rules.sonarsource.com

Kotlin static code analysis: Authorizing non-authenticated users to use keys in the Android KeyStore is security-sensitive

3-4 minutes

Android KeyStore is a secure container for storing key materials, in particular it prevents key materials extraction, i.e. when the application process is compromised, the attacker cannot extract keys but may still be able to use them. It’s possible to enable an Android security feature, user authentication, to restrict usage of keys to only authenticated users. The lock screen has to be unlocked with defined credentials (pattern/PIN/password, biometric).

Ask Yourself Whether

- The application requires prohibiting the use of keys in case of compromise of the application process.
- The key material is used in the context of a highly sensitive application like a e-banking mobile app.

There is a risk if you answered yes to any of those questions.

Recommended Secure Coding Practices

It’s recommended to enable user authentication (by setting `setUserAuthenticationRequired` to `true` during key generation) to use keys for a limited duration of time (by setting appropriate values to `setUserAuthenticationValidityDurationSeconds`), after which the user must re-authenticate.

Noncompliant Code Example

Any users can use the key:

```
val keyGenerator: KeyGenerator =
KeyGenerator.getInstance(KeyProperties.KEY_ALGORITHM_AES,
"AndroidKeyStore")

var builder: KeyGenParameterSpec =
KeyGenParameterSpec.Builder("test_secret_key",
KeyProperties.PURPOSE_ENCRYPT or
KeyProperties.PURPOSE_DECRYPT) // Noncompliant
  .setBlockModes(KeyProperties.BLOCK_MODE_GCM)

.setEncryptionPaddings(KeyProperties.ENCRYPTION_PADDING_NONE)
  .build()

keyGenerator.init(builder)
```

Compliant Solution

The use of the key is limited to authenticated users (for a duration of time defined to 60 seconds):

```
val keyGenerator: KeyGenerator =
KeyGenerator.getInstance(KeyProperties.KEY_ALGORITHM_AES,
"AndroidKeyStore")

var builder: KeyGenParameterSpec =
KeyGenParameterSpec.Builder("test_secret_key",
KeyProperties.PURPOSE_ENCRYPT or
KeyProperties.PURPOSE_DECRYPT)
  .setBlockModes(KeyProperties.BLOCK_MODE_GCM)

.setEncryptionPaddings(KeyProperties.ENCRYPTION_PADDING_NONE)
  .setUserAuthenticationRequired(true) // Compliant
  .setUserAuthenticationParameters (60,
KeyProperties.AUTH_DEVICE_CREDENTIAL)
  .build()

keyGenerator.init(builder)
```

See

- OWASP Top 10 2021 Category A4 - Insecure Design
- developer.android.com - Android keystore system
- developer.android.com - Require user authentication for key use
- Mobile AppSec Verification Standard - Authentication and Session Management Requirements
- OWASP Mobile Top 10 2016 Category M4 - Insecure Authentication
- MITRE, CWE-522 - Insufficiently Protected Credentials