rules.sonarsource.com

Kotlin static code analysis: Cipher algorithms should be robust

3 minutes

Strong cipher algorithms are cryptographic systems resistant to cryptanalysis, they are not vulnerable to well-known attacks like brute force attacks for example.

A general recommendation is to only use cipher algorithms intensively tested and promoted by the cryptographic community.

More specifically for block cipher, it's not recommended to use algorithm with a block size inferior than 128 bits.

Noncompliant Code Example

```
import javax.crypto.NoSuchPaddingException
import java.security.NoSuchAlgorithmException
import javax.crypto.Cipher

class test {
  fun main(args: Array<String>) {
    try {
      val c1 = Cipher.getInstance("DES") // Noncompliant: DES works with 56-bit keys allow attacks via exhaustive search
      val c7 = Cipher.getInstance("DESede") // Noncompliant: Triple DES is vulnerable to meet-in-the-middle attack
      val c13 = Cipher.getInstance("RC2") // Noncompliant: RC2 is vulnerable to a related-key attack
      val c19 = Cipher.getInstance("RC4") // Noncompliant: vulnerable to several attacks (see https://en.wikipedia.org/wiki/RC4#Security)
      val c25 = Cipher.getInstance("Blowfish") // Noncompliant: Blowfish use a 64-bit block size makes it vulnerable to birthday attacks
      val nc = NullCipher() // Noncompliant: the NullCipher class provides an "identity cipher" one that does not transform or encrypt the plaintext in any way.


    } catch (e: NoSuchAlgorithmException) {
    } catch (e: NoSuchPaddingException) {
    }
  }
}
```

Compliant Solution

```
import javax.crypto.NoSuchPaddingException
import java.security.NoSuchAlgorithmException
import javax.crypto.Cipher

class test {
  fun main(args: Array<String>) {
    try {
      val c31 = Cipher.getInstance("AES/GCM/NoPadding") // Compliant

    } catch (e: NoSuchAlgorithmException) {
    } catch (e: NoSuchPaddingException) {
    }
  }
}
```

See

- OWASP Top 10 2021 Category A2 - Cryptographic Failures
- OWASP Top 10 2017 Category A3 - Sensitive Data Exposure
- 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