rules.sonarsource.com

Kotlin static code analysis: Encryption algorithms should be used with secure mode and padding scheme

3 minutes

Encryption operation mode and the padding scheme should be chosen appropriately to guarantee data confidentiality, integrity and authenticity:

- For block cipher encryption algorithms (like AES):
- The GCM (Galois Counter Mode) mode which works internally with zero/no padding scheme, is recommended, as it is designed to provide both data authenticity (integrity) and confidentiality. Other similar modes are CCM, CWC, EAX, IAPM and OCB.
- The CBC (Cipher Block Chaining) mode by itself provides only data confidentiality, it's recommended to use it along with Message Authentication Code or similar to achieve data authenticity (integrity) too and thus to prevent padding oracle attacks.
- The ECB (Electronic Codebook) mode doesn't provide serious message confidentiality: under a given key any given plaintext block always gets encrypted to the same ciphertext block. This mode should not be used.
- For RSA encryption algorithm, the recommended padding scheme is OAEP.

Noncompliant Code Example

```
val c1 = Cipher.getInstance("AES") // Noncompliant: by default ECB mode is chosen
val c2 = Cipher.getInstance("AES/ECB/NoPadding") // Noncompliant: ECB doesn't provide serious message confidentiality

val c3 = Cipher.getInstance("RSA/None/NoPadding") // Noncompliant: RSA without OAEP padding scheme is not recommanded
```

Compliant Solution

```
// Recommended for block ciphers
val c1 = Cipher.getInstance("AES/GCM/NoPadding"); // Compliant

// Recommended for RSA
val c3 = Cipher.getInstance("RSA/None/OAEPWITHSHA-256ANDMGF1PADDING") // Compliant
// or the ECB mode can be used for RSA when "None" is not available with the security provider used - in that case, ECB will be treated as "None" for RSA.
val c3 = Cipher.getInstance("RSA/ECB/OAEPWITHSHA-256ANDMGF1PADDING"); // Compliant
```

See

- OWASP Top 10 2021 Category A2 - Cryptographic Failures
- OWASP Top 10 2017 Category A6 - Security Misconfiguration
- Mobile AppSec Verification Standard - Cryptography Requirements
- OWASP Mobile Top 10 2016 Category M5 - Insufficient Cryptography
- MITRE, CWE-327 - Use of a Broken or Risky Cryptographic Algorithm

- CERT, MSC61-J. - Do not use insecure or weak cryptographic algorithms
- SANS Top 25 - Porous Defenses