HackCtrl

https://hackcontrol.org/

ANDROID APPLICATION
PENETRATION TESTING

Report for:	
Date:	

This document contains confidential information about IT systems and network infrastructure of the customer, as well as information about potential vulnerabilities and methods of their exploitation. This confidential information is for internal use by the customer only and shall not be disclosed to third parties.

Table of Content

Introduction

We thank _____________ for giving us the opportunity to conduct Security Assessment of their mobile application and its backend API. This document outlines our methodology, limitations and results of the security assessment.

Executive Summary

Hackcontrol (Consultant) was contracted by _________ (Customer) to conduct the penetration testing of their mobile application.

This report presents the findings of the penetration testing of CLIENT`s mobile application conducted between February 04th, 2018 – February 22nd, 2018.

The main subject of the penetration testing is ___________`s mobile systems & API.

Application Security Assessment has the following objectives:

- identify technical and functional vulnerabilities;
- estimate their severity level (ease of use, impact on information systems, etc);
- modelling the "most likely" attack vector against the Customer's Information System;
- proof of concept and exploitation of vulnerabilities;
- draw up a prioritized list of recommendations to address identified weaknesses.

According to our research, the mobile application is of **high security rating** for Customer and Backend systems; Several high-level vulnerabilities have been detected, however it requires a considerable amount of time and efforts to exploit them.

Three (3) High vulnerabilities of sensitive info logging and bypass root and developer mode checks were diagnosed during the security assessment. Also, three (3) Medium and a number of low and Informative vulnerabilities and errors were identified.

Team

Role	Name	EMAIL
Project Manager	John Doe (CEH, ISO27001 LA)	info@hackcontrol.org
Penetration Testing Engineer	John Doe (OSCP, eWPT, eCPPT)	engineer@hackcontrol.org

Scope of the Security Assessment

The following list of systems was in the scope of the Security Assessment.

#	Name	Description
1	___________v_0.9.2.apk	

Security Assessment start and end dates were coordinated by email according to the following table.

Methodology

The testing methodology is based on generally accepted industry-wide approaches to perform penetration testing for mobile applications – Mobile Security Testing Guide (MSTG);

Application-level penetration tests include, at a minimum, checking for the following types of vulnerabilities:

- lack of binary protections;
- insecure data storage;
- unintended data leakage;
- client-side injection;
- weak encryption;
- implicit trust of all certificates;
- execution of activities using root;
- private key exposure;
- exposure of database parameters and SQL queries;
- insecure random number generator;

Severity Definition

The level of criticality of each risk is determined based on the potential impact of loss from successful exploitation as well as ease of exploitation, existence of exploits in public access and other factors.

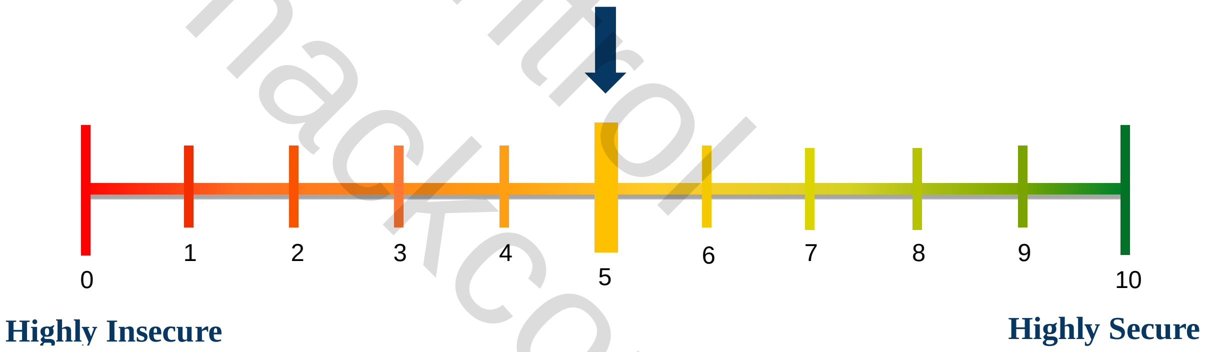
Severity	Description
High 🟥	High-level vulnerabilities are easy in exploitation and may provide an attacker with full control of the affected systems, also may lead to significant data loss or downtime. There are exploits or PoC available in public access.
Medium 🟨	Medium-level vulnerabilities are much harder to exploit and may not provide the same access to affected systems. No exploits or PoCs available in public access. Exploitation provides only very limited access.
Low 🟩	Low-level vulnerabilities provide an attacker with information that may assist them in conducting subsequent attacks against target information systems or against other information systems, which belong to an organization. Exploitation is extremely difficult, or impact is minimal.
Info 🟦	These vulnerabilities are informational and can be ignored.

Summary of Findings

According to the following in-depth testing of the mobile application & API, those require improvements.

Value	Numbers of risks
High	3
Medium	3
Low	2
Info	5

Based on our understanding of the mobile application and backend API, as well as the nature of the vulnerabilities discovered, their exploitability, and their potential impact, we have assessed the level of risk for your organization to be Medium.



Three (3) high, three (3) medium, two (2) low and five (5) informational level vulnerabilities have been found.

Despite the number of vulnerabilities and errors, there was no way to gain an unauthorized access or steal and modify the sensitive information like database data at backend system. However, the number of potential issues may increase during implementation of new functionality and its modification.

There were no prevention and blocking actions during the testing from the ________________’s security team and systems. Also, no account blocking was provided during a malicious activity and scanning process.

We have not performed test money transfer between devices. Application is crashing when one of the devices initiate a transfer and money is not sent. We have tested on 5 different devices and Android versions.

Key Findings

Root and developer mode bypass

#1	Description	Type: Real

In Android devices, rooting is the process of allowing smartphones, tablets and other devices to attain privileged control (known as "root access") within Android's sub-system.

Rooting is often performed with the goal of overcoming limitations that carriers and hardware manufacturers put on some devices. Thus, rooting gives the ability (or permission) to alter or replace system applications and settings, run specialized apps that require administrator-level permissions, or perform other operations that are otherwise inaccessible to a normal Android user. On Android, rooting can also facilitate the complete removal and replacement of the device's operating system, usually with a more recent release of its current operating system.

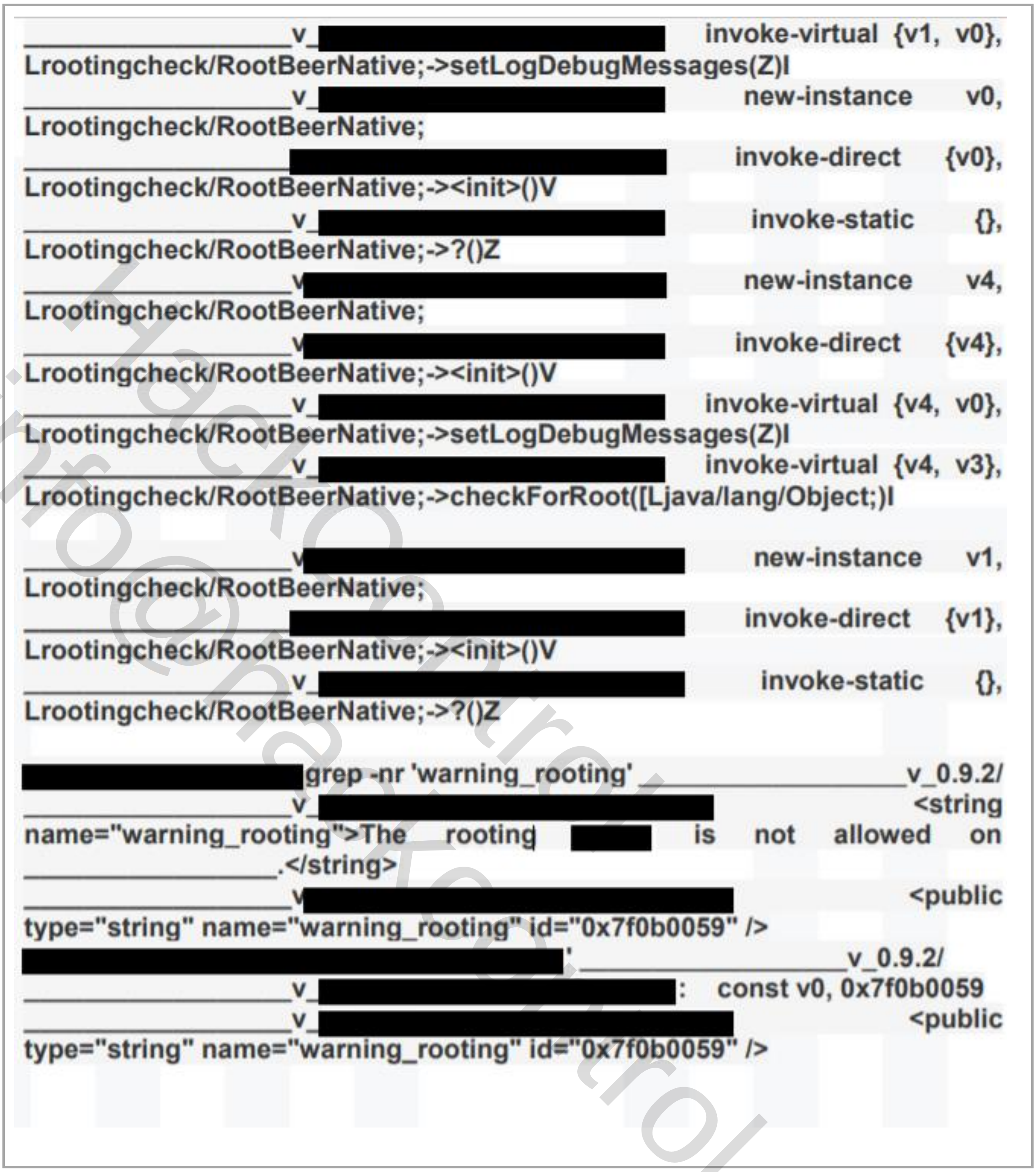
Rooted devices can be used to gain information about an application. The Settings app on Android includes a screen called Developer options that allows configuring system behavior and debugging application. For example, it can be used for enabling debugging over USB, capture a bug report, enable visual feedback for taps and more.

Evidences

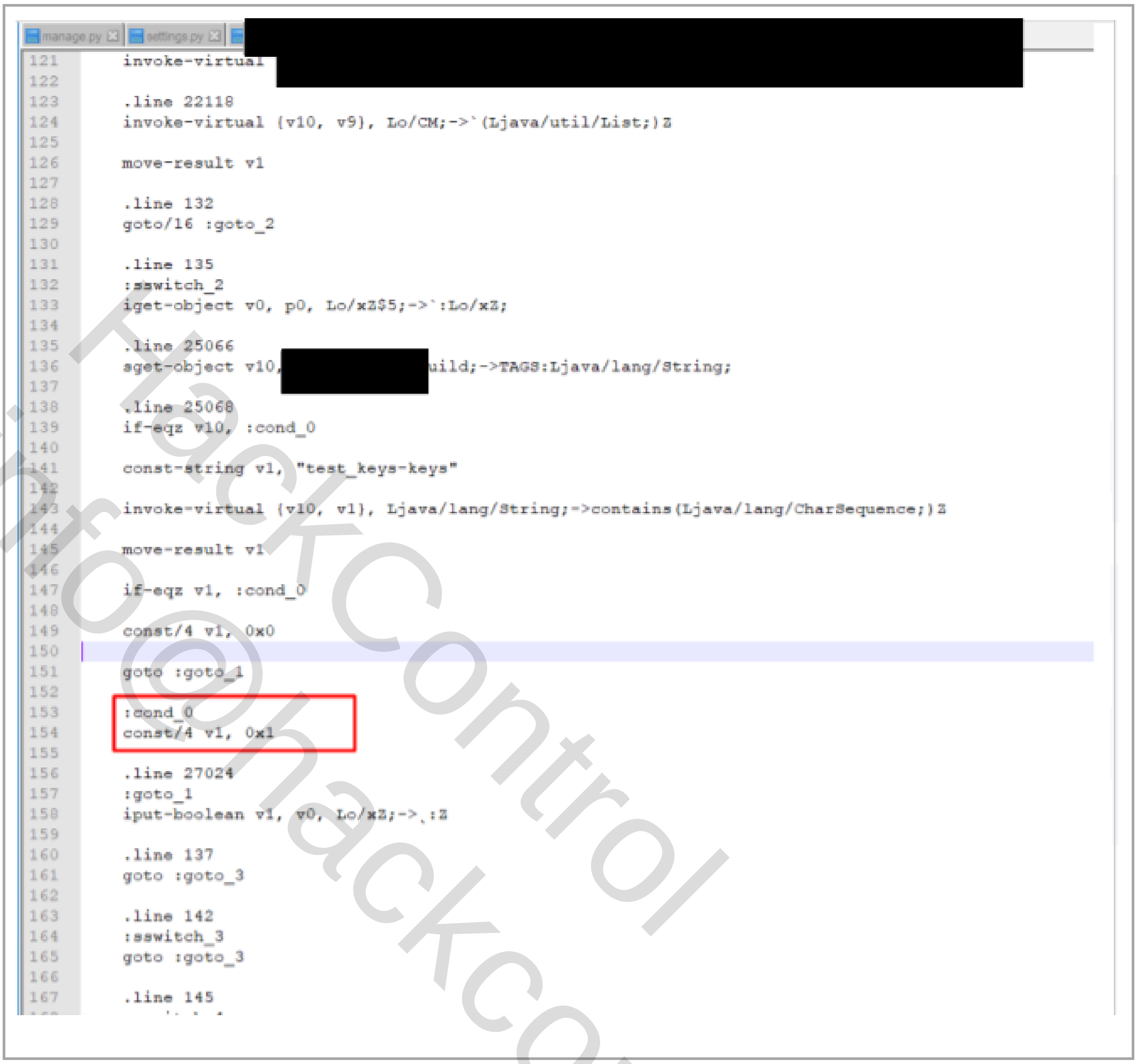
Steps to reproduce:

- Decompile application
- Search keywords for developer-mode and rootchecks class in source code
- Change associated strings and results of checks
- ReCompile application
- Sign application

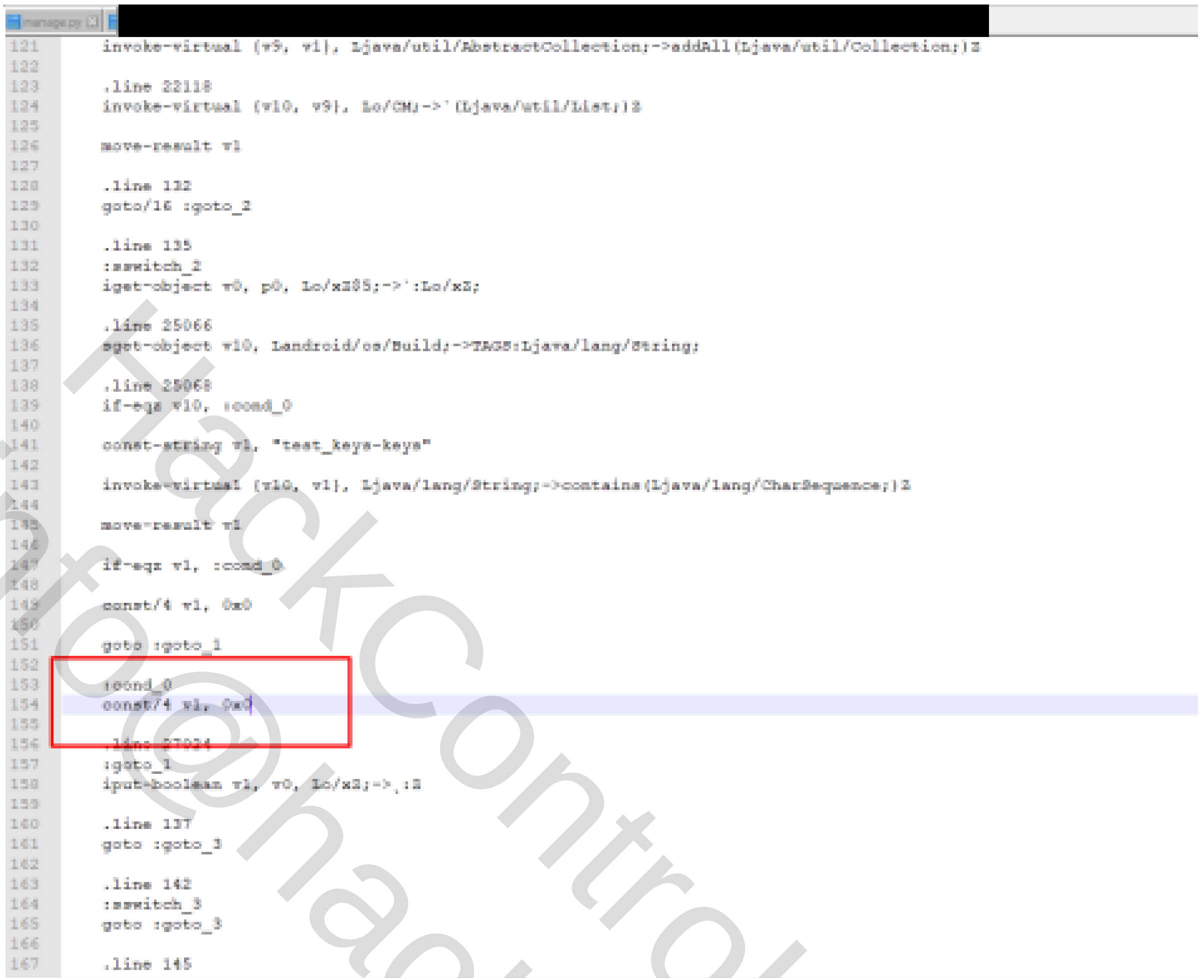
[redacted]________________.apk



```
____________________v_[redacted]      invoke-virtual {v1, v0},
Lrootingcheck/RootBeerNative;->setLogDebugMessages(Z)I
____________________v_[redacted]         new-instance     v0,
Lrootingcheck/RootBeerNative;
____________________[redacted]         invoke-direct     {v0},
Lrootingcheck/RootBeerNative;-><init>()V
____________________v_[redacted]          invoke-static      {},
Lrootingcheck/RootBeerNative;->?()Z
____________________v[redacted]         new-instance     v4,
Lrootingcheck/RootBeerNative;
____________________v[redacted]         invoke-direct     {v4},
Lrootingcheck/RootBeerNative;-><init>()V
____________________v_[redacted]      invoke-virtual {v4, v0},
Lrootingcheck/RootBeerNative;->setLogDebugMessages(Z)I
____________________v_[redacted]      invoke-virtual {v4, v3},
Lrootingcheck/RootBeerNative;->checkForRoot([Ljava/lang/Object;)I

____________________v[redacted]         new-instance     v1,
Lrootingcheck/RootBeerNative;
____________________[redacted]         invoke-direct     {v1},
Lrootingcheck/RootBeerNative;-><init>()V
____________________v_[redacted]          invoke-static      {},
Lrootingcheck/RootBeerNative;->?()Z

[redacted]grep -nr 'warning_rooting' ____________________v_0.9.2/
____________________v_[redacted]                        <string
name="warning_rooting">The   rooting  [redacted]  is   not   allowed   on
____________________.</string>
____________________v[redacted]                         <public
type="string" name="warning_rooting" id="0x7f0b0059" />
[redacted]'____________________v_0.9.2/
____________________v_[redacted]:  const v0, 0x7f0b0059
____________________v_[redacted]                         <public
type="string" name="warning_rooting" id="0x7f0b0059" />
```



```
    invoke-virtual [redacted]

    .line 22118
    invoke-virtual {v10, v9}, Lo/CM;->`(Ljava/util/List;)Z

    move-result v1

    .line 132
    goto/16 :goto_2

    .line 135
    :sswitch_2
    iget-object v0, p0, Lo/xZ$5;->`:Lo/xZ;

    .line 25066
    sget-object v10,[redacted]uild;->TAGS:Ljava/lang/String;

    .line 25068
    if-eqz v10, :cond_0

    const-string v1, "test_keys-keys"

    invoke-virtual {v10, v1}, Ljava/lang/String;->contains(Ljava/lang/CharSequence;)Z

    move-result v1

    if-eqz v1, :cond_0

    const/4 v1, 0x0

    goto :goto_1

    :cond_0
    const/4 v1, 0x1

    .line 27024
    :goto_1
    iput-boolean v1, v0, Lo/xZ;->,:Z

    .line 137
    goto :goto_3

    .line 142
    :sswitch_3
    goto :goto_3

    .line 145
```



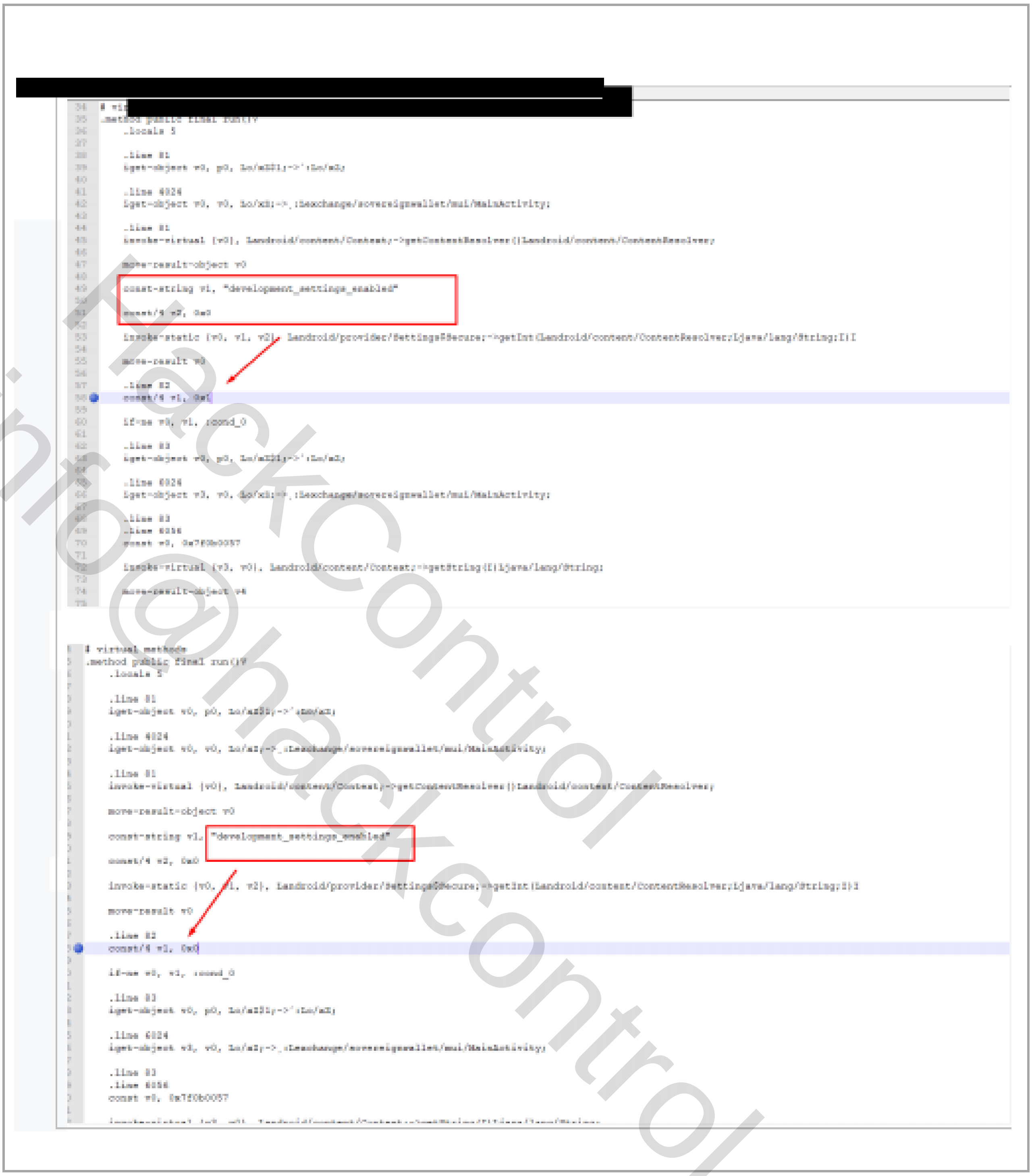
[redacted]

```
    invoke-virtual {v9, v1}, Ljava/util/AbstractCollection;->addAll(Ljava/util/Collection;)Z

    .line 22118
    invoke-virtual {v10, v9}, Lo/CM;->`(Ljava/util/List;)Z

    move-result v1

    .line 132
    goto/16 :goto_2

    .line 135
    :sswitch_2
    iget-object v0, p0, Lo/x2S5;->`:Lo/xZ;

    .line 25066
    sget-object v10, Landroid/os/Build;->TAGS:Ljava/lang/String;

    .line 25068
    if-eqz v10, :cond_0

    const-string v1, "test_keys-keys"

    invoke-virtual {v10, v1}, Ljava/lang/String;->contains(Ljava/lang/CharSequence;)Z

    move-result v1

    if-eqz v1, :cond_0

    const/4 v1, 0x0

    goto :goto_1

    :cond_0
    const/4 v1, 0x0

    .line 27024
    :goto_1
    iput-boolean v1, v0, Lo/x2;->,:Z

    .line 137
    goto :goto_3

    .line 142
    :sswitch_3
    goto :goto_3

    .line 145
```

[redacted]

```
     * Enabled aggressive block sorting
     * Enabled unnecessary exception pruning
     * Enabled aggressive exception aggregation
     */
    @Override
    protected void onCreate(Bundle object) {
        block10 : {
            Object object2;
            int n2;
            boolean bl2;
            block17 : {
                super.onCreate((Bundle)object);
[illegible]
```

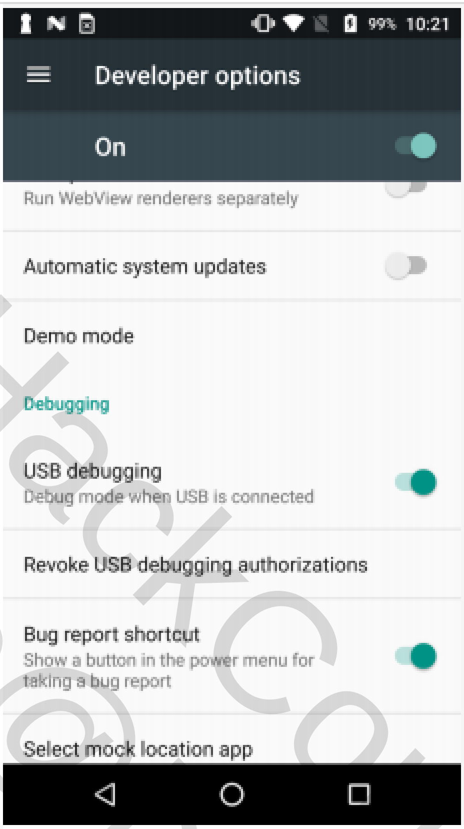
$ grep -nr "development_settings_enabled" ______________v_0.9.2\ Binary file ______________v_[redacted]s.dex matches ______________[redacted] const-string v1, "development_settings_enabled"



[redacted]

```
# vi[redacted]
.method public final run()V
    .locals 5

    .line 81
    iget-object v0, p0, Lo/a221;->'`:Lo/a2;

    .line 4024
    iget-object v0, v0, Lo/a2;->_:Lexchange/sovereignwallet/mui/MainActivity;

    .line 81
    invoke-virtual {v0}, Landroid/content/Context;->getContentResolver()Landroid/content/ContentResolver;

    move-result-object v0

    const-string v1, "development_settings_enabled"

    const/4 v2, 0x0

    invoke-static {v0, v1, v2}, Landroid/provider/Settings$Secure;->getInt(Landroid/content/ContentResolver;Ljava/lang/String;I)I

    move-result v0

    .line 82
    const/4 v1, 0x1

    if-ne v0, v1, :cond_0

    .line 83
    iget-object v0, p0, Lo/a221;->'`:Lo/a2;

    .line 6024
    iget-object v3, v0, Lo/a2;->_:Lexchange/sovereignwallet/mui/MainActivity;

    .line 83
    .line 6056
    const v0, 0x7f0b0057

    invoke-virtual {v3, v0}, Landroid/content/Context;->getString(I)Ljava/lang/String;

    move-result-object v4
```

```
# virtual methods
.method public final run()V
    .locals 5

    .line 81
    iget-object v0, p0, Lo/a221;->'`:Lo/a2;

    .line 4024
    iget-object v0, v0, Lo/a2;->_:Lexchange/sovereignwallet/mui/MainActivity;

    .line 81
    invoke-virtual {v0}, Landroid/content/Context;->getContentResolver()Landroid/content/ContentResolver;

    move-result-object v0

    const-string v1, "development_settings_enabled"

    const/4 v2, 0x0

    invoke-static {v0, v1, v2}, Landroid/provider/Settings$Secure;->getInt(Landroid/content/ContentResolver;Ljava/lang/String;I)I

    move-result v0

    .line 82
    const/4 v1, 0x0

    if-ne v0, v1, :cond_0

    .line 83
    iget-object v0, p0, Lo/a221;->'`:Lo/a2;

    .line 6024
    iget-object v3, v0, Lo/a2;->_:Lexchange/sovereignwallet/mui/MainActivity;

    .line 83
    .line 6056
    const v0, 0x7f0b0057

    invoke-virtual {v3, v0}, Landroid/content/Context;->getString(I)Ljava/lang/String;
```



Developer options

On

Run WebView renderers separately

Automatic system updates

Demo mode

Debugging

USB debugging
Debug mode when USB is connected

Revoke USB debugging authorizations

Bug report shortcut
Show a button in the power menu for taking a bug report

Select mock location app

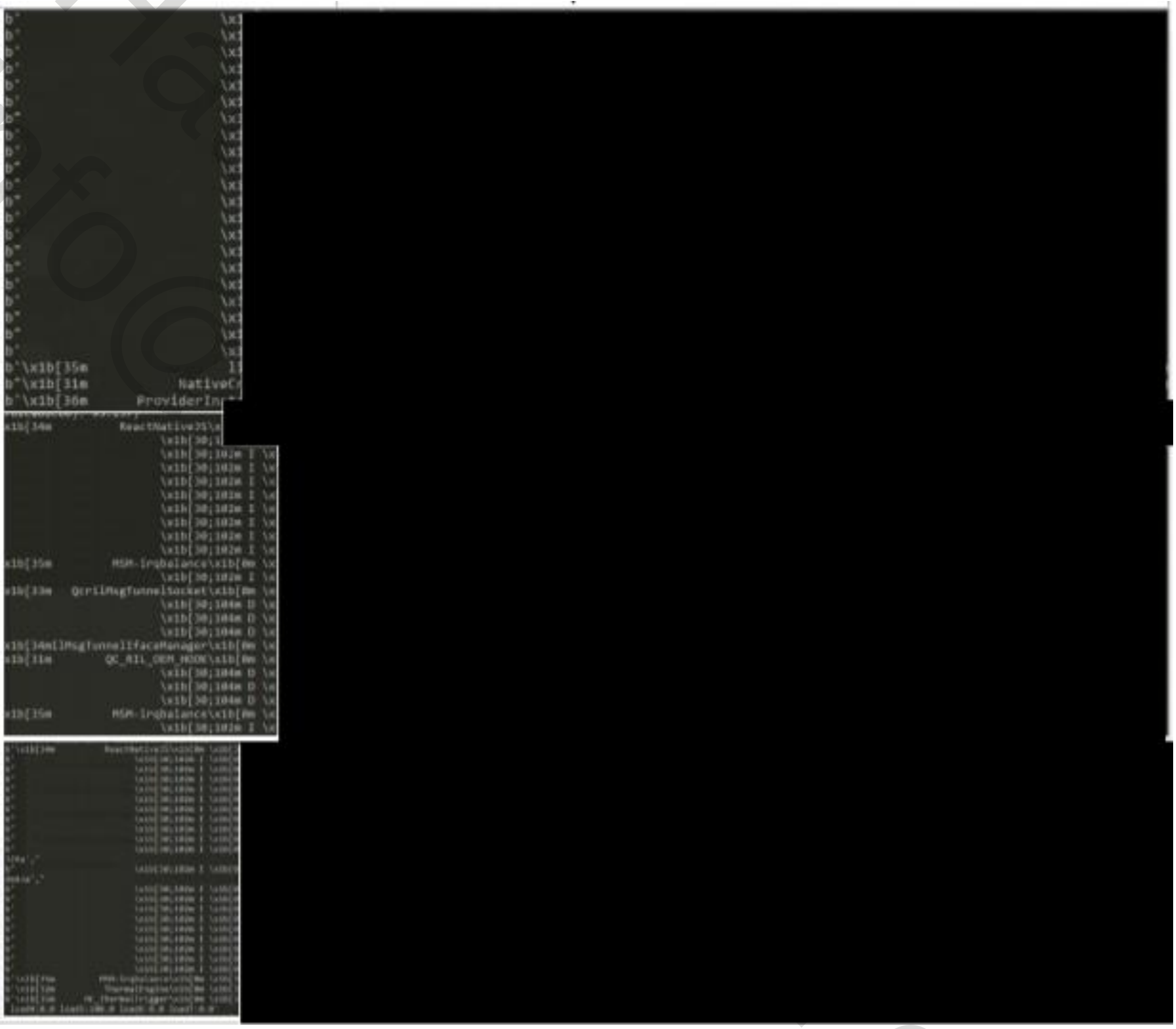
[redacted]

Recommendations	1. Obfuscate source code 2. Import function for checking modification of source code

Critical bug in money transfer

#2	Description	Type: Real
The application crashes, when money is transferred between two different accounts		
Evidences		
Steps to reproduce: – Log in application – Enter into money transfer – Get valet address with help QR-code – Input data – Press send button		
Recommendations	Check transfer work on different version of Android.	

Personal data in logs

#3	Description	Type: Real
Personal data can be stolen from application logs. Often Developers leave debugging information publicly. So any application with READ_LOGS permission can access those logs and can gain sensitive information through that.		
Evidences		
Perform pidcat 		
Recommendations	Turn off READ_LOGS permissions.	

```
b'                                                   \x1
b'                                                   \x1
b'                                                   \x1
b'                                                   \x1
b'                                                   \x1
b'                                                   \x1
b'                                                   \x1
b'                                                   \x1
b'                                                   \x1
b'                                                   \x1
b'                                                   \x1
b'                                                   \x1
b'                                                   \x1
b'                                                   \x1
b'                                                   \x1
b'                                                   \x1
b'                                                   \x1
b'                                                   \x1
b'                                                   \x1
b'                                                   \x1
b'                                                   \x1
b'\x1b[35m                                           11
b'\x1b[31m                                     NativeC
b'\x1b[36m                                ProviderIn
x1b[34m                       ReactNativeJS\x
                                   \x1b[30;1
                                   \x1b[30;102m I \x
                                   \x1b[30;102m I \x
                                   \x1b[30;102m I \x
                                   \x1b[30;102m I \x
                                   \x1b[30;102m I \x
                                   \x1b[30;102m I \x
                                   \x1b[30;102m I \x
                                   \x1b[30;102m I \x
x1b[35m                 MSM-Irqbalance\x1b[0m \x
                                   \x1b[30;102m I \x
x1b[33m      QcrilMsgTunnelSocket\x1b[0m \x
                                   \x1b[30;104m D \x
                                   \x1b[30;104m D \x
                                   \x1b[30;104m D \x
x1b[34mIlMsgTunnelIfaceManager\x1b[0m \x
x1b[31m              QC_RIL_OEM_HOOK\x1b[0m \x
                                   \x1b[30;104m D \x
                                   \x1b[30;104m D \x
                                   \x1b[30;104m D \x
x1b[35m                 MSM-Irqbalance\x1b[0m \x
                                   \x1b[30;102m I \x
[illegible]
```

[redacted]

Absence of source code obfuscation

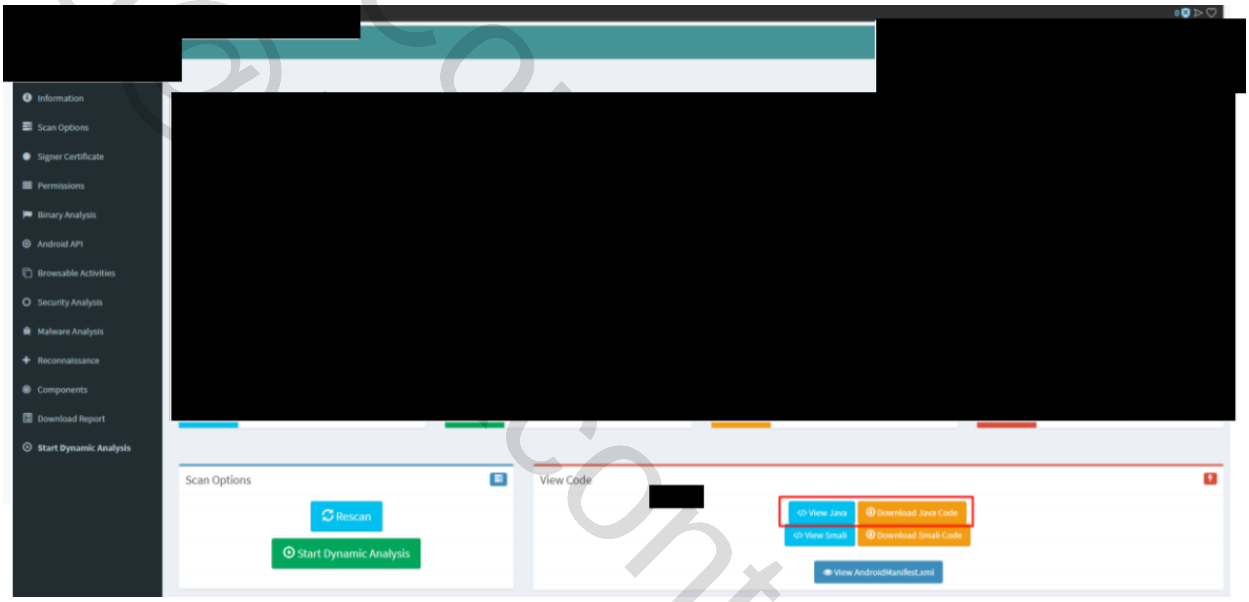
#4	Description	Type: Real

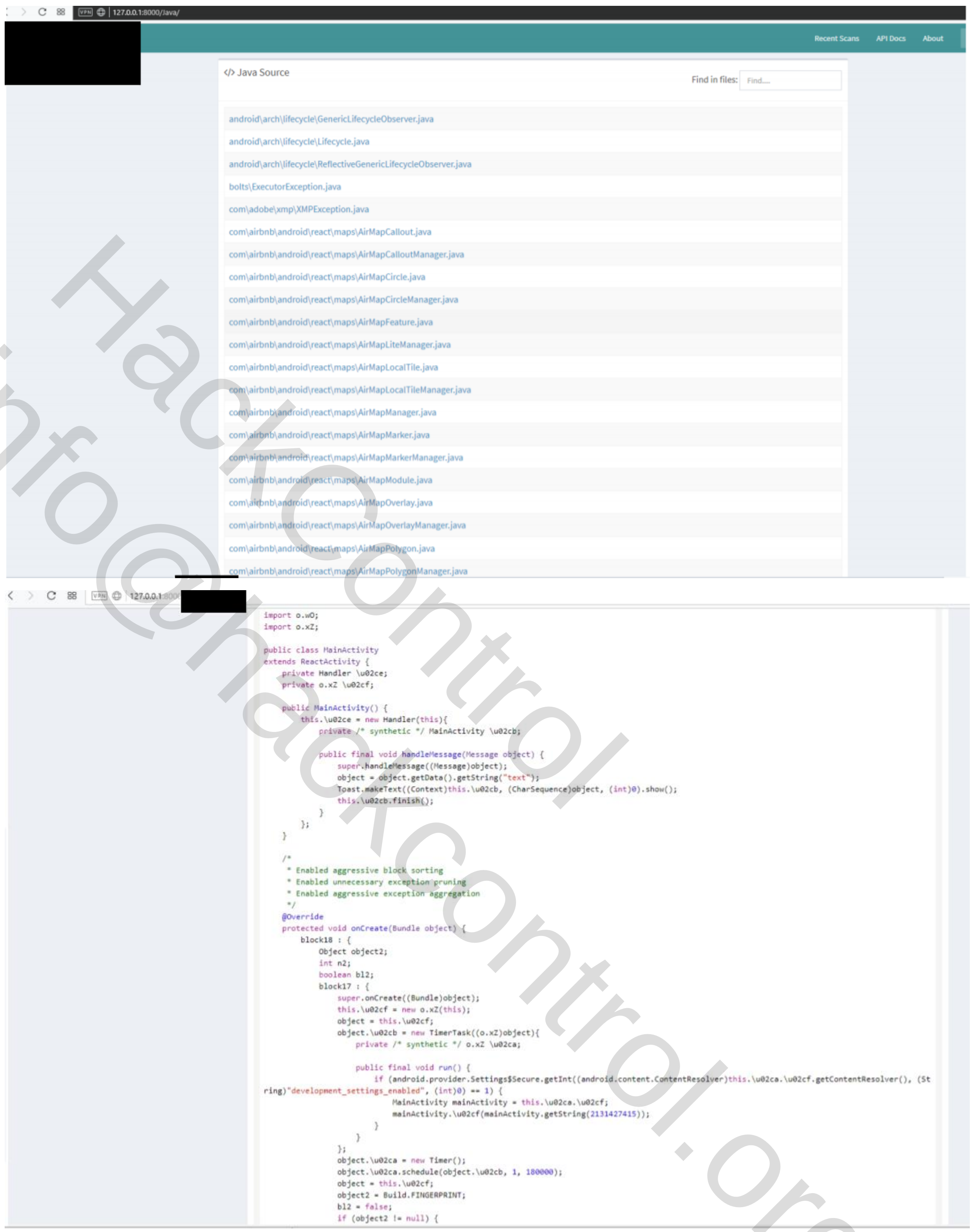
Android Application are delivered through an. apk file format which can be exploited by someone to see all the code contained in it. Below are scenarios of reverse engineering an application:

- A hacker can analyze and determine which defensive measure are implemented in the app and also find a way to bypass those mechanisms.
- Also a hacker can also insert the malicious code, recompile it and deliver to normal users.
- For example, gaming apps which have some features unlocked are widely downloaded by youngster through insecure sources (sometimes through Google PlayStore as well). Most of those modified apps contain malware and some contain advertising to gain profit from those users. This can lead to code analysis.

Evidences

- Upload APK into MobSF
- Click button Java code


[redacted]
[redacted]
[redacted]
[redacted]



[redacted]

android\arch\lifecycle\GenericLifecycleObserver.java
android\arch\lifecycle\Lifecycle.java
android\arch\lifecycle\ReflectiveGenericLifecycleObserver.java
bolts\ExecutorException.java
com\adobe\xmp\XMPException.java
com\airbnb\android\react\maps\AirMapCallout.java
com\airbnb\android\react\maps\AirMapCalloutManager.java
com\airbnb\android\react\maps\AirMapCircle.java
com\airbnb\android\react\maps\AirMapCircleManager.java
com\airbnb\android\react\maps\AirMapFeature.java
com\airbnb\android\react\maps\AirMapLiteManager.java
com\airbnb\android\react\maps\AirMapLocalTile.java
com\airbnb\android\react\maps\AirMapLocalTileManager.java
com\airbnb\android\react\maps\AirMapManager.java
com\airbnb\android\react\maps\AirMapMarker.java
com\airbnb\android\react\maps\AirMapMarkerManager.java
com\airbnb\android\react\maps\AirMapModule.java
com\airbnb\android\react\maps\AirMapOverlay.java
com\airbnb\android\react\maps\AirMapOverlayManager.java
com\airbnb\android\react\maps\AirMapPolygon.java
com\airbnb\android\react\maps\AirMapPolygonManager.java

[redacted]

```
import o.wO;
import o.xZ;

public class MainActivity
extends ReactActivity {
    private Handler \u02ce;
    private o.xZ \u02cf;

    public MainActivity() {
        this.\u02ce = new Handler(this){
            private /* synthetic */ MainActivity \u02cb;

            public final void handleMessage(Message object) {
                super.handleMessage((Message)object);
                object = object.getData().getString("text");
                Toast.makeText((Context)this.\u02cb, (CharSequence)object, (int)0).show();
                this.\u02cb.finish();
            }
        };
    }

    /*
     * Enabled aggressive block sorting
     * Enabled unnecessary exception pruning
     * Enabled aggressive exception aggregation
     */
    @Override
    protected void onCreate(Bundle object) {
        block18 : {
            Object object2;
            int n2;
            boolean bl2;
            block17 : {
                super.onCreate((Bundle)object);
                this.\u02cf = new o.xZ(this);
                object = this.\u02cf;
                object.\u02cb = new TimerTask((o.xZ)object){
                    private /* synthetic */ o.xZ \u02ca;

                    public final void run() {
                        if (android.provider.Settings$Secure.getInt((android.content.ContentResolver)this.\u02ca.\u02cf.getContentResolver(), (St
ring)"development_settings_enabled", (int)0) == 1) {
                            MainActivity mainActivity = this.\u02ca.\u02cf;
                            mainActivity.\u02cf(mainActivity.getString(2131427415));
                        }
                    }
                };
                object.\u02ca = new Timer();
                object.\u02ca.schedule(object.\u02cb, 1, 180000);
                object = this.\u02cf;
                object2 = Build.FINGERPRINT;
                bl2 = false;
                if (object2 != null) {
```

Recommendations	Application Code can be obfuscated with help of Proguard or DashO, but it is only able to slow down a hacker from reverse engineering android application, obfuscation doesn't prevent reverse engineering.

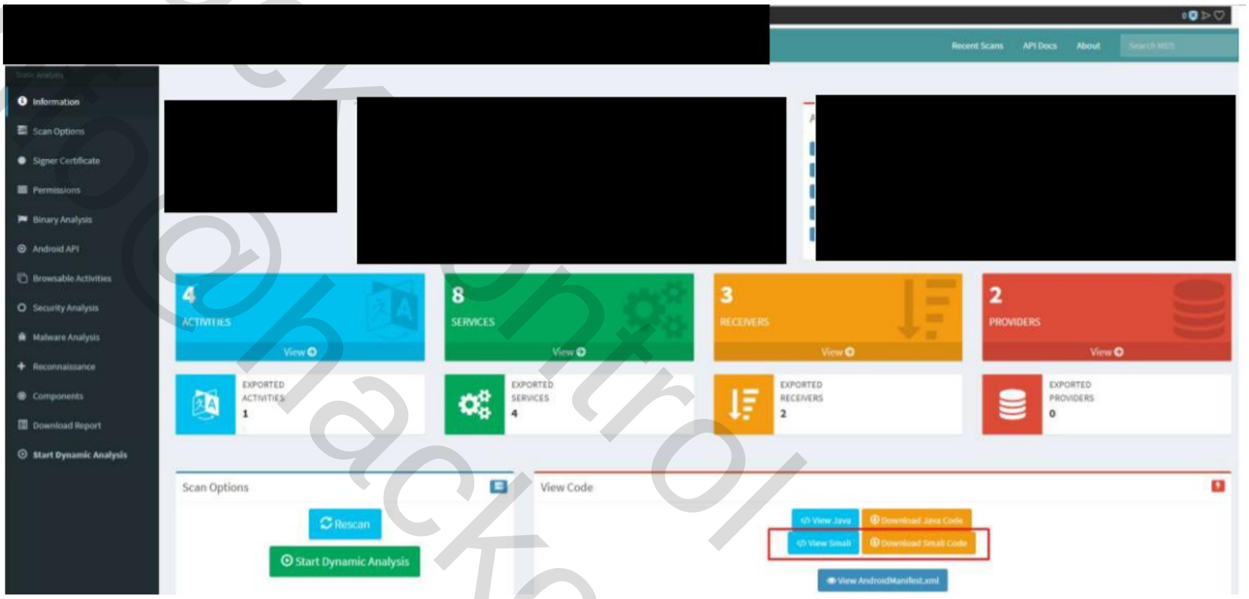
Check modify source code

#5	Description	Type: Real

Binary protections prevent an adversary from modifying the underlying code or behavior to disable or add additional functionality on behalf of the adversary. This is likely to occur if an application stores, transmits, or processes personally identifiable information (PII) or other sensitive information assets like passwords or credit cards. Code modification often takes the form of repackaging or insertion of malware into existing mobile apps.

Evidences

Upload APK into MobSF code

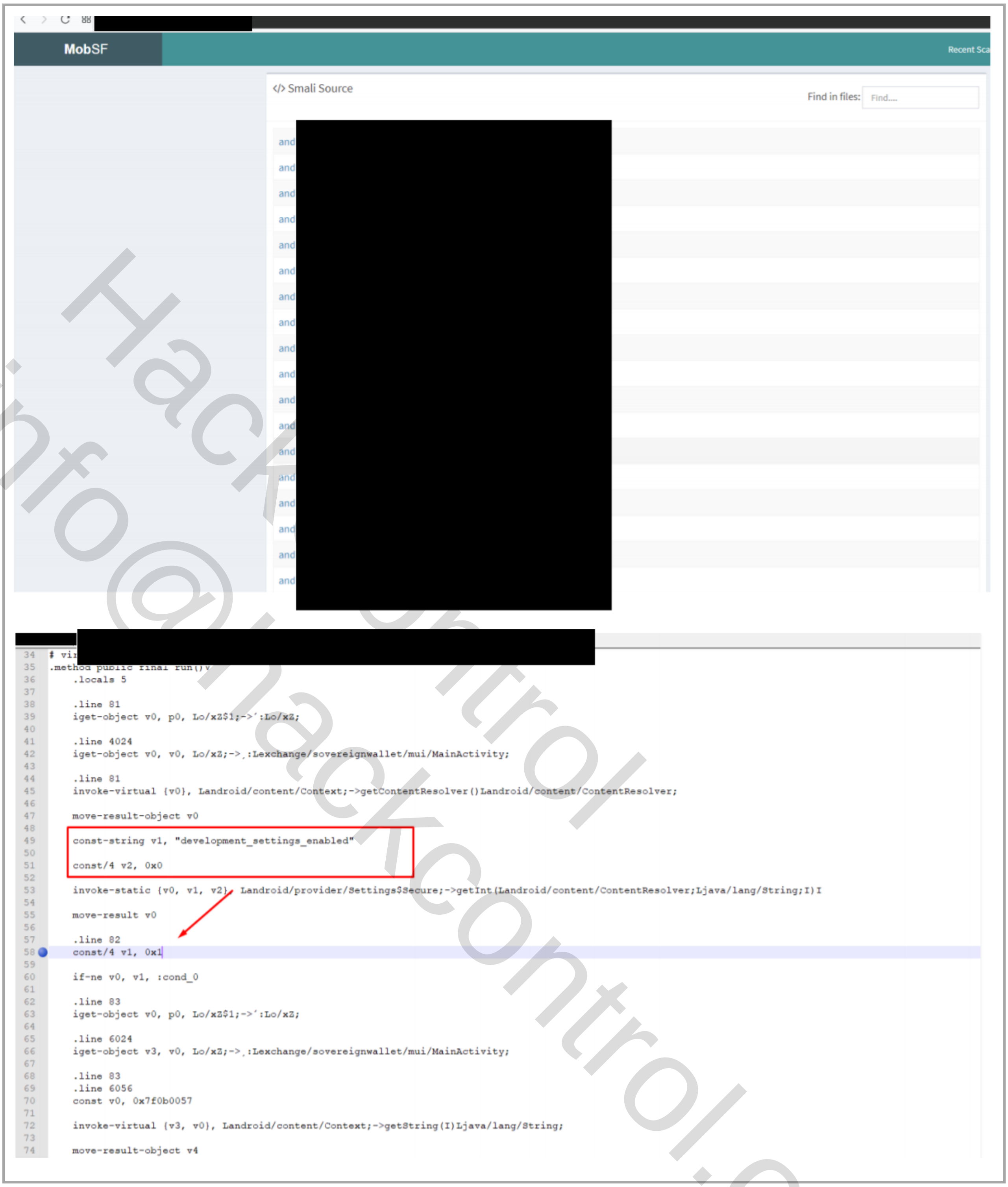


[redacted]

[redacted]

[redacted]

[redacted]



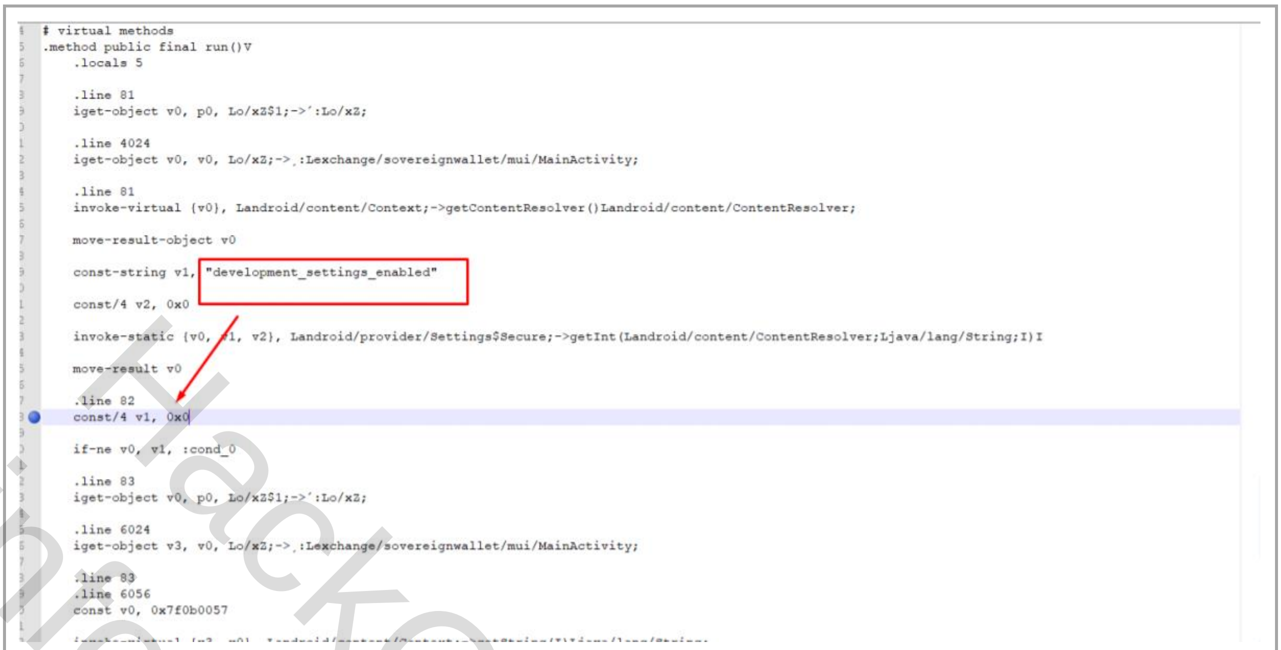
Smali Source

Find in files: Find....

and [redacted]
and
and
and
and
and
and
and
and
and
and
and
and
and
and
and
and
and

[redacted]

```
# vi[redacted]
.method public final run()V
    .locals 5

    .line 81
    iget-object v0, p0, Lo/xZ$1;->':Lo/xZ;

    .line 4024
    iget-object v0, v0, Lo/xZ;->¸:Lexchange/sovereignwallet/mui/MainActivity;

    .line 81
    invoke-virtual {v0}, Landroid/content/Context;->getContentResolver()Landroid/content/ContentResolver;

    move-result-object v0

    const-string v1, "development_settings_enabled"

    const/4 v2, 0x0

    invoke-static {v0, v1, v2}, Landroid/provider/Settings$Secure;->getInt(Landroid/content/ContentResolver;Ljava/lang/String;I)I

    move-result v0

    .line 82
    const/4 v1, 0x1

    if-ne v0, v1, :cond_0

    .line 83
    iget-object v0, p0, Lo/xZ$1;->':Lo/xZ;

    .line 6024
    iget-object v3, v0, Lo/xZ;->¸:Lexchange/sovereignwallet/mui/MainActivity;

    .line 83
    .line 6056
    const v0, 0x7f0b0057

    invoke-virtual {v3, v0}, Landroid/content/Context;->getString(I)Ljava/lang/String;

    move-result-object v4
```



```
# virtual methods
.method public final run()V
    .locals 5

    .line 81
    iget-object v0, p0, Lo/xZ$1;->':Lo/xZ;

    .line 4024
    iget-object v0, v0, Lo/xZ;->¸:Lexchange/sovereignwallet/mui/MainActivity;

    .line 81
    invoke-virtual {v0}, Landroid/content/Context;->getContentResolver()Landroid/content/ContentResolver;

    move-result-object v0

    const-string v1, "development_settings_enabled"

    const/4 v2, 0x0

    invoke-static {v0, v1, v2}, Landroid/provider/Settings$Secure;->getInt(Landroid/content/ContentResolver;Ljava/lang/String;I)I

    move-result v0

    .line 82
    const/4 v1, 0x0

    if-ne v0, v1, :cond_0

    .line 83
    iget-object v0, p0, Lo/xZ$1;->':Lo/xZ;

    .line 6024
    iget-object v3, v0, Lo/xZ;->¸:Lexchange/sovereignwallet/mui/MainActivity;

    .line 83
    .line 6056
    const v0, 0x7f0b0057

    invoke-virtual {v3, v0}, Landroid/content/Context;->getString(I)Ljava/lang/String;
```

Recommendations	Adds tamper detection to let your application react accordingly if a hacker has tried to modify it or is accessing it illegitimately.

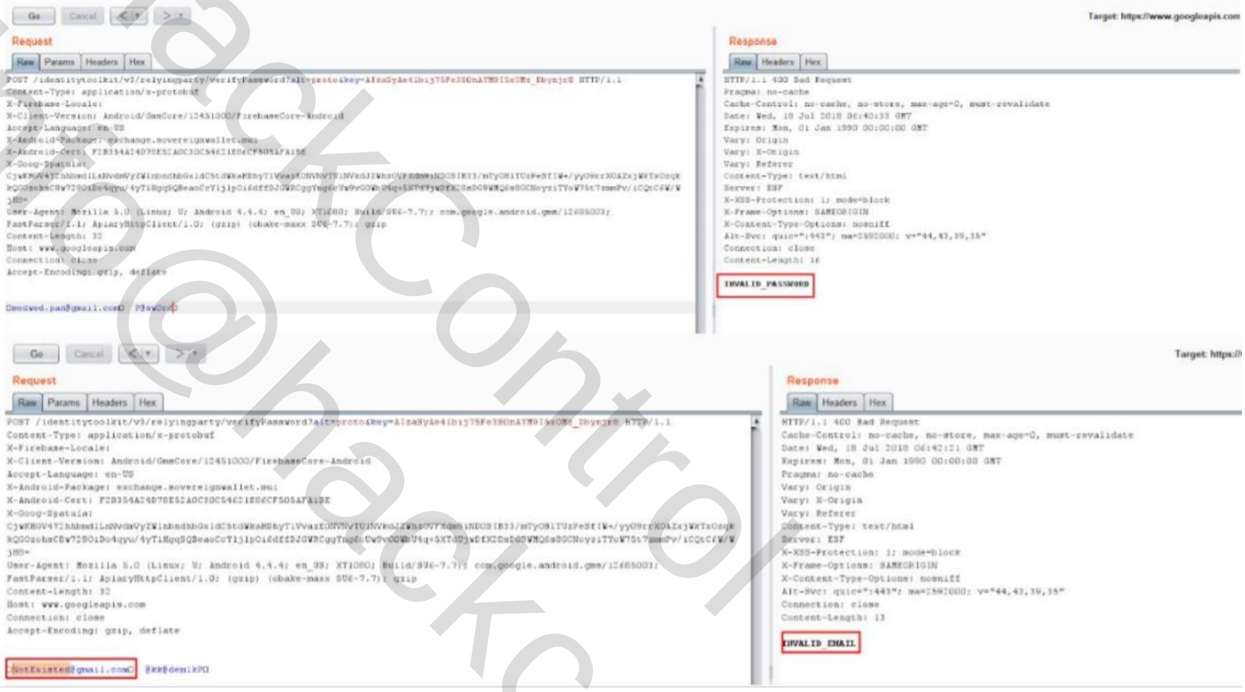
User enumeration

#6	Description	Type: Real

Authorization header contains both email address and authentication token. It was discovered that by sending existing and not existing email address it is possible to enumerate valid users because of different responses. Before the token is checked, the application looks up if email address belongs to a registered user. If the user is not registered, an error “User not found” occurs.

Evidences

- Check response from existing and not existing user during authorization



```
POST /identitytoolkit/v3/relyingparty/verifyPassword?alt=proto&key=AIzaSyAe4Ibij75Fe3SOnAYM9ISxOMs_DbynjcO HTTP/1.1
Content-Type: application/x-protobuf
X-Firebase-Locale:
X-Client-Version: Android/GmsCore/12451000/FirebaseCore-Android
Accept-Language: en-US
X-Android-Package: exchange.sovereignwallet.mui
X-Android-Cert: F2B354A24D70E52A0C30C5462IE06CF505AFA1DE
X-Goog-Spatula:
[illegible]
[illegible]
jHS=
User-Agent: Mozilla 5.0 (Linux; U; Android 4.4.4; en_US; XT1080; Build/SU6-7.7); com.google.android.gms/12685003;
FastParser/1.1; ApiaryHttpClient/1.0; (gzip) (obake-maxx SU6-7.7); gzip
Content-Length: 32
Host: www.googleapis.com
Connection: close
Accept-Encoding: gzip, deflate

Dmedved.pan@gmail.com P@ssw0rd
```

```
HTTP/1.1 400 Bad Request
Pragma: no-cache
Cache-Control: no-cache, no-store, max-age=0, must-revalidate
Date: Wed, 18 Jul 2018 06:40:33 GMT
Expires: Mon, 01 Jan 1990 00:00:00 GMT
Vary: Origin
Vary: X-Origin
Vary: Referer
Content-Type: text/html
Server: ESF
X-XSS-Protection: 1; mode=block
X-Frame-Options: SAMEORIGIN
X-Content-Type-Options: nosniff
Alt-Svc: quic=":443"; ma=2592000; v="44,43,39,35"
Connection: close
Content-Length: 16

INVALID_PASSWORD
```

```
POST /identitytoolkit/v3/relyingparty/verifyPassword?alt=proto&key=AIzaSyAe4Ibij75Fe3SOnAYM9ISxOMs_DbynjcO HTTP/1.1
Content-Type: application/x-protobuf
X-Firebase-Locale:
X-Client-Version: Android/GmsCore/12451000/FirebaseCore-Android
Accept-Language: en-US
X-Android-Package: exchange.sovereignwallet.mui
X-Android-Cert: F2B354A24D70E52A0C30C5462IE06CF505AFA1DE
X-Goog-Spatula:
[illegible]
[illegible]
jHS=
User-Agent: Mozilla 5.0 (Linux; U; Android 4.4.4; en_US; XT1080; Build/SU6-7.7); com.google.android.gms/12685003;
FastParser/1.1; ApiaryHttpClient/1.0; (gzip) (obake-maxx SU6-7.7); gzip
Content-Length: 32
Host: www.googleapis.com
Connection: close
Accept-Encoding: gzip, deflate

NotExisted@gmail.com @kk@demik PO
```

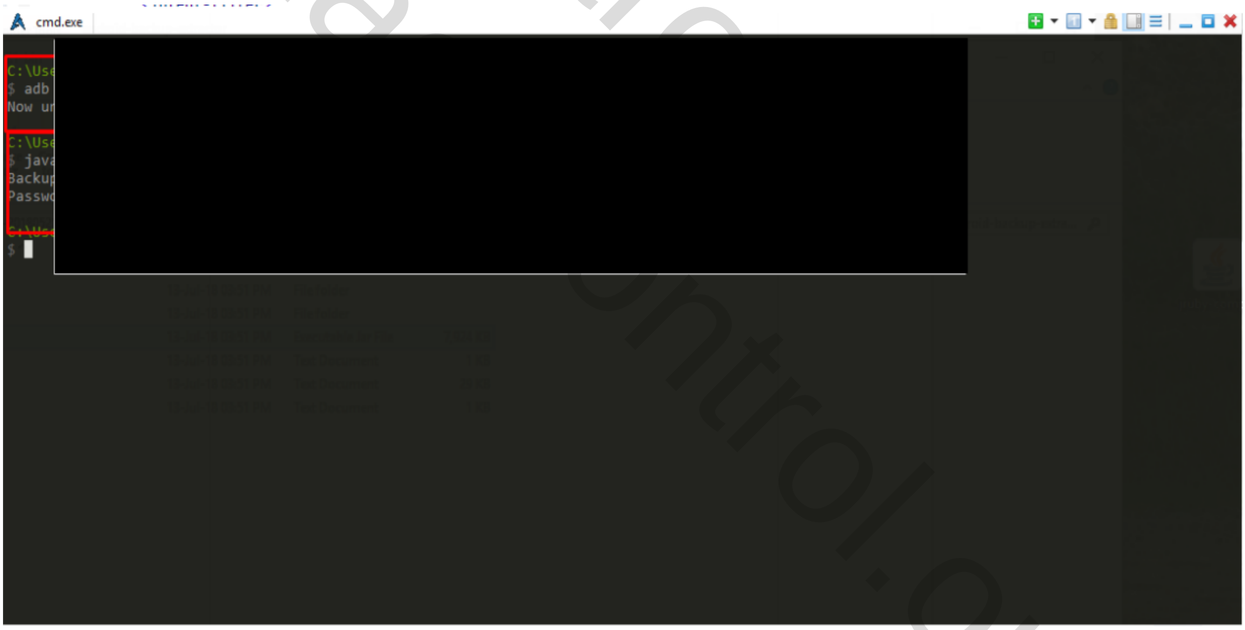
```
HTTP/1.1 400 Bad Request
Cache-Control: no-cache, no-store, max-age=0, must-revalidate
Date: Wed, 18 Jul 2018 06:42:21 GMT
Expires: Mon, 01 Jan 1990 00:00:00 GMT
Pragma: no-cache
Vary: Origin
Vary: X-Origin
Vary: Referer
Content-Type: text/html
Server: ESF
X-XSS-Protection: 1; mode=block
X-Frame-Options: SAMEORIGIN
X-Content-Type-Options: nosniff
Alt-Svc: quic=":443"; ma=2592000; v="44,43,39,35"
Connection: close
Content-Length: 13

INVALID_EMAIL
```

Recommendations	It is recommended to provide the same response irrespective of whether password was incorrect or username does not exist.

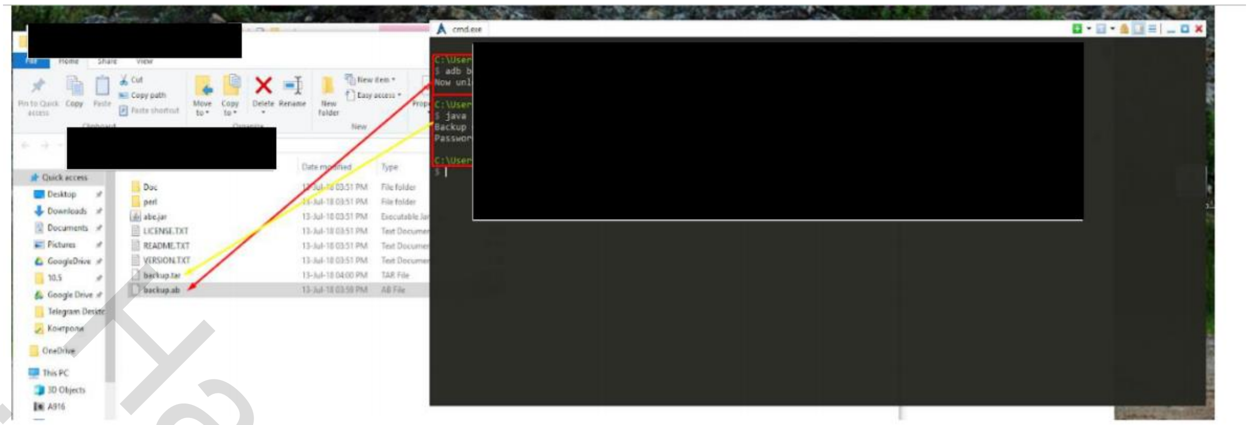
Application data can be backup

<table>
<tr><th>#7</th><th>Description</th><th>Type: Real</th></tr>
<tr><td colspan="3">This flag allows anyone to backup your application data via adb. It allows users who have enabled USB debugging to copy application data of the device.</td></tr>
<tr><th colspan="3">Evidences</th></tr>
<tr><td colspan="3">Checks flag “android:allowBackup” adb backup –f backup.ab -apk exchange.________________

```
<uses-permission android:name="com.huawei.android.launcher.permission.WRITE_SETTINGS" />
<us [redacted]
<us
<us
<us
<us
<us
<ap
on" and
t" andr
```



```
C:\Use [redacted]
$ adb
Now u
C:\Use
$ java
Backup
Passw
C:\Use
$
```
</td></tr>
</table>



[redacted]

[redacted]

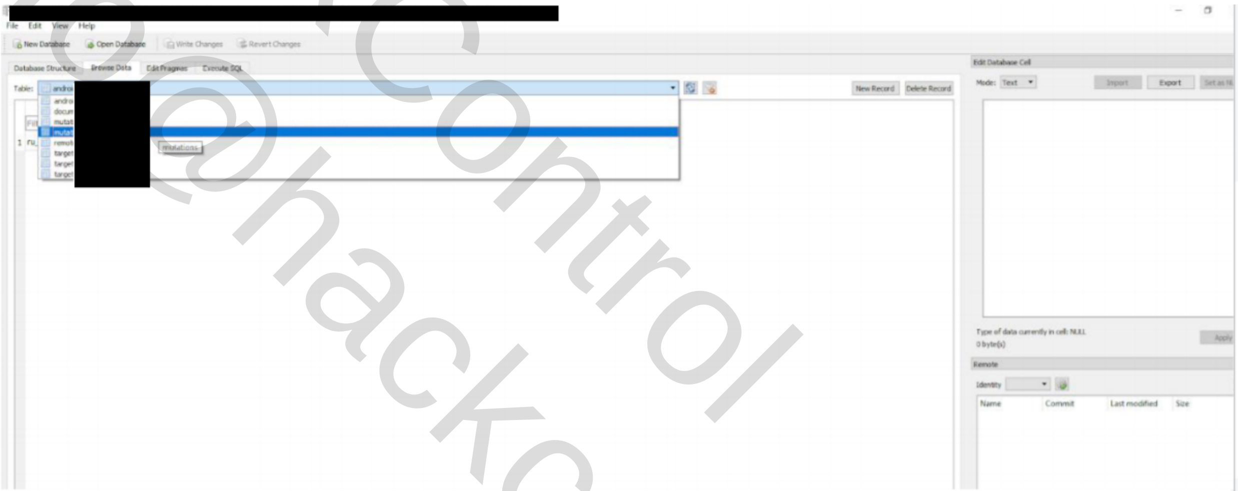
	Date modified	Type
Doc	13-Jul-18 03:51 PM	File folder
perl	13-Jul-18 03:51 PM	File folder
abe.jar	13-Jul-18 03:51 PM	Executable Jar
LICENSE.TXT	13-Jul-18 03:51 PM	Text Documen
README.TXT	13-Jul-18 03:51 PM	Text Documen
VERSION.TXT	13-Jul-18 03:51 PM	Text Documen
backup.tar	13-Jul-18 04:00 PM	TAR File
backup.ab	13-Jul-18 03:59 PM	AB File

```
C:\User
$ adb b
Now unl

C:\User
$ java
Backup
Passwor

C:\User
$ |
```

[redacted]

C:\Users\Us [redacted]

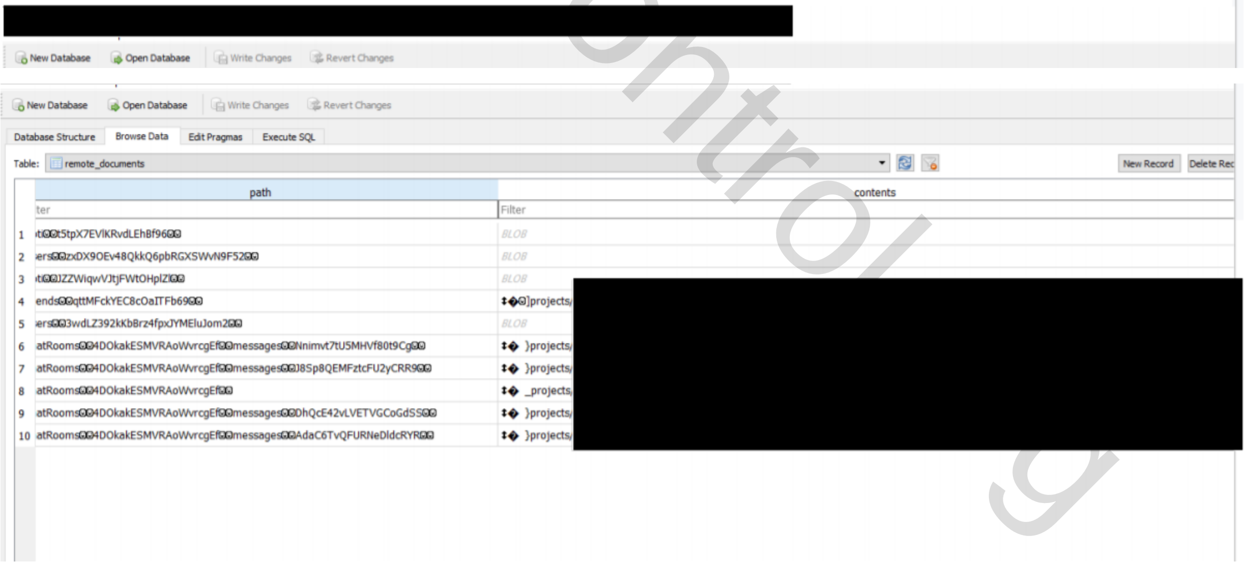
[redacted]



[redacted]

[redacted]

Type of data currently in cell: NULL
0 byte(s)



[redacted]

Table: remote_documents

	path	contents
	ter	Filter
1	ti⊠⊠t5tpX7EVlKRvdLEhBf96⊠⊠	BLOB
2	ers⊠⊠zxDX9OEv48QkkQ6pbRGXSWvN9F52⊠⊠	BLOB
3	ti⊠⊠JZZWiqwVJtjFWtOHplZl⊠⊠	BLOB
4	ends⊠⊠qttMFckYEC8cOaITFb69⊠⊠	‡�⊠]projects/ [redacted]
5	ers⊠⊠3wdLZ392kKbBrz4fpxJYMEluJom2⊠⊠	BLOB
6	atRooms⊠⊠4DOkakESMVRAoWvrcgEf⊠⊠messages⊠⊠Nnimvt7tU5MHVf80t9Cg⊠⊠	‡� }projects/
7	atRooms⊠⊠4DOkakESMVRAoWvrcgEf⊠⊠messages⊠⊠J8Sp8QEMFztcFU2yCRR9⊠⊠	‡� }projects/
8	atRooms⊠⊠4DOkakESMVRAoWvrcgEf⊠⊠	‡� _projects/
9	atRooms⊠⊠4DOkakESMVRAoWvrcgEf⊠⊠messages⊠⊠DhQcE42vLVETVGCoGdSS⊠⊠	‡� }projects/
10	atRooms⊠⊠4DOkakESMVRAoWvrcgEf⊠⊠messages⊠⊠AdaC6TvQFURNeDldcRYR⊠⊠	‡� }projects/

Recommendations	Set value in flag android:allowBackup=”false”

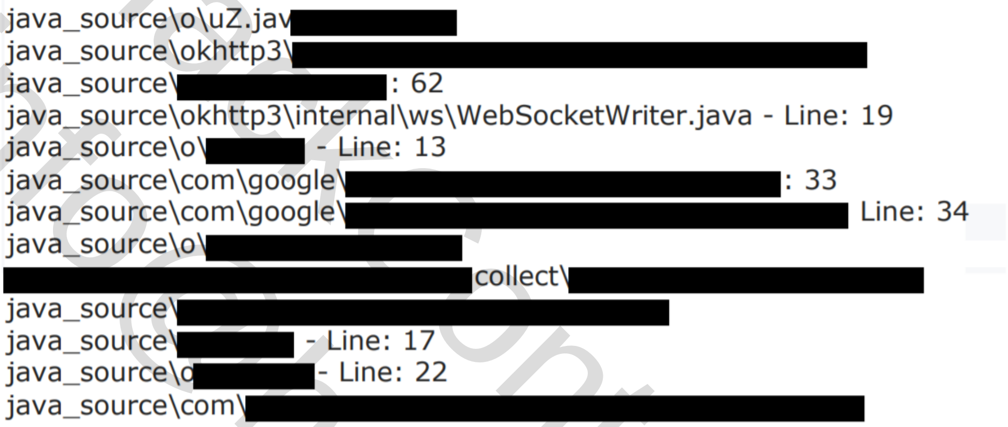
🟩 No certificate and public key pinning

#8	Description	Type: Real
There was no Certificate and Public Key Pinning found during the mobile application test. Absence of the mechanism makes it more convenient and faster to intercept and decrypt traffic between an application and a server. Pinning is the process of associating a host with their expected X509 certificate or public key. Once a certificate or public key is known or seen for a host, the certificate or public key is associated or 'pinned' to the host.		
Evidences		
Steps to reproduce: - Configure the Burp Proxy listener - Configure your device to use the proxy - Test the configuration. If the traffic can be captured and decrypted the Pinning mechanism is not implemented		
Recommendations	It is recommended to implement Certificate and Public Key Pinning. For more details please visit https://www.owasp.org/index.php/Certificate_and_Public_Key_Pinning	

Http without headers

#9	Description	Type: Real
Unless directed otherwise, browsers may store a local cached copy of content received from web servers. Some browsers, including Internet Explorer, cache content accessed via HTTPS. If sensitive information in application responses is stored in the local cache, then this may be retrieved by other users who have access to the same computer at a future time.(Cache-control: nostore, Pragma: no-cache)		
Recommendations	Add the following headers: Cache-control: no-store Pragma: no-cache	

Out of date library

<table>
<tr><th>#10</th><th>Description</th><th>Type: Real</th></tr>
<tr><td colspan="3">When software generates predictable values in a context requiring unpredictability, it may be possible for an attacker to guess the next value that will be generated, and use this guess to impersonate another user or access sensitive information. As the java.util.Random class relies on a pseudorandom number generator, this class and relating java.lang.Math.random() method should not be used for security-critical applications or for protecting sensitive data java.util.Random. This package is flawed and produces predictable values for any given seed which are easily reproducible once the starting seed is identified.

java_source\o\uZ.jav[redacted]
java_source\okhttp3\[redacted]
java_source\[redacted]: 62
java_source\okhttp3\internal\ws\WebSocketWriter.java - Line: 19
java_source\o\[redacted] - Line: 13
java_source\com\google\[redacted]: 33
java_source\com\google\[redacted] Line: 34
java_source\o\[redacted]
[redacted]collect\[redacted]
java_source\[redacted]
java_source\[redacted] - Line: 17
java_source\o[redacted]- Line: 22
java_source\com\[redacted]</td></tr>
<tr><td>Recommendations</td><td colspan="2">Use library java.security.SecureRandom, read more https://resources.infosecinstitute.com/randomnumber-generation-java/</td></tr>
</table>

Bugs in key word display

#11	Description	Type: Real
Keywords has incorrect position		
Recommendations	Check position of key words	

Export components

#12	Description	Type: Real

A Service is found to be shared with other apps on the device therefore leaving it accessible to any other application on the device. The presence of intent-filter indicates that the Service is explicitly exported. A Broadcast Receiver is found to be shared with other apps on the device therefore leaving it accessible to any other application on the device. It is protected by a permission which is not defined in the analyzed application. As a result, the protection level of the permission should be checked where it is defined. If it is set to normal or dangerous, a malicious application can request and obtain the permission and interact with the component. If it is set to signature, only applications signed with the same certificate can obtain the permission.

Evidences

- run [redacted] [redacted] [redacted].info -package exchange.____________________.mui
- run [redacted]o -a exchange.____________________.mui

```
dz> run ap[redacted]
Package: e
  io.inver
    Permis
  io.inver
    Permis
  com.goog
    Permis
  com.goog
    Permis
```

```
dz> run app[redacted]
Package: ex
  com.googl
    Permiss
  com.googl
    Permiss
```

Recommendations	Set flag android:exported=true in:

	- io.invertase.firebase.messaging.RNFirebaseMessagingService - io.invertase.firebase.messaging.RNFirebaseInstanceIdService - com.google.firebase.messaging.FirebaseMessa gingService - - com.google.firebase.iid.FirebaseInstanceIdSer vice - - com.google.android.gms.measurement.AppMeasurementInstallReferrerReceiver - - com.google.firebase.iid.FirebaseInstanceIdRec eiver

Vulnerability in webview

#13	Description	Type: Real

This vulnerability can lead for privilege escalation in Android < 4.2's WebView component that arises when untrusted Javascript code is executed by a WebView that has one or more Interfaces added to it. The untrusted Javascript code can call into the Java Reflection APIs exposed by the Interface and execute arbitrary commands. Some distributions of the Android Browser app have an addJavascriptInterface call tacked on, and thus are vulnerable to RCE. The Browser app in the Google APIs 4.1.2 release of Android is known to be vulnerable. A secondary attack vector involves the WebViews embedded inside a large number of Android applications. Ad integrations are perhaps the worst offender here. If you can MITM the WebView's HTTP connection, or if you can get a persistent XSS into the page displayed in the WebView, then you can inject the html/js served by this module and get a shell. Note: Adding a .js to the URL will return plain javascript (no HTML markup). (https://www.rapid7.com/db/modules/exploit/android/browser/webview_addj avascriptinterface)

Evidences

```
$ grep -nr 'setAllowUniversalAccessFromFileURLs' java_source\
java_source\/com/facebook/react/views/webview/ReactWebViewManag er.java:227: public
void setAllowUniversalAccessFromFileURLs(WebView webView, boolean bl2) {
java_source\/com/facebook/react/views/webview/ReactWebViewManag er.java:228:
webView.getSettings().setAllowUniversalAccessFromFileURLs(bl2); $grep -nr
'setJavaScriptEnabled' java_source\
java_source\/com/facebook/react/views/webview/ReactWebViewManag er.java:242: public
void setJavaScriptEnabled(WebView webView, boolean bl2) {
java_source\/com/facebook/react/views/webview/ReactWebViewManag er.java:243:
webView.getSettings().setJavaScriptEnabled(bl2); $ grep -nr 'JavascriptInterface'
java_source\ java_source\/com/facebook/react/views/webview/ReactWebViewManag
er.java:21: android.webkit.JavascriptInterface
java_source\/com/facebook/react/views/webview/ReactWebViewManag er.java:58: import
android.webkit.JavascriptInterface;

java_source\/com/facebook/react/views/webview/ReactWebViewManag er.java:420:
this.addJavascriptInterface((Object)new If(this, this), "__REACT_WEB_VIEW_BRIDGE");
java_source\/com/facebook/react/views/webview/ReactWebViewManag er.java:426:
this.removeJavascriptInterface("__REACT_WEB_VIEW_BRIDGE");
java_source\/com/facebook/react/views/webview/ReactWebViewManag er.java:447:
@JavascriptInterface
```

Recommendations	In AndroidManifest.xml sets minSdk=24.

Appendix B. Automated Tools

Scope	Tools used

Application Security	Drozer Xposed 3.1.5 MobSF Dex2Jar JD-GUI BurpSuite 1.7.30 Nmap Sqlmap VisualCodeGrepper SonarQube
Devices	Samsung Note 8 – Android 8.0 Lenovo A968 – Android 4.4.2 Motorola Z Force – Android 8.0 Motorola Droid Turbo 2 – Android 7.0 Motorola Droid Maxx – Android 4.4.4