Dossier thématique

Sécurité des données dans les applications mobiles

RESUME

L'arrivée des smartphones et tablettes modernes, à commencer par l'iPhone, en 2008 a fait apparaître plusieurs risques informatiques nouveaux. Pour le DSI et ses équipes, connaître ces risques nouveaux et les stratégies pour les limiter est crucial. Il s'agit, bien sûr, de risques liés à la sécurité des données de l'entreprise : comment assurer la confidentialité et l'intégrité de ces dernières ?

Sommaire

Introduction

L'arrivée des smartphones et tablettes modernes, à commencer par l'iPhone, en 2008 a fait apparaître plusieurs risques informatiques nouveaux.

Pour le DSI et ses équipes, connaître ces risques nouveaux et les stratégies pour les limiter est crucial. Il s'agit, bien sûr, de risques liés à la sécurité des données de l'entreprise : comment assurer la confidentialité et l'intégrité de ces dernières ?

Deux exemples permettent d'illustrer ces risques :

- Un collaborateur installe sur le même device un CRM mobile qui télécharge et sauvegarde un fichier client et une application malveillante. Cette application vole le fichier "client" et l'envoie par le réseau.
- Un collaborateur égare un device mobile qui tombe entre les mains d'une personne experte et mal intentionnée ; cette personne accède à des données confidentielles : emails, fichiers clients, logins et mots de passe, notamment en faisant une lecture systématique de la mémoire flash grâce à des outils informatiques.

Quels sont les modes d'accès des collaborateurs équipés de mobiles aux données de l'entreprise ? Quels sont les scénarios de risques possibles ? (NDLR : Nous avons privilégié cette démarche concrète par rapport à une construction de taxonomies abstraites des risques liés au mobile). Plus loin, nous aborderons la question des risques liés au mobile et qui par leur nature échappent à la vigilance de la DSI. En conséquence, quel est l'arsenal mis à notre disposition pour réduire ces risques ? NB : Le choix a été fait d'étudier séparément le dispositif de sécurité d'Apple/iOS et le dispositif de sécurité d'Android.

Pour limiter les risques liés au mobile, nous recommanderons ainsi :

1- De limiter l'accès aux données de l'entreprise à une flotte identifiée.

2 - De contrôler la flotte à travers une MDM.

3- Quand cela est possible, de choisir une flotte de devices Apple (iPhones et iPads).

Architecture des accès mobiles aux données de l'entreprise

D'abord, comment accède-t-on physiquement aux données de l'entreprise ?

Le schéma ci-dessous représente 5 cas d'accès aux données de l'entreprise :

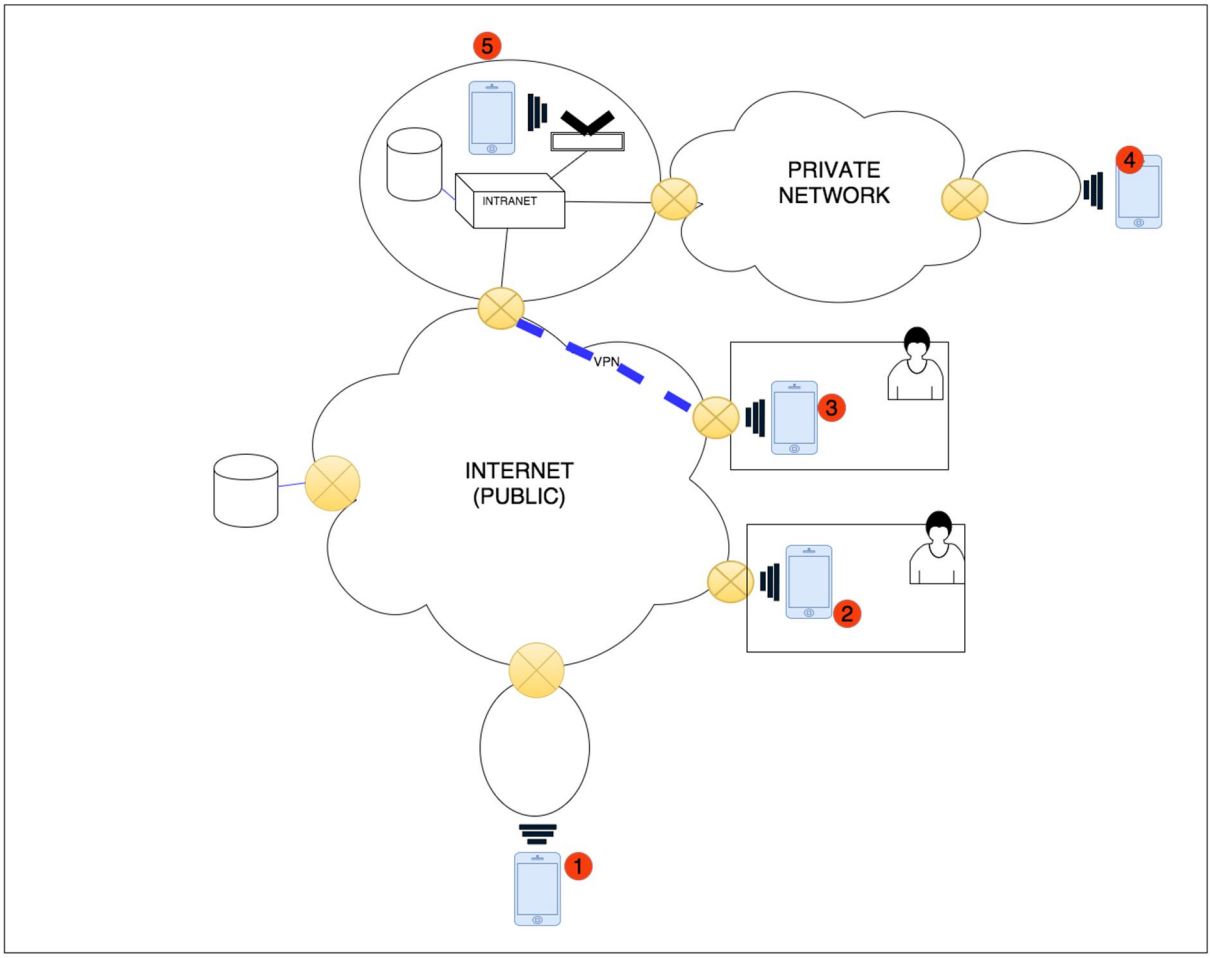

Fig.1 : Les différents modes d'accès aux données de l'entreprise

Cas 1	Le collaborateur accède via le réseau GSM à l'internet public. Il dispose alors d'une IP attribuée par son opérateur de téléphonie mobile qui joue dans ce cas le rôle de FAI. Le collaborateur peut accéder aux serveurs hébergés sur l'intranet (ouverture de ports) et les serveurs hébergés par des tiers.

Cas 2	Le collaborateur accède à l'internet depuis son domicile via sa box. Ce cas est similaire au cas1 pour l'essentiel. Le collaborateur ne peut accéder à l'intranet que via ouverture de ports sur des gateways INTRANET/INTERNET.
Cas 3	Le collaborateur est relié à l'intranet de sa société par activation de VPN. Cas ressemblant au cas d'accès numéro 5.
Cas 4	Le collaborateur accède à l'intranet de l'entreprise via un réseau privé (APN privé) d'opérateur. (cas proche du cas numéro 5)
Cas 5	Le collaborateur accède à l'intranet dans les locaux de son entreprise via un lien wifi. Par défaut, le device a alors accès aux serveurs et desktops visibles sur l'intranet.

Les scenarios à risque qu'il faut envisager

Scénario A	Le collaborateur est sur intranet via wifi, son device contient un virus qui infecte des serveurs, desktops et autres matériels connectés à l'intranet.
Scénario B	Le smartphone du collaborateur tombe entre des mains malveillantes (mais peu expertes). Le verrouillage du téléphone est négligé (Code PIN trop simple. Ex : "0000"). Les données professionnelles sont ainsi exposées : mails, carnets d'adresses, agendas, fichiers clients…
Scénario C	Le collaborateur installe une application maligne (virus, cheval de Troie) qui va accéder aux données installer sur le téléphone et les dégrader.
Scénario D	Le collaborateur compromet la sécurité du téléphone par un root ou un jailbreak. Il peut être piraté à son insu, et des applications malignes auront accès aux informations stockées sur le device.

Scénario E	Un hacker diffuse une application contenant un virus par un e-mailing. Le collaborateur installe l'application et compromet la sécurité de son téléphone.
Scénario F	Un pirate accède physiquement au device du collaborateur, installe un malware et le lui retourne sans que celui-ci ne se rende compte que son téléphone est compromis. Un pirate accède physiquement au device du collaborateur et vole toutes les données par la lecture pure et simple de la mémoire flash du téléphone.
Scénario G	Un collaborateur accède aux données de l'entreprise via un webservice sans SSL. Un pirate connecté au même réseau (wifi public par exemple) renifle les paquets et vole les informations transportées (login, password, etc…)
Scénario H	L'entreprise ou même un tiers édite une application mobile embarquant une technologie de type aspirateur de site web pour analyser et exploiter des espaces web privés. Cette application stocke alors sur le device des données extraites de l'espace client. Ces données sont ainsi exposées au vol de device ou à la présence de malwares.
Scénario I	Le collaborateur active un service de type cloud sur son device, il se connecte au même cloud à partir de plusieurs devices. Les données des applications métiers se trouvent alors dupliquées sur plusieurs devices à la fois. Le risque est ainsi démultiplié d'exposer les données de l'entreprise.

Des risques qui par nature échappent à la vigilance des DSI

Les technologies mobiles encouragent les utilisateurs à recourir à des plateformes collaboratives ouvertes au public : les drives de Google, des Dropbox, des comptes e-mails grand public.

Par définition, la DSI n'a pas la maîtrise de ces comportements. Rien n'empêche, par exemple, un trader dans une banque de passer certains ordres par une messagerie grand public. Récemment, Mme Clinton s'est trouvée au cœur d'un scandale pour

avoir utilisé une messagerie personnelle afin de transmettre des informations classifiées secret défense. Mais que faisait la DSI ? La DSI n'était simplement pas au courant et ne pouvait l'être…

Bien qu'ils ne soient pas liés aux technologies mobiles, ces risques sont fortement augmentés par l'utilisation de ces dernières. En effet, le téléphone par nature est exposé aux risques de vol et piratage, par son accès à des bornes publiques.

Que faire, sinon sensibiliser les collaborateurs et les informer ?

Quelles stratégies pour sécuriser les données de l'entreprise ?

Nous avons choisi de différencier les stratégies de sécurisation selon la plateforme (iOS ou Android). En effet, malgré des points communs très nombreux, les différences de philosophie, d'implémentation et les risques résiduels sont suffisamment marqués pour que chaque plateforme soit abordée indépendamment.

Il convient d'ailleurs de noter une différence évidente : iOS implique un matériel dédié (iPhone, iPad et certains iPods) produit par un seul fabricant - Apple - ; Android, par contre, est shippé sur du matériel divers et varié, produit par un panel de constructeurs : Samsung, HTC, Sony, Motorola, LG, et autres …

Aussi, dans le cas de l'iOS, le hardware est un élément central de la stratégie de sécurité, et confère un avantage important dans l'édifice de défense de la firme à la pomme.

Le process sécurité Apple

Sécurité système

Le secure boot chain

Le secure boot chain est l'enchaînement de protocoles de sécurité mis en place du démarrage du téléphone au chargement du noyau iOS. Celui-ci se décompose en plusieurs étapes :

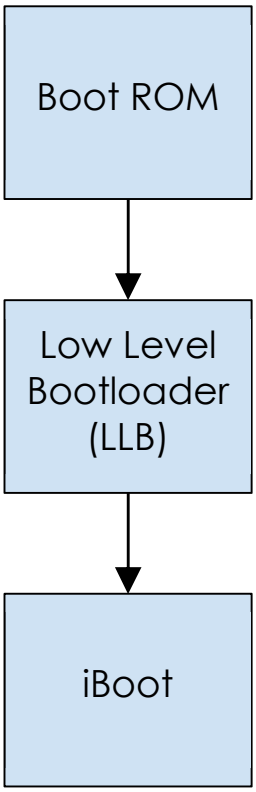

Fig.2 : Le processus de secure boot chain

Le boot ROM fait partie intégrante du hardware Apple. Il exécute du code contenu dans une partie de la mémoire vive en lecture seule. Il contient par ailleurs une clef publique permettant de vérifier l'intégrité du LLB et que celui-ci est bel et bien signé par Apple. Le boot ROM ne peut-être altéré, le reste du système lui fait donc implicitement confiance.

Le LLB quant à lui se charge d'amorcer le processus de démarrage. Une fois sa tâche terminée, à l'instar du Boot ROM, il va vérifier et démarrer iBoot. Lorsque le téléphone démarre, on est alors sûr dans le cas nominal que le système est intègre et que tous les processus lancés ont été approuvés par Apple. Si une de ces étapes fait défaut, le téléphone se met alors en mode DFU (Device Firmware Upgrade), le seul moyen de le réutiliser étant de le remettre à zéro ou en "factory mode".

Ce secure boot chain est le point névralgique de la sécurité d'Apple. Corrompre ou bypasser cette étape permet rend tout processus tiers légitime auprès du système et permet une plus grande liberté d'action sur le téléphone tout en sacrifiant sa sécurité. Le processus de "jailbreak", que nous expliquerons en détail dans cet article, s'applique exclusivement à briser ou ignorer le secure boot.

Le chiffrement des données

Le module de chiffrement hardware

Tout appareil Apple possède dans sa mémoire vive un module dédié au chiffrement. Étant aussi situé dans une zone en lecture seule, celui-ci n'est pas corruptible. D'autre part, il utilise l'algorithme AES 256, réputé inviolable face au "brute force" (il faudrait des dizaines de milliards d'années pour aboutir à un résultat dans l'état actuel des choses !).

Les deux clefs de chiffrement sont le GID (Group IDentifier) propre à une série de téléphone (l'iPhone 6 par exemple) et l'UID (User IDentifier) propre à l'appareil et que personne, a priori, ne peut connaitre, y compris Apple. Le GID est utilisé pour les opérations non critiques alors que l'UID est la clef utilisée pour toute opération pouvant compromettre la sécurité de l'appareil ou la vie privée de l'utilisateur.

Le module de chiffrement software

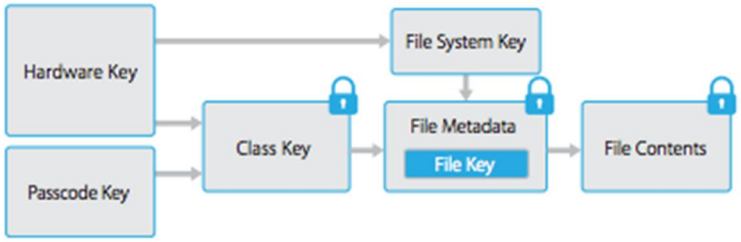

Fig.3 : Module de chiffrement logiciel d'iOS

A chaque fichier créé dans la partition data est associé une clef de chiffrement. Cette clef est alors donnée au module de chiffrement matériel. Ce module écrit la clef dans les métadonnées du fichier et fournit au système une nouvelle clef permettant de déchiffrer les métadonnées du fichier.

A l'ouverture du fichier, le système déchiffre les métadonnées, extrait la clef de chiffrement et laisse la main au module matériel qui aura pour mission de déchiffrer le contenu du fichier.

Le keychain

C'est l'espace de stockage sécurisé d'Apple, permettant d'enregistrer des données critiques comme des mots de passe ou des token d'authentification par exemple.

Concrètement, c'est un fichier sqlite situé dans */private/var/Keychains/keychain-2.db*. Toutes les entrées dans cette base de données sont bien entendu chiffrées.

Il est aisé de lire le keychain sur un appareil jailbroken. Les données sont chiffrées mais possiblement accessible à l'aide d'un brute force.

Protection de la vie privée

Le sandbox applicatif

Chaque application est installée dans une sandbox, zone cloisonnée et dédiée à l'application empêchant l'accès aux fichiers extérieurs, hardware, préférences, etc…

Cette sandbox est aussi en place sur un iPhone jailbreaké. Cependant, les applications systèmes n'ont pas cette restriction et il est ainsi possible, dans le cas d'un iPhone jailbreaké, de créer une application système tierce ayant accès à l'intégralité des fichiers d'application (en particulier les bases de données sqlite).

Il est donc intéressant de considérer une sécurité "in-app" en plus de celle d'Apple, considérant qu'un iPhone jailbreaké laisse potentiellement toute application malveillante accéder aux ressources de chacune des autres applications.

Permissions

Tout accès aux données personnelles de l'utilisateur est soumis à son approbation préalable (sous forme de pop-up). L'accès ou le blocage peuvent être paramétrés pour chaque application dans les paramètres du téléphone. Il est intéressant de noter que l'accès au iCloud drive est permis par défaut.

Contourner la sécurité Apple : le jailbreak

Définition

Le jailbreak est l'action de donner à l'utilisateur un accès root au système d'exploitation iOS. On peut alors installer des applications tierces, non vérifiées par Apple.

Le jailbreak n'est possible qu'en brisant le "secure boot chain" ou en l'ignorant. Dans les deux cas, la mise en place de cette opération nécessite l'exploitation d'une vulnérabilité du noyau iOS ou d'une application système. Le jailbreak injectera ainsi du code permettant de lancer des processus qui deviendront légitimes ou d'ignorer certaines vérifications du "secure boot chain".

Comment détecter le jailbreak ?

Détecter le jailbreak est une opération relativement simple. Il s'agit de détecter certains comportements propres à un téléphone jailbroken ou de chercher les traces d'applications tierces connues. Dans le premier cas, on pourra tester si une application peut écrire dans un répertoire qui est hors du périmètre de son sandbox. Si cela est possible, on sait immédiatement que le téléphone est jailbroken. Similairement, on peut aller chercher la trace dans certains répertoires d'applications comme Cydia, l'appstore non officiel des appareils jailbroken.

Cependant, les jailbreaks les plus évolués placent dans des répertoires non triviaux les applications nécessaire à leur fonctionnement et vont même jusqu'à modifier le comportement de certaines méthodes du kit de développement Apple afin de tromper les routines de détection d'application "anti-jailbreak".

Comment assurer la sécurité de son parc mobile ?

La réponse réside dans le MDM (pour Mobile Device Management). De nombreuses solutions professionnelles comme AirWatch ou MobileIron proposent de gérer la configuration, les mises à jour et le monitoring des téléphones mobiles de collaborateurs dans une entreprise.

La sécurité des appareils Apple, tant sur le plan hardware que software, étant particulièrement poussée, il ne reste plus qu'à protéger les échanges réseau et ce que les utilisateurs installent sur leur téléphone. Dans le premier cas, la meilleure solution est la mise en place d'un VPN (Virtual Private Network) permettant un contrôle poussé des échanges réseau entre les téléphones et le reste du monde.

Dans le second cas, cela nécessite de déployer les applications autorisées sur les téléphones du parc mobile et de désactiver l'accès à l'App Store. Cette sécurité peut toutefois être contournée par des publications ad hoc ou in house, permises

par Apple, et qui consistent à télécharger l'ipa (le fichier exécutable iOS) directement depuis n'importe quel serveur.

La question du jailbreak reste cependant en suspens. Nous avons précédemment vu que la détection du jailbreak peut-être compliquée voire impossible dans certains cas. Certains fournisseurs prétendent pouvoir détecter à 100% les appareils jailbreakés et pouvoir les exclure d'un parc mobile. Mais il existe de nombreux tutoriels sur le net expliquant comment outrepasser les applications mises en place par les principaux fournisseurs de solution MDM.

Conclusion sur la sécurité native d'Apple

La sécurité fournie par Apple est extrêmement élaborée, utilise des technologies à la pointe et saurait difficilement être mise à mal quand leurs appareils ne sont pas jailbroken. D'autre part, les mises à jour régulières, et notamment la sortie d'iOS 9, rendent de plus en plus difficile la réalisation d'un jailbreak.

Ces propos sont à nuancer tant les pirates sont prompts à s'adapter aux mises à jour d'iOS. Alors que la détection du jailbreak n'est pas fiable, on peut se demander comment protéger les téléphones de ses collaborateurs.

Face au danger du JailBreak, on peut recourir à la sensibilisation des usagers aux conséquences du recours à cette procédure.

Le process sécurité Android

Le noyau

Android repose sur un noyau linux dont la fiabilité et la sécurité ont été éprouvées dans des environnements sensibles. Des mises à jour régulières corrigent les failles de sécurité connues.

Vue d'ensemble

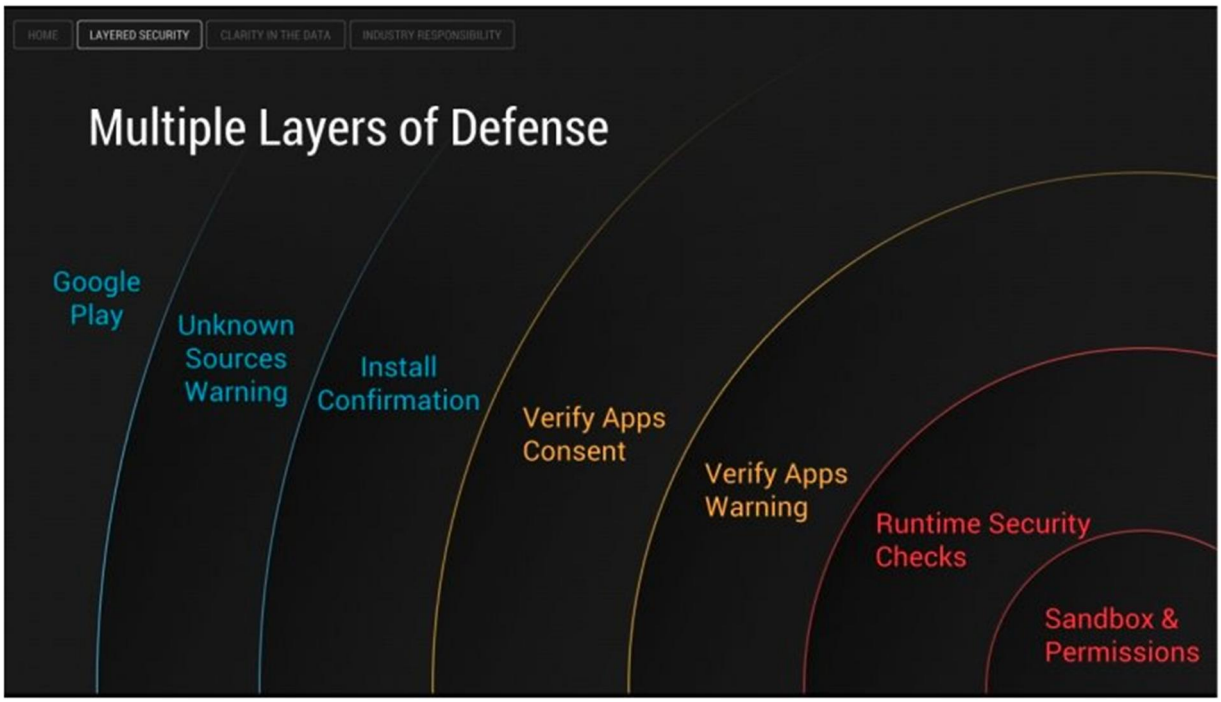

Fig.4 : Process sécurité multicouches d'Android

Le process sécurité pour téléphone non rooté se divise en plusieurs couches. Au préalable une vérification est faite sur Google Play (le store applicatif officiel d'Android) à chaque upload d'application. Celui-ci vérifie en particulier la signature d'un malware connu. Si l'application provient d'une source inconnue, l'utilisateur est alors prévenu du risque qu'il encourt. Installer une application nécessite forcément le consentement de l'utilisateur, il n'existe en effet pas d'installation silencieuse. Par ailleurs, de nombreuses vérifications sont faites pendant que l'application est en vie. Enfin, tout comme sur iOS, les applications sont "sandboxées" et doivent obtenir la permission de l'utilisateur pour avoir accès à des données personnelles comme les contacts, les sms ou encore le calendrier.

Protection de la vie privée

Le chiffrement des données

Par défaut, les fichiers ne sont pas chiffrés sur Android. Il est toutefois possible d'activer une option le permettant. Le chiffrement proposé est AES 128 (256 optionnel), le même algorithme qu'utilise Apple.

Si l'option de chiffrement est activée, alors le système chiffre le fichier lors de l'écriture sur le disque et le déchiffre lorsqu'un processus essaie de le lire.

Le sandbox applicatif

Le principe est le même qu'Apple, chaque application a sa propre zone personnelle et ne peut interférer sur celle des autres. Il est possible de contourner ce sandbox en autorisant explicitement d'autres applications à accéder à certaines données (habituellement au moyen de ce qu'on appelle un "content provider").

Ce sandbox repose sur le fait que chaque application a son propre user, au sens "linuxien" du terme. Ainsi, les applications n'ont pas les accès en lecture ou en écriture dans les dossiers hors de leur périmètre, à l'exception de l'"external storage" (la carte SD par exemple) qui est une zone commune et accessible par l'ensemble du système.

Contourner la sécurité : le rooting

Définition

L'utilisateur root a tous les privilèges possibles sur Linux. Sur Android, dans le cas nominal, les droits root sont accordés au noyau et aux applications systèmes.

Dans le cas d'un téléphone rooté, on permet à chacun des utilisateurs système de devenir "super utilisateur". Il est alors aisément possible de changer le comportement système et d'accéder à l'ensemble des données des autres applications.

Soulignons tout de même qu'à chaque fois qu'une application nécessite le statut de "super utilisateur", le propriétaire du device est prévenu par l'apparition d'une "pop-up" et doit explicitement accepter l'opération.

Détection d'un téléphone root

Les méthodes de détection de root sur Android sont extrêmement similaires à celles d'Apple pour le jailbreak. Il s'agit encore une fois de chercher les traces d'applications connues, qui ne fonctionnent qu'avec les droits root ou d'essayer de faire des opérations normalement impossibles avec des privilèges standards.

Encore une fois, ces méthodes étant assez peu précises, de nombreuses techniques permettent de les contourner.

Conclusion sur la sécurité Android

La sécurité mise en place sur Android est similaire en de nombreux points à celle d'Apple. Toutefois, si la sécurité d'Android repose uniquement sur le software, Apple propose une protection supplémentaire au niveau hardware qui la rend plus efficace. D'autre part, Android laisse plus de liberté à l'utilisateur, notamment celle de donner la permission à des applications d'effectuer des opérations dangereuses pour l'intégrité des données personnelles. Les solutions MDM sur Android sont équivalentes à celles d'iOS, il est possible de mettre de nombreuses barrières empêchant l'utilisation de fonctionnalités dangereuses comme l'installation d'applications issues de sources inconnues (en dehors du store officiel). Il reste néanmoins difficile d'empêcher les contournements que sont le root et le jailbreak. Il est tout à fait possible de contourner la détection de telles pratiques et ainsi d'introduire des appareils potentiellement dangereux dans un parc mobile a priori sécurisé par une solution de mobile device management.

Conclusion générale

iOS présente une sécurité plus contraignante et les utilisateurs sont statistiquement plus à jour sur cette plateforme que sur celle de son principal concurrent, ce qui réduit de facto les risques d'attaque. Ce système d'exploitation semble donc être à privilégier dans des environnements sensibles, pour les professionnels. Il demeure que la sécurité mise en place, bien que robuste, n'est pas fiable à 100% en raison d'un possible jailbreak.

Il semblerait donc que ce soit le comportement des collaborateurs vis à vis de la politique de sécurité de l'entreprise qui requiert une véritable attention. On remarque en effet que les attaques sur des systèmes Android ou iOS "sains" sont extrêmement rares et demandent un niveau d'expertise élevé aux pirates pour un résultat dont l'impact reste généralement faible. Les conséquences sont bien plus dramatiques si l'utilisateur permet à une application douteuse d'avoir un accès administrateur.

A ce moment-là il est difficile de faire marche arrière...

Dossier rédigé par Julien Datour, Ingénieur Développeur Mobile chez Qualia Systèmes.

Qualia Systèmes est une société experte dans le développement d'applications mobiles sur smartphones et tablettes (iOS, Android).