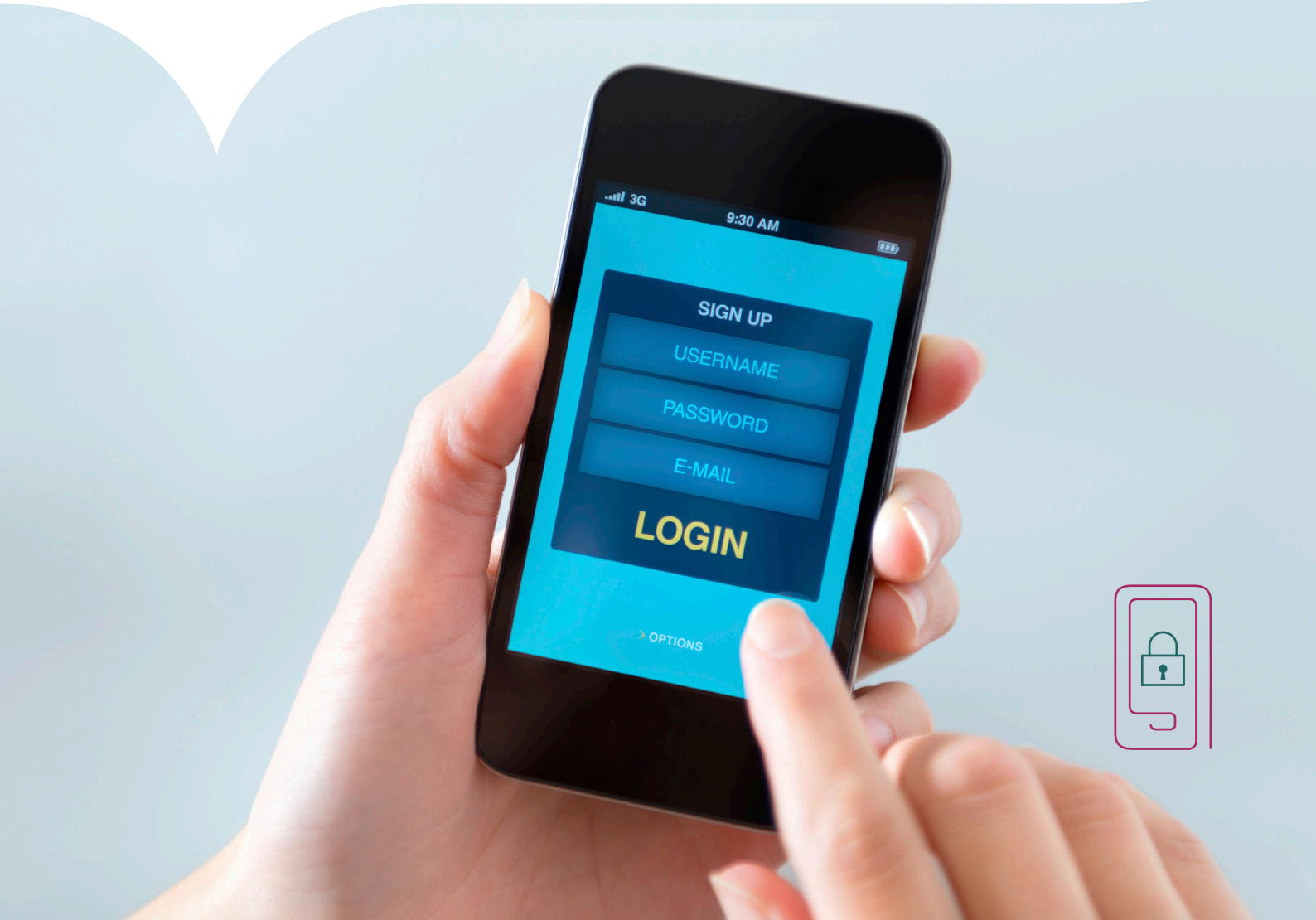
Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

SOGETI

Taking Mobile Security to the Next Level

Delivering Secured Mobile Solutions




9:41 AM
3G
Enter Passcode
OK
1
2 ABC
3 DEF
4 GHI
5 JKL
7 PQRS
8 TUV
0
Emergency Call

Introduction

The transformation of mobile communications brought about by the advent of tablets and smartphones over the last five years is one of the most dramatic in the history of technology. The number and variety of mobile devices in use are growing at an exponential rate[1]. For the enterprise, the correct mobile strategy can undoubtedly boost profits, productivity and profile.

However, concerns have been raised that this brave new mobile world is vulnerable to an increased level and a greater variety of security risks than the established world of IT. These threats are not specific to mobile security, but mobility itself brings additional risks. Creating the right security approach will help enterprises to address these threats while taking advantage of the huge benefits mobile offers.

There are three main targets for threats – information, identity, and availability.

A mobile device is personal, and is used for both private and business productivity even more in the context of a Bring Your Own Device (BYOD) company scenario. Devices store valuable and sensitive information. Keeping that information secure is vital. The consequences of a malicious third party gaining access to personal financial information, for example, could be disastrous and irreparable. With mobile devices gathering personal, often photographic, information about where the owner lives, works and spends his or her leisure time, the consequences of a mobile device getting into the wrong hands could be truly catastrophic. And with devices being used for work and leisure, the possibility of sensitive business information getting into the wrong hands could have much, much wider – and potentially massive – implications.

Moreover, the major new challenge to security is that information is now scattered rather than being centrally stored. For this reason, rather than being on the datacenter, security must now be on the transport medium and the information itself. Addressing the linked, unprecedented security challenges of scattered information being accessed and exchanged via mobile devices is the subject of this point of view.

This paper looks to examine the issues IT leadership will grapple with for enterprises with security concerns when adopting mobile solutions. Some enterprises will be exploring the policy-led challenges of Bring Your Own Device (BYOD)—this is a topic in itself and is covered in a separate paper.

In the following sections, this paper covers the enterprise need to design, implement, and maintain a mobile security architecture that mitigates risks by keeping data and identities secure while ensuring availability.

1 Source: http://www.kasalis.com/blog/2012/06/22/exponential-growth-in-smartphone-and-tablet-industry-fuels-need-for-high-quality-optical-components/

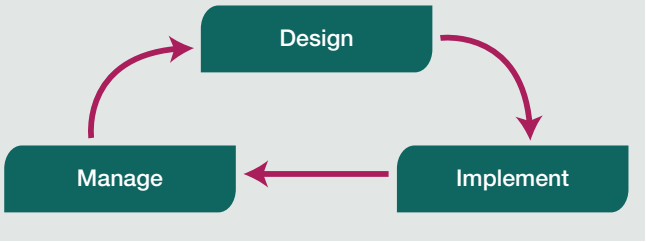
A Structured Approach to Mobile Security

Given the previously mentioned threats and many more, there is clearly a need for attention to mobile security. Several aspects must be considered when working with mobile security and apps (for which enterprise mobility is often predicated on):

- Confidentiality: Does the app keep private data private?
- Integrity: Can data passed to and from the app be trusted and verified?
- Authentication: Does the app verify the user's identity to an appropriate degree of certainty?
- Authorization: Does the app properly limit user privileges?
- Availability: Can an attacker harm the mobile solution in any way?
- Non-Repudiation: Does your app keep records of events?

A structured approach to working with mobile security is to define mobile security architecture with the following lifecycle (see Figure 1).

Figure 1: Lifecycle of a mobile security architecture



The lifecycle begins with the **design** of the mobile security architecture, and consists of a structured process that defines the contextual, conceptual, logical, and physical architectures for mobile security. It starts by defining the business requirements for mobile security in a contextual mobile security architecture, which is refined all the way down to the physical level.

The physical mobile security architecture defines the actual products and technologies used to **implement** the mobile security architecture. This is the lifecycle's second phase. It includes elements such as selecting mobile platforms, system design of mobile apps, handling secure access to data, secure transfer of data, secure storage of data, testing mobile security, as well as managing devices and apps.

The last, and probably most important, phase of the lifecycle defines how to **manage** the mobile security architecture over time. This involves keeping up to date on threats, improving implementation based on a changing technology landscape and best practices.

Designing a Mobile Security Architecture

Designing a secure corporate mobile environment can be done in four steps. We can summarize the whole process by giving an answer to the following questions:

- Why do we need to design a secure architecture for mobility? – Security principles and drivers for mobility
- What do we need to protect? –Assets to be protected. People involved.
- How do we protect the mobile environment? – Functions needed to achieve security.
- With what do we implement? – The physical aspects of mobile security such as material and location.

These four steps help define the business requirements for security and are the founding principles used to build sustainable mobile security architecture[2]. In essence, an audit of the situation with inventory of vulnerabilities is performed to inform the design requirements. In more detail:

- **WHY?** The Contextual Mobile Security Architecture

The contextual architecture takes in input from the business requirements and all the constraints (policies, guidance, legal, regulation) as well as assumptions. It will then define a clear and shared view on the scope and the principles that will drive the secure mobile architecture. This means there is ultimately a focus on data and application level security instead of relying only on network security only.

- **WHAT?** The Conceptual Mobile Security Architecture

The conceptual security architecture aims to identify the security requirements. The way to identify security requirements is mainly to perform a risk assessment; what are the most significant threats and consequently what are the security services that must be implemented to reduce the corresponding risks.

- **HOW?** The Logical Mobile Security Architecture

This logical architecture intends to provide a logical model which delivers the security services while conforming to the principles and models as set out in the Contextual Architecture and the Conceptual Mobile Security Architecture. The purpose of the Logical Security Architecture is to communicate how security should be implemented.

- **WITH WHAT?** The Physical Mobile Security Architecture

The physical architecture is the selection of technologies and products that will be used to implement the Logical Mobile Security Architecture patterns defined in the step before.

2 For further information on methodology applied here, see: www.sabsa.org/the-sabsa-method.aspx

3 Source: http://www.idc.com/getdoc.jsp?containerId=prUS23946013

In parallel runs the important task of defining how the mobile security architecture will be maintained and updated over time. The result is referred to as the Operational Mobile Security Architecture and is covered in the next section.

With risk assessment in hand and processes defined, the last piece in design is to validate the plan complies with applicable law and regulations. The legal framework a company must adhere to will be dictated by their own local and industry-related circumstances. In some countries, for example, companies have a responsibility for any malicious or illegal utilization of the platforms used by employees. In this situation, companies could use the legal framework to ensure employees comply with the right and secure way to use platforms. Attention to this legal step will ensure stakeholders are accountable and users compliant when it comes to implementing mobile security measures.

Implementing Mobile Security

Having established the approach of creating a mobile security architecture in a structured way, it's vital to look at the concrete challenges that need to be dealt with. The most important areas are:

- Mobile Platforms: Evaluate security considerations for iOS, Android, and Windows Phone
- Mobile Apps, websites, and architecture: Security for apps, websites accessed from mobile browsers, and the important role of a solid software architecture
- Access Control: Select an authentication mechanism
- Data in Transit: Choose how to encrypt data communication
- Data at Rest: Set up secure data storage and containerization
- Mobile Testing: Test the security aspects (confidentiality, integrity, etc.) of the mobile solution
- Mobile Enterprise Platforms: managing mobile devices, apps, and content in a secure way.

Mobile Platforms

Several mobile operating systems drive millions of applications on billions of devices. In February 2013, IDC reported Google's Android and Apple iOS as the two most prevalent, ahead of BlackBerry and Windows Phone[3]. Android is the only operating system built on open source. Its open nature, spread across multiple device manufacturers, means that manufacturers should have the policy to distribute updates at the required frequency, which is not always the case[4]. For this reason, security holes on Android devices can be left unpatched for a long time. The closed source code models of iOS and Windows Phone tend to update all devices within a matter of weeks of updates being available[5], thereby quickly fixing security issues.

Regarding mobile application distribution, all three operating systems have app stores with a built-in aim of preventing the downloading of malicious software (malware). In the

iOS App Store and Windows Phone Store, each app will go through an approval process before they are made available to users, making it significantly harder for malware to be spread. There is no such rigorous process for Android apps in Google's Play Store. High risk apps are removed but the risk of distributing apps tainted with malware is clearly higher with Android devices. The fact that it's even possible to install apps not distributed through Google Play, further increases that risk. Windows Phone and iOS don't allow distribution of apps from outside their own app stores except in very specific enterprise cases.

It is possible to remove built-in security restrictions on devices that use any of the three operating systems by "jailbreaking" or "rooting". Indeed, over fourteen million devices on iOS 6.x have been jailbroken[6]. So policies and solutions must be implemented to counter this vulnerability. The enterprise policy should indicate whether users are supported if they choose to modify their device and a technical solution could be implemented to test for such jailbreaks (while noting jailbreaks are not always detectable by automatic means).

Two features of Android and Windows Phone that iOS owners don't have access to (and are therefore safe from), are the ability to use SD cards and USB Mass Storage. In the case of SD cards, neither Android or Windows Phone encrypt the content by default and therefore there are two risks: that sensitive information can be leaked and that malware can enter the mobile device. Similarly, when the device is connected to the computer and allowed to be used as USB storage, this also means that information can leak and malware enter the mobile device.

It is relatively easy to reverse-engineer complied Java[7] (Android) and .NET[8] (Windows Phone). The dynamic nature of the Objective C language used within iOS also enables users to reverse-engineer applications. This ability to reverse-engineer an app to reveal its source code, can provide valuable information to hackers[9].

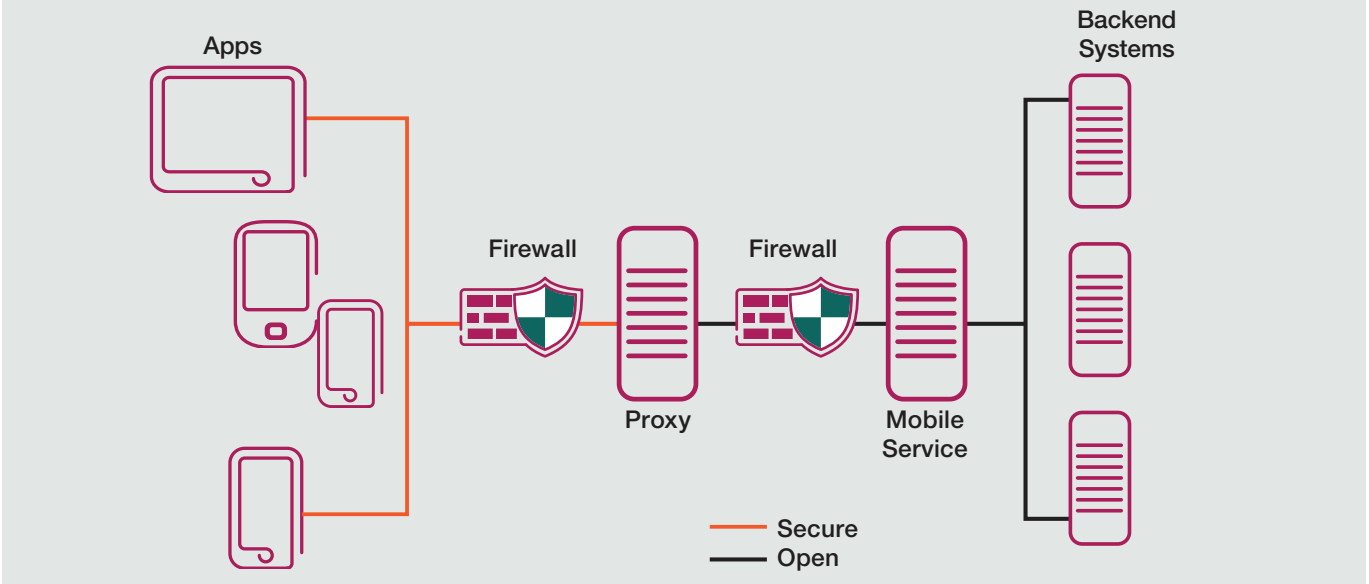
When addressing what operating systems should be supported by the corporate infrastructure, it's important to consider the differences in security features of each platform.

Mobile Apps, Websites, and Architecture

There are two main ways to deliver content and functionality to mobile devices: via mobile apps or via websites that are viewed through a web browser.

The majority of apps available are written in a programming language that is platform-specific. All are compiled into a binary executable file that is made to download and run entirely on each specific platform. These are generally referred to as native apps. However, most of these apps also include some web content and functionality that is either distributed with the app or accessed in real-time from a web server through a native component (often called a web view). Such apps are referred to as hybrid apps. The parts of a native app that consist of web content and functionality have the same security risks as any web site run in the browser (see Figure 2).

Figure 2: A typical software architecture for a mobile solution



4 Source: http://www.slashgear.com/aclu-asks-feds-to-investigate-major-carriers-over-infrequent-android-updates-17278278/

5 Source: http://news.cnet.com/8301-13579_3-57450474-37/apple-365-million-ios-devices-sold-80-percent-running-ios-57

6 Source: http://www.iphonehacks.com/2013/03/over-14-million-ios-6-devices-jailbroken.html

7 Source: http://android.amberfog.com/?p=582

8 Source: http://www.net-security.org/secworld.php?id=12253

9 Source: http://media.hacking-lab.com/scs3/scs3_pdf/SCS3_2011_Bachmann.pdf

The fact that developers of native (and hybrid) apps have full control of the functionality implemented is both a security risk and an opportunity. Therefore, any organizations implementing these kinds of apps need to outline software architectural and coding guidelines that focus on mobile security issues and recommendations.

For example, a solid software architecture would consider a mobile (or multi-channel) service on the intranet that is made accessible in a secure way (e.g. via SSL and Basic Authentication) from the Internet through a reverse proxy. That way none of the backend systems are exposed directly, and the integrations are made easier in a secure environment (the intranet).

The security considerations for websites accessed through the browser on a mobile device are mostly the same as for any web application, and include risks such as cross-site scripting and request forgery, broken access control, file inclusion and creation, and various types of injection (SQL, command, scripting, etc). There are also some specific mobile security risks that relate to the form factor, such as smaller screens. For example, the URL is often abbreviated, preventing the user from knowing what site is actually accessed and any mobile browser specific security flaws.

The best general advice is to include mobile browsers on the most important platforms when designing, implementing, and testing websites for both customers and employees.

Access Control

Depending on the nature of the enterprise there are several established means of authenticating users. Many systems authenticate users via username and password. Others use one-time passwords. Once a user is logged in there is normally policy around how long they can remain logged in before the session expires. This traditional approach may not be ideal for mobile.

In deciding to use a native (or hybrid) app, certain things must be considered:

- Must the user enter their username and password each time they use the application or can it be remembered?
- How long should a session last?
- Must users enter their credentials every time the device is unlocked or resumes from sleep?
- If the password is invalid what should be done with data on the device?

How these questions are answered will inform the organization's security needs as well as impact the user experience. It will be necessary to strike a balance between security and user experience based on needs. It is important to assess the risks being created by providing a better user experience.

Instead of requiring the username and password every time, consider a design that uses an alternative password solely for the mobile app. For example, the user might first log in using their username and password, they would then generate a short, four digit PIN to access the application in future. Although this password itself is less secure than a normal password it can be designed to work only from that one mobile device and through no other service and can be revoked if the device is compromised without inconveniencing the whole account.

Table 1: Two-factor versus one-factor authentication

	One Factor	Two Factor
User requires	A username and password; something they know.	A username and password; something they know. Something they have (i.e. a keyfob, NFC identifier or other similar device) or something they are (i.e. biometrics).
Works offline	If password has been cached at least once.	Typically No – need to confirm the "something user has" online.
User experience	Easy – quick to access.	Poor – takes additional time. User must ensure they have "something user has" to access the app.
Security risk	Single point of failure if password leaks, may expose other systems.	Minimal – no single piece of information can gain access to the system and "something user has" can be deactivated remotely.
Worst issue	Security risk.	User experience and user forgetting their identifier.

Two-factor authentication (TFA) for apps might be considered. Popular with online banking and corporate VPN applications, this is where, in addition to a password or PIN, a user has a separate device that generates a one-time password. It is ideal in areas requiring extra security. As near-field-communication (NFC) becomes more popular there are possibilities to develop TFA solutions using NFC rather than one-time passwords. Such authentication will always be more secure but it requires users to carry another item of technology. What impact would this have on users and would it limit people wanting to use the app?

Associated to authentication are user interface and spyware issues. By essence, mobile apps are used in public spaces with surrounding people who may not be known to the user. Warding against the curious or malicious onlooker, the recommendation would be to design a secure Graphical User Interface that uses a random numeric keypad, or ensures a password is not displayed in clear text. It is also recommended to equip corporate smartphones with a privacy screen filter.

The key to any decisions around authentication involves trade-offs between user experience and risk. The more layers of complexity are added the more risk of frustrating the user. What is being stored on the device? And if it were compromised how will this impact the enterprise's business? Is the act of creating a more secure authentication system due to perceived rather than actual risks? Once these issues are understood it will be possible to select the most suitable authentication mechanism. This may vary from app to app and from internal to external user depending on the overall mobile strategy.

Internet access control and interface blockage issues must also be addressed. These issues cannot be covered in detail within this paper but will include how to address direct access authorization, prohibition of split tunneling, and compulsory use of company Internet access. One way to avoid counterproductive employee activity, for example, is to establish a company policy that requires users to agree that when a device is being used for work, it will be on the company network. If those employees know they must use the company's wireless network, they will be more inclined to keep activities to company policy.

Data in Transit

There are three main ways to securely transmit data to mobile devices: VPN, SSL, or custom encryption using third party or bespoke solutions. VPNs are the known and standard way to secure sensitive data transmission for employees within an organization that requires access from a remote site. In most organizations they are used to access internal systems and are designed for traditional, not mobile, computing applications. Although VPNs can be used for mobile devices there are a number of drawbacks (unreliable and slow over 2G/3G connections, significant configuration overhead, poor user experience, etc).

Ideally, a mobile security strategy that works for both internal and external users and that complements the mobile experience should be designed. The most popular way to secure data in transport from the physical connection is to use HTTPS (secure HTTP or SSL, more correctly referred to as TLS). The technology is tried and trusted and for all but the most secure data, when twinned with suitable authentication and access control mechanisms, offers a high level of security.

Table 2: Comparison of data in transit methods

	VPN	SSL	Bespoke
User Configuration	Difficult.	None – transparent to user.	Depends on solution – normally one-off and occasional password entry.
Mobile Suitability	Low – poor user experience.	High – transparent to user.	High – designed specifically for mobile apps.
Secure Transmission	High.	Medium – potential of man in the middle attacks.	High.
Development	Easy – no additional overhead.	Easy – minimal additional checks to prevent man in the middle attacks.	High – integration of third party components or complex key exchange.
Testing	Easy – no additional overhead.	Medium – minimal testing for man in the middle attacks.	High – need to ensure key exchange, storage and data integrity are secure.

There are two approaches to bespoke encryption if there is a question of whether or not HTTPS will provide sufficient encryption. Transmitting all data with bespoke encryption will necessitate writing a native (or hybrid) application to take advantage of the device's capabilities. Investing in bespoke encryption can be both time-consuming and risky. Without a suitable strategy around key exchange and storage, there is a risk of data being less secure than if SSL is used.

If there is a willingness to invest in third party components, security frameworks and mobile enterprise application platforms (MEAPs) exist that can deliver high levels of security. Some of these have received FIPS 140-2 accreditation in the USA and similar levels elsewhere. There are many third party options to consider. Some are capable of securing data at rest, as well as data in transit.

Data at Rest

In a native (or hybrid) app, consideration must be given to what is done with any data that is downloaded. It can be stored on the device's internal storage but that obviously poses a security risk. To store data securely requires it to be encrypted. There are two ways to do this: either relying on the device's built in encryption systems or by using some form of bespoke or third party encryption technology within the application.

It may be necessary to store many details: usernames, passwords, authentication tokens, encryption keys and personal data related to the user or the organization. Taking additional precautions to protect access to anything stored and data backup is prudent. Depending on the level of security needed there are several approaches.

Most mobile operating systems support some form of device encryption – this is operating system data protection. Android, iOS and Windows Phone all operate with full disk encryption, similar to that employed on laptops. The entire device is encrypted when it is not being used, offering great protection for stolen devices being cracked but none from malware on the device itself. These forms of encryption tend to rely on the user's device PIN, but if the PIN is a simple four-digit number it is often possible to decrypt the data forensically in a matter of minutes[10]. Requiring longer, more complex alphanumeric passwords has been proven to be secure[11]. The information being stored on the device will determine whether these forms of encryption are sufficient, but what if a device is lost or stolen? It may be possible to wipe a corporate device, using a mobile device management (MDM) system to protect the data, but what wider impact would occur if a customer's device were stolen?

If these standard operating system levels of protection are insufficient, it may be necessary to consider third party or custom solutions to encryption. This can be baked into the development process. The ultimate level of security can be obtained by using a secure container on the device where sensitive information is stored separately from the standard OS data on the device, even being protected in the case of a jailbreak or rooting.

Table 3: Operating system versus containerization protection

	Operating System	Containerization
Lost Device	No inherent protection. For iOS, where user is registered for iCloud, they are offered capability to wipe the entire device. Android device users using Google Sync are also offered a secure wipe capability.	Offer additional mechanisms to selectively wipe the device. Data in the container can include not only PIM data but Enterprise applications and data.
Speed to access data	Quick – data is encrypted once the device is started and unlocked.	Can be slower – data decrypted on-demand and also with more CPU intensive algorithms.
Malware Protection	Minimal – most operating systems decrypt entire disk once booted meaning malware can access anything.	Protected – any data stored within the application while at rest is encrypted so reading the data would be meaningless to malware on the device.
Jailbreak Protection	No – a jailbreak or root attack will reveal the operating system. Native OS does not block or prevent operation of jailbroken/rooted devices.	Yes – data and/or applications protected by containers can be blocked or removed upon jailbreak/root detection.
Forensic Protection	Yes – device is fully encrypted meaning without the keys brute force attempts are required, although standard passcodes used for devices may be weak if only four characters long. Subject to known exploits with OS native algorithms.	Yes with stronger encryption algorithms than the native OS – the data is fully encrypted. If device-level encryption is also used there are two forms of encryption that would have to be broken. Algorithms used by containers are stronger than the native OS.
Third Party / Built-In Apps	Yes – applies to entire device so protects third party apps as well as standard device apps such as e-mail or calendars	No – typically for applications created using such technology. Depending on the selected technology container it may protect built-in apps or 3rd party apps not aware of the container technology.

There are numerous ways to protect data containerization via mobile content management (MCM). This approach creates a silo on the device whereby local databases and file access are secured in addition to data transmission to the device[12]. MCM provides a set of tools and functions to implement forms of containerization and typically includes authentication, encryption, compliance, geo-fencing, access control and sharing within the organization. This allows the leverage of data protection without creating a bespoke solution from scratch. Mobile application management (MAM) can be twinned with an MCM solution to control who can access and download applications to a device in the first place.

Mobile device management systems (MDMs) are extending their functionality to implement MCM and MAM functionality. This functionality set is called Enterprise Mobility Management (EMM) and is currently limited to a small subset of players. There are also third party MCM solutions just to implement containerization without device-level control. Depending on whether an app is internal or external facing will help to determine the right approach for the organization.

If an off-the-shelf solution is not required, or a suitable one is unavailable, it is possible to create such a solution from scratch, but the amount of work can be extensive. Third party tools that have already received security accreditations can vastly reduce the time to market, while increasing security.

Websites that are accessed through the browser on a mobile device are concerned with data that may be stored in the browser's cache. Similar approaches can be taken with mobile web apps as with traditional web apps to prevent caching (by using HTTP headers). With HTML5 storage it will be necessary to ensure that anything stored is low-risk: securing the HTML5 environment is hard to control.

Circling back to the availability target for threats and the vulnerabilities identified in the design phase, protection against the risk of malware injection in the devices must also be addressed according to the sensitivity of the context of use. Malware attacks are on the rise, with mobile malware hitting an all time high[13]. Malware includes viruses, worms, Trojan horses, keyloggers, and other malicious software programs. To avoid compromise of data through the injection of malware in the devices, special measures such as anti-virus and rapid application of security patches on the device should be implemented.

In all of the above, it is possible to place too much emphasis on concerns over data at rest. As with all aspects of mobility the key question should be risk mitigation. There are several options depending on what is at risk. The question should be asked of how much more at risk a mobile device might be than other access channels. With customer-facing websites, how much protection has been created in case customers have key logging and Trojan viruses on their computer? In many cases, home computers are more susceptible to a range of malware than mobile devices. If possible, risk levels and the cross-channel approach should be matched.

Mobile security testing is becoming increasingly **important**. Validating specified and implemented security measures often reveal **critical security holes and threats.**

Mobile Security Testing

Mobile security testing is becoming increasingly important. Validating specified and implemented security measures often reveal critical security holes and threats. In a typical mobile security testing effort tools can be used to validate the common security aspects:

- **Confidentiality:** Does the app keep your private data private? Penetrate data storage locations looking for private data or data that should have been deleted during app exit. Analyze network traffic and validate whether or not sensitive information is appropriately encrypted.
- **Integrity:** Can the data passed to and from the app be trusted and verified? Validate the integrity of the data being passed to and from the app by monitoring network traffic and, where relevant, validate whether or not the data is appropriately encrypted.
- **Authentication:** Does the app verify the user's identity to an appropriate degree of certainty? Test if the right level of authentication is implemented or not, for example, by validating the implementation of two factor authentication or by checking correct round tripping of mail-based confirmations.
- **Authorization:** Does the app properly limit user privileges? Test if server-based services are provided at the right level of privileges and only there, by trying to invoke functions or reaching for data beyond authenticated users' privileges.

10 Source: http://www.iosresearch.org
http://resources.infosecinstitute.com/iphone-penetration-testing-3
http://sit.sit.fraunhofer.de/studies/en/sc-iphone-passwords-faq.pdf

11 Source: http://www.macrumors.com/2012/08/13/apples-unbreakable-ios-device-encryption-highlighted

12 Source: http://www.consumerizeit.com/blogs/consumerization/archive/2012/02/03/byod-vendor-briefing-notes-good-technology.aspx

13 Source: http://www.webpronews.com/mcafee-reports-that-malware-is-on-the-rise-2012-09

- **Availability:** Can an attacker harm the solution in any way? Apply common attacking methods on the server-based services by first monitoring open network traffic, then trying to either get to restricted functions or data or trying to halt the entire service.
- **Non-Repudiation:** Does your app keep records of events? Where relevant, validate both client and server logs to make sure that it's possible to use them to prove the user's activities through the mobile solution.

The focus here is exclusively on the mobile solution's security, disregarding functional and usability aspects that are addressed in other types of testing. It is possible to engage through both manual and automated tests using high profile and market leading security testing tools.

Mobile Enterprise Platforms

To support a mobile security architecture, there are a several commercial products available for MDM, MAM and MCM. They provide various security features during the lifecycle of devices, apps, and content. The lifecycle typically consist of three phases, beginning with provisioning (when the device, app, and content is first installed), followed by production (when the device, app, and content is in use), and ends with decommission (when the device, app, or content is lost or removed). Some of the security features that are important during each phase are as follows:

- **Provisioning:** Establish policies and configuration, e.g. initialize power-on password, install and secure (encrypt) apps and data, install and configure antivirus and firewall
- **Production:** Back-up data, update apps, apply patches and security updates, enforce updated security policies, monitor and track security violations and threats, compliance activity logging
- **Decommission:** Disable a lost or stolen device, remote lock, wipe and kill, access violation lock, data fading and time bombs, disable device, network, app, or data access.

Even if similar solutions can be realized using custom implementations, these MDM, MAM, and MCM systems will make it easier to handle many of the security risks already mentioned, related to access control, secure data communication, and secure data storage.

Managing the Mobile Security Architecture

Ideally, a mobile security architecture is not created as a one-off effort, it is a living thing that needs to be maintained and applied constantly. It's a reference that should be used by project teams as they design and implement their specific mobile solutions. However, the world is constantly changing. Business requirements evolve, and the front end of the architecture, the contextual architecture, must be reviewed and updated periodically. An important question is: at what point do the contextual changes create sufficient pressure to change the underlying conceptual architecture and other layers?

Security is a vital part of any large **IT deployment** and is arguably even more important with mobile devices. The key to securing these devices is **understanding the risks** posed to the enterprise by their use.

The changing behavior of mobile users affects the security aspects of the solutions they use. There is also a need to keep pace with changes in the world of mobile security, such as new threats and best practices to handle them. The question arises: how do we monitor and measure the security aspects of our mobile solutions and keep up to date with changes affecting security in the mobile world? Technology also changes and new mobile security solutions become available. This also raises a question: when should you change decisions in the physical architecture from one technology or product to another? These questions suggest a continual architecture review process that is governed in a structured way and monitors how well mobile operations are performing to meet business security requirements.

Crucial to ongoing efficiency and security of mobile solutions is user awareness. All the required attention to architecture and operations is worth nothing unless continued effort is made to ensure users are aware and comply with relevant and up to date policy and governance.

Conclusion

Security is a vital part of any large IT deployment and is arguably even more important with mobile devices. The key to securing these devices is understanding the risks posed to the enterprise by their use.

The trend towards "scattering" of information across many locations has grown in parallel with the uptake of mobile devices for personal and business use. With the acceptance of mobile device use within the enterprise, these separate, but related trends offer many benefits, but present many new and unprecedented security challenges.

A careful selection and enforcement of technologies, policies and governance can ensure that a mobile strategy provides adequate support to build secured mobile solutions and, potentially, could be more secure than current systems. A suitable strategy will allow the enterprise to reap the business benefits of the mobile revolution, while safeguarding it and its employees and clients from the risks.

Acknowledgement

Thanks are due to various contributors:

- Andreas Sjöström
- Christian Forsberg
- Guy Powell
- Hans Scholten
- Jérôme Filipozzi
- Stéphane Janichewski, and
- Tobias Hutzler

About Capgemini and Sogeti

With more than 125,000 people in 44 countries, Capgemini is one of the world's foremost providers of consulting, technology and outsourcing services. The Group reported 2012 global revenues of EUR 10.3 billion. Together with its clients, Capgemini creates and delivers business and technology solutions that fit needs and drive the results they want. A deeply multicultural organization, Capgemini has developed its own way of working, the Collaborative Business Experience™, and draws on Rightshore®, its worldwide delivery model.

Sogeti is a wholly-owned subsidiary of Cap Gemini S.A., providing local professional services, specializing in Application Management, Infrastructure Management and High-Tech Engineering. Sogeti offers cutting-edge solutions around Testing, Business Intelligence, Mobility, Cloud and Security. Sogeti brings together more than 20,000 professionals in 15 countries and is present in over 100 locations.

Capgemini and Sogeti offer end-to-end Mobile Solutions for mobile strategy and services as an Enterprise Mobility Orchestrator. Deploying a framework of harmonized methods, accelerators and industrialized services, the Enterprise Mobility Orchestrator services can help create, implement and support an organization's mobile strategy. To address all areas of a business going mobile, the service portfolio covers: Strategy; Mobile Applications; Mobile Platforms; Managed Mobility; and Reselling. Together, Capgemini and Sogeti have combined their extensive capabilities in strategic consulting, technology excellence, industry solutions and global delivery to help organizations optimize their mobile business potential.

Learn more about us at
www.capgemini.com/mobility
www.sogeti.com/mobile-security

For more information, please contact:

Josean Mendez Rios
Global Chief Mobile Technology Architect, Capgemini
jose.mendez-rios@capgemini.com

Stéphane Janichewski
Head of Security Global Line, Sogeti
stephane.janichewski@sogeti.com