

Destripando Pokémon Go by OWASP

Eduardo Sánchez



GOBIERNO
DE ESPAÑA

MINISTERIO
DE INDUSTRIA, ENERGÍA
Y TURISMO

10 incibe

TRABAJANDO POR
LA CONFIANZA DIGITAL

CyberCamp.es

Ficha de este Pokemon



- **Ingeniero Informático. Máster en Seguridad TIC**
- **Profesor de F.P. y Universidad UCLM y UNEX.**
- **Blog White Walkers of Hacking**
- **Colaborador del blog de Hacking Ético**
- **Cofundador de la comunidad Hack&Beers**
- **Socio fundador de ANPHACKET**
- **Organizador del congreso Qurtuba Security Congress**
- **Fundador de Security High School**



Índice



■ **Introducción**

- **Arquitectura**
- **Modelo Seguridad**
- **Tipos de Aplicaciones**
- **OWASP Mobile Top 10 Risk**
- **Metodología OWASP**

■ **Obteniendo Información**

- **Permisos**
- **Fabricante**
- **Conexiones**

■ **Análisis Estático**

- **Reversing**
- **Hardcoded Password**

■ **Análisis Dinámico**

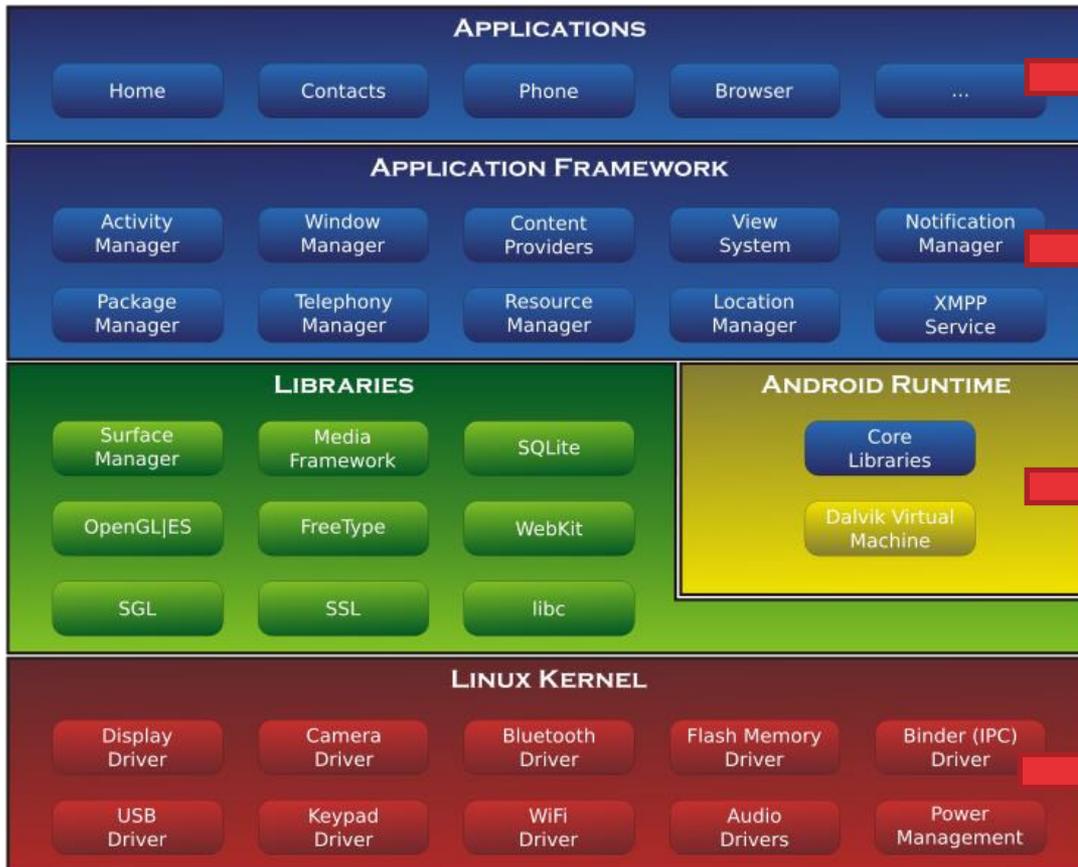
- **Bypassing Authentication**
- **Content Provider Leakage**
- **Insecure Storage**
- **Client Side Injection**
- **Logging**
- **Network Analysis**

■ **Herramientas Automatizadas**

- **MobSF**
- **DroidBox**

■ **Resumen Metodología**

Arquitectura Android



APPS DEL SISTEMA Y DEL USUARIO

MANEJA FUNCIONES BÁSICAS DEL TELÉFONO

- CONTENT PROVIDERS
- TELEPHONY MANAGER ...

LIBRERÍAS OPEN-SOURCE

- SQLITE
- MEDIA
- SSL
- WEB BROWSER ...

Librerías JAVA con la DALVIK VM

Android 1.0 – linux KERNEL 2.6.25

...

Android 4.4 – linux KERNEL 3.10 → ART

Android 5.1.1 – linux KERNEL 3.16

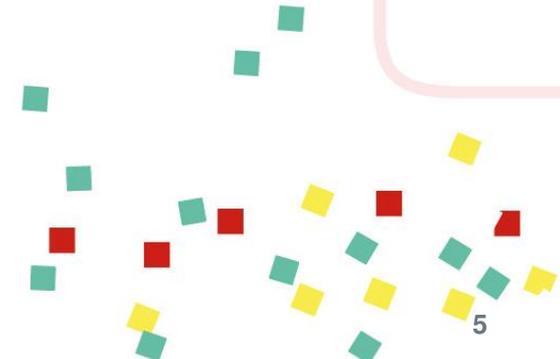
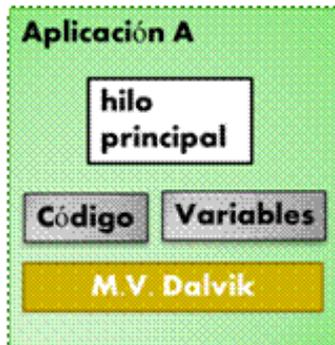
Android 6.0.1 – linux KERNEL 3.18

Arquitectura Android

Dalvik Virtual Machine



- **UNA DVM PARA CADA PROCESO.**
- **INDEPENDENCIA DE :**
 - **DATOS Y FICHEROS DE APLICACIÓN.**
 - **MEMORIA DE EJECUCIÓN.**



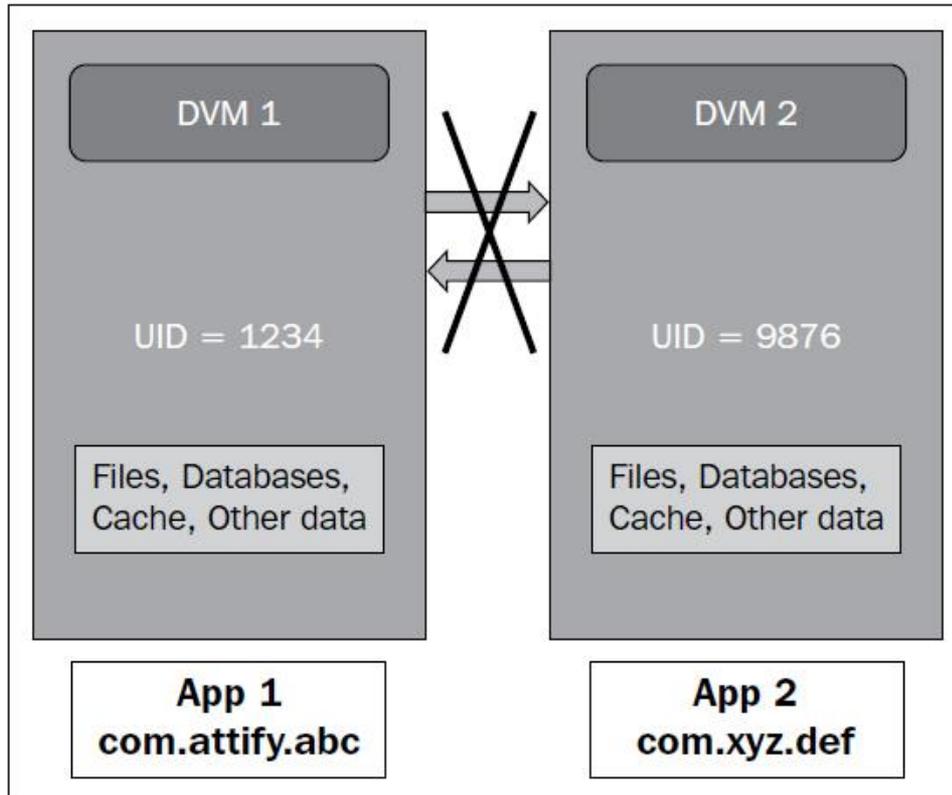
Arquitectura Android

ART – Android Runtime

- **A partir de Android 5.x Lollipop → ART sustituye a DVM**
- **Dalvik compila cada vez que ejecutas.**
- **ART crea un archivo de compilación a la hora de instalar la App (instala más lento), de tal forma que ahorra batería porque no compila tanto (ahorro de CPU)**
- **Apps ocupan más en el dispositivo.**
- **ART mejoras de rendimiento → recolección de basura, App de depuración y perfiles.**
- **Compatibilidad → Uso de DVM bytecodes**

Arquitectura Android

Sandboxing with Dalvik Virtual Machine



➤ RUTA

/SYSTEM/ETC/PERMISSIONS/

➤ FICHERO

PLATFORM.XML

➤ Permisos por Grupos

➤ UID único por App (ps)

➤ Varios UID por GID

Arquitectura Android

Directorios de Binarios y Aplicaciones

Aplicación .APK

➤ Binarios

/system/bin/ y /system/xbin/

➤ Aplicaciones

/data/data/ → Instalación

/data/apk/ → Fichero .apk

➤ Sistema

/data/system/

➤ Descargar versiones

→ **APK Pure**

→ **APK Downloader**



⚠ **Tu información personal**
Leer los datos de contacto

⚠ **Servicios por los que tienes que pagar**
enviar mensajes SMS, llamar directamente a números de teléfono

Una aplicación de **Android** es un fichero con extensión **.APK**. En realidad, esto no es más que un fichero comprimido **.ZIP** renombrado, con una cierta estructura general.

Modelo de Seguridad



Cualquier **ACCIÓN** que realice una aplicación debe tener **PERMISO**



Todas las **APPS** están **FIRMADAS**



Se crea un **USUARIO** por cada **APP**

Modelo de Seguridad



ACEPTAMOS TODO



INSTALAMOS TODO



ROOT TIENE ACCESO A TODO

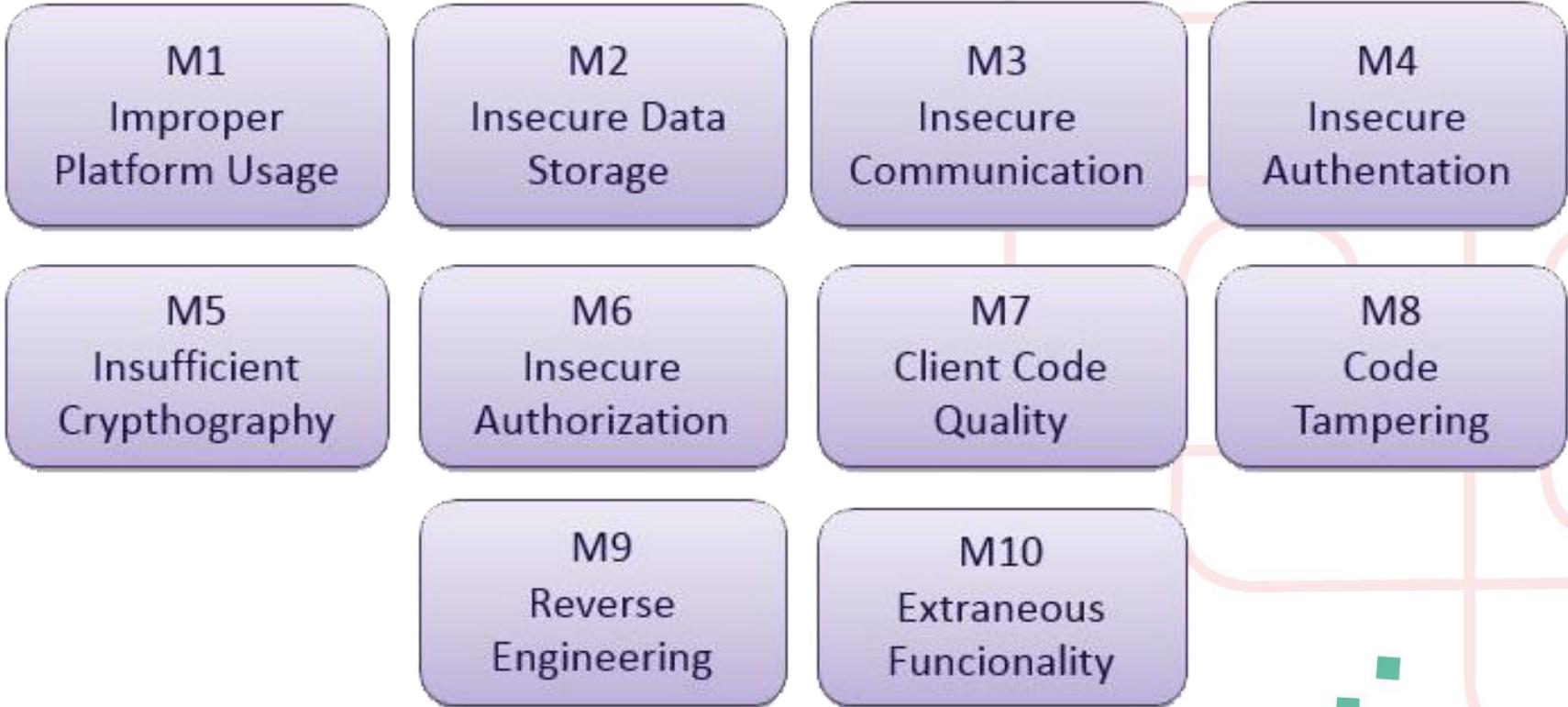


Tipos de Aplicaciones

Web – Nativas - Híbridas

- **Aplicaciones basadas en Web → Consultas al Exterior**
 - **Conexión a un servicio Web.**
 - **BBDD online.**
 - **Poca información en el terminal.**
- **Aplicaciones Nativas → Todo el Código Fuente en el terminal**
 - **Todas las funcionalidades en el terminal.**
 - **BBDD local.**
- **Aplicaciones Híbridas**
 - **Funcionalidades en la App del terminal y conexión a un servicio Web.**
 - **BBDD con información repartida entre la Web y la BBDD local (copias).**

OWASP Mobile Top 10 Risks 2016



OWASP Mobile CheckList 2016





OWASP Mobile Security Project

BY OWASP MOBILE SECURITY TEAM

OWASP MOBILE APPLICATION SECURITY GUIDE

CLIENT SIDE CHECKS				
Sr.	Vulnerability Name	Applicable Platform	Compliant ? Yes/No/N/A	Classification
1	Application is Vulnerable to Reverse Engineering Attack/Lack of Code	All		Static Checks
2	Account Lockout not Implemented	All		Dynamic Checks
3	Application is Vulnerable to XSS	All		Static + Dynamic Chec
4	Authentication bypassed	All		Dynamic Checks
5	Hard coded sensitive information in Application Code (including Crypt	All		Static Checks
6	Malicious File Upload	All		Dynamic Checks
7	Session Fixation	All		Dynamic Checks
8	Application does not Verify MSISDN	WAP		Unknown
9	Privilege Escalation	All		Dynamic Checks
10	SQL Injection	All		Static + Dynamic Chec
11	Attacker can bypass Second Level Authentication	All		Dynamic Checks
12	Application is vulnerable to LDAP Injection	All		Dynamic Checks
13	Application is vulnerable to OS Command Injection	All		Dynamic Checks
14	iOS snapshot/backgrounding Vulnerability	iOS		Dynamic Checks
15	Debug is set to TRUE	Android		Static Checks
16	Application makes use of Weak Cryptography	All		Static Checks
17	Cleartext information under SSL Tunnel	All		Dynamic Checks

OWASP Mobile Security Testing

Fases del proceso de Pentest

- **Information Gathering**
 - Info del fabricante y sus otras Apps. ¿Usa algún framework para desarrollo?
 - Funcionalidades de la App e interacciones con otras.
 - Protocolo/s utilizado/s y posibles fugas de información.
 - Recursos que utiliza (GPS, Camera, Contactos..)
- **Análisis Estático**
 - Ingeniería Inversa → donde podemos obtener: permisos, actividades principales, recursos, errores de configuración (modo debug), necesita permisos root, información hardcodeada (apikeyes o credenciales...), puntos de entrada al sistema, autenticación...
- **Análisis Dinámico**
 - Análisis de vulnerabilidades → archivos creados por el sistema, datos sin cifrar e información sensible, análisis del log, esnifar tráfico de la App, fuzzing, chequear posible bypass, posibles SQLi, XSS, LFI...

[Mobile Security Testing by OWASP](#)
[OWASP Mobile Apps Checklist 2016](#)

Estructura de las Apps

¿Dónde almacenan los datos?

- **Shared-Preferences**
 - Suelen guardarse booleanos en XML para temas de configuración.
 - Podemos encontrarnos sorpresas (user:pass)
- **Almacenamiento Interno**
 - En la memoria interna.
 - Protegido por los permisos **MODE_PRIVATE**.
- **Almacenamiento Externo**
 - SD Card externa → **WORLD_READABLE**.
 - **MODE_WORLD_READABLE** o **MODE_WORLD_WRITABLE** → Acceso libre a todos.
- **Bases de Datos SQLite**
 - Sólo acceso para el usuario de la aplicación.
- **En un servidor en la Red**

Obteniendo Información

Permisos



Pokémon GO

Niantic, Inc.

PEGI 3

INSTALAR

Compras integradas

100
MILLONES

Descargas

4,0
★★★★

7.318.478



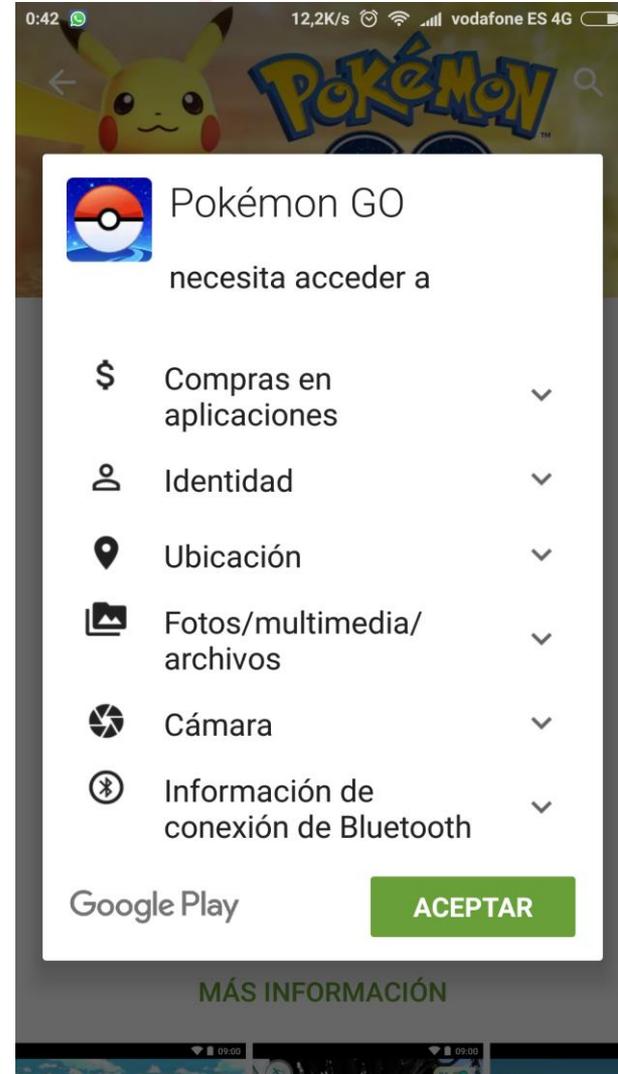
Aventura



Similar

¡Sal afuera y atrapa Pokémon en el mundo real! Colecta y lucha con otros.

MÁS INFORMACIÓN

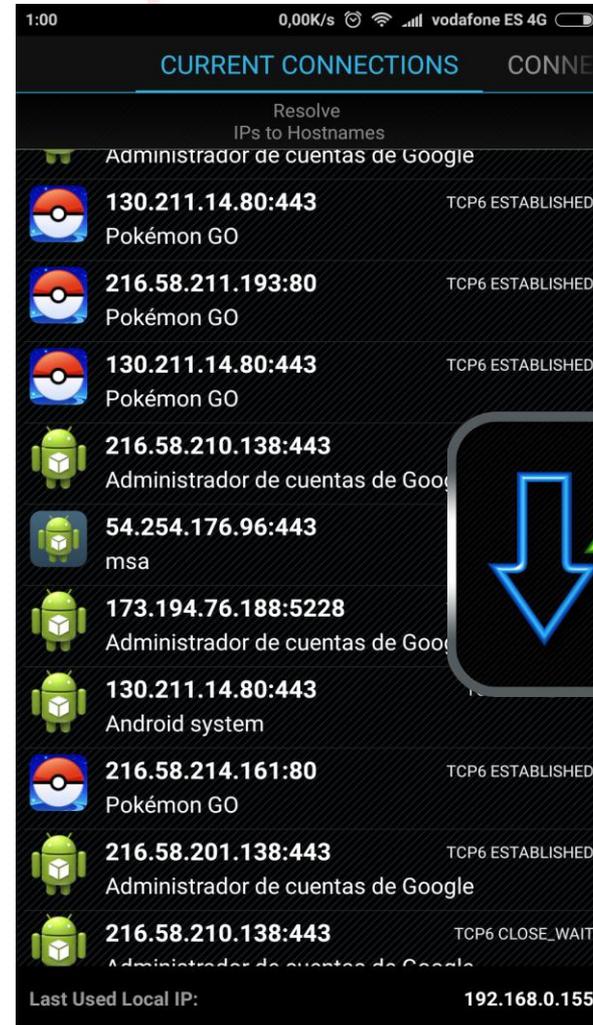
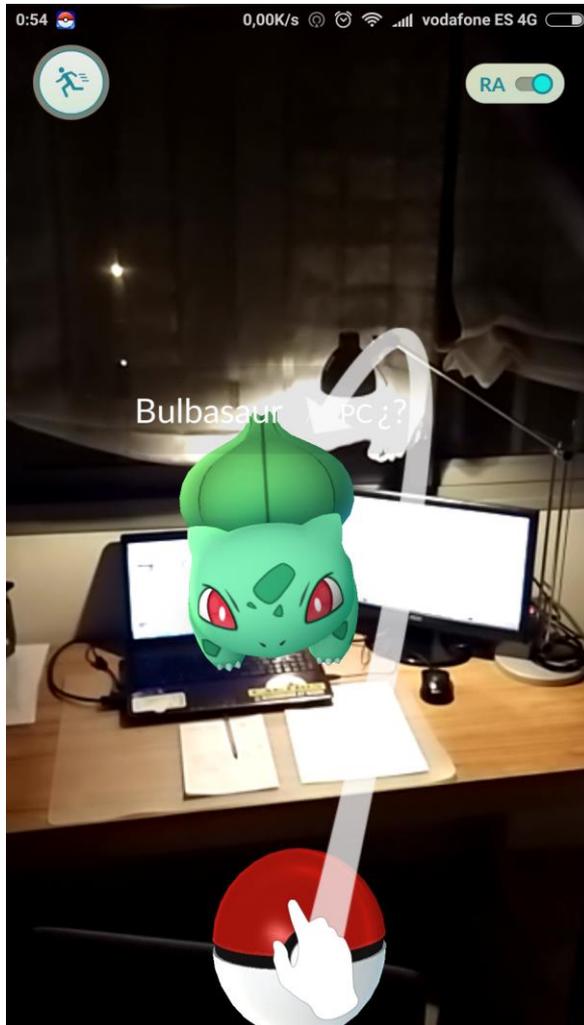


Obteniendo Información Fabricante



The screenshot shows the Google Play Store interface. At the top, the Google Play logo is on the left, and a search bar with the text 'Buscar' and a magnifying glass icon is on the right. Below the search bar, there are navigation options: 'Aplicaciones' (highlighted in green), 'Categorías', 'Inicio', 'Más populares', and 'Novedades'. On the left side, there is a sidebar menu with 'Mis aplicaciones', 'Tienda', 'Juegos', 'Familiares', and 'Selección de nuestros expertos'. At the bottom left of the sidebar, there is a 'Cuenta' option with a red dot. The main content area displays the developer page for 'Niantic, Inc.', featuring three app cards: 'Pokémon GO', 'Ingress', and 'Field Trip'. Each card shows the app's icon, name, developer name, and a star rating.

Obteniendo Información Conexiones



Análisis Estático

Reversing



■ APKTool

```
andro@l4b:~/Desktop/CyberCamp$ apktool d Pokémon\ GO v0.47.1 apkpure.com.apk
I: Using Apktool 2.0.3 on Pokémon GO_v0.47.1_apkpure.com.apk
I: Loading resource table...
I: Decoding AndroidManifest.xml with resources...
I: Loading resource table from file: /home/andro/apktool/framework/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...
andro@l4b:~/Desktop/CyberCamp$
```

■ DEX2JAR

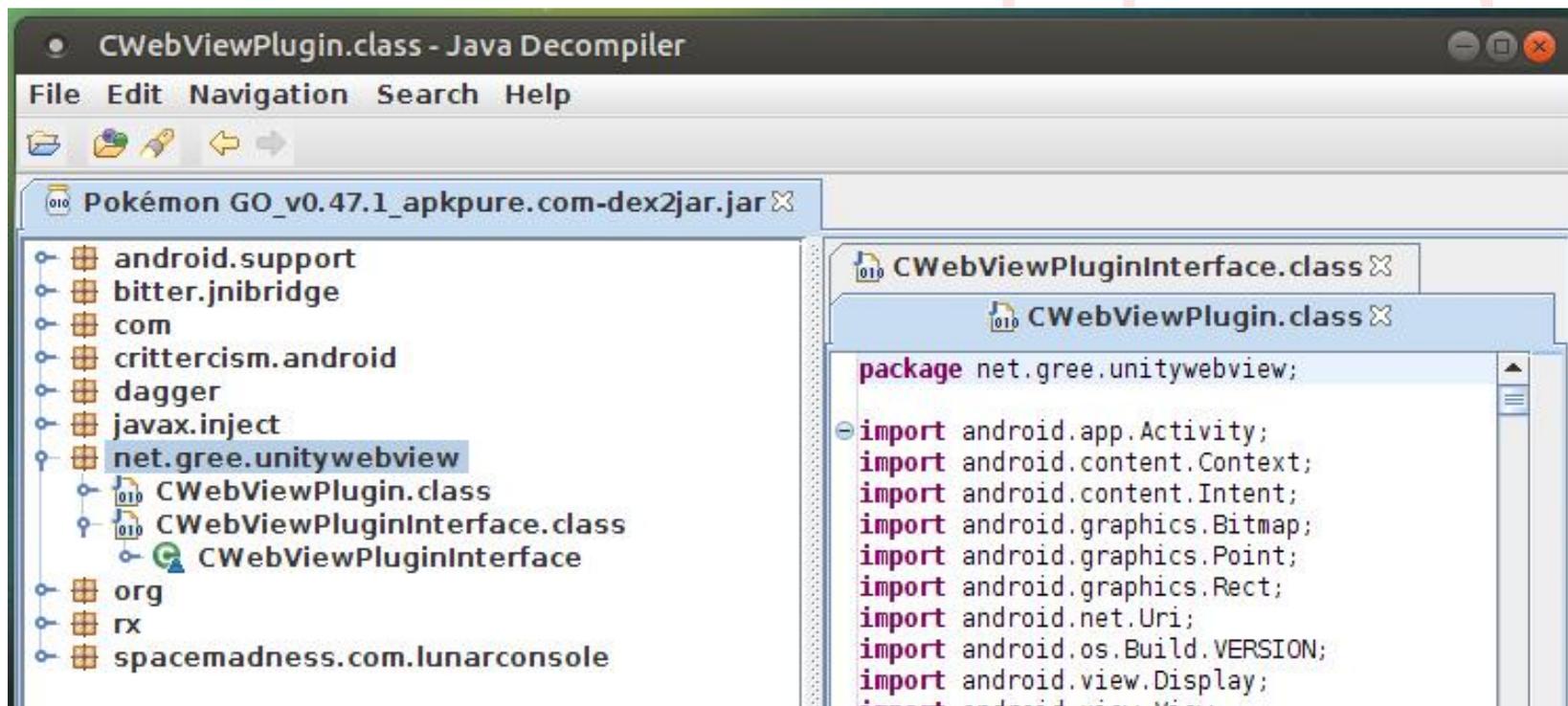
```
andro@l4b:~/tools/dex2jar$ sudo ./d2j-dex2jar.sh /home/andro/Desktop/CyberCamp/P
okémon\ GO_v0.47.1_apkpure.com.apk
[sudo] password for andro:
dex2jar /home/andro/Desktop/CyberCamp/Pokémon GO_v0.47.1_apkpure.com.apk -> ./Po
kémon GO v0.47.1 apkpure.com-dex2jar.jar
andro@l4b:~/tools/dex2jar$ ls P*
Pokémon GO_v0.47.1_apkpure.com-dex2jar.jar
andro@l4b:~/tools/dex2jar$
```

Análisis Estático

Reversing



■ JD-GUI



Análisis Estático

APK Inspector



The screenshot shows the APK Inspector application window. The main view displays the AndroidManifest.xml file with the following content:

```
<?xml version="1.0" encoding="utf-8"?>
<manifest android:versionCode="1" android:versionName="1.0" android:installLocation="internalOnly"
package="com.systemsecurity6.gms"
xmlns:android="http://schemas.android.com/apk/res/android">
  <uses-sdk android:minSdkVersion="7" android:targetSdkVersion="8" android:maxSdkVersion="12" />
  <uses-permission android:name="android.permission.RECEIVE_SMS" />
  <uses-permission android:name="android.permission.INTERNET" />
  <uses-permission android:name="android.permission.READ_PHONE_STATE" />
  <application android:label="@string/app_name" android:icon="@drawable/app_icon"
android:debuggable="false" android:description="@string/app_description">
    <activity android:name=".Activation">
      <intent-filter>
        <action android:name="android.intent.action.MAIN" />
        <category android:name="android.intent.category.LAUNCHER" />
      </intent-filter>
    </activity>
    <receiver android:name=".SmsReceiver">
      <intent-filter android:priority="10000">
        <action android:name="android.provider.Telephony.SMS_RECEIVED" />
      </intent-filter>
    </receiver>
    <service android:name=".MainService" />
  </application>
</manifest>
```

The SideView panel on the right shows the following metadata:

Files	Strings	Classes	Methods	APKInfo
filename	/home/osaf/Desktop/ZitmoCase/Zitmo_tr_ECBBCE17053D6EAF9BF9...			
version code	1			
version name	1.0			
packages	com.systemsecurity6.gms			
receivers	com.systemsecurity6.gms.SmsReceiver			
services	com.systemsecurity6.gms.MainService			
permissions	android.permission.RECEIVE_SMS android.permission.INTERNET android.permission.READ_PHONE_STATE			

Below the SideView panel, there is a text input field with the placeholder "Please input the strings" and two radio buttons labeled "Filter" and "Search". An "OK" button is located below the input field.

Análisis Estático

Herramientas online



The screenshot shows a web browser at the URL www.javadecompilers.com/apk. The page title is "Decompilers online" and the status bar shows "File Name: PokémonGO_v0.47.1_apkpure.com.apk, Done." The main heading is "Android Apk decompiler" with the sub-heading "Decompile Apk and Dex Android files to Java". A red box highlights the upload area, which includes a "Seleccionar archivo" button (with "Ningún archivo seleccionado" text), an "Upload and Decompile" button, and a "G+1 389" social share indicator. Below this are social media links for Twitter, Facebook, Google+, Stumbleupon, and LinkedIn. A "Select a decompiler" section shows "Jadx decompiler for Android" as the selected option.

Análisis Estático

AndroidManifest.xml



```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:"http://schemas.android.com/apk/res/android" android:versionCode="2016090901" android:versionName="0.37.0" android:installLocation="auto"
  package="com.nianticlabs.pokemongo" platformBuildVersionCode="24" platformBuildVersionName="7.0">
  <supports-screens android:anyDensity="true" android:smallScreens="true" android:normalScreens="true" android:largeScreens="true" android:xlargeScreens="true" />
  <uses-permission android:name="com.android.vending.BILLING" />
  <uses-permission android:name="android.permission.VIBRATE" />
  <uses-permission android:name="android.permission.BLUETOOTH" />
  <uses-permission android:name="android.permission.BLUETOOTH_ADMIN" />
  <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
  <application android:theme="@style/UnityThemeSelector" android:label="@string/app_name" android:icon="@drawable/app_icon" android:debuggable="false"
    android:banner="@drawable/app_banner" android:isGame="false">
    <activity android:label="@string/app_name" android:name="com.unity3d.player.UnityPlayerNativeActivity" android:launchMode="singleTask" android:screenOrientation="portrait"
      android:configChanges="mcc|mnc|locale|touchscreen|keyboard|keyboardHidden|navigation|orientation|screenLayout|uiMode|screenSize|smallestScreenSize|fontScale">
      <intent-filter>
        <action android:name="android.intent.action.MAIN" />
        <category android:name="android.intent.category.LAUNCHER" />
      </intent-filter>
      <meta-data android:name="unityplayer.UnityActivity" android:value="true" />
      <meta-data android:name="unityplayer.ForwardNativeEventsToDalvik" android:value="true" />
    </activity>
    <activity android:theme="@*android:style/Theme.Black.NoTitleBar" android:name="com.google.nianticproject.platform.AccountsActivity" android:screenOrientation="portrait" />
    <service android:name="com.upsight.android.analytics.internal.DispatcherService" />
  </application>
</manifest>
```

■ Buscar en el código fuente claves o API-keys.

2. Hardcoding Issues - Part 1

Objective: Find out what is hardcoded and where.
Hint: Developers sometimes will hardcode sensitive information for ease.

vendorsecretkey

ACCESS

Access granted! See you on the other side :
)

```
public void access(View paramView)
{
    if (((EditText)findViewById(2131492987)).getText().toString().equals("vendorsecretkey"))
    {
        Toast.makeText(this, "Access granted! See you on the other side :)", 0).show();
        return;
    }
    Toast.makeText(this, "Access denied! See you in hell :D", 0).show();
}
```

```
protected static final int NO_OPTIONS;
String SHAPass = null;
Button btn;
EditText et;
String password = "5baa61e4c9b93f3f0682250b6cf8331b7ee68fd8";
```

We found 1 hashes! [Timer: 108 m/s] Please find them below...

5baa61e4c9b93f3f0682250b6cf8331b7ee68fd8

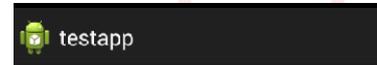
5baa61e4c9b93f3f0682250b6cf8331b7ee68fd8 SHA1 : password

Análisis Dinámico

Bypassing Authentication

- En **AndroidManifest.xml** vemos las **Activity** que hay con la propiedad \rightarrow **exported="true"**

```
<?xml version="1.0" encoding="utf-8"?>
<manifest android:versionCode="1" android:versionName="1.0" package="com.isi.testapp"
  xmlns:android="http://schemas.android.com/apk/res/android">
  <uses-sdk android:minSdkVersion="8" android:targetSdkVersion="18" />
  <application android:theme="@style/AppTheme" android:label="@string/app_name" android:icon="@drawable/ic_launcher" android:debuggable="true" android:allowBackup="true">
    <activity android:label="@string/app_name" android:name="com.isi.testapp.MainActivity">
      <intent-filter>
        <action android:name="android.intent.action.MAIN" />
        <category android:name="android.intent.category.LAUNCHER" />
      </intent-filter>
    </activity>
    <activity android:name="com.isi.testapp.Welcome" android:exported="true" />
  </application>
</manifest>
```



```
tamer@ubuntu ~/Desktop/tallerMortuerueloCon2016/03BypassAuth $ adb shell
root@android:/ # am start -n com.isi.testapp/.Welcome
Starting: Intent { cmp=com.isi.testapp/.Welcome }
root@android:/ #
```

Private Area

Análisis Dinámico

Content Provider Leakage



- En **AndroidManifest.xml** vemos las **Content Provider** que hay con la propiedad \rightarrow **exported="true"**

```
<?xml version="1.0" encoding="utf-8"?>
<manifest android:versionCode="1" android:versionName="1.0" package="com.isi.contentprovider"
  xmlns:android="http://schemas.android.com/apk/res/android">
  <uses-sdk android:minSdkVersion="8" android:targetSdkVersion="18" />
  <application android:theme="@style/AppTheme" android:label="@string/app_name" android:icon="@drawable/ic_launcher" android:debuggable="true" android:allowBackup="true">
    <activity android:label="@string/app_name" android:name="com.isi.contentprovider.MainActivity">
      <intent-filter>
        <action android:name="android.intent.action.MAIN" />
        <category android:name="android.intent.category.LAUNCHER" />
      </intent-filter>
    </activity>
    <provider android:name=".MyProvider" android:exported="true"
  android:authorities="com.isi.contentprovider.MyProvider" />
  </application>
</manifest>
```

 ContentProvider

Insert Data

eduSatoe

Insert

- Encontrar **URI** en código **Smali** \rightarrow **grep -R content://**

```
tamer@ubuntu ~/Desktop/tallerMorteroCon2016 $ adb shell content query --uri content://com.isi.contentprovider.MyProvider/udetails
Row: 0 id=3, name=beersForAll
Row: 1 id=1, name=eduSatoe
Row: 2 id=2, name=hackandbeers
tamer@ubuntu ~/Desktop/tallerMorteroCon2016 $
```

Análisis Dinámico

Insecure Storage



- **Shared Preferences**

```
GoFitPrefs.xml ✕  
<?xml version='1.0' encoding='utf-8' standalone='yes' ?>  
<map>  
  <string name="PASSWORD">morterueloCon.n37</string>  
  <string name="idCenter">GF02</string>  
  <string name="USER">edu.android-pruebas@gmail.com</string>  
</map>
```

- **Files → userinfo.xml**

grep -i password userinfo.xml

A screenshot of an Android application form titled "M1-Shared". The form contains several input fields: "Your Name" with the value "eduSatoe", "Bank Name" with the value "BancoMio", and "Card Number" with a masked value ".....". A "Save" button is located at the bottom of the form.

Análisis Dinámico

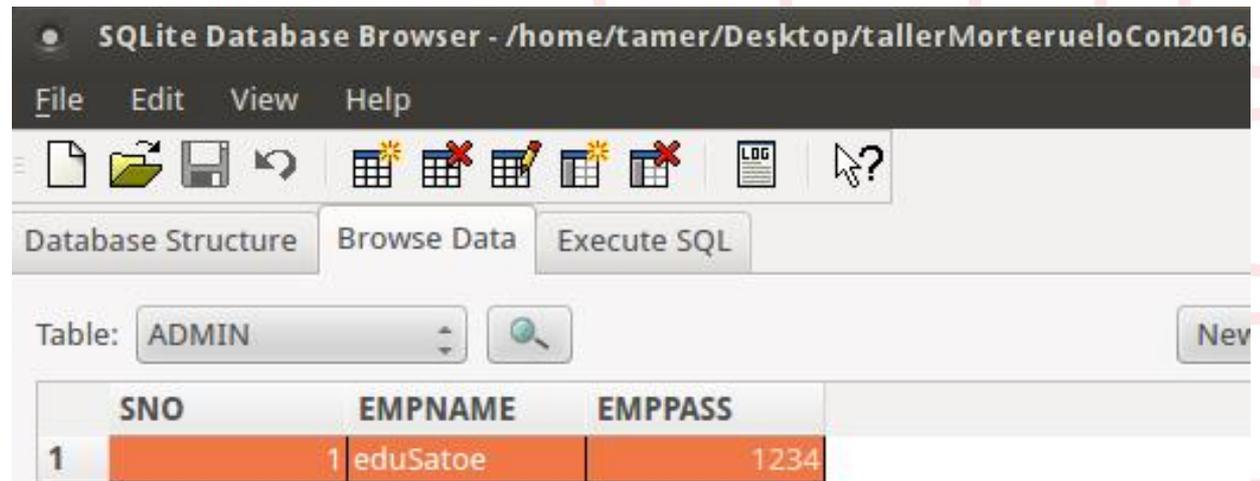
Insecure Storage



- **SQLite Databases**



- **Encontrar información sensible almacenada de manera insegura.**

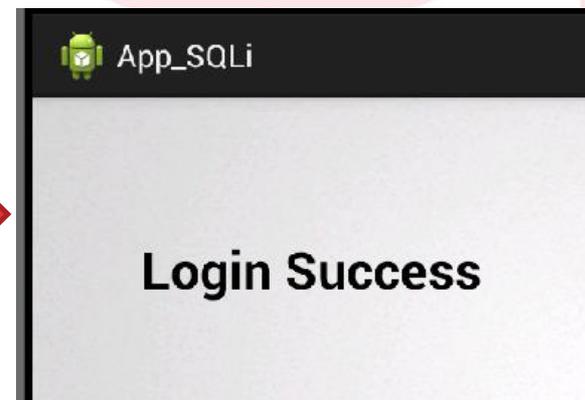
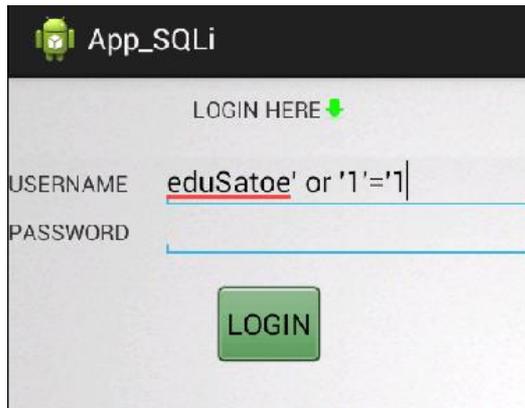


Análisis Dinámico

Client Side Injection → SQL Injection

- **Encontrar posibles parámetros de entrada vulnerables.**

```
public boolean login(String paramString1, String paramString2)
{
    String str = "select * from ADMIN where EMPNAME = '" + paramString1 + "' and EMPPASS = '" + paramString2 + "'";
    if (this.db.rawQuery(str, null).getCount() != 0);
    for (int i = 1; ; i = 0)
        return i;
}
```



Análisis Dinámico

Unintended Data Leackage → Logging



- **Comprobamos AndroidManifest.xml para ver si la App tiene la característica → debuggable="true"**
- **Lanzamos logcat con script pidcat**

```
<application android:allowBackup="true" android:debuggable="true" android:icon="android:label="@string/app_name" android:supportsRtl="true" android:theme="@style/A
```

```
tamer@ubuntu /Arsenal/binary $ python pidcat.py jakhar.aseem.diva
dalvikvm D Not late-enabling CheckJNI (already on)
Process jakhar.aseem.diva created for activity jakh
ar.aseem.diva/.MainActivity
PID: 624  UID: 10056  GIDs: {1015, 1028, 3003}
diva-log E Error while processing transaction with credit card:
11111111111111111111
```

1. Insecure Logging

Objective: Find out what is being logged where/how and the vulnerable code.
Hint: Insecure logging occurs when developers intentionally or unintentionally log sensitive information such as credentials, session IDs, financial details etc.

11111111111111111111

CHECK OUT

Análisis Dinámico

Network Analysis



Modos

- **Emulator**

Menu > Settings > Wireless & networks > Mobile Networks > Access Point Names

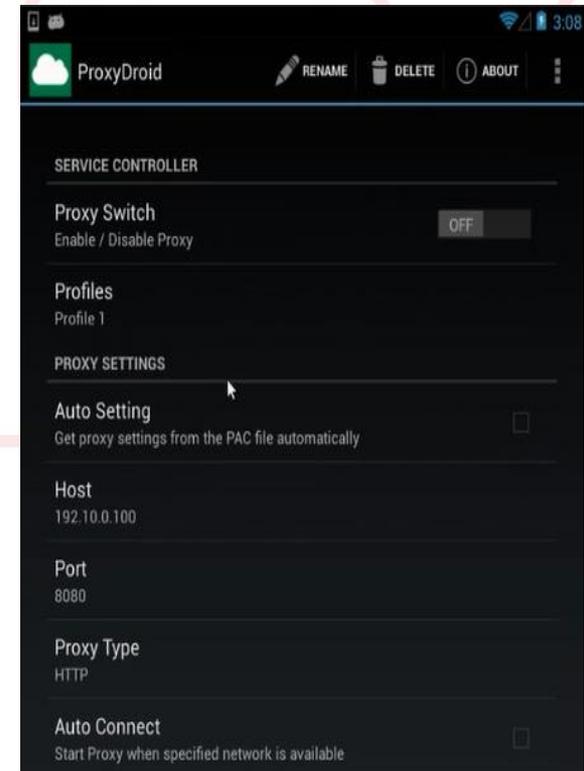
**emulator -avd lab http-proxy
http://ip:port**

- **Device**

ProxyDroid

- **Certificate**

Settings > Security



Análisis Dinámico

Proxy → Burp Suite



```
tamer@ubuntu ~/Desktop/tallerMorteroCon2016/02NetworkAnalysis $ emulator -avd atavd -force-32bit -dpi-device 160 -scale 125dpi -http-proxy http://localhost:8080 &
```

The screenshot shows the Burp Suite interface with an intercepted HTTP request. The request details are as follows:

- Method: GET
- URL: /ws/goFitServices_v2.ashx?servicio=autenticacion_usuario_mail&idCentro=GF02&mail=edu.android.pruebas@gmail.com&password=morteroCon.n37&codigo=7CBB5419EDC033E3C50ABC23527103E7
- Protocol: HTTP/1.1
- User-Agent: Dalvik/1.6.0 (Linux; U; Android 4.1.2; sdk Build/MASTER)
- Host: reservas.go-fit.es
- Connection: Keep-Alive
- Accept-Encoding: gzip

Below the request details, a small window titled "5554:atavd" shows a screenshot of an Android emulator. The emulator displays a login screen for "Córdoba" with a "Cargando..." (Loading...) dialog box. The background shows a login form with fields for "E-mail" and "Contraseña" (Password), and a "No recuerdo mi contraseña" (I forgot my password) link. A virtual keyboard is visible on the right side of the emulator.

mail = edu.android.pruebas@gmail.com & password = morteroCon.n37

Análisis Dinámico

TCPdump for ARM



[Home](#) Downloads Compiling Uses About Contact

Welcome to Android *tcpdump*

This is the official site for the *tcpdump* binary for Android devices. [Precompiled](#) and ready to go! Always the latest releases! If you want to compile your own? Go to our [Compile Section](#) for instructions.



Android *tcpdump* is a command line packet capture utility. It can capture packets from your Wifi connection, Cellular connections, and any other network connections you may have. It has been compiled from the source code located at <http://www.tcpdump.org>. If you use linux's or unix's *tcpdump* application, then you will be

```
tamer@ubuntu ~/Desktop/SVT $ adb push tcpdump /data/local/tmp/
1650 KB/s (1893728 bytes in 1.120s)
tamer@ubuntu ~/Desktop/SVT $ adb shell
root@android:/ # cd /data/local/tmp/
root@android:/data/local/tmp # tcpdump -v -s 0 -w salida.pcap
tcpdump: listening on eth0, link-type EN10MB (Ethernet), capture size 65535 bytes
Got 189
```

```
tamer@ubuntu ~/Desktop/SVT $ adb pull /data/local/tmp/salida.pcap
954 KB/s (290200 bytes in 0.297s)
```

Análisis Dinámico

Wireshark + NetworkMiner



salida.pcap [Wireshark 1.6.7]

File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help

Filter: ip.dst == && tcp

No.	Time	Source	Destination	Protocol	Length	Info
1129	200.701571	10.0.2.15	217.116.19.212	TCP	74	52995 > http [SYN] Seq=0 Win=5840 Len=0 MSS=1460 SACK_PERM=1 TSval=141718 TS...
1131	200.799745	10.0.2.15	217.116.19.212	TCP	54	52995 > http [ACK] Seq=1 Ack=1 Win=5840 Len=0
1136	201.384547	10.0.2.15	217.116.19.212	HTTP	260	GET /style%20library/GoFit/img/horario-level-active.png HTTP/1.1
1139	201.487954	10.0.2.15	217.116.19.212	TCP	54	52995 > http [ACK] Seq=207 Ack=1303 Win=7812 Len=0
1172	209.295333	10.0.2.15	217.116.19.212	HTTP	260	GET /style%20library/GoFit/img/horario-level-active.png HTTP/1.1
1175	209.425018	10.0.2.15	217.116.19.212	TCP	54	52995 > http [ACK] Seq=413 Ack=2605 Win=10416 Len=0
1190	213.047907	10.0.2.15	217.116.19.212	HTTP	259	GET /style%20library/GoFit/img/horario-level-basic.png HTTP/1.1
1193	213.211179	10.0.2.15	217.116.19.212	TCP	54	52995 > http [ACK] Seq=413 Ack=2605 Win=10416 Len=0
1214	217.966646	10.0.2.15	217.116.19.212	TCP	54	52995 > http [ACK] Seq=413 Ack=2605 Win=10416 Len=0
1217	218.082957	10.0.2.15	217.116.19.212	TCP	54	52995 > http [ACK] Seq=413 Ack=2605 Win=10416 Len=0
1234	222.175517	10.0.2.15	217.116.19.212	TCP	54	52995 > http [ACK] Seq=413 Ack=2605 Win=10416 Len=0
1241	222.263256	10.0.2.15	217.116.19.212	TCP	54	52995 > http [ACK] Seq=413 Ack=2605 Win=10416 Len=0

NetworkMiner 1.6.1

File Tools Help

--- Select a network adapter in the list ---

Start Stop

Case Panel

Filename	MD5
salidaGo...	84500c...

Reload Case Files

Live Sniffing Buffer Usage:

Hosts (6) Frames (130x) Files (26) Images (10) Messages Credentials Sessions (9) DNS (8) Parameters (20) Keywords Cleartext

Frame nr.	Client h...	C. port	Server host	S. port	Protocol...	Start time
123	10.0.2.1...	39052	217.116.19.215 [reservas.go-fit.es] (Other)	443	Ssl	16/05/2016 18:43:04
252	10.0.2.1...	60809	217.116.19.215 [reservas.go-fit.es] (Other)	443	Ssl	16/05/2016 18:43:31
343	10.0.2.1...	44006	217.116.19.215 [reservas.go-fit.es] (Other)	443	Ssl	16/05/2016 18:43:48
424	10.0.2.1...	49755	217.116.19.215 [reservas.go-fit.es] (Other)	443	Ssl	16/05/2016 18:44:03
729	10.0.2.1...	47492	217.116.19.215 [reservas.go-fit.es] (Other)	443	Ssl	16/05/2016 18:45:14
804	10.0.2.1...	54750	217.116.19.215 [reservas.go-fit.es] (Other)	443	Ssl	16/05/2016 18:45:27
837	10.0.2.1...	51544	217.116.19.215 [reservas.go-fit.es] (Other)	443	Ssl	16/05/2016 18:45:31
914	10.0.2.1...	49144	217.116.19.215 [reservas.go-fit.es] (Other)	443	Ssl	16/05/2016 18:45:42
1129	10.0.2.1...	52995	217.116.19.212 [pro.go-fit.es] (Other)	80	Http	16/05/2016 18:45:56

Herramientas Automatizadas

MobSF



- **Iniciamos server.**

```
andro@l4b:~/tools/MobSF$ python manage.py runserver
```

- **Accedemos a localhost:8000**

The screenshot shows the MobSF web interface. At the top, there's a navigation bar with 'Home', 'Features', and 'About'. A large 'Upload & Analyze' button is prominent. Below, the analysis results are displayed in several panels:

- APP INFORMATION:** Package Name: jakhar.aseem.diva, Main Activity: jakhar.aseem.diva.MainActivity, Target SDK: 23, Min SDK: 15, Max SDK: [button], Android Version Name: 1.0, Android Version Code: 1.
- CODE NATURE:** Native: False, Dynamic: False, Reflection: True, Crypto: True.
- DISTRIBUTION:** A donut chart showing a small green slice and a larger red slice.
- DECOMPILE & DISASSEMBLE:** Buttons for 'View Java', 'Download Java', 'View Smali', and 'Download Smali'.

Herramientas Automatizadas

DroidBox



- **Arrancando la aplicación**
./startmenu.sh nombreAVD
./droidbox.sh APK segundos
- **¿Qué tenemos que mirar?**
 - **Hashes para analizar APK**
 - **Datos entrada/salida**
 - **Operaciones lectura/escritura**
 - **Servicios iniciados**
 - **Fugas de información**
 - **Etc**

```

[Info]
-----
File name: /home/tamer/Desktop/SVT/Shazam_3.10.0-BB76011.apk
MD5: f6d3bce7761a91f8b4914f1446e0382b
SHA1: 86e7519483c05f68f27ff9bfa8f843d1a35a3350
SHA256: 1cb8c4f270c98c3685d851b219176fc8fe84a41b90ff5b75300dbb35
f00604c
Duration: 100.583075047s

[File activities]
```

Resumen Metodología

Tres fases



▪ **Recogida de información**

- 1. Fabricante en Google Play y ver otras posibles Apps del mismo.**
- 2. Instalar la App y manejar todas las posibles Actividades.**
- 3. Identificar los posibles puntos de entrada.**
- 4. Recursos que utiliza o necesita (GPS, Internet...)**
- 5. Posible framework de desarrollo.**
- 6. Intentar localizar versiones anteriores (apkpure).**
- 7. Tipo de App: Nativa, Web o Híbrida.**
- 8. Identificar posibles intenciones a las que llama (tlf, otras Apps...)**
- 9. TCPdump activo para una primera fase de análisis.**

NOTA: Tener la check-list delante para ir rellenando toda la info.

Resumen Metodología

Tres fases



▪ **Análisis Estático**

- 1. Llevar a cabo el proceso con APKtool o similares para obtener AndroidManifest.xml**
- 2. Buscar información de la App para ver si es depurable, si tiene actividades, content provider, etc exported=true.**
- 3. Obtener el nombre del paquete y del launcher (actividad principal).**
- 4. Identificar servicios que pueden correr en paralelo.**
- 5. Permisos de la aplicación (¿alguno que no corresponda con el uso de la misma?). ¿Necesita Root? ¿Por qué?**
- 6. Versión de la API y posible metadatos sobre el framework que emplea. Lo cual puede llevarnos a información sobre el sistema donde se encuentre puede ser vulnerable.**
- 7. Información hardcodeada: API-Keys, Secret-keys, password, etc.**
- 8. ¿Cómo almacena los datos? (imprescindible hacer un buen testing en la fase 1).**
- 9. Otras fallos como posible detección de SQLi u otras vulnerabilidades vista en el código fuente.**
- 10. Identificación de socket de conexión en código fuente.**
- 11. Búsqueda de cadenas URIS o string con información valiosa.**

NOTA: Tener la check-list delante para ir rellenando toda la info.

Resumen Metodología

Tres fases



- **Análisis Dinámico**
 - 1. Análisis de puntos de entrada con TCPdump y con Proxy Inverso.**
 - 2. Monitoreo del log del sistema (pidcat) → logcat.x (E I D W)**
 - 3. Información que se almacena en el terminal después de interactuar con la aplicación (databases, shared_preferences).**
 - 4. Análisis dinámico y posibles llamadas a intenciones, servicios u otros recursos (DroidBox).**

NOTA: Tener la check-list delante para ir rellenando toda la info.



Gracias por
su atención



GOBIERNO
DE ESPAÑA

MINISTERIO
DE INDUSTRIA, ENERGÍA
Y TURISMO

