Lookout

Whitepaper

Enterprise Mobile Security

Managing App Sideloading Threats on iOS

I. Introduction

Through rigorous app review Apple has lowered the risk of downloading malware from its App Stores to near zero. Companies, however, increasingly rely on an app-distribution mechanism called enterprise provisioning that allows them to distribute apps to employees without Apple's review as long as the apps are signed with an Apple-issued enterprise signing certificate.

Unfortunately, attackers have managed to hijack this app-distribution mechanism to sideload apps on non-jailbroken devices, as demonstrated in the recent Wirelurker attack. Organizations today face a real security threat that attackers will continue to abuse enterprise provisioning and use it to sideload malware, especially since:

1) The widespread prevalence of legitimate, enterprise-provisioned iOS apps in the workplace has conditioned employees to seeing (and ignoring) the security warnings triggered on devices when installing these apps. With minor social engineering, many employees would likely accept and install a sideloaded app.

2) Sideloaded apps have unrestricted access to the device, including APIs that Apple otherwise prohibits. These private APIs can empower sideloaded apps with a wide range of dangerous capabilities, such as the ability to install or launch additional code or collect location data without notification.

This whitepaper examines the technical underpinnings that enable app-sideloading and also highlights Lookout's unique approach to managing this emerging security threat on iOS devices.

II. The Path to App Sideloading

Signing Certificates

Apple offers two types of signing certificates for app distribution outside of their App Stores and both types allow users to install and execute signed apps on non-jailbroken devices:

1) A developer certificate, intended to sign and deploy test apps to a limited number of devices.

2) An enterprise certificate, intended to sign and widely deploy apps to devices within an organization.

To obtain these certificates you must enroll in one of Apple's two iOS developer programs. Table 1 on the following page summarizes the enrollment requirements for each program and their app provisioning restrictictions.

Both types of signing certificates expire after a year, whereupon developers can apply for new ones. Apple can also revoke certificates if it learns of abuse and an app signed with an expired or revoked certificate will no longer run on an iOS device.

Table 1: iOS Developer Program Characteristics

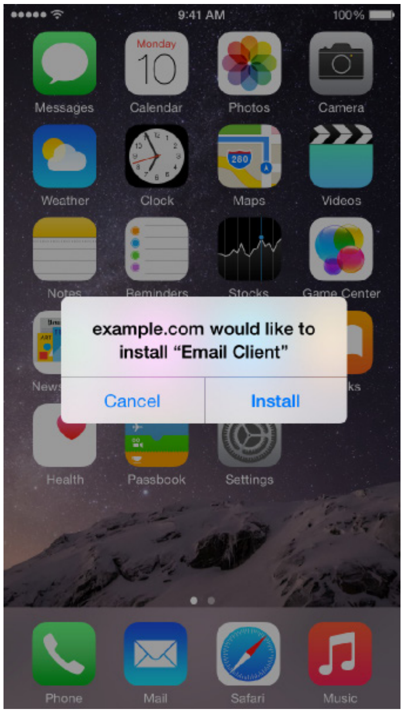
Developer Program	Cost	Enrollment Requirements	Certificate Issued	Provisioning Limit	Device Provisioning Requirements
iOS Developer Program[1]	$99/year	- Valid credit card - Valid Apple ID	Developer Certificate	100 devices	Devices must be manually authorized via their UDID[2]
iOS Developer Enterprise Program[3]	$299/year	- Valid credit card - D-U-N-S number[4]	Enterprise Certificate	Unlimited devices[5]	None

Provisioning Profiles

To run an enterprise or developer-signed app, an iOS device must first install a file called a provisioning profile. Provisioning profiles contain the signing certificate and the app ID, verifying that the app's developer is registered with a valid iOS developer account.

To streamline the installation process, developers increasingly embed provisioning profiles inside apps. When a user attempts to download an app outside the App Store that contains a provisioning profile, their iOS device will alert them (see Figure 1a at right), asking them to confirm the download. In iOS 9 Apple added an interim step that requires iOS users to then open their device settings to trust the provisioning profile. Their device will then alert them again (see Figure 1b on the following page) to confirm they trust the developer behind the provisioning profile. When a user trusts a developer their device will then automatically trust any apps that use the same provisioning profile in the future and will not trigger additional security alerts for those apps.

Figure 1a: Download Security Notice[6]



example.com would like to install "Email Client"

[1] See: https://developer.apple.com/programs/ios/

[2] UDID: Unique Device Identifier

[3] See: https://developer.apple.com/programs/ios/enterprise/

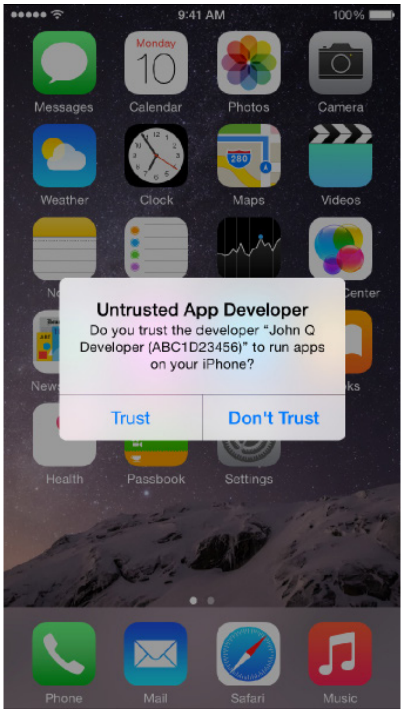
[4] D-U-N-S Number: a unique, nine-digit identification number issued by Dun & Bradstreet that attests to a company's physical address.

[5] While Apple contractually requires organizations to only distribute enterprise-signed apps to their employee devices, outside of a signing certificate expiration or revocation, enterprise-signed apps can be installed on an unlimited number of iOS devices.

[6] Image source: https://support.apple.com/en-us/HT204460

Many companies today deploy enterprise-provisioned apps using a Mobile Device Management (MDM) service. Devices under MDM management can have multiple enterprise-provisioning profiles installed. In addition, an MDM profile does not prevent devices from downloading enterprise-provisioned apps from third parties as long as those apps are signed with a valid enterprise certificate.

Figure 1b: Installation security notice



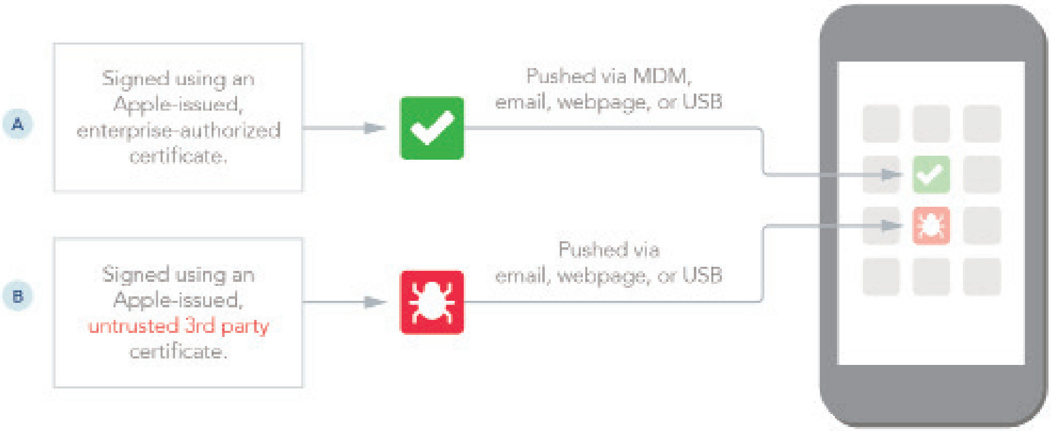
Untrusted App Developer

Do you trust the developer "John Q Developer (ABC1D23456)" to run apps on your iPhone?

III. The App Sideloading Threat

The fundamental challenge with the app sideloading threat is that a sideloaded app and a legitimately-provisioned app look much the same to an iOS device since both use Apple-issued signing certificates as depicted in Figure 2 below.

Figure 2: Enterprise App Provisioning (A) vs. App Sideloading (B)



A
Signed using an Apple-issued, enterprise-authorized certificate.
Pushed via MDM, email, webpage, or USB
B
Signed using an Apple-issued, untrusted 3rd party certificate.
Pushed via email, webpage, or USB

When it comes to signing certificate abuse, developer certificates make for a less practical attack vector since an attacker would also need to obtain the UDIDs of target devices. Enterprise certificates, however, do not have this constraint since an enterprise-signed app will run on any iOS device provided the user accepts the security notices, even on non-jailbroken devices.

To obtain an enterprise certificate through their developer program, Apple requires applicants to provide a D-U-N-S number as proof that they have a legally registered business. Unfortunately, motivated attackers can circumvent this requirement by registering legal entities themselves. With minimal effort, for example, an attack can register a new business online via a variety of legal services and then use this registration to obtain a valid D-U-N-S number, and thereby, an enterprise certificate.

Attackers that can't be bothered with this effort can also reuse enterprise certificates issued to other parties.

In 2013, for example, it came to light that mobile development services company, MacBuildServer, was openly signing any third party apps with its own enterprise certificate[7]. Although Apple promptly revokes a certificate when it learns of signing abuse (which occurred in the case of MacBuildServer) it's difficult to keep tabs on this problem. Security researchers, for example, recently documented more than 1,000 iOS apps available for public download outside the App Store using enterprise and developer certificates[8].

Ultimately, attackers have taken notice of enterprise provisioning abuse as a viable means to sideload iOS malware. Table 2 below documents two recent iOS attacks that abused enterprise provisioning and sideloaded surveillanceware on non-jailbroken devices:

Table 2: Recent Sideloaded iOS Threats That Abused Enterprise Provisioning

iOS Threat	Year	Description
XAgent	2015	XAgent is iOS surveillanceware that collects a range of sensitive data from compromised devices including SMS, contacts, photos, and GPS locations; it can also remotely activate voice recording on compromised devices.[9]
Wirelurker	2014	Wirelurker is iOS surveillanceware delivered via USB connections to infected OS X devices. Wirelurker can capture contact lists and SMS messages from compromised devices.[10]

[7] "Apple Slams The Door On Super Mario". ReadWrite. July 2013. http://readwrite.com/2013/07/17/apple-slams-the-door-on-super-mario

[8] "Enpublic Apps: Security Threats Using iOS Enterprise and Developer Certificates". Zheng, Min, Hui Xue, Yulong Zhang, Tao Wei, and John C.S Lui. April 2015. http://www.cs.cuhk.hk/~cslui/PUBLICATION/ASIACCS15.pdf

[9] "XAgent iPhone Malware Attack Steals Data without Jailbreaking". Mac Observer. February 2015. http://www.macobserver.com/tmo/article/xagent-iphone-malware-attack-steals-data-without-jailbreaking

[10] "Malicious Software Campaign Targets Apple Users in China". The New York Times. November 2014. http://bits.blogs.nytimes.com/2014/11/05/malicious-software-campaign-targets-apple-users-in-china/?_r=0

IV. Lookout's Approach to App Sideloading Protection

To protect an organization from sideloaded apps, Lookout's platform automatically analyzes apps delivered from outside the App Store and determines whether they can run on the device or not. Figure 3 below demonstrates this approach in greater detail, in the following processes:

1) An attacker distributes an enterprise (or developer) signed app to employee devices via email attachments, webpage links, or USB cable.

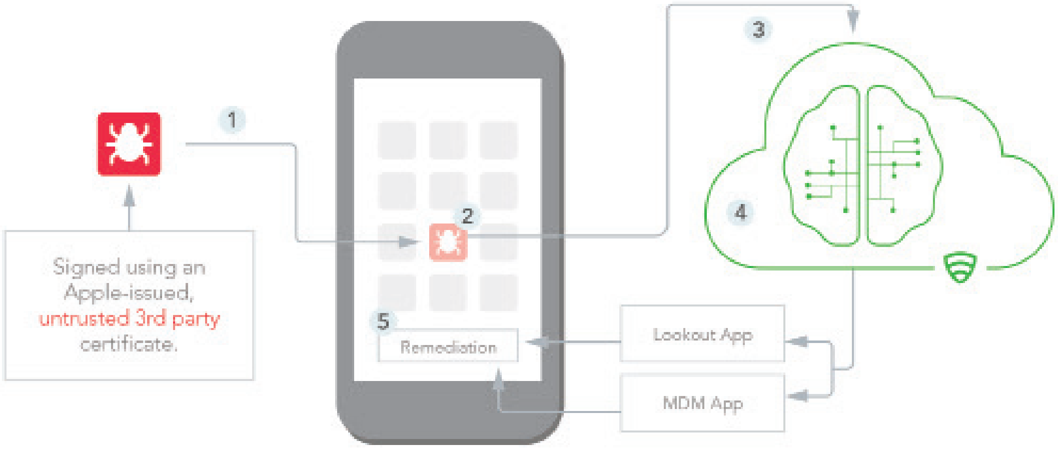
2) An employee downloads and installs the app after accepting the standard device security notices.

3) Before the app is executed, Lookout sends a range of security telemetry about the app to the cloud where it is instantly analyzed and Lookout then immediately alerts the user if the app is not approved by their organization.

4) The analysis performed in the cloud includes app metadata analysis to determine the app's source as well as analysis of the app's signing certificate to determine if it's authorized for use within the organization.

5) Lookout's remedial actions include not only an immediate user alert to remove the app, but also an alert to Lookout's admin console. Where organizations have integrated their MDM service, Lookout can also alert the MDM to enforce device policy (e.g. shutting down network access).

Figure 3: Lookout's Approach to App Sideload Protection


1
2
3
4
5
Signed using an Apple-issued, untrusted 3rd party certificate.
Remediation
Lookout App
MDM App

Lookout's approach to app sideloading detection offers more comprehensive protection when compared to existing security approaches. Advanced threats can fool app reputation and app behavioral analyses by replacing legitimate apps with undetectable, trojanized versions (e.g. Masque Attack) or by suppressing undesirable behaviors.

In each case, Lookout's platform would not be fooled by these evasive maneuvers since it would block the threats based purely on their unauthorized distribution method. This novel approach to sideload detection, combined with Lookout's advanced threat and jailbreak detection capabilities, better positions Lookout to detect iOS threats, including novel iOS attacks.

In summary, Lookout's unique approach lowers the risk of a sideloaded attack to almost zero. To evade this approach an attacker would need to compromise the enterprise or developer certificates that belong to the target organization. Otherwise, if an attacker uses their own certificate or even a third-party certificate to sign a sideloaded app then Lookout will detect this threat.