Intro to Android Assessment in No Time

@warlockk87

FARADAY

Quien soy?

Security Researcher para Infobyte Security

Pentester de WebApps y aplicaciones Mobile

Desarrollador Java

Aficionado a los CTF

Aficionado a los tabletop RPGs

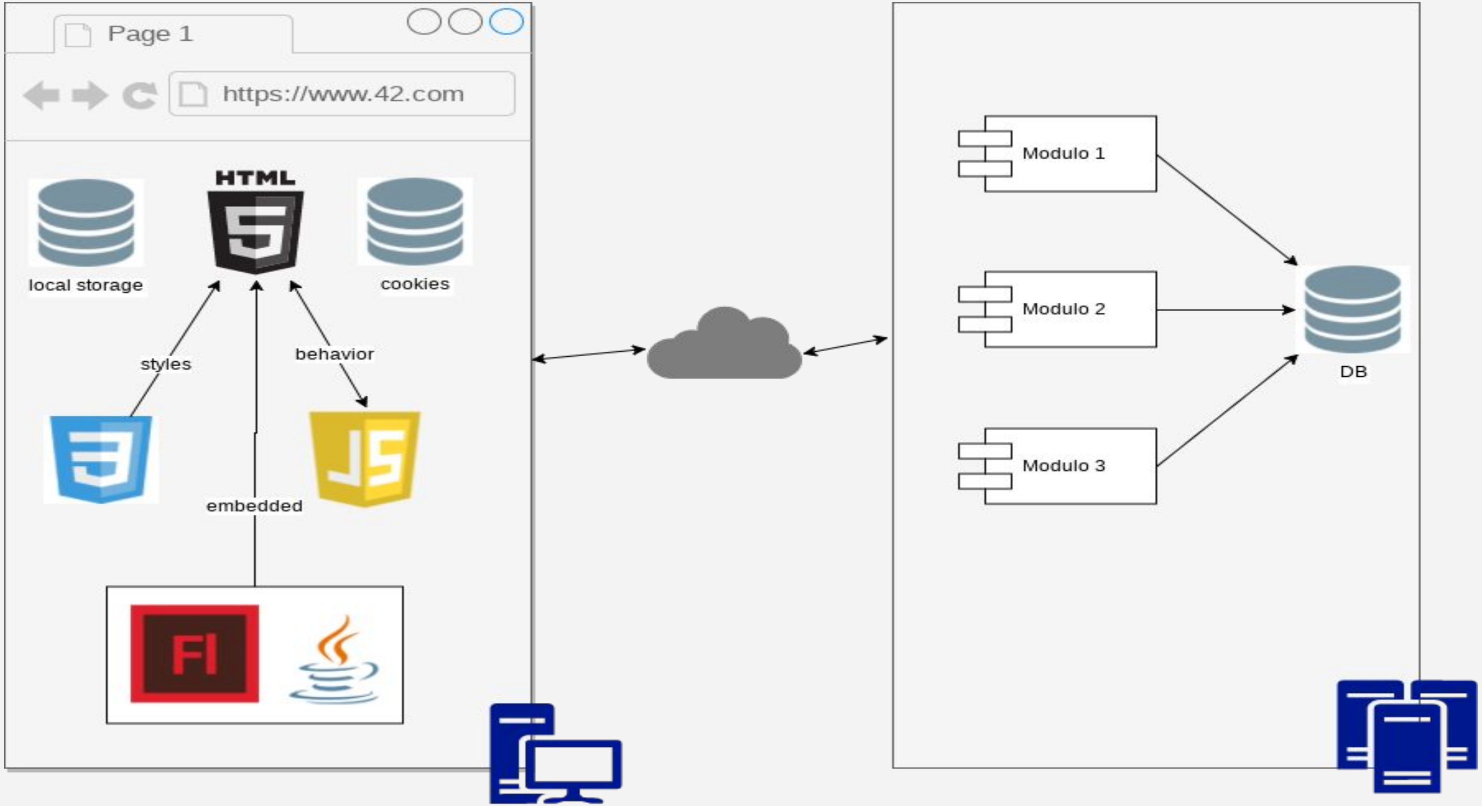
Fanatico de League Of Legends

Tendras este pentest para ayer?

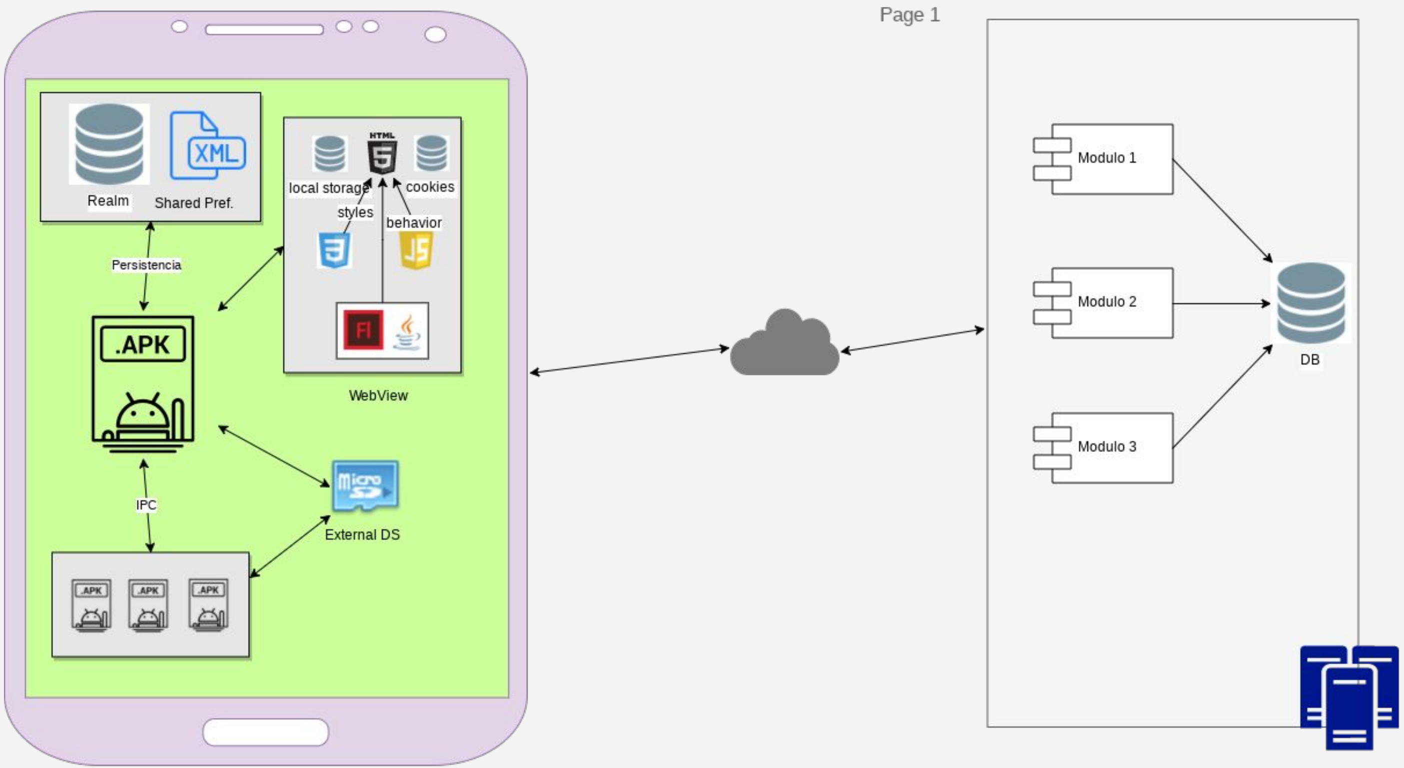
Hoja de ruta

- Estructura de aplicativo Android y ecosistema
- Pasos para la instalación de aplicativo
- Puesta a punto de entorno para hacer pruebas
- Pasos generales de pentesting en Android
- Recomendaciones durante la ejecución de un pentest
- Conclusiones

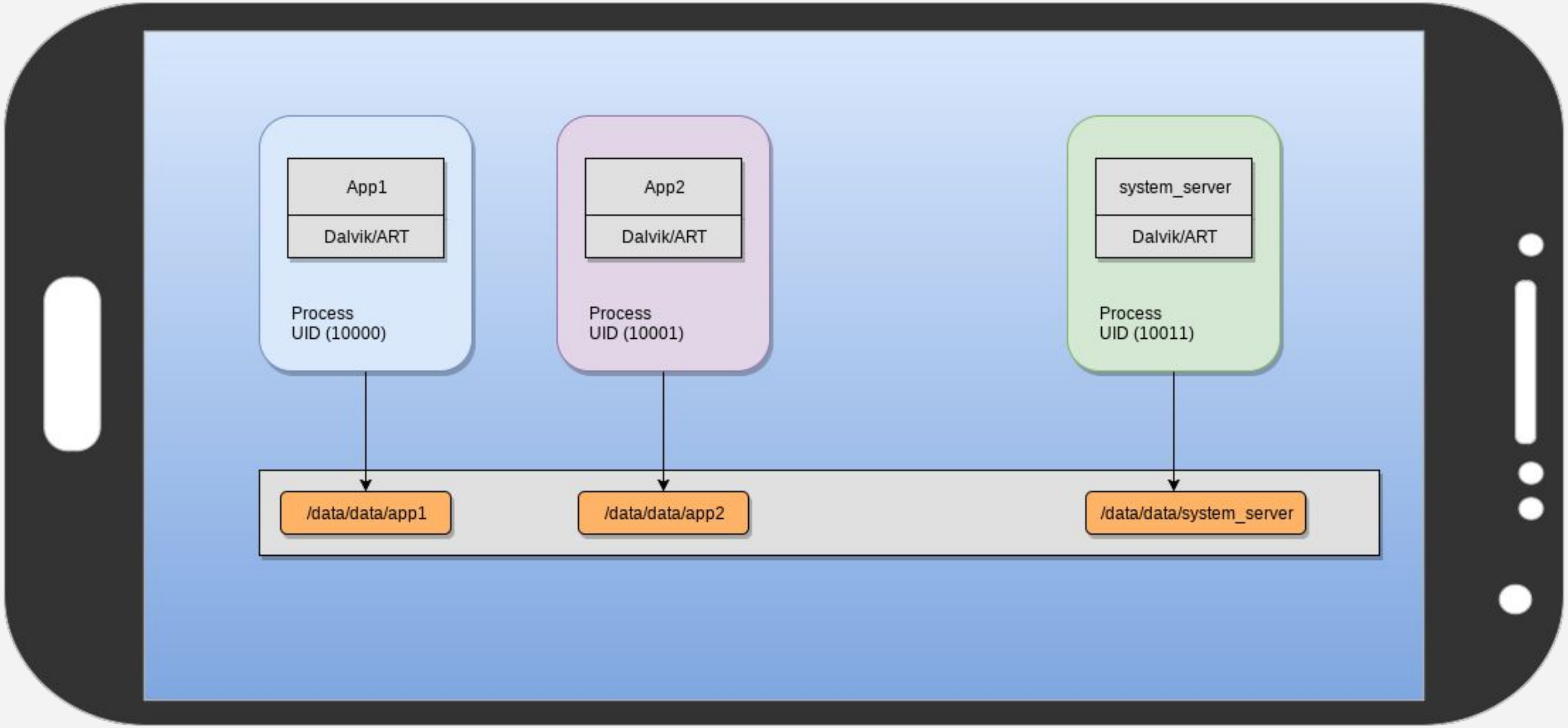
Estructura aplicaciones web



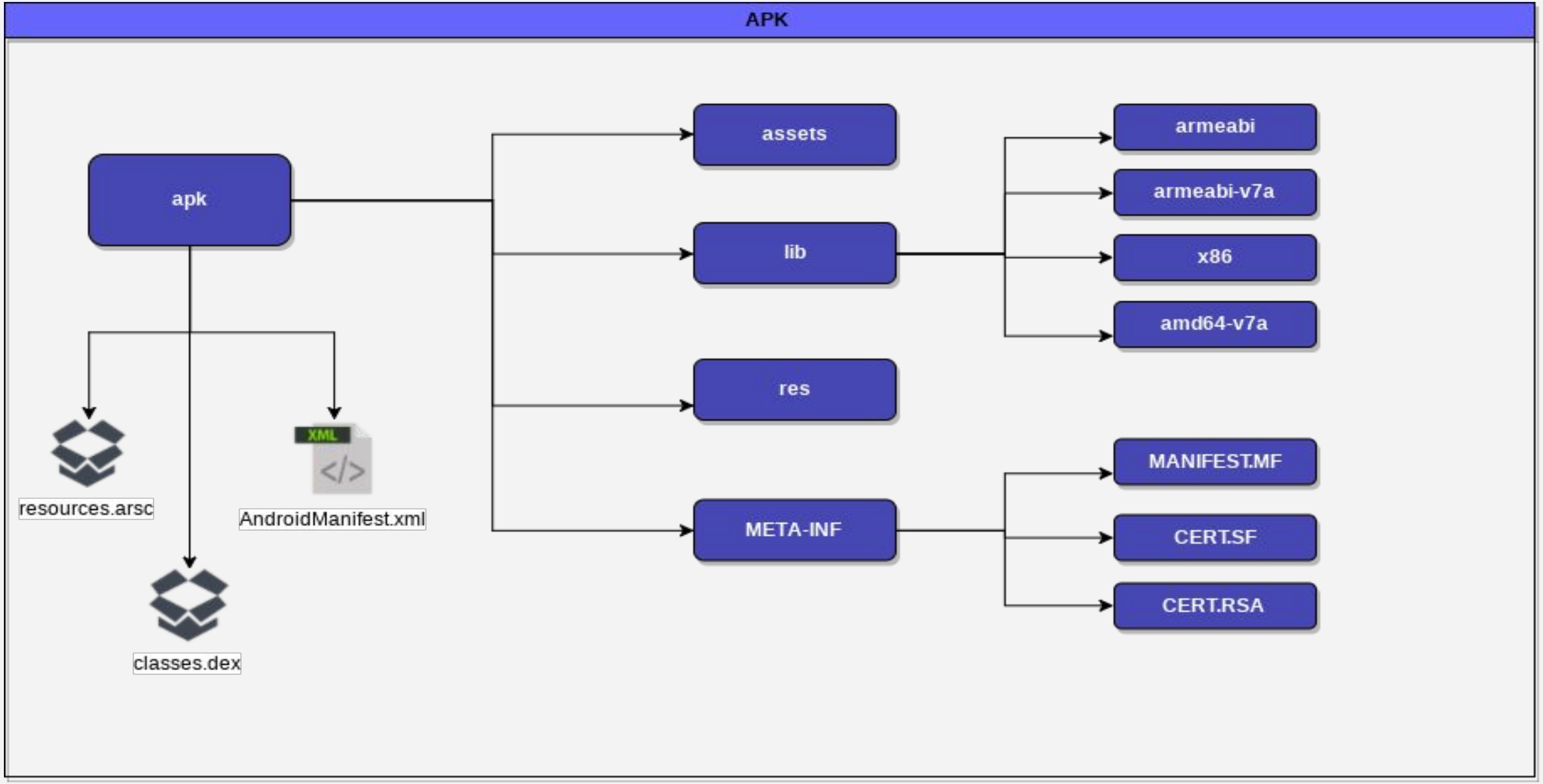
Estructura aplicaciones mobile



Android sandboxing

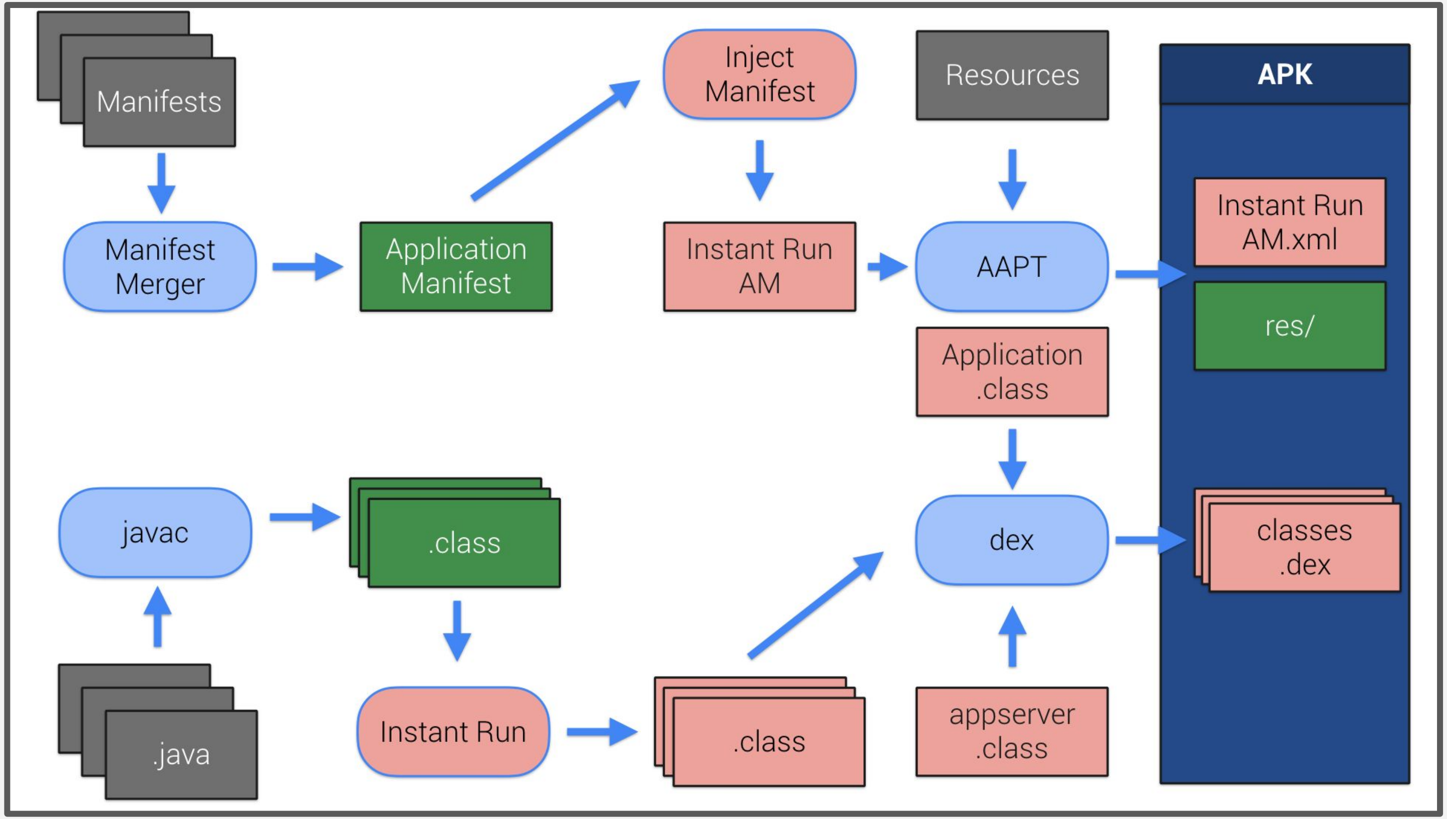


Estructura de un APK

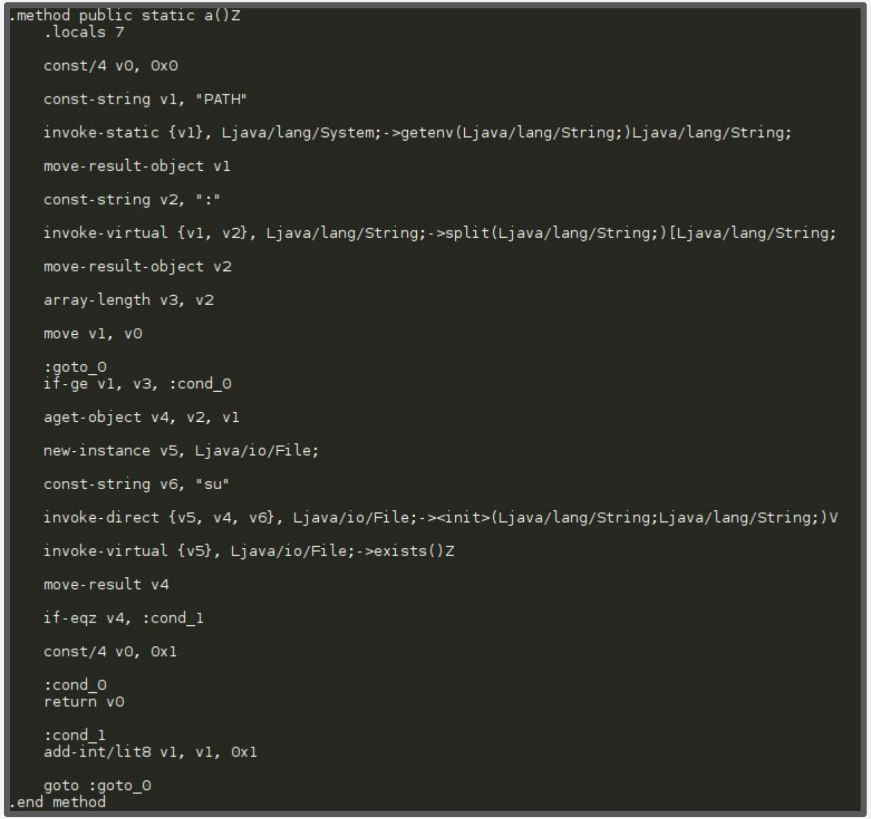


Estructura de un APK

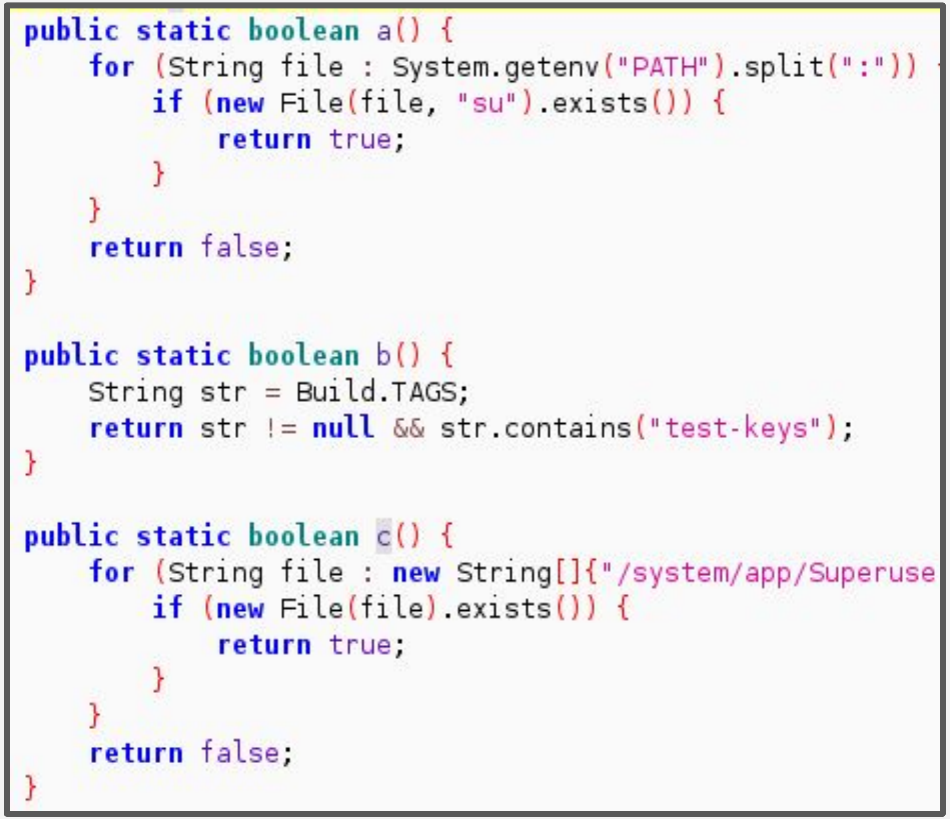
FARADAY



Bypass de control de rooteo



```
.method public static a()Z
    .locals 7

    const/4 v0, 0x0

    const-string v1, "PATH"

    invoke-static {v1}, Ljava/lang/System;->getenv(Ljava/lang/String;)Ljava/lang/String;

    move-result-object v1

    const-string v2, ":"

    invoke-virtual {v1, v2}, Ljava/lang/String;->split(Ljava/lang/String;)[Ljava/lang/String;

    move-result-object v2

    array-length v3, v2

    move v1, v0

    :goto_0
    if-ge v1, v3, :cond_0

    aget-object v4, v2, v1

    new-instance v5, Ljava/io/File;

    const-string v6, "su"

    invoke-direct {v5, v4, v6}, Ljava/io/File;-><init>(Ljava/lang/String;Ljava/lang/String;)V

    invoke-virtual {v5}, Ljava/io/File;->exists()Z

    move-result v4

    if-eqz v4, :cond_1

    const/4 v0, 0x1

    :cond_0
    return v0

    :cond_1
    add-int/lit8 v1, v1, 0x1

    goto :goto_0
.end method
```



```
public static boolean a() {
    for (String file : System.getenv("PATH").split(":"))
        if (new File(file, "su").exists()) {
            return true;
        }
    }
    return false;
}

public static boolean b() {
    String str = Build.TAGS;
    return str != null && str.contains("test-keys");
}

public static boolean c() {
    for (String file : new String[]{"/system/app/Superuse
        if (new File(file).exists()) {
            return true;
        }
    }
    return false;
}
```

Arsenal

- Adb (android device bridge)
- Android Studio (opcional)
- Emulador (genymotion / avd / ISO VirtualBox - VMware) o Celular
- Jadx-gui o dex2jar + jd-gui
- Apktool
- Jarsigner
- jdb
- BurpSuite / ZAP
- Wireshark
- Analizadores estaticos (MobSF / Androbugs / QARK / JAADAS)
- Frida
- Drozer
- XPosed (rooteo requerido)

Arsenal

- **Adb (android device bridge)**
- Android Studio (opcional)
- **Emulador (genymotion / avd / ISO VirtualBox - VMware) o Celular**
- **Jadx-gui** o dex2jar + jd-gui
- Apktool
- Jarsigner
- jdb
- **BurpSuite / ZAP**
- Wireshark
- Analizadores estaticos (MobSF / Androbugs / QARK / JAADAS)
- Frida
- **Drozer**
- XPosed (rooteo requerido)

ANDROID

Conseguir e instalar el APK

a. Descargar de Google Play

   i. https://apkpure.com/

   ii. https://apps.evozi.com/apk-downloader/

b. Instalar desde PC

   i. adb install com.example.apk

c. Conseguir APK de celular (USB modo debug)

   i. adb shell pm list packages

   ii. adb shell path com.fitstart.pt

   iii. adb pull /data/app/com.fitstar.pt-1/base.apk /path/destino

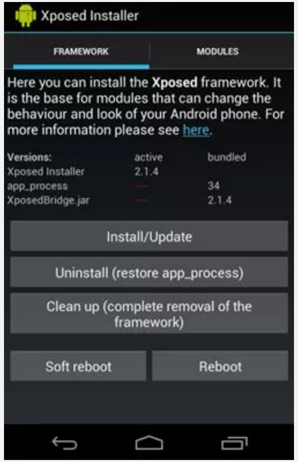
Detectando dispositivos rooteados

a. Existencia de paquetes o archivos particulares como
    i. /system/app/Superuser.apk
    ii. eu.chainfire.supersu
b. Existencia de “su”
    i. Buscar en directorios (/sbin/su, /system/su, etc)
    ii. Ejecutar mediante Runtime.getRuntime().exec()
c. Revisar los procesos que corren en /proc
d. Ver permisos de diferentes directorios
e. etc...

Bypass de control de rooteo

FARADAY

1. Usar XPosed Framework
2. Instalar RootCloak
3. Habilitar la app en RootCloak
4. Volver a abrir la app


Xposed Installer

FRAMEWORK | MODULES

Here you can install the **Xposed** framework. It is the base for modules that can change the behaviour and look of your Android phone. For more information please see here.

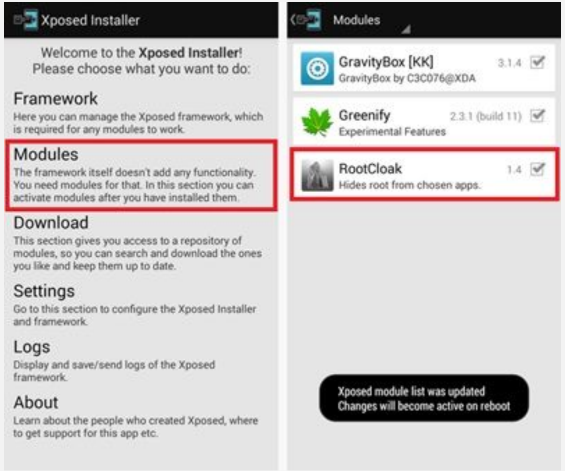
Versions:	active	bundled
Xposed Installer	2.1.4	
app_process	---	34
XposedBridge.jar	---	2.1.4

Install/Update

Uninstall (restore app_process)

Clean up (complete removal of the framework)

Soft reboot | Reboot


Xposed Installer

Welcome to the **Xposed Installer**! Please choose what you want to do:

Framework
Here you can manage the Xposed framework, which is required for any modules to work.

Modules
The framework itself doesn't add any functionality. You need modules for that. In this section you can activate modules after you have installed them.

Download
This section gives you access to a repository of modules, so you can search and download the ones you like and keep them up to date.

Settings
Go to this section to configure the Xposed Installer and framework.

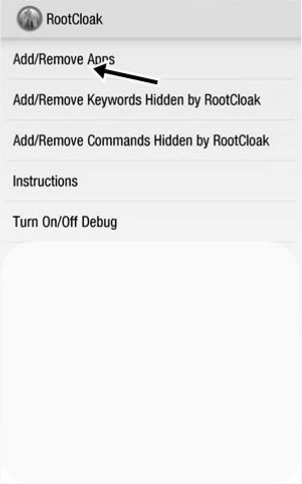
Logs
Display and save/send logs of the Xposed framework.

About
Learn about the people who created Xposed, where to get support for this app etc.

Modules

GravityBox [KK] GravityBox by C3C076@XDA	3.1.4	☑
Greenify Experimental Features	2.3.1 (build 11)	☑
RootCloak Hides root from chosen apps.	1.4	☑

Xposed module list was updated
Changes will become active on reboot


RootCloak

Add/Remove Apps

Add/Remove Keywords Hidden by RootCloak

Add/Remove Commands Hidden by RootCloak

Instructions

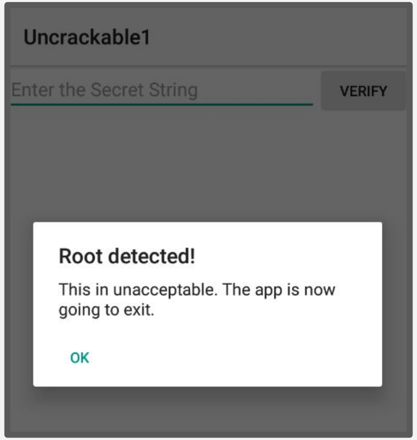
Turn On/Off Debug

Bypass de control de rooteo

a. Decompilar el apk con jadx-gui

b. Buscar el código o mensaje de error

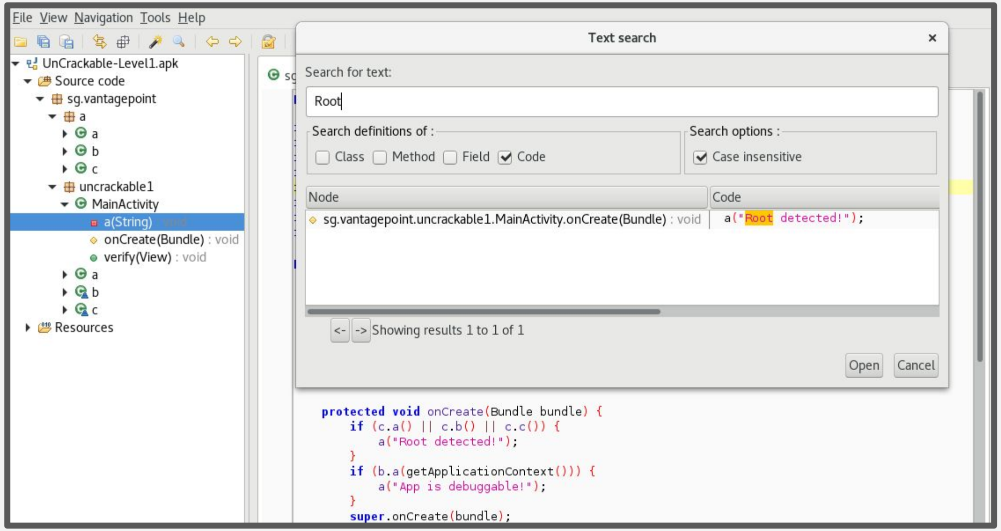
c. Ver desde donde se llama



Root detected!

This in unacceptable. The app is now going to exit.

OK



Search for text: Root

Search definitions of : □ Class □ Method □ Field ☑ Code

Search options : ☑ Case insensitive

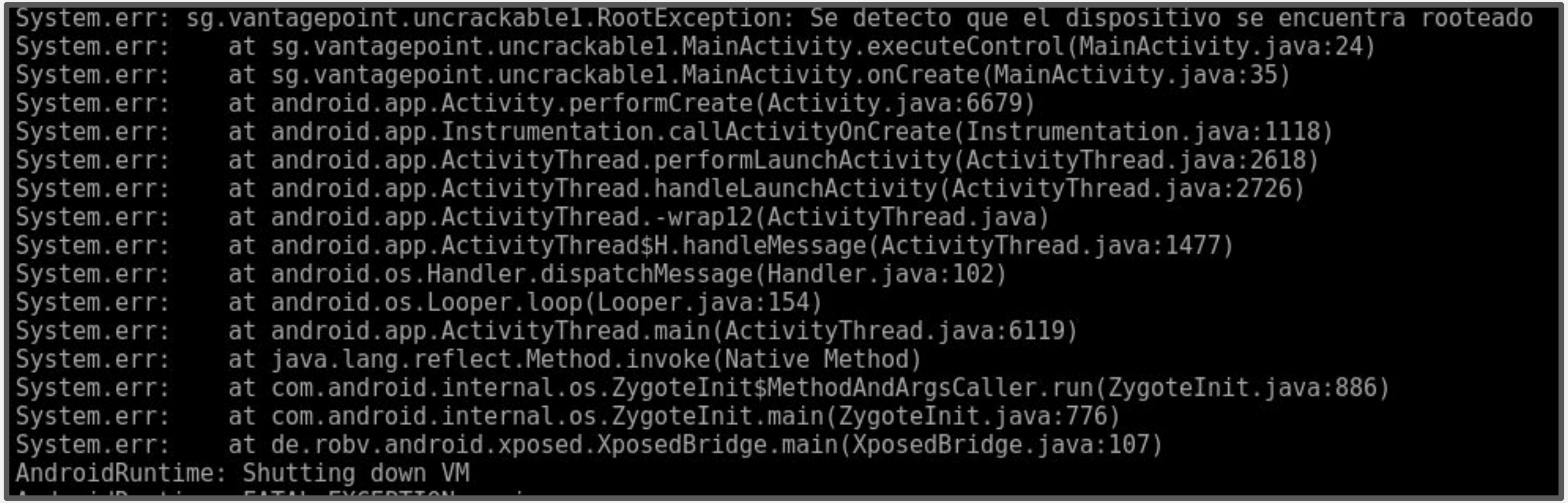
Node	Code
sg.vantagepoint.uncrackable1.MainActivity.onCreate(Bundle) : void	a("Root detected!");

<- -> Showing results 1 to 1 of 1

```
protected void onCreate(Bundle bundle) {
    if (c.a() || c.b() || c.c()) {
        a("Root detected!");
    }
    if (b.a(getApplicationContext())) {
        a("App is debuggable!");
    }
    super.onCreate(bundle);
```

Bypass de control de rooteo

Revisar los logs con adb logcat, y buscar la excepción



```
System.err: sg.vantagepoint.uncrackable1.RootException: Se detecto que el dispositivo se encuentra rooteado
System.err:     at sg.vantagepoint.uncrackable1.MainActivity.executeControl(MainActivity.java:24)
System.err:     at sg.vantagepoint.uncrackable1.MainActivity.onCreate(MainActivity.java:35)
System.err:     at android.app.Activity.performCreate(Activity.java:6679)
System.err:     at android.app.Instrumentation.callActivityOnCreate(Instrumentation.java:1118)
System.err:     at android.app.ActivityThread.performLaunchActivity(ActivityThread.java:2618)
System.err:     at android.app.ActivityThread.handleLaunchActivity(ActivityThread.java:2726)
System.err:     at android.app.ActivityThread.-wrap12(ActivityThread.java)
System.err:     at android.app.ActivityThread$H.handleMessage(ActivityThread.java:1477)
System.err:     at android.os.Handler.dispatchMessage(Handler.java:102)
System.err:     at android.os.Looper.loop(Looper.java:154)
System.err:     at android.app.ActivityThread.main(ActivityThread.java:6119)
System.err:     at java.lang.reflect.Method.invoke(Native Method)
System.err:     at com.android.internal.os.ZygoteInit$MethodAndArgsCaller.run(ZygoteInit.java:886)
System.err:     at com.android.internal.os.ZygoteInit.main(ZygoteInit.java:776)
System.err:     at de.robv.android.xposed.XposedBridge.main(XposedBridge.java:107)
AndroidRuntime: Shutting down VM
AndroidRuntime: FATAL EXCEPTION: main
```

Bypass de control de rooteo

a. Decompilar el apk con **apktool d <apk>.apk**

b. Buscar la clase y métodos a modificar

c. Modificar el código smali del método

d. Volver a generar el apk

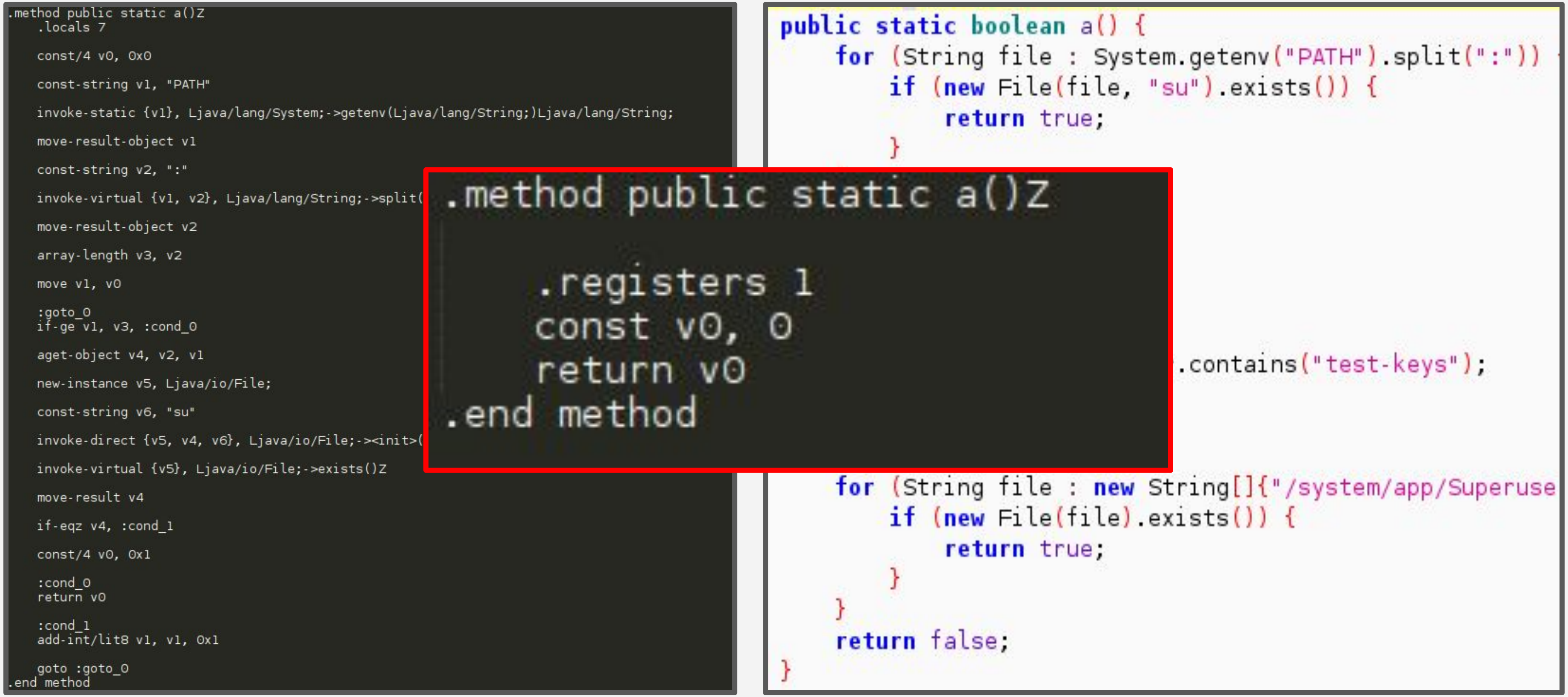
   i. apktool b <directorio de apk en smali>

   ii. keytool -genkey -v -keystore my-release-key.keystore -alias <alias> -keyalg RSA -keysize 2048 -validity 10000

   iii. jarsigner -verbose -sigalg SHA1withRSA -digestalg SHA1 -keystore my-release-key.keystore <destiny_folder>/dist/<apk>.apk <alias>

   iv. adb install <apk>.apk

Bypass de control de rooteo

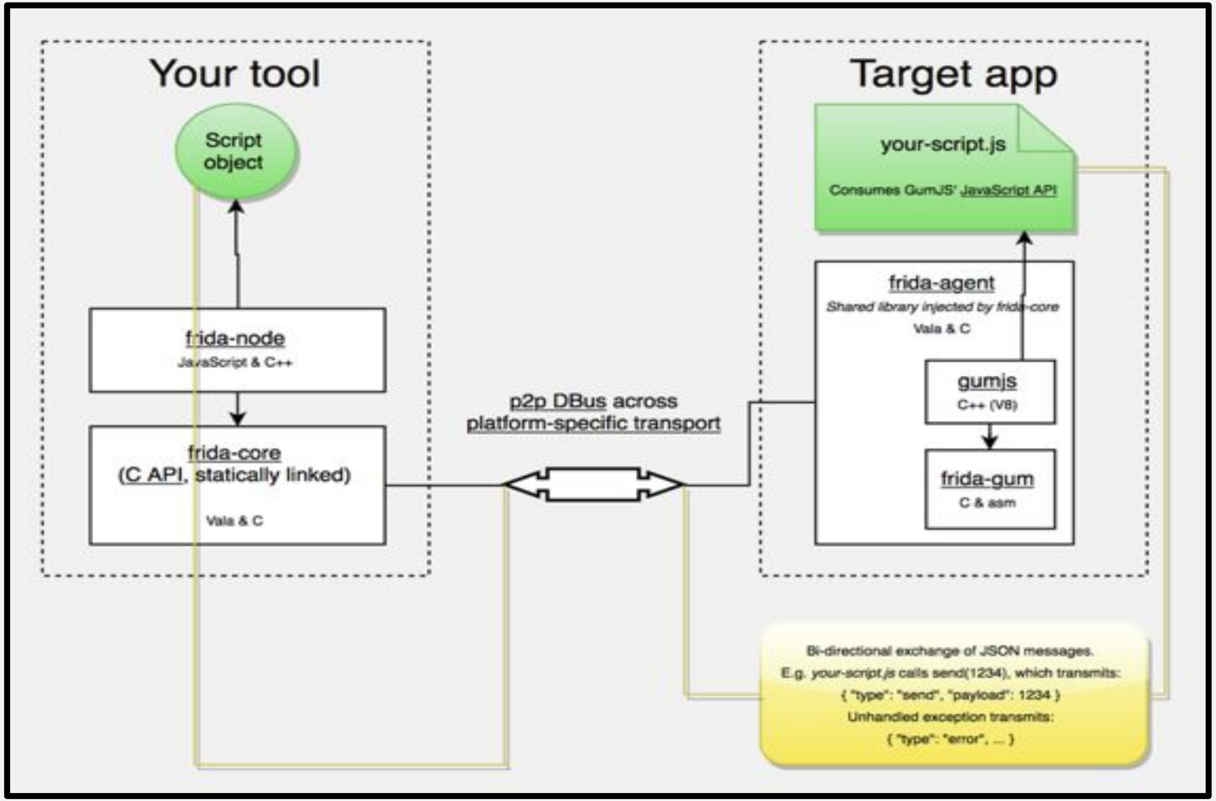


```
.method public static a()Z
    .locals 7

    const/4 v0, 0x0

    const-string v1, "PATH"

    invoke-static {v1}, Ljava/lang/System;->getenv(Ljava/lang/String;)Ljava/lang/String;

    move-result-object v1

    const-string v2, ":"

    invoke-virtual {v1, v2}, Ljava/lang/String;->split(

    move-result-object v2

    array-length v3, v2

    move v1, v0

    :goto_0
    if-ge v1, v3, :cond_0

    aget-object v4, v2, v1

    new-instance v5, Ljava/io/File;

    const-string v6, "su"

    invoke-direct {v5, v4, v6}, Ljava/io/File;-><init>(

    invoke-virtual {v5}, Ljava/io/File;->exists()Z

    move-result v4

    if-eqz v4, :cond_1

    const/4 v0, 0x1

    :cond_0
    return v0

    :cond_1
    add-int/lit8 v1, v1, 0x1

    goto :goto_0
.end method
```

```
.method public static a()Z

    .registers 1
    const v0, 0
    return v0
.end method
```

```
public static boolean a() {
    for (String file : System.getenv("PATH").split(":"))
        if (new File(file, "su").exists()) {
            return true;
        }

.contains("test-keys");

    for (String file : new String[]{"/system/app/Superuse
        if (new File(file).exists()) {
            return true;
        }
    }
    return false;
}
```

Bypass de control de rooteo con Frida



Bi-directional exchange of JSON messages.
E.g. *your-script.js* calls send(1234), which transmits:
{ "type": "send", "payload": 1234 }
Unhandled exception transmits:
{ "type": "error", ... }

Bypass de control de rooteo con Frida

a. Bajar misma version de Frida y frida-server

b. Copiar frida-server a dispositivo:

   i. adb push /home/michael/Downloads/frida-server-<ver> /data/local/tmp/frida-server

c. Llamar frida-server en dispositivo

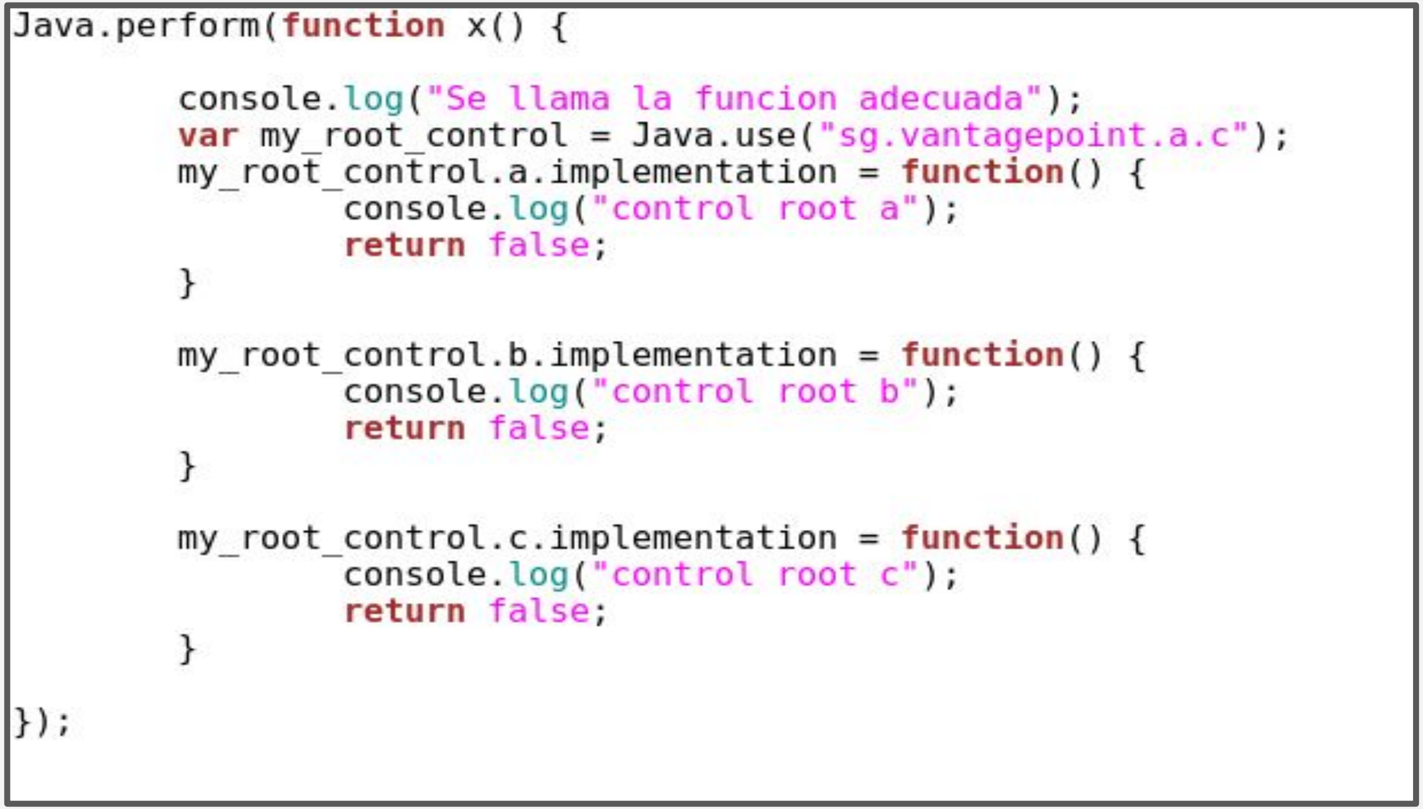
   i. /data/local/tmp/frida-server &

d. Crear Script js que modifique las funciones sensibles

e. Ejecutar script:

   i. frida -U -l script.js --no-pause <package>

Bypass de control de rooteo con Frida

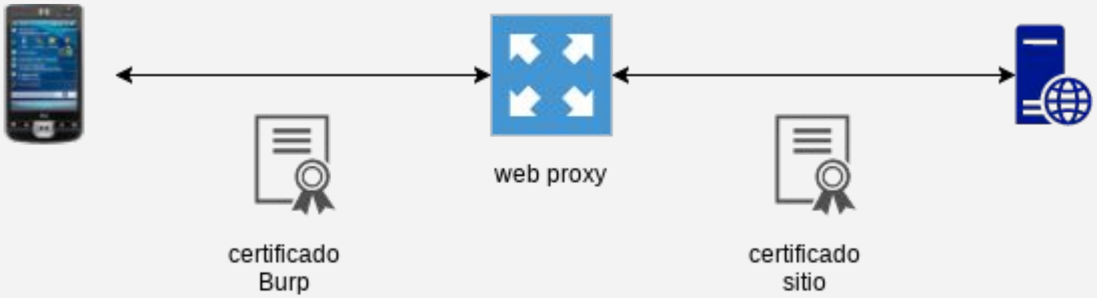


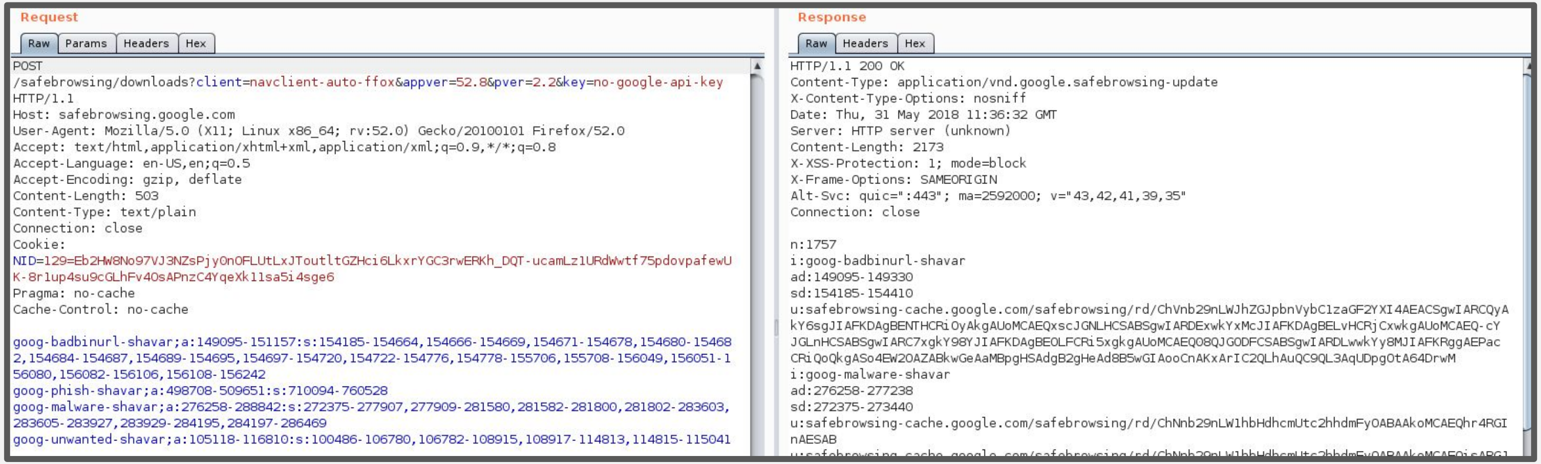
```
Java.perform(function x() {

        console.log("Se llama la funcion adecuada");
        var my_root_control = Java.use("sg.vantagepoint.a.c");
        my_root_control.a.implementation = function() {
                console.log("control root a");
                return false;
        }

        my_root_control.b.implementation = function() {
                console.log("control root b");
                return false;
        }

        my_root_control.c.implementation = function() {
                console.log("control root c");
                return false;
        }

});
```

Interceptando el tráfico

a. Ver lógica de las APIs
b. Identificar funcionamiento sin leer código.
c. Encontrar anomalías que son difíciles de detectar de otra forma.
d. Identificar información sensible.
e. Entender output e input a la aplicación (por ej. XSS)
f. Acelerar proceso de testing

Interceptando el tráfico





```
POST
/safebrowsing/downloads?client=navclient-auto-ffox&appver=52.8&pver=2.2&key=no-google-api-key
HTTP/1.1
Host: safebrowsing.google.com
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:52.0) Gecko/20100101 Firefox/52.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Length: 503
Content-Type: text/plain
Connection: close
Cookie:
NID=129=Eb2HW8No97VJ3NZsPjyOnOFLUtLxJToutltGZHci6LkxrYGC3rwERKh_DQT-ucamLz1URdWwtf75pdovpafewU
K-8r1up4su9cGLhFv40sAPnzC4YqeXk11sa5i4sge6
Pragma: no-cache
Cache-Control: no-cache

goog-badbinurl-shavar;a:149095-151157:s:154185-154664,154666-154669,154671-154678,154680-15468
2,154684-154687,154689-154695,154697-154720,154722-154776,154778-155706,155708-156049,156051-1
56080,156082-156106,156108-156242
goog-phish-shavar;a:498708-509651:s:710094-760528
goog-malware-shavar;a:276258-288842:s:272375-277907,277909-281580,281582-281800,281802-283603,
283605-283927,283929-284195,284197-286469
goog-unwanted-shavar;a:105118-116810:s:100486-106780,106782-108915,108917-114813,114815-115041
```

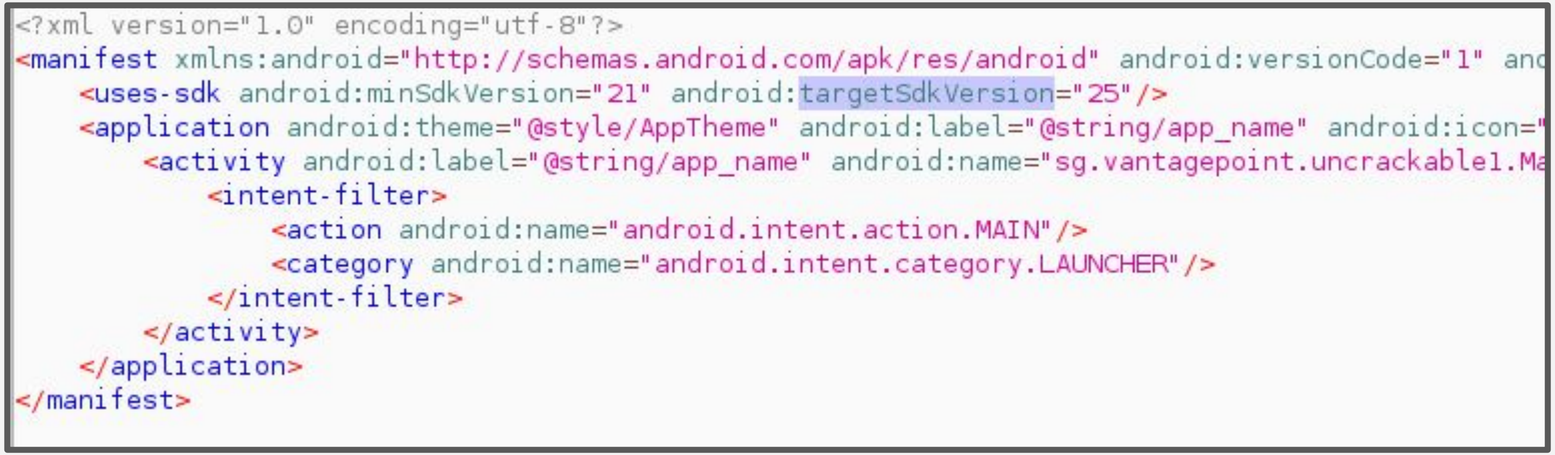
```
HTTP/1.1 200 OK
Content-Type: application/vnd.google.safebrowsing-update
X-Content-Type-Options: nosniff
Date: Thu, 31 May 2018 11:36:32 GMT
Server: HTTP server (unknown)
Content-Length: 2173
X-XSS-Protection: 1; mode=block
X-Frame-Options: SAMEORIGIN
Alt-Svc: quic=":443"; ma=2592000; v="43,42,41,39,35"
Connection: close

n:1757
i:goog-badbinurl-shavar
ad:149095-149330
sd:154185-154410
u:safebrowsing-cache.google.com/safebrowsing/rd/ChVnb29nLWJhZGJpbnVybC1zaGF2YXI4AEACSgwIARCQyA
kY6sgJIAFKDAgBENTHCRiOyAkgAUoMCAEQxscJGNLHCSABSgwIARDExwkYxMcJIAFKDAgBELvHCRjCxwkgAUoMCAEQ-cY
JGLnHCSABSgwIARC7xgkY98YJIAFKDAgBEOLFCRi5xgkgAUoMCAEQ08QJGODFCSABSgwIARDLwwkYy8MJIAFKRggAEPac
CRiQoQkgASo4EW2OAZABkwGeAaMBpgHSAdgB2gHeAd8B5wGIAooCnAKxArIC2QLhAuQC9QL3AqUDpgOtA64DrwM
i:goog-malware-shavar
ad:276258-277238
sd:272375-273440
u:safebrowsing-cache.google.com/safebrowsing/rd/ChNnb29nLW1hbHdhcmUtc2hhdmFyOABAAkoMCAEQhr4RGI
nAESAB
u:safebrowsing-cache.google.com/safebrowsing/rd/ChNnb29nLW1hbHdhcmUtc2hhdmFyOABAAkoMCAEQisARGI
```

Interceptando el tráfico

a. Configurando proxy
   i. Http OK
   ii. Https NO OK
b. Agregando certificado API <= 23
   i. Http OK
   ii. Https OK
c. Agregando certificado API > 23
   i. Http OK
   ii. Https NO OK

Interceptando tráfico con SDK > 23

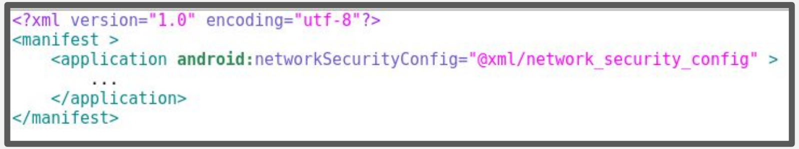
Cambiando targetSdkVersion:



```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android" android:versionCode="1" anc
    <uses-sdk android:minSdkVersion="21" android:targetSdkVersion="25"/>
    <application android:theme="@style/AppTheme" android:label="@string/app_name" android:icon="
        <activity android:label="@string/app_name" android:name="sg.vantagepoint.uncrackable1.Ma
            <intent-filter>
                <action android:name="android.intent.action.MAIN"/>
                <category android:name="android.intent.category.LAUNCHER"/>
            </intent-filter>
        </activity>
    </application>
</manifest>
```

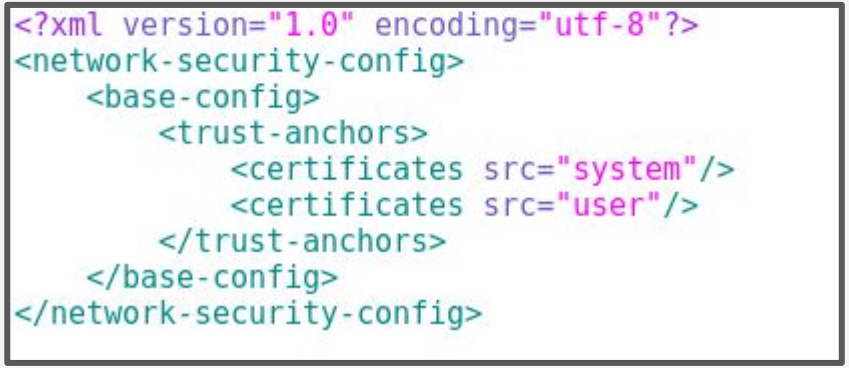
Interceptando tráfico con SDK > 23

Modificando o agregando configuración de red en AndroidManifest.xml:



```xml
<?xml version="1.0" encoding="utf-8"?>
<manifest >
    <application android:networkSecurityConfig="@xml/network_security_config" >
        ...
    </application>
</manifest>
```

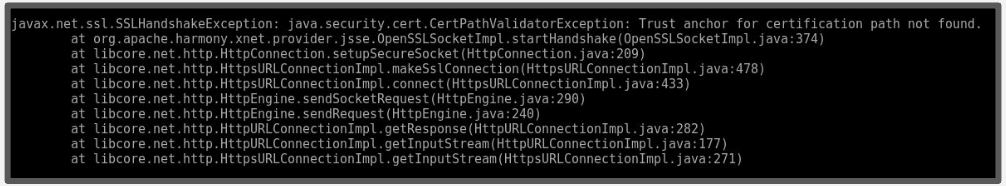
Agregando archivo res/xml/network_security_config.xml:



```xml
<?xml version="1.0" encoding="utf-8"?>
<network-security-config>
    <base-config>
        <trust-anchors>
            <certificates src="system"/>
            <certificates src="user"/>
        </trust-anchors>
    </base-config>
</network-security-config>
```

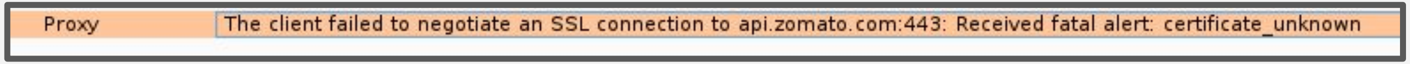
Certificate Pinning

Ver errores en el log de la siguiente forma:



```
javax.net.ssl.SSLHandshakeException: java.security.cert.CertPathValidatorException: Trust anchor for certification path not found.
        at org.apache.harmony.xnet.provider.jsse.OpenSSLSocketImpl.startHandshake(OpenSSLSocketImpl.java:374)
        at libcore.net.http.HttpConnection.setupSecureSocket(HttpConnection.java:209)
        at libcore.net.http.HttpsURLConnectionImpl.makeSslConnection(HttpsURLConnectionImpl.java:478)
        at libcore.net.http.HttpsURLConnectionImpl.connect(HttpsURLConnectionImpl.java:433)
        at libcore.net.http.HttpEngine.sendSocketRequest(HttpEngine.java:290)
        at libcore.net.http.HttpEngine.sendRequest(HttpEngine.java:240)
        at libcore.net.http.HttpURLConnectionImpl.getResponse(HttpURLConnectionImpl.java:282)
        at libcore.net.http.HttpURLConnectionImpl.getInputStream(HttpURLConnectionImpl.java:177)
        at libcore.net.http.HttpsURLConnectionImpl.getInputStream(HttpsURLConnectionImpl.java:271)
```

Ver mensajes de error en el proxy:



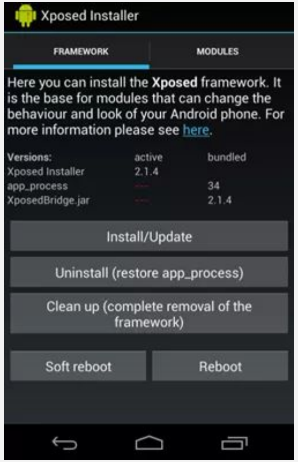
```
Proxy    The client failed to negotiate an SSL connection to api.zomato.com:443: Received fatal alert: certificate_unknown
```

Comportamiento erratico en la aplicacion como:

- Activities que no se abren
- Pantallas que se quedan en blanco
- Mensajes de error al hacer operaciones que no deberían fallar

Bypass de Certificate Pinning

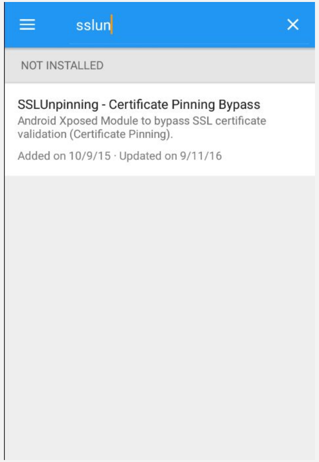
1. Usar XPosed Framework
2. Instalar SSLUnpinning/JustTrustMe
3. Habilitar la app en el nuevo módulo
4. Volver a abrir la app


Xposed Installer

Here you can install the **Xposed** framework. It is the base for modules that can change the behaviour and look of your Android phone. For more information please see here.

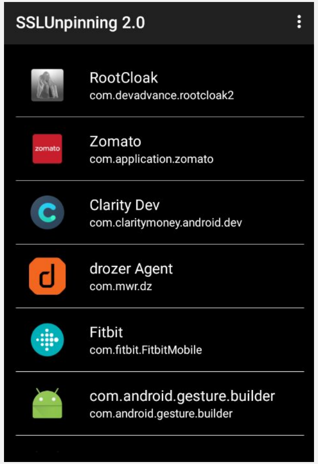
```
Versions:          active    bundled
Xposed Installer   2.1.4
app_process        ----      34
XposedBridge.jar   ----      2.1.4
```

Install/Update
Uninstall (restore app_process)
Clean up (complete removal of the framework)
Soft reboot
Reboot


sslun

NOT INSTALLED

SSLUnpinning - Certificate Pinning Bypass
Android Xposed Module to bypass SSL certificate validation (Certificate Pinning).
Added on 10/9/15 · Updated on 9/11/16


SSLUnpinning 2.0

RootCloak	com.devadvance.rootcloak2
Zomato	com.application.zomato
Clarity Dev	com.claritymoney.android.dev
drozer Agent	com.mwr.dz
Fitbit	com.fitbit.FitbitMobile
com.android.gesture.builder	com.android.gesture.builder

Bypass de Certificate Pinning

a. Buscar por palabras claves

   i. SSLSocketFactory

   ii. CertificatePinner

   iii. TrustManager

   iv. X509Certificate

   v. checkServerTrusted

   vi. X509TrustManager

   vii. sha1/

   viii. sha256/

   ix. BKS

Bypass de Certificate Pinning

a. Encontrar puntos importantes

b. Debuggear el aplicativo, poniendo breakpoint en esos puntos

   i. AndroidManifest.xml -> **android:debuggable=true**

   ii. Correr el aplicativo

   iii. adb shell ps | grep “<apk>”

   iv. adb forward tcp:9000 jdwp:<pid>

   v. jdb attach localhost:9000

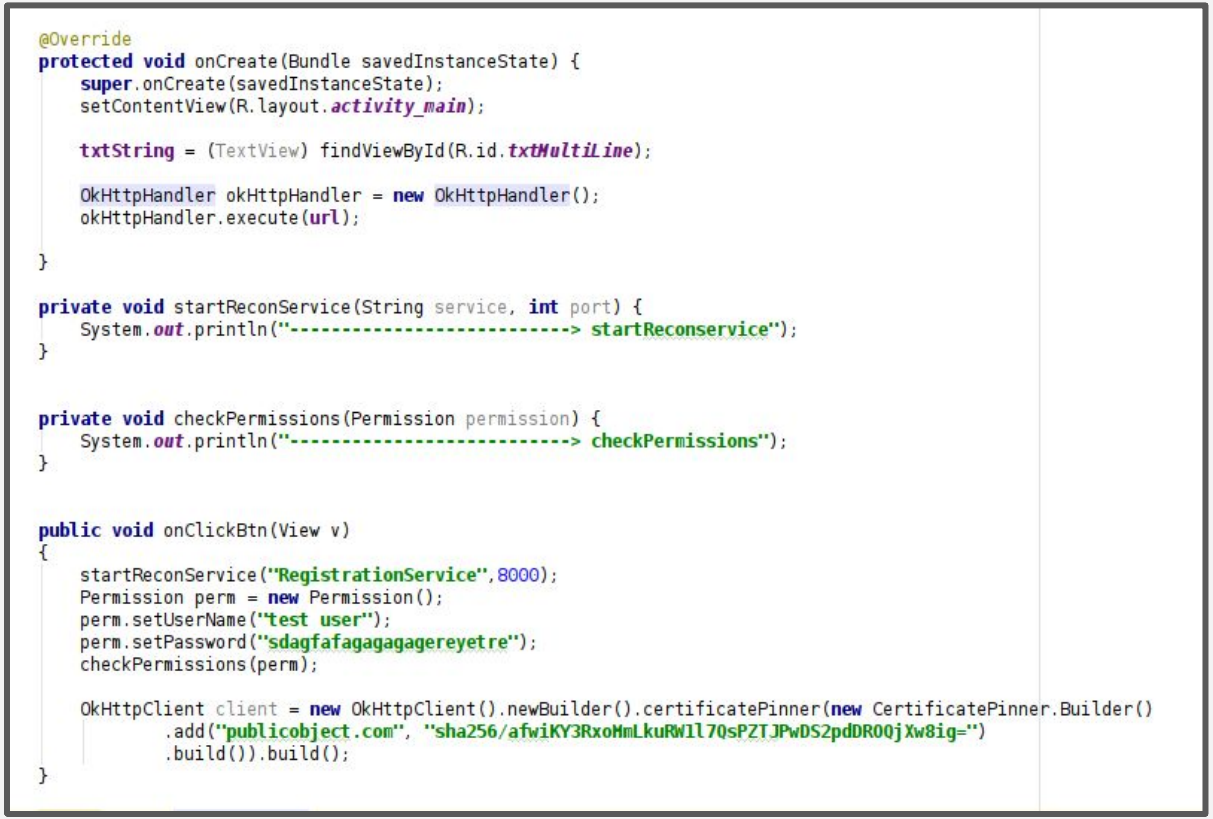
   vi. stop in <package.clase.metodo> (pe. java.lang.String.length)

   vii. debug...

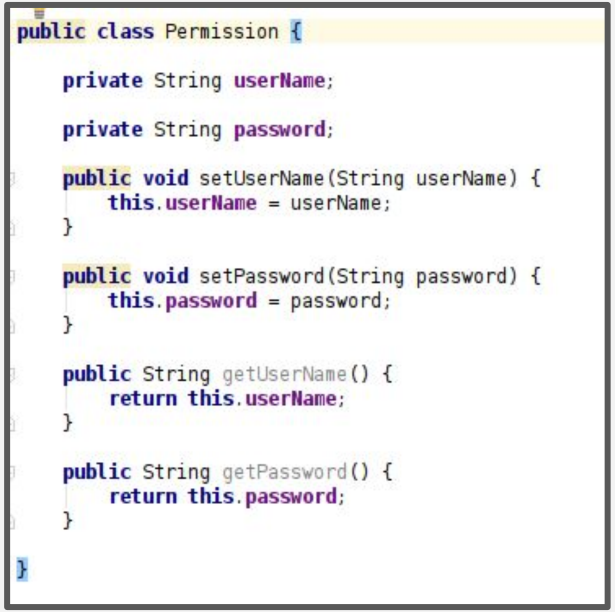
Bypass de Certificate Pinning

a. Encontrar puntos importantes
b. Agregar hooks y prints con Frida
c. Ejecutar aplicativo
d. Chequear output en consola, como para ver qué función se llama
e. Una vez analizado el código, se puede modificar la implementación como se hizo con el control antirooteo

Bypass de Certificate Pinning

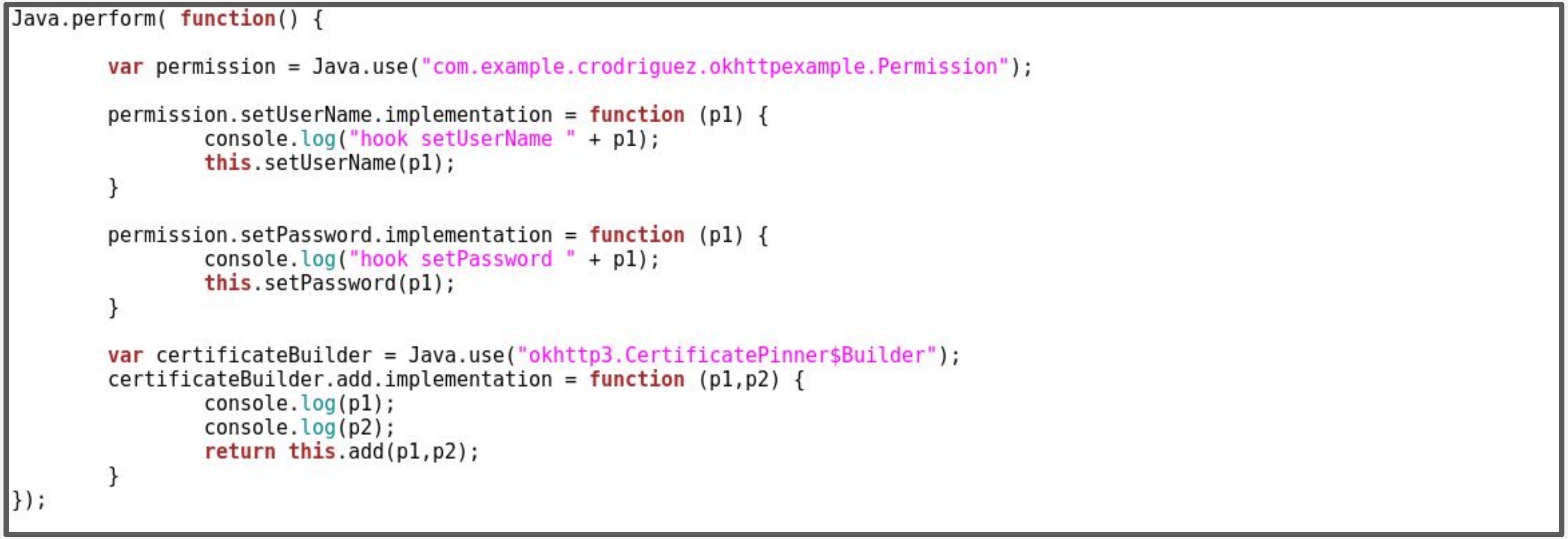


```java
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    txtString = (TextView) findViewById(R.id.txtMultiLine);

    OkHttpHandler okHttpHandler = new OkHttpHandler();
    okHttpHandler.execute(url);

}

private void startReconService(String service, int port) {
    System.out.println("---------------------------> startReconservice");
}


private void checkPermissions(Permission permission) {
    System.out.println("---------------------------> checkPermissions");
}


public void onClickBtn(View v)
{
    startReconService("RegistrationService",8000);
    Permission perm = new Permission();
    perm.setUserName("test user");
    perm.setPassword("sdagfafagagagagereyetre");
    checkPermissions(perm);

    OkHttpClient client = new OkHttpClient().newBuilder().certificatePinner(new CertificatePinner.Builder()
            .add("publicobject.com", "sha256/afwiKY3RxoMmLkuRW1l7QsPZTJPwDS2pdDROQjXw8ig=")
            .build()).build();
}
```

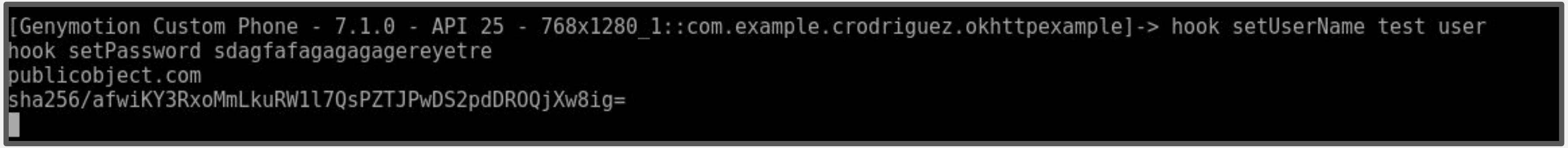


```java
public class Permission {

    private String userName;

    private String password;

    public void setUserName(String userName) {
        this.userName = userName;
    }

    public void setPassword(String password) {
        this.password = password;
    }

    public String getUserName() {
        return this.userName;
    }

    public String getPassword() {
        return this.password;
    }

}
```

Bypass de Certificate Pinning



```
Java.perform( function() {

        var permission = Java.use("com.example.crodriguez.okhttpexample.Permission");

        permission.setUserName.implementation = function (p1) {
                console.log("hook setUserName " + p1);
                this.setUserName(p1);
        }

        permission.setPassword.implementation = function (p1) {
                console.log("hook setPassword " + p1);
                this.setPassword(p1);
        }

        var certificateBuilder = Java.use("okhttp3.CertificatePinner$Builder");
        certificateBuilder.add.implementation = function (p1,p2) {
                console.log(p1);
                console.log(p2);
                return this.add(p1,p2);
        }
});
```



```
[Genymotion Custom Phone - 7.1.0 - API 25 - 768x1280_1::com.example.crodriguez.okhttpexample]-> hook setUserName test user
hook setPassword sdagfafagagagagereyetre
publicobject.com
sha256/afwiKY3RxoMmLkuRW1l7QsPZTJPwDS2pdDROQjXw8ig=
```

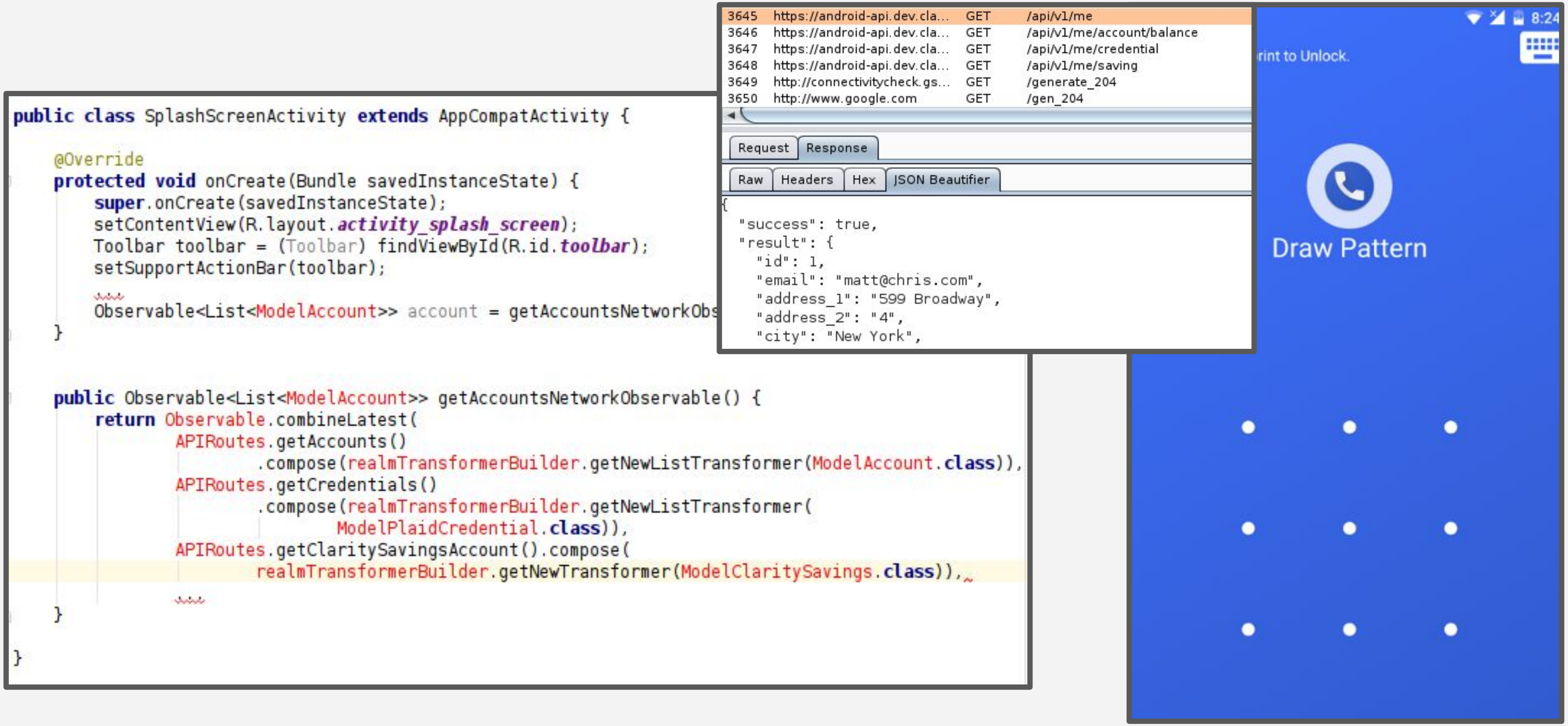
VAW-113
6J
VAW 113
P P
PR3 110 09

Metodología

a. Probar aplicativo
b. Análisis estático de aplicación
c. Probar persistencia de datos
d. Probar criptografia
e. Probar autenticación y manejo de sesiones
f. Probar seguridad en comunicaciones
g. Probar interacción con plataforma
h. Probar componentes de terceros
i. Probar fallas en lógica

Probar aplicación

FARADAY



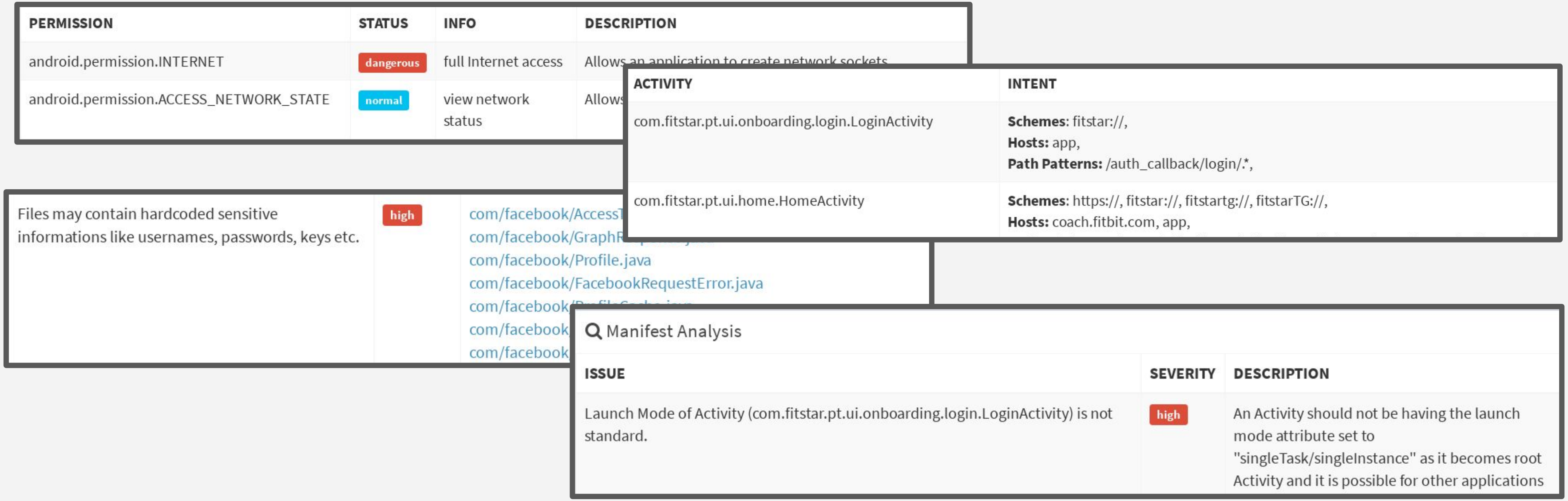
```
public class SplashScreenActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_splash_screen);
        Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);
        setSupportActionBar(toolbar);

        Observable<List<ModelAccount>> account = getAccountsNetworkObs
    }


    public Observable<List<ModelAccount>> getAccountsNetworkObservable() {
        return Observable.combineLatest(
                APIRoutes.getAccounts()
                        .compose(realmTransformerBuilder.getNewListTransformer(ModelAccount.class)),
                APIRoutes.getCredentials()
                        .compose(realmTransformerBuilder.getNewListTransformer(
                                ModelPlaidCredential.class)),
                APIRoutes.getClaritySavingsAccount().compose(
                        realmTransformerBuilder.getNewTransformer(ModelClaritySavings.class)),

    }

}
```

```
3645  https://android-api.dev.cla...  GET  /api/v1/me
3646  https://android-api.dev.cla...  GET  /api/v1/me/account/balance
3647  https://android-api.dev.cla...  GET  /api/v1/me/credential
3648  https://android-api.dev.cla...  GET  /api/v1/me/saving
3649  http://connectivitycheck.gs...  GET  /generate_204
3650  http://www.google.com           GET  /gen_204
```

```
{
  "success": true,
  "result": {
    "id": 1,
    "email": "matt@chris.com",
    "address_1": "599 Broadway",
    "address_2": "4",
    "city": "New York",
```

Análisis estático de aplicación FARADAY

Hay varias alternativas: (mobSF, JAADAS, qark)



PERMISSION	STATUS	INFO	DESCRIPTION
android.permission.INTERNET	dangerous	full Internet access	Allows an application to create network sockets
android.permission.ACCESS_NETWORK_STATE	normal	view network status	Allows

Files may contain hardcoded sensitive informations like usernames, passwords, keys etc.	high	com/facebook/AccessT com/facebook/GraphR com/facebook/Profile.java com/facebook/FacebookRequestError.java com/facebook/ com/facebook com/facebook

ACTIVITY	INTENT
com.fitstar.pt.ui.onboarding.login.LoginActivity	**Schemes**: fitstar://, **Hosts:** app, **Path Patterns:** /auth_callback/login/.*,
com.fitstar.pt.ui.home.HomeActivity	**Schemes**: https://, fitstar://, fitstartg://, fitstarTG://, **Hosts:** coach.fitbit.com, app,

Manifest Analysis

ISSUE	SEVERITY	DESCRIPTION
Launch Mode of Activity (com.fitstar.pt.ui.onboarding.login.LoginActivity) is not standard.	high	An Activity should not be having the launch mode attribute set to "singleTask/singleInstance" as it becomes root Activity and it is possible for other applications

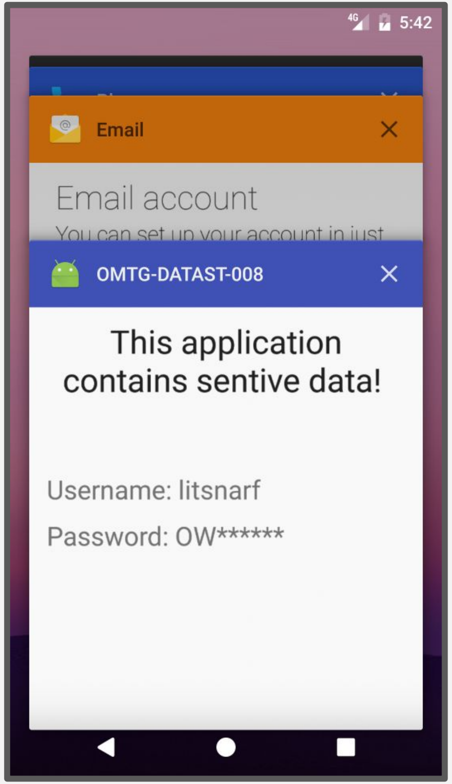
Probar persistencia de datos

Incidencias de mayor interés e impacto para el negocio

Navegando la aplicacion despues de probar funcionalidades

	Localización (por defecto)	Expuesto por defecto	allowBackup= true	WORLD_REA DABLE	Rooted	Content Provider	Services	BroadCast Receiver
SharedPreferences	/data/data/<apk>/shared_prefs	no	si	si	si	parc	parc	parc
DB sqlite	/data/data/<apk>/databases	no	si	si	si	parc	parc	parc
DB Realm	/data/data/<apk>/files	no	parc	si	si	parc	parc	parc
Archivo en storage interno	/data/data/<apk>/files	no	parc	si	si	parc	parc	parc
Archivo en storage externo	?	si	parc	si	si	parc	parc	parc
KeyStore	?	no	no	no	si	parc	parc	parc

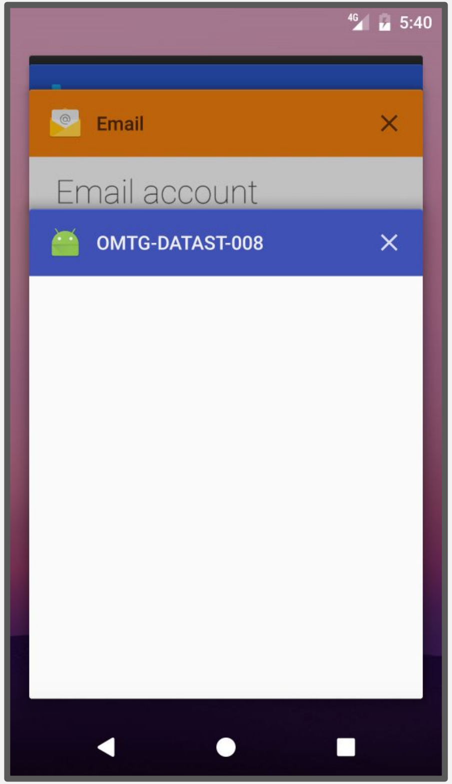
FLAG_SECURE

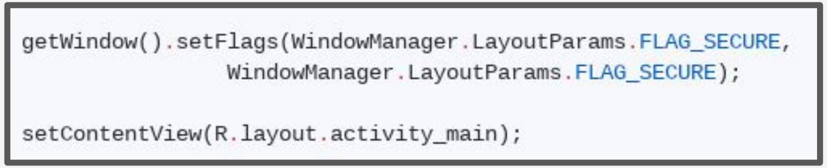


This application contains sentive data!

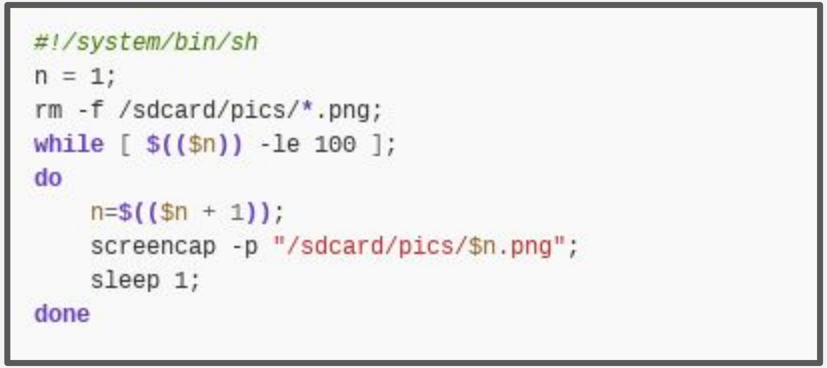
Username: litsnarf

Password: OW******



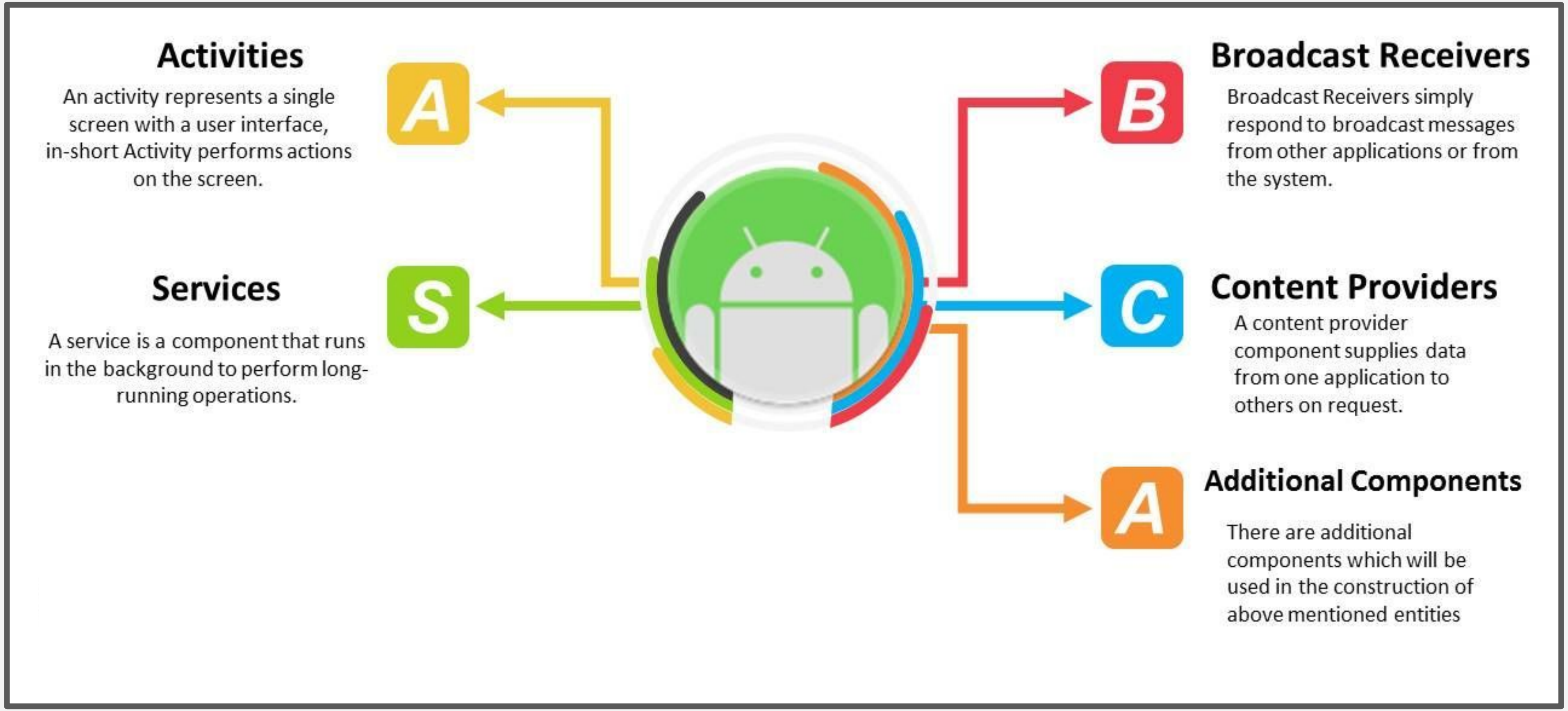


```
getWindow().setFlags(WindowManager.LayoutParams.FLAG_SECURE,
                     WindowManager.LayoutParams.FLAG_SECURE);

setContentView(R.layout.activity_main);
```



```
#!/system/bin/sh
n = 1;
rm -f /sdcard/pics/*.png;
while [ $(($n)) -le 100 ];
do
    n=$(($n + 1));
    screencap -p "/sdcard/pics/$n.png";
    sleep 1;
done
```

interacción con plataforma



Activities

An activity represents a single screen with a user interface, in-short Activity performs actions on the screen.

Services

A service is a component that runs in the background to perform long-running operations.

Broadcast Receivers

Broadcast Receivers simply respond to broadcast messages from other applications or from the system.

Content Providers

A content provider component supplies data from one application to others on request.

Additional Components

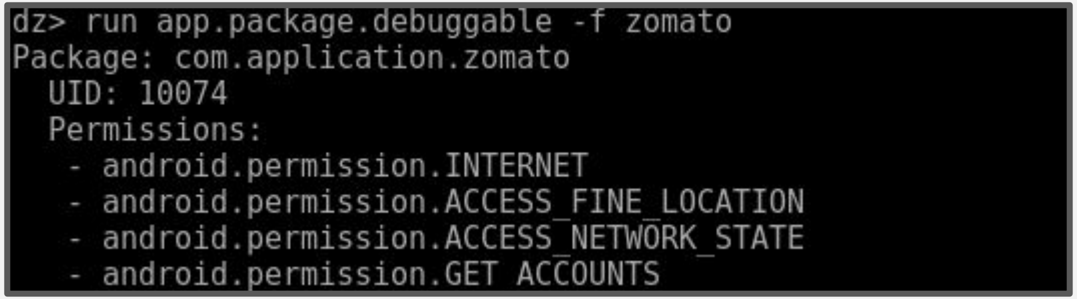
There are additional components which will be used in the construction of above mentioned entities

interacción con plataforma

Usar drozer:

run app.package.list -f <filtro>

run app.package.debuggable -f <filtro>

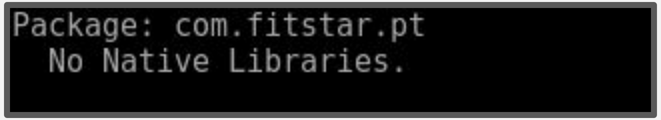


```
dz> run app.package.debuggable -f zomato
Package: com.application.zomato
  UID: 10074
  Permissions:
   - android.permission.INTERNET
   - android.permission.ACCESS_FINE_LOCATION
   - android.permission.ACCESS_NETWORK_STATE
   - android.permission.GET_ACCOUNTS
```

interacción con plataforma

run app.package.info -f <filtro>

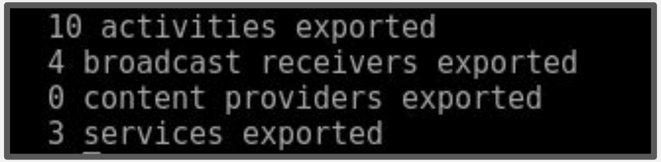
run app.package.native <app>



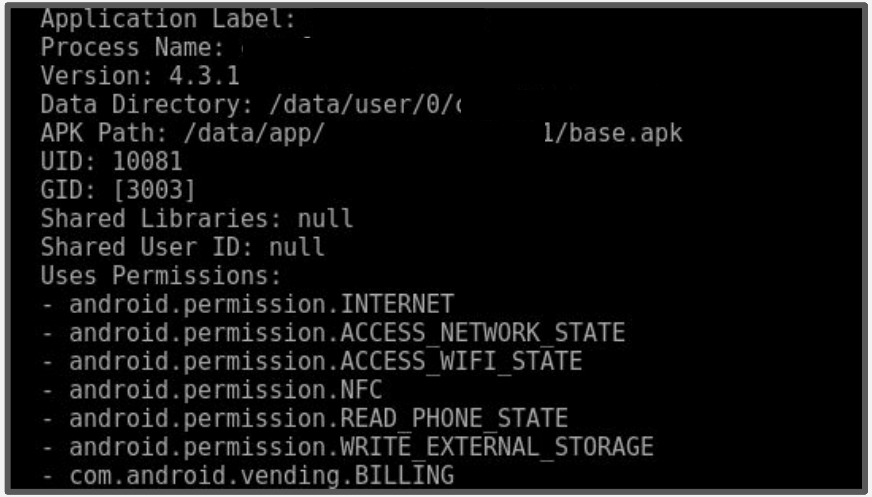
```
Package: com.fitstar.pt
  No Native Libraries.
```

run app.package.backup -f <filtro>

run app.package.attacksurface <app>

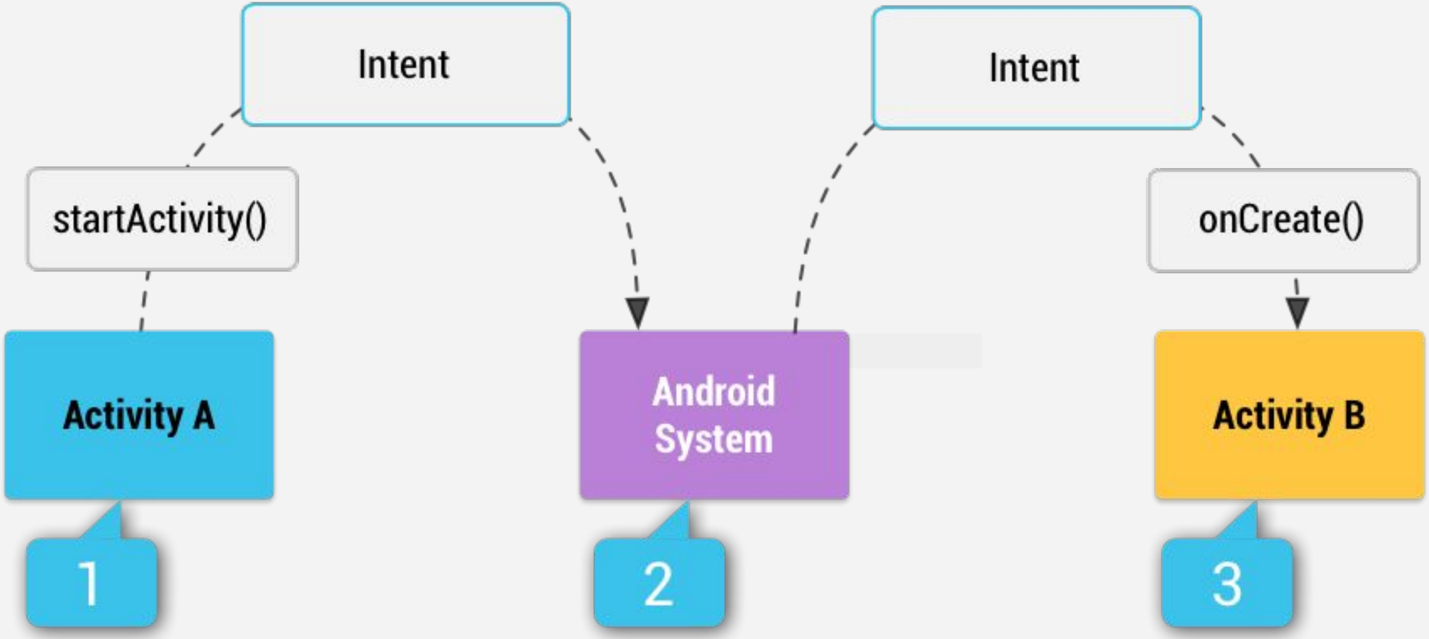


```
  10 activities exported
  4 broadcast receivers exported
  0 content providers exported
  3 services exported
```



```
Application Label: [redacted]
Process Name: [redacted]
Version: 4.3.1
Data Directory: /data/user/0/[redacted]
APK Path: /data/app/[redacted]1/base.apk
UID: 10081
GID: [3003]
Shared Libraries: null
Shared User ID: null
Uses Permissions:
- android.permission.INTERNET
- android.permission.ACCESS_NETWORK_STATE
- android.permission.ACCESS_WIFI_STATE
- android.permission.NFC
- android.permission.READ_PHONE_STATE
- android.permission.WRITE_EXTERNAL_STORAGE
- com.android.vending.BILLING
```

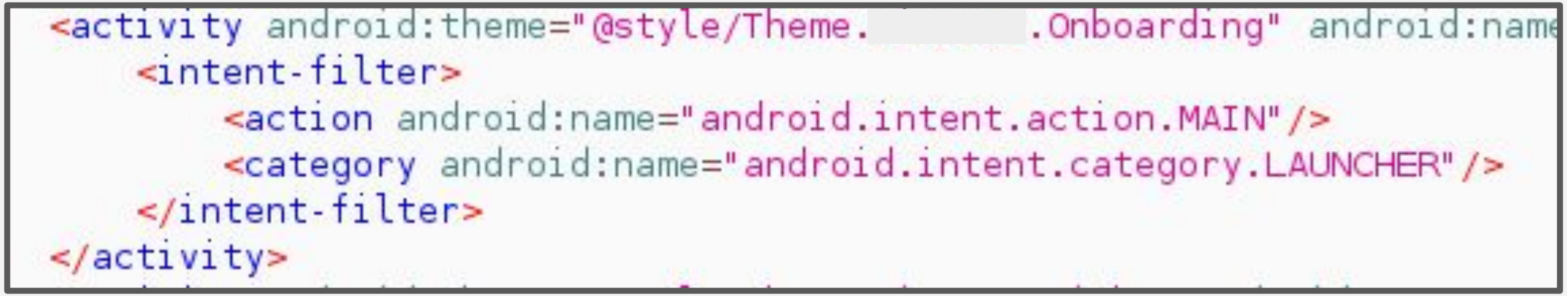
Probar activities



Probar activities

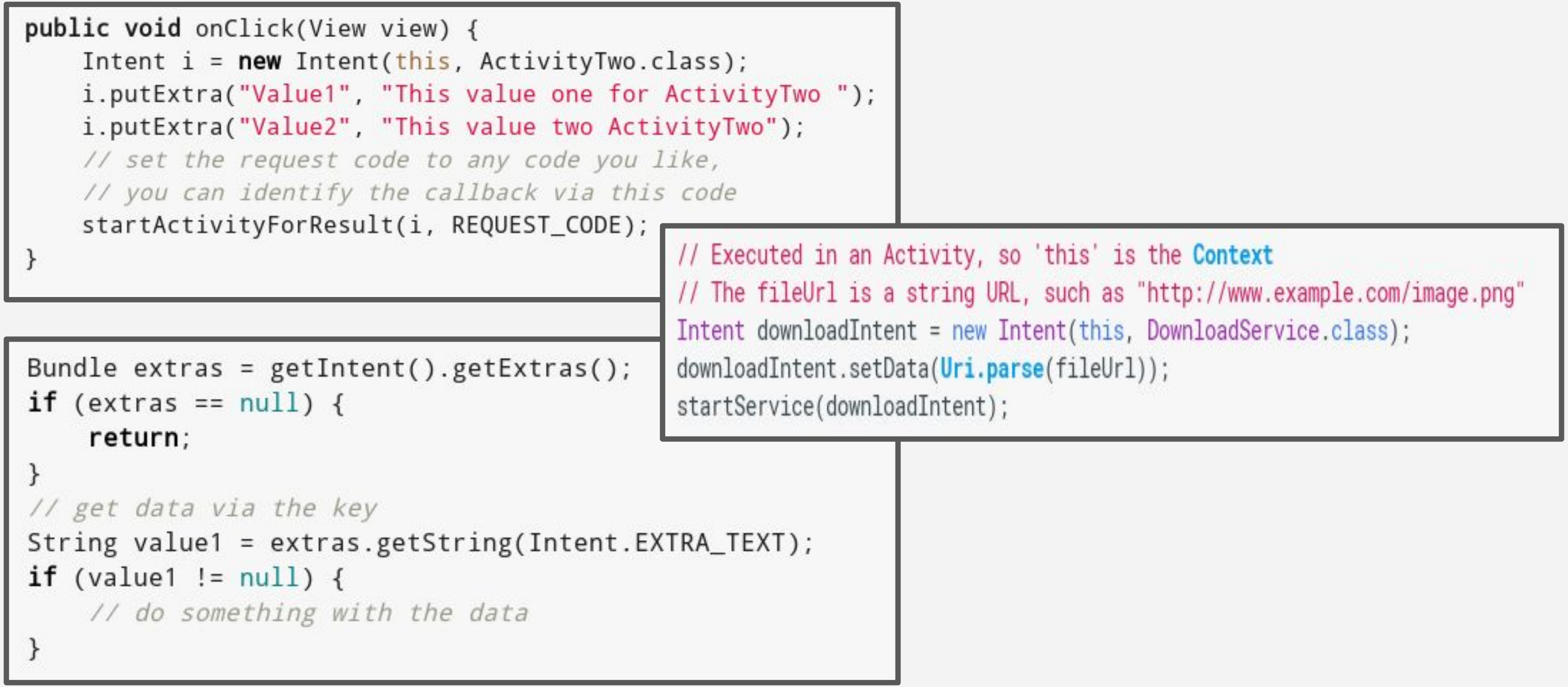
run app.activity.info -f <filtro>

run app.activity.start --component[package name] [activity name] --data-uri [data-uri] --extra [type name value] --extra [type name value]



```
<activity android:theme="@style/Theme.[redacted].Onboarding" android:name
    <intent-filter>
        <action android:name="android.intent.action.MAIN"/>
        <category android:name="android.intent.category.LAUNCHER"/>
    </intent-filter>
</activity>
```

Probar activities

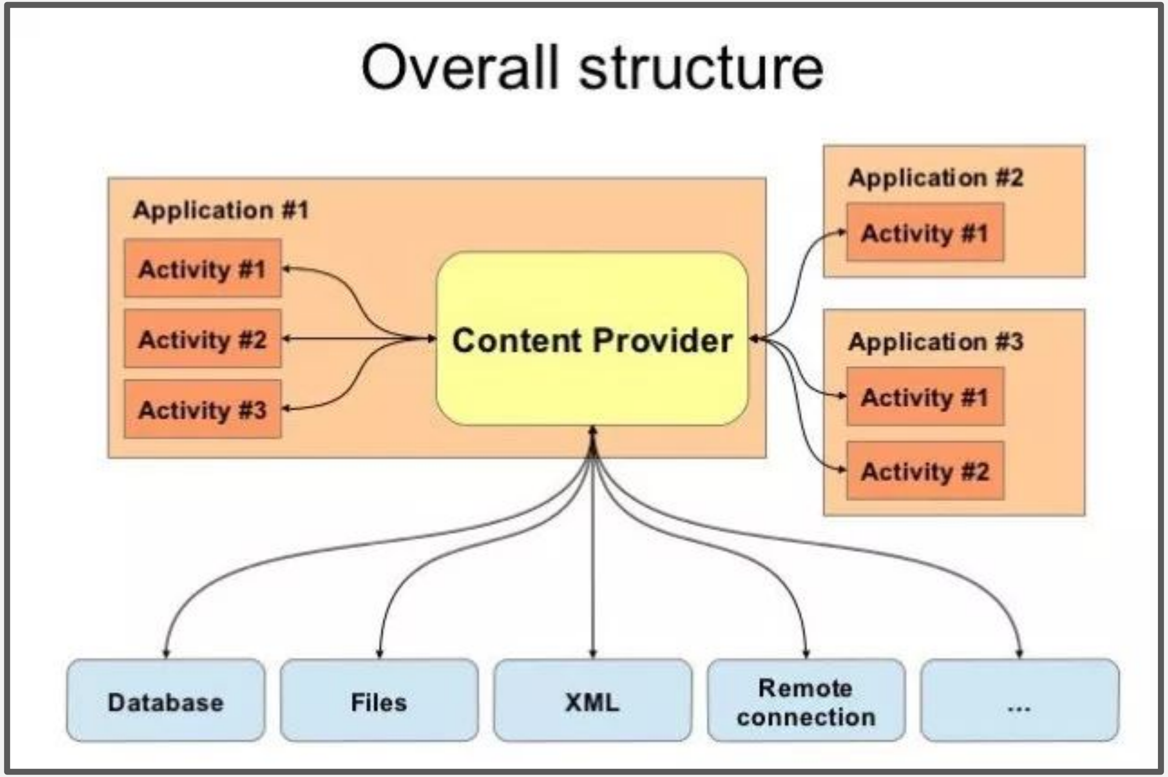


```
public void onClick(View view) {
    Intent i = new Intent(this, ActivityTwo.class);
    i.putExtra("Value1", "This value one for ActivityTwo ");
    i.putExtra("Value2", "This value two ActivityTwo");
    // set the request code to any code you like,
    // you can identify the callback via this code
    startActivityForResult(i, REQUEST_CODE);
}
```

```
Bundle extras = getIntent().getExtras();
if (extras == null) {
    return;
}
// get data via the key
String value1 = extras.getString(Intent.EXTRA_TEXT);
if (value1 != null) {
    // do something with the data
}
```

```
// Executed in an Activity, so 'this' is the Context
// The fileUrl is a string URL, such as "http://www.example.com/image.png"
Intent downloadIntent = new Intent(this, DownloadService.class);
downloadIntent.setData(Uri.parse(fileUrl));
startService(downloadIntent);
```

Probar content provider



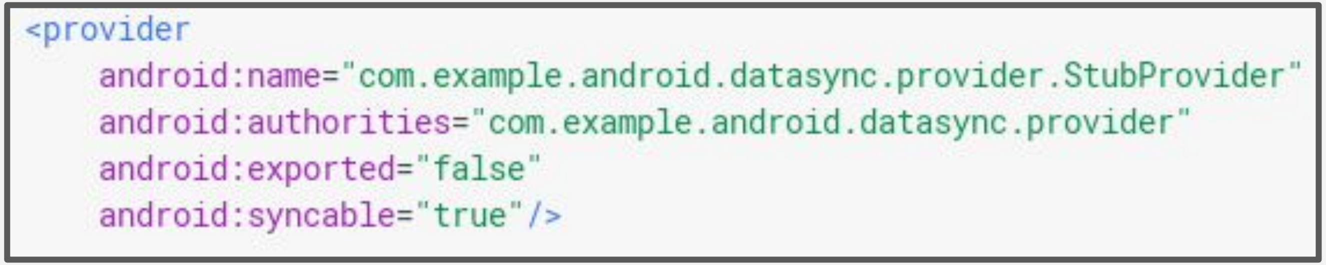
Probar content provider

run scanner.provider.finduris -a <package>

run scanner.provider.injection -a <package>

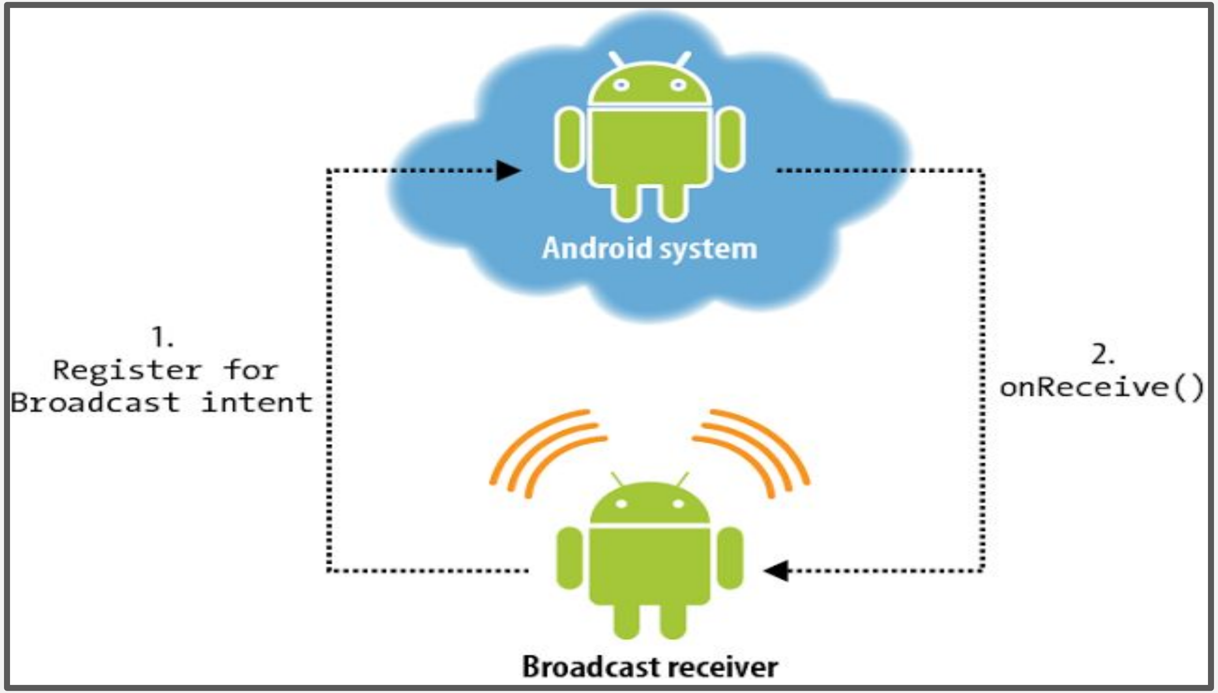
run app.provider.query [uri]

run scanner.provider.injection -a [package name]



```
<provider
    android:name="com.example.android.datasync.provider.StubProvider"
    android:authorities="com.example.android.datasync.provider"
    android:exported="false"
    android:syncable="true"/>
```

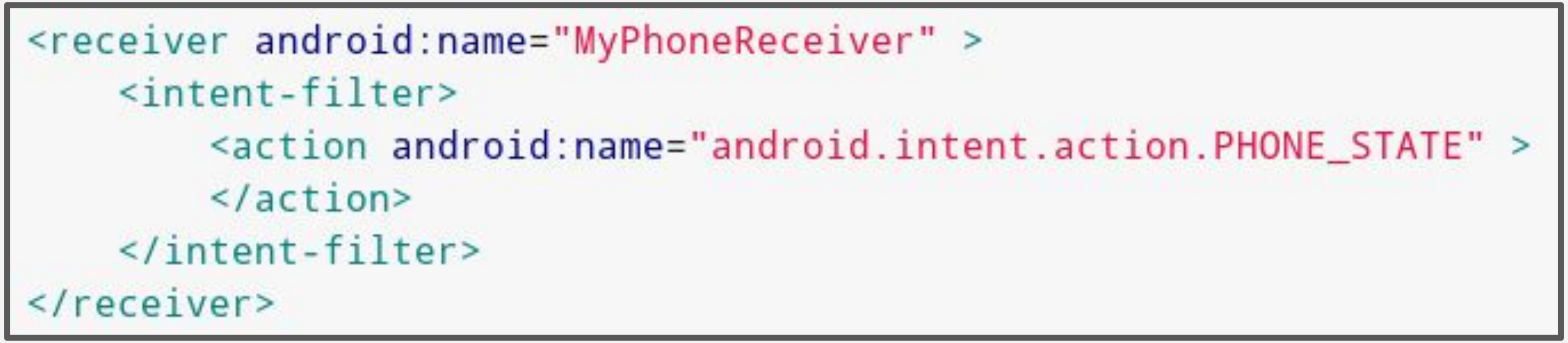
Probar broadcast receiver



Probar broadcast receiver

run app.broadcast.info -a <filtro>

run app.broadcast.send --action [action from android_manifest file] --component [package name] [broadcast reciever] --extra [type name value] --extra [type name value]

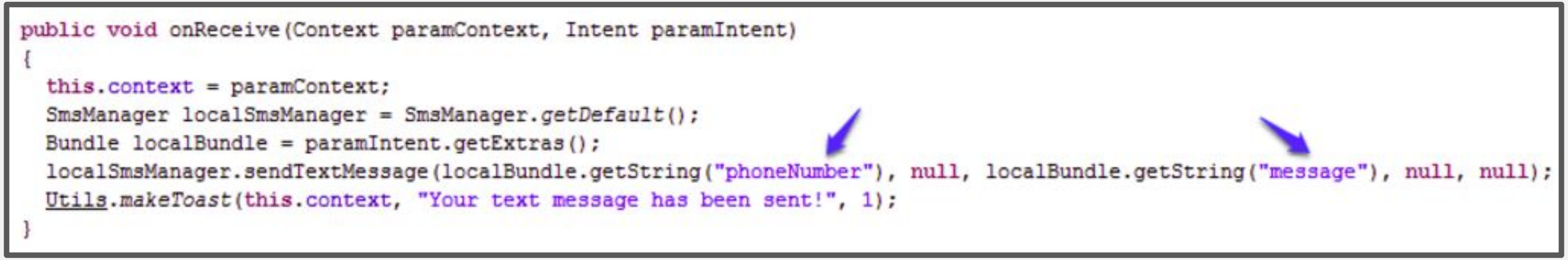


```
<receiver android:name="MyPhoneReceiver" >
    <intent-filter>
        <action android:name="android.intent.action.PHONE_STATE" >
        </action>
    </intent-filter>
</receiver>
```

Probar broadcast receiver

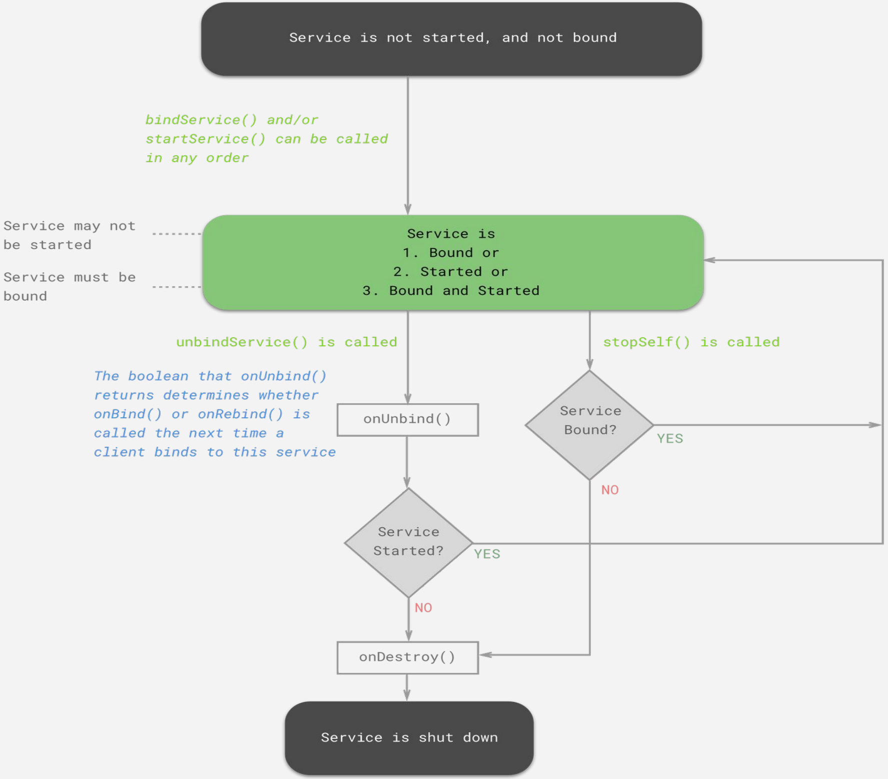
```
IntentFilter intentFilter = new IntentFilter();
intentFilter.addAction(CustomBroadCastReceiver.ACTION_SHOW_TOAST);

mReceiver = new AlarmReceiver();
registerReceiver(mReceiver, intentFilter);
```

```
Intent i = new Intent(CustomBroadCastReceiver.ACTION_SHOW_TOAST);
sendBroadcast(i);
```



```
public void onReceive(Context paramContext, Intent paramIntent)
{
  this.context = paramContext;
  SmsManager localSmsManager = SmsManager.getDefault();
  Bundle localBundle = paramIntent.getExtras();
  localSmsManager.sendTextMessage(localBundle.getString("phoneNumber"), null, localBundle.getString("message"), null, null);
  Utils.makeToast(this.context, "Your text message has been sent!", 1);
}
```

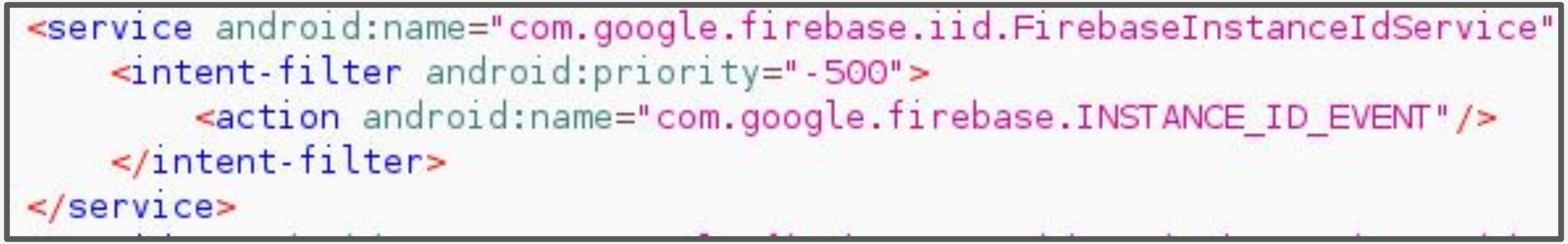
Probar services


Service is not started, and not bound
bindService() and/or startService() can be called in any order
Service may not be started
Service must be bound
Service is
1. Bound or
2. Started or
3. Bound and Started
unbindService() is called
stopSelf() is called
The boolean that onUnbind() returns determines whether onBind() or onRebind() is called the next time a client binds to this service
onUnbind()
Service Bound?
YES
NO
Service Started?
YES
NO
onDestroy()
Service is shut down

Probar services

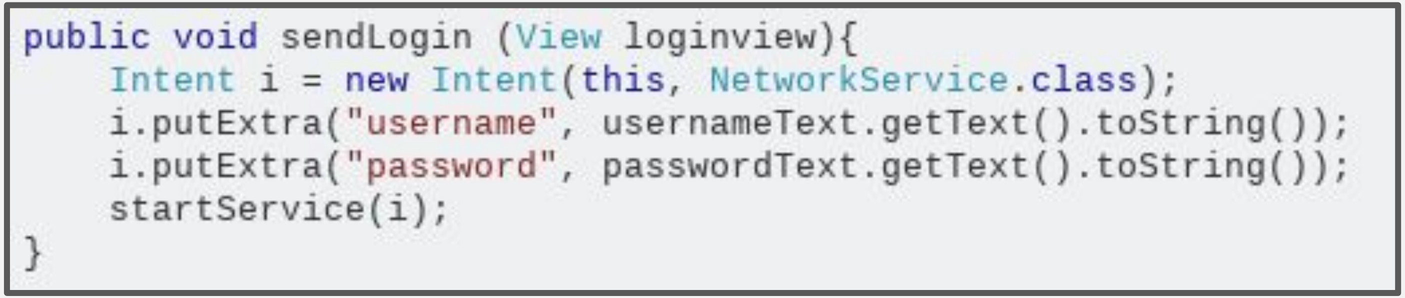
run app.service.info -a <filtro>

run app.service.send [package name] [service name] --msg [msg to service]

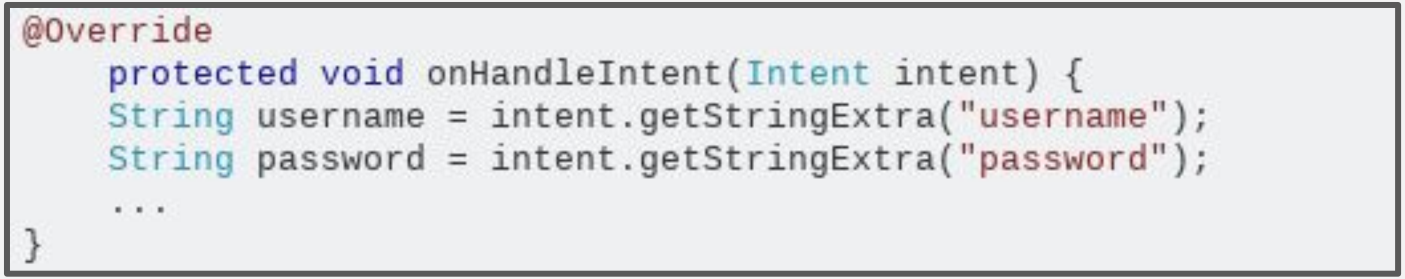


```
<service android:name="com.google.firebase.iid.FirebaseInstanceIdService"
    <intent-filter android:priority="-500">
        <action android:name="com.google.firebase.INSTANCE_ID_EVENT"/>
    </intent-filter>
</service>
```

Probar services

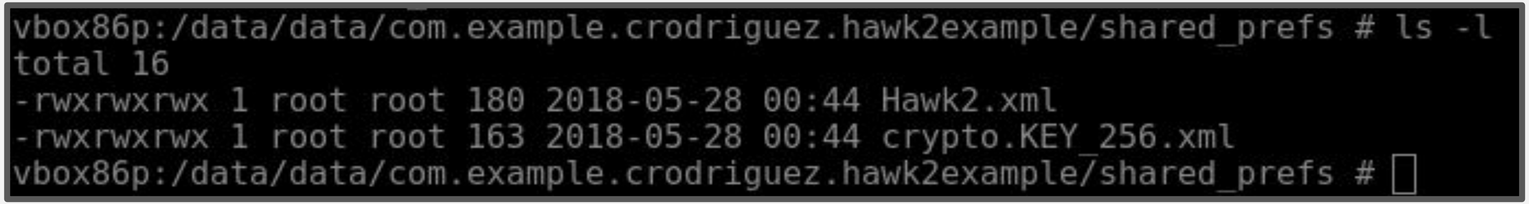


```
public void sendLogin (View loginview){
    Intent i = new Intent(this, NetworkService.class);
    i.putExtra("username", usernameText.getText().toString());
    i.putExtra("password", passwordText.getText().toString());
    startService(i);
}
```

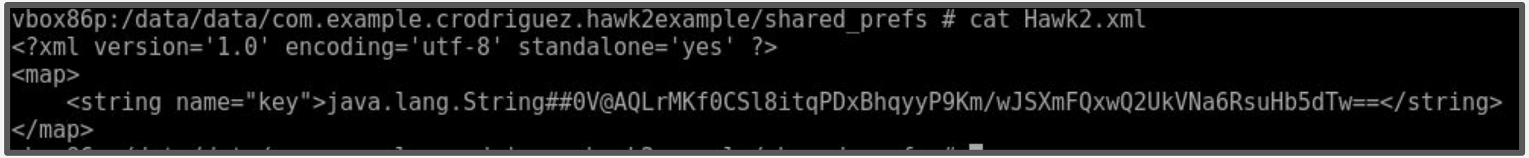


```
@Override
    protected void onHandleIntent(Intent intent) {
    String username = intent.getStringExtra("username");
    String password = intent.getStringExtra("password");
    ...
}
```

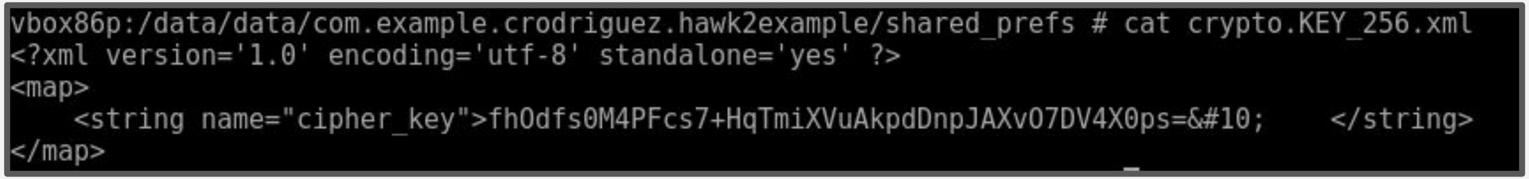
Componentes de terceros



```
vbox86p:/data/data/com.example.crodriguez.hawk2example/shared_prefs # ls -l
total 16
-rwxrwxrwx 1 root root 180 2018-05-28 00:44 Hawk2.xml
-rwxrwxrwx 1 root root 163 2018-05-28 00:44 crypto.KEY_256.xml
vbox86p:/data/data/com.example.crodriguez.hawk2example/shared_prefs # 
```



```
vbox86p:/data/data/com.example.crodriguez.hawk2example/shared_prefs # cat Hawk2.xml
<?xml version='1.0' encoding='utf-8' standalone='yes' ?>
<map>
    <string name="key">java.lang.String##0V@AQLrMKf0CSl8itqPDxBhqyyP9Km/wJSXmFQxwQ2UkVNa6RsuHb5dTw==</string>
</map>
```



```
vbox86p:/data/data/com.example.crodriguez.hawk2example/shared_prefs # cat crypto.KEY_256.xml
<?xml version='1.0' encoding='utf-8' standalone='yes' ?>
<map>
    <string name="cipher_key">fhOdfs0M4PFcs7+HqTmiXVuAkpdDnpJAXvO7DV4X0ps=&#10;    </string>
</map>
```

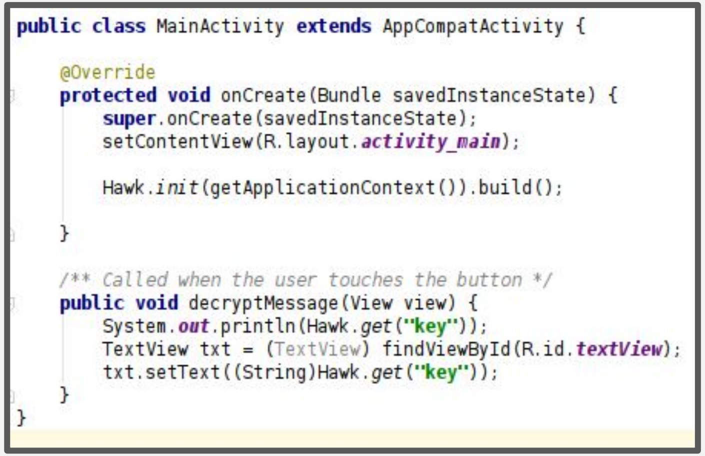
Componentes de terceros

FARADAY

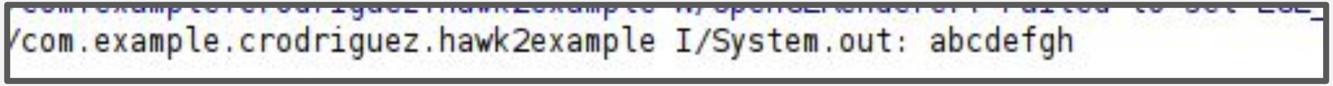
- Hawk2 es una librería de **código abierto** que permite guardar información en shared_preferences de un modo encriptado.
- Usa facebook conceal para encriptar, otra librería de **código abierto.**
- Por defecto la clave que usa se genera en un archivo dentro de shared_prefs.
- allowBackup=true

Componentes de terceros

Backup con adb y cree un proyecto en AndroidStudio para levantar estas preferencias:



```
public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        Hawk.init(getApplicationContext()).build();

    }

    /** Called when the user touches the button */
    public void decryptMessage(View view) {
        System.out.println(Hawk.get("key"));
        TextView txt = (TextView) findViewById(R.id.textView);
        txt.setText((String)Hawk.get("key"));
    }
}
```



```
com.example.crodriguez.hawk2example W/OpenGLRenderer: Failed to set EGL_
/com.example.crodriguez.hawk2example I/System.out: abcdefgh
```

Conclusiones

Conocer el aplicativo y sus funcionalidades más sensibles

Cuanta más información se tenga, más rápido se hace el pentest

Priorizar vulnerabilidades de persistencia de datos, de manejo de sesiones y errores de lógica

Muchos bugs se encuentran concatenando fallas de configuración con menor criticidad

La practica hace al maestro

Fuentes

FARADAY

https://github.com/OWASP/owasp-mstg

Mobile Application Hacker’s Handbook

Android Security Internals

https://github.com/ashishb/android-security-awesome

https://github.com/enaqx/awesome-pentest

@mobilesecurity_

Alguna consulta?

FARADAY

@warlockk87

crodriguez@infobytesec.com

¡CHAU CHICOS!