Kony Solutions

A Kony White Paper

DECEMBER 2010

SECURING A MOBILE EDGE

MULTI-PLATFORM MOBILE APPLICATION DEVELOPMENT

Avoiding Lowest Common Denominator Security

Table of Contents

Introduction

The mobile platform has become ubiquitous for conducting business and engaging with consumers. The always-connected, portable devices give direct access to systems and data, anywhere a cellular or wireless connection can be established.

But, with mobile application flexibility comes complexity and insecurity. While the mobile OS providers have attempted to build a more secure mobile operating system s compared to their desktop ancestors, applications can still be built using insecure coding practices, on top of insecure development platforms, and with insecure features and functions. As security is often an afterthought in application development, the tradeoff between delivering a secure application late and delivering a functional application on time often gets made at the expense of security. However, this doesn't have to be the case; selecting the right platform can enable secure applications to be built across multiple platforms, delivering them with full functionality, on time.

The Importance of a Secure Mobile Application Development Platform

Portable devices and the mobile applications that run on them are must-have business tools for today's organizations. Mobile applications are designed to provide features, functions, and benefits for on-the-go customers and employees alike. However, as with any tool that conveniently connects various users to business systems and data, we can often experience significant risk; threats to take down mobile security systems or even the prospect of a breach of personal data or confidential business information.

As more users turn to mobile devices for conducting both business and personal transactions, the demand for greater functionality and rich computing capabilities once relegated to desktops and laptops have aggressively moved to the mobile space. Mobile devices and applications allow for relatively complex, multi-system, multi-user business transactions to take place from anywhere, at any time; each of which could involve money changing hands, financial accounts being managed, goods and services being sent through the supply chain, sensitive business data being accessed, and private or personal information being supplied to a service provider.

With each of these transactions comes the risk of system compromise, system downtime, unauthorized access, and data theft. The risks associated with this move toward complete mobility continue to both evolve and expand as an increasing number of users engage with each other through public networks and cloud-based services. An overwhelming amount of sensitive data is shuttled amongst users, service providers, cloud hosting locations, and social network databases.

History Repeats Itself, Only Faster

The evolution of the desktop operating platform is one that began simple and grew; first, as a single mass-market operating system with a move to include a handful of key operating systems.

The desktop platform's connection points and data transfer capabilities primarily included floppy drives and later moved to local area networks, followed by the use of modem-based wide area networks, and then to high-speed Internet connections. Ultimately, I/O ports (USB, FireWire, etc.), and sometimes the removable storage (SD cards), become a common way to exchange data.

Similarly, the move from simple command-line user interfaces grew to include a variety of graphical interfaces and then browser-based interfaces that connect to multiple networks through a multitude of protocols – some secure, some insecure.

The applications available on the platform came primarily from the platform providers themselves – but this quickly changed, as third-party ISVs (independent software vendors) began to develop their own applications to run on the desktop platforms.

The mobile platform is very similar to this model. The only real difference being in that it is growing at speeds far greater than that of the early desktop platform space. One might argue that the mobile platforms are much more stable, secure, and expandable than its desktop-based cousin. But, others could argue that the rate of growth limits the ability of the mobile platform to adequately control how it is used, securely, while enabling as many features as available to its counterpart at the desktop.

In short, the risk surface of the mobile platform is as equal to that of the desktop platform, if not more, due to the fact that there are fewer controls, fewer organizations focusing on security for these platforms, and a much broader use and mix of both business users and consumers using and sharing a mix of business and personal applications and information.

Just like building applications for the now-ubiquitous desktop operating platform, directly building application to run on each mobile operating system doesn't inherently address the risks related to system and data security; security must be thought through when building an application for the iPhone just as it does when building an application for the Windows desktop operating system.

Building an application to run across an entire set of mobile operating systems such that all mobile operating systems are brought to market at the same time certainly adds complexity to the delivery of the application. Additionally, building a secure mobile application for each platform may have its own challenges as each could possess their own individual intricacies for how they handle data storage, network connections, login procedures, and more.

Of course, simply relying on a platform abstraction layer that covers 'all' of the functions available across 'all' mobile platforms in one fail swoop doesn't necessarily mean that the abstraction layer itself is secure. This then means that what gets built using an insecure application could ultimately mean that the application is now insecure across all platforms.

To bring this set of challenges back up a level to where it really matters, we can categorize the risks associated with building and maintaining mobile applications into three main risk areas: Business, Operations, and Brand.

Business Risk

Business is driven by having a product or service that solves a problem in a way that the consumers of the product/service will pay for it and successfully use it. Compromise the company's ability to do this, and the business will likely fail. Below are some clear examples of where affected product quality, service stability, or customer service delivery could impact the success of the business:

Fraud: Applications that collect, store, transfer, and use credit card and other financial information expose these capabilities and data to users that could misuse them and/or hi-jack them to perform fraudulent credit card transactions, account skimming, or other inappropriate account activity. Each of these inappropriate activities could cost the organization significant time and money to track down the culprit, recover from the damage, and resolve the problem both near term (disaster recovery) and long term (fix the root cause of the problem).

Unauthorized Access: There are many ways for individuals and software to gain unauthorized access to the device, its connections, and the data. Picture a multi-dimensional matrix of users, applications, networks, physical inputs/outputs, and data storage locations (memory, device, memory) – each with the ability to create, modify, and communicate with the other.

With a trusted system, it is easy to see how access to the network via an application would expose data stored on the device to the network. Similarly, one can understand how easy it would be to inject malicious code through the installation of an infected application, granting it the same rights as the user that installed it, thereby giving it access to the network and data.

Each and every aspect of the mobile operating environment can be used to gain unauthorized access to the rest of the operating environment; the device itself, the cellular and wifi networks, local and SD storage, USB connections to a desktop, and intercepted communications and data transmissions.

System Breach: Applications that leverage user logins, network connections, and data transfer capabilities expose the mobile device to attack, primarily through application vulnerabilities introduced via the application development process. For example, an improperly coded application could leave a hole open for would-be hackers to tap in to, granting them access to the

applications' functions, back-end services, and data.

Once access has been granted, the mobile device can then be used maliciously to perform all sorts of dirty deeds ranging from performing mobile-driven denial of service attacks against the application's server to take it offline, compromising the application's in order server to gain access to its functions/services such that they can manipulate the transactions in their favor, or even by directly stealing business and/or personal information that is stored on the application's server for sale on the black market.

A system compromise can introduce a series of serious disruptions to a business; time spent identifying the culprit device(s), time spent identifying the vulnerability or other reason for the breach, time and money spent rectifying the problem(s) on the device(s) and the server(s), and time cleaning up from the payload left behind by the breach (restoring the systems, recovering the data, notifying the victims of the breach, managing any legal activities and related costs, and more).

System Downtime: Similar to the system breach scenario, the system downtime scenario can dramatically impact business operations and revenue generation. In this case, however, it is less about a system being compromised to gain access to system functions and data, but rather to take the system down and offline in order to prevent business transactions from taking place. With a system taken down, an organization can find it difficult, maybe even impossible, to recover and recoup the revenue lost during the outage.

Brand Reputation: Even if the business stays up and running, the operations team is able to combat most of the attempted attacks, and the security team is able to quickly clean up from any successful attacks that surface, the damage to an organization's brand is essentially irreparable if they lose their customer's financial or personal information.

Mobile applications are a great way to build a cutting edge brand for a company. Conversely, all it takes is one high profile news spot of a security breach, and it could easily be the one tool that destroys the brand in a matter of seconds.

What good are the systems, applications, and supporting business processes if the customers, employees, and partners can't trust the company building and running them?

Regulations and Compliance

For the cases where security is not driven by a desire to protect the business and its users, the government and large industry institutions have stepped in to help guide organizations along. Below are a few of the most common examples:

PCI-DSS: The Payment Card Industry Data Security Standard (PCI-DSS) is a worldwide set of standards which mandates the configuration of firewalls, passwords, encryption technology, and general maintenance of secure systems and applications in order to protect cardholder information. Established by the Payment Card Industry Security Standards Council, its primary goal is to prevent credit card fraud. The standard is enforced by Visa/MasterCard acquirers as well as by American Express directly. Organizations and merchants who process electronic credit card transactions must adhere to an annual assessment to determine compliance. Any and all mobile applications and related back-end services that touch cardholder data in any way would be subject to this assessment.

HIPAA: The Health Insurance Portability & Accountability Act (HIPAA) is a set of provisions and programs used to protect health insurance coverage for individuals and families, and to prevent health care fraud and abuse. Title II of HIPAA specifically regulates and controls the disclosure, transmission, and billing of health care data, benefits, eligibility, notifications, reviews and claims, including retail pharmacy claims. HIPAA Security Standards defines administrative, physical and technical safeguards to protect data

from interception and other malicious activity as it gets transmitted across open networks. Any mobile applications that allow users to interact with their personally-identifiable health information would be subject to the requirements imposed by this act.

SOX: The Sarbanes-Oxley Act (SOX) is a U.S. federal law that set the initial standards for U.S. auditing accountability and responsibility. The law governs management, the boards of public companies, and the public accounting firms. Compliance encourages companies to centralize data in order to prevent financial reporting fraud. Mobile applications that operate using data related to publicly-traded companies and firms are subject to the requirements and guidelines defined by SOX.

FISMA: The Federal Information Security Management Act (FISMA) is a U.S. federal law that requires federal agencies to implement data security systems to protect data confidentiality and availability, and to prevent unauthorized access, modification, destruction or other malicious activity. Mobile applications and services interacting with data related to the US Federal government would be subject to the requirements under the FISMA standards.

Mobile Threats Are on the Rise

While there is some debate over the breadth and depth of threats faced by the mobile device space, there is certainly no lack of sample attacks to draw from.

Backdoor invasions, exploited web hosts, stolen identities, stolen data, decrypted secret keys, cloned hosts, and unauthorized servers are just some of the weapons hackers can readily deploy to compromise systems and gain access to sensitive data.

Below are some common actions used by a variety of mobile attacks:

- The malicious software is launched on the mobile device by an unsuspecting user clicking a link in an email, downloading and launching an infected application, or through self-propagation by the malicious software itself via one of the networks available to the mobile device (Bluetooth, WiFi, or even the mobile device's cellular network)
- The malicious software gains access to the root level of operating system, changes the default root operating system password to prevent authorized administrator login attempts – preventing the administrator from detecting and cleaning up from the compromise
- The malicious software installs itself on the device and then copies itself to any removable media found on the device (such as an integrated SD card)
- The malicious software attempts to send MMS messages to every contact within the device where the MMS would contain an infected file, which, upon open, would infect the receiving mobile device, thereby further propagating itself
- The malicious software attempts to further propagate itself through the networks available on the mobile device (Bluetooth, WiFi, and the cellular network)
    - Note: constant attempts to propagate through the cellular network will eventually cause the battery to drain on the mobile device
- The malicious software collects valuable information stored on the phone (such as banking information sent to the user via a SMS) and send it off to a control host using one of the available networks on the mobile device
- After grabbing the information it needs, the malicious software damages and/or replaces the system files to prevent the mobile device from starting up correctly (if at all) at its next reboot

Breaking this set of actions down, the following can be analyzed to see how and where insecure applications can be exploited to gain control of a mobile system and access to its data and connected networks.

Users/Accounts

Users can do things accidently which put the mobile device in harm's way, unbeknownst to them.

Malicious software can leverage the rights granted to the mobile device's user account(s) such that they can use those rights to compromise the system and to gain access to functions, systems, and data. The malicious software can leverage vulnerabilities within the operating system to gain root level (core level) access to the system, its networks, and its data.

Real-World Scenario: Android OS Compromised at Black Hat 2010

At the Black Hat 2010 Conference in Las Vegas, NV, researchers from Lookout Mobile Security were able to exploit a known, un-patched vulnerability known as CVE-2009-1185 to gain root level access and control of the Android operating system.

According to the researchers as reported in a TechWorld article[1], part of the permissions system in Android allows applications to tap into each other's resources. An application without permission to access the Internet might have access to another application on the device that does have Internet access and use the Internet resources through that application.

Network Connections

Networks and network connections can be used by both authorized and unauthorized users and processes (applications) to conduct fraudulent and/or otherwise damaging transactions.

The networks can be used by malicious software, accessing them covertly behind the scenes, to regularly send information from the device and its user(s) back to the control server on the other end of the network.

Data Transfers

Improperly and unprotected transfers of confidential and sensitive information can result in the data finding its way into the hands of the wrong entity; a hacker, a thief, or a competitor.

Data Storage

Improperly and insecurely stored data – both at the mobile device and at the hosting service – can leave the data open to leakage to inappropriate parties and theft by malicious parties.

Application Service Hosting

An insecure hosting service with insecure supporting networks and networked systems (such as the mail gateway or database system connected to the service) could leave the entire process, its systems, and it data at risk.

Mobile Platform Security Challenges

In addition to the mobile platform being new and unique, the drive to deliver applications for multiple operating environments all at once coupled can present a significant challenge. Add to this the fact that these mobile devices are powered by public-facing networking services that collect and use both personal and business information to provide value to their users makes the need to incorporate security even more challenging.

The Mobile Platform is Unique

The mobile platform can be very similar to the desktop platform in that the threat landscape increases with each **different mobile operating system** introduced to operate the devices:

- iPhone
- Blackberry
- Android
- Windows Mobile
- Symbian
- Palm
- Java
- Brew

…with each new operating permutation that arises when crossing **different mobile web connections**:

- Internet Explorer
- Firefox
- Safari
- Blackberry
- Google Chrome

… and where the devices are connecting to a variety of desktop-enabled and server-side components running **different desktop operating systems**:

- Windows
- Mac OSX
- Linux

However, the mobile operating platform begins to differentiate itself as the devices are connected to the corporate network and the Internet through built-in **network channels**:

- GSM (Global System for Mobile Communications)
- CDMA (Code Division Multiple Access)
- WiFi (Wireless Fidelity)
- Bluetooth

…such as those provided by the following **mobile service providers**:

- AT&T
- Sprint
- Verizon
- Boingo
- T-Mobile

…when the users are able to employ **mobile-specific communication channels** to transfer information and files:

- SMS (Short Message Service)
- MMS (Multimedia Message Service)
- BBM (BlackBerry Messenger)
- USSD (Unstructured Supplementary Service Data)

… and when the users can also communicate and share information through social networks using numerous **gadgets & widgets** to access their online accounts:

- Facebook login
- Twitter login
- OpenID login
- Custom login controls

Key Distinctions

The most important distinctions between the desktop are that the mobile platform is that the mobile platform is always-networked, always-on, comes in different forms, communicates through different channels, and thrives on applications sharing information amongst each in order other to improve the overall user experience throughout their entire use of the mobile platform.

Another key differentiator is that, while the platform uses http to communicate and the gateway proxy forces the communications to go through the carrier gateway, all page views are cached in the browser and the browser-enabled applications on the phone:

- Geo-location information
- Contacts and phonebook information
- Call and message history
- Calendar and event information
- Application use and overall browsing history

Native Apps vs. Wrapper Apps

Today's mobile apps can be built as either "wrapper" apps, which rely on mobile websites as hosts (an on-line environment), and "on-device native applications", which are stored entirely on the device itself (an off-line environment).

The native app requires little to no transfer of data, users can enjoy high-performance functionality of the application, including the ability to use the advanced features of the device – with or without a network connection. With a native app functions are not delegated to component.

A wrapper, in many ways is the same as a mobile website with the addition of a shell to primarily get past App Store certification without actually

writing a native application. Like a browser and website working together, a wrapper still requires a connection to the web server to transmit the user interface (UI). With a wrapper it is possible the browser will cache data or perform other behaviors unwanted in a banking app.

For example, a Smartphone user may employ an on-device native app to exchange phone numbers and business information with another Smartphone user, and then send their contact data to be stored in the cloud via a cloud storage service and/or through a synchronization tool connected through the user's computer. Additionally, the user's contact data may be transferred to another computer or otherwise shared with a third-party via another app – either intentionally or unsuspectingly.

Ultimately, the data could even be sent to an app server host for future retrievals and modifications through another application residing on another mobile device – bringing in the social network aspect of data sharing and collaboration.

Each of these instances introduces user interactions, network connections, data communications, and data storage. Each, in turn, introduces risk to the sending and receiving users.

Real-World Scenario: Banking Application Security Flaws Uncovered

Mobile applications from some of the top baking institutions in the U.S. were found to be storing customer information, such as usernames and passwords, in plain text, locally in memory on the mobile device.

The mobile user's data could be gleaned if a criminal got physical access to the phone. More troubling, is that the data could also be obtained remotely if an attacker were able to con a user into visiting a malicious website simply by tricking them to click a link sent to them in a fake email or SMS text message. Once at the website, malicious code could be used to compromise the device in order to read the memory, allowing the remote hacker to steal the data. A wrapper may increase exposure to this risk.

> ***"For mobile app providers, there are no shortcuts to protecting customers' data. It must be engineered from the start and thoroughly tested after any change in the app or underlying OS."***
>
> Andrew Hoog
> Chief Investigative Officer, viaForensics
> *Wall Street Journal Article, Nov 5, 2010*

Resiliency and Redundancy

Developing a cross-platform mobile application and service that is resilient to system failure and system compromise can be challenging at best as each operating platform (both device and server side) will possess its own specific configurations and nuances that can't easily be addressed with a one-size-fits-all development process.

Cross-Platform Troubleshooting

Identifying problems across multiple platforms can be difficult (from where does the core problem stem). Getting them resolved without impacting the rest of the system on other platforms can be a nightmare (fixing a problem 'here' exposes another problem on another platform 'there').

Built-in OS Security to the Rescue

Each of the top mobile platform providers have attempted to address the primary security concerns by incorporating security at the heart of the device, its operating system.

Some of the top security mechanisms in place across the board, not specific to any particular mobile OS provider, are:

Passcode Enforcement: Each device will support a protection mechanism controlled by password, passcode, or pass design. They can be enforced locally or over-the-air with built-in expiration and progressive failed attempt protections.

Protocol Authentication: SSL TLS is used to authenticate the client/device in order to prevent unauthorized penetration into the back-end network that hosts the applications service(s) and data.

Device Discovery and Access: Bluetooth discovery mode and desktop connections can be disabled to safeguard device integrity while preventing unauthorized access to the device, its I/O ports, and the data stored on the device. Additionally, the devices can control which applications can access the Global Position Services (GPS) functions and data.

Application Signing: Applications can be signed with a certificate whose private key is held by the OS provider, the application's developer, or within the OS providers' application marketplace. The application certificate is used to identify the author of the application and ensures that the application hasn't been tampered with or altered since it was signed.

Runtime checks can be made against the signature to ensure that an application hasn't become compromised since it was uploaded, installed, or last used, thereby preventing the execution of compromised applications on the device.

Application Sandbox: Applications on the device are placed in a "security sandbox" in order to prevent access to data and processes available via the other applications running on the device. In addition, system files, resources, and the operating system kernel are typically shielded from the user's application execution space.

If an application needs to access data from the system and/or another application, it can only do so by requesting permissions via the use of the built-in operating system APIs. Permissions required by an application can be declared statically in the application, so they can be known up-front at install time and will not change after that, thereby preventing access rights escalations from occurring behind the scenes.

Data Encryption: Some platforms incorporate encryption through APIs that allow developers to further protect the users' data created, used, and stored by an application. Data can be encrypted using methods such as AES, DES, 3DES, or SHA-1.

Event Injection: The BlackBerry platform can control which applications can inject synthetic (non-human) input events such as pressing keys and using the device navigation.

Device Wipe: Data can be erased from the device to prevent unauthorized use. A local device wipe can be set to take place after a specified number of incorrect login attempts have been reached; wiping phone storage and/or SD storage. A remote device wipe can be set by an enterprise mobile device administrator where the wipe will occur immediately (if push is enabled) or the next time the device connects to the enterprise system (if push is disabled).

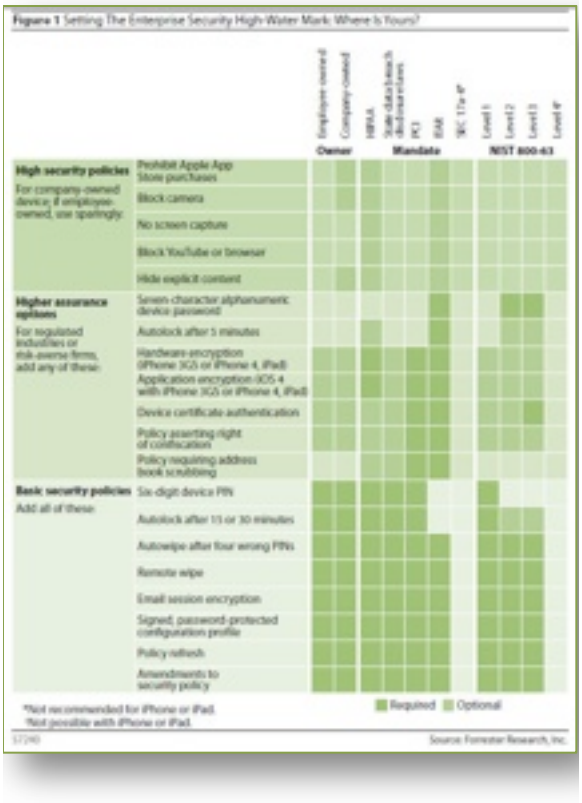
Secure Communications: Support for SSL and TLS for Internet applications, enabling encrypted communication channels between the device and the back-end mobile services. Additionally, secure WiFi connections can be utilized to secure the data transmissions when the device is communicating through wireless networks.

VPN Integration: Most mobile platforms offer native integration with leading VPN protocols such as Cisco IPSec, L2TP, and PPTP.

Segmented Networks: RIM's BlackBerry security architecture supports the separation of corporate networks or LANs (both wired and wireless) into multiple firewall segmented components, thereby containing the network traffic and improving the security and performance of each network segment by filtering out data that is not destined for that particular segment.

Enterprise Manageability: The ability to centrally set and enforce fine-grained access control and usage policies for the devices, their applications, and their network connections will become a critical component for any device that finds its way into the enterprise.

From an enterprise perspective, Forrester Research suggests that RIM's BlackBerry platform will be the primary of choice when it comes to security.



Lowest Common Denominator Security

When selecting a security systems platform, look for an industry leader that supports true native development for mobile applications across all leading platforms and leading browsers. They should be capable of delivering a rich and secure cross-platform experience for both the developer and the applications' users. A platform that eliminates the developer's burden of having to configure and develop for each individual platform at a time, per modification and per upgrade, can provide greater release quality, quicker release times, improved application scalability and reliability, and ultimately proper data integrity.

Fragmented development across multiple platforms coupled with the potential for the end users' environment to encompass a nearly-endless set of network and browser permutations can easily result in "lowest common denominator" security.

Since bringing a product to market quickly is critical for the success of today's business', devising separate solutions to apply to multiple mobile operating systems can force developers to turn a blind eye to secure coding, secure testing, and even secure services.

Furthermore, taking the time to secure the hosting services and to build in resiliency and redundancy are at risk of being stripped bare, ultimately affecting the functionality and reliability of the application and system. In effect, a fragmented approach to security results in little to no security.

Mobile App Security Requirements

There are many things to consider when building mobile applications to sell, use for your own business, and to enable others to benefit from your business services while on the move. Security is one key area to consider, and these are the must-have security requirements.

Secure Design and Coding Practices

When looking for a build-once, run everywhere mobile development platform, be sure to identify and select a firm that embraces and employs core security-oriented design and coding principles that put the platform through a series of checks and balances. Some examples of secure coding best practices would include:

- All developer code, including any re-used core application code, to pass through static code analyzers
- All developer code, including any re-used core application code, to pass through a peer code review process

Once the code is implemented, changes should only be made to the external properties of the code. The code should otherwise be locked down, preventing it from being manipulated by consumers of the platform or any other third party, including hackers or malicious software. To accomplish this, the platform selected and built to should support application signing.

Redundancy and Resiliency

The solution should be highly scalable, the architecture configured to utilize multiple servers behind a redundant load balancer. Multiple instances of the supporting infrastructure components should be employed in order to sustain the system in the event of any application server failure.

Multi-Factor Authentication (MFA)

The Multi-Factor Authentication process involves:

1. The first factor for Authentication utilizes core User Credentials (User ID, password, or PIN).
2. The second factor for Authentication would utilize a device identifier, which should be generated by the application platform provider for each device. These device identifiers will permit lock-outs after loss of the device, upon too many failed logins, or any other defined misuse of the device.
3. The third factor for Authentication would utilize a One Time Password (OTP). The OTP is generated directly by the device using a Server Synch Time and the device identifier.

If the Multi Factor Authentication is successful, the server will send the authentication request to the enterprise's authentication service for further approval, granting access to the features, functions, systems, and data as appropriate.

The solution should support open standards for authorization, such as the OAuth (Open Authorization) standard. Such standards provide users with a site-specific and time-bound token that allows them to share their private content, such as pictures and contact lists, between third-party websites without having to hand out their username/password credentials nor forcing them to grant full access to all of their private resources.

Data Transport Security

Select a solution with a security architecture that is predicated on the 128-bit SSL v3 or TLS v1.0 protocol standard when managing the flow of all data communications between the mobile device and the mobile servers.

Data Storage

Select a solution that has implemented a storage access layer wrapper around the server-side operating system's file system. All access to the data should be made through this layer. This storage layer should reside on a separate system from the location for which the data is actually stored, preventing auto-access to the data if for some reason the system that hosts the wrapper is compromised.

The stored data should be encrypted to prevent unauthorized viewing and manipulation of the data is for some reason the data storage system is compromised.

Data Backups

Select a solution with an architecture that implements critical backups for the deployed libraries and system configuration along with the log data which includes events, alerts, error

messages and reporting information. Backups should be both automated and manual depending upon the type of event.

Secure Transactions

Select a solution that also adheres to the industry's most rigorous benchmarks for secure credit card transactions — including the Payment Card Industry (PCI) Data Security Standard (DSS). In properly meeting PCI-DSS, the security and protection of consumers' sensitive credit and debit card account information in connection with mobile banking and commerce transactions can be better affirmed.

Security Library Integration

Select a platform that is able to integrate with third-party libraries and tools in order to provide additional security capabilities for the device, its applications, and data.

Some key examples of integration points include federated single-sign-on (SSO), VPN integration for enhanced network-based user authentication, and biometrics capabilities to extend second-factor authentication capabilities to the device itself.

Be Prepared To Follow the Future

Just as changes have occurred in the desktop space over the years, there will certainly be changes in the mobile operating and device space in the future. New platforms, multiple versions of the platforms, new ways for people to use the devices, new ways for data to be input, collected, stored, shared, transferred, and deleted.

With this, it is safe to state that the long term viability of the applications being built will be entirely based on the long term viability of the tools and processes employed to build your mobile applications. The platform must address the security risks associated with the systems, devices, and uses of today while also being flexible and resilient enough to change and support the systems, devices, and uses of tomorrow.

Run Everywhere - No Compromises

Kony's Write-Once, Run-Everywhere technology gives today's mobile developers the flexibility and tools they need to service the sophisticated users in need of secure data while computing on-the-go. The platform delivers an environment which allows developers to deliver solid, secure, resilient applications that adhere to the system and data protection requirements both mandated and inferred.

Kony delivers the key components necessary to keep developers at peak performance while building secure applications for the mobile world:

- Multi-factor authentication with SSO (Single Sign-On) support to ensure authorized access to the mobile application and its data
- Secure transactions through redundant and resilient systems
- Integration with third-party security functions to further extend the protections of the device using industry recognized security technologies
- Secure authentication and authorization that is OAuth compliant
- Support for VPN
- Secure storage of confidential data using industry-standard encryption methods
- Secure connections among users, phone carriers, and third-party hosts
- Secure communications via HTTPS-based POSTs and 128-bit encryption over SSL
- Highly-scalable logging system for high-volume reporting
- Device identifier & progress access management, enabling lock-outs due to unauthorized access or application misuse
- Mobile apps security brand assurance, enhancing the developer's own brand
- Brand assurance that helps attract and retain future customers
- Write Once, Read Everywhere provides cost-saving, time-saving and security solutions for rich mobile apps with high functionality

- Streamlined process for upgrading and modifying rich mobile apps across a multitude of platforms; no feeling or need to rush the projects
- Reporting functionality includes traffic/visits reports, server/production analytics, and application-specific analytics

Kony's Security Architecture protects data, data transfers, stored data, securely authenticates devices, and authorizes transactions against spoofing, back door entry and other hacker tactics. By continuously investing in serving rich functionality and cutting-edge computing, Kony can support stored data that is encrypted or in plain text form, both offline and online – depending upon the need of the target use of the application. When the native platform capabilities for offline storage are not available, Kony can implement a storage layer over the existing file system.

While the basic design principle of Kony is to be OFX & PII compliant by not storing any personal data on the user device, the Kony Platform imposes no limitations on the developer, allowing rich implementations for data that can be securely stored locally or hosted elsewhere with the protection measures necessary to ensure proper data security.

Most importantly, it's not about a built it, configure it, and forget. Kony provides the tools and the platform necessary to ensure that your mobile applications will survive the tests of time, twisting and turning with each new platform, device, and use that may come your way.

REFERENCES

1) Article: TechWorld: "Hackers break into Android phone at Black Hat" http://news.techworld.com/security/3233833/hackers-break-into-android-phone-at-black-hat/
2) Article: Wall Street Journal: "Banks Rush to Fix Security Flaws in Wireless Apps" http://online.wsj.com/article/SB10001424052748703805704575594581203248658.html
3) NIST Vulnerability Information: CVE-2009-1185: http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2009-1185
4) Forrester Mobile Security Report: http://www.forrester.com/rb/Research/apples_iphone_and_ipad_secure_enough_for/q/id/57240/t/2

Kony Solutions, Inc.
1825 S. Grant Street, Suite 450
San Mateo, CA 94402
Tel: 1-650-645-2200
Toll free: 1-888-323-9630
Fax: 1-650-645-2201

About Kony

Kony and the Kony Mobile Application Platform™ enable Fortune 500 companies to offer consumers and employees feature-rich mobile applications in less time and at lower costs than any other solution. Leveraging a *Write Once, Run Everywhere* single application definition, applications are designed and developed just once, in a device-independent manner, and deployed across multiple channels, including native applications, device-optimized mobile web, SMS, web gadgets, kiosks, and tablets. Kony's unique platform is proven to future-proof a company's mobile investment by enabling applications to be changed once for all channels, ensuring faster adoption of new operating systems and standards as they are introduced, while eliminating maintenance, upgrade and future development costs.

More information can be found at www.Kony.com