

Práctica 1 - Introducción a APKs en Android

Jaime García González

Índice general

1. Introducción	2
2. Desarrollo de una aplicación Android	3
3. Estudio de kontaktos.apk	4
3.1. Verificación de firma	4
3.2. Firmas y certificados	6

Capítulo 1

Introducción

Capítulo 2

Desarrollo de una aplicación Android

Capítulo 3

Estudio de kontaktos.apk

3.1. Verificación de firma

En este apartado se procederá a verificar la firma de la aplicación *kontaktos.apk*, para ello, se utilizará la herramienta *apktool*. Se descargarán las herramientas necesarias, la aplicación a analizar y se moverán al mismo directorio. Los pasos seguidos han sido:

1. Ejecución de la aplicación, se generarán los archivos necesarios en un directorio temporal. Véase la figura 3.1.

```
D:\Jaime\Programas\Apktool>java -jar apktool.jar if kontaktos.apk
I: Using Apktool 2.3.4 on kontaktos.apk
S: WARNING: Could not write to (C:\Users\Jaime\AppData\Local\apktool\framework), using C:\Users\Jaime\AppData\Local\Temp\ instead...
S: Please be aware this is a volatile directory and frameworks could go missing, please utilize --frame-path if the default storage directory is unavailable
I: Framework installed to: C:\Users\Jaime\AppData\Local\Temp\127.apk
```

Figura 3.1: Ejecución de apktool para kontaktos.apk

2. Se procede a decompilar la aplicación para su estudio. Véase la figura 3.2.

```
D:\Jaime\Programas\Apktool>java -jar apktool.jar d kontaktos.apk
I: Using Apktool 2.3.4 on kontaktos.apk
I: Loading resource table...
I: Decoding AndroidManifest.xml with resources...
S: WARNING: Could not write to (C:\Users\Jaime\AppData\Local\apktool\framework), using C:\Users\Jaime\AppData\Local\Temp\ instead...
S: Please be aware this is a volatile directory and frameworks could go missing, please utilize --frame-path if the default storage directory is unavailable
I: Loading resource table from file: C:\Users\Jaime\AppData\Local\Temp\1.apk
I: Regular manifest package...
I: Decoding file-resources...
I: Decoding values */* XMLs...
I: Baksmaling classes.dex...
I: Copying assets and libs...
I: Copying unknown files...
I: Copying original files...
```

Figura 3.2: Decompilación de kontaktos.apk

3. Se generará automáticamente un directorio que contiene la aplicación decompilada. Véase la figura 3.3.





 kontaktos	06/02/2019 9:44	Carpeta de archivos	
 apktool	06/02/2019 9:30	Archivo por lotes ...	1 KB
 apktool	06/02/2019 9:27	Executable Jar File	10.746 KB
 kontaktos.apk	06/02/2019 9:33	Archivo APK	8.206 KB

Figura 3.3: Aplicación decompilada

4. Abrimos el fichero *AndroidManifest.xml* para comprobar los permisos. Véase la figura 3.4.

```
<?xml version="1.0" encoding="utf-8" standalone="no"?><manifest xmlns:android="http://schemas.android.com/apk
<uses-permission android:name="android.permission.INTERNET"/>
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"/>
<uses-permission android:name="android.permission.CHANGE_NETWORK_STATE"/>
<uses-permission android:name="android.permission.READ_CONTACTS"/>
<uses-permission android:name="android.permission.WRITE_CONTACTS"/>
<uses-permission android:name="android.permission.CALL_PHONE"/>
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
<uses-permission android:name="android.permission.READ_PHONE_STATE"/>
<uses-permission android:name="android.permission.GET_ACCOUNTS"/>
<uses-permission android:name="android.permission.MANAGE_ACCOUNTS"/>
<uses-permission android:name="android.permission.AUTHENTICATE_ACCOUNTS"/>
<uses-permission android:name="android.permission.READ_SYNC_SETTINGS"/>
<uses-permission android:name="android.permission.WRITE_SYNC_SETTINGS"/>
<uses-permission android:name="android.permission.ACCESS_WIFI_STATE"/>
<uses-permission android:name="android.permission.BATTERY_STATS"/>
<uses-permission android:name="android.permission.RECEIVE_BOOT_COMPLETED"/>
<uses-permission android:name="android.permission.WAKE_LOCK"/>
<uses-permission android:name="android.permission.MODIFY_PHONE_STATE"/>
<uses-permission android:name="android.permission.WRITE_APN_SETTINGS"/>
<uses-permission android:name="android.permission.READ_APN_SETTINGS"/>
<uses-permission android:name="android.permission.VIBRATE"/>
<uses-permission android:name="android.permission.READ_LOGS"/>
<uses-permission android:name="android.permission.WRITE_CALL_LOG"/>
<uses-permission android:name="android.permission.READ_CALL_LOG"/>
<uses-permission android:name="android.permission.READ_SMS"/>
<uses-permission android:name="android.permission.SEND_SMS"/>
<uses-permission android:name="android.permission.RECEIVE_SMS"/>
<uses-permission android:name="android.permission.RECEIVE_MMS"/>
<uses-permission android:name="android.permission.WRITE_SMS"/>
<uses-permission android:name="android.permission.DISABLE_KEYGUARD"/>
<uses-permission android:name="com.google.android.providers.gsf.permission.READ_GSERVICES"/>
<permission android:name="com.android.launcher.action.INSTALL_SHORTCUT"/>
<uses-permission android:name="com.android.launcher.permission.INSTALL_SHORTCUT"/>
<permission android:name="com.contapps.android.permission.C2D_MESSAGE" android:protectionLevel="signature
<uses-permission android:name="com.contapps.android.permission.C2D_MESSAGE"/>
<uses-permission android:name="com.google.android.c2dm.permission.RECEIVE"/>
<permission android:name="com.contapps.android.permission.MAPS_RECEIVE" android:protectionLevel="signature
<uses-permission android:name="com.contapps.android.permission.MAPS_RECEIVE"/>
<uses-feature android:name="android.hardware.telephony" android:required="false"/>
<supports-screens android:anyDensity="true" android:largeScreens="true" android:normalScreens="true" andr
<application android:allowBackup="true" android:icon="@drawable/icon" android:label="@string/app_name" an
<uses-library android:name="com.google.android.maps"/>
```

Figura 3.4: AndroidManifest.xml de la aplicación

Como se puede observar, esta aplicación requiere numerosos permisos, algunos de ellos son:

READ_CONTACTS. Permite a la aplicación la lectura de contactos del dispositivo.

VIBRATE. Permite a la aplicación vibrar.

INTERNET. Permite a la aplicación abrir sockets de red.

CALL_PHONE. Permite a la aplicación efectuar una llamada telefónica sin hacer uso del teclado.

BATTERY_STATS. Permite a la aplicación conocer las estadísticas de la batería del dispositivo.

Los *permisos runtime* solo están disponibles a partir de Android 6.0 (API 23), son notificados al usuario cuando se ejecuta la aplicación, no en el momento de instalación. Estos permisos se pueden aceptar o denegar, para que no se pregunte al usuario cada vez que inicia la aplicación. Sin embargo, los *permisos estáticos* son aquellos que están disponibles en versiones anteriores a 5.1 (API 22) y son notificados al usuario en el momento de instalación de la aplicación.

3.2. Firmas y certificados

En esta sección, se procederá a estudiar las firmas y los certificados digitales de la aplicación anterior.

Lo primero que se debe hacer es instalar la aplicación en nuestro dispositivo virtual (AVD), para ello, se inicializa nuestro dispositivo virtual y una terminal para instalar el .apk, véase la figura .

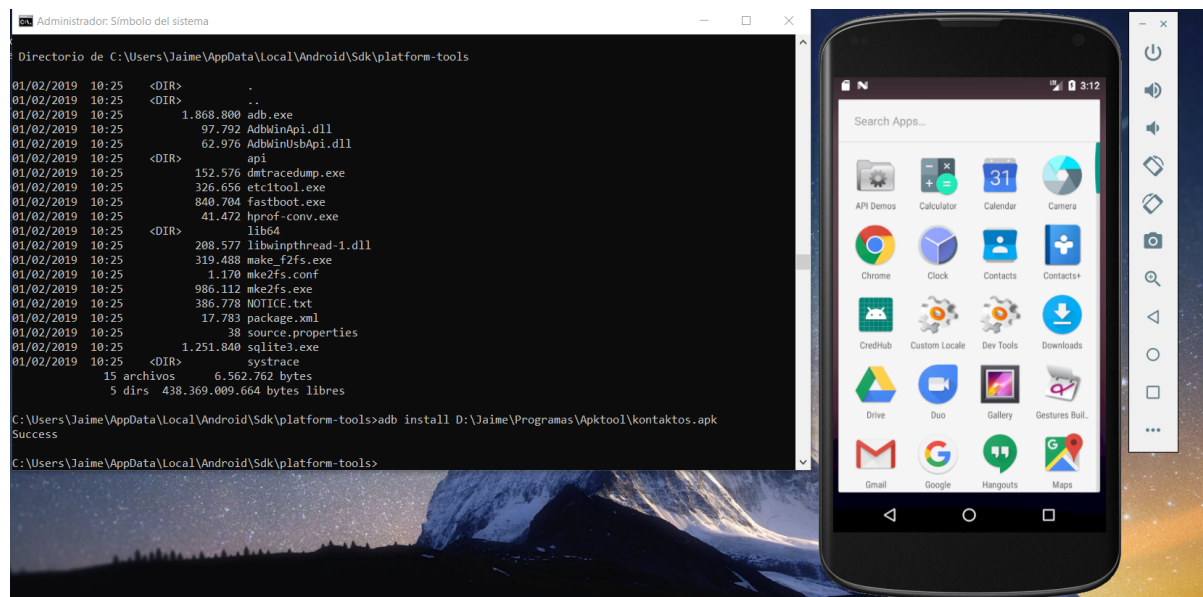


Figura 3.5: Instalación de una apk vía adb

Seguidamente, se procede a descomprimir el .apk para visualizar los certificados, que se encuentran en el directorio "META-INF". Se descomprime la aplicación haciendo uso de la herramienta *Winrar*. A continuación, hacemos uso de la herramienta *keytool* para visualizar el contenido del certificado digital declarado por el dueño de la llave pública, véase la figura 3.6.

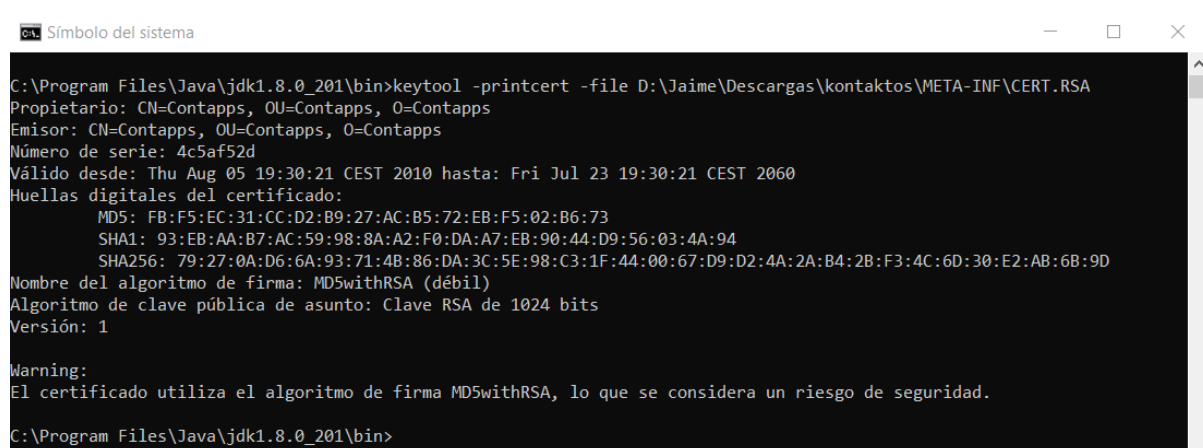


Figura 3.6: Certificado de la apk

Como se puede comprobar, el certificado consta de varias partes, son las siguientes:

Propietario. Propietario de la aplicación.

Emisor. Emisor de la aplicación.

Serie. Número de serie de la aplicación.

Validez. Indica el periodo de validez del certificado.

Huellas digitales. Funciones resumen únicas que identifican la aplicación.

Algoritmo de firma. Algoritmo de firmado utilizado en la aplicación.

Algoritmo de clave pública. Algoritmo utilizado para generar la clave pública.

Versión. Versión de la aplicación.

Los valores obtenidos de la aplicación *kontaktos*, son los siguientes:

Serial number	4c5af52d
Validez	Aug 2010 - Jul 2060
Codificación clave pública	RSA
Tamaño clave pública	1024 bits
Modulus	????????
Algoritmo firma	MD5withRSA

La siguiente tarea consiste en obtener los hash criptográficos del logotipo de la aplicación y de al menos tres imágenes distintas que tengan densidad de píxeles diferentes, por ello, se debe analizar el fichero CERT.SF que se puede visualizar con cualquier editor de texto, en este caso, *Notepad ++*.

El fichero contiene más de 5000 líneas de código, así que, para localizar el nombre del icono de la aplicación se accede al directorio *res* se busca el icono en los diferentes subdirectorios. El logo de la aplicación recibe el nombre de *icon*, así que, se busca en el fichero este archivo y se encuentra su información:

Name: res/drawable-hdpi/icon.png

SHA1-Digest: X5/mtN+YdCFiZSlqEC9QRQ4eXFo=

Para localizar tres imágenes diferentes se realiza de manera análoga, siendo cada directorio la densidad de píxeles de cada una, por ejemplo, *mdpi* (medium-dpi), *hdpi* (large-dpi) ... Por tanto, tres archivos válidos serían:

Name: res/drawable-hdpi/welcome_pic.png

SHA1-Digest: kEtv1qXDVEaipT5tpE6yJFbTVag=

Name: res/drawable-ldpi/wizard_thank_you.png

SHA1-Digest: oaQL3lC7JfkqM+rjezbyhW9iO3A=

Name: res/drawable-mdpi/wizard_free_sms_pic.png

SHA1-Digest: aiczu6u6zKX4kkjBzKZmODovr44=

Para la codificación del hash, se ha utilizado el algoritmo *SHA-1*, que separa la información en bloques de 512 bits y luego añade 80 vueltas utilizando una serie de vectores y mezclando la información con los siguientes hasta obtener un resumen de 160 bits.