STANDARDIZING THE SECURITY OF MOBILE APP STORE PLATFORMS

by

Michael Clapsadl

A Capstone Project Submitted to the Faculty of

Utica College

November 2012

in Partial Fulfillment of the Requirements for the Degree

Master of Science in Cyber-Intelligence and Forensics

Abstract

This research evaluates the security of mobile device application platforms, better known as app stores. Nearly 98% of devices use five primary operating systems, and therefore the app stores common to those systems were investigated. This includes the Google Play Store, Amazon App Store for Android, Apple iTunes Store, Research In Motion's Blackberry App World, and the Windows Phone Marketplace by Microsoft. The Cydia store, an open source system for iOS, was also included in this research for security comparison. The goal was to compare existing stores security practices and policies and develop a comprehensive industry standard for use in consumer comparison of app stores.

App store security is important due to an increase in app based malware. New technology is often not built with security in mind and app stores are no exception. What do stores do to prevent malware and data theft and to protect user privacy? In total, twenty factors were established that form an industry standard app store security model. This includes a rating system for consumers to use for comparing stores and better understand security.

This research finds that none of the stores reviewed met even 70% of the necessary security factors. In fact most stores scored less than five on a ten point scale. Implementation of a standard would encourage stores to do more to ensure security is enforced and will provide increased transparency allowing consumers to understand what stores are doing to protect them. A standard is vital to the future of mobile device app stores and stopping the increase in malware. Keywords: Cybersecurity, Christopher Riddell, C3 Security.

Acknowledgements

I would like to thank my family, friends, and colleagues who helped me through the completion of this paper and Masters degree. Many people made sacrifices to allow me the time required to complete this work and I am grateful for their help and patience.

Table of Contents

List of Illustrative Materials

Standardizing the Security of Mobile App Store Platforms

The purpose of this study was to develop a security ranking structure for mobile app stores to aide consumer comparison of app store platform security. Through research, several questions will be answered. What must developers do to prove their apps are secure and legitimate? What do app stores currently do to ensure security? What must be added to current practices in order to form an effective security standard for app stores?

Mobile devices have become ubiquitous in today's society. Smartphones, such as the iPhone and Droid, touch screen tablets, such as the Apple iPad and Kindle Fire, and even smart televisions and media players allow people to be connected to the Internet like never before. As of the last quarter of 2011, 6 billion mobile phone devices were in use globally, 1.2 billion of them with available Internet access, and these numbers are expected to increase (ICT, 2011). These devices on their own, as provided from their respective manufacturers, are powerful and have many built in features, but consumers want and need more.

Consumers desire infinite customizations and the innovation that resulted from this demand is the 'app'. A term coined as shorthand for application, an app allows the device user to add functionality ranging from simple social networking to online banking and shopping. As the popularity of mobile devices and their apps grew, a new requirement became obvious: the need for app repositories or marketplaces. Users needed systems in which available apps were organized and available freely or for sale. To fit this need, companies created what we know colloquially as app stores.

Five primary operating systems make up 97.9% of all mobile devices and each has one or two primary sources for apps. The Google Play Store and Amazon App Store for Android provide apps to the Google Android OS. The Apple iTunes Store is the main source for Apple

iOS device apps. Research In Motion's Blackberry App World is the primary app store for the popular BlackBerry platform and the Windows Phone Marketplace by Microsoft supports the apps of the Windows Mobile OS platform (comScore, 2012). There are also countless minor app stores, like Cydia and GetJar, which users can access depending upon their device. Users have a choice of which store they use to obtain their apps; however, they may not have any way of knowing how those stores implement security.

From these stores many billions of apps have been downloaded, 25 billion from the Apple App store alone, each download being a potential security risk to the device user (Desmarais, 2012). As is often the case with new technology, security is not always considered when trying to develop new systems. The potential lack of security in relation to a popular and widespread technology could become a major problem. Device users currently have no way to determine if the apps they are downloading are safe. They have no way to know what the various app stores are doing to protect them from malicious developers who may make illegitimate or harmful apps.

The threats are numerous and varied. Viruses and malware have already been created that affect the majority of the mobile operating systems, and the problem will only get worse. At the time of this writing, the Symantec Corporation has a record of 526 known threats affecting mobile operating systems (Symantec, 2012). Fake and malicious versions of popular apps have also made their way into major app marketplaces. This includes copies of Angry Birds which were laced with Rufraud malware and was used to defraud 14,000 users (Schwartz, 2012). Apps have also been found in the wild which steal the users contact list data and sometimes use it to spam those contacts or even spread the malware to others (Maslennikov, 2012). Location data is also at risk at times. A form of phishing has also been seen in the world of mobile apps. Some

apps will legitimately act as advertised and be available from legitimate sources; however advertisements or promotions within the app may direct the user to fraudulent information and other apps (Astar, 2011). Malware is a serious enough issue for individuals, but it can be devastating to companies. With more and more businesses allowing employees to bring personal mobile devices into the corporate network, known as bring your own device (BYOD), and often times allowing corporate data to be stored on those devices, the existence of mobile malware could result in severe data breaches. A study by BT Assure found that 39 percent of surveyed enterprises which allowed BYOD experienced a security breach due to their employee's devices (BT, 2012).

The results of a lack in app store security are similar to what is seen in the personal computer field. Malware can cause a loss of privacy, financial losses, breach of confidentiality, and more. One example is the August 2012 Finfish Trojan virus that can steal data from iOS, Android, BlackBerry, Windows Mobile, and Windows operating systems (Spasojevic, 2012). The Android.Opfake malware is an example of malware that sends premium SMS messages, known more commonly as text messages, charging the consumer's cellular account without their permission (Suenaga, 2012). These apps' malicious actions negatively affect society and the economy, and attempts to prevent them should be made by app stores in as many ways as possible.

Research was unable to uncover any defined industry standard of security that app stores should comply with. There is also no oversight or policing of the individual app stores and their practice or policies. They can act as they wish, with as many or as few security precautions as they want, often without transparency to the end user. This leaves consumers in a situation where they do not know if they could be vulnerable or not. A standard, including a ranking system,

would allow consumers to choose their app store of preference based on the security decisions each app store makes.

Literature Review

In order to form an industry standard, it is useful to see what problems currently exist and what is currently done in the industry. In the case of app stores, the problems are primarily malicious apps. Research into what is currently done will be based on the aforementioned app stores of the Google Play Store, Amazon App Store for Android, Apple iTunes Store, Blackberry App World and the Windows Phone Marketplace. The less popular, but well known Cydia store will also be included for comparison, as it works on an open source approach rather than a centralized corporate marketplace.

Malware

Nearly all technology is susceptible to viruses, and mobile devices are no exception. The following examples show some of the malware which has affected mobile devices, what they have done, and how those apps made it into the devices of consumers. The existence of these examples can show weaknesses in app stores and app distribution controls.

AndroidOS.Tapsnake is a malicious app from 2010 that takes advantage of the mobile device's GPS location features to track the user of the app. This is an example of a Trojan virus where an undesired functionality is included in the app, hidden from the user. The hidden GPS tracking functions sends the device's location to a remote server which another individual can then access with a separate app. No other data is gathered nor other functions of the phone affected, but users lose privacy of their locations by simply playing a game. The developer of the app does properly list the GPS tracking functions of the app in the description, but it is not apparent if you are simply using the device (Symantec Security Response, 2010).

Not considered malicious, but also including a hidden function unknown to the app store, was an app named Handy Light. This was a simple "flashlight" app for Apple iOS devices which allowed the user to make the device screen white, or another color, to use as a flashlight. The developer included a hidden function, unlocked only when a user selected the correct color sequence, which allowed the app to share the device's Internet connection. This connection sharing, known as tethering, was usually not allowed by cellular carriers without an additional fee (Engadget, 2010).

Opfake is another Android based app that is deemed malicious by the Symantec Corporation. Its primary purpose is to utilize the mobile device's ability to send SMS, or text, messages. These messages are not free to the consumer and they can be defrauded even if they decline the terms of use in the app. If the user accepts the vaguely written agreement they repeatedly get charged for premium text messages sent by the app for the sole purpose of making the developer money. This scam has taken form of various apps, the functionality of which is often maintained (Suenaga, 2012). It may take the device's owner some time to determine that the premium charges were made from their account and more time to determine what caused them.

Affecting nearly every mobile operating system and even standard Windows based computers is Finfish. Finding its way into the wild in August of 2012, the Finfish Trojan is able to create a security hole, or backdoor, in the operating system and exfiltrate data from the device by posing as a fake update. Information sent to the originator of the malware includes the user's text and multimedia messages, contacts, emails, photos, ID information of the device, and more. Despite its unusual characteristic of operating on many different operating systems Symantec reports that it had a low distribution level (Spasojevic, 2012).

A malicious app can also act much in the same way a worm does, spreading itself. The Find and Call app, classified for iOS and Android devices as Trojan.IphoneOS.Fidall.a and Trojan.AndroidOS.Fidall.a, copied all of the user's address book information to a server and then text messaged those contacts asking them to install the app. The text messages contacts received appeared to be sent from the original device's owner, thus convincing many to install the malicious app because it was recommended by a friend. The only warning consumers could see in advance was negative reviews in the app stores left by other users (Maslennikov, 2012). The servers operated by the app developer could also retain the contact information and use it in future communications to the victims, or the information could be sold to other third parties.

Called Android.Maistealer, the Anaru app is notable not because it steals information from a user's phone, but rather that it is distributed through a fake app store. There is no evidence to show that this app was on the Google Play store, but the app store it could be obtained from used the same graphics and logo to make the user believe that it was the legitimate Google operated marketplace. This same technique was also used in malicious apps packaged as flashlights and battery saver programs; in each case the app did function, but in the background stole personal data. Figure 1 depicts what Japanese users would see when obtaining one of these apps from a fake app store (Hamada, 2012). Figure 2 shows what an app on the legitimate Google Play store looks like (Crider, 2012). Despite the language differences, these look remarkably similar.

Figure 1. A fake Japanese app store

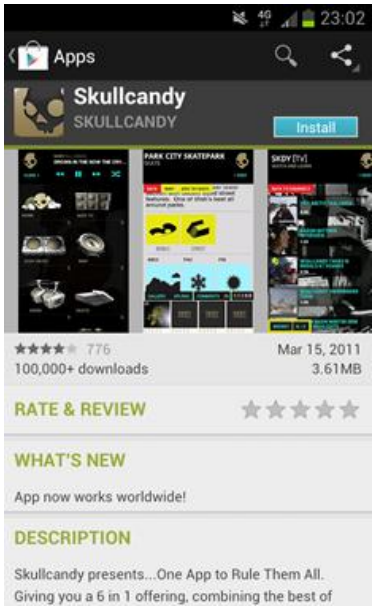

Figure 2. The Google play store

Another way mobile malware is mimicking the typical computer virus is in the uninstallation of the malware, or rather the lack thereof. In the case of Android.SMSzombie.A, the malware prevents the normal administrative functions in the Android operating so the user is unable to remove the software as they normally could with an app. SMSzombie was also covert by using another malicious app to get installed onto the device and not displaying a Cancel

button during initialization. Functionally, this is yet another app which sends premium text messages and defrauds the consumer (McAfee, 2012).

Rufraud is yet another SMS fraud type malware app affecting Android. Unlike other apps the main technique used to get installed is by masquerading as legitimate, generally very popular, apps. Users assume they are the legitimate apps available for free, when really they are malicious copies of those normally for-profit apps. Popular games like Where's My Water, have been copied, the malicious code added, and then posted in app stores for unsuspecting consumers to download. The Rufraud malware itself is not interesting – another simple text message fraud – however its method of tricking users into installing the malware is of significance (Xie, 2011). InformationWeek reports that as many as 27 different infected copy-cat apps made it into the Google Play store (Schwartz, 2012). In addition to the consumer fraud, malware using this method of delivery also violates the copyright of the original app developers.

App Store Policies and Procedures

The individual app stores control security primarily through their policies and procedures. Developers have to accept certain terms and their apps must comply with certain restrictions in order to be allowed into most app stores. Each of the major five app stores and one non-traditional marketplace were researched to find what they do in regards to security.

Apple App Store

The Apple store was one of the first and is currently the largest with over 680,000 apps available (App Store Metrics, 2012). Of the top five app stores, this comprises 30.7% of the market (comScore, 2012). This store is operated by the same corporation that makes the mobile devices themselves - iOS based devices such as the iPhone, iPad and iPod Touch - and is the only store providing those devices with apps. Users have only two ways to use the store. They

must install the Apple iTunes program on their computer, or they can use the Apple App Store mobile app on their device. Apps cannot be installed via websites, email, or other methods without overriding the built in operating system protections. This override process is known as jailbreaking, and results in the user's ability to use other app sources, although they do so at their own risk. The Apple's iOS Security publication states, "iOS does not allow users to install potentially malicious unsigned apps from websites" (iOS Security, 2012, p. 6).

One of Apple's strongest security measures is that the company reviews every app that is submitted by developers to the app store. Unfortunately, this review process is not completely transparent to consumers or app developers. The review itself is not documented publicly; however it can be at least partially inferred from the developer and app submission policies. Apple's policies are also not available to the public. One must apply to be a developer to obtain even basic information concerning the store's security regulations. The following quote from Apple's documentation appears to indicate that the review process varies by app and is not static. "In general the more expensive your app the more thoroughly we will review it" (App Store Review Guidelines, 2012, p. 5).

Developers are required to pay for access to submit apps to the store. The $99 USD annual cost could provide a disincentive for a developer who may want to quickly and inexpensively put malware into an app store. Sandboxing is required of all apps, which prevents them from accessing data on the device to which the app should not have permission (iOS Security, 2012). An obvious but important policy is that Apple forbids apps which include malware or viruses. The documentation does not state how the app store checks for malicious code. Code signing is also required, allowing for some accountability and checks that code has

not been modified. Developers must obtain the signatures from Apple, presumably so the store controls as much of the security process as possible. The store's security guide notes:

> iOS does not allow users to install potentially malicious unsigned apps from websites, or run untrusted code. At runtime, code signature checks of all executable memory pages are made as they are loaded to ensure that an app has not been modified since it was installed or last updated. (iOS Security, 2012)

Requirements also prohibit apps from unauthorized use of location data, namely GPS (App Store Review Guidelines, 2012). Violating the privacy of user's information in any way is banned as well. This could include accessing contact lists, files, or photos on the device without permission (App Store Approval Process, 2012). Hidden features not listed in the description of the app, like the tethering built into Handy Light, are also strictly prohibited (App Store Review Guidelines, 2012). If the app will be provided for consumption in foreign countries, Apple checks it for any export restricted content, like encryption software (App Store Approval Process, 2012).

Beyond the technical requirements, Apple also has user focused security and content rules. Content wise, developers are not allowed to create an app containing adult content, violence, or obscenity. Their content guidelines state "we will reject Apps for any content or behavior that we believe is over the line (App Store Review Guidelines, 2012, p. 1)." This store has also been known to reject apps due to moral or ethical considerations. For example, an app that allowed consumers to avoid police driving while intoxicated checkpoints was removed from the store (Copeland, 2011).

Users may be able to determine the legitimacy of an app by the information listed in the app store and reviews left by others. Apple protects both of these pieces of data in their review

process and policies. First, their review process checks not only the app itself, but all of the metadata, or descriptive information, listed by the developer. If the developer uses inaccurate or misleading wording the app will be rejected from the store until the information is corrected (App Store Approval Process, 2012). Review information is protected in two ways. A policy is in place that disallows any type of ranking manipulation. This prevents the developer from artificially inflating their number of reviews or their ranking, in one to five stars, in order to get on the Best Seller or Top App lists (Moren, 2012). Second, developers have no access to the reviews users post. They are unable to remove or hide negative comments. Even advertisements are carefully controlled by Apple to ensure a good user experience. Developers who choose to include ads in their apps must only use Apple's iAds system. This ensures that users are not lured to a malicious page by an ad and that ads are not abused purely to increase traffic to a website outside of the app (App Store Review Guidelines, 2012).

Google Play Store

Operated by Google, the distributors of the Android operating system, the Play Store is the primary and default source of apps for Android device users. Android comprises 51% of the mobile device market, although the Google Play Store is only one of many which provide apps (comScore, 2012). As of June 2012 the store has over 600,000 apps available to consumers (Warren, 2012). Although Google does operate this store due to their own interest in Android being successful, the operating system allows users to choose if they obtain apps directly from Google or if they wish to use an alternate store. To do this a person would simply select the "Unknown sources" checkbox in their device's settings (Publishing Overview). Access to the store is via a website, http://play.google.com, or by using the Google Play App on Android devices which support it. Similar to jailbreaking, users of many Android devices have the option

to bypass the operating system controls and hack, or 'root', their device. This allows unlimited customization and even installation of alternate versions of the operating system itself (Tyler, 2012). This is a potential security risk, but well outside of the app store's control.

Google conducts an app review process to check that each app submitted meets the store's requirements. Like Apple, this process is not very transparent and the actions they take to review an app are not obvious. In fact, section 7.2 of the Google Developer agreement states "Google does not undertake an obligation to monitor the Products or their content (Developer Distribution Agreement, p. 8)." An automated system known as Bouncer checks apps for common malicious actions, such as sending premium text messages or unauthorized access to data (Lockheimer, 2012). It is unknown if the review includes any human interaction in addition to Bouncer's automated scanning. Despite the unknowns Google's system is slightly more transparent than the Apple App Store because developer information is available to anyone and signing up to be a developer is free.

Malware is very high priority in the documentation for developers. Not only is it prohibited in the developer agreement, and scanned for by Bouncer, Google also has the ability to remove malicious apps remotely if they make it through the marketplace controls and get installed onto devices (Cannings, 2010). The store states the following in the "Dangerous Products" bullet of the Google Play Developer Program Policies document:

> Don't transmit viruses, worms, defects, Trojan horses, malware, or any other items that may introduce security vulnerabilities to or harm user devices, applications, or personal data. We don't allow content that harms, interferes with the operation of, or accesses in an unauthorized manner, networks, servers, or other infrastructure. Apps that collect information (such as the user's location or

> behavior) without the user's knowledge (spyware), malicious scripts and password phishing scams are also prohibited on Google Play, as are applications that cause users to unknowingly download or install applications from sources outside of Google Play. (Developer Program Policies, p. 2)

This section prohibits not only malware, but privacy violations and malicious activities outside of the apps themselves. They also reserve the right to remove any apps and/or developer accounts from the Google system if any breach of their regulations is found (Developer Distribution Agreement).

From the developer point of view, apps are required to use code signing. Unlike with iOS, Google allows the signature keys to come from any source as opposed to creating them for developers. These keys allow authentication between the store and the developer and also allows apps signed with the same keys to trust each other (Preparing for Release). Software creators are also prohibited from including any hidden features in their apps, but only vaguely in wording that requires them to not mislead users about the apps or services being offered (Developer Program Policies). For app internal advertisements Google allows developers to use their own system or another.

On the user side, the Google Play store has a number of features to protect the consumer. Reviews by other users are allowed so one may see other consumer's thoughts prior to purchase. Before installation a list of permissions is also listed; example shown in Figure 3 (SecureIT How-To Guides). This indicates to the user what features of the device the app is requesting to use. For example a game may require Internet access. If this same game has listed that it wants access to photos or contacts it may be a warning sign. Google also prohibits adult content, violence, and obscene materials (Developer Program Policies).

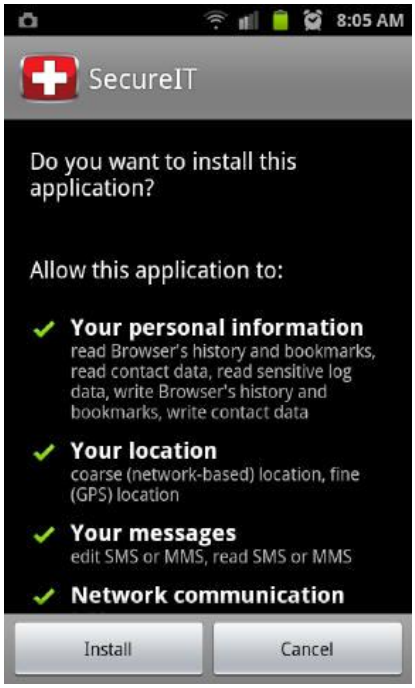

Figure 3. App install permissions

Amazon App Store for Android

Another contender in the world of Android apps is the consumer giant Amazon. Amazon developed the Amazon App Store for Android in competition to Google Play and to provide apps for their own Kindle Fire series of tablet devices. As mentioned previously, Android allows users to select "unknown sources" for their apps. Anyone wishing to use the Amazon store will need to change the setting on their device to allow it. Once done, apps can be obtained from the Amazon website or by using an app on the device (FAQ).

This store is unique in its transparency to consumers in that all their information is published in only two web pages; the App Distribution Agreement and the Developer Frequently Asked Questions page. The files are public if you know the page, but links to the pages are only available with developer access. Developer accounts are free for the first year, $99 annually after that (FAQ). The ease of finding and reading Amazon's policies can be a benefit to their security.

Developers can easily find the policies they must comply with, and if a consumer finds the pages they will see all the relevant information in one place.

As with other stores, Amazon reserves the right to review apps submitted by developers. The frequently asked questions document states, "we may test the apps you submit prior to making them available to verify that each app works as outlined in your product description, does not impair the functionality of the mobile device or put customer data at risk" (FAQ, p. 11). No details are provided that indicate how this review process is conducted. Amazon's terms also discuss privacy concerns by requiring apps to provide notice to the user and obtain consent before using device data in any way.

Clauses also forbid any copyright infringement, trademark or other legal violations an app could conduct. Disallowing of hidden features in apps is not clearly defined, but is only mentioned in a statement regarding fulfilling "In-App Products by delivering additional executable code to the applicable App" (App Distribution Agreement, 2012, p. 8). The documentation does note that Amazon does not review apps for export compliance and that responsibility is the developers. Code signing is required and generally Amazon requires using a certificate provided by them. Developers can request an exception to use their own certificate (FAQ).

The user end environment is also controlled. Apps are not to include adult content or offensive content. Beyond consumers being able to read reviews before they install an app, Amazon also has a unique feature they call Test Drive. A user can run the prospective app within their web browser and try it before they install it on their mobile device (FAQ).

BlackBerry App World

Research In Motion, the creators of BlackBerry mobile phones and the BlackBerry Playbook tablet, have created an app store specifically for their devices. Known as BlackBerry App World it is the sole source of apps for these devices, which make up only 12.3% of the mobile market (comScore, 2012). A relatively small selection of apps is available to users, only 59,806 as of October 13th 2012 (Download Apps). As is typical, access to the store is via their website, http://appworld.blackberry.com/webstore, or by using an app on the mobile device.

Developer access is free, but is not needed to obtain the store's policy documents. This however does not mean there is complete transparency into their app process. Very little information was provided by BlackBerry that indicates their restrictions. Reviews are conducted on each app, and developers are required to submit only the code for the app, not a finished compiled version. Presumably this allows the store to review the base code to look for malware, which is prohibited (Vendor Portal).

In addition to malware, the App World store prohibits a number of other app actions. Being the sole source for apps, it does not allow any app to link to another store, or allow additional apps to be obtained from other sources. Policies also closely protect user data, requiring the app to request permission or provide a warning before collecting location data, collecting personal data, or sending an SMS message or other communications without the user's consent (Vetting Criteria). Apps with any legal issue, like copyright infringement, will also be denied in the app review process (Vendor Portal). Additionally code signing is required of developers and the signatures used are provided by the store (Code Signing Support).

Users have the ability to read reviews posted by other consumers before purchasing. They also are protected from developers who may attempt to fake app information because BlackBerry

carefully checks app metadata to be sure the description of the app is accurate and does not make attempts to inflate rankings in any way. Content is also tightly controlled. Adult content is allowed, except for nudity, but is rated. Hate speech, drug use, violence, and crude content are all disallowed from any app (Vetting Criteria).

Windows Phone Store

The final major app store is the Windows Phone store, by Microsoft. Microsoft has created this store to complement their line of devices running on the Windows Phone operating system. It is a proprietary system, with no competing app stores, consisting of over 88,000 apps available globally (Blandford, 2012). Consumers use an app on their mobile device, or the store's website, www.windowsphone.com/store, to obtain apps. Overall, the Windows Mobile operating systems constitute only 3.9% of the mobile market (comScore, 2012).

The Windows Phone Store, like most of the other app stores, reviews apps submitted by developers, but is not completely open regarding their review process. A number of policies and requirements are available publicly, but others are only accessible by paying $99 annually for a developer account. Their Application Provider Agreement document states, regarding the review process, that Microsoft has the right to outsource their obligations to a third party, which is not seen in other app stores (Application Provider Agreement, 2012). The review, known as certification by Microsoft, can take up to five days, and developers can request that their apps are not reviewed (Certification, 2012). It is unclear under which circumstances an app could avoid certification.

The list of regulations for developers is lengthy and well detailed, although the information is split over a number of different documents. One restriction is that developers must provide a list of permissions needed by the app. This list is included as a manifest in the app and

is used by the operating system to allow the app access to data or phone features (Application Submission Requirements, 2012). Developers must ensure that their apps metadata is correct, which is reviewed upon submission to the store and any time changes are made. Code signing is also required, however Microsoft does this for the developer using signatures they create (Certification, 2012).

Malware and viruses are prohibited in the store's provider agreement as well as their technical certification documentation (Technical Certification Requirements, 2012). If malware does get past the review process, and is found later, Microsoft can remove the offending app from the store and remotely from user's devices. Apps may also be removed from the store, or denied during the certification process, if they violate user privacy or violate laws in any way, if they harm the cellular carrier network, or hidden features exist in the app (Application Provider Agreement, 2012).

Consumers are also protected in a number of ways. Reviews are available for each app which can indicate the experience others had with an app. Microsoft also states that they will not share customer information with developers. If the developer needs information about the user the app will need to request that information (Application Provider Agreement, 2012). Apps also must request user permission before accessing location data (Application Policies, 2012). Adult content, violence, and obscenity are also prohibited under Microsoft's app guidelines (Content Policies, 2012).

Cydia

The Cydia store can be considered a black market of sorts for iOS based apps. Apple mobile devices which have been jailbroken by their users can access this store via an app on their device (Cydia). In a letter to the US Copyright office Jay Freeman, the store's proprietor,

indicates that 12% of iPhones are jailbroken and use Cydia (Freeman, 2012). This store is not associated, or approved, by Apple. This store is unique in that it is not controlled centrally by one organization. Rather it is structured in the same way as the Linux operating system, where software is available from online repositories that developers create. A developer can make their own repository and provide users with instructions to configure their device for use with that repository; an example of which is shown in Figure 4 (Add Us To Cydia).

Figure 4. Adding a repository to Cydia

Alternately users can utilize existing repositories already listed in Cydia's settings like BigBoss or ModMyi. Due to this non centralized system, the policies and regulations between app sources vary considerably.

The BigBoss repository prides itself on saying "Developers, I will host your apps. Apple decline your app? I will host it (Hosting, 2012, p. 1)." There are five total regulations for submitting an app to this store. The developer must own the app content, it cannot have pornography or adult content, the app cannot be listed in other repositories, screenshots of the app are required for the store, and the app has to be digitally signed in a certain way in order to function correctly. This site also claims to have a submitted app available in the store for consumers within 24 hours (Submit Your App, 2012). This could indicate that the store does not review the apps extensively, if at all.

The ModMyi repository is similar. Developers must check a box indicating they own the content of the app and that it will not be listed in another repository (Developer Portal, 2012). Nothing is listed regarding viruses, hidden content, misleading descriptions, or other possible security concerns. Many other repositories like these exist and are unregulated. In some cases of malware outbreaks, such as Anaru, these stores are likely responsible (Hamada, 2012).

Discussion of Findings

In analysis of the available literature, a number of security features and consumer protections were identified in existing app stores. No one store contained all of the available security features found, so there are areas for all of these stores to improve. In the following pages each of the important features will be listed, including malware that would be prevented if the feature was in place as part of a security standard. Additionally, attempts will be made to determine if there are security features not currently in use by any store which should be implemented.

Possibly the most important factor for security is that the app stores perform reviews of the apps submitted by developers. Without this review many of the subsequent security factors will fail to be implemented. The review process, though it may vary by store, allows for quality control and ensures that the developers complied with the store's policies and regulations. Microsoft is unique in their review process in that they outsource it to a third party. No information was provided to indicate what this third party is, or why Microsoft does not do this work themselves. It is expected that this review process discovers and prevents at least a portion of malware submitted by developers. Reviews may also require tests of uninstalling of the app. Such a test would discover that an app infected with the SMSzombie malware could not be removed as a normal app could.

None of the current stores have put into place controls which allow consumers to know the store is legitimate. In situations where users choose to use a store other than the operating system default, they may not be able to properly identify which sources they are using. Having ways to identify the system would act as anti-phishing and may reduce the occurrences of fake or look-a-like app stores as seen in Figure 1. One option would be to adopt 'sitekeys' like many banks do for online commerce. Sitekeys work by showing a user an image that they choose when creating their user account. If a store shows an incorrect image, or none at all, the consumer knows they are not visiting the legitimate store. Implementing this security measure may aid in preventing phishing attacks.

Prohibiting malware is often a strict app store policy and also a major portion of the review process. Since the review process is usually proprietary, and in many cases not documented, consumers are dependent upon the store to perform their due diligence in protecting devices from being infected with malware. Optimally, the stores would be more transparent in their review process, although it is possible that publicly documenting this information would better allow malicious developers to bypass the security controls put in place to find malware.

Hidden features are not allowed by some stores. At times, like in the case of Handy Light on iOS, a hidden feature may not be malicious and actually benefit the consumer. More often the hidden features or code are of malicious intent and the user may not even know the hidden features are operating in the background. Many times, such as with the Tapsnake app, these hidden features are malware.

Most mobile devices, unless they are jailbroken or rooted, initialize apps in a virtual sandbox which controls the access apps have to data. The stores generally ensure that apps submitted to them do not violate the rules of this sandbox and gain unauthorized access to user

data or location information. Enforcing this strictly would have prevented the Finfish malware from stealing data.

Code signing is required by most platforms. This may or may not be implemented in a way that allows apps to be attributed back to the original developer. It varies if the store provides the digital signatures needed to sign the code or if the developer does. The store can better ensure the signatures are correct and legitimate. If developers can provide their own signatures or keys, they could obtain them illegally or from a source which does not confirm the developer's identity.

Some stores only permit consumers to install apps via an official website, computer software, or mobile app while others are more open. Allowing installations by SMS or email provides an avenue for phishing attempts and accidental install of malware. Implementing a control to prevent SMS and email based installs would stop malware like Find and Call that attempts to spread by text messaging itself to all contacts on the infected device.

In the store, before purchase or installation of an app, users may be able to see what permissions the app requires. If, for example, a chess game app lists that it needs access to contacts, the user may identify that access as a red flag and look for a different app. Apple does not provide this to customers at this time, but most Android app stores do. This simple information may allow consumers to avoid malware such as Rufraud, which requires access to send text messages.

Some app markets have a policy against harming the network. Generally speaking this refers to the cellular carrier network but could also include WiFi connections, Bluetooth, or future types of connectivity. This policy could prevent excessive use of bandwidth, which may

cost the consumer a large sum of money. It could also discourage developers from making apps that scan the network looking for, or exploiting, vulnerabilities.

The cost of an app store developer account may influence the amount of malware in a given store. Stores that allow joining for free may see more malicious apps submitted because there is a lower cost of entry. Stores which require a fee - those that do are typically $99 - pose a financial risk to those developers who want to get malware posted in an app store. Developer accounts currently do not appear to involve a background check or some method to confirm the identity of the developer. It is possible for a new developer to create accounts created with false credentials which cannot be tracked back to an individual. Improving accountability by doing this could reduce the number of developers who post malicious data because they would be held responsible. This is especially true of malware which may violate the law.

Stores sometimes check the accuracy of app metadata to ensure that the developer has correctly listed and described the features and functionality of the app. This review may also include checking that any support information, such as the developer's website or email address, is valid. Apps not described correctly or that provide fake support information are more likely to contain malware and/or mislead consumers.

Consumers are able to post reviews and ratings in the app stores and read the reviews and ratings others have provided. Reviews are typically a short written commentary, while ratings are usually a one to five star system. These create a form of checks and balances. Users who have had bad experiences with an app can post a review regarding their experience thus protecting others in the future. In some cases of malicious apps, such as Find and Call in the Google Play store, a major indication of a problem is user posted reviews.

Policies against ranking manipulation are in place in some app stores. The review system checking for ranking manipulation protects the store and ensures the developer cannot tamper with the information. The ability to delete, change, or tamper with the results is prohibited. The store may also look for patterns that may indicate inflation of positive ratings to ensure accurate information. This will not directly prevent malicious apps, but it would help prevent a malicious app from fraudulently appearing on a "Top App" or "Popular App" list leading to additional installations and spreading of the malware.

It has been seen that some malware has been installed when users select an advertisement in the app which then sent the user to a malicious site. Stores operating their own app systems and require all developers to use that system tend to be more secure. This allows the store to select advertisers and review the ads for legitimacy before they appear on user's devices.

App stores may have a policy indicating they will not share user info with app developers. This protects user's identities by keeping their store account information private. Information could include their full name, username, address, phone number, credit card information, and more. This policy also benefits the store because more people will do business with them if they know their personally identifiable information is safe.

A test drive feature allows users can try an app before they buy it. Currently the Amazon App Store for Android is the only store known to have this capability. Not only does it allow users to preview an app that may be malicious or not work properly, but it could also save them money if they decide an app just is not what they wanted before they purchase it.

With apps being sold worldwide many laws apply. The most common legality to cover is that the author of the app owns all of the content of that app and is not infringing on copyrights or trademarks. If an app is to be sold in other countries the app must meet U.S. export laws as

well as laws in the target destination. Some developers perform this step on their apps, but they may not be well versed on the applicable laws. App stores also sometimes check for legal compliance; however, this may be restrictively time consuming for the app store to do. The compromise would be to combine the knowledge of the store and the time of the developer. The app store's legal team can create documentation relating to common laws and the developer can ensure compliance.

A few stores have the ability to remotely uninstall an app. Remote uninstallation allows stores to protect consumers in the situation where malicious software has gotten past all other protections. Devices may be infected with botnets or Trojan viruses that the user is unaware of or does not know how to remove. The remote access allows the store to remove all infections from all devices very quickly. Currently only Microsoft and Google claim that they have this capability.

A centrally controlled store differs from an open source app store. Open source stores are not managed by a specific entity. Cydia is an example of a store which does not have a central control structure and allows any developer to add their apps with no consistent rules. The stores operated by a central company have more security policies and restrictions.

In the end none of these factors help users decide which app store to use if they do not know about them. Stores that are transparent in their policies better educate the end user. When information concerning the stores policies are all on one page, and accessible from the user's account when they login, it is easier for people to find that information. Currently, for all stores researched, this information is spread through numerous web pages and documents. Consumers need to be educated in what they can do to stay safe, like checking permissions and reviews

before buying, and what the app store does to protect them. It is also important that the end user understands the risks of jailbreaking, rooting, or allowing unknown sources on their device.

Results

From analysis of current mobile app store practices and security needs, a total of twenty factors have been found that play a part in app store security. The combination of factors used by app stores can form an industry standard and rating system. These are listed by short description in Table 1.

Table 1

List of Security Factors

App stores must conduct a review of submitted apps, based on other factors.
Consumers must have an ability to identify legitimate app stores against fakes.
Policy and review actions that prohibit apps with any form of malicious activity or intent.
Hidden features must not be allowed and must be part of the app review process.
Apps must be reviewed to comply with operating system sandbox if applicable.
Apps should only be installable from app store specific web sites or apps and not email or SMS.
All permissions needed by an app should be listed and available to consumers before purchase.
Policy and review actions should be in place to prevent apps that could harm any network.
Developers should be required to pay for access and/or submit to background check.
Policy and review actions must ensure that app descriptions and other metadata is correct.
Consumers must be able to review/rate their apps & see reviews/ratings of apps before purchase.
Stores must enact policies preventing manipulation or falsification of ratings or reviews.
Stores should operate their own advertising systems and review ads for malicious intent.
Stores must keep consumers account information private, including preventing developer access.
It is recommended that users can test apps before they purchase.
Stores must ensure that developers comply with all applicable laws.
Stores should have the ability to remotely remove apps which are found to be harmful.

Stores which are centrally controlled are preferred over open access stores.
Stores must be transparent in their policies and actions to the best of their ability.
Stores must educate consumers regarding security risks and best practices.

To make this standard easy to understand for the average mobile device consumer, each factor is being kept to just one sentence and is worded as clearly as possible. Any store wishing to advertise that they comply with this standard would need to clearly list the complete standard as it is in Table 1, and provide links to more detailed information concerning how they comply with each factor. The store would also have to clearly list their overall score. This would allow consumers to compare stores on an even playing field and choose which store they wish to do business with.

Since security is often considered as strong as the weakest link, each factor in this standard is weighted the same, each worth half of one point. A store completely complying with the standard would receive a score of ten out of ten. This however many not be ideal to an app store because of their values, structure, or abilities. Due to this, the standard allows for stores to make their own decisions and choose in which areas they wish to comply. The stores will have a lower overall rating, but then consumers can choose if the security standard rating is more important than other factors or features of the store.

Enforcement of this app store security standard could be handled in a few ways. The first option would be for stores to adopt the standard of their own volition and police themselves. This would not necessarily be reliable as the honor system could be violated, and consumers could not guarantee that stores were compliant. The second option would be for a third party to control the standard and certify stores which claim to comply. This is currently common in the computer industry with services like VeriSign providing security verification for pages they have deemed

secure. For this purpose the company of C3 Security is being used as an example. C3 Security would own the standard, maintain it over time as changes are needed, and uphold it. The company would provide a non-biased review of app stores for consumer protection. Store owners like Apple and Google would pay a fee to this company to review their store's compliance to the C3 app store security standard. Very small or new app stores could apply for a fee waiver or deferment so there is less financial detriment required to be approved for the standard. In return for the licensing costs, C3 would allow the store to use a security logo in their consumer advertisements, apps, software, websites, etc. This common logo, which would appear consistently across all certified app stores, would allow consumers to know that the store was reviewed for security and would display the score that the store received. Example logos are shown in Figure 5.

Figure 5. C3 security logos

A variety of sizes and layouts would be available for different uses. Larger versions may be used on a website while smaller versions of the logo, like those shown on the right side of the figure, would be used on a mobile app or in software where space is limited. Consumers could also click or tap on the logo anywhere it appears digitally to be taken to the C3 website. The website would host the certification results of each store so a given store could not post false

information and fraudulently claim they comply. Additionally, consumers could compare the given store to others on the C3 site to essentially comparison shop.

Under this standard, each of the researched app stores were evaluated to see how they currently comply and rate. This is indicated in Table 2, with each factor described in only a few words for readability. Stores compliant with a factor are marked with a ” •” and the resulting score. Each factor counts for 1/2 point for a total possible score of ten.

Table 2

App Store Security Ratings

	Apple	Google	Amazon	BlackBerry	Microsoft	Cydia
Store Reviews	•	•	•	•		•
Legitimate Store						
Malware Prevention	•	•	•	•	•	
Hidden Features	•	•			•	
Sandboxing Enforced	•	•	•	•	•	
Install Restrictions	•			•		
Permissions Listed		•	•			
Network Protection		•			•	
Developer Access	•			•	•	
App Information	•			•		
User Reviews	•	•	•	•	•	•
Ranking Manipulation	•			•		
Advertising Controls	•					
User Privacy	•	•		•	•	
Test Drive Ability			•			
Legal Compliance	•			•		
Remote Removal		•			•	
Central Control	•	•	•	•	•	
Transparency			•			
Education of Users						
Resulting Score:	6.5	5.0	4.0	5.5	4.5	1.0

Note that Microsoft was not given credit for reviewing apps since there is an option for developers to bypass the review process. Also of importance is the fact that none of the stores

currently meet the need to educate users regarding security risks nor do they provide a way for users to know for certain that they are using the legitimate store and not a fake. Transparency, legal compliance, test drive ability, and advertising restrictions were also not existent or effective in most cases. The fact that the mean score for these six stores is 4.4 out of 10 confirms the industry is in need of improvement.

Conclusion

The current societal move to an always connected mobile lifestyle has caused the smartphone to become a commonplace device. As such, criminals have begun to take advantage of weak security and uneducated consumers. Malware can steal phone data, track the user's location, and charge the user for premium text messages that were not authorized. So what is being done to alleviate this problem?

Five major app stores, covering four different mobile operating systems, were evaluated for their current security practices. These ranged from review of developer's apps and policies against malware, to remote app removal and user reviews. In total, twenty factors were determined to play a part in app store security. None of the stores checked met even 70% of these factors, with most falling below 50%. This makes it obvious that app stores need to do more. Adopting an industry standard would allow stores to be compared to each other on an even level and allow consumers to shop around for the most secure app stores.

It is recommended that a company, such as C3 Security, maintain and uphold the security standard. A company such as this would provide non-biased security testing of app stores and provide ratings that consumers could trust. Ratings would allow the public to have the knowledge they need to be responsible mobile device users. Existence of a standard also can lead to peer pressure which would encourage the app stores to comply. They may not have any reason

to enhance their current practices otherwise. Even if app stores never achieve implementation of all twenty security features, having an industry standard allows people to know how they are protected by the stores they do business with and then decide what additional steps they may need to do on their own to fill in the gaps.

Additional Research

Much is still unknown regarding the security of mobile devices, apps, and app stores. The topics and questions below are just a few areas related to mobile app stores that need further investigation and research.

- Developer coding practices. The app stores do not currently require any specific code checks, such as measures that ensure SQL injection or Cross-Site Scripting are prevented.
- Security in the purchase process. Stores do not currently document for the public how their apps and websites are secured in the purchase process. Could man-in-the-middle attacks gather transaction information?
- Security in the app store client or website. It is unknown what stores do to keep hackers out of their systems. Could the website be vulnerable to leaking information to criminals?
- Third party app review sites or programs lending credibility to apps before purchase. Like the proposed company of C3 Security, are there currently third party sites or software which help consumers check apps before they purchase them?
- Jailbreaking. Is jailbreaking or rooting of devices gaining popularity? Should manufacturers continue their attempts to thwart the process or should they instead try to educate consumers of the risks? Why is it that U.S. copyright law allows jailbreaking of smart phones but not tablets, even if they are using the same operating system?

- Anti-malware apps. Are antivirus and anti-malware applications for mobile devices actually working? For what platforms are this software available, and if some do not allow this, why? How many consumers have installed such software?
- What steps, or best practices, should mobile device users perform to ensure security? What can users to do add to the security the app stores already have in place? This could include researching apps before installing, carefully looking at permissions apps are requesting, installing antivirus software, and much more.

References

App Distribution Agreement. (2012, October 3). Retrieved from Amazon Developer: https://developer.amazon.com/help/da.html

App Store Approval Process. (2012, August 9). Retrieved August 9, 2012, from Apple Developer: https://developer.apple.com/appstore/resources/approval/index.html

App Store Metrics. (2012, October 1). Retrieved from 148Apps: http://148apps.biz/app-store-metrics/?mpage=appcount

App Store Review Guidelines. (2012, August 9). Retrieved from Apple Developer: https://developer.apple.com/appstore/resources/approval/guidelines.html

Application Policies. (2012, July 26). Retrieved from Windows Phone: http://msdn.microsoft.com/en-us/library/windowsphone/develop/hh184841(v=vs.92)

Application Provider Agreement. (2012, August). Retrieved from Windows Phone: http://cmsresources.windowsphone.com/devcenter/en-us/legal/Windows-Phone-Marketplace-Application-Provider-Agreement.pdf

Application Submission Requirements. (2012, July 26). Retrieved from Windows Phone: http://msdn.microsoft.com/en-us/library/windowsphone/develop/hh184844(v=vs.92)

Certification. (2012, June 19). Retrieved from Windows Phone: http://msdn.microsoft.com/en-us/library/windowsphone/develop/hh202928(v=vs.92).aspx

Content Policies. (2012, July 26). Retrieved from Windows Phone: http://msdn.microsoft.com/en-us/library/windowsphone/develop/hh184842(v=vs.92)

Developer Portal. (2012, October 16). Retrieved from ModMyi: http://modmyi.com/mmi/

Hosting. (2012, August 11). Retrieved October 16, 2012, from The Big Boss: http://thebigboss.org/hosting-repository-cydia

iOS Security. (2012, May). Retrieved from Apple: http://images.apple.com/ipad/business/docs/iOS_Security_May12.pdf

Submit Your App. (2012, March 1). Retrieved October 16, 2012, from The Big Boss: http://thebigboss.org/hosting-repository-cydia/submit-your-app

Technical Certification Requirements. (2012, July 26). Retrieved from Windows Phone: http://msdn.microsoft.com/en-us/library/windowsphone/develop/hh184840(v=vs.92).aspx

Add Us To Cydia. (n.d.). Retrieved October 16, 2012, from iHacksRepo: http://blog.ihacksrepo.com/add-us-to-cydia/

Astar, I. (2011, July 20). *Android Threat Trend Shows That Criminals are Thinking Outside the Box*. Retrieved September 10, 2012, from Symantec: http://www.symantec.com/connect/blogs/android-threat-trend-shows-criminals-are-thinking-outside-box

Blandford, R. (2012, July 5). *100,000 apps published to Windows Phone Marketplace*. Retrieved October 16, 2012, from All About Windows Phone: http://allaboutwindowsphone.com/news/item/14960_100000_apps_published_to_Windo.php

BT. (2012, May 16). *BYOD gives competitive advantage, say IT managers* . Retrieved September 10, 2012, from BT: https://www.btplc.com/news/Articles/Showarticle.cfm?ArticleID=741139D3-592C-426E-9904-EB4540663C19

Cannings, R. (2010, June 23). *Exercising Our Remote Application Removal Feature*. Retrieved from Android Developers Blog: http://android-developers.blogspot.com/2010/06/exercising-our-remote-application.html

Code Signing Support. (n.d.). Retrieved October 13, 2012, from BlackBerry: https://developer.blackberry.com/CodeSigningHelp/codesignhelp.html

comScore. (2012, May 1). *comScore*. Retrieved August 29, 2012, from comScore Reports March 2012 U.S. Mobile Subscriber Market Share: http://www.comscore.com/Press_Events/Press_Releases/2012/5/comScore_Reports_March_2012_U.S._Mobile_Subscriber_Market_Share

Copeland, L. (2011). Apple to stop accepting DUI checkpoint apps. *USA Today*, 03a.

Crider, M. (2012, April 13). *Spyware found in Japanese Google Play Store*. Retrieved September 29, 2012, from Android Community: http://androidcommunity.com/spyware-found-in-japanese-google-play-store-20120413/

Cydia. (n.d.). Retrieved October 16, 2012, from Cydia: http://cydia.saurik.com/

Desmarais, C. (2012). Apple App Store Hits 25 Billion Downloads. *PC World*, 26.

Developer Distribution Agreement. (n.d.). Retrieved October 8, 2012, from Android Developer: http://www.android.com/us/developer-distribution-agreement.html

Developer Program Policies. (n.d.). Retrieved October 9, 2012, from Google Play: https://play.google.com/about/developer-content-policy.html

Download Apps. (n.d.). Retrieved October 13, 2012, from BlackBerry App World: https://appworld.blackberry.com/webstore/product/1/?page=534&recordsPerPage=100&lang=en#licenseRadio

Engadget. (2010). Handy Light for iPhone's dirty little secret: tethering. *Engadget*. Retrieved from http://search.proquest.com/docview/615019132?accountid=28902

FAQ. (n.d.). Retrieved October 11, 2012, from Amazon Developer: https://developer.amazon.com/help/faq.html

Freeman, J. (2012, February 24). *Comments on Classes of Works*. Retrieved from U.S. Copyright Office: http://www.copyright.gov/1201/2012/comments/Jay_Freeman.pdf

Hamada, J. (2012, September 7). *Anaru Malware Now Live and Ready to Steal*. Retrieved September 29, 2012, from Symantec: http://www.symantec.com/connect/blogs/anaru-malware-now-live-and-ready-steal

ICT. (2011). *ICT Facts and Figures*. Retrieved August 28, 2012, from International Telecommunication Union: http://www.itu.int/ITU-D/ict/facts/2011/material/ICTFactsFigures2011.pdf

Lockheimer, H. (2012, February 2). *Android and Security*. Retrieved from Google Mobile Blog: http://googlemobile.blogspot.com/2012/02/android-and-security.html

Maslennikov, D. (2012, July 5). *Find and Call: Leak and Spam*. Retrieved September 10, 2012, from SecureList: https://www.securelist.com/en/blog/208193641/Find_and_Call_Leak_and_Spam

McAfee. (2012, August 20). *Android/SMSzombie.A*. Retrieved October 2, 2012, from McAfee: http://home.mcafee.com/virusinfo/virusprofile.aspx?key=1428711

Moren, D. (2012). Apple Clamps Down on App Store Ranking Manipulation. *Macworld*, 28.

Preparing for Release. (n.d.). Retrieved October 9, 2012, from Android Developer: https://developer.android.com/tools/publishing/preparing.html

Publishing Overview. (n.d.). Retrieved October 8, 2012, from Android Developer: https://developer.android.com/tools/publishing/publishing_overview.html

Schwartz, M. J. (2012, May 25). *Angry Birds Malware Sparks $78,000 Fine*. Retrieved September 10, 2012, from Information Week: https://www.informationweek.com/security/mobile/angry-birds-malware-sparks-78000-fine/240000966

SecureIT How-To Guides. (n.d.). Retrieved October 9, 2012, from Security Coverage: http://www.securitycoverage.com/support/secureit/howto.php?id=30

Spasojevic, B. (2012, August 30). *IOS.Finfish*. Retrieved September 22, 2012, from Symantec: http://www.symantec.com/security_response/writeup.jsp?docid=2012-083015-4511-99&inid=us_sr_carousel

Suenaga, M. (2012, May). *Android.Opfake In-Depth*. Retrieved September 22, 2012, from Symantec: http://www.symantec.com/content/en/us/enterprise/media/security_response/whitepapers/android_opfake_in_depth.pdf

Symantec. (2012, September 5). *A-Z Listing of Threats & Risks*. Retrieved September 5, 2012, from Symantec Corporation: https://www.symantec.com/security_response/landing/azlisting.jsp

Symantec Security Response. (2010, August 19). *AndroidOS.Tapsnake: Watching Your Every Move*. Retrieved September 27, 2012, from Symantec: http://www.symantec.com/connect/blogs/androidostapsnake-watching-your-every-move

Tyler, J. (2012). XDA Developers' Android Hacker's Toolkit. John Wiley & Sons.

Vendor Portal. (n.d.). Retrieved October 13, 2012, from BlackBerry: https://appworld.blackberry.com/isvportal/guidelines.do;jsessionid=1FC4C68594909917227AE7403C287E08

Vetting Criteria. (n.d.). Retrieved October 13, 2012, from BlackBerry: https://appworld.blackberry.com/isvportal/downloadAWVettingCriteriaDoc.do

Warren, C. (2012, June 27). *With Jelly Bean, Google Play Embraces Its Inner iTunes*. Retrieved from Mashable: http://mashable.com/2012/06/27/google-play-io-jelly-bean/

Xie, X. J. (2011, December 13). *Android.Rufraud*. Retrieved October 4, 2012, from Symantec: https://www.symantec.com/security_response/writeup.jsp?docid=2011-121306-2304-99