Mobile Application Security Testing Report

Intern name: Cinchana S
Date: 23-01-2026

Introduction

The objective of this task was to analyze an Android application, identify common security issues, and document the findings in a clear and practical manner. The target application used for this assessment was DIVA (Damn Insecure and Vulnerable App), a deliberately vulnerable Android application commonly used for learning mobile security testing.

Scope of Testing

The assessment focused on identifying common mobile application security weaknesses through APK analysis and security review. The scope included static analysis, manual code review, authentication logic assessment, and IPC exposure analysis.

Tools

The following tools were used during the assessment:

- **MobSF (Mobile Security Framework)** – Static analysis of the APK
- **JADX** – Decompilation and manual source code review
- **Frida** – Dynamic testing approach
- **Drozer** – IPC testing

Application Overview

- Application Name: DIVA (Damn Insecure and Vulnerable App)
- APK File: diva-beta.apk
- Package Name: jakhar.aseem.diva

The APK was successfully decompiled, allowing analysis of the application manifest, source code, and resources.

Static Analysis (MobSF)

Static analysis was performed by uploading the APK to MobSF. MobSF automatically scanned the application and highlighted multiple security concerns related to insecure coding practices.

Key Findings from MobSF

- Insecure data storage mechanisms
- Hardcoded secrets present in the application code
- Exported application components increasing IPC attack surface

These results were further validated through manual analysis using JADX.

Manual Code Review (JADX)

Using JADX, the application source code and AndroidManifest.xml file were reviewed to better understand the identified issues.

Exported Component Identified

<provider
android:name="jakhar.aseem.diva.NotesProvider"
android:enabled="true"
android:exported="true"
android:authorities="jakhar.aseem.diva.provider.notesprovider" />

The content provider is exported without any visible permission checks, which could allow other applications to access sensitive data.

Vulnerability Findings

Insecure Data Storage

Description: The application stores sensitive information locally without proper encryption or protection. Static analysis indicates the use of insecure storage mechanisms, making data accessible through device compromise or backup extraction.

Impact

- Exposure of sensitive application data

Severity: High

Hardcoded Authentication Logic

Affected Component

- **Activity:** *HardcodeActivity*

Description The authentication mechanism relies on a hardcoded secret key (**vendorsecretkey**) within the client-side code. An attacker can reverse engineer the APK and bypass authentication.

Impact

- Authentication bypass
- Unauthorized access to application functionality

Severity: High

Exported Content Provider (IPC Exposure)

Affected Component

- **Content Provider:** *jakhar.aseem.diva.NotesProvider*

Description The exported content provider can be accessed by other applications without proper access control, leading to potential unauthorized data access.

Impact

- Information disclosure
- Abuse of inter-process communication

Severity: High

Dynamic Testing (Frida)

Dynamic testing techniques were applied to analyze authentication-related logic within the application. The hardcoded authentication check in *HardcodeActivity* was identified as a candidate for runtime manipulation using Frida. This demonstrates how client-side security controls can be bypassed through function hooking.

IPC Testing (Drozer)

IPC testing was conducted by reviewing exported components and preparing Drozer commands to assess the application's inter-process communication exposure. These commands are commonly used to enumerate attack surfaces and identify unauthorized access to exported components.

The following Drozer commands were prepared as part of the assessment:

- Identify attack surface:

  run app.package.attacksurface jakhar.aseem.diva

- List exported activities:

  run activity.info -a jakhar.aseem.diva

- Query exported content provider:

  run provider.content query
  content://jakhar.aseem.diva.provider.notesprovider/notes --vertical

- Scan content providers for injection:

  run scanner.provider.injection -a jakhar.aseem.diva

Risk Assessment Summary

Test ID	Vulnerability	Severity	Impact
001	Insecure Data Storage	High	Sensitive data exposure
002	Hardcoded Authentication Logic	High	Authentication bypass
003	Exported Content Provider	High	Unauthorized data access

Recommendations

- Avoid storing sensitive data locally without encryption
- Remove hardcoded secrets from client-side code
- Implement server-side authentication controls
- Set *android:exported="false"* for unnecessary components
- Apply permission checks to content providers

Conclusion

The mobile application assessment identified several high-risk security issues related to insecure storage, client-side authentication, and IPC exposure. The findings demonstrate realistic attack paths that could be exploited by an attacker if appropriate controls are not implemented. Addressing these issues would significantly improve the security posture of the application.

Summary

This assessment revealed that the application relies heavily on client-side security controls and exposes internal components unnecessarily. These weaknesses make the application vulnerable to reverse engineering, authentication bypass, and unauthorized data access. Proper secure coding practices and IPC restrictions are required to mitigate these risks.