

by Sergio Arcos Sebastián



Destripando y protegiendo aplicaciones Android

Índice



- **Introducción**
- Marco de seguridad en el diseño de Android
- App1 – Difamación
- App2 – Invasión
- App3 – Recopilación
- App4 – Suplantación
- Consejos y conclusiones

Disclaimer



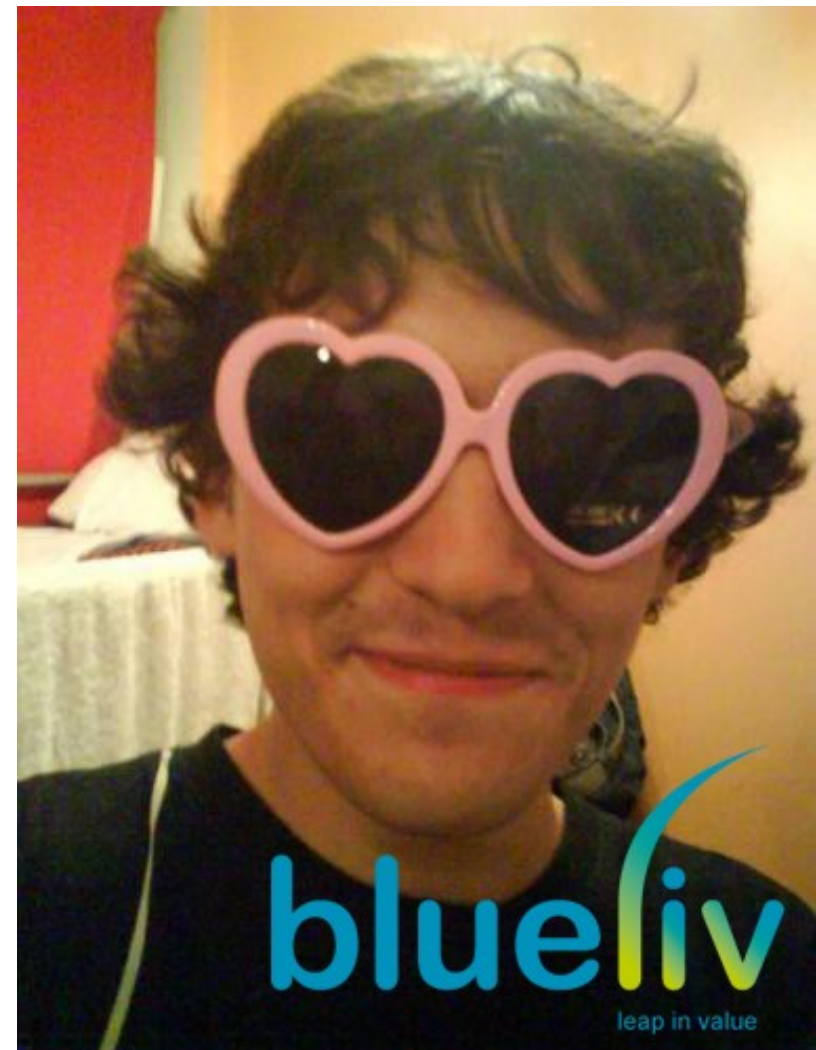
- 1) No aprendáis de mis aplicaciones...
... vosotros programáis mejor :)
- 2) No sé si durará 1 hora o 3 horas...
... ¡ups!
- 3) No sé si será Master.class...
... pero esperemos que sí sea divertida



¿Quién soy?



- Poca memoria
- Autodidácta
- Consultor de seguridad informática en www.blueliv.com
- Miembro del equipo técnico de www.pirata.cat



Mi primera APP

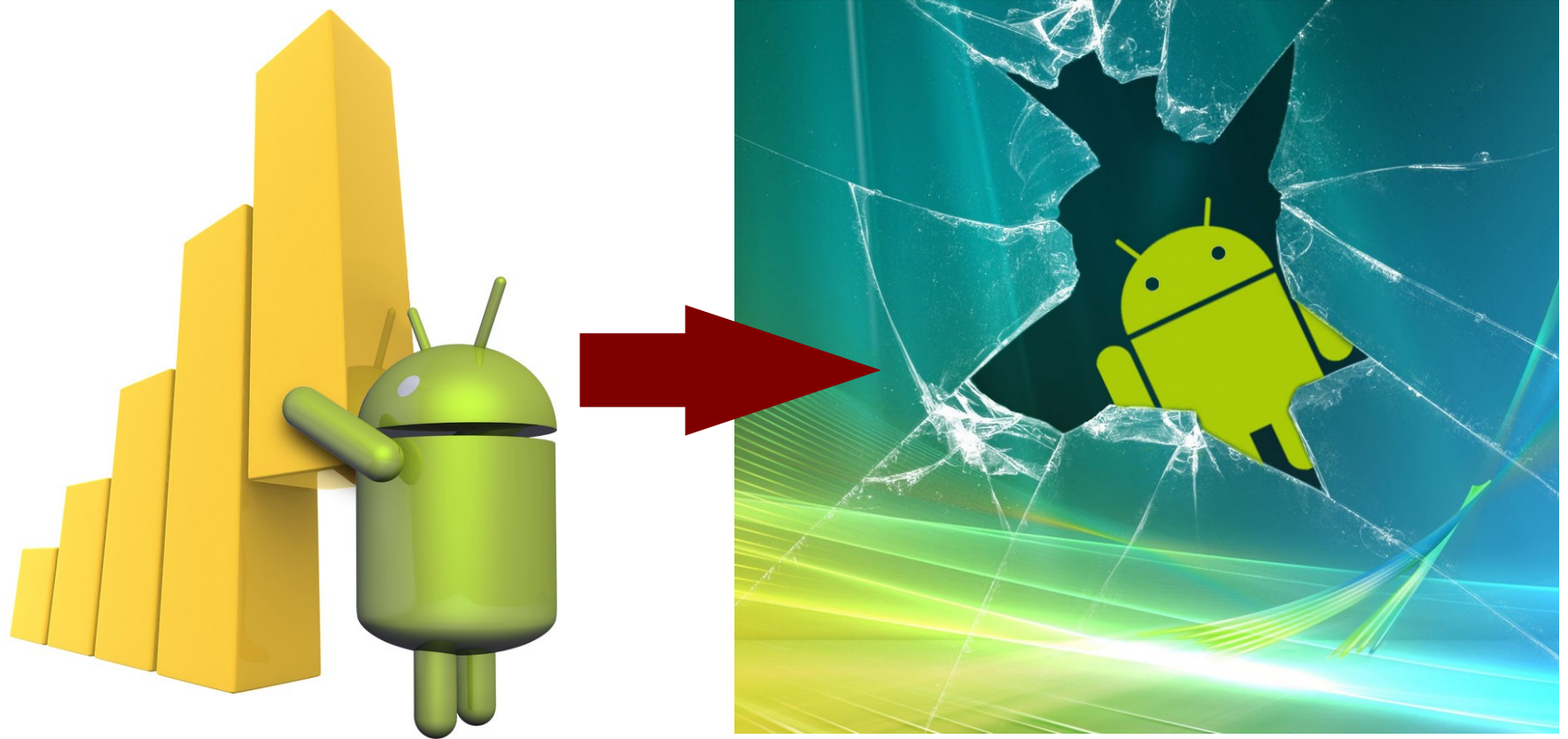


- Lector RSS, Google Calendar, Sistema de participación, ...
- Permite autenticarse y votar
- Actualmente v2.3
- Falta mejorar tanto su **usabilidad** como su **seguridad**

(pname:cat.pirata.activities)



¿Porqué esta presentación?



“Se aprende enseñando”

¿Qué es la seguridad?



"Algo es seguro hasta que deja de serlo"

(Definición cutre pero real)

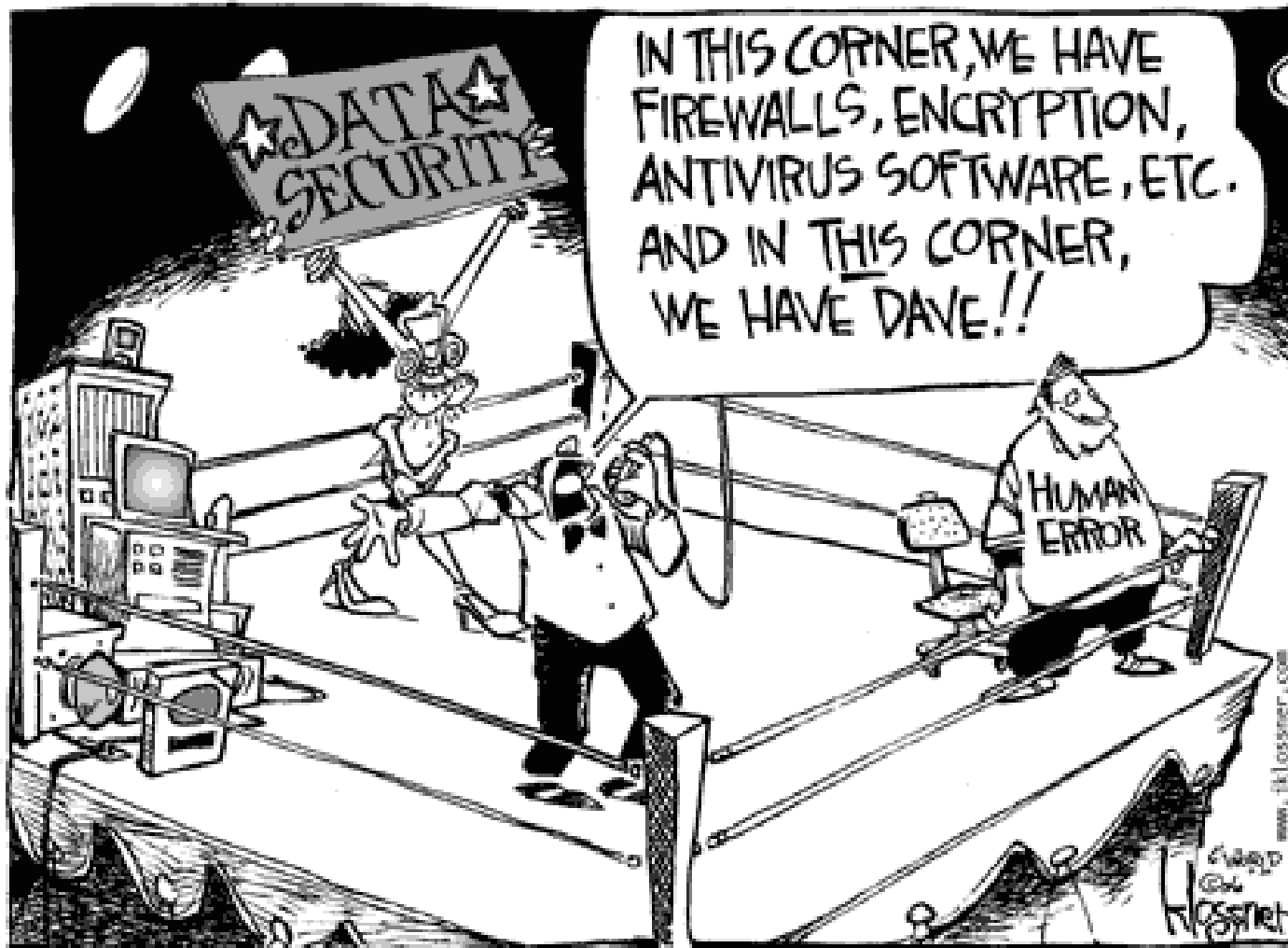
Son succulentos. ¿Por qué?



- Bastante potencia de CPU
- Muchas cuentas de muchas plataformas juntas
- Conexión permanente 3G
- Funciones de llamada y SMS
- Poco tiempo en el mercado
- Baja frecuencia de actualización

... pero ninguno de estos es el mayor problema...

Esta autoadministrado :(



¿Qué podemos controlar?



- Legitimidad del software (fiabilidad)
- Información y transparencia (confianza)
- Comunicaciones seguras (protección)
- Datos protegidos (privacidad)
- Actualizaciones constantes (soporte)
- Buenas prácticas de uso (consciencia)

Objetivo



mitigar posibles futuros daños elevando el
numero de capas de protección
implementadas en el sistema/**aplicaciones**

¿Cómo?

Capa oculta

(Interacción Humano-Computador Segura, ¡te invocó!)

Nota informativa



Como mis conocimientos de la *Interacción Humano-Computador Segura* son bastante introductorios, usaremos otro método:

A palos. (lease: a base de precedentes)

"La letra con sangre entra", Goya



Índice



- Introducción
- **Marco de seguridad en el diseño de Android**
- App1 – Difamación
- App2 – Invasión
- App3 – Recopilación
- App4 – Suplantación
- Consejos y conclusiones

Hardware ~ ARMvX



- Instrucciones RISC 32 bits, Bi-Endian
- Uso de **Registros**
- ARM-Thumb

```
payload = "" +  
    "\\x01\\x30\\x8f\\xe2" +  
    "\\x13\\xff\\x2f\\xe1" +  
    "\\x78\\x46\\x0c\\x30" +  
    "\\xc0\\x46\\x01\\x90" +  
    "\\x49\\x1a\\x92\\x1a" +  
    "\\x0b\\x27\\x01\\xdf" + cmd
```

- Syscalls = x86
- Modulos para

Metasploit

<http://dev.metasploit.com/redmine/attachments/459/exec.rb>



Hardware ~ Memoria



Interna:

- ROM & RAM
- FileSystem: **YAFFS**
- Varias particiones
- # df -h
- /dev/block/dm-X
-> /mnt/**asec**/App/

Externa:

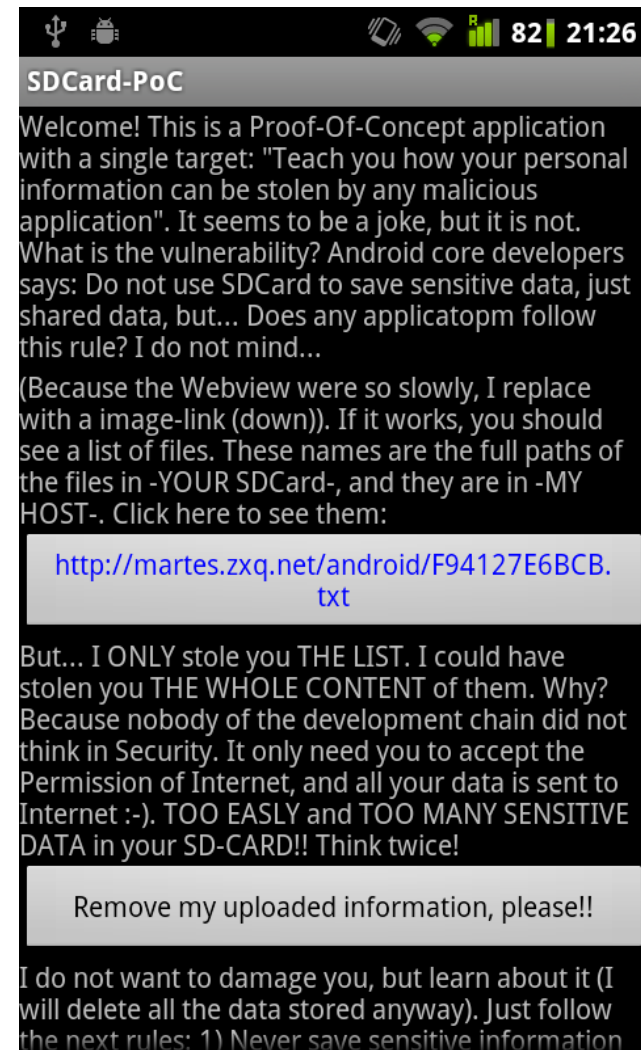
- SDCard – NAND
- FileSystem: **FAT32**
(por defecto [mkcard])
- "Única" partición
- # fdisk -l
/dev/block/mmcblk0
- Parecido a **/tmp (-t)**
(sticky bit)



Hardware ~ Memoria – App 1



- Espacio **compartido**
- **Leer/ejecutar** no requiere permisos
- **Escribir** requiere *android.permission.WRITE_EXTERNAL_STORAGE*
- Poco espacio en memoria interna
- Común y **natural!!**



Woops!.. [Never happened]

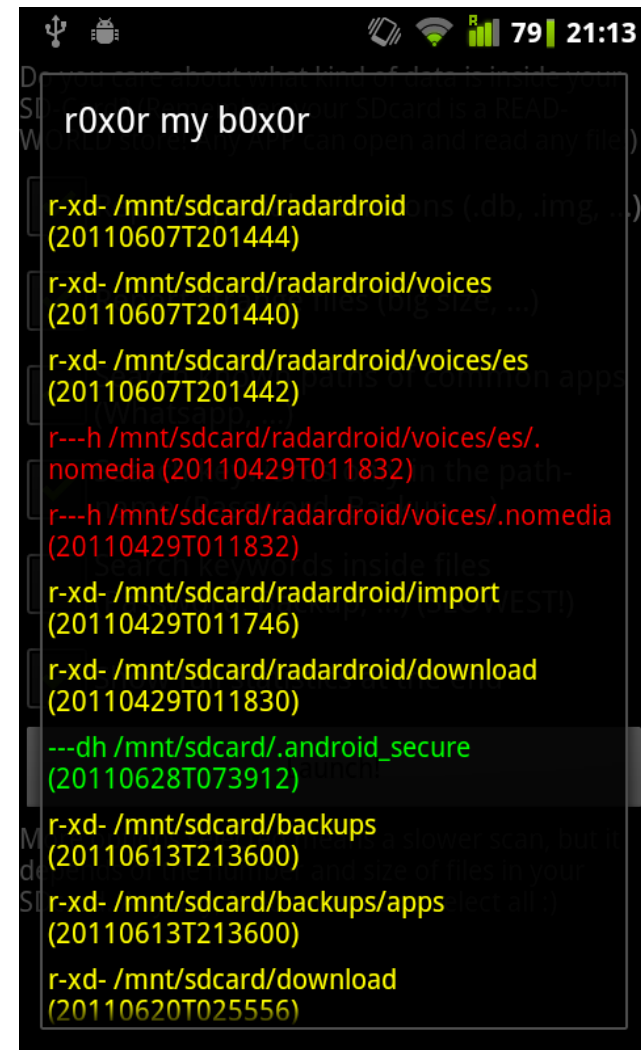
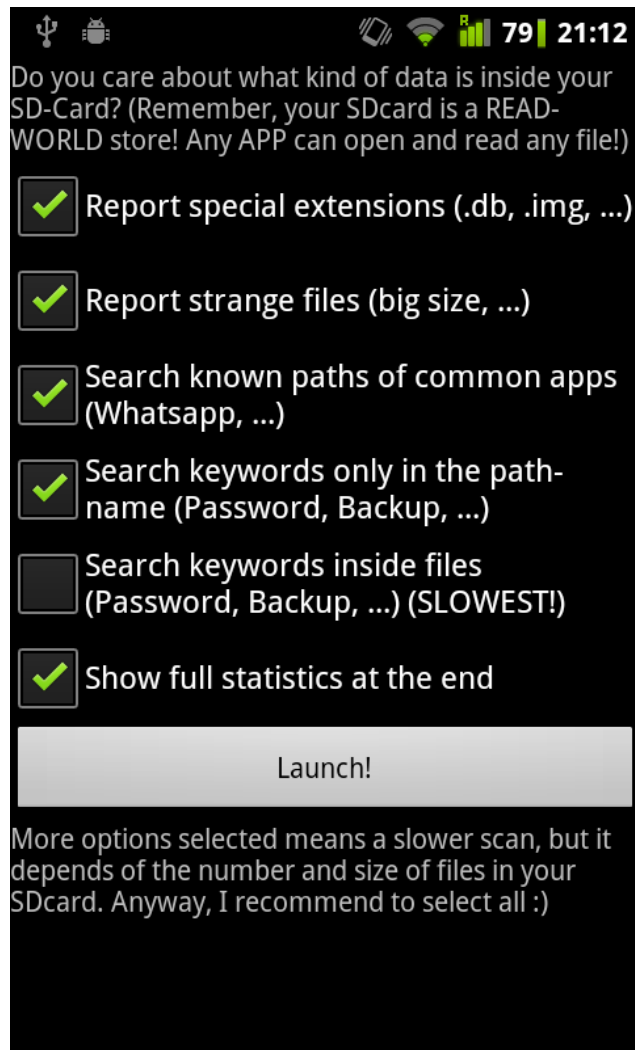


Total de ficheros	72071
Total de "voluntarios"	59
Total de extensiones diferentes	300
Archivos en la carpeta de la camara de fotos	2297
Imagenes de backups (.img)	16
Ofimatica (docx, doc, ppt, xls, pdf)	249
Ficheros APK	345
Nombres de correos (+logs)	67
Palabra "backup"	290
Bases de datos	92

```
46444 download/LiveJasmin_Android_v1016.apk
46445 download/LulzSec Delivers - Copy (2).txt
46446 download/LulzSec Delivers - Copy.txt
46447 download/META-INF
```



Hardware ~ Memoria – App 2



Hardware ~ Comunicación



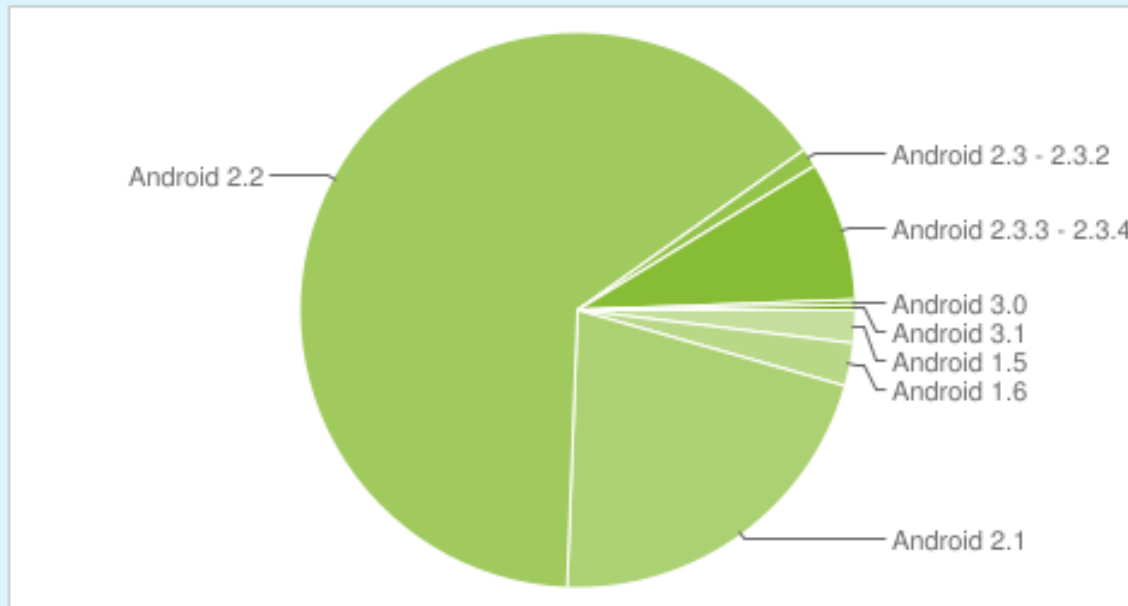
- Mini-USB
- Wireless
- Bluetooth
- SDCard
- Tarjeta SIM
- Pantalla
- Otros (infrarojos, ...)

Android ~ Estadísticas 1



- **Mayo 2009** – 30,000 por día
 - **Mayo 2010** – 100,000 por día
 - **Agosto 2010** – 200,000 por día
 - **Diciembre 2010** – 300,000 por día
 - **Mayo 2011** – 400,000 por día
 - **Junio 2011** – 500,000 por día
-
- 4.500.000.000 aplicaciones descargadas

Android ~ Estadísticas 2



Platform	API Level	Distribution
Android 1.5	3	1.9%
Android 1.6	4	2.5%
Android 2.1	7	21.2%
Android 2.2	8	64.6%
Android 2.3 - Android 2.3.2	9	1.1%
Android 2.3.3 - Android 2.3.4	10	8.1%
Android 3.0	11	0.3%
Android 3.1	12	0.3%

Data collected during a 14-day period ending on June 1, 2011



A graphic logo for GTUG Barcelona. It features four large, colorful spheres (blue, red, yellow, green) with the letters 'G', 'T', 'U', and 'G' respectively. Behind the spheres are stylized line drawings of Barcelona landmarks: the Guggenheim Museum, the Sagrada Família, and the Montjuïc Castle. The word 'BARCELONA' is written in bold black letters below the spheres, with colorful paint splatters around the base.



Android ~ basado en Linux 2



Android ~ Google



- ro.secure = {0,1}
- El Market es **\$deity**
- Base de datos con las **firmas**
- Reporte de infracciones
- Permite instalar apps a través de su web
- Mejora las funciones (wipe, encriptar, ...)
- Posibles condiciones a los proveedores.



Android ~ Particiones



- /boot -> bootloader + kernel
- /system -> sistema operativo + apps (de sistema)
- /recovery -> boot alternativo
- /data -> **datos del usuario** + apps
- /cache -> datos reaprovechables
- /misc -> configuraciones (hardware)
- /sdcard -> Environment.getExternalStorageDirectory();
- /sd-ext -> APP2SD+ / data2ext

¿y /var/log?
¿y encriptar?

Android ~ VirtualMachine



- Todo corre en diferentes **Sandbox** (falsooo!)
- Sistema basado en **PRIVILEGIOS**
 - A) Ejecutamos **código nativo** (JNI)
 - Se hizo una charla GTUG de esto :) (buscad NDK)
 - B) Ejecutamos **código DalvikEXecutable (DEX)**
 - Similar a Java; optimizado y reducido para perifericos
- **Protecciones** de datos **entre** diferentes **instancias** (ej: navegador->Webview)
- **Zygote**: sistema de padres-hijos

Android ~ Usuarios



¿mono-usuario por ser mono-propietario?

```
CA: Símbolo del sistema - adb shell
#
# pwd
pwd
/data/data
# ls -la | tail -n 14
ls -la | tail -n 14
drwxr-x--x    1 app_95    app_95          2048 Jun
drwxr-x--x    1 app_84    app_84          2048 Jun
drwxr-x--x    1 app_71    app_71          2048 Jun
drwxr-x--x    1 app_70    app_70          2048 Jun
drwxr-x--x    1 app_72    app_72          2048 Jun
drwxr-x--x    1 app_73    app_73          2048 Jun
drwxr-x--x    1 app_54    app_54          2048 Apr
drwxr-x--x    1 app_86    app_86          2048 Jun
drwxr-x--x    1 app_93    app_93          2048 Jun
drwxr-x--x    1 app_82    app_82          2048 Jun

drwxr-x--x    1 app_37    app_37          2048 Apr
ager
drwxr-x--x    1 app_85    app_85          2048 Jun

drwxr-x--x    1 app_103   app_103         2048 Jun
drwxr-x--x    1 app_104   app_104         2048 Jun
#
```

```
CA: Símbolo del sistema - adb shell
#
# pwd
pwd
/mnt/sdcard
# ls -la
ls -la
d---rwxr-x    12 system    sdcard_r        4096 Jun
drwxrwxr-x     6 root      system           0 Jun
d-----      2 root      root            40 Jun
d---rwxr-x     3 system    sdcard_r        4096 Jun
d---rwxr-x     5 system    sdcard_r        4096 Jun
d---rwxr-x     2 system    sdcard_r        4096 Jun
d---rwxr-x     5 system    sdcard_r        4096 Jun
d---rwxr-x     3 system    sdcard_r        4096 Jun
d---rwxr-x     3 system    sdcard_r        4096 Jun
d---rwxr-x     2 system    sdcard_r        4096 Jun
---rwxr-x     1 system    sdcard_r        2934 Jun
---rwxr-x     1 system    sdcard_r         112 Jun
---rwxr-x     1 system    sdcard_r         539 Jun
d---rwxr-x     5 system    sdcard_r        4096 Jun
d---rwxr-x     3 system    sdcard_r        4096 Jun
---rwxr-x     1 system    sdcard_r     1169998 Jun
#
```

Android ~ APK 1



- Usando **adb**

> adb install /path/pkg.apk

- En la **shell**

\$ am start -a android.intent.action.VIEW -d file:///path/pkg.apk -t text
-n com.android.packageinstaller/.PackageInstallerActivity

- Con el **navegador** iendo a <file:///path/pkg.apk>
- Con la cuenta de Google desde Google Market

Android ~ APK 2



- **Mover** a /system/app
- **Solapar** una app ya existente /data/app
 - El sistema comprueba firmas en 2 ocasiones:
 - Instalando una aplicación
 - Reiniciando el sistema
- **# pm install -t /path/pkg.apk**
- Usando un paquete `update.zip` (firmado) a través del modo de arranque **/recovery**

Android ~ APK 3



- Necesitan estar **firmadas**
(Modo **debug** o no)
- APK = ZIP + jarsigner
+ align
- Objetivo: Control
absoluto de la app

```
// INSTALAR
String fileName =
    Environment.getExternalStorageDirectory() +
        "/myApp.apk";
Intent intent =
    new Intent(Intent.ACTION_VIEW);
File file =
    new File(fileName);
intent.setDataAndType(Uri.fromFile(file),
    "application/vnd.android.package-archive");
startActivity(intent);

// DESINSTALAR
Uri packageURI =
    Uri.parse("package:com.android.myapp");
Intent uninstallIntent = new Intent(
    Intent.ACTION_DELETE,
    packageURI);
startActivity(uninstallIntent);


// ¿INSTALABLE?
int result =
    Settings.Secure.getInt(
        getContentResolver(),
        Settings.Secure.INSTALL_NON_MARKET_APPS,
        0);
if (result == 0) {
    Intent intent = new Intent();
    intent.setAction(
        Settings.ACTION_APPLICATION_SETTINGS);
    startActivity(intent);
}
```



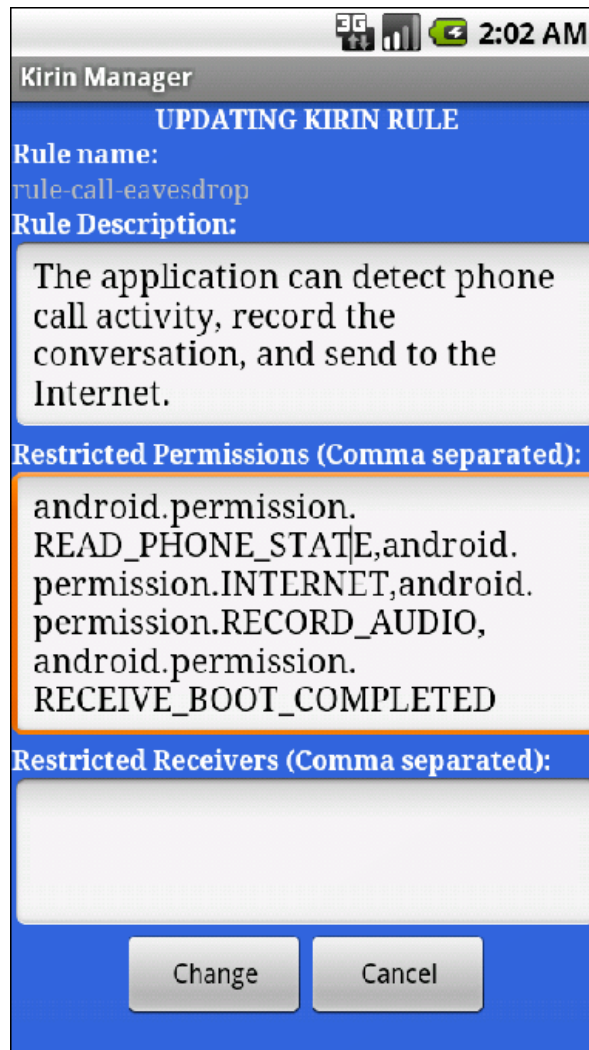
Android ~ Permisos 1



- Regular, peligroso, sistema y firmados (116)
 - Consultar **Settings.Secure** (`import android.provider.Settings;`)
 - Combinaciones más que peligrosas. Ej:
PHONE_STATE + RECORD_AUDIO + INTERNET
 - Filosofía: **"O todo o nada"**
 - 06-29 21:33:38.540: WARN/InputDispatcher(162): Permission denied: injecting event from pid 13267 uid 10122 to window with input channel 40a21648
com.android.packageinstaller/com.android.packageinstaller.PackageInstallerActivity (server)
owned by uid 10046

(Error obtenido AÚN habiendo asignado el permiso `INJECT_EVENT`. ¡Mola!)
 - Existe el permiso "`BRICK`" (9 apps en el market). ¡No mola!
- Destripando y protegiendo aplicaciones Android ~ 

Android ~ Permisos 2



AppFence can enable new interfaces that give users control over the use of their info.



Android ~ Cuentas

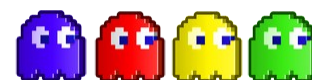


```
Símbolo del sistema - adb shell
# sqlite3 -line accounts.db "SELECT * FROM accounts"
sqlite3 -line accounts.db "SELECT * FROM accounts"          Ruta: /data/system/
  _id = 1
  name = sergio.arcos@gmail.com
  type = com.google
password = AF...Lwj5yu63qS3...n9Q-QVGC...
cidw6WuiRlKM8...p_Qvwe9wA7L...qFQ44Fbh...
yhrc_49VBTcg0...lYqZA==

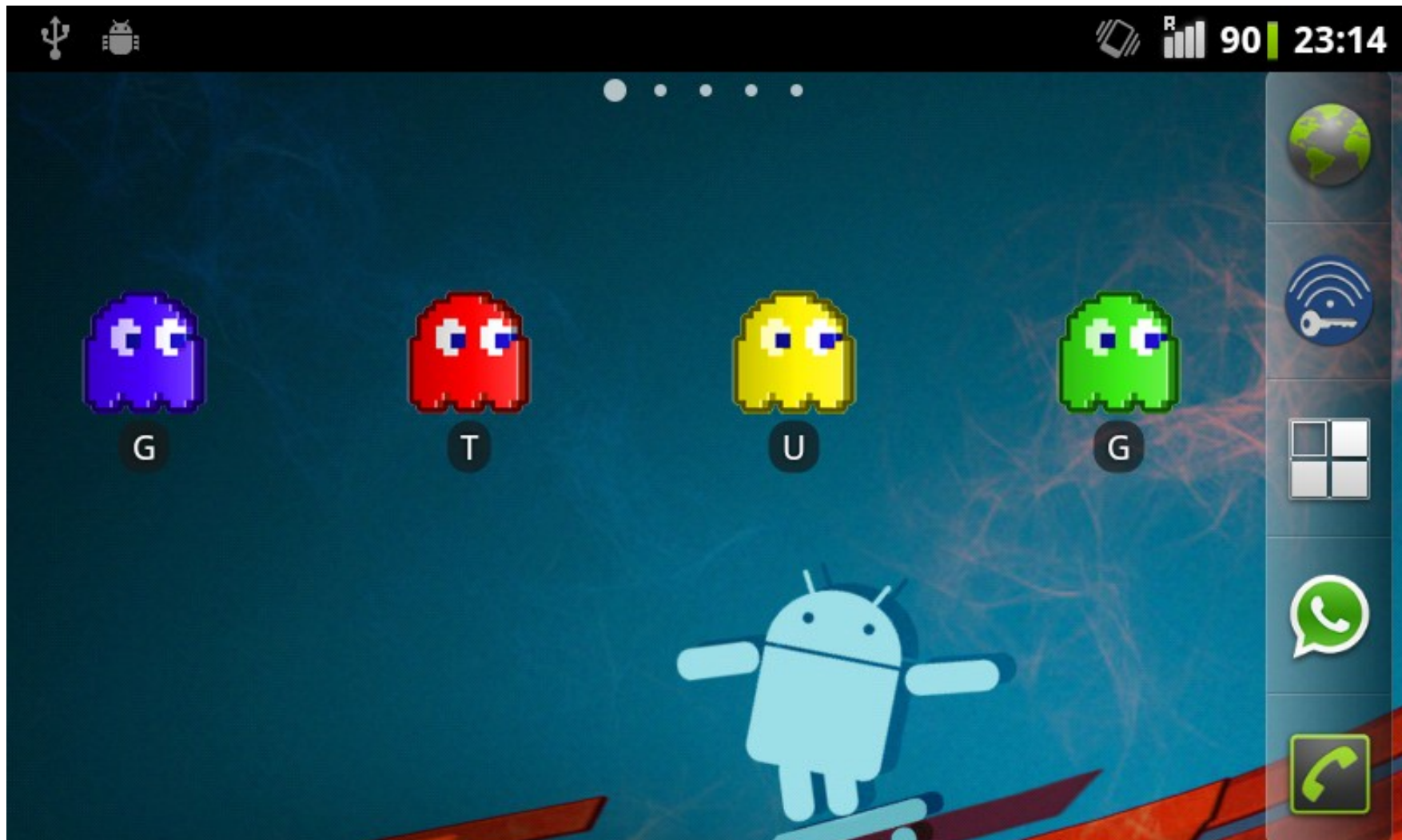
  _id = 2
  name = martes_trece
  type = com.twitter.android.auth.login
password =

  _id = 6
  name = sergio.arcos@gmail.com
  type = com.linkedin.android
password =
  ← Con un "# grep password accounts.db" podemos localizar
  otros passwords... (yo saque el mio de LinkedIn)

  _id = 12
  name = 346...165
  type = com.whatsapp
password =
```



Manos a la obra!!



Destripando y protegiendo aplicaciones Android ~ 

Índice



- Introducción
- Marco de seguridad en el diseño de Android
- **App1 – Difamación**
- App2 – Invasión
- App3 – Recopilación
- App4 – Suplantación
- Consejos y conclusiones

Objetivos

- Aplicación común (Webview) sin ofuscar
- Análisis del APK
- Apktool, dex2jar, dexdump
- Código dinámico
- Análisis del DroidDread



Bienvenidos al Barcelona Google Technology User Group

El Barcelona GTUG está formado por un grupo de desarrolladores sin ánimo de lucro con ganas de realizar reuniones periódicas con el fin de difundir las tecnologías Google. Un foro de intercambio de conocimientos donde se realizan sesiones técnicas, charlas, mesas redondas... y donde además después de las reuniones podemos irnos a pasear por la bonita ciudad de Barcelona y así conocernos mejor.



Índice



- Introducción
- Marco de seguridad en el diseño de Android
- App1 – Difamación
- **App2 – Invasión**
- App3 – Recopilación
- App4 – Suplantación
- Consejos y conclusiones

Objetivos



- Aplicación común (Webview) ofuscada con Proguard
- Uso de AndroGuard
- Intrusión web
- Retrace de Proguard
- Keystore y las claves seguras



tecnologías Google. Un foro de intercambio de conocimientos donde se realizan sesiones técnicas, charlas, mesas redondas... y donde además después de las reuniones podemos irnos a pasear por la bonita ciudad de Barcelona y así conocernos mejor.

La participación está abierta a novatos, desarrolladores, managers y organizaciones que estén interesadas en las tecnologías de Google o que las usan como parte de sus proyectos.

Mantente informado de las reuniones y de todo lo relacionado con el grupo en:

contacto: israel@gtugs.org

Holaaaa

Esto es un comentario



Índice



- Introducción
- Marco de seguridad en el diseño de Android
- App1 – Difamación
- App2 – Invasión
- **App3 – Recopilación**
- App4 – Suplantación
- Consejos y conclusiones

Objetivos



- Aplicación nativa con autenticación SSL mediante webservice
- Ataque del Hotspot
- Introducción al sniffing vía wireless
- AndroidAuditTools
- WhisperSys tools



Índice

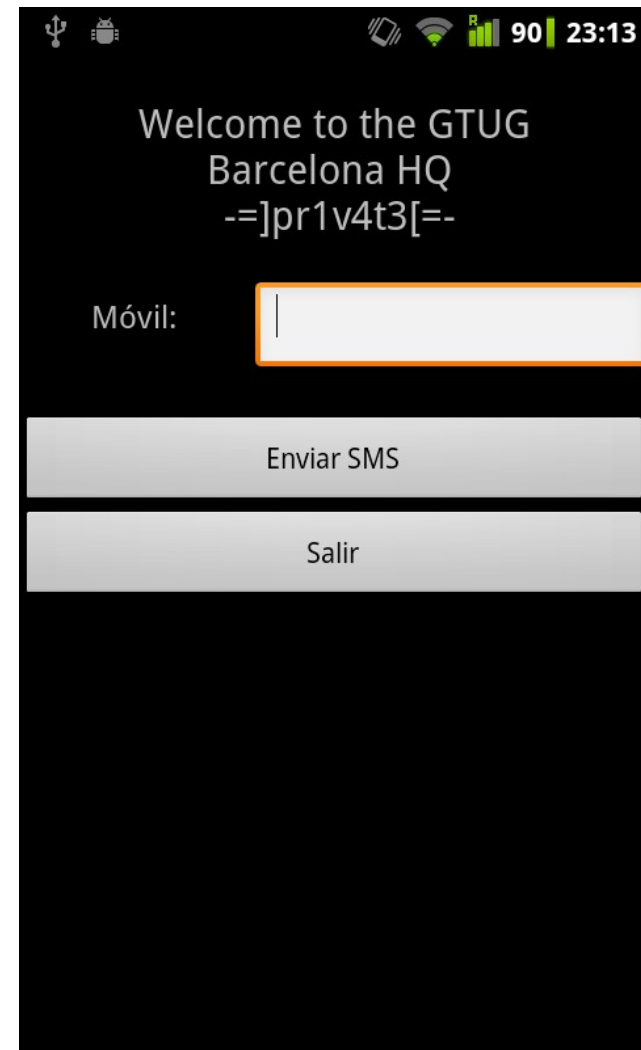


- Introducción
- Marco de seguridad en el diseño de Android
- App1 – Difamación
- App2 – Invasión
- App3 – Recopilación
- **App4 – Suplantación**
- Consejos y conclusiones

Objetivos



- Autenticación por lista blanca de móviles
- Técnicas anti-debug
- Introducción al emulador como herramienta de reversing
- DroidBox & TaintDroid

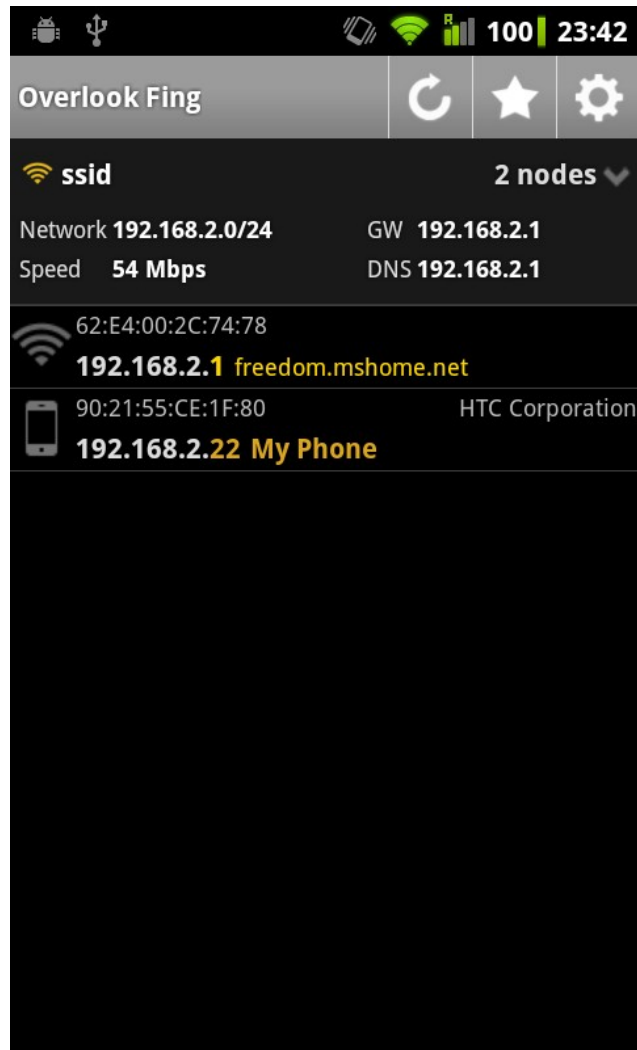


Índice



- Introducción
- Marco de seguridad en el diseño de Android
- App1 – Difamación
- App2 – Invasión
- App3 – Recopilación
- App4 – Suplantación
- **Consejos y conclusiones**

Derivados...



Consejos



- Guarda muy bien los ficheros:
 - Keystore de firmar
 - Mapping.txt
- Recuerda:
 - Mínimos permisos
 - SSL verificado
 - Datos en la Sdcard
 - Tenla actualizada

**Sois el
objetivo
número 1
de las
mafias de
malware**



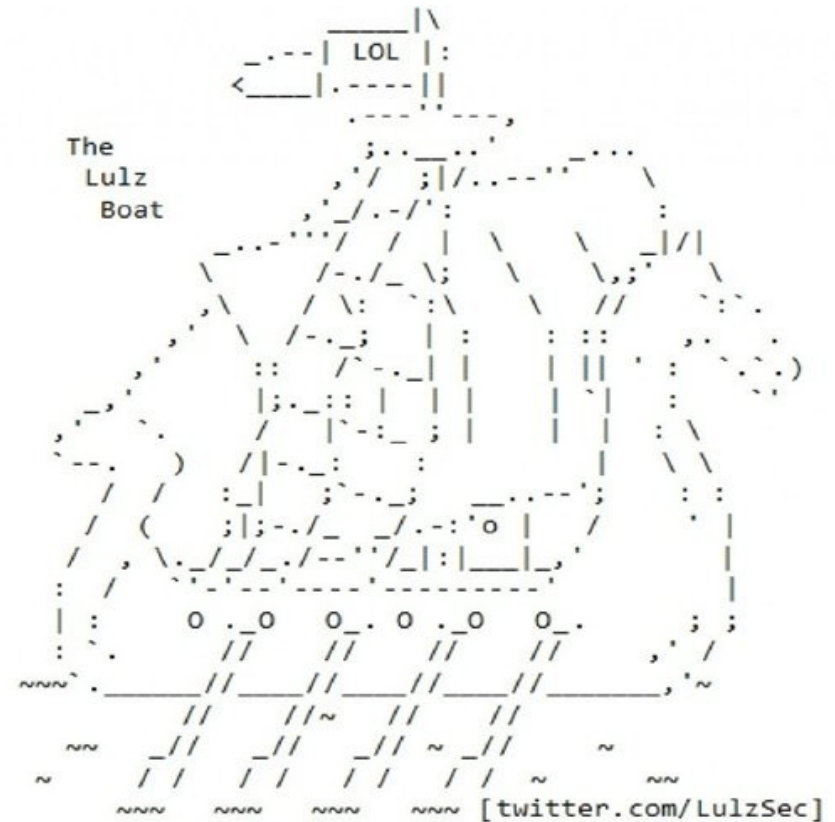
Conclusiones



ANONYMOUS



NEVER FORGIVE



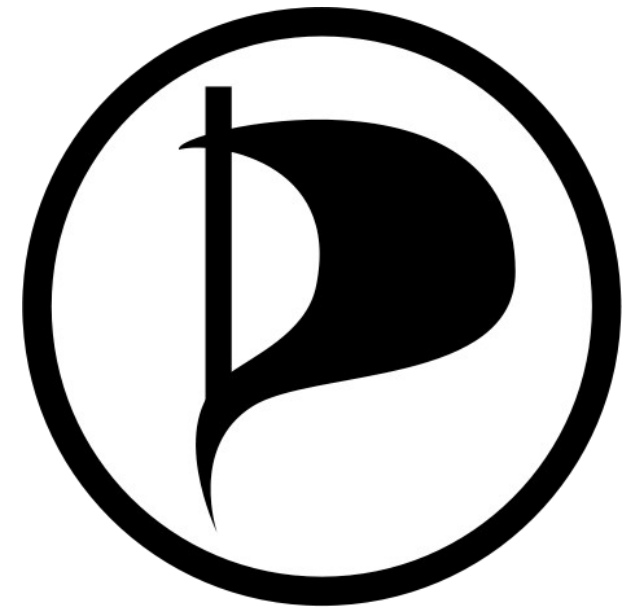
Retos interesantes



```
m13@ubuntu:/dfrws$ file mtblock*
mtblock0.img: VMS Alpha executable
mtblock1.img: DOS executable (device driver)
mtblock3.img: data
mtblock4.img: VMS Alpha executable
mtblock5.img: VMS Alpha executable
mtblock6.img: VMS Alpha executable
mtblock7.img: DOS executable (device driver)
m13@ubuntu:/dfrws$ unyaffs mtblock0.img
broken image file
m13@ubuntu:/dfrws$ unyaffs mtblock1.img
broken image file
m13@ubuntu:/dfrws$ unyaffs mtblock3.img
broken image file
m13@ubuntu:/dfrws$ unyaffs mtblock4.img
Segmentation fault
m13@ubuntu:/dfrws$ unyaffs mtblock5.img
broken image file
m13@ubuntu:/dfrws$ unyaffs mtblock6.img
broken image file
m13@ubuntu:/dfrws$ unyaffs mtblock7.img
broken image file
m13@ubuntu:/dfrws$ █
http://www.dfrws.org/2011/challenge/
```



Preguntas, dudas,
comentarios < /dev/random



Sergio Arcos Sebastián