BYPASSING THE iOS GATEKEEPER

AVI BASHAN
Technology Leader
Check Point Software Technologies, Ltd.

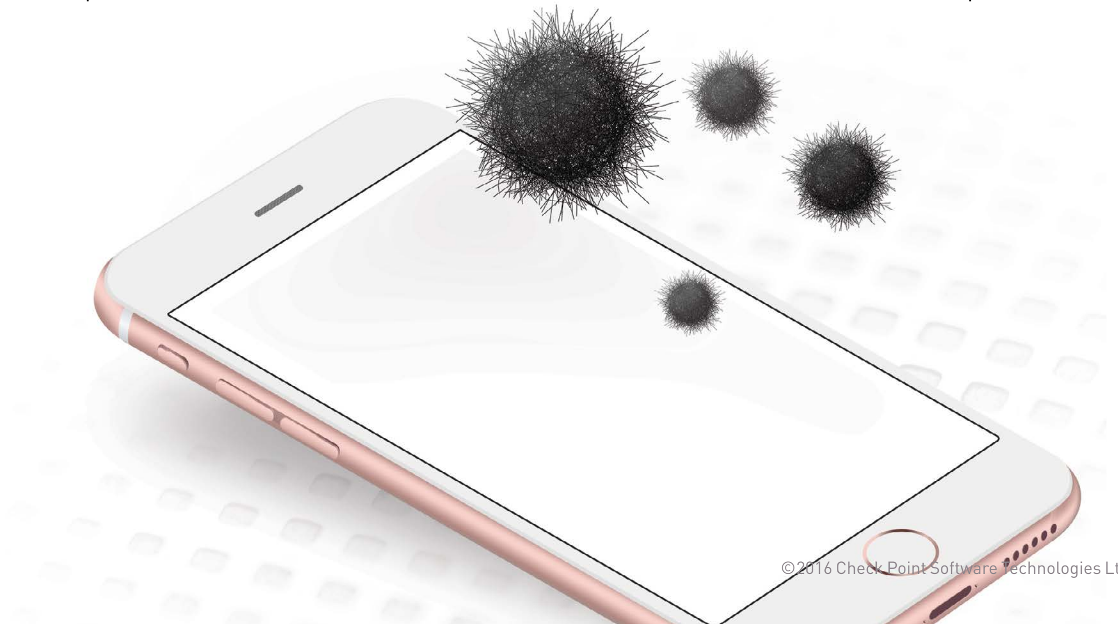
OHAD BOBROV
Director, Mobile Threat Prevention
Check Point Software Technologies, Ltd.

EXECUTIVE SUMMARY

The Apple® iOS security paradigm is built on several main factors, one of them being that the Apple App Store® offers a central distribution process that allows verification of nearly all code executed on iPhones and iPads. However, recent attacks targeting iOS (e.g., Masque Attack, WireLurker, Hacking Team, YiSpecter, to name a few) demonstrate that there are some inherent vulnerabilities in this ecosystem. This report covers one such vulnerability, which exposes iOS users in the enterprise to cyberattacks.

The Check Point research team has highlighted an attack vector—a possible security flaw in Apple's iOS 9 that allows threat actors to install malicious apps on enterprise employees' iPhones and iPads. The flaw enables threat actors to stage a Man-in-the-Middle attack that hijacks communications between managed iOS devices and Mobile Device Management (MDM) solutions. This exploit could give threat actors control of devices, the data that resides on them, and even enterprise services, potentially impacting millions of iOS users worldwide whose devices are managed by an MDM.

Check Point's policy in such cases is to immediately notify the manufacturer of the product in which an attack vector has been found, to give the company an opportunity to resolve the situation. Accordingly, the Check Point research team's findings and a video demonstrating how a threat actor could attack an iOS device were submitted to Apple's security team in October 2015. Apple responded in November 2015 that the behavior the research team demonstrated "is expected."



DEVELOPING APPS FOR IOS

A substantial part of the iOS value proposition hinges on providing superior security and privacy for users and their sensitive data. Apple uses several distinct approaches to achieve this, including:

- **Sandboxing:** Each app is embedded within a sandbox that limits its privileges and capabilities to access or operate within protected parts of the OS.
- **Permissions:** Apps that require certain access to resources to function must request specific permissions from the user.
- **Signed code:** Unless a device is jailbroken, only signed code can run on a device. This is done to ensure all code running on a device has received final approval from Apple and is safe to use.

iOS apps are strictly controlled by Apple. As part of its app strategy, Apple provides a single App Store for users to download apps to their devices. Developers must be registered in order to publish apps on the App Store. This enables Apple to control who's developing apps for its ecosystem and to banish developers who fail to meet its standards. iOS developers can't develop apps anonymously if they want their apps distributed through the App Store.

APP REVIEW PROCESS

Each time a developer publishes a new version of an app, it must undergo a rigorous security review. This review process may take several weeks and is conducted according to a basic set of rules outlined in the App Store Review Guidelines. These rules filter out apps that contain improper content, that are low quality, or that have malicious intent.

During the review, the app's content, functionality, behavior, APIs, and many other aspects are scrutinized to prevent malicious or dangerous apps from being published in the App Store. Although this process is thorough, Apple's review process can sometimes allow apps that include malicious code into the App Store (Apple Confirms Discovery of Malicious Code in Some App Store Products, New York Times, September 20, 2015).

APPLE DEVELOPER ENTERPRISE PROGRAM

Apple created the Apple Developer Enterprise Program to offer organizations a way to develop and distribute apps for internal enterprise use. These apps can be distributed quickly and directly to devices, enabling enterprises to develop apps that meet their business requirements without publishing them on the App Store. In this way, organizations can avoid a lengthy review process and, more importantly, keep these apps only for internal use by employees.

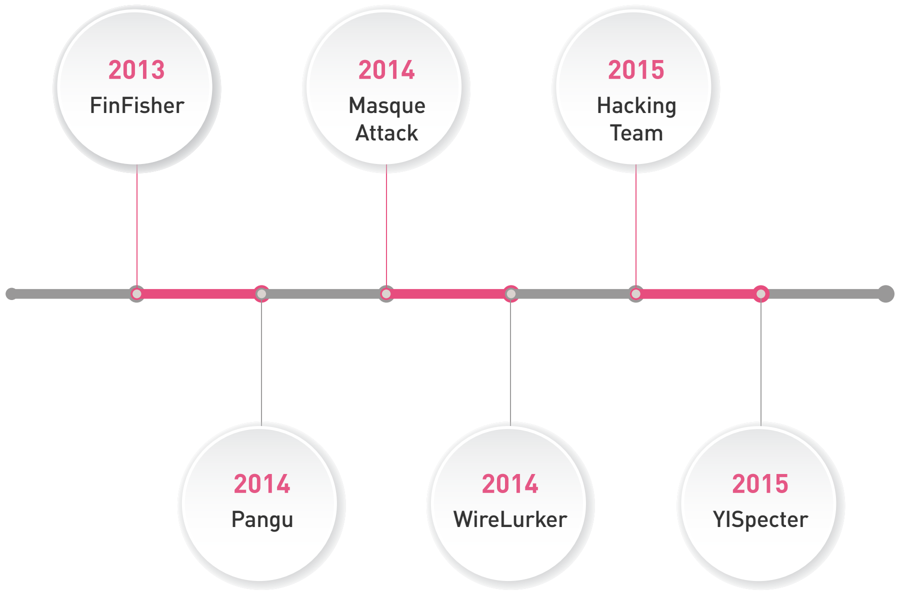
WHAT IS AN ENTERPRISE CERTIFICATE?

An enterprise certificate is a certificate signed by Apple that developers can use for signing apps they create in XCode. Apps signed with this certificate can be installed on iOS devices without having to be vetted through the traditional App Store process. This is done not only for testing purposes but for enterprises who may want to develop apps themselves then distribute them to their employees without requiring that these employees install the app through the App Store. Developer Enterprise certificates have relaxed restrictions on the number and type of devices on which apps signed using these certificates can be installed.

FROM THEORY TO PRACTICE: ENTERPRISE CERTIFICATE ABUSE

Enterprise certificates are abused on a regular basis. Third-party app stores like vShare, 25PP, Kuaiyong, 7659, and others abuse certificates as a distribution method. These third-party app stores register as an enterprise with Apple to enter the program and obtain an enterprise certificate. They use the certificate to install apps on their customers' devices, claiming they are "staff members."

Unfortunately, many examples of malicious abuse of enterprise apps can be found:



TEST CASE: HACKING TEAM

One example of an attack that abused enterprise developer certificates is the case of the Hacking Team's iOS malware, which in 2015 targeted iOS versions 8.1.3 and earlier.

The malware leveraged a vulnerability dubbed Masque Attack, which was discovered in 2014 and allowed enterprise apps to replace existing apps on a device, including system apps. The newly installed app kept the original app's directory and could steal any local cache that existed. To create an app with a matching bundle identifier (ID), a threat actor must use an enterprise certificate. This is because the bundle ID verification is performed when a developer submits an app to the App Store for review.

Hacking Team used this attack method and abused an enterprise certificate they owned. Using this certificate, they created a malicious app disguised as the Newsstand app, which was native to iOS until iOS 9. The malware accessed the user's location, photos, and address book, and sent their data to a server they controlled. In addition, the malware installed a keyboard that recorded all keystrokes and sent them to the same server.

The only way Hacking Team was able to leverage Masque Attack was by using an enterprise certificate to bypass the App Store bundle identifier check.

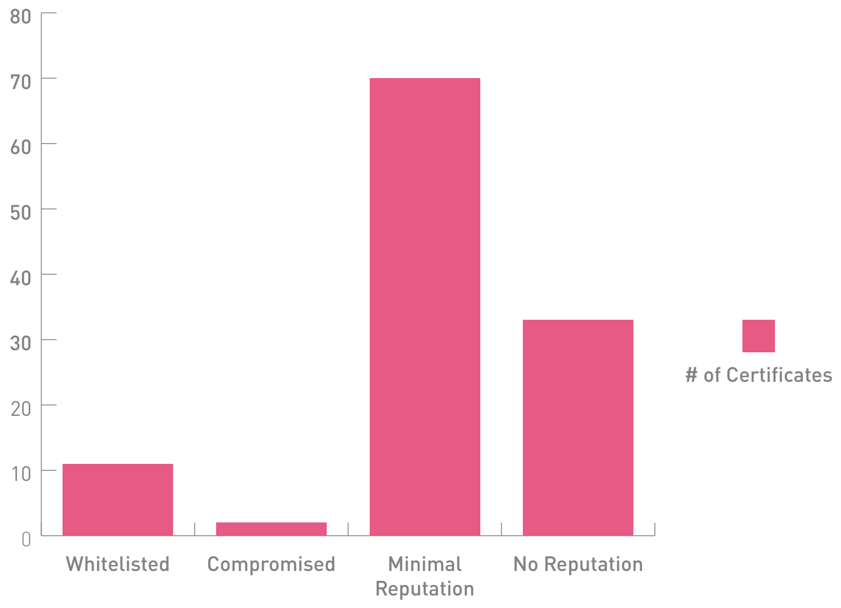
CASE STUDY: FORTUNE 100 COMPANY

Check Point researchers studied the use of enterprise apps in a Fortune 100 company, analyzing approximately 5,000 devices. These were the results:

- 318 unique enterprise apps were installed
- 116 unique enterprise certificates were used

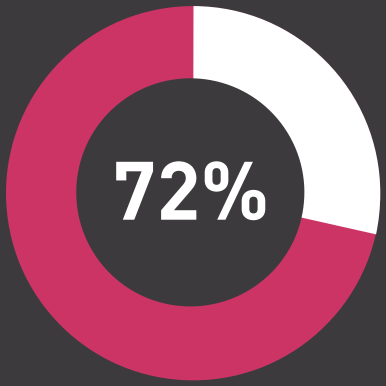
Of the 116 unique enterprise certificates, only 11 belonged to whitelisted developers with positive reputations and with a record of having previously developed apps. Most of the certificates belonged to developers with little or no information about their reputation.

CASE STUDY: NON-APP STORE CERTIFICATES IN A FORTUNE 100 COMPANY



ORIGIN OF ENTERPRISE CERTIFICATES

MORE THAN 70% OF THE ENTERPRISE APPS ORIGINATED IN CHINA AND OTHER COUNTRIES IN ASIA. BECAUSE THE APP STORE IN CHINA IS NOT AS WIDELY ADOPTED AS IN OTHER PARTS OF THE WORLD, DEVELOPERS THERE DISTRIBUTE APPS USING OTHER, ALTERNATIVE APP MARKETPLACES.



A HARD PROBLEM TO SOLVE

Following the attacks shown on the timeline chart above, Apple understood the problem with enterprise apps. Enterprise apps cannot be eliminated, since many organizations are already heavily invested in this solution. However, Apple took certain steps to mitigate the threat.

In response to what amounted to a significant vulnerability to their ecosystem, Apple introduced new security measures for enterprise apps in iOS 9. The first action Apple took was to increase the complexity of executing enterprise apps. For instance, when the enterprise app is initially downloaded, the user must go through a maze of settings screens to verify the app's developer. Only after this verification process is complete can the app be executed. This process differs from previous versions of iOS, in which the user was merely shown a message the first time the app was opened that stated it was from an unknown developer.

Apple did leave a loophole, however. Enterprises use apps in myriad ways, and many users can't handle the new workflow for actively trusting apps. So iOS natively trusts any app installed by MDM solutions, which are exclusively used by businesses. In fact, an app installed by an MDM will not show any indication of its origin.

MOBILE DEVICE MANAGEMENT

MDM solutions are often used by enterprises to support Bring Your Own Device (BYOD) programs, in which the company allows personal devices to be used to access corporate email and other corporate services. MDM is a central management tool that enables enterprises to manage policies on the devices used by their employees. Actions they can take include deploying security policies, remote wiping of lost or stolen devices, installing applications, and more.

However, MDMs can also be exposed to Man-in-the-Middle (MitM) attacks. These attacks can allow easy installation of malicious enterprise apps over-the-air, because Apple gives apps installed using MDMs a free pass from heightened security measures.

Malicious MDM-distributed apps can be abused by using the following process:

1. Install a malicious iOS configuration profile. This is a native way to distribute a set of configuration settings like networking, security settings, root CAs, and more. A threat actor can craft a configuration profile that will install a root CA and route traffic through a VPN or a proxy to a malicious server, and then initiate a MitM attack. This configuration could be deployed using phishing attack.
2. Set up a remote enterprise app server to serve the malicious app.
3. Wait for a command to be sent to an iOS device by an MDM: then, using a MitM attack, intercept and replace the command with a request to install a malicious app. The iOS device will fetch from the remote enterprise app server and install it.
4. Execute commands using the malicious enterprise app which, because of the method used to install it, does not require explicit user trust. This means that users will not be able to distinguish between a legitimate enterprise app, an App Store app, or a bogus app installed by a threat actor.

CONCLUSION

1. Apple's iOS ecosystem has certain flaws in the enterprise app installation process that can allow unverified code to be introduced into the iOS ecosystem.
2. Non-jailbroken devices are exposed to attacks by a threat actor using bogus enterprise apps.
3. Enterprises cannot rely on an end user's judgment in BYOD environments, since doing so introduces significant risk of exposure to malicious apps.
4. Enterprises should implement solutions that provide a clear way to view and assess the risk of malicious enterprise apps on mobile devices.

"IN-HOUSE APPS ARE NOT SUBMITTED TO THE APP STORE AND ARE NOT REVIEWED, APPROVED, OR HOSTED BY APPLE. YOU CAN DISTRIBUTE IN-HOUSE APPS EITHER BY HOSTING YOUR APP ON A SIMPLE INTERNAL WEB SERVER OR BY USING A THIRD-PARTY MDM OR APP MANAGEMENT SOLUTION."

- iOS Deployment Overview for Enterprise

Check Point SOFTWARE TECHNOLOGIES LTD.

Learn More About
Check Point Mobile Threat Prevention

Request A Demo