THE 6 LAYERS

AUTHENTICATION AND AUTHORIZATION

APPLICATION CODE SECURITY

DATA AT REST

DATA IN MOTION

DATA IN AN MBaaS OR PRIVATE DATA CENTER

MOBILE SECURITY

APPLICATION DISTRIBUTION AND MANAGEMENT
Now more than ever mobile applications are becoming commonplace in the workplace and our daily lives. People now use their mobile devices for much more than phone calls and emails. They are becoming the primary way users interact with work, the Internet, and friends and family. In fact, there are almost 6 billion mobile devices in the world and Gartner states mobile payment transactions will reach over $500 billion by 2016. While mobility brings great opportunity, it can also introduce significant risk to businesses.

Recently, a large social networking site hit the headlines when millions of users’ passcodes were compromised as a result of improper security standards associated with the handling of data transferred while the users were accessing the site from their mobile devices. According to the Ponemon Institute, a leader in independent research on privacy, data protection and information security policy, 6 out of every 10 cyber-security breaches occur as a result of a mobile device. In the past year security breaches on “smart” mobile devices grew over 150% and similar attacks on Android devices alone grew over 3000%, according to Juniper Networks.

As is true in traditional application development, security cannot be an afterthought. It must be baked into the entire lifecycle of the mobile application, from planning, development and testing, to release and analysis. A security breach not only affects the users/customers that have been compromised, it can have a deep impact to a company’s bottom line and brand reputation, and it will destroy user/customer confidence and loyalty.

In the mobile world, applications often need to be written for many different types of devices (phones, tablets, mini-tablets, etc...) and platforms (iOS, Android, Windows, etc...). All of these permutations introduce different levels of risk and bring with them unique vulnerabilities. For example, one operating system has a way of securing private keys, keeping the data encrypted, while another stores them in accessible memory, thus providing an opportunity for an attack to decrypt sensitive information. Another runs applications in a “secure” sandbox that affords a level of protection by keeping application data and its running processes isolated from all other applications running on the device, while another OS allows applications free range into its memory structure. Regardless of the device, platform, or data source, companies must develop secure mobile applications if they want to minimize the risk to their customers, bottom line, and brand.

Securing mobile applications is complex. Having the right knowledge and tools will make the job easier. When securing a mobile application, there are 6 different layers of security that must be considered.
Authentication and authorization of mobile users

Authentication is the first line of defense to build into any mobile application. It identifies users and allows them access to the application on the device and the connected backend servers. The more rigorous the authentication process, the more secure the application.

Basic authentication consists simply of a username and password. Putting stringent password rules in place, such as forcing a variation of numbers, different case letters and symbols, will help ensure more secure access. According to Unisys, it takes 5 minutes to break a passcode that is 6 characters or less and all lowercase. 9 characters that have a mix of numbers, symbols, and different case letters will take 19,985 years to break! Even with that level of protection, applications are still open to vulnerabilities such as having passwords stolen. In order to make the authentication more robust, organizations should also require passwords to change on a regular basis and forcing re-authentication via a session timeout. This makes it that much harder for people to break in and limits the possible damage if a passcode has been compromised.

2 Factor Authentication is the next level of protection; this is often used when more complex authentication is needed such as accessing a bank account or a company’s financial records. This method requires a username and password, plus a one-time use code that is either emailed, texted, or hardware key generated.
After authentication has passed the next level is authorization. Authorization determines where in the app and what data a user is permitted to access. Not every user gets the exact same access rights to every application. The finance department is allowed access all employees’ salaries where an individual employee is only permitted to see his/her salary. If LDAP is used for authentication then LDAP groups can be used for authorization. But when it comes to mobile apps a more flexible authorization method is generally needed. Many calls are asynchronous and you cannot rely on a continuous connection because of the nature of mobile devices.

OAuth is a recommended approach for authorization. OAuth works by providing the user a token (or “valet key”), this token is then used to grant access based on credentials. The token is time based and can work across different networks. This works well in the “hit and miss” connections of the mobile world. The token can also be used with 3rd party resources (the main purpose of OAuth) such as cloud-based services without having to provide the users actual credentials. Authorization becomes easier and credentials remain secure.

Each application can create it’s own method for authorization, but it is best to use a standard method that is proven and reliable. Regardless of how it is built into the application, authorization is just as important as authentication.
LAYERS 2

Data at rest on the mobile device

Data stored on a private device does not, by itself, make the data secure. The application might be created in its own sandbox, however that still doesn’t fully secure the data. The device might be lost or stolen. According to McAfee, about 5% or 30 million smartphones are lost or stolen every year. Malware can also get on the device and access the data.

Data, especially private data, such as credit cards and passcodes, needs to be encrypted when stored on a device. The encryption key should be at least 128 bits in order cause a significant challenge to potential hackers. There are many different public encryption algorithms that a developer can use, for example Advanced Encryption Standard (AES) or Data Encryption Standard (DES).

Only sensitive data being stored on the device should be encrypted. Encrypting all data could create significant performance issues. Encrypting the data, on its own, is not enough to ensure data protection.

Protecting the private key of the encryption algorithm is necessary to properly secure the data. Should the key be stored on or off the mobile device? If possible it is almost always better to store the private key off the device. Storing the private key off the device makes it more difficult for anyone to hack the data, if the device is lost or stolen, since the private key that is needed to decrypt the data is not accessible locally. If the key was stored on the device, it may only be a matter of time before it is discovered and cracked by a savvy hacker.
A good rule of thumb is that sensitive information, if possible, should never be stored on a mobile device. For example, caching customers' personal information on a device might allow for improved performance in a Customer Relationship Management application (CRM); but it also increases the chance of a security breach. Storing data on a secure server, off of the mobile device, is always preferred no matter the promise of improved performance. If data must be stored on the device, because of the need for offline accessibility or some other reason, properly encrypting the data as a minimum precaution is critical.

In the case of enterprise applications, the need for a comprehensive approach to data security is even more important. The organizations are not just concerned with one user’s sensitive data, but their entire customer base, all other sensitive data stored on the companies network, and the many different industry and government regulations that need to be followed, such as Payment Card Industry (PCI), Health Insurance Portability and Accountability Act (HIPAA), and Sarbanes-Oxley Act (SOX).

In order to manage the risk of lost or stolen devices, most enterprises will turn to a Mobile Device Management (MDM) system. MDMs secure, monitor, and manage mobile devices. They can manage both company-provided and employee-owned devices and most enterprise grade MDMs allow for an additional level of security where the enterprise can remotely wipe a device clean. Consequently, this means that if a device does get lost or stolen, the data or application will not get compromised. The Mobile Center of Excellence (MCoE) in a company is the organization that would be responsible for identifying and implementing the MDM system across the entire company. Mobile Centers of Excellence will be discussed later.
Objective C, Java, JavaScript and HTML5 are development languages used for creating mobile applications. Java, JavaScript, and HTML5 are largely human readable. This allows them to be easily understood and portable to any platform. As a result, this makes them much easier to reverse engineer than traditional pre-compiled applications like Objective C. Reverse engineering, or “reversing” allows malicious hackers to understand application logic and in turn allow a hacker to bypass certain application security code or obtain specific algorithms that may be used to steal personal information.

Obfuscating or obscuring the mobile application code, so that generally readable code is harder to interpret, will not affect the function of the code but will help prevent prying eyes from reading or understanding it. Obfuscating code will not prevent users from taking the code and running it through application will beautify or render that code readable again, but it is one other step to make hackers to go through.

The preferred and more secure method is to encrypt the code. In this approach, the code can be encrypted and placed in libraries that will be read and decrypted at runtime. HTML5 does not have the ability to be encrypted and therefore obfuscation is the only option for HTML5.
Data in motion is any data that is being transmitted to or from a mobile device across a wireless network. If a username or password is transmitted to a server in plain text, then there is almost no reason to have the password since reading plain text off of a wireless network is very easy to do. Therefore, encrypting data being transferred is really the only option. Most secure mobile applications use Security Socket Layer (SSL) as the encryption method. SSL is a point-to-point secure channel; it does not secure the data end to end.

Enterprises can employ other means of securing their data transfer. From Virtual Private Networks (VPNs) to Mobile Application Management Systems (MAM).

**Virtual Private Networks (VPNs)**

VPNs provide security so that traffic sent through the VPN connections stay isolated from other computers on the intermediate networks. VPNS not only encrypt the traffic but they also provide authentication to the internal private network. Most enterprises today already allow VPN access through laptops and home computers in order to enable their employees to work remotely. Mobile VPNs can be implemented and enabled across the entire mobile device and all applications that require the use of a corporate network will tunnel through the VPN. These generally use what is called Internet Protocol Security VPN (IPSec VPN). The other option is to have the mobile applications create the tunnel as needed. This uses Secure Sockets Layer (SSL) VPN. The SSL VPN is becoming a preferred method for mobile devices since the applications specify access and the device itself is not given full access to the corporate network.

**Mobile Application Management (MAM)**

Another weapon in the enterprise arsenal is known as a Mobile Application Management (MAM). With MAM, IT administrators can force all data traffic to be encrypted and/or go through a VPN, whether or not the application was designed for it. This provides a layer of security across all corporate mobile applications. MAMs will be discussed in more detail later.
This is a variant of Data at Rest. Data at rest is generally concerned with data on a mobile device, as mentioned above. However, more and more mobile applications are storing all or part of their data on servers either in a Mobile Backend as a Service (MBaaS) or in private data centers. As long as the servers are secure, this is a safer way of storing protected data.

As stated earlier, SSL protects the data in transit, but once the data gets to the server, it has to be encrypted again for storage. As with data stored on a mobile device, securing the private key is a concern that needs to be addressed. A universal private key, controlled by the data service, can encrypt the data. The security hole with this method is if the one and only private key gets discovered, hackers can gain access to everyone’s data.

A more preferred approach would be to create a unique private key for each user of the application. This will limit the exposure if a single private key gets discovered. If the backend servers are controlled and managed internally, then there should be discussions with the security group and the MCoE to validate the appropriate security measures are taken. If the backend service is cloud or 3rd party based, then the company must understand the security measures that the service provider offers.
Distribution and management plays an important role with enterprise-based applications. Customer facing are either distributed via the web for mobile web apps, or public app stores such as Apple iTunes and Google Play for native apps. Enterprise organizations use MAMs or MDMs to manage the access and distribution of applications. MAMs manage the delivery and administration of enterprise software to end users’ corporate and personal devices. They can allow IT to manage access to applications based on permission levels. While most MAMs are primarily focused on app distribution, increasingly they have begun adding more features for security. Using a MAM, applications can be wrapped in a security blanket where all data gets encrypted (as mentioned above) and administrators can remove the application and data from the device when a user no longer needs access or the device gets lost or stolen.

Similarly, MDMs are becoming commonplace within companies who are trying to handle the rise of “Bring Your Own Device” (BYOD). Before, enterprise organizations could control data access and application usage by issuing pre-configured computers or mobile devices to their employees. Increasingly, employees have decided that they do not want to carry multiple devices and computers with them wherever they go (i.e. the personal and work device are becoming one). Companies have also decided that they no longer want to bear the cost of the devices. As a result, companies are implementing new ways of protecting their internal data, while still providing their employees the flexibility of BYOD. MDMs are one part of the solution to this complex problem. They help ensure secure access to corporate networks and ensure that data and applications on the devices are also secure. In short, an MDM can allow or deny devices from having access to internal systems. They can push security software on devices and they enable the company to scan devices for security breaches in a regular and controlled fashion.

Combining the functionality of an MDM with a MAM gives enterprise organizations an additional level of security to help protect their users, customers, and data from security breaches.
After identifying the basic building blocks of security, the next logical question is, “Which is more secure, a native mobile application or a mobile web application?”

There are many factors to consider when choosing an application architecture and if security is a major concern, native is always preferred.

For the different security layers of data at rest, in transit, or application code security, native applications have many advantages over mobile web.

**Data at Rest**
HTML5 provides unencrypted local storage and session storage. Session Storage is preferred because the data is removed once the browser session has ended. Data in local storage remains even when the browser is closed. Both session storage and local storage can be viewed by the user and is susceptible to Cross-Site-Scripting (XSS) attacks. XSS will be explained in more detail later.

Native applications have the ability to store encrypted data in a secure file system or database. If storing private data on a mobile device is required, then native remains the safe option.

**Application Code Security**
As mentioned previously Objective-C, Java, and HTML5 can be obfuscated and/or encrypted to help prevent people from reading the code. However, tools such as FireBug can easily overcome HTML5 obfuscation. If the application code has proprietary algorithms that needs protection then native applications are also the clear option over HTML5.

**Data in Transit**
Both native and web applications use SSL for transit encryption; however, native applications also allow the ability to encrypt data at the application layer. This means that data, even after the client to server transmission, remains encrypted. HTML5 cannot encrypt data at the application layer.

<table>
<thead>
<tr>
<th>SECURITY CONCERN</th>
<th>NATIVE</th>
<th>MOBILE WEB</th>
</tr>
</thead>
<tbody>
<tr>
<td>Encrypted Data at Rest</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Encrypt Application Code</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Encrypt Data in Transit point-to-point with SSL</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Encrypt Data in Transit end-to-end</td>
<td>Yes</td>
<td>No</td>
</tr>
</tbody>
</table>
Native applications also do not connect as frequently to the network as web applications and in some cases skip the process altogether. Less time communicating over the wireless network allows less time hackers have to hijack the communication stream.

**Additional Mobile Web Security Challenges**

Mobile web applications also provide additional security challenges that do not exist or are significantly muted within native applications.

**Cross-Site Scripting (XXS)** enables attackers to inject client-side script into web pages. This script can then send malicious commands to unsuspecting users. The end user’s browser has no way to detect that malicious code has been added, and will execute the script. Scripts such as these can then access data on the device in local and session storage.

**SQL Injection** is a method used to attack databases through websites. This is done by including SQL statements in a web form entry field in an attempt to get the website to pass the new SQL statement to the backend database. Once the attacker understands how to gain access to the database, they can do whatever they want with it. They can potentially grab credit card or social security numbers or create new accounts that provide them access to the internals of the company.

Other applications can be vulnerable to SQL injection, but it is overwhelmingly considered an issue for mobile web applications.

**XPath Injection** is a very similar technique to SQL injection. Instead of attempting to create malicious SQL statements, attackers try to create an XPath Query for XML. The dangers are just the same as with SQL injections.

While these and other web security challenges can be overcome with appropriate development, they are difficult to completely secure. Even after 15+ years of web development, they are still major causes of security breaches today.

---

**What Are Companies Doing Today?**

Most companies today are taking a fragmented approach to mobile security. Many leave it up to the developers or testers. Security is often a low priority requirement if it’s listed as a requirement at all. Organizations, many times, are sacrificing security for time to market. Sacrificing security will eventually have undesirable consequences. It is not a question of “if”; it is a question of “when”.

Enterprise organizations are just beginning to understand the potential for security breaches of their data and passcodes being stored on their employee’s mobile devices. They realize developers are using 3rd party services for improved functionality and quicker development cycles but may not be adhering to the security standards set by a company.
What is Needed?

To set themselves up for success companies need to begin to take a holistic approach to mobile security. They need to:

Produce proper security requirements along with a proper security review process that spans across the entire application lifecycle. Many companies have industry or government regulations that must be followed, such as PCI, HIPAA, and SOX. These need to be added to the security requirements. This will involve product managers, developers, quality engineers, the Mobile CoE and SecOps. This needs to happen periodically similar to code reviews and there are many 3rd party services who can perform this function.

Create a Mobile CoE. It is becoming clear that enterprises need to have a central group focused on mobility with an expertise in security. Mobile is one of the biggest strategic initiatives in companies today and they cannot afford to have a fragmented approach. Even, if this starts with one person or a best practice development guide, it is a step in the right direction.

The mCoE will focus on:

- Internal and external mobile app development, the security of the mobile applications, and the management of the mobile devices.
- Working with existing security policies for data at rest and data in motion.
- Understanding and enforcing PCI, HIPPA and other standards for mobile devices and applications.

Design security plans for data on the device, in the cloud, and in transit. The Security Operations (SecOps) group, central IT and the Mobile CoE need to work together to ensure that the plans are complete and realistic. There should be an annual audit from either an internal or external source to sign off on the security policies.

Test the applications. Security testing is extremely important. It is essentially the last defense before end-users take control.

---

**MOBILE SECURITY APPLICATION LIFECYCLE**

- **PLANNING**
  - App Security Requirements
  - Device Security Requirements
  - Mobile Enterprise Application Platform

- **DEVELOPMENT**
  - Industry Standard Encryption / Authentication

- **TESTING**
  - Static Code Security Analysis
  - Dynamic Application Security Testing

- **RELEASE**
  - Public App Store
  - Mobile Application Management (MAM)

- **ANALYTICS**
  - Device
  - Application
  - Logs
Security testing involves static and dynamic testing of the code. Static code security testing analyses the code itself for security vulnerabilities. Dynamic testing discovers potential security gaps while running the application.

**Decide how to handle application distribution.** Will the apps go on the app stores or will they be managed through a MAM? Companies will have to choose if they need to implement security precautions and control the devices themselves with a MDM system. This would enable them to enforce VPN access, have the ability to wipe the devices clean, and limit the types of applications users can install on the devices.

Having a fragmented mobile security plan is not sustainable. Fragmented security methodologies lead to fragmented security capabilities, which lead to security breaches, as well as significant audit and compliance issues for any organizations. Understanding the securities needs is the first step, but putting an enforceable plan into place is also required.

---

**What Can Appcelerator Do to Help?**

At Appcelerator, we build Titanium SDK, Titanium Studio and our Cloud Services to deliver cross-platform (iOS, Android and mobile web) applications at a fraction of the cost without sacrificing a high performing, rich mobile experience. A single team with JavaScript experience can build an app for iOS, Android, Blackberry, Windows (coming soon), and HTML5, with up to 90% code reuse. The result is significantly lower cost and faster time to market. No Objective-C or Java skills are required. We do this by delivering a mobile platform that allows developers to create apps built in JavaScript with access to over 5,000 APIs and parity across platforms.

Appcelerator’s development environment also includes a broad set of cloud-based services for push, email, photo sharing and more. No knowledge of IaaS, infrastructure provisioning or server programming languages such as Ruby or PHP is required. To further extend our platform, our Open Marketplace has 310+ modules to improve UI, integrate with 3rd party services, and improve application security.

The Titanium mobile development platform is highly extensible. Developers and partners alike can extend Titanium to meet their needs, through modules extensions, cloud services, and integrations into Titanium Studio. As demands of mobile development quickly grow, especially around security, Titanium can respond with additions to our core sdk, the creation of new modules, and with integrations from a wide variety of partners. Appcelerator has multiple security integration partnerships including:

- MDM companies (MobileIron, Apperian, Good Technologies)
- MAM companies (Nukona/Symantec, Bluebox)
- Mobile authentication partners (RSA, Centrify)
- Security Risk Management and assessment vendors (Appthority, Cenzic)
**Titanium Security**
By generating native, hybrid, and mobile web apps, Titanium allows organizations to determine what level of security risk is acceptable when choosing their application architecture.

Titanium has built in libraries and functions to help with the different layers of security.

**Authentication & Authorization**
Titanium supports multiple authentication methods. It supports 2-Factor authentication with module extensions as well as authorization with OAuth 2.0.

**Data at Rest**
Titanium includes crypto methods. Titanium’s SDK also supports encryption methods for iOS and Android. Additionally, there are Appcelerator and 3rd party encryption modules on the Appcelerator Open Marketplace.

**Application Code Security**
Titanium provides the option to obfuscate code before releasing to production.

**Data in Motion**
Titanium supports SSL and other encryption methods supported by iOS and Android.

**Data in an mBaaS**
Appcelerator Cloud Services (ACS) can protect your data in the cloud. For more details please see the “Appcelerator Cloud Services (ACS) Security” Section.

**Application Distribution**
Titanium has various partner integrations with different MDM and MAM vendors such as MobileIron, Nukona/Symantec , and Apperian to provide enterprise app stores and device security.

<table>
<thead>
<tr>
<th>LAYERS OF MOBILE SECURITY</th>
<th>DOES APPCELERATOR SUPPORT?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Authentication</td>
<td>Yes</td>
</tr>
<tr>
<td>Data at Rest</td>
<td>Yes (through Appcelerator modules &amp; partner modules)</td>
</tr>
<tr>
<td>Application Code Security</td>
<td>Yes</td>
</tr>
<tr>
<td>Data in Motion</td>
<td>Yes</td>
</tr>
<tr>
<td>Data in mBaaS</td>
<td>Yes</td>
</tr>
<tr>
<td>Application Distribution</td>
<td>Yes (through partners)</td>
</tr>
</tbody>
</table>
Titanium also lends itself well to integrating into a proper framework that analyzes code for security issues. Implementing a standardized framework will help security engineers and developers alike work together and focus adequate resources on security concerns and issues.

**Titanium Modules and Security**
Companies, partners, and developers can use existing modules or create new modules that extend the capabilities of the Titanium platform. These modules can be used and stored internally or can be published on the Open Appcelerator Marketplace. There are hundreds of modules already available for consumption today, including many focused on security, such as data at rest with SQL Encryption and AES Security.

**Appcelerator Cloud Services (ACS) and Security**
Appcelerator’s Cloud Services provide the ability to remove sensitive data from mobile devices themselves and store them in secure file and database servers. ACS is available in 3 flavors: public cloud, for more security, virtual private cloud, and for the most secure, private cloud. With the Virtual Private Cloud, the services are deployed in a public cloud, but use dedicated infrastructure, wrapped in a VPN for private access to a particular company. Our Private Cloud, the services are deployed in a private data center and can take advantage of internal security practices already in use within an organization.

**Security Risk Management and Assessment**
Appcelerator Professional Services can provide personalized security assessments, recommendations, and application development services. Appcelerator’s partners (e.g. Cenzic) also provide expert mobile application security assessments.

*For more information, please visit [www.appcelerator.com](http://www.appcelerator.com)*