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
GreenCheck

APP ID
at.itsv.mobile.cochap

APP VERSION
1.18

DEVICE TYPE
Android

TEST STARTED
May 12th 2022, 12:20

TEST FINISHED
May 12th 2022, 12:55

Mobile App Permissions and Privacy	OWASP Mobile Top 10 Security Test	Mobile App External Communications	Software Composition Analysis
8 PERMISSIONS	3 MAJOR RISKS FOUND	7 MAJOR RISKS FOUND	30 COMPONENTS FOUND

Malware test: no malicious code or behavioral patterns detected in the mobile app.

Mobile Application Permissions and Privacy Test

If the application processes or stores any PII of EU residents, the following requirements of EU GDPR may apply:

Privacy Policy

Privacy Policy was not found in application — Misconfiguration or weakness

Mobile Application Functionality

The mobile application requests access to the following functionality that may endanger user's privacy under certain circumstances:

Camera

The mobile application can use the camera for taking pictures or videos.

Mobile Application Permissions

The mobile application requests the following permissions that may endanger user's privacy under certain circumstances:

CAMERA dangerous

Allows application to take pictures and videos with the camera. This allows the application to collect images that the camera is seeing at any time.

ACCESS_NETWORK_STATE normal

Allows an application to view the status of all networks.

ACCESS_WIFI_STATE normal

Allows an application to view the information about the status of Wi-Fi.

CHANGE_WIFI_MULTICAST_STATE normal

Allows an application to receive packets not directly addressed to your device. This can be useful when discovering services offered nearby. It uses more power than the non-multicast mode.

INTERNET normal

Allows an application to create network sockets.

USE_BIOMETRIC normal

Allows an app to use device supported biometric modalities.

USE_FINGERPRINT normal

This constant was deprecated in API level 28. Applications should request USE_BIOMETRIC instead.

VIBRATE normal

Allows the application to control the vibrator.

OWASP Mobile Top 10 Security Test

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

HIGH RISKS	MEDIUM RISK	LOW RISKS	WARNINGS

2	1	3	7

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

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

HIGH

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 **'rawQuery()'** found in file 'com/reactnativecommunity/asyncstorage/AsyncLocalStorageUtil.java':

```
[line 95:             ReactDatabaseSupplier instance =
ReactDatabaseSupplier.getInstance(context);]
[line 96:             instance.get().rawQuery("PRAGMA wal_checkpoint", null).close();]
[line 97:             instance.closeDatabase();]
```

CVSSv3 Base Score:

7.3 (CVSS:3.1/AV:N/AC:L/PR:N/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

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

HIGH

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/koushikdutta/async/util/FileCache.java':

```
[line 112:         try {]
[line 113:             messageDigest = MessageDigest.getInstance("MD5");]
[line 114:         } catch (NoSuchAlgorithmException e) {]
```

There is **'getInstance("MD5")'** found in file 'com/koushikdutta/async/http/spdy/ByteString.java':

```
[line 241:         try {]
[line 242:             return String.format("ByteString[size=%s md5=%s]", new Object[]
{Integer.valueOf(bArr.length),
of(MessageDigest.getInstance("MD5").digest(this.data)).hex()});]
[line 243:         } catch (NoSuchAlgorithmException unused) {]
```

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

WEAK ENCRYPTION [M5] [CWE-327] [SAST]

MEDIUM

Description:

Weak or badly implemented encryption algorithms can endanger data storage and transmission used by the mobile application.

Example of insecure code:

```
Cipher c = Cipher.getInstance("AES/ECB/NoPadding");
c.init(Cipher.ENCRYPT_MODE, k, iv);
byte[] cipherText = c.doFinal(plainText);
```

Example of secure code:

```
Cipher c = Cipher.getInstance("AES/GCM/NoPadding");
c.init(Cipher.ENCRYPT_MODE, k, iv);
byte[] cipherText = c.doFinal(plainText);
```

Details:

There is **'RSA/ECB/PKCS1Padding'** found in file 'com/taluttasgiran/rnsecurestorage/RNKeyStore.java':

```
[line 51:     private byte[] encryptRsaPlainText(PublicKey publicKey, byte[] bArr)
throws GeneralSecurityException, IOException {]
[line 52:         Cipher instance = Cipher.getInstance("RSA/ECB/PKCS1Padding");]
[line 53:         instance.init(1, publicKey);]
```

```
[line 108:     private byte[] decryptRsaCipherText(PrivateKey privateKey, byte[] bArr)
throws GeneralSecurityException, IOException {]
[line 109:         Cipher instance = Cipher.getInstance("RSA/ECB/PKCS1Padding");]
[line 110:         instance.init(2, privateKey);]
```

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/javax/crypto/Cipher.html
- http://find-sec-bugs.github.io/bugs.htm

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://pwall.net/schema/schema.json in following files:

- **net/pwall/json/validation/JSONValidation.java:**

```
[line 401: URI.create("http://pwall.net/schema/schema.json").resolve(str);]
```

http:// with value http://example.com/missing in following files:

- **ehn/techiop/hcert/kotlin/data/ValueSetHolder.java:**

```
[line 86: return new ValueSetEntryAdapter(str, new ValueSetEntry(str, "en",
false, "http://example.com/missing", "0", str2));]
```

http:// with value http://purl.org/dc/elements/1.1/ in following files:

- **com/adobe/xmp/XMPConst.java:**

```
[line 11: public static final String NS_DC =
"http://purl.org/dc/elements/1.1/";]
```

- **com/adobe/xmp/impl/Utils.java:**

```
[line 100: if ("http://purl.org/dc/elements/1.1/".equals(str)) {]
```

- **com/adobe/xmp/impl/XMPSchemaRegistryImpl.java:**

```
[line 32: registerAlias("http://ns.adobe.com/xap/1.0/", "Author",
"http://purl.org/dc/elements/1.1/", "creator", arrayOrdered);]
[line 33: registerAlias("http://ns.adobe.com/xap/1.0/", "Authors",
```

```
"http://purl.org/dc/elements/1.1/", "creator", null);]
[line 34: registerAlias("http://ns.adobe.com/xap/1.0/", "Description",
"http://purl.org/dc/elements/1.1/", AuthPromptOptions.DESCRIPTION, null);]
[line 35: registerAlias("http://ns.adobe.com/xap/1.0/", "Format",
"http://purl.org/dc/elements/1.1/", "format", null);]
[line 36: registerAlias("http://ns.adobe.com/xap/1.0/", "Keywords",
"http://purl.org/dc/elements/1.1/", "subject", null);]
[line 37: registerAlias("http://ns.adobe.com/xap/1.0/", "Locale",
"http://purl.org/dc/elements/1.1/", "language", null);]
[line 38: registerAlias("http://ns.adobe.com/xap/1.0/", "Title",
"http://purl.org/dc/elements/1.1/", "title", null);]
[line 39: registerAlias(XMPConst.NS_XMP_RIGHTS, ExifInterface.TAG_COPYRIGHT,
"http://purl.org/dc/elements/1.1/", "rights", null);]
[line 40: registerAlias(XMPConst.NS_PDF, "Author",
"http://purl.org/dc/elements/1.1/", "creator", arrayOrdered);]
[line 46: registerAlias(XMPConst.NS_PDF, "Subject",
"http://purl.org/dc/elements/1.1/", AuthPromptOptions.DESCRIPTION,
aliasOptions);]
[line 47: registerAlias(XMPConst.NS_PDF, "Title",
"http://purl.org/dc/elements/1.1/", "title", aliasOptions);]
[line 48: registerAlias(XMPConst.NS_PHOTOSHOP, "Author",
"http://purl.org/dc/elements/1.1/", "creator", arrayOrdered);]
[line 49: registerAlias(XMPConst.NS_PHOTOSHOP, "Caption",
"http://purl.org/dc/elements/1.1/", AuthPromptOptions.DESCRIPTION,
aliasOptions);]
[line 50: registerAlias(XMPConst.NS_PHOTOSHOP, ExifInterface.TAG_COPYRIGHT,
"http://purl.org/dc/elements/1.1/", "rights", aliasOptions);]
[line 51: registerAlias(XMPConst.NS_PHOTOSHOP, "Keywords",
"http://purl.org/dc/elements/1.1/", "subject", null);]
[line 53: registerAlias(XMPConst.NS_PHOTOSHOP, "Title",
"http://purl.org/dc/elements/1.1/", "title", arrayAltText);]
[line 55: registerAlias("http://ns.adobe.com/tiff/1.0/",
ExifInterface.TAG_ARTIST, "http://purl.org/dc/elements/1.1/", "creator",
arrayOrdered);]
[line 56: registerAlias("http://ns.adobe.com/tiff/1.0/",
ExifInterface.TAG_COPYRIGHT, "http://purl.org/dc/elements/1.1/", "rights",
null);]
[line 58: registerAlias("http://ns.adobe.com/tiff/1.0/",
ExifInterface.TAG_IMAGE_DESCRIPTION, "http://purl.org/dc/elements/1.1/",
AuthPromptOptions.DESCRIPTION, null);]
[line 60: registerAlias(XMPConst.NS_PNG, "Author",
"http://purl.org/dc/elements/1.1/", "creator", arrayOrdered);]
[line 61: registerAlias(XMPConst.NS_PNG, ExifInterface.TAG_COPYRIGHT,
"http://purl.org/dc/elements/1.1/", "rights", arrayAltText);]
[line 63: registerAlias(XMPConst.NS_PNG, "Description",
"http://purl.org/dc/elements/1.1/", AuthPromptOptions.DESCRIPTION,
arrayAltText);]
[line 66: registerAlias(XMPConst.NS_PNG, "Title",
"http://purl.org/dc/elements/1.1/", "title", arrayAltText);]
[line 71: registerNamespace("http://purl.org/dc/elements/1.1/", "dc");]
```

- **com/adobe/xmp/impl/ParseRDF.java:**

```
[line 42: namespaceURI = "http://purl.org/dc/elements/1.1/";]
```

- **com/adobe/xmp/impl/XMPNormalizer.java:**

```
[line 101: XMPNode findSchemaNode =
XMPNodeUtils.findSchemaNode(((XMPMetaImpl) xMPMeta).getRoot(),
"http://purl.org/dc/elements/1.1/", true);]
[line 110: xMPMeta2.setLocalizedText("http://purl.org/dc/elements/1.1/",
"rights", "", XMPConst.X_DEFAULT, findSchemaNode.getChild(1).getValue(),
null);]
[line 142: xMPMeta3.setLocalizedText("http://purl.org/dc/elements/1.1/",
"rights", "", XMPConst.X_DEFAULT, stringBuilder3.toString(), null);]
[line 258: String str = "http://purl.org/dc/elements/1.1/";]
```

- **com/drew/metadata/Schema.java:**

```
[line 3: public static final String DUBLIN_CORE_SPECIFIC_PROPERTIES =
"http://purl.org/dc/elements/1.1/";]
```

http:// with value http://purl.org/dc/1.1/ in following files:

- **com/adobe/xmp/XMPConst.java:**

```
[line 12: public static final String NS_DC_DEPRECATED =
"http://purl.org/dc/1.1/";]
```

http:// with value http://cipa.jp/exif/1.0/ in following files:

- **com/adobe/xmp/XMPConst.java:**

```
[line 16: public static final String NS_EXIFX = "http://cipa.jp/exif/1.0/";]
```

http:// with value http://iptc.org/std/Iptc4xmpCore/1.0/xmlns/ in following files:

- **com/adobe/xmp/XMPConst.java:**

```
[line 18: public static final String NS_IPTCCORE =
"http://iptc.org/std/Iptc4xmpCore/1.0/xmlns/";]
```

http:// with value http://iptc.org/std/Iptc4xmpExt/2008-02-29/ in following files:

- **com/adobe/xmp/XMPConst.java:**

```
[line 19: public static final String NS_IPTCEXT =
"http://iptc.org/std/Iptc4xmpExt/2008-02-29/";]
```

http:// with value http://www.aiim.org/pdfa/ns/extension/ in following files:

- **com/adobe/xmp/XMPConst.java:**

```
[line 24: public static final String NS_PDFA_EXTENSION =
"http://www.aiim.org/pdfa/ns/extension/";]
```

http:// with value http://www.aiim.org/pdfa/ns/field# in following files:

- **com/adobe/xmp/XMPConst.java:**

```
[line 25: public static final String NS_PDFA_FIELD =
"http://www.aiim.org/pdfa/ns/field#";]
```

http:// with value http://www.aiim.org/pdfa/ns/id/ in following files:

- **com/adobe/xmp/XMPConst.java:**

```
[line 26: public static final String NS_PDFA_ID =
"http://www.aiim.org/pdfa/ns/id/";]
```

http:// with value http://www.aiim.org/pdfa/ns/property# in following files:

- **com/adobe/xmp/XMPConst.java:**

```
[line 27: public static final String NS_PDFA_PROPERTY =
"http://www.aiim.org/pdfa/ns/property#";]
```

http:// with value http://www.aiim.org/pdfa/ns/schema# in following files:

- **com/adobe/xmp/XMPConst.java:**

```
[line 28: public static final String NS_PDFA_SCHEMA =
"http://www.aiim.org/pdfa/ns/schema#";]
```

http:// with value http://www.aiim.org/pdfa/ns/type# in following files:

- **com/adobe/xmp/XMPConst.java:**

```
[line 29: public static final String NS_PDFA_TYPE =
"http://www.aiim.org/pdfa/ns/type#";]
```

http:// with value http://www.npes.org/pdfx/ns/id/ in following files:

- **com/adobe/xmp/XMPConst.java:**

```
[line 31: public static final String NS_PDFX_ID =
"http://www.npes.org/pdfx/ns/id/";]
```

http:// with value http://ns.useplus.org/ldf/xmp/1.0/ in following files:

- **com/adobe/xmp/XMPConst.java:**

```
[line 33: public static final String NS_PLUS =
"http://ns.useplus.org/ldf/xmp/1.0/";]
```

http:// with value http://javax.xml.XMLConstants/feature/secure-processing in following files:

- **com/adobe/xmp/impl/XMPMetaParser.java:**

```
[line 28:
newInstance.setFeature("http://javax.xml.XMLConstants/feature/secure-
processing", true);]
```

dummy:// with value dummy:/// in following files:

- **ehn/techiop/hcert/kotlin/chain/impl/JvmSchemaLoader.java:**

```
[line 65: URI create = URI.create("dummy:///");]
[line 66: Intrinsics.checkNotNullExpressionValue(create,
"create(\"dummy:///\")");]
```

https:// with value https://greencheck.gv.at/ in following files:

- **at/itsv/mobile/cochap/BuildConfig.java:**

```
[line 8: public static final String baseUrl = "https://greencheck.gv.at/";]
```

https:// with value https://github.com/software-mansion/react-native-screens/issues/17#issuecomment-424704067 in following files:

- **com/swmansion/rnscreens/ScreenStackFragment.java:**

```
[line 57: throw new IllegalStateException("ScreenStack fragments should
never be restored. Follow instructions from https://github.com/software-
mansion/react-native-screens/issues/17#issuecomment-424704067 to properly
configure your main activity.");]
```

- **com/swmansion/rnscreens/ScreenFragment.java:**

```
[line 74: throw new IllegalStateException("Screen fragments should never be
restored. Follow instructions from https://github.com/software-
mansion/react-native-screens/issues/17#issuecomment-424704067 to properly
configure your main activity.");]
```

https:// with value https://drewnoakes.com/code/exif/ in following files:

- **com/drew/tools/ProcessAllImagesInFolderUtility.java:**

```
[line 672: writer.write("https://drewnoakes.com/code/exif/\n");]
```

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/mtrl_alert_dialog_actions.xml
- 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-land/mtrl_picker_header_dialog.xml
- android/res/layout/text_view_with_theme_line_height.xml
- android/res/layout/mtrl_picker_text_input_date.xml
- android/res/layout/mtrl_alert_dialog_actions.xml
- android/res/layout/abc_list_menu_item_checkbox.xml
- android/res/layout/design_layout_snackbar_include.xml
- android/res/layout/design_text_input_start_icon.xml
- android/res/layout/abc_alert_dialog_button_bar_material.xml
- android/res/layout/redbox_item_frame.xml
- android/res/layout/mtrl_picker_header_fullscreen.xml
- android/res/layout/abc_activity_chooser_view_list_item.xml
- android/res/layout/notification_template_big_media.xml
- android/res/layout/text_view_with_line_height_from_layout.xml
- android/res/layout/design_layout_tab_icon.xml
- android/res/layout/abc_screen_content_include.xml
- android/res/layout/mtrl_picker_dialog.xml
- android/res/layout/abc_activity_chooser_view.xml
- android/res/layout/abc_select_dialog_material.xml
- android/res/layout/test_toolbar_elevation.xml
- android/res/layout/abc_action_bar_title_item.xml
- android/res/layout/mtrl_calendar_year.xml
- android/res/layout/mtrl_alert_dialog_title.xml
- android/res/layout/test_toolbar.xml
- android/res/layout/abc_screen_toolbar.xml
- android/res/layout/text_view_with_line_height_from_appearance.xml
- android/res/layout/abc_dialog_title_material.xml
- android/res/layout/mtrl_picker_header_dialog.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/mtrl_calendar_months.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_text_input_end_icon.xml
- android/res/layout/design_navigation_item_separator.xml
- android/res/layout/mtrl_calendar_day_of_week.xml
- android/res/layout/mtrl_picker_header_toggle.xml
- android/res/layout/mtrl_calendar_month_navigation.xml
- android/res/layout/test_reflow_chipgroup.xml
- android/res/layout/notification_media_cancel_action.xml
- android/res/layout/mtrl_calendar_month.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/texture_view.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/mtrl_calendar_horizontal.xml
- android/res/layout/mtrl_alert_select_dialog_multichoice.xml
- android/res/layout/test_toolbar_custom_background.xml
- android/res/layout/abc_screen_simple_overlay_action_mode.xml
- android/res/layout/mtrl_calendar_vertical.xml
- android/res/layout/abc_popup_menu_item_layout.xml
- android/res/layout/mtrl_picker_header_title_text.xml
- android/res/layout/mtrl_picker_fullscreen.xml
- android/res/layout/design_menu_item_action_area.xml
- android/res/layout/select_dialog_singlechoice_material.xml
- android/res/layout/autofill_inline_suggestion.xml
- android/res/layout/select_dialog_item_material.xml
- android/res/layout/design_navigation_menu.xml
- android/res/layout/mtrl_picker_header_selection_text.xml
- android/res/layout/test_action_chip.xml
- android/res/layout/surface_view.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/mtrl_alert_select_dialog_item.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/test_design_checkbox.xml
- android/res/layout/mtrl_picker_text_input_date_range.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/mtrl_calendar_month_labeled.xml
- android/res/layout/abc_action_mode_close_item_material.xml

- android/res/layout/test_toolbar_surface.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/mtrl_calendar_days_of_week.xml
- android/res/layout/text_view_without_line_height.xml
- android/res/layout/mtrl_calendar_day.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/text_view_with_line_height_from_style.xml
- android/res/layout/redbox_item_title.xml
- android/res/layout/fingerprint_dialog_layout.xml
- android/res/layout/mtrl_alert_select_dialog_singlechoice.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/mtrl_alert_dialog.xml
- android/res/layout/mtrl_picker_actions.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
- android/res/layout-v26/mtrl_calendar_month.xml

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

```
[line 20: ]
[line 21: class TSpanView extends TextView {]
[line 22:     private static final String FONTS = "fonts/";]
```

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

```
[line 10: ]
[line 11: class TextPathView extends TextView {]
[line 12:     private String mHref;]
```

There is **'extends ViewGroup'** found in file 'com/lwansbrough/RCTCamera/RCTCameraView.java':

```
[line 9: ]
[line 10: public class RCTCameraView extends ViewGroup {]
[line 11:     private int _actualDeviceOrientation = -1;]
```

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

```
[line 20: ]
[line 21: public class RNGestureHandlerButtonViewManager extends
ViewGroupManager<ButtonViewGroup> {]
[line 22: ]
[line 23:     static class ButtonViewGroup extends ViewGroup {]
[line 24:         public static final String SELECTABLE_ITEM_BACKGROUND =
"selectableItemBackground";]
```

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

```
[line 27: ]
[line 28: public class ScreenStackHeaderConfig extends ViewGroup {]
[line 29:     private boolean mBackButtonInCustomView;]
```

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

```
[line 17: ]
[line 18: public class Screen extends ViewGroup {]
[line 19:     private static final OnAttachStateChangeListener
sShowSoftKeyboardOnAttach = new OnAttachStateChangeListener() {]
```

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

```
[line 18: ]
[line 19: public class ScreenContainer<T extends ScreenFragment> extends ViewGroup {]
[line 20:     private FragmentTransaction mCurrentTransaction;]
```

There is **'extends TextureView'** found in file 'com/lwansbrough/RCTCamera/RCTCameraViewFinder.java':

```
[line 35: ]
[line 36: class RCTCameraViewFinder extends TextureView implements
SurfaceTextureListener, PreviewCallback {]
[line 37:     public static volatile boolean barcodeScannerTaskLock = false;]
```

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 CoordinatorLayout'** found in file 'com/swmansion/rnscreens/ScreenStackFragment.java':

```
[line 27: ]
[line 28:     private static class NotifyingCoordinatorLayout extends CoordinatorLayout
{]
[line 29:         private final AnimationListener mAnimationListener = new
AnimationListener() {]
```

CVSSv3 Base Score:

3.3 (AV:L/AC:H/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 21:         <activity
android:configChanges="keyboard|keyboardHidden|orientation|screenSize|uiMode"
android:exported="true" android:label="@string/app_name"
android:launchMode="singleTask" android:name="at.itsv.mobile.cochap.MainActivity"
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

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 31:         Intrinsics.checkNotNullParameter(str, "prefix");]
[line 32:         File createTempFile = File.createTempFile(str, str2, file);]
[line 33:         createTempFile.delete();]
```

```
[line 54:         }]
[line 55:         return createTempFile(str, str2, file);]
[line 56:     }]
```

```
[line 58:     @Deprecated(message = "Avoid creating temporary files in the default temp
location with this function due to too wide permissions on the newly created file. Use
kotlin.io.path.createTempFile instead or resort to java.io.File.createTempFile.")]
[line 59:     public static final File createTempFile(String str, String str2, File
file) {]
[line 60:         Intrinsics.checkNotNullParameter(str, "prefix");]
[line 61:         File createTempFile = File.createTempFile(str, str2, file);]
[line 62:         Intrinsics.checkNotNullExpressionValue(createTempFile,
"File.createTempFile(prefix, suffix, directory)");]
[line 63:         return createTempFile;]
```

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

```
[line 728:         }]
[line 729:         Path createTempFile = Files.createTempFile(str, str2,
(FileAttribute[]) Arrays.copyOf(fileAttributeArr, fileAttributeArr.length));]
[line 730:         Intrinsics.checkNotNullExpressionValue(createTempFile,
"Files.createTempFile(prefix, suffix, *attributes)");]
[line 731:         return createTempFile;]
```

```
[line 733: ]
[line 734:     private static final Path createTempFile(String str, String str2,
FileAttribute<?>... fileAttributeArr) throws IOException {]
[line 735:         Path createTempFile = Files.createTempFile(str, str2,
(FileAttribute[]) Arrays.copyOf(fileAttributeArr, fileAttributeArr.length));]
[line 736:         Intrinsics.checkNotNullExpressionValue(createTempFile,
"Files.createTempFile(prefix, suffix, *attributes)");]
[line 737:         return createTempFile;]
```

```
[line 746:         }]
[line 747:         return createTempFile(path, str, str2, fileAttributeArr);]
[line 748:     }]
[line 749: ]
[line 750:     public static final Path createTempFile(Path path, String str, String
str2, FileAttribute<?>... fileAttributeArr) throws IOException {]
[line 751:         Intrinsics.checkNotNullParameter(fileAttributeArr, "attributes");]
[line 752:         if (path != null) {]
[line 753:             path = Files.createTempFile(path, str, str2, (FileAttribute[])
Arrays.copyOf(fileAttributeArr, fileAttributeArr.length));]
[line 754:             Intrinsics.checkNotNullExpressionValue(path,
"Files.createTempFile(dir…fix, suffix, *attributes)");]
[line 755:             return path;]
[line 756:         }]
[line 757:         path = Files.createTempFile(str, str2, (FileAttribute[])
Arrays.copyOf(fileAttributeArr, fileAttributeArr.length));]
[line 758:         Intrinsics.checkNotNullExpressionValue(path,
"Files.createTempFile(prefix, suffix, *attributes)");]
[line 759:         return path;]
```

There is **'createTempFile()'** found in file 'com/lwansbrough/RCTCamera/RCTCameraModule.java':

```
[line 1123:                 stringBuilder.append(format);]
[line 1124:                 createTempFile =
File.createTempFile(stringBuilder.toString(), ".jpg", cacheDir);]
[line 1125:             } else if (i == 2) {]
```

```
[line 1128:                 stringBuilder.append(format);]
[line 1129:                 createTempFile =
File.createTempFile(stringBuilder.toString(), ".mp4", cacheDir);]
[line 1130:             } else {]
```

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 'com/zoontek/rnpermissions/RNPermissionsModule.java':

```
[line 180:            ReactApplicationContext reactApplicationContext =
getReactApplicationContext();]
[line 181:            Intent intent = new Intent();]
[line 182:            String packageName = reactApplicationContext.getPackageName();]
```

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 15:         <activity android:label="@string/app_name"
android:name="at.itsv.mobile.cochap.SplashActivity"
android:theme="@style/SplashTheme">]
[line 16:             <intent-filter>]
[line 17:                 <action android:name="android.intent.action.MAIN"/>]
```

Reference:

- https://developer.android.com/guide/components/intents-filters.html
- https://developer.android.com/training/articles/security-tips.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 'j$/time/zone/ZoneRulesProvider.java':

```
[line 57:                     try {]
[line 58:                         ZoneRulesProvider zoneRulesProvider =
(ZoneRulesProvider) cls.cast(Class.forName(property, true,
cls.getClassLoader()).newInstance());]
[line 59:
ZoneRulesProvider.registerProvider(zoneRulesProvider);]
```

There is **'ClassLoader'** found in file 'kotlin/reflect/jvm/internal/UtilKt.java':

```
[line 72: ]
[line 73:  "defaultPrimitiveValue", "", "type", "Ljava/lang/reflect/Type;",
"deserializeToDescriptor", "D", "M",
"Lkotlin/reflect/jvm/internal/impl/protobuf/MessageLite;", "moduleAnchor",
"Ljava/lang/Class;", "proto", "nameResolver",
"Lkotlin/reflect/jvm/internal/impl/metadata/deserialization/NameResolver;",
"typeTable", "Lkotlin/reflect/jvm/internal/impl/metadata/deserialization/TypeTable;",
"metadataVersion",
"Lkotlin/reflect/jvm/internal/impl/metadata/deserialization/BinaryVersion;",
"createDescriptor", "Lkotlin/Function2;",
"Lkotlin/reflect/jvm/internal/impl/serialization/deserialization/MemberDeserializer;",
"Lkotlin/ExtensionFunctionType;", "
(Ljava/lang/Class;Lorg/jetbrains/kotlin/protobuf/MessageLite;Lorg/jetbrains/kotlin/meta
data/deserialization/NameResolver;Lorg/jetbrains/kotlin/metadata/deserialization/TypeTa
ble;Lorg/jetbrains/kotlin/metadata/deserialization/BinaryVersion;Lkotlin/jvm/functions/
Function2;)Lorg/jetbrains/kotlin/descriptors/CallableDescriptor;", "loadClass",
"ClassLoader"[CUTTED]]
[line 74: /* compiled from: util.kt */]
```

```
[line 99:             }]
[line 100:             klass =
loadClass(ReflectClassUtilKt.getSafeClassLoader(classDescriptor.getClass()), classId,
0);]
[line 101:         }]
```

```
[line 104: ]
[line 105:     static /* synthetic */ Class loadClass$default(ClassLoader ClassLoader,
ClassId classId, int i, int i2, Object obj) {]
[line 106:         if ((i2 & 4) != 0) {]
```

```
[line 108:         }]
[line 109:         return loadClass(ClassLoader, classId, i);]
[line 110:     }]
[line 111: ]
[line 112:     private static final Class<?> loadClass(ClassLoader ClassLoader, ClassId
classId, int i) {]
[line 113:         JavaToKotlinClassMap javaToKotlinClassMap =
JavaToKotlinClassMap.INSTANCE;]
```

```
[line 123:         Intrinsics.checkNotNullExpressionValue(asString2,
"javaClassId.relativeClassName.asString()");]
[line 124:         return loadClass(ClassLoader, asString, asString2, i);]
[line 125:     }]
[line 126: ]
[line 127:     private static final Class<?> loadClass(ClassLoader ClassLoader, String
str, String str2, int i) {]
[line 128:         if (Intrinsics.areEqual((Object) str, (Object) "kotlin")) {]
```

```
[line 189:         }]
[line 190:         return ReflectJavaClassFinderKt.tryLoadClass(ClassLoader, str);]
[line 191:     }]
```

```
[line 246:             ConstantValue constantValue = (ConstantValue) entry.getValue();]
[line 247:             ClassLoader classLoader = toJavaClass.getClassLoader();]
[line 248:             Intrinsics.checkNotNullExpressionValue(classLoader,
"annotationClass.ClassLoader");]
[line 249:             Object toRuntimeValue = toRuntimeValue(constantValue,
ClassLoader);]
[line 250:             toRuntimeValue = toRuntimeValue != null ?
TuplesKt.to(name.asString(), toRuntimeValue) : null;]
```

```
[line 257: ]
[line 258:     private static final Object toRuntimeValue(ConstantValue<?>
constantValue, ClassLoader ClassLoader) {]
[line 259:         if (constantValue instanceof AnnotationValue) {]
```

```
[line 265:             for (ConstantValue toRuntimeValue : iterable) {]
[line 266:                 arrayList.add(toRuntimeValue(toRuntimeValue, ClassLoader));]
[line 267:             }]
```

```
[line 274:             Name name = (Name) pair.component2();]
[line 275:             Class loadClass$default = loadClass$default(ClassLoader,
classId, 0, 4, null);]
[line 276:             if (loadClass$default == null) {]
```

```
[line 284:                 NormalClass normalClass = (NormalClass) value;]
[line 285:                 return loadClass(ClassLoader, normalClass.getClassId(),
normalClass.getArrayDimensions());]
[line 286:             } else if (value instanceof LocalClass) {]
```

```
[line 368:         Intrinsics.checkNotNullParameter(function22, "createDescriptor");]
[line 369:         RuntimeModuleData orCreateModule =
ModuleByClassLoaderKt.getOrCreateModule(cls);]
[line 370:         if (m2 instanceof Function) {]
```

There is **'ClassLoader'** found in file 'kotlin/reflect/jvm/internal/KDeclarationContainerImpl$Data$moduleData$2.java':

```
[line 19:     public final RuntimeModuleData invoke() {]
[line 20:         return
ModuleByClassLoaderKt.getOrCreateModule(KDeclarationContainerImpl.this.getJClass());]
[line 21:     }]
```

There is **'ClassLoader'** found in file 'kotlin/reflect/jvm/internal/ModuleByClassLoaderKt.java':

```
[line 10: ]
[line 11: @Metadata(bv = {1, 0, 3}, d1 = {"\u0000
\n\u0000\n\u0002\u0018\u0002\n\u0002\u0018\u0002\n\u0002\u0018\u0002\n\u0002\u0018\u000
2\n\u0000\n\u0002\u0010\u0002\n\u0000\n\u0002\u0018\u0002\n\u0000\u001a\b\u0010\u0005\u
001a\u00020\u0006H\u0000\u001a\u0010\u0010\u0007\u001a\u00020\u0004*\u0006\u0012\u0002\
b\u00030\bH\u0000\"
\u0010\u0000\u001a\u0014\u0012\u0004\u0012\u00020\u0002\u0012\n\u0012\b\u0012\u0004\u00
12\u00020\u00040\u00030\u0001X\u0004¢\u0006\u0002\n\u0000¨\u0006\t"}, d2 =
{"moduleByClassLoader", "Ljava/util/concurrent/ConcurrentMap;",
"Lkotlin/reflect/jvm/internal/WeakClassLoaderBox;", "Ljava/lang/ref/WeakReference;",
"Lkotlin/reflect/jvm/internal/impl/descriptors/runtime/components/RuntimeModuleData;",
"clearModuleByClassLoaderCache", "", "getOrCreateModule", "Ljava/lang/Class;", "kotlin-
reflection"}, k = 2, mv = {1, 4, 0})]
[line 12: /* compiled from: moduleByClassLoader.kt */]
[line 13: public final class ModuleByClassLoaderKt {]
[line 14:     private static final ConcurrentMap<WeakClassLoaderBox,
WeakReference<RuntimeModuleData>> moduleByClassLoader = new ConcurrentHashMap();]
[line 15:]
```

```
[line 18:         Intrinsics.checkNotNullParameter(cls, "$this$getOrCreateModule");]
[line 19:         ClassLoader safeClassLoader =
ReflectClassUtilKt.getSafeClassLoader(cls);]
[line 20:         WeakClassLoaderBox weakClassLoaderBox = new
WeakClassLoaderBox(safeClassLoader);]
[line 21:         ConcurrentMap concurrentMap = moduleByClassLoader;]
[line 22:         WeakReference weakReference = (WeakReference)
concurrentMap.get(weakClassLoaderBox);]
[line 23:         if (weakReference != null) {]
```

```
[line 28:             }]
[line 29:             concurrentMap.remove(weakClassLoaderBox, weakReference);]
[line 30:         }]
[line 31:         RuntimeModuleData create =
RuntimeModuleData.Companion.create(safeClassLoader);]
[line 32:         while (true) {]
[line 33:             try {]
[line 34:                 ConcurrentMap concurrentMap2 = moduleByClassLoader;]
[line 35:                 WeakReference weakReference2 = (WeakReference)
concurrentMap2.putIfAbsent(weakClassLoaderBox, new WeakReference(create));]
[line 36:                 if (weakReference2 != null) {]
```

```
[line 40:                     }]
[line 41:                     concurrentMap2.remove(weakClassLoaderBox,
weakReference2);]
[line 42:                 } else {]
[line 43:                     weakClassLoaderBox.setTemporaryStrongRef((ClassLoader)
null);]
[line 44:                     return create;]
```

```
[line 46:             } finally {]
[line 47:                 weakClassLoaderBox.setTemporaryStrongRef((ClassLoader)
null);]
[line 48:             }]
```

```
[line 52: ]
[line 53:     public static final void clearModuleByClassLoaderCache() {]
[line 54:         moduleByClassLoader.clear();]
[line 55:     }]
```

There is **'ClassLoader'** found in file 'kotlin/reflect/jvm/internal/ReflectionFactoryImpl.java':

```
[line 136:          KClassCacheKt.clearKClassCache();]
[line 137:          ModuleByClassLoaderKt.clearModuleByClassLoaderCache();]
[line 138:      }]
```

There is **'ClassLoader'** found in file 'kotlin/reflect/jvm/internal/KPackageImpl$Data$multifileFacade$2.java':

```
[line 52:          r0 = r0.getJClass();]
[line 53:          r0 = r0.getClassLoader();]
[line 54:          r3 = 47;]
```

There is **'ClassLoader'** found in file 'kotlin/reflect/jvm/internal/WeakClassLoaderBox.java':

```
[line 6: ]
[line 7: @Metadata(bv = {1, 0, 3}, d1 =
{"\u00000\n\u0002\u0018\u0002\n\u0002\u0010\u0000\n\u0000\n\u0002\u0018\u0002\n\u0002\b
\u0002\n\u0002\u0010\b\n\u0002\b\u0003\n\u0002\u0018\u0002\n\u0002\b\u0007\n\u0002\u001
0\u000b\n\u0002\b\u0003\n\u0002\u0010\u000e\n\u0000\b\u0002\u0018\u00002\u00020\u0001B\
r\u0012\u0006\u0010\u0002\u001a\u00020\u0003¢
\u0006\u0002\u0010\u0004J\u0013\u0010\u0011\u001a\u00020\u00122\b\u0010\u0013\u001a\u00
04\u0018\u00010\u0001H\u0002J\b\u0010\u0014\u001a\u00020\u0006H\u0016J\b\u0010\u0015\u
001a\u00020\u0016H\u0016R\u0011\u0010\u0005\u001a\u00020\u0006¢
\u0006\b\n\u0000\u001a\u0004\b\u0007\u0010\bR\u0017\u0010\t\u001a\b\u0012\u0004\u0012\u
00020\u00030\n¢
\u0006\b\n\u0000\u001a\u0004\b\u000b\u0010\fR\u001c\u0010\r\u001a\u0004\u0018\u00010\u0
003X\u000e¢
\u0006\u000e\n\u0000\u001a\u0004\b\u000e\u0010\u000f\"\u0004\b\u0010\u0010\u0004¨\u0006
\u0017"}, d2 = {"Lkotlin/reflect/jvm/internal/WeakClassLoaderBox;", "", "classLoader",
"Ljava/lang/ClassLoader;", "(Ljava/lang/ClassLoader;)V", "identityHashCode", "",
"getIdentityHashCode", "()I", "ref", "Ljava/lang/ref/WeakReference;", "getRef", "
()Ljava/lang/ref/WeakReference;", "temporaryStrongRef", "getTemporaryStrongRef", "
()Ljava/lang/ClassLoader;", "setTemporaryStrongRef", "equals", "", "other", "hashCode",
"toString", "", "kotlin-reflection"}, k = 1, mv = {1, 4, 0})]
[line 8: /* compiled from: moduleByClassLoader.kt */]
[line 9: final class WeakClassLoaderBox {]
[line 10:     private final int identityHashCode;]
[line 11:     private final WeakReference<ClassLoader> ref;]
[line 12:     private ClassLoader temporaryStrongRef;]
[line 13: ]
[line 14:     public WeakClassLoaderBox(ClassLoader ClassLoader) {]
[line 15:         Intrinsics.checkNotNullParameter(classLoader, "ClassLoader");]
[line 16:         this.ref = new WeakReference(ClassLoader);]
[line 17:         this.identityHashCode = System.identityHashCode(ClassLoader);]
[line 18:         this.temporaryStrongRef = ClassLoader;]
[line 19:     }]
[line 20: ]
[line 21:     public final void setTemporaryStrongRef(ClassLoader ClassLoader) {]
[line 22:         this.temporaryStrongRef = ClassLoader;]
[line 23:     }]
```

```
[line 25:     public boolean equals(Object obj) {]
[line 26:         return (obj instanceof WeakClassLoaderBox) && ((ClassLoader)
this.ref.get()) == ((ClassLoader) ((WeakClassLoaderBox) obj).ref.get());]
[line 27:     }]
```

```
[line 33:     public String toString() {]
[line 34:         ClassLoader classLoader = (ClassLoader) this.ref.get();]
[line 35:         if (ClassLoader != null) {]
[line 36:             String classLoader2 = ClassLoader.toString();]
[line 37:             if (ClassLoader2 != null) {]
[line 38:                 return ClassLoader2;]
[line 39:             }]
```

There is **'ClassLoader'** found in file 'kotlin/reflect/jvm/internal/KDeclarationContainerImpl.java':

```
[line 317:             if (z) {]
[line 318:                 ClassLoader safeClassLoader =
ReflectClassUtilKt.getSafeClassLoader(cls3);]
[line 319:                 StringBuilder stringBuilder = new StringBuilder();]
```

```
[line 321:                 stringBuilder.append("$DefaultImpls");]
[line 322:                 superclass =
ReflectJavaClassFinderKt.tryLoadClass(safeClassLoader, stringBuilder.toString());]
[line 323:                 if (superclass != null) {]
```

```
[line 520:         } else if (charAt == 'L') {]
[line 521:             ClassLoader safeClassLoader =
ReflectClassUtilKt.getSafeClassLoader(getJClass());]
[line 522:             i++;]
```

```
[line 526:             Intrinsics.checkNotNullExpressionValue(substring, "(this as
java.lang.Strin…ing(startIndex, endIndex)");]
[line 527:             obj =
safeClassLoader.loadClass(StringsKt__StringsJVMKt.replace$default(substring, '/', '.',
false, 4, null));]
[line 528:         } else if (charAt == 'S') {]
```

There is **'ClassLoader'** found in file 'kotlin/reflect/jvm/internal/calls/AnnotationConstructorCallerKt.java':

```
[line 100:         Lazy lazy2 = LazyKt__LazyJVMKt.lazy(new
AnnotationConstructorCallerKt$createAnnotationInstance$toString$2(cls, map));]
[line 101:         Object newProxyInstance =
Proxy.newProxyInstance(cls.getClassLoader(), new Class[]{cls}, new
AnnotationConstructorCallerKt$createAnnotationInstance$result$1(cls, lazy2, null, lazy,
null, annotationConstructorCallerKt$createAnnotationInstance$2, map));]
[line 102:         Objects.requireNonNull(newProxyInstance, "null cannot be cast to
non-null type T");]
```

There is **'ClassLoader'** found in file 'kotlin/reflect/jvm/internal/impl/resolve/OverridingUtil.java':

```
[line 58:     private static final TypeConstructorEquality
DEFAULT_TYPE_CONSTRUCTOR_EQUALITY;]
[line 59:     private static final List<ExternalOverridabilityCondition>
EXTERNAL_CONDITIONS =
CollectionsKt___CollectionsKt.toList(ServiceLoader.load(ExternalOverridabilityCondition
.class, ExternalOverridabilityCondition.class.getClassLoader()));]
[line 60:     private final TypeConstructorEquality equalityAxioms;]
```

There is **'ClassLoader'** found in file 'kotlin/reflect/jvm/internal/impl/serialization/deserialization/builtins/BuiltInsResourceLoader.java':

```
[line 9:         Intrinsics.checkNotNullParameter(str, "path");]
[line 10:         ClassLoader classLoader = getClass().getClassLoader();]
[line 11:         if (ClassLoader != null) {]
[line 12:             InputStream resourceAsStream =
ClassLoader.getResourceAsStream(str);]
[line 13:             if (resourceAsStream != null) {]
```

```
[line 16:         }]
[line 17:         return ClassLoader.getSystemResourceAsStream(str);]
[line 18:     }]
```

There is **'ClassLoader'** found in file 'kotlin/reflect/jvm/internal/impl/builtins/BuiltInsLoader$Companion$Instance$2.java':

```
[line 16:     public final BuiltInsLoader invoke() {]
[line 17:         ServiceLoader load = ServiceLoader.load(BuiltInsLoader.class,
BuiltInsLoader.class.getClassLoader());]
[line 18:         Intrinsics.checkNotNullExpressionValue(load, "implementations");]
```

There is **'ClassLoader'** found in file 'kotlin/reflect/jvm/internal/impl/descriptors/Visibilities.java':

```
[line 448:         DEFAULT_VISIBILITY = anonymousClass5;]
[line 449:         Iterator it = ServiceLoader.load(ModuleVisibilityHelper.class,
ModuleVisibilityHelper.class.getClassLoader()).iterator();]
[line 450:         MODULE_VISIBILITY_HELPER = it.hasNext() ? (ModuleVisibilityHelper)
it.next() : EMPTY.INSTANCE;]
```

There is **'ClassLoader'** found in file 'kotlin/reflect/jvm/internal/impl/descriptors/runtime/components/ReflectJavaClassFinderKt.java':

```
[line 6: public final class ReflectJavaClassFinderKt {]
[line 7:     public static final Class<?> tryLoadClass(ClassLoader ClassLoader, String
str) {]
[line 8:         Intrinsics.checkNotNullParameter(ClassLoader, "$this$tryLoadClass");]
[line 9:         Intrinsics.checkNotNullParameter(str, "fqName");]
[line 10:         try {]
[line 11:             return Class.forName(str, false, ClassLoader);]
[line 12:         } catch (ClassNotFoundException unused) {]
```

There is **'ClassLoader'** found in file 'kotlin/reflect/jvm/internal/impl/descriptors/runtime/components/ReflectJavaClassFinder.java':

```
[line 16: public final class ReflectJavaClassFinder implements JavaClassFinder {]
[line 17:     private final ClassLoader ClassLoader;]
[line 18:]
```

```
[line 23: ]
[line 24:     public ReflectJavaClassFinder(ClassLoader ClassLoader) {]
[line 25:         Intrinsics.checkNotNullParameter(classLoader, "ClassLoader");]
[line 26:         this.classLoader = ClassLoader;]
[line 27:     }]
```

```
[line 43:         }]
[line 44:         Class tryLoadClass =
ReflectJavaClassFinderKt.tryLoadClass(this.ClassLoader, replace$default);]
[line 45:         return tryLoadClass != null ? new ReflectJavaClass(tryLoadClass) :
null;]
```

There is **'ClassLoader'** found in file 'kotlin/reflect/jvm/internal/impl/descriptors/runtime/components/RuntimeModuleData.java':

```
[line 43: ]
[line 44:         public final RuntimeModuleData create(ClassLoader ClassLoader) {]
[line 45:             ClassLoader classLoader2 = ClassLoader;]
[line 46:             Intrinsics.checkNotNullParameter(classLoader2, "ClassLoader");]
[line 47:             StorageManager lockBasedStorageManager = new
LockBasedStorageManager("RuntimeModuleData");]
```

```
[line 50:             stringBuilder.append("<runtime module for ");]
[line 51:             stringBuilder.append(ClassLoader2);]
[line 52:             stringBuilder.append(Typography.greater);]
[line 53:             Name special = Name.special(stringBuilder.toString());]
[line 54:             Intrinsics.checkNotNullExpressionValue(special, "Name.special(\"
<runtime module for $ClassLoader>\")");]
[line 55:             ModuleDescriptorImpl moduleDescriptorImpl = new
ModuleDescriptorImpl(special, lockBasedStorageManager, jvmBuiltIns, null, null, null,
56, null);]
```

```
[line 58:             jvmBuiltIns.initialize(moduleDescriptor, true);]
[line 59:             ReflectKotlinClassFinder reflectKotlinClassFinder = new
ReflectKotlinClassFinder(ClassLoader2);]
[line 60:             DeserializedDescriptorResolver deserializedDescriptorResolver =
new DeserializedDescriptorResolver();]
```

```
[line 66:             StorageManager storageManager = lockBasedStorageManager;]
[line 67:             LazyJavaPackageFragmentProvider
makeLazyJavaPackageFragmentFromClassLoaderProvider$default =
RuntimeModuleDataKt.makeLazyJavaPackageFragmentFromClassLoaderProvider$default(ClassLoa
der2, moduleDescriptor, lockBasedStorageManager, notFoundClasses, kotlinClassFinder,
deserializedDescriptorResolver, singleModuleClassResolver, null, 128, null);]
[line 68:             DeserializationComponentsForJava
makeDeserializationComponentsForJava =
RuntimeModuleDataKt.makeDeserializationComponentsForJava(moduleDescriptor,
storageManager, notFoundClasses2,
makeLazyJavaPackageFragmentFromClassLoaderProvider$default, kotlinClassFinder,
deserializedDescriptorResolver);]
[line 69:
deserializedDescriptorResolver.setComponents(makeDeserializationComponentsForJava);]
```

```
[line 71:             Intrinsics.checkNotNullExpressionValue(javaResolverCache,
"JavaResolverCache.EMPTY");]
[line 72:             singleModuleClassResolver.setResolver(new
JavaDescriptorResolver(makeLazyJavaPackageFragmentFromClassLoaderProvider$default,
javaResolverCache));]
[line 73:             classLoader2 = Unit.class.getClassLoader();]
[line 74:             Intrinsics.checkNotNullExpressionValue(classLoader2,
"stdlibClassLoader");]
[line 75:             StorageManager storageManager2 = storageManager;]
```

```
[line 78:             ReflectKotlinClassFinder reflectKotlinClassFinder2 =
reflectKotlinClassFinder;]
[line 79:             JvmBuiltInsPackageFragmentProvider
jvmBuiltInsPackageFragmentProvider = new
JvmBuiltInsPackageFragmentProvider(storageManager3, new
ReflectKotlinClassFinder(ClassLoader2), moduleDescriptor, notFoundClasses2,
jvmBuiltIns.getSettings(), jvmBuiltIns.getSettings(), Default.INSTANCE,
NewKotlinTypeChecker.Companion.getDefault(), new
SamConversionResolverImpl(storageManager2, CollectionsKt__CollectionsKt.emptyList()));]
[line 80:             ModuleDescriptorImpl[] moduleDescriptorImplArr = new
ModuleDescriptorImpl[1];]
```

There is **'ClassLoader'** found in file 'kotlin/reflect/jvm/internal/impl/descriptors/runtime/components/ReflectKotlinClassFinder.java':

```
[line 15: public final class ReflectKotlinClassFinder implements KotlinClassFinder {]
[line 16:     private final ClassLoader ClassLoader;]
[line 17: ]
[line 18:     public ReflectKotlinClassFinder(ClassLoader ClassLoader) {]
[line 19:         Intrinsics.checkNotNullParameter(classLoader, "ClassLoader");]
[line 20:         this.classLoader = ClassLoader;]
[line 21:     }]
```

```
[line 24:         KotlinClass kotlinClass;]
[line 25:         Class tryLoadClass =
ReflectJavaClassFinderKt.tryLoadClass(this.ClassLoader, str);]
[line 26:         if (tryLoadClass != null) {]
```

```
[line 57:         if (fqName.startsWith(KotlinBuiltIns.BUILT_INS_PACKAGE_NAME)) {]
[line 58:             return
this.ClassLoader.getResourceAsStream(BuiltInSerializerProtocol.INSTANCE.getBuiltInsFile
Path(fqName));]
[line 59:         }]
```

There is **'ClassLoader'** found in file 'kotlin/reflect/jvm/internal/impl/descriptors/runtime/components/RuntimeModuleDataKt.java':

```
[line 41: public final class RuntimeModuleDataKt {]
[line 42:     public static /* synthetic */ LazyJavaPackageFragmentProvider
makeLazyJavaPackageFragmentFromClassLoaderProvider$default(ClassLoader ClassLoader,
ModuleDescriptor moduleDescriptor, StorageManager storageManager, NotFoundClasses
notFoundClasses, KotlinClassFinder kotlinClassFinder, DeserializedDescriptorResolver
deserializedDescriptorResolver, ModuleClassResolver moduleClassResolver,
PackagePartProvider packagePartProvider, int i, Object obj) {]
[line 43:         return
makeLazyJavaPackageFragmentFromClassLoaderProvider(ClassLoader, moduleDescriptor,
storageManager, notFoundClasses, kotlinClassFinder, deserializedDescriptorResolver,
moduleClassResolver, (i & 128) != 0 ? Empty.INSTANCE : packagePartProvider);]
[line 44:     }]
[line 45: ]
[line 46:     public static final LazyJavaPackageFragmentProvider
makeLazyJavaPackageFragmentFromClassLoaderProvider(ClassLoader ClassLoader,
ModuleDescriptor moduleDescriptor, StorageManager storageManager, NotFoundClasses
notFoundClasses, KotlinClassFinder kotlinClassFinder, DeserializedDescriptorResolver
deserializedDescriptorResolver, ModuleClassResolver moduleClassResolver,
PackagePartProvider packagePartProvider) {]
[line 47:         ClassLoader classLoader2 = ClassLoader;]
[line 48:         ModuleDescriptor moduleDescriptor2 = moduleDescriptor;]
```

```
[line 56:         PackagePartProvider packagePartProvider2 = packagePartProvider;]
[line 57:         Intrinsics.checkNotNullParameter(classLoader2, "ClassLoader");]
[line 58:         Intrinsics.checkNotNullParameter(moduleDescriptor2, "module");]
```

```
[line 66:         AnnotationTypeQualifierResolver annotationTypeQualifierResolver2 =
new AnnotationTypeQualifierResolver(storageManager2, Jsr305State.DISABLED);]
[line 67:         JavaClassFinder reflectJavaClassFinder = new
ReflectJavaClassFinder(ClassLoader2);]
[line 68:         SignaturePropagator signaturePropagator =
SignaturePropagator.DO_NOTHING;]
```

There is **'ClassLoader'** found in file 'kotlin/reflect/jvm/internal/impl/descriptors/runtime/structure/ReflectClassUtilKt.java':

```
[line 48: ]
[line 49:     public static final ClassLoader getSafeClassLoader(Class<?> cls) {]
[line 50:         Intrinsics.checkNotNullParameter(cls, "$this$safeClassLoader");]
[line 51:         ClassLoader classLoader = cls.getClassLoader();]
[line 52:         if (ClassLoader != null) {]
[line 53:             return ClassLoader;]
[line 54:         }]
[line 55:         classLoader = ClassLoader.getSystemClassLoader();]
[line 56:         Intrinsics.checkNotNullExpressionValue(classLoader,
"ClassLoader.getSystemClassLoader()");]
[line 57:         return ClassLoader;]
[line 58:     }]
```

There is **'ClassLoader'** found in file 'kotlin/reflect/jvm/internal/impl/descriptors/runtime/structure/Java8ParameterNamesLoader.java':

```
[line 41:         try {]
[line 42:             return new Cache(cls.getMethod("getParameters", new Class[0]),
ReflectClassUtilKt.getSafeClassLoader(cls).loadClass("java.lang.reflect.Parameter").get
Method("getName", new Class[0]));]
[line 43:         } catch (NoSuchMethodException unused) {]
```

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

```
[line 19:         r1 = "ClassCastException(\"Inst…baseTypeCL\").initCause(e)";]
[line 20:         r2 = ", base type ClassLoader: ";]
[line 21:         r3 = "Instance ClassLoader: ";]
[line 22:         r4 = "null cannot be cast to non-null type
kotlin.internal.PlatformImplementations";]
```

```
[line 43:         r6 = r6.getClass();    Catch:{ ClassNotFoundException -> 0x0060 }]
[line 44:         r6 = r6.getClassLoader();      Catch:{ ClassNotFoundException ->
0x0060 }]
[line 45:         r8 = kotlin.internal.PlatformImplementations.class;]
[line 46:         r8 = r8.getClassLoader();      Catch:{ ClassNotFoundException ->
0x0060 }]
[line 47:         r9 = new java.lang.ClassCastException;         Catch:{
ClassNotFoundException -> 0x0060 }]
```

```
[line 77:         r6 = r6.getClass();    Catch:{ ClassNotFoundException -> 0x00ad }]
[line 78:         r6 = r6.getClassLoader();      Catch:{ ClassNotFoundException ->
0x00ad }]
[line 79:         r8 = kotlin.internal.PlatformImplementations.class;]
[line 80:         r8 = r8.getClassLoader();      Catch:{ ClassNotFoundException ->
0x00ad }]
[line 81:         r9 = new java.lang.ClassCastException;         Catch:{
ClassNotFoundException -> 0x00ad }]
```

```
[line 115:         r0 = r0.getClass();   Catch:{ ClassNotFoundException -> 0x0101 }]
[line 116:         r0 = r0.getClassLoader();     Catch:{ ClassNotFoundException ->
0x0101 }]
[line 117:         r7 = kotlin.internal.PlatformImplementations.class;]
[line 118:         r7 = r7.getClassLoader();     Catch:{ ClassNotFoundException ->
0x0101 }]
[line 119:         r8 = new java.lang.ClassCastException;        Catch:{
ClassNotFoundException -> 0x0101 }]
```

```
[line 150:         r0 = r0.getClass();   Catch:{ ClassNotFoundException -> 0x014e }]
[line 151:         r0 = r0.getClassLoader();     Catch:{ ClassNotFoundException ->
0x014e }]
[line 152:         r5 = kotlin.internal.PlatformImplementations.class;]
[line 153:         r5 = r5.getClassLoader();     Catch:{ ClassNotFoundException ->
0x014e }]
[line 154:         r6 = new java.lang.ClassCastException;        Catch:{
ClassNotFoundException -> 0x014e }]
```

```
[line 182:         } catch (ClassCastException e) {]
[line 183:             ClassLoader classLoader = obj.getClass().getClassLoader();]
[line 184:             Intrinsics.reifiedOperationMarker(4, str);]
[line 185:             ClassLoader classLoader2 = Object.class.getClassLoader();]
[line 186:             StringBuilder stringBuilder = new StringBuilder();]
[line 187:             stringBuilder.append("Instance ClassLoader: ");]
[line 188:             stringBuilder.append(ClassLoader);]
[line 189:             stringBuilder.append(", base type ClassLoader: ");]
[line 190:             stringBuilder.append(ClassLoader2);]
[line 191:             Throwable initCause = new
ClassCastException(stringBuilder.toString()).initCause(e);]
```

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

```
[line 7: ]
[line 8: @Metadata(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\u001a3\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\bø\u0001\u0000¢
\u0006\u0002\u0010\u0013\u0002\u0007\n\u0005\b20\u0001¨\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, 5, 1})]
[line 9: /* compiled from: Thread.kt */]
[line 10: public final class ThreadsKt {]
[line 11:     public static /* synthetic */ Thread thread$default(boolean z, boolean
z2, ClassLoader ClassLoader, String str, int i, Function0 function0, int i2, Object
obj) {]
[line 12:         boolean z3 = (i2 & 1) != 0 ? true : z;]
```

```
[line 14:         if ((i2 & 4) != 0) {]
[line 15:             classLoader = (ClassLoader) null;]
[line 16:         }]
[line 17:         ClassLoader classLoader2 = ClassLoader;]
[line 18:         if ((i2 & 8) != 0) {]
```

```
[line 20:         }]
[line 21:         return thread(z3, z4, ClassLoader2, str, (i2 & 16) != 0 ? -1 : i,
function0);]
[line 22:     }]
[line 23: ]
[line 24:     public static final Thread thread(boolean z, boolean z2, ClassLoader
ClassLoader, String str, int i, Function0<Unit> function0) {]
[line 25:         Intrinsics.checkNotNullParameter(function0, "block");]
```

```
[line 35:         }]
[line 36:         if (ClassLoader != null) {]
[line 37:             threadsKt$thread$thread$1.setContextClassLoader(ClassLoader);]
[line 38:         }]
```

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

```
[line 64:         try {]
[line 65:             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(HintConstants.AUTOFILL_HINT_NAME, new Class[0]));]
[line 66:             cache = cache;]
```

There is **'ClassLoader'** found in file 'ehn/techiop/hcert/kotlin/chain/impl/JvmSchemaLoader.java':

```
[line 9: ]
[line 10: @Metadata(d1 =
{"\u0000(\n\u0002\u0018\u0002\n\u0002\u0018\u0002\n\u0002\u0018\u0002\n\u0002\b\u0002\n
\u0002\u0018\u0002\n\u0002\b\u0002\n\u0002\u0018\u0002\n\u0002\b\u0002\n\u0002\u0010\u0
00e\n\u0002\b\u0007\u0018\u00002\b\u0012\u0004\u0012\u00020\u00020\u0001B\u0005¢
\u0006\u0002\u0010\u0003J\u0010\u0010\u0004\u001a\n
\u0006*\u0004\u0018\u00010\u00050\u0005H\u0002J\b\u0010\u0007\u001a\u00020\bH\u0002J\u0
010\u0010\t\u001a\u00020\b2\u0006\u0010\n\u001a\u00020\u000bH\u0002J\r\u0010\f\u001a\u0
0020\u0002H\u0010¢
\u0006\u0002\b\rJ\u0015\u0010\u000e\u001a\u00020\u00022\u0006\u0010\n\u001a\u00020\u000
bH\u0010¢
\u0006\u0002\b\u000fJ\u0010\u0010\u0010\u001a\u00020\u00022\u0006\u0010\u0011\u001a\u00
020\bH\u0002¨\u0006\u0012"}, d2 =
{"Lehn/techiop/hcert/kotlin/chain/impl/JvmSchemaLoader;",
"Lehn/techiop/hcert/kotlin/chain/impl/SchemaLoader;",
"Lnet/pwall/json/schema/JSONSchema;", "()V", "classLoader", "Ljava/lang/ClassLoader;",
"kotlin.jvm.PlatformType", "getFallbackSchema", "Ljava/io/InputStream;",
"getSchemaResource", "version", "", "loadFallbackSchema",
"loadFallbackSchema$hcert_kotlin", "loadSchema", "loadSchema$hcert_kotlin", "parse",
"resource", "hcert-kotlin"}, k = 1, mv = {1, 5, 1}, xi = 48)]
[line 11: /* compiled from: SchemaValidationAdapter.kt */]
```

```
[line 74:     private final InputStream getSchemaResource(String str) {]
[line 75:         ClassLoader classLoader = ClassLoader();]
[line 76:         StringBuilder stringBuilder = new StringBuilder();]
```

```
[line 79:         stringBuilder.append("/DCC.combined-schema.json");]
[line 80:         InputStream resourceAsStream =
ClassLoader.getResourceAsStream(stringBuilder.toString());]
[line 81:         if (resourceAsStream != null) {]
```

```
[line 87:     private final InputStream getFallbackSchema() {]
[line 88:         InputStream resourceAsStream =
ClassLoader().getResourceAsStream("json/schema/fallback/DCC.combined-schema.json");]
[line 89:         if (resourceAsStream != null) {]
```

```
[line 94: ]
[line 95:     private final ClassLoader ClassLoader() {]
[line 96:         return SchemaValidationAdapter.class.getClassLoader();]
[line 97:     }]
```

There is **'ClassLoader'** found in file 'com/koushikdutta/async/http/spdy/SpdyMiddleware.java':

```
[line 99:                 String stringBuilder2 = stringBuilder.toString();]
[line 100:                 this.nativeGetNpnNegotiatedProtocol =
Class.forName(stringBuilder2, true,
this.sslParameters.getType().getClassLoader()).getDeclaredMethod("SSL_get_npn_negotiate
d_protocol", new Class[]{Long.TYPE});]
[line 101:                 this.nativeGetAlpnNegotiatedProtocol =
Class.forName(stringBuilder2, true,
this.sslParameters.getType().getClassLoader()).getDeclaredMethod("SSL_get0_alpn_selecte
d", new Class[]{Long.TYPE});]
[line 102:                 this.peerHost.setAccessible(true);]
```

There is **'ClassLoader'** found in file 'com/learnium/RNDeviceInfo/RNInstallReferrerClient.java':

```
[line 91:                 this.mReferrerClient = invoke.getClass().getMethod("build",
new Class[0]).invoke(invoke, new Object[0]);]
[line 92:                 this.installReferrerStateListener =
Proxy.newProxyInstance(InstallReferrerStateListenerClazz.getClassLoader(), new Class[]
{InstallReferrerStateListenerClazz}, new InstallReferrerStateListenerProxy());]
[line 93:                 InstallReferrerClientClazz.getMethod("startConnection", new
Class[]{InstallReferrerStateListenerClazz}).invoke(this.mReferrerClient, new Object[]
{this.installReferrerStateListener});]
```

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

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 'net/i2p/crypto/eddsa/math/Curve.java':

```
[line 5: ]
[line 6: public class Curve implements Serializable {]
[line 7:     private static final long serialVersionUID = 4578920872509827L;]
```

There is **'implements Serializable'** found in file 'net/i2p/crypto/eddsa/math/Field.java':

```
[line 4: ]
[line 5: public class Field implements Serializable {]
[line 6:     private static final long serialVersionUID = 8746587465875676L;]
```

There is **'implements Serializable'** found in file 'net/i2p/crypto/eddsa/math/GroupElement.java':

```
[line 7: ]
[line 8: public class GroupElement implements Serializable {]
[line 9:     private static final long serialVersionUID = 2395879087349587L;]
```

There is **'implements Serializable'** found in file 'net/i2p/crypto/eddsa/math/FieldElement.java':

```
[line 4: ]
[line 5: public abstract class FieldElement implements Serializable {]
[line 6:     private static final long serialVersionUID = 1239527465875676L;]
```

There is **'implements Serializable'** found in file 'net/i2p/crypto/eddsa/math/bigint/BigIntegerLittleEndianEncoding.java':

```
[line 8: ]
[line 9: public class BigIntegerLittleEndianEncoding extends Encoding implements
Serializable {]
[line 10:     private static final long serialVersionUID = 3984579843759837L;]
```

There is **'implements Serializable'** found in file 'net/i2p/crypto/eddsa/math/bigint/BigIntegerFieldElement.java':

```
[line 7: ]
[line 8: public class BigIntegerFieldElement extends FieldElement implements
Serializable {]
[line 9:     private static final long serialVersionUID = 4890398908392808L;]
```

There is **'implements Serializable'** found in file 'net/pwall/util/Ref.java':

```
[line 4: ]
[line 5: public class Ref<T> implements Serializable {]
[line 6:     private static final long serialVersionUID = 6434890119379911833L;]
```

There is **'implements Serializable'** found in file 'net/pwall/util/Pair.java':

```
[line 5: ]
[line 6: public class Pair<F, S> implements Serializable {]
[line 7:     private static final long serialVersionUID = 3881544420119098149L;]
```

There is **'implements Serializable'** found in file 'net/pwall/yaml/YAMLDocument.java':

```
[line 4: ]
[line 5: public class YAMLDocument implements Serializable {]
[line 6:     public static final int defaultMajorVersion = 1;]
```

There is **'implements Serializable'** found in file 'j$/util/concurrent/ConcurrentHashMap.java':

```
[line 63: ]
[line 64:     static class Segment extends ReentrantLock implements Serializable {]
[line 65:         Segment(float f) {]
```

There is **'implements Serializable'** found in file 'j$/time/ZoneRegion.java':

```
[line 7: ]
[line 8: final class ZoneRegion extends ZoneId implements Serializable {]
[line 9:     private final String id;]
```

There is **'implements Serializable'** found in file 'j$/time/ZoneId.java':

```
[line 15: ]
[line 16: public abstract class ZoneId implements Serializable {]
[line 17:     public static final Map SHORT_IDS;]
```

There is **'implements Serializable'** found in file 'j$/time/Clock.java':

```
[line 7: ]
[line 8:     static final class SystemClock extends Clock implements Serializable {]
[line 9:         private final ZoneId zone;]
```

There is **'implements Serializable'** found in file 'j$/time/temporal/ValueRange.java':

```
[line 5: ]
[line 6: public final class ValueRange implements Serializable {]
[line 7:     private final long maxLargest;]
```

There is **'implements Serializable'** found in file 'j$/time/chrono/IsoChronology.java':

```
[line 21: ]
[line 22: public final class IsoChronology extends AbstractChronology implements
Serializable {]
[line 23:     public static final IsoChronology INSTANCE = new IsoChronology();]
```

There is **'implements Serializable'** found in file 'j$/time/zone/ZoneRules.java':

```
[line 18: ]
[line 19: public final class ZoneRules implements Serializable {]
[line 20:     private static final ZoneOffsetTransitionRule[] EMPTY_LASTRULES = new
ZoneOffsetTransitionRule[0];]
```

There is **'implements Serializable'** found in file 'j$/time/zone/ZoneOffsetTransitionRule.java':

```
[line 4: ]
[line 5: public final class ZoneOffsetTransitionRule implements Serializable {]
[line 6: }]
```

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

```
[line 6: /* compiled from: Tuples.kt */]
[line 7: public final class Pair<A, B> implements Serializable {]
[line 8:     private final A first;]
```

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

```
[line 6: /* compiled from: Tuples.kt */]
[line 7: public final class Triple<A, B, C> implements Serializable {]
[line 8:     private final A first;]
```

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

```
[line 10: /* compiled from: Result.kt */]
[line 11: public final class Result<T> implements Serializable {]
[line 12:     public static final Companion Companion = new Companion();]
```

```
[line 35:     /* compiled from: Result.kt */]
[line 36:     public static final class Failure implements Serializable {]
[line 37:         public final Throwable exception;]
```

There is **'implements Serializable'** found in file 'kotlin/reflect/jvm/internal/impl/incremental/components/Position.java':

```
[line 6: /* compiled from: LookupLocation.kt */]
[line 7: public final class Position implements Serializable {]
[line 8:     public static final Companion Companion = new Companion();]
```

There is **'implements Serializable'** found in file 'kotlin/reflect/jvm/internal/impl/protobuf/GeneratedMessageLite.java':

```
[line 17: ]
[line 18: public abstract class GeneratedMessageLite extends AbstractMessageLite
implements Serializable {]
[line 19:]
```

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

```
[line 4: ]
[line 5: final class MapEntry<K, V> implements Serializable {]
[line 6:     public final K key;]
```

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

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

```
[line 20: /* compiled from: Regex.kt */]
[line 21: public final class Regex implements Serializable {]
[line 22:     public static final Companion Companion = new Companion();]
```

```
[line 61:     /* compiled from: Regex.kt */]
[line 62:     private static final class Serialized implements Serializable {]
[line 63:         public static final Companion Companion = new Companion();]
```

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

```
[line 8: /* compiled from: XorWowRandom.kt */]
[line 9: public final class XorWowRandom extends Random implements Serializable {]
[line 10:     private static final Companion Companion = new Companion();]
```

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

```
[line 16:     /* compiled from: Random.kt */]
[line 17:     public static final class Default extends Random implements Serializable
{]
[line 18:]
```

```
[line 20:         /* compiled from: Random.kt */]
[line 21:         private static final class Serialized implements Serializable {]
[line 22:             public static final Serialized INSTANCE = new Serialized();]
```

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

```
[line 10: /* compiled from: PlatformRandom.kt */]
[line 11: final class PlatformRandom extends AbstractPlatformRandom implements
Serializable {]
[line 12:     private static final Companion Companion = new Companion();]
```

There is **'implements Serializable'** found in file 'com/drew/metadata/jpeg/JpegComponent.java':

```
[line 4: ]
[line 5: public class JpegComponent implements Serializable {]
[line 6:     private static final long serialVersionUID = 61121257899091914L;]
```

There is **'implements Serializable'** found in file 'com/koushikdutta/async/http/spdy/ByteString.java':

```
[line 16: ]
[line 17: final class ByteString implements Serializable {]
[line 18:     public static final ByteString EMPTY = of(new byte[0]);]
```

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

External Communications and Outgoing Traffic

Mobile Application Endpoints

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
pwall.net:80	92.243.0.172:80	N	Not Tested Yet	Not Tested Yet
purl.org:80	207.241.239.241:80	N	Not Tested Yet	Not Tested Yet
cipa.jp:80	118.82.81.189:80	N	Not Tested Yet	Not Tested Yet
iptc.org:80	3.64.29.21:80	N	Not Tested Yet	Not Tested Yet

Hostname	IP:Port	SSL Encryption	Websec Server Security	Domain Domain Security
www.aiim.org:80	199.60.103.31:80	N	Not Tested Yet	Not Tested Yet
www.npes.org:80	216.33.126.92:80	N	Not Tested Yet	Not Tested Yet
ns.useplus.org:80	54.83.4.77:80	N	Not Tested Yet	Not Tested Yet
javax.xml.xmlconstants:80	Not Resolved:80	Not Tested Yet	Not Tested Yet	Not Tested Yet
greencheck.gv.at:443	Not Resolved:443	Not Tested Yet	Not Tested Yet	Not Tested Yet
drewnoakes.com:443	50.17.237.32:443	A	Not Tested Yet	Not Tested Yet

Software Composition Analysis Test

The mobile application uses the following external and native libraries:

External

- net.i2p
- net.pwall
- org.devio
- org.webkit
- org.reactnative
- kotlinx.datetime
- kotlinx.serialization
- io.github
- defpackage
- ehn.techiop
- COSE
- com.tozny
- com.tectiv3
- com.th3rdwave
- com.horcrux
- com.lugg
- com.lwansbrough
- com.oblador
- com.bitgo
- com.reactnativecommunity
- com.swmansion
- com.securepreferences
- com.drew
- com.koushikdutta
- com.zoontek
- com.tradle

Android Native

- androidx.transition
- androidx.asynclayoutinflater
- androidx.tracing
- androidx.collection
- androidx.exifinterface
- androidx.core
- androidx.drawerlayout
- androidx.interpolator
- androidx.versionedparcelable
- androidx.activity
- androidx.viewpager
- androidx.vectordrawable
- androidx.lifecycle
- androidx.legacy
- androidx.documentfile
- androidx.annotation
- androidx.autofill
- androidx.print
- androidx.localbroadcastmanager
- androidx.cursoradapter
- androidx.biometric
- androidx.swiperefreshlayout
- androidx.fragment
- androidx.loader
- androidx.recyclerview
- androidx.arch

- com.learnium
- com.upokecenter
- com.peel
- com.taluttasgiran

- androidx.appcompat
- androidx.viewpager2
- androidx.customview
- androidx.slidingpanelayout
- androidx.coordinatorlayout
- androidx.savedstate
- androidx.cardview
- androidx.media
- bolts
- kotlin
- javax.inject
- javax.annotation
- android.support
- com.android