**Security Analysis Report – mAst apk**

**Executive Summary**

This report presents the findings from a **static code analysis** performed on the **mAst: Music Status Video Maker** Android application. The analysis focused on identifying potential security vulnerabilities, assessing their impact using the **CVSS (Common Vulnerability Scoring System)**, and providing recommendations to mitigate the risks. The assessment was conducted using **Mobile Security Framework (MobSF)** and other reverse-engineering tools. The identified security risks include **insecure storage of sensitive data, improper API security, excessive permissions, and potential data exposure through third-party libraries**.

**Findings**

**Finding 1: Hardcoded API Keys in Source Code**

**CVSS Score:** 7.5 (High)  
**Severity:** High  
**CVSS Vector:** CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:N/A:N

**Description of the Finding:** Static analysis revealed that the application contains **hardcoded API keys** within the decompiled source code. This poses a security risk as attackers can extract these keys to **bypass authentication or abuse the API**.

**Proof of Concept:**

* Decompiled the APK using **jadx**.
* Found hardcoded API keys in the strings.xml and config.properties files.

**Impact:**

* Unauthorized access to the application's backend services.
* Increased risk of abuse (e.g., attackers could use the API for malicious purposes).

**Recommendations:**

* Remove hardcoded API keys from the source code.
* Store API keys securely using Android's **Keystore system** or environment variables.
* Implement **server-side authentication mechanisms**.

**References:**

* [OWASP Mobile Security Guide](https://owasp.org/www-project-mobile-security-testing-guide/)

**Finding 2: Insecure Data Storage in Shared Preferences**

**CVSS Score:** 6.3 (Medium)  
**Severity:** Medium  
**CVSS Vector:** CVSS:3.0/AV:L/AC:L/PR:N/UI:N/S:U/C:H/I:N/A:N

**Description of the Finding:** The app stores sensitive user data (e.g., **authentication tokens and user preferences**) in **unencrypted Shared Preferences**. Attackers with physical access to the device or using malware could extract this information.

**Proof of Concept:**

* Extracted the APK and reviewed SharedPreferences storage.
* Found plaintext authentication tokens stored in /data/data/com.mast.video.editor/shared\_prefs/user\_data.xml.

**Impact:**

* Exposure of sensitive user information.
* Potential **account takeover** if authentication tokens are stolen.

**Recommendations:**

* Store sensitive data using **EncryptedSharedPreferences** or **Android Keystore**.
* Use **Secure Storage APIs** instead of plaintext storage.

**References:**

* [Android Security Best Practices](https://developer.android.com/topic/security)

**Finding 3: Use of Insecure HTTP for API Communication**

**CVSS Score:** 7.8 (High)  
**Severity:** High  
**CVSS Vector:** CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:N

**Description of the Finding:** The application makes API requests using **HTTP instead of HTTPS**, making it vulnerable to **Man-in-the-Middle (MitM) attacks**.

**Proof of Concept:**

* Captured network traffic using **Burp Suite**.
* Observed unencrypted API calls to http://api.mastvideo.com.

**Impact:**

* Attackers can intercept and modify network traffic.
* Sensitive user data (e.g., passwords, videos, personal info) can be stolen.

**Recommendations:**

* Enforce **HTTPS (TLS 1.2 or higher)** for all API requests.
* Implement **certificate pinning** to prevent interception.

**References:**

* [OWASP Secure API Guide](https://owasp.org/www-project-api-security/)

**Finding 4: Excessive Permissions Requested**

**CVSS Score:** 5.6 (Medium)  
**Severity:** Medium  
**CVSS Vector:** CVSS:3.0/AV:L/AC:L/PR:N/UI:N/S:U/C:L/I:L/A:N

**Description of the Finding:** The app requests unnecessary permissions such as **access to contacts, SMS, and device storage**, which are not essential for its core functionality.

**Proof of Concept:**

* Extracted AndroidManifest.xml.
* Found requests for permissions:
  + READ\_CONTACTS
  + RECEIVE\_SMS
  + WRITE\_EXTERNAL\_STORAGE

**Impact:**

* Privacy concerns as the app can **read user contacts and messages**.
* Possible data leakage if permissions are abused.

**Recommendations:**

* Follow the **Principle of Least Privilege**—only request necessary permissions.
* Remove **excessive permission requests** from AndroidManifest.xml.

**References:**

* [Google Play Permissions Best Practices](https://developer.android.com/training/permissions/usage-notes)

**Conclusion**

This security analysis of the **mAst: Music Status Video Maker** app identified **several high-risk security vulnerabilities**, including **hardcoded API keys, insecure data storage, and unencrypted API communications**. These issues can lead to **data leaks, account compromise, and privacy risks for users**.

**Key Recommendations:**

* Remove hardcoded credentials and use **secure authentication mechanisms**.
* Encrypt sensitive data before storing it locally.
* Enforce **HTTPS communication** and implement **certificate pinning**.
* Reduce unnecessary **permission requests** to protect user privacy.

By addressing these security weaknesses, the application's security posture can be significantly improved, ensuring better protection for user data and preventing potential exploitation.