Test security and privacy of your mobile application (iOS & Android), detect OWASP Mobile Top 10 and other weaknesses.

Summary of Mobile Application Security Test

APP NAME
COVID Safe Paths

APP ID
org.pathcheck.covidsafepaths

APP VERSION
1.0.1

DEVICE TYPE
Android

TEST STARTED
May 10th 2020, 15:38

TEST FINISHED
May 10th 2020, 15:51

OWASP Mobile Top 10

The automated audit revealed the following security flaws and weaknesses that may impact the application:

WARNINGS	LOW RISKS	MEDIUM RISKS	HIGH RISK
9	6	5	0

Zero false-positive SLA and advanced manual testing of application is only available in ImmuniWeb® MobileSuite.

HARDCODED SENSITIVE DATA [M10] [CWE-200] [SAST]

MEDIUM

Description:
The mobile application contains potentially sensitive hardcoded data. An attacker with an access to the mobile application file can easily extract this data from the application and use it in any further attacks.

Details:
There is **'DEFAULT_KEYSTORE_PASSWORD'** found in file 'ch/qos/logback/core/net/ssl/SSL.java':

```
line 4:     public static final String DEFAULT_KEYSTORE_PASSWORD = "changeit";
```

CVSSv3 Base Score:
6.5 (AV:L/AC:L/PR:L/UI:N/S:C/C:H/I:N/A:N)

EXTERNAL DATA IN RAW SQL QUERIES [M7] [CWE-89] [SAST]

MEDIUM

Description:
Inclusion of input into raw SQL queries can potentially lead to a local SQL injection vulnerability in the mobile application. The correct approach is to use prepared SQL statements beyond user's control.

Example of insecure code:

```
db.rawQuery("SELECT username FROM users_table WHERE id = '"+ input_id +"'");
db.execSQL("SELECT username FROM users_table WHERE id = '"+ input_id +"'");
```

Example of secure code:

```
PreparedStatement pstmt = con.prepareStatement("UPDATE EMPLOYEES SET SALARY = ? WHERE
ID = ?");
pstmt.setBigDecimal(1, 153833.00)
pstmt.setInt(2, 110592)
```

Details:

There is **'execSQL()'** found in file 'com/marianhello/bgloc/data/sqlite/SQLiteLocationDAO.java':

```
line 355:             stringBuilder2 = stringBuilder.toString();
line 356:             this.db.execSQL(stringBuilder2, new Object[]
{backgroundLocation.getProvider(), Long.valueOf(backgroundLocation.getTime()),
Float.valueOf(backgroundLocation.getAccuracy()),
Float.valueOf(backgroundLocation.getSpeed()),
Float.valueOf(backgroundLocation.getBearing()),
Double.valueOf(backgroundLocation.getAltitude()),
Float.valueOf(backgroundLocation.getRadius()),
Double.valueOf(backgroundLocation.getLatitude()),
Double.valueOf(backgroundLocation.getLongitude()),
Integer.valueOf(backgroundLocation.hasAccuracy()),
Integer.valueOf(backgroundLocation.hasSpeed()),
Integer.valueOf(backgroundLocation.hasBearing()),
Integer.valueOf(backgroundLocation.hasAltitude()),
Integer.valueOf(backgroundLocation.hasRadius()),
backgroundLocation.getLocationProvider(), backgroundLocation.getBatchStartMillis(),
Integer.valueOf(backgroundLocation.getStatus()),
Integer.valueOf(backgroundLocation.getMockFlags()), Long.valueOf(j)});
line 357:             this.db.setTransactionSuccessful();
```

```
line 359:             if (valueOf.booleanValue()) {
line 360:                 this.db.execSQL("VACUUM");
line 361:             }
```

CVSSv3 Base Score:

5.3 (AV:L/AC:L/PR:L/UI:N/S:U/C:L/I:L/A:L)

Reference:

- https://developer.android.com/reference/android/database/sqlite/SQLiteDatabase.html
- https://developer.android.com/reference/java/sql/PreparedStatement.html

USAGE OF UNENCRYPTED HTTP PROTOCOL [M3] [CWE-319] [SAST]

MEDIUM

Description:

The mobile application uses HTTP protocol to send or receive data. The design of HTTP protocol does not provide any encryption of the transmitted data, which can be easily intercepted if an attacker is located in the same network or has access to data channel of the victim.

Example of insecure code:

```
HttpURLConnection conn = (HttpURLConnection) url.openConnection();
```

Example of secure code:

```
HttpsURLConnection conn = (HttpsURLConnection) url.openConnection();
```

Details:
There is **'(HttpURLConnection)'** found in file 'com/marianhello/bgloc/HttpPostService.java':

```
line 30:         if (this.mHttpURLConnection == null) {
line 31:             this.mHttpURLConnection = (HttpURLConnection) new
URL(this.mUrl).openConnection();
line 32:         }
```

CVSSv3 Base Score:
6.5 (AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:L/A:N)

Reference:
- https://developer.android.com/reference/java/net/HttpURLConnection.html
- https://developer.android.com/reference/javax/net/ssl/HttpsURLConnection.html

PREDICTABLE RANDOM NUMBER GENERATOR [M5] [CWE-338] [SAST]

MEDIUM

Description:
The mobile application uses a predictable Random Number Generator (RNG).
Under certain conditions this weakness may jeopardize mobile application data encryption or other protection based on randomization. For example, if encryption tokens are generated inside of the application and an attacker can provide application with a predictable token to validate and then execute a sensitive activity within the application or its backend.

Example of insecure code:

```
Random random = new Random();
byte bytes[] = new byte[20];
random.nextBytes(bytes);
```

Example of secure code:

```
SecureRandom random = new SecureRandom();
byte bytes[] = new byte[20];
random.nextBytes(bytes);
```

Details:
There is **'new Random()'** found in file 'net/lingala/zip4j/crypto/AESEncrpyter.java':

```
line 140:             for (int i3 = 0; i3 < i2; i3++) {
line 141:                 int nextInt = new Random().nextInt();
line 142:                 int i4 = i3 * 4;
```

There is **'new Random()'** found in file 'net/lingala/zip4j/crypto/StandardEncrypter.java':

```
line 70:             byte[] bArr = new byte[i];
line 71:             Random random = new Random();
line 72:             for (int i2 = 0; i2 < bArr.length; i2++) {
```

There is **'new Random()'** found in file 'kotlin/random/FallbackThreadLocalRandom$implStorage$1.java':

```
line 15:     public Random initialValue() {
line 16:         return new Random();
line 17:     }
```

There is **'new Random()'** found in file 'com/dieam/reactnativepushnotification/modules/RNPushNotification.java':

```
line 32:     private RNPushNotificationHelper mRNPushNotificationHelper;
line 33:     private final Random mRandomNumberGenerator = new
Random(System.currentTimeMillis());
line 34:
```

There is **'new Random()'** found in file 'com/dieam/reactnativepushnotification/modules/RNPushNotificationListenerService.java':

```
line 100:         if (bundle.getString(str) == null) {
line 101:             bundle.putString(str, String.valueOf(new
Random(System.currentTimeMillis()).nextInt()));
line 102:         }
```

There is **'new Random()'** found in file 'com/dieam/reactnativepushnotification/modules/RNPushNotificationListenerServiceGcm.java':

```
line 87:         if (bundle.getString(str) == null) {
line 88:             bundle.putString(str, String.valueOf(new
Random(System.currentTimeMillis()).nextInt()));
line 89:         }
```

CVSSv3 Base Score:
4.8 (AV:N/AC:H/PR:N/UI:N/S:U/C:L/I:L/A:N)

Reference:

- https://developer.android.com/reference/java/util/Random.html
- https://developer.android.com/reference/java/security/SecureRandom.html

CLEARTEXT SQLITE DATABASE [M2] [CWE-312] [DAST]

MEDIUM

Description:
The mobile application uses an unencrypted SQLite database.
This database can be accessed by an attacker with physical access to the mobile device or a malicious application with root access to the device. The application should not store sensitive information in clear text.

Details:
In file cordova_bg_geolocation.db:

```
TABLES:
android_metadata
location
configuration
time_idx
batch_id_idx

RAW DUMP:
CREATE TABLE android_metadata (locale TEXT);CREATE TABLE location (_id INTEGER PRIMARY
KEY,time INTEGER,accuracy REAL,speed REAL,bearing REAL,altitude REAL,latitude
REAL,longitude REAL,radius REAL,has_accuracy INTEGER,has_speed INTEGER,has_bearing
INTEGER,has_altitude INTEGER,has_radius INTEGER,provider TEXT,service_provider
INTEGER,valid INTEGER,batch_start INTEGER,mock_flags INTEGER );CREATE TABLE
configuration (_id INTEGER PRIMARY KEY,stationary_radius REAL,distance_filter
INTEGER,desired_accuracy INTEGER,debugging INTEGER,notification_title
TEXT,notification_text TEXT,notification_icon_small TEXT,notification_icon_large
TEXT,notification_icon_color TEXT,stop_terminate INTEGER,stop_still INTEGER,start_boot
INTEGER,start_foreground INTEGER,notifications_enabled INTEGER,service_provider
TEXT,interval INTEGER,fastest_interval INTEGER,activities_interval INTEGER,url
TEXT,sync_url TEXT,sync_threshold INTEGER,http_headers TEXT,max_locations
INTEGER,template TEXT );CREATE INDEX time_idx ON location (time);CREATE INDEX
batch_id_idx ON location (batch_start);
```

CVSSv3 Base Score:
5.5 (AV:L/AC:L/PR:L/UI:N/S:U/C:H/I:N/A:N)

HARDCODED DATA [M2] [CWE-200] [SAST]

LOW

Description:
The mobile application contains debugging or other technical information that may be extracted and used by an attacker to facilitate further attacks.

http:// with value http://logback.qos.ch/codes.html#layoutInsteadOfEncoder in following files:

- **ch/qos/logback/core/OutputStreamAppender.java:**

```
line 78: addWarn("See also
http://logback.qos.ch/codes.html#layoutInsteadOfEncoder for details");
```

http:// with value http://logback.qos.ch/codes.html in following files:

- **ch/qos/logback/core/CoreConstants.java:**

```
line 7: public static final String CODES_URL =
"http://logback.qos.ch/codes.html";
line 52: public static final String SEE_FNP_NOT_SET = "See also
http://logback.qos.ch/codes.html#tbr_fnp_not_set";
```

http:// with value http://logback.qos.ch/codes.html#tbr_fnp_not_set in following files:

- **ch/qos/logback/core/CoreConstants.java:**

```
line 52: public static final String SEE_FNP_NOT_SET = "See also
http://logback.qos.ch/codes.html#tbr_fnp_not_set";
```

- **ch/qos/logback/core/rolling/TimeBasedRollingPolicy.java:**

```
line 138: throw new IllegalStateException("The FileNamePattern option must
be set before using TimeBasedRollingPolicy. See also
http://logback.qos.ch/codes.html#tbr_fnp_not_set");
```

- **ch/qos/logback/core/rolling/FixedWindowRollingPolicy.java:**

```
line 174: throw new IllegalStateException("The \"FileNamePattern\" property
must be set before using FixedWindowRollingPolicy. See also
http://logback.qos.ch/codes.html#tbr_fnp_not_set");
```

http:// with value http://logback.qos.ch/codes.html#1andOnly1 in following files:

- **ch/qos/logback/core/sift/SiftingJoranConfiguratorBase.java:**

```
line 19: static final String ONE_AND_ONLY_ONE_URL =
"http://logback.qos.ch/codes.html#1andOnly1";
line 56: str = "Only and only one appender can be nested the <sift> element
in SiftingAppender. See also http://logback.qos.ch/codes.html#1andOnly1";
```

http:// with value http://logback.qos.ch/codes.html#syslog_layout in following files:

- **ch/qos/logback/core/net/SyslogAppenderBase.java:**

```
line 10: static final String SYSLOG_LAYOUT_URL =
"http://logback.qos.ch/codes.html#syslog_layout";
line 186: addWarn("The layout of a SyslogAppender cannot be set directly.
See also http://logback.qos.ch/codes.html#syslog_layout");
```

http:// with value http://logback.qos.ch/codes.html#socket_no_port in following files:

- **ch/qos/logback/core/net/AbstractSocketAppender.java:**

```
line 226: stringBuilder.append(" For more information, please visit
http://logback.qos.ch/codes.html#socket_no_port");
```

http:// with value http://logback.qos.ch/codes.html#socket_no_host in following files:

- **ch/qos/logback/core/net/AbstractSocketAppender.java:**

```
line 235: stringBuilder2.append(" For more information, please visit
http://logback.qos.ch/codes.html#socket_no_host");
```

http:// with value http://logback.qos.ch/codes.html#smtp_no_layout in following files:

- **ch/qos/logback/core/net/SMTPAppenderBase.java:**

```
line 196: name = "]. For more information, please visit
http://logback.qos.ch/codes.html#smtp_no_layout";
```

http:// with value http://logback.qos.ch/codes.html#tbr_fnp_prudent_unsupported in following files:

- **ch/qos/logback/core/rolling/FixedWindowRollingPolicy.java:**

```
line 13: static final String PRUDENT_MODE_UNSUPPORTED = "See also
http://logback.qos.ch/codes.html#tbr_fnp_prudent_unsupported";
```

http:// with value http://logback.qos.ch/codes.html#fwrp_parentFileName_not_set in following files:

- **ch/qos/logback/core/rolling/FixedWindowRollingPolicy.java:**

```
line 14: static final String SEE_PARENT_FN_NOT_SET = "Please refer to
http://logback.qos.ch/codes.html#fwrp_parentFileName_not_set";
```

http:// with value http://logback.qos.ch/codes.html#sbtp_size_format in following files:

- **ch/qos/logback/core/rolling/SizeBasedTriggeringPolicy.java:**

```
line 7: public static final String SEE_SIZE_FORMAT =
"http://logback.qos.ch/codes.html#sbtp_size_format";
line 86: r2 = "http://logback.qos.ch/codes.html#sbtp_size_format";
```

http:// with value http://logback.qos.ch/codes.html#rfa_collision in following files:

- **ch/qos/logback/core/rolling/RollingFileAppender.java:**

```
line 7: private static String COLLISION_URL =
"http://logback.qos.ch/codes.html#rfa_collision";
```

http:// with value http://logback.qos.ch/codes.html#rfa_no_rp in following files:

- **ch/qos/logback/core/rolling/RollingFileAppender.java:**

```
line 8: private static String RFA_NO_RP_URL =
"http://logback.qos.ch/codes.html#rfa_no_rp";
```

http:// with value http://logback.qos.ch/codes.html#rfa_no_tp in following files:

- **ch/qos/logback/core/rolling/RollingFileAppender.java:**

```
line 9: private static String RFA_NO_TP_URL =
"http://logback.qos.ch/codes.html#rfa_no_tp";
```

http:// with value http://logback.qos.ch/codes.html#rfa_file_after in following files:

- **ch/qos/logback/core/rolling/RollingFileAppender.java:**

```
line 80: addError("Visit http://logback.qos.ch/codes.html#rfa_file_after for
more information");
```

http:// with value http://logback.qos.ch/codes.html#renamingError in following files:

- **ch/qos/logback/core/rolling/helper/RenameUtil.java:**

```
line 9: static String RENAMING_ERROR_URL =
"http://logback.qos.ch/codes.html#renamingError";
```

http:// with value http://logback.qos.ch/codes.html#appender_order in following files:

- **ch/qos/logback/core/joran/action/AppenderRefAction.java:**

```
line 32: addError("See http://logback.qos.ch/codes.html#appender_order for
more details.");
```

http:// with value http://logback.qos.ch/codes.html#missingRightParenthesis in following files:

- **ch/qos/logback/core/pattern/parser/Parser.java:**

```
line 16: public static final String MISSING_RIGHT_PARENTHESIS =
"http://logback.qos.ch/codes.html#missingRightParenthesis";
line 67: addError("See also
http://logback.qos.ch/codes.html#missingRightParenthesis");
```

http:// with value http://logback.qos.ch/codes.html#receiver_no_port in following files:

- **ch/qos/logback/classic/net/SocketReceiver.java:**

```
line 227: addError("No port was configured for receiver. For more
information, please visit
http://logback.qos.ch/codes.html#receiver_no_port");
```

http:// with value http://logback.qos.ch/codes.html#receiver_no_host in following files:

- **ch/qos/logback/classic/net/SocketReceiver.java:**

```
line 234: addError("No host name or address was configured for receiver. For
more information, please visit
http://logback.qos.ch/codes.html#receiver_no_host");
```

http:// with value http://logback.qos.ch/css/classic.css in following files:

- **ch/qos/logback/classic/html/UrlCssBuilder.java:**

```
line 4: String url = "http://logback.qos.ch/css/classic.css";
```

http:// with value http://mindprod.com in following files:

- **com/mindprod/ledatastream/LEDataInputStream.java:**

```
line 7: private static final String EMBEDDED_COPYRIGHT = "copyright (c)
1999-2010 Roedy Green, Canadian Mind Products, http://mindprod.com";
```

https:// with value https://www.facebook.com/sharer/sharer.php in following files:

- **cl/json/social/FacebookPagesManagerShare.java:**

```
line 6: private static final String DEFAULT_WEB_LINK =
"https://www.facebook.com/sharer/sharer.php?u={url}";
```

https:// with value https://plus.google.com/share in following files:

- **cl/json/social/GooglePlusShare.java:**

```
line 6: private static final String DEFAULT_WEB_LINK =
"https://plus.google.com/share?url={url}";
```

https:// with value https://twitter.com/intent/tweet in following files:

- **cl/json/social/TwitterShare.java:**

```
line 6: private static final String DEFAULT_WEB_LINK =
"https://twitter.com/intent/tweet?text={message}&url={url}";
```

https:// with value https://pinterest.com/pin/create/button/ in following files:

- **cl/json/social/PinterestShare.java:**

```
line 6: private static final String DEFAULT_WEB_LINK =
"https://pinterest.com/pin/create/button/?url=
{url}&media=$media&description={message}";
```

CVSSv3 Base Score:
3.3 (AV:L/AC:L/PR:L/UI:N/S:U/C:L/I:N/A:N)

MISSING TAPJACKING PROTECTION [M1] [CWE-451] [SAST]

LOW

Description:
The mobile application does not have a tapjacking protection required to mitigate tapjacking attacks.
By default, Android OS permits a mobile application to display its user interface over the user interface of another application installed and run on the device. When user touches the screen, application may pass the touch event to another application below its user interface layer that the user does not see, serving like a proxy to pass unintended touch activities. This attack is quite similar to clickjacking but for mobile devices. In order to be successfully exploited, a malicious application shall be already installed on the mobile phone of the victim. An example of exploitation would be a malware app that tricks user to unwittingly tap on a payment button (or any other functionality) of a sensitive application when playing a game or doing other innocent activity in the malicious application screen.

Example of secure code:

```
public class MyActivity extends Activity {
    protected void onCreate(Bundle bundle) {
        super.onCreate(bundle);

        final Button myButton = (Button)findViewById(R.id.button_id);
        myButton.setFilterTouchesWhenObscured(true);

        myButton.setOnClickListener(new View.OnClickListener() {
            // Perform action on click
        }
    }
}

<Button
    android:layout_height="wrap_content"
    android:layout_width="wrap_content"
    android:text="@string/self_destruct"
    android:onClick="selfDestruct"
    android:filterTouchesWhenObscured="true" />
```

Details:
There is **android:filterTouchesWhenObscured="true"** missing in files:

- android/res/layout-v22/abc_alert_dialog_button_bar_material.xml
- android/res/layout-watch/abc_alert_dialog_button_bar_material.xml
- android/res/layout-watch/abc_alert_dialog_title_material.xml
- android/res/layout/abc_list_menu_item_checkbox.xml
- android/res/layout/design_layout_snackbar_include.xml
- android/res/layout/abc_alert_dialog_button_bar_material.xml
- android/res/layout/redbox_item_frame.xml
- android/res/layout/abc_activity_chooser_view_list_item.xml
- android/res/layout/notification_template_big_media.xml
- android/res/layout/design_layout_tab_icon.xml
- android/res/layout/abc_screen_content_include.xml
- android/res/layout/abc_activity_chooser_view.xml

- android/res/layout/abc_select_dialog_material.xml
- android/res/layout/abc_action_bar_title_item.xml
- android/res/layout/abc_screen_toolbar.xml
- android/res/layout/abc_dialog_title_material.xml
- android/res/layout/abc_search_view.xml
- android/res/layout/abc_search_dropdown_item_icons_2line.xml
- android/res/layout/abc_action_menu_item_layout.xml
- android/res/layout/design_bottom_sheet_dialog.xml
- android/res/layout/fps_view.xml
- android/res/layout/design_bottom_navigation_item.xml
- android/res/layout/abc_action_bar_up_container.xml
- android/res/layout/notification_template_icon_group.xml
- android/res/layout/abc_list_menu_item_layout.xml
- android/res/layout/design_navigation_item_separator.xml
- android/res/layout/notification_media_cancel_action.xml
- android/res/layout/notification_action_tombstone.xml
- android/res/layout/redbox_view.xml
- android/res/layout/abc_action_menu_layout.xml
- android/res/layout/notification_template_part_time.xml
- android/res/layout/notification_action.xml
- android/res/layout/abc_list_menu_item_radio.xml
- android/res/layout/mtrl_layout_snackbar_include.xml
- android/res/layout/custom_dialog.xml
- android/res/layout/abc_screen_simple_overlay_action_mode.xml
- android/res/layout/abc_popup_menu_item_layout.xml
- android/res/layout/design_menu_item_action_area.xml
- android/res/layout/select_dialog_singlechoice_material.xml
- android/res/layout/select_dialog_item_material.xml
- android/res/layout/design_navigation_menu.xml
- android/res/layout/support_simple_spinner_dropdown_item.xml
- android/res/layout/notification_media_action.xml
- android/res/layout/launch_screen.xml
- android/res/layout/notification_template_media.xml
- android/res/layout/abc_alert_dialog_material.xml
- android/res/layout/design_navigation_item.xml
- android/res/layout/abc_tooltip.xml
- android/res/layout/abc_action_mode_bar.xml
- android/res/layout/design_navigation_item_header.xml
- android/res/layout/notification_template_lines_media.xml
- android/res/layout/design_layout_snackbar.xml
- android/res/layout/notification_template_big_media_narrow_custom.xml
- android/res/layout/abc_action_mode_close_item_material.xml
- android/res/layout/design_navigation_item_subheader.xml
- android/res/layout/abc_list_menu_item_icon.xml
- android/res/layout/abc_expanded_menu_layout.xml
- android/res/layout/notification_template_media_custom.xml
- android/res/layout/dev_loading_view.xml
- android/res/layout/abc_cascading_menu_item_layout.xml
- android/res/layout/design_text_input_password_icon.xml
- android/res/layout/mtrl_layout_snackbar.xml
- android/res/layout/design_navigation_menu_item.xml
- android/res/layout/notification_template_custom_big.xml
- android/res/layout/redbox_item_title.xml
- android/res/layout/abc_popup_menu_header_item_layout.xml
- android/res/layout/abc_alert_dialog_title_material.xml
- android/res/layout/notification_template_big_media_custom.xml
- android/res/layout/notification_template_big_media_narrow.xml
- android/res/layout/notification_template_part_chronometer.xml

- android/res/layout/abc_screen_simple.xml
- android/res/layout/select_dialog_multichoice_material.xml
- android/res/layout/design_layout_tab_text.xml
- android/res/layout-sw600dp/design_layout_snackbar.xml
- android/res/layout-sw600dp/mtrl_layout_snackbar.xml
- android/res/layout-v26/abc_screen_toolbar.xml

There is **'extends ViewGroup'** found in file 'androidx/transition/ViewOverlayApi14.java':

```
line 21:
line 22:     static class OverlayViewGroup extends ViewGroup {
line 23:         static Method sInvalidateChildInParentFastMethod;
```

There is **'extends ViewGroup'** found in file 'androidx/transition/GhostViewPort.java':

```
line 13: @SuppressLint({"ViewConstructor"})
line 14: class GhostViewPort extends ViewGroup implements GhostView {
line 15:     @Nullable
```

There is **'extends ViewGroup'** found in file 'androidx/drawerlayout/widget/DrawerLayout.java':

```
line 48:
line 49: public class DrawerLayout extends ViewGroup {
line 50:     private static final boolean ALLOW_EDGE_LOCK = false;
```

There is **'extends ViewGroup'** found in file 'androidx/viewpager/widget/ViewPager.java':

```
line 51:
line 52: public class ViewPager extends ViewGroup {
line 53:     private static final int CLOSE_ENOUGH = 2;
```

There is **'extends ViewGroup'** found in file 'androidx/viewpager/widget/PagerTitleStrip.java':

```
line 27: @DecorView
line 28: public class PagerTitleStrip extends ViewGroup {
line 29:     private static final int[] ATTRS = new int[]{16842804, 16842901, 16842904,
16842927};
```

There is **'extends ViewGroup'** found in file 'androidx/swiperefreshlayout/widget/SwipeRefreshLayout.java':

```
line 33:
line 34: public class SwipeRefreshLayout extends ViewGroup implements
NestedScrollingParent, NestedScrollingChild {
line 35:     private static final int ALPHA_ANIMATION_DURATION = 300;
```

There is **'extends ViewGroup'** found in file 'androidx/recyclerview/widget/RecyclerView.java':

```
line 69:
line 70: public class RecyclerView extends ViewGroup implements ScrollingView,
NestedScrollingChild2 {
line 71:     static final boolean ALLOW_SIZE_IN_UNSPECIFIED_SPEC = (VERSION.SDK_INT >=
23);
```

There is **'extends ViewGroup'** found in file 'androidx/appcompat/widget/ActionBarOverlayLayout.java':

```
line 32: @RestrictTo({Scope.LIBRARY_GROUP_PREFIX})
line 33: public class ActionBarOverlayLayout extends ViewGroup implements
DecorContentParent, NestedScrollingParent, NestedScrollingParent2,
NestedScrollingParent3 {
line 34:     private static final int ACTION_BAR_ANIMATE_DELAY = 600;
```

There is **'extends ViewGroup'** found in file 'androidx/appcompat/widget/Toolbar.java':

```
line 54:
line 55: public class Toolbar extends ViewGroup {
line 56:     private static final String TAG = "Toolbar";
```

There is **'extends ViewGroup'** found in file 'androidx/appcompat/widget/ActivityChooserView.java':

```
line 40: @RestrictTo({Scope.LIBRARY_GROUP_PREFIX})
line 41: public class ActivityChooserView extends ViewGroup implements
ActivityChooserModelClient {
line 42:     private static final String LOG_TAG = "ActivityChooserView";
```

There is **'extends ViewGroup'** found in file 'androidx/appcompat/widget/LinearLayoutCompat.java':

```
line 22:
line 23: public class LinearLayoutCompat extends ViewGroup {
line 24:     private static final String ACCESSIBILITY_CLASS_NAME =
"androidx.appcompat.widget.LinearLayoutCompat";
```

There is **'extends ViewGroup'** found in file 'androidx/appcompat/widget/AbsActionBarView.java':

```
line 17:
line 18: abstract class AbsActionBarView extends ViewGroup {
line 19:     private static final int FADE_DURATION = 200;
```

There is **'extends ViewGroup'** found in file 'androidx/slidingpanelayout/widget/SlidingPaneLayout.java':

```
line 40:
line 41: public class SlidingPaneLayout extends ViewGroup {
line 42:     private static final int DEFAULT_FADE_COLOR = -858993460;
```

There is **'extends ViewGroup'** found in file 'androidx/coordinatorlayout/widget/CoordinatorLayout.java':

```
line 61:
line 62: public class CoordinatorLayout extends ViewGroup implements
NestedScrollingParent2 {
line 63:     static final Class<?>[] CONSTRUCTOR_PARAMS = new Class[]{Context.class,
AttributeSet.class};
```

There is **'extends ViewGroup'** found in file 'com/swmansion/gesturehandler/react/RNGestureHandlerButtonViewManager.java':

```
line 17:
line 18: public class RNGestureHandlerButtonViewManager extends
ViewGroupManager<ButtonViewGroup> {
line 19:
line 20:     static class ButtonViewGroup extends ViewGroup {
line 21:         static TypedValue sResolveOutValue = new TypedValue();
```

There is **'extends ViewGroup'** found in file 'com/swmansion/rnscreens/ScreenStackHeaderConfig.java':

```
line 13:
line 14: public class ScreenStackHeaderConfig extends ViewGroup {
line 15:     private OnClickListener mBackClickListener = new OnClickListener() {
```

There is **'extends ViewGroup'** found in file 'com/swmansion/rnscreens/Screen.java':

```
line 16:
line 17: public class Screen extends ViewGroup implements ReactPointerEventsView {
line 18:     private static OnAttachStateChangeListener sShowSoftKeyboardOnAttach = new
OnAttachStateChangeListener() {
```

There is **'extends ViewGroup'** found in file 'com/swmansion/rnscreens/ScreenContainer.java':

```
line 16:
line 17: public class ScreenContainer<T extends ScreenFragment> extends ViewGroup {
line 18:     private final Set<ScreenFragment> mActiveScreenFragments = new HashSet();
```

There is **'extends FrameLayout'** found in file 'androidx/transition/GhostViewHolder.java':

```
line 12: @SuppressLint({"ViewConstructor"})
line 13: class GhostViewHolder extends FrameLayout {
line 14:     private boolean mAttached = true;
```

There is **'extends FrameLayout'** found in file 'androidx/core/widget/NestedScrollView.java':

```
line 47:
line 48: public class NestedScrollView extends FrameLayout implements
NestedScrollingParent3, NestedScrollingChild3, ScrollingView {
line 49:     private static final AccessibilityDelegate ACCESSIBILITY_DELEGATE = new
AccessibilityDelegate();
```

There is **'extends FrameLayout'** found in file 'androidx/appcompat/widget/ContentFrameLayout.java':

```
line 12: @RestrictTo({Scope.LIBRARY})
line 13: public class ContentFrameLayout extends FrameLayout {
line 14:     private OnAttachListener mAttachListener;
```

There is **'extends FrameLayout'** found in file 'androidx/appcompat/widget/ActionBarContainer.java':

```
line 20: @RestrictTo({Scope.LIBRARY_GROUP_PREFIX})
line 21: public class ActionBarContainer extends FrameLayout {
line 22:     private View mActionBarView;
```

There is **'extends FrameLayout'** found in file 'androidx/appcompat/widget/FitWindowsFrameLayout.java':

```
line 11: @RestrictTo({Scope.LIBRARY_GROUP_PREFIX})
line 12: public class FitWindowsFrameLayout extends FrameLayout implements
FitWindowsViewGroup {
line 13:     private OnFitSystemWindowsListener mListener;
```

There is **'extends FrameLayout'** found in file 'androidx/appcompat/view/menu/MenuItemWrapperICS.java':

```
line 84:
line 85:     static class CollapsibleActionViewWrapper extends FrameLayout implements
CollapsibleActionView {
line 86:         final android.view.CollapsibleActionView mWrappedView;
```

There is **'extends FrameLayout'** found in file 'androidx/cardview/widget/CardView.java':

```
line 19:
line 20: public class CardView extends FrameLayout {
line 21:     private static final int[] COLOR_BACKGROUND_ATTR = new int[]{16842801};
```

There is **'extends ProgressBar'** found in file 'androidx/core/widget/ContentLoadingProgressBar.java':

```
line 8:
line 9: public class ContentLoadingProgressBar extends ProgressBar {
line 10:     private static final int MIN_DELAY = 500;
```

There is **'extends PagerTitleStrip'** found in file 'androidx/viewpager/widget/PagerTabStrip.java':

```
line 20:
line 21: public class PagerTabStrip extends PagerTitleStrip {
line 22:     private static final int FULL_UNDERLINE_HEIGHT = 1;
```

There is **'extends View'** found in file 'androidx/legacy/widget/Space.java':

```
line 12: @Deprecated
line 13: public class Space extends View {
line 14:     @SuppressLint({"MissingSuperCall"})
```

There is **'extends View'** found in file 'androidx/appcompat/widget/ViewStubCompat.java':

```
line 18: @RestrictTo({Scope.LIBRARY_GROUP_PREFIX})
line 19: public final class ViewStubCompat extends View {
line 20:     private OnInflateListener mInflateListener;
```

There is **'extends View'** found in file 'com/BV/LinearGradient/LinearGradientView.java':

```
line 14:
line 15: public class LinearGradientView extends View {
line 16:     private float mAngle = 45.0f;
```

There is **'extends View'** found in file 'com/swmansion/rnscreens/LifecycleHelper.java':

```
line 39:
line 40:     public <T extends View & LifecycleObserver> void register(T t) {
line 41:         t.addOnLayoutChangeListener(this.mRegisterOnLayoutChange);
```

```
line 43:
line 44:     public <T extends View & LifecycleObserver> void unregister(T t) {
line 45:         Lifecycle lifecycle = (Lifecycle) this.mViewToLifecycleMap.get(t);
```

There is **'extends ImageView'** found in file 'androidx/swiperefreshlayout/widget/CircleImageView.java':

```
line 15:
line 16: class CircleImageView extends ImageView {
line 17:     private static final int FILL_SHADOW_COLOR = 1023410176;
```

There is **'extends ImageView'** found in file 'androidx/appcompat/widget/AppCompatImageView.java':

```
line 17:
line 18: public class AppCompatImageView extends ImageView implements
TintableBackgroundView, TintableImageSourceView {
line 19:     private final AppCompatBackgroundHelper mBackgroundTintHelper;
```

There is **'extends TabHost'** found in file 'androidx/fragment/app/FragmentTabHost.java':

```
line 25: @Deprecated
line 26: public class FragmentTabHost extends TabHost implements OnTabChangeListener {
line 27:     private boolean mAttached;
```

There is **'extends ListView'** found in file 'androidx/appcompat/app/AlertController.java':

```
line 341:
line 342:     public static class RecycleListView extends ListView {
line 343:         private final int mPaddingBottomNoButtons;
```

There is **'extends ListView'** found in file 'androidx/appcompat/widget/DropDownListView.java':

```
line 22:
line 23: class DropDownListView extends ListView {
line 24:     public static final int INVALID_POSITION = -1;
```

There is **'extends ListView'** found in file 'androidx/appcompat/view/menu/ExpandedMenuView.java':

```
line 14: @RestrictTo({Scope.LIBRARY_GROUP_PREFIX})
line 15: public final class ExpandedMenuView extends ListView implements ItemInvoker,
MenuView, OnItemClickListener {
line 16:     private static final int[] TINT_ATTRS = new int[]{16842964, 16843049};
```

There is **'extends RadioButton'** found in file 'androidx/appcompat/widget/AppCompatRadioButton.java':

```
line 17:
line 18: public class AppCompatRadioButton extends RadioButton implements
TintableCompoundButton, TintableBackgroundView {
line 19:     private final AppCompatBackgroundHelper mBackgroundTintHelper;
```

There is **'extends RatingBar'** found in file 'androidx/appcompat/widget/AppCompatRatingBar.java':

```
line 9:
line 10: public class AppCompatRatingBar extends RatingBar {
line 11:     private final AppCompatProgressBarHelper mAppCompatProgressBarHelper;
```

There is **'extends ImageButton'** found in file 'androidx/appcompat/widget/AppCompatImageButton.java':

```
line 18:
line 19: public class AppCompatImageButton extends ImageButton implements
TintableBackgroundView, TintableImageSourceView {
line 20:     private final AppCompatBackgroundHelper mBackgroundTintHelper;
```

There is **'extends LinearLayout'** found in file 'androidx/appcompat/widget/ButtonBarLayout.java':

```
line 16: @RestrictTo({Scope.LIBRARY_GROUP_PREFIX})
line 17: public class ButtonBarLayout extends LinearLayout {
line 18:     private static final int PEEK_BUTTON_DP = 16;
```

There is **'extends LinearLayout'** found in file 'androidx/appcompat/widget/FitWindowsLinearLayout.java':

```
line 11: @RestrictTo({Scope.LIBRARY_GROUP_PREFIX})
line 12: public class FitWindowsLinearLayout extends LinearLayout implements
FitWindowsViewGroup {
line 13:     private OnFitSystemWindowsListener mListener;
```

There is **'extends LinearLayout'** found in file 'androidx/appcompat/view/menu/ListMenuItemView.java':

```
line 25: @RestrictTo({Scope.LIBRARY_GROUP_PREFIX})
line 26: public class ListMenuItemView extends LinearLayout implements ItemView,
SelectionBoundsAdjuster {
line 27:     private static final String TAG = "ListMenuItemView";
```

There is **'extends EditText'** found in file 'androidx/appcompat/widget/AppCompatEditText.java':

```
line 24:
line 25: public class AppCompatEditText extends EditText implements
TintableBackgroundView {
line 26:     private final AppCompatBackgroundHelper mBackgroundTintHelper;
```

There is **'extends LinearLayoutCompat'** found in file 'androidx/appcompat/widget/ActionMenuView.java':

```
line 24:
line 25: public class ActionMenuView extends LinearLayoutCompat implements ItemInvoker,
MenuView {
line 26:     static final int GENERATED_ITEM_PADDING = 4;
```

There is **'extends LinearLayoutCompat'** found in file 'androidx/appcompat/widget/AlertDialogLayout.java':

```
line 15: @RestrictTo({Scope.LIBRARY_GROUP_PREFIX})
line 16: public class AlertDialogLayout extends LinearLayoutCompat {
line 17:     public AlertDialogLayout(@Nullable Context context) {
```

There is **'extends Button'** found in file 'androidx/appcompat/widget/AppCompatButton.java':

```
line 22:
line 23: public class AppCompatButton extends Button implements TintableBackgroundView,
AutoSizeableTextView {
line 24:     private final AppCompatBackgroundHelper mBackgroundTintHelper;
```

There is **'extends SeekBar'** found in file 'androidx/appcompat/widget/AppCompatSeekBar.java':

```
line 8:
line 9: public class AppCompatSeekBar extends SeekBar {
line 10:     private final AppCompatSeekBarHelper mAppCompatSeekBarHelper;
```

There is **'extends CheckBox'** found in file 'androidx/appcompat/widget/AppCompatCheckBox.java':

```
line 17:
line 18: public class AppCompatCheckBox extends CheckBox implements
TintableCompoundButton, TintableBackgroundView {
line 19:     private final AppCompatBackgroundHelper mBackgroundTintHelper;
```

There is **'extends TextView'** found in file 'androidx/appcompat/widget/AppCompatTextView.java':

```
line 34:
line 35: public class AppCompatTextView extends TextView implements
TintableBackgroundView, TintableCompoundDrawablesView, AutoSizeableTextView {
line 36:     private final AppCompatBackgroundHelper mBackgroundTintHelper;
```

There is **'extends TextView'** found in file 'com/horcrux/svg/TSpanView.java':

```
line 22: @SuppressLint({"ViewConstructor"})
line 23: class TSpanView extends TextView {
line 24:     private static final String FONTS = "fonts/";
```

There is **'extends TextView'** found in file 'com/horcrux/svg/TextPathView.java':

```
line 12: @SuppressLint({"ViewConstructor"})
line 13: class TextPathView extends TextView {
line 14:     private String mHref;
```

There is **'extends ToggleButton'** found in file 'androidx/appcompat/widget/AppCompatToggleButton.java':

```
line 6:
line 7: public class AppCompatToggleButton extends ToggleButton {
line 8:     private final AppCompatTextHelper mTextHelper;
```

There is **'extends CheckedTextView'** found in file 'androidx/appcompat/widget/AppCompatCheckedTextView.java':

```
line 12:
line 13: public class AppCompatCheckedTextView extends CheckedTextView {
line 14:     private static final int[] TINT_ATTRS = new int[]{16843016};
```

There is **'extends CompoundButton'** found in file 'androidx/appcompat/widget/SwitchCompat.java':

```
line 36:
line 37: public class SwitchCompat extends CompoundButton {
line 38:     private static final String ACCESSIBILITY_EVENT_CLASS_NAME =
"android.widget.Switch";
```

There is **'extends AppCompatAutoCompleteTextView'** found in file 'androidx/appcompat/widget/SearchView.java':

```
line 376:     @RestrictTo({Scope.LIBRARY_GROUP_PREFIX})
line 377:     public static class SearchAutoComplete extends
AppCompatAutoCompleteTextView {
line 378:         private boolean mHasPendingShowSoftInputRequest;
```

There is **'extends AppCompatImageView'** found in file 'androidx/appcompat/widget/ActionMenuPresenter.java':

```
line 164:
line 165:     private class OverflowMenuButton extends AppCompatImageView implements
ActionMenuChildView {
line 166:         private final float[] mTempPts = new float[2];
```

There is **'extends AppCompatTextView'** found in file 'androidx/appcompat/widget/DialogTitle.java':

```
line 11: @RestrictTo({Scope.LIBRARY_GROUP_PREFIX})
line 12: public class DialogTitle extends AppCompatTextView {
line 13:     public DialogTitle(Context context, AttributeSet attributeSet, int i) {
```

There is **'extends AppCompatTextView'** found in file 'androidx/appcompat/view/menu/ActionMenuItemView.java':

```
line 25: @RestrictTo({Scope.LIBRARY_GROUP_PREFIX})
line 26: public class ActionMenuItemView extends AppCompatTextView implements ItemView,
OnClickListener, ActionMenuChildView {
line 27:     private static final int MAX_ICON_SIZE = 32;
```

There is **'extends AutoCompleteTextView'** found in file 'androidx/appcompat/widget/AppCompatAutoCompleteTextView.java':

```
line 20:
line 21: public class AppCompatAutoCompleteTextView extends AutoCompleteTextView
implements TintableBackgroundView {
line 22:     private static final int[] TINT_ATTRS = new int[]{16843126};
```

There is **'extends Spinner'** found in file 'androidx/appcompat/widget/AppCompatSpinner.java':

```
line 45:
line 46: public class AppCompatSpinner extends Spinner implements
TintableBackgroundView {
line 47:     private static final int[] ATTRS_ANDROID_SPINNERMODE = new int[]
{16843505};
```

There is **'extends HorizontalScrollView'** found in file 'androidx/appcompat/widget/ScrollingTabContainerView.java':

```
line 35: @RestrictTo({Scope.LIBRARY_GROUP_PREFIX})
line 36: public class ScrollingTabContainerView extends HorizontalScrollView implements
OnItemSelectedListener {
line 37:     private static final int FADE_DURATION = 200;
```

There is **'extends MultiAutoCompleteTextView'** found in file 'androidx/appcompat/widget/AppCompatMultiAutoCompleteTextView.java':

```
line 18:
line 19: public class AppCompatMultiAutoCompleteTextView extends
MultiAutoCompleteTextView implements TintableBackgroundView {
line 20:     private static final int[] TINT_ATTRS = new int[]{16843126};
```

There is **'extends ViewPager'** found in file 'com/reactnativecommunity/viewpager/VerticalViewPager.java':

```
line 10:
line 11: public class VerticalViewPager extends ViewPager {
line 12:     private GestureDetector mGestureDetector;
```

There is **'extends WebView'** found in file 'com/reactnativecommunity/webview/RNCWebViewManager.java':

```
line 106:
line 107:     protected static class RNCWebViewClient extends WebViewClient {
line 108:         protected boolean mLastLoadFailed = false;
```

```
line 307:
line 308:     protected static class RNCWebView extends WebView implements
LifecycleEventListener {
line 309:         protected boolean hasScrollEvent = false;
```

CVSSv3 Base Score:
3.9 (AV:L/AC:L/PR:L/UI:R/S:U/C:L/I:L/A:N)

Reference:

- https://developer.android.com/guide/topics/ui/declaring-layout.html
- https://developer.android.com/guide/topics/resources/layout-resource.html
- https://blog.lookout.com/blog/2010/12/09/android-touch-event-hijacking/

EXPORTED ACTIVITIES [M1] [CWE-926] [SAST]

LOW

Description:
The mobile application contains exported activities that can be loaded and executed by other applications residing on the mobile device, including malicious ones, to trigger a legitimate application activity.
An activity is an Android component that allows to interact with the application in a particular way (e.g. perform certain actions or functions).

Example of insecure code:

```
<activity
    android:name=".SomeActivity"
    ....
    android:exported="true" />
```

Example of secure code:

```
<activity
    android:name=".SomeActivity"
    ....
    android:exported="false" />
```

Details:
There is **'MainActivity'** found in file 'android/AndroidManifest.xml':

```
line 45:         <activity
android:configChanges="keyboard|keyboardHidden|orientation|screenSize"
android:exported="true" android:label="@string/app_name"
android:launchMode="singleTask"
android:name="org.pathcheck.covidsafepaths.MainActivity"
android:screenOrientation="portrait" android:windowSoftInputMode="adjustResize"/>
```

CVSSv3 Base Score:

3.6 (AV:L/AC:H/PR:L/UI:N/S:U/C:L/I:L/A:N)

Reference:

- https://developer.android.com/reference/android/app/Activity.html
- https://developer.android.com/guide/topics/manifest/activity-element.html

EXPORTED BROADCAST RECEIVERS [M1] [CWE-925] [SAST]

LOW

Description:

The mobile application contains an exported receiver enabling other applications, including malicious ones, to send intents without restrictions.
By default, Broadcast Receivers is exported in Android, as the result any application will be able to send an intent to the Broadcast Receiver of the application. To define which applications can send intents to mobile application's Broadcast Receiver set relevant permissions in the Android Manifest file.

Example of insecure code:

```
<receiver
    android:name=".receivers.BatteryMonitoringReceiver"
    ...
</receiver>
```

Example of secure code:

```
<permission
    android:name="com.yourpage.permission.YOUR_PERMISSION"
    android:protectionLevel="signature" />

<uses-permission
    android:name="com.yourpage.permission.YOUR_PERMISSION" />

<receiver
    android:name=".receivers.BatteryMonitoringReceiver"
    android:permission="com.yourpage.permission.YOUR_PERMISSION"
    ...
</receiver>
```

Details:

There is **'BootCompletedReceiver'** found in file 'android/AndroidManifest.xml':

```
line 70:         <receiver android:enabled="true" android:exported="true"
android:name="com.marianhello.bgloc.BootCompletedReceiver">
```

There is **'FetchAlarmReceiver'** found in file 'android/AndroidManifest.xml':

```
line 75:         <receiver
android:name="com.transistorsoft.tsbackgroundfetch.FetchAlarmReceiver"/>
```

There is **'BootReceiver'** found in file 'android/AndroidManifest.xml':

```
line 77:         <receiver
android:name="com.transistorsoft.tsbackgroundfetch.BootReceiver">
```

CVSSv3 Base Score:
3.3 (AV:L/AC:L/PR:L/UI:N/S:U/C:N/I:L/A:N)

Reference:
- https://developer.android.com/guide/topics/manifest/receiver-element.html
- https://developer.android.com/reference/android/content/BroadcastReceiver.html

EXPORTED SERVICES [M1] [CWE-926] [SAST]

LOW

Description:
The mobile application contains an exported service.
By default, in Android services are not exported and cannot be invoked by other applications. However, if an intent filter is defined in Android Manifest file, it is exported by default. Particular attention should be given to the exported services, as without the specific permissions, they can be used by any other applications including malicious ones.

Example of insecure code:

```
<service
    android:name=".receivers.BatteryMonitoringReceiver"
    android:exported="true"
    ...
</service>
```

Example of secure code:

```
<service
    android:name=".receivers.BatteryMonitoringReceiver"
    ...
</service>
```

Details:
There is **'SyncService'** found in file 'android/AndroidManifest.xml':

```
line 55:         <service android:exported="true"
android:name="com.marianhello.bgloc.sync.SyncService" android:process=":sync">
```

There is **'FirebaseMessagingService'** found in file 'android/AndroidManifest.xml':

```
line 92:         <service android:exported="true"
android:name="com.google.firebase.messaging.FirebaseMessagingService">
```

There is **'FirebaseInstanceIdService'** found in file 'android/AndroidManifest.xml':

```
line 104:         <service android:exported="true"
android:name="com.google.firebase.iid.FirebaseInstanceIdService">
```

CVSSv3 Base Score:
3.3 (AV:L/AC:L/PR:L/UI:N/S:U/C:L/I:N/A:N)

Reference:
- https://developer.android.com/guide/components/services.html
- https://developer.android.com/reference/android/app/Service.html

- https://developer.android.com/guide/topics/manifest/service-element.html

EXPOSURE OF POTENTIALLY SENSITIVE DATA [M2] [CWE-200] [DAST]

LOW

Description:
The mobile application may expose potentially sensitive information during its runtimes.

- POTENTIAL USERNAME FOUND 05-10 15:45:05.071 I/PreBootBroadcaster(1712): Pre-boot of {com.google.android.gsf/com.google.android.gsf.loginservice.MigrateToAccountManagerBroadcastReceiver} for user 0
- POTENTIAL USERNAME FOUND 05-10 15:45:05.210 D/GoogleLoginService(2459): disabling the migration script since the migration is complete
- POTENTIAL USERNAME FOUND 05-10 15:45:05.210 D/GoogleLoginService(2459): no legacy android id exists to migrate
- POTENTIAL USERNAME FOUND 05-10 15:45:05.210 D/GoogleLoginService(2459): skipping migration because /data/user/0/com.google.android.gsf/databases/accounts.db doesn't exist
- POTENTIAL USERNAME FOUND 05-10 15:45:13.686 I/CheckinChimeraService(2409): Starting Checkin Task: CheckinNowTaskTag Reason : 1 Force : false UserId: 0
- POTENTIAL URL FOUND 05-10 15:45:15.034 W/OkHttpClient(1712): A connection to https://www.google.com/ was leaked. Did you forget to close a response body?
- POTENTIAL URL FOUND 05-10 15:45:18.864 I/CheckinRequestProcessor(2409): CheckinNowTaskTag : Checkin Succeeded: https://android.googleapis.com/checkin (fragment #1):
- POTENTIAL URL FOUND 05-10 15:45:47.638 D/NetworkMonitor/NetworkAgentInfo [MOBILE (LTE) - 100](1712): PROBE_HTTPS https://www.google.com/generate_204 time=567ms ret=502 request={User-Agent=[Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/60.0.3112.32 Safari/537.36]} headers={null=[HTTP/1.1 502 Bad Gateway], Connection=[close], Content-Length=[396], Content-Type=[text/html], Server=[mitmproxy 4.0.4], X-Android-Received-Millis=[1589118347637], X-Android-Response-Source=[NETWORK 502], X-Android-Selected-Protocol=[http/1.1], X-Android-Sent-Millis=[1589118347600]}
- POTENTIAL URL FOUND 05-10 15:45:48.134 D/NetworkMonitor/NetworkAgentInfo [MOBILE (LTE) - 100](1712): PROBE_HTTP http://connectivitycheck.gstatic.com/generate_204 time=1065ms ret=204 request={User-Agent=[Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/60.0.3112.32 Safari/537.36]} headers={null=[HTTP/1.1 204 No Content], Content-Length=[0], Date=[Sun, 10 May 2020 13:45:47 GMT], X-Android-Received-Millis=[1589118348134], X-Android-Response-Source=[NETWORK 204], X-Android-Selected-Protocol=[http/1.1], X-Android-Sent-Millis=[1589118347069]}
- POTENTIAL URL FOUND 05-10 15:45:49.696 D/NetworkMonitor/NetworkAgentInfo [MOBILE (LTE) - 100](1712): PROBE_FALLBACK http://www.google.com/gen_204 time=1562ms ret=204 request={User-Agent=[Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/60.0.3112.32 Safari/537.36]} headers={null=[HTTP/1.1 204 No Content], Content-Length=[0], Content-Type=[text/html; charset=UTF-8], Date=[Sun, 10 May 2020 13:45:49 GMT], Server=[gws], Set-Cookie=[1P_JAR=2020-05-10-13; expires=Tue, 09-Jun-2020 13:45:49 GMT; path=/; domain=.google.com; Secure], X-Android-Received-Millis=[1589118349696], X-Android-Response-Source=[NETWORK 204], X-Android-Selected-Protocol=[http/1.1], X-Android-Sent-Millis=[1589118348135], X-Frame-Options=[SAMEORIGIN], X-XSS-Protection=[0]}
- POTENTIAL EMAIL FOUND 05-10 15:45:02.070 I/ChromeSync(2409): [Persistence,AffiliationManager] One of affiliation or grouping data for android://SdkWC7t-Fplp-xCyrBC3HYMRo98n3PGCOwfQ3fwZLTBt7eLYiPMypamRcDx0ENO7gfvnNunrNutuW5aC5-I2eg==@org.pathcheck.covidsafepaths/ was not found. Marking affiliation data as stale...

CVSSv3 Base Score:
3.3 (AV:L/AC:L/PR:L/UI:N/S:U/C:L/I:N/A:N)

TEMPORARY FILE CREATION [SAST]

WARNING

Description:
The mobile application creates temporary files. Despite that cache files are usually private by default, it is recommended to make sure that temporary files are securely deleted when they are not required by the application anymore.

Details:
There is **'createTempFile()'** found in file 'kotlin/io/FilesKt__UtilsKt.java':

```
line 38:         Intrinsics.checkParameterIsNotNull(str, "prefix");
line 39:         File createTempFile = File.createTempFile(str, str2, file);
line 40:         createTempFile.delete();
```

```
line 61:         }
line 62:         return createTempFile(str, str2, file);
line 63:     }
```

```
line 65:     @NotNull
line 66:     public static final File createTempFile(@NotNull String str, @Nullable
String str2, @Nullable File file) {
line 67:         Intrinsics.checkParameterIsNotNull(str, "prefix");
line 68:         File createTempFile = File.createTempFile(str, str2, file);
line 69:         Intrinsics.checkExpressionValueIsNotNull(createTempFile,
"File.createTempFile(prefix, suffix, directory)");
line 70:         return createTempFile;
```

There is **'createTempFile()'** found in file 'com/reactnativecommunity/webview/RNCWebViewModule.java':

```
line 342:         }
line 343:         return File.createTempFile(str, str2,
getReactApplicationContext().getExternalFilesDir(null));
line 344:     }
```

There is **'createTempFile()'** found in file 'com/RNFetchBlob/RNFetchBlobBody.java':

```
line 275:         r4 = "rnfb-form-tmp";
line 276:         r4 = java.io.File.createTempFile(r4, r3, r0);
line 277:         r5 = new java.io.FileOutputStream;
```

There is **'createTempFile()'** found in file 'com/marianhello/bgloc/sync/BatchManager.java':

```
line 185:         r2 = ".json";
line 186:         r13 = java.io.File.createTempFile(r13, r2);    Catch:{ all -> 0x009d
}
line 187:         r2 = new java.io.FileOutputStream;     Catch:{ all -> 0x009d }
```

Reference:
- https://developer.android.com/training/basics/data-storage/files.html

USAGE OF IMPLICIT INTENT [M1] [CWE-927] [SAST]

WARNING

Description:

The mobile application uses implicit intent that may be insecure under certain conditions.
Intents enable mobile applications to communicate with each other by requesting to perform different actions for which they are better suited. An implicit intent, however, does not specify to which particular application it sends a request to perform an action. If a malicious application is installed on victim's device, it may also receive the implicit intent, respond to it and perform some action instead, or in addition to, a legitimate application.

Example of insecure code:

```
Intent sendIntent = new Intent();
```

Example of secure code:

```
Intent downloadIntent = new Intent(this, DownloadService.class);
```

Details:

There is **'new Intent()'** found in file 'androidx/core/app/JobIntentService.java':

```
line 347:             if (intent == null) {
line 348:                 intent = new Intent();
line 349:             }
```

There is **'new Intent()'** found in file 'androidx/core/app/NavUtils.java':

```
line 69:             } else {
line 70:                 makeMainActivity = new Intent().setComponent(componentName);
line 71:             }
```

```
line 93:         } else {
line 94:             makeMainActivity = new Intent().setComponent(componentName);
line 95:         }
```

```
line 109:         } else {
line 110:             makeMainActivity = new Intent().setComponent(componentName2);
line 111:         }
```

There is **'new Intent()'** found in file 'androidx/core/app/RemoteInput.java':

```
line 211:             if (clipDataIntentFromIntent == null) {
line 212:                 clipDataIntentFromIntent = new Intent();
line 213:             }
```

```
line 238:             if (clipDataIntentFromIntent == null) {
line 239:                 clipDataIntentFromIntent = new Intent();
line 240:             }
```

```
line 262:             if (clipDataIntentFromIntent == null) {
line 263:                 clipDataIntentFromIntent = new Intent();
line 264:             }
```

There is **'new Intent()'** found in file 'androidx/core/app/ShareCompat.java':

```
line 35:         private CharSequence mChooserTitle;
line 36:         private Intent mIntent = new
Intent().setAction("android.intent.action.SEND");
line 37:         private ArrayList<Uri> mStreams;
```

There is **'new Intent()'** found in file 'androidx/core/app/NotificationManagerCompat.java':

```
line 191:                 this.mCachedEnabledPackages = enabledListenerPackages;
line 192:                 List queryIntentServices =
this.mContext.getPackageManager().queryIntentServices(new
Intent().setAction(NotificationManagerCompat.ACTION_BIND_SIDE_CHANNEL), 0);
line 193:                 HashSet<ComponentName> hashSet = new HashSet();
```

There is **'new Intent()'** found in file 'androidx/core/widget/TextViewCompat.java':

```
line 170:         private Intent createProcessTextIntent() {
line 171:             return new
Intent().setAction("android.intent.action.PROCESS_TEXT").setType("text/plain");
line 172:         }
```

There is **'new Intent()'** found in file 'androidx/core/content/pm/ShortcutManagerCompat.java':

```
line 78:         if (createShortcutResultIntent == null) {
line 79:             createShortcutResultIntent = new Intent();
line 80:         }
```

There is **'new Intent()'** found in file 'cl/json/social/ShareIntent.java':

```
line 240:             String str = resolveInfo.activityInfo.packageName;
line 241:             Intent intent2 = new Intent();
line 242:             intent2.setComponent(new ComponentName(str,
resolveInfo.activityInfo.name));
```

There is **'new Intent()'** found in file 'com/reactnativecommunity/rnpermissions/RNPermissionsModule.java':

```
line 62:            ReactApplicationContext reactApplicationContext =
getReactApplicationContext();
line 63:            Intent intent = new Intent();
line 64:            intent.setAction("android.settings.APPLICATION_DETAILS_SETTINGS");
```

Reference:

- https://developer.android.com/guide/components/intents-filters.html
- https://developer.android.com/training/articles/security-tips.html

USAGE OF INTENT FILTER [M1] [CWE-927] [SAST]

WARNING

Description:

The mobile application uses an intent filter that may be a serious security risk if not properly implemented and filtered. Developers should not solely rely on intent filters for security purposes because they place no restrictions on explicit intents. Intent filters are defined in the Android Manifest file, they let developers choose which type of intents their application components are supposed to receive and handle.

Example of insecure code:

```
<intent-filter>
    <action android:name="android.intent.action.VIEW" />
    <action android:name="android.intent.action.EDIT" />
    <action android:name="android.intent.action.PICK" />
    <category android:name="android.intent.category.DEFAULT" />
    <data mimeType:name="vnd.android.cursor.dir/vnd.google.note" />
</intent-filter>
```

Example of secure code:

```
// When you use intent-filter, you have to perform input validation in your code.
```

Details:

There is **'<intent-filter>'** found in file 'android/AndroidManifest.xml':

```
line 39:        <activity android:label="@string/app_name_short"
android:name="org.pathcheck.covidsafepaths.SplashActivity"
android:theme="@style/SplashTheme">
line 40:            <intent-filter>
line 41:                <action android:name="android.intent.action.MAIN"/>
```

```
line 50:        <service android:exported="false"
android:name="com.dieam.reactnativepushnotification.modules.RNPushNotificationListenerS
ervice">
line 51:            <intent-filter>
line 52:                <action android:name="com.google.firebase.MESSAGING_EVENT"/>
```

```
line 55:         <service android:exported="true"
android:name="com.marianhello.bgloc.sync.SyncService" android:process=":sync">
line 56:             <intent-filter>
line 57:                 <action android:name="android.content.SyncAdapter"/>
```

```
line 61:         <service
android:name="com.marianhello.bgloc.sync.AuthenticatorService">
line 62:             <intent-filter>
line 63:                 <action android:name="android.accounts.AccountAuthenticator"/>
```

```
line 70:         <receiver android:enabled="true" android:exported="true"
android:name="com.marianhello.bgloc.BootCompletedReceiver">
line 71:             <intent-filter>
line 72:                 <action android:name="android.intent.action.BOOT_COMPLETED"/>
```

```
line 77:         <receiver
android:name="com.transistorsoft.tsbackgroundfetch.BootReceiver">
line 78:             <intent-filter>
line 79:                 <action android:name="android.intent.action.BOOT_COMPLETED"/>
```

```
line 98:         <receiver android:exported="true"
android:name="com.google.firebase.iid.FirebaseInstanceIdReceiver"
android:permission="com.google.android.c2dm.permission.SEND">
line 99:             <intent-filter>
line 100:                 <action
android:name="com.google.android.c2dm.intent.RECEIVE"/>
```

Reference:

- https://developer.android.com/guide/components/intents-filters.html
- https://developer.android.com/training/articles/security-tips.html

JS ENABLED IN A WEBVIEW [M10] [CWE-749] [SAST]

WARNING

Description:

The mobile application has enabled JavaScript in WebView. By default, JavaScript is disabled in WebView, if enabled it can bring various JS-related security issues, such as Cross-Site Scripting (XSS) attacks.

Example of insecure code:

```
WebSettings settings = webView.getSettings();
settings.setJavaScriptEnabled(true);
```

Example of secure code:

```
// Don't enable Javascript in WebView
```

Details:

There is **'setJavaScriptEnabled()'** found in file 'com/reactnativecommunity/webview/RNCWebViewManager.java':

```
line 556:     public void setJavaScriptEnabled(WebView webView, boolean z) {
line 557:         webView.getSettings().setJavaScriptEnabled(z);
```

Reference:

- https://developer.android.com/reference/android/webkit/WebView.html
- https://developer.android.com/reference/android/webkit/WebSettings.html

DYNAMIC LOAD OF CODE [M7] [CWE-94] [SAST]

WARNING

Description:

The mobile application uses dynamic load of executable code. Under certain circumstances, dynamic load of code can be dangerous. For example, if the code is located on an external storage (e.g. SD card), this can lead to code injection vulnerability if the external storage is world readable and/or writable and an attacker can access it.

Example of insecure code:

```
Object test = loader.loadClass("Test", true).newInstance();
```

Example of secure code:

```
// If you are using code from unsafe place (like external storage),
// you should sign and cryptographically verify your code.
```

Details:

There is **'ClassLoader'** found in file 'androidx/transition/TransitionInflater.java':

```
line 167:                     if (constructor == null) {
line 168:                         Class asSubclass = Class.forName(attributeValue,
false, this.mContext.getClassLoader()).asSubclass(cls);
line 169:                         if (asSubclass != null) {
```

There is **'ClassLoader'** found in file 'androidx/core/app/AppComponentFactory.java':

```
line 15: public class AppComponentFactory extends android.app.AppComponentFactory {
line 16:     public final Activity instantiateActivity(ClassLoader classLoader, String
str, Intent intent) throws InstantiationException, IllegalAccessException,
ClassNotFoundException {
line 17:         return (Activity)
CoreComponentFactory.checkCompatWrapper(instantiateActivityCompat(classLoader, str,
intent));
line 18:     }
line 19:
line 20:     public final Application instantiateApplication(ClassLoader classLoader,
String str) throws InstantiationException, IllegalAccessException,
ClassNotFoundException {
line 21:         return (Application)
CoreComponentFactory.checkCompatWrapper(instantiateApplicationCompat(classLoader,
str));
line 22:     }
line 23:
line 24:     public final BroadcastReceiver instantiateReceiver(ClassLoader
classLoader, String str, Intent intent) throws InstantiationException,
IllegalAccessException, ClassNotFoundException {
line 25:         return (BroadcastReceiver)
CoreComponentFactory.checkCompatWrapper(instantiateReceiverCompat(classLoader, str,
intent));
line 26:     }
line 27:
line 28:     public final ContentProvider instantiateProvider(ClassLoader classLoader,
String str) throws InstantiationException, IllegalAccessException,
ClassNotFoundException {
line 29:         return (ContentProvider)
CoreComponentFactory.checkCompatWrapper(instantiateProviderCompat(classLoader, str));
line 30:     }
line 31:
line 32:     public final Service instantiateService(ClassLoader classLoader, String
str, Intent intent) throws InstantiationException, IllegalAccessException,
ClassNotFoundException {
line 33:         return (Service)
CoreComponentFactory.checkCompatWrapper(instantiateServiceCompat(classLoader, str,
intent));
line 34:     }
```

```
line 36:     @NonNull
line 37:     public Application instantiateApplicationCompat(@NonNull ClassLoader
classLoader, @NonNull String str) throws InstantiationException,
IllegalAccessException, ClassNotFoundException {
line 38:         try {
line 39:             return (Application) Class.forName(str, false,
classLoader).getDeclaredConstructor(new Class[0]).newInstance(new Object[0]);
line 40:         } catch (NoSuchMethodException | InvocationTargetException e) {
```

```
line 45:     @NonNull
line 46:     public Activity instantiateActivityCompat(@NonNull ClassLoader
classLoader, @NonNull String str, @Nullable Intent intent) throws
InstantiationException, IllegalAccessException, ClassNotFoundException {
line 47:         try {
line 48:             return (Activity) Class.forName(str, false,
classLoader).getDeclaredConstructor(new Class[0]).newInstance(new Object[0]);
line 49:         } catch (NoSuchMethodException | InvocationTargetException e) {
```

```
line 54:     @NonNull
line 55:     public BroadcastReceiver instantiateReceiverCompat(@NonNull ClassLoader
classLoader, @NonNull String str, @Nullable Intent intent) throws
InstantiationException, IllegalAccessException, ClassNotFoundException {
line 56:         try {
line 57:             return (BroadcastReceiver) Class.forName(str, false,
classLoader).getDeclaredConstructor(new Class[0]).newInstance(new Object[0]);
line 58:         } catch (NoSuchMethodException | InvocationTargetException e) {
```

```
line 63:     @NonNull
line 64:     public Service instantiateServiceCompat(@NonNull ClassLoader classLoader,
@NonNull String str, @Nullable Intent intent) throws InstantiationException,
IllegalAccessException, ClassNotFoundException {
line 65:         try {
line 66:             return (Service) Class.forName(str, false,
classLoader).getDeclaredConstructor(new Class[0]).newInstance(new Object[0]);
line 67:         } catch (NoSuchMethodException | InvocationTargetException e) {
```

```
line 72:     @NonNull
line 73:     public ContentProvider instantiateProviderCompat(@NonNull ClassLoader
classLoader, @NonNull String str) throws InstantiationException,
IllegalAccessException, ClassNotFoundException {
line 74:         try {
line 75:             return (ContentProvider) Class.forName(str, false,
classLoader).getDeclaredConstructor(new Class[0]).newInstance(new Object[0]);
line 76:         } catch (NoSuchMethodException | InvocationTargetException e) {
```

There is **'ClassLoader'** found in file 'androidx/core/app/JobIntentService.java':

```
line 253:         /* JADX WARNING: Missing block: B:10:0x0013, code skipped:
line 254:
r1.getIntent().setExtrasClassLoader(r3.mService.getClassLoader());
line 255:      */
```

```
line 280:             r2 = r3.mService;
line 281:             r2 = r2.getClassLoader();
line 282:             r0.setExtrasClassLoader(r2);
line 283:             r0 = new
androidx.core.app.JobIntentService$JobServiceEngineImpl$WrapperWorkItem;
```

There is **'ClassLoader'** found in file 'androidx/core/app/CoreComponentFactory.java':

```
line 22:
line 23:     public Activity instantiateActivity(ClassLoader classLoader, String str,
Intent intent) throws InstantiationException, IllegalAccessException,
ClassNotFoundException {
line 24:         return (Activity)
checkCompatWrapper(super.instantiateActivity(classLoader, str, intent));
line 25:     }
line 26:
line 27:     public Application instantiateApplication(ClassLoader classLoader, String
str) throws InstantiationException, IllegalAccessException, ClassNotFoundException {
line 28:         return (Application)
checkCompatWrapper(super.instantiateApplication(classLoader, str));
line 29:     }
line 30:
line 31:     public BroadcastReceiver instantiateReceiver(ClassLoader classLoader,
String str, Intent intent) throws InstantiationException, IllegalAccessException,
ClassNotFoundException {
line 32:         return (BroadcastReceiver)
checkCompatWrapper(super.instantiateReceiver(classLoader, str, intent));
line 33:     }
line 34:
line 35:     public ContentProvider instantiateProvider(ClassLoader classLoader, String
str) throws InstantiationException, IllegalAccessException, ClassNotFoundException {
line 36:         return (ContentProvider)
checkCompatWrapper(super.instantiateProvider(classLoader, str));
line 37:     }
line 38:
line 39:     public Service instantiateService(ClassLoader classLoader, String str,
Intent intent) throws InstantiationException, IllegalAccessException,
ClassNotFoundException {
line 40:         return (Service)
checkCompatWrapper(super.instantiateService(classLoader, str, intent));
line 41:     }
```

There is **'ClassLoader'** found in file 'androidx/core/content/pm/ShortcutManagerCompat.java':

```
line 151:                     Class[] clsArr = new Class[]{Context.class};
line 152:                     sShortcutInfoCompatSaver = (ShortcutInfoCompatSaver)
Class.forName("androidx.sharetarget.ShortcutInfoCompatSaverImpl", false,
ShortcutManagerCompat.class.getClassLoader()).getMethod("getInstance",
clsArr).invoke(null, new Object[]{context});
line 153:                 } catch (Exception unused) {
```

There is **'ClassLoader'** found in file 'androidx/core/os/ParcelableCompat.java':

```
line 3: import android.os.Parcel;
line 4: import android.os.Parcelable.ClassLoaderCreator;
line 5: import android.os.Parcelable.Creator;
```

```
line 9:
line 10:     static class ParcelableCompatCreatorHoneycombMR2<T> implements
ClassLoaderCreator<T> {
line 11:         private final ParcelableCompatCreatorCallbacks<T> mCallbacks;
```

```
line 20:
line 21:         public T createFromParcel(Parcel parcel, ClassLoader classLoader) {
line 22:             return this.mCallbacks.createFromParcel(parcel, classLoader);
line 23:         }
```

There is **'ClassLoader'** found in file 'androidx/core/os/ParcelableCompatCreatorCallbacks.java':

```
line 6: public interface ParcelableCompatCreatorCallbacks<T> {
line 7:     T createFromParcel(Parcel parcel, ClassLoader classLoader);
line 8:
```

There is **'ClassLoader'** found in file 'androidx/drawerlayout/widget/DrawerLayout.java':

```
line 14: import android.os.Parcelable;
line 15: import android.os.Parcelable.ClassLoaderCreator;
line 16: import android.os.Parcelable.Creator;
```

```
line 261:     protected static class SavedState extends AbsSavedState {
line 262:         public static final Creator<SavedState> CREATOR = new
ClassLoaderCreator<SavedState>() {
line 263:             public SavedState createFromParcel(Parcel parcel, ClassLoader
classLoader) {
line 264:                 return new SavedState(parcel, classLoader);
line 265:             }
```

```
line 280:
line 281:         public SavedState(@NonNull Parcel parcel, @Nullable ClassLoader
classLoader) {
line 282:             super(parcel, classLoader);
line 283:             this.openDrawerGravity = parcel.readInt();
```

There is **'ClassLoader'** found in file 'androidx/versionedparcelable/VersionedParcelParcel.java':

```
line 207:     public <T extends Parcelable> T readParcelable() {
line 208:         return this.mParcel.readParcelable(getClass().getClassLoader());
line 209:     }
```

```
line 211:     public Bundle readBundle() {
line 212:         return this.mParcel.readBundle(getClass().getClassLoader());
line 213:     }
```

There is **'ClassLoader'** found in file 'androidx/versionedparcelable/VersionedParcel.java':

```
line 745:         i = 0;
line 746:         if ((exception instanceof Parcelable) &&
exception.getClass().getClassLoader() == Parcelable.class.getClassLoader()) {
line 747:             i = EX_PARCELABLE;
```

```
line 1028:                 public Class<?> resolveClass(ObjectStreamClass
objectStreamClass) throws IOException, ClassNotFoundException {
line 1029:                     Class cls = Class.forName(objectStreamClass.getName(),
false, getClass().getClassLoader());
line 1030:                     if (cls != null) {
```

```
line 1092:         System.currentTimeMillis();
line 1093:         method = Class.forName(str, true,
VersionedParcel.class.getClassLoader()).getDeclaredMethod("read", new Class[]
{VersionedParcel.class});
line 1094:         this.mReadCache.put(str, method);
```

```
line 1114:         }
line 1115:         cls2 = Class.forName(String.format("%s.%sParcelizer", new Object[]
{cls.getPackage().getName(), cls.getSimpleName()}), false, cls.getClassLoader());
line 1116:         this.mParcelizerCache.put(cls.getName(), cls2);
```

There is **'ClassLoader'** found in file 'androidx/versionedparcelable/ParcelUtils.java':

```
line 60:             }
line 61:             bundle.setClassLoader(ParcelUtils.class.getClassLoader());
line 62:             return fromParcelable(bundle.getParcelable(INNER_BUNDLE_KEY));
```

```
line 82:             bundle = (Bundle) bundle.getParcelable(str);
line 83:             bundle.setClassLoader(ParcelUtils.class.getClassLoader());
line 84:             Iterator it =
bundle.getParcelableArrayList(INNER_BUNDLE_KEY).iterator();
```

There is **'ClassLoader'** found in file 'androidx/viewpager/widget/ViewPager.java':

```
line 11: import android.os.Parcelable;
line 12: import android.os.Parcelable.ClassLoaderCreator;
line 13: import android.os.Parcelable.Creator;
```

```
line 132:     private Parcelable mRestoredAdapterState = null;
line 133:     private ClassLoader mRestoredClassLoader = null;
line 134:     private int mRestoredCurItem = -1;
```

```
line 278:     public static class SavedState extends AbsSavedState {
line 279:         public static final Creator<SavedState> CREATOR = new
ClassLoaderCreator<SavedState>() {
line 280:             public SavedState createFromParcel(Parcel parcel, ClassLoader
classLoader) {
line 281:                 return new SavedState(parcel, classLoader);
line 282:             }
```

```
line 292:         Parcelable adapterState;
line 293:         ClassLoader loader;
line 294:         int position;
```

```
line 315:
line 316:         SavedState(Parcel parcel, ClassLoader classLoader) {
line 317:             super(parcel, classLoader);
line 318:             if (classLoader == null) {
line 319:                 classLoader = getClass().getClassLoader();
line 320:             }
line 321:             this.position = parcel.readInt();
line 322:             this.adapterState = parcel.readParcelable(classLoader);
line 323:             this.loader = classLoader;
line 324:         }
```

```
line 444:             if (this.mRestoredCurItem >= 0) {
line 445:                 this.mAdapter.restoreState(this.mRestoredAdapterState,
this.mRestoredClassLoader);
line 446:                 setCurrentItemInternal(this.mRestoredCurItem, false, true);
```

```
line 448:                 this.mRestoredAdapterState = null;
line 449:                 this.mRestoredClassLoader = null;
line 450:             } else if (z) {
```

```
line 1478:                 this.mRestoredAdapterState = savedState.adapterState;
line 1479:                 this.mRestoredClassLoader = savedState.loader;
line 1480:             }
```

There is **'ClassLoader'** found in file 'androidx/viewpager/widget/PagerAdapter.java':

```
line 37:
line 38:     public void restoreState(@Nullable Parcelable parcelable, @Nullable
ClassLoader classLoader) {
line 39:     }
```

There is **'ClassLoader'** found in file 'androidx/fragment/app/FragmentPagerAdapter.java':

```
line 28:
line 29:     public void restoreState(@Nullable Parcelable parcelable, @Nullable
ClassLoader classLoader) {
line 30:     }
```

There is **'ClassLoader'** found in file 'androidx/fragment/app/FragmentFactory.java':

```
line 11:     @NonNull
line 12:     private static Class<?> loadClass(@NonNull ClassLoader classLoader,
@NonNull String str) throws ClassNotFoundException {
line 13:         Class<?> cls = (Class) sClassMap.get(str);
```

```
line 16:         }
line 17:         Class cls2 = Class.forName(str, false, classLoader);
line 18:         sClassMap.put(str, cls2);
```

```
line 21:
line 22:     static boolean isFragmentClass(@NonNull ClassLoader classLoader, @NonNull
String str) {
line 23:         try {
line 24:             return Fragment.class.isAssignableFrom(loadClass(classLoader,
str));
line 25:         } catch (ClassNotFoundException unused) {
```

```
line 30:     @NonNull
line 31:     public static Class<? extends Fragment> loadFragmentClass(@NonNull
ClassLoader classLoader, @NonNull String str) {
line 32:         StringBuilder stringBuilder;
```

```
line 34:         try {
line 35:             return loadClass(classLoader, str);
line 36:         } catch (ClassNotFoundException e) {
```

```
line 51:     @NonNull
line 52:     public Fragment instantiate(@NonNull ClassLoader classLoader, @NonNull
String str) {
line 53:         StringBuilder stringBuilder;
```

```
line 57:         try {
line 58:             return (Fragment) loadFragmentClass(classLoader,
str).getConstructor(new Class[0]).newInstance(new Object[0]);
line 59:         } catch (InstantiationException e) {
```

There is **'ClassLoader'** found in file 'androidx/fragment/app/FragmentTabHost.java':

```
line 287:                 if (tabInfoForTag.fragment == null) {
line 288:                     tabInfoForTag.fragment =
this.mFragmentManager.getFragmentFactory().instantiate(this.mContext.getClassLoader(),
tabInfoForTag.clss.getName());
line 289:                     tabInfoForTag.fragment.setArguments(tabInfoForTag.args);
```

There is **'ClassLoader'** found in file 'androidx/fragment/app/Fragment.java':

```
line 18: import android.os.Parcelable;
line 19: import android.os.Parcelable.ClassLoaderCreator;
line 20: import android.os.Parcelable.Creator;
```

```
line 168:         @NonNull
line 169:         public static final Creator<SavedState> CREATOR = new
ClassLoaderCreator<SavedState>() {
line 170:             public SavedState createFromParcel(Parcel parcel) {
```

```
line 173:
line 174:             public SavedState createFromParcel(Parcel parcel, ClassLoader
classLoader) {
line 175:                 return new SavedState(parcel, classLoader);
line 176:             }
```

```
line 191:
line 192:         SavedState(@NonNull Parcel parcel, @Nullable ClassLoader classLoader)
{
line 193:             this.mState = parcel.readBundle();
line 194:             if (classLoader != null) {
line 195:                 Bundle bundle = this.mState;
line 196:                 if (bundle != null) {
line 197:                     bundle.setClassLoader(classLoader);
line 198:                 }
```

```
line 349:         try {
line 350:             Fragment fragment = (Fragment)
FragmentFactory.loadFragmentClass(context.getClassLoader(), str).getConstructor(new
Class[0]).newInstance(new Object[0]);
line 351:             if (bundle != null) {
line 352:                 bundle.setClassLoader(fragment.getClass().getClassLoader());
line 353:                 fragment.setArguments(bundle);
```

There is **'ClassLoader'** found in file 'androidx/fragment/app/FragmentStatePagerAdapter.java':

```
line 170:
line 171:     public void restoreState(@Nullable Parcelable parcelable, @Nullable
ClassLoader classLoader) {
line 172:         if (parcelable != null) {
line 173:             Bundle bundle = (Bundle) parcelable;
line 174:             bundle.setClassLoader(classLoader);
line 175:             Parcelable[] parcelableArray =
bundle.getParcelableArray("states");
```

There is **'ClassLoader'** found in file 'androidx/fragment/app/FragmentState.java':

```
line 76:
line 77:     public Fragment instantiate(@NonNull ClassLoader classLoader, @NonNull
FragmentFactory fragmentFactory) {
line 78:         if (this.mInstance == null) {
```

```
line 80:             if (bundle != null) {
line 81:                 bundle.setClassLoader(classLoader);
line 82:             }
line 83:             this.mInstance = fragmentFactory.instantiate(classLoader,
this.mClassName);
line 84:             this.mInstance.setArguments(this.mArguments);
```

```
line 86:             if (bundle2 != null) {
line 87:                 bundle2.setClassLoader(classLoader);
line 88:                 this.mInstance.mSavedFragmentState = this.mSavedFragmentState;
```

There is **'ClassLoader'** found in file 'androidx/fragment/app/FragmentManagerImpl.java':

```
line 970:         r1 = r1.getContext();
line 971:         r1 = r1.getClassLoader();
line 972:         r0.setClassLoader(r1);
line 973:         r0 = r7.mSavedFragmentState;
```

```
line 2926:                         if (fragmentState.mSavedFragmentState != null) {
line 2927:
fragmentState.mSavedFragmentState.setClassLoader(this.mHost.getContext().getClassLoader
());
line 2928:                             fragment.mSavedViewState =
fragmentState.mSavedFragmentState.getSparseParcelableArray(VIEW_STATE_TAG);
```

```
line 2937:                     if (fragmentState2 != null) {
line 2938:                         instantiate =
fragmentState2.instantiate(this.mHost.getContext().getClassLoader(),
getFragmentFactory());
line 2939:                         instantiate.mFragmentManager = this;
```

```
line 3300:                 @NonNull
line 3301:                 public Fragment instantiate(@NonNull ClassLoader
classLoader, @NonNull String str) {
line 3302:                     return
FragmentManagerImpl.this.mHost.instantiate(FragmentManagerImpl.this.mHost.getContext(),
str, null);
```

```
line 3614:         obtainStyledAttributes.recycle();
line 3615:         if (str3 == null ||
!FragmentFactory.isFragmentClass(context.getClassLoader(), str3)) {
line 3616:             return null;
```

```
line 3650:         if (fragment == null) {
line 3651:             Fragment instantiate =
getFragmentFactory().instantiate(context.getClassLoader(), str3);
line 3652:             instantiate.mFromLayout = true;
```

There is **'ClassLoader'** found in file 'androidx/recyclerview/widget/StaggeredGridLayoutManager.java':

```
line 465:             this.mLastLayoutRTL = z;
line 466:             this.mFullSpanItems =
parcel.readArrayList(FullSpanItem.class.getClassLoader());
line 467:         }
```

There is **'ClassLoader'** found in file 'androidx/recyclerview/widget/RecyclerView.java':

```
line 18: import android.os.Parcelable;
line 19: import android.os.Parcelable.ClassLoaderCreator;
line 20: import android.os.Parcelable.Creator;
```

```
line 4437:     public static class SavedState extends AbsSavedState {
line 4438:         public static final Creator<SavedState> CREATOR = new
ClassLoaderCreator<SavedState>() {
line 4439:             public SavedState createFromParcel(Parcel parcel, ClassLoader
classLoader) {
line 4440:                 return new SavedState(parcel, classLoader);
line 4441:             }
```

```
line 4452:
line 4453:         SavedState(Parcel parcel, ClassLoader classLoader) {
line 4454:             super(parcel, classLoader);
line 4455:             if (classLoader == null) {
line 4456:                 classLoader = LayoutManager.class.getClassLoader();
line 4457:             }
line 4458:             this.mLayoutState = parcel.readParcelable(classLoader);
line 4459:         }
```

```
line 4683:                 try {
line 4684:                     ClassLoader classLoader;
line 4685:                     Constructor constructor;
line 4686:                     if (isInEditMode()) {
line 4687:                         classLoader = getClass().getClassLoader();
line 4688:                     } else {
line 4689:                         classLoader = context.getClassLoader();
line 4690:                     }
line 4691:                     Class asSubclass =
classLoader.loadClass(str).asSubclass(LayoutManager.class);
line 4692:                     Object[] objArr = null;
```

There is **'ClassLoader'** found in file 'androidx/appcompat/app/AppCompatDelegateImpl.java':

```
line 26: import android.os.Parcelable;
line 27: import android.os.Parcelable.ClassLoaderCreator;
line 28: import android.os.Parcelable.Creator;
```

```
line 236:         private static class SavedState implements Parcelable {
line 237:             public static final Creator<SavedState> CREATOR = new
ClassLoaderCreator<SavedState>() {
line 238:                 public SavedState createFromParcel(Parcel parcel, ClassLoader
classLoader) {
line 239:                     return SavedState.readFromParcel(parcel, classLoader);
line 240:                 }
```

```
line 268:
line 269:             static SavedState readFromParcel(Parcel parcel, ClassLoader
classLoader) {
line 270:                 SavedState savedState = new SavedState();
```

```
line 277:                 if (savedState.isOpen) {
line 278:                     savedState.menuState = parcel.readBundle(classLoader);
line 279:                 }
```

There is **'ClassLoader'** found in file 'androidx/appcompat/app/AppCompatViewInflater.java':

```
line 439:             str2 = str;
line 440:             constructor = Class.forName(str2, false,
context.getClassLoader()).asSubclass(View.class).getConstructor(sConstructorSignature);
line 441:             sConstructorMap.put(str, constructor);
```

There is **'ClassLoader'** found in file 'androidx/appcompat/widget/Toolbar.java':

```
line 8: import android.os.Parcelable;
line 9: import android.os.Parcelable.ClassLoaderCreator;
line 10: import android.os.Parcelable.Creator;
```

```
line 281:     public static class SavedState extends AbsSavedState {
line 282:         public static final Creator<SavedState> CREATOR = new
ClassLoaderCreator<SavedState>() {
line 283:             public SavedState createFromParcel(Parcel parcel, ClassLoader
classLoader) {
line 284:                 return new SavedState(parcel, classLoader);
line 285:             }
```

```
line 301:
line 302:         public SavedState(Parcel parcel, ClassLoader classLoader) {
line 303:             super(parcel, classLoader);
line 304:             this.expandedMenuItemId = parcel.readInt();
```

There is **'ClassLoader'** found in file 'androidx/appcompat/widget/SearchView.java':

```
line 19: import android.os.Parcelable;
line 20: import android.os.Parcelable.ClassLoaderCreator;
line 21: import android.os.Parcelable.Creator;
```

```
line 335:     static class SavedState extends AbsSavedState {
line 336:         public static final Creator<SavedState> CREATOR = new
ClassLoaderCreator<SavedState>() {
line 337:             public SavedState createFromParcel(Parcel parcel, ClassLoader
classLoader) {
line 338:                 return new SavedState(parcel, classLoader);
line 339:             }
```

```
line 354:
line 355:         public SavedState(Parcel parcel, ClassLoader classLoader) {
line 356:             super(parcel, classLoader);
line 357:             this.isIconified = ((Boolean)
parcel.readValue(null)).booleanValue();
```

There is **'ClassLoader'** found in file 'androidx/appcompat/view/SupportMenuInflater.java':

```
line 276:             try {
line 277:                 Constructor constructor = Class.forName(str, false,
SupportMenuInflater.this.mContext.getClassLoader()).getConstructor(clsArr);
line 278:                 constructor.setAccessible(true);
```

There is **'ClassLoader'** found in file 'androidx/customview/view/AbsSavedState.java':

```
line 4: import android.os.Parcelable;
line 5: import android.os.Parcelable.ClassLoaderCreator;
line 6: import android.os.Parcelable.Creator;
```

```
line 10: public abstract class AbsSavedState implements Parcelable {
line 11:     public static final Creator<AbsSavedState> CREATOR = new
ClassLoaderCreator<AbsSavedState>() {
line 12:         public AbsSavedState createFromParcel(Parcel parcel, ClassLoader
classLoader) {
line 13:             if (parcel.readParcelable(classLoader) == null) {
line 14:                 return AbsSavedState.EMPTY_STATE;
```

```
line 53:
line 54:     protected AbsSavedState(@NonNull Parcel parcel, @Nullable ClassLoader
classLoader) {
line 55:         Parcelable readParcelable = parcel.readParcelable(classLoader);
line 56:         if (readParcelable == null) {
```

There is **'ClassLoader'** found in file 'androidx/slidingpanelayout/widget/SlidingPaneLayout.java':

```
line 13: import android.os.Parcelable;
line 14: import android.os.Parcelable.ClassLoaderCreator;
line 15: import android.os.Parcelable.Creator;
```

```
line 273:     static class SavedState extends AbsSavedState {
line 274:         public static final Creator<SavedState> CREATOR = new
ClassLoaderCreator<SavedState>() {
line 275:             public SavedState createFromParcel(Parcel parcel, ClassLoader
classLoader) {
line 276:                 return new SavedState(parcel, null);
```

```
line 292:
line 293:         SavedState(Parcel parcel, ClassLoader classLoader) {
line 294:             super(parcel, classLoader);
line 295:             this.isOpen = parcel.readInt() != 0;
```

There is **'ClassLoader'** found in file 'androidx/coordinatorlayout/widget/CoordinatorLayout.java':

```
line 14: import android.os.Parcelable;
line 15: import android.os.Parcelable.ClassLoaderCreator;
line 16: import android.os.Parcelable.Creator;
```

```
line 552:     protected static class SavedState extends AbsSavedState {
line 553:         public static final Creator<SavedState> CREATOR = new
ClassLoaderCreator<SavedState>() {
line 554:             public SavedState createFromParcel(Parcel parcel, ClassLoader
classLoader) {
line 555:                 return new SavedState(parcel, classLoader);
line 556:             }
```

```
line 567:
line 568:         public SavedState(Parcel parcel, ClassLoader classLoader) {
line 569:             super(parcel, classLoader);
line 570:             int readInt = parcel.readInt();
```

```
line 572:             parcel.readIntArray(iArr);
line 573:             Parcelable[] readParcelableArray =
parcel.readParcelableArray(classLoader);
line 574:             this.behaviorStates = new SparseArray(readInt);
```

```
line 1031:             if (constructor == null) {
line 1032:                 constructor =
context.getClassLoader().loadClass(str2).getConstructor(CONSTRUCTOR_PARAMS);
line 1033:                 constructor.setAccessible(true);
```

There is **'ClassLoader'** found in file 'androidx/savedstate/Recreator.java':

```
line 67:         try {
line 68:             Class asSubclass = Class.forName(str, false,
Recreator.class.getClassLoader()).asSubclass(AutoRecreated.class);
line 69:             try {
```

There is **'ClassLoader'** found in file 'androidx/media/MediaBrowserServiceCompat.java':

```
line 501:                     Bundle bundle2 = data.getBundle(str5);
line 502:                     MediaSessionCompat.ensureClassLoader(bundle2);
line 503:                     this.mServiceBinderImpl.connect(data.getString(str4),
data.getInt(str3), data.getInt(str2), bundle2, new
ServiceCallbacksCompat(message.replyTo));
```

```
line 509:                    bundle =
data.getBundle(MediaBrowserProtocol.DATA_OPTIONS);
line 510:                    MediaSessionCompat.ensureClassLoader(bundle);
line 511:
this.mServiceBinderImpl.addSubscription(data.getString(str7),
BundleCompat.getBinder(data, str), bundle, new
ServiceCallbacksCompat(message.replyTo));
```

```
line 520:                    Bundle bundle3 = data.getBundle(str5);
line 521:                    MediaSessionCompat.ensureClassLoader(bundle3);
line 522:                    this.mServiceBinderImpl.registerCallbacks(new
ServiceCallbacksCompat(message.replyTo), data.getString(str4), data.getInt(str3),
data.getInt(str2), bundle3);
```

```
line 528:                    bundle =
data.getBundle(MediaBrowserProtocol.DATA_SEARCH_EXTRAS);
line 529:                    MediaSessionCompat.ensureClassLoader(bundle);
line 530:
this.mServiceBinderImpl.search(data.getString(MediaBrowserProtocol.DATA_SEARCH_QUERY),
bundle, (ResultReceiver) data.getParcelable(str6), new
ServiceCallbacksCompat(message.replyTo));
```

```
line 533:                    bundle =
data.getBundle(MediaBrowserProtocol.DATA_CUSTOM_ACTION_EXTRAS);
line 534:                    MediaSessionCompat.ensureClassLoader(bundle);
line 535:
this.mServiceBinderImpl.sendCustomAction(data.getString(MediaBrowserProtocol.DATA_CUSTO
M_ACTION), bundle, (ResultReceiver) data.getParcelable(str6), new
ServiceCallbacksCompat(message.replyTo));
```

```
line 551:             Bundle data = message.getData();
line 552:             data.setClassLoader(MediaBrowserCompat.class.getClassLoader());
line 553:             data.putInt(MediaBrowserProtocol.DATA_CALLING_UID,
Binder.getCallingUid());
```

There is **'ClassLoader'** found in file 'androidx/media/MediaBrowserServiceCompatApi21.java':

```
line 38:         public android.service.media.MediaBrowserService.BrowserRoot
onGetRoot(String str, int i, Bundle bundle) {
line 39:             MediaSessionCompat.ensureClassLoader(bundle);
line 40:             BrowserRoot onGetRoot = this.mServiceProxy.onGetRoot(str, i,
bundle == null ? null : new Bundle(bundle));
```

There is **'ClassLoader'** found in file 'androidx/media/MediaBrowserServiceCompatApi26.java':

```
line 61:         public void onLoadChildren(String str, Result<List<MediaItem>> result,
Bundle bundle) {
line 62:             MediaSessionCompat.ensureClassLoader(bundle);
line 63:             ((ServiceCompatProxy) this.mServiceProxy).onLoadChildren(str, new
ResultWrapper(result), bundle);
```

There is **'ClassLoader'** found in file 'ch/qos/logback/core/util/OptionHelper.java':

```
line 78:     public static Object instantiateByClassName(String str, Class<?> cls,
Context context) throws IncompatibleClassException, DynamicClassLoadingException {
line 79:         return instantiateByClassName(str, (Class) cls,
Loader.getClassLoaderOfObject(context));
line 80:     }
line 81:
line 82:     public static Object instantiateByClassName(String str, Class<?> cls,
ClassLoader classLoader) throws IncompatibleClassException,
DynamicClassLoadingException {
line 83:         return instantiateByClassNameAndParameter(str, (Class) cls,
classLoader, null, null);
line 84:     }
```

```
line 86:     public static Object instantiateByClassNameAndParameter(String str,
Class<?> cls, Context context, Class<?> cls2, Object obj) throws
IncompatibleClassException, DynamicClassLoadingException {
line 87:         return instantiateByClassNameAndParameter(str, (Class) cls,
Loader.getClassLoaderOfObject(context), (Class) cls2, obj);
line 88:     }
line 89:
line 90:     public static Object instantiateByClassNameAndParameter(String str,
Class<?> cls, ClassLoader classLoader, Class<?> cls2, Object obj) throws
IncompatibleClassException, DynamicClassLoadingException {
line 91:         if (str != null) {
line 92:             try {
line 93:                 Class loadClass = classLoader.loadClass(str);
line 94:                 if (!cls.isAssignableFrom(loadClass)) {
```

There is **'ClassLoader'** found in file 'ch/qos/logback/core/util/Loader.java':

```
line 11:
line 12: public class Loader {
line 13:     private static boolean HAS_GET_CLASS_LOADER_PERMISSION = ((Boolean)
AccessController.doPrivileged(new PrivilegedAction<Boolean>() {
```

```
line 15:             try {
line 16:                 AccessController.checkPermission(new
RuntimePermission("getClassLoader"));
line 17:                 return Boolean.valueOf(true);
```

```
line 33:
line 34:     public static ClassLoader getClassLoaderAsPrivileged(final Class<?> cls) {
line 35:         return !HAS_GET_CLASS_LOADER_PERMISSION ? null : (ClassLoader)
AccessController.doPrivileged(new PrivilegedAction<ClassLoader>() {
line 36:             public ClassLoader run() {
line 37:                 return cls.getClassLoader();
line 38:             }
```

```
line 41:
line 42:     public static ClassLoader getClassLoaderOfClass(Class<?> cls) {
line 43:         ClassLoader classLoader = cls.getClassLoader();
line 44:         return classLoader == null ? ClassLoader.getSystemClassLoader() :
classLoader;
line 45:     }
line 46:
line 47:     public static ClassLoader getClassLoaderOfObject(Object obj) {
line 48:         if (obj != null) {
line 49:             return getClassLoaderOfClass(obj.getClass());
line 50:         }
```

```
line 53:
line 54:     public static URL getResource(String str, ClassLoader classLoader) {
line 55:         try {
line 56:             return classLoader.getResource(str);
line 57:         } catch (Throwable unused) {
```

```
line 61:
line 62:     public static URL getResourceBySelfClassLoader(String str) {
line 63:         return getResource(str, getClassLoaderOfClass(Loader.class));
line 64:     }
line 65:
line 66:     public static Set<URL> getResourceOccurrenceCount(String str, ClassLoader
classLoader) throws IOException {
line 67:         HashSet hashSet = new HashSet();
line 68:         Enumeration resources = classLoader.getResources(str);
line 69:         while (resources.hasMoreElements()) {
```

```
line 74:
line 75:     public static ClassLoader getTCL() {
line 76:         return Thread.currentThread().getContextClassLoader();
line 77:     }
```

```
line 91:     public static Class<?> loadClass(String str, Context context) throws
ClassNotFoundException {
line 92:         return getClassLoaderOfObject(context).loadClass(str);
line 93:     }
```

There is **'ClassLoader'** found in file 'ch/qos/logback/core/util/LocationUtil.java':

```
line 12:         if (str != null) {
line 13:             URL resourceBySelfClassLoader;
line 14:             if (!str.matches(SCHEME_PATTERN)) {
line 15:                 resourceBySelfClassLoader =
Loader.getResourceBySelfClassLoader(str);
line 16:             } else if (str.startsWith(CLASSPATH_SCHEME)) {
```

```
line 21:                 if (substring.length() != 0) {
line 22:                     resourceBySelfClassLoader =
Loader.getResourceBySelfClassLoader(substring);
line 23:                 } else {
```

```
line 26:             } else {
line 27:                 resourceBySelfClassLoader = new URL(str);
line 28:             }
line 29:             if (resourceBySelfClassLoader != null) {
line 30:                 return resourceBySelfClassLoader;
line 31:             }
```

There is **'ClassLoader'** found in file 'ch/qos/logback/core/property/ResourceExistsPropertyDefiner.java':

```
line 14:         }
line 15:         return
PropertyDefinerBase.booleanAsStr(Loader.getResourceBySelfClassLoader(this.resourceStr)
!= null);
line 16:     }
```

There is **'ClassLoader'** found in file 'ch/qos/logback/core/joran/util/PropertySetter.java':

```
line 119:             stringBuilder2.append("[");
line 120:             stringBuilder2.append(clsArr[0].getClassLoader());
line 121:             stringBuilder2.append("] whereas object of type ");
```

```
line 126:             stringBuilder2.append("\" was loaded by [");
line 127:             stringBuilder2.append(cls.getClassLoader());
line 128:             stringBuilder2.append("].");
```

There is **'ClassLoader'** found in file 'ch/qos/logback/core/joran/action/PropertyAction.java':

```
line 50:             str =
interpretationContext.subst(attributes.getValue(RESOURCE_ATTRIBUTE));
line 51:             URL resourceBySelfClassLoader =
Loader.getResourceBySelfClassLoader(str);
line 52:             if (resourceBySelfClassLoader == null) {
line 53:                 stringBuilder2 = new StringBuilder();
```

```
line 61:             try {
line 62:                 loadAndSetProperties(interpretationContext,
resourceBySelfClassLoader.openStream(), stringToScope);
line 63:             } catch (IOException e3) {
```

There is **'ClassLoader'** found in file 'ch/qos/logback/core/joran/action/AbstractIncludeAction.java':

```
line 125:     private URL resourceAsURL(String str) {
line 126:         URL resourceBySelfClassLoader =
Loader.getResourceBySelfClassLoader(str);
line 127:         if (resourceBySelfClassLoader != null) {
line 128:             return resourceBySelfClassLoader;
line 129:         }
```

There is **'ClassLoader'** found in file 'ch/qos/logback/core/android/SystemPropertiesProxy.java':

```
line 10:
line 11:     private SystemPropertiesProxy(ClassLoader classLoader) {
line 12:         try {
line 13:             setClassLoader(classLoader);
line 14:         } catch (Exception unused) {
```

```
line 57:
line 58:     public void setClassLoader(ClassLoader classLoader) throws
ClassNotFoundException, SecurityException, NoSuchMethodException {
line 59:         if (classLoader == null) {
line 60:             classLoader = getClass().getClassLoader();
line 61:         }
line 62:         this.SystemProperties =
classLoader.loadClass("android.os.SystemProperties");
line 63:         this.getString = this.SystemProperties.getMethod("get", new Class[]
{String.class, String.class});
```

There is **'ClassLoader'** found in file 'ch/qos/logback/classic/util/ContextInitializer.java':

```
line 17:     public static final String STATUS_LISTENER_CLASS =
"logback.statusListenerClass";
line 18:     final ClassLoader classLoader = Loader.getClassLoaderOfObject(this);
line 19:     final LoggerContext loggerContext;
```

```
line 27:     /* JADX WARNING: Missing block: B:19:?, code skipped:
line 28:             r2 = ch.qos.logback.core.util.Loader.getResource(r0,
r4.classLoader);
line 29:      */
```

```
line 36:     /* JADX WARNING: Missing block: B:22:0x0049, code skipped:
line 37:             r5 = r4.classLoader;
line 38:      */
```

```
line 54:     /* JADX WARNING: Missing block: B:28:0x0057, code skipped:
line 55:             r5 = r4.classLoader;
line 56:      */
```

```
line 72:     /* JADX WARNING: Missing block: B:34:0x0067, code skipped:
line 73:             statusOnResourceSearch(r0, r4.classLoader, null);
line 74:      */
```

```
line 92:     L_0x001c:
line 93:         r3 = r4.classLoader;    Catch:{ MalformedURLException -> 0x003f }
line 94:         r4.statusOnResourceSearch(r0, r3, r0);  Catch:{ MalformedURLException
-> 0x003f }
```

```
line 104:     L_0x0031:
line 105:         r5 = r4.classLoader;
line 106:         if (r2 == 0) goto L_0x0039;
```

```
line 116:     L_0x003f:
line 117:         r2 = r4.classLoader;   Catch:{ all -> 0x003d }
line 118:         r2 = ch.qos.logback.core.util.Loader.getResource(r0, r2);      Catch:
{ all -> 0x003d }
```

```
line 122:     L_0x0049:
line 123:         r5 = r4.classLoader;
line 124:         if (r2 == 0) goto L_0x0051;
```

```
line 133:     L_0x0057:
line 134:         r5 = r4.classLoader;
line 135:         if (r2 == 0) goto L_0x0060;
```

```
line 146:     L_0x0067:
line 147:         r5 = r4.classLoader;
line 148:         r4.statusOnResourceSearch(r0, r5, r1);
```

```
line 161:         stringBuilder.append(AUTOCONFIG_FILE);
line 162:         return getResource(stringBuilder.toString(), this.classLoader, z);
line 163:     }
line 164:
line 165:     private InputStream getResource(String str, ClassLoader classLoader,
boolean z) {
line 166:         InputStream resourceAsStream = classLoader.getResourceAsStream(str);
line 167:         if (z) {
```

```
line 171:             }
line 172:             statusOnResourceSearch(str, classLoader, str2);
line 173:         }
```

```
line 176:
line 177:     private void statusOnResourceSearch(String str, ClassLoader classLoader,
String str2) {
line 178:         StatusManager statusManager = this.loggerContext.getStatusManager();
```

There is **'ClassLoader'** found in file 'ch/qos/logback/classic/spi/PackagingDataCalculator.java':

```
line 11:
line 12:     private Class bestEffortLoadClass(ClassLoader classLoader, String str) {
line 13:         Class loadClass = loadClass(classLoader, str);
line 14:         if (loadClass != null) {
```

```
line 16:         }
line 17:         ClassLoader contextClassLoader =
Thread.currentThread().getContextClassLoader();
line 18:         if (contextClassLoader != classLoader) {
line 19:             loadClass = loadClass(contextClassLoader, str);
line 20:         }
```

```
line 35:
line 36:     private ClassPackagingData computeBySTEP(StackTraceElementProxy
stackTraceElementProxy, ClassLoader classLoader) {
line 37:         String className = stackTraceElementProxy.ste.getClassName();
```

```
line 41:         }
line 42:         Class bestEffortLoadClass = bestEffortLoadClass(classLoader,
className);
line 43:         ClassPackagingData classPackagingData2 = new
ClassPackagingData(getCodeLocation(bestEffortLoadClass),
getImplementationVersion(bestEffortLoadClass), false);
```

```
line 56:
line 57:     private Class loadClass(ClassLoader classLoader, String str) {
line 58:         if (classLoader == null) {
line 59:             return null;
```

```
line 61:         try {
line 62:             return classLoader.loadClass(str);
line 63:         } catch (ClassNotFoundException | NoClassDefFoundError unused) {
```

```
line 128:     /* Access modifiers changed, original: 0000 */
line 129:     public void populateUncommonFrames(int i, StackTraceElementProxy[]
stackTraceElementProxyArr, ClassLoader classLoader) {
line 130:         int length = stackTraceElementProxyArr.length - i;
```

```
line 132:             StackTraceElementProxy stackTraceElementProxy =
stackTraceElementProxyArr[i];
line 133:
stackTraceElementProxy.setClassPackagingData(computeBySTEP(stackTraceElementProxy,
classLoader));
line 134:         }
```

There is **'ClassLoader'** found in file 'ch/qos/logback/classic/android/AndroidManifestPropertiesUtil.java':

```
line 21:         StatusManager statusManager = context.getStatusManager();
line 22:         InputStream resourceAsStream =
Loader.getClassLoaderOfObject(context).getResourceAsStream("AndroidManifest.xml");
line 23:         if (resourceAsStream == null) {
```

There is **'ClassLoader'** found in file 'kotlin/internal/PlatformImplementationsKt.java':

```
line 25:         r1 = "ClassCastException(\"Inst…baseTypeCL\").initCause(e)";
line 26:         r2 = ", base type classloader: ";
line 27:         r3 = "Instance classloader: ";
line 28:         r4 = "null cannot be cast to non-null type
kotlin.internal.PlatformImplementations";
```

```
line 49:          r6 = r6.getClass();     Catch:{ ClassNotFoundException -> 0x0060 }
line 50:          r6 = r6.getClassLoader();       Catch:{ ClassNotFoundException ->
0x0060 }
line 51:          r8 = kotlin.internal.PlatformImplementations.class;
line 52:          r8 = r8.getClassLoader();       Catch:{ ClassNotFoundException ->
0x0060 }
line 53:          r9 = new java.lang.ClassCastException;  Catch:{ ClassNotFoundException
-> 0x0060 }
```

```
line 83:          r6 = r6.getClass();     Catch:{ ClassNotFoundException -> 0x00ad }
line 84:          r6 = r6.getClassLoader();       Catch:{ ClassNotFoundException ->
0x00ad }
line 85:          r8 = kotlin.internal.PlatformImplementations.class;
line 86:          r8 = r8.getClassLoader();       Catch:{ ClassNotFoundException ->
0x00ad }
line 87:          r9 = new java.lang.ClassCastException;  Catch:{ ClassNotFoundException
-> 0x00ad }
```

```
line 121:          r0 = r0.getClass();    Catch:{ ClassNotFoundException -> 0x0100 }
line 122:          r0 = r0.getClassLoader();      Catch:{ ClassNotFoundException ->
0x0100 }
line 123:          r7 = kotlin.internal.PlatformImplementations.class;
line 124:          r7 = r7.getClassLoader();      Catch:{ ClassNotFoundException ->
0x0100 }
line 125:          r8 = new java.lang.ClassCastException;         Catch:{
ClassNotFoundException -> 0x0100 }
```

```
line 156:          r0 = r0.getClass();    Catch:{ ClassNotFoundException -> 0x014d }
line 157:          r0 = r0.getClassLoader();      Catch:{ ClassNotFoundException ->
0x014d }
line 158:          r5 = kotlin.internal.PlatformImplementations.class;
line 159:          r5 = r5.getClassLoader();      Catch:{ ClassNotFoundException ->
0x014d }
line 160:          r6 = new java.lang.ClassCastException;         Catch:{
ClassNotFoundException -> 0x014d }
```

```
line 189:          } catch (ClassCastException e) {
line 190:              ClassLoader classLoader = obj.getClass().getClassLoader();
line 191:              Intrinsics.reifiedOperationMarker(4, str);
line 192:              ClassLoader classLoader2 = Object.class.getClassLoader();
line 193:              StringBuilder stringBuilder = new StringBuilder();
line 194:              stringBuilder.append("Instance classloader: ");
line 195:              stringBuilder.append(classLoader);
line 196:              stringBuilder.append(", base type classloader: ");
line 197:              stringBuilder.append(classLoader2);
line 198:              Throwable initCause = new
ClassCastException(stringBuilder.toString()).initCause(e);
```

There is **'ClassLoader'** found in file 'kotlin/concurrent/ThreadsKt.java':

```
line 11:
line 12: @Metadata(bv = {1, 0, 3}, d1 =
{"\u0000:\n\u0000\n\u0002\u0018\u0002\n\u0000\n\u0002\u0010\u000b\n\u0002\b\u0002\n\u00
02\u0018\u0002\n\u0000\n\u0002\u0010\u000e\n\u0000\n\u0002\u0010\b\n\u0000\n\u0002\u001
8\u0002\n\u0002\u0010\u0002\n\u0002\b\u0002\n\u0002\u0010\u0000\n\u0002\u0018\u0002\n\u
0002\b\u0003\u001aJ\u0010\u0000\u001a\u00020\u00012\b\b\u0002\u0010\u0002\u001a\u00020\
u00032\b\b\u0002\u0010\u0004\u001a\u00020\u00032\n\b\u0002\u0010\u0005\u001a\u0004\u001
8\u00010\u00062\n\b\u0002\u0010\u0007\u001a\u0004\u0018\u00010\b2\b\b\u0002\u0010\t\u00
1a\u00020\n2\f\u0010\u000b\u001a\b\u0012\u0004\u0012\u00020\r0\f\u001a0\u0010\u000e\u00
1a\u0002H\u000f\"\b\b\u0000\u0010\u000f*\u00020\u0010*\b\u0012\u0004\u0012\u0002H\u000f
0\u00112\f\u0010\u0012\u001a\b\u0012\u0004\u0012\u0002H\u000f0\fH□\b¢
\u0006\u0002\u0010\u0013¨\u0006\u0014"}, d2 = {"thread", "Ljava/lang/Thread;", "start",
"", "isDaemon", "contextClassLoader", "Ljava/lang/ClassLoader;", "name", "",
"priority", "", "block", "Lkotlin/Function0;", "", "getOrSet", "T", "",
"Ljava/lang/ThreadLocal;", "default", "
(Ljava/lang/ThreadLocal;Lkotlin/jvm/functions/Function0;)Ljava/lang/Object;", "kotlin-
stdlib"}, k = 2, mv = {1, 1, 16})
line 13: @JvmName(name = "ThreadsKt")
```

```
line 15: public final class ThreadsKt {
line 16:     public static /* synthetic */ Thread thread$default(boolean z, boolean z2,
ClassLoader classLoader, String str, int i, Function0 function0, int i2, Object obj) {
line 17:         boolean z3 = (i2 & 1) != 0 ? true : z;
```

```
line 19:         if ((i2 & 4) != 0) {
line 20:             classLoader = (ClassLoader) null;
line 21:         }
line 22:         ClassLoader classLoader2 = classLoader;
line 23:         if ((i2 & 8) != 0) {
```

```
line 25:         }
line 26:         return thread(z3, z4, classLoader2, str, (i2 & 16) != 0 ? -1 : i,
function0);
line 27:     }
```

```
line 29:     @NotNull
line 30:     public static final Thread thread(boolean z, boolean z2, @Nullable
ClassLoader classLoader, @Nullable String str, int i, @NotNull Function0<Unit>
function0) {
line 31:         Intrinsics.checkParameterIsNotNull(function0, "block");
```

```
line 41:         }
line 42:         if (classLoader != null) {
line 43:             threadsKt$thread$thread$1.setContextClassLoader(classLoader);
line 44:         }
```

There is **'ClassLoader'** found in file 'kotlin/coroutines/jvm/internal/ModuleNameRetriever.java':

```
line 75:         try {
line 76:             Cache cache = new Cache(Class.class.getDeclaredMethod("getModule",
new Class[0]),
baseContinuationImpl.getClass().getClassLoader().loadClass("java.lang.Module").getDecla
redMethod("getDescriptor", new Class[0]),
baseContinuationImpl.getClass().getClassLoader().loadClass("java.lang.module.ModuleDesc
riptor").getDeclaredMethod(Action.NAME_ATTRIBUTE, new Class[0]));
line 77:             cache = cache;
```

There is **'ClassLoader'** found in file 'com/marianhello/bgloc/BackgroundGeolocationFacade.java':

```
line 75:                     BackgroundGeolocationFacade.this.logger.debug("Received
MSG_ON_LOCATION");
line 76:
extras.setClassLoader(LocationServiceImpl.class.getClassLoader());
line 77:
BackgroundGeolocationFacade.this.mDelegate.onLocationChanged((BackgroundLocation)
extras.getParcelable(str));
```

```
line 80:                     BackgroundGeolocationFacade.this.logger.debug("Received
MSG_ON_STATIONARY");
line 81:
extras.setClassLoader(LocationServiceImpl.class.getClassLoader());
line 82:                     BackgroundLocation backgroundLocation =
(BackgroundLocation) extras.getParcelable(str);
```

```
line 87:                     BackgroundGeolocationFacade.this.logger.debug("Received
MSG_ON_ACTIVITY");
line 88:
extras.setClassLoader(LocationServiceImpl.class.getClassLoader());
line 89:
BackgroundGeolocationFacade.this.mDelegate.onActivityChanged((BackgroundActivity)
extras.getParcelable(str));
```

Reference:

- https://developer.android.com/reference/java/lang/ClassLoader.html
- https://developer.android.com/reference/dalvik/system/DexClassLoader.html
- https://developer.android.com/reference/java/security/SecureClassLoader.html
- https://developer.android.com/reference/java/net/URLClassLoader.html

WEAK HASHING ALGORITHMS [M5] [CWE-916] [SAST]

WARNING

Description:

The mobile application uses weak hashing algorithms. Weak hashing algorithms (e.g. MD2, MD4, MD5 or SHA-1) can be vulnerable to collisions and other security weaknesses, and should not be used when reliable hashing of data is required.

Example of insecure code:

```
MessageDigest md = MessageDigest.getInstance("SHA-1");
```

Example of secure code:

```
MessageDigest md = MessageDigest.getInstance("SHA-256");
```

Details:
There is **'getInstance("MD5")'** found in file 'com/RNFetchBlob/RNFetchBlobUtils.java':

```
line 23:         try {
line 24:             MessageDigest instance = MessageDigest.getInstance("MD5");
line 25:             instance.update(str.getBytes());
```

CVSSv3 Base Score:
5.5 (AV:L/AC:L/PR:L/UI:N/S:U/C:H/I:N/A:N)

Reference:
- https://developer.android.com/reference/java/security/MessageDigest.html

OBJECT DESERIALIZATION FOUND [M7] [CWE-502] [SAST]

WARNING

Description:
Object deserialization performed on an untrusted resource (e.g. user-supplied input or external storage), can be dangerous if the data for deserialization is tampered by an attacker.

Example of insecure code:

```
bundle.putSerializable("exampleClass", exampleOfSerializabledClass);
exampleOfSerializabledClass = bundle.getSerializable("exampleClass");
```

Example of secure code:

```
// Use only serialization when you have the control over data
```

Details:
There is **'implements Serializable'** found in file 'ch/qos/logback/classic/Level.java':

```
line 5:
line 6: public final class Level implements Serializable {
line 7:     public static final Level ALL = new Level(Integer.MIN_VALUE,
SqlExpression.SqlShowTypeAll);
```

There is **'implements Serializable'** found in file 'ch/qos/logback/classic/spi/ClassPackagingData.java':

```
line 4:
line 5: public class ClassPackagingData implements Serializable {
line 6:     private static final long serialVersionUID = -8046432812183370001L;
```

There is **'implements Serializable'** found in file 'ch/qos/logback/classic/spi/StackTraceElementProxy.java':

```
line 4:
line 5: public class StackTraceElementProxy implements Serializable {
line 6:     private static final long serialVersionUID = -2374374378980555982L;
```

There is **'implements Serializable'** found in file 'ch/qos/logback/classic/spi/LoggerRemoteView.java':

```
line 5:
line 6: public class LoggerRemoteView implements Serializable {
line 7:     static final /* synthetic */ boolean $assertionsDisabled = false;
```

There is **'implements Serializable'** found in file 'ch/qos/logback/classic/spi/LoggerContextVO.java':

```
line 7:
line 8: public class LoggerContextVO implements Serializable {
line 9:     private static final long serialVersionUID = 5488023392483144387L;
```

There is **'implements Serializable'** found in file 'kotlin/Pair.java':

```
line 8: /* compiled from: Tuples.kt */
line 9: public final class Pair<A, B> implements Serializable {
line 10:     private final A first;
```

There is **'implements Serializable'** found in file 'kotlin/Triple.java':

```
line 8: /* compiled from: Tuples.kt */
line 9: public final class Triple<A, B, C> implements Serializable {
line 10:     private final A first;
```

There is **'implements Serializable'** found in file 'kotlin/Result.java':

```
line 13: /* compiled from: Result.kt */
line 14: public final class Result<T> implements Serializable {
line 15:     public static final Companion Companion = new Companion();
```

```
line 41:     /* compiled from: Result.kt */
line 42:     public static final class Failure implements Serializable {
line 43:         @NotNull
```

There is **'implements Serializable'** found in file 'kotlin/jvm/internal/Ref.java':

```
line 6:
line 7:     public static final class BooleanRef implements Serializable {
line 8:         public boolean element;
```

```
line 14:
line 15:     public static final class ByteRef implements Serializable {
line 16:         public byte element;
```

```
line 22:
line 23:     public static final class CharRef implements Serializable {
line 24:         public char element;
```

```
line 30:
line 31:     public static final class DoubleRef implements Serializable {
line 32:         public double element;
```

```
line 38:
line 39:     public static final class FloatRef implements Serializable {
line 40:         public float element;
```

```
line 46:
line 47:     public static final class IntRef implements Serializable {
line 48:         public int element;
```

```
line 54:
line 55:     public static final class LongRef implements Serializable {
line 56:         public long element;
```

```
line 62:
line 63:     public static final class ObjectRef<T> implements Serializable {
line 64:         public T element;
```

```
line 70:
line 71:     public static final class ShortRef implements Serializable {
line 72:         public short element;
```

There is **'implements Serializable'** found in file 'kotlin/jvm/internal/CallableReference.java':

```
line 24:     @SinceKotlin(version = "1.2")
line 25:     private static class NoReceiver implements Serializable {
line 26:         private static final NoReceiver INSTANCE = new NoReceiver();
```

There is **'implements Serializable'** found in file 'kotlin/coroutines/CombinedContext.java':

```
line 28:     /* compiled from: CoroutineContextImpl.kt */
line 29:     private static final class Serialized implements Serializable {
line 30:         public static final Companion Companion = new Companion();
```

There is **'implements Serializable'** found in file 'kotlin/text/Regex.java':

```
line 23: /* compiled from: Regex.kt */
line 24: public final class Regex implements Serializable {
line 25:     public static final Companion Companion = new Companion();
```

```
line 67:     /* compiled from: Regex.kt */
line 68:     private static final class Serialized implements Serializable {
line 69:         public static final Companion Companion = new Companion();
```

There is **'implements Serializable'** found in file 'com/marianhello/bgloc/data/ArrayListLocationTemplate.java':

```
line 9:
line 10: public class ArrayListLocationTemplate extends AbstractLocationTemplate
implements Serializable {
line 11:     private static final long serialVersionUID = 1234;
```

There is **'implements Serializable'** found in file 'com/marianhello/bgloc/data/HashMapLocationTemplate.java':

```
line 10:
line 11: public class HashMapLocationTemplate extends AbstractLocationTemplate
implements Serializable {
line 12:     private static final long serialVersionUID = 1234;
```

There is **'getSerializable()'** found in file 'com/marianhello/bgloc/Config.java':

```
line 118:         Bundle readBundle = parcel.readBundle();
line 119:         setHttpHeaders((HashMap) readBundle.getSerializable("httpHeaders"));
line 120:         setTemplate((LocationTemplate)
readBundle.getSerializable("template"));
line 121:     }
```

Reference:

- https://developer.android.com/reference/android/os/Bundle.html

MISSING ANTI-EMULATION [SAST]

WARNING

Description:

The mobile application does not use any anti-emulation or anti-debugger techniques (e.g. detecting rooted devices or checking if contacts are authentic).

This can significantly facilitate application debugging and reverse-engineering processes.

Reference:

- https://github.com/strazzere/anti-emulator

NETWORK SECURITY CONFIGURATION IS NOT PRESENT [SAST]

WARNING

Description:
The mobile application does not use Network Security Configuration to define which certificates and Certificate Authorities (CA) can be used for different environments (e.g. Development, Test and Production). The Network Security Configuration on Android feature lets application developers customize their network security settings in a safe, declarative configuration file without modifying the application code.

Reference:

- https://developer.android.com/training/articles/security-config.html

Mobile Application Behaviour

Mobile Application Functionality

The mobile application uses the following functionality that can endanger user's privacy under certain circumstances:

Storage

The mobile application can access external storage (e.g. SD card) in a write or read mode.

Location

The mobile application has an access to geographical location of the mobile phone.

Mobile Application Outgoing Traffic

The following HTTP/S requests were sent by mobile application without interaction with user:

`POST https://android.googleapis.com HTTP/1.1`

Date/Time: 10.05.2020 11:03 CEST
Requested URL: https://android.googleapis.com/auth/devicekey

`POST https://cloudconfig.googleapis.com HTTP/1.1`

Date/Time: 10.05.2020 15:44 CEST
Requested URL: https://cloudconfig.googleapis.com/config

`POST https://www.googleapis.com HTTP/1.1`

Date/Time: 10.05.2020 15:44 CEST
Requested URL: https://www.googleapis.com/experimentsandconfigs/v1/getExperimentsAndConfigs

POST https://www.googleapis.com HTTP/1.1

Date/Time: 10.05.2020 15:44 CEST
Requested URL: https://www.googleapis.com/placesandroid/v1/getPlaceInferenceModelWeights?key=AIzaSyAP-gfH3qvi6vgHZbSYwQ_XHqV_mXHhzIk

POST https://www.googleapis.com HTTP/1.1

Date/Time: 10.05.2020 15:44 CEST
Requested URL: https://www.googleapis.com/usercontext/v1/controllerhub/getconfig?key=AIzaSyAP-gfH3qvi6vgHZbSYwQ_XHqV_mXHhzIk

POST https://www.googleapis.com HTTP/1.1

Date/Time: 10.05.2020 15:44 CEST
Requested URL: https://www.googleapis.com/androidantiabuse/v1/x/create?alt=PROTO&key=AIzaSyBofcZsgLSS7BOnBjZPEkk4rYwzOIz-lTI

POST https://www.googleapis.com HTTP/1.1

Date/Time: 10.05.2020 15:44 CEST
Requested URL: https://www.googleapis.com/androidantiabuse/v1/x/create?alt=PROTO&key=AIzaSyBofcZsgLSS7BOnBjZPEkk4rYwzOIz-lTI

POST https://www.googleapis.com HTTP/1.1

Date/Time: 10.05.2020 15:44 CEST
Requested URL: https://www.googleapis.com/experimentsandconfigs/v1/getExperimentsAndConfigs

POST https://www.googleapis.com HTTP/1.1

Date/Time: 10.05.2020 15:44 CEST
Requested URL: https://www.googleapis.com/androidantiabuse/v1/x/create?alt=PROTO&key=AIzaSyBofcZsgLSS7BOnBjZPEkk4rYwzOIz-lTI

POST https://www.googleapis.com HTTP/1.1

Date/Time: 10.05.2020 15:44 CEST
Requested URL: https://www.googleapis.com/affiliation/v1/affiliation:lookup?alt=proto&key=AIzaSyAP-gfH3qvi6vgHZbSYwQ_XHqV_mXHhzIk

POST https://android.googleapis.com HTTP/1.1

Date/Time: 10.05.2020 15:45 CEST
Requested URL: https://android.googleapis.com/checkin

POST https://www.googleapis.com HTTP/1.1

Date/Time: 10.05.2020 15:44 CEST
Requested URL: https://www.googleapis.com/androidantiabuse/v1/x/create?alt=PROTO&key=AIzaSyBofcZsgLSS7BOnBjZPEkk4rYwzOIz-lTI

POST https://www.googleapis.com HTTP/1.1

Date/Time: 10.05.2020 15:44 CEST
Requested URL: https://www.googleapis.com/androidantiabuse/v1/x/create?alt=PROTO&key=AIzaSyBofcZsgLSS7BOnBjZPEkk4rYwzOIz-lTI

POST https://cloudconfig.googleapis.com HTTP/1.1

Date/Time: 10.05.2020 15:45 CEST
Requested URL: https://cloudconfig.googleapis.com/config

POST https://mobilenetworkscoring-pa.googleapis.com HTTP/1.1

Date/Time: 10.05.2020 15:45 CEST
Requested URL: https://mobilenetworkscoring-pa.googleapis.com/v1/GetWifiQuality?key=AIzaSyBrlVtd67QvwS_Wz0Do4ZIdOeA6ThtVczU

POST https://www.googleapis.com HTTP/1.1

Date/Time: 10.05.2020 15:45 CEST
Requested URL: https://www.googleapis.com/affiliation/v1/affiliation:lookup?alt=proto&key=AIzaSyAP-gfH3qvi6vgHZbSYwQ_XHqV_mXHhzIk

GET https://pagead2.googlesyndication.com HTTP/1.1

Date/Time: 10.05.2020 15:45 CEST
Requested URL: https://pagead2.googlesyndication.com/pagead/gen_204?id=gmob-apps&device=Google%20Android%20SDK%20built%20for%20x86&js=16089022.16089000&os=8.1.0&api=27&lat=0&eids=318475417&tslu=11&appid=com.google.android.gms

GET https://pagead2.googlesyndication.com HTTP/1.1

Date/Time: 10.05.2020 15:45 CEST
Requested URL: https://pagead2.googlesyndication.com/pagead/gen_204?id=gmob-apps&device=Google%20Android%20SDK%20built%20for%20x86&js=16089022.16089000&os=8.1.0&api=27&lat=0&eids=318475417&tslu=-1&appid=com.google.android.gms

POST https://www.googleapis.com HTTP/1.1

Date/Time: 10.05.2020 15:45 CEST
Requested URL: https://www.googleapis.com/androidantiabuse/v1/x/create?alt=PROTO&key=AIzaSyBofcZsgLSS7BOnBjZPEkk4rYwzOIz-lTI

POST https://cloudconfig.googleapis.com HTTP/1.1

Date/Time: 10.05.2020 15:45 CEST
Requested URL: https://cloudconfig.googleapis.com/config

GET https://www.googleapis.com HTTP/1.1

Date/Time: 10.05.2020 15:45 CEST
Requested URL: https://www.googleapis.com/android_video/v1/config?make=Google&model=Android%20SDK%20built%20for%20x86&product=sdk_gphone_x86&device=generic_x86&devid=3749234000110016402&apptype=1&devtype=1&fs=f

POST https://www.googleapis.com HTTP/1.1

Date/Time: 10.05.2020 15:45 CEST
Requested URL: https://www.googleapis.com/androidcheck/v1/attestations/adAttest?key=AIzaSyDaepk5bynjTA7ZhzF_0fzIHIXAkZlz3dA

GET https://www.googleadservices.com HTTP/1.1

Date/Time: 10.05.2020 15:45 CEST
Requested URL: https://www.googleadservices.com/pagead/conversion/1001680686/?bundleid=com.google.android.youtube&appversion=12.17.41&osversion=8.1.0&sdkversion=ct-sdk-a-v2.2.4&gms=1&lat=0&rdid=41e8270e-9180-4d11-a1ba-f902e91a89bb×tamp=1589118307.855&remarketing_only=1&usage_tracking_enabled=0&data.screen_name=%3CAndroid_YT_Open_App%3E

POST https://www.googleapis.com HTTP/1.1

Date/Time: 10.05.2020 15:45 CEST
Requested URL: https://www.googleapis.com/androidcheck/v1/attestations/adAttest?key=AIzaSyDaepk5bynjTA7ZhzF_0fzIHIXAkZlz3dA

GET https://www.youtube.com HTTP/1.1

Date/Time: 10.05.2020 15:45 CEST
Requested URL: https://www.youtube.com/csi_204?v=3&s=youtube_android&action=process&yt_lt=frozen&mod_li=0&conn=3&it=ndps.555,proc_k.-215,app_l.1010,f_proc.1125&cplatform=mobile&cbr=com.google.android.youtube&c=android&cmodel=Android%20SDK%20built%20for%20x86&cos=Android&csdk=27&cbrver=12.17.41&cver=12.17.41&cosver=8.1.0&cbrand=Google&proc=2

POST https://youtubei.googleapis.com HTTP/1.1

Date/Time: 10.05.2020 15:45 CEST
Requested URL: https://youtubei.googleapis.com/deviceregistration/v1/devices?key=AIzaSyA8eiZmM1FaDVjRy-df2KTyQ_vz_yYM39w&rawDeviceId=e9dfa67a35b2b572

POST https://www.googleapis.com HTTP/1.1

Date/Time: 10.05.2020 15:45 CEST
Requested URL: https://www.googleapis.com/androidantiabuse/v1/x/create?alt=PROTO&key=AIzaSyBofcZsgLSS7BOnBjZPEkk4rYwzOIz-lTI

POST https://www.googleapis.com HTTP/1.1

Date/Time: 10.05.2020 15:45 CEST
Requested URL: https://www.googleapis.com/experimentsandconfigs/v1/getExperimentsAndConfigs?r=6

POST https://www.googleapis.com HTTP/1.1

Date/Time: 10.05.2020 15:45 CEST
Requested URL: https://www.googleapis.com/androidcheck/v1/attestations/adAttest?key=AIzaSyDaepk5bynjTA7ZhzF_0fzIHIXAkZlz3dA

POST https://android.googleapis.com HTTP/1.1

Date/Time: 10.05.2020 15:45 CEST
Requested URL: https://android.googleapis.com/checkin

POST https://www.googleapis.com HTTP/1.1

Date/Time: 10.05.2020 15:45 CEST
Requested URL: https://www.googleapis.com/androidantiabuse/v1/x/create?alt=PROTO&key=AIzaSyBofcZsgLSS7BOnBjZPEkk4rYwzOIz-lTI

POST https://www.googleapis.com HTTP/1.1

Date/Time: 10.05.2020 15:45 CEST
Requested URL: https://www.googleapis.com/androidcheck/v1/attestations/adAttest?key=AIzaSyDaepk5bynjTA7ZhzF_0fzIHIXAkZlz3dA

POST https://www.googleapis.com HTTP/1.1

Date/Time: 10.05.2020 15:45 CEST
Requested URL: https://www.googleapis.com/androidantiabuse/v1/x/create?alt=PROTO&key=AIzaSyBofcZsgLSS7BOnBjZPEkk4rYwzOIz-lTI

POST https://www.googleapis.com HTTP/1.1

Date/Time: 10.05.2020 15:45 CEST
Requested URL: https://www.googleapis.com/youtubei/v1/browse?key=AIzaSyA8eiZmM1FaDVjRy-df2KTyQ_vz_yYM39w

POST https://android.googleapis.com HTTP/1.1

Date/Time: 10.05.2020 15:45 CEST
Requested URL: https://android.googleapis.com/auth/devicekey

POST https://www.googleapis.com HTTP/1.1

Date/Time: 10.05.2020 15:45 CEST
Requested URL: https://www.googleapis.com/experimentsandconfigs/v1/getExperimentsAndConfigs?r=8

GET https://www.youtube.com HTTP/1.1

Date/Time: 10.05.2020 15:45 CEST
Requested URL: https://www.youtube.com/gen_204?cplatform=mobile&cbr=com.google.android.youtube&c=android&cmodel=Android%20SDK%20built%20for%20x86&cos=Android&csdk=27&cbrver=12.17.41&cver=12.17.41&cosver=8.1.0&cbrand=Google&a=delayed_request&batch_size=100&max_queue_size=1000&max_age_hours=720&age_of_oldest_request_hours=0¤t_queue_size=0&peak_queue_size=0&total_enqueued_requests=0&total_successful_requests=0&total_server_http_errors=0&total_client_http_errors=0&report_cap_hours=24&total_dropped_requests=0

POST https://android.googleapis.com HTTP/1.1

Date/Time: 10.05.2020 15:46 CEST
Requested URL: https://android.googleapis.com/auth/devicekey

Software Composition Analysis

The mobile application uses the following external and native libraries:

External

- brut.util
- brut.common
- brut.androlib
- net.lingala
- org.devio

Android Native

- androidx.transition
- androidx.asynclayoutinflater
- androidx.collection
- androidx.core
- androidx.drawerlayout

- org.webkit
- org.chromium
- org.reactnative
- ch.qos
- io.github
- ru.andremoniy
- me.leolin
- cl.json
- com.dieam
- com.vitorpamplona
- com.mindprod
- com.th3rdwave
- com.iodine
- com.horcrux
- com.BV
- com.transistorsoft
- com.reactnativecommunity
- com.intentfilter
- com.swmansion
- com.rnfs
- com.ocetnik
- com.RNFetchBlob
- com.marianhello
- com.rnziparchive
- com.tenforwardconsulting
- androidx.interpolator
- androidx.versionedparcelable
- androidx.activity
- androidx.viewpager
- androidx.vectordrawable
- androidx.lifecycle
- androidx.legacy
- androidx.documentfile
- androidx.annotation
- androidx.print
- androidx.localbroadcastmanager
- androidx.cursoradapter
- androidx.swiperefreshlayout
- androidx.fragment
- androidx.loader
- androidx.recyclerview
- androidx.arch
- androidx.appcompat
- androidx.customview
- androidx.slidingpanelayout
- androidx.coordinatorlayout
- androidx.savedstate
- androidx.cardview
- androidx.media
- bolts
- kotlin
- javax.inject
- javax.annotation
- android.support

Mobile App External Communications

Static mobile application security test revealed the following remote hosts where the mobile application may send or receive data:

Hostname	IP:Port	SSL Encryption	Websec Server Security	Domain Domain Security
logback.qos.ch:80	83.166.144.67:80	N	C+	No risks found
mindprod.com:80	65.110.21.43:80	N	N/A	10 malicious websites found
www.facebook.com:443	31.13.80.36:443	B-	A	4010 malicious websites found
twitter.com:443	104.244.42.129:443	A+	A	1341 malicious websites found

Hostname	IP:Port	SSL Encryption	Websec Server Security	Domain Domain Security
pinterest.com:443	151.101.0.84:443	A	N/A	536 malicious websites found
www.youtube.com:443	172.217.13.174:443	A-	A	1543 malicious websites found