Bring Your Own Device

Vergleich: Work Profile (Android) gegen User Enrollment (iOS)

Google hat mit den Work-Profilen in Android Enterprise bereits seit längerem Maßstäbe für den Einsatz von Endgeräten im BYOD-Umfeld gesetzt. Mit iOS 13 versucht Apple nun zu seinem Wettbewerber aufzuschließen.

› iOS und Android verfolgen unterschiedliche Ansätze, um Berufliches und Privates auf den mobilen Endgeräten zu trennen. (Foto: GaudiLab, Shutterstock.com)

Auf iOS-Geräten konnte man bislang lediglich firmennahe Ausgabeverfahren für Mobilgeräte wie **COPE** (Corporate Owned, Personally Enabled) oder **COBO** (Corporate Owned, Business Only) ohne Sorgenfalten umsetzen. Das wurde mit dem Device Enrollment Program (DEP), welches Ende 2019 durch den Apple Business Manager abgelöst wurde, auch automatisiert möglich. Die Geräte bekamen dank Volumen Purchase Program (VPP) zudem Software aufgespielt, ohne dass der Anwender das Geld vorstrecken und kompliziert rückvergütet werden musste.

Für den Einsatz von BYOD (Bring your own Device), bei dem es dem Mitarbeiter freigestellt ist, sein privates Gerät zu verwenden, hatte iOS nur die oben gezeigte Geräteregistrierung vorgesehen. Das Problem dabei: Obwohl der Anwender sein Gerät sowohl privat als auch beruflich nutzt, wird es in Gänze administrativ verwaltet. Auch wenn diese Geräte nur selten im Betreuungsmodus (Supervised) betrieben werden, der sehr tiefgreifende administrative Vorgaben erlaubt, sind die tech-

nischen Möglichkeiten der Administratoren mit dem Datenschutz der Anwender und dem gewünschten Komfort selten oder gar nicht vereinbar. So erhält der Geräte-Admin etwa potenziell Einsicht in installierte Apps, kann die PIN-Sperre und die Verknüpfung mit einer Apple-ID aufheben, diverse Apps verbieten oder das Gerät sogar remote löschen.

Eine Alternative stellen Lösungen für das Mobile Application Management (MAM) dar. Hier unterliegen nur spezifische, beruflich genutzte Apps einer Verwaltung. Damit soll sichergestellt werden, dass bestimmte Firmen-Policies eingehalten werden und sensible geschäftliche Daten nicht in falsche Hände geraten.

Auch dies ist im iOS-Umfeld nicht der Weisheit letzter Schluss. Während ein Mobile-Device-Management-System es erlaubt, die Anwendungen auf mobilen Endgeräten zwischen einem Firmenbereich (durch die Firma verteilte Managed Apps) und einem Anwenderbereich (durch den Anwender installierte private Apps) zu unterscheiden, können MAM-Apps nur durch den Anwender selbst installiert werden. Die Trennung und Konfiguration erfolgen hier auf App-Ebene, etwa bei Microsoft-Office-Apps mit jeweils dienstlichem und privatem Umfeld. Auch der Schutz vor Datenabfluss muss in MAM-Szenarien in den jeweiligen Apps aufwändig durch die Hersteller realisiert werden. Orientiert man sich an den Microsoft-Apps, gehören dazu etwa:

- Steuern der Möglichkeit von Benutzern, Unternehmensdateien zu verschieben;
- Konfigurieren von Einschränkungen für die Zwischenablage;
- Erzwingen der Verschlüsselung von gespeicherten Daten (iOS: AES256, Android: OpenSSL128bitAES);
- Remote-Zurücksetzung von Unternehmensdaten;
- Erzwingen der Verwendung eines verwalteten Browsers;
- Erzwingen einer PIN-Richtlinie;
- Überprüfen der Geräteintegrität und -konformität (z.B. Routing/Jailbreak Detection);
- Unterstützung mehrerer Identitäten.

Um keine Missverständnisse aufkommen zu lassen: Microsoft und andere Anbieter haben diese Funktionen sehr vorbildlich und umfangreich implementiert. Auch die Möglichkeit der Zwei-Faktor-basierten Anmeldung (modernAuth) durch den Microsoft Authenticator ist – meiner Meinung nach – einmalig am Markt. Er versorgt die installierten Apps mit Einmal-Kennwörtern zur Anmeldung und hält diese selbst mit einem 30 Tage gültigen Dauer-Token bereit.

Aber das Ganze führt nicht an dem eigentlichen Problem vorbei, denn nicht alle Apps stammen von Microsoft und die MAM-Apps der unterschiedlichen Hersteller sind unter einander (meist) nicht kompatibel.

iOS 13 macht BYOD massentauglich

Mit iOS 13 setzt Apple auf die Verwaltung per Benutzerregistrierung (User Enrollment). Es handelt sich hierbei um eine sehr tiefgreifende architektonische Veränderung, die jedem Anwender zugutekommt, der seine persönlichen Geräte verwenden möchte, um auf Arbeitsinformationen zuzugreifen. Das Interesse der Firmen nach Datensicherheit wird gleichermaßen eingehalten. So liefert das MDM-Protokoll bei einer Aktivierung per Benutzerregistrierung keine persistente verfügbare Identifizierungsmöglichkeiten für ein Gerät, etwa Identifier wie UDID, IMEI und dergleichen, an den Administrator, beziehungsweise das MDM-System. Zur Identifikation wird stattdessen eine individuelle User-Enrollment-ID bei jeder Aktivierung generiert und anschließend für die Kommunikation zum MDM-Server genutzt. Im Umfeld von Exchange ActiveSync wird ein spezieller EASDeviceIdentificer erstellt.

Bei der Aktivierung der Benutzerregistrierung legt iOS 13 ein Managed-APFS-Volumen mit einem eigenen abgeleiteten Verschlüsselungsschlüssel an. Verwaltete Apps legen ihre Daten dann ausschließlich auf diesem Volume ab. Dieses ist Speicherort für:

- verwaltete, der Managed-Apple-ID zugewiesene Apps;
- Daten der Notizen-App;
- Dokumente des Enterprise iCloud Drive;
- dienstliche Schlüsselbund-Einträge;
- E-Mail-Anhänge und Mail-Body
  (restliche Metadaten bleiben im normalen System);
- Kalender-Anhänge.

Hebt der Anwender (oder Admin) die Benutzerregistrierung auf, wird der zugehörige Verschlüsselungsschlüssel von iOS 13 nachhaltig vernichtet. Das Volume ist damit unbrauchbar und steht nicht mehr zur Einsicht zur Verfügung. Ein neuer Rollout per Benutzerregistrierung legt einen neuen Schlüssel und ein neues Volume an.

Technisch ist dies jedoch kein Multi-User-Ansatz, sondern ein Multi-Account-Ansatz. Denn die Benutzerregistrierung erfolgt nicht auf Basis der privat persönlichen Apple-ID des Anwenders, vielmehr muss hierzu eine „Managed Apple ID" für ihn angelegt werden. Administratoren (und Personenverantwortliche) können dies im Apple Business Manager (ABM) vornehmen. Die Zuordnung verwalteter Apps erfolgt an diese Managed-Apple-ID. Die Eigentumsverhältnisse bleiben so immer bei der Firma.

Auf die Geräte selbst, die per Benutzerregistrierung aktiviert werden, haben Firmen nur noch geringen Einfluss. So sind Administratoren etwa nicht in der Lage, diese Devices aus der Ferne zu löschen und detaillierte persönliche Daten wie zum Beispiel installierte Apps einzusehen. Kennwortregeln können nicht mehr mit (extremer)

Komplexität eigefordert werden. Auch das Entfernen einer PIN (falls der Anwender die PIN vergessen hat) geht nicht mehr. Wird eine komplexe PIN benötigt, kann der Administrator das einfordern. Die „Definition", was eine komplexe PIN ist, gibt allerdings Apple vor.

Die VPN-Konfigurationen beschränken sich auf App-VPN-Konfigurationen. Nur der Zugriff auf definierte (verwaltete) Domänen, beispielsweise über verwaltete Apps, kann ein VPN aufbauen. Ein Full VPN, bei dem der komplette Datenfluss umgeleitet wird, sowie das Konfigurieren genereller Proxys, ist technisch nicht mehr möglich.. Betreuungsmodus-Konfigurationen zu nutzen ist gänzlich unmöglich.

Trotzdem können Geräteadministratoren weiterhin etwa:

- dienstliche Apps und dienstliche Konten installieren und konfigurieren (siehe AppConfig);
- einen Passcode im Gerät erzwingen;
- Daten abfragen, die sich auf die dienstlichen Apps, Zertifikate und Konfigurationsprofile beziehen;
- per-App-VPN konfigurieren.

Viele MDM-Anbieter haben hinsichtlich des Datenschutzes einiges an Auswertungen von Vornherein proaktiv bei sich eingeschränkt. Allerdings waren das nicht alle Hersteller und „technisch möglich" war viel mehr als das, was Administratoren gesehen haben. Das Vertrauensverhältnis zwischen Anwendern und Administratoren war daher oft gestört. Die Benutzerregistrierung kann das ändern.

Vergleich mit Android: Der Multi-User-Modus

Mit Android Q wird es nun möglich, Arbeitsprofile für Android Q per QR-Code oder einer dem iOS DEP ähnlichen Zero-Touch-Funktionalität zu aktivieren. Während der Bereitstellung eines firmeneigenen Geräts können nun Device Policy Controller Apps (DPCs) ein Work-Profil oder eine komplette Geräteverwaltung initiieren.

Android Enterprise Work Profile stellen eine Multi-User-Umsetzung dar. Diese erlaubt es Android, ein paar Dinge anders zu handhaben. So können Apps unter Android in beiden „User-Kontexten" installiert werden und damit parallel als eine private und eine berufliche Version existieren. Ein Koffersymbol macht den Kontext deutlich.

Die Dual-Persona-Umsetzung hat außerdem den Vorteil, dass Anwender ihren dienstlichen Bereich zum Beispiel im Urlaub deaktivieren und am ersten Arbeitstag wieder aktivieren können. Unter iOS gibt es diese Möglichkeit (noch) nicht.

Viele erwähnen an dieser Stelle den Google Play Store for Work und weisen darauf hin, dass es sich um einen eigenen App Store für geschäftliche Apps handelt. Unternehmen können hier eine verwaltete Version von Google Play nutzen, um so geschäftlich genutzte Apps zu verteilen. Dies ist im Grunde mit dem VPP/ABM von Apple vergleichbar. Apps, die von anderen Quellen als Google Play (oder anderen vertrauenswürdigen App-Stores) heruntergeladen werden, werden als Apps aus unbekannten Quellen bezeichnet. In Android Q können Administratoren von Arbeitsprofilen nun verhindern, dass solche Apps an beliebiger Stelle auf dem Gerät installiert werden können. Trotz dieser Einschränkung kann ein Anwender mit Gerätezugriff per Android Debug Bridge (adb) weiterhin Apps – auch aus unbekannter/nicht vertrauter Quelle – installieren.

Unter iOS kann dies komplett unterbunden werden, dafür gibt es bei iOS nur den AppStore als verlässliche Quelle, beziehungsweise durch das MDM selbst verteilte Apps. Weitere AppStores, die in einer Firma durchaus existieren können, haben diese Möglichkeit nicht. Es muss somit alles über den AppStore und/oder das MDM gehen. Der Vollständigkeit halber sei gesagt, dass natürlich auch MDM-Einstellungen im Company-Umfeld existieren, die Software aus fremder Quelle komplett erlauben. Diese Einstellung ist jedoch nicht zu empfehlen.

OEMConfig soll MDM-Einstellungen unter Android vereinheitlichen

Bisher haben OEM-Hersteller „Plain Android" so angepasst und erweitert, dass ihre Endgeräte besser über ein MDM-System gesteuert und konfiguriert werden können. Die Hersteller von MDM-Systemen mussten diese Funktionen dann aufwändig in ihre Lösungen integrieren. Dies wird sich nun mit OEMConfig ändern. Zwar passen die Firmen weiterhin Android mit eigenen MDM-Erweiterungen an, die über die Fähigkeiten von Android Enterprise hinausgehen. Zusätzlich erstellen sie aber eine eigene OEMConfig-Anwendung, um diese neuen, selbst definierten APIs zu referenzieren und zu registrieren.

Der OEM-Hersteller lädt diese OEMConfig-Anwendung in Google Play und stellt sie so zur Verfügung. Ein Administrator einer Organisation kann nun diese App auf seinen Geräten verteilen und per App-Konfiguration ansprechen. Die App-Konfiguration übermittelt die Konfiguration an die OEMConfig-Anwendung und diese führt die Gerätekonfiguration durch. So erhält das Unternehmen sofort die neueste Version (Tag-0-Unterstützung) der referenzierten APIs, die der OEM hinzugefügt hat.

Bei OEMConfig handelt es sich um nichts anderes als eine kleine Anwendung, die vom Hersteller erstellt und vorinstalliert wird – quasi ein spezieller MDM-Client für die entsprechende Marke oder oft auch ein entsprechendes Gerät. Im Gegensatz zu iOS herrscht somit weiterhin je nach Hersteller und Device ein Unterschied in der „Mächtigkeit" von MDM-Kommandos.

Fazit

Mit iOS 13 ist Apple endlich an dem Punkt angelangt, den sich viele Administratoren gewünscht haben. Ich bin mir sicher, dass Apple das Thema „Eine App doppelt installieren" zukünftig auch angehen wird. Im Grunde müsste Apple nur den „Bundle-Identifier" im dienstlichen Kontext ergänzen. Der Bundle-Identifier ist eine eindeutige Kennung für eine Anwendung und beginnt normalerweise mit einem Domainnamen in umgekehrter Reihenfolge der Wörter (z.B. de.computerwoche). Daran wird dann noch der Name der jeweiligen App angehängt. Apple könnte hier noch das Attribut „.managed" anfügen, um eine dienstlich genutzte App von der privaten Version zu unterscheiden.

Mark Zimmermann