TASAM - Towards the smart devices App-Stores Applications security Management related best practices

Zafar Kazmi & Toni Felguera
MobSec Research Group
Barcelona Digital Technology Centre
Barcelona, Spain
{zkazmi, afelguera} @bdigital.org

Jorge Aguila Vila & Mario Maawad Marcos
CSIRT
La Caixa
Barcelona, Spain
{mmaawad, jorge.aguila} @lacaixa.es

Abstract — **Mobile phone applications (Apps) including the financial services related mobile Apps such as the banking Apps, are becoming the wave of the future and these must be developed, managed and monitored with a great consideration, in order to ensure the Apps security. It is therefore, a huge responsibility of different Apps publishers and the application stores (App-Stores) to develop and maintain an up-to-date Apps delivery security mechanism. This paper aims to present a proposal on mobile application stores Apps management security related good practices, by reviewing the current security measures deployed by different major mobile application stores and by developing the Apps management best practices, based on the finding of the review.**

Keywords-mobile applications management; applications security; appstores applications delivery management best practices; mobile device security; smartphone security

I. INTRODUCTION

Mobile devices, such as the smartphones and tablets are playing an increasingly important role in how people communicate, socialize, and carry out other important daily tasks including accessing their banking activities. Newer network technologies as well as the mobile devices will progressively increase security aspects that would contribute to the wider range of mobile applications and services.

Mobile applications stores (App-Stores) are no exception and the recent popularity of different mobile App-Stores such as the Google's Android Market, and Apple's App-Store, is a clear evidence to support the above statement.

According to an article by Minda Zetlin (Zetlin, 2011), a number of Android users, in the year 2010, downloaded different mobile banking applications (Apps) from the Google's Android Market at a cost of $1.50 each. The Apps enabled the users to connect with about 40 major banks, including the Bank of America and the Wells Fargo of the United States. It later appeared that the banks did not upload those Apps and that those Apps were in fact submitted by some unknown fraudsters to different App-Stores, seeking to only make $1.50 from each download. Needless to mention the potential threat of the fraudsters being able to steal the users' banking login credentials, as a result of the users downloading those malicious banking Apps. It was also the reason why many banks asked their customers to actually have their mobile service provider remove those malicious Apps from their mobile devices (Zetlin, 2011).

Another work (Goodin, 2011) stated that the Google's Android Market had to remove at least a dozen smartphone games out of the Android Market after discovering they contained secret code that was sending text messages (SMS) to a premium number and the users having to bear the high costs of those text messages (Goodin, 2011).

Moving on, the recent evolution within the mobile banking ecosystem clearly indicates that the access to banking through mobile devices will move from browsers to mobile Apps. Moreover, a great part of the security will depend on how secure is the actual App-Store as that is where a user would download their banking App from, in order to access their mobile banking and other financial activities. It is therefore vital to address these growing malicious Apps threats and one of the measures could include outlining some sort of common policies for different App-Stores, enabling them to better manage their Apps delivery mechanisms.

In this paper, we aim to outline a proposal on common policies for mobile Apps management mechanisms for different mobile App-Stores. Therefore, an assessment of the major mobile App-Stores was carried out in order to establish an understanding of their current Apps delivery/monitoring mechanisms. This included the current procedures involved in uploading, downloading, and upgrading the Apps to/from an App-Store along with the current reporting mechanisms deployed by different App-Stores for the malicious Apps reporting.

The paper will start off by presenting a comparison of different App-Stores in the next section. Furthermore, a proposal of common policies/best practices for Apps management will be presented in the third section of this document before outlining the future-work and concluding this study in the sections four and five (respectively) of this paper.

978-1-4673-0229-6/12/$31.00 ©2012 IEEE
Authorized licensed use limited to: UNIVERSITY OF NOTTINGHAM. Downloaded on May 13,2023 at 14:49:01 UTC from IEEE Xplore. Restrictions apply.

II. A DISCUSSION ON DIFFERENT MOBILE APP-STORES APPLICATIONS MANAGEMENT MECHANISMS

There are a number of unique challenges and pieces to application security that most devices currently, are not able to provide. Similarly, all major mobile platforms bear different Apps delivery mechanisms and hence the associated risks also differ to a certain extent. One of the common major tasks however, for all mobile Apps platforms/providers is to ensure the Apps authenticity and a secure management/delivery mechanism for all Apps available on their App-Stores.

The differences between the capabilities of platforms have a significant impact on the management of their security. Managing the Apps delivery mechanism is all about the capabilities of a mobile devices platform/App-Store to monitor/authenticate the legality of an App's source. Being able to keep a track/record of where an App came from can be vital to deploy appropriate countermeasures against a malicious or a fraudulent App.

Furthermore, the rapid changes in this largely consumer driven mobile devices market mean that code is quickly written, deployed and replaced or even upgraded. Development platforms that support the writing of a secure code are currently lacking for the mobile devices. This is particularly the case for the mobile devices Operating Systems (OS) which are often written in either "C" or other native languages leaving security totally at the discretion of the developer.

Therefore, balancing the restrictions imposed by application delivery mechanisms while ensuring an acceptable level of versatility and usability of the smart phone is a challenge for the Apps providers/vendors. Some vendors provide for encoded signatures on applications and some restrict applications to a single controlled source, while others have no restrictions on the source of an application. Table 1 below, outlines the Apps delivery security mechanisms which are put in place by different mobile platforms:

Table 1 - Outlines the Apps Signing, Revocation, and Approval Procedures for Different Mobile App-stores (Source: Veracode[1])

Platform	Signing	Revocation	Approval
Symbian	Signed by Vendor	Yes	Quality
Android	Anonymous, self-signed	Yes	No
iOS	Signed by Vendor	Yes	Policy and Quality
Windows Phone	Signed by Vendor	Yes	Policy, Quality and Security
Blackberry	Signed with Vendor issued key	Yes	No

As it can be gathered from the above table, all mobile platforms have some sort of App signing mechanism in place. Similarly, it is also apparent that all mobile platforms support "revocation" in order to remove malicious Apps, once reported or detected.

Moving on, the mobile OS will not allow Apps that are not signed to execute with an exception of the "jail-broken" or "rooted" OS's which are in fact "jail-broken or rooted" by the users specifically to allow unsigned apps to be executed.

Depending on the implementation of the signing mechanism, it can be a great achievement in terms of improved security. For instance, if the App is signed by the developer with a self-generated key, there is little security gain but if the application is signed by a key issued by the platform provider then there will be a security benefit based on the policies the platform provider adheres to, for approving Apps. Moreover, it is important to stress that the mobile jail-breaking removes the security benefits of the platform signing mechanism altogether.

Android's Market App store for instance, is probably not up to the challenge when it comes to malicious Apps publishing and distribution. This is evident in the ease with which malicious Apps can be uploaded and distributed on the Android Market.

Fig.1 outlines the process of applications development and publishing on the Android Market:

[1] http://info.veracode.com/Whitepaper-2011-Mobile.html

Authorized licensed use limited to: UNIVERSITY OF NOTTINGHAM. Downloaded on May 13,2023 at 14:49:01 UTC from IEEE Xplore. Restrictions apply.

Figure 1. Android applications development and publishing process[2]

Those malicious Apps can access the sensitive OS resources such as the text messages, mobile device location via GPS, camera, voice recording, to name a few. It can therefore be stated that developing, publishing and distributing a powerful fraudulent Android App which could steal one's personal information including their financial information is almost trivial as the current security measures deployed around the App submission process by Android are inadequate to identify and prevent submission of malicious applications to the Android Market.

The Apple's App Store on the other hand, appears to be a "walled garden" and it does employ an approval process but the details of its approval process are not well documented or publicized. It is therefore difficult to establish exactly what sort of details the iOS Security Team at the Apple App store takes into consideration, when it comes to an App approval or screening. Having said that, it is quite clear, based on the deployment of the aforementioned approval process, that the Apple App store is putting a lot of effort in ensuring the user experiences and the compliant to its policies.

Fig.2 outlines a procedure of sharing an Apple's mobile App with tester:

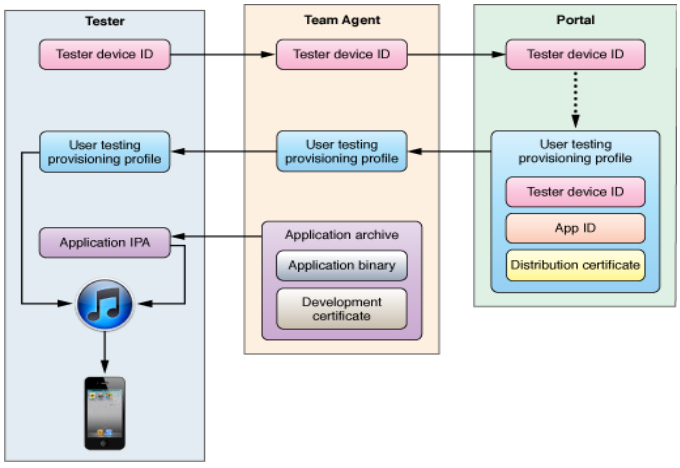

Figure 2. A process of sharing Apple's mobile Apps with testers[3]

[2] http://developer.android.com/guide/publishing/preparing.html

It is therefore difficult to establish exactly what sort of details the iOS Security Team at the Apple App store takes into consideration, when it comes to an App approval or screening. Having said that, it is quite clear, based on the deployment of the aforementioned approval process, that the Apple App store is putting a lot of effort in ensuring the user experiences and the compliant to its policies.

According to a recent report from Symantec[4], Apple and Google are very different when it comes to mobile security, "creating distinct potential vulnerabilities for enterprises embracing devices running these operating systems". Apple employs an "Application Provenance" strategy which basically involves identifying, certifying and vetting an App before it is published on to the Apple App store. For Android Market on the other hand, the course of action is completely different as there is no vetting process as there are far more self-signed applications and the Apps can be uploaded from just about anywhere on the internet. So although these user-friendly Apps are continuously getting more and more popular amongst the users, these are also bringing new security threats and breaches with them.

One of the proposed solutions could be that the Android apps are not only self-signed but are also signed by certified keys issued by the trusted authorities. This strategy on its own may not prevent malicious Apps but it would certainly assist in deploying a tracing mechanism which would allow tracking down the fraudulent App owner/publisher.

As far as the malicious/fraudulent App reporting procedures are concerned, these differ from each app-store and the most common issue is that the reporting procedures are not clearly defined by the App-stores. The same goes for the removal/take-down procedures of a malicious App. The Google's Android Market for instance, does have an option to report a malicious/fraudulent App on its website, which can be used to request Android Market to review and remove an inappropriate App from its store but it could become a difficult task for an end user to find the "reporting option" on Android Market website.

Furthermore, there have been a number of instances whereby a malicious App was reported but it took a while until that App was taken down by the Android Market. Therefore, it can be stated that the process of identifying/reporting and removing/take-down a malicious App from the Android Market requires serious considerations and perhaps improvements as a result of those considerations.

[3] http://developer.apple.com/library/ios/#documentation/Xcode/Conceptual/ios_development_workflow/145-Distributing_Applications/distributing_applications.html#//apple_ref/doc/uid/TP40007959-CH10-SW2

[4] http://www.networkworld.com/news/2011/062811-symantec-mobile-report.html

Authorized licensed use limited to: UNIVERSITY OF NOTTINGHAM. Downloaded on May 13,2023 at 14:49:01 UTC from IEEE Xplore. Restrictions apply.

In the case of Nokia's Ovi Store, there is a revocation process in place for malicious Apps. Revocation provides a final sanction to handle applications which may pose a threat to the Nokia users, its network or the Nokia's Ovi community as a whole to ensure those malicious Apps do not spread any further. Fig.3 outlines an App submission and signing process for Nokia's Symbian:

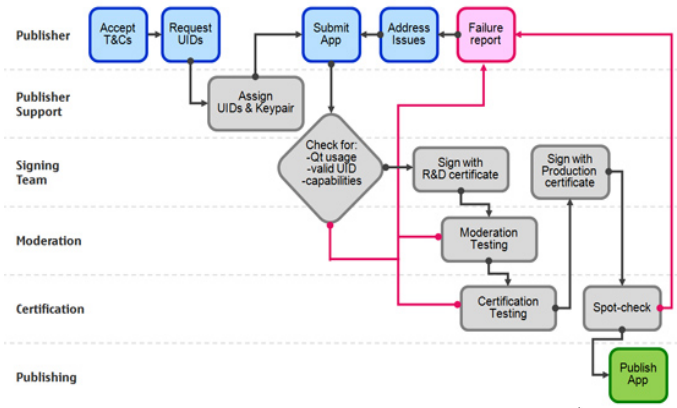

Figure 3. Symbian App submission and signing process[5]

Based on the findings of this study, the next section of this paper will aim to propose common policies or best practices for Apps management in order to improve the security of mobile Apps ecosystem.

III. Mobile Applications Management Best Practices for Different App-Stores

1. While conducting a security review of the submitted Apps, the App-Stores should consider a number of different known threats associated to the mobile Apps. These threats could include but are not limited to[6]:
   - Activity monitoring and data retrieval
   - Unauthorized premium rates dialing and premium rates SMS
   - Mobile banking and mobile payments related frauds (since the mobile banking Apps are increasingly being uploaded and downloaded)
   - Unauthorized network connectivity
   - UI (Unique Identifier) impersonation
   - Sensitive data leakage
   - Unsafe sensitive data transmission

2. The App-Stores should increase the user awareness regarding malicious Apps and the consequences of downloading such Apps by introducing a number of different measures including an informative message, clearly visible on the App-Stores' websites. This should also be the case when it comes to the reporting procedures so that a user could follow simple steps in order to report malicious Apps to the relevant App-Store.

3. An introduction of a warning message for the Apps developers/authors could also be introduced, stating that a malicious App upload will lead to their accounts being deleted permanently and possibility of their details being passed onto the law enforcement agencies, in case of a criminal activity as a result of their App being downloaded by the users.

4. All App-Stores should look into placing a quality framework in the form of a series of standard security tests which could issue health certificates for different Apps in order to combat the growing threat of fraudulent Apps.

5. Apps Isolation - A strong security mechanism for ensuring mobile Apps access permissions could be introduced by the App-Stores, since it would not be sufficient to allow an App access to certain functions as the functionality needs to be further restricted by type, by time and conditions.

6. Banking and other Mobile Financial Services related Apps should only be allowed to be published by the banks and related financial institutions, ensuring their credentials are fully verified prior to submission of those Apps.

7. In order to improve the security, the App-Stores could carry out a pre-screening of the Apps and could also deploy a continuous monitoring procedure to ensure that the Apps which have been upgraded could also be screened / monitored.
8. A comprehensive vetting mechanism for Apps developers/authors could be deployed for all app-stores to ensure a successful tracking-down of the owner of the App in case of an App turning out to be a malicious App. For instance, all Apps publishers must have an account with the appropriate app store before they could submit an App to the app-store. Also, the app-stores should verify those publishers' accounts by emails, postal addresses, telephone numbers, and also through their credit card details.

9. All App-Stores should consider deploying some sort of Apps and Apps developers/authors reviewing procedures including the review counters, which would automatically raise the alarm in case of a negative review. Perhaps an Apps auto-suspension procedure could also be deployed in case of reaching so many numbers of negative reviews (e.g. 5), until the App is investigated by the app-store security team. A serious consideration should also be given to monitor how those feedbacks were being submitted

[5] http://www.developer.nokia.com/Distribute/Packaging_and_signing.xhtml
[6] http://info.veracode.com/Whitepaper-2011-Mobile.html

Authorized licensed use limited to: UNIVERSITY OF NOTTINGHAM. Downloaded on May 13,2023 at 14:49:01 UTC from IEEE Xplore. Restrictions apply.

in order to ensure that the fraudsters could not abuse the review process.

10. An App review rating from one App-Store could also be used by other App-Stores through some sort of a shared reviewing mechanism implemented jointly by all app-stores. One of the most serious concerns here could be that the most users would rate Apps for their actual functionality and not for their security aspects. It would therefore be important to have two separate categories for the App reviewing mechanism, one for security and privacy issues, which could include an App asking for excessive privileges at install, and the other for the general functionality issues of an App, which could include information such as the App worked as it was supposed to, etc.

11. A continuous monitoring of the Apps developers/authors in a way that the prior Apps contribution should not be taken into account and each new submitted App is considered on its own basis.

12. The malicious Apps reporting procedures should be well defined and easily visible (highlighted) and accessible to the users of an App-Store. One of the options could be a simple reporting form in order to submit a complaint which should be dealt with and resolved, as quickly as possible. Another suggestion could be a live chat channel in place on all App-Stores so that a user who wishes to report a malicious App could do so, as and when needed.

13. A simple but effective and appropriate, "take-down" or "removal" procedure for all different App-Stores could be implemented. Perhaps a unified "shared" take-down mechanism could be employed throughout different App-Stores in order to ensure the malicious App publisher gets barred from all stores.

14. All App-Stores should have a kill-switch to remotely kill a malicious App upon reporting/discovery. A generic policy and procedure on "remote wipe" could also be implemented in case of a malicious App disaster ensuring that the users are fully aware of those procedures.

15. All App-Stores should consider priority vetting for updates of existing Apps in order to enable the Apps developers/publishers to patch up the vulnerabilities quickly and effectively.

16. A knowledge share mechanism between different App-Stores could be deployed in order to report and monitor a fraudulent developer's/author's activities.

17. A generic (for all App-Stores), mobile devices Apps downloading and user's best practices could be developed in order to ensure an increased awareness for the end users.

IV. FUTURE WORK

The aforementioned best practices for different App-Stores Apps management mechanisms and the comparison of different App-Stores Apps management strategies could further assist in enhancing the security of mobile Apps ecosystem. These practices could be further developed in the future and could also be an App-Store specific. Hence, the findings of this study could be used as a starting-point by different entities interested in further improving the security of the Apps publishing and delivery mechanisms.

Moreover, the findings of this study could be tested through a test-bed environment by actually implementing the aforementioned best practices to different App-Stores. This would enable us to further assess the efforts needed by the different App-Stores in order to implement the proposed best practices.

V. CONCLUSIONS

The mobile App-Stores should ensure they take into consideration the current security issues, including the known threats and vulnerabilities when permitting an App to be published on their stores, in order to enhance the overall security of the App delivery management mechanism.

Similarly, the Apps developers, while developing an App, should not only consider an App's functionality and usability but must also take into account the security aspects of that specific App. This could include App isolation, ensuring that an App must not attempt to access unnecessary resources, etc.

ACKNOWLEDGMENT (HEADING 5)

This research was carried out in conjunction with Mobey Forum's Security Task Force (STF) and la Caixa bank's CSIRT group.

We would also like to thank Prof. Roberto Di Pietro of Università di Roma Tre, Italy, for his valuable contribution to this paper.

REFERENCES

[1] Goodin, D. (2011), Malicious apps infiltrate Google's Android Market. Available at: http://www.theregister.co.uk/2011/12/12/android_market_malware/ [Accessed, 12th December 2011]

[2] Zetlin, M. (2011), 8 Tips to Stop Banking App Fraud. Available at: http://www.creditcards.com/credit-card-news/eight-tips-stop-banking-app-fraud-1282.php [Accessed, 24th August 2011]

Authorized licensed use limited to: UNIVERSITY OF NOTTINGHAM. Downloaded on May 13,2023 at 14:49:01 UTC from IEEE Xplore. Restrictions apply.