupcake). The Hero is a little different than a standard Android phone because HTC employs its own Sense user interface (UI) on the device, which will not be used on any Google-branded devices.
It is clear that the sensitivity and confidentiality of users and data transiting in such digital cellular networks is paramount both to businesses and private users. Security and privacy in such networks is achieved at several levels in their architectures, such as the air interface, the operator’s internal network and the inter-operator links. The main assumption underlying the security of legacy mobile networks, such as GSM and UMTS, is the trust that each operator has in its own infrastructure and in other operators with whom it has a roaming agreement. Our goal in this paper is to raise awareness about security and privacy issues in Android Mobile, by reviewing some significant security threats and. Our solutions are inspired from similar research efforts mobile ad hoc network (MANET).

II. Anatomy of Android Application

Activities are classes that provide an interface. An Activity is given a window in which to add User Interface to. Therefore, creating multi-screen applications involves creating multiple Activities and transitioning between them. The Activity class inherits from the abstract Context class. Context is the closest Android gets to a reference to the current application and provides a mechanism for accessing the Android system. A Context is needed to perform many operations in Android such as:

- Accessing Android services, Accessing preferences, Creating views, Accessing Device Resources.

Intents are used throughout Android to make things happen by sending messages. Intents are most commonly used within applications to launch Activities. To launch a new Activity, we create a new Intent, set the Context and the Activity class to launch and then tell the OS to handle the Intent, which launches the Activity. Intents are a powerful concept as they allow the creation of loosely coupled applications. Intents can be used to communicate between any installed application components on the device. An Intent object can contain information for the receiving component. For example if your application calls via Intent a browser it may send the URL to the browser component. Intent also contains information for the Android system so that the Android system can determine which component should handle the request.

Every Android application needs to include a file called AndroidManifest.xml. This file contains information about the application such as:

- The components that make up the app, including registration of Activities and Intents.
- The permissions the app requires
- The minimum Android API level the application support.

Service is code that is long-lived and runs without a UI. A good example of this is a media player playing songs from a playlist.

III. Security and Privacy Challenges In Android Mobiles

The attacks are categorized based on two threat models, depicted in Figure 1. We study how malicious web pages can attack Android applications.

Attacks from Malicious Web Pages

In this attack model, we assume that apps are beginning, and they are intended to serve a web application, such as Facebook. These apps can be owned by the intended web application and third-party (owned by an independent entity). The objective of attackers is to compromise the apps and their intended web application. To achieve this, the attackers need to trick the victim to load their web pages into the apps, and then launch attacks on the target Web View. The attack is depicted in Figure.
3(a). Getting the victim to load attacker's web pages is not very difficult, and it can be done through various means, such as emails, social networks, advertisements.

**Attacks from Malicious Apps**
In this threat model, we assume that an attacker owns a malicious app, designed specifically for a web application, e.g., Facebook. The goal of the attacker is to directly launch attacks on the web application. The attack is depicted in Figure 3(b). These attacks only make sense for third-party apps. To prepare for such attacks, the attacker needs to allure users to use their apps for the intended web application. In addition to manipulating the contents/cookies of the web page, the malicious application can also ask its injected JavaScript code to send out sensitive information from the page.

Besides the powerful interaction mechanism between Android Applications and web pages, WebView also exposes a number of hooks to Android applications, allowing them to intercept events, and potentially change the consequences of events.

![Malicious Web Pages](image1)

(a) Malicious Web Pages

![Malicious Apps](image2)

(b) Malicious Apps

Figure 3: Threat Models

**IV. Services of Android Application**

**Free and Open Source**
Android is an open source platform. The Android Operating system is licensed under GNU General Public License Version 2. The Android framework is distributed under the Apache Software which allows for the distribution of both open and closed source derivations of the source code. The Android SDK and tools are freely available. Developers can download the Android SDK from the Android Web site after agreeing to the terms of the Android Software Development Kit License Agreement.

**Familiar and Inexpensive Development Tools**
The Android application framework includes programming constructs, such as threads and processes and specially designed data structures to encapsulate objects used in mobile applications. Developers can use familiar class libraries, such as java.net and java.text. Unlike some other proprietary platforms that require developer uses charges, expensive compilers, there are no costs to developing Android applications. Android applications are written in a well-respected programming language like Java. Special libraries like SQLite is used for database management and graphics.

**A “Free Market” for Applications**
With Android, developers can write and successfully publish any kind of application they want. Developers can tailor applications to small demographics, instead of just large-scale money-making ones often insisted upon by mobile operators. Android developers are free to choose any kind of model they want. They can develop freeware, shareware, or trial-ware applications and paid applications. Because developers have a variety of application distribution mechanisms to choose from, they can pick the methods that work.

**V. The Lifecycle Hierarchy Of Android application**
The following events follow a basic hierarchy as indicated by indentation. They are all:

- onCreate: Called to set up the Java class for the instance of the app.
- onStart: Technically, it is called to initiate the “visible” lifespan of the app at any time between onStart and onStop.
We can either be onResume or onStop from this state. There is also an event for onRestart, which is called before onStart if the application is transitioning from onStop to onStart instead of being started from scratch.

onResume: Technically, the start of the “foreground” lifespan of the app, but this does not mean that the app is fully visible.

onPause: The app is losing its foregrounded state; this is normally an indication that something is fully covering the app. We can either be onResume or onStop from this state.

onStop: The end of the current visible lifespan of the app, we may transition to on(Re)Start to become visible again, or to onDestroy.

onDestroy: It is called when the Java class is about to be destroyed. Once this function is called, there is only one option for transition (other than being killed):

Fig4: Life cycle of Android Application

VI.Conclusion

Mobile software development has evolved over so many times with increase with time. With the increase in research and practical use towards mobile devices, we hope to not just follow the trend but to supply programmer a more interactive, convenient, efficient way of capturing e-evidences so Android has emerged out as a new mobile development platform, building avoiding past failures of other platforms and brought success in this field. Android was designed to empower the developer to write innovative applications. The platform is open source; with no charges developers can take many benefits over other mobile platforms. The number of Android phones will be continuously increasing as more manufacturers adopt the budding OS. As it stands now, Android sales, by some estimates, will overtake iPhone sales within the next two to three years (Lomas, 2009). While Android is powerful, complex, has multiple firmware implementations and some with manufacturers making custom UIs, the standardization will make mobile forensics simpler in the long run. Indeed, as the market for Android continues to grow.

References