Android Malware

A Study of Known and Potential Malware Threats

Troy Vennon, GTC Research Engineer
February 24, 2010

TABLE OF CONTENTS

ABSTRACT

Open source versus closed source. It's a discussion that often leads to heated arguments and one that will likely continue well beyond its usefulness. The discussion began before many of us realized there would need to be terms such as "malware" and the often incorrectly used "hacker". Regardless of what side of the discussion you come down on, the term Android has not helped to lessen the veracity of the debate. Since Google released the first Smartphone operating system that was supposed to be completely open source, the debate between BlackBerry, Windows Mobile, iPhone and Symbian users continues to get louder.

Whether you're new to the Smartphone revolution or are an Android convert from some other platform, there is a reason that you chose Android. Some wanted to break the stuffy business-like feel of the BlackBerry. Others were excited about the possibilities that an operating system built on a Linux kernel with incredible customization capabilities brings. Some wanted something that was friendly or easier to use than the Windows Mobile or their Symbian device. Then there are the ones that just want to be anti-Apple. There are just as many anti-everything-Apple as there are Apple "fanboys" in the world. There are also those that just got a deal from their provider that they couldn't refuse. Regardless of the reason, Andriod's market share is growing.

In October 2009, the world renowned Gartner predicted that the Google-backed Android platform would obtain the 2nd largest market share among Smartphones in the world by 2012. At the time of his prediction, Android's market share was only at 2%. A new report indicates that in just the few short months between October 2009 and February 2010 that market share more than doubled to 5.2%.

The Christmas season saw a few key Android handsets hitting each of the major providers in the U.S., with varying levels of success. Handset insiders have been hinting that nearly every major handset manufacturer is zeroing in on Android as the platform for which they will be developing new, groundbreaking handsets. Motorola and HTC are already well on their way to developing singularly for Android platforms. When so many manufacturers exist to flood the market with Android-ready devices for every provider that was cut out of Apple's and AT&T's exclusive iPhone deal, who can argue that Android is well on its way to taking over the world?

The final questions that Google and Android need to answer have already been asked. Are they ready to play in an enterprise environment and is it secure? And both of the answers are debatable, at best.

PURPOSE

The question of whether Android is ready to play in an enterprise environment is something that SMobile could answer, but we choose not to. What we can answer is whether it is secure or not. However the scope of this discussion will be limited to the threat posed to Android users by malware or malicious applications. To address whether the Android platform presents a secure posture from the threat of a traditional, directed attack will be a discussion for another time and another paper.

The purpose of this discussion will be to look at a few of the known malware threats that exist for Android users, as well as a general look at the malware landscape that could affect users in the future.

DISCUSSION

From the Android Developer's Guide – Security and Permissions "Android is a multi-process system, in which each application (and parts of the system) runs in its own process. Most security between applications and the system is enforced at the process level through standard Linux facilities, such as user and group IDs that are assigned to applications. Additional finer-grained security features are provided through a "permission" mechanism that enforces restrictions on the specific operations that a particular process can perform, and per-URI permissions for granting ad-hoc access to specific pieces of data."

The sticking point in the Android Security Framework is such that, by default, no application has permission to perform any operation that could adversely affect another application, the operating system, or the user. Each application process can and should be considered to be its own "sandbox". However, it is entirely possible for an application to meddle in the actions of another, but it must first explicitly declare the permissions it needs in order to perform the additional functions not provided by the basic "sandbox".

These extraneous permissions can be disallowed based upon certificates that were used to sign the application or by prompting the user. Finally, the permissions that an application would need in order to function outside of its "sandbox" are declared statically within the application and will be displayed to the user, up-front, at installation. Once they are declared at installation and accepted by the user, the permissions will not change. This last piece becomes very important as we address how users can be affected by malware on an Android handset.

So what's the point? The point is that the Android Security Framework includes provisions that prohibit one application from affecting another application, the OS or the user. However, they also know that developers are going to want to develop applications that are useful to users. In order to do so, they thought ahead (at least in this one area) and allow developers to build applications that users want by enforcing rules that their applications must follow to inform the user that the application performs a function, or several, that would cause the application to affect another aspect of the device in some way. Nice of them, isn't it? If users paid attention, yes it's very nice of them.

PERMISSSIONS

Up to this point, we've heard a lot about these permissions that the framework relies upon. Let's take a look at them. A basic Android application has no permissions to manipulate the user experience or the device's data. Some of the features that an application may need access to in order to be useful to the user could be:

- Do things that can cost you money
- Read and write your SMS, email, and other messages
- Gain access to your contacts and calendar stored on the device
- Monitor you physical location
- Access various network features
- Access available Google accounts
- Directly access hardware components of the handset
- Monitor, record and process phone calls
- Access and control aspects of the system
- Access the SD card
- In some cases, access only features needed for application developers

As you read through those activities, think about the different types of applications that could be developed based upon them. What you just read was the list of Android Permissions Groups that exist in the Android Framework. Every activity that an application could perform will fall into one or more of those categories.

Those permissions groups are quite general in their scope. Each of the permissions groups contains a few to a few dozen different permissions that allow aspects of that particular activity to occur. For example, let's say we have an SMS application. In order to access that particular protected feature of the device, our application would need to request permissions from the *permgrouplab_messages* group, which allows an application to read and write your SMS, email, and other messages. However, there are a handful of different capabilities inside of the "messages" group that may or not apply to our particular application. Let's take a quick look at some of the individual permissions that make up the "messages" permission group:

- permlab_receiveSms--receive SMS
- permdesc_receiveSms--Allows application to receive and process SMS messages. Malicious applications may monitor your messages or delete them without showing them to you.
- permlab_receiveMms--receive MMS
- permdesc_receiveMms--Allows application to receive and process MMS messages. Malicious applications may monitor your messages or delete them without showing them to you.
- permlab_sendSms--send SMS messages
- permdesc_sendSms--Allows application to send SMS messages. Malicious applications may cost you money by sending messages without your confirmation.
- permlab_readSms--read SMS or MMS

- permdesc_readSms--Allows application to read SMS messages stored on your phone or SIM card. Malicious applications may read your confidential messages.
- permlab_writeSms--edit SMS or MMS
- permdesc_writeSms--Allows application to write to SMS messages stored on your phone or SIM card. Malicious applications may delete your messages.

As you read through the narratives following each of the "permdesc" fields, something might have jumped out at you. These descriptions already include what a malicious application may be able to do with this type of permission if it is granted. Something else may have jumped out as well. Notice the "permlab_sendsms" permission and its associated description. This particular permission could potentially cost the user money. So, this particular permission may also be a member of the "Do things that cost you money" permissions group.

Listing all of the permissions that are available to an Android application may be out of the scope of this document. However, for the remainder of the study, it's important to understand how permissions play a role in what a developer of a malicious Android application is capable of doing and how they would do it.

WHAT DOES SMOBILE CONSIDER TO BE MALWARE?

The next logical question is what would be considered malware on an Android device? Malware is considered to be malicious software that is designed to infiltrate a computer system without the user's informed consent. An Android Smartphone is definitely a computer, albeit small and usually resides in the user's pocket. It would be a mistake to consider an Android phone anything other than a computer. Heck, the only thing that resembles a phone in today's Smartphone devices is the fact that someone can actually call it. Every other aspect of the device functions in exactly the same manner as a regular old PC or laptop device.

Information security engineers have spent years labeling and categorizing malicious applications. To date we have viruses, Trojans, worms, spyware, rootkits, backdoors, adware and a few other names that could arguable fall under the umbrella of those we just mentioned. What is important to specify is that malware is categorized based upon what it actually does once it has infected a system. SMobile currently categorizes malware much the same way that InfoSec professionals would and labels them as such:

- Virus – A malicious or destructive program that does not have self-reproductive capability
- Trojan - A malicious program that appears to perform a needed function, but allows an attacker to gain unauthorized, remote access to the system
- Worm - a malicious or destructive program that is able to leverage a network or system vulnerability in order to automatically replicate to another system
- Spyware - a malicious application that pretends to be something it is not or actively hides itself from the user while collecting bits of information about the user without the user's knowledge or consent

The first three categories should be self explanatory in terms of why an anti-malware vendor, such as SMobile, takes a proactive approach towards defending Smartphone devices. The

spyware category can get a little tricky when considering applications that are specifically designed to allow individuals of authority (parents or possibly employers) to monitor certain types of use and activity.

For SMobile's purposes of detection and removing spyware from infected devices, we rely on a simple rule of thumb when categorizing a threat as spyware. If the application allows a 3rd party to spy on the activities of the user and the application actively hides itself from the 3rd party at the same time, it is considered to be spyware. In essence, if a user cannot look at the name or a visible icon for the application in the applications list and determine its function while it is monitoring certain activities, SMobile considers it to be spyware and will remove it as such.

ANDROID MALWARE

Android malware development is currently in its infancy. This paper opened with a short discussion about open source versus closed source and about market share. That was by design! The reason that this paper began by discussing Android's market share and the open source nature is because they are responsible for the current threat landscape for Android devices. If you ask any credible security researcher why Microsoft Windows is plagued with malware and Mac or Linux is not, you'll likely get one of two answers…or a combination of the two. The first would be a discussion of closed source OS development not allowing security researchers to assist Microsoft in identifying weaknesses in the platforms before attackers do. This is truly where the open versus closed source discussion began.

The second possible answer you will receive is that Microsoft dominates all of its competitors in the marketplace. Unless it is a directed attack for a single, solitary purpose and goal, it is likely that an attacker is going to develop malicious code that will affect the largest amount of people at once. Latest estimates show that some version of Microsoft Windows currently resides on 91% of all computer systems in the world. Market share drives research from attackers as well as from defenders. This is a rule that cannot be left unconsidered when trying to determine the threat landscape affecting a computing platform. It is also the reason why Android malware development is in its infant stages. But if you trust the predictions (and you definitely should) things will not stay the same for Android…and there is one giant reason why.

By developing an open source operating system that is based upon a kernel that has been repeatedly and extensively dissected and torn apart by thousands and thousands of researchers, Android offers a very secure foundation. By opening the development of the security framework up to industry professionals that have been fighting against age old failures in PC security frameworks, Android was able to come up with the concept of "sandboxing" applications from one another. These two aspects make up the core of the reason why Android may go forward as a secure platform that may never see a game changing direct attack on the operating system.

However, there is one aspect of the open source model that was chosen for Android that may be its undoing. That is the aspect of the Android Market. While Android, iPhone, BlackBerry, Windows Mobile and Symbian devices will always find themselves vulnerable to the same old user-inflicted vulnerabilities as PC's, the Android Market poses a very interesting problem for the future of the movement.

We'll come back to the theoretical threat that exists in the Market shortly. First, we should take a look at some of the threats against Android users that we already know exist. Because SMobile

focuses its attention not just on the consumer but enterprise users as well, we view the mobile spyware trend to be very concerning. Mobile spyware today has the ability to monitor every aspect of a user's communication, activity and physical location without the user knowing that it is occurring. Financial transactions, web surfing habits, every type of messaging, incoming and outgoing call logs, physical location, content downloaded and stored of storage cards and even conversations that occur near the handset can all be monitored by mobile spyware today.

Spyware makes the potential of having sensitive corporate policies or deals, trade secrets, customer information, and all manner of secret data being lost to competitors become a reality.

MOBILESPY

There are two major spyware applications that SMobile has been detecting and removing from Android handsets. The first is MobileSpy, which was released in November 2009. MobileSpy offers the following functionality for Android devices:

- Monitor SMS messages
- View inbound and outbound call details
- GPS location
- View all websites visited from the device
- Web interface to view and manage the captured logs

When it is received, MobileSpy arrives on the device as ms.apk. It must be manually downloaded or transferred to the device. The attacker must have physical access to the device in order to install the software as well. This application should not be considered malware that can automatically arrive and install on the handset. Someone (the attacker) must gain physical access to the device to install it.

None of this would be considered abnormal for any properly identified parental control or monitoring software. The reason that MobileSpy runs into trouble with being identified as spyware is because the application is installed and identified as "Retinax.Android" in the list of installed applications on the device. Additionally, there is no application icon available to the user that indicates "Retinax.Android" or MobileSpy to be a viable application, even though it runs in the background.

While installing, MobileSpy requests the following permissions be granted:

- android.permission.RECEIVE_BOOT_COMPLETED
- android.permission.READ_PHONE_STATE
- android.permission.READ_CONTACTS
- android.permission.READ_LOGS
- android.permission.RECEIVE_SMS
- android.permission.ACCESS_FINE_LOCATION
- android.permission.INJECT_EVENTS
- android.permission.DISABLE_KEYGUARD
- android.permission.PROCESS_OUTGOING_CALLS
- android.permission.INTERNET

- android.permission.READ_SMS
- android.permission.ACCESS_NETWORK_STATE
- android.permission.CHANGE_WIFI_STATE
- android.permission.ACCESS_WIFI_STATE
- android.permission.WRITE_SETTINGS
- android.permission.WRITE_SECURE_SETTINGS

MobileSpy can only be purchased through the Mobile Spy website at a cost of $49.97 per quarter. Once the application is purchased, the attacker is able to setup an online account that he/she would log in to in order to view the contents of the communication that had been monitored by the application. Assuming that a user did not know that "Retinax.Android" was a malicious application, the only way a user would know if their handset was infected with MobileSpy would be to install and run an Anti-Virus/Anti-Spyware application that could detect the existence of MobileSpy.

MOBISTEALTH

The second major spyware application that we'll discuss is MobiStealth. MobiStealth is a much more robust spyware application than MobileSpy, as the cost would indicate. MobiStealth is also not available in the Android Market. MobiStealth comes in three versions with increasing functionality:

MobiStealth Lite:	SMS Logging Call History Logging Browser History Logging Bookmark Logging Appointment/Calendar Logging Contact Details GPS Tracking
MobiStealth Pro:	Recording of Surrounding Picture Logging SMS Logging Call History Logging Browser History Logging Bookmark Logging Appointment/Calendar Logging Contact Details GPS Tracking SIM Change Notification Tracking without GPS Location through SMS
MobiStealth ProX:	Video Logging Recording of Calls Recording of Surrounding Picture Logging SMS Logging

Call History Logging
Browser History Logging
Bookmark Logging
Appointment/Calendar Logging
Contact Details
GPS Tracking
SIM Change Notification
Tracking without GPS
Location through SMS

The pricing for MobiStealth breaks down as follows:

MobiStealth Lite - $49.99/ 3 months
$69.99/ 6 months
$99.99/ 12 months

MobiStealth Pro - $79.99/ 3 months
$119.99/ 6 months
$149.99/ 12 months

MobiStealth Pro-X - $99.99/ 3 months
$149.99/ 6 months
$199.99/12 months

In order for an attacker to install MobiStealth on a target device, the attacker would need to gain physical access to the handset. Once purchased, the application would need to be manually downloaded or transferred to the device and is received as mobistealth.apk. Once the attacker begins the install process the following permissions are requested by the application:

- android.permission.RECEIVE_SMS
- android.permission.READ_SMS
- android.permission.SEND_SMS
- android.permission.WRITE_SMS
- android.permission.READ_CONTACTS
- android.permission.WRITE_CONTACTS
- android.permission.INTERNET
- android.permission.ACCESS_NETWORK_STATE
- android.permission.READ_PHONE_STATE
- android.permission.MODIFY_PHONE_STATE
- android.permission.ACCESS_FINE_LOCATION
- android.permission.ACCESS_COARSE_LOCATION
- android.permission.READ_CALENDAR
- android.permission.WRITE_CALENDAR
- android.permission.RECEIVE_BOOT_COMPLETED
- com.android.browser.permission.READ_HISTORY_BOOKMARKS
- com.android.browser.permission.WRITE_HISTORY_BOOKMARKS

- android.permission.RECORD_AUDIO
- android.permission.PROCESS_OUTGOING_CALLS
- android.permission.WRITE_EXTERNAL_STORAGE
- android.permission.MOUNT_UNMOUNT_FILESYSTEMS
- android.permission.CHANGE_WIFI_STATE
- android.permission.ACCESS_WIFI_STATE
- android.permission.ACCESS_COARSE_UPDATES
- android.permission.WAKE_LOCK

MobiStealth also offers a web interface that can be used to view the contents of the monitored data, which is accessed by logging into the "stealth club account" that is created upon registration of the application. MobiStealth also bills itself as a parental control and monitoring suite, but is considered by SMobile to be spyware because the application is actually installed on the device and listed in the applications list as "EmailClient". After the application has been installed and registered by the attacker, no instance of an application icon is displayed and the tool runs in the background, without the user knowing they are being monitored.

DROID09

Not a lot is known about what this particular application did to Android devices other than that it was some sort of phishing application that targeted banks. The first public reports of a malicious application to infiltrate the Android Market began appearing on tech sites on January 11, 2010. The reports indicated that an Android developer was able to create an online banking application that was named "Droid09" that was able to connect users from various banks to their specific banking institution to view balances, transactions and even transfer funds.

Droid09 would prompt new users to select from a limited list of banks, complete with company logos that they would like to connect with. Once an institution was selected, the application would move to another screen that prompted the user to supply the login information for their online bank account. At this point, it is unclear as to whether this application also asked for the specific bank account and routing number, but in the grand scheme of things, this information is probably unimportant.

For many users that downloaded this application, it was common that an application purporting to provide online banking functionality would ask for these types of credentials. What led to the application being outed as a phishing application was the fact that when the users attempted to actually use the application, all it would do is open a web browser to web portal of the bank that they initially configured. It would not automatically log the users in or even display any sensitive information. They were just presented with the webpage of their online bank.

There are quite a few pieces of information that are unknown about this particular attack and requests to Google to provide the following answers have received no response:

- How many users downloaded the Droid09 application?
- How long was Droid09 available on the Android Market?
- How many users reported unauthorized account activity following a Droid09 application installation

- Who is the individual that developed the Droid09 application and what were his/her intentions
- Did the application actually transfer account credentials to a 3rd party and how?

OPEN SOURCE ANDROID MARKET

Even though there are still a myriad of unanswered questions surround the Droid09 attack of late 2009, we can make some assumptions about the users who trusted the application and the nature of the open Android Market. When Android decided they were going to be the anti-iPhone in every possible way, they were focused on one particular aspect of how Apple operates. As many already know, Apple maintains a very tight grasp over the applications that are approved or disapproved for sale in the App Store. Whether you agree with Apple's approach or not, they do not foster the debate of whether malicious apps will run rampant in their application store. Granted, no approach is perfect. Apple is currently waging a suit against an iPhone game developer who was able to slip a video game past the screening process that was harvesting cell phone numbers and shipping them off to a 3rd party.

But what many security-minded individuals have been asking for a long time is now the question at the top the mind of every enterprise security manager that is considering making an enterprise shift to Android devices, as well as any vaguely informed consumers. What will Android do about regulating malicious applications in the Market? As it stands, anyone can sign and publish an application to the Android Market. Anyone can download an application from that Market, assuming their device is configured with an authentic Google Checkout account.

The idea behind the open Market is that the community will regulate broken or malicious applications. The community will identify applications that are either not performing their intended function, don't work at all or are just plain malicious in nature. But where does that leave the individual consumer who to find one of those applications in the Market and downloads it before the "community" has had a chance to "regulate" it out of the space?

Android has been diplomatic by pointing to the terms of service that developers must follow when building and publishing applications. This response does not go far enough for some. Android also goes above and beyond what its competition currently does in requiring applications to declare their "intent" by prompting users to accept permissions before the application can be installed. One could even go as far as to say that the language Android uses to try to inform users of what the requested permissions can possibly do goes much further than BlackBerry's attempt at informing the user. But what remains evident from the Droid09 spectacle is that Smartphone users are just as gullible as PC users when it comes to installing software.

The purpose of this study is not to answer the questions that Android seems to be unwilling to answer. The purpose is only to discuss the threats that currently exist. As it stands, Android's current threat exists in the realm of malicious applications. Whether they are physically installed by an attacker on the device or whether they are downloaded from the Market by the user themselves.

One thing remains clear, as Android's market share grows so must the awareness of the security professional charged with protecting sensitive information that may be accessed from the device. Android as a Smartphone platform is not going anywhere in the near future. In fact, the

presence and influence is only going to grow. Until these questions are answered, enterprises, government agencies and consumers are going to need to rely on reputable 3rd party software vendors to assist in identifying malicious applications, just as much as PC users have had to.

REFERENCES

http://www.computerworld.com/s/article/9139026/Android_to_grab_No._2_spot_by_2012_says_Gartner

http://www.businessweek.com/the_thread/techbeat/archives/2010/02/android_doubles.html

http://developer.android.com/guide/topics/security/security.html

http://www.mobile-spy.com/

http://www.mobistealth.com/

http://www.macobserver.com/tmo/article/iphone_game_developer_sued_for_collecting_users_cell_numbers/

About SMobile Systems

SMobile Systems, founded in 2002 and headquartered in Columbus, Ohio, is the world leader in providing comprehensive software security solutions for all major mobile device platforms, including BlackBerry, Windows Mobile, Symbian, Palm, iPhone and Android.

In response to the growing demand for mobile device security, SMobile has created a complete mobile security suite including Antivirus, Firewall, AntiSpam, Anti-Theft and Identity Protection, Secure Mobile Banking, and Parental and Enterprise Controls.

SMobile's mission is to enable end-to-end voice and data continuity on wireless networks by providing a range of specialized, leading edge security software and services, all built on a proprietary secure back plane infrastructure.