

# Low Hanging Fruit On Android



# Zezadas

AKA José Moreira

Currently working at **S21**<sub>SEC</sub>

-  <https://peidei.me>
-  <https://sefod.eu>
-  [@0xZ3Z4d45](https://twitter.com/0xZ3Z4d45)



# Reverse, patch, repack, repeat

1. Obtain and unpack APK
2. Analyze code, resources, files
3. Patch smali code
4. Rebuild and sign APK
5. Install into device / emulator
6. ???
7. PROFIT!

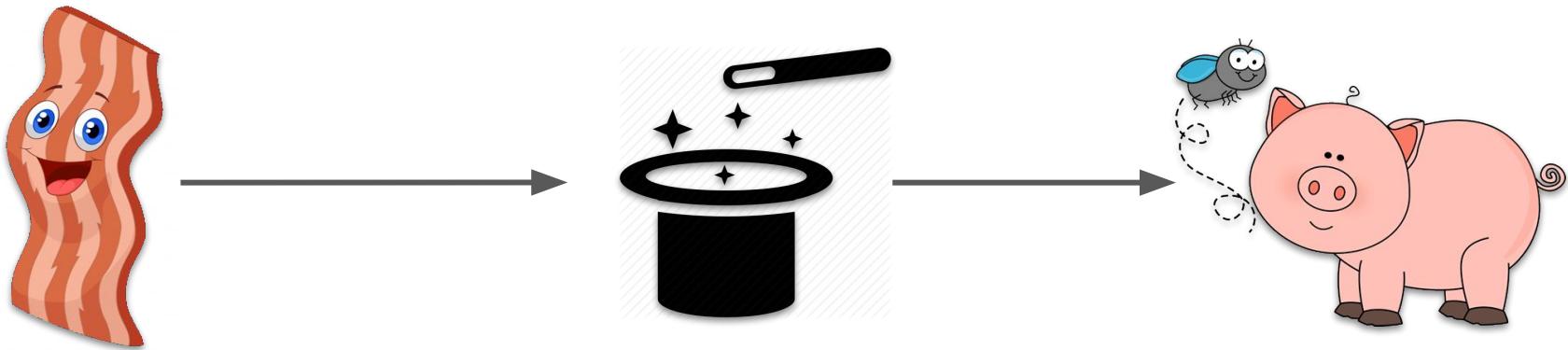


# Reversing Android - 101

101



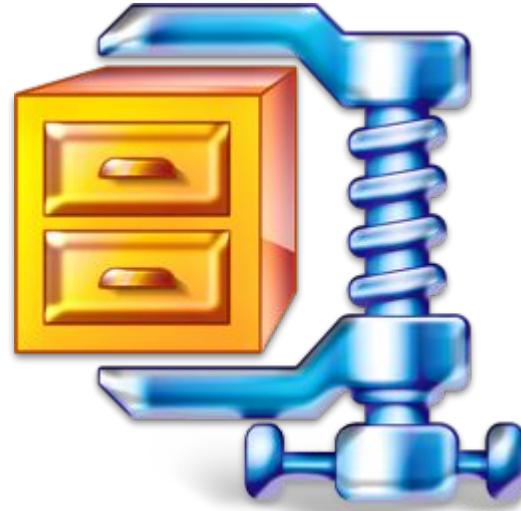
# What is reversing?



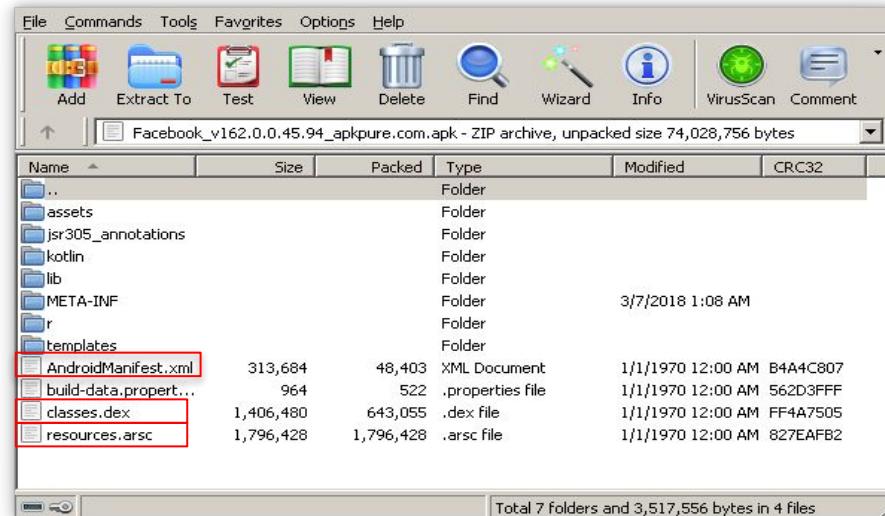
# APK



# APK



# APK

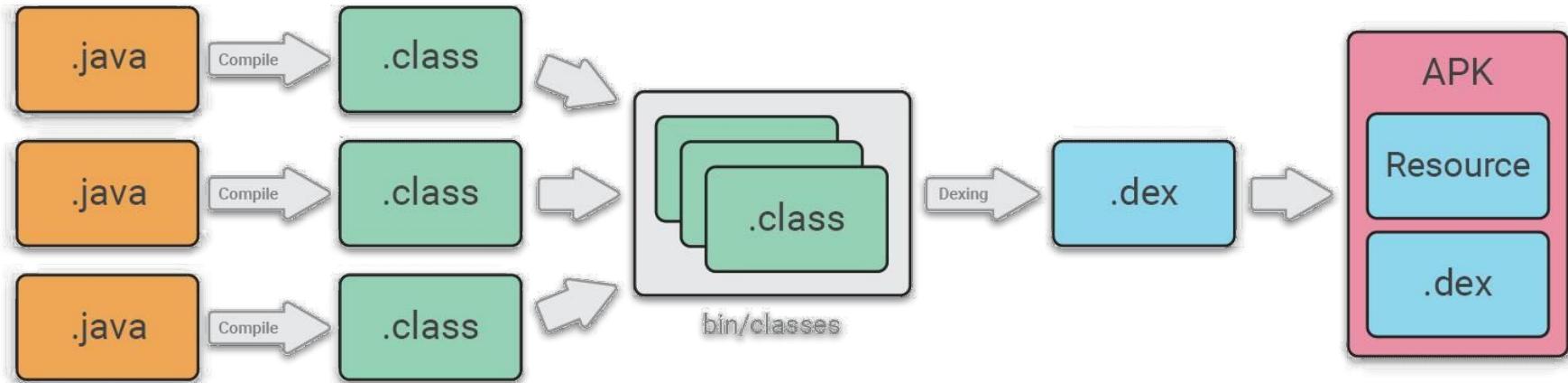


The screenshot shows the WinRAR interface displaying the contents of an APK file. The file path is "Facebook\_v162.0.0.45.94\_apkpure.com.apk". The table lists the following files:

Name	Size	Packed	Type	Modified	CRC32
...			Folder		
assets			Folder		
jsr305_annotations			Folder		
kotlin			Folder		
lib			Folder		
META-INF			Folder	3/7/2018 1:08 AM	
r			Folder		
templates			Folder		
AndroidManifest.xml	313,684	48,403	XML Document	1/1/1970 12:00 AM	B4A4C807
build-data.propert...	964	522	.properties file	1/1/1970 12:00 AM	562D3FFF
classes.dex	1,406,480	643,055	.dex file	1/1/1970 12:00 AM	FF4A7505
resources.arsc	1,796,428	1,796,428	.arsc file	1/1/1970 12:00 AM	827EAFB2

Total 7 folders and 3,517,556 bytes in 4 files

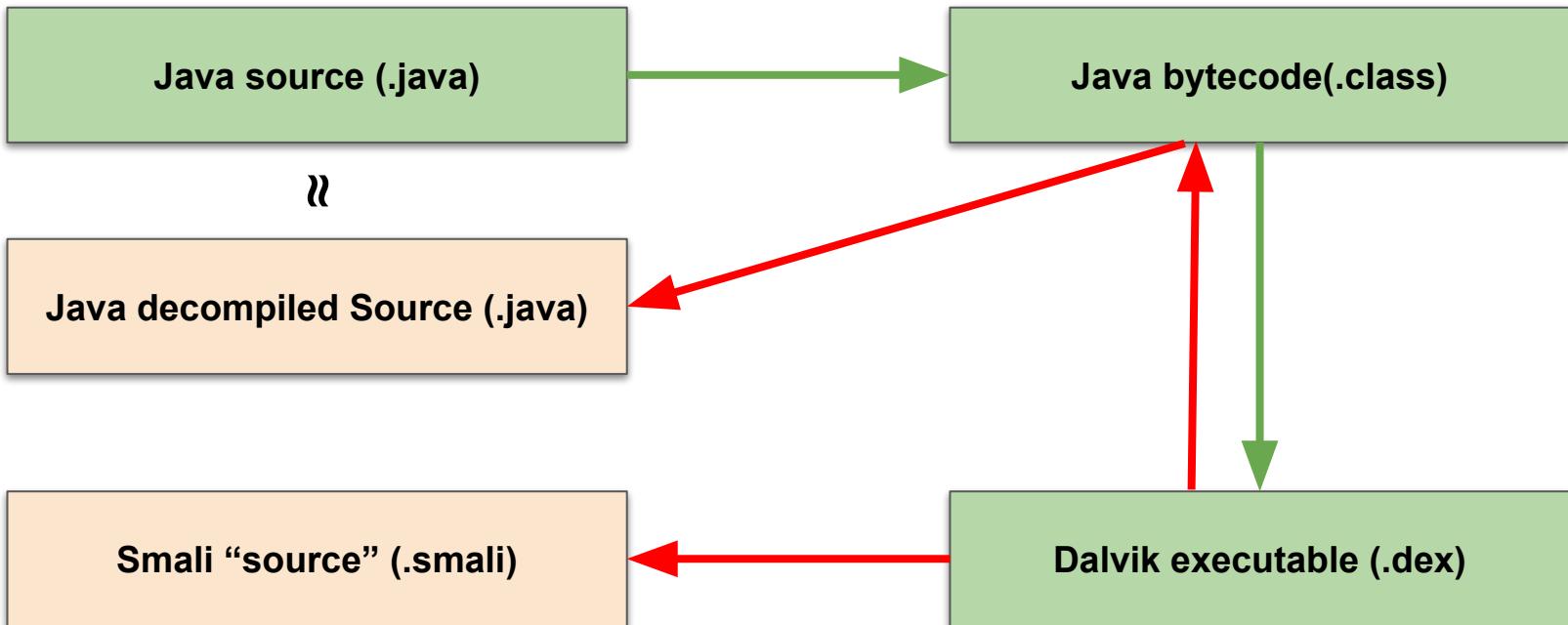
# Android compilation (simplified)



# Java Compilation / reversing

→ Compilation

→ Reversing



# Java Vs Smali Vs Byte Code

Java code:

```
import java.io.PrintStream;  
  
public class HelloWorld  
{  
    public static void main(String[] paramArrayOfString)  
    {  
        System.out.println("Hello World!");  
    }  
}
```

Java Bytecode:

???

Dalvik Bytecode:

???

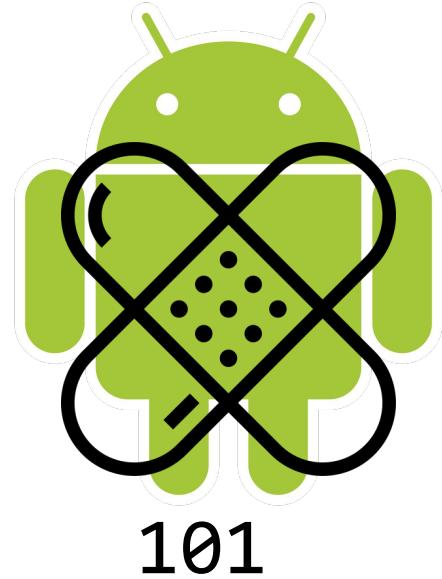
Decompiled Java bytecode:

```
public class HelloWorld {  
  
    public static main(java.lang.String[] arg0) { //([Ljava/lang/String;)V  
        getstatic java/lang/System.out:java.io.PrintStream  
        ldc "Hello World!" (java.lang.String)  
        invokevirtual java/io/PrintStream println((Ljava/lang/String;)V)  
        return  
    }  
}
```

Smali code:

```
.class public LHelloWorld;  
.super Ljava/lang/Object;  
  
.method public static main([Ljava/lang/String;)V  
    .registers 3  
    sget-object v0, Ljava/lang/System;->out:Ljava/io/PrintStream;  
    const-string v1, "Hello World!"  
    invoke-virtual {v0, v1}, Ljava/io/PrintStream;->println(Ljava/lang/String;)V  
    return-void  
.end method
```

# Patching Android - 101





**Target**

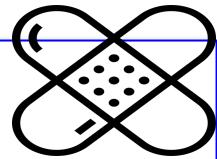


**APK**

```
.class public LHelloWorld;
.super Ljava/lang/Object;

# direct methods
.method public static main([Ljava/lang/String;)V
    .registers 3

    sget-object v0, Ljava/lang/System;->out:Ljava/io/PrintStream;
    const-string v1, "Hello World!"
    invoke-virtual {v0, v1}, Ljava/io/PrintStream;->println(Ljava/lang/String;)V
    return-void
.end method
```



**Smali code**

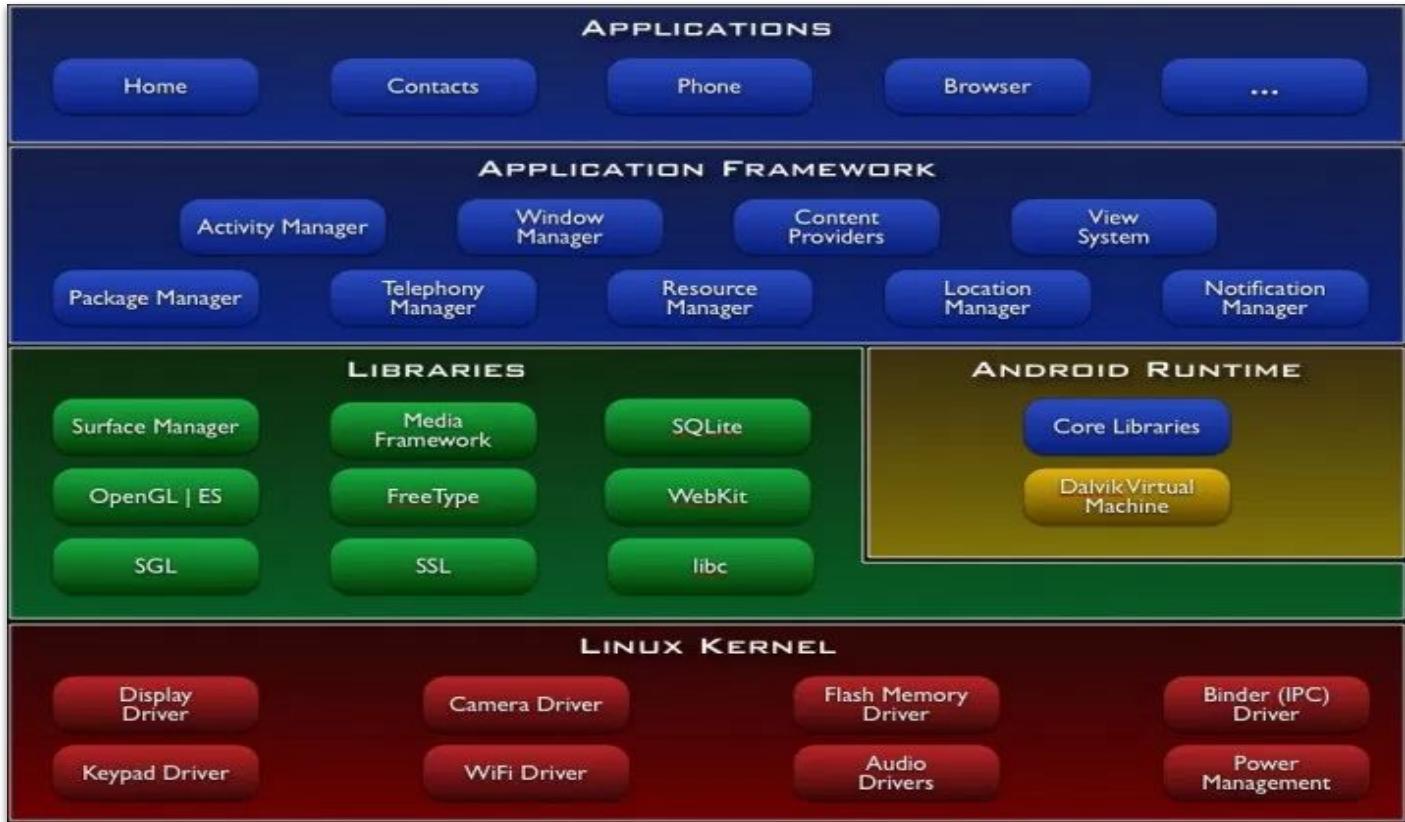
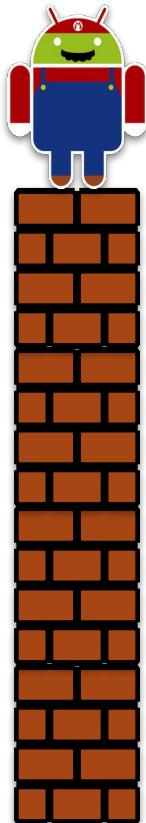


**Signing key**



**APK  
Untrusted**

# Android Stack



# START HACKING! - Tools

- Install:

- Android Studio (emulator && platform-tools) - <https://developer.android.com/studio>
- JDK - <https://jdk.java.net/>
- APKtool - <https://ibotpeaches.github.io/Apktool/>
- Burp - <https://portswigger.net/burp>
- Dex2jar - <https://github.com/pxb1988/dex2jar>
- Bytecodeviewer - <https://github.com/Konloch bytecode-viewer>
- jadx-gui - <https://github.com/skylot/jadx>



# Practical - ENEI App 2018

Ways to obtain APKs:

- <https://apk-dl.com>
- <http://www.aptoide.com/>
- <https://www.apkmonk.com/>
- Pull from android phone:
  - \$ adb shell pm list packages
  - \$ adb shell pm path com.package.name
  - \$ adb pull /full/path/to/apk



# ENEI 2019 - App

ENEI 2018 - Apps on Google +

<https://play.google.com/store/apps/details?id=com.enei.eneimobile>

Apps Categories Home Top Charts New Releases

My apps Shop Games Family Editors' Choice

Account Payment methods My subscriptions Redeem Buy gift card My wishlist My Play activity Parent Guide

ENEI 2018

André Lago Events

PEGI 3

This app is compatible with all of your devices.

Installed

23/MAR	24/MAR	25/MAR	26/MAR
Palestras			
14:00	Boas-vindas		
14:30	14:00 - 17:30   Entrada FEUP		
15:00			
16:00			
16:30			
17:00			
17:30	Sessão de abertura		
18:00	17:30 - 19:30   Auditório FEUP		
18:30			
19:00			
19:30	Jantar   Cantina de Engenharia		
20:00			
21:00	Festarola   AEFEUP		
02:00			

# Google + APKMonk

APKMonk

Apks >> Events >> ENEI 2018



 ENEI 2018 apk

Updated On March 24, 2018

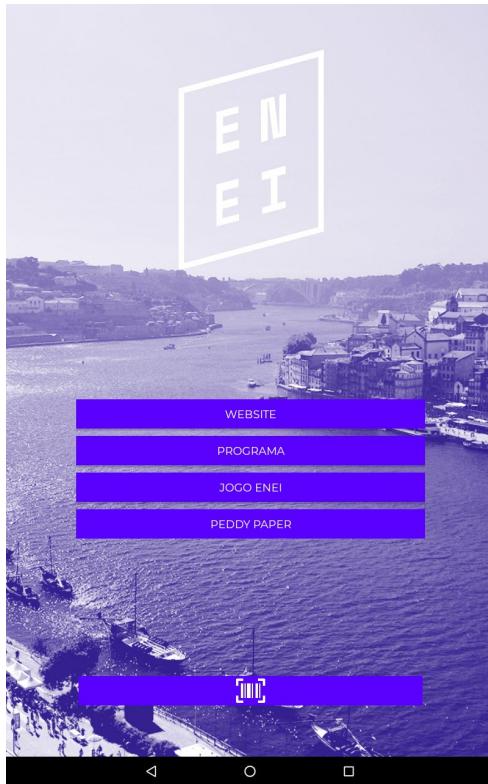
App Version N/A

[DOWNLOAD APK](#)

This apk is safe to download from this mirror and free of any virus.

[Check Previous Versions](#)

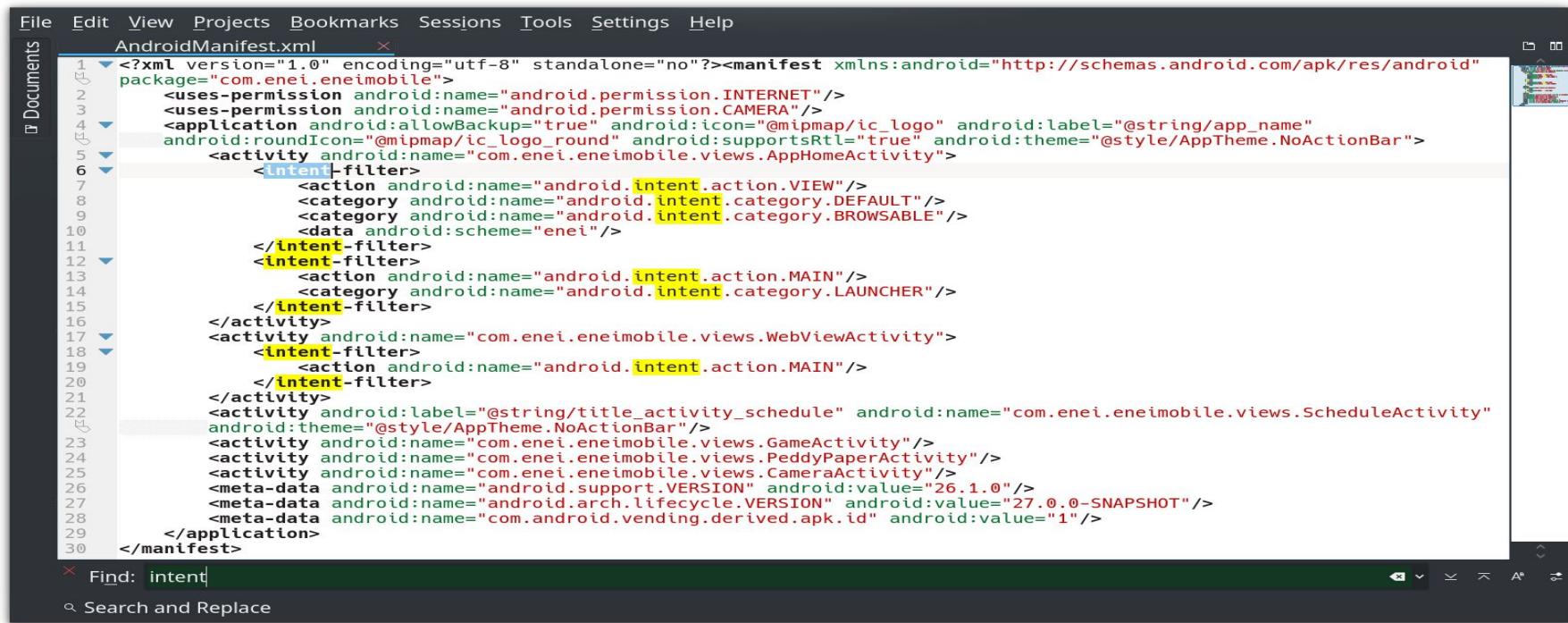
# ENEI 2018 App



# Extract && Decompile

```
File Edit View Bookmarks Settings Help
anon@unknown ~ /oposec/enei apktool d com.enei.eneimobile_2018-03-24.apk
I: Using Apktool 2.3.4 on com.enei.eneimobile_2018-03-24.apk
I: Loading resource table...
I: Decoding AndroidManifest.xml with resources...
I: Loading resource table from file: /home/anon/.local/share/apktool/framework/1.apk
I: Regular manifest package...
I: Decoding file-resources...
I: Decoding values */* XMLs...
I: Baksmaling classes.dex...
I: Copying assets and libs...
I: Copying unknown files...
I: Copying original files...
anon@unknown ~ /oposec/enei _
```

# AndroidManifest.XML



The screenshot shows an AndroidManifest.xml file open in an IDE. The file defines the application's permissions, activities, and meta-data.

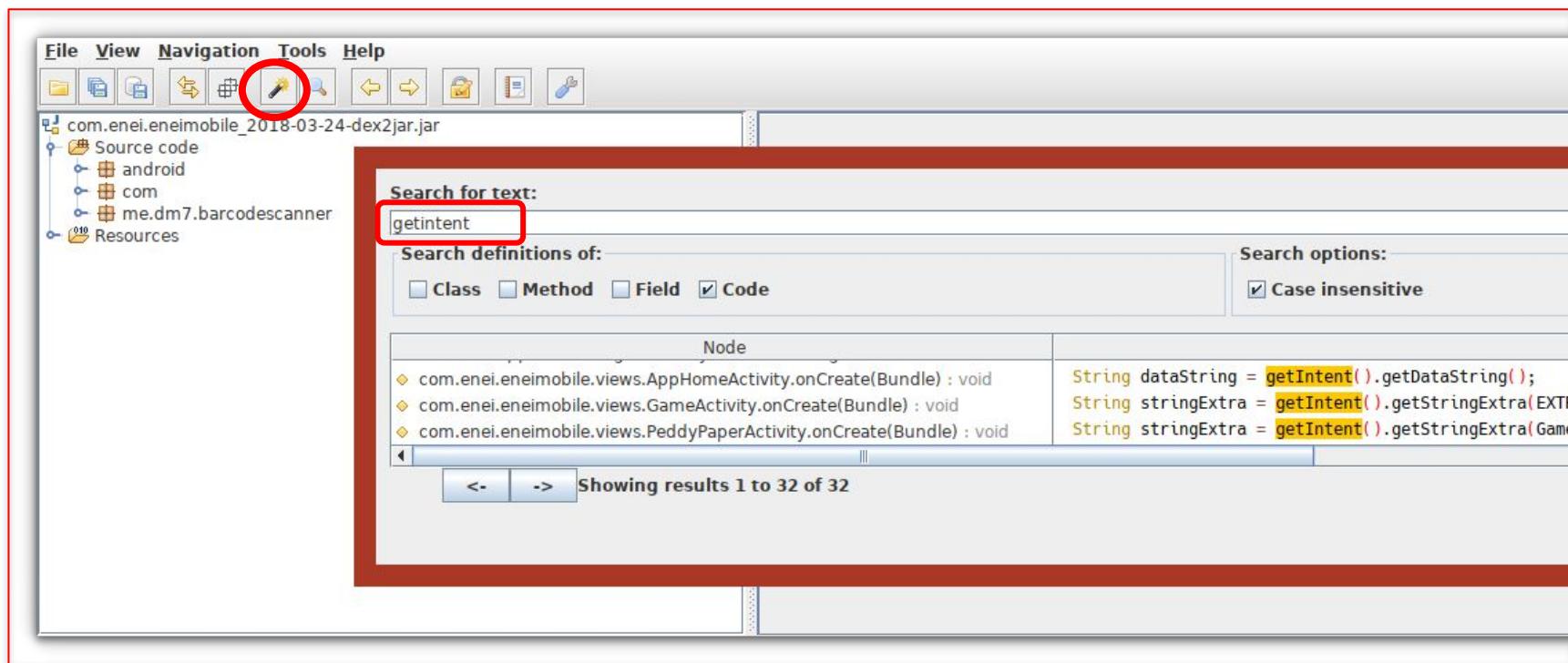
```
<?xml version="1.0" encoding="utf-8" standalone="no"?><manifest xmlns:android="http://schemas.android.com/apk/res/android" package="com.enei.eneimobile">
    <uses-permission android:name="android.permission.INTERNET"/>
    <uses-permission android:name="android.permission.CAMERA"/>
    <application android:allowBackup="true" android:icon="@mipmap/ic_logo" android:label="@string/app_name" android:roundIcon="@mipmap/ic_logo_round" android:supportsRtl="true" android:theme="@style/AppTheme.NoActionBar">
        <activity android:name="com.enei.eneimobile.views.AppHomeActivity">
            <intent-filter>
                <action android:name="android.intent.action.VIEW"/>
                <category android:name="android.intent.category.DEFAULT"/>
                <category android:name="android.intent.category.BROWSABLE"/>
                <data android:scheme="enei"/>
            </intent-filter>
            <intent-filter>
                <action android:name="android.intent.action.MAIN"/>
                <category android:name="android.intent.category.LAUNCHER"/>
            </intent-filter>
        </activity>
        <activity android:name="com.enei.eneimobile.views.WebViewActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN"/>
            </intent-filter>
        </activity>
        <activity android:label="@string/title_activity_schedule" android:name="com.enei.eneimobile.views.ScheduleActivity" android:theme="@style/AppTheme.NoActionBar"/>
        <activity android:name="com.enei.eneimobile.views.GameActivity"/>
        <activity android:name="com.enei.eneimobile.views.PeddyPaperActivity"/>
        <activity android:name="com.enei.eneimobile.views.CameraActivity"/>
        <meta-data android:name="android.support.VERSION" android:value="26.1.0"/>
        <meta-data android:name="android.arch.lifecycle.VERSION" android:value="27.0.0-SNAPSHOT"/>
        <meta-data android:name="com.android.vending.derived.apk.id" android:value="1"/>
    </application>
</manifest>
```

The code editor has a search bar at the bottom with the text "Find: intent".

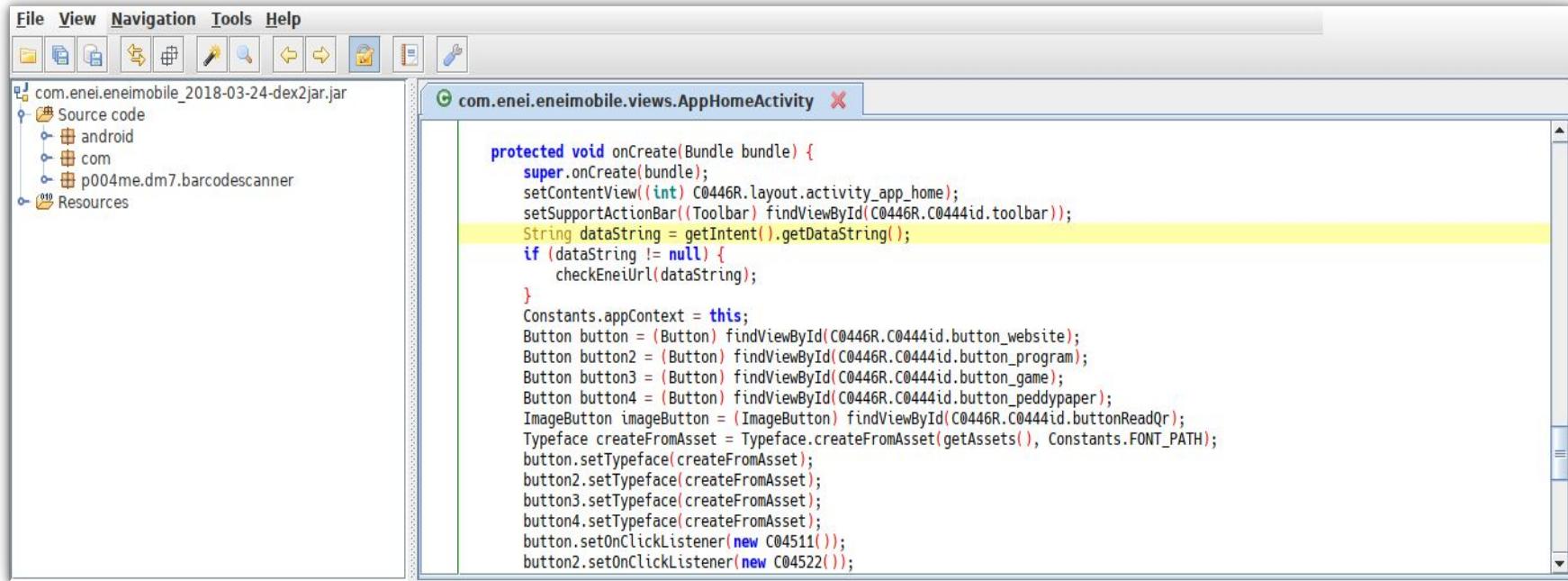
# Dex2Jar && Jadx-GUI

```
File Edit View Bookmarks Settings Help
anon@unknown ~ /oposec/enei d2j-dex2jar com.enei.eneimobile_2018-03-24.apk
dex2jar com.enei.eneimobile_2018-03-24.apk -> ./com.enei.eneimobile_2018-03-24-dex2jar.jar
anon@unknown ~ /oposec/enei jadx-gui com.enei.eneimobile_2018-03-24-dex2jar.jar
INFO - output directory: com.enei.eneimobile_2018-03-24-dex2jar
INFO - loading ...
INFO - converting to dex: com.enei.eneimobile_2018-03-24-dex2jar.jar ...
```

# Jadx-GUI (1/5)- Search Intents



# Jadx-GUI (2/5)- Code Analysis



The screenshot shows the Jadx-GUI interface with the following details:

- File Menu:** File, View, Navigation, Tools, Help.
- Toolbar:** Includes icons for Open, Save, Find, Replace, and others.
- Project Tree:** Shows the project structure: com.enei.eneimobile\_2018-03-24-dex2jar.jar, Source code, android, com, p004me.dm7.barcodescanner, and Resources.
- Code Editor:** Displays the Java code for AppHomeActivity. A specific line of code, `String dataString = getIntent().getStringExtra("url");`, is highlighted with a yellow background.

```
protected void onCreate(Bundle bundle) {
    super.onCreate(bundle);
    setContentView(C0446R.layout.activity_app_home);
    setSupportActionBar(findViewById(C0446R.C0444id.toolbar));
    String dataString = getIntent().getStringExtra("url");
    if (dataString != null) {
        checkEneiUrl(dataString);
    }
    Constants.appContext = this;
    Button button = (Button) findViewById(C0446R.C0444id.button_website);
    Button button2 = (Button) findViewById(C0446R.C0444id.button_program);
    Button button3 = (Button) findViewById(C0446R.C0444id.button_game);
    Button button4 = (Button) findViewById(C0446R.C0444id.button_peddypaper);
    ImageButton imageView = (ImageButton) findViewById(C0446R.C0444id.buttonReadQr);
    Typeface createFromAsset = Typeface.createFromAsset(getAssets(), Constants.FONT_PATH);
    button.setTypeface(createFromAsset);
    button2.setTypeface(createFromAsset);
    button3.setTypeface(createFromAsset);
    button4.setTypeface(createFromAsset);
    button.setOnClickListener(new C04511());
    button2.setOnClickListener(new C04522());
}
```

# Jadx-GUI (3/5)- Code Analysis

The screenshot shows the Jadx-GUI interface with the following details:

- File Menu:** File, View, Navigation, Tools, Help.
- Toolbar:** Includes icons for file operations like Open, Save, and a search icon.
- Project Tree:** Shows the project structure under "com.enei.eneimobile\_2018-03-24-dex2jar.jar".
  - Source code:
    - android
    - com
    - p004me.dm7.barcodescanner
  - Resources
- Code Editor:** Displays the Java code for `AppHomeActivity`. The code implements a method `checkEneiUrl` which iterates through lists of locations to start an activity based on a substring of the URL. Red annotations highlight the string comparison in the first loop and the intent creation in the second loop, with arrows pointing to the corresponding code lines.
- Bottom Line:** Shows the start of another method: `private boolean haveCameraPermission()`.

# Jadx-GUI (4/5)- Code Analysis

File View Navigation Tools Help

com.enei.eneimobile\_2018-03-24-dex2jar.jar

- Source code
  - android
  - com
  - p004me.dm7.barcodescanner
- Resources

com.enei.eneimobile.views.AppHomeActivity

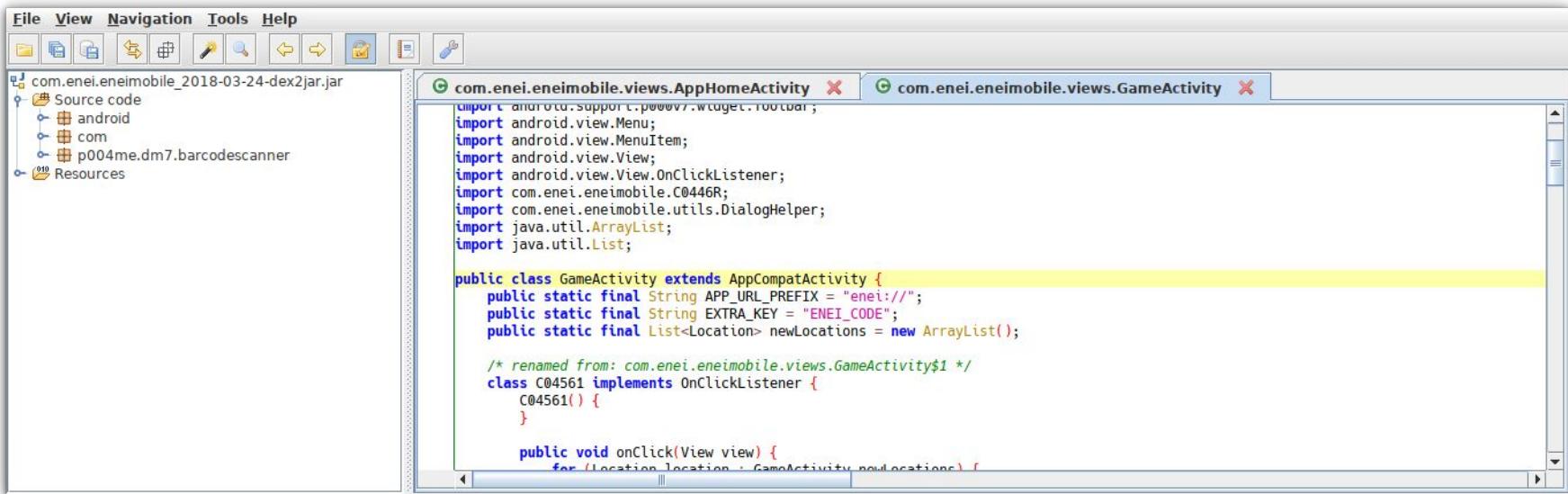
```
}

static {
    newLocations.add(new Location("sonaeim", "8a5zM0bcPiKx2bawyD7F", C0446R.C0444id.sonaeim, "SONAE IM", "Patrocinador principal do
    newLocations.add(new Location("deloitte", "TXRarKo4okU6uwnJXBQ", C0446R.C0444id.deloitte, "Deloitte", "Conta com aproximadamen
    newLocations.add(new Location("efacec", "Hr17IKDLTwc50EVGwyF1", C0446R.C0444id.efacec, "Efacec", "É a maior empresa industrial
    newLocations.add(new Location("nativixis", "aZqGuRqMRpV8nSQzQm1l", C0446R.C0444id.nativixis, "Nativixis", "Segmento bancário para emp
    newLocations.add(new Location("wipro", "AXcHTURcEbHwHbb@LPj2", C0446R.C0444id.wipro, "Wipro", "Empresa líder mundial em tecnolo
    newLocations.add(new Location("biblioteca", "53lGHSRx4xv9@FPUpyo", C0446R.C0444id.biblioteca, "Biblioteca", "O principal local
    newLocations.add(new Location("portafeup", "mAVGiCur5VCxk6mvlkI4", C0446R.C0444id.entrada, "Entrada FEUP", "Principal entrada d
    newLocations.add(new Location("niafeup", "zCzELx4qqI7ymMT85bZB", C0446R.C0444id.niafeup, "NIAEFEUP", "Núcleo de Informática d
    newLocations.add(new Location("ieee", "Zmk8VLxxSzHYKgUsh9uP", C0446R.C0444id.ieee, "IEEE UP Student Branch", "Ramo da Universid
    newLocations.add(new Location("cantina", "DRqdnOPemYQ5l1hwPpJ8", C0446R.C0444id.cantina, "Cantina de Engenharia", "Cantina que
    newLocations.add(new Location("falcao", "yEAi5oancUzkDPJX01Ug", C0446R.C0444id.falcao, "Pavilhão Luís Falcão", "Pavilhão despor
    newLocations.add(new Location("queijos", "l6odYe8qYQakHYKS00qi", C0446R.C0444id.queijos, "Queijos", "Embora seja o local onde s
    newLocations.add(new Location("dei", "QrsTEPDcSTBWhogUxkCF", C0446R.C0444id.dei, "DET", "Este é o departamento responsável pelo
    newLocations.add(new Location("cica", "C1XcR3jvBbFBbUy2sBvF", C0446R.C0444id.cica, "CICA", "Centro de Informática da FEUP, resp
    newLocations.add(new Location("aefeup", "Bl3Lu4JFGxb5zCrUhJw", C0446R.C0444id.aefeup, "AEEFUP", "Principal local de convívio p
```

}

private void checkSuccessfulCodes() {

# Jadx-GUI (5/5)- Code Analysis



The screenshot shows the Jadx-GUI interface with the following details:

- File Menu:** File, View, Navigation, Tools, Help.
- Toolbar:** Includes icons for file operations like Open, Save, Find, and others.
- Project Tree:** Shows the project structure with files like com.enei.eneimobile\_2018-03-24-dex2jar.jar, Source code, Resources, and specific Java files.
- Code Editor:** Displays the source code of `com.enei.eneimobile.views.GameActivity`. The code includes imports for various Android classes and interfaces, and defines a class `GameActivity` that extends `AppCompatActivity`.

```
import android.support.v7.widget.Toolbar;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.view.View.OnClickListener;
import com.enei.eneimobile.C0446R;
import com.enei.eneimobile.utils.DialogHelper;
import java.util.ArrayList;
import java.util.List;

public class GameActivity extends AppCompatActivity {
    public static final String APP_URL_PREFIX = "enei://";
    public static final String EXTRA_KEY = "ENEI_CODE";
    public static final List<Location> newLocations = new ArrayList();

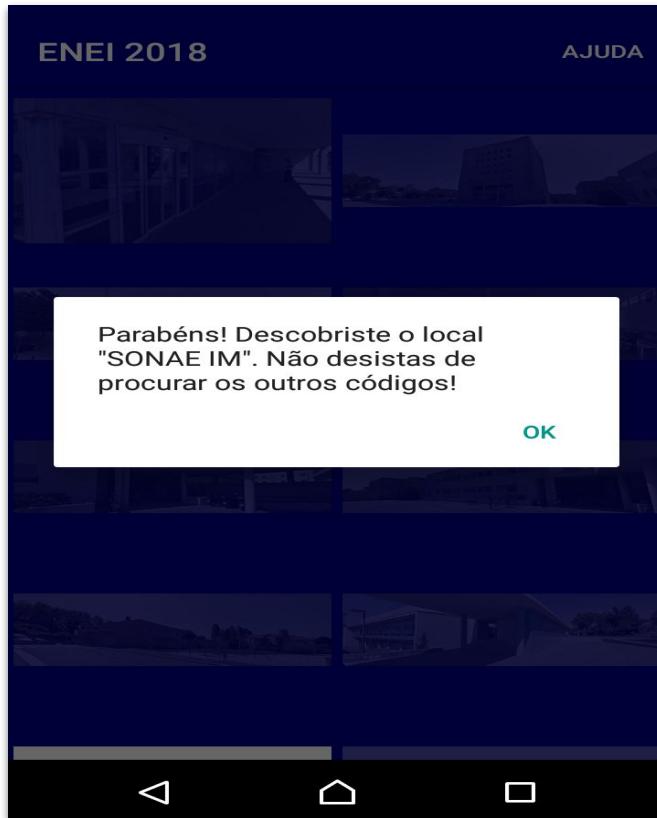
    /* renamed from: com.enei.eneimobile.views.GameActivity$1 */
    class C04561 implements OnClickListener {
        C04561() {
        }

        public void onClick(View view) {
            for (Location location : GameActivity.newLocations) {
                ...
            }
        }
    }
}
```

Exploit 1 - enei://8a5zMObcPiKx2bawyD7F



# Solve 1



# Exploit 2 - Stealth

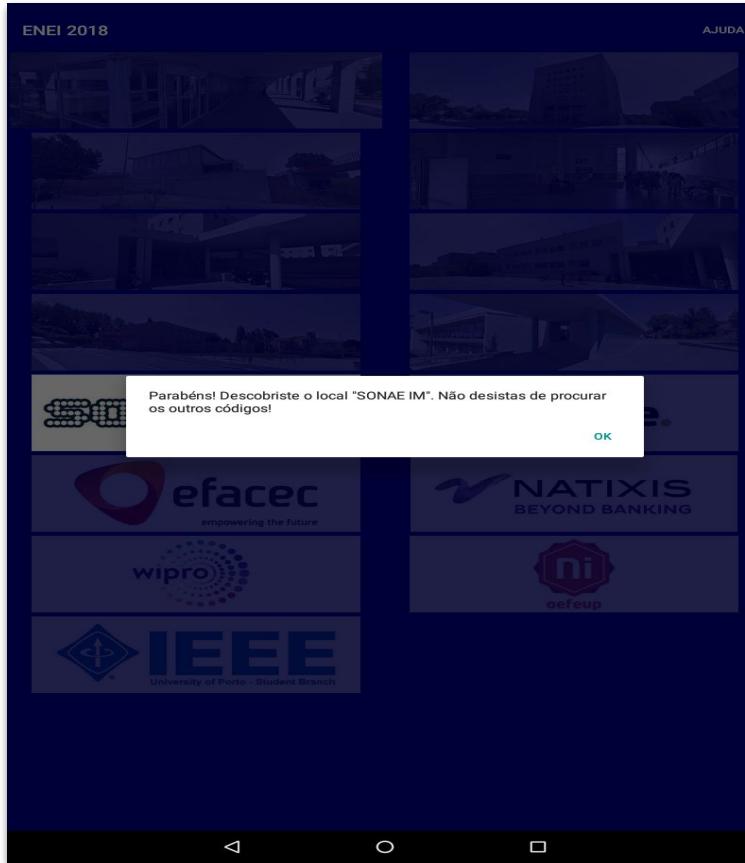
File Edit View Bookmarks Settings Help

```
→ ~ adb shell am start -a android.intent.action.VIEW -d "enei://8a5zM0bcPiKx2bawyD7F"  
Starting: Intent { act=android.intent.action.VIEW dat=enei://8a5zM0bcPiKx2bawyD7F }
```

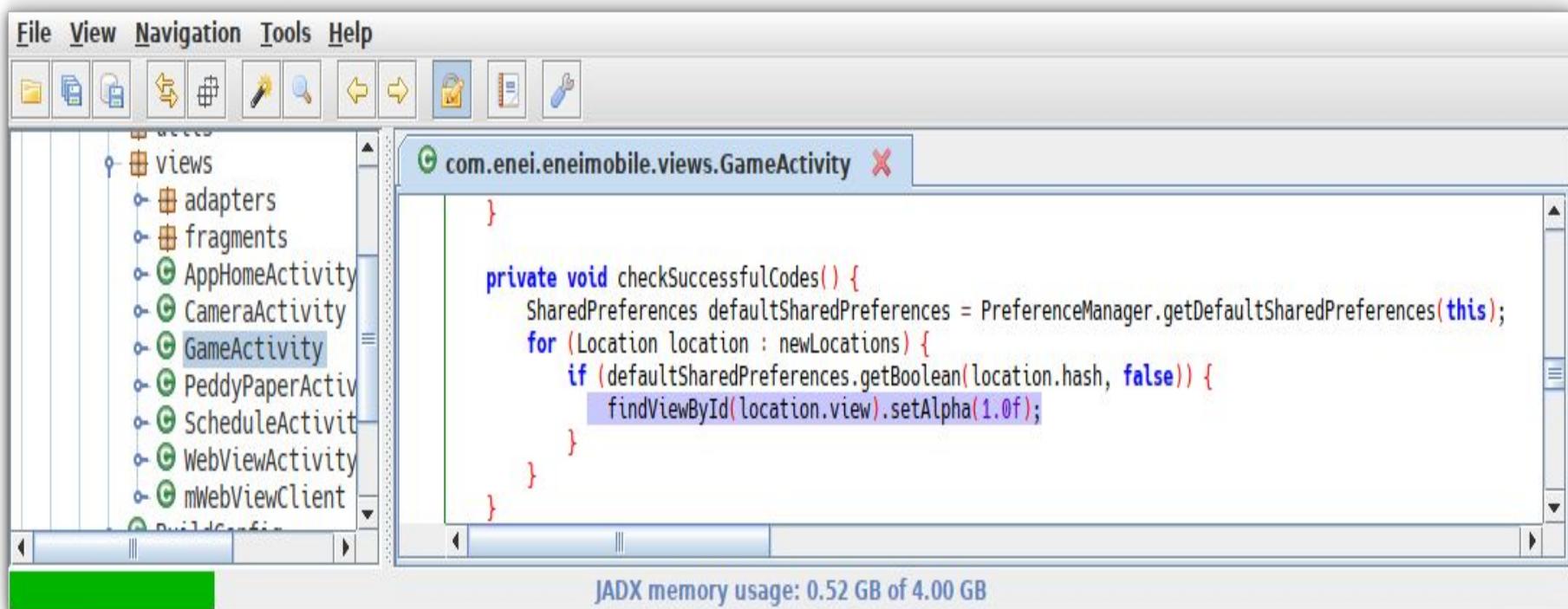
```
→ ~ |
```

**adb shell am start -a android.intent.action.VIEW -d "enei://8a5zM0bcPiKx2bawyD7F"**

# Solve 2



# Exploit 3 - Analysing



The screenshot shows the JADX tool interface. The menu bar includes File, View, Navigation, Tools, and Help. The toolbar contains various icons for file operations and analysis. The left sidebar displays a tree view of the application's package structure, including views, adapters, fragments, and several Activity classes: AppHomeActivity, CameraActivity, GameActivity, PeddyPaperActivity, ScheduleActivity, WebViewActivity, and mWebViewClient. The main window shows the decompiled Java code for the GameActivity class. The code is as follows:

```
    }
}

private void checkSuccessfulCodes() {
    SharedPreferences defaultSharedPreferences = PreferenceManager.getDefaultSharedPreferences(this);
    for (Location location : newLocations) {
        if (defaultSharedPreferences.getBoolean(location.hash, false)) {
            findViewById(location.view).setAlpha(1.0f);
        }
    }
}
```

At the bottom of the main window, a status bar indicates "JADX memory usage: 0.52 GB of 4.00 GB".

# Exploit 3 - Patching

The screenshot shows a Smali editor interface with the following details:

- File Menu:** File, Edit, View, Projects, Bookmarks, Sessions, Tools, Settings, Help.
- Document List:** Shows a list of documents on the left, with "GameActivity.smali" currently selected.
- Code Editor:** The main area displays the following Smali code:

```
357 move-result-object v2
358 :cond_0
359 :goto_0
360 invoke-interface {v2}, Ljava/util/Iterator;-->hasNext()Z
361
362 move-result v3
363
364 if-eqz v3, :cond_1
365
366 invoke-interface {v2}, Ljava/util/Iterator;-->next()Ljava/lang/Object;
367
368 move-result-object v0
369
370 check-cast v0, Lcom/enei/eneimobile/views/GameActivity$Location;
371
372 .line 140
373 .local v0, "location":Lcom/enei/eneimobile/views/GameActivity$Location;
374 igeget-object v3, v0, Lcom/enei/eneimobile/views/GameActivity$Location;-->hash:Ljava/lang/String;
375
376 const/4 v4, 0x0
377
378 invoke-interface {v1, v3, v4}, Landroid/content/SharedPreferences;-->getBoolean(Ljava/lang/String;Z)Z
379
380 move-result v3
381
382 #if-eqz v3, :cond_0
383
384 .line 141
385 igeget v3, v0, Lcom/enei/eneimobile/views/GameActivity$Location;-->view:I
386
387 invoke-virtual {p0, v3}, Lcom/enei/eneimobile/views/GameActivity;-->findViewById(I)Landroid/view/View;
```
- Search Bar:** At the bottom, there is a search bar with the text "Find: alpha".
- Right Panel:** On the right side, there is a vertical panel containing several tabs or sections, likely related to the current file's analysis or resources.

# Exploit 3 - Recompile

File Edit View Bookmarks Settings Help

```
→ oposeczadas apktool b com.enei.eneimobile_2018-03-24
I: Using Apktool 2.3.4
I: Checking whether sources has changed...
I: Smaling smali folder into classes.dex...
I: Checking whether resources has changed...
I: Building resources...
S: WARNING: Could not write to (/home/anon/.local/share/apktool/framework), using /tmp instead...
S: Please be aware this is a volatile directory and frameworks could go missing, please utilize --frame-path if the default storage directory is unavailable
I: Building apk file...
I: Copying unknown files/dir...
I: Built apk...
→ oposeczadas █
```

# Exploit 3 - Compare

File View Navigation Tools Help

The screenshot shows the JADX interface with the file tree on the left and the code editor on the right. The code editor displays the `com.enei.eneimobile.views.GameActivity` class. The code is as follows:

```
        }
    }

    private void checkSuccessfulCodes() {
        SharedPreferences defaultSharedPreferences = PreferenceManager.getDefaultSharedPreferences(this);
        for (Location location : newLocations) {
            if (defaultSharedPreferences.getBoolean(location.hash, false)) {
                findViewById(location.view).setAlpha(1.0f);
            }
        }
    }
}
```

JADX memory usage: 0.53 GB of 4.00 GB

File View Navigation Tools Help

The screenshot shows the JADX interface with the file tree on the left and the code editor on the right. The code editor displays the `com.enei.eneimobile.views.GameActivity` class. A search bar at the top of the code editor has the word "checks" entered. The code is identical to the one in the first screenshot.

```
        }
    }

    private void checkSuccessfulCodes() {
        SharedPreferences defaultSharedPreferences = PreferenceManager.getDefaultSharedPreferences(this);
        for (Location location : newLocations) {
            defaultSharedPreferences.getBoolean(location.hash, false);
            findViewById(location.view).setAlpha(1.0f);
        }
    }

    private void checkWon() {
    }
}
```

JADX memory usage: 0.35 GB of 4.00 GB

# Exploit 3 - Sign && Install

File Edit View Bookmarks Settings Help

```
→ dist sign_apk com.enei.eneimobile_2018-03-24.apk  
jar signed.
```

Warning:

The signer's certificate is self-signed.

```
→ dist adb install com.enei.eneimobile_2018-03-24.apk
```

Success

```
→ dist █
```

# Solve 3



# Practical - Vuln App

Ways to obtain APKs:

- Source:
  - [https://github.com/zezadas/0xOPOSEC\\_0x73\\_VulnApp\\_Android](https://github.com/zezadas/0xOPOSEC_0x73_VulnApp_Android)
- Pull from android phone:
  - \$ adb shell pm list packages | grep -i vulnapp
  - \$ adb shell pm path pt.oposec.vulnapp
  - \$ adb pull  
/data/app/pt.oposec.vulnapp-1/base.apk



# VulnApp

The screenshot shows a mobile application interface with a dark green header bar at the top. On the left side of the header are three small icons: a person, a network signal, and a battery. On the right side are two more icons: a signal strength and a battery level, followed by the time '11:27'. Below the header, the main content area has a light gray background. It contains two text input fields: one labeled 'Username' with the value 'john' and another labeled 'Password' with the value '123456'. Both inputs have red underline highlights. Below these fields is a large, light gray button with the word 'LOGIN' centered in capital letters.

Username

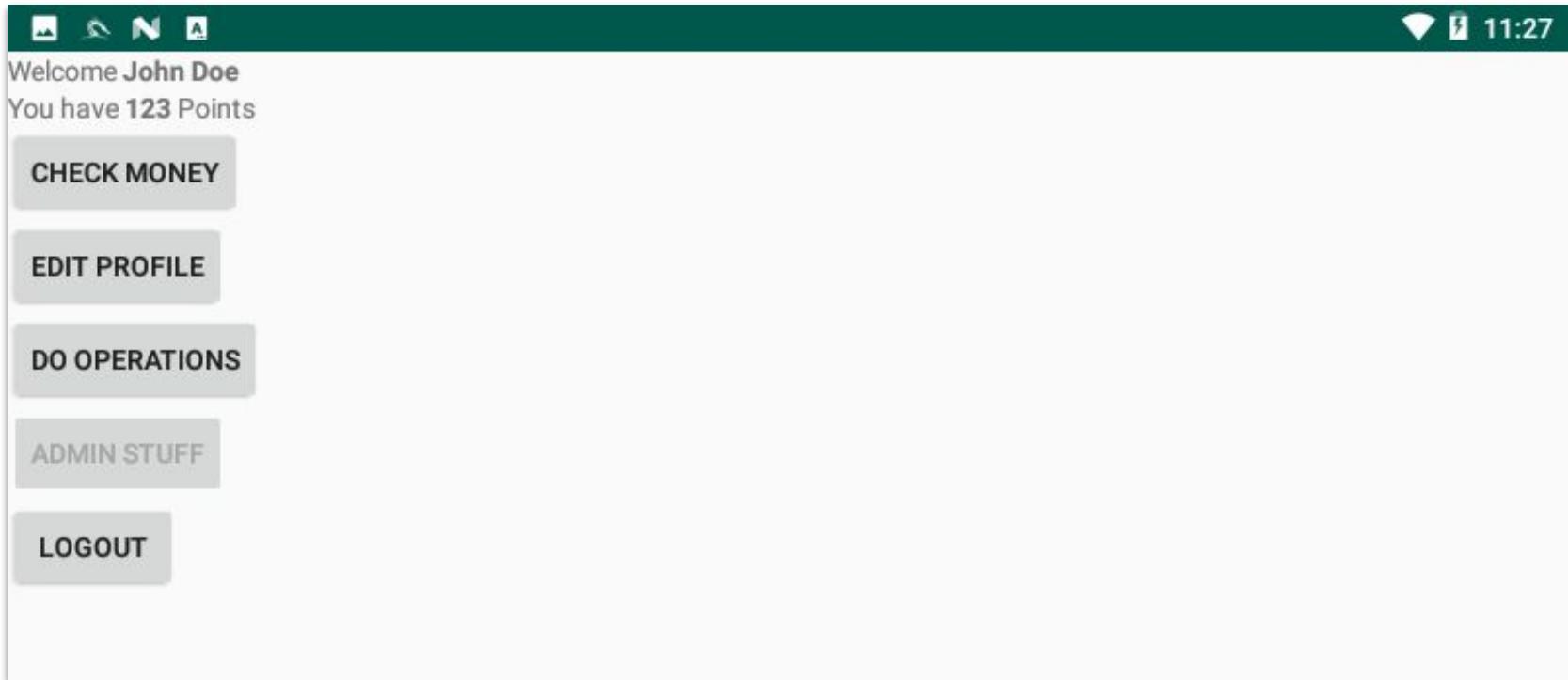
john

Password

123456

LOGIN

# VulnApp - Valid User



# VulnApp - Wrong User

The image shows a screenshot of an Android mobile application. At the top, there is a dark green header bar with several icons: a square, a person, a 'N' for network, and a 'A' for battery level. On the right side of the header, it shows signal strength, battery level, and the time '11:27'. Below the header is a white login form. It has two text input fields: 'Username' and 'Password'. The 'Username' field contains the text 'johnwrong' and is underlined with a red line, indicating it is the current focus or has errors. The 'Password' field contains the text '123456'. Below the password field, a message says 'Wrong Password 4 attempts left.' In the center of the form is a large grey button with the word 'LOGIN' in capital letters.

Username

johnwrong

Password

123456

Wrong Password 4 attempts left.

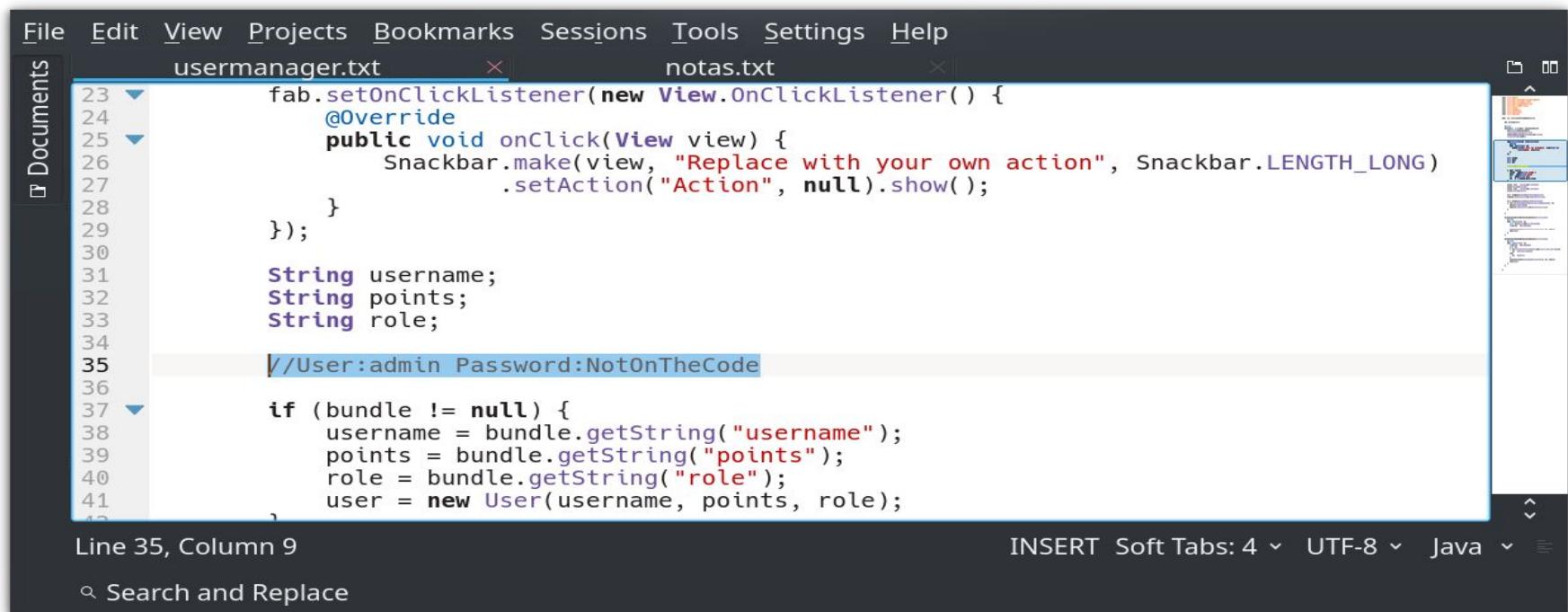
LOGIN

# VulnApp - AndroidManifest.xml

The screenshot shows a code editor interface with the following details:

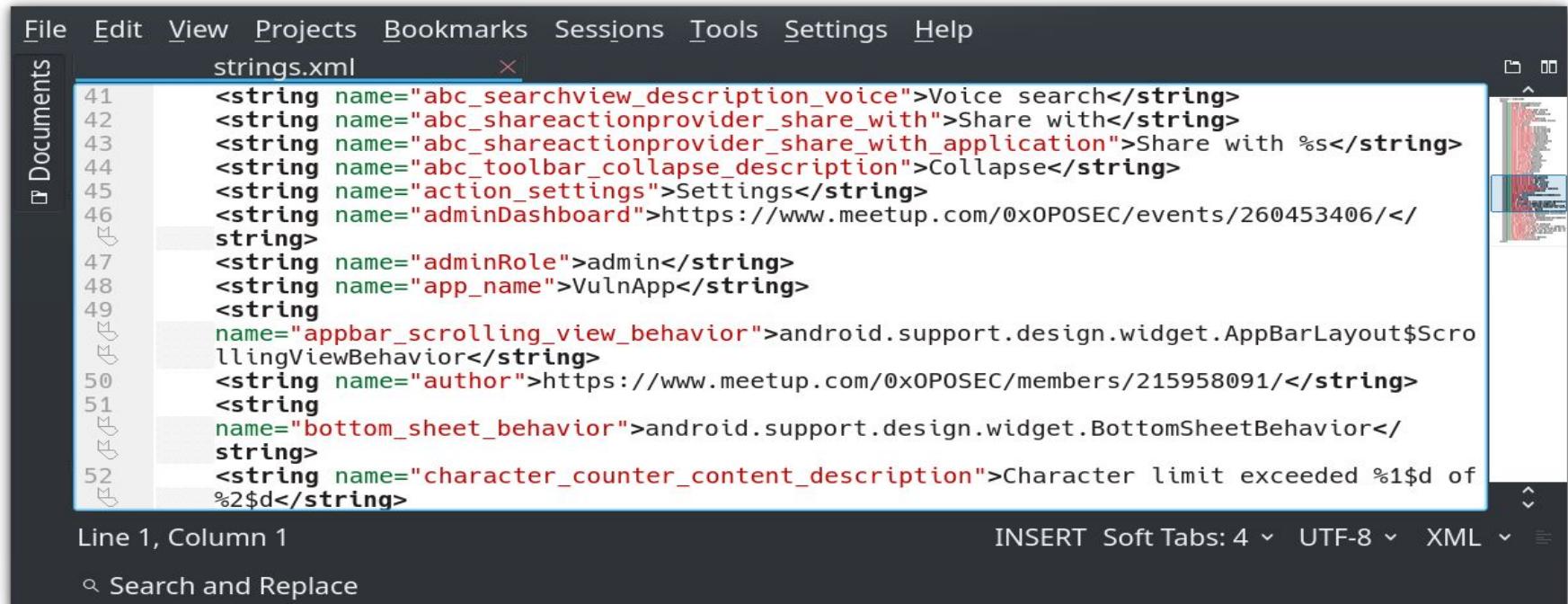
- File Menu:** File, Edit, View, Projects, Bookmarks, Sessions, Tools, Settings, Help.
- Documents:** Shows two tabs: "AndroidManifest.xml" and "notas.txt".
- Code Editor:** The "AndroidManifest.xml" tab displays the XML manifest file. Lines 1 through 12 are visible, with line numbers on the left.
- Annotations:** Three specific lines are highlighted with red circles:
  - Line 4: The attribute `android:allowBackup="true"`.
  - Line 5: The attribute `android:debuggable="true"`.
  - Line 10: The attribute `android:exported="true"`.
- Status Bar:** Shows "Line 12, Column 12" at the bottom-left, "INSERT Soft Tabs: 4" and "UTF-8" at the bottom-right, and "XML" in the status bar.
- Bottom Bar:** Shows a search bar with the placeholder "Search and Replace".

# VulnApp - Information left inside APK



```
File Edit View Projects Bookmarks Sessions Tools Settings Help
Documents usermanager.txt x notas.txt x
23 fab.setOnClickListener(new View.OnClickListener() {
24     @Override
25     public void onClick(View view) {
26         Snackbar.make(view, "Replace with your own action", Snackbar.LENGTH_LONG)
27             .setAction("Action", null).show();
28     }
29 );
30
31     String username;
32     String points;
33     String role;
34
35     //User:admin Password:NotOnTheCode
36
37     if (bundle != null) {
38         username = bundle.getString("username");
39         points = bundle.getString("points");
40         role = bundle.getString("role");
41         user = new User(username, points, role);
42     }
Line 35, Column 9
INSERT Soft Tabs: 4 ▾ UTF-8 ▾ Java ▾
Search and Replace
```

# VulnApp - Information left inside APK



The screenshot shows a code editor interface with a menu bar (File, Edit, View, Projects, Bookmarks, Sessions, Tools, Settings, Help) and a toolbar on the left labeled "Documents". The main window displays the contents of a file named "strings.xml". The code is as follows:

```
File Edit View Projects Bookmarks Sessions Tools Settings Help
strings.xml ×
41 <string name="abc_searchview_description_voice">Voice search</string>
42 <string name="abc_shareactionprovider_share_with">Share with</string>
43 <string name="abc_shareactionprovider_share_with_application">Share with %s</string>
44 <string name="abc_toolbarCollapse_description">Collapse</string>
45 <string name="action_settings">Settings</string>
46 <string name="adminDashboard">https://www.meetup.com/0xOPOSEC/events/260453406/</string>
47 <string name="adminRole">admin</string>
48 <string name="app_name">VulnApp</string>
49 <string
50   name="appbar_scrolling_view_behavior">android.support.design.widget.AppBarLayout$Scro
51   llingViewBehavior</string>
52 <string name="author">https://www.meetup.com/0xOPOSEC/members/215958091/</string>
<string
53   name="bottom_sheet_behavior">android.support.design.widget.BottomSheetBehavior</
54   string>
<string name="character_counter_content_description">Character limit exceeded %1$d of
55   %2$d</string>
```

The code editor status bar at the bottom indicates "Line 1, Column 1" and "INSERT Soft Tabs: 4 ▾ UTF-8 ▾ XML ▾". A search bar at the bottom left says "Search and Replace".

# VulnApp - Information left inside APK

The screenshot shows the IDA Pro debugger interface with the following details:

- File menu:** File, Edit, Jump, Search, View, Debugger, Options, Windows, Help.
- Toolbars:** Standard toolbar with icons for file operations, search, and debugger controls.
- Windows:**
  - Functions window:** Shows a list of function names, many of which are highlighted in pink, indicating they are part of the standard C++ library or NDK.
  - IDB View-A:** Pseudocode view showing the assembly code for the current function.
  - Strings window:** Hex dump of the string "This could be more interesting. But it's empty".
  - Hex View-1:** Hex dump of the assembly code.
  - Structures:** Enums, Imports, Exports.
- Registers:** Registers A1, V1, V2, V3, V4, V5, V6, V7, V8, V9, V10, V11, V12, V13, V14, V15, V16, V17, V18, V19, V20, V21, V22, V23, V24, V25, V26, V27, V28, V29, V30, V31, V32.
- Memory dump:** Shows the memory dump for the string "This could be more interesting. But it's empty".
- Output window:** Displays the message "762C: using guessed type int \_\_fastcall std::basic\_string<char, std::char\_traits<char>, std::allocator<char>>;~basic\_string(\_DWORD);".
- Python tab:** Shows the status "AU: idle".

**Assembly Code (IDA View-A):**

```
1 int __fastcall Java_pt_oposec_vulnapp_MainActivity_stringFromJNI(JNIEnv *a1)
2 {
3     int v1; // ST14_4
4     int result; // r0
5     int v3; // [sp+4h] [bp-A4h]
6     const char *v4; // [sp+8h] [bp-A0h]
7     JNIEnv *v5; // [sp+38h] [bp-70h]
8     int v6; // [sp+90h] [bp-18h]
9     int v7; // [sp+94h] [bp-14h]
10    const char *v8; // [sp+98h] [bp-10h]
11    int v9; // [sp+9Ch] [bp-Ch]
12
13    v5 = a1;
14    v1 = 0;
15    v6 = 0;
16    v7 = 0;
17    v1 = std::__ndk1::char_traits<char>::length("This could be more interesting. But it's empty");
18    std::__ndk1::basic_string<char, std::__ndk1::char_traits<char>, std::__ndk1::allocator<char>>::__init(
19        &v6,
20        "This could be more interesting. But it's empty",
21        v1);
22    if ((unsigned __int8)v6 << 31)
23        v4 = v8;
24    else
25        v4 = (char *)&v6 + 1;
26    v3 = JNIEnv::NewStringUTF(v5, v4);
27    std::__ndk1::basic_string<char, std::__ndk1::char_traits<char>, std::__ndk1::allocator<char>>::~basic_string(&v6);
28    result = __stack_chk_guard;
29    if (__stack_chk_guard == v9)
30        result = v3;
31    return result;
32 }
```

**Memory Dump (Strings window):**

```
00007B94 Java_pt_oposec_vulnapp_MainActivity_stringFromJNI:20 (7B94) This could be more interesting. But it's empty
```

# VulnApp - Dev features

The screenshot shows a development environment for an Android application named 'VulnApp'. The interface includes a top navigation bar with File, View, Navigation, Tools, and Help menus, along with various toolbar icons. On the left is a file browser showing the project structure under 'base-dex2jar.jar' and 'Source code'. The main area contains three tabs: 'p007pt.oposec.vulnapp.LoginFragment', 'p007pt.oposec.vulnapp.MainActivity', and 'p007pt.oposec.vulnapp'. The current tab is 'LoginFragment'. A search bar at the top of the code editor says 'Find: reset'. The code shown is:

```
/* renamed from: pt.oposec.vulnapp.LoginFragment$LoginButtonLongClick */
private class LoginButtonLongClick implements OnLongClickListener {
    private LoginButtonLongClick() {
    }

    /* synthetic */ LoginButtonLongClick(LoginFragment loginFragment, C06701 c06701) {
        this();
    }

    public boolean onLongClick(View view) {
        if (LoginFragment.this.DEBUG) {
            LoginFragment.this.resetAttempts();
            Toast.makeText(LoginFragment.this.getContext(), "Attempts reset!", 0).show();
        }
        return true;
    }

    private void CheckCredentialsOnInternet(String str, String str2) {
        User user = new User(str, str2);
        this.apiService.getUser(user.username, user).enqueue(new C06701());
    }
}
```

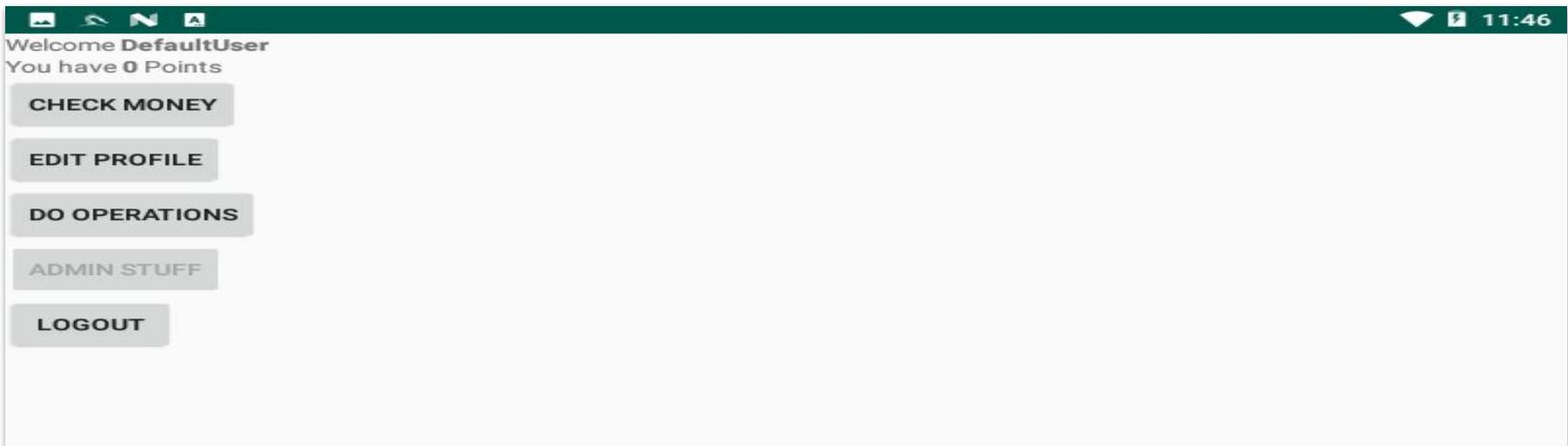
# VulnApp - Debuggable Feature (1/2)

```
File Edit View Bookmarks Settings Help
anon@unknown ~/oposec/vulnapp/base adb shell run-as pt.oposec.vulnapp find .
.
./cache
./cache/0ECE53DABA921940.toc
./cache/0ECE53DABA921940.bin
./code_cache
./lib
./files
./files/config.txt
anon@unknown ~/oposec/vulnapp/base adb shell run-as pt.oposec.vulnapp cat ./files/config.txt
42
anon@unknown ~/oposec/vulnapp/base _
```

# VulnApp - Debuggable Feature (2/2)

```
File Edit View Bookmarks Settings Help
anon@unknown ~/oposec > adb shell run-as pt.oposec.vulnapp
echo 99 > ./files/config.txt
cat ./files/config.txt
99
```

# VulnApp - Exported Activity



```
adb shell am start -n pt.oposec.vulnapp/.UserActivity
```

# VulnApp - Backup/Restore Feature (1/2)

```
File Edit View Bookmarks Settings Help
anon@unknown ~/oposec ➤ adb backup -f backup.ab pt.oposec.vulnapp
Now unlock your device and confirm the backup operation...
anon@unknown ~/oposec ➤ abe unpack backup.ab backup_opo.tar 123456
Calculated MK checksum (use UTF-8: true): 764D23C2736F81C0787764034E17492E27BDE03A3DDDF395A36B80B8A237A476
43% 53%
3584 bytes written to backup_opo.tar.
anon@unknown ~/oposec ➤ x backup_opo.tar
apps/pt.oposec.vulnapp/_manifest
apps/pt.oposec.vulnapp/f/config.txt
anon@unknown ~/oposec ➤ _
```

# VulnApp - Backup/Restore Feature (2/2)

File Edit View Bookmarks Settings Help

```
anon@unknown ~$ echo 99 > apps/pt.oposec.vulnapp/f/config.txt  
anon@unknown ~$ tar -tf backup_opo.tar > backup_opo.list  
anon@unknown ~$ cat backup_opo.list | grep pt.oposec.vulnapp > vulnapp.list  
anon@unknown ~$ cat vulnapp.list | pax -wd > vulnapp.tar  
anon@unknown ~$ abe pack vulnapp.tar vulnapp.ab 123456
```

10240 bytes written to vulnapp.ab.

```
anon@unknown ~$ adb restore vulnapp.ab
```

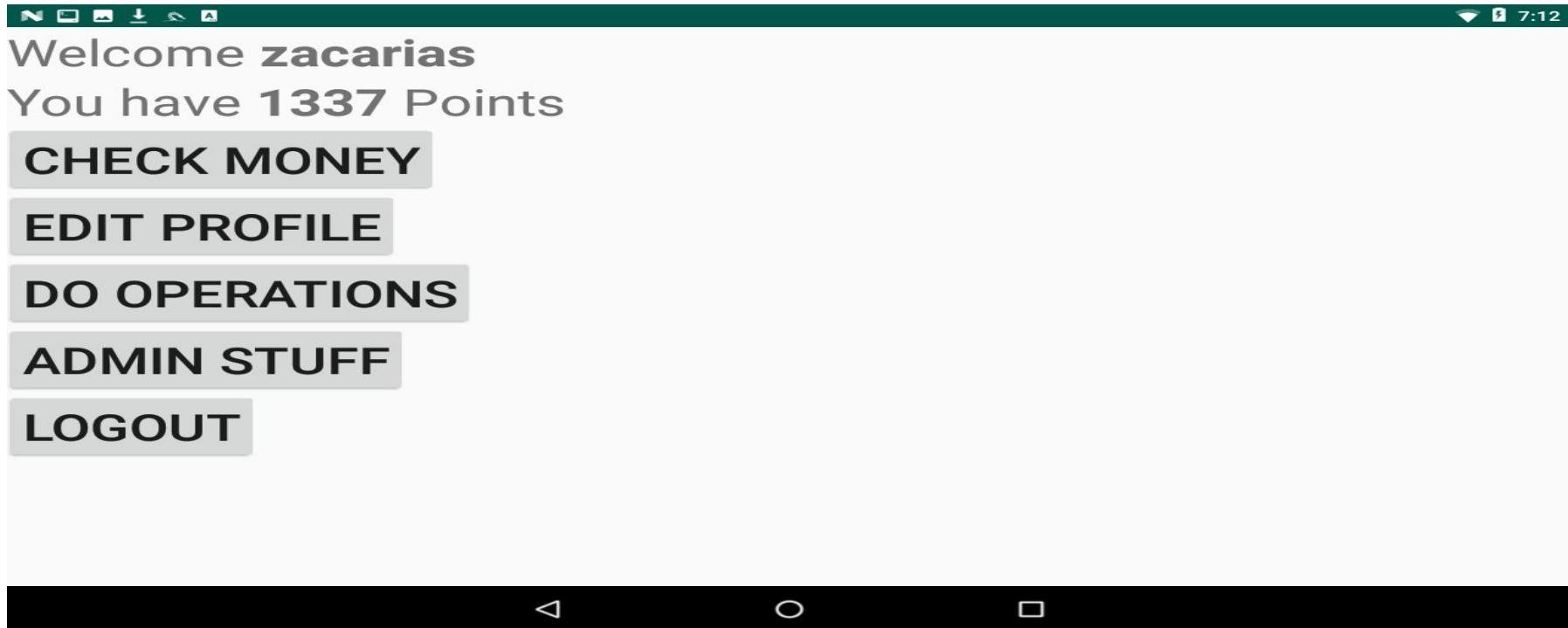
Now unlock your device and confirm the restore operation.

```
anon@unknown ~$ _
```

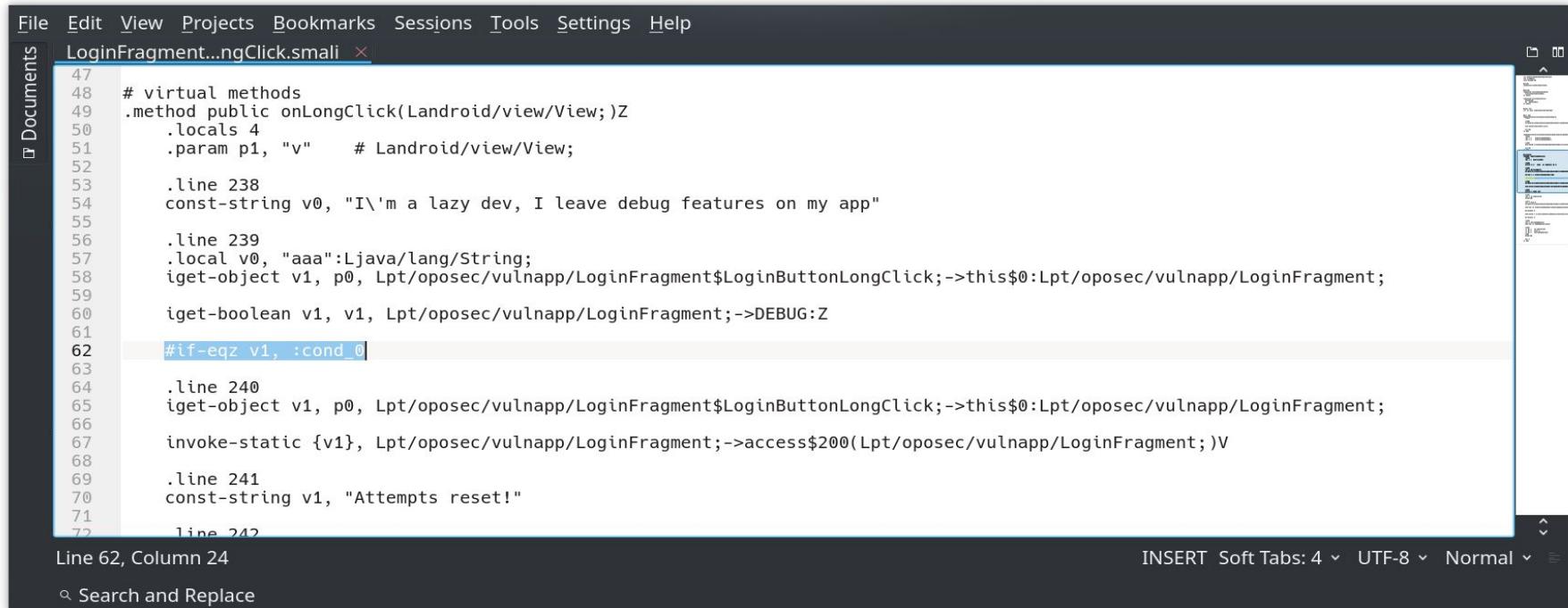
# VulnApp - Exported Activity (1/2)

```
File Edit View Bookmarks Settings Help
anon@unknown ~/oposec/original/dist > adb shell am start -n pt.oposec.vulnapp/.UserActivity --es "username" "zacarias" --es "points" "1337"
--es "role" "admin"
Starting: Intent { cmp=pt.oposec.vulnapp/.UserActivity (has extras) }
anon@unknown ~/oposec/original/dist >
```

# VulnApp - Exported Activity (2/2)



# VulnApp - Patch & Recompile (1/2)



The screenshot shows a code editor interface with the following details:

- File Menu:** File, Edit, View, Projects, Bookmarks, Sessions, Tools, Settings, Help.
- Document Title:** LoginFragment...ngClick.smali
- Code Content:** The code is a decompiled Java method in Smali format. It handles a long click event on a login button. The code includes comments, string constants, local variable declarations, and bytecode instructions. A specific line (Line 62) is highlighted with a light gray background.
- Editor Status:** Line 62, Column 24. Insert mode is active. Soft Tabs: 4, UTF-8, Normal.
- Search Bar:** Search and Replace.

```
File Edit View Projects Bookmarks Sessions Tools Settings Help
LoginFragment...ngClick.smali ×
Documents
47
48 # virtual methods
49 .method public onLongClick(Landroid/view/View;)Z
50     .locals 4
51     .param p1, "v"    # Landroid/view/View;
52
53     .line 238
54     const-string v0, "I'm a lazy dev, I leave debug features on my app"
55
56     .line 239
57     .local v0, "aaa":Ljava/lang/String;
58     ige-object v1, p0, Lpt/oposec/vulnapp/LoginFragment$LoginButtonLongClick;->this$0:Lpt/oposec/vulnapp/LoginFragment;
59
60     ige-boolean v1, v1, Lpt/oposec/vulnapp/LoginFragment;->DEBUG:Z
61
62     #if-eqz v1, :cond_0
63
64     .line 240
65     ige-object v1, p0, Lpt/oposec/vulnapp/LoginFragment$LoginButtonLongClick;->this$0:Lpt/oposec/vulnapp/LoginFragment;
66
67     invoke-static {v1}, Lpt/oposec/vulnapp/LoginFragment;->access$200(Lpt/oposec/vulnapp/LoginFragment;)V
68
69     .line 241
70     const-string v1, "Attempts reset!"
71
72     .line 242
```

# VulnApp - Patch & Recompile (2/2)

```
File Edit View Bookmarks Settings Help
anon@unknown ~/oposec apktool b original
I: Using Apktool 2.3.4
I: Checking whether sources has changed...
I: Smaling smali folder into classes.dex...
I: Checking whether sources has changed...
I: Checking whether resources has changed...
I: Building apk file...
I: Copying unknown files/dir...
I: Built apk...
anon@unknown ~/oposec cd original/dist
anon@unknown ~/oposec/original/dist assina original.apk
jar signed.

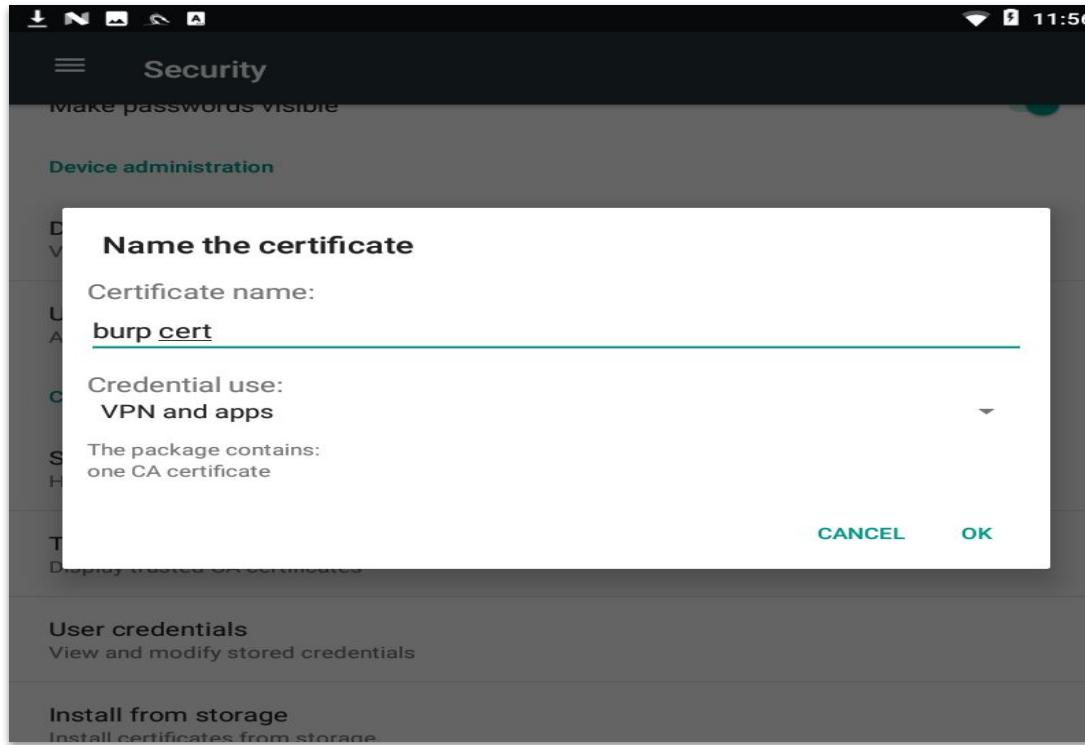
Warning:
The signer's certificate is self-signed.
No -tsa or -tsacert is provided and this jar is not timestamped. Without a timestamp, users may not be able to validate this jar after
the signer certificate's expiration date (2045-01-01) or after any future revocation date.
anon@unknown ~/oposec/original/dist adb install original.apk
Success
anon@unknown ~/oposec/original/dist _
```

# VulnApp - No certificate Pinning

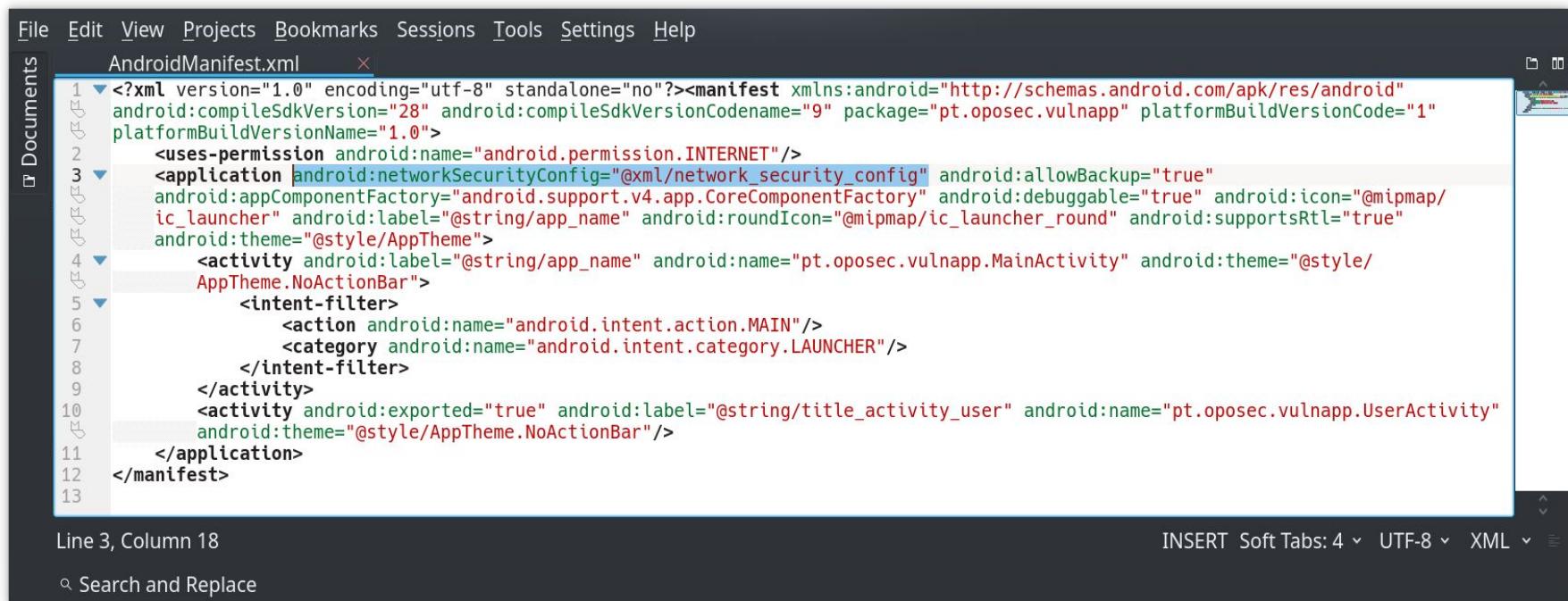
The screenshot shows an IDE interface with the following details:

- File menu:** File, View, Navigation, Tools, Help.
- Toolbar:** Standard file operations like Open, Save, Find, etc.
- Sidebar:** Shows project structure with packages like base-dex2jar.jar, Source code, and Resources. Under Source code, there are several Java files: ApiService, BuildConfig, C0675R, LoginFragment, MainActivity, MainActivityFragment, User, UserActivity, and UserActivity. A method named "onCreate(Bundle)" is highlighted in the sidebar.
- Code Editor:** The main window displays the code for `LoginFragment`. The code includes:
  - A `String` builder for logging errors.
  - An `onAttach` method that calls `super.onAttach(context)`.
  - An `onCreate` method that creates a `Builder` for the API service, setting the URL to `C0675R.string.API_ENDPOINT` and adding a converter factory.
  - An `onCreateView` method that inflates the layout `C0675R.layout.fragment_login`, finds views by ID (`ButtonLogin`, `UserTxt`, `PassTxt`, `MessageLogin`), and sets click listeners for the login button.

# Traffic Interception - Install CA must have SDK < 23



# Traffic Interception - Install CA must have SDK >= 23 (1/4)



The screenshot shows an AndroidManifest.xml file being edited in an IDE. The XML code is color-coded for syntax: green for tags like <manifest>, <application>, <activity>, <uses-permission>, <intent-filter>, <action>, and <category>; red for attributes like android:compileSdkVersion="28", android:allowBackup="true", android:debuggable="true", android:icon="@mipmap/ic\_launcher", android:label="@string/app\_name", android:roundIcon="@mipmap/ic\_launcher\_round", android:supportsRtl="true", android:theme="@style/AppTheme.NoActionBar", android:name="pt.oposec.vulnapp.MainActivity", android:exported="true", android:label="@string/title\_activity\_user", and android:name="pt.oposec.vulnapp.UserActivity"; and blue for namespaces like xmlns:android="http://schemas.android.com/apk/res/android". There are several syntax errors highlighted in red, particularly around the <application> tag and its nested elements. The code includes declarations for network security configuration, permissions, activities, and intent filters.

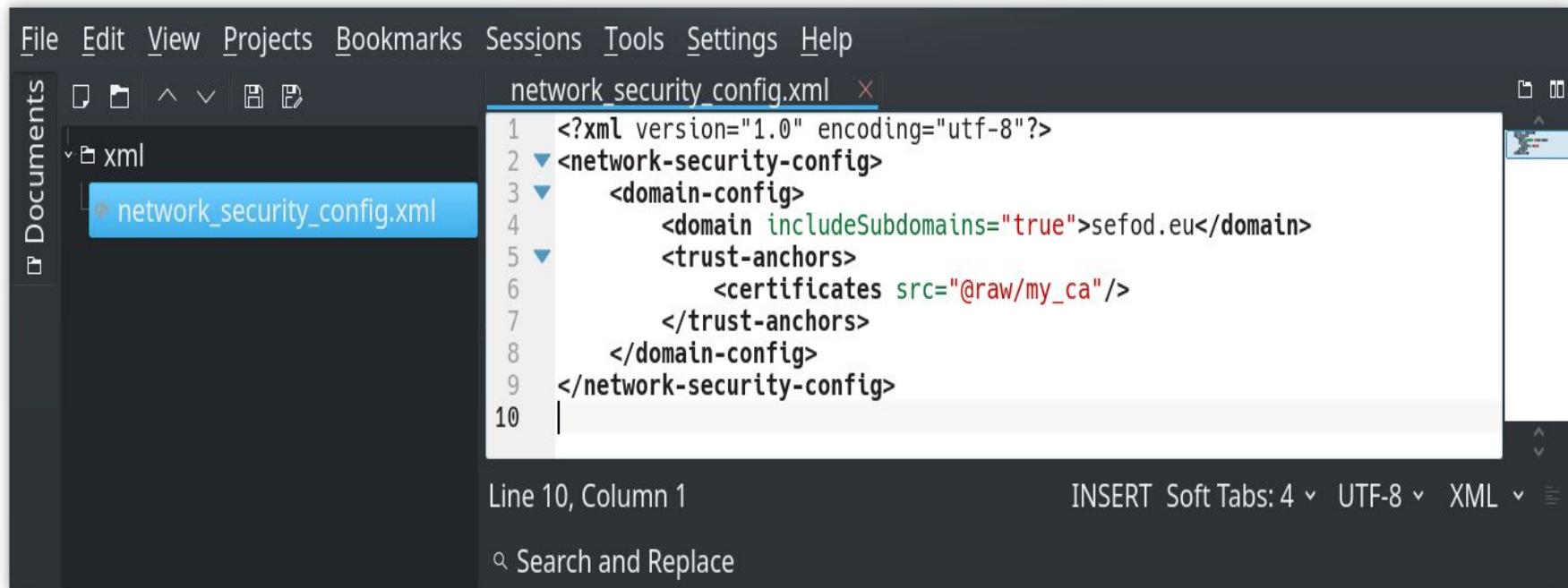
```
<?xml version="1.0" encoding="utf-8" standalone="no"?><manifest xmlns:android="http://schemas.android.com/apk/res/android"  
    android:compileSdkVersion="28" android:compileSdkVersionCodename="9" package="pt.oposec.vulnapp" platformBuildVersionCode="1"  
    platformBuildVersionName="1.0">  
    <uses-permission android:name="android.permission.INTERNET"/>  
    <application android:networkSecurityConfig="@xml/network_security_config" android:allowBackup="true"  
        android:appComponentFactory="android.support.v4.app.CoreComponentFactory" android:debuggable="true" android:icon="@mipmap/  
        ic_launcher" android:label="@string/app_name" android:roundIcon="@mipmap/ic_launcher_round" android:supportsRtl="true"  
        android:theme="@style/AppTheme">  
        <activity android:label="@string/app_name" android:name="pt.oposec.vulnapp.MainActivity" android:theme="@style/  
        AppTheme.NoActionBar">  
            <intent-filter>  
                <action android:name="android.intent.action.MAIN"/>  
                <category android:name="android.intent.category.LAUNCHER"/>  
            </intent-filter>  
        </activity>  
        <activity android:exported="true" android:label="@string/title_activity_user" android:name="pt.oposec.vulnapp.UserActivity"  
            android:theme="@style/AppTheme.NoActionBar"/>  
    </application>  
</manifest>
```

Line 3, Column 18

SEARCH AND REPLACE

INSERT Soft Tabs: 4 ▾ UTF-8 ▾ XML ▾

# Traffic Interception - Install CA must have SDK >= 23 (2/4)



The screenshot shows the Android Studio interface with the code editor open to the file `network_security_config.xml`. The file contains XML configuration for network security, specifically defining a domain configuration for the subdomain `sefod.eu` and specifying a certificate source from a raw resource named `my_ca`.

```
<?xml version="1.0" encoding="utf-8"?>
<network-security-config>
    <domain-config>
        <domain includeSubdomains="true">sefod.eu</domain>
        <trust-anchors>
            <certificates src="@raw/my_ca"/>
        </trust-anchors>
    </domain-config>
</network-security-config>
```

The code editor interface includes a menu bar with File, Edit, View, Projects, Bookmarks, Sessions, Tools, Settings, Help. On the left is a sidebar with Documents, XML, and the selected file `network_security_config.xml`. At the bottom are status bars for Line 10, Column 1, and settings for INSERT, Soft Tabs: 4, UTF-8, XML, and a search bar.

# Traffic Interception - Install CA must have SDK >= 23 (3/4)

```
File Edit View Bookmarks Settings Help
anon@unknown:~/oposec/original/res/raw$ cp /home/anon/my_ca ./my_ca
anon@unknown:~/oposec/original/res/raw$
```

# Traffic Interception - Install CA must have SDK >= 23 (4/4)

```
File Edit View Bookmarks Settings Help
anon@unknown ~/oposec apktool b original
I: Using Apktool 2.3.4
I: Checking whether sources has changed...
I: Smaling smali folder into classes.dex...
I: Checking whether sources has changed...
I: Checking whether resources has changed...
I: Building apk file...
I: Copying unknown files/dir...
I: Built apk...
anon@unknown ~/oposec cd original/dist
anon@unknown ~/oposec/original/dist assina original.apk
jar signed.

Warning:
The signer's certificate is self-signed.
No -tsa or -tsacert is provided and this jar is not timestamped. Without a timestamp, users may not be able to validate this jar after
the signer certificate's expiration date (2045-01-01) or after any future revocation date.
anon@unknown ~/oposec/original/dist adb install original.apk
Success
anon@unknown ~/oposec/original/dist _
```

# OWASP - Mobile Top 10 2016

- M1-Improper Platform Usage
- M2-Insecure Data Storage
- M3-Insecure Communication
- M4-Insecure Authentication
- M5-Insufficient Cryptography
- M6-Insecure Authorization
- M7-Poor Code Quality
- M8-Code Tampering
- M9-Reverse Engineering
- M10-Extraneous Functionality



**OWASP**

Open Web Application  
Security Project

# Other Resources

- Qark - <https://github.com/linkedin/qark>
- Drozer - <https://github.com/mwrlabs/drozer>
- Frida - <https://github.com/frida>/
- Android-security-awesome - <https://github.com/ashishb/android-security-awesome>
- OWASP ASVS -  
[https://www.owasp.org/index.php/Category:OWASP\\_Application\\_Security\\_Verification\\_Standard\\_Project](https://www.owasp.org/index.php/Category:OWASP_Application_Security_Verification_Standard_Project)
- OWASP Mob Testing Guide -  
[https://www.owasp.org/index.php/OWASP\\_Mobile\\_Security\\_Testing\\_Guide](https://www.owasp.org/index.php/OWASP_Mobile_Security_Testing_Guide)



**END**