Seven Tips to Keeping Financial Apps Safe and Secure

jack henry & ASSOCIATES INC. | jack henry Banking | Symitar | ProfitStars

Seven Tips to Keeping Financial Apps Safe and Secure

Mobile applications provide users with access to critical information while on the go. In order to quickly empower users with immediate access to their financial data, there's an interesting dance that needs to take place between convenience and security. When considering a mobile solution for your financial institution, ensure these seven best practices are in place so your customers remain safe and secure.

1. Sensitive Data

Many financial applications use a person's bank/credit union account or card number as a way to establish identity. This data is sent over the Internet each time a new transaction and balance is received. There is simply no reason for sending this data over the wire, or worse, storing it on the device! Applications should always use a different key for identifying a user's account. One way to do this is to simply use the product name such as "Preferred Checking," rather than passing a user's account number back and forth over the Internet.

2. Device Signature

A native app is very intelligent about a phone's identity. It knows the phone's carrier, model number, mobile equipment identifier (MEID), international mobile subscriber identity (IMSI), and phone number. Unlike a cookie and IP address that can be spoofed with software, a mobile device stores most of this information on a SIM card, which is a read-only device. When a mobile banking application is installed and activated for the first time, additional steps should be taken in order to identify the authenticity of the user by leveraging out-of-wallet questions. For obvious reasons, a text message, email, or phone call should not be used to pass a token to the user for this initial setup. All Web services that allow a user to consume data should validate that the requesting device is from a known device list, regardless of credentials.

3. Passcode Access

Many app users will turn off the passcode (or PIN) access to an application if they have enabled the security password for their device. It is important to have your application check if the user ever disables the security settings on the device. If they do, the user should automatically be prompted to re-enable the device security or enable the application security features. It is also important to re-validate a user when they are performing tasks such as funds transfers, RDC, bill pay, or peer-to-peer payments after the action has been submitted. Doing this will not slow down the user experience but will act as a confirmation for the action taking place. If the application detects any malicious activity from the Web services side, it should push an additional question to the user before the action can be completed.

4. Storing Text Data

Native applications can quickly show account balance and previous transactions. This data is refreshed when a user opens the application, and the previous stored data is usually stored in a split database. There are many developer tools that allow the raw access to the underlying database, which can be used to query data even if a password exists on the device. This can be used to access information such as account number and the last four transactions. Having this data paired with a user's name and phone number, which is associated with the device in the contacts, would allow a hacker to call the financial institution and request a password reset or to transfer funds. To prevent this, we suggest using the advance encryption standard, or AES128, to encrypt the most recent transactions that could be used to validate an external user. The seed, or shared secret, can be unique per device and retrieved from the server when the application starts. The key is to never store data locally on the device, which would make it usable for a thief.

5. Data Service Access

All data should be requested over a secure socket layer (SSL) for encrypted communication. The data SSL Certificate should be a 256-Bit Encryption strength. The native application client should utilize OAuth or the emerging OAuth 2.0 specification. OAuth allows for the application to connect to the data services without having to store username and passwords on the device or send the credentials over the wire with each request. OAuth 2.0 is widely used by social media sites such as Twitter, Facebook, and MeetUp but has not yet garnered mass adoption in financial services.

6. Images

Images of checks contain all of a user's confidential information such as routing number, account number, and billing address. Encrypting large images on a device is slow compared to text data. It is our recommendation that images of checks that are stored for remote deposit capture should be sent to the server immediately after the photo is taken. If the application allows for review of the image, it should state that an image is available and as a user selects it, download the image for presentation. The check image should never be stored or cached on the device for later retrieval.

7. Remote Wipe

With enough time, systems can be compromised through developer tools, stack overflows, or brute force attempts. However, all of these efforts do take some time. If a user loses their phone, it takes six minutes on average until they acknowledge that it is lost or stolen and not just misplaced. A defined workflow should exist which allows the consumer to contact the bank or credit union about the lost phone. The financial institution should be able to flag the device as compromised, and the next time the application is launched, it should remove all sensitive data and brick the application. The app would then need to be reinstalled and reconfigured for it to connect to the Web services and properly function.

We understand that our business depends on security, and we know that this is a vital area that needs to be understood. Our goal with this white paper has been to help you understand the measures we take to secure our apps. If the above areas are still unclear, we encourage you to contact the Jack Henry & Associates, Inc.® iPay Solutions™ team with any questions.