Datensicherheit in iOS-Geräten aus Unternehmenssicht

Ein AdNovum IT Consulting Whitepaper

Datensicherheit in iOS-Geräten aus Unternehmenssicht

Immer mehr Unternehmen bieten ihre Dienstleistungen über den mobilen Kanal an. Dabei vertrauen sie vielfach implizit den Sicherheitsfunktionen der mobilen Geräte und setzen sich somit deren Sicherheitsschwachstellen aus. Bei der App-Entwicklung ist deshalb die Spezifikation, welche beschreibt ob und wenn ja, wie und wo schützenswerte Daten auf dem mobilen Gerät gespeichert werden, essenziell. Bei bereits bestehenden Apps muss gewährleistet sein, dass die schützenswerten Daten vorschriftsgemäss behandelt werden.

Einerseits profitieren die Mitarbeiter selbst von den Dienstleistungen, wenn sie beispielsweise mit ihren mobilen Geräten vor Ort beim Kunden Präsentationen zeigen, Produkte vorstellen, Offerten oder Verträge ad hoc erstellen oder Bestellungen aufnehmen. Hierfür werden nicht nur Stammdaten benötigt, sondern auch weitere schützenswerte Geschäftsdaten, welche unternehmenskritisch sind. Um Effizienz zu garantieren, müssen diese Daten auch offline verfügbar sein, das heisst, sie müssen zumindest auf dem mobilen Gerät zwischengespeichert werden.

Andererseits werden die Dienstleistungen via den mobilen Kanal auch den Kunden angeboten. Hier gelten selbstverständlich dieselben Regeln wie bei den Dienstleistungen für die Mitarbeiter und es müssen vor allem persönliche Daten geschützt werden. Erschwerend in diesem Fall ist, dass die mobilen Geräte natürlich nicht der Kontrolle der Unternehmung unterliegen.

Unabhängig vom Kanal oder dem Endbenutzer gilt es, die firmeninterne IT-Sicherheitsstrategie zu beachten. Sie stützt sich auf Gesetze und Vorschriften ab und definiert Massnahmen zum Umgang mit schützenswerten Daten. Die Missachtung der IT-Sicherheitsstrategie kann zu rechtlichen und existenziellen Bedrohungen führen, wie zum Beispiel wenn persönliche Daten nicht genügend geschützt verarbeitet oder gespeichert werden.

Firmenintern sind zurzeit die Geräte der iOS-Familie populär, das heisst die Produkte iPhone oder iPad von Apple, welche als Betriebssystem das iOS verwenden. Dies, unter anderem, weil sie eine umfassendere Geräteverwaltung und mehr Sicherheit versprechen als z.B. Android-basierte Geräte.

Es gilt, die Daten auf dem mobilen Gerät zu schützen, sodass diese von Dritten weder unerlaubt gelesen noch verändert werden können.

Bedrohungssituation

Die Bedrohungen, welche direkt auf das Endgerät einwirken, sind äusserst vielfältig. Die zurzeit bekannteste und grösste Bedrohung ist, dass dem Benutzer sein mobiles Gerät abhandenkommt, sei es durch Verlust oder Diebstahl. Damit hat ein Datendieb sämtliche Möglichkeiten offen, die Sicherheitsmechanismen zu umgehen, um an die gewünschten Daten zu gelangen. Er kann beispielsweise beliebig viel Zeit aufwenden, beliebige Ressourcen einsetzen und alle gewünschten Technologien verwenden. Im Zusammenspiel mit Social-Engineering kann der Angreifer auch gewisse Abkürzungen nehmen, indem er zum Beispiel zuvor den Passcode ausspioniert hat und diesen somit nicht mehr erraten muss.

Wie in der Presse zu lesen war und ist, gelangt immer mehr Malware für mobile Geräte in Umlauf, welche versucht, an persönliche Daten zu gelangen, damit sich ein Angreifer beispielsweise als der bestohlene Benutzer ausgeben kann. Durch Schwachstellen in der App-Sandbox oder auf iOS-Geräten mit unterzogenem Jailbreak ist es auch möglich, dass solche Apps oder andere Malware Daten aus Dritt-Apps lesen können.

Es existieren auch verschiedene Szenarien, welche nicht direkt das iOS-Gerät attackieren, sondern externe Datenablagen, wie iTunes-Backups oder Cloud-Services. Malware auf PCs hat die Möglichkeit, unverschlüsselte iTunes-Backups zu lesen. Ist ein iTunes-Backup verschlüsselt, kann mit einem Tastatur-Logger das iTunes-Backup-Passwort ausspioniert werden und damit das Backup entschlüsselt werden.

Abbildung 1: Auswahl von Bedrohungssituationen

Alle oben dargestellten Bedrohungen lassen sich jeweils auf eine oder mehrere der bekannten Grundbedrohungen einschränken: Verlust der Verfügbarkeit, Integrität, Vertraulichkeit, Authentizität sowie Nachweisbarkeit (Non-Repudiation).

Die Bedrohungen agieren auf verschiedenen Ebenen und unterscheiden sich auch im zeitlichen Aspekt, wie beispielsweise vor, während oder nach der Laufzeit einer App. Dies muss bei Sicherheitsvorkehrungen berücksichtigt werden.

Wir fokussieren uns im Folgenden auf die iOS-spezifischen Sicherheitsmerkmale und dessen Möglichkeiten mit Schwerpunkt Datensicherheit auf dem Endgerät. Für eine ganzheitliche Sicherheitsbetrachtung ist jedoch nicht nur das mobile Gerät zu berücksichtigen, sondern wie in der oben dargestellten Grafik ersichtlich, die gesamte Infrastruktur, welche tangiert wird, um vertrauliche Informationen zwischen einem Server und einem Endgerät auszutauschen.

Sicherheitsmassnahmen in iOS

Apple stellt ein Sicherheits-Framework zur Verfügung und deckt damit die Ebenen der Gerätesicherheit, Software-Sicherheit und Datensicherheit ab. Die Einstellungen der Gerätesicherheit und Software-Sicherheit haben einen erheblichen Einfluss auf die Datensicherheit.

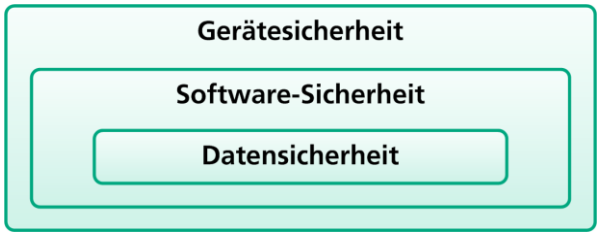

Abbildung 2: Ebenen der Sicherheitsmassnahmen auf mobilen Geräten

Gerätesicherheit

Die Gerätesicherheit beinhaltet die grundlegenden Sicherheitseinstellungen des iOS-Geräts wie Passcode-Komplexität, zwingende Verschlüsselung des Backups, Nutzung von Cloud-Services und so weiter. Bei privaten Geräten muss der Endbenutzer diese Einstellungen selber wählen. Dabei wird vielfach der Bequemlichkeit gegenüber der Sicherheit der Vortritt gelassen.

Bei Geräten mit Zugriff auf firmeninterne Daten wie E-Mail, Kalender oder spezifische App-Daten ist es ein Muss, eine Mobile-Device-Management-Plattform (MDM-Plattform) einzurichten. Sie bietet die Möglichkeit, Geräte zentral zu verwalten sowie die Konfigurationsprofile zu erstellen. Die Konfigurationsprofile erzwingen gewünschte Basis-Sicherheitseinstellungen des mobilen Geräts wie beispielsweise Passcode-Komplexität oder Exchange-ActiveSync-Einstellungen. Neben Apple stellen auch einige andere Hersteller Tools für die MDM-Unterstützung her.

Software-Sicherheit

Die Software-Sicherheit beinhaltet die Sicherheitsmechanismen, die für den Schutz der Integrität und die Garantie der Authentizität von Software implementiert sind, und zwar vor und während der Laufzeit der App.

Vor der Laufzeit einer App wird geprüft, ob die Integrität und Authentizität der App gegeben ist. Dieser Mechanismus wird „Mandatory Code Signing“ genannt. Eine App muss entweder von Apple selbst signiert oder im Falle einer Enterprise-App von der Unternehmung signiert sein, welche im Enterprise-Provisioning-Profil definiert worden ist. Die Signaturpflicht gilt auch für Apps während der Test- und Entwicklungsphase.

Mit „Code Signing Enforcement“ (CSE) wird anschliessend während der Laufzeit der App geprüft, dass der Code, welcher ausgeführt wird, immer signiert ist. Dadurch kann kein neuer unsignierter Code eingeschleust werden. Das bedeutet aber auch, dass native Just–in-Time-Kompilierungen nicht möglich sind. Für den Mobile-Safari-Browser kommt aus Performance-Gründen jedoch eine Ausnahme zur Anwendung. Er kann JavaScript-Code zur Laufzeit kompilieren und ist damit einiges effizienter, als wenn er den JavaScript-Code immer interpretieren müsste.

Wie bei den meisten modernen Betriebssystemen wird auch bei iOS ab Version 4.3 „Address Space Layout Randomization“ (ASLR) eingesetzt. Dieser Mechanismus erlaubt es, die Speicheradressen der zu ladenden Software zu randomisieren. Dies erschwert es dem Angreifer, Speicheradressen einfach zu erraten und Schwachstellen auszunutzen.

Aufgrund der Rückwärtskompatibilität gibt es in iOS die beiden Arten “limited“ und “full“ ASLR.

Jailbreaking und Daten-Sicherheit

Beim Jailbreaking geht es darum, die Nutzungseinschränkungen von Apple zu umgehen. Dies wird erreicht, indem man Schwachstellen im iOS-Gerät und im iOS-Betriebssystem ausnutzt, um modifizierte Versionen von iOS zu laden. Bei einem „untethered Jailbreak“ kann das Gerät jeweils von sich aus in den manipulierten Jailbreak-Zustand starten, während das iOS-Gerät bei einem „tethered Jailbreak“ immer mit Hilfe eines Computers und spezieller Software gestartet werden muss, um in den Jailbreak-Zustand zu gelangen.

Für die Endbenutzer mag Jailbreaking einige Vorteile bieten: Sie können Apps installieren, auch wenn sie nicht von Apple signiert wurden, die Standard-Apps verändern, sodass sie neue und benutzerfreundlichere Funktionen enthalten, oder ganz generell das iOS-Gerät komplett anders nutzen als vorgesehen (zum Beispiel als Unix-basierten Rechner).

Aus Unternehmenssicht birgt Jailbreaking des iOS-Geräts gravierende Risiken. Beim Jailbreaking werden fast alle Software-Sicherheitsmechanismen umgangen:

- Die Sicherheit der App-Sandbox ist nicht mehr gegeben und es können Daten aus anderen Apps ausgelesen werden.
- Daten sind potenziell über weitere Kanäle lesbar, wie z.B. ftp oder ssh.
- Analog den Trojanern auf dem PC lassen sich bösartige Apps installieren, welche Daten mitlesen, verändern und senden können.

Bekannte Malware existiert zur Zeit nur für mittels Jailbreak manipulierte iOS-Geräte, da die Malware mehr Rechte oder Funktionen benötigt, als von Apple im Standard-iOS zur Verfügung gestellt wird. Für Unternehmen, welche iOS-Geräte im Einsatz haben, ist es somit zentral für die Datensicherheit, dass die Geräte keinem Jailbreak unterzogen wurden.

Dies ist die einzige Software-Sicherheitseinstellung, welche zur Kompilierungszeit vom Entwickler gesetzt werden kann. Alle anderen Software-Sicherheitseinstellungen werden von Apple vorgenommen.

Im Gegensatz zu Android laufen in iOS alle Apps unter demselben Unix-Benutzer. Das bedeutet, dass iOS ein containerbasiertes App-Sandbox-Konzept nutzt, um Apps und ihre Daten gegenüber anderen Apps zu schützen. Mit sogenannten Sandbox-Profilen wird definiert, welche Rechte eine App hat. Bis auf gewisse im Gerät eingebaute Standard-Apps haben alle Apps dasselbe Sandbox-Profil, welches Operationen erlaubt wie Zugriff auf Adressbuch, Fotos, Musik, Netzwerk, wohldefinierte Betriebssystemfunktionen und Filesystemzugriff im App-spezifischen Bereich.

Datensicherheit

Ergänzend zu den von Apple vordefinierten Software-Sicherheits-Mechanismen stehen die Mechanismen zur Datensicherheit. Diese müssen nun explizit von den Software-Entwicklern korrekt umgesetzt werden.

Heute werden die Anforderungen bezüglich mobiler Datenhaltung nur selten bewusst in die Anforderungsspezifikation einer App aufgenommen, da die Möglichkeiten vielfach nicht bekannt sind oder fälschlicherweise davon ausgegangen wird, dass die Daten per se sicher auf dem mobilen Gerät gespeichert werden. Man kann jedoch zeigen, dass ohne spezielle Massnahmen alle App-Daten eines mittels Jailbreak manipulierten iOS-Geräts innert weniger Minuten lesbar sind.

Bei iOS-Geräten wird unterschieden zwischen File-basierten Daten und Daten, welche in der sogenannten Keychain abgelegt werden. Die Keychain wird benötigt, um kurze, sensitive Daten wie Passwörter für Wireless-LANs oder SSL/TLS-Zertifikate zu speichern.

Für filebasierte Daten stellt Apple verschiedene Schutzklassen zur Verfügung (siehe Kasten). Ohne Angabe einer Schutzklasse werden die Daten nicht abhängig vom Passcode gespeichert und sind damit immer lesbar, sobald das Gerät eingeschaltet ist.

Analoges gilt für die sogenannten Keychain-Elemente. Wichtigster Unterschied ist jedoch, dass in diesem Fall zusätzlich definiert werden kann, ob die Elemente auf andere iOS-Geräte übertragen werden dürfen. Dürfen sie nicht übertragen werden, so werden diese Elemente auch nicht in ein Backup abgelegt und sind somit nicht auf einem PC angreifbar. Bei Keychain-Elementen wird, sofern vom Entwickler nicht anders gesetzt, kSecAttrAccessibleAlways verwendet, was bedeutet, dass das Element immer lesbar ist und auch in ein Backup abgelegt wird. Der Default ist also bei file- und keychain-basierten Daten die unsichere Variante.

Schutzklassen für filebasierte Daten

NSFileProtectionNone

Das File ist immer lesbar, sobald das iOS-Gerät eingeschaltet ist (Standard-Schutzklasse, wenn nichts angegeben wird).

NSFileProtectionComplete

Das File ist verschlüsselt gespeichert und man kann nur darauf zugreifen wenn das iOS-Gerät entsperrt ist.

NSFileProtectionCompleteUnlessOpen

Das File ist verschlüsselt gespeichert. Wurde es im entsperrten Zustand des iOS-Geräts geöffnet, so ist es danach immer lesbar, auch im gesperrten Zustand des iOS-Geräts, bis das File wieder geschlossen wird.

NSFileProtectionCompleteUntilFirstUserAuthentication

Das File ist verschlüsselt gespeichert. Nach dem ersten Öffnen des Files kann es immer gelesen werden, auch wenn das File wieder geschlossen und das iOS-Gerät gesperrt wurde.

Schutzmöglichkeiten der Keychain-Elemente

kSecAttrAccessibleAlways

Das Keychain-Element ist immer lesbar, sobald das iOS-Gerät eingeschaltet ist.

kSecAttrAccessibeAfterFirstUnlock

Das Keychain-Element ist verschlüsselt gespeichert und man kann nach dem ersten Entsperren des iOS-Geräts darauf zugreifen.

kSecAttrAccessibleWhenUnlocked

Das Keychain-Element ist verschlüsselt gespeichert und man kann nur im entsperrten Zustand des iOS-Geräts darauf zugreifen.

Bei allen Attributen kann zusätzlich spezifiziert werden, dass das Keychain-Element nicht auf ein anderes Gerät übertragbar ist und damit auch nicht in einem Backup abgelegt wird.

Die passcode-abhängig geschützten Daten auf dem iOS-Gerät sind folglich so gut geschützt, wie der Passcode komplex ist, das heisst, der Passcode nicht erraten oder mittels Brute-Force-Methoden geknackt werden kann. Vielfach wird der Passcode aufgrund der Benutzerfreundlichkeit einfach gehalten.

Bei den heutigen iOS-Geräten, die mittels Jailbreak manipuliert werden können, hält ein vierstelliger, nummerischer Passcode einem Brute-Force-Angriff nur 20 Minuten stand. Anschliessend können sämtliche Daten aus dem iOS-Gerät gelesen und kopiert werden. Um komplexere Passcodes zu erraten, können auf einem iOS-Gerät, das einem Jailbreak unterzogen werden kann, beliebige Tools installiert werden, um damit Dictionary-Attacken durchzuführen.

Bei den Standard-Applikationen ist wichtig zu wissen, dass zurzeit nur E-Mails passcode-abhängig gespeichert werden. Alle anderen filebasierten Daten wie Kontakte, Notizen, SMS etc., können ausgelesen werden, ohne den Passcode eingegeben zu haben.

Reichen die oben genannten Schutzklassen für Ihre Daten nicht aus, müssen sie applikationsspezifisch geschützt werden. Dies bedeutet in den meisten Fällen ein applikationsspezifischer Passcode, aus welchem ein kryptografischer Schlüssel generiert wird. Da dieser Passcode ebenfalls genügend komplex gewählt werden muss, geht es auch hier um den Zielkonflikt Sicherheit versus Benutzerfreundlichkeit.

Weitere Möglichkeiten, Daten auszulesen

Das reine Schützen der in der App gespeicherten Daten reicht nicht aus, um die Vertraulichkeit zu gewährleisten. Üblicherweise werden schützenswerte Daten aufgrund verschiedener iOS-Funktionen noch an einigen anderen Orten gespeichert, wie zum Beispiel:

- Tastatur Cache
- Screenshot, welcher entsteht, wenn die App in den Hintergrund tritt
- Clipboard bei Copy- und Paste-Funktionen

Je nach Einsatzbereich der App müssen auch andere persönliche schützenswerte Daten der Benutzer in Betracht gezogen werden, z.B. GPS-Koordinaten, Fotoablage, Browser-Cache, Browser-Suchanfragen.

Es existieren zudem weitere Angriffsszenarien, welche betrachtet werden müssen. Darunter fällt, wie zu Beginn erwähnt, der Angriff auf den Backup-Mechanismus. Falls ein Angreifer Zugriff auf ein Backup des iOS-Geräts hat, kann er unter Umständen Daten dieses Backups wiederherstellen. Wenn das Backup nicht Passwort-geschützt ist, können alle Daten mit Ausnahme der Keychain wiederhergestellt werden, d.h. auch die App-spezifischen Daten, jedoch ohne den temporären und den Cache-Ordner.

Wenn ein iOS-Backup mit einem Passwort geschützt ist, so wird das Backup-Passwort zwingend benötigt. Ein Angreifer muss das Passwort mit bekannten Methoden erraten oder mit einer Brut-Force-Methode in Erfahrung bringen. Eine weitere Möglichkeit ist, das Backup-Passwort aus der Keychain im iOS-Gerät auszulesen, sofern der Angreifer den Passcode des Geräts kennt und darauf Zugriff hat. Falls der Angreifer in den Besitz des Backup-Passwortes kommt, kann er auf alle Daten inklusive Keychain zugreifen. Die einzige Ausnahme sind in diesem Fall die Keychain-Elemente, die an das Gerät gebunden sind.

Fazit

Die iOS-Sicherheitsfunktionalitäten sind gut durchdacht und im Vergleich zu anderen modernen Betriebssystemen effektiv – sofern die beschriebenen Massnahmen korrekt umgesetzt sind. Dazu müssen unter anderem die folgenden Bedingungen erfüllt sein:

- Die zur Verfügung stehenden Datenschutz-Mechanismen müssen korrekt eingesetzt sein.
- Interna des iOS müssen bekannt sein, um Data-Leakage über Seitenkanäle verhindern zu können.
- Das iOS-Gerät darf keinem Jailbreak unterzogen worden sein.
- Der Passcode muss genügend komplex gewählt sein.

Die ersten beiden Punkte sind nicht trivial und benötigen fundiertes Fachwissen, um die Daten zu schützen.

Bei jeder Implementation einer App, welche schützenswerte Daten bearbeitet, muss analysiert werden, ob und, wenn ja, wie diese Daten auf dem mobilen Gerät gespeichert werden sollen. Es ist essentiell, den Schutzbedarf sämtlicher App-Daten zu spezifizieren. Dabei muss auch geklärt werden, wann die Daten lesbar sein müssen (müssen sie z.B. lesbar bleiben, wenn die App im Hintergrund ist).

Wie einführend erwähnt, ist die Datensicherheit auf dem mobilen Gerät nur ein Aspekt, welcher bei einer Implementation einer App berücksichtigt werden muss. Benutzerauthentisierung, sichere Kommunikation, Integration in bestehende Infrastrukturen sowie Wartbarkeit der App bei Multiplattform-Unterstützung sind weitere zentrale Punkte, die es zu beachten gilt.

Über AdNovum IT Consulting

AdNovum konzipiert, implementiert und pflegt seit bald 25 Jahren anspruchsvolle Software-Lösungen für Firmen und Behörden. Das Wissen und die Erfahrung aus der Projektarbeit geben wir in Form von Beratung an unsere Kunden weiter. Sie finden bei uns hersteller- und produktunabhängige Unterstützung für komplexe IT-Vorhaben. Unser Angebot umfasst alle Lösungsebenen, von technologischen Fragen über Prozessgestaltung bis hin zur IT-Strategie-Beratung.

Mobile Computing ist einer der Kernbereiche der IT-Consulting-Dienstleistung von AdNovum.

http://www.adnovum.ch

Über den Autor

Aldo Rodenhäuser, seit 2000 bei AdNovum, ist dipl. El.-Ing. FH mit NDK ETH in Information Security. Als Projektleiter befasste er sich mehrere Jahre mit dem Engineering und der Integration von Identity- und Access-Management-Systemen sowie Authentisierungslösungen, bevor er sich mobilen Applikationen zuwandte. Für diese genauso wie für den klassischen Bereich erstellt er als IT Consultant Sicherheitskonzepte und Risikoanalysen.

Kontakt

AdNovum Informatik AG
Aldo Rodenhäuser, Senior IT Consultant
Röntgenstrasse 22, CH-8005 Zürich
Tel. +41 44 272 6111
E-Mail: info@adnovum.ch, http://www.adnovum.ch