rules.sonarsource.com

Swift static code analysis: Cipher algorithms should be robust

2 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

CommonCrypto library:

```
import CommonCrypto

let algorithm = CCAlgorithm(kCCAlgorithmDES) // Noncompliant: 64 bits block size
```

IDZSwiftCommonCrypto library:

```
import IDZSwiftCommonCrypto

let algorithm = .des // Noncompliant: 64 bits block size
```

CryptoSwift

```
import CryptoSwift

let blowfish = try Blowfish(key: key, blockMode: GCM(iv: iv, mode: .combined), padding: .pkcs7) // Noncompliant: 64 bits block size
```

Compliant Solution

Swift Crypto library: prefer using this library which is native and officially supported by Apple

```
import Crypto

let sealedBox = try AES.GCM.seal(input, using: key) // Compliant
```

CommonCrypto library:

```
import CommonCrypto

let algorithm = CCAlgorithm(kCCAlgorithmAES) // Compliant
```

IDZSwiftCommonCrypto library:

```
import IDZSwiftCommonCrypto

let algorithm = .aes // Compliant
```

CryptoSwift

```
import CryptoSwift

let aes = try AES(key: key, iv: iv) // Compliant
```

See

- OWASP Top 10 2021 Category A2 - Cryptographic Failures
- OWASP Top 10 2017 Category A3 - Sensitive Data Exposure
- MITRE, CWE-327 - Use of a Broken or Risky Cryptographic Algorithm
- SANS Top 25 - Porous Defenses