Defending Data on iOS 7

Version 2.0
Released: February 10, 2014

Securosis, L.L.C. http://securosis.com

Author's Note

The content in this report was developed independently of any sponsors. It is based on material originally posted on the Securosis blog but has been enhanced and professionally edited.

Special thanks to Chris Pepper for editing and content support.

Licensed by Watchdox

WatchDox enables organizations to access, share and control their critical documents wherever they go: on any tablet, smartphone, or PC, even those beyond the IT department's control. Available as SaaS or on premise, the WatchDox document-centric security platform allows organizations to collaborate with partners, adopt BYOD initiatives, and control or wipe their documents remotely, all while providing users an intuitive experience across every device.

WatchDox also tracks document access and use for compliance and visibility. The platform integrates with SharePoint, salesforce.com and other enterprise applications via comprehensive APIs. More than 500 organizations — including top-10 global financial institutions, governments, and Fortune 500 companies — depend on WatchDox to protect their business-critical information.

http://watchdox.com

Copyright

This report is licensed under the Creative Commons Attribution-Noncommercial-No Derivative Works 3.0 license.

http://creativecommons.org/licenses/by-nc-nd/4.0/deed.en_US

Table of Contents

Introduction 4
- Why iOS and Not Android 4
- Information-Centric Security 5

iOS Security and Data Protection 6
- Apple's BYOD Philosophy 6
- Device and OS Security 7
- Device Management 11
- iOS Data Flow 13

The iOS Data Security Spectrum 16
- Unmanaged Devices 17
- Partially Managed Devices 22
- Enterprise-Owned (Supervised) Devices 23

Defining Your iOS Data Security Strategy 25
- Factors 25
- Make a Decision 27
- Conclusion 27

Who We Are 29
- About the Author 29
- About Securosis 29

Introduction

Survey after survey shows growing enterprise adoption of iOS, rivaled only by Android — even aside from the tidal wave called iPad. The phrase "the consumerization of IT" was in circulation before the iPhone, but no other vendor is so effectively driving the adoption of consumer technologies into the enterprise as Apple.

In years past, we in IT security served as the gatekeepers for new technologies in the enterprise. As much as we like to say we are the last to find out about new tools and toys, mobility is one area where we held tight control by restricting access to the network. But in this post-PC consumer-centric world we are losing our ability to stop or slow adoption of consumer technologies, even when they don't support all our enterprise needs.

Two years ago at the RSA Security Conference I asked a group of 150 operational security professionals how many were under pressure to support non-BlackBerry devices. Nearly every hand in the room went up, almost universally to support iOS, and only a relatively small percentage had technical capabilities or policies in place to manage this transition. In a recent meeting with a group of CISOs from mid-sized organizations, nearly all supported iOS to some degree, but none felt fully comfortable with their understanding of the security ramifications. Or that they were adopting the best approach to managing devices.

In nearly every conversation, the key concern was for the safety of data.

The question is no longer *if* or *when* to allow these devices, but *how* to support non-PC computing platforms while safely protecting enterprise data.

In order to stay focused, this report will lay out options for protecting enterprise data on iOS, rather than talking about the myriad other issues around mobile device management.

Why iOS and Not Android

Of course Apple isn't single-handedly driving the consumerization of IT, but the numbers above (and a quick glance around the office) show that the company from Cupertino is a major force. They have done more to drive the smartphone and tablet markets than any competitor. And we are asked about securing iOS for the enterprise more than any other platform.

Until a few years ago BlackBerry was the dominant platform — largely because it was designed specifically to address enterprise needs — so most organizations are comfortable securing those tools. Some organizations also supported Microsoft and perhaps Palm, but one of those companies no longer exists and the other has since completely tossed out its platform to start fresh.

The real activity is with iOS and Google's Android. Pressure has been mounting to support Android phones, but there is far less pressure to support Android tablets, and device fragmentation complicates all those discussions.

iOS is also a stronger platform in terms of security. Nothing is invulnerable but there is essentially no iOS malware and few known security breaches. The software ties strongly into the hardware, and current versions are very difficult to hack. Android, by its more open nature, represents a greater security risk, as demonstrated by ongoing malware issues — still lower than PC levels, but much higher than iOS.

The main problem is that Apple provides different tools for enterprise management of iOS than we are accustomed to. There is no ability to run background security applications, so we need to rely on policy management, the platform's inherent security, and a spectrum of security architectures. Apple has a clear philosophy on how their devices should be managed, and makes it effectively impossible to secure devices without working inside its model.

Information-Centric Security

We are focused on *data* for this series, so we take an information-centric approach. We won't talk about network management or device restrictions that aren't relevant to protecting data, but we will discuss managing data even before it hits the device.

Previously I wrote up some principles of information-centric security:

- Information (data) must be self describing and defending.
- Policies and controls must account for business context.
- Information must be protected as it moves from structured to unstructured, in and out of applications, and between changing business contexts.
- Policies must work consistently through the different defensive layers and technologies we implement.

These sound a bit like typical analyst flimflam, but we actually have the technologies to implement much of it today. When managing data for mobility and iOS, we can hit every one of those points.

This report will show how to manage what data ends up on devices, how to protect it once it's there, and how to build and manage policies to enable users without violating risk tolerances. To accomplish this we will present a spectrum of options designed to satisfy different organizational needs — all supported by existing products, some of which you probably already have.

iOS Security and Data Protection

Before we delve into management options we need to understand the iOS 7 security and data protection models. These controls are built into the platform, and utilized by the various enterprise options we will discuss. We are focused on data but will also cover iOS security basics — they play an important role in data security. iOS 7 adds a plethora of new data security features for the enterprise, arguably more than any previous release.

The short version is that iOS is quite secure — far more than a general-purpose computer. But you need to understand Apple's *security philosophy* to understand their design decisions and your integration options. Apple has a clear vision of the future for BYOD, and it is very different from the way most organizations have managed personal devices in the past.

Note: We are only discussing iOS 7 and later (as of this writing 7.03 is the current version of the iOS operating system for iPhone, iPad, and iPod touch). We do not recommend supporting previous versions of iOS.

Apple's BYOD Philosophy

Apple has a very clear vision of the role of iOS devices in the enterprise. There is BYOD, and there are enterprise-owned devices, with almost completely different models for the two. ***Ownership*** *of the device defines the security and management model*.

In BYOD users own their devices, enterprises own enterprise data and apps on them, and the user experience should never suffer to support this division. No dual personas. No virtual machines. A seamless experience, with data and apps intermingled yet sandboxed. The model is far from perfect today, with one major gap, but iOS 7 is the clearest expression of this direction yet, and only the foolish would expect Apple to change any time soon.

Enterprise-owned devices support absolute control by IT, down to the new device provisioning experience. Organizations can degrade features as much as they want and need but devices still, as much as allowed, provide the complete iOS experience.

In the first case, users *allow the enterprise space on their device, while the enterprise allows users access to enterprise resources*; in the second model, *the enterprise owns everything*. The divide is so clear that it is actually difficult for an enterprise to implement supervised mode on an employee-owned device.

We will explain the specifics as we go along, but here are a few examples to highlight the different models.

- On *employee owned devices*:
  - The enterprise sends a configuration profile that the user can choose to accept or decline.
  - If the user accepts it, certain minimal security can be required, such as passcode settings.
  - The user gains access to their corporate email, but cannot move messages to other email accounts without permission.

  - The enterprise can install *managed apps*, which can be set to only allow data to flow between them and *managed accounts* (email). These may be enterprise apps or enterprise licenses for other commercial apps. If the enterprise pays for it, they own it.
  - The user otherwise controls all their personal accounts, apps, and information on the device.
  - All this is done without exposing any user data (like the user's iTunes Store account) to the enterprise.
  - If the user opts out of enterprise control (which they can do whenever they want) they lose access to all enterprise features, accounts, and apps. The enterprise can also erase their 'footprint' remotely whenever they want.
  - The device is still tied to the user's iCloud account, including Activation Lock to prevent anyone, even the enterprise, from taking the device and using it without permission.
- On *enterprise owned devices:*
  - The enterprise controls the entire provisioning process, from before the box is even opened.
  - When the user first opens the box and starts their assigned device, the entire experience is managed by the enterprise, down to which setup screens display.
  - The enterprise controls all apps, settings, and features of the device, down to disabling the camera and restricting network settings.
  - The device can never be associated with a user's iCloud account for Activation Lock; the enterprise owns it.

This model is quite different from the way security and management were handled on iOS 6, and runs deeper than most people realize. While there are gaps, especially in the BYOD controls, it is a safe bet that they will be slowly cleaned up over time, following Apple's normal iterative improvement process.

It is hard to fully explain the differences without actually covering the security controls and technologies, so let's jump in.

Device and OS Security

No computing device is ever completely secure, but iOS has an excellent track record. There has never been a widespread remote attack or malware used against (non-jailbroken) iOS devices, although we have seen proof of concept attacks and plenty of reported vulnerabilities. This is thanks to a series of anti-exploitation features built into the OS, some tied to the hardware.

Devices may be vulnerable to local exploitation if the attacker has physical access (using the same techniques as jailbreakers), but it has become increasingly difficult on newer iOS devices (the iPhone 4S and iPad 2 and later), and basic precautions can protect data even if you lose physical control.

Let's quickly review the built-in security controls.

> There are dramatic differences in the security of different iOS hardware revisions. Anything earlier than the iPhone 4S or iPad 2 is vulnerable to attacks that circumvent many of the security controls discussed in this paper. But the 4S and iPad 2, as well as later devices including the iPad Air and mini, are much more resilient to attack if they are lost or stolen.

Operating System Hardening

Five key features of iOS are designed to minimize the chances of successful exploitation, even in case of an unpatched vulnerability:

- **Data Execution Protection:** DEP is an operating system security feature that marks memory locations as non-executable, which is then enforced by the CPU itself. This reduces the opportunity for memory corruption attacks.

- **Address Space Layout Randomization:** ASLR randomizes the memory locations of system components to make it difficult for attackers to complete exploitation and run their own code, even if they do find and exploit a vulnerability. Randomizing the locations of system components makes it difficult for attackers to know exactly where to find and execute their exploit code to take over the system.
- **Application Code Signing:** All applications on iOS must be cryptographically signed. Better yet, they must be signed using an official Apple digital certificate, or an official enterprise certificate installed on the device for custom enterprise applications — more on this later. This prevents unsigned code from running on devices, including exploit code. Apple only signs applications sold through the App Store and system updates, minimizing the danger of malicious apps.
- **Sandboxing:** All applications are highly compartmentalized from each other, with no central document/file store. Applications cannot influence each other's behavior or access shared data unless both applications explicitly allow and support such communication.
- **The App Store:** For consumers, only applications distributed by Apple through the App Store can be installed on iOS. Enterprises can develop and distribute custom applications, using a model very similar to Apple's App Store, and such applications only work on devices with the corresponding enterprise digital certificate installed. All App Store apps undergo code review by Apple — this isn't perfect but dramatically reduces the likelihood of a malicious application ending up on a device.

There are, of course, techniques to circumvent DEP and ASLR, but it is extremely difficult to circumvent a proper implementation of them working together. Combined with code signing and additional software and hardware security beyond the scope of this discussion, iOS is very difficult to exploit.

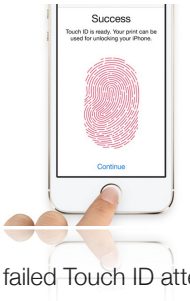
Of course it isn't impossible — we *have* seen exploits (especially local attacks such as tethered jailbreaks and individual application failures), but their rarity, in light of the popularity of these devices, makes clear that these security controls work well enough to thwart widespread attacks. Specifically, we have yet to see *any* malware spread among un-jailbroken iPhones or iPads.

Hardware Security Features

iOS devices also leverage device hardware to enhance security. Apple is tight-lipped about these capabilities, and in a few spots we need to read between the lines. These capabilities improve with each revision of the hardware, beginning with the first hardware encryption in the iPhone 3GS. *Enterprises should only support the iPhone 4S or later, or iPad 2 or later* due to their hardware security features. Many iOS security controls can be circumvented more easily prior to those models.

- **64-Bit Memory Protection:** The iPhone 5S and late 2013 model (Retina) iPad Air and iPad Mini use a 64-bit ARM A7 processor. 64-bit hardware often includes more advanced memory corruption attack protection than 32-bit systems. We know Apple uses these features in the latest versions of OS X for the Mac, and iOS and OS X share a code base. It is likely this feature is also used in 64-bit iOS devices, although *this is educated speculation*.
- **Hardware Encryption Processor:** iOS devices (3GS and later) include a cryptographic accelerator for encryption operations. As we explain below, it is used for both full-device and per-application data encryption. It is also used for VPN, code signing, and other crypto operations.
- **Unique Hardware Device Identifier:** Every iPhone and iPad has a unique device key burned into the hardware that is not recorded by Apple or their suppliers, and is not accessible by anything other than the crypto processor. This identifier is used during device and data encryption, which means that even if a device is physically lost, *any brute force attack must occur on the device*. The device hardware is rate-limited to resist attacks based on imaging the device and attempting a brute force attack on more powerful hardware.

- **Activation Lock:** iOS 7 introduced an anti-theft feature that ties an iPhone's activation to the user's Apple ID account. This combines hardware and software support. Once registered, with Find My iPhone (or iPad/iPad touch) turned on, a device is permanently tied to a user's Apple ID unless they disassociate it. The device cannot be activated until the user signs in with their Apple ID credentials to release it from that account, even after it has been wiped. Enterprises note this also means that IT will be unable to repurpose a device associated with a user's personal Apple ID until the user deactivates Find My iPhone.



- **Touch ID and the Secure Enclave:** The iPhone 5S includes a capacitive fingerprint sensor known as Touch ID (we expect it appear in additional future models). The Touch ID system stores the user's fingerprint templates in the *Secure Enclave* on the A7 processor. We believe this uses a version of the ARM TrustZone architecture for hardware-enforced processing and memory segregation (although Apple will not confirm this), and that the user's passcode (and Apple ID password) are also stored in the Secure Enclave. Touch ID doesn't replace passcodes — it adds a layer on top, allowing users to unlock their devices with their fingerprints. The passcode is still required every time the device reboots, after a number of failed Touch ID attempts, or if it isn't used for 48 hours. *For most organizations, this can be considered reasonably secure and enhances security by allowing a long passcode with the convenience of no passcode during routine use*. But Touch ID should not be relied on in high-security environments. Additionally, Touch ID increases the likelihood of a user forgetting their passcode.

Enterprises should give as much preference as possible to newer hardware; as we mentioned, we don't recommend supporting anything earlier than the iPhone 4S or iPad 2. Even top digital forensics firms and law enforcement are unable to recover protected data on an iOS device with a long passcode, thanks to these features. They need to rely on side-channel attacks, such as recovering an unencrypted or poorly encrypted backup. We will cover which data is protected in a moment.

Configurable Security Features

In addition to fundamental architectural and hardware security controls, iOS 7 includes basic security features that users can configure themselves or employers can manage through policies:

- **Device PIN or Passcode:** The most basic security for any device, iOS supports either a simple 4-digit PIN or full alphanumeric passphrase. Either flavor ties into the Data Protection and device wipe features.
- **Activation Lock:** As mentioned under *Hardware*, when a user enables Find My iPhone their device is permanently tied to their Apple ID until they disable the feature, even if the device is wiped. The Apple ID is distinct from the device passcode — the two are not related.
- **Remote Wipe:** iOS supports remote wipe via Find My iPhone, Exchange ActiveSync, and Mobile Device Management (MDM) APIs. Of course the device must be accessible via the Internet to receive the wipe command.
- **Passcode Wipe:** When a PIN or passphrase is set, if the code is entered incorrectly enough times, the device can erase all user data.
- **Geolocation:** The device's physical location can be tracked using location services, which are part of Find My iPhone and can be incorporated into third-party applications through MDM support.
- **VPN and on-demand VPN:** Virtual private networks can be activated manually or automatically when the device accesses any network service (although not all VPNs support on-demand connection). VPNs can also be activated when a *Managed Application or Account* launches, as we will discuss under *Device Management*.

- **Configuration Profiles:** Many security features — especially those used in enterprise environments — can be managed using profiles installed on the device. These include options far beyond those available to consumers configuring iOS casually, such as restricting which applications and activities the user can access.

These are the core features we will build on as we discuss enterprise management. But iOS also includes data protection features that are the cornerstone of most iOS data security strategies.

Data Protection

Although it was nearly impossible to protect data on early iPhones, modern devices use a combination of hardware and software to provide data security:

- **Device Encryption:** The iPhone 3GS and later, and all iPads, support built-in hardware encryption. All user data can be automatically encrypted in hardware at all times. This is used primarily for wiping the device rather than to stop attacks. Erasing the entire flash storage would be slow, so instead wiping works by destroying the encryption key, which instantly makes all user data inaccessible. Data is encrypted with a device key the OS has full access to, which means even encrypted data is exposed if someone jailbreaks or otherwise accesses the device directly. Hardware encryption is also used to provide some protection against unauthorized physical access.

Basic policies should include the following:

- Require Passcode: After *n* minutes
- Simple Passcode: **OFF**
- Erase Data: **ON**
- Remote Wipe: **ON**

- **Data Protection:** As noted above, hardware encryption is relatively easy to circumvent because it is primarily designed to wipe the device rather than to secure data from attack. To address this requirement Apple added a *Data Protection* option in iOS and made it available to applications with iOS 4. When a user sets a passcode lock all application data is encrypted using the passcode (and the device identifier) as the key. With this enabled, even if the device is physically lost and exploited, any data protected with this feature is encrypted. Prior to iOS 7 only mail, attachments, and applications using the Data Protection API were encrypted, but *in iOS 7 all application data is encrypted by default*. Some data is still less protected with only device encryption, which is subject to exposure if the device is jailbroken. This includes the camera roll, contacts, calendars, reminders, and location data. Attackers may still attempt brute-force guessing attacks against the key, of course, but only on the device hardware itself. Research indicates that passcodes longer than 8 characters are nearly impossible to crack.
- **Backup Encryption:** All iOS devices automatically back themselves up to iTunes or iCloud when they are plugged in (or connected to their linked computer). If backup encryption is enabled, all data is encrypted *on the device* using the designated password before transferring to the computer. Note that if the user sets a weak password this protection might not be worth much; additionally the password may be stored in the iTunes computer's system keychain. The encrypted iOS keychain is included in the backup, when the user has a passcode set.

These capabilities have improved steadily since the first iPhone, and current hardware revisions running iOS 7 with a strong passcode appear unrecoverable, even to law enforcement agencies with physical control of the device. We do suspect some governments have zero-day exploits they can use to hack the device locally or remotely to circumvent encryption, but no device is immune to that sort of thing.

Device Management

iOS 7 introduced significantly more device management options. So many that it almost completely changes the calculus on determining how to best manage iOS devices. As we covered under *Apple's BYOD Philosophy*, there are two approaches to iOS 7 Mobile Device Management:

- *Supervised devices* are owned and managed by the organization. MDM allows complete and absolute control of devices, even before the box is opened.
- *Employee-owned devices* are owned by the employee rather than the enterprise. The organization can grant access to enterprise resources (specifically accounts and apps), and, using MDM, can control their use based on enterprise policies. But there are still gaps in coverage that could allow data to leak, although we expect them to be plugged with future updates.

These updates are so significant that *we do not recommend supporting any version of iOS prior to iOS 7.*

As a reminder, with few exceptions Apple does not allow background apps capable of monitoring or interacting with other apps on a system-wide basis. All apps are heavily sandboxed, with connections only allowed using special techniques that must be built into the apps themselves. There is no ability to deploy antivirus, DLP, or other third-party monitoring technologies. Fortunately the security model itself obviates the need for most of them.

First we will cover device management features for employee-owned devices, and then follow with supervised devices. Supervised devices add a few capabilities to the employee-owned model.

Employee-Owned Devices

The core of managing an employee-owned device comes down to four key features:

- Configuration Profiles, which you deploy to devices to attach them to a Mobile Device Management server and enable management.
- The Mobile Device Management (MDM) framework and protocol for communicating with and managing devices.
- Managed Accounts, which are enterprise email accounts, separate from the user's personal accounts, which the organization can manage through MDM.
- Managed Apps, which are owned and deployed by the enterprise — including App Store apps purchased by the organization using volume licensing.

You use the *configuration profile* to enable *MDM* and then provision *managed accounts* and *managed apps,* which the enterprise can restrict. *If the user removes the Configuration Profile, removing MDM and enterprise control,* ***they lose access to Managed Accounts and Managed Apps!***

Let's walk through these components to see how they fit together and what they mean for managing iOS devices:

Configuration Profiles

Profiles are small files deployed on iOS devices that hook them into the identity management infrastructure. They are generated using the *Apple Configurator* application or an MDM tool, and can be issued via email, web links, or any other way of moving a file around, including iMessage. The key is that *the device owner decides* whether to install the profile.

An installed profile can establish a variety of configuration settings. Everything from passcode requirements to Exchange server settings to VPNs and network restrictions can be managed — along with much more. The organization decides

how deep these requirements will go, as we will discuss later in this report. An installed profile can be remotely updated, assuming the tool or configuration supports it.

Configuration profiles can be locked so users cannot remove them, although there are ways around this if the employee owns the device — such as restoring from a pre-profile backup or resetting the device and starting over. We don't recommend locking profiles for devices which the organization does not own.

Mobile Device Management Protocol and Framework

If configuration profiles are the hooks, MDM is the fishing line tying devices to a management server.

MDM on iOS consists of a framework on the device for running management tasks and an MDM protocol to communicate with the device. Once registered using a configuration profile the device contacts the MDM server, and can be enrolled once the profile and device are validated.

On the device the MDM framework handles enforcement of policies and updates. MDM servers issue updates using Apple's Push Notification Service, which prompts the device to check back with the server for updated settings. MDM only allows control of basic device settings and enterprise accounts and apps — a user's personal accounts and data are equally protected against access by enterprise IT.

The key difference between MDM on iOS, compared to on most other platforms, is that all features and connections are handled by the operating system and Apple. MDM vendors do not need to, nor can they even attempt to, create and install their own device-specific agents. Everyone is on an equal playing field, with the same set of (Apple's) device management features. Additionally, Apple manages communications with devices to trigger policy updates.

Managed Accounts

Accounts (mail/calendar/etc.) installed using MDM are flagged as 'managed' by iOS. They are distinct from user accounts, and are managed by enterprise policies issued and enforced via MDM. The idea is to allow organizations to segregate enterprise data and access from the user's personal use of the device, without infringing on that use.

Control over managed accounts is very granular. For example enterprises can restrict moving an email message from a managed account to a personal mailbox, block backing up managed accounts to iCloud, or set attachments to only open in managed apps (using *Managed Open In*, described below). Conceptually, the user has full freedom with their own accounts, while the enterprise controls managed accounts, all within the standard iOS user experience.

If a user removes the enterprise configuration profile they lose access to managed accounts. This ensures enterprise data is still protected (especially if backup restrictions are enabled). *But there is a major hole in this model.* Currently there are no policies to restrict moving files using AirDrop, or text with copy/paste or supported social media services. Considering the robustness of the framework otherwise, these are glaring omissions.

Managed Apps

Managed Apps are the biggest change to MDM in iOS 7. When installed by an MDM server, these apps are flagged as 'managed' like a managed account. Managed apps can be private corporate apps, or regular apps from the App Store licensed using Apple's *Volume Purchasing Program (VPP).*

Managed apps are installed, updated, and removed using MDM. They support four key features:

- *Managed Open In* restricts what other apps a user can open files in. As of this writing, when enabled a user can only open files from managed accounts and apps in other managed accounts and apps. It is all-or-nothing, without support for restricting different apps with different settings. We discuss this more in under *iOS Data Flow*.

- *Managed configurations* for pushing per-app configuration settings to managed apps, if the app supports it.
- *Per-app VPN* to require a particular VPN when a managed app is opened.
- *App restrictions*, such as disabling iCloud backups.

Apps created and digitally signed by the enterprise are issued using an enterprise app store, MDM, or other download technique. But VPP enables enterprises to bulk purchase any App Store app (supported by the developer) and install it using MDM or manually through Apple's App Store. If the developer enables an app configuration, enterprises can configure the app as needed for the organization. If a user already has the app on their device, they will need to uninstall it and then accept the managed version, because an app cannot be installed twice on one iPhone or iPad (or iWatch — we like to future-proof our papers with popular rumors).

Your MDM tool must support managed apps because it requires a connection to the App Store. The process links the user's iTunes account with an enterprise identifier, without revealing the user's information to the MDM server. If the user removes the configuration profile or is disenrolled from the MDM server, they get 30 days to convert the app to a personal license, or it is automatically removed.

See Apple showing its colors again? The user opts in and gains access to enterprise apps, which are restricted without affecting the rest of the device or user experience. And the user doesn't need to permanently tie their personal account to the enterprise. Remove enterprise restrictions and they lose access to enterprise resources.

Except, of course, for the huge and unexplained AirDrop and sharing hole. But hey, Apple always likes to leave us something to complain about, so they can fix it later.

Supervised Mode

Supervised Mode allows the enterprise complete control of the IOS device, including in the sealed box if your order it pre-configured from Apple. It plugs all the gaps in the employee-owned model, allowing complete control. If you want to block AirDrop, restrict networks, enforce web filtering, and even lock down the wallpaper, this is how you do it.

Supervised mode requires devices pre-enrolled by Apple, or a physical connection to a Mac running the Configurator application. Supervision is tied to a specific Mac, so you need the same physical (or virtual) machine to remove supervised mode. It supports managed apps and accounts and all other MDM features, plus the additional supervision restrictions.

This option is best for limited-use devices such as those in schools, hospitals, kiosks, and warehouse floors. It should never be used on an employee-owned device because it effectively transfers ownership — just as employees should never be able to enroll corporate-owned devices into Find My iPhone/Activation Lock under their personal Apple ID.

iOS Data Flow

We need to take some time to understand how data moves onto and around iOS devices before delving into security and management options.

Data on iOS devices falls into one of a few categories, each with different data protection properties. For this discussion we assume that Data Protection is enabled by setting a passcode, because otherwise iOS provides no real data security. The data categories are:

- Email messages and attachments.
- Calendars, contacts, and other non-email user information.

- Application data.

Data follows one of two flows in iOS 7:

- In the *unmanaged data flow*, data can move from any account to any supported app, without restriction.
- The *managed data flow* restricts data from managed accounts and apps to other managed accounts and apps. If configured, data from unmanaged accounts and apps can also enter the managed flow… and stay there.

Let's take a look at both of these.

Data Protection and Unmanaged Data

The biggest data security changes in iOS 7 are managed accounts and apps, and the fact that all application data is encrypted with data protection by default once a passcode is set (configuring Touch ID automatically configures, and requires, a passcode). Reports from forensics firms indicate that Data Protection on an iPad 2 or iPhone 4S (or later, we presume) running iOS 5 cannot currently be cracked, by other than brute force. Data Protection on earlier devices *can* be cracked.

There are two common ways files find themselves on an iOS device — via email, or downloaded into an app. In both cases the content is strongly encrypted in the app's sandbox. If the user views the file in the app (including Mail) with the built-in viewer or app features, the file is still inside the sandbox. If the user selects "Open In…" and another app, the file is *copied into the other app's sandbox*. There are now two copies (or more) of the file. Our editor goes into seizures every time we write one of these papers on iOS and leave multiple versions scattered all over the place in different apps and services. It can be a version control nightmare.

The only exception is files stored in a shared service like Dropbox. Apps which access Dropbox store local copies in their own private document stores, but other apps can access the same data online to retrieve their own (private) copies. These apps can save versions back to Dropbox, keeping everything in sync as if there was a 'normal' (local) file system, aside from the necessary local duplication. The files are encrypted on the device, but Apple's encryption does not protect the version stored in the cloud — it is local only.

This workflow is specific to email and apps — calendars, contacts, photos, and other system-accessible user information is not similarly protected, and is generally recoverable by a reasonably sophisticated attacker with physical possession of the device. Data in these apps is also available system-wide to any application. It is a special class of iOS data using a shared store, unlike third-party app data.

Because Data Protection is enabled by default for all apps, this is effectively the same as the protected data flow in iOS 6, except that files can never slip through the gap and end up unencrypted on the device. All data strongly encrypted all the time.

The main concern for enterprises, in this model, is that although the data is *protected* it is not *managed*. Users can move files into any app or service on the iPhone or iPad.

Managed Data

If an enterprise enables managed accounts and apps, it can manage the flow of enterprise data on devices. Any files downloaded into a managed account or app can be restricted, using Managed Open In, to only approved apps and accounts. This enables an enterprise, for example, to volume purchase a popular iOS document editor, install it on every device, and allow employees to open attachments in that editor and email them back out, only via the corporate mail account.

As mentioned earlier, this option is all or nothing. Managed apps and accounts can transfer files to any other managed app or account — you cannot lock it down tighter. You also have the option to allow users to open personal files into the managed pathway, at which point those copies are henceforth managed (except for the original copy). Clearly you should back this up with DLP or other server-side data protection, because otherwise the user can just email the file to a personal account, as users tend to do from any computer when not restricted.

And perhaps, by the time we publish this, Apple will close the AirDrop hole to the managed data flow.

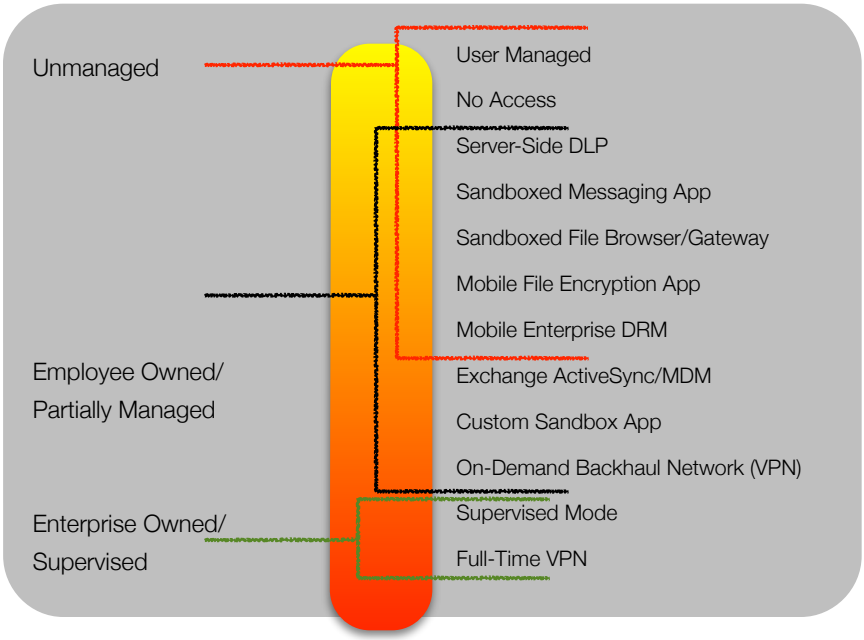
Managed accounts and apps aren't perfect, but offer a compelling way for enterprises to protect data in BYOD scenarios without having to take full control of employee devices. The key is to provide the right apps to enable employee workflows within a managed data flow.

The iOS Data Security Spectrum

Managing iOS devices involves a lot more than merely determining whether they will be owned by the employee or the enterprise. There are a whole spectrum of options available for securing enterprise data on iOS, depending on how tightly you want to manage the device and data. 'Spectrum' isn't quite the right word, though, because these options aren't on a linear continuum — despite falling into our three main categories:

- Options for unmanaged devices
- Options for employee-owned/partially managed devices
- Options for enterprise-owned (supervised) devices

We define these categories as follows:



- *Unmanaged devices* are fully in the control of the end user. No enterprise polices are enforced, and the user can install anything and use the device however they please.
- *Employee-owned/partially managed devices* use a configuration profile or Exchange ActiveSync policies to manage certain settings, but the user is otherwise still in control of the device. We previously referred to them as *partially managed* devices, but here we use Apple's term. The device belongs to the user but they agree to some level of corporate management. They can install arbitrary applications and change most settings. Typical policies require users to use a strong passcode and enable remote wipe by the enterprise. They may also need to use an on-demand VPN for at least some network traffic (*e.g.,* to the enterprise mail server and intranet web services), but other traffic goes unmonitored through whatever network connection they are currently using.

- *Enterprise-owned devices* are deployed in supervised mode. The enterprise controls what apps can be installed, enforces an always-on VPN (if desired) which users cannot disable, and has the ability to monitor and manage all traffic to and from the device.

Even with all the time we spent describing Apple's BYOD philosophy and features, there are many implementation nuances. Some options fit into multiple categories, so we will start with the least protected and work our way up through higher levels of protection. We will note which options carry forward to the higher (tighter) buckets.

Note: This paper is focused exclusively on data security — it does not discuss mobile device management in general, or the myriad other device management options!

Let's start with a brief discussion of data protection options in the first bucket:

Unmanaged Devices

Unmanaged devices are completely under the user's control, and the enterprise is unable to enforce any device polices. This means no configuration profiles and no Exchange ActiveSync policies to enforce device settings such as passcode requirements. Yes, believe it or not, some organizations (especially smaller ones) choose this option.

User Managed Security with Written Policies

Under this model you don't restrict data or devices in any way, but institute written policies requiring users to protect data on the devices themselves. This option is not very secure but we believe in being comprehensive.

Basic written policies should include the following:

- Require Passcode: After *n* minutes
- Simple Passcode: **OFF**
- Erase Data: **ON**

Additionally we highly recommend you enable some form of remote wipe — either the free Find My iPhone, Exchange ActiveSync, or a third-party app.

These settings enable data protection and offer the highest level of device security possible without additional tools, but they aren't generally sufficient for an enterprise or anything other than the smallest businesses.

We will discuss employee policies in more detail later, but make sure the user signs a mobile device policy saying they agree to these settings, then help them get the device configured. But if you are reading this paper we figure you care enough about security that unmanaged devices aren't a viable option.

No Access to Enterprise Data

The simplest choice for security is to completely exclude iOS devices, rather than attempting to secure them. But depending on how your environment is set up this might be very difficult. There are a few key areas you need to check to ensure an iOS device won't slip through:

- **Email server:** If you support IMAP/POP or even Microsoft Exchange mailboxes, if the user knows the server settings and you haven't implemented any preventative controls, they will be able to access email from their iPhone or iPad. There are numerous ways to prevent this (too many to cover in this paper), but as a rule of thumb if the device can access the server, and you don't have per-device restrictions, they can probably get email on the iDevice.

- **File servers:** Like email servers, if you allow the device to connect to the corporate network and have open file shares, users can access corporate content. There are plenty of file server clients in the App Store capable of accessing most server types. If you rely on username and password protection (rather than network credentials) users can fetch content to their devices.
- **Remote access:** iOS offers support for a variety of VPNs. Unless you use certificate or other device restrictions, and especially if your VPN is based on a standard like IPSec, there is nothing to prevent end users from configuring the VPN on their device. Don't assume users won't figure out how to VPN in, even if you don't provide direct support.

To put this in perspective, in the Securosis environment we allow extensive use of iOS. We didn't have to configure anything special to support iOS devices — we simply had to *not* configure anything to block them.

Email Access with Server-side Data Loss Prevention (DLP)

With this option you allow access to enterprise email but enforce content-based restrictions using DLP to filter messages and attachments before they reach devices.

Most DLP tools filter at the mail gateway (MTA) — **not** at the mail store (*e.g.,* Exchange). Unless your DLP tool offers explicit support for filtering based on content *and device* you won't be able to use this option since mobile devices pull email at the mail server level, not from the MTA.

If your DLP tool is sufficiently flexible, though, you can use it to prevent sensitive content from going to the device, while allowing normal communications. You can either build this off existing DLP policies or create completely new device-specific ones.

Sandboxed Messaging App / Walled Garden

One of the more popular options today is to install a sandboxed app for messaging and file access, to isolate and control enterprise data. These apps do not use the iOS email client, and handle all enterprise email and attachments internally. They also typically manage calendars and contacts, and some offer access to intranet web pages.

These are a great option if you don't trust Apple's MDM model, or want similar protections over email without having to deploy MDM. They might also be a good option if you want consistency across Android or Windows Mobile.

The app may implement its own encryption and hardening on top of the default Data Protection, especially if it was written for earlier iOS versions. Some of these apps can be installed without *requiring* a configuration profile to enforce a passcode, remote wipe, client certificate, and other settings, but in practice policies are almost universally required (placing these apps more in the *Partially Managed* category). You don't *necessarily* have to enforce settings, so we include these in the *Unmanaged Devices* category, but they will show up again in the *Enterprise-Owned/Partially Managed* section.

A sandboxed messaging app may support one are all of the following, depending on the product and how you have it configured:

- Isolated and encrypted enterprise email, calendars, and contacts.
- Encrypted network connection to the enterprise without requiring a separate VPN client (end-to-end encryption).

Enterprises tend to use their own app stores for enterprise-specific applications signed with their own certificates. There are vendors which provide this infrastructure to ease deployment. VPP software allows you to purchase and manage applications through Apple's App Store.

- In-app document viewing for common document types (typically running the built-in iOS document viewer within the sandbox).
- Document isolation. Documents can be viewed within the app, but "Open In…" is restricted for all or some document types (this is different than a *managed app* because the app enforces the restriction rather than iOS).
- Remote wipe of the app (and data store), the device, or both.
- Intranet web site and/or file access.
- Detection of jailbroken iOS devices to block use.

The app becomes the approved portal to enterprise data, while the user is otherwise free to do whatever they want on the device (subject to simple security policies).

To finish our discussion of securing data on unmanaged devices, let's consider three categories of apps designed for secure file access:

Sandboxed File Browsers and Mobile File Gateways

While messaging apps generally do a good job of handling email, they don't necessarily link into file servers or integrate with enterprise encryption. Secure file management apps skip email features and focus on access to enterprise file repositories. They support the following core features:

- Use of either iOS Data Protection alone or their own embedded encryption.
- A secure connection to the file repository (which may require a VPN for remote access to internal sources).
- Support for the iOS document viewer for supported document types (iWork, Microsoft Office, PDF, etc.).
- Authentication and authorization to enable or restrict access on a per-user, per-device basis.
- Ability to restrict or allow "Open In…" to control file movement to other apps, independent of *managed apps*.

There are a few different flavors. Most require server components or plugins to repositories like Microsoft SharePoint.

- **Sandboxed file browser:** These allow connections to enterprise file shares using standard connections and store downloaded documents in an encrypted container. Most now rely on Data Protection rather than their own encryption schemes. This is usually read-only, although some support annotation of PDF files.
- **Sandboxed cloud file browser:** Rather than relying on direct network connections to enterprise file stores, these apps access cloud storage repositories in a cloud service.
- **Mobile file management gateway:** This is a refinement on the sandboxed file browser. Rather than allowing access directly to file repositories, mobile devices connect to the gateway using a sandboxed app and are then given access to files through the gateway. These support more granular policies, monitoring, and directory integration. They often also support multiple mobile platforms (yes, there is a world outside Apple).
- **Document management system extensions:** These are similar to mobile file management gateways, but instead of a separate server they run as *plugins* to an existing document management system. Users connect directly to the document management system (such as SharePoint) via the extension/plugin, which might be centrally managed.

Some of these tools support commenting and annotating files (usually restricted to PDFs) but expanded document editing is on several roadmaps.

Sandboxed Mobile File Encryption Apps

Mobile computing is one of the big drivers of cloud computing, and cloud storage in turn is expanding use of encryption. Encryption apps extend the sandboxed file browser by integrating with enterprise encryption. They expand on the file browser by:

- Maintaining file and document isolation in the sandbox.
- Transparently decrypting files accessed by the app (when integrated into an enterprise encryption scheme and key management server).
- Accepting files from other apps via "Open In…" and keeping them encrypted in private storage, then offering protected access to such files.
- Support for connections to common cloud storage platforms such as Box and Dropbox.

The main division within this category is between apps designed to open files passed to them by other applications (such as encrypted mail attachments) and those which integrate directly into cloud storage or other file browsers. Some tools can also decrypt password-protected files sent by email, unlike those managed using centralized enterprise keys.

When integrated with enterprise key management, the entire process of accessing encrypted files on iOS is completely transparent to the user. They open the app, which connects to the file store, and files are cached within the app's secure data store and decrypted as needed. Documents can be restricted so they are only accessible within the app, as with our other sandboxing examples. Some apps also support encryption of files from other apps.

This actually provides *more* protection than normal desktop encryption because it is far easier to isolate documents and keep them within the app.

Mobile Enterprise Digital Rights Management

The next option for handling files securely on unmanaged devices expands encryption into Enterprise Digital Rights Management. EDRM provides more granular controls that travel with the documents, a step closer to true information-centric security. The easiest way to distinguish between a simple encryption app and EDRM on iOS is:

- An encrypted document opened in a sandbox may be isolated in that app, but generally isn't protected when accessed on other systems which also have access (such as a laptop or desktop). Protection is binary, like a lockbox — controlling only who can access the file. We rely on the sandbox app for additional controls — such as restricting movement into other apps — usually on an all-or-nothing basis.
- An EDRM protected document stays encrypted, but can only be opened by applications that respect the more granular controls applied to the file — including compatible mobile apps. This allows a wide range of control — including who can open the file, who can edit it, who can forward it via email, which devices can access it, and even time limits for access.

> Less control doesn't necessarily mean less risk. Some options are very secure even on unmanaged or partially managed devices. They restrict the user to their container more, while fully managed devices allow safe use of more apps.

In the mobile space EDRM is better for protecting files you want to share externally but still protect. Encryption, in contrast, is generally only suitable for internal use, or secure transmission of documents, because it offers no restriction on what others can do once they decrypt the contents. EDRM is strongly oriented towards office documents, while encryption works better with

arbitrary files.

Mobile EDRM requires a server or service to manage keys. The rights themselves are embedded in the documents. There are a variety of deployment models, including:

- Mobile file gateway
- File server/SharePoint integration
- Email client integration
- Email server integration
- Microsoft Office integration

To simplify a bit: documents can either be manually protected when you create them in Office or email them, when you upload them to an EDRM-enabled file gateway/storage platform, or automatically when you save them into a protected directory or email them to a certain destination.

These documents can only be read using the vendor's proprietary solution (app), which enforces the rights. Some tools integrate into Microsoft's Windows Rights Management Services (RMS) or another EDRM platform which is integrated into Office. Another valuable feature is the ability to *edit* documents, not merely to annotate them.

Rights you can manage on a *per file* basis generally include:

- Who can read the file.
- Who can edit the file (with the same annotation/commenting limitations we see in most iOS apps, although that should be changing soon).
- Who can transfer the file out of the sandbox ("Open In…").
- Who can print the file.
- Who can share the file (allow others to read it).
- How long the file is accessible.
- Who can copy out of the file and paste elsewhere.

Unless you allow users to remove rights (or copy out of the document), content is always encrypted and protected as it moves between locations, users, and platforms. Rights are tied directly to users in enterprise environments through directory servers, so there is little sharing of credentials to allow access. If you want to exchange protected documents with external users you have them download the (usually free) app and send them the file, then use an alternate authentication and authorization model such as federation.

For external users or mobile users without VPN access, the keys and rights management server must be Internet accessible — perhaps hosted by a SaaS provider.

Sandboxed file browsers, mobile encryption, and mobile EDRM all use sandboxed apps for handling files — with different degrees of security, flexibility, and integration.

This covers all our options for unmanaged devices — *nearly all of which are also relevant on partially managed and enterprise-owned devices*.

Partially Managed Devices

We now turn to *employee-owned/partially managed* devices.

Our definition is:

Devices that use a configuration profile or Exchange ActiveSync policies to manage certain settings, but the user is otherwise still in control of the device. The device is owned by the user, but they agree to some level of corporate management.

With iOS 7 management of the device is minimally intrusive to the user's overall experience, the enterprise is segregated from the user's personal data and applications, and vice-versa (assuming you configure policies that way). The following policies are typically deployed onto partially managed devices via Exchange ActiveSync:

- Enforce passcode lock.
- Disable simple passcode.
- Enable remote wipe.

This, in turn, enables Data Protection on supported hardware (including all iOS models currently for sale).

Apple's iOS configuration profiles offer all EAS features except remote wipe (which requires a remote wipe/MDM server), as well as the following additions:

- On-demand VPN for specific domains (not all traffic, but all enterprise traffic).
- Manual VPN for access to corporate resources.
- Per-app VPN.
- Digital certificates for access to corporate resources (VPN or SSL).
- Installation of custom enterprise applications — managed or unmanaged.
- Access to Volume Purchase Program licenses for apps in the App Store.
- Automatic wipe on failed passcode attempts (the number of attempts is configurable, unlike the simple ON/OFF user setting in the Settings app, which is fixed at 10 failures).
- Enforcement of managed apps and managed accounts restrictions.

One key point for administering managed policies on a user-owned device is to ensure that you *obtain the user's consent and notify them of what will happen.* The user should sign a document saying they understand that although they own the device, by accessing corporate resources they are allowing management of it — which may include remote wiping a lost or stolen device. And the user is responsible for their own backups of personal data. A remote wipe clears all data off a device — not just enterprise data.

Unlike other platforms, you don't install an MDM agent on iOS devices. Apple handles MDM for you and your MDM tools connects using standard APIs and configuration profiles. This limits MDM vendors' ability to differentiate themselves on iOS.

Enhanced Security for Existing Options

Most of the previous options we have discussed are significantly enhanced when digital certificate, passcode, and Data Protection policies are enforced. This is especially true of all the sandboxed app options — and in fact many

vendors don't support use of their tools without a configuration profile which requires at least a passcode. It also opens up managed app and managed account restrictions to (almost) fully segregate enterprise data from the user's data and apps (except for the AirDrop & Copy/Paste holes).

Managed Exchange ActiveSync (or Equivalent)

Microsoft's Exchange ActiveSync protocol, despite its name, is separate from the Exchange mail server and included with other products, including some that compete with Exchange. iOS natively supports it, so it is the backbone for managed email on iDevices when a sandboxed messaging app or MDM isn't used. EAS is unable to enforce the same degree of policy control as MDM on iOS 7; if MDM is an option it provides a lot more management bang for the buck.

By configuring the policies listed above, all email is encrypted under the user's passcode using Data Protection.

Custom Enterprise Sandboxed Application

Now that you can install an enterprise digital certificate onto the device and guarantee Data Protection is active, you can also deploy custom enterprise applications that leverage this built-in encryption.

This option allows you to use the built-in iOS document viewer within your application's sandbox, which makes it fairly easy to deploy a custom application that provides fully sandboxed and encrypted access to enterprise documents. Combine it with an on-demand VPN tied to the domain name of the server or a manual VPN, and you get data encrypted both in transit and in storage.

Today a few vendors provide toolkits to build this sort of application. Some are adding document annotation for PDF files and full editing capabilities for MS Office document formats.

Managed/Backhaul Network (Cloud or Enterprise)

This option works with other data security tools, and is not sufficient to protect data by itself. Users own their devices, but agree to route all traffic through an enterprise-managed network. This might be a VPN back to the corporate network or a VPN service.

On the data security side this enables monitoring of all network traffic — possibly including SSL traffic (by installing a special certificate on the device). This is primarily for malware protection and to reduce the likelihood of malicious apps on devices, but it also enables more complete DLP.

We rarely see this option in the real world.

Enterprise-Owned (Supervised) Devices

Enterprise-owned devices in supervision mode make life easier for security administrators. Full control of the device enables enforcement of any desired policies, although users might not be thrilled.

Remember that full control doesn't mean the device must be in a highly restricted kiosk mode — you can allow a range of activities while maintaining security. All the previous data security options are still available, along with:

Supervised Mode

Using a Mobile Device Management tool, the iOS device is completely managed and restricted. The user is unable to install unapproved applications, email is limited to the approved enterprise account, and all security settings are enabled for Data Protection.

Restricting the applications allowed on the device and enforcing security policies makes it much more difficult for users to leak data through unapproved services. Plus you gain full Data Protection, strong passcodes, and remote wipe. Some MDM tools even detect jailbroken devices. This is the one option that closes the AirDrop hole.

To gain the full benefit of Data Protection you need to block unapproved apps which could leak data (such as Dropbox and iCloud apps). This isn't always viable, which is why this option is often combined with a captive network (always-on VPN) to give users a bit more flexibility to install apps on their own, vs. a total lock down.

Managed/Backhaul Network with DLP, etc.

The device uses an on-demand VPN for all network traffic, at all times, through an enterprise or cloud portal. We call it an "on-demand" VPN because the device automatically shuts it down when there is no network traffic and brings it up before sending traffic — the VPN 'coverage' is comprehensive. 'On-demand' in this context definitely does *not* mean that users can bring the VPN up and down as desired.

Combined with full device management, the captive network affords complete control over all data moving onto and off devices. This is primarily used with DLP to manage sensitive data, but may also be used for application control or even to allow monitored access to non-enterprise email accounts.

For DLP, in addition to management of enterprise email without a full captive network, this option also enables management of web traffic.

Full control of the device and network doesn't remove the need for other security options. For example you might still need encryption or DRM to allow use of otherwise insecure cloud and sharing services.

Defining Your iOS Data Security Strategy

Now that we have covered the different data security options for iOS, it is time to focus on building a strategy. In many ways figuring out the technology is the easy part — the problems start when you need to apply technology in a dynamic business environment, with users who have already made technology choices.

Factors

Most organizations we speak with — of all sizes and verticals — are under intense pressure to support iOS, to *expand* support for iOS, or to wrangle control over data security on iDevices already deployed and in active use. Developing a strategy depends on where you are starting from as much as on your overall goals. Here are the major factors to consider:

Device Ownership

Device ownership is no longer a simple question of "ours or theirs". Some companies are able to maintain strict management of everything that connects to their networks and accesses data, but this has become the exception rather than the rule. Nearly all organizations are being forced to accept at least some level of employee-owned device access to enterprise assets, whether that means remote access for a home PC or access to corporate email on an iPad.

The fact that Apple *offers* additional control over supervised devices doesn't make it the best choice. The first question to ask yourself is whether you can maintain strict ownership of all devices you support — followed by whether you even want to. The gut instinct of most security professionals is to only allow organization-owned devices, but this is rarely a viable long-term strategy. Fortunately allowing employee-owned devices doesn't require you to give up on enterprise ownership completely, especially with the MDM enhancements in iOS 7.

Many of the data security options we have discussed work in a variety of scenarios. Here's how to piece together your options:

- **Employee-owned devices:** Your options are either partially managed or unmanaged. With unmanaged you have few viable security options; so you should focus on sandboxed messaging, encryption, and DRM apps. Even if you use one of these options, it will be more secure if you can use least minimal partial management to enable Data Protection (by requiring a passcode and enabling automatic wipe on passcode failure), enabling remote wipe, and installing an enterprise digital certificate. The key is to sell this option to users, as we will detail below. Ideally, link the device into your MDM infrastructure and leverage managed accounts and apps. You can still use the security applications we discussed, but gain better control overall of data flow.
- **Organization-owned devices:** These fall into two categories: general and limited use. Limited use devices are highly restricted and serve a single purpose — such as flight manuals for pilots, mobility apps for health care, and sales or sales engineering support. They are locked down with only necessary apps available. General use devices are issued to employees for a variety of duties and support a wider range of applications. For data security focus on

techniques that manage data moving on and off devices — typically managed email and networking — with good app support for what users need to get their jobs done.

If the employee owns the device you need their permission for any device management. Define clear and simple policies that include the following points:

- It is the employee's device, but in exchange for access to [employer] resources the employee allows the organization to install a work profile on the device.
- The work profile requires a strong passcode to protect the device and the data stored on it.
- In the event the device is lost or stolen, the employee must report it within [time period]. If there is reasonable belief that the device is at risk, [employer] will remotely wipe the device. This protects both personal and company data. If you use a sandboxed app that only wipes itself, specify that here.
- If you use a backhaul network, detail when it is used.
- Devices cannot be shared with others, including family.
- How the user is allowed to back up the device (or a recommended backup option).

Emphasize that these restrictions protect both personal and organizational data. The user must understand and accept that they are giving up some control of their device in order to gain access to work resources. **They must sign the policy**, because you are installing something on their personal device, and you need clear evidence they know what that means. Remind them that, per Apple's model, their personal data is also protected from the enterprise.

Culture

Financial services companies, defense contractors, healthcare organizations, and tech startups all have very different cultures. Some expect and accept tightly restricted access to employer resources, while others assume unrestricted access to consumer technology.

Don't underestimate culture when defining your strategy — we have presented a range of data security options, and some may not work with your particular culture. If more freedom is expected look to sandboxed apps. If management is expected you can support a wider range of work activities with tighter device control.

Sensitivity of Data

Not every organization has the same data security needs. There are fields with information that simply shouldn't be allowed onto a mobile device with any chance of loss. Fortunately most organizations have more flexibility.

The more sensitive the data, the more it needs to be isolated (or restricted from being on devices). This ties into both network security options (including DLP to prevent sensitive data from going onto the device) and messaging/file access options (such as managed apps, Exchange ActiveSync, and sandboxed apps of all flavors).

Not all data is equal. Assess your risk and then tie it back into an appropriate technology strategy.

Business Needs and Workflow

If you need to exchange documents with partners you will use different tools than if you only want to allow access to employee email. If you use cloud storage or care about document-level security you might need a different tool.

Determine what the business *wants* to do with devices, then figure out which components you need to support that. Don't forget to look at what they are *already doing*, which might surprise you. Keep in mind that managed apps are tied

to the VPP program, and you will likely want to purchase and provide the apps employees need so you can maintain management.

Existing Infrastructure

If you have backhaul networks or existing encryption tools, they may incline you in a particular direction. Document storage and sharing technologies (both internal and cloud) are also likely to influence your decision.

The trick is to *follow the workflow*. Map out existing and desired employee workflows and note where they intersect with your infrastructure, which will help feed strategy requirements.

Compliance

Will the device access any data or applications with compliance ramifications? If so it may need to comply with specific requirements which could include anything from encryption to email archiving. Or even preventing the devices completely.

Make a Decision

Here is a suggested process to pull all these factors together:

- Determine the ownership model to support: employee, enterprise, or both.
- Determine which devices to support (we focused on iOS for this paper, but the available options change with additional device types).
- Identify business processes and applications to support. This includes: a) email and communications, b) data repositories, c) enterprise applications, and d) external services such as cloud storage and SaaS applications.
- Map out business workflows for the identified processes.
- Determine data security and compliance requirements for identified data and workflows. These should include how the data needs to be stored (*e.g.,* encrypted), where it can be exchanged (*e.g.,* email to external parties), and where it can be accessed.
- Map business workflows first to device (where the data may transfer onto the device) and then to the on-device workflow (which apps are used). Don't map security controls yet — this part is more about figuring out *how* employees want to use the data on the device.
- Identify potential security controls and tools to enforce security requirements at each step of each identified workflow.
- Review and determine which tool categories to support.
- Identify and select specific tools.

You'll notice that although we opened with a discussion of information-centric security, at this point we are more concerned with identifying the workflows involved. That's because we need to bridge the business and security requirements — to protect the data we need to know how it's used **and** how employees want to use it. The best data security in the world is useless if it interferes so much with business process that it kills off what the business wants to do, or if users decide they need to work around it.

Conclusion

iPhones, iPads, and cloud computing are the 1–2–3 punch knocking down our traditional expectations for securing enterprise data and managing employee devices and services. Simultaneously, they are creating new opportunities for

information-centric security approaches we have long ignored, as we fixated on our fantasy of the enterprise perimeter. I am firmly convinced that these new models create more security opportunities than risks.

But intense pressure to support new things in a short time frame is always a challenge.

The good news is that iOS is a relatively secure platform that is completely suitable for most organizations. Of course it isn't perfect, and employee ownership and expectations further complicate the situation. For some organizations, the risks are still simply too great.

For the rest of us who want to embrace iOS, tools are available to do so securely, with a range of deployment options. We can start with something as simple as filtering out sensitive emails before they hit the iPhone, and work up to something as complex as multi-organization secure document workflows with managed accounts and apps. Hopefully this paper has given you some good starting tips, and as new technologies appear we will try to keep it up to date.

Who We Are

About the Author

Rich Mogull, Analyst and CEO

Rich has twenty years of experience in information security, physical security, and risk management. He specializes in data security, application security, emerging security technologies, and security management. Prior to founding Securosis, Rich was a Research Vice President at Gartner on the security team where he also served as research co-chair for the Gartner Security Summit. Prior to his seven years at Gartner, Rich worked as an independent consultant, web application developer, software development manager at the University of Colorado, and systems and network administrator. Rich is the Security Editor of TidBITS, a monthly columnist for Dark Reading, and a frequent contributor to publications ranging from Information Security Magazine to Macworld. He is a frequent industry speaker at events including the RSA Security Conference and DefCon, and has spoken on every continent except Antarctica (where he is happy to speak for free — assuming travel is covered).

About Securosis

Securosis, L.L.C. is an independent research and analysis firm dedicated to thought leadership, objectivity, and transparency. Our analysts have all held executive level positions and are dedicated to providing high-value, pragmatic advisory services.

We provide services in four main areas:

- Publishing and speaking: Including independent objective white papers, webcasts, and in-person presentations.
- Strategic consulting for end users: Including product selection assistance, technology and architecture strategy, education, security management evaluations, and risk assessments.
- Strategic consulting for vendors: Including market and product analysis and strategy, technology guidance, product evaluations, and merger and acquisition assessments.
- Investor consulting: Technical due diligence including product and market evaluations, available in conjunction with deep product assessments with our research partners.

Our clients range from stealth startups to some of the best known technology vendors and end users. Clients include large financial institutions, institutional investors, mid-sized enterprises, and major security vendors.

Securosis has partnered with security testing labs to provide unique product evaluations that combine in-depth technical analysis with high-level product, architecture, and market analysis.