# AndroidManifest.xml File Security

**D** by Durgesh Thakare

# **AndroidManifest.xml**

File is an important component of an Android application as it contains essential information and configurations about the app. It serves as a roadmap for the Android system to understand and interact with the app. Therefore, it is crucial to ensure the security and integrity of the AndroidManifest.xml file to prevent unauthorized access or tampering.

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"</pre>
    package="com.deneme2.uygulama1">
    <application</pre>
        android:allowBackup="true"
        android:icon="@mipmap/ic launcher"
        android:label="Uygulama1"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">
        <activity android:name=".MainActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <activity android:name=".CustomList"></activity>
   </application>
</manifest>
```



# WHAT INSIDE ANDROIDMANIFEST FILE

### 1 MinSDK Version:

MinSDK specifies the minimum API level on which the application can run. It is essential to ensure that your app functions correctly on different Android versions while keeping security features up-to-date.

### **2** Permissions:

Define what data and hardware components the app needs to access.

Permissions should be explicitly declared for:

#### Camera

#### Contacts

Internet access

Read/write external storage

Package management

Bluetooth, etc.

Make sure that you request only the permissions necessary for your app's functionality to reduce the attack surface.

## **3** Activities:

Activities represent UI elements or screens in the application. Some activities, particularly those dealing with sensitive data, need extra protection. Consider using Intent filters to restrict access to screens containing:

**Account details** 

Money transfer functions

Hidden screens

## **4** Content Providers:

Content providers serve data from your app to other applications. Be cautious about the data you expose and ensure it's adequately protected, especially if it's sensitive information.

# Structure of AndroidManifest.xml

The basic structure of the Android Manifest.xml file includes the following main elements:

- <manifest> tag
- <application> tag
- <activity>, <service>, <receiver>, and or components
- <intent-filter> for defining intent filters
- permission> for specifying permissions

# Security Issues in AndroidManifest.xml

### **Backup Mode** If backup mode is enabled, an attacker can take a backup of your application, **Debug Mode** 2 potentially exposing sensitive data. It should be set to false to mitigate this Debug mode can make your risk. application vulnerable. When set to true, an attacker can debug the APK and potentially access sensitive 3 **Permissions** information. Debug mode must be set to false to protect against this critical Dangerously sensitive permissions vulnerability. (e.g., read and write external storage) must be carefully managed and minimized. Request only the **exported Attribute** permissions your app truly requires. The exported attribute should be set to false for components that should not be accessible by other apps. This restricts external access to your app's components and data, enhancing security. 5 **Intent Filters** If the app is public, it can be accessed Examine <intent-filter> elements by other apps (set to true). If it's private, associated with activities and services. it cannot be accessed by other apps Determine if any sensitive data is (set to false). passed through intents without proper validation or protection. Unauthorized access to sensitive data Misconfigured URIs is a common security risk. Look for any URIs or deep links specified in the manifest. Misconfigured URIs could lead to

# security issues, such as open redirects or unintended access to resources. 7 Data Leakage

9

10

Assess broadcast receivers to determine if they are overly permissive or handle broadcasts that could lead to vulnerabilities, such as data leakage or code execution.

**Broadcast Receivers** 

# Components Handling Sensitive Data

Identify activities, services, and providers that handle sensitive data, such as user credentials or financial information. Ensure they have the necessary security controls in place.

Examine the manifest for any unintentional data leakage, such as the inclusion of sensitive data in exported components or the use of insecure intent filters.

#### Code Obfuscation

While not directly visible in the manifest, it's important to inquire whether code obfuscation is used to protect the app's source code and binaries from reverse engineering.