Third-Party App Stores Delivered via the iOS App Store

By Trend Micro (words)

Published: 2017-03-23 · Archived: 2026-04-05 21:51:17 UTC

The iOS ecosystem is usually described as a closed ecosystem, under the strict control of Apple. However, there are still ways to get around this tight control. Remember the Haima app? That method relied on enterprise certificates from Apple—which are costly, since the certificates needed are changed very frequently.

We are currently seeing how third-party app stores are improving. Recently, we saw an app that leads to a third-party app store being offered *on* the official iOS App Store. To evade detection, this app was concealed as a legitimate app. In at least one case, an app used for jailbreaking was available via this third-party app store.

It is unclear exactly who the target of this malicious app is. The account book app is designed with Japanese characters, but the app store itself is written in Mandarin Chinese. In addition, it was found in the App Store of multiple countries.

This app is named “こつこつ家計簿-無料のカレンダー家計簿”, which translates to “Household Accounts App”. This app appears to be a financial helper app for families, but it is actually a third-party app store. Apple has since removed it from the App Store.

Figure 1. Household account app in iOS App Store

Figures 2-4. Various stages of app launch

The code (Figure 5, below) reveals that it checks the *PPAASSWOpenKey* key in the system’s user preference *plist* when it first starts up. This key is used by the app to determine if the app has run before: as it hasn’t, the key does not exist. The app switches to the *else* branch, which requests the right to use data to access the third-party store. Because of iOS’s permission mechanism, this request needs to be approved specifically by the user (Figure 2). The first request therefore fails, so the app jumps to the account book view and pretends to be a legitimate app (Figure 3). The text in Figure 3 claims that data access is necessary for exporting information from the app.

Figure 5. Code for switching to account book view (Click to enlarge)

So long as the app is not closed or goes to the background, the app will stay in the account book view. However, once it enters the background, it again attempts to connect to the third-party store. This time, it should succeed, and instead of seeing the account book view, the user will instead see the third-party app store (Figure 4).

Figure 6. Code for switching to third-party app store view (Click to enlarge)

Why would the creators of the third-party app store need this particular behavior? Putting their app store inside Apple's official one makes it easier for would-be users to access it, but subterfuge is needed to pass Apple's scrutiny.

Installing the apps

After successfully making it into the App Store, the third-party app store then needs to be able to install apps. To do this, the app's creator employs a technique generally used to install apps signed with enterprise certificates. The technique involves the creation of a plist file, which is used to install apps. Figure 7 shows an example plist file (note that this file is not the same as those used by this third-party store).

Figure 7. Sample plist file

Then, creating a link as shown below:

> itms-services://?action=download-manifest&url=https://{web server address}/install_app.plist.

By opening the URL, the app is installed. For apps signed with an Apple certificate, there's an additional catch: the app must have been purchased with the user's Apple ID. This is why the app asks for the user's Apple ID, so it can complete the purchase process:

Figures 8-11. Stages of app installation

Figure 12. Code for creation of plist file and installation link (Click to enlarge)

Figure 13. Code for creation of app installation process (Click to enlarge)

Malware Distribution

In addition to apps already present in the App Store, it can also sell apps which are not normally distributed via the App Store. Unfortunately, this can include malware and other unwanted applications.

An example is the app called "PG Client", which is a tool for jailbreaking iOS devices. It was once available on the App Store, but has already been removed. It is still available for download via the third-party app store:

Figure 14. PG Client

Other malicious apps will often ask users to download the *PPHelper* app on their PC, which is also an iOS jailbreaking tool. This is installed on a PC/Mac and the user is asked to connect their iOS device to the said

PC/Mac. *PPHelper* gets some files associated with the user's authorization from the device and communicates with it as if it was iTunes, effectively bypassing some of the DRM protection of iOS.

Figures 15-17. Stages of requesting user to connect PC helper app

 

Figures 18-19. Code requesting the user's device authorization

Promoting Other Apps For Money

This particular app was not the only problematic app we saw in the App Store. We found another one called “爱应用助手”, which translates to “LoveApp”. This is designed to be used to promote apps that are already in the App Store. This, in effect, bypasses either Apple’s arrangement of apps in searches and the paid Search Ads option. This app makes its money from developers who want to promote their wares without going through Apple's promotion service.

iOS includes various APIs that are meant to allow a developer to easily display their app's page. LoveApp uses this to easily direct users from its own listings to the App Store listing of the promoted apps:

Figure 20. Installation of promoted app

The "LoveApp" is in the background, with the App Store window of the promoted app in the foreground. The code that does this can be seen here:

Figure 21. Code for opening the App Store window (Click to enlarge)

From a privacy perspective, LoveApp has multiple issues. Firstly, during installation, it uploads some user attributes to their servers, including their advertising identifier (idfa). This is primarily used to count the number of downloads.

Figure 22. User attributes uploaded

In addition, it also uses a third-party SDK called *TalkingData*. The app uses it to gather information about the user's behavior:

Figure 23. TalkingData call

This SDK, however, has many aggressive API calls. Its capabilities include acquiring various parts of the user's system information (including the Wi-Fi network name, running processes, and IP address). If the user's phone has been jailbroken, the SDK can also gather the user's Apple ID and installed apps. This is enough for us to consider it a potentially unwanted application.

Figure 24. Other aggressive API calls (Click to enlarge)

Risks and Mitigation

We recommend that users be careful about downloading apps from third-party app stores. Apple can't endorse the safety of any of the apps delivered via third-party stores, and such is the case here: users are still exposing themselves to various security threats (including malware and other unwanted apps). Organizations should put in place policies to reduce the risk from these malicious apps, such as blocking unapproved app stores and safeguarding personal devices.

We notified Apple about the presence of both of these apps in the iOS App Store prior to publication of this blog post. The following files are related to this incident:

SHA256 hash	Detection name
212015dbae6e36c703c513f762413ffe fe5ad58720c22abb696bca94f3b6c14b	IOS_FakeAppStore.A
adcfa3d540f78297dde3dcbf0191271d 8592911d71703ce853b6de622421c1fb	IOS_JailBreakTool.A
c75777079d72c43516adc7bdee4db447 f22bbd25af26c08bcee42f885a820866	IOS_FakeAppStore.A
