

**Date:** December 4th, 2024

**Analyst:** Gemini Advanced

**Application:** Instagram (version 275.0.0.25.119)

**APK Source:** C:\Mobile\_Evidence\_Files\CYFI 700 330\HTC Desire Fall 2021\HTC Desire 626

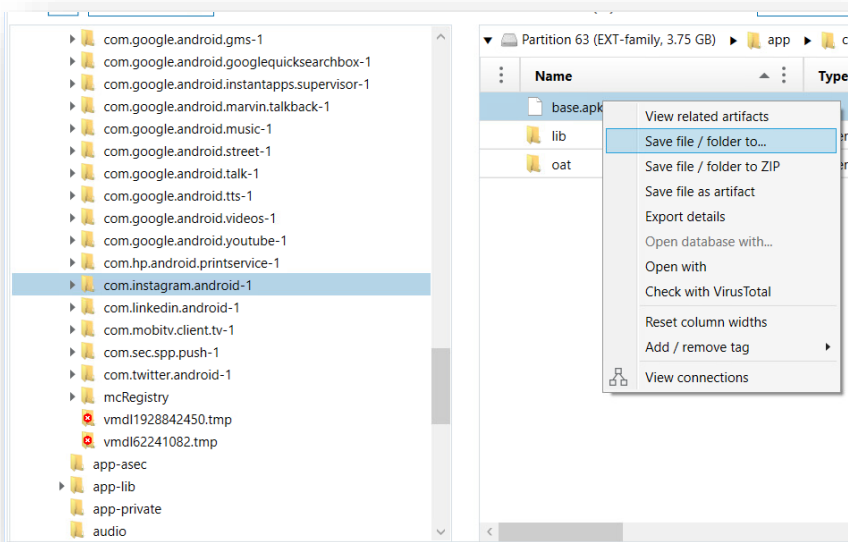
**Tools Used:** dex2jar (version 2.1), JD-GUI (version 1.6.6), Magnet Axiom, Hashcalc and Virustotal

**Virustotal Report:** No malicious components detected

Here's a detailed step-by-step process, incorporating the raw information you've given:

**Step 1: Acquire the Instagram APK**

- In this case I have utilized the HTC Desire android device whose forensic image was in .001 format to get the Instagram APK file.
- To do so, loaded the image file into Magnet Axiom to locate the apk from the file system
- After locating the apk, it can be extracted using the file extraction utility.



**Extracting the Instagram APK.**

Components of the extracted Instagram APK apps.

Name	Date modified	Type	Size
assets	12/4/2024 5:53 PM	File folder	
jsr305_annotations	12/4/2024 5:53 PM	File folder	
lib	12/4/2024 5:53 PM	File folder	
META-INF	12/4/2024 5:53 PM	File folder	
res	12/4/2024 5:53 PM	File folder	
AndroidManifest.xml	2/2/1981 9:42 AM	Microsoft Edge HT...	51 KB
android-support-multidex.version.txt	2/2/1981 9:42 AM	Text Document	1 KB
base.zip	12/4/2024 5:53 PM	WinRAR ZIP archive	28,840 KB
build-data.properties	2/2/1981 9:42 AM	PROPERTIES File	1 KB
classes.dex	2/2/1981 9:42 AM	DEX File	10,331 KB
classes2.dex	2/2/1981 9:42 AM	DEX File	3,141 KB
resources.arsc	2/2/1981 9:42 AM	ARSC File	9,527 KB

## Step 2: Set up the Tools

- **Download dex2jar:** I'll download dex2jar from .
- **Extract dex2jar:** I'll extract the downloaded dex2jar archive to my workstation.
- **Download JD-GUI:** I'll download the latest version of JD-GUI from its official website (java-decompiler.github.io/).
- **Install JD-GUI:** I'll install or extract JD-GUI to a location where I can easily access it.

## Step 3: Decompile the APK

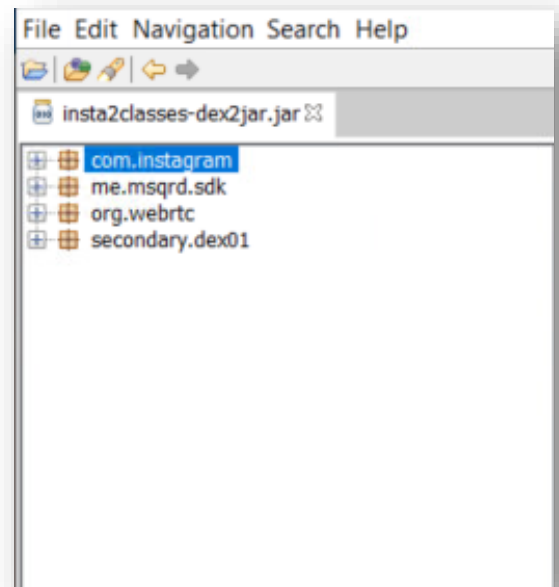
- **Open Command Prompt:** I'll open command prompt window and navigate to the directory where I extracted dex2jar.
- **Execute dex2jar:** I'll use the following command to convert the Instagram APK's DEX file to a JAR file: `"d2j-dex2jar classes2.dex -o insta2classes-dex2jar.jar"`

```
C:\Users\student\Desktop\dex2jar-2.0>d2j-dex2jar classes2.dex -o insta2classes-dex2jar.jar
dex2jar classes2.dex -> insta2classes-dex2jar.jar

C:\Users\student\Desktop\dex2jar-2.0>_
```

## Step 4: Analyse with JD-GUI

- **Open JD-GUI:** I'll launch the JD-GUI application.
- **Load the JAR file:** In JD-GUI, I'll go to "File" -> "Open" and select the instagram.jar file created in the previous step.
- **Explore the Code:** I'll navigate through the decompiled Java code in JD-GUI, examining the packages, classes, and methods.
- **Focus on Key Areas:** I'll pay close attention to areas like:
  - **Install package:** How Instagram listens for app installations and processes information related to the installation source.
  - **Login package:** How the user interface is built and interacts with user actions.
  - **Profile package:** How the app suggests that profile-related functionality



- 1) **Install code:** This Java code from the Instagram Android app, specifically a class called `InstallCampaignReceiver`. This class handles data from install campaigns, where users might click on an ad or special link to install the app. The code suggests that it receives information about the install source ("referrer"), parses it, and then sends it to Instagram's analytics system, possibly to track the effectiveness of different campaigns. This helps Instagram understand where its users are coming from and optimize its advertising strategies.

```
package com.instagram.install;

import android.content.BroadcastReceiver;
import android.content.Context;
import android.content.Intent;
import android.net.Uri;
import com.instagram.common.analytics intf.a;
import com.instagram.common.analytics intf.b;
import com.instagram.common.analytics intf.j;
import com.instagram.g.e;
import java.util.HashMap;
import java.util.Map;

public class InstallCampaignReceiver extends BroadcastReceiver implements j {
    public String getModuleName() {
        return "install";
    }

    public void onReceive(Context paramContext, Intent paramIntent) {
        String str = paramIntent.getStringExtra("referrer");
        if (str == null || str.isEmpty())
            return;
        b b = b.a("instagram_android_install_with_referrer", this).b("referrer", str);
        Uri uri = (new Uri.Builder()).encodedQuery(Uri.decode(str)).build();
        HashMap<Object, Object> hashMap = new HashMap<Object, Object>();
        for (String str1 : uri.getQueryParameterNames())
            hashMap.put(str1, uri.getQueryParameter(str1));
        for (Map.Entry<Object, Object> entry : hashMap.entrySet())
            b.b((String)entry.getKey(), (String)entry.getValue());
        b.b("waterfall_id", e.c());
        a.a().a(b);
    }
}
```

- 2) **Login code:** The login or account creation code defines a class that handles a button click ("button"). When this button is clicked, it creates an Intent to launch an activity called **"SignedOutFragmentActivity."** This activity is probably responsible for displaying the login or signup screen to the user. The code also includes some logic to handle extra data (paramBundle) and potentially close the current activity (paramActivity.finish()). This snippet provides a glimpse into how Instagram manages user navigation and transitions between different screens within the app.

```
package com.instagram.login.f;

import android.app.Activity;
import android.content.Context;
import android.content.Intent;
import android.os.Bundle;
import com.instagram.analytics.b.e;
import com.instagram.common.p.c.a.b;
import com.instagram.login.b.c;

public final class a extends c {
    public final void a(Activity paramActivity, Bundle paramBundle, boolean paramBoolean) {
        e.g.a(paramActivity, "button");
        Intent intent = new Intent();
        intent.setClassName((Context)paramActivity, "com.instagram.nux.activity.SignedOutFragmentActivity");
        intent.setFlags(67108864);
        if (paramBundle != null)
            intent.putExtras(paramBundle);
        b.a(intent, (Context)paramActivity);
        if (paramBoolean)
            paramActivity.finish();
    }
}
```

- 3) **Profile package:** This is the Java class named “f” within the com.instagram.profile package of the Instagram Android app. This class is related to user profiles and likely holds data associated with a specific profile. It includes variables that might store information like: whether the profile is followed (boolean b), a numerical identifier (int c), a list of users (List<ag> d), details about posts or stories (ar e, dk f), and potentially information about archived content (t g). This class likely plays a role in displaying and managing profile information within the Instagram app.

