Mobile App Security – 5 worst things

Most people aren't thinking about security and data privacy when buying a scone at Starbucks with their phones, playing Angry Birds while commuting, or using whatever clever app is the cornerstone of your digital business strategy. If they give security any thought, they figure the developers took care of all that for them. The app is from a reputable company, and they got it from the app store -- or even directly from their employer. What could go wrong? A lot.

The global mobile infrastructure is a complex, interconnected, and desirable target. Companies must tackle potential security problems when formulating a B2B or B2C mobile strategy. While security particulars vary widely, depending on the type of app being deployed, it's up to IT leaders to ensure that user convenience never trumps protection of valuable enterprise or consumer information.

1. Insecure data storage

The Starbucks mobile app is one of the most widely used mobile payment apps in the US. Consumers simply enter their passwords once when activating the payment portion of the app and use it again and again to make unlimited purchases without having to re-input their password or user name.

While that might be convenient for a caffeine-starved public, Starbucks recently confirmed that its app was storing usernames, email addresses, and passwords in clear text. That allowed anyone with access to the phone to see passwords and usernames just by connecting the phone to a PC. Clear text also displayed users' geo-location tracking points. With this information in hand, unauthorized individuals would have the credentials to log in to the Starbucks website as well. It's common for users to employ the same username and password across systems, so if someone compromises that particular password, the potential also exists for them to compromise additional user accounts.

Design apps in such a way that critical information such as passwords and credit card numbers do not reside directly on a device. If they do, they must be stored securely. For iOS, passwords should be stored within an encrypted data section in the iOS keychain. For Android, they should reside within encrypted storage in the internal app data directory, and the app should be marked to disallow backup.

2. Weak server-side controls

When creating their first mobile applications, businesses often expose systems that had not previously been accessible from outside of their

networks. Often, these formerly sheltered systems are not fully vetted against security flaws. A number of back-end APIs assume (quite wrongly) that an app will be only thing that will access it. However, the servers that an app is accessing (whether they're your own or the servers of any third-party system your app may be accessing) should have security measures in place to prevent unauthorized users from accessing data. It's critical that back-end services be hardened against malicious attackers. This means all APIs should be verified and proper security methods be employed to ensure only authorized personnel have access.

3. Unintended data leakage

Brands covet the kind of personal information some mobile apps glean. Being able to personalize marketing offers to consumers is a key digital business goal. But it's essential that this desire to gather personal data doesn't compromise a consumer's privacy.

For instance, media reports recently contended that the NSA had tapped popular smart phone apps like Angry Birds to gather the huge amounts of personal data -- including age, location, gender, and more -- that they collect. This is what's meant by a "leaky" app.

It's not just consumer apps that are at risk. Consider a healthcare app this is used to track how often a patient experiences a particular symptom of a disease. If the app also contained analytics that reported how often that same section of the application was viewed, it would be possible for someone with analytics access to determine the medical condition of a specific user -- and place the provider in violation of HIPAA compliance.

Use caution when choosing analytics providers and implementing advertising. Watching what, how, when, and where data moves can give an attacker a gold mine of information. Do this tracking before the bad guys and obfuscate where necessary.

[You can keep only three security products. Which ones stay? Tell us in our 2014 Strategic Security Survey and enter to win a 64 GB Apple iPad Air with WiFi + cellular or one of three 60-minute one-on-one consultations with the survey author, Michael A. Davis.]

4. Broken cryptography

Many widely used cryptographic algorithms and protocols, like MD5 and SHA1, have proven to be insufficient for modern security requirements. But there's no easier way to mishandle mobile encryption than for an organization to create and use its own encryption algorithms or protocols.

Always use modern algorithms that are accepted as strong by the security community, and whenever possible use state-of-the art encryption APIs within

mobile platforms -- think AES with a 256-bit key for encryption and SHA-256 for hashing. If you're not sure about your cryptography, invest in manual analysis, such as penetration testing, threat modeling, and interactive tools that allow the tester to record and modify an active session.

Poor key management is another issue that warrants close consideration. Many organizations make the mistake of using strong encryption algorithms, but implement their own keys and certificates in areas that are vulnerable to attackers. An example is when an app ships with keys stored in the byte code. Since the keys are common across all app installs, the security is negated because anyone who gains access to someone's encrypted data can decrypt it.

5. Security decisions via untrusted inputs

A mobile app can accept data from all kinds of sources. In the absence of sufficient encryption, attackers could modify inputs such as cookies and environment variables. When security decisions on authentication and authorization are made based on the values of these inputs, attackers can bypass your security.

For example, in 2012 a flaw in Skype security allowed hackers to open the Skype app and dial arbitrary phone numbers using a simple link in the contents of an email. Similarly, a bug in the iPhone 1 OS enabled hackers to listen in on phone conversations when those phones were connected to insecure wireless networks. Any app that has openings to accept data from external sources must include checks to all inputs used to build the app.

If all this sounds complicated, that's because it is. Before embarking on a DIY mobile strategy, make sure your mobile developers can think through unintended consequences of app design. If they can't, get help. Delivering an easy-to-use app won't win you any points if you put customer or enterprise data at risk.