Guide to the Secure Configuration of Apple iOS 6

Status: **DRAFT** (as of 2013-12-31-00:00)
Version: 0.5

SCAP-on-Apple Project

Description

This guide is designed to provide comprehensive discussion of security-relevant configuration settings for Apple iOS 6. Providing system administrators with such information enables them to securely configure systems under their control in a variety of network roles. This guide also provides policy makers with a comprehensive catalog of settings, from which security baselines can be constructed. The XCCDF format enables granular selection and adjustment of settings, and their association with OVAL and OCIL content provides an automated checking capability. Transformations of this document, and its associated automated checking content, are capable of providing baselines that meet a diverse set of policy objectives.

Notice

Do not attempt to implement any of the settings in this guide without first testing them in a non-operational environment. The creators of this guidance assume no responsibility whatsoever for its use by other parties, and makes no guarantees, expressed or implied, about its quality, reliability, or any other characteristic.

Contents

1 Introduction

Purpose. This document provides security-related usage and configuration recommendations for Apple iOS devices such as the iPhone, iPad, and iPod touch. This document does not constitute government policy, nor is it an endorsement of any particular platform; its purpose is solely to provide security guidance. This guide may outline procedures required to implement or secure certain features, but it is also not a general-purpose configuration manual.

Audience. This guide is primarily intended for network/system administrators deploying Apple's iOS devices or supporting their integration into enterprise networks. Some information relevant to IT decision makers and users of the devices is also included. Readers are assumed to possess basic network and system administration skills for Mac OS X or Microsoft Windows systems, and they should have some familiarity with Apple's documentation and user interface conventions.

Scope. Apple's mobile devices, including the iPhone and iPad, are prominent examples of a new generation of mobile devices that combine into a single device the capabilities of a cellular phone, laptop computer, portable music player, camera, audio recorder, GPS receiver and other electronics. The capabilities of such devices are considerable but, as with any information system, also pose some security risks. Design features can seriously mitigate some risks, but others must be considered as part of a careful, holistic risk decision that also respects the capabilities enabled by such devices.

Security guidance for mobile devices must cut across many previously discrete boundaries between technologies. For example, scrupulous deployment of an iPhone includes consideration not just the settings on the device itself, but those of the Wi-Fi networks to which it will connect, the VPNs through which it will tunnel, and the servers from which it will receive its configuration. This guide provides recommendations for the settings on an iOS device itself, as well as closely-related information for the network and configuration resources upon which deployment of iOS devices depends.

1.1 How to Use This Guide

1.1.1 Read Sections Completely and in Order

Each section tends to build on information discussed in prior sections. Each section should be read and understood completely. Instructions should never be blindly applied. Relevant discussion may occur immediately *after* instructions for an action, so be sure to read the whole section before beginning implementation. Careful consideration is essential for deploying iOS devices in an enterprise environment where multiple supporting devices and software components may need to be configured properly in order to function.

1.1.2 Understand the Purpose of this Guidance

The purpose of the guidance is to provide seurity-relevant configuration recommendations. It does not imply the suitability or unsuitability of any product for any particular situation, which entails a risk decision.

1.1.3 Limitations

This guide is limited in its scope to security-related issues. This guide does not claim to offer comprehensive configuration guidance. For general configuration and implementation guidance refer to other sources such as Apple.

1.1.4 Test in Non-Production Environment

To the extent possible, guidance should be tested in a non-production environment before deployment. Ensure that any test environment simulates the configuration in which the devices will be deployed as closely as possible.

1.1.5 Formatting Conventions

Commands intended for shell execution, file paths, and configuration file text, are featured in a `monospace` `font`. Menu options and GUI elements will be set in a **Bold, sans-serif font**. Settings appropriate to the device itself will be typeset in-line (i.e. **Settings > Airplane Mode**). Actionable instructions are typically embedded in a box.

1.2 General Principles

The following general principles motivate much of the advice in this guide and should also influence any configuration decisions that are not explicitly addressed.

1.2.1 Encrypt Transmitted Data Whenever Possible

Data transmitted over a network, whether via wire or wirelessly, is susceptible to passive monitoring. Whenever practical mechanisms exist for encrypting this data-in-transit, they should be applied. Even if data is expected to be transmitted only over a local network, it should still be encrypted if possible. Encrypting authentication data, such as passwords, is particularly important. iOS's support for SSL, WPA2, and IPsec protocols demonstrates its capabilities, when such features are activated, to adhere to this principle.

1.2.2 Encrypt Stored Data Whenever Possible

Data on mobile devices is particularly susceptible to compromise due to loss of physical control. Whenever practical solutions exist, they should be employed to protect this data. The Data Protection API on iOS devices is used by some applications, and demonstrates the devices' capability to provide such protection. Drawing on applications which use this capability (and ensuring that internally-developed enterprise applications also use it), and setting an appropriately complex passcode, follow this principle.

1.2.3 Minimize Software to Minimize Vulnerability

The easiest and simplest way to avoid the vulnerabilities in a particular piece of software is to avoid installing the software altogether. Hundreds of thousands of 3rd-party applications, or "apps," written by thousands of different developers are available for iOS devices. These developers may have willfully or accidentally introduced vulnerabilities. For some environments, a particular app may fulfill a mission-critical need. In other cases an app might needlessly introduce additional risk to the system. Certain risk scenarios may call for minimizing apps. BYOD scenarios, on the other hand, generally imply the consideration and acceptance of such risks.

1.2.4 Leverage Security Features, Never Disable Them

Security features should be effectively used to improve a system's resistance to attacks. These features can improve a system's robustness against attack for only the cost of a little effort spent doing configuration. For example, iOS's enforcement of code signing of apps provides assurance of integrity both during installation and at runtime. Disabling this feature through the use of "jailbreaking" tools provided by the hacker community significantly decreases an iOS device's resistance to attack.

1.2.5 Grant Least Privilege

Grant the least privilege necessary for users to perform tasks. The more privileges (or capabilities) that a user has, the more opportunities he or she will have to enable the compromise of the system (and be a victim of such a compromise). For example, a configuration profile can disallow use of the Safari web browser and the camera. Disabling the camera prevents a malicious or careless user from photographing sensitive areas, while disabling Safari helps ensure the user is protected from any web-based attacks (albeit at significant loss of capability). Similarly, it is possible to restrict the installation of third party apps, and this may be the right balance between security and functionality for some environments.

1.3 Risks, Mitigations, and Consequences

Understanding the risks – and available mitigations – involved in the deployment of smartphone platforms such as iOS provides a background for certain risk decisions. An attacker who has compromised any mobile device, and can remotely maintain control of that device, can use this access to gather a great deal of information about the user of the device and his or her environment. As described by NIST Special Publication 800-124, the consequences of such attacks include:

- collecting audio ("hot-microphone" eavesdropping)
- using the cameras
- geolocation of the device (and presumably the user)
- collecting all data, including credentials stored on the device or accessed by it
- acting as the user on any network to which the device later connects.

The following table describes risks (with attack vector) along with applicable mitigations that are either built into the iOS platform or can be employed by administrators or users. The following table is provided as a summary for risk decision makers - and to motivate administrators to apply mitigations whenever practical. *It should not be used to draw comparisons between iOS and other platforms.*

Risk	Mitigation
Data Compromise due to Lost Device (still reachable over any network interface --- cellular or WiFi)	Enabling a Passcode provides protection for Apps that leverage the Data Protection API, such as Apple's Mail app and 3rd party apps, as well as for credential storage in Keychains. Using the latest hardware currently prevents usage of public Jailbreak tools to access other data on a lost device. Activating a remote wipe can be performed via ActiveSync, MDM, or iCloud. Find My iPhone or other geolocation could permit the lost device to be located.
Data Compromise due to Lost Device (not reachable over any network interface)	Enabling a Passcode encrypts some data on the device. Using the latest hardware currently prevents usage of public Jailbreak tools to access other data on a lost device.
Data Compromise due to Casual Access Attempt	Enabling a Passcode prevents a casual snoop from accessing the device. Provide user training to stress importance of physical control at all times.
Data Compromise via Host Computer Backup/Sync	Ensure proper hygiene and configuration of systems used for backup/sync. This may entail enterprise rollout of iTunes, to ensure protection of backup data. Train users not to connect their device to any untrusted computers/devices and provide additional AC outlet chargers. Encrypting iOS device backups in iTunes can mitigate data loss if the host computer is later compromised or lost.
Exploitation of Device via Malicious app	The Sandboxing feature prevents apps from carrying out certain malicious activities. The App Store's app vetting process provides accountability for developers, which discourages creation of malicious apps. Disabling the App Store, or permitting only installation of Enterprise-created Apps, further mitigates any threat from 3rd party app developers (at significant cost to capability).
Exploitation of Device via Malicious WiFi Network	Apply software updates. Provide user training on connecting only to trusted networks. Provide user training to encourage use of the VPN.
Exploitation of Device via Bluetooth Communications	Apply software updates. Monitor compliance with MDM software. iOS only implements a small subset of the available BlueTooth profiles, which decreases its likelihood to contain vulnerabilities that would give rise to exploitation.

Exploitation of Device via Cellular Network (e.g. SMS/MMS, baseband communications)	Apply software updates. Monitor compliance with MDM software. Provide user training to ensure awareness during travel.
Exploitation of Device via Malicious Email or Web Page	Apply software updates, with particular vigilance after public release of jailbreak tools. Monitor compliance with MDM software.

2 Configuration Deployment

This chapter presents information about creating and deploying settings to iOS devices, which are generally contained in configuration profiles. Configuration profiles are simply XML files that conform to Apple's XML DTD and the plist format. Additional information is available at **http://www.apple.com/iphone/business/it-center/**.

2.1 Nature of Configuration Profiles

Understand that a user who controls an iOS device can opt to erase the device, which erases all data from the device including any configuration profiles. Understand also that users can typically append further restrictive settings, as well as services, onto the device, even in the presence of a configuration profile. Configuration Profile settings enforcement on the iOS devices are cumulative indicating that they can further restrict existing settings when applied.

iOS configuration profiles specify a collection of settings that can control some security-relevant behavior of an iOS device, but are not designed to provide an enterprise with total, arbitrary control over the user's device.

A "carrot and stick" approach can be employed to avoid tempting users to remove a configuration profile (either directly or via device reset). Bind "carrots" (such as credentials needed for enterprise access) to "sticks" (such as a passcode policy) in a single configuration profile. Removing a configuration profile implies that credentials necessary for accessing enterprise services (such as VPN certificates or e-mail accounts) would also be lost, and thus deny the user such services. Also in this case, MDM software would become unable to query the device and the enterprise would be alerted as to the device's unmanaged status.

2.2 Mobile Device Management Software

Third-party MDM products, as well as Apple's own OS X Server, can automate the deployment of configuration profiles and carry out the operational management of devices. Configuration profiles can also be provided via secure web-based services. Configuration profiles can also be created using Apple's iPhone Configuration Utility ("iPCU"), but it does not provide mechanisms for automated deployment or reporting. iPCU provides a convenient means of surveying the settings which can be deployed to devices, although there is no guarantee that a particular MDM product will support all settings.

2.2.1 Select Mobile Device Management (MDM) Software

Select an MDM product which uses Apple's MDM API, unless enterprise management of the devices is not needed.

Apple's MDM API provides the supported mechanism for enterprise device management, and various 3rd party vendors have built products upon it. For more information, see **http://www.apple.com/iphone/business/it-center/deployment-mdm.html**.

2.2.2 Understand Capabilities of MDM Software

Mobile device management software may also include features that are not part of the supported Apple MDM API:

- "Jailbreak detection" can determine if a user has chosen to jailbreak his or her device, which is a useful feature for enterprises who monitor compliance. However, it does not provide high assurance that a device has not been maliciously jailbroken by a sophisticated attacker. The situation is analogous to "root detection" on another mobile platform. It is also analogous to the historical and difficult problem of rootkit detection on desktop or server operating systems. In all these cases, the operating system itself becomes compromised. Since it alone operates at the most privileged levels, there are limits to the extent to which any add-on software can "ask a liar if he is lying."

  The system's cryptographically-verified boot process, runtime enforcement of code signatures, app sandboxing mechanism, controlled software distribution model via "app stores", and rapid software update capability very strongly address the problem of jailbreak-based attacks by themselves. Using add-on software to query for signs of jailbreak may provide an additional layer of defense.

- "Secure containers" can provide data-at-rest protection and data-in-transit protection. These are typically software libraries included by 3rd party apps, which then make use of their functionality instead of that provided by the system's software libraries. These "containers" (which are really just apps) can be useful if the system's capable, built-in mechanisms which already provide these features do not meet particular requirements. Note, however, that they cannot protect their contents against privileged code running on the device, such as would result from a sophisticated, malicious jailbreak attack during system operation.

  They should also *not be confused* with the Sandboxing feature of the iOS kernel as described in [Blazakis]. Rather, the Sandboxing feature strongly addresses the problem of malicious or co-opted apps trying to perform undesirable activities on the system (such as elevating their privileges) in the first place. Sandboxing constitutes a significant obstacle to attackers, but it does not allow apps to (rather inconceivably) protect themselves if the underlying operating system is compromised. App sandboxing may serve as a means of jailbreak detection as discussed above, in that an app which can access beyond its Sandbox may infer that it is running on a compromised device.

2.3 Deploying Configuration Profiles

After a configuration profile is created — typically in an MDM console — it must be deployed to devices. This section discusses methods available for installing a configuration profile onto an iOS device, along with their security implications.

Customizing profiles to individual users implies embedding sensitive authentication information within transmitted profiles. This introduces a need for confidentiality during transmission of such files.

2.3.1 Deploy Over-the-Air with Encryption and Authentication

If configuration profiles will be deployed over-the-air, ensure use of authentication and encryption.

If the iPhone can authenticate a configuration profile during its installation, the **Settings > General > Profile** screen will display **Verified**.

Over-the-air deployment that is authenticated and encrypted requires the support of enterprise infrastructure, such as directory services, an enterprise certificate trusted by iPhone, an SCEP server, and a web server. The server component of MDM products may provide some of this infrastructure.

Deploying configuration profiles to a device over-the-air consists of three major steps:

- *Authentication* of the user, typically leveraging existing directory services.
- *Enrollment*, which involves the device transmitting device-specific information to the enterprise, and receiving a device certificate in return.
- *Installation* of an encrypted, authenticated configuration profile onto the device.

Some MDM products include a server component that provides a web-based service for users to initiate this process, while others initiate the process by requiring users to download a particular MDM client app from the App Store which can facilitate the process.

Transmission and data formats used by the MDM protocol are thoroughly standards-based. Detailed, authoritative description of the transactions between the device and the enterprise are available to Apple-registered developers at **https://developer.apple.com**.

Additional description and security analysis is available in **https://github.com/intrepidusgroup/imdmtools/blob/master/Presentations/InsideAppleMDM_ShmooCon_2012** linked from **intrepidusgroup.com/insight/2012/01/changes-to-apple-mdm-for-ios-5-x/**.

2.3.2 Manual Deployment with iPhone Configuration Utility

Manually using the iPhone Configuration Utility (iPCU) is the safest means of deploying configuration profiles to devices, but does not scale well as it depends on administrators' manual intervention. It also implies that an MDM server will not be used to remotely monitor device status. Nevertheless, transferring the profile to a device which is directly connected via USB cable avoids threats to confidentiality and integrity inherent in network transfers.

iPCU is available at **http://www.apple.com/support/iphone/enterprise/** (cryptographic checksum unavailable). Documentation is available at **http://help.apple.com/iosdeployment-ipcu/**.

2.3.3 Avoid Unauthenticated, Unencrypted Deployment Methods

Avoid deployment of configuration profiles through methods that do not provide encryption and authentication. Email is especially discouraged.

It is possible to distribute configuration profiles to individual devices by emailing the profile to the user of the device or providing a link via SMS. Once the profile is accepted by the end user, the iOS device facilitates its installation. These methods are not recommended because they do not generally provide authentication of the sender of the configuration profile, or encryption of the profile itself during transmission. Users should generally be taught not to have confidence about the origin of email attachments or SMS messages.

Emailing configuration profiles also presupposes that the user has configured an email account on the iPhone.

Furthermore, once the configuration profile is in the receiver's mailbox, it will remain there until it is manually deleted. If the mobile profile contains sensitive information, its prolonged existence in an unmanaged mailbox poses additional risk.

3 System Settings

3.1 Passcode

The remarkable attention paid to passcode quality requirements represents misplaced priorities in the current network environment, as passcodes do not protect against many contemporary threats. However, setting a passcode enables cryptographic features that can protect data on the device if it is lost, stolen or temporarily out of possession. Hardware and software cryptographic features of iOS devices – not present on typical desktop or server systems – provide significant protections against the password-guessing threat when the passcode feature is enabled. Furthermore, iOS devices are likely to store only a single user's credentials, while complex passcode policies are designed to protect against the compromise of large numbers of credentials when they are stored on a single system that becomes compromised (such as a directory server). Onerous passcode policies may also drive users to attempt to subvert the settings. For these reasons, IT decision makers should understand that onerous passcode policies on iOS devices provide little value (in the best case), and may end up being counterproductive. The following publicly-available research provides rationale for these recommendations:

- *Apple iOS 4 Security Evaluation* by Dino Dai Zovi at: **http://www.trailofbits.com/resources/ios4_security_evaluation_slides.pdf**. The slides' section

"Attacking Passcode" provides highly relevant platform-specific discussion for iOS devices.
- NIST Special Publication 800-118 *DRAFT Guide to Enterprise Password Management* at **http://csrc.nist.gov/publications/PubsDrafts.html#SP-800-118** provides discussion about factors that should be considered in order to create effective password strength recommendations.

3.1.1 Enable Passcode

The following passcode settings are recommended, and can be deployed via a configuration profile.

3.1.1.1 *Disable Simple Value for Passcode*

Set **Allow simple value** to **Unchecked**.

Disabling the use of simple values for a passcode enables display of the entire keyboard for passcode entry, instead of only a numeric keypad.

References: IA-5(1)

3.1.1.2 *Require Alphanumeric Value for Passcode*

Set **Require alphanumeric value** to **Checked**.

Requiring alphanumeric values should increase the search space in a password-guessing attack.

References: IA-5(1)

3.1.1.3 *Set Minimum Passcode Length*

Set **Minimum passcode length** to a value.

Setting a minimum length should increase the search space in a password-guessing attack. A passcode length between 4 and 6 may be a reasonable balance between user experience and security for many deployment scenarios.

References: IA-5(1)

3.1.1.4 *Set Minimum Number of Complex Characters*

Set **Minimum number of complex characters** to a value.

Requiring complex characters should increase the search space in a password-guessing attack.

References: IA-5(1)

3.1.1.5 *Set Maximum Passcode Age*

Set **Maximum passcode age** to a number of a days, if there is a need for such rotation.

Changing passcodes prevents an attacker who has compromised the password from re-using it to regain access. This is an unlikely scenario, but is addressed by setting a password expiration. A value of 120 days is likely to be adequate for many deployment scenarios.

References: IA-5(1)

3.1.1.6 *Set Auto-Lock Time*

Set **Auto-Lock (in minutes)** to a number of minutes. A value of 5 minutes may be a reasonable balance between user experience and security for many deployment scenarios.

Automatically locking the device after a period of inactivity ensures that the device will require passcode entry if lost or left unattended.

3.1.1.7 *Set Passcode History*

Set **Passcode History** to **3**.

Setting a passcode history ensures that users cannot trivially alternate between passcodes.

References: IA-5(1)

3.1.1.8 *Set Grace Period for Device Lock*

Set **Grace period for device lock** to 5 minutes, or less.

This permits unlock of the device after a certain period of time has passed since the last device lock. Allowing a Grace Period enhances usability and makes users more likely to tolerate passcode requirements.

References: IA-11

3.1.1.9 *Set Maximum Number of Failed Attempts*

Set **Maximum number of failed attempts** to 10 attempts, or fewer.

Setting the device to automatically erase after a number of failed attempts can protect against witless password guessing attacks conducted through the unlock screen.

References: IA-5(1)

3.1.2 Understand Which Files are Protected by Encryption

Enabling a passcode activates the Data Protection feature of iOS. The Data Protection feature encrypts items with a key whose availability depends on entry of the passcode. Currently, the following items are protected:

- **Email messages** stored by the built-in Mail app
- **Inactive Apps' Screenshots** displayed at app re-launch to create impression of "instant resume"
- **Some Keychain Items** such as email passwords and iTunes backup password
- **Data stored by third-party apps** which use the Data Protection API

In fact, the rest of the files on the device are encrypted, but they are still available to an attacker who can get privileged code to execute on the device. This is because the encryption key for these files is available even without the passcode (unlike the files above). This was possible for a time using publicly-available tools which provide the ability to execute privileged code on any device in physical possession. Examples of these tools include jailbreak software such as *evasi0n* from the evad3rs and *redsn0w* from The iPhone Dev Team. These applications leverage a collection of exploits and other jailbreak utilities including *limera1n* by George Hotz (geohot). Note also that even if privileged code can be run by an attacker on a lost or stolen iPhone, a password-guessing attack against the protected files must be executed on the device itself. This is because the key which encrypts the items listed above is derived from the passcode as well as a key that is bound to the hardware of the device (and not considered extricable from it). The following references provide detailed explanation:

- *iPhone data protection in depth* by Jean-Baptiste Bédrune and Jean Sigwald (Sogeti ESEC) available at **http://esec-lab.sogeti.com/dotclear/public/publications/11-hitbamsterdam-iphonedataprotection.pdf** linked from **http://esec-lab.sogeti.com/post/iOS-5-data-protection-**

updates.

- *Apple iOS 4 Security Evaluation* by Dino Dai Zovi (Trail of Bits) available at **http://trailofbits.files.wordpress.com/2011/08/apple-ios-4-security-evaluation-whitepaper.pdf** linked from **http://trailofbits.wordpress.com/2011/08/10/ios-4-security-evaluation/**.

3.2 Restrictions

Some security-relevant restrictions can be placed upon the user of the iOS device.

3.2.1 Device Functionality

3.2.1.1 *Disable Installation of Third-Party Apps*

Unless necessary, disable **Allow installing apps**. This is unusual for a general-purpose device.

While iOS includes features such as sandboxing that are designed to prevent third-party apps from influencing the integrity of the operating system, they do have the ability to access data stored on the device such as Address Book, Location Data, or the Photo Library, and have the ability to transmit this information.

References: CM-11(2)

3.2.1.2 *Disable Camera*

Disable **Allow use of camera**, if concerns exist about capturing sensitive images.

References: SC-42

3.2.1.3 *Disable Screen Capture*

Disable **Allow screen capture**, if concerns exist about storing screen contents in the Photo Library.

While unlikely, this feature could accidentally be triggered (by simultaneously pressing the Home and Sleep buttons) and lead to storage of sensitive information outside the intended storage area.

3.2.2 Applications

3.2.2.1 *Disable use of iTunes Store*

Deselect **Allow use of iTunes Store**, to prevent access to media content and applications.

iTunes provides access to media content and applications.

3.2.2.2 Safari

The Safari web browser is included with iOS, and the following section provides discussion of security-relevant settings. Additional information about Safari security settings on iOS is available at **http://support.apple.com/kb/HT1677**.

3.2.2.2.1 *Disable Safari*

If Safari can be disabled, uncheck **Allow use of Safari**. ***This is very unusual for a general-purpose device.***

Devices in specialized use cases may not require the use of a web browser such as Safari.

3.2.2.2.2 *Disable Safari Autofill*

Deselect **Enable autofill**, to prevent storage of some potentially sensitive information by Safari.

Storage of autofill information permits an attacker to harvest previously-used credentials in the event that the device becomes compromised.

3.2.2.2.3 *Enable Safari Fraud Warning*

Select **Force fraud warning**, to ensure users are warned when visiting known-fraudulent web sites.

Fraudulent web sites may try to collect sensitive information from users.

3.2.2.2.4 *Disable JavaScript*

Deselect **Enable javascript**, to prevent JavaScript from running in Safari. *This makes the Internet virtually useless and is only appropriate for unusual, high-security scenarios*.

JavaScript permits client-side scripts to run within the browser. Nearly all modern web sites use JavaScript, though its functionality constitutes an attack surface.

3.2.2.2.5 *Enable Safari Pop-up Blocking*

Set **Block pop-ups** to **Checked**.

Blocking pop-ups prevents malicious or obnoxious web sites from interfering with the user in an unexpected and possibly security-relevant manner.

3.2.2.2.6 *Accept Cookies from Visited Sites Only*

Set **Accept cookies** to **From visited sites**.

Accepting cookies from only visited sites decreases the extent to which third-party web sites can track user activity.

3.2.3 iCloud configuration

Policies regarding the usage of cloud-based storage services continue to evolve, as do the assurances of safety by cloud providers. In general, if there is a need to store potentially sensitive information on the iOS device, then the following restrictions should be considered.

3.2.3.1 *Disable iCloud Backups*

Set **Allow backup** to **Unchecked**.

Backing up iOS devices to iCloud may involve placing enterprise data on systems that are not controlled by the enterprise.

3.2.3.2 *Disable iCloud Document Sync*

Set **Allow document sync** to **Unchecked**.

Synchronizing documents to iCloud involves placing data on systems that are not controlled by the enterprise.

3.2.3.3 *Disable iCloud Photo Stream*

Set **Allow Photo Stream** to **Unchecked**.

If photographs may constitute sensitive enterprise data, then transferring photos to iCloud could involve placing such data on systems that are not controlled by the enterprise.

3.2.4 Security and Privacy

3.2.4.1 *Disable Sending Diagnostic Data to Apple*

Disable **Allow diagnositic data to be sent to Apple** to Unchecked, if this presents concerns about inadvertent transmission of sensitive data.

Diagnostic data could be considered sensitive in some environments and not transmitted outside enterprise control.

3.2.4.2 *Disable User Acceptance of Untrusted TLS Certificates*

Set **Allow user to accept untrusted TLS certificates** to Unchecked. Root CAs trusted by iOS are available at **http://support.apple.com/kb/HT5012**.

Presentation of an untrusted TLS certificate to client software could indicate a man-in-the-middle attack.

3.2.4.3 *Force Encrypted Backups*

Set **Force encrypted backups** to Checked, in order to ensure that the contents of iOS devices are encrypted when backed up onto host computers.

Encrypting device backups protects hosts backups even if the host later becomes compromised.

3.3 Wi-Fi

iOS devices support 802.1X authentication for WPA2 Enterprise networks, and this is strongly recommended. A RADIUS server is required for 802.1X authentication and typically involves the use of public key infrastructure. User education and training is also important, since the user has control over the device's network settings. Section contains information for users. DoD Instruction 8420.01, available at **http://www.dtic.mil/whs/directives/corres/pdf/842001p.pdf**, provides information for DoD entities regarding the configuration and deployment of WiFi networks.

3.3.1 *Disable Auto-Join for Wi-Fi*

Ensure that **Auto Join** is disabled for WiFi networks.

Disabling auto join ensures that users are aware of when connections to WiFi networks are being made.

3.3.2 *Use WPA / WPA2 Enterprise for Wi-Fi Encryption*

Using **WPA / WPA2 Enterprise** with **TLS** for authentication is recommended. If TLS support is not available, **PEAP** is the next best choice for authentication. All other authentication protocols are not recommended.

Use **WPA / WPA2** if authentication support is not available. Proxy servers can be configured with WiFi as another layer for providing control of the connection.

3.4 VPN

VPN connectivity obviously depends on an enterprise's available infrastructure, but VPNs which use IPsec are preferred. Several SSL VPNs are also supported by iOS. Actual VPNs are preferred over SSL VPNs as they are designed to protect systemwide network communications. Note, however, that at this time iOS VPNs cannot be configured to route all traffic through a VPN, and operate in split tunnel mode. This behavior occurs even if **Send All Traffic** is selected as part of any VPN's configuration.

3.4.1 *Select IPsec (Cisco) or L2TP for Use as VPN*

Select **IPsec (Cisco)** or **L2TP** (which also uses IPsec) for use as the Connection Type if possible. Use of hardware tokens is generally preferred over passwords for user authentication.

Apple provides documentation regarding iOS VPN capabilities in the following documents:

- *VPN Server Configuration for iOS Devices*, available at: **http://help.apple.com/iosdeployment-vpn/**
- *iOS: Supported protocols for VPN*, available at: **http://support.apple.com/kb/HT1288**

The following documents provide recommendations for configuring VPNs in an enterprise infrastructure:

- *Guide to IPSec VPNs* (NIST SP 800-77), available at: **http://csrc.nist.gov/publications/nistpubs/800-77/sp800-77.pdf**.
- *Guide to SSL VPNs* (NIST SP 800-113), available at: **http://csrc.nist.gov/publications/nistpubs/800-113/SP800-113.pdf**.

References: AC-18(1)

3.5 Email

Permitting users to access only enterprise-supported email accounts decreases the risk posed by email-based attacks. It ensures that enterprise-provided countermeasures against email attacks (such as content filters or anti-virus software) can scan email transmitted to the device.

3.5.1 *Prevent Moving Messages between Mail Accounts*

Disable **Allow Move** for all email accounts.

Moving mail directly between accounts could permit the movement of enterprise information to personal email accounts. While this is possible simply through email forwarding, in that case the enterprise email server (when used as the MTA) has at least the possibility of logging such transfers.

3.5.2 *Enable S/MIME Support for Mail if Needed*

Set **Enable S/MIME** to **Checked**, to provide encrypted and authenticated email support. Ensure that transmission of configuration profiles to devices is encrypted and authenticated if S/MIME certificates containing private keys are embedded. The iOS device can also be configured to use an SCEP server to retrieve S/MIME certificates for use with Mail.

S/MIME provides the capability for users to send and receive S/MIME email messages from wireless email devices. S/MIME and digital signatures provide assurance the message is authentic. Without

S/MIME users will not be able to read encrypted email and will not be able to encrypt email with sensitive information.

References: SI-7(6)SC-8

3.5.3 *Enable SSL for Mail Connections*

Ensure **Use SSL** is enabled for all incoming and outgoing email accounts.

SSL provides authentication of the mail server to which the mail client connects, and also encrypts traffic between the mail server and the email client.

References: SI-7(6)SC-8

3.6 Exchange ActiveSync

If your organization employs Microsoft Exchange to manage user accounts and maintain device policies, configuring Exchange ActiveSync will bind the device to the user's Microsoft Exchange account, syncing email, calendars and contacts with the device.

3.6.1 *Prevent moving messages between ActiveSync accounts*

Disable **Allow Move** for all Exchange ActiveSync accounts.

3.6.2 *Allow Mail from this Account Only from the Mail App*

Enable **Use Only in Mail** for all Exchange ActiveSync accounts.

3.6.3 *Enable SSL for ActiveSync Communications*

Ensure **Use SSL** is **Checked** for all Exchange ActiveSync accounts.

3.6.4 *Enable S/MIME Support for ActiveSync if Needed*

Set **Enable S/MIME** to **Checked**, if encrypted and authenticated email support is needed. Ensure that transmission of configuration profiles to devices is encrypted and authenticated if S/MIME certificates containing private keys are embedded. The iOS device can also be configured to use an SCEP server to retrieve S/MIME certificates for use with Mail.

3.7 LDAP

3.7.1 *Enable SSL for LDAP Connections*

Ensure **Use SSL** is enabled if using an LDAP service.

SSL provides authentication of the LDAP server to which the system connects, and also encrypts traffic between the iOS device and the LDAP server.

3.8 Calendars and Address Books

3.8.1 CalDav

3.8.1.1 *Enable SSL for CalDav Connections*

> Ensure **Use SSL** is enabled if using a CalDav service.

SSL provides authentication of the CalDav server to which the iOS device connects, and also encrypts traffic between the CalDav server and the iOS device.

3.8.2 Subscribed Calendars

3.8.2.1 *Enable SSL for Subscribed Calendar Connections*

> Ensure **Use SSL** is enabled if connecting to calendar subscriptions.

SSL provides authentication of servers that provide calender subscriptions, and also encrypts traffic between the calendar subscription server and the iOS device.

3.8.3 CardDav

3.8.3.1 *Enable SSL for CardDav Connections*

> Enable **Use SSL** if using a CardDav service.

SSL provides authentication of the CardDav server to which the iOS device connects, and also encrypts traffic between the CardDav server and the email client.

3.9 Certificates

3.9.1 Credentials

If your organization employs any self-signed certificates, embed them in the configuration profile or use SCEP to distribute. Note that embeddeding any credentials into the configuration profile introduces the need for encryption during profile deployment.

3.9.1.1 SCEP

If your organization will use the Simple Certificate Enrollment Protocol to distribute certificates and configuration profiles, include its settings with the configuration profile. These settings may be handled by MDM products in a manner that is automated, and not tied to individual users.

3.9.1.1.1 *Set a Challenge Password*

> In the **Challenge** field, enter a strong passphrase to be used as a pre-shared secret for automatic enrollment.

References: IA-2(2)

3.10 Mobile Device Management

Some behavior of Mobile Device Management (MDM) software can be configured inside a configuration profile. This includes how much information an MDM server can retrieve from a device, whether an MDM server can update profiles remotely, whether an MDM server can remotely wipe a device, and whether an MDM server can reset a device's password. These settings allow for more fine-grained adjustment between enterprise control versus user control of a device. Some MDM products may not permit administrators to disable some of their capability for querying devices. Information about enterprise deployment using MDM is available from Apple at **http://www.apple.com/ipad/business/it-center/deployment-mdm.html**.

3.10.1 Sign Messages

Set **Sign messages** to Checked.

This setting causes responses generated by the device (in response to commands from the MDM server) to be signed with the device's identity certificate.

3.10.2 Check Out When Removed

Set **Check out when removed** to Checked.

This causes the device to send a message to the MDM server whenever the configuration profile is removed.

3.10.3 Access Rights for Remote Administrators

The following settings control what an MDM server is permitted to query from an iOS device. For an enterprise-owned, enterprise-controlled device, permitting the enterprise administrator to query as much information as possible is appropriate. Some MDM products may simply include these access rights by default and offer options to retrieve less information from the device.

At the same time, querying all of these types of information may not be appropriate even for some enterprise users, and for enterprises that support BYOD scenarios. The terms of any individual organization's "BYOD Contract" with their users is beyond the scope of this document.

3.10.3.1 *Allow Remote Query of General Settings*

In the **Query device for** section, set **General settings** to Checked.

ALlowing an enterprise to be able to query general device settings such as profile names and identifiers permits the enterprise to have insight into the ownership of profiles installed onto the device.

References: CA-9

3.10.3.2 *Allow Remote Query of Security Settings*

In the **Query device for** section, set **Security settings** to Checked.

Allowing an enterprise to be able to query security settings permits the enterprise to have insight into the device's security posture.

References: CA-9

3.10.3.3 *Allow Remote Query of Network Settings*

In the **Query device for** section, set **Network settings** to Checked.

Allowing an enterprise to be able to query network settings permits the enterprise to have insight into the device's network connection capabilities.

3.10.3.4 *Allow Remote Query of Restrictions*

In the **Query device for** section, set **Restriction** to Checked.

Allowing an enterprise to be able to query restrictions enforced on the device permits the enterprise to have insight into whether certain device behavior is configured, such as encrypted backups.

3.10.3.5 *Allow Remote Query of Configuration Profiles*

In the **Query device for** section, set **Configuration Profiles** to Checked.

Allowing an enterprise to be able to query installed configuration profiles permits the enterprise to understand the overall composed configuration applied to the device.

References: CA-9

3.10.3.6 *Allow Remote Query of Applications*

In the **Query device for** section, set **Applications** to Checked.

Allowing an enterprise to query installed applications provides insight into the device's software inventory, which is valuable in incident response.

3.10.3.7 *Allow Remote Query of Provisioning Profiles*

In the **Query device for** section, set **Provisioning Profiles** to Checked.

Allowing an enterprise to be able to query installed provisioning profiles permits the enterprise to be aware of expiry dates for these.

3.10.3.8 *Allow Remote Addition/Removal of Configuration Profiles*

In the **Add / Remove** section, set **Configuration Profiles** to Checked.

Allowing an enterprise to add or remove configuration profiles permits an enterprise MDM server to be able to update these profiles remotely.

3.10.3.9 *Allow Remote Addition/Removal of Provisioning Profiles*

In the **Add / Remove** section, set **Provisioning Profiles** to Checked.

Allowing an enterprise to add or remove provisioning profiles permits an enterprise MDM server to be able to update these profiles remotely.

3.10.3.10 *Allow Remote Addition/Removal of Apps*

In the **Add / Remove** section, set **Apps** to Checked. This applies to managed apps, which are purchased by the enterprise via the Volume Purchasing Program.

Allowing an enterprise to add or remove managed apps provides the enterprise with the ability to easily deploy enterprise applications to devices. The MDM server can also specify that the data in managed apps be removed when the MDM profile is removed, to protect enterprise data.

3.10.3.11 *Allow Remote Addition/Removal of Settings*

In the **Add / Remove** section, set **Settings** to Checked.

Allowing an enterprise to add or remove settings provides agility with regard to individual settings management.

3.10.3.12 *Allow or Disallow Remote Change of Device Password*

In the **Security** section, set **Change device password** to Unchecked or Checked. This entails a risk decision, though Checked is likely to be appropriate for most scenarios.

Most enterprises are likely well-served by permitting the MDM administrator to remotely send a change password command, in order to allow users with forgotten passcodes to maintain access to their devices. This would also permit an enterprise with appropriate authority, and which needed to overcome the Data Protection feature (such as for forensics purposes) the ability to do so.

At the same time, an attacker who has compromised communications between the device and its MDM server (or the MDM server itself), could maliciously send a password-change command to defeat the data-at-rest protection on the devices. This would depend upon an attacker's physical compromise of the device as well as compromise of TLS communications (or the MDM server itself).

3.10.3.13 *Allow Remote Wipe*

Set **Remote wipe** to Checked.

Permitting remote wipe allows the enterprise to remotely wipe an iOS device in the event that it is lost or stolen.

Note also that a layered configuration profile approach which involves specific configuration profiles permitting access to specific services (which can be removed by an MDM server), effectively permits selective removal of access to services (and their local data). This can provide a form of remote wipe that is more appropriate with BYOD scenarios that are incompatible with IT staff wiping entire devices.

3.11 Manually-Configured Device Settings

The following security-relevant settings can be manually applied. These settings depend on the user's control of the device, and thus training users can help them make appropriate choices.

3.11.1 *Disable Wi-Fi, if Practical*

If the iOS device is not to be connected to a Wi-Fi network, disable Wi-Fi. Set **Settings > Wi-Fi > Off**.

Disabling **Ask to Join Networks** will prevent the phone from automatically associating with previously known (but potentially-spoofed) access points without user interaction, and should be disabled whenever possible. Users should be instructed to use only trusted WiFi networks.

References: AC-18(3)

3.11.2 *Disable Bluetooth Manually, if Practical*

To disable Bluetooth, set **Settings > Bluetooth > Off**, when practical.

Leaving Bluetooth enabled can expose the presence of an iOS device, although the device provides visual cues when it is in the Bluetooth "discoverable" mode which allows it to pair with other devices. The

Bluetooth profiles supported by iOS are described at **http://support.apple.com/kb/HT3647**.

References: AC-18(3)

3.11.3 *Disable Location Services, if Practical*

If the ability of apps and web pages to determine the location of the device poses an unacceptable risk, disable Location Services. Set **Settings > General > Restrictions > Privacy: Location Services > Off**. Note also that usage of Location Service can be controlled on a per-app basis, at the user's discretion. Given the utility of location information for some apps (such as Maps), user-determined settings may be most practical.

If an application (such as Maps) wishes to use Location Services while being disabled, the user will be prompted to return to **Settings** to enable it.

3.11.4 *Disable Loading of Remote Images, if Practical*

To disable the automatic loading of images in e-mail, set **Settings > Mail, Contacts, Calendars> Mail: Load Remote Images > Off** , if this is practical.

Automatically loading images in e-mail messages can leak usage information to authors of fraudulent e-mail. It can also provide an opportunity for malicious images to exploit any implementation flaws in complex graphics libraries. At the same time, this may also inhibit viewing of images that are useful.

4 Device Usage and Handling

This chapter provides recommendations on device usage and handling, for both administrators and users. Section provides handling and usage guidance for administrators. These topics include issuing devices, managing and accounting for devices once in users' hands, and effectively educating users on secure device usage. Further, there are important recommendations such as maintaining physical control of the device, not jailbreaking the device, and preventing connections to untrusted networks. This section closes with suggested usage statements that could be provided to users.

4.1 Handling Guidance for Administrators

If the enterprise is planning to procure and distribute devices to users, administrators should establish procedures for this activity. Some items from this section may not apply to BYOD scenarios, however, such as inventory management and prompt retirement of unsupported devices.

4.1.1 Establish a User Training Program

Create or make available training resources to educate users about device security issues and organization policies. Ensure that all device users are aware of risks and properly trained to mitigate them.

Security and policy awareness training reduces the risk of user-originated security compromise. This relates closely to any agreements between the user and enterprise regarding device handling, which should be verified for each user prior to their being issued a device, as described in Section .

4.1.2 Issuing Devices

This section provides recommendations for enterprises issuing iOS devices.

4.1.2.1 Issue Only Supported Devices

Ensure that only supported hardware versions are issued. Supported hardware versions are defined as those that can run the latest version of iOS and receive all updates. To determine this, administrators will need to manually note which systems can be updated whenever security updates are provided.

Sometimes only the current version and the previous version of the iPhone or iPad hardware can run all updates. This suggests that IT planners should anticipate a 2 year (or 3 year, at most) refresh cycle for enterprise-purchased devices.

4.1.2.2 Erase and Reset Devices, if Re-issuing

If re-issuing devices, erase them before distributing them to users.

Use the command **Settings > General > Reset > Erase All Content and Settings** to erase a device. Clearing content and settings returns the device to a stable state and prevents accidental exposure of the prior user's data.

4.1.2.3 Update Device-to-User Registration

Establish a system for attributing individual devices to users prior to issuance. This information must be updated every time a device is issued or transferred. Existing inventory tracking systems or MDM software can enable automation of this process. The following pieces of information from each device can be useful:

- UDID (Unique Device Identifier)
- Serial Number
- IMEI (if equipped with a cellular connectivity)
- Model Number
- Wi-Fi MAC Address
- Bluetooth MAC Address

MDM products may also report this information. This information should be protected accordingly.

4.1.2.4 Verify User Training History

Ensure that users are familiar with the training before receiving a new device, and at regular intervals afterward.

4.1.2.5 Provide Recharging Hardware with Device

Distribute AC power adapters to users when issuing devices and warn users not to connect their devices to unauthorized systems. It may be prudent to distribute additional AC power adapters to remove the temptation to connect the devices to unknown PCs.

Connecting iOS devices to unauthorized systems, even if only intending to recharge the device, presents a security risk. Providing a power adapter, and easy access to replacements and additional adapters, will help combat temptation to connect to other systems. Users should never be left with connecting to a computer as their only option to recharge their device.

4.1.3 Dealing with a Lost or Stolen iOS Device

If an iOS device is reported as lost or stolen, the device should be immediately disabled to prevent unauthorized use or access. The system administrator can issue a remote "Wipe" command to erase all media, data and settings from the device, restoring it to factory settings. Be aware of the circumstances under which issuing a wipe may not be possible, such as keeping a device in Airplane Mode or simply lacking network connectivity.

4.1.3.1 Establish Procedure for Lost or Stolen iOS devices

Establish and test a procedure to issue a wipe command to erase data from a lost or stolen iOS device. Ensure that users are also aware of their responsibilities to report lost or stolen devices. Wipe commands can be issued by an MDM server or by Exchange ActiveSync. Users can also initiate remote wipe using iCloud, if the device is enrolled.

4.1.4 Retire Devices Which Cannot Run Latest OS Version

Immediately retire any devices which cannot run the latest iOS version. This requires vigilance on the part of administrators, to monitor when an update is issued but is not supported on older devices.

iOS updates include both security patches as well as new functionality. Ensure that all iOS device hardware provided and managed by the enterprise can always run the latest iOS. For example, all iPhone 3G devices should be immediately retired, because they cannot run iOS 6.

4.1.5 Monitor Devices Using MDM, Especially for Updates

MDM products enable enterprise integration and reporting for iOS devices. Regularly monitor the status of devices using MDM software and respond accordingly. Particularly important is ensuring that the version of iOS is kept up to date, which implies that all available security updates are installed. Some MDM products include the ability to disable access to enterprise resources if devices are not kept up to date or are otherwise not compliant.

4.2 Handling Guidance for Users

User education is one of the strongest tools an organization can use to minimize the risk of security issues. Educating users helps raise awareness of their actions and helps them understand the reasoning behind policy decisions. This section details physical handling guidance and security policy topics to be reinforced to users through an organization-developed user education program.

4.2.1 Physical Control

Maintain physical control of your iOS device at all times.

All guidance contained in this document depends upon uninterrupted physical control of your iOS device. It is your responsibility to maintain possession of the device. Never leave your iOS device unattended in an insecure location. An unattended device is at high risk for loss, theft, and other forms of compromise that could violate the confidentiality, integrity, or availability of the device and the information contained therein.

4.2.1.1 Surrendering Physical Control

Learn the proper procedure for relinquishing control of the iOS device to another entity.

There are times when physical control of the iOS device must be surrendered, such as when passing through security or customs inspections. The following are possible methods of mitigating potential loss of personal, financial or company information.

- Before entering security or customs checkpoints, power down the iOS device, remove its SIM card using the SIM eject tool or an unwound paper clip, and place the SIM card in a physically separate location such as a bag or your coat pocket.
- Place the device in a clear, tamper evident bag.
- Ensure passcode is enabled.

Organizations may elect to require all of these steps based on their security policy.

4.2.1.2 Notify Security or Administrative Personnel Upon Loss of Physical Control

Obtain the contact information of your System Security Officer (SSO) for use in reporting the loss of physical control of your iOS device and learn which scenarios require SSO notification. If there is any suspicion that a device has been accessed by an unauthorized user, report the incident immediately to the appropriate SSO or administrative personnel.

If a device is lost or stolen, the administrator or SSO should be contacted immediately in order to execute the remote wipe procedure through Microsoft Exchange, and to create a detailed incident report describing the event. Even if you lose control of your iOS device for a period of time but regain it later, it should be inspected for signs of physical compromise by system administration or security personnel. If a compromise is suspected, actions should be taken to sanitize or destroy the device, depending on the sensitivity of the data and severity of the situation in which it was compromised.

4.2.1.3 Be Aware of Your Surroundings

Be aware of the danger of "shoulder surfing," which refers to the ability of others to see your entry or viewing of sensitive information on the phone.

Because anyone nearby can potentially view any information displayed on the device, be wary of your environment when viewing any sensitive information, and particularly wary when entering passwords. Due to obvious physical and user interface constraints, password entry is susceptible to shoulder surfing, whether by observation of finger position or brief display of each character on-screen. Some third-party products may be available to mitigate this risk.

4.2.1.4 Follow Procedures for Secure Areas

Learn the proper procedure for handling your iOS device in a secure area.

If your organization has a secure area for talking about confidential information, you should be educated about the risks of bringing your iOS device into those areas. The following policies may be implemented for device security in secure areas:

- Leave iOS devices outside conference rooms.
- Applications that record audio or video must be removed or their use restricted.
- Ensure the camera on the back of the iOS device is blocked (e.g. opaque tape) to prevent photo or video recording.
- Ensure that all iOS devices, if present, are in airplane mode with Wi-Fi turned off.

4.2.2 Do Not Jailbreak or Unlock Your iOS Device

Jailbreaking is a term that describes the process of modifying the iOS device's operating system, often with the goal of running unsigned code or performing unsupported customizations to the operating system. Unlocking allows users to operate an iOS device on a cellular network it is not authorized to connect to. Unlocking an iOS device requires a jailbroken iOS device first.

Jailbreaking significantly increases the iOS device's susceptibility to compromise. It disables the enforcement of code signatures, a critical security feature. This enables access to a wide range of software with little accountability and minimal vetting. Jailbreaking tools also typically install and activates services that make the device easier to remotely access, such as SSH.

Warning

Do not jailbreak or unlock your iOS device. Doing so makes it much easier for attackers to introduce malicious code onto the device.

4.2.3 Install Software Updates When Available

Install software updates as quickly as possible. Updates can be applied over-the-air, and are indicated by a red circle on the "Settings" app. Apply by following **Settings > General > Software Update**.

Software updates for iOS devices can contain fixes for security vulnerabilities. As these vulnerabilities often become public knowledge when the updates are released, installing updates as soon as they are available is strongly recommended. Supported 3rd party software should not be broken by updates. Prior to iOS 5, applying iOS updates always required the use of iTunes. Migrating from iOS 5 to iOS 6 also requires the use of iTunes.

4.2.4 Connect Only to Trusted Networks

Do not connect your iOS device to untrusted wireless networks.

Connections to untrusted WiFi networks introduce some risks. Attacks on the iOS device, or eavesdropping on the data it transmits, can occur due to use of such networks. Because the user controls the WiFi settings, he or she must understand the risks associated with untrusted wireless networks and behave responsibly. Some organizations have policies that forbid connection to non-enterprise controlled networks. Other organizations forbid or prevent the use of personal devices on enterprise networks.

4.2.5 Email Accounts

4.2.5.1 Consider Risks of Using Personal Email Accounts

Do not add personal email accounts to your iOS device, unless you are comfortable with (or approved for) the additional risk.

Adding personal email accounts implies that personal, non-company data will be transferred to and stored on the device. This likely violates organizational policy with regard to use of company resources for personal use, but it also increases risk. It increases the risk of your personal information being compromised as a result of an attack against the device, and also increases the risks of company information being compromised as a result of an attack carried out against your personal email account. See the next section for more information about phishing attacks and the motivation for segregating email accounts between different systems.

4.2.5.2 Be Aware of Phishing

Be aware of phishing attempts, including receiving mobile profiles from attackers.

Phishing is a term referring to a fraudulent communication (usually email) pretending to be from a reputable source asking for personal, financial or company information. Adding personal email accounts to your iOS device greatly increases your availability to receive phishing emails, which may accidentally release important information about yourself or your organization. By removing personal email accounts from the device, you are protecting your organization from divulging information through your device to these malicious actors.

4.2.6 Disable Bluetooth if Practical

Disable Bluetooth communication if not necessary.

Disabling Bluetooth reduces the possible attack surface for exploitation, although such vulnerabilities are rare and the iOS over-the-air update process enables rapid patching upon any public disclosure. Bluetooth also permits wireless device discovery and can be used to reveal a limited amount of information from the device. If practical, it is safest to keep Bluetooth disabled. The Bluetooth profiles available on iOS devices are documented at **http://support.apple.com/kb/HT3647**.

4.2.7 Recharge Device Only Through Approved Methods

Recharge your device by either connecting to an organization-approved system or by using the AC power adapter you received when you were issued your device.

Connecting your iOS device to unknown systems exposes the device to unnecessary risks, including the loss of personal or company information. Syncing only with trusted systems also helps maintain the integrity of your iOS device.

4.2.8 Do Not Use Unapproved Peripherals on Your iOS Device

Many different types of iOS device accessories are available for use by consumers. Some of these devices allow access to a keyboard, magnetic card reader, or USB flash drive reader. Users should only use accessories approved for use by the enterprise.

Due to the physical access nature of these accessories, security risks may be present when attached to an iOS device.