Android Security Issues

Sanjeev Srivatsa, sksrivatsa (at) wustl.edu (A paper written under the guidance of Prof. Raj Jain) Download

Abstract:

Android is the most widely used mobile operating system today. As the world grows more data-driven, people are storing more and more sensitive information on their mobile devices; security is more important than ever. This paper details the possible security issues with Android devices, countermeasures, and real-world examples of past attacks in order to provide users and developers with a better understanding of the risks and limitations surrounding the platform.

Keywords:

Android, Android security, Linux, mobile, smartphone, mobile security, mobile threats, Java

Table of Contents

1. Introduction

This section will provide a brief overview of the Android platform and some of the reasons that mobile device security is so imperative.

1.1 What is Android?

Android is a mobile OS developed by Google that is based on the Linux kernel and written in Java. The OS was initially developed by Android, Inc. (which was backed and later acquired by Google) and released in 2007. The user interface is designed for use with touchscreen devices such as phones or tablets. Androidâ€™s source code is open source and widely available, although most Android devices have a mix of open source and proprietary software [Android(OS)].

1.2 Why is mobile security important?

Nowadays nearly all of the tasks that you could only perform on a computer are achievable on mobile devices as well. This means that more sensitive information will be stored on peoplesâ€™ mobile devices than before. Employees are even able to do work on their mobile devices, so there are more risks for proprietary information leaks as well. Additionally, the number of attempts of cybercrime has been increasing steadily in the recent years. This is even more important for Android because it is the most targeted platform due to its widespread usage and open source properties. The need for security is greater than ever for not only consumers, but large enterprises as well [Farmer].

2. Android Security and Threats

This section will start off with high-level details of the Android security model and risks associated with it and then delve into more detail. It will also cover measures that Android developers need to take in order to build a secure application, as well as things that users should look out for.

2.1 Android Security Model

The Android security model was designed with multiple layers that provide flexibility as well as sufficient protection for all of the consumers of the platform. The flexibility of the platform allows developers of all experience levels to easily work with the SDK to build secure applications. Visibility to users is also very stressed with the Android security model. Users are given information on how applications work and what permissions the applications have on their device.

Security Architecture

The Android operating system's goal is to protect user data, protect system resources, and provide application isolation. To achieve these goals the following security features are provided [SecurityOverview]:

- Robust security at the OS level through the Linux kernel
- Mandatory application sandbox for all applications
- Secure interprocess communication
- Application signing
- Application-defined and user-granted permissions

Figure 1 shows the different components and considerations of the Android software stack. Each part of the stack operates under the assumption that everything below it is properly secured.

The core of the Android security model is the Linux kernel. Linux itself has been around for a very long time and is a very robust kernel now after being constantly improved. It is used in the industry and trusted by many professionals. This kernel provides the Android OS with a user-based permissions model, process isolation, a mechanism for secure IPC, and the ability to remove parts of the kernel.

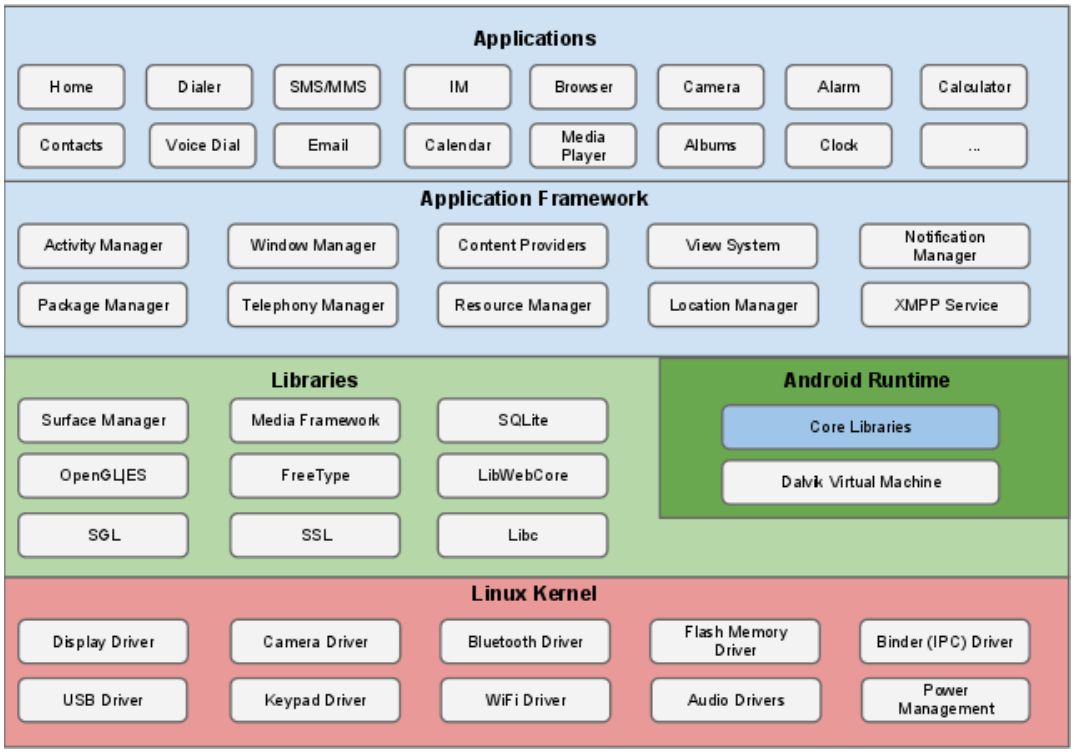

Figure 1: Android software stack

User Data

There are protected APIs in place that that provide applications access to user data. In general, Android devices will store user data over time within the applications that are downloaded on them. Certain applications can choose to share this data, and can use the Android OS permission checks to protect it from other parties.

It should be very clear to users what sort of data an application is trying to access on a device. Always check to make sure that you are only granting an application the appropriate permissions that you want to, because once you grant permission the application can have access to the data at any time. Applications can share data through permissions applied to the secure IPC mechanisms used by the OS [SecurityOverview].

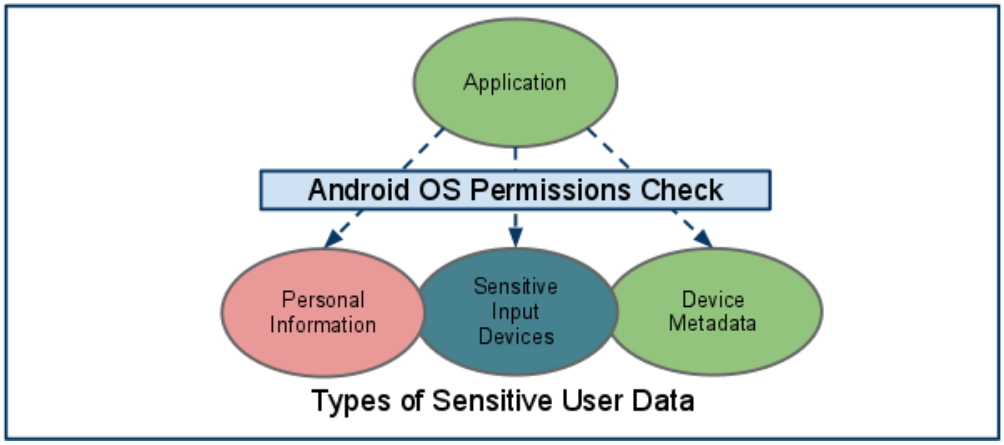

Figure 2: Permissions check required for access to sensitive user data

Applications

Applications extend the functionality of a device by providing a useful service or tool. For the Android platform, applications are developed in Java using the Android SDK. The SDK provides API libraries and tools that are required to build, test, and debug Android applications [Android SDK]. The majority of security concerns are related to applications since they need access to usersâ€™ personal information and other permissions on the device.

There are two sources for applications on any Android device. There are pre-installed applications such as phone, email, calendar, web browser, and contacts. The other would be user-installed applications that can be downloaded from any third party vendor or distributor. Most security risks associated with applications will come from user-installed applications that are actually malware masquerading as an innocent program.

Every application is given its own unique UID when it is installed, and when the application is run it will always use the same UID. This UID is to protect the applicationâ€™s data from other applications and it forces developers to be explicit about sharing data between applications. This effectively sandboxes the applications from one another and keeps malicious applications from infecting other programs on a device.

Permissions are the rights that a specific application has that allow it to perform certain actions on a device. Examples of these actions include taking pictures, using the GPS, reading contacts, or making phone calls. All applications have their permissions available for users to check; you should always make sure you are only installing an application with permissions that you want to give it access to.

2.2 User Concerns

Versions

There are many different versions of the Android OS, and not all devices use the latest version. Android devices are not updated automatically; individual phone manufacturers have the responsibility to push out updates [AndroidGuide]. This means that if there is a threat that is exploitable in an early version, it can still exist for some users of that old version even if it is fixed in a newer version. Users should make sure that their Android version stays up to date especially if there are any security exploits that are found that they are not protected from.

Rooting

Normally, a user does not have full permissions on their Android device, but there is a process called rooting where a user can give himself root privileges on his device. The reasons for doing this include full customization, improved performance, etc., but there are security risks associated with it as well [Six12]. Since your phone now has root access, the security restrictions on your device are bypassed. A rooted device is susceptible to worms, viruses, spyware, and Trojans that can take control of devices without the userâ€™s knowledge [AndroidRooting]. This paper does not go into more detail about security for rooted devices as it is completely different from a normal Android device. Additionally, there are many more concerns that must be addressed when a device is rooted because of the fact that malicious applications gain more control if they are able to take over a rooted device.

Applications

The Google Play store is an application that comes preinstalled on Android devices in general. Users use this application to search for and download other apps onto their device. Developers publish their completed applications to this store where they become available for the public to download them. It is the equivalent of the Apple store for iOS. There are other third party ways to download applications as well that are more widely used in other countries. In order to have a better chance of downloading a trusted applications, users should generally stick the the Google Play Store although it is not guaranteed that there are no malicious applications on this more trusted platform.

Android users have to be careful when installing new applications, as there are malicious applications that pretend to be useful or try to trick people into downloading them. By tricking users into granting them permissions, they can do harmful things like steal user information, destroy personal data, and even make calls. This is why it is very important to look over the permissions of a specific application and checking that the developer is a trusted source before downloading it. These malicious applications are typically discovered and removed by Google when they are found but they are still a real threat to the uninformed user.

Version	Codename	API	Distribution
2.2	Froyo	8	0.6%
2.3.3 - 2.3.7	Gingerbread	10	9.8%
4.0.3 - 4.0.4	Ice Cream Sandwich	15	8.5%
4.1.x	Jelly Bean	16	22.8%
4.2.x		17	20.8%
4.3		18	7.3%
4.4	KitKat	19	30.2%

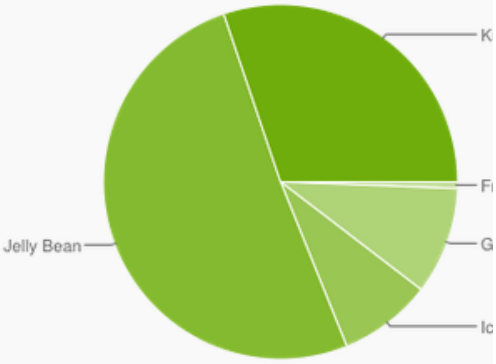

Figure 3: The distribution of the different Android versions.

2.3 Developer Concerns

This section will cover a wide range of precautions that developers should take while building Android applications. It will not provide a full explanation of all of the application components that are discussed in this section. For more information on these, check out the [DevGuide].

Permissions

Permissions are the rights that a specific application has that allow it to perform certain actions on a device. Examples of these actions include taking pictures, using the GPS, reading contacts, or making phone calls (see Figure 4). All applications have their permissions available for users to check; you should always make sure you are only installing an application with permissions that you want to give it access to.

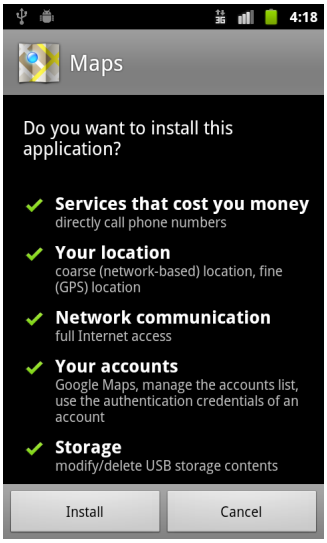

Figure 4: A screenshot of the permissions required for an application.

Developers also should take certain measures when it comes to requiring permissions for their applications. They should only require permissions that are essential to the functionality of the application so that there is a minimal chance of a security breach through the application. Additionally, some users may refuse to install software that requires too many unwanted permissions as well.

Intents

Intents are a mechanism that allow data to be sent between Android processes. They are essentially messages that can cross system security boundaries, and they do not enforce any security policy themselves. Intents can be sent to the other Android application components like Activities, Services, and BroadcastReceivers. It is possible to set up IntentFilters to pick up certain types of Intents and read them with a given priority.

Activities

Activities are single functional components of an application. They can also be used to allow applications to call each other, and reuse each otherâ €™s features. Intents are used to send data to an Activity to initialize or set it up. It is important to not put any vital information that would interest an attacker into an Intent that is going to be used to start an Activity. This is because a malicious program could register an IntentFilter that could pick up on the Intents sent in your application and read the sensitive data inside.

Broadcasts

Broadcasts provide a way for applications and system components to communicate securely. They send messages as Intents, and the system handles the details for delivering them correctly. Broadcasts can explicitly define their target receiver, so there is not the same problem as with sending Intents to Activities, although developers still need to make sure this is done.

There is a special type of broadcast known as a sticky broadcast that has some different properties. They stay around after they have been sent, and any applications with the BROADCAST_STICKY privilege can read any sticky broadcast that is left lying around. So since they can be read by malicious applications that may have this permission, there should not be any sensitive information sent in these type of broadcasts. They should typically only be used to inform other processes about system state.

Services

Services are essentially background processes that can potentially run for a long period of time. They are used for things like running of a game server of playing music. Certain applications require a call to a running Service which could be potentially insecure. Developers must validate the Service they are connecting to and not an unknown program, especially if they are sending it information such as passwords or emails.

SQL Injection

SQL injection is a fairly common exploit that is present in all forms of applications with database calls. For the Android platform, SQL injection can be easily avoided through the use of parameterized queries which explicitly distinguish between the data being sent and the query logic. One caveat would be that if string concatenation is being used in the application (and then these strings are passed along), there could still be the possibility of SQL injection if the data is not being passed directly into a parameterized query.

File Permissions

The Android file system is very similar to any Linux computer, and has similar permissions. Developers should take similar precautions as with any other sort of program and only create files and grant permissions for these files as they intend. Make sure that the permissions are explicitly defined: i.e. log files, and temporary files are fine for writing, but they should probably not be given global read permissions.

3. Real-world Attacks

This section details some real vulnerabilities that have been found in the Android OS.

3.1 Fake ID

All android applications have their own unique identity, and there was a vulnerability that allowed identities to be copied so that one application could impersonate another. This â€œFake IDâ€ breach allowed malicious applications to be recognized as a trusted one by the user without the

user knowing about it. This could potentially allow malicious software to steal user information from a trusted application and even take control of the security mechanisms on a device [FakeID].

The problem arises from the Android package installer not verifying the validity of a chain of certificates. Normally an applicationâ€™s certificate is verified before installing it or updating a version. However, an identity can claim to be issued by another identity, providing a certificate that could potentially be malicious as well as the verified one. The malicious certificate will be ignored since the certificate chaining verification was not done properly, and this will allow that malicious application access to the trusted application.

3.2 SOP Vulnerability

The Same Origin Policy is a security mechanism that is necessary for web application security. The policy allows scripts that originate from the same site to access information from that site found in the DOM or elsewhere, but not to access any information found on pages from different sites. This prevents a web application from reading information that is open in another tab that a user might be using at the same time.

There was a vulnerability with the Android browser AOSP that allowed hackers to bypass the SOP, so they could access sensitive information open in a userâ€™s email tab when the user is on a different site. This was done by sending a malformed JavaScript: URL handler with a null byte, which led to the SOP not being enforced. This has been fixed as the AOSP browser is actually no longer available on the Android devices. Google has released mobile versions of Chrome that do not have this issue so the problem has been resolved [Hoog11].

3.3 GinMaster

Android GinMaster (short for GingerMaster) is a Trojan application family that is primarily distributed through Chinese third party stores that infects Android devices. It was originally named GingerMaster as it attacked Android version 2.3 which was named Gingerbread. The attacks were first found in 2011 and continued for over 2 years. The newer variants of GinMaster were able to avoid detection by most anti-virus software in order to get into devices. Using polymorphic tecchniques, the program would obfuscate class names for infected objects and randomize package names and certificates for applications. Other functionality of this malware was to steal confidential information, gain more permissions on the device, and install applications without user approval [Yu13].

There are many other similar Trojan or other malicious applications and GinMaster is just a single example of this type of problem. Other general categories of Android malware are Rootkit, Trojan spy, Malicious downloader, Click fraudster, Data stealer, and Premium service abuser. Users should always double check an application's permissions and whether or not they are getting the application from a trusted source before downloading.

3.4 Master Key

The Master Key vulnerability was found by researchers at SophosLabs in 2013. Usually, when an application is installed, the Android Package, or APK, verifies that all of the necessary files and certificates for installation correctly check out. If one were to put two files with the same name into the APK (which would normally serve no purpose), the Android device verifies the first file, but then installs and uses the second one. This means that if you were able to borrow another party's package, programs, and other data, you could run and install something that this party has never approved or seen before. In that sense, it works sort of like a master key although it does not actually crack any cryptographic keys in the Android system. This was a flaw with the Android OS as it should not have been doing the check in this way but after it had been identified it was patched.

4. Summary

Android is a very prevalent mobile operating system and will probably be around for many years to come. As mobile devices become more and more advanced, they continue to have more uses and thereby more information stored on them. It is important for consumers and developers to understand the security risks surrounding the platform and what they can do to protect their information. Users need to be aware of what applications they are installing and developers need to take the proper countermeasures to prevent any security breaches or issues.

5. References

[Dubev13]	Abhishek Dubey, Anmol Misra, Android Security: Attacks and Defenses, Taylor and Francis Group, 2013, ISBN: 978-1-4822-0986-0
[Six12]	Jeff Six, Application Security for the Android Platform: Processes, Permissions, and Other Safeguards, Oâ€™Reilly Media, 2012, ISBN: 978-1-449-31507-8
[Rai13]	Pragati Ogai Rai, Android Application Security Essentials, Packt Publishing, 2013, ISBN: 978-1-84951-560-3

[Gunasakera12] Sheran Gunasekera, Android Apps Security, Apress, 2012, ISBN: 978-1430240624

[Hoog11] Andrew Hoog, Android Forensics: Investigation, Analysis and Mobile Security for Google Android, Elsevier Inc., 2011, 978-1-59749-651-3

[Farmer] Ryan Farmer, A Brief Guide to Android Security, Acumin Consulting

[Fedler12] Rafael Fedler, Christian Banse, Christoph Krauss, Volker Fusenig, Android OS Security: Risks and Limitations, 2012, Fraunhofer Research Institution for Applied and Integrated Security

[Mohini13] Tiwari Mohini, Srivastava Ashish Kumar, Gupta Nitesh, Review on Android and Smartphone Security, 2013, Research Journal of Computer and Information Technology Sciences

[Yu13] Rowland Yu, Ginmaster: A Case Study in Android Malware, 2013, Virus Bulletin Conference October 2013

[Enck09] William Enck, Damien Octeau, Patrick McDaniel, Swarat Chaudhuri, A Study of Android Application Security, 2011, Systems and Internet Infrastructure Security Laboratory, Department of Computer Science and Engineering, Pennsylvania State University

[Burns09] Jesse Burns, Mobile Application Security on Android, 2009, Black Hat USA

[Casteel12] Kelly Casteel, Owen Derby, Dennis Wilson, Exploiting common Intent vulnerabilities in Android applications, 2012, Massachusetts Institute of Technology, Computer Systems Security

[Marble14] Marble Labs, Marble Labs Mobile Threat Report, June 2014, 2014, Marble Security

[FakeID] Android Fake ID Vulnerability Lets Malware Impersonate Trusted Applications, Puts All Android Users Since January 2010 At Risk, https://bluebox.com/technical/android-fake-id-vulnerability, A web article detailing an Android security vulnerability known as Fake ID which allows malicious applications to impersonate trusted applications without user knowledge.

[Android(OS)] Android (operating system, http://en.wikipedia.org/wiki/Android_(operating_system), A Wikipedia article with some introductory information about the Android platform.

[DevDashboards] Android Developer Dashboards, https://developer.android.com/about/dashboards/index.html, Documentation on the distribution of Android devices and some other useful information.

[AndroidGuide] The Complete Android Guide for Everyone, http://www.makeuseof.com/tag/download-these-are-the-droids-youre-looking-for-an-android-guide/, A simple guide for users new to Android devices.

[AndroidRooting] Android Rooting Risks, http://www.bullguard.com/bullguard-security-center/mobile-security/mobile-threats/android-rooting-risks.aspx, An article detailing the security risks associated with rooting an Android device.

[AndroidSDK] Get the Android SDK, https://developer.android.com/sdk/index.html?hl=i, The developer page for the Android SDK with some details and a link to download.

[DevGuide] Introduction to Android, https://developer.android.com/guide/index.html, A comprehensive guide with all necessary information for developing on the Android platform.

[SecurityOverview] Android Security Overview, https://source.android.com/devices/tech/security/, A webpage that goes into detail on how the Android security model was built and its functionality.

6. List of Acronyms

OS	Operating system
UID	User identifier
SDK	Software development kit
API	Application programming interface
SQL	Structured query language
SOP	Same Origin Policy
DOM	Document Object Model
AOSP	Android Open Source Platform
URL	Uniform resource locator
APK	Android Package Manager
IPC	Inter-process communication

Last Modified: December 1, 2014
This and other papers on current issues in network security are available online at http://www.cse.wustl.edu/~jain/cse571-14/index.html
Back to Raj Jain's Home Page