Defending Data on iOS

Version 1.0
Released: June 14, 2012

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/3.0/us/

Table of Contents

Introduction 2
- Why iOS and Not Android 2
- Information-Centric Security 3

iOS Security and Data Protection 4
- Device and OS Security 4
- iOS Data Flow 7

The iOS Data Security Spectrum 9
- Unmanaged Devices 10
- Partially Managed Devices 14
- Managed Devices 16

Defining Your iOS Data Security Strategy 18
- Factors 18
- Make a Decision 20
- Conclusion 20

Who We Are 22
- About the Author 22
- About Securosis 22

Introduction

The numbers alone can't tell the story. In 2011 Apple sold 315 million iOS devices (62 million in the fourth quarter). There are over 100 million iCloud users – using a service less than a year old. And these numbers are for Apple alone – never mind all the other mobile devices. Apple calls this the dawn of the "post-PC era", and with numbers like those it's hard to argue. Even Microsoft is in the midst of what is shaping up to be the largest change in its platform strategy since Windows, in an attempt to address this market.

These devices aren't confined to home. Survey after survey shows growing enterprise adoption of iOS, including major migrations off RIM BlackBerry and other business-centric smartphones – 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're the last to find out about new tools and toys, mobility is one area where we have held tight control by restricting access to the network. But in this post-PC consumer world we are losing our ability to stop or slow adoption of consumer technologies, even when they don't support all our enterprise needs.

In a recent session 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.

And while there was some concern about the impact of these devices on the network, the universal 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 shift the smartphone and tablet markets than any other provider. And, not coincidentally, we are asked about securing iOS for the enterprise more than any other platform.

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

The real activity is with iOS and Google's Android. But for a variety of reasons enterprises face more pressure to support iOS. Android-based tablets are not yet competitive or in wide use, and the fractured nature of Android phones and software versions makes it far easier to justify restricting those devices.

iOS is also a stronger platform in terms of security. While nothing is invulnerable, 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 limited tools for enterprise management of iOS. There is no ability to run background security applications, so we need to rely on policy management and a spectrum of security architectures.

We will focus on iOS because:

- You already know how to manage BlackBerry.
- Android isn't mature or safe enough for us to endorse for enterprise use (unless you highly restrict which devices you support), and the fractured operating system levels make strategic management difficult.
- Windows Mobile is not in widespread use and the Metro tablet platform is still in development.
- Clients tell us they are under pressure to support iOS more than other platforms – particularly the iPad.
- Most of the options we will discuss also apply to other platforms – especially the latest version of Android (4.0 Ice Cream Sandwich, which isn't widely available yet).

Information-Centric Security

We are focusing 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 the data even before it hits the device.

Previously I wrote the following 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 changing business contexts.
- Policies must work consistently through the different defensive layers and technologies we implement.

These sound a bit like the usual analyst mumbo-jumbo, but we actually have the technologies to implement much of this today. In terms of managing data for mobility and iOS we can hit every one of those points except movement between structured and unstructured data.

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 security and data protection models. These are the controls 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.

The short version is that iOS is quite secure – far more secure than a general-purpose computer. The downside is that Apple supports only limited third-party security management options.

Note: We are only discussing iOS 5 and later (as of this writing 5.1 is the current version of the iOS operating system – for iPhone, iPad, and iPod touch). We do not recommend supporting previous versions of iOS.

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). This is 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.

> 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 we discuss in this paper.* Those devices, and later devices like the iPad (3), are much more resilient to attack if they are lost or stolen.

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

Operating System Hardening

Five key features of iOS are designed to minimize the chances of successful exploitation, even in the face of a possible 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 extremely difficult for an attacker to complete exploitation and run their own code, even if they do find and take

advantage of 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.

- **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 the device, including exploit code. Apple only signs applications sold through the App Store, minimizing the danger of malicious apps.
- **Sandboxing:** All applications are highly compartmentalized from each other, with no central document/file store. Applications can't 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, but this uses a model similar to the 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. Throw in code signing and additional software and hardware security beyond the scope of our discussion, and iOS is very difficult to exploit.

Of course it isn't impossible, and we *have* seen exploits (especially local attacks such as tethered jailbreaks), 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.

Security Features

In addition to its fundamental architectural security controls, iOS also 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 way they tie into the Data Drotection and device wipe features.
- **Passcode Wipe:** When a PIN or passphrase is set, if the code is entered incorrectly enough times the device can erase all user data (this is based on the encryption features discussed next).
- **Remote Wipe:** iOS supports remote wipe via Find My iPhone and Exchange ActiveSync. Of course the device must be accessible on the Internet to receive the wipe command.
- **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.
- **VPN and on-demand VPN:** Virtual private networks can be activated manually or automatically when the device accesses any network service. (Not all VPNs support on-demand connection.)
- **Configuration Profiles:** Many of the 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 on the phone or tablet.

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:

- **Hardware 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.
- **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, their email data (including attachments) is encrypted using their passcode as the key. Data Protection also applies to applications that leverage the Data Protection API. With this enabled, even if the device is physically lost and exploited, any data protected with this feature (specifically mail and data for third-party applications which implement the Data Protection API) is encrypted. Attackers may still attempt brute-force guessing attacks against the key, of course.
- **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.

This is not as robust as BlackBerry full device encryption, but it forms the basis for most iOS data security strategies. Prior to the availability of hardware encryption and Data Protection it was nearly impossible to protect data on iOS. Now that those features are available to all applications, however, Apple has provided an enterprise-class mechanism for securing data even when devices are lost or stolen.

Management Options and Limitations

Overall, iOS includes fewer enterprise management options than organizations are used to having – particularly in comparison to general-purpose desktop and laptop computers. Apple does not support full background applications, which means there is no capability to install constantly-running security software like antivirus or DLP on iOS devices.

To us this is a net positive because all applications are sandboxed and there is no ability for background tasks that could snoop on user activity or otherwise compromise security.

Although we don't feel this is a serious security risk – despite the cries of antivirus vendors as they watch the hot new market slip through their grasp – it *could* be a compliance issue if you are required to run such software on all devices to satisfy someone's checklist mentality.

The inability to run background tasks also means that device management is restricted to the features Apple exposes through configuration profiles. These include application and feature restrictions, as well as the ability to manage VPNs, digital certificates, passcodes, and other features. No matter what mobile device management tool you use, they all tie back to these profiles.

This should help you understand the key security features of iOS; next we will review the options for protecting enterprise data; then we will conclude with a framework for deciding which options are best for your organization.

iOS Data Flow

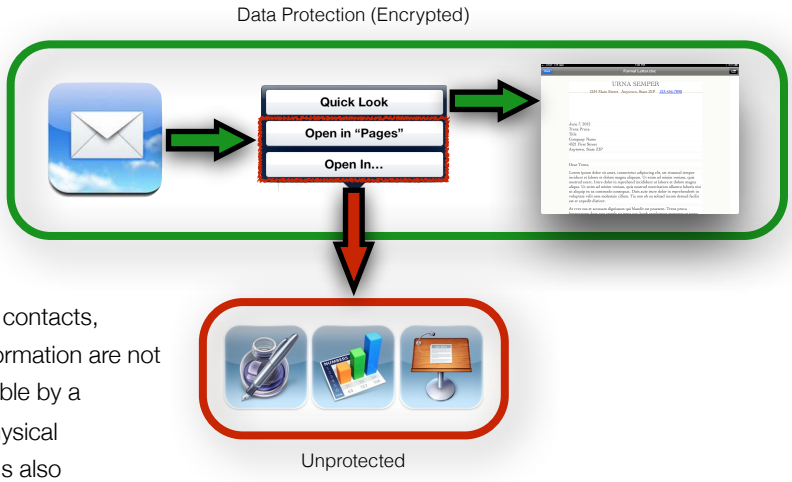
We need to take some time to understand how data moves onto and around iOS devices before delving into security and management options.

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, because otherwise iOS provides no real data security.

- Emails and email attachments.
- Calendars, contacts, and other non-email user information.
- Application data.

When the iOS Mail app downloads mail, message contents and attachments are stored securely and encrypted using Data Protection (under the user's passphrase). If the user doesn't set a passcode the data is stored along with the rest of user data, and only encrypted with the device key. 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.

Assuming the user properly uses Data Protection, mail attachments viewed with the built-in viewer app are also safe. But once a user uses "Open In…", the file is copied into the target application's storage sandbox, and may thus be exposed. When a user downloads an email and an attachment, and views them in the Mail app, both are encrypted twice (once by the underlying FDE and once by Data Protection). But when the user opens the document with Pages to edit it, a copy is stored in the Pages store, which does not use Data Protection – and the data can be exposed.



This workflow is specific to email – calendars, contacts, photos, and other system-accessible user information are not similarly protected, and are generally recoverable by a reasonably sophisticated attacker who has 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.

Other (third party) application data may or may not utilize Data Protection – this is up to the app developer – and is always sandboxed in the application's private store. Data in each application's local store is encrypted with the user's passcode. This data may include whatever the programmer chooses – which means some data may be exposed, although documents are nearly always protected when Data Protection is enabled. The programmer can also restrict what other apps a given document is allowed to open in, although this is generally an all-or-nothing affair. If Data Protection isn't enabled, data is only protected with the device's default hardware encryption, but sandboxing still prevents each app's data from access by other apps.

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 from the online service to retrieve their own (private) copies.

So application data (files) may be exposed despite Data Protection if the app supports “Open In…”. Otherwise data in applications is well protected. If a network storage service is used the copy within the app is still protected and isolated, but the network copy is accessible to other compatible apps. This isn’t a fault of iOS but must be kept in mind. Especially if a document is opened in a Data Protection enabled app (where it’s secure), but then saved to a storage service that allows insecure apps to access it and store unencrypted copies.

So iOS provides both protected and unprotected data flows. A protected data flow places content in a Data Protection encrypted container and only allows it to move to other encrypted containers (apps). An unprotected flow allows data to move into unencrypted apps. Some kinds of data (iOS system calendars, contacts, photos, etc.) *cannot* be protected and are always exposed.

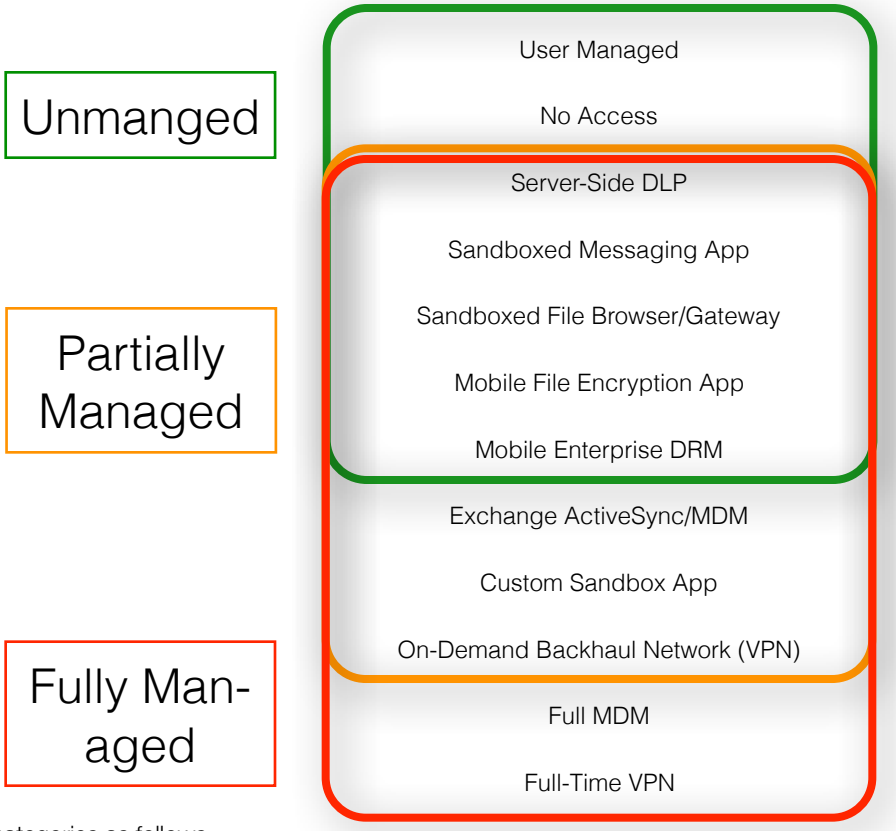
On top of this, some apps use their own internal encryption, which isn’t tied to the device hardware or the user’s passcode. Depending on implementation, this could be more or less secure than using the Data Protection APIs.

The key for security is to understand how enterprise data moves onto the device (which app pulls it in), whether that app uses Data Protection or some other form of encryption, and what other apps that data can move into. If data ever moves into an app that doesn’t encrypt, it is exposed.

The iOS Data Security Spectrum

There are a whole spectrum of options available for securing enterprise data on iOS, depending on how much 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 – instead they fall into three major buckets:

- Options for unmanaged devices
- Options for partially managed devices
- Options for fully managed 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.

- *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. The device is the user's, but they agree to some level of corporate management. They can install arbitrary applications and change most settings. Typical policies require them 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 the user's other traffic goes unmonitored through whatever network connection they are currently using.
- *Fully managed devices* also use a configuration profile, but are effectively enterprise-owned. The enterprise controls what apps can be installed, enforces an always-on VPN the user cannot disable, and has the ability to monitor and manage all traffic to and from the device.

Some options fit into multiple categories, so we will start with the least protected and work our way up through the 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!

With that reminder, 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.

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 option but we are 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 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 this is not a good option for you.

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 right 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 post), 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 the 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) then the user can fetch content to their device.
- **Remote access:** iOS includes decent 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 the 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.

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 mail client, and handle all enterprise emails and attachments internally. They also typically manage calendars and contacts, and some include access to intranet web pages.

The app may use iOS Data Protection, implement its own encryption and hardening, or use both. 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 *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).
- In-app document viewing for common document types (usually 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.
- 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 (albeit often with a few minor security policies enforced).

To finish our discussion of securing data on unmanaged devices, let's focus on 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 messaging and focus on access to enterprise file repositories. They support the following core features:

- Use of either iOS Data Protection 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.

There are a few different flavors. Most require server components or plugins to repositories like Microsoft SharePoint. If a tool cannot isolate documents by restricting the "Open In…" feature it is unsuitable for enterprise use.

- **Sandboxed file browser:** These allow connections to enterprise file shares using standard connections and store downloaded documents in an encrypted container. Most use Data Protection rather than their own encryption schemes. They are 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 and are tied to their 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 on 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 enabling protected access to such files.
- Support for connections to common cloud storage platforms such as Box.net 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) versus those that integrate directly into cloud storage or other file browsers. Some tools also support decryption of password-protected files versus 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's 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 an simple encryption app and EDRM on iOS is:

- An encrypted document opened in a sandbox may be isolated in that app, but isn't generally 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.

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 securely transmission of documents, because it offers no restriction on what others can do once they decrypt the contents. EDRM is very oriented towards office documents, while encryption works better with arbitrary files.

Mobile EDRM requires a server or service to manage the 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 this 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.

The documents can only be read using the vendor's proprietary solution (app), which enforces all the rights. Some tools integrate into Microsoft's Windows Rights Management (RMS) service or another EDRM platform which is integrated into Office.

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/paste out of the file.

Unless you allow users to remove rights (or copy/paste 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 to be Internet accessible – perhaps hosted by a SaaS provider.

Sandboxed file browsers, mobile encryption, and mobile EDRM all use a sandboxed app 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 fully-managed devices.

Partially Managed Devices

We now turn our sights to *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 the user's, but they agree to some level of corporate management.

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 supporting hardware (including all iOS models currently for sale).

Apple's iOS configuration profiles offer all the EAS features except remote wipe (which requires a remote wipe server), and the following in addition:

- On-demand VPN for specific domains (not all traffic, but all *enterprise* traffic).
- Manual VPN for access to corporate resources.
- Digital certificates for access to corporate resources (VPN or SSL).
- Installation of custom enterprise applications.
- 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).

The key differences between partially and fully managed devices are a) the user can still install arbitrary applications and make settings changes, and b) not all traffic is routed through a mandatory full-time VPN.

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.

Basic policies should include the following:

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

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 in those categories don't support use of their tools without a configuration profile which requires at least a passcode.

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 isn't used.

By configuring the policies listed above, all email is encrypted under the user's passcode using Data Protection. Some other content is not protected, but remote wipe is supported.

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 based on recent announcements we expect full editing capabilities for MS Office document formats.

User-owned device with managed/backhaul network (cloud or enterprise)

This option works with other data security tools, and isn't sufficient to protect data by itself. The users own their devices, but agree to route all traffic through an enterprise-managed network. This might be via a VPN back to the corporate network or through 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 more for malware protection and to reduce the likelihood of malicious apps on devices, but it also enables more complete DLP.

Managed Devices

Managed devices make life easier for security administrators. Full control of the device enables enforcement any policies desired, although users might not be thrilled.

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

MDM managed device with Data Protection

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 wiping. Some MDM tools even detect jailbroken devices.

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 to give users a bit more flexibility.

Managed/backhaul network with DLP, etc.

The device uses an on-demand VPN to route 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 it 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 certain other security options. For example, you might still need encryption or DRM, as these 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 that technology in a dynamic business environment, with users who have already made technology choices.

Factors

Most organizations we talk with – of all sizes and verticals – are under intense pressure to support iOS, to *expand* support of iOS, or to wrangle control over data security on iDevices already deployed and in active use. So 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 "ours or theirs". Although some companies are able to maintain strict management of everything that connects to their networks and accesses data, this has become the exception more 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.

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

The first question to ask yourself is whether you can maintain strict ownership of all devices you support – and 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.

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

If the employee owns the device you need their permission for any device management. Define simple and clear 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.

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. But fortunately most organizations have more flexibility.

The more sensitive the data, the more it needs to be isolated (or restricted from being on the device). This ties into both network security options (including DLP to prevent sensitive data from going to the device) and messaging/file access options (such as 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 may need a different tool.

Determine what the business *wants* to do with devices, then figure out which components you need to support that. And don't forget to look at what they are *already doing*, which might surprise you.

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 compliance requirements which could include anything from encryption to email archiving. Or even restricting the devices completely.

Make a Decision

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

- Determine the ownership model to support: personal, employer, or both.
- Determine which devices to support (we focused on iOS, 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. 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 your security controls yet – this part is more about figuring out *how* employees want to use the data on the device.
- Identify potential security controls/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 security risks.

But every time we face intense pressure to support new things in a short time frame, it is 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, to something as complex as multi-organization secure document workflows. 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's 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.