Android

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

Android

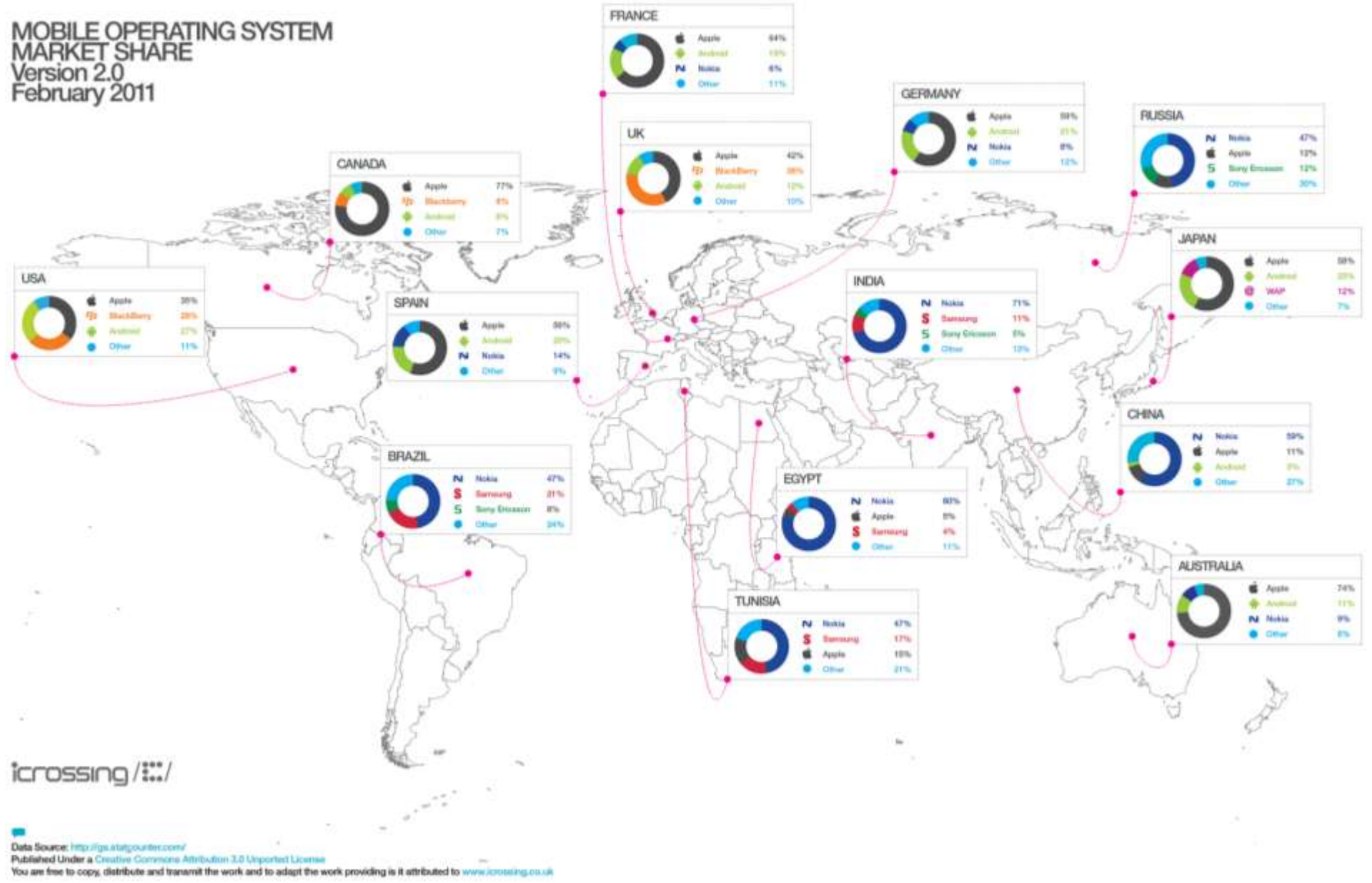
- Contexte technique
  - Un système d'exploitation Open Source
    - … ou presque (ex. drivers)
  - Basé sur Linux 2.6
  - Orienté "smartphones"
    - GUI et modèle de programmation ≠ PC
    - L'essentiel du parc déployé s'exécute sur architecture ARM
  - Extensible par des développement en Java
    - Android Market … ou autres (ex. Amazon)
  - Machine virtuelle Java = "Dalvik VM"
    - Bytecode incompatible avec la JVM ~~Sun~~Oracle

Android

- Contexte historique
  - 2003: création d'Android, Inc.
  - 2005: rachat par Google
  - 2007: sortie du produit
    - Création de l'Open Handset Alliance
  - 2010: le projet Android devient rentable

Android

- Contexte commercial
  - Source: http://www.gartner.com/it/page.jsp?id=1689814
  - Il s'est vendu environ 428 millions de mobiles dans le premier trimestre 2011
    - Dont 23,6% de smartphones
  - Répartition des smartphones vendus par OS
    - Android: 36%
    - Symbian: 27%
    - iOS: 17%
    - RIM: 13%
  - Attention aux chiffres
    - Les marchés sont très régionalisés
    - Les smartphones représentent une minorité des téléphones en circulation



http://paidcontent.org/images/editorial/_original/global-os-marketshare-feb.2011-o.png

Android et la sécurité

- Version officielle
  - Les applications sont signées
    - Un certificat auto-signé suffit
    - Permet la révocation (à distance)
  - Les permissions "sensibles" doivent être acceptées par l'utilisateur à l'installation
    - Atomique et irrévocable
  - Chaque application possède un uid/gid unique
    - Permet d'utiliser le cloisonnement naturel d'Unix
    - Note: la machine Java n'est pas une frontière de sécurité

- Et … c'est tout ?

- Heureusement non !
  - Failles système
  - Erreurs logiques
  - Failles intrinsèques au modèle de sécurité
  - Applications boguées
  - Applications malveillantes
  - Fuites d'information
  - Bonus des constructeurs

Android et l'(in)sécurité

- Failles système
  - Deux cibles de choix
    - Le système (Linux + LibC + ADB + …)
    - Le moteur WebKit
    - Le lecteur Flash embarqué nativement
  - Quelques exemples
    - Linux
      - xSports: Exploid (udev), RageAgainstTheCage (adb), KillingInTheNameOf (ashmem)
    - WebKit
      - http://www.exploit-db.com/exploits/15423/
      - http://www.exploit-db.com/exploits/16974/
      - http://blog.metasploit.com/2011/01/mobile-device-security-and-android-file.html

Android et l'(in)sécurité

- Erreurs logiques
  - Contournement du verrouillage
    - Ex. Motorola Droid
      - http://techcrunch.com/2010/01/11/verizon-droid-security-bug/
  - Backdoor "null"
    - http://code.google.com/p/android/issues/detail?id=3006

Android et l'(in)sécurité

- Failles intrinsèques au modèle de permissions
  - Le modèle a été prouvé en langage Coq
    - (Une faille a d'ailleurs été trouvée …)
      - "A Formal Model to Analyze the Permission Authorization and Enforcement in the Android Framework"
  - Mais …
    - L'utilisateur est le rempart ultime
    - Une application peut créer de nouvelles permissions
      - Ex. com.htc.*, com.sprint.*
    - Chaque application doit vérifier ses entrées
      - Cf. page suivante
    - Attaquer directement https://market.android.com/ ?
      - http://jon.oberheide.org/blog/2011/03/07/how-i-almost-won-pwn2own-via-xss/

Android et l'(in)sécurité

- Applications boguées
  - Mauvaise utilisation des API
    - API = Intent, Content Provider, Broadcast Listener, Service …
    - Ex. Intents ouverts à tous (Skype)
      - http://www.privateerlabs.net/research/whitepapers/PRIVATEERLABS_MALICIOUS_INTENT.pdf
    - Note: un Intent peut contenir un objet Java sérialisé …
    - Ex. "am start −a android.intent.action.CALL tel:1234"
      - http://www.ei.rub.de/media/trust/veroeffentlichungen/2010/11/13/DDSW2010_Privilege_Escalation_Attacks_on_Android.pdf

Android et l'(in)sécurité

- Applications en mode "debug"
  - Ex. journal système: URLs avec paramètres, n° de CB, …
- Rappel: il n'y a aucune permission sur la carte SD
  - (Système FAT)
  - Ex. Skype (à nouveau)
- SQLite … et injections SQL
  - https://media.blackhat.com/bh-ad-10/Nils/Black-Hat-AD-2010-android-sandcastle-slides.pdf
- NDK … et *buffer overflows*

Android et l'(in)sécurité

- Applications malveillantes
  - Très simple
    - Demander très permissions très larges
    - (ou) Embarquer tout ce qu'il faut pour passer "root"
  - Très classique
    - DroidDream, etc.
  - Le plus dur: se diffuser
    - Via une application "hôte"
      - Avec ou sans l'accord de l'éditeur
      - http://www.reddit.com/r/Android/comments/fm3cu/spyware_company_wants_us_to_embed_their_code_into/
    - En republiant une application populaire

Android et l'(in)sécurité

- Fuites d'information
  - Hotspot WiFi : le trou noir
    - Clients Facebook, Twitter …
    - Même les applications Google sont vulnérables !
      - Vol du AuthToken
      - http://www.uni-ulm.de/en/in/mi/staff/koenings/catching-authtokens.html
  - … mais pas seulement
    - Un journal de crash contient beaucoup d'information
      - http://events.ccc.de/congress/2010/Fahrplan/events/4151.en.html

Android et l'(in)sécurité

- Bonus des constructeurs
  - Codes secrets
    - http://www.blackhat.com/presentations/bh-usa-09/BURNS/BHUSA09-Burns-AndroidSurgery-SLIDES.pdf
  - Applications sans support du SSL
    - Mortel sur un WiFi
  - Shell "root" sur le port TCP/12345
    - HTC Evo et HTC Hero commercialisés par Sprint
  - HTCSense.com
  - Téléphones verrouillés et "end of life"
    - … conduisant à des firmwares "non officiels"

Android

- Mais alors … où est la sécurité ?
  - Protection du "/system"
    - Parfois hardware avec les cartes "eMMC"
  - Protection du "/mnt/asec"
    - Applications sur SDCard
  - Signature des firmwares
    - Sauf si le téléphone est en mode "S-OFF"
  - ro.secure & ro.debuggable
  - *Remote kill switch* applicatif (chez Google)
  - Mise à jour "over-the-air" (FOTA)
  - Mise à jour automatique des applications
    - à permissions équivalentes
- … il y a quand même du boulot

Conclusion

from Dave Aitel

subject **[Dailydave] Immunity's Guide to Being Mobile and Secure**

to dailydave <dailydave@lists.immunityinc.com>

Immunity's Guide to Being Mobile and Secure

Choose your OS:
 - Sorry Google-Fans. Android is the least secure mobile phone operating system that you'll actually use - it's accessible and easy to write applications for - and that means less secure.
- - The Blackberry is the least secure mobile phone OS that you won't use (at least, not if you don't have to)
- - Windows Phone 7 is the most secure operating system, partially because no one has ever seen it in the wild. But both the iPhone and WP7 are built from the ground up to restrict what the consumer does with their phone. This makes them "secure" both for you, and for large media companies who want to make money off you.

Choose what you do:
- - Don't ever do internet banking on your phone
- - Don't submit checks to your bank from your phone
- - Don't take naughty pictures on your phone
- - There's no halfway here. You either want someone to take all your money or you don't.

Choose your connection:
- - Stick with 3G if you can, while traveling. WiFi is short for "I like it when other people log into my facebook as me".
- - Buy yourself a local phone when you go out-of-country.

Conclusion

- Android n'est pas un mauvais système
  - Mais il y a trop de gens qui cherchent à gagner de l'argent avec pour qu'il reste sûr bien longtemps
- Nous sommes revenus dans les années 1990
  - Mais avec des attaques de 2011
- @Consultants
  - Il est temps de s'y mettre avant qu'il ne soit trop tard !