GREY BOX APPLICATION PT

Tarasol Android

APPLICATION PT REPORT

Effective date: 18th Nov 2025

Version #: 1.0

VERSION CONTROL

Document Control ID	EDGE-TARASOL-AND-APP
Date Issued:	18-Nov-2025
Date Effective:	
Next Review Date:	
Owner:	

VERSION HISTORY

Version	Date	Summary of Changes	Author
1.0	18-Nov-2025	Application PT Report	Saswati Bal

REVIEW AND APPROVAL

Task	Name	Title	Signature	Date
Prepared by:	Saswati Bal	Information Security Consultant		18-Nov-2025
Reviewed by:	Pinaki Bhattacharya	Information Security Consultant		18-Nov-2025
Approved by:	Mrutyunjay Sahoo	Technical Manager		18-Nov-2025

TABLE OF CONTENTS

1. EXECUTIVE SUMMARY ... 4
1.1 INTRODUCTION ... 4
1.2 SCOPE ... 4
1.3 OVERALL SUMMARY ... 5
1.4 SUMMARY OF FINDINGS ... 6
1.5 OWASP FINDING SUMMARY ... 7
1.6 OWASP MOBILE FINDING SUMMARY ... 8
1.7 TESTCASES OVERVIEWS ... 9
1.8 APPLICATION RISK DISCOVERED ... 11
1.9 TECHNICAL FINDINGS ... 13
2. KEY FINDINGS & ACTION ITEM ... 14
3. APPENDIX – 1: RISK RATING FRAMEWORK ... 25
4. APPENDIX – 2: TOOL USED ... 26

1. EXECUTIVE SUMMARY

1.1 INTRODUCTION

SecurEyes performed a Grey Box Application PT of **Tarasol** Android Application. The objective of the testing was to find out vulnerabilities that can be seen and compromised by malicious users/ adversaries over the internet.

This report documents the results of the Grey Box android Application PT activity. The activity was carried out from the source EDGE VDI. The application was found to be vulnerable to multiple security risks including **Improper Input Validation, Insecure Android Permission** etc. The following sections provide details of the scope of the application security audit and the detailed findings from the audit exercise.

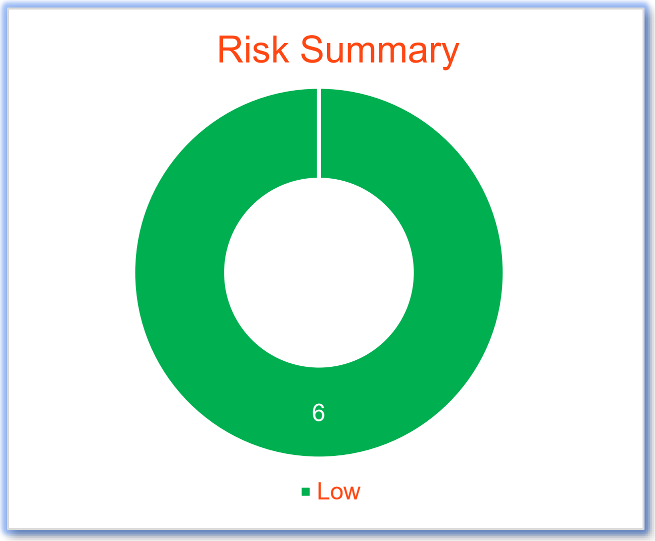
Our opinion provided in this report is valid for the period during which the PT was carried out and is based on the information and the application contents available for audit. Projection of any conclusions based on our findings for future periods and application versions is subject to the risk that the validity of such conclusions may be altered because of changes made to the network or application or system or the failure to make the changes to the network, application when required.

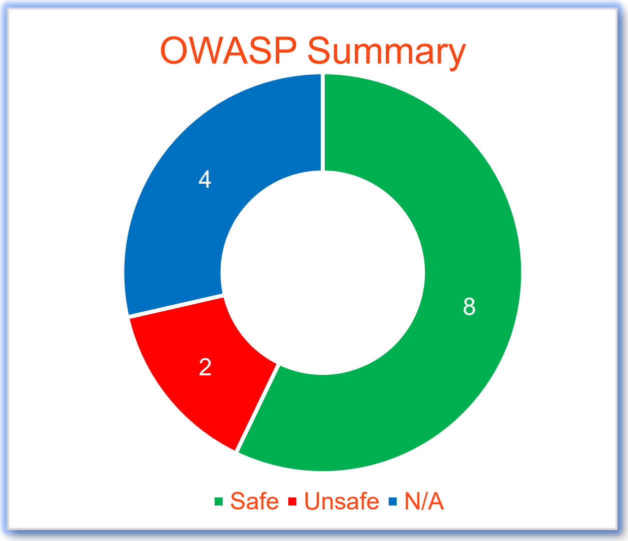
1.2 SCOPE

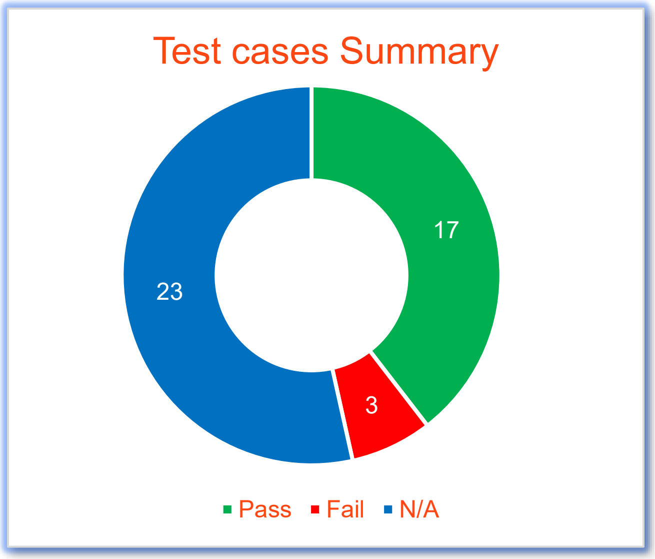
The following table provides an overview on the scope of the Grey Box application PT.

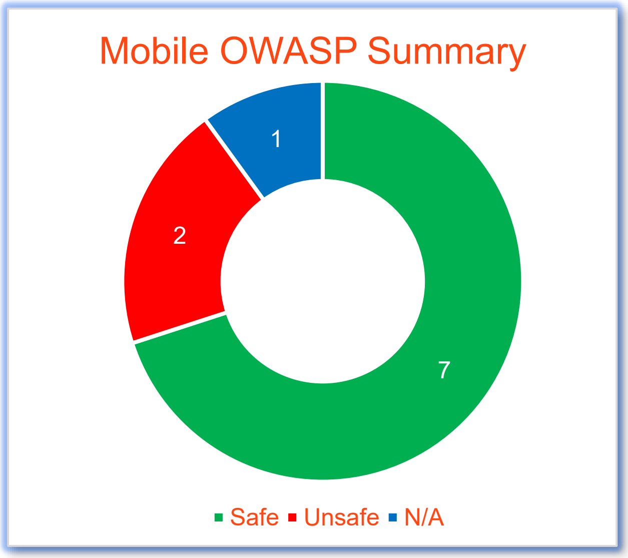
Details	
Application Name	Tarasol Android Application
Audit URL	TarasovX20250922_Test.apk
Mode of Test	Automated and Manual
Testing Type	Grey Box Android Application PT
Duration	O3 Oct 2025 to 18 Oct 2025

1.3 OVERALL SUMMARY





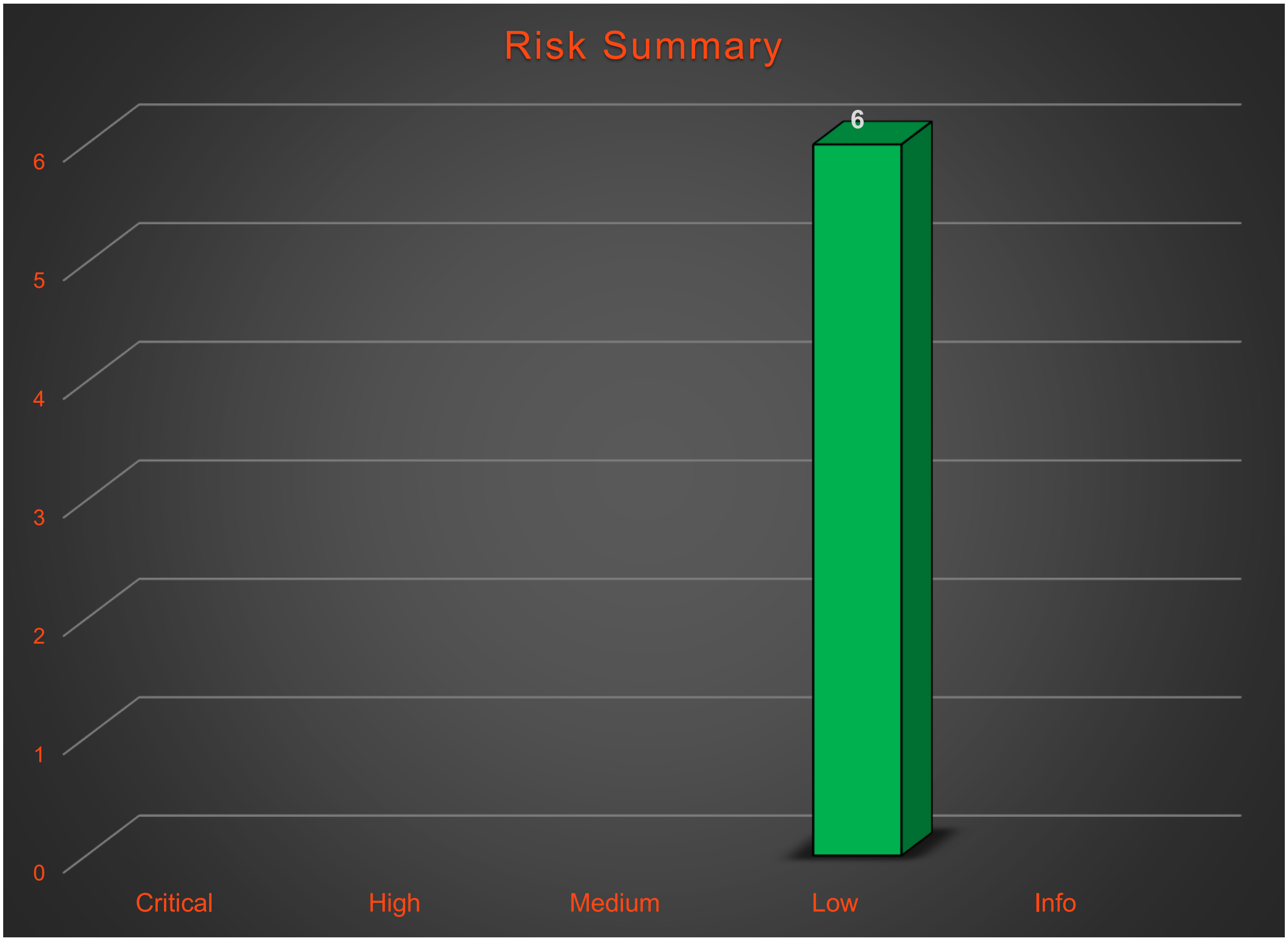




1.4 SUMMARY OF FINDINGS

We have identified 06 Low Risk vulnerabilities during the grey box android application PT activity. The following table & graph depicts the number of observed vulnerabilities across risk levels.

No of Vulnerabilities	Low
06	06



1.5 OWASP FINDING SUMMARY

SecurEyes application security PT are modelled along the methodologies specified by the Open Web Applications Security Project (OWASP). OWASP has rated the Top Ten Vulnerabilities found in web applications worldwide. The table shows how the application compares with respect to the OWASP Top 10 2021 list including additional 4 vulnerabilities as listed in previous OWASP top 10 2004, 2010, 2013 & 2017. The below status is pertaining to the test cases as applicable against a grey box mobile application PT activity.

#	Vulnerabilities	Status
1.	A1- Broken Access Control	N/A
2.	A2-Cryptographic Failure	Safe
3.	A3- Injection	Safe
4.	A4- Insecure Design	Safe
5.	A5- Security Misconfiguration	Unsafe
6.	A6- Vulnerable and Outdated Components	Safe
7.	A7- Identification and Authentication Failures	Safe
8.	A8- Software and Data Integrity Failures	Safe
9.	A9- Security Logging and Monitoring Failures	N/A
10.	A10- Server-side Request Forgery	N/A
11.	Unvalidated Input (2004)	Unsafe
12.	Denial of Service (2004)	Safe
13.	Malicious File Execution	Safe
14.	Cross Site Request Forgery	N/A

1.6 OWASP MOBILE FINDING SUMMARY

#	Vulnerabilities	Status
1.	M1- Improper Credential Usage	Safe
2.	M2- Inadequate Supply Chain Security	Safe
3.	M3- Insecure Authentication/Authorization	Safe
4.	M4- Insufficient Input/Output Validation	Unsafe
5.	M5- Insecure Communication	Safe
6.	M6- Inadequate Privacy Controls	Safe
7.	M7- Insufficient Binary Protections	NA
8.	M8- Security Misconfiguration	Unsafe
9.	M9- Insecure Data Storage	NA
10.	M10- Insufficient Cryptography	Safe

1.7 TESTCASES OVERVIEWS

No.	Test Cases	Status
1.	Testing for Insecure Password Transmission	N/A
2.	Testing for Browser Refresh Attack	N/A
3.	Testing for Shoulder Surfing of Critical Data	Pass
4.	Testing for Weak Password Policy	N/A
5.	Testing for Session Management related Vulnerabilities	Pass
6.	Testing for Cookies attributes	N/A
7.	Testing for Session Fixation	N/A
8.	Testing for Session Hijacking	N/A
9.	Failure to restrict Direct URL Access to Internal Pages	Pass
10.	Testing for Improper Redirection	N/A
11.	Testing for File Upload Functionality	Pass
12.	Testing for Logout and Browser Cache Management	N/A
13.	Improper Error Handling	Pass
14.	Testing for Critical/Sensitive Information Disclosure	Pass
15.	Testing 2nd /Multiple Factors Authentication	N/A
16.	Testing for Insecure Direct Object Reference	N/A
17.	Testing for Insecure Deserialization	N/A
18.	Testing for Privilege Escalation	N/A
19.	Testing for Insecure HTTP Methods	N/A
20.	Source Code Disclosure	Pass
21.	Testing for Host Header Manipulation	N/A
22.	Testing for Internal Path Disclosure	Pass
23.	Testing for SSL/TLS Related vulnerabilities	Pass
24.	Brute Force Attack	Pass
25.	Testing for Denial of Service	Pass
26.	Testing for CAPTCHA Related Vulnerabilities	N/A
27.	Testing for Cross site request forgery	N/A
28.	Testing for Cross Site Scripting	Pass
29.	Testing for Injection Attacks	Pass
30.	Testing for External XML Entities (XXE)	N/A

31.	Testing for Mandatory Server Configuration	Pass
32.	Testing for HTTP Splitting/Smuggling	N/A
33.	Testing for Clickjacking attack	N/A
34.	Testing for Parameter Tampering	Pass
35.	Testing for login module implementation	Pass
36.	Bypassing the Business Logic	Pass
37.	Testing for Replay Attacks	N/A
38.	Testing for OTP Related Issues	N/A
39.	Local File Inclusion	Pass
40.	Remote File Inclusion	Pass
41.	Testing for user enumeration	N/A
42.	Testing for Directory Traversal / Directory Listing	N/A
43.	An Adversary Tries to Bypass Mandatory Fields	Pass

1.8 APPLICATION RISK DISCOVERED

Following is the list of vulnerabilities observed during the grey box android application PT activity.

#	Observation and Impact	Technical Finding Name	Risk Rating	CVE/CWE Reference No
1.	The application is not validating the user inputs in various input fields as a result malicious user may insert malicious scripts into the application and compromise it.	Improper Input Validation	Low	**CWE-20**
2.	The mobile application's 'apk' file has set insecure permissions in the Manifest File such as • READ_EXTERNAL_STORAGE • WRITE_EXTERNAL_STORAGE As a result, it allows an attacker to do reverse engineering and initiate attacks based on the given permission.	Insecure Android Permission	Low	**CWE-926**
3.	The application components such as Exported is set to "True." which allows an attacker to enter in developer mode of the application and do reverse engineering.	Insecure Manifest File	Low	**CWE-287**
4.	The application discloses SQL Queries in java class files which can potentially lead to SQL injection vulnerability in the mobile application.	SQL Queries present in Class File	Low	**CWE-200**
5.	The application is using weak SHA1 and MD5 hashing algorithm to transmit password form client to server as a result an attacker may be able to extract password and perform further targeted attacks.	Weak Hashing Algorithm	Low	**CWE-916**

6.	The application has enabled WebView which may allow an attacker to execute malicious scripts to capture the cookie of the application. This will be possible if JavaScript execution is not restricted in web view.	Insecure WebView Enabled	**Low**	**CWE-749**

1.9 TECHNICAL FINDINGS

As a result of the comprehensive security PT conducted on **‘Tarasol’** Android application, the following technical vulnerabilities were discovered.

1. Improper Input Validation
2. Insecure Android Permission
3. Insecure Manifest File
4. SQL Queries present in Class File
5. Weak Hashing Algorithm
6. Insecure WebView Enabled

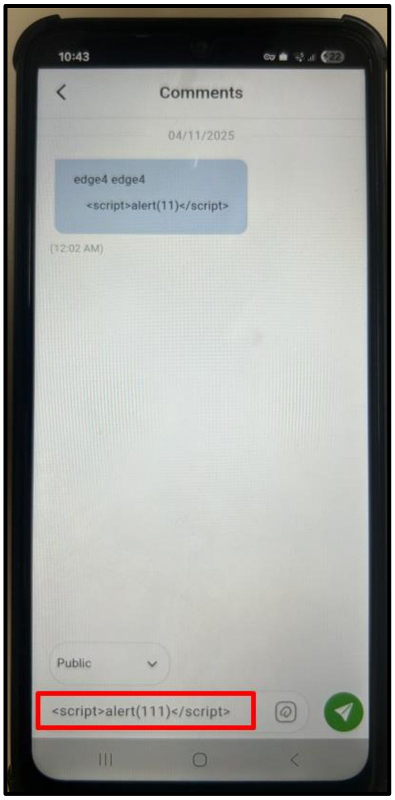
2. KEY FINDINGS & ACTION ITEM

EDGE-TARASOL-AND-APP-01. The application is not validating the user inputs in various input fields- Improper input Validation.

Observation Details and Impact	Technical Finding Name	Risk Rating	Impact on Application	Probability of Attack
The application is not validating the user inputs in various input fields as a result malicious user may insert malicious scripts into the application and compromise it.	Improper Input Validation	Low	Low	Low
Built APK	[TarasovX20250922_Test.apk]			
CVE Reference No	**CWE-20**			

How to Test:

Step #1: A malicious user navigates to the **"Comment"** page of the application and enters a malicious script into an input field.



10:43

Comments

04/11/2025

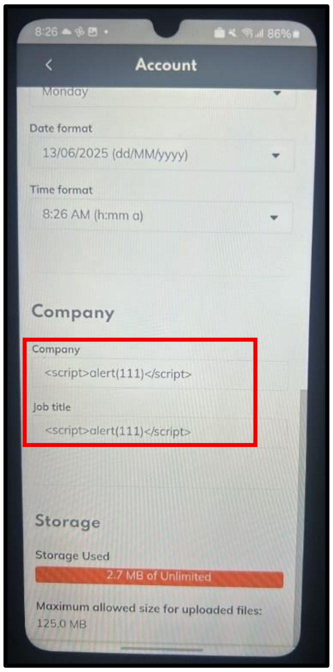
edge4 edge4
<script>alert(11)</script>

(12:02 AM)

Public

<script>alert(111)</script>

Step #2: As shown in the screenshot below, the malicious script is stored and displayed in the field.



Monday

Date format

13/06/2025 (dd/MM/yyyy)

Time format

8:26 AM (h:mm a)

Company

Company

<script>alert(111)</script>

Job title

<script>alert(111)</script>

Storage

Storage Used

2.7 MB of Unlimited

Maximum allowed size for uploaded files:

125.0 MB

Recommendations:

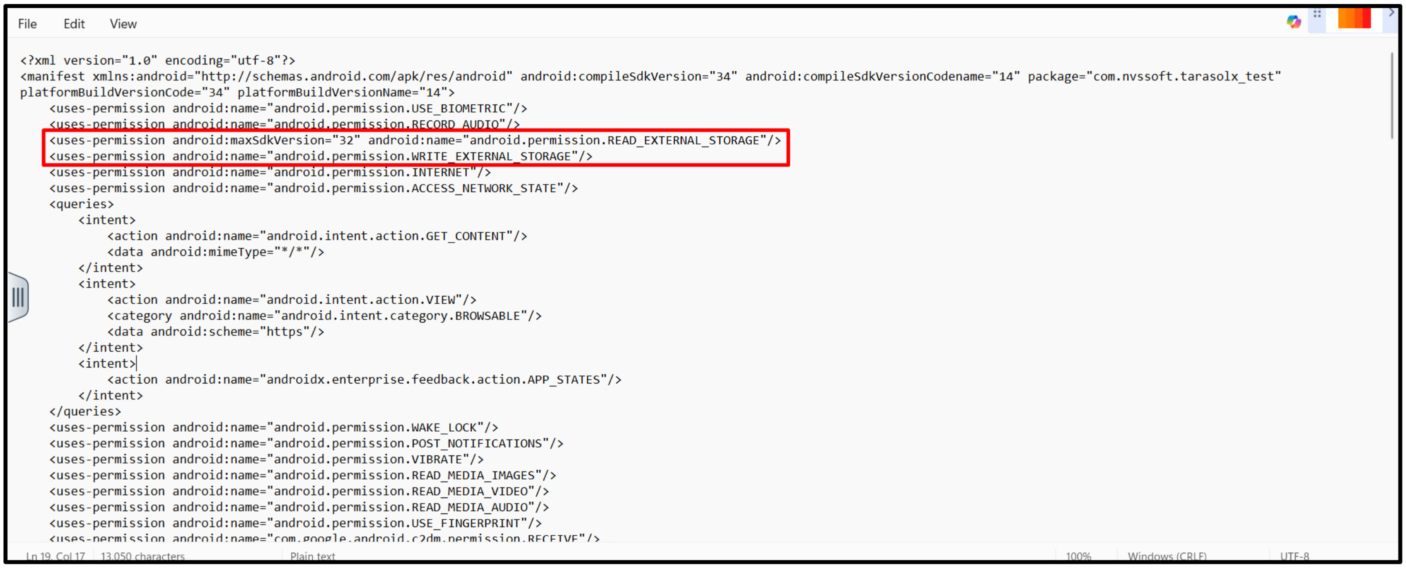
- The application should properly validate all the client requests against business logic implemented in the application.

EDGE-TARASOL-AND-APP-02. The mobile application’s ‘APK’ file has set insecure permission in the Manifest File - Insecure Android Permission.

Observation Details and Impact	Technical Finding Name	Risk Rating	Impact on Application	Probability of Attack
The mobile application’s ‘apk’ file has set insecure permissions in the Manifest File such as • READ_EXTERNAL_STORAGE • WRITE_EXTERNAL_STORAGE As a result, it allows an attacker to do reverse engineering and initiate attacks based on the given permission.	Insecure Android Permission	Low	Low	Low
Built APK	TarasovX20250922_Test.apk			
CVE Reference No	**CWE-926**			

How to Test:

Step #1: From below screenshot, it is observed that the application has application allows attackers to take screenshots of the application.



```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android" android:compileSdkVersion="34" android:compileSdkVersionCodename="14" package="com.nvssoft.tarasolx_test"
platformBuildVersionCode="34" platformBuildVersionName="14">
    <uses-permission android:name="android.permission.USE_BIOMETRIC"/>
    <uses-permission android:name="android.permission.RECORD_AUDIO"/>
    <uses-permission android:maxSdkVersion="32" android:name="android.permission.READ_EXTERNAL_STORAGE"/>
    <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
    <uses-permission android:name="android.permission.INTERNET"/>
    <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"/>
    <queries>
        <intent>
            <action android:name="android.intent.action.GET_CONTENT"/>
            <data android:mimeType="*/*"/>
        </intent>
        <intent>
            <action android:name="android.intent.action.VIEW"/>
            <category android:name="android.intent.category.BROWSABLE"/>
            <data android:scheme="https"/>
        </intent>
        <intent>
            <action android:name="androidx.enterprise.feedback.action.APP_STATES"/>
        </intent>
    </queries>
    <uses-permission android:name="android.permission.WAKE_LOCK"/>
    <uses-permission android:name="android.permission.POST_NOTIFICATIONS"/>
    <uses-permission android:name="android.permission.VIBRATE"/>
    <uses-permission android:name="android.permission.READ_MEDIA_IMAGES"/>
    <uses-permission android:name="android.permission.READ_MEDIA_VIDEO"/>
    <uses-permission android:name="android.permission.READ_MEDIA_AUDIO"/>
    <uses-permission android:name="android.permission.USE_FINGERPRINT"/>
    <uses-permission android:name="com.google.android.c2dm.permission.RECEIVE"/>
```

Recommendations:

- ➢ It is recommended that the insecure permission should not be mentioned in the Android Manifest File.
- ➢ The permissions provided to the app should be based on the business requirement.

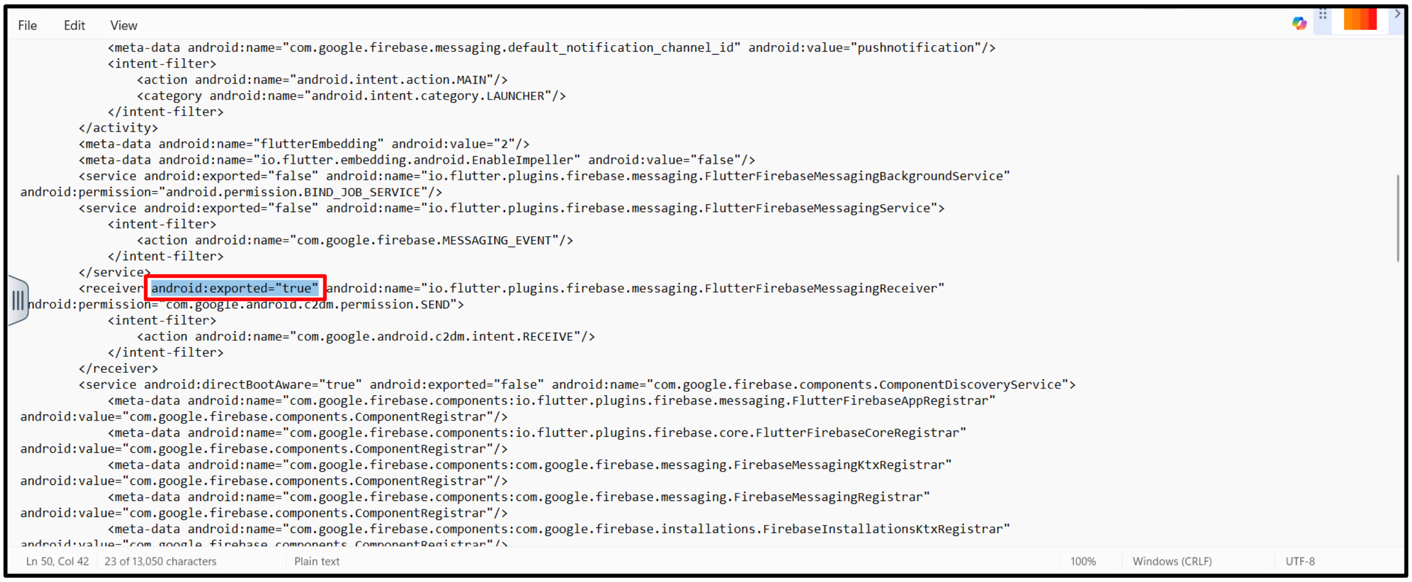
EDGE-TARASOL-AND-APP-03. The application component such as Exported is set to “True” - Insecure Manifest File.

Observation Details and Impact	Technical Finding Name	Risk Rating	Impact on Application	Probability of Attack
The application components such as Exported is set to “True.” which allows an attacker to enter in developer mode of the application and do reverse engineering.	Insecure Manifest File	Low	Low	Low
Built APK	TarasovX20250922_Test.apk			
CVE Reference No	**CWE-287**			

How to Test:

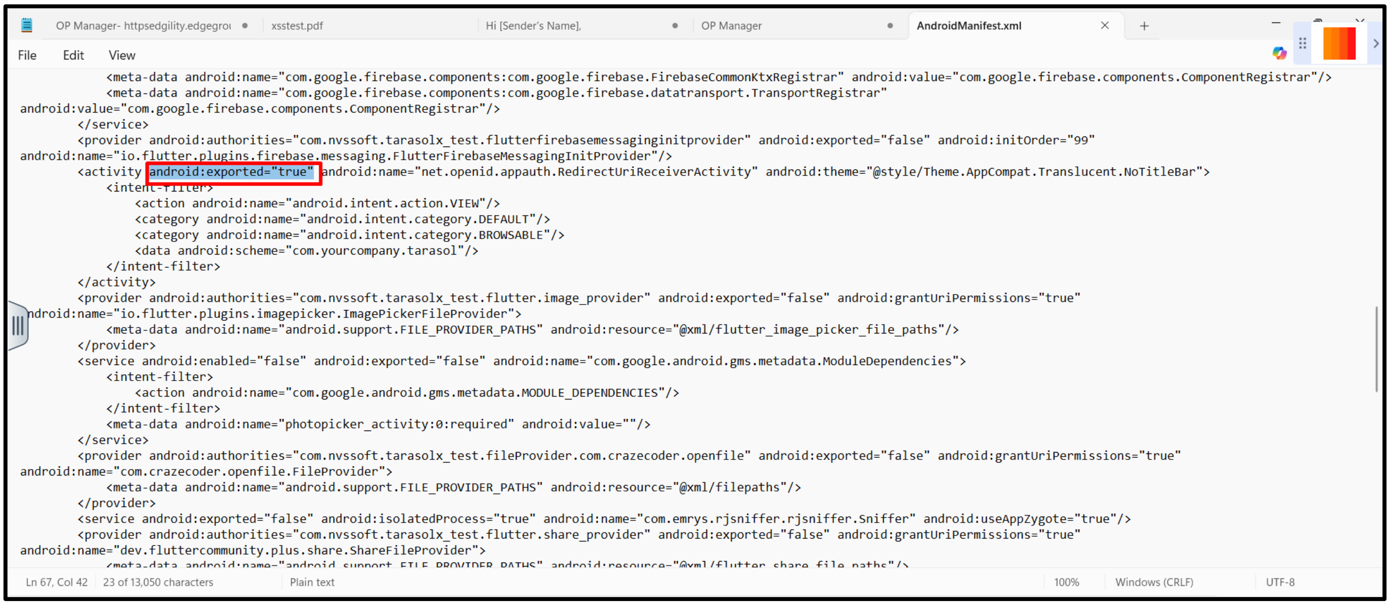
Step 1: Skimming through the manifest file an attacker observes that the exported flag is set to “True’’ as shown below:

Screenshot #1:



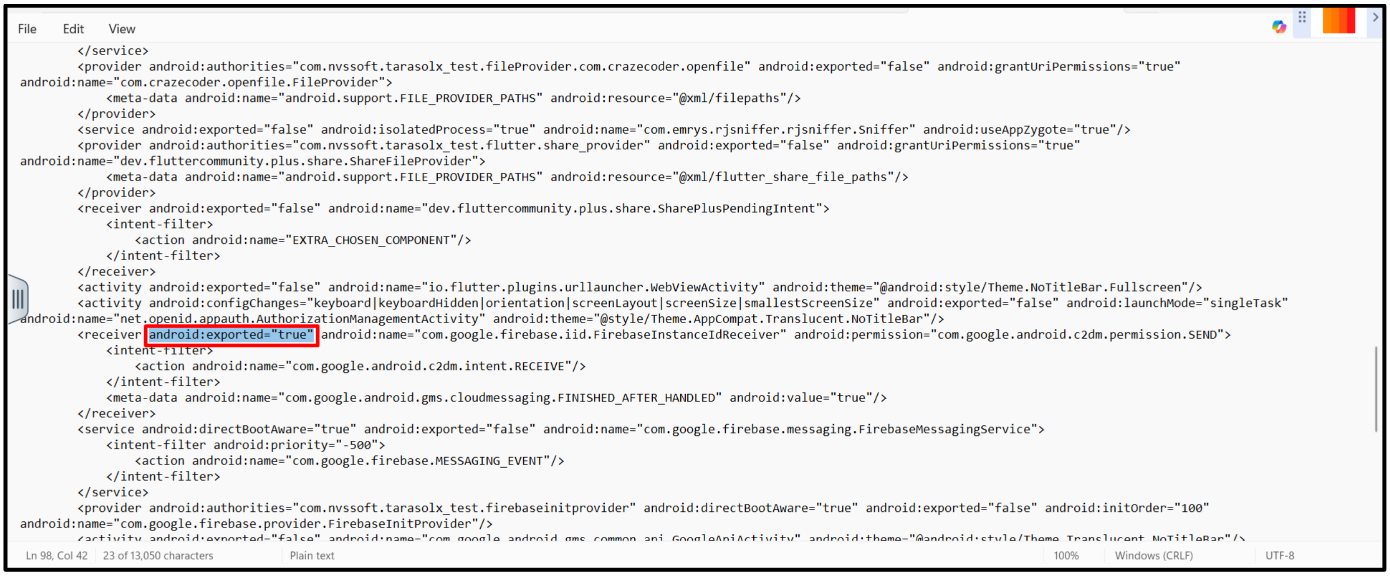
```
            <meta-data android:name="com.google.firebase.messaging.default_notification_channel_id" android:value="pushnotification"/>
            <intent-filter>
                <action android:name="android.intent.action.MAIN"/>
                <category android:name="android.intent.category.LAUNCHER"/>
            </intent-filter>
        </activity>
        <meta-data android:name="flutterEmbedding" android:value="2"/>
        <meta-data android:name="io.flutter.embedding.android.EnableImpeller" android:value="false"/>
        <service android:exported="false" android:name="io.flutter.plugins.firebase.messaging.FlutterFirebaseMessagingBackgroundService"
android:permission="android.permission.BIND_JOB_SERVICE"/>
        <service android:exported="false" android:name="io.flutter.plugins.firebase.messaging.FlutterFirebaseMessagingService">
            <intent-filter>
                <action android:name="com.google.firebase.MESSAGING_EVENT"/>
            </intent-filter>
        </service>
        <receiver android:exported="true" android:name="io.flutter.plugins.firebase.messaging.FlutterFirebaseMessagingReceiver"
ndroid:permission="com.google.android.c2dm.permission.SEND">
            <intent-filter>
                <action android:name="com.google.android.c2dm.intent.RECEIVE"/>
            </intent-filter>
        </receiver>
        <service android:directBootAware="true" android:exported="false" android:name="com.google.firebase.components.ComponentDiscoveryService">
            <meta-data android:name="com.google.firebase.components:io.flutter.plugins.firebase.messaging.FlutterFirebaseAppRegistrar"
android:value="com.google.firebase.components.ComponentRegistrar"/>
            <meta-data android:name="com.google.firebase.components:io.flutter.plugins.firebase.core.FlutterFirebaseCoreRegistrar"
android:value="com.google.firebase.components.ComponentRegistrar"/>
            <meta-data android:name="com.google.firebase.components:com.google.firebase.messaging.FirebaseMessagingKtxRegistrar"
android:value="com.google.firebase.components.ComponentRegistrar"/>
            <meta-data android:name="com.google.firebase.components:com.google.firebase.messaging.FirebaseMessagingRegistrar"
android:value="com.google.firebase.components.ComponentRegistrar"/>
            <meta-data android:name="com.google.firebase.components:com.google.firebase.installations.FirebaseInstallationsKtxRegistrar"
android:value="com.google.firebase.components.ComponentRegistrar"/>
```

Screenshot #2:



```
            <meta-data android:name="com.google.firebase.components:com.google.firebase.FirebaseCommonKtxRegistrar" android:value="com.google.firebase.components.ComponentRegistrar"/>
            <meta-data android:name="com.google.firebase.components:com.google.firebase.datatransport.TransportRegistrar"
android:value="com.google.firebase.components.ComponentRegistrar"/>
        </service>
        <provider android:authorities="com.nvssoft.tarasolx_test.flutterfirebasemessaginginitprovider" android:exported="false" android:initOrder="99"
android:name="io.flutter.plugins.firebase.messaging.FlutterFirebaseMessagingInitProvider"/>
        <activity android:exported="true" android:name="net.openid.appauth.RedirectUriReceiverActivity" android:theme="@style/Theme.AppCompat.Translucent.NoTitleBar">
            <intent-filter>
                <action android:name="android.intent.action.VIEW"/>
                <category android:name="android.intent.category.DEFAULT"/>
                <category android:name="android.intent.category.BROWSABLE"/>
                <data android:scheme="com.yourcompany.tarasol"/>
            </intent-filter>
        </activity>
        <provider android:authorities="com.nvssoft.tarasolx_test.flutter.image_provider" android:exported="false" android:grantUriPermissions="true"
ndroid:name="io.flutter.plugins.imagepicker.ImagePickerFileProvider">
            <meta-data android:name="android.support.FILE_PROVIDER_PATHS" android:resource="@xml/flutter_image_picker_file_paths"/>
        </provider>
        <service android:enabled="false" android:exported="false" android:name="com.google.android.gms.metadata.ModuleDependencies">
            <intent-filter>
                <action android:name="com.google.android.gms.metadata.MODULE_DEPENDENCIES"/>
            </intent-filter>
            <meta-data android:name="photopicker_activity:0:required" android:value=""/>
        </service>
        <provider android:authorities="com.nvssoft.tarasolx_test.fileProvider.com.crazecoder.openfile" android:exported="false" android:grantUriPermissions="true"
android:name="com.crazecoder.openfile.FileProvider">
            <meta-data android:name="android.support.FILE_PROVIDER_PATHS" android:resource="@xml/filepaths"/>
        </provider>
        <service android:exported="false" android:isolatedProcess="true" android:name="com.emrys.rjsniffer.rjsniffer.Sniffer" android:useAppZygote="true"/>
        <provider android:authorities="com.nvssoft.tarasolx_test.flutter.share_provider" android:exported="false" android:grantUriPermissions="true"
android:name="dev.fluttercommunity.plus.share.ShareFileProvider">
            <meta-data android:name="android.support.FILE_PROVIDER_PATHS" android:resource="@xml/flutter_share_file_paths"/>
```

Screenshot #3:



```
        </service>
        <provider android:authorities="com.nvssoft.tarasolx_test.fileProvider.com.crazecoder.openfile" android:exported="false" android:grantUriPermissions="true"
android:name="com.crazecoder.openfile.FileProvider">
            <meta-data android:name="android.support.FILE_PROVIDER_PATHS" android:resource="@xml/filepaths"/>
        </provider>
        <service android:exported="false" android:isolatedProcess="true" android:name="com.emrys.rjsniffer.rjsniffer.Sniffer" android:useAppZygote="true"/>
        <provider android:authorities="com.nvssoft.tarasolx_test.flutter.share_provider" android:exported="false" android:grantUriPermissions="true"
android:name="dev.fluttercommunity.plus.share.ShareFileProvider">
            <meta-data android:name="android.support.FILE_PROVIDER_PATHS" android:resource="@xml/flutter_share_file_paths"/>
        </provider>
        <receiver android:exported="false" android:name="dev.fluttercommunity.plus.share.SharePlusPendingIntent">
            <intent-filter>
                <action android:name="EXTRA_CHOSEN_COMPONENT"/>
            </intent-filter>
        </receiver>
        <activity android:exported="false" android:name="io.flutter.plugins.urllauncher.WebViewActivity" android:theme="@android:style/Theme.NoTitleBar.Fullscreen"/>
        <activity android:configChanges="keyboard|keyboardHidden|orientation|screenLayout|screenSize|smallestScreenSize" android:exported="false" android:launchMode="singleTask"
android:name="net.openid.appauth.AuthorizationManagementActivity" android:theme="@style/Theme.AppCompat.Translucent.NoTitleBar"/>
        <receiver android:exported="true" android:name="com.google.firebase.iid.FirebaseInstanceIdReceiver" android:permission="com.google.android.c2dm.permission.SEND">
            <intent-filter>
                <action android:name="com.google.android.c2dm.intent.RECEIVE"/>
            </intent-filter>
            <meta-data android:name="com.google.android.gms.cloudmessaging.FINISHED_AFTER_HANDLED" android:value="true"/>
        </receiver>
        <service android:directBootAware="true" android:exported="false" android:name="com.google.firebase.messaging.FirebaseMessagingService">
            <intent-filter android:priority="-500">
                <action android:name="com.google.firebase.MESSAGING_EVENT"/>
            </intent-filter>
        </service>
        <provider android:authorities="com.nvssoft.tarasolx_test.firebaseinitprovider" android:directBootAware="true" android:exported="false" android:initOrder="100"
android:name="com.google.firebase.provider.FirebaseInitProvider"/>
        <activity android:exported="false" android:name="com.google.android.gms.common.api.GoogleApiActivity" android:theme="@android:style/Theme.Translucent.NoTitleBar"/>
```

Recommendations:

- The application should set the exported value to false i.e., android: exported = “False”.

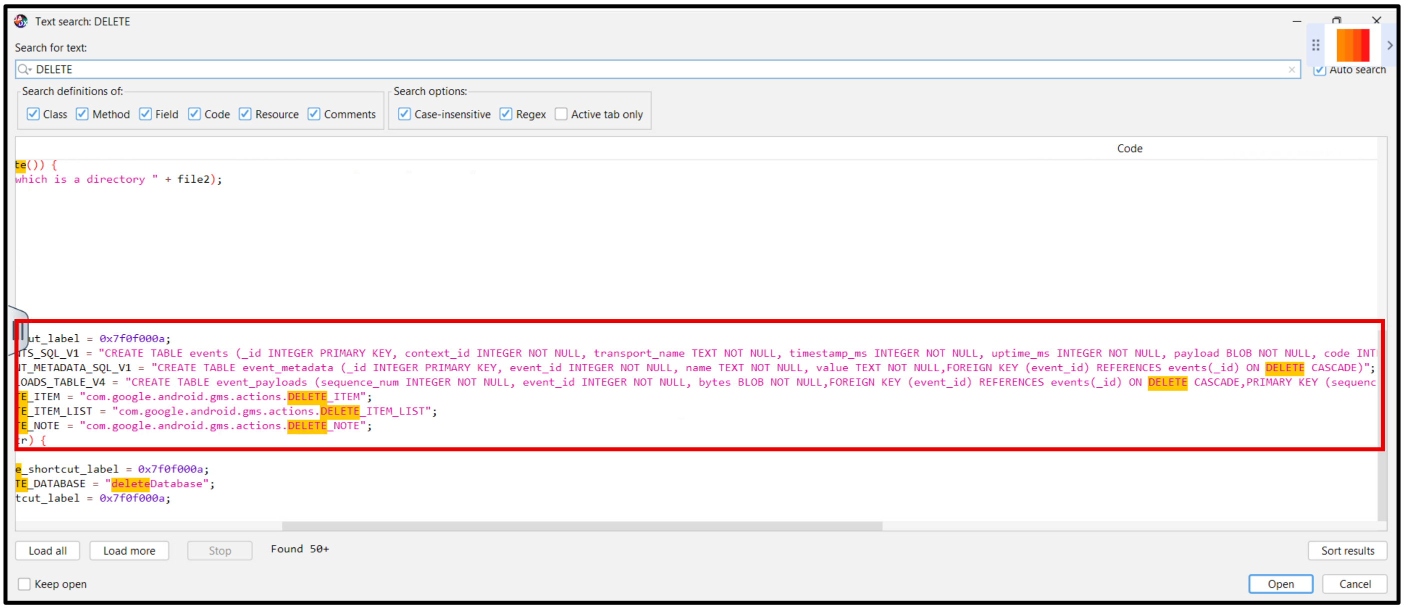
EDGE-TARASOL-AND-APP-04. The application discloses SQL Queries in java class file – SQL Queries present in Class File

Observation Details and Impact	Technical Finding Name	Risk Rating	Impact on Application	Probability of Attack
The application discloses SQL Queries in java class files which can potentially lead to SQL injection vulnerability in the mobile application.	SQL Queries present in Class File	Low	Low	Low
Built APK	TarasovX20250922_Test.apk			
CVE Reference No	**CWE-200**			

How to Test:

Step #1: In the class files of the application, it can be observed that the application is disclosing SQL Queries as shown in the below screenshot:

Screenshot #1:



DELETE

☑ Class ☑ Method ☑ Field ☑ Code ☑ Resource ☑ Comments

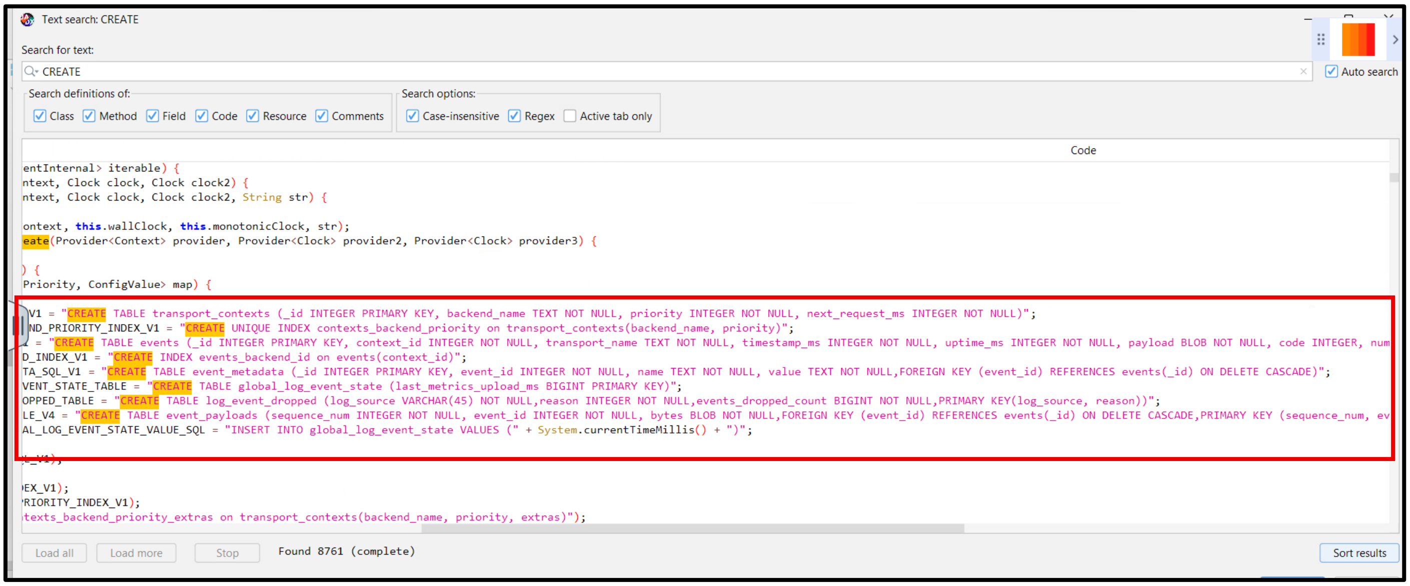
☑ Case-insensitive ☑ Regex □ Active tab only

```
te()) {
which is a directory " + file2);

ut_label = 0x7f0f000a;
TS_SQL_V1 = "CREATE TABLE events (_id INTEGER PRIMARY KEY, context_id INTEGER NOT NULL, transport_name TEXT NOT NULL, timestamp_ms INTEGER NOT NULL, uptime_ms INTEGER NOT NULL, payload BLOB NOT NULL, code INT
T_METADATA_SQL_V1 = "CREATE TABLE event_metadata (_id INTEGER PRIMARY KEY, event_id INTEGER NOT NULL, name TEXT NOT NULL, value TEXT NOT NULL,FOREIGN KEY (event_id) REFERENCES events(_id) ON DELETE CASCADE)";
OADS_TABLE_V4 = "CREATE TABLE event_payloads (sequence_num INTEGER NOT NULL, event_id INTEGER NOT NULL, bytes BLOB NOT NULL,FOREIGN KEY (event_id) REFERENCES events(_id) ON DELETE CASCADE,PRIMARY KEY (sequenc
E_ITEM = "com.google.android.gms.actions.DELETE_ITEM";
E_ITEM_LIST = "com.google.android.gms.actions.DELETE_ITEM_LIST";
E_NOTE = "com.google.android.gms.actions.DELETE_NOTE";
r) {

e_shortcut_label = 0x7f0f000a;
TE_DATABASE = "deleteDatabase";
tcut_label = 0x7f0f000a;
```

Found 50+

□ Keep open

Screenshot #2:



Search for text: CREATE

Search definitions of: ☑ Class ☑ Method ☑ Field ☑ Code ☑ Resource ☑ Comments

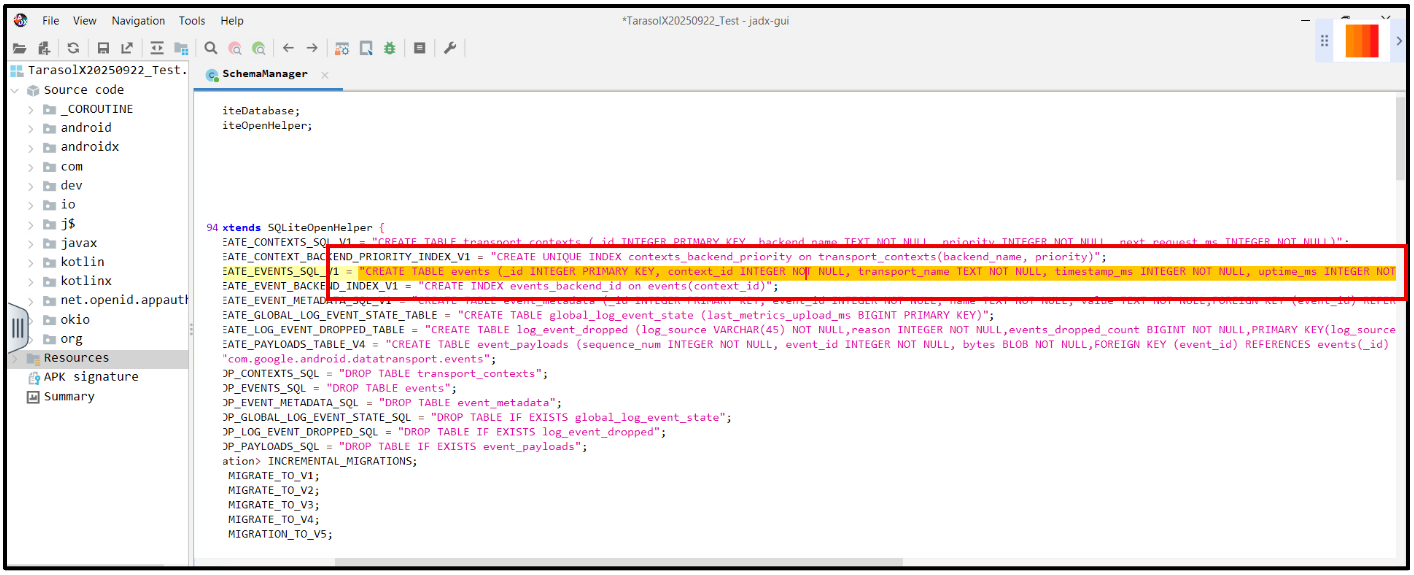
Search options: ☑ Case-insensitive ☑ Regex □ Active tab only

☑ Auto search

```
entInternal> iterable) {
ntext, Clock clock, Clock clock2) {
ntext, Clock clock, Clock clock2, String str) {

ontext, this.wallClock, this.monotonicClock, str);
eate(Provider<Context> provider, Provider<Clock> provider2, Provider<Clock> provider3) {

) {
Priority, ConfigValue> map) {

V1 = "CREATE TABLE transport_contexts (_id INTEGER PRIMARY KEY, backend_name TEXT NOT NULL, priority INTEGER NOT NULL, next_request_ms INTEGER NOT NULL)";
ND_PRIORITY_INDEX_V1 = "CREATE UNIQUE INDEX contexts_backend_priority on transport_contexts(backend_name, priority)";
1 = "CREATE TABLE events (_id INTEGER PRIMARY KEY, context_id INTEGER NOT NULL, transport_name TEXT NOT NULL, timestamp_ms INTEGER NOT NULL, uptime_ms INTEGER NOT NULL, payload BLOB NOT NULL, code INTEGER, num
D_INDEX_V1 = "CREATE INDEX events_backend_id on events(context_id)";
TA_SQL_V1 = "CREATE TABLE event_metadata (_id INTEGER PRIMARY KEY, event_id INTEGER NOT NULL, name TEXT NOT NULL, value TEXT NOT NULL,FOREIGN KEY (event_id) REFERENCES events(_id) ON DELETE CASCADE)";
VENT_STATE_TABLE = "CREATE TABLE global_log_event_state (last_metrics_upload_ms BIGINT PRIMARY KEY)";
OPPED_TABLE = "CREATE TABLE log_event_dropped (log_source VARCHAR(45) NOT NULL,reason INTEGER NOT NULL,events_dropped_count BIGINT NOT NULL,PRIMARY KEY(log_source, reason))";
LE_V4 = "CREATE TABLE event_payloads (sequence_num INTEGER NOT NULL, event_id INTEGER NOT NULL, bytes BLOB NOT NULL,FOREIGN KEY (event_id) REFERENCES events(_id) ON DELETE CASCADE,PRIMARY KEY (sequence_num, ev
AL_LOG_EVENT_STATE_VALUE_SQL = "INSERT INTO global_log_event_state VALUES (" + System.currentTimeMillis() + ")";

L_V1);

EX_V1);
RIORITY_INDEX_V1);
texts_backend_priority_extras on transport_contexts(backend_name, priority, extras)");
```

Found 8761 (complete)

Screenshot #3:



```
iteDatabase;
iteOpenHelper;




94 xtends SQLiteOpenHelper {
EATE_CONTEXTS_SQL_V1 = "CREATE TABLE transport_contexts (_id INTEGER PRIMARY KEY, backend_name TEXT NOT NULL, priority INTEGER NOT NULL, next_request_ms INTEGER NOT NULL)";
EATE_CONTEXT_BACKEND_PRIORITY_INDEX_V1 = "CREATE UNIQUE INDEX contexts_backend_priority on transport_contexts(backend_name, priority)";
EATE_EVENTS_SQL_V1 = "CREATE TABLE events (_id INTEGER PRIMARY KEY, context_id INTEGER NOT NULL, transport_name TEXT NOT NULL, timestamp_ms INTEGER NOT NULL, uptime_ms INTEGER NOT
EATE_EVENT_BACKEND_INDEX_V1 = "CREATE INDEX events_backend_id on events(context_id)";
EATE_EVENT_METADATA_SQL_V1 = "CREATE TABLE event_metadata (_id INTEGER PRIMARY KEY, event_id INTEGER NOT NULL, name TEXT NOT NULL, value TEXT NOT NULL,FOREIGN KEY (event_id) REFER
EATE_GLOBAL_LOG_EVENT_STATE_TABLE = "CREATE TABLE global_log_event_state (last_metrics_upload_ms BIGINT PRIMARY KEY)";
EATE_LOG_EVENT_DROPPED_TABLE = "CREATE TABLE log_event_dropped (log_source VARCHAR(45) NOT NULL,reason INTEGER NOT NULL,events_dropped_count BIGINT NOT NULL,PRIMARY KEY(log_source
EATE_PAYLOADS_TABLE_V4 = "CREATE TABLE event_payloads (sequence_num INTEGER NOT NULL, event_id INTEGER NOT NULL, bytes BLOB NOT NULL,FOREIGN KEY (event_id) REFERENCES events(_id)
"com.google.android.datatransport.events";
OP_CONTEXTS_SQL = "DROP TABLE transport_contexts";
OP_EVENTS_SQL = "DROP TABLE events";
OP_EVENT_METADATA_SQL = "DROP TABLE event_metadata";
OP_GLOBAL_LOG_EVENT_STATE_SQL = "DROP TABLE IF EXISTS global_log_event_state";
OP_LOG_EVENT_DROPPED_SQL = "DROP TABLE IF EXISTS log_event_dropped";
OP_PAYLOADS_SQL = "DROP TABLE IF EXISTS event_payloads";
ation> INCREMENTAL_MIGRATIONS;
 MIGRATE_TO_V1;
 MIGRATE_TO_V2;
 MIGRATE_TO_V3;
 MIGRATE_TO_V4;
 MIGRATION_TO_V5;
```

Recommendations:

- It is recommended that the application should not disclose SQL related Queries in the java .class file.

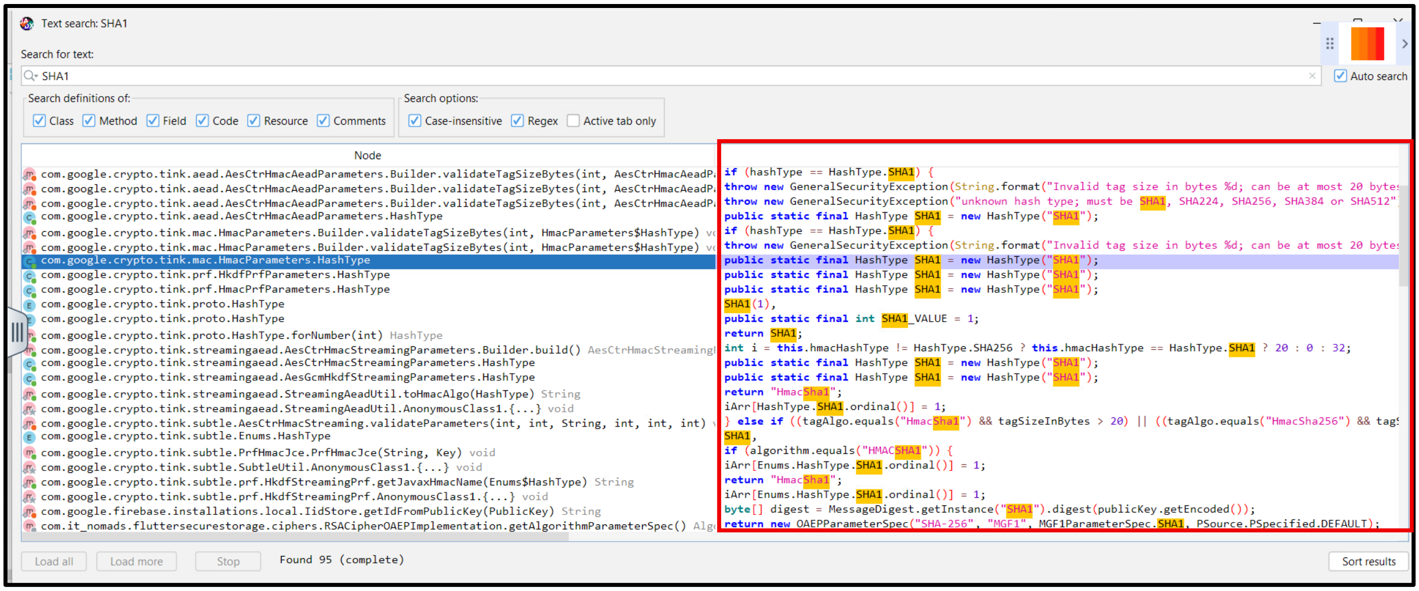
EDGE-TARASOL-AND-APP-05. The application is using weak SHA1 and MD5 hashing algorithm – Weak Hashing Algorithm.

Observation Details and Impact	Technical Finding Name	Risk Rating	Impact on Application	Probability of Attack
The application is using weak SHA1 and MD5 hashing algorithm to transmit password form client to server as a result an attacker may be able to extract password and perform further targeted attacks.	Weak Hashing Algorithm	Low	Low	Low
Built APK	TarasovX20250922_Test.apk			
CVE Reference No	**CWE-916**			

How to Test:

Step #1: From the below screenshot, it can be observed that the application is using weak hashing algorithm like SHA1 and MD5.

Screenshot #1:



Text search: SHA1

Search for text: SHA1

Search definitions of: ☑ Class ☑ Method ☑ Field ☑ Code ☑ Resource ☑ Comments

Search options: ☑ Case-insensitive ☑ Regex ☐ Active tab only

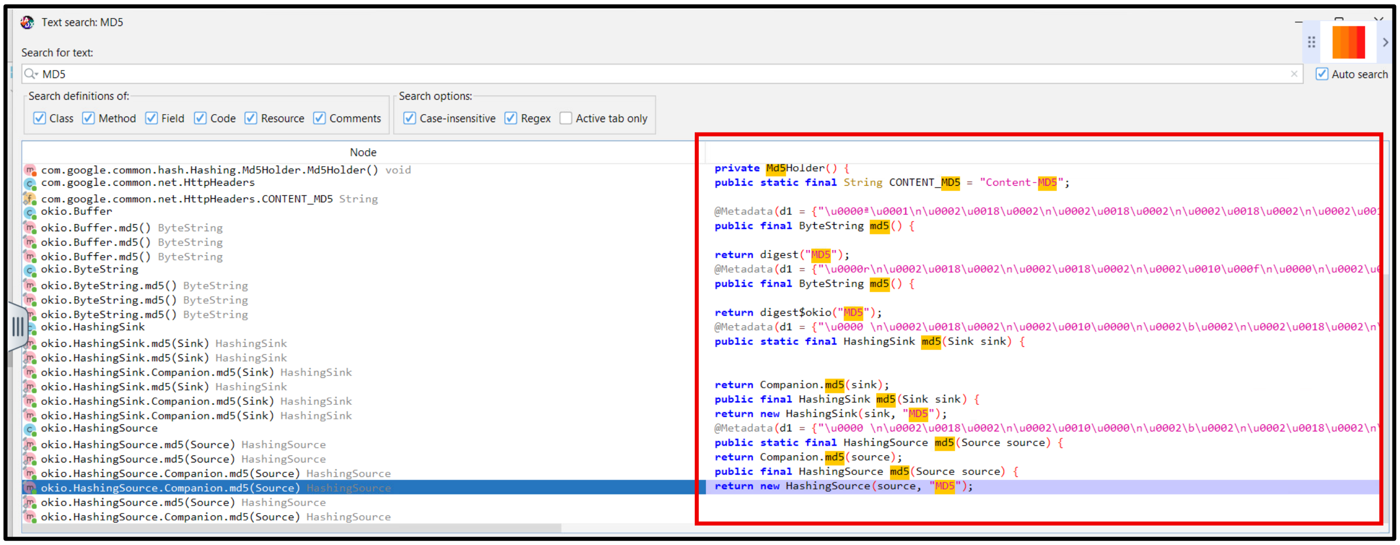
☑ Auto search

```
Node
com.google.crypto.tink.aead.AesCtrHmacAeadParameters.Builder.validateTagSizeBytes(int, AesCtrHmacAeadPa
com.google.crypto.tink.aead.AesCtrHmacAeadParameters.Builder.validateTagSizeBytes(int, AesCtrHmacAeadPa
com.google.crypto.tink.aead.AesCtrHmacAeadParameters.Builder.validateTagSizeBytes(int, AesCtrHmacAeadPa
com.google.crypto.tink.aead.AesCtrHmacAeadParameters.HashType
com.google.crypto.tink.mac.HmacParameters.Builder.validateTagSizeBytes(int, HmacParameters$HashType) vo
com.google.crypto.tink.mac.HmacParameters.Builder.validateTagSizeBytes(int, HmacParameters$HashType) vo
com.google.crypto.tink.mac.HmacParameters.HashType
com.google.crypto.tink.prf.HkdfPrfParameters.HashType
com.google.crypto.tink.prf.HmacPrfParameters.HashType
com.google.crypto.tink.proto.HashType
com.google.crypto.tink.proto.HashType
com.google.crypto.tink.proto.HashType.forNumber(int) HashType
com.google.crypto.tink.streamingaead.AesCtrHmacStreamingParameters.Builder.build() AesCtrHmacStreamingP
com.google.crypto.tink.streamingaead.AesCtrHmacStreamingParameters.HashType
com.google.crypto.tink.streamingaead.AesGcmHkdfStreamingParameters.HashType
com.google.crypto.tink.streamingaead.StreamingAeadUtil.toHmacAlgo(HashType) String
com.google.crypto.tink.streamingaead.StreamingAeadUtil.AnonymousClass1.{...} void
com.google.crypto.tink.subtle.AesCtrHmacStreaming.validateParameters(int, int, String, int, int, int) v
com.google.crypto.tink.subtle.Enums.HashType
com.google.crypto.tink.subtle.PrfHmacJce.PrfHmacJce(String, Key) void
com.google.crypto.tink.subtle.SubtleUtil.AnonymousClass1.{...} void
com.google.crypto.tink.subtle.prf.HkdfStreamingPrf.getJavaxHmacName(Enums$HashType) String
com.google.crypto.tink.subtle.prf.HkdfStreamingPrf.AnonymousClass1.{...} void
com.google.firebase.installations.local.IidStore.getIdFromPublicKey(PublicKey) String
com.it_nomads.fluttersecurestorage.ciphers.RSACipherOAEPImplementation.getAlgorithmParameterSpec() Alg
```

```
if (hashType == HashType.SHA1) {
throw new GeneralSecurityException(String.format("Invalid tag size in bytes %d; can be at most 20 byte
throw new GeneralSecurityException("unknown hash type; must be SHA1, SHA224, SHA256, SHA384 or SHA512"
public static final HashType SHA1 = new HashType("SHA1");
if (hashType == HashType.SHA1) {
throw new GeneralSecurityException(String.format("Invalid tag size in bytes %d; can be at most 20 byte
public static final HashType SHA1 = new HashType("SHA1");
public static final HashType SHA1 = new HashType("SHA1");
public static final HashType SHA1 = new HashType("SHA1");
SHA1(1),
public static final int SHA1_VALUE = 1;
return SHA1;
int i = this.hmacHashType != HashType.SHA256 ? this.hmacHashType == HashType.SHA1 ? 20 : 0 : 32;
public static final HashType SHA1 = new HashType("SHA1");
public static final HashType SHA1 = new HashType("SHA1");
return "HmacSha1";
iArr[HashType.SHA1.ordinal()] = 1;
} else if ((tagAlgo.equals("HmacSha1") && tagSizeInBytes > 20) || ((tagAlgo.equals("HmacSha256") && tag
SHA1,
if (algorithm.equals("HMACSHA1")) {
iArr[Enums.HashType.SHA1.ordinal()] = 1;
return "HmacSha1";
iArr[Enums.HashType.SHA1.ordinal()] = 1;
byte[] digest = MessageDigest.getInstance("SHA1").digest(publicKey.getEncoded());
return new OAEPParameterSpec("SHA-256", "MGF1", MGF1ParameterSpec.SHA1, PSource.PSpecified.DEFAULT);
```

Found 95 (complete)

Screenshot #2:



Text search: MD5

Search for text:

MD5

☑ Auto search

Search definitions of: ☑ Class ☑ Method ☑ Field ☑ Code ☑ Resource ☑ Comments

Search options: ☑ Case-insensitive ☑ Regex ☐ Active tab only

Node

```
com.google.common.hash.Hashing.Md5Holder.Md5Holder() void
com.google.common.net.HttpHeaders
com.google.common.net.HttpHeaders.CONTENT_MD5 String
okio.Buffer
okio.Buffer.md5() ByteString
okio.Buffer.md5() ByteString
okio.Buffer.md5() ByteString
okio.ByteString
okio.ByteString.md5() ByteString
okio.ByteString.md5() ByteString
okio.ByteString.md5() ByteString
okio.HashingSink
okio.HashingSink.md5(Sink) HashingSink
okio.HashingSink.md5(Sink) HashingSink
okio.HashingSink.Companion.md5(Sink) HashingSink
okio.HashingSink.md5(Sink) HashingSink
okio.HashingSink.Companion.md5(Sink) HashingSink
okio.HashingSink.Companion.md5(Sink) HashingSink
okio.HashingSource
okio.HashingSource.md5(Source) HashingSource
okio.HashingSource.md5(Source) HashingSource
okio.HashingSource.Companion.md5(Source) HashingSource
okio.HashingSource.Companion.md5(Source) HashingSource
okio.HashingSource.md5(Source) HashingSource
okio.HashingSource.Companion.md5(Source) HashingSource
```

```
private Md5Holder() {
public static final String CONTENT_MD5 = "Content-MD5";

@Metadata(d1 = {"\u0000ª\u0001\n\u0002\u0018\u0002\n\u0002\u0018\u0002\n\u0002\u0018\u0002\n\u0002\u00
public final ByteString md5() {

return digest("MD5");
@Metadata(d1 = {"\u0000r\n\u0002\u0018\u0002\n\u0002\u0018\u0002\n\u0002\u0010\u000f\n\u0000\n\u0002\u
public final ByteString md5() {

return digest$okio("MD5");
@Metadata(d1 = {"\u0000 \n\u0002\u0018\u0002\n\u0002\u0010\u0000\n\u0002\b\u0002\n\u0002\u0018\u0002\n
public static final HashingSink md5(Sink sink) {


return Companion.md5(sink);
public final HashingSink md5(Sink sink) {
return new HashingSink(sink, "MD5");
@Metadata(d1 = {"\u0000 \n\u0002\u0018\u0002\n\u0002\u0010\u0000\n\u0002\b\u0002\n\u0002\u0018\u0002\n
public static final HashingSource md5(Source source) {
return Companion.md5(source);
public final HashingSource md5(Source source) {
return new HashingSource(source, "MD5");
```

Recommendations:

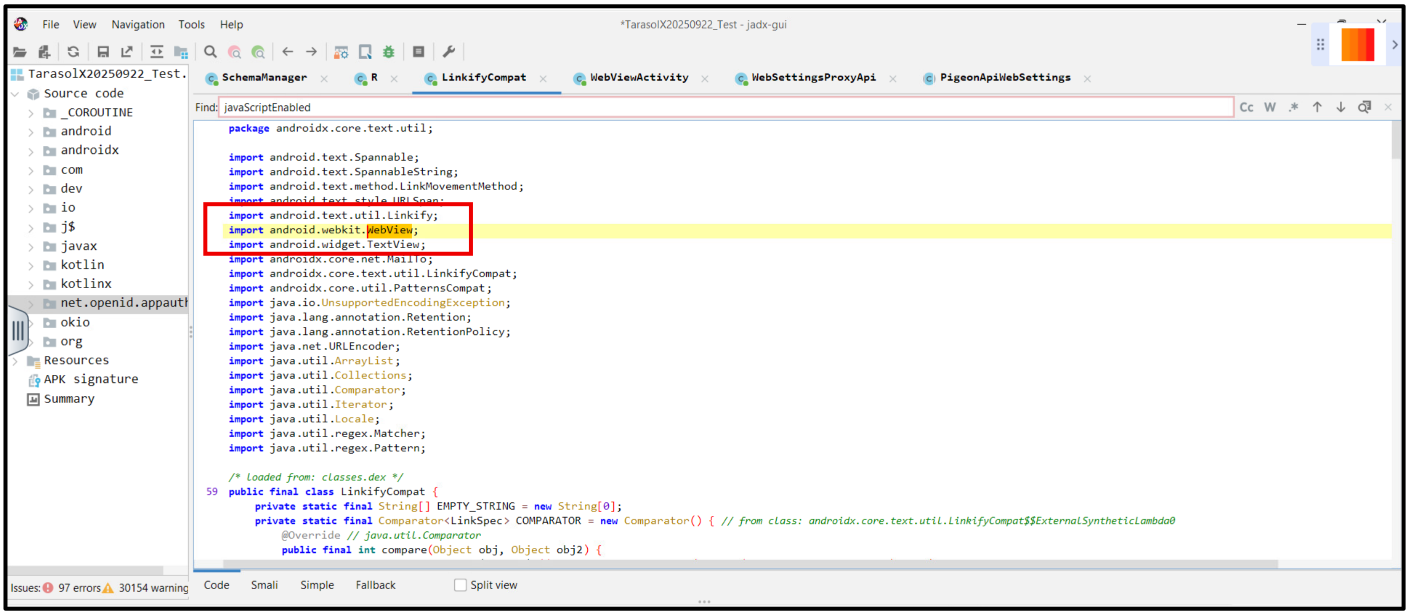
- Password should travel in SHA256/512 or encrypted form respectively.
- Password should be always hashed with random salt and salt should be unique for every request.
- Salt should be generated at server side and properly validated.

EDGE-TARASOL-AND-APP-06. The application has enabled WebView – Insecure WebView Enabled.

Observation Details and Impact	Technical Finding Name	Risk Rating	Impact on Application	Probability of Attack
The application has enabled WebView which may allow an attacker to execute malicious scripts to capture the cookie of the application. This will be possible if JavaScript execution is not restricted in web view.	Insecure WebView Enabled	**Low**	**Low**	**Low**
Built APK	TarasovX20250922_Test.apk			
CVE Reference No	**CWE-749**			

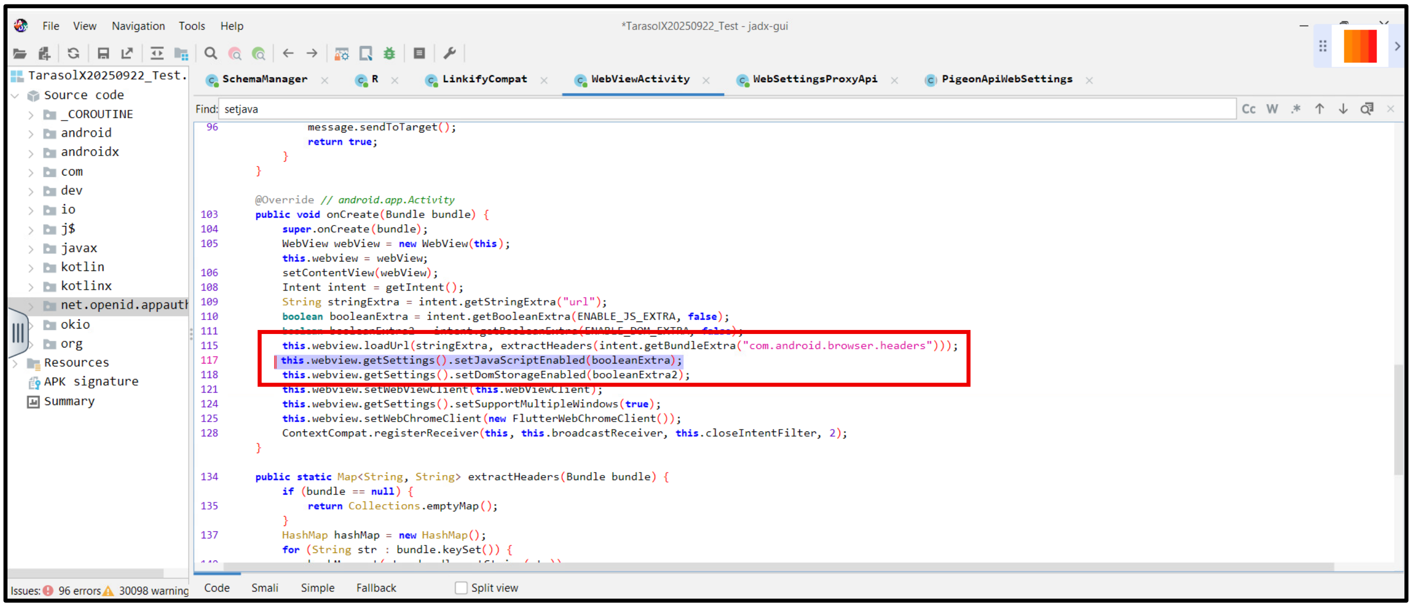
How to Test:

Step #1: From the below screenshot, it can be observed from the below screenshots that the application has allowed WebView in its class file.



Find: javaScriptEnabled

```
package androidx.core.text.util;

import android.text.Spannable;
import android.text.SpannableString;
import android.text.method.LinkMovementMethod;
import android.text.style.URLSpan;
import android.text.util.Linkify;
import android.webkit.WebView;
import android.widget.TextView;
import androidx.core.net.MailTo;
import androidx.core.text.util.LinkifyCompat;
import androidx.core.util.PatternsCompat;
import java.io.UnsupportedEncodingException;
import java.lang.annotation.Retention;
import java.lang.annotation.RetentionPolicy;
import java.net.URLEncoder;
import java.util.ArrayList;
import java.util.Collections;
import java.util.Comparator;
import java.util.Iterator;
import java.util.Locale;
import java.util.regex.Matcher;
import java.util.regex.Pattern;

/* loaded from: classes.dex */
public final class LinkifyCompat {
    private static final String[] EMPTY_STRING = new String[0];
    private static final Comparator<LinkSpec> COMPARATOR = new Comparator() { // from class: androidx.core.text.util.LinkifyCompat$$ExternalSyntheticLambda0
        @Override // java.util.Comparator
        public final int compare(Object obj, Object obj2) {
```



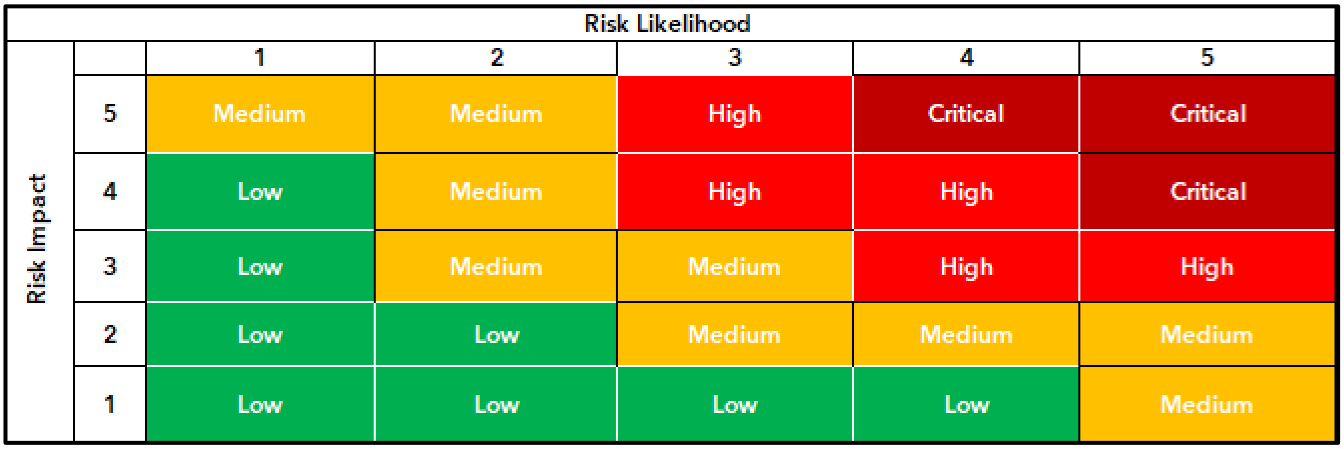
```java
            message.sendToTarget();
            return true;
        }
    }

    @Override // android.app.Activity
    public void onCreate(Bundle bundle) {
        super.onCreate(bundle);
        WebView webView = new WebView(this);
        this.webview = webView;
        setContentView(webView);
        Intent intent = getIntent();
        String stringExtra = intent.getStringExtra("url");
        boolean booleanExtra = intent.getBooleanExtra(ENABLE_JS_EXTRA, false);
        boolean booleanExtra2 = intent.getBooleanExtra(ENABLE_DOM_EXTRA, false);
        this.webview.loadUrl(stringExtra, extractHeaders(intent.getBundleExtra("com.android.browser.headers")));
        this.webview.getSettings().setJavaScriptEnabled(booleanExtra);
        this.webview.getSettings().setDomStorageEnabled(booleanExtra2);
        this.webview.setWebViewClient(this.webViewClient);
        this.webview.getSettings().setSupportMultipleWindows(true);
        this.webview.setWebChromeClient(new FlutterWebChromeClient());
        ContextCompat.registerReceiver(this, this.broadcastReceiver, this.closeIntentFilter, 2);
    }

    public static Map<String, String> extractHeaders(Bundle bundle) {
        if (bundle == null) {
            return Collections.emptyMap();
        }
        HashMap hashMap = new HashMap();
        for (String str : bundle.keySet()) {
```

Recommendations:

- It is recommended to disable JavaScript for web view.

3. APPENDIX – 1: RISK RATING FRAMEWORK

The Severity for each finding in this report is based on the Impact and Ease of exploitation of the vulnerability. Here's a guide to interpreting the Severity:



		Risk Likelihood				
		1	2	3	4	5
Risk Impact	5	Medium	Medium	High	Critical	Critical
	4	Low	Medium	High	High	Critical
	3	Low	Medium	Medium	High	High
	2	Low	Low	Medium	Medium	Medium
	1	Low	Low	Low	Low	Medium

Low Risk:

Three types of vulnerabilities get this rating: first, vulnerabilities that can only be exploited locally, secondly, vulnerabilities that are easy to exploit but have a low impact, and finally vulnerabilities that reveal information that might aid an attacker in crafting an attack easily. Descriptive error or informational messages are an example of this type. Clear text protocols being enabled and unnecessary services running on the target system would typically come under this category. Resumes of employees that discuss internal architecture and features they have configured on security devices are another example. Please fix them before the next testing cycle.

4. APPENDIX – 2: TOOL USED

The following is a list of tools that were used during the application PT activity.

1. **Burp Suite:** Burp Proxy is a tool which intercepts traffic between client and server. It is available for download at http://www.portswigger.net/.
2. **APK Easy Tool**
3. **Jdax**