

ASSESSMENT REPORT
MOBILE APPLICATION

One by IFG

Prepared by



Table of Contents

1. Pendahuluan.....	3
2. Code Protection (Perlindungan Statis)	4
a. Before Hardened	4
b. After Hardened.....	5
3. Runtime Protection (Perlindungan Saat Aplikasi Berjalan)	6
a. Before Hardened	6
b. After Hardened.....	8
4. Integrity Protection.....	10
a. Before Hardened	10
b. After Hardened.....	12
5. Man-in-the-Middle (MITM) Protection	15
a. Before Hardened	15
a. After Hardened.....	16
6. Penyimpanan Data di Shared Preferences	18
a. Before Hardened	18
b. After Hardened.....	19
7. Resource Protection.....	20
a. Before Hardened	20
b. After Hardened.....	20
8. Kesimpulan.....	22

1. Pendahuluan

Aplikasi One by IFG merupakan salah satu kanal layanan digital yang digunakan untuk mendukung aktivitas dan layanan kepada pengguna, sehingga memproses serta menyimpan berbagai data penting di dalamnya. Kondisi tersebut menuntut penerapan aspek keamanan aplikasi yang memadai guna memastikan **kerahasiaan, integritas, dan keandalan** aplikasi tetap terjaga.

Seiring dengan meningkatnya pemanfaatan aplikasi mobile, potensi ancaman terhadap aplikasi juga semakin beragam, baik yang menargetkan struktur aplikasi, proses aplikasi saat berjalan, maupun mekanisme pertukaran data yang digunakan. Oleh karena itu, diperlukan suatu proses evaluasi keamanan yang terstruktur untuk memastikan bahwa aplikasi telah dilengkapi dengan mekanisme perlindungan yang sesuai terhadap berbagai skenario ancaman yang mungkin terjadi.

Asesmen keamanan ini dilakukan untuk mengevaluasi kondisi keamanan aplikasi **sebelum dan sesudah** penerapan mekanisme *mobile application hardening* menggunakan SecIron. Ruang lingkup asesmen mencakup beberapa aspek utama, antara lain **perlindungan kode aplikasi (code protection)**, **perlindungan pada saat aplikasi berjalan (runtime protection)**, **pemeriksaan integritas aplikasi**, **perlindungan terhadap intersepsi komunikasi aplikasi (MITM protection)**, serta **keamanan penyimpanan data secara lokal**.

Melalui asesmen ini, diharapkan dapat diperoleh gambaran yang lebih komprehensif mengenai **postur keamanan aplikasi** serta tingkat efektivitas mekanisme proteksi yang diterapkan dalam meningkatkan ketahanan aplikasi terhadap potensi ancaman keamanan.

2. Code Protection (Perlindungan Statis)

a. Before Hardened

Pada tahap *reverse engineering* menggunakan tools seperti **JADX**, diketahui bahwa aplikasi **belum menerapkan mekanisme proteksi kode secara optimal**. Kondisi ini menyebabkan struktur source code masih dapat dibaca dengan relatif mudah melalui proses dekompilasi.

Sebagai dampaknya, sejumlah informasi sensitif berhasil diidentifikasi secara langsung di dalam kode aplikasi, antara lain:

- URL endpoint API
- API key

Eksposur informasi tersebut meningkatkan risiko keamanan, khususnya terhadap aktivitas *reverse engineering* lanjutan serta potensi penyalahgunaan kredensial API oleh pihak tidak berwenang. Dalam skenario terburuk, hal ini dapat dimanfaatkan untuk melakukan akses ilegal maupun pengiriman data ke server tanpa kontrol yang semestinya.

[Gambar: Cuplikan source code hasil dekompilasi dari JADX]

```

*com.one.ifg v1.4.1_antisplit - jadx-gui
File View Navigation Tools Plugins Help
New version 1.5.3 available!
Source code
BuildConfig
2
3 /* loaded from: classes3.dex */
4 public final class BuildConfig {
5     public static final String ANDROID_VERSION_CODE = "106";
6     public static final String ANDROID_VERSION_NAME = "1.4.1";
7     public static final String API_BASE_URL = "https://one.ifg.id/api";
8     public static final String APPCENTER_APPNAME_ANDROID = "OneByIfg";
9     public static final String APPCENTER_APPNAME_IOS = "OneByIfg-1";
10    public static final String APPCENTER_KEY_ID = "9cf5bd8e5becb5d5a92817f607d73f862db9";
11    public static final String APPCENTER_OWNER_ANDROID = "dev-one-ifg-life.id";
12    public static final String APPCENTER_OWNER_IOS = "app.dev-ifg-life.id";
13    public static final String APPLICATION_ID = "com.one.ifg";
14    public static final String APPSFLYER_APP_ID = "1564248452";
15    public static final String APPSFLYER_KEY = "mZNKxWbhrRufWwPjJ3";
16    public static final String APPSFLYER_URL = "http://one.ifg.id/api";
17    public static final String CODE_PUSH_DEPLOYMENT_KEY = "jeWW7ugXfajPi1TMSFhNcaBgsixdhu80FeEB3";
18    public static final String DD_APPLICATION_ID = "38977436-d9fb-4bc4-9988-0b0894b15e0e";
19    public static final String DD_CLIENT_TOKEN = "pub480d39cfa2bf18e598726068e311dc71";
20    public static final boolean DEBUG_APP = false;
21    public static final String DISPLAY_NAME = "One By Ifg";
22    public static final String ENV_NAME = "prod";
23    public static final String FB_APP_ID = "369646873951078";
24    public static final String FB_CLIENT_TOKEN = "abfc5608e9b6950f000326dc8f2fcfe";
25    public static final String INSIDER_PARTNER_NAME = "ifgindonesia";
26    public static final String IOS_BUNDLE_IDENTIFIER = "com.one.ifg";
27    public static final String IOS_CODE_PUSH_DEPLOYMENT_KEY = "KHWTpjZ5kPZ0aw7zmjWuCblI3sg2bu80FeEB3";
28    public static final String IOS_VERSION_CODE = "106";
29    public static final String IOS_VERSION_NAME = "1.4.1";
30    public static final String LIVESAVER_APP_ID = "106";
31    public static final Boolean IS_NEW_ARCHITECTURE_ENABLED = false;
32    public static final String LIVESAVER_RIPPLAY = "https://life.id/api/v1/customer/content/riplays";
33    public static final String LIVESAVER_TINC = "https://life.id/api/v1/customer/content/terms-conditions?lang=en";
34    public static final String LIFE_URL = "https://life.id/api";
35    public static final String LYFARIS_LICENSE_URL = "https://apl.advance.ai/openapi/liveness/v1/auth-license";
36    public static final String LYFARIS_RESULT_URL = "https://apl.advance.ai/openapi/liveness/v3/detection-result";
37    public static final String LYFARIS_SDK_KEY = "80af1cb6b704d9c9c";
38    public static final String LYFARIS_SECRET_KEY = "6c06ccbd62903fcd";
39    public static final String LYFARIS_X_ADVAT_KEY = "c43b4a475f80f379";
40    public static final String MYAPP_UPLOAD_KEY_APP = "one-by-ifg";
41    public static final String MYAPP_UPLOAD_PASSWORD = "P@ssw0rd09!";
42    public static final String MYAPP_UPLOAD_STORE_FILE = "one-by-ifg.keystore";
43    public static final String MYAPP_UPLOAD_STORE_PASSWORD = "P@ssw0rd09!";
44    public static final String PRIVACY_POLICY_URL = "https://aman.ifg-life.id/syarat-dan-ketentuan";
45    public static final String RECAPTCHA_DOMAIN = "https://one.ifg.id";
46    public static final String VERSION_NAME = "1.4.1";
47    public static final String WEB_BASE_URL = "https://one.ifg.id";
48    public static final String WEB_BASE_URL_AUTH = "https://one.ifg.id/apps/authentication";
49    public static final String WEB_BASE_URL_HEALTH = "https://one.ifg.id/apps/health";
50    public static final String WEB_BASE_URL_WEALTH = "https://one.ifg.id/apps/wealth";
51
52
53
54
55
}

```

Issues: 297 errors 5685 warnings

Code Small Simple Fallback Split view

b. After Hardened

Setelah proses *hardening* diterapkan menggunakan **SecIron**, terlihat perbedaan yang signifikan dibandingkan kondisi sebelumnya. Kode aplikasi tidak hanya mengalami proses **obfuscation**, tetapi juga telah melalui mekanisme **enkripsi dan penyembunyian (code hiding)**.

Sebagai hasilnya, kelas-kelas yang sebelumnya dapat diidentifikasi melalui proses dekompilasi kini **tidak lagi terlihat**. Struktur kode menjadi jauh lebih bersih, di mana hanya komponen yang berkaitan langsung dengan kebutuhan **UI** yang masih dapat terdeteksi. Informasi sensitif serta logika inti aplikasi berhasil disembunyikan secara efektif, sehingga **tidak dapat dianalisis secara langsung melalui tools reverse engineering seperti JADX**.

Kondisi ini secara signifikan menurunkan risiko *reverse engineering*, penyalahgunaan kredensial, serta kebocoran informasi sensitif dari sisi aplikasi.

[Gambar: Cuplikan source code hasil dekompilasi dari JADX]

*com.one.ifg v1.4.1_antisplit_20260209102111_sec - jadx-gui

New version 1.5.3 available

File View Navigation Tools Plugins Help

Inputs Files Scripts

Source code

com

- AppGuard
- one.ifg
- anim
- animator
- array
- attr
- bool
- color
- dimen
- drawable
- font
- id**
- integer
- interpolator
- layout
- menu
- mipmap
- plurals
- R
- raw
- string
- style
- xml

kotlin.coroutines.jvm.internal

Resources APK signature Summary

package com.one.ifg;

(JADINFO: This class is generated by JADX *)*

public final class R {

public static final class anim {

public static final int abc_fade_in = 0x7f010000;

public static final int abc_fade_out = 0x7f010001;

public static final int abc_grow_fade_in_from_bottom = 0x7f010002;

public static final int abc_grow_fade_in_from_top = 0x7f010003;

public static final int abc_popup_exit = 0x7f010004;

public static final int abc_shrink_fade_out_from_bottom = 0x7f010005;

public static final int abc_slide_in_bottom = 0x7f010006;

public static final int abc_slide_in_top = 0x7f010007;

public static final int abc_slide_out_bottom = 0x7f010008;

public static final int abc_slide_out_top = 0x7f010009;

public static final int abc_tooltip_enter = 0x7f01000a;

public static final int abc_tooltip_exit = 0x7f01000b;

public static final int btn_checkbox_to_checked_box_inner_merged_animation = 0x7f01000c;

public static final int btn_checkbox_to_checked_box_outer_merged_animation = 0x7f01000d;

public static final int btn_checkbox_to_checked_icon_null_animation = 0x7f01000e;

public static final int btn_checkbox_to_unchecked_box_inner_merged_animation = 0x7f01000f;

public static final int btn_checkbox_to_unchecked_check_path_merged_animation = 0x7f010010;

public static final int btn_checkbox_to_unchecked_icon_null_animation = 0x7f010011;

public static final int btn_radio_to_off_mtrl_dot_group_animation = 0x7f010012;

public static final int btn_radio_to_off_mtrl_ring_outer_animation = 0x7f010013;

public static final int btn_radio_to_on_mtrl_dot_group_animation = 0x7f010014;

public static final int btn_radio_to_on_mtrl_ring_outer_animation = 0x7f010015;

public static final int catalyst_fade_in = 0x7f010016;

public static final int catalyst_fade_out = 0x7f010017;

public static final int catalyst_push_up_in = 0x7f010018;

public static final int catalyst_push_up_out = 0x7f010019;

public static final int catalyst_push_up_up = 0x7f01001a;

public static final int catalyst_slide_down = 0x7f01001b;

public static final int catalyst_slide_up = 0x7f01001c;

public static final int design_bottom_sheet_slide_in = 0x7f01001d;

public static final int design_bottom_sheet_slide_out = 0x7f01001e;

public static final int design_snackbar_in = 0x7f010020;

public static final int design_snackbar_out = 0x7f010021;

public static final int fragment_fast_out_extra_slow_in = 0x7f010022;

public static final int ins_anim_xcv_fadein = 0x7f010023;

public static final int ins_anim_xcv_fadeout = 0x7f010024;

public static final int ins_anim_xcv_slidedown = 0x7f010025;

public static final int ins_anim_xcv_slidetop = 0x7f010026;

public static final int mtrl_bottom_sheet_slide_in = 0x7f010027;

public static final int mtrl_bottom_sheet_slide_out = 0x7f010028;

public static final int mtrl_card_lowers_interpolator = 0x7f010029;

public static final int oneshot_fade_in = 0x7f01002a;

public static final int oneshot_fade_out = 0x7f01002b;

public static final int rns_default_enter_in = 0x7f01002c;

public static final int rns_default_enter_out = 0x7f01002d;

public static final int rns_default_exit_in = 0x7f01002e;

public static final int rns_default_exit_out = 0x7f01002f;

} // class anim

} // class R

Issues: 1 warnings

Code Small Simple Fallback Split view

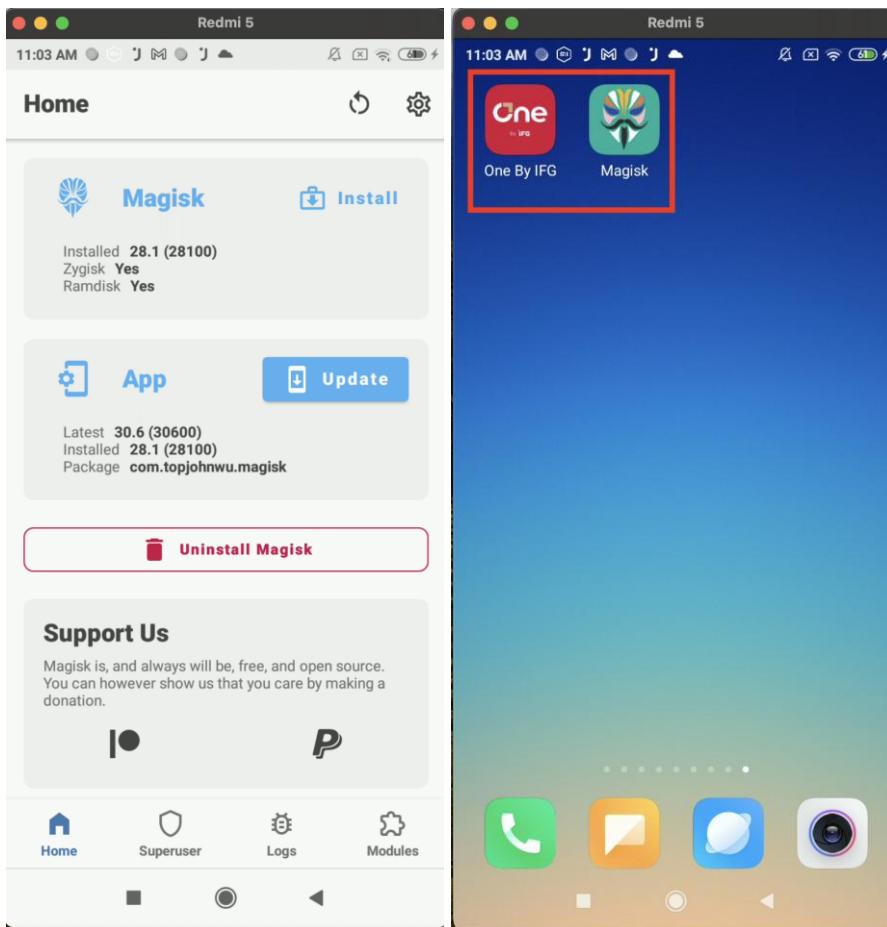
3. Runtime Protection (Perlindungan Saat Aplikasi Berjalan)

a. Before Hardened

Pada kondisi sebelum penerapan *runtime protection*, aplikasi **belum dilengkapi dengan mekanisme deteksi dan mitigasi terhadap lingkungan perangkat yang tidak aman**. Hal ini dibuktikan dengan aplikasi yang masih dapat berjalan secara normal pada perangkat yang telah di-root menggunakan **Magisk**.

Ketiadaan kontrol ini membuka celah keamanan yang signifikan, khususnya terhadap potensi serangan pada saat aplikasi berjalan (*runtime attack*), seperti **code injection**, **method hooking**, maupun manipulasi perilaku aplikasi melalui framework pihak ketiga. Dalam kondisi tersebut, integritas aplikasi tidak dapat terjamin karena lingkungan eksekusinya telah berada di luar kontrol keamanan yang semestinya.

[Gambar: Device yang telah di-root menggunakan magisk]

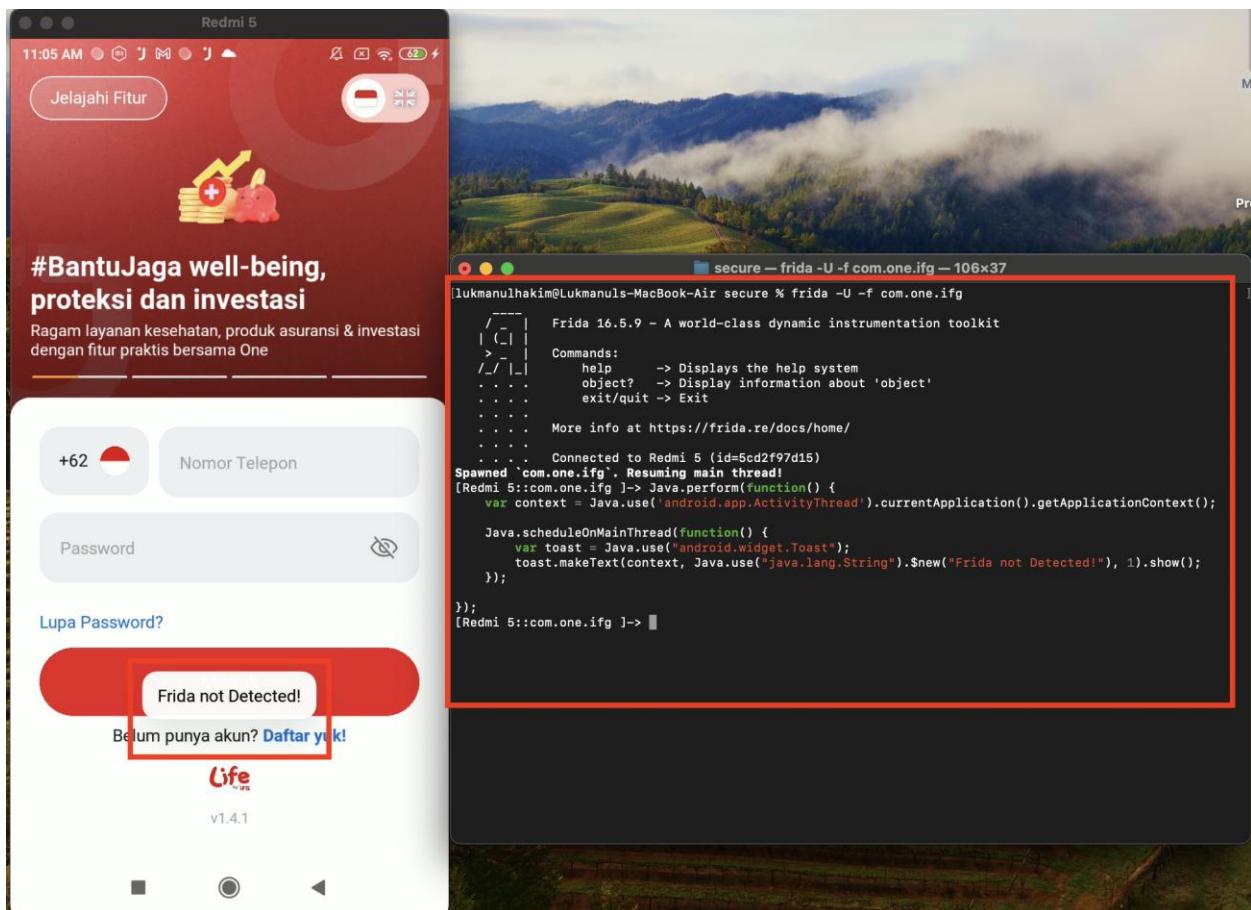


Sebagai tindak lanjut, dilakukan pengujian *runtime attack* menggunakan **Frida** untuk mensimulasikan skenario **code injection**. Pada pengujian ini, skrip Frida di-inject ke dalam aplikasi dengan tujuan menampilkan *toast message* bertuliskan “**Frida not detected**”.

Hasil pengujian menunjukkan bahwa proses injeksi **berhasil dilakukan** dan aplikasi **tetap berjalan secara normal** tanpa adanya indikasi pemblokiran, peringatan, maupun terminasi aplikasi. Kondisi ini mengindikasikan bahwa aktivitas injeksi melalui Frida **tidak terdeteksi oleh aplikasi**, serta tidak terdapat mekanisme proteksi yang mampu mendeteksi atau mencegah penggunaan *dynamic instrumentation framework* tersebut.

Temuan ini menegaskan bahwa pada kondisi *before hardened*, aplikasi **rentan terhadap serangan injeksi pada saat runtime**, termasuk manipulasi fungsi, pengambilan data sensitif, maupun perubahan alur logika aplikasi.

[Gambar: Hooking menggunakan Frida untuk menampilkan toast message]



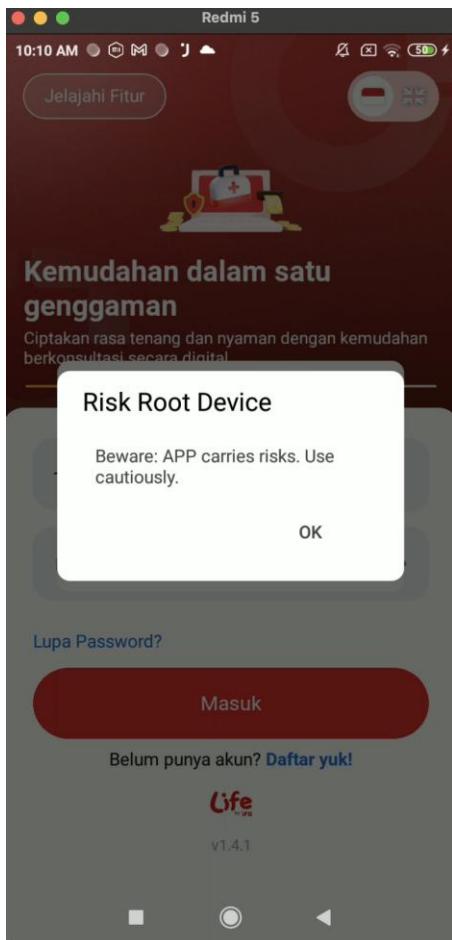
b. After Hardened

Setelah proses *hardening* diterapkan menggunakan **SeIron**, aplikasi kini telah dilengkapi dengan mekanisme **deteksi aktivitas hooking, manipulasi runtime, serta kondisi perangkat yang tidak aman (rooted device)**. Mekanisme ini memungkinkan aplikasi untuk mengidentifikasi upaya serangan yang dilakukan melalui *dynamic instrumentation*, teknik hooking, maupun eksekusi aplikasi pada lingkungan perangkat yang telah di-root.

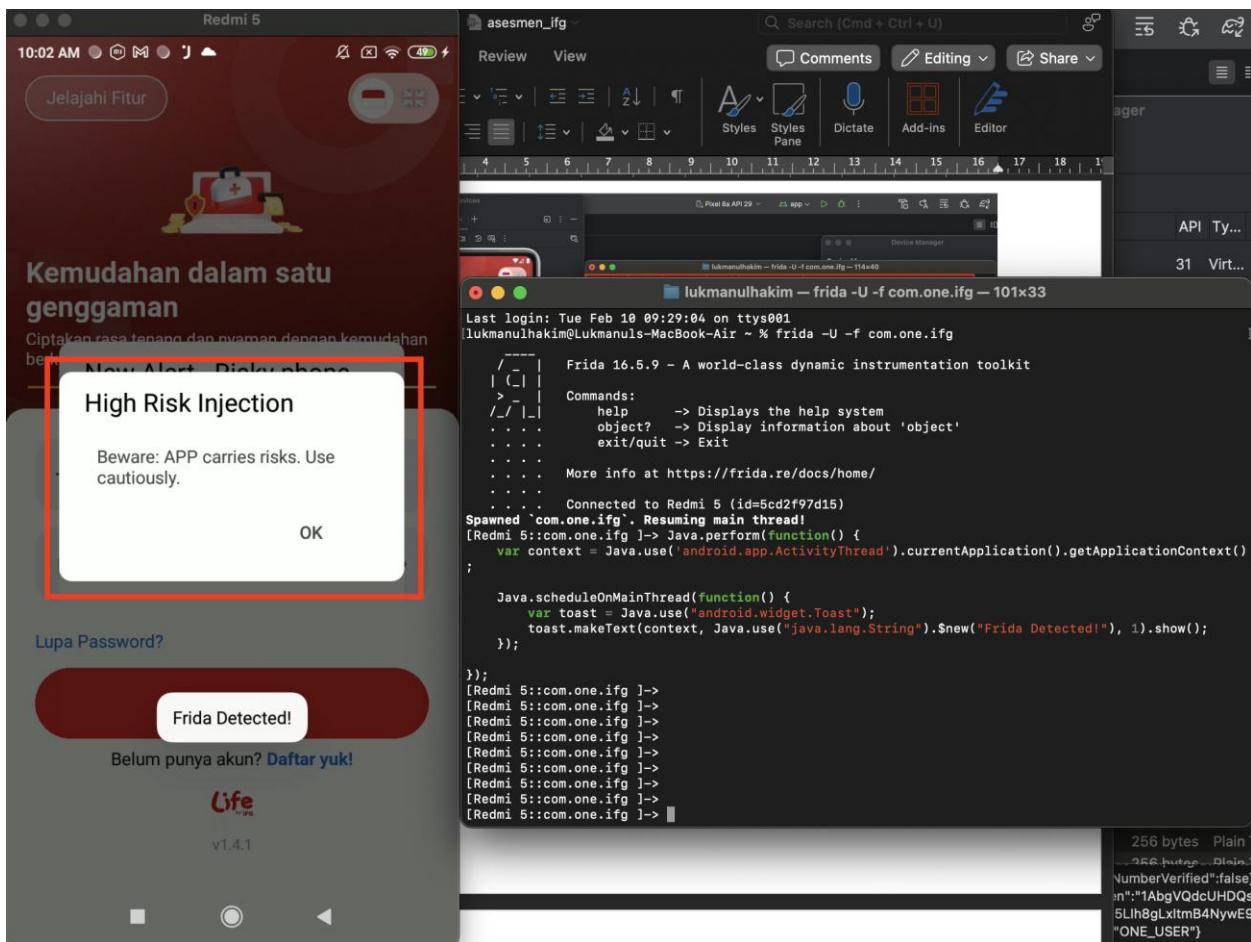
Pada saat aktivitas hooking atau indikasi perangkat *rooted* terdeteksi, aplikasi secara otomatis **mengambil tindakan mitigasi**, seperti **penghentian eksekusi aplikasi**, sehingga upaya serangan tidak dapat dilanjutkan. Selain itu, informasi terkait insiden keamanan tersebut **tercatat dan terkirim ke dashboard IronSKY**, memungkinkan tim keamanan untuk melakukan pemantauan, analisis, serta respons lanjutan secara terpusat dan real-time.

Penerapan proteksi ini secara signifikan meningkatkan **ketahanan aplikasi terhadap serangan pada saat runtime**, sekaligus memberikan **visibilitas yang lebih baik terhadap pola serangan, kondisi perangkat, dan sumber ancaman** yang terjadi di sisi klien.

[Gambar: Aplikasi mendeteksi device root setelah di-hardened]



[Gambar: Aplikasi mendeteksi aktivitas hooking yang terjadi setelah di-hardened]



[Gambar: Data monitoring yang tercatat pada dashboard IronSKY]

4. Integrity Protection

a. Before Hardened

Selain pengujian pada aspek *runtime protection*, dilakukan pula evaluasi terhadap **integrity protection** aplikasi. Hasil pengujian menunjukkan bahwa pada kondisi *before hardened*, aplikasi **belum dilengkapi dengan mekanisme pemeriksaan integritas** untuk mendeteksi perubahan pada kode atau struktur aplikasi.

Pada skenario pengujian ini, aplikasi didekompilasi dan dilakukan modifikasi dengan menambahkan *toast message* pada **Main Activity** berupa teks “**Modified Build**”, kemudian aplikasi dibangun ulang (*rebuild*) dan diinstall kembali ke perangkat uji.

[Gambar: Aplikasi setelah di dekompilasi dan ditambahkan Toast dengan teks Modified Build]

The screenshot shows the Android Studio interface with the decompiler tool open. The left sidebar displays the project structure under 'COM.ONE.IFG V1.4.1_ANTI...', with 'MainActivity.smali' selected. The main pane shows the SMALI code for the constructor of MainActivity. A red box highlights a section of code where a custom toast message has been inserted:

```
    .method public constructor <init>()V
        .end method

        # virtual methods
        .method protected onCreate(Landroid/os/Bundle;)V
            .locals 3

            .line 1
            const/4 p1, 0x0

            .line 2
            invoke-super {p0, p1}, Lcom/facebook/react/q;->onCreate(Landroid/os/Bundle;)V

            .line 3
            invoke-static {p0}, Lrq/d;->f(Landroid/app/Activity;)V

# ===== TAMBAHAN TOAST =====
const-string v0, "MODIFIED BUILD"
const/4 v1, 0x1

        invoke-static {p0, v0, v1}, Landroid/widget/Toast;->makeText(Landroid/content/Context;Lj;
        move-result-object v2
        invoke-virtual {v2}, Landroid/widget/Toast;->show()V
# =====

        .line 4
        return-void
    .end method

.method protected x0()Lcom/facebook/react/r;
    .locals 3

    .line 1
    new-instance v0, Lq9/a;

    .line 2
```

Setelah aplikasi terpasang, hasil pengujian menunjukkan bahwa aplikasi **tetap dapat dijalankan secara normal** tanpa adanya peringatan, pemblokiran, maupun terminasi aplikasi. *Toast message* yang ditambahkan muncul sebagaimana fungsi bawaan aplikasi, seolah-olah merupakan bagian dari aplikasi yang sah (*legitimate*).

[Gambar: Aplikasi tetap berjalan setelah dimodifikasi]

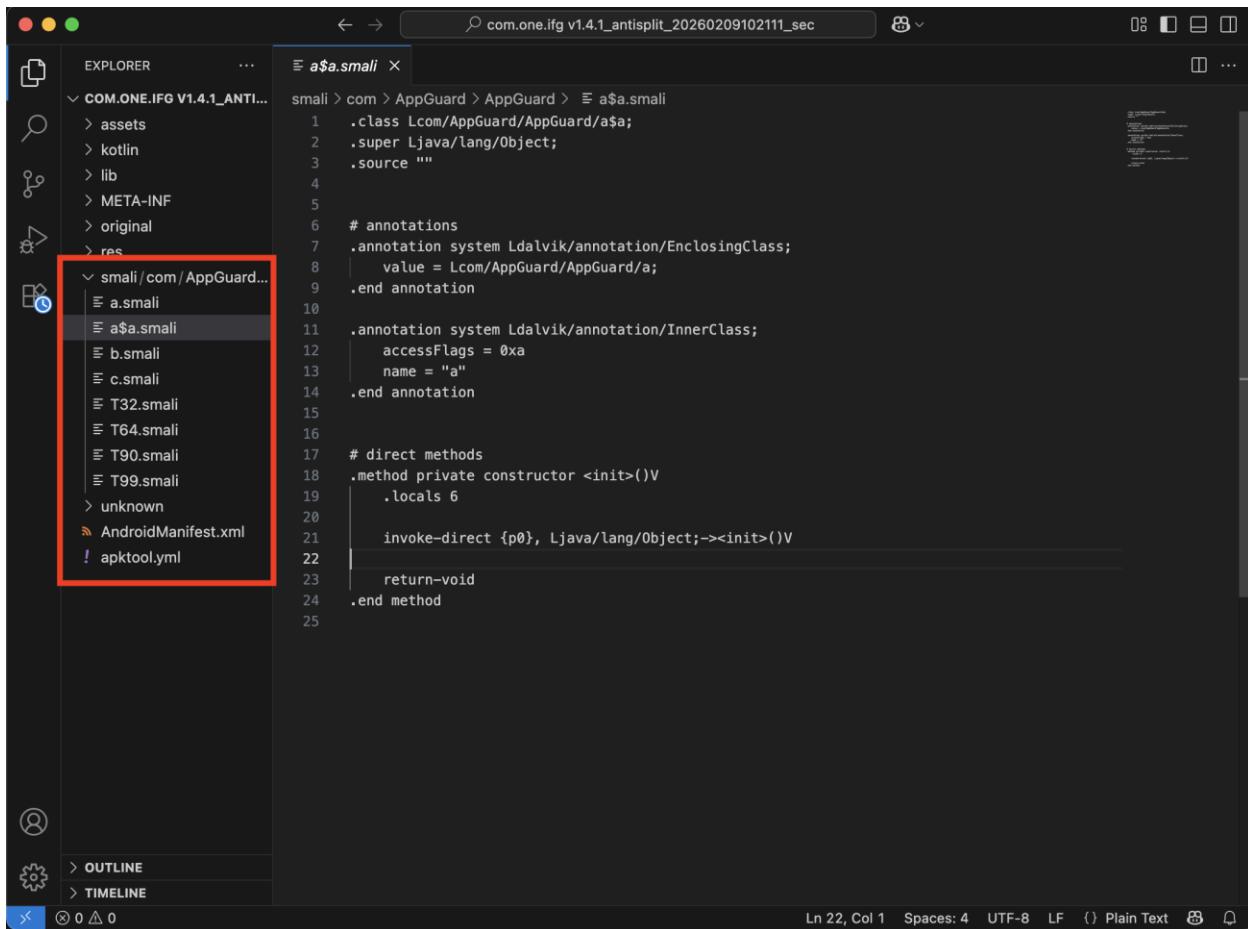


Kondisi ini menunjukkan bahwa aplikasi **tidak mendeteksi adanya modifikasi kode**, sehingga membuka risiko keamanan yang serius. Dalam skenario penyalahgunaan, penyerang dapat memodifikasi satu halaman penuh agar menyerupai tampilan asli aplikasi, namun pada kenyataannya mengarahkan pengguna ke **endpoint berbahaya milik pihak tidak berwenang**. Risiko ini berpotensi menyebabkan pencurian data, penyalahgunaan kredensial, hingga kompromi akun pengguna.

b. After Hardened

Pada saat dilakukan upaya modifikasi terhadap aplikasi yang telah di-*harden* menggunakan SecIron, seluruh kelas yang sebelumnya dapat diakses, termasuk **Main Activity**, **tidak lagi dapat dilihat maupun dianalisis melalui proses dekompilasi**. Hal ini menunjukkan bahwa mekanisme proteksi kode telah diterapkan secara efektif, sehingga menyulitkan pihak tidak berwenang untuk melakukan *reverse engineering* terhadap logika inti aplikasi.

[Gambar: Hasil dekompilasi aplikasi setelah hardening, di mana kelas-kelas internal tidak lagi dapat diidentifikasi]



```

smali > com > AppGuard > AppGuard > a$a.smali
1 .class Lcom/AppGuard/AppGuard/a$a;
2 .super Ljava/lang/Object;
3 .source ""
4
5
6 # annotations
7 .annotation system Ldalvik/annotation/EnclosingClass;
8 | value = Lcom/AppGuard/AppGuard/a;
9 .end annotation
10
11 .annotation system Ldalvik/annotation/InnerClass;
12 | accessFlags = 0xa
13 | name = "a"
14 .end annotation
15
16
17 # direct methods
18 .method private constructor <init>()V
19 | .locals 6
20 |
21 | invoke-direct {p0}, Ljava/lang/Object;--><init>()V
22 |
23 | return-void
24 .end method
25

```

Sebagai bagian dari pembuktian (*proving*) mekanisme **integrity protection** SecIron, dilakukan pengujian lanjutan dengan melakukan perubahan minor pada aplikasi, yaitu **mengubah label aplikasi menjadi “modded – IFG”**.

[Gambar: Merubah label aplikasi menjadi “modded -IFG”]

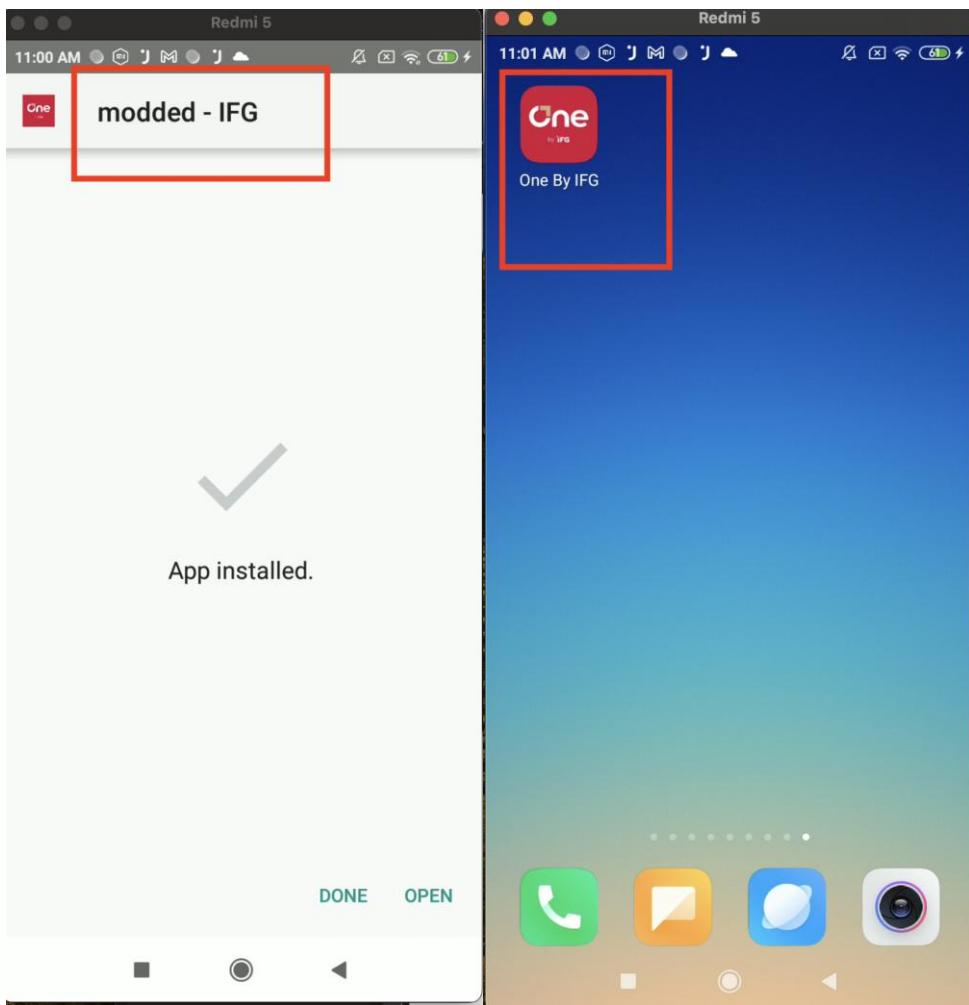
The screenshot shows the Android Studio interface with the following details:

- Top Bar:** Shows the project name "com.one.ifg v1.4.1_antisplit_20260209102111_sec".
- Left Sidebar (EXPLORER):** Lists the project structure:
 - assets
 - kotlin
 - lib
 - META-INF
 - original
 - res
 - smali/com/AppGuard... (with 5 files)
 - a.smali
 - a\$a.smali
 - b.smali
 - c.smali
 - T32.smali
 - T64.smali
 - T90.smali
 - T99.smali
 - unknown
- Central Area:** Displays the decompiled code of `AndroidManifest.xml`. A specific line of code is highlighted with a red box:

```
    <activity android:name="com.AppGuard.MainActivity" android:screenOrientation="Portrait" android:windowSoftInputMode="AdjustPan">  
        <intent-filter>  
            <action android:name="android.intent.action.MAIN" />  
            <category android:name="android.intent.category.LAUNCHER" />  
        </intent-filter>  
    </activity>
```
- Bottom Status Bar:** Shows the current line and column (Ln 75, Col 287), the number of spaces (Spaces: 4), the encoding (UTF-8), and the file format (LF).

Setelah dilakukan *rebuild* dan instalasi ulang ke perangkat uji, aplikasi langsung mengalami ***force close*** pada saat dijalankan.

[Gambar: Aplikasi hasil modifikasi yang diinstal pada perangkat dan langsung mengalami force close akibat mekanisme integrity protection]



Hasil pengujian ini membuktikan bahwa SecIron **mampu mendeteksi setiap perubahan pada paket aplikasi**, termasuk modifikasi yang bersifat non-fungsional seperti perubahan label. Dengan demikian, mekanisme *integrity protection* bekerja secara menyeluruh untuk mencegah eksekusi aplikasi yang telah dimodifikasi.

5. Man-in-the-Middle (MITM) Protection

a. Before Hardened

Berdasarkan hasil pengujian **Man-in-the-Middle (MITM)** menggunakan **Burp Suite**, diketahui bahwa aplikasi **belum menerapkan mekanisme SSL Pinning maupun deteksi terhadap aktivitas MITM**. Pada kondisi ini, komunikasi antara aplikasi dan server dapat diintersepsi dengan mudah, sehingga **endpoint API dan payload komunikasi dapat terlihat secara jelas**, sebagaimana ditunjukkan pada pengujian berikut.

[Gambar Percobaan Man-in-the-Middle menggunakan Burp Suite sebelum hardening]

#	Host	Method	URL	Params	Edited	Status code	Length	MIME type	Extension	Title
175	https://one.ig.id	POST	/api/v2/one/user/auth/login		✓	200	2120	JSON		
177	https://play.googleapis.com	GET	/pagead/dr/m?is_lgt=0		✓	200	931	HTML		
178	https://play.googleapis.com	POST	/log/batch		✓	200	324	text		
179	https://play.googleapis.com	POST	/log/batch		✓	200	324	text		
180	https://play.googleapis.com	POST	/log/batch		✓	200	324	text		
181	https://play.googleapis.com	POST	/log/batch		✓	200	324	text		
182	https://play.googleapis.com	POST	/log/batch		✓	200	324	text		
183	https://play.googleapis.com	POST	/log/batch		✓	200	324	text		
184	https://one.ig.id	POST	/api/v2/one/user/auth/login		✓	401	699	JSON		
185	https://one.ig.id	POST	/api/v2/one/user/auth/login		✓	200	2126	JSON		
186	https://play.googleapis.com	GET	/play/log/messages		✓	200	891	text		
187	https://play.googleapis.com	POST	/play/log?format=raw&proto_v2=true		✓	200	656	text		

Ketidaaan proteksi terhadap serangan MITM membuka risiko keamanan yang signifikan. Dalam skenario ini, penyerang berpotensi untuk:

- Mencegat dan memantau *request* serta *response* antara aplikasi dan server
- Memodifikasi data pada saat runtime
- Melakukan serangan *replay* maupun *spoofing* data

Kondisi tersebut dapat berdampak langsung pada kebocoran data sensitif, manipulasi transaksi, hingga penyalahgunaan sesi komunikasi aplikasi.

a. After Hardened

Setelah proses *hardening* diterapkan menggunakan **SecIron**, aplikasi kini mampu **mendeteksi aktivitas Man-in-the-Middle secara efektif**. Upaya intersepsi komunikasi yang dilakukan melalui Burp Suite berhasil teridentifikasi oleh mekanisme proteksi yang diterapkan.

Pada saat aktivitas MITM terdeteksi, aplikasi dapat **menjalankan respons keamanan yang bersifat konfigurable**, seperti melakukan *force exit* secara langsung atau menampilkan notifikasi peringatan sebelum aplikasi dihentikan. Fleksibilitas ini memungkinkan penyesuaian kebijakan keamanan sesuai dengan kebutuhan operasional dan tingkat risiko yang ditetapkan.

Selain itu, seluruh aktivitas serangan yang terdeteksi **tercatat dan dimonitor melalui dashboard IronSKY**, sehingga tim keamanan memperoleh visibilitas penuh terhadap upaya MITM yang terjadi di sisi klien.

[Gambar: Percobaan MITM menggunakan Burp Suite yang terdeteksi setelah hardening]

The screenshot illustrates a mobile application running on a Redmi 5 device. A prominent red box highlights a 'High Risk MITM' warning dialog box. The dialog contains the text: 'Beware: APP carries risks. Use cautiously. High Risk MITM' and an 'OK' button. Below the dialog, there's a 'Lupa Password?' link and a 'Masuk' button. At the bottom, it says 'Belum punya akun? Daftar yuk!' and shows the 'Life' logo with 'v1.4.1'. To the right of the phone screen is the Burp Suite Community Edition interface. The 'Proxy' tab is selected, showing a list of captured requests. One specific request is highlighted, showing a GET request to '/s/vdpbs5' with a status code of 307. The 'Request' and 'Response' panes below show the detailed headers and body of this request. The 'Inspector' pane on the right displays request and response headers. The overall environment demonstrates how the application handles a detected MITM attack by displaying a warning and logging the interaction in the proxy tool.

[Gambar: Data monitoring aktivitas MITM pada dashboard IronSKY]

e23441a5-4f4a-38ee-9f5d-7e2c412c5052	unknow	1141	One By IFG	1.4.1	High Risk MITM	1770692987355	Android 8.1.0	5.5.1	Pop-up message with close button	True
<hr/>										
THREAT ID	THREAT NAME									
MH100002	HTTP Proxy	Reason: ["agency"]; Proxy address: http://192.168.100.16:8080;								
MRI10002	Root/Jailbreak	root_manage_apps:#com.tsng.hidemyapplist; su_paths_exist/sbin/su; which_su/sbin/su; su_files_exist/sbin/su; magisk_root/fileStat[/sbin/								
<hr/>										
MC10001	Root certificate error	Reason: null; Issuer:CN=PortSwigger CA OU=PortSwigger CA O=PortSwigger L=PortSwigger ST=PortSwigger C=PortSwigger; CN=PortSwigger CA OU=PortSwigger CA O=PortSwigger L=PortSwigger ST=PortSwigger C=PortSwigger; Signature:795AB0F15CE2CD076218227F242848661C408D75FC0FE04F04330F0D245F7604F74610CD6408F36B92A09D154BF58CDF67E16DDC719D74B06: 795AB0F15CE2CD076218227F242848661C408D75FC0FE04F04330F0D245F7604F74610CD6408F36B92A09D154BF58CDF67E16DDC719D74B0673A33D7E9								

6. Penyimpanan Data di Shared Preferences

a. Before Hardened

Berdasarkan hasil analisis, ditemukan bahwa aplikasi menyimpan sejumlah konfigurasi di dalam **SharedPreferences**. Pada kondisi *before hardened*, mekanisme penyimpanan tersebut **belum dilengkapi dengan enkripsi maupun proteksi tambahan**.

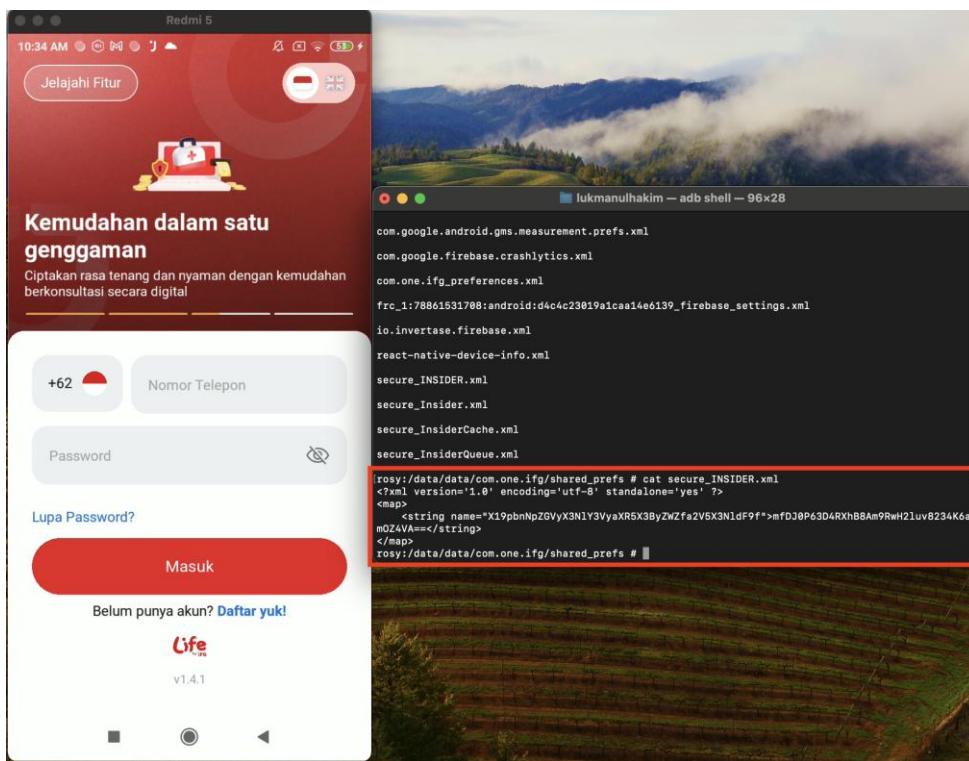
Apabila data sensitive, seperti **token autentikasi, ID pengguna, atau status login**, disimpan dalam SharedPreferences tanpa perlindungan yang memadai, maka informasi tersebut berpotensi untuk diakses maupun dimodifikasi, terutama pada perangkat yang berada dalam kondisi **rooted** atau **debuggable**.

Kondisi ini meningkatkan risiko keamanan, antara lain:

- Data dapat dibaca atau diubah secara langsung melalui akses ke penyimpanan aplikasi
- Terbukanya peluang terjadinya *spoofing* identitas pengguna
- Peningkatan risiko *session hijacking* akibat manipulasi token atau status autentikasi

Tanpa mekanisme perlindungan yang memadai, penyimpanan data pada sisi klien menjadi salah satu titik lemah yang dapat dimanfaatkan oleh pihak tidak berwenang.

[Gambar: Isi SharedPreferences hasil eksplorasi sebelum hardening]



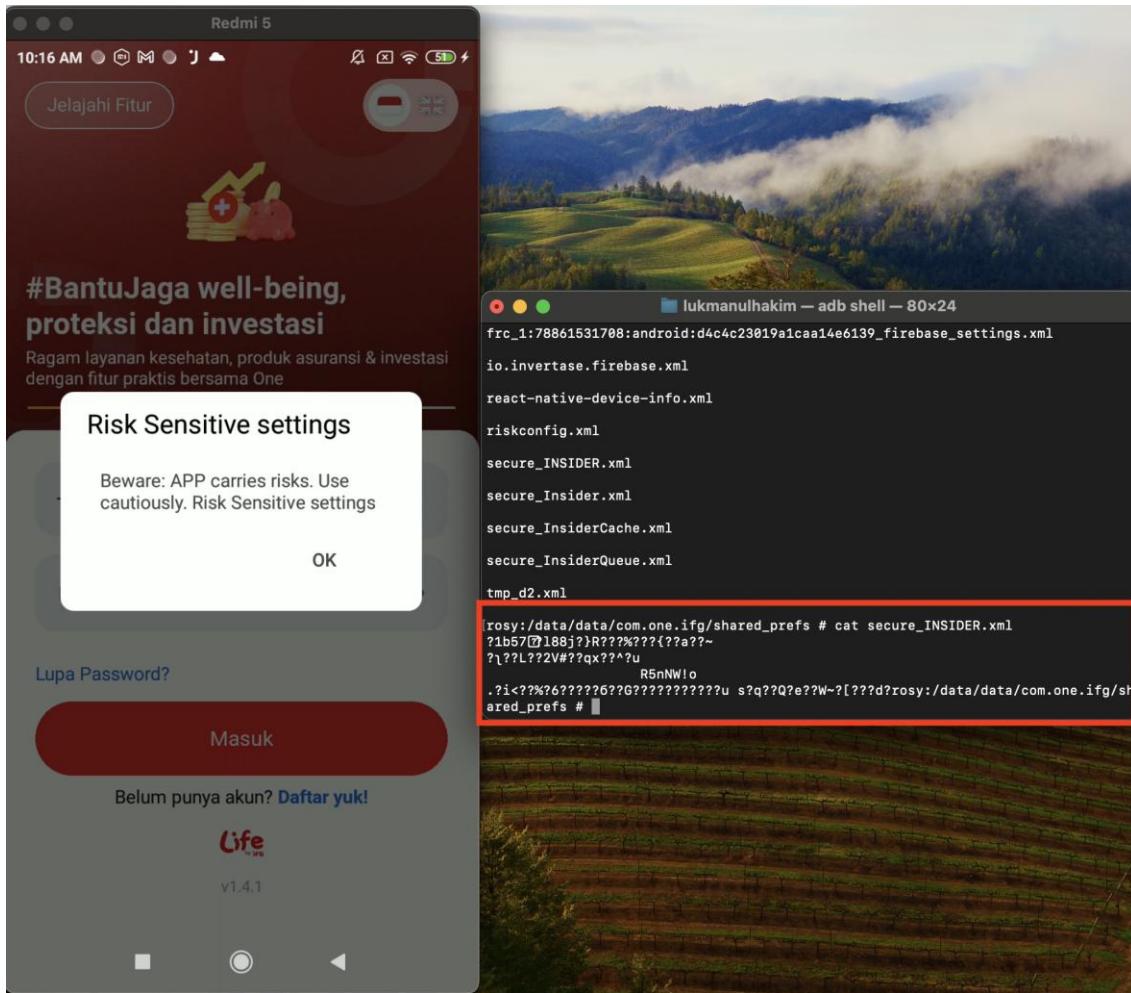
b. After Hardened

Setelah proses *hardening* diterapkan menggunakan **SecIron**, mekanisme **enkripsi pada SharedPreferences** telah diaktifkan. Dengan penerapan enkripsi ini, data yang disimpan di sisi klien **tidak lagi dapat dibaca dalam bentuk teks yang dapat dimengerti**, karena seluruh nilai tersimpan dalam format terenkripsi.

Hasil pengujian menunjukkan bahwa isi *SharedPreferences* yang sebelumnya dapat diakses dan dibaca secara langsung kini **berubah menjadi karakter acak yang tidak bermakna**, sehingga tidak dapat diinterpretasikan meskipun berhasil diakses secara tidak sah. Kondisi ini secara efektif melindungi data sensitif dari upaya pencurian informasi melalui akses langsung ke penyimpanan aplikasi.

Penerapan enkripsi *SharedPreferences* ini secara signifikan meningkatkan **keamanan penyimpanan data lokal**, serta meminimalkan risiko kebocoran informasi pada sisi perangkat pengguna.

[Gambar: Isi SharedPreferences pada aplikasi yang telah di-harden]



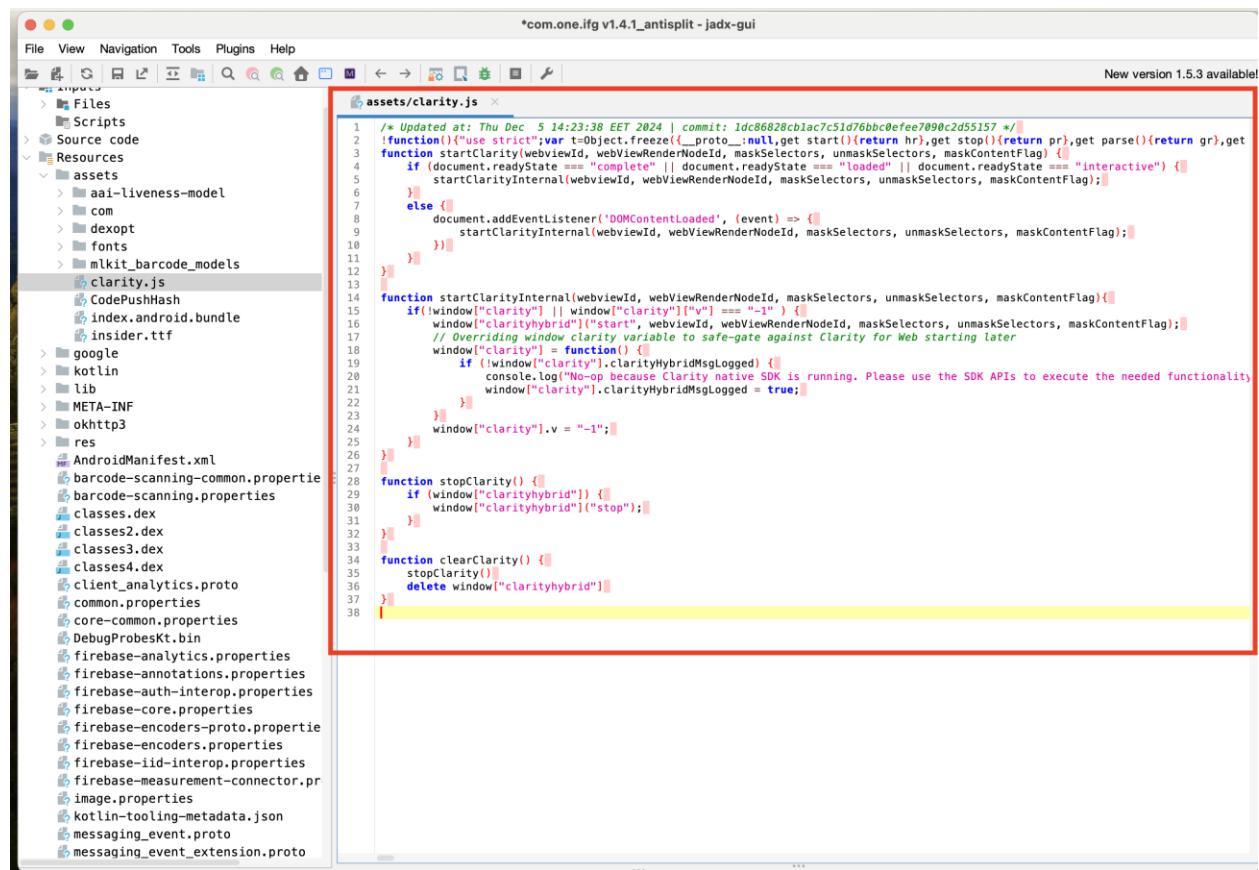
7. Resource Protection

a. Before Hardened

Berdasarkan hasil asesmen keamanan yang dilakukan, diketahui bahwa pada kondisi **sebelum penerapan mekanisme hardening**, aplikasi masih menyimpan sejumlah *resource* dalam bentuk tidak terenkripsi. *Resource* tersebut dapat diakses, ditampilkan, dan dianalisis secara langsung, sehingga berpotensi membuka informasi internal aplikasi.

Kondisi ini menunjukkan bahwa struktur aplikasi masih relatif mudah untuk dianalisis, yang dapat meningkatkan risiko terhadap upaya *reverse engineering* maupun penyalahgunaan *resource* oleh pihak yang tidak berwenang.

[Gambar: Contoh file resource aplikasi sebelum dilakukan hardening]



```
*com.one.ifg v1.4.1_antisplit - jadx-gui
File View Navigation Tools Plugins Help
Imports > Files > Scripts > Source code > Resources > assets > clarity.js
New version 1.5.3 available

assets/clarity.js
1 /* Updated at: Thu Dec 5 14:23:38 EET 2024 | commit: 1dc86828cb1ac7c51d76bbc0effe7890c2d55157 */
2 "use strict";var t=Object.freeze({__proto__:null,get start(){return hr},get stop(){return pr},get parse(){return gr},get
3 function startClarity(webviewId, webViewRenderNodeId, maskSelectors, unmaskSelectors, maskContentFlag) {
4   if (document.readyState === "complete" || document.readyState === "loaded" || document.readyState === "interactive") {
5     startClarityInternal(webviewId, webViewRenderNodeId, maskSelectors, unmaskSelectors, maskContentFlag);
6   }
7   else {
8     document.addEventListener('DOMContentLoaded', (event) => {
9       startClarityInternal(webviewId, webViewRenderNodeId, maskSelectors, unmaskSelectors, maskContentFlag);
10    })
11  }
12 }
13
14 function startClarityInternal(webviewId, webViewRenderNodeId, maskSelectors, unmaskSelectors, maskContentFlag) {
15   if(!window["clarity"] || window["clarity"]["v"] === "-1") {
16     window["clarityhybrid"] = startClarityInternal;
17     // Overriding window.clarity variable to safe-gate against Clarity for Web starting later
18     window["clarity"] = function() {
19       if (!window["clarity"]) {
20         console.log("No-op because Clarity native SDK is running. Please use the SDK APIs to execute the needed functionality");
21         window["clarityhybridMsgLogged"] = true;
22       }
23     }
24   }
25   window["clarity"].v = "-1";
26 }
27
28 function stopClarity() {
29   if (window["clarityhybrid"]) {
30     window["clarityhybrid"]("stop");
31   }
32 }
33
34 function clearClarity() {
35   stopClarity();
36   delete window["clarityhybrid"];
37 }
38 */


```

b. After Hardened

Setelah penerapan mekanisme hardening, *resource* pada aplikasi telah terenkripsi, sehingga tidak lagi dapat dianalisis secara langsung. Kondisi ini secara signifikan meningkatkan tingkat kesulitan terhadap upaya *reverse engineering* serta mengurangi risiko pemanfaatan *resource* aplikasi oleh pihak yang tidak berwenang.

Penerapan enkripsi pada *resource* aplikasi menunjukkan bahwa mekanisme hardening yang diterapkan telah berjalan dengan baik dalam melindungi komponen internal aplikasi dari proses analisis statis.

[Gambar: Contoh file resource aplikasi setelah dilakukan hardening]

8. Kesimpulan

Berdasarkan hasil asesmen keamanan yang telah dilakukan terhadap aplikasi **One by IFG**, dapat disimpulkan bahwa sebelum penerapan *hardening*, aplikasi masih memiliki sejumlah celah keamanan yang signifikan. Celah tersebut mencakup aspek *static protection*, *runtime protection*, *integrity protection*, *network security (MITM)*, serta *local data storage*, yang berpotensi dimanfaatkan oleh pihak tidak berwenang untuk melakukan *reverse engineering*, manipulasi aplikasi, pencurian data, maupun penyalahgunaan kredensial pengguna.

Setelah penerapan mekanisme *hardening* menggunakan **SecIron**, terjadi peningkatan postur keamanan aplikasi secara menyeluruh dan terukur, dengan hasil sebagai berikut:

- **Code Protection**
Kode aplikasi berhasil diamankan melalui obfuscation, enkripsi, dan code hiding, sehingga struktur internal serta informasi sensitif tidak lagi dapat dianalisis melalui proses dekompilasi.
- **Runtime Protection**
Aplikasi mampu mendeteksi dan memitigasi serangan pada saat runtime, termasuk aktivitas hooking, dynamic instrumentation (Frida), serta eksekusi aplikasi pada perangkat yang telah di-root. Upaya serangan dihentikan secara otomatis dan tercatat pada dashboard IronSKY.
- **Integrity Protection**
Setiap perubahan pada paket aplikasi, baik perubahan logika maupun modifikasi minor seperti perubahan label aplikasi, berhasil terdeteksi dan menyebabkan aplikasi tidak dapat dijalankan (*force close*). Hal ini membuktikan bahwa integritas aplikasi terlindungi secara menyeluruh.
- **Man-in-the-Middle (MITM) Protection**
Upaya intersepsi komunikasi jaringan berhasil dideteksi, dengan respons keamanan yang dapat dikonfigurasi sesuai kebijakan. Aktivitas MITM juga tercatat dan dapat dimonitor secara terpusat melalui IronSKY.
- **Local Data Protection (SharedPreferences)**
Data yang disimpan secara lokal telah terenkripsi, sehingga tidak dapat dibaca atau dimodifikasi meskipun diakses secara tidak sah, secara signifikan menurunkan risiko kebocoran data dan *session hijacking*.
- **Resources Encryption**
Resource internal aplikasi telah dienkripsi sebagai bagian dari mekanisme hardening, sehingga tidak lagi dapat dianalisis secara langsung. Penerapan enkripsi ini meningkatkan ketahanan aplikasi terhadap analisis statis dan upaya reverse engineering, serta melindungi komponen internal aplikasi dari penyalahgunaan oleh pihak yang tidak berwenang.

Secara keseluruhan, penerapan SecIron telah meningkatkan **ketahanan aplikasi terhadap berbagai vektor serangan**, memperkuat perlindungan data pengguna, serta memberikan **visibilitas dan kontrol keamanan yang lebih baik** bagi tim melalui sistem monitoring.