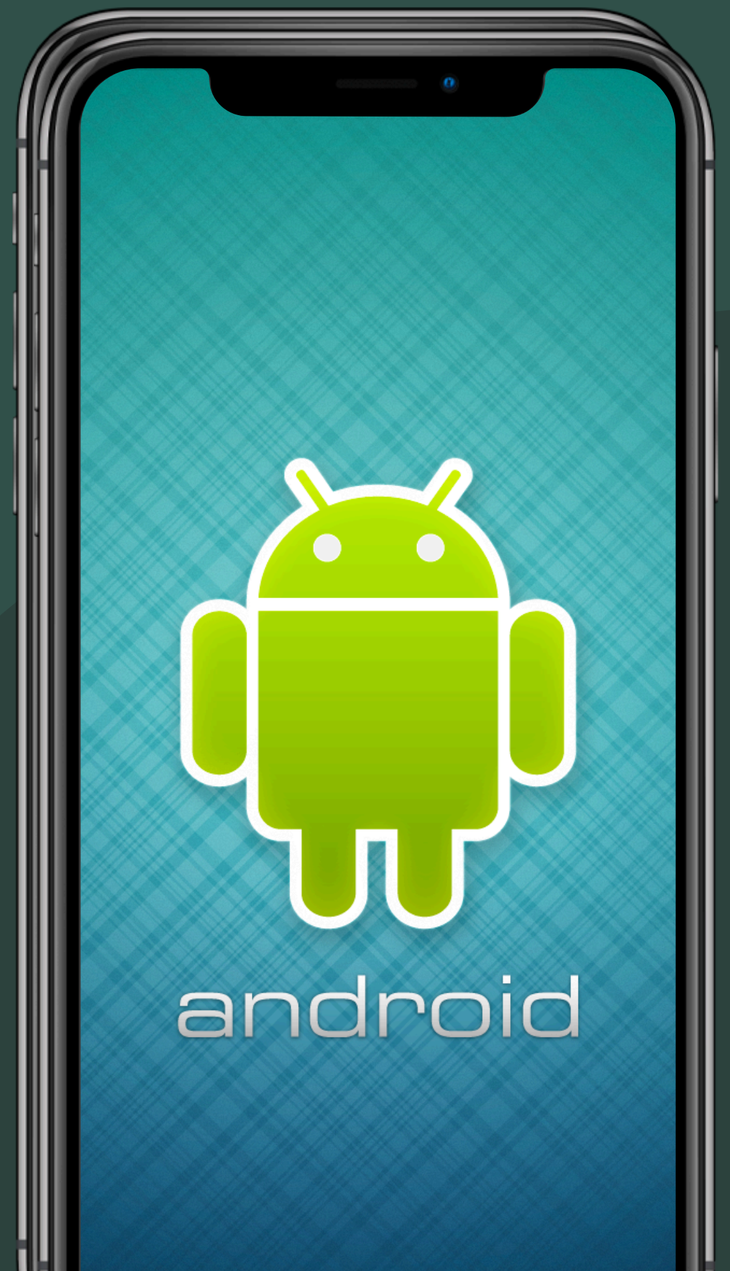
Static & Dynamic Software Security Analysis Project

ZHUSUPOV DANIIAR
BAPE ANSER

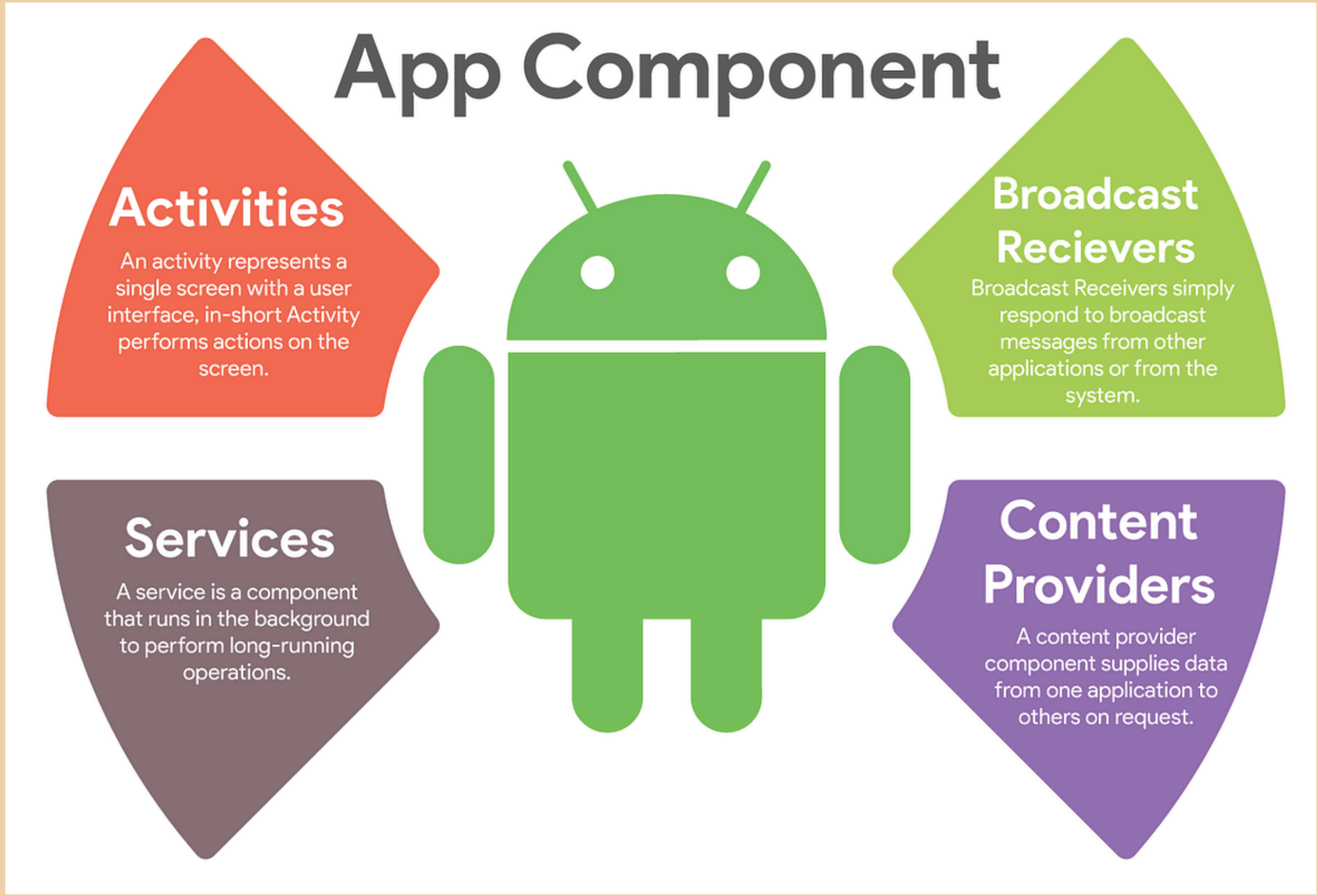


Section 1: Introduction to FlowDroid for Taint Analysis

Content

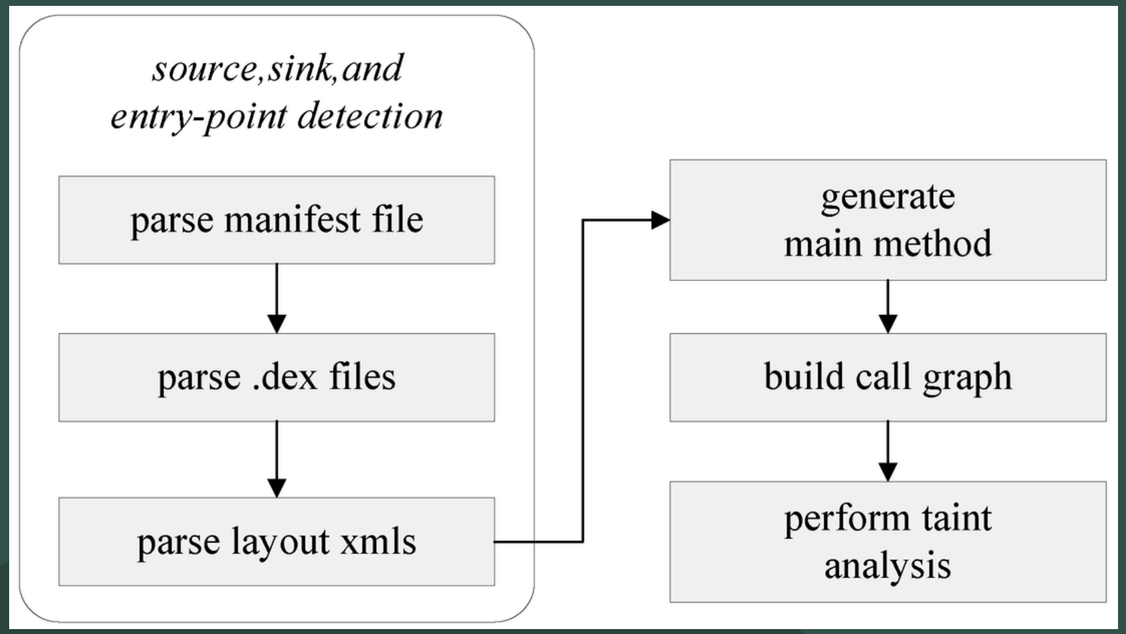
- Section 1: Introduction to FlowDroid for Taint Analysis
- Android App Components
- FlowDroid and Configuration Options
- Sources and Sinks
- FlowDroid Analysis
- Hardware and Configuration
- Results
- Optional task
- Section 2: Privacy Requirements Compliance Checking in Android Apps
- Conclusion

Android app components



FlowDroid

- Static taint analysis tool
- Designed for Android applications.
- Helps to identify how sensitive data ("taints") flows through an application by tracing its sources and sinks.
- Highly configurable.



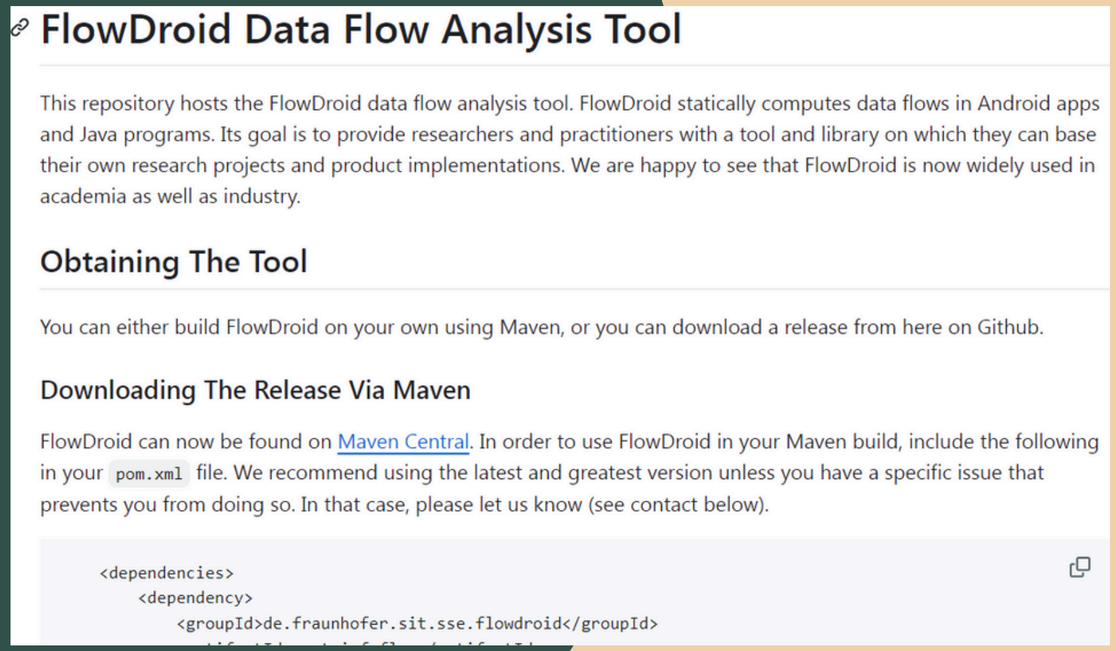
Timeout settings, Memory and resource allocation, Flow sinks and sources, Granularity and depth, Output options.

Sources and Sinks

Source – points in the code where sensitive data originates.
Examples: device data (e.g., location, IMEI, contacts, photos).

Sink – points in the code where data is used in a potentially unsafe or sensitive way. Examples: system logs (e.g., Log.d, Log.e in Android),sending data via SMS or email.

Setup and Hardware configuration



FlowDroid Data Flow Analysis Tool

This repository hosts the FlowDroid data flow analysis tool. FlowDroid statically computes data flows in Android apps and Java programs. Its goal is to provide researchers and practitioners with a tool and library on which they can base their own research projects and product implementations. We are happy to see that FlowDroid is now widely used in academia as well as industry.

Obtaining The Tool

You can either build FlowDroid on your own using Maven, or you can download a release from here on Github.

Downloading The Release Via Maven

FlowDroid can now be found on Maven Central. In order to use FlowDroid in your Maven build, include the following in your `pom.xml` file. We recommend using the latest and greatest version unless you have a specific issue that prevents you from doing so. In that case, please let us know (see contact below).

```
<dependencies>
    <dependency>
        <groupId>de.fraunhofer.sit.sse.flowdroid</groupId>
```

- FlowDroid.jar v2.13 with dependencies
- JDK 17
- Android.jar v11
- OS: Lubuntu 64-bit.
- Allocated Resources: RAM: 4096 MB (4 GB). CPU: 4 cores allocated with a CPU usage cap of 100%. Disk Space: 20 GB Virtual Disk (SATA interface).
- Graphics: Controller: VMSVGA. Video Memory: 16 MB.
- Networking: Intel PRO/1000 MT Desktop adapter (NAT mode).

Used command

```
java -jar /home/micsispa/Downloads/soot-infoflow-cmd-2.13.0-jar-with-dependencies.jar \
    -a /home/micsispa/Downloads/APKs/com.delhi.metro.dtc.apk \
    -p /home/micsispa/Android-platforms/jars/stubs/android-11/android.jar \
    -s /home/micsispa/Downloads/SourcesAndSinks.txt \
    --timeout 60 > apk160.txt 2>&1
```

```
^Cmicsispa@micsispa:~$ java -jar /home/micsispa/Downloads/soot-infoflow-cmd-2.13.0-jar-with-dependencies.jar     -a /home/micsispa/Downloads/APKs/com.delhi.metro.dtc.apk     -p /home/micsispa/Android-platforms/jars/stubs/android-11/androi
jar     -s /home/micsispa/Downloads/SourcesAndSinks.txt     --timeout 60 > apk160.txt 2>&1
micsispa@micsispa:~$ java -jar /home/micsispa/Downloads/soot-infoflow-cmd-2.13.0-jar-with-dependencies.jar     -a /home/micsispa/Downloads/APKs/com.delhi.metro.dtc.apk     -p /home/micsispa/Android-platforms/jars/stubs/android-11/android.
jar     -s /home/micsispa/Downloads/SourcesAndSinks.txt     --timeout 300 > apk1300.txt 2>&1
micsispa@micsispa:~$ java -jar /home/micsispa/Downloads/soot-infoflow-cmd-2.13.0-jar-with-dependencies.jar     -a /home/micsispa/Downloads/APKs/com.delhi.metro.dtc.apk     -p /home/micsispa/Android-platforms/jars/stubs/android-11/android.
jar     -s /home/micsispa/Downloads/SourcesAndSinks.txt     --timeout 1200 > apk11200.txt 2>&1
```

Analysis Results

№	App	Number of leaks (60 sec)	Number of leaks (300 sec)	Number of leaks (1200 sec)
1	com.delhi.metro.dtc	3	3	3
2	com.hawaiianairlines.app	33	33	33
3	com.imo.android.imoim	4	4	4
4	com.tado	21	21	21
5	com.walkme.azores.new	4	4	4
6	com.wooxhome.smart	0	0	0
7	com.yourdelivery.pyszne	5	5	5
8	linko.home	24	24	24
9	mynt.app	11	11	11
10	nz.co.stuff.android.news	26	26	26

com.delhi.metro.dtc

- Source: $d0 = virtualinvoke $r6.<android.location.Location: double getLatitude()>. $d0 = virtualinvoke $r6.<android.location.Location: double getLongitude()> - captures location data (getLatitude() and getLongitude()).

- Sink: staticinvoke <android.util.Log: int d(java.lang.String,java.lang.String)>("sandeep", $r8) - data is being logged using android.util.Log.

com.hawaiianairlines.app

- Source: virtualinvoke $r8.<android.net.wifi.WifiInfo: java.lang.String getSSID()> - sensitive data: Wi-Fi SSID.

- Sink: virtualinvoke $r0.<androidx.activity.ComponentActivity: void startActivityForResult(android.content.Intent,int)> - found in multiple activities such as MediaActivity, FavoritesActivity, and ShoppingActivity.

com.imo.android.imoim

- Source: virtualinvoke $r8.<android.content.pm.PackageManager: java.util.List queryBroadcastReceivers(android.content.Intent,int)> - information queried from the PackageManager, potentially related to broadcast receivers.

- Sink: staticinvoke <android.util.Log: int i(java.lang.String,java.lang.String)>("AppsFlyer_6.11.0", $r2) - data is logged using android.util.Log.

com.tado

- Source: virtualinvoke $r9.<java.net.HttpURLConnection: java.io.InputStream getInputStream()> - reads data from an HTTP connection's input stream.

- Sink: staticinvoke <android.util.Log: int d(java.lang.String,java.lang.String) > ("AccountResponse", $r1) - logs sensitive data as part of API response debugging.

com.tado

- Source: virtualinvoke r0. <com.tado.android.installation.CreateAccountActivity: android.view.View findViewById(int)> - extracts UI elements during account creation.

- Sink: virtualinvoke $r12.<java.io.OutputStream: void write(byte[])> - writes data to an output stream.

com.walkme.azores.new

- Source: virtualinvoke $r4.<java.util.Locale: java.lang.String getCountry()> - retrieves the user's country code from the system locale.

- Sink: interfaceinvoke $r7.<android.content.SharedPreferences$Editor: android.content.SharedPreferences$Editor putString(java.lang.String,java.lang.String)> - stores the country code in shared preferences.

com.linko.home

- Source: $r9 = virtualinvoke $r1.<java.io.File: java.io.File getAbsoluteFile()> - accesses the absolute file path.

- Sink: virtualinvoke $r16.<java.io.FileOutputStream: void write(byte[])> - writes the file to an output stream.

com.linko.home

- Source: $r10 = interfaceinvoke $r13.<android.database.Cursor: java.lang.String getString(int)> - retrieves data from a database cursor.

- Sink: virtualinvoke $r6.<android.os.Bundle: void putString(java.lang.String,java.lang.String)> - stores sensitive data in a Bundle.

com.mynt.app

- Source: $r21 = virtualinvoke $r19. <android.content.pm.PackageManager: java.util.List queryIntentServices(android.content.Intent,int)> - retrieves information about services from the package manager.

- Sink: staticinvoke <android.util.Log: int d(java.lang.String,java.lang.String)> - logs sensitive service query results.

nz.co.stuff.android.news

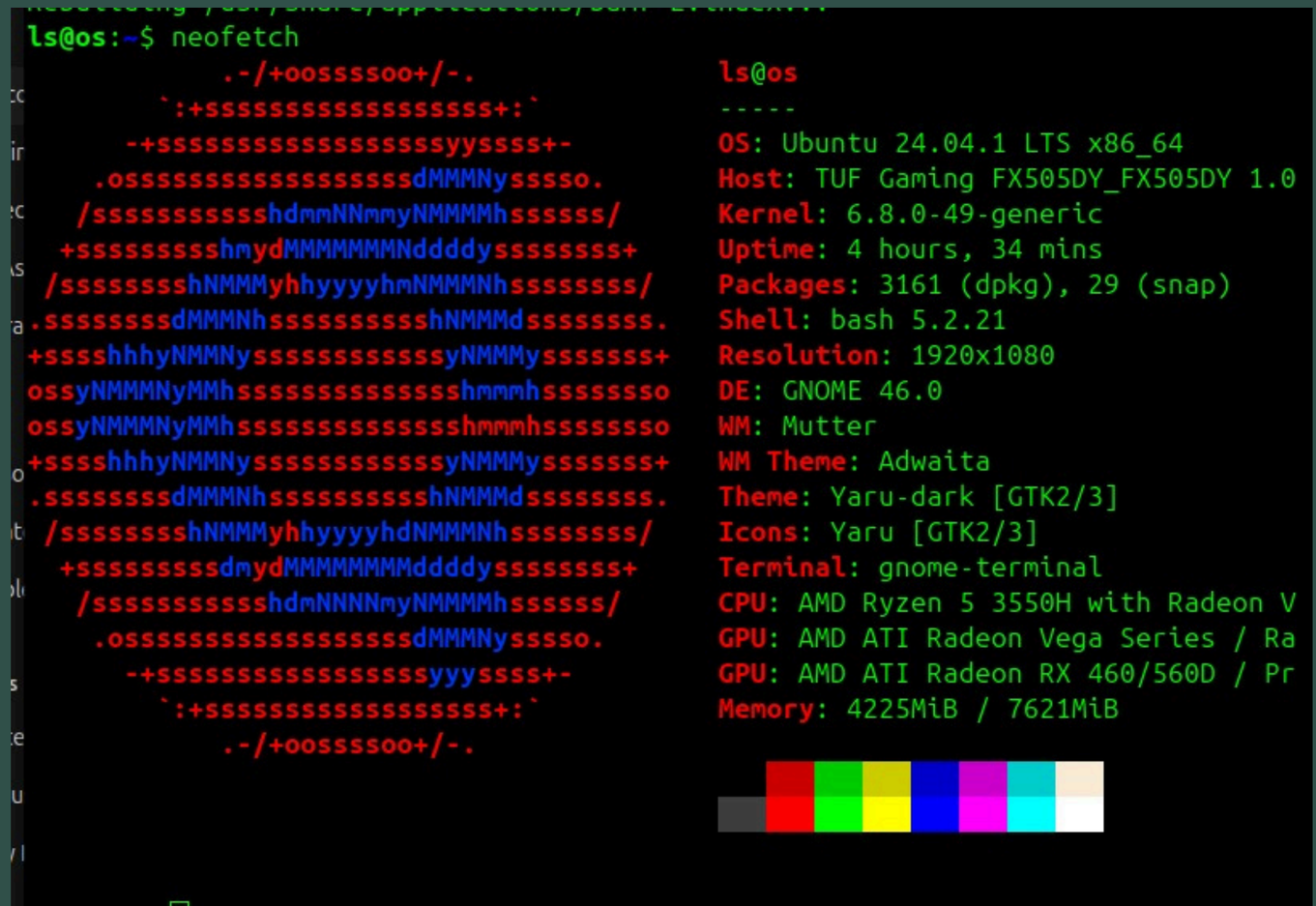
- Source: $r7 = interfaceinvoke $r6.<android.database.Cursor: java.lang.String getString(int)> - retrieves data from a database cursor.

- Sink: virtualinvoke $r3.<java.io.Writer: void write(java.lang.String,int,int)>($r2, 0, $i0) - writes the data into a file.

Optional task

Configurations:

- FlowDroid.jar v2.13 with dependencies
- JDK 21
- Android.jar v24



Optional task

Configurations:

- FlowDroid.jar v2.13 with dependencies
- JDK 11
- Android.jar v11

General

Name: MICS_ISPA_LABS_ORIGIN
Operating System: Ubuntu (64-bit)

System

Base Memory: 1900 MB
Boot Order: Optical, Hard Disk
Acceleration: Nested Paging, KVM Paravirtualization

Display

Video Memory: 16 MB
Graphics Controller: VMSVGA
Remote Desktop Server: Disabled
Recording: Disabled

Preview

Storage

Controller: IDE
IDE Secondary Device 0: [Optical Drive] Empty
Controller: SATA
SATA Port 0: MICS_ISPA_LABS-disk001.vdi (Normal, 20.00 GB)

Audio

Host Driver: PulseAudio
Controller: ICH AC97

Network

Adapter 1: Intel PRO/1000 MT Desktop (NAT)

USB

USB Controller: OHCI, EHCI
Device Filters: 0 (0 active)

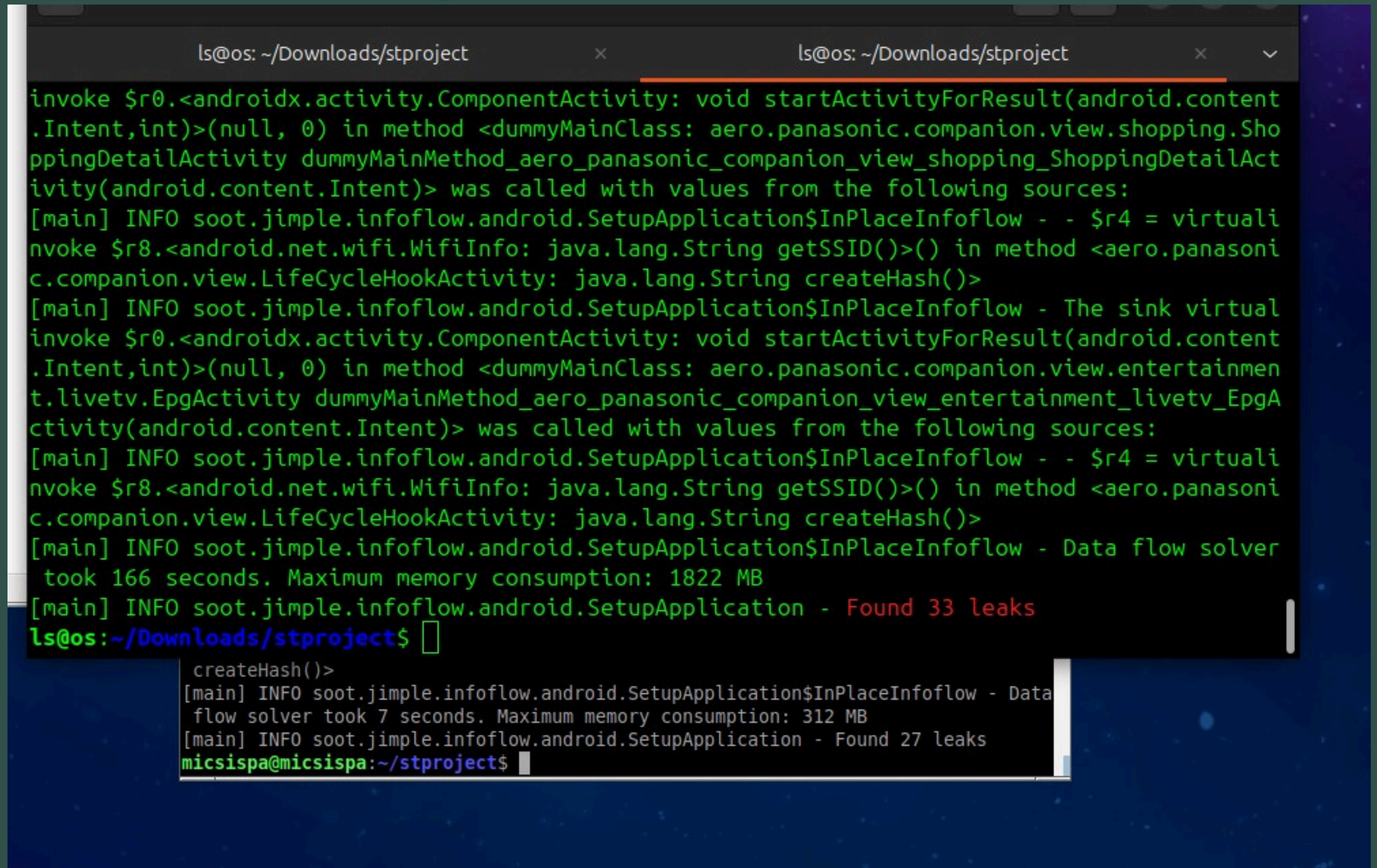
Shared folders

None

Description

None

Implementaion



Optional task results

№	App	Ubuntu 24.04	Professor's VM
1	com.delhi.metro.dtc	3	3
2	com.hawaiianairlines.app	33	27
3	com.imo.android.imoim	4	4
4	com.tado	21	14
5	com.walkme.azores.new	4	4
6	com.wooxhome.smart	0	0
7	com.yourdelivery.pyszne	5	5
8	linko.home	24	24
9	mynt.app	11	10
10	nz.co.stuff.android.news	26	26

Section 2: Privacy Requirements Compliance Checking in Android Apps

Analysis of Privacy and GDPR Compliance

Account Deletion & Data Removal Process

Privacy Requirement 1 (R1)

Objective: Verify if location data is sent outside the app (potential data leak).

- Methodology: Static analysis using FlowDroid, a taint analysis tool.
- Sources: Methods retrieving location data, such as android.location.Location and LocationManager.
- Sinks: Methods sending data over the network, like java.net.HttpURLConnection and org.apache.http.
- Configuration: Custom sources and sinks file (r1SourcesAndSinks.txt) created for FlowDroid analysis, with increased memory allocation for large app size.

We executed FlowDroid

*java -Xmx4g -jar soot-infoflow-cmd-2.13.0-jar-with-dependencies.jar *

Results of the Analysis for R1

The analysis found one potential data leak in the com.tado.apk application

```
ow.data.pathBuilders.DefaultPathBuilderFactory$RepeatableContextSensitivePathBuilder - Building path 1...
ow.data.pathBuilders.DefaultPathBuilderFactory$RepeatableContextSensitivePathBuilder - Building path 2...
ow.data.pathBuilders.DefaultPathBuilderFactory$RepeatableContextSensitivePathBuilder - Building path 3...
ow.data.pathBuilders.DefaultPathBuilderFactory$RepeatableContextSensitivePathBuilder - Building path 4...
ow.memory.MemoryWarningSystem - Shutting down the memory warning system...
ow.android.SetupApplication$InPlaceInfoflow - Memory consumption after path building: 436 MB
ow.android.SetupApplication$InPlaceInfoflow - Path reconstruction took 3 seconds
ow.android.SetupApplication$InPlaceInfoflow - The sink $r11 = virtualinvoke $r9.<java.net.HttpURLConnection
ow.android.SetupApplication$InPlaceInfoflow - - $r6 = virtualinvoke r0.<com.tado.android.LoginActivity: and
ow.android.SetupApplication$InPlaceInfoflow - - $r6 = virtualinvoke r0.<com.tado.android.LoginActivity: and
ow.android.SetupApplication$InPlaceInfoflow - Data flow solver took 18 seconds. Maximum memory consumption:
ow.android.SetupApplication - Found 1 leaks
```

Results for R1

Sink Method (where data goes):

- Method Involved: java.net.HttpURLConnection.getOutputStream()
- This method is used to send data over the network (e.g., to a server).

Source Methods (where data comes from):

- Source 1: findViewById(2131755329) inside LoginActivity.onCreate()
- Source 2: findViewById(2131755331) inside LoginActivity.onCreate()
- These methods are used to retrieve data, potentially location or user-related information (e.g., from UI elements).

Data Flow:

- The data retrieved from the sources (LoginActivity) is passed to the sink (HttpURLConnection), suggesting that this data is being sent over the network.

Assessment & Challenges

Assessment of Genuine Leak:

- Potential Leak: The data (e.g., user data or location) is being sent outside the app using the network, which could be a privacy violation.
- Consideration: If this is done without user consent or for unauthorized purposes, it’s a data leak.
- Further Investigation Needed: We need to verify if sending this data is legitimate, like for location-based services, and whether user consent was obtained.

Challenges:

- Memory issues and missing callback methods affected completeness.

Observation R1

Potential Data Leak: The analysis identified one potential data leak involving the HttpURLConnection class sending data over the network. This suggests that location data might be sent outside the app.

Compliance with R1: To ensure compliance with R1, the app should not send location data outside the app without user consent. The identified data leak needs further investigation to determine whether it is legitimate or a violation of privacy requirements

Privacy Requirement (R2)

Objective: Verify the existence of a mechanism for account deletion.

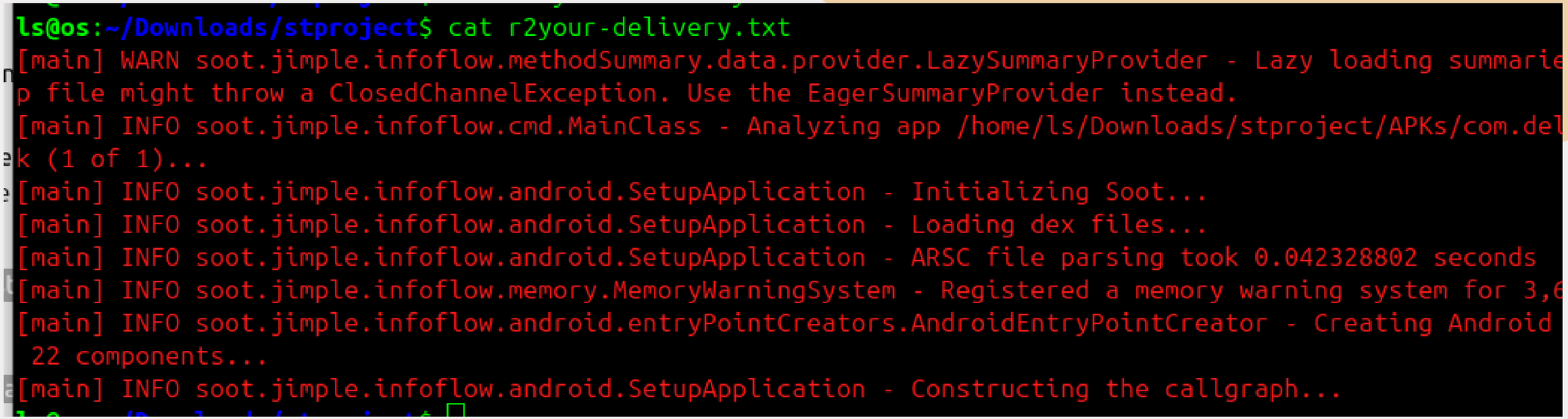
- **Methodology**: Custom sources and sinks file created to analyze potential account deletion flow.

```
# Sources
<com.yourdelivery.pyszne.AccountManager: void getUserProfile()> -> _SOURCE_
<com.yourdelivery.pyszne.AuthManager: String getAuthToken()> -> _SOURCE_

# Sinks
<com.yourdelivery.pyszne.AccountManager: void deleteAccount(java.lang.String)>
<com.yourdelivery.pyszne.ApiClient: void deleteUser(java.lang.String)> -> _SIN
```

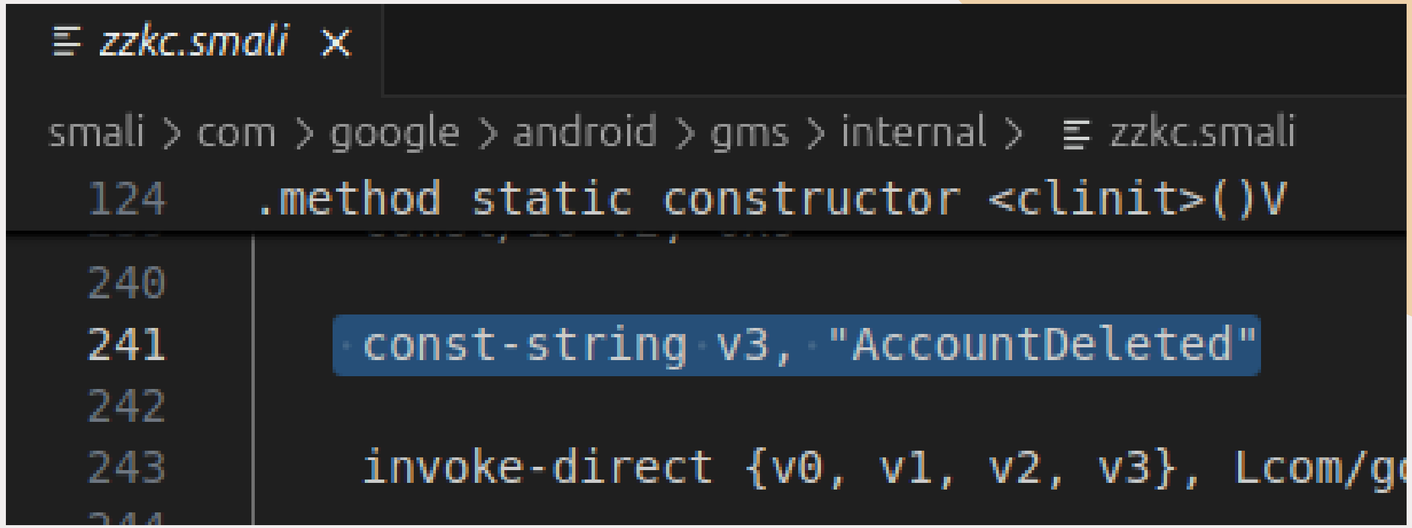
Privacy Requirement (R2)

- **Findings:** No direct evidence of an account deletion mechanism (client-side or server-side) found.
- **Next Steps:** Reverse engineer APK using apktool to inspect the Smali code for indirect or hidden references to account deletion.



Reverse Engineering and Code Review:

apktool d com.yourdelivery.pyszne.apk



The presence of ACCOUNT_DELETED suggests the app recognizes a state where a user account can be deleted

Reverse Engineering and Code Review:

grep removeUserCredentials

```
/yopeso/tteferando/activity
ls@os:~/Downloads/stproject$ grep -rEi "deleteAccount|removeUser|closeAccount" output_youdeliver-decompiled/smali/
output_youdeliver-decompiled/smali/com/kahuna/sdk/KahunaAnalytics$6.smali:    value = Lcom/kahuna/sdk/KahunaAnalytics
;->removeUserCredential(Ljava/lang/String;)V
output_youdeliver-decompiled/smali/com/kahuna/sdk/KahunaAnalytics$6.smali:    const-string v3, "Handled exception in
KahunaAnalytics.removeUserCredential(): "
output_youdeliver-decompiled/smali/com/kahuna/sdk/KahunaAnalytics.smali:.method public static removeUserCredential(Lj
```

The method name suggests that it is intended to remove user credentials

```
sFragment$DeleteAddressListener;-><init>(Lcom/yopeso/lieferando/fragments/account/MapAddressFragment;)V
-decompiled/smali/com/yopeso/lieferando/fragments/account/MapAddressFragment$DeleteAddressListener.smali:    iget-object v0, p0, Lcom/yopeso/lieferando/
gment$DeleteAddressListener;->this$0:Lcom/yopeso/lieferando/fragments/account/MapAddressFragment;
-decompiled/smali/com/yopeso/lieferando/fragments/account/MapAddressFragment$DeleteAddressListener.smali:    iget-object v0, p0, Lcom/yopeso/lieferando/
gment$DeleteAddressListener;->this$0:Lcom/yopeso/lieferando/fragments/account/MapAddressFragment;
-decompiled/smali/com/yopeso/lieferando/fragments/account/MapAddressFragment$DeleteAddressListener.smali:    iget-object v0, p0, Lcom/yopeso/lieferando/
gment$DeleteAddressListener;->this$0:Lcom/yopeso/lieferando/fragments/account/MapAddressFragment;
-decompiled/smali/com/yopeso/lieferando/fragments/account/MapAddressFragment$DeleteAddressListener.smali:    iget-object v0, p0, Lcom/yopeso/lieferando/
gment$DeleteAddressListener;->this$0:Lcom/yopeso/lieferando/fragments/account/MapAddressFragment;
-decompiled/smali/com/yopeso/lieferando/fragments/account/MapAddressFragment$DeleteAddressListener.smali:    iget-object v0, p0, Lcom/yopeso/lieferando/
gment$DeleteAddressListener;->this$0:Lcom/yopeso/lieferando/fragments/account/MapAddressFragment;
-decompiled/smali/com/yopeso/lieferando/fragments/account/MapAddressFragment$DeleteAddressListener.smali:    iget-object v0, p0, Lcom/yopeso/lieferando/
gment$DeleteAddressListener;->this$0:Lcom/yopeso/lieferando/fragments/account/MapAddressFragment;
-decompiled/smali/com/yopeso/lieferando/fragments/account/MapAddressFragment$DeleteAddressListener.smali:    iget-object v0, p0, Lcom/yopeso/lieferando/
gment$DeleteAddressListener;->this$0:Lcom/yopeso/lieferando/fragments/account/MapAddressFragment;
-decompiled/smali/com/yopeso/lieferando/fragments/account/MainAddressFragment$5.smali:    invoke-direct {v4, v5, v6}, Lcom/yopeso/lieferando/fragments/a
eAddressListener;-><init>(Lcom/yopeso/lieferando/fragments/account/MainAddressFragment;Lcom/yopeso/lieferando/fragments/account/MainAddressFragment$Dele

/stproject$ grep -rEi "delete.*account|remove.*user|terminate.*account|close.*account" output_youdeliver-decompiled/smali

-decompiled/smali/com/kahuna/sdk/KahunaAnalytics$6.smali:    value = Lcom/kahuna/sdk/KahunaAnalytics;->removeUserCredential(Ljava/lang/String;)V
-decompiled/smali/com/kahuna/sdk/KahunaAnalytics$6.smali:    const-string v2, "Cannot remove user credential with empty string or null key value."
-decompiled/smali/com/kahuna/sdk/KahunaAnalytics$6.smali:    const-string v3, "Handled exception in KahunaAnalytics.removeUserCredential(): "
-decompiled/smali/com/kahuna/sdk/KahunaAnalytics.smali:.method public static removeUserCredential(Ljava/lang/String;)V
-decompiled/smali/com/kahuna/sdk/KahunaInternalEventKeys.smali:.field protected static final REMOVE_USER_CREDENTIAL:Ljava/lang/String; = "k_remove_crede
-decompiled/smali/com/facebook/TestSession.smali:.method private deleteTestAccount(Ljava/lang/String;Ljava/lang/String;)V
-decompiled/smali/com/facebook/TestSession.smali:    const-string v6, "Could not delete test account %s: %s"
-decompiled/smali/com/facebook/TestSession.smali:    const-string v6, "Could not delete test account %s: unknown reason"
-decompiled/smali/com/facebook/TestSession.smali:    invoke-direct {p0, v0, v1}, Lcom/facebook/TestSession;->deleteTestAccount(Ljava/lang/String;Ljava/l
-decompiled/smali/com/yopeso/lieferando/fragments/account/MainAddressFragment$DeleteAddressListener.smali:    iput-object p1, p0, Lcom/yopeso/lieferando
agment$DeleteAddressListener;->this$0:Lcom/yopeso/lieferando/fragments/account/MainAddressFragment;
-decompiled/smali/com/yopeso/lieferando/fragments/account/MainAddressFragment$DeleteAddressListener.smali:    invoke-direct {p0, p1}, Lcom/yopeso/liefer
essFragment$DeleteAddressListener;-><init>(Lcom/yopeso/lieferando/fragments/account/MainAddressFragment;)V
-decompiled/smali/com/yopeso/lieferando/fragments/account/MainAddressFragment$DeleteAddressListener.smali:    iget-object v0, p0, Lcom/yopeso/lieferando
agment$DeleteAddressListener;->this$0:Lcom/yopeso/lieferando/fragments/account/MainAddressFragment;
-decompiled/smali/com/yopeso/lieferando/fragments/account/MainAddressFragment$DeleteAddressListener.smali:    iget-object v0, p0, Lcom/yopeso/lieferando
agment$DeleteAddressListener;->this$0:Lcom/yopeso/lieferando/fragments/account/MainAddressFragment;
-decompiled/smali/com/yopeso/lieferando/fragments/account/MainAddressFragment$DeleteAddressListener.smali:    iget-object v0, p0, Lcom/yopeso/lieferando
agment$DeleteAddressListener;->this$0:Lcom/yopeso/lieferando/fragments/account/MainAddressFragment;
```

Reverse Engineering and Code Review:

The removeUserCredential method:

```
.method public static removeUserCredential(Ljava/lang/String;)V
    .locals 3
    .param p0, "key"    # Ljava/lang/String;

    .prologue
    .line 677
    sget-object v0, Lcom/kahuna/sdk/KahunaAnalytics;->mKahunaExecutor:Ljava/util/concurrent/ExecutorService;

    new-instance v1, Lcom/kahuna/sdk/KahunaBackgroundRunnable;

    new-instance v2, Lcom/kahuna/sdk/KahunaAnalytics$6;

    invoke-direct {v2, p0}, Lcom/kahuna/sdk/KahunaAnalytics$6;-><init>(Ljava/lang/String;)V

    invoke-direct {v1, v2}, Lcom/kahuna/sdk/KahunaBackgroundRunnable;-><init>(Ljava/lang/Runnable;)V

    invoke-interface {v0, v1}, Ljava/util/concurrent/ExecutorService;->execute(Ljava/lang/Runnable;)V

    .line 715
    return-void
.end method
```

Delegates the work to KahunaAnalytics$6, which seems to handle the actual logic of credential removal.
Runs the task asynchronously in the background, ensuring it doesn't block the main thread.

Reverse Engineering and Code Review:

clearUserData

```
ls@os:~/Downloads/stproject$ grep -A 20 "clearUserData" output_youdeliver-decompiled/smali/
com/yopeso/lieferando/custom/LRActivity.smali
.method private clearUserData()V
    .locals 4
```

```
ls@os:~/Downloads/stproject$ grep -rE "(deleteAccount|removeAccount|clearUserData)" outpu
youdeliver-decompiled/smali/
output_youdeliver-decompiled/smali/com/yopeso/lieferando/custom/LRActivity.smali:.method
vate clearUserData()V
output_youdeliver-decompiled/smali/com/yopeso/lieferando/custom/LRActivity.smali:    invo
direct {p0}, Lcom/yopeso/lieferando/custom/LRActivity;->clearUserData()V
ls@os:~/Downloads/stproject$
```

This could potentially be related to clearing user data, which might include account deletion functionality.

Reverse Engineering and Code Review:

clearUserData

```
ls@os:~/Downloads/stproject$ grep -A 20 "clearUserData" output_youdeliver-decompiled/smali/
com/yopeso/lieferando/custom/LRActivity.smali
.method private clearUserData()V
    .locals 4
```

```
ls@os:~/Downloads/stproject$ grep -rE "(deleteAccount|removeAccount|clearUserData)" outpu
youdeliver-decompiled/smali/
output_youdeliver-decompiled/smali/com/yopeso/lieferando/custom/LRActivity.smali:.method
vate clearUserData()V
output_youdeliver-decompiled/smali/com/yopeso/lieferando/custom/LRActivity.smali:    invo
direct {p0}, Lcom/yopeso/lieferando/custom/LRActivity;->clearUserData()V
ls@os:~/Downloads/stproject$
```

This could potentially be related to clearing user data, which might include account deletion functionality.

Reverse Engineering and Code Review:

Account Deletion or Data Removal Process

Key Method: removeUserCredential in the KahunaAnalytics class.

- **What It Does:** Removes user credentials (e.g., authentication tokens).

Possible Relevance to GDPR:

- Could be part of a larger process that removes personal data (e.g., shopping cart, order history, delivery addresses).
- If this broader data removal is implemented, it may fulfill GDPR’s "right to erasure."

Reverse Engineering and Code Review:

User Data Handling

Key Method: clearUserData in the LRActivity class.

- What It Does: Clears local user data, like preferences and authentication tokens.

GDPR Compliance Concerns:

- Only removes data from the user’s device (local deletion).
- Does not address server-side deletion, which is essential for full GDPR compliance.
- Without server-side action, it may not fully comply with GDPR’s "right to erasure."

Reverse Engineering and Code Review:

User Data Handling

Key Method: clearUserData in the LRActivity class.

- What It Does: Clears local user data, like preferences and authentication tokens.

GDPR Compliance Concerns:

- Only removes data from the user’s device (local deletion).
- Does not address server-side deletion, which is essential for full GDPR compliance.
- Without server-side action, it may not fully comply with GDPR’s "right to erasure."

LLM Response

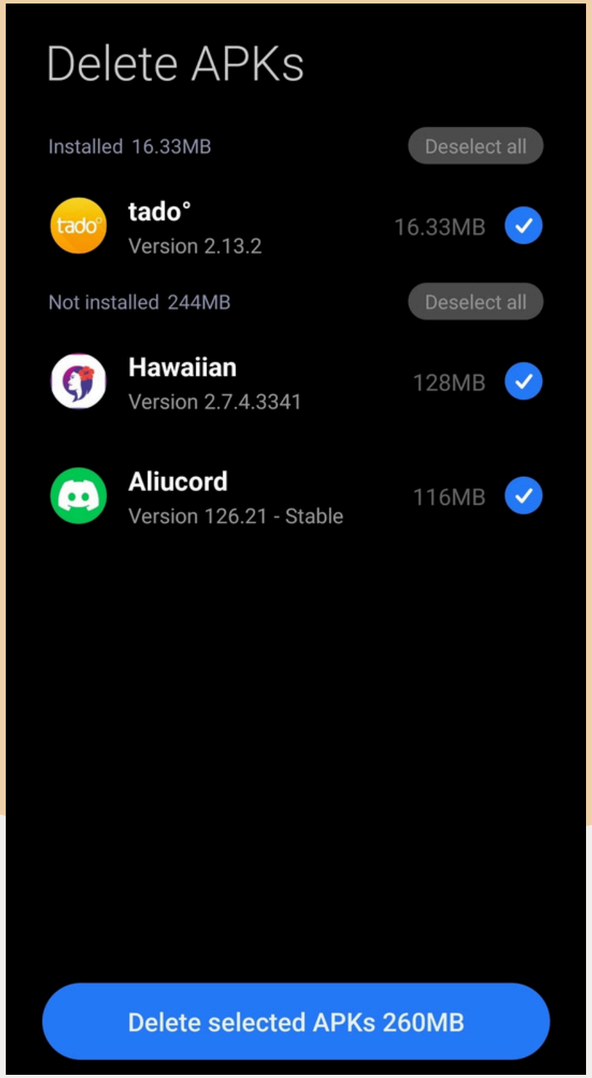
```
GDPR Compliance Assessment:

    Compliance Status: Not Fully Compliant
    Reasoning: The app has features for clearing data and deleting user accounts, but it lacks

For full GDPR compliance, the app should address these areas and ensure all data processing ac
```

Challenges

- Memory and Performance Issues
- Abstract and Interface Classes
- Missing Callback Methods
- Server Issues



Conclusion

R1

- The analysis consistently identified data leaks across varying timeout durations, confirming FlowDroid's efficiency in detecting taint flows within minimal time.
- The identified leaks appear genuine and require investigation to ensure compliance with privacy standards and improve data security.

R2

- The app has mechanisms to remove some user data locally.
- However, it lacks evidence of server-side deletion and transparency, which are critical for GDPR compliance.

- Improvements needed:
  - Ensure server-side deletion processes.
  - Clearly communicate data removal practices to users

Questions?
Comments?
Let us know!

Thank you for your attention!