Sécurité du Système Android

Nicolas RUFF
EADS Innovation Works
nicolas.ruff (à) eads.net

Panorama des OS mobiles

- Propriétaire / fermé
  - Souvent réservé aux téléphones d'entrée de gamme
  - Non extensible
    - Pas de SDK public / impossibilité d'installer de nouvelles applications
  - En voie de disparition
- Symbian OS
  - Anciennement majoritaire
  - Mais en voie de disparition également
    - Dernière défection en date: Sony-Ericsson
- iOS
  - Disponible uniquement sur iPhone / iPod / iPad
- BlackBerry OS
  - Disponible uniquement sur BlackBerry
- Windows Mobile
  - Il y a un "avant 7" et un "après 7"
- Autres
  - Ex. webOS, badaOS, MeeGo, …

- Et bien sûr … Android ☺

Panorama des OS mobiles

- Remarques préliminaires
  - Le marché se fragmente de plus en plus
    - Y compris au niveau du *hardware*
  - Tous les systèmes qui n'ont pas de *marketplace* vont probablement disparaitre
  - *Tous* les systèmes cités précédemment ont été attaqués avec succès
    - Plus beau *hack* à ce jour: Adam Gowdiak vs. Nokia 6310i (2002)

Android, pourquoi

- Gratuit
- Open Source / personnalisable
  - http://android.git.kernel.org/
  - Attention: dans un téléphone "physique", tout n'est pas Open Source !
- Technologies maitrisées
  - Linux / Java
- Environnement de développement convivial
  - Eclipse + SDK Google
  - http://developer.android.com/sdk/index.html
- Support de Google
  - Pérenne
  - Techniquement avancé
- Présence d'une *marketplace* dynamique

Android en détail

- Architecture
  - Noyau Linux
  - Librairie C ("Bionic")
  - Java Virtual Machine ("Dalvik")

- Mécanismes de sécurité notables
  - Signature des applications
  - Isolation des applications
    - Uid par application
    - Gid par signataire
  - Plus de 100 permissions attachées à des opérations "sensibles"
    - http://developer.android.com/reference/android/Manifest.permission.html

Les limites d'Android

- A mon avis, voici les principaux risques:

1. Failles logicielles
2. Marché de la téléphonie mobile
3. Applications

1. Failles logicielles

- Cause(s)
  - (Désolé, mais) le code Open Source est truffé de failles
    - … et facile à auditer
  - Quelques exemples célèbres
    - Exploitation d'une faille dans "LibTIFF" (PSP et iPhone)
    - Exploitation d'une faille dans "FreeType" (iPhone)
  - Méfiez-vous du *bullshit*
    - "Le noyau est sûr"
    - … car un outil automatique ne trouve que 359 bogues !
      - http://www.lemagit.fr/article/securite-google-bugs-android-qualite-opensource-code-source/7454/1/android-code-sans-trop-defauts-grace-noyau-linux/
    - Tout est une question d'interprétation des résultats
      - http://linux.slashdot.org/article.pl?sid=10/11/02/2238205

1. Failles logicielles

- Facteurs aggravants
  - Facile à *patcher* … mais très difficile à déployer !
    - La mise à jour automatique à distance de tous les téléphones est impossible
    - … car elle implique le constructeur *et* l'opérateur
    - … et n'apporte aucun revenu, que des ennuis
  - Pourtant il existe des mécanismes de mise à jour
    - Ex. protocole FOTA (*Firmware Over-the-Air*)
  - Aucun produit de sécurité sérieux n'est disponible pour Android
    - A cause du modèle d'isolation des applications (!)

1. Failles logicielles

– La prévalence des versions Android en temps réel
  - http://developer.android.com/resources/dashboard/platform-versions.html

1. Failles logicielles

- Conséquences
  - Faille dans le navigateur
    - Exploitable à distance sur Android 2.0 et 2.1
      - http://www.exploit-db.com/exploits/15423/
      - http://imthezuk.blogspot.com/2010/11/float-parsing-use-after-free.html
  - Elévation de privilèges locale
    - Une variante de la faille "udev" exploitable sur Android < 2.2 (au moins)
      - http://stealth.openwall.net/xSports/exploid.tgz
  - N'oublions pas qu'Android intègre également Flash Player …
      - http://www.appbrain.com/app/flash-player-10-1/com.adobe.flashplayer

2. Marché de la téléphonie

- J'identifie 4 acteurs dans le modèle actuel

- … et aucun de ces acteurs ne travaille pour la sécurité du modèle !

2. Marché de la téléphonie

- 1/ Fabricants de matériel
  - Minimisent le coût de développement et le "*time to market*"
  - Qualité logicielle déplorable
  - Aucun support après-vente
    - La durée de vie "commerciale" d'un téléphone est d'environ 6 mois
  - Fonctions de débogage disponibles en production
    - Ex. shell "root" sur HTC Evo et HTC Hero (port TCP/12345)
      - http://www.unrevoked.com/rootwiki/doku.php/public/unrevoked1_disclosure

2. Marché de la téléphonie

- 2/ Fabricants de logiciel
  - Cherchent à séduire les autres acteurs
  - Simplifient le développement au maximum
    - Ex. Google App Inventor
      - http://appinventor.googlelabs.com/about/
    - Ex. pas de signature avant Symbian 9
      - Et on a pu lire que ça serait la fin des développeurs indépendants !
  - Proposent des fonctions/options "dangereuses"
    - Ex. Windows CE: "Enable Full Kernel Mode"
      - http://msdn.microsoft.com/en-us/library/ms934283.aspx
    - Ex. Android: déverrouillage magique avec le mot de passe "null"
      - http://www.onlineshoppingfree.com/2010/08/24/android-unlock/
      - http://code.google.com/p/android/issues/detail?id=3006
  - … ou débiles ☺
      - http://developer.android.com/reference/android/app/ActivityManager.html#isUserAMonkey%28%29
  - Permettent aux autres acteurs de générer des revenus
    - Qui a dit "même illégaux" ?

2. Marché de la téléphonie

- 3/ Opérateurs de téléphonie
  - Maximisent leurs revenus
  - Personnalisent les logiciels
    - Essentiellement pour brider des fonctions
      - Ex. "tethering", géolocalisation, DRM …
  - Ajoutent des bogues
    - Ex. contournement du verrouillage sur Motorola Droid
      - https://theassurer.com/p/756.html
      - http://techcrunch.com/2010/01/11/verizon-droid-security-bug/
  - Et incitent (indirectement) au piratage par les utilisateurs !

2. Marché de la téléphonie

- 4/ Développeurs d'applications
  - Maximisent leurs revenus
    - Dans un modèle essentiellement basé sur le "gratuit" ou le "*low cost*" (0,99c/application)
  - Se financent par la publicité
    - Lecture indispensable: http://arronla.com/2010/08/android-revenue-advanced-task-manager/
  - Ajoutent des fonctions douteuses (en lien avec la publicité)
    - Ex. accès au carnet d'adresses, géolocalisation indue, …
  - N'hésitent pas à employer des méthodes carrément frauduleuses
    - http://www.01net.com/editorial/520345/gare-aux-pubs-pieges-dans-les-applications-iphone/
    - http://www.f-secure.com/weblog/archives/00002063.html
  - Et ça n'est pas du domaine du FUD !
    - http://french.people.com.cn/VieSociale/7194155.html
- Note: le consommateur final n'est pas partie prenante dans le modèle
  - Hors effet de *buzz*

3. Applications

- La *marketplace* est un composant fermé
  - Géré par l'opérateur
  - Un filtrage "minimal" des applications est mis en œuvre
    - La révocation comme principal outil de sécurité ?
    - Les certificats autosignés sont acceptés
    - Les certificats *doivent* avoir une durée de vie extrêmement longue (au-delà du 22 octobre 2033)
      - http://developer.android.com/guide/publishing/app-signing.html

- Il n'est pas "trivial" de récupérer les applications pour analyse
  - Accès "*root*" au téléphone
  - Mode "*debug*"
  - (Eventuellement) fonction "enregistrer sur la SDCard" (Android 2.2+)

- Ne donne pas accès au contenu téléchargé en continu
  - Ex. publicités

3. Applications

- La *marketplace* n'est pas la seule source d'application
  - N'importe qui peut compiler, signer et diffuser un APK
    - Option du téléphone "autoriser les sources inconnues"
  - Il existe des bases de données d'applications "piratées" ou nécessitant un accès "*root*" au téléphone
    - Note: le piratage d'applications Java est trivial (*bytecode* non obscurci)
      - http://intrepidusgroup.com/insight/2010/10/android-app-decompilation-bake-off/
- La sécurité est basée sur des permissions
  - Déclarées par le développeur
  - Non sélectionnables individuellement par l'utilisateur
  - Qui peuvent changer lors d'une mise à jour

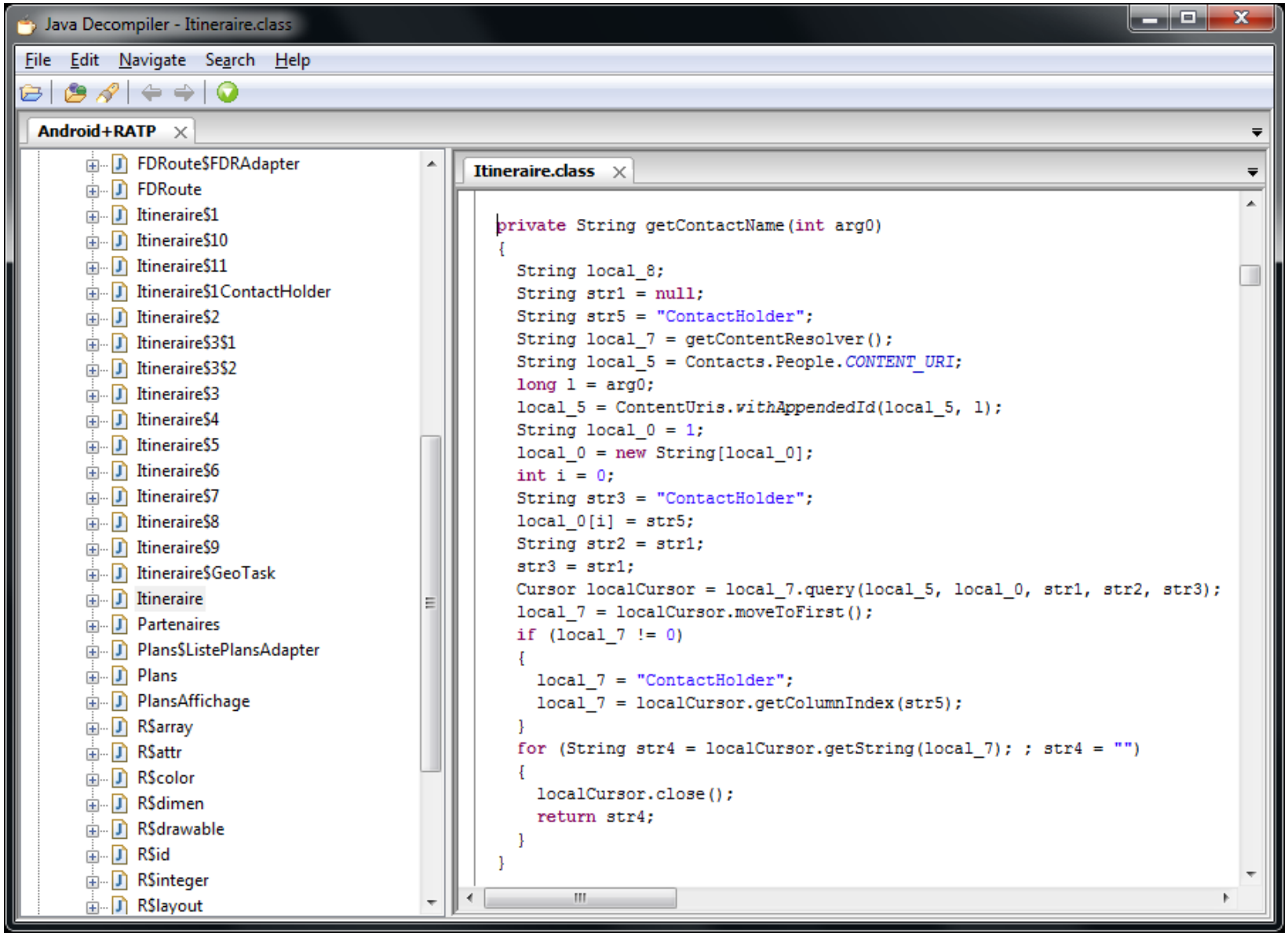
3. Applications

- Cas d'école: l'application "RATP Lite"
  - Alias "com.fabernovel.ratp"
  - Caractéristiques
    - Utilise le logo "officiel" de la RATP
    - Editée par "FaberNovel"
      - A développé deux applications: "RATP Lite" (gratuit) et "RATP Premium" (0,79c)
    - Demande l'accès au carnet d'adresses
  - Vous installez ?

3. Applications

- Etape 1: récupérer l'application
  - Donc la télécharger et l'installer …
- Etape 2: décompiler l'application
  - Outil "baksmali"
  - Ou mieux: "undx"
    - Mais ne fonctionne pas (bien) "*out of the box*"
- Etape 3: chercher les références aux contacts
- Verdict: l'application est "probablement" saine
  - L'accès aux contacts sert à trouver la station la plus proche !
    - … dans cette version de l'application du moins

3. Applications



Conclusion: un virus pour demain ?

- Non
  - Les virus, c'est tellement "années 90"
    - Il n'existera donc pas de virus pour Linux ? ☺
  - Pour survivre, un code malveillant doit rester "en dessous du radar"
    - La propagation automatique est exclue
  - Note: il existe bien quelques vers (comme "Ikee" sur iPhone)
    - Mais ils tiennent plus de la "preuve de concept"

Conclusion: un virus pour demain ?

- Ce qui ~~risque plutôt d'arriver~~ existe déjà: les applications malveillantes
  - Le développement et la distribution d'applications sont faciles
  - Les possibilités d'abus sont nombreuses
    - Accès aux contacts
    - Messages / appels surtaxés
    - Vol de données
    - Espionnage
  - Il n'existe pas d'outil permettant de s'en protéger
    - L'utilisateur final doit faire le travail de vérification (!)
  - L'accès Internet permanent autorise la création d'attaques modulaires
    - Composant malveillant téléchargé et exécuté ultérieurement

Conclusion

- DEMO
- QUESTIONS