Task 5 Report: Mobile Application Testing Lab

Target: InsecureBankv2.apk
Tool: Mobile Security Framework (MobSF)
Technique: Static Application Security Testing (SAST)

1. Findings Log

Test ID	Vulnerability	Severity	Target App
016	Excessive Application Permissions	Medium	`InsecureBankv2.apk`
017	Insecure Data Storage	High	`InsecureBankv2.apk`

2. Technical Summary

Static analysis of the target APK using MobSF revealed several security misconfigurations and vulnerabilities. The application requests excessive permissions beyond its functional requirements and implements insecure data storage practices, potentially exposing sensitive user information on compromised devices.

3. Evidence

3.1. MobSF Dashboard and APK Analysis

The Mobile Security Framework dashboard shows the successful upload and analysis of the target mobile application, confirming the assessment environment was properly configured.
Evidence: `11_MobSF_Dashboard.png`

3.2. Security Scoring and Vulnerability Assessment

MobSF's automated analysis provided a comprehensive security score and categorized vulnerabilities by severity, highlighting multiple medium and high-risk findings that require remediation.
Evidence: 12_MobSF_Security_Score.png

3.3. Excessive Permission Request Analysis

The application requests unnecessary permissions including full network access and read/write external storage capabilities, exceeding its functional requirements and expanding the attack surface.
Evidence: 13_MobSF_Application_Permissions.png

4. Mobile Testing Summary

MobSF analysis revealed excessive permission requests and insecure data storage practices. The application demands unnecessary device access permissions while storing sensitive data insecurely. These vulnerabilities could allow threat actors to extract confidential information from compromised devices or gain unauthorized access to user data through permission abuse attacks.

5. Remediation Recommendations

1. Permission Minimization: Revise the application's permission request list to follow the principle of least privilege, removing unnecessary permissions.
2. Secure Storage Implementation: Utilize Android Keystore for sensitive data encryption and secure storage practices.
3. Permission Rationale: Provide clear explanations for all required permissions within the application description.
4. Runtime Permission Requests: Implement runtime permission requests for Android 6.0+ to enhance user awareness and control.
5. Security Review: Conduct regular security assessments and code reviews to identify permission abuse vulnerabilities.