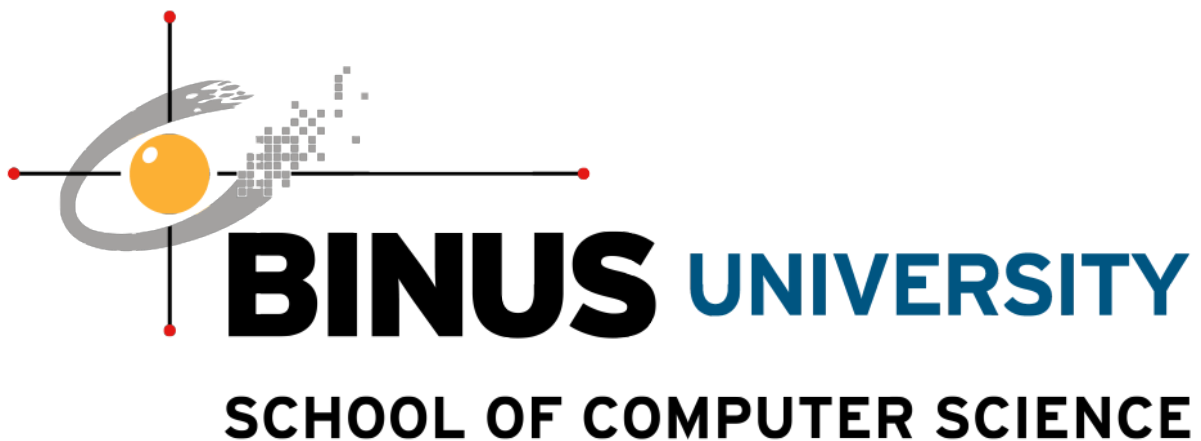
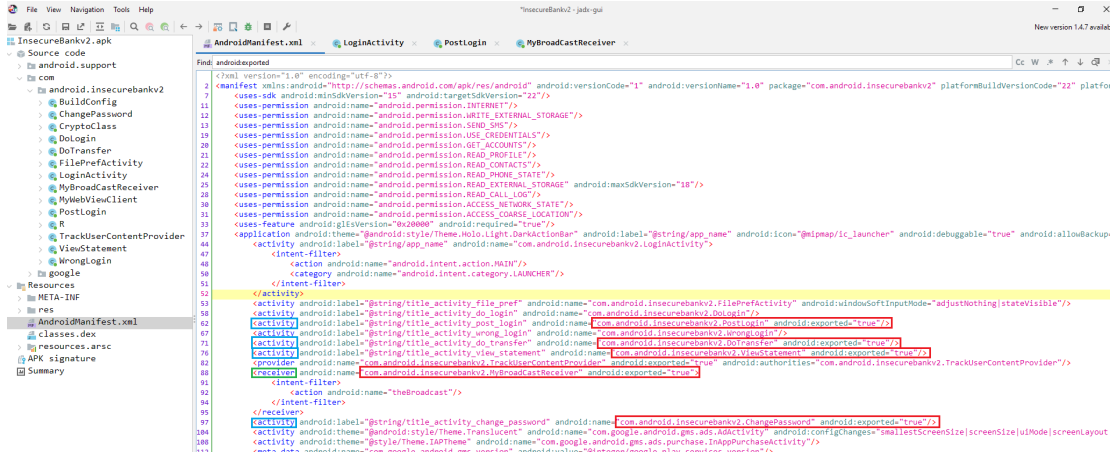


REPORT MOBILE PENTEST
INSECURE BANK V2 APP
KELOMPOK 11



Anggota:

- Mikael Wiryamanta Wijaya - 2540119633
- Satya Kusuma - 2540124740
- Khumaira Malik - 2540132080
- William Sulasman - 2540125421

Report 1	
Issue	Testing for Sensitive Functionality Exposure Through IPC (MSTG-PLATFORM-4)
Executive Summary	Kami menemukan beberapa aktivitas sensitif dalam aplikasi yang tidak memiliki validasi akses yang baik, sehingga beberapa aktivitas seperti <i>main page</i> , <i>transfer page</i> , dan <i>change password page</i> dapat diakses sebelum user melakukan login akun. Hal ini terjadi karena aplikasi mengizinkan ekspor aktivitas tanpa adanya validasi <i>permission</i> lebih lanjut.
POC [Static Analysis]	<p>1. Pertama disini saya menganalisa bagian AndroidManifest.xml menggunakan tools JADX, dan ditemukan terdapat android:exported="true" pada 4 activity dan 1 receiver.</p>  <p>2. Saya melakukan analisa lebih lanjut terhadap 4 activity diatas diantaranya adalah PostLogin, DoTransfer, ViewStatement, dan ChangePassword. fungsi-fungsi ini merupakan fungsi yang akan ditampilkan setelah user melakukan <i>login account</i></p> <p>3. Receiver MyBroadcastReceiver merupakan bagian yang akan melakukan <i>broadcast</i> terkait password yang berhasil diubah, namun disini bisa dilihat pada line 37, password yang dioutput sudah dalam bentuk <i>decrypted</i> sehingga nantinya akan berupa <i>plaintext</i></p>

```

package com.android.insecurebankv2;

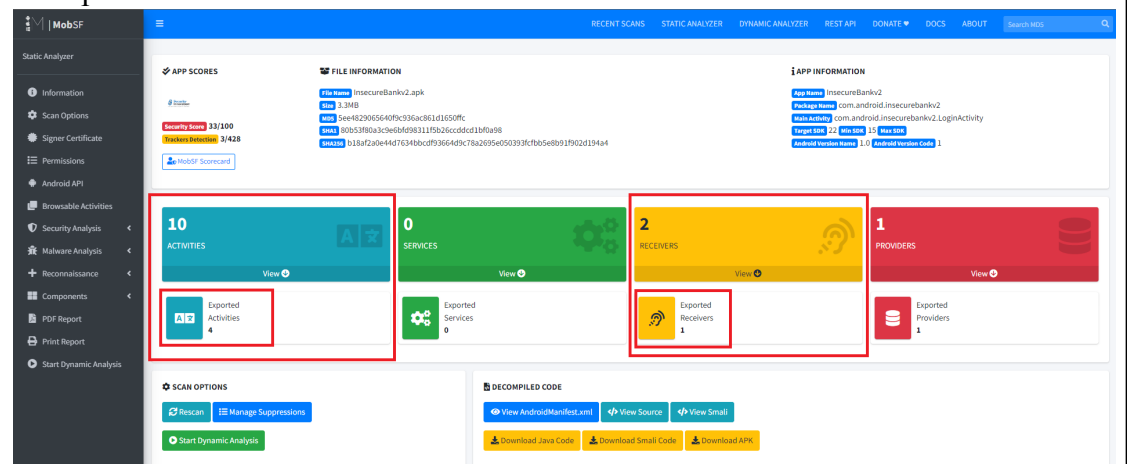
import android.content.BroadcastReceiver;
import android.content.Context;
import android.content.Intent;
import android.content.SharedPreferences;
import android.telephony.SmsManager;
import android.util.Base64;

/* loaded from: classes.dex */
public class MyBroadcastReceiver extends BroadcastReceiver {
    public static final String MY_PREFS = "mySharedPreferences";
    String usernameBase64ByteString;

    @Override // android.content.BroadcastReceiver
    public void onReceive(Context context, Intent intent) {
        String phn = intent.getStringExtra("phonenumber");
        String newpass = intent.getStringExtra("newpass");
        if (phn != null) {
            try {
                SharedPreferences settings = context.getSharedPreferences("mySharedPreferences", 1);
                String username = settings.getString("EncryptedUsername", null);
                byte[] usernameBase64Byte = Base64.decode(username, 0);
                this.usernameBase64ByteString = new String(usernameBase64Byte, "UTF-8");
                String password = settings.getString("supersecurePassword", null);
                CryptoClass crypt = new CryptoClass();
                String decryptedPassword = crypt.aesDecryptedString(password);
                String textPhoneno = phn.toString();
                String textMessage = "Updated Password from: " + decryptedPassword + " to: " + newpass;
                SmsManager smsManager = SmsManager.getDefault();
                System.out.println("For the changepassword - phonenumber: " + textPhoneno + " password is: " + textMessage);
                smsManager.sendTextMessage(textPhoneno, null, textMessage, null, null);
            } catch (Exception e) {
                e.printStackTrace();
            }
            return;
        }
        System.out.println("Phone number is null");
    }
}

```

4. Berikutnya saya menggunakan tools **MobSF**, untuk memeriksa lebih lanjut bagian-bagian yang *exported*, dan disini ditemukan hasil yang sama dengan yang ditampilkan oleh **JADX**



*Exported activity dan exported receiver mengakibatkan komponen tersebut bisa diakses oleh aplikasi lain, maka dari itu komponen yang diberikan akses exported harus menimbang kebutuhan dari komponen itu sendiri

POC [Dynamic Analysis]

1. Page **PostLogin**, **DoTransfer**, **ViewStatement**, dan **ChangePassword** hanya bisa diakses user ketika sudah melakukan *login account*. Karena *activity* ini *exported* maka saya menggunakan command **adb shell am start -n <activity>** pada cmd
 *command diatas merupakan perintah untuk *start activity*

```
C:\Windows\System32\cmd.exe

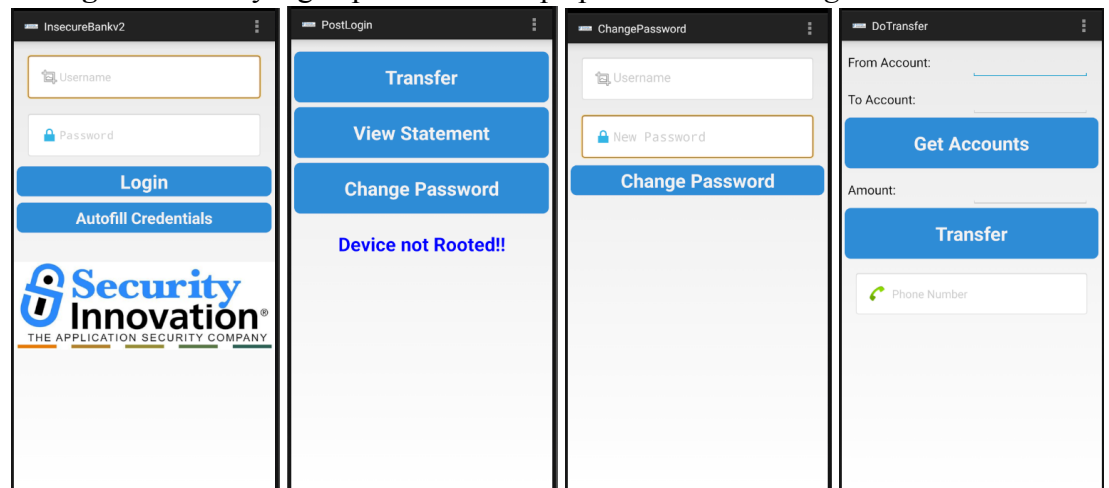
D:\Android-InsecureBankv2-master>adb shell am start -n com.android.insecurebankv2/com.android.insecurebankv2.PostLogin
Starting: Intent { cmp=com.android.insecurebankv2/.PostLogin }

D:\Android-InsecureBankv2-master>adb shell am start -n com.android.insecurebankv2/com.android.insecurebankv2.ChangePassword
Starting: Intent { cmp=com.android.insecurebankv2/.ChangePassword }

D:\Android-InsecureBankv2-master>adb shell am start -n com.android.insecurebankv2/com.android.insecurebankv2.DoTransfer
Starting: Intent { cmp=com.android.insecurebankv2/.DoTransfer }

D:\Android-InsecureBankv2-master>
```

2. Berikut *screen capture* dari halaman **PostLogin**, **DoTransfer**, dan **ChangePassword** yang dapat diakses tanpa perlu melakukan *login account*



3. Selanjutnya saya kembali memasukkan command **adb shell am broadcast -a theBroadcast -es phonenum "num" -es newpass "newpassword"**, command ini saya rancang dengan melihat fungsi tersebut pada **JADX**

```
AndroidManifest.xml | LoginActivity | PostLogin | BroadcastReceiver

24 <uses-permission android:name="android.permission.READ_PHONE_STATE"/>
25 <uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE" android:maxSdkVersion="18"/>
26 <uses-permission android:name="android.permission.READ_CALL_LOG"/>
27 <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"/>
28 <uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION"/>
29 <uses-feature android:glEsVersion="0x00000000" android:required="true"/>
30 <application android:theme="@android:style/Theme.Holo.Light.DarkActionBar" android:label="@string/app_name" android:icon="@mipmap/ic_launcher" and
31 <activity android:label="@string/app_name" android:name="com.android.insecurebankv2.LoginActivity">
32 <intent-filter>
33 <action android:name="android.intent.action.MAIN"/>
34 <category android:name="android.intent.category.LAUNCHER"/>
35 </intent-filter>
36 </activity>
37 <activity android:label="@string/title_activity_file_prof" android:name="com.android.insecurebankv2.FilePrefActivity" android:windowSoftInputM
38 <activity android:label="@string/title_activity_do_login" android:name="com.android.insecurebankv2.DoLogin"/>
39 <activity android:label="@string/title_activity_post_login" android:name="com.android.insecurebankv2.PostLogin" android:exported="true"/>
40 <activity android:label="@string/title_activity_wrong_login" android:name="com.android.insecurebankv2.WrongLogin"/>
41 <activity android:label="@string/title_activity_do_transfer" android:name="com.android.insecurebankv2.DoTransfer" android:exported="true"/>
42 <activity android:label="@string/title_activity_view_statement" android:name="com.android.insecurebankv2.ViewStatement" android:exported="true"/>
43 <provider android:name="com.android.insecurebankv2.TrackerContentProvider" android:exported="true" android:authorities="com.android.insecure
44 <receiver android:name="com.android.insecurebankv2.BroadcastReceiver" android:exported="true">
45 <intent-filter>
46 <action android:name="theBroadcast"/>
47 </intent-filter>
48 </receiver>
49 <activity android:label="@string/title_activity_change_password" android:name="com.android.insecurebankv2.ChangePassword" android:exported="tr
50 <activity android:theme="@android:style/Theme.Translucent" android:name="com.google.android.gms.ads.AdActivity" android:configChanges="smalles
51 <activity android:theme="@style/Theme.L10Theme" android:name="com.google.android.gms.ads.purchase.InAppPurchaseActivity"/>
52 <meta-data android:name="com.google.android.gms.version" android:value="@integer/google_play_services_version"/>
53 <meta-data android:name="com.google.android.gms.wallet.api.enabled" android:value="true"/>
54 <receiver android:name="com.google.android.gms.wallet.EnableWalletOptimizationReceiver" android:exported="false">
55 <intent-filter>
56 <action android:name="com.google.android.gms.wallet.ENABLE_WALLET_OPTIMIZATION"/>
57 </intent-filter>
58 </receiver>
59 </application>
60 </manifest>
```

* [-a theBroadcast] = nama actionnya

```

package com.android.insecurebankv2;

import android.content.BroadcastReceiver;
import android.content.Context;
import android.content.Intent;
import android.content.SharedPreferences;
import android.telephony.SmsManager;
import android.util.Base64;

/* Loaded from: classes.dex */
22 public class MyBroadcastReceiver extends BroadcastReceiver {
    public static final String MY_PREFS = "mysharedPreferences";
    String usernameBase64ByteString;

    @Override // android.content.BroadcastReceiver
    23 public void onReceive(Context context, Intent intent) {
    24     String phn = intent.getStringExtra("phonenumber");
    25     String newpass = intent.getStringExtra("newpass");
    26     if (phn != null) {
    27         try {
    28             SharedPreferences settings = context.getSharedPreferences("mysharedPreferences", 1);
    29             String username = settings.getString("encryptedusername", null);
    30             byte[] usernameBase64Byte = Base64.decode(username, 0);
    31             this.usernameBase64ByteString = new String(usernameBase64Byte, "UTF-8");
    32             String password = settings.getString("superSecurePassword", null);
    33             CryptoClass crypt = new CryptoClass();
    34             String decryptedPassword = crypt.aesDecryptedString(password);
    35             String textPhoneno = phn.toString();
    36             String textMessage = "Updated Password from: " + decryptedPassword + " to: " + newpass;
    37             SmsManager smsManager = SmsManager.getDefault();
    38             System.out.println("for the changepassword - phonenumber: " + textPhoneno + " password is: " + textMessage);
    39             smsManager.sendTextMessage(textPhoneno, null, textMessage, null, null);
    40             return;
    41         } catch (Exception e) {
    42             e.printStackTrace();
    43             return;
    44         }
    45     }
    46     System.out.println("Phone number is null");
    }
}

```

* [-es phonenumber “num” -es newpass “newpassword”] = ekstra data dari **phn** dan **newpass**

```

C:\Windows\System32\cmd.exe

D:\Android-InsecureBankv2-master>adb shell am broadcast -a theBroadcast --es phonenumber "082122121212" --es newpass "test1234"
Broadcasting: Intent { act=theBroadcast flg=0x400000 (has extras) }
Broadcast completed: result=0

D:\Android-InsecureBankv2-master>S

```

4. Didapati hasil broadcast melalui SMS dari nomor 082122121212 berupa password baru sesuai dengan yang dituliskan pada terminal cmd



Result

- Login Bypass to Change Password Page using exported activity
- Login Bypass to Main Page using exported activity

	<ul style="list-style-type: none"> • Login Bypass to Transfer Page using exported activity • Exploit Broadcast Message using exported receiver
Recommendation	<p>Saya merekomendasikan pada bagian PostLogin, DoTransfer, ViewStatement, ChangePassword, dan BroadcastReceiver menggunakan <i>permission tag</i> (android:permission), sehingga tidak sembarang aplikasi dapat mengaksesnya dan mengatur <i>user permission</i> agar user dengan akses tertentu saja yang dapat membukanya.</p>

Report 2	
Issue	Sensitive data is written to application logs (MSTG-STORAGE-3)
Executive Summary	Kami menemukan bahwa aplikasi tersebut menaruh data sensitif seperti username dan password yang bisa dilihat secara jelas pada log emulatorenya.
POC [Static Analysis]	<p>Di sini kami melihat code aplikasi tersebut menggunakan JADX dan mulai menganalisis class DoLogin dan ChangePassword</p>  <p>Setelah dianalisis, terdapat beberapa code yang membuat aplikasi tersebut print data sensitif user tanpa terencrypt pada log emulator.</p>
POC [Dynamic Analysis]	<ol style="list-style-type: none"> 1. Untuk melihat log emulator, kita perlu connect emulatorenya terlebih dahulu dan cek apakah device tersebut sudah terconnect atau belum menggunakan command “adb devices” dan apabila sudah muncul nama devicesnya, kita hanya perlu menggunakan command “adb logcat” untuk melihat log

```

min: 499.37ms max: 661.02ms count: 2
PS C:\Users\malik> adb devices
List of devices attached
emulator-5554    device

PS C:\Users\malik> adb logcat

```

2. Setelah itu kita hanya perlu membuka aplikasinya dan login menggunakan creds yang sudah disediakan pada github.



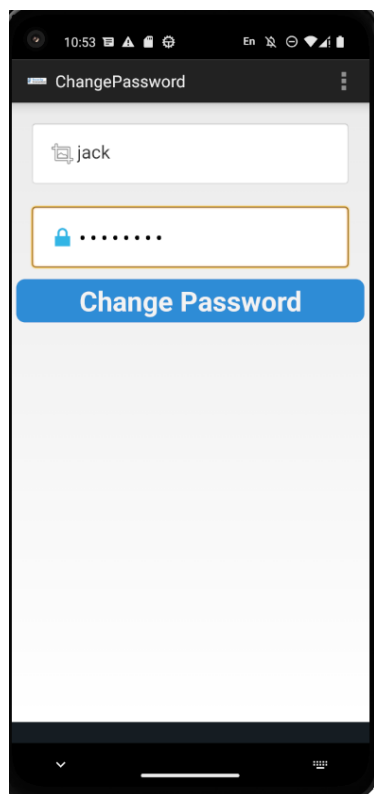
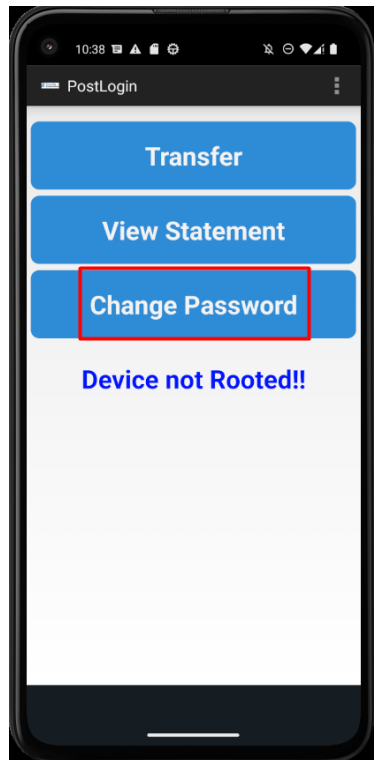
dan apabila dilihat pada lognya, terdapat informasi mengenai akun dan password akun yang saya masukkan sebelumnya.

```

chTime (0) < mLastActivityLaunchTime (30571386)
06-18 22:20:01.578 410 449 D goldfish-address-space: claimShared
sk to claim region [0x3ef29d000 0x3ef8c9000]
06-18 22:20:01.615 9035 9077 D TrafficStats: tagSocket(89) with st
Tag=0xffffffff, statsUid=-1
06-18 22:20:01.646 9035 9077 I TrafficStats: untagSocket(89)
06-18 22:20:01.649 9035 9077 D Successful Login:: , account=jack:J
0123$
06-18 22:20:01.687 9035 9035 W OnBackInvokedCallback: OnBackI
kedCallback is not enabled for the application.
06-18 22:20:01.687 9035 9035 W OnBackInvokedCallback: Set 'android
ableOnBackInvokedCallback="true"' in the application manifest.
06-18 22:20:01.696 459 482 W TransactionTracing: Could not find
er handle 0x7acecd867870
06-18 22:20:01.702 611 1093 I ActivityTaskManager: START u0 {cmp=
.android.insecurebankv2/.PostLogin (has extras)} from uid 10161
06-18 22:20:01.704 611 1093 W ActivityTaskManager: startActivity

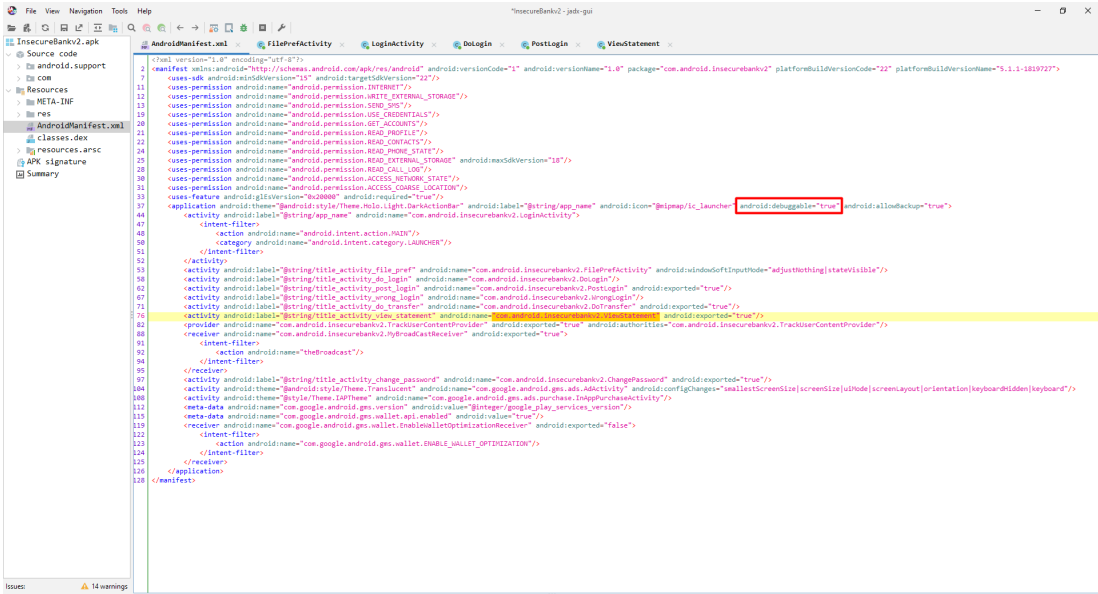
```


3. Lalu kami mencoba untuk mengganti passwordnya




Dan ternyata pada lognya juga terdapat informasi mengenai phone number yang digunakan untuk mengirim sms, password lama akun tersebut, beserta password barunya

	<pre>isplayId=0, eventId=172981790 } 06-18 22:58:40.734 9512 9540 D TrafficStats: tagSocket(96) with statsTag=0 xxxxxxxxxx, statsUid=-1 06-18 22:58:40.801 9512 9540 I TrafficStats: untagSocket(96) 06-18 22:58:40.805 9512 9512 D CompatibilityChangeReporter: Compat change id reported: 147798919; UID 10161; state: DISABLED 06-18 22:58:40.826 9512 9512 I System.out: phonno:+15551234567 06-18 22:58:40.845 611 1820 D CoreBackPreview: Window{ddd22235 u0 Toast}: Setting back callback OnBackInvokedCallbackInfo{mCallback=android.window.IOn BackInvokedCallback\$Stub\$Proxy@1964358, mPriority=0} 06-18 22:58:40.853 9512 9512 I System.out: For the changepassword - phonen umber: +15551234567 password is: Updated Password from: Jack@123\$ to: New@12 3\$ 06-18 22:58:40.867 1033 1051 D SmsNumberUtils: enter filterDestAddr. destA ddr="[GGHjlcXNrWq9L-5GGyTJfxCapeA]" 06-18 22:58:40.869 1033 1051 D SmsNumberUtils: destAddr is not formatted. 06-18 22:58:40.869 1033 1051 D SmsNumberUtils: leave filterDestAddr, new d estAddr="[GGHjlcXNrWq9L-5GGyTJfxCapeA]" 06-18 22:58:40.901 410 457 D goldfish-address-space: claimShared: Ask to claim region [0x3ebb7d000 0x3ebc22000] 06-18 22:58:40.907 410 457 D goldfish-address-space: claimShared: Ask to</pre>
Result	User data leak
Recommendation	Seharusnya developer app harus menghilangkan log class ketika ingin merilis appnya.

Report 3	
Issue	Testing Whether the App is Debuggable (MSTG-CODE-2)
Executive Summary	Kami menemukan bahwa aplikasi tersebut dapat di-debug. Ini dikarenakan developer membiarkan attribute android:debuggable di setel menjadi true di dalam file “AndroidManifest.xml”
POC [Static Analysis]	<p>Saya melihat code aplikasi tersebut dengan menggunakan JADX GUI. Disini saya menemukan “android:debuggable=true” di file “AndroidManifest.xml”.</p> 
POC [Dynamic Analysis]	<p>Aplikasi ditandai sebagai dapat di-debug maka penyerang dapat mengakses data aplikasi dengan mengasumsikan hak istimewa aplikasi tersebut. Misalnya, saya dapat memulai shell di emulator saya dan kemudian beralih ke pengguna non-root. Saya kemudian dapat menggunakan run-as untuk melihat konten direktori paket yang biasanya saya tidak memiliki izin. Setelah itu, Saya menggunakan command “adb exec-out run-as com.android.insecurebankv2 cat databases/mydb > mydb-copy”</p> <pre>emulator: \$ ls acct cache debug_ramdisk linkerconfig odm_dkrm sdcard system_dkrm adb_keys config dev lost+found oem second_stage_resources system_ext apex d etc metadata postinstall storage vendor bin data init mnt proc sys vendor_dkrm bugreports data_mirror init.environ.rc odm product system emulator: \$ run-as com.android.insecurebankv2 emulator:/data/user/0/com.android.insecurebankv2 \$ ls app_textures app_wbview cache code cache databases shared_prefs emulator:/data/user/0/com.android.insecurebankv2 \$ cd databases/ emulator:/data/user/0/com.android.insecurebankv2/databases \$ ls mydb mydb-journal emulator:/data/user/0/com.android.insecurebankv2/databases \$ exit emulator: \$ exit emulator: \$ adb exec-out run-as com.android.insecurebankv2 cat databases/mydb > mydb-copy</pre>
Result	Kita mendapatkan file database “mydb-copy”

	 <p>SQLite format 3.00 DB 1000</p> <p>CREATE TABLE sqlite_sequence (seq INTEGER PRIMARY KEY AUTOINCREMENT, name TEXT NOT NULL);</p> <p>DB 1000</p>
Recommendation	Setelah dibuka dengan notepad didapatkan seperti gambar diatas.
	Untuk mencegah app data backup, setel (android:debuggable=”true”) true menjadi false. Jika attribute ini tidak tersedia, pengaturan “debuggable” diaktifkan secara default, dan backup tersebut akan dinonaktifkan

Report 4	
Issue	The app's login system can be try repeatedly after a lot of failed authentication attempts (MSTG-STORAGE-15)
Executive Summary	Kami menemukan bahwa sistem login yang ada pada aplikasi memungkinkan user untuk melakukan login tanpa batas. Hal ini memungkinkan user untuk melakukan serangan <i>brute-force</i> (berulang kali sampai berhasil) sampai dapat mengambil alih akun.
POC [Static Analysis]	
POC [Dynamic Analysis]	 <p>Disini saya mencoba memasukkan credensial secara asal, yakni berupa “a” untuk username dan passwordnya. Setelah saya ulang sekitar 15 kali, saya masih bisa mencoba loginnya tanpa ada warning seperti tidak bisa mengisi selama 1 menit, 5 menit, yang akan terus berakumulatif.</p>
Result	Not banned or blocked or warned after multiple logins.
Recommendation	Menerapkan sistem untuk memperlambat login atau mengunci akun untuk meningkatkan security dari aplikasi.

