La sécurité des smartphones

Romain Raboin
rraboin(@)atlab.fr

atlab.fr

Résumé Le nombre de smartphones est en constante évolution. Toujours plus puissant, les systèmes qu'ils embarquent sont proches de ceux des ordinateurs de bureau. Cet article présente un état de la sécurité de ces appareils. Nous verrons tout d'abord la résistance des systèmes d'exploitation qu'ils utilisent et les attaques qu'ils ont subies. Ensuite après avoir étudié les fonctionnalités du système Windows Mobile, nous présenterons un ensemble d'attaques avancées sur ce système et les risques que représente un smartphone compromis notamment pour les réseaux avec lesquels il est interconnecté. Pour terminer nous conclurons sur les méthodes de sécurisation des mobiles.

1 Introduction

Le téléphone portable a subi des révolutions en alliant à sa fonction initiale (échange de voix) des fonctionnalités complémentaires personnelles (sonneries personnalisées, musique, radio, caméra, photos, etc.) ou professionnelles (agenda, messagerie, stockage, etc.). Il se rapproche donc plus aujourd'hui d'un ordinateur de poche que d'un équipement de téléphonie sans fil. Nous parlons alors de smartphone, soit l'association d'un téléphone portable et d'un PDA. Il fait dorénavant partie intégrante du système d'information de l'entreprise, mais reste aussi utilisé dans un cadre personnel. Grâce aux technologies de synchronisation, il détient une grande partie des informations du poste de travail de l'entreprise (contacts, calendriers, mails et documents divers) et est souvent équipé d'une caméra ou encore d'un GPS.

Connecté à Internet, branché sur les postes clients de l'entreprise, équipé des technologies de communication sans fil, le smartphone est à la frontière entre téléphonie et informatique. Il est de ce fait un risque pour la sécurité du système d'information de l'entreprise et une cible de choix dans le cadre de l'espionnage industriel. La sécurité des smartphones est un sujet d'actualité, de nombreuses recherches ont été effectuées récemment, on a pu voir plusieurs présentations sur le sujet à des conférences comme le Chaos Communication Congress [1] ou CanSecWest [2].

Cet article présente une vue d'ensemble des attaques sur les systèmes d' exploitation des smartphones et plus particulièrement sur le système Microsoft Windows Mobile.

2 Différents OS sur les smartphones

Le marché mondial des smartphones est en forte croissance et représente aujourd'hui environ 10% du marché de la téléphonie mondiale [3]. Les ventes de smartphones ont augmenté considérablement pour atteindre le nombre record de 39,9 millions de smartphones vendus sur le dernier quart de l'année 2008. Les chiffres *(Fig. 1)* montrent que le marché des systèmes d'exploitation des smartphones est principalement dominé par quatre systèmes :

- Symbian OS
- iPhone OS
- RIM BlackBerry Operating system
- Windows Mobile de Microsoft

Company	3Q08 Sales	3Q08 Market Share (%)	3Q07 Sales	3Q07 Market Share (%)	3Q08 - 3Q07 Growth (%)
Symbian	18,179	49.8	20,664	63.1	-12.0
Research In Motion	5,800	15.9	3,192	9.7	81.7
Mac OS X	4,720	12.9	1,104	3.4	327.5
Microsoft Windows Mobile	4,053	11.1	4,180	12.8	-3.0
Linux	2,622	7.2	2,884	8.8	-9.1
Palm OS	780	2.1	383	1.2	103.3
Others	361	1.0	345	1.1	4.6
Total	**36,515**	**100.0**	**32,753**	**100.0**	**11.5**

Fig. 1. Répartition des ventes de smartphone dans le monde, pour la fin 2008, en fonction des systèmes [3].

Ces terminaux mobiles de plus en plus présents dans notre environnement sont aussi de plus en plus puissants et permettent de stocker un grand nombre d'informations souvent confidentielles, jusqu'à 16Go pour les iPhones. Ces évolutions ont alors un impact non négligeable sur la sécurité de ces terminaux mobiles :

- de nombreux moyens de communication (réseau WiFi, Bluetooth, synchronisation) favorisent les possibilités d'infection par un malware ;
- leurs tailles et leurs usages nomades favorisent leurs pertes ou vol ;

– les technologies de géolocalisation peuvent engendrer des atteintes à la vie privée.

2.1 Symbian OS

Le système Symbian OS est la cible de nombreux malwares, plus nombreux que sous d'autres OS. C'est certainement dû au fait qu'il soit très répandu et qu'il soit très facile de développer dessus notamment grâce à des interpréteurs de langage de scripts inclus nativement, comme python. Cependant Symbian OS depuis la version 9 intègre un modèle de sécurité pour contrer les malwares en utilisant un système de signature des binaires. Tous les binaires doivent être signés pour être exécutés sur la configuration par défaut. Il existe plusieurs façons de signer un binaire. La première est d'auto-signer un binaire mais celui-ci ne pourra pas être exécuté sur un système original fourni. Les deux autres méthodes pour obtenir un binaire signé sont les suivantes :

– utiliser le programme *Open Signed Online* ;
– acheter un certificat chez Symbian.

La première méthode, gratuite impose de fortes restrictions. Les binaires sont signés pour un unique IMEI (numéro qui permet d'identifier de manière unique chacun des terminaux mobiles) et n'auront accès qu'à une partie limitée du système et des API qu'il propose. Autrement dit, un malware signé avec cette méthode n'aura pas accès à l'ensemble des fonctionnalités du système et pourra fonctionner uniquement sur un seul mobile. La deuxième méthode consiste à acheter un certificat auprès d'une autorité de certification. En plus du coût, l'auteur d'un malware devra donner des informations personnelles, vérifiées par l'autorité associée à ce certificat. Une présentation a été faite durant la conférence BlackHat Japon fin 2008 [4] qui détaille davantage ces systèmes de signature et leurs faiblesses. Malgré cette protection, des malwares existent aussi pour la version 9 de Symbian, principalement pour deux raisons :

– des malwares arrivent parfois à obtenir des certificats valides ;
– les utilisateurs modifient leurs systèmes pour enlever cette protection contraignante.

Une fois le mobile infecté, ces malwares utilisent divers moyens de propagation. Par exemple, Worm.SymbOS.Comwar.a [5] se propage par exemple via MMS et Bluetooth. Pour d'autres comme Worm.SymbOS.Lasco.a [6], l'infection fonctionne en deux étapes :

– sous Windows XP en cherchant à infecter les fichiers .sis (fichiers d'installation de logiciels) présents sur la machine ;
– sur le téléphone en tentant une propagation via Bluetooth.

On voit aussi apparaître depuis peu des malwares dont les objectifs sont financiers pour les auteurs. Par exemple Trojan-SMS.Python.Flocker découvert fin janvier 2009 a pour but d'envoyer des SMS surtaxés. D'autres encore plus évolués comme SymbOS/Yxes.A !worm [7] utilisent le même principe de numéro surtaxé avec des techniques de propagation via l'envoie d'emails ou SMS. Ce vers possède la particularité d'être signé par un certificat officiel Symbian, ce certificat ayant été rapidement invalidé limitant ainsi la propagation.

Pour terminer, Symbian OS possède aussi le plus grand nombre de produits d'espionnage commerciaux. Certains de ces produits comme FlexiSPY (détaillé par la suite) possèdent une signature valide fournie par Symbian.

2.2 iPhone OS

Le système iPhone OS est un dérivé du système Mac OS X. Il possède une séparation de privilèges et des droits de fichiers proches des systèmes UNIX. En plus du compte root, la configuration de l'OS fournit un compte utilisateur nommé mobile pour les applications qui n'ont pas besoin de privilèges systèmes.

Même s'il est relativement simple de développer des applications pour l'iPhone grâce au SDK et à une suite d'outils fournie gratuitement par Apple, l'iPhone OS possède un système de signature proche de celui utilisé par Symbian OS. Tous les binaires doivent être signés par Apple qui se réserve le droit de refuser de signer une application suite à un audit de fonctionnalité. Cette sécurité étant contraignante pour les utilisateurs, une pratique courante consiste à déverrouiller « *jailbreaker* » l'iPhone afin de supprimer cette limitation. Une fois la protection supprimée, il est possible d'exécuter sans restriction des binaires non signés. L'exécution de malwares ou de logiciels espions devient alors possible.

FlexiSPY utilise ce procédé pour s'installer sur l'iPhone, il possède aussi une fonctionnalité permettant de cacher le *jailbreak* qu'il peut faire lors de son installation. Il faut noter aussi que le procédé de *jailbreaking* rend l'iPhone plus vulnérable à des attaques distantes notamment à cause de l'activation du serveur d'administration à distance *ssh* et des mots de passe par défaut utilisé sur ces systèmes.

2.3 RIM Blackberry OS

BlackBerry OS est développé par la société RIM. Il est aujourd'hui distribué pré-installé sur les BlackBerry dans sa version 4.6. Ce système possède peu de sécurité contre les malwares. Il est relativement facile de développer une application pour cette plate-forme et de la distribuer sans contrainte. Malgré tout et bien que son utilisation massive dans le milieu de l'entreprise et les informations qu'il contient peut

en faire une très bonne cible pour des attaques d'espionnage, il est pour le moment moins ciblé par les malwares et autres logiciels espions. Des travaux de recherche sur cet OS présente des utilisations malicieuses possibles :

- Blackjacking, 0wning the Enterprise via Blackberry (DefCon14) [8]
- RedBerry, Advanced Attacks via a Trojaned blackberry (KiwiCon 2007) [9]

Ces présentations ont pour objectif de démontrer l'utilisation malicieuse, dans un réseau d'entreprise, d'un BlackBerry compromis.

3 Windows Mobile

Principalement deux versions de Windows Mobile pour smartphones sont en circulation actuellement :

- Windows Mobile 5.0 lancé en 2005 et basé sur Windows CE 5.1
- Windows Mobile 6.0 lancé début 2007 et basé sur la version 5.2 de Windows CE

La version Windows mobile 6.1 est prévue pour avril 2009 et aussi basé sur Windows CE 5.2. D'un point de vue sécurité, Windows Mobile connait des risques et attaques proches des systèmes vus précédemment. La sécurité par défaut de Windows Mobile est assez faible (plus de détails par la suite) :

- possibilité d'exécuter des binaires de façon silencieuse ;
- auto-exécution à partir des médias amovibles activés par défaut ;
- pas de séparation des privilèges.

Le logiciel de synchronisation des smartphones Windows Mobile est ActiveSync, aujourd'hui dans sa version 4.5. Depuis ActiveSync 4.0, il est devenu impossible de synchroniser son mobile directement via le réseau. Cette fonctionnalité présentait en effet des faiblesses [10] en termes de sécurité : outre le fait que les données étaient envoyées de manière non chiffrées sur le réseau, aucune demande d'authentification n'était effectuée lors de la synchronisation, même avec un smartphone verrouillé. Ces deux faiblesses permettaient le vol d'informations. Premièrement, il était possible d'effectuer une attaque de type *Man In The Middle* entre le smartphone et le poste de synchronisation. Le protocole ne possédant pas non plus de système d'identification, l'attaquant pouvait alors récupérer les informations en ayant lui-même installé ActiveSync. Deuxièmement, il était possible de récupérer les informations de synchronisation en interrogeant un service ActiveSync et en se faisant passer pour un client valide. Cette technique ne nécessitait pas d'attaque préalable de type *Man In The Middle*, mais obligeait à trouver l'identifiant d'un smartphone ayant déjà été synchronisé sur l'ActiveSync interrogé. L'identifiant, encodé sur 31 bits, pouvait être obtenu par des attaques de type force brute.

3.1 Différents malwares pour Windows Mobile

Assez peu de malwares existent pour la plate-forme Windows Mobile, la plupart ne dépassant pas le stade de preuve de concept. Parmi les plus répandus, nous retrouvons par exemple Backdoor.WinCE.Brador.a [11] et Trojan.WinCE.InfoJack [12].

La backdoor WinCE.Brador.a été entièrement écrite en assembleur ARM et permet le contrôle du smartphone à distance. Elle ouvre un port en écoute sur le réseau afin de recevoir les ordres de l'attaquant. Les fonctionnalités de ce malware sont classiques : manipulation de fichiers (listage, envoi, téléchargement et suppression) et exécution de programmes. Elle se cache sur le système de fichiers sous le nom de svchost.exe, dans le dossier de démarrage du système d'exploitation. Il n'y a par contre aucun vecteur de propagation connu pour ce malware.

Le cas du trojan WinCE.InfoJack est différent. Il se propage via l'infection de fichiers d'installation .cab téléchargeables principalement sur des sites chinois (ce vecteur sera décrit en détail dans la section sur les différents types d'infection). Il se déclare comme étant du contenu additionnel au .cab original et infecte le smartphone lors de l'installation. Il attend ensuite que l'appareil soit connecté à Internet pour télécharger des malwares additionnels et envoyer des informations confidentielles à ses créateurs.

Tout comme les plate-formes Win32, il existe aussi des virus de type parasites d'exécutables pour Windows Mobile. Le Virus.WinCE.Duts.a[3] fait parti de cette catégorie. Il s'agit malgré tout plus d'une preuve de concept que d'un code ayant vraiment des intentions malveillantes.

Windows Mobile possède aussi des malwares qui ont pour but de faire gagner de l'argent à leurs auteurs comme WinCE.PmCryptic.A [13]. Ce malware utilise des techniques de polymorphisme et utilise l'infection des médias amovibles insérés sur le smartphone pour se propager.

Enfin, en Juin 2008, Petr Matousek a présenté ses recherches [14] sur la modification du noyau du système d'exploitation. Cette présentation démontre qu'il est possible de faire un pilote permettant de cacher des fichiers, des processus ou encore des clefs de la base de registre. Le code source de son rootkit n'est pas public actuellement.

3.2 Logiciels espions commerciaux

A l'inverse des malwares que nous pouvons trouver sur Internet et qui sont à l'état de preuve de concept, il existe de nombreux logiciels d'espionnage commerciaux complètement fonctionnels pour la plate-forme Windows Mobile.

Parmi les différents produits sur le marché, nous avons choisi d'étudier le plus connu, à savoir FlexiSPY. Il est sans doute le plus complet en termes de possibilité

mais aussi le seul à fonctionner sur les quatre plate-formes les plus répandues. Il s'installe via un fichier *.cab* et crée un service afin de s'assurer d'être présent à chaque lancement du smartphone. Une fois déployé, il est possible d'atteindre l'interface de configuration en composant un numéro spécifique, qui sera intercepté par le logiciel. Il offre de multiples fonctionnalités :

- interception d'appels ;
- déclenchement du micro à distance ;
- avertissement de changement de carte SIM ;
- surveillance via GPS ;
- vol de SMS, mails, MMS et historique d'appels ;
- reconfiguration à distance par SMS.

Toutes les informations collectées sont envoyées sur le site de l'éditeur et deviennent accessibles via un couple identifiant / mot de passe fourni lors de l'achat du produit. Au-delà de l'aspect légal de cette solution, on peut alors se poser la question de la confidentialité des données récoltées par l'éditeur de ce logiciel. Pour en mesurer réellement la gravité, nous avons créé un serveur pour la récolte des informations grâce à nos travaux de *reverse engineering* et ainsi comprendre pleinement le fonctionnement de ce logiciel.

4 Vecteurs d'infections sur les plate-formes Windows Mobile

La surface d'attaque des téléphones fonctionnant sous Windows Mobile est assez vaste. La plupart des malwares utilisent comme méthode d'infection des techniques d'ingénierie sociale comme l'envoi de fichiers par mail, MMS ou partage Bluetooth.

Il ne faut cependant pas oublier la possibilité d'exploitation distante de vulnérabilités logicielles. Une des premières sur Windows Mobile permet l'exécution de code sans action de l'utilisateur via l'envoi d'un MMS [15]. Des vulnérabilités dans la pile Bluetooth [16] peuvent aussi potentiellement permettre l'exécution de code à distance.

Durant nos travaux de recherche, nous avons développé plusieurs méthodes d'infections, présentées dans la suite de ce document, dont une sans aucune interaction avec l'utilisateur.

4.1 Les archives .cab

Les fichiers d'extension .cab permettent d'installer simplement une application sur Windows mobile. Un fichier .cab n'est rien d'autre qu'une archive comprenant les

fichiers nécessaires au déploiement du logiciel. Un de ces fichiers contient les détails de l'installation du logiciel au format XML. Via ce fichier XML, il est possible de créer des raccourcis ou encore d'effectuer des modifications dans la base de registre. Il est donc relativement facile d'ajouter un programme malicieux dans une archive afin qu'il soit déployé avec l'application originale.

4.2 Auto-exécution via média amovible

Les smartphones Windows Mobile 5 et 6 possèdent des fonctionnalités d'auto-exécution [17] via l'insertion d'un média amovible souvent de type *Secure Digital*. Pour pouvoir exécuter du code avec cette fonctionnalité, il faut déposer le binaire dans un chemin particulier sur la carte de stockage :
`\Carte de stockage\<`*type de processeur*`>\autorun.exe` Ce chemin peu aussi être obtenu via l'appel à la fonction *SHGetAutoRunPatH*. Un attaquant peut donc créer sur une carte plusieurs chemins pour chaque type de processeur et y placer ses logiciels malicieux pour chaque architecture. Le type de processeur du smartphone peut être facilement récupéré avec la fonction *GetSystemInfo*, la liste des processeurs supportés est inclue avec la documentation de cette fonction.

Une fois que les binaires sont correctement placés sur la carte SD, l'exécution du binaire sera faite non seulement à l'insertion de la carte mais aussi au moment où elle est retirée. En effet lorsque la carte est insérée, si elle contient un système d'auto-exécution, le binaire sera copié ici :
`\windows\Carte de stockage\autorun.exe` Lorsqu'on retire la carte, ce binaire est exécuté puis effacé. Le risque d'auto-exécution est malgré tout limité si l'application n'est pas signée puisque l'utilisateur doit donner son accord pour l'exécution.

4.3 Bluetooth et OBEX FTP

Les smartphones sous Windows Mobile peuvent recevoir ou envoyer des fichiers via le protocole OBEX (Object Exchange). Lors de la connexion d'un client sur un mobile, le client peut accéder au répertoire `\Mes Documents\Partage Bluetooth`.

Même si on peut imaginer envoyer un malware et espérer que la cible lance le malware, cette attaque a peu de chance de réussir du fait de la forte interaction avec l'utilisateur final. Cependant fin janvier 2009, une vulnérabilité de type *directory traversal* a été publiée [18]. Elle permet, si la cible accepte notre fichier, de copier le malware dans le répertoire : `..\..\Windows\Démarrage\` Lors du prochain redémarrage, le malware est exécuté. Il est aussi possible de copier le malware dans `..\..\windows\Carte de stockage\` avec pour nom `autorun.exe`, cela aura pour effet de lancer l'exécutable au moment de la manipulation d'un média amovible, mais aussi au redémarrage lorsque Windows Mobile détecte le média amovible.

4.4 ActiveSync et RAPI

Les téléphones Windows Mobile se synchronisent sur les postes clients grâce à la technologie ActiveSync afin d'actualiser les données du serveur Exchange, d'échanger des fichiers ou encore de partager une connexion Internet.

Microsoft fournit une API qui permet aux développeurs d'applications bureautiques de communiquer avec des appareils Windows Mobile. Grâce à cette API (nommé *RAPI* pour Remote API), nous avons pu développer une méthode d'infection silencieuse, sans aucune action de l'utilisateur.

Cette technique utilise un poste client Windows compromis. A la synchronisation du téléphone Windows Mobile avec le poste, il est alors possible d'utiliser *RAPI* pour copier et exécuter du code sur le smartphone.

Par défaut une demande de confirmation est faite sur le téléphone avant d'exécuter un programme non signé par un certificat reconnu. L'utilisateur est alors prié d'accepter la requête avant l'exécution effective. Cependant nous avons découvert des solutions techniques pour contourner cette demande en réduisant le niveau de sécurité du système d'exploitation (voir paragraphe suivant).

5 Les règles de sécurité de Windows Mobile

Windows Mobile possède des règles (*Policies*) qui permettent de limiter les accès à certaines fonctions de la *RAPI*. Il existe trois configurations de règles possibles :

- Closed mode : tous les accès via *RAPI* sont interdits.
- Restricted mode : certaines fonctions privilégiées sont interdites, modification dans la base de registre limitée, modification de fichiers limités, etc.
- Open mode : aucune restriction de la *RAPI*.

La configuration de ces règles se retrouve dans la base de registre à `HKLM\Security\Policies\Policies`. En plus de ces trois modes, il est possible d'affiner les règles comme par exemple modifier la politique de sécurité concernant les exécutables ou les fichiers *.cab* non signés.

Par défaut les smartphones Windows Mobile sont configurés en *restricted mode*. Il n'est pas possible alors d'exécuter un programme non signé sans afficher un message d'avertissement ou l'utilisateur doit confirmer pour autoriser son lancement.

On peut trouver une liste des réglages sur le site de Microsoft [19] avec la documentation du SDK. Parmi les réglages les plus intéressants pour un malware :

Policy Setting	Policy ID	Description
RAPI Policy	4097	This setting restricts the access of remote applications that are using Remote API (*RAPI*) to implement ActiveSync operations on Windows Mobile-based devices. Default value is 2 for Windows Mobile-based Pocket PC and smartphone : – 0 - indicates that the ActiveSync service is shut down. *RAPI* calls are rejected. – 1 - indicates full access to ActiveSync is provided. – 2 - indicates that access to ActiveSync is restricted to the *SECROLE_USER_AUTH*
Unsigned Applications Policy	4102	This setting indicates whether unsigned applications are allowed to run on a Windows Mobile-based devices.
Unsigned Prompt Policy	4122	This setting indicates whether a user is prompted to accept or reject unsigned .cab, theme, .dll and .exe files. – 0 - indicates user will be prompted. – 1 - indicates user will not be prompted.

Les clefs dans `HKLM\Security\Policies\Policies` ne peuvent pas non plus être modifiées directement via *RAPI* avec les fonctions prévues pour cette tâche comme *CeRegSetValueEx*, à cause de la configuration par défaut de *RAPI Policy* qui restreint l'accès à certaines fonctions de l'API.

Cependant il existe un utilitaire, *rapiconfig.exe*, fournit avec le SDK de Windows Mobile qui permet la modification de toutes les règles, via un fichier XML, même en étant en *resticted mode*.

Une étude approfondie de l'exécutable *rapiconfig.exe* nous a permis de voir qu'il appelle une fonction de la librairie *rapi.dll* via son ordinal. Cette fonction non documentée, *CeProcessConfig*, prend en paramètre un fichier XML décrivant les nouvelles règles à appliquer. Voici un exemple de fichier XML permettant la modification des Policies :

```
<wap-provisioningdoc>
  <characteristic type="SecurityPolicy">
    <parm name="4097" value="1"/>
    <parm name="4102" value="1"/>
    <parm name="4122" value="1"/>
  </characteristic>
</wap-provisioningdoc>
```

Ces règles, détaillées dans le tableau précédent, permettent de désactiver le contrôle de fonctionnalités privilégiées via *RAPI* et de désactiver le contrôle des signatures des fichiers. Dès lors, il est possible pour un malware sur un poste client de modifier les règles de sécurité d'un Windows Mobile et d'y exécuter du code sans alarmer l'utilisateur.

Il existe d'autres valeurs intéressantes pour un malware qui voudrait diminuer la sécurité d'un mobile. À noter aussi la possibilité pour un utilisateur d'améliorer la sécurité de son mobile en modifiant par exemple les règles concernant l'auto-exécution des médias amovibles.

6 Développement d'un logiciel espion

Nous travaillons actuellement sur *Hantaan*, une implémentation de la plupart de nos découvertes en environnement Windows Mobile.

Elle est pour le moment capable d'infecter un smartphone depuis un poste client sans aucune interaction utilisateur et sans alerter les deux parties. Elle ne nécessite aucun redémarrage et est difficilement détectable avec les outils présents par défaut sur Windows Mobile. L'infection se déroule de la manière suivante :

- Attente de connexion d'un smartphone de type Windows Mobile.
- Modification des *Policies* avec *CeProcessConfig()*
- Identification du mobile avec *CeGetVersionEx()*
- Ecriture et exécution de *Hantann* via *CeWriteFile()* et *CeCreateProcess()*

Une fois lancée, *Hantaan* attend une connexion Internet afin d'envoyer toutes les informations sensibles du téléphone au travers d'un tunnel HTTP.

Toutes les données synchronisées avec un serveur Microsoft Exchange (mails, événements du calendrier de l'entreprise, liste des appels, SMS, etc.) sont transférées sur un serveur tiers via WiFi, GPRS ou encore ActiveSync. Elles y sont alors triées et

stockées ensuite dans un serveur de base de données et affichées enfin sur un serveur Web.

De nombreuses fonctionnalités supplémentaires dans *Hantaan* sont envisageables (la liste est non exhaustive) :

- limitation de la consommation du forfait pour davantage de discrétion ;
- vol des historiques de navigation ;
- vol des mots de passe sauvegardés ;
- etc.

Fig. 2. Interface de récupération des informations.

7 Les risques d'un smartphone compromis

En plus du vol d'informations, un smartphone compromis peut comporter des risques pour le système d'information sur lequel il est connecté.

Nos recherches se concentrent sur des procédés d'infection et d'attaque du réseau auquel un smartphone est connecté. En effet, lors de la synchronisation du mobile sur le poste client, la connexion au réseau local est partagée avec le périphérique, ouvrant de nombreuses perspectives au niveau des vecteurs de propagation.

Il devient alors possible d'attaquer les serveurs internes *(Fig. 3)*, les postes clients, voire même de rebondir sur d'autres smartphones. Des interpréteurs de langages de scripts tels que Python ou Ruby existent pour Windows Mobile. Il est donc tout à fait envisageable et sans aucune difficulté, soit d'utiliser des exploits existants, voire des frameworks d'exploitation déjà existants, soit de développer ses propres exploits ou framework rapidement utilisables sur ces plateformes.

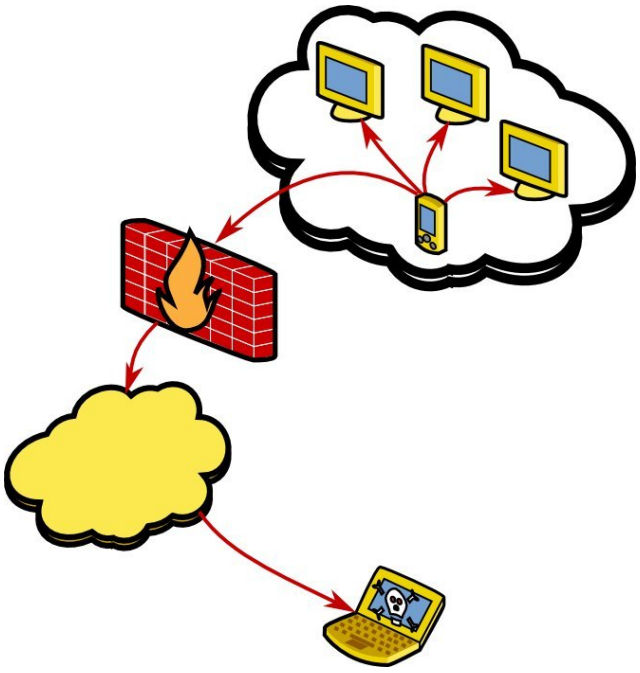

Fig. 3. Risque d'un smartphone compromis.

La boucle est alors bouclée. Un salarié synchronise son smartphone depuis son poste de travail personnel compromis. Son smartphone est alors infecté. Il le synchronise ensuite pendant son temps de travail avec son poste de travail professionnel. Le smartphone compromet ce dernier et lance des attaques sur l'ensemble du système d'information de l'entreprise. Etc. Il est même possible d'imaginer le smartphone servant de passerelle entre le poste d'un attaquant et le réseau de l'entreprise.

8 Conclusion : comment sécuriser son mobile ?

Pour terminer nos travaux, nous avons testé quelques solutions de sécurité côté utilisateur. Les anti-virus testés ont révélé de nombreuses faiblesses. Le plus connu de ces derniers n'a pas été capable de détecter un logiciel espion commercial répandu. Les logiciels de type pare-feu testés peuvent éventuellement prévenir des connexions entrantes malicieuses mais en sont incapables quand un processus illégitime envoie des informations sur le réseau.

Il existe différentes solutions de chiffrement pour les systèmes mobiles, soit par la création d'un conteneur chiffré, soit par des systèmes plus évolués qui permettent de chiffrer de façon transparente les données confidentielles comme les mails ou le carnet d'adresses. Ces solutions ont uniquement pour objectif de prévenir la fuite d'information en cas de perte ou de vol du mobile. Pour une entreprise, des solutions de gestion de flotte existent (*Mobile Device Management*). D'un point de vue sécurité, elles apportent un certain nombre de fonctionnalité :

- inventaire des ressources mobiles de l'entreprise ;
- contrôle des logiciels installés ;
- renforcement du contrôle d'accès au système ;
- sauvegarde automatisée ;
- blocage et effacement à distance du mobile en cas de perte ou de vol ;
- chiffrement des données.

Même si ces solutions peuvent apporter des avantages en termes de sécurité, elles restent malgré tout peu déployées, sont vendues à un prix élevé difficile à rentabiliser pour des flottes de petites tailles et ont souvent des difficultés à gérer une flotte de mobiles possédant des systèmes hétérogènes.

De par la nature du matériel présent dans les smartphones, il est difficile de réaliser des outils performants pour la détection de code malicieux. Le plus important pour éviter au maximum les risques de compromission est d'appliquer les règles de sécurité de base comme par exemple verrouiller automatiquement son mobile, ne pas installer de logiciels provenant de source non sure ou faire attention aux postes de travail sur lequel il est interconnecté.

Références

1. Chaos Communication Congress :
   http://events.ccc.de/congress/
2. CanSecWest 2009 Conference :
   http://cansecwest.com/speakers.html/
3. Répartition de la vente des smartphones par OS :
   http://www.gartner.com/it/page.jsp?id=827912
4. Exploiting symbian :
   www.blackhat.com/presentations/bh-jp-08/bh-jp-08-Mulliner/BlackHat-Japan-08-Mulliner-Hacking-Symbian-OS.pdf
5. Worm.SymbOS.Comwar.a :
   http://www.viruslist.com/fr/viruses/encyclopedia?virusid=75541
6. Worm.SymbOS.Lasco.a :
   http://www.viruslist.com/fr/viruses/encyclopedia?virusid=69735
7. SymbOS/Yxes.A !worm :
   http://www.fortiguardcenter.com/virusency/SymbOS/Yxes.A
8. Blackjacking, 0wning the Enterprise via Blackberry :
   www.praetoriang.net/download/Blackjacking%20-%20Defcon%2014.ppt
9. RedBerry, Advanced Attacks via a Trojaned blackberry :
   www.aurasoftwaresecurity.co.nz/Publications/gn-redberry-kiwicon07.pdf
10. Microsoft ActiveSync information leak and spoofing :
    http://www.derkeiler.com/Mailing-Lists/securityfocus/bugtraq/2005-08/0051.html
11. Backdoor.WinCE.Brador.a :
    http://www.viruslist.com/en/viruslist.html?id=1984055
12. Trojan.WinCE.InfoJack :
    http://www.f-secure.com/v-descs/trojan_wince_infojack.shtml
13. WinCE.PmCryptic.A :
    https://forums2.symantec.com/t5/Mobile-Wireless/A-Smart-Worm-for-a-smartphone-WinCE-PmCryptic-A/ba-p/365445
14. Rootkit Windows CE : https://www.rootkit.com/newsread.php?newsid=899
15. Windows Mobile MMS Exploit : http://www.mulliner.org/pocketpc/
16. Microsoft Windows Mobile Overly Long Bluetooth Device Name Denial of Service Vulnerability :
    http://www.securityfocus.com/bid/31420/discuss
17. Windows CE Autorun :
    http://msdn.microsoft.com/en-us/library/aa454179.aspx
18. Microsoft Windows Mobile OBEX FTP Service Directory Traversal Vulnerability :
    http://www.securityfocus.com/bid/33359
19. Default Security Policy Settings for Windows Mobile-Based Devices :
    http://msdn.microsoft.com/en-us/library/ms889564.aspx