sonar RULES

Products ⌄

- Secrets
- ABAP
- Apex
- C
- C++
- CloudFormation
- COBOL
- C#
- CSS
- Flex
- Go
- HTML
- Java
- **JavaScript**
- Kotlin
- Objective C
- PHP
- PL/I
- PL/SQL
- Python
- RPG
- Ruby
- Scala
- Swift
- Terraform
- Text
- TypeScript
- T-SQL
- VB.NET
- VB6
- XML

JavaScript static code analysis

Unique rules to find Bugs, Vulnerabilities, Security Hotspots, and Code Smells in your JAVASCRIPT code

| All rules 285 | 🔒 Vulnerability 29 | 🐛 Bug 62 | 🛡 Security Hotspot 43 | Code Smell 151 | Quick Fix 41 |

Tags ⌄   Search by name... 🔍

Code Smell

Conditionals should start on new lines

Code Smell

Cognitive Complexity of functions should not be too high

Code Smell

"void" should not be used

Code Smell

Loop counters should not be assigned to from within the loop body

Code Smell

"for" loop increment clauses should modify the loops' counters

Code Smell

Functions should not be empty

Code Smell

Server-side requests should not be vulnerable to forging attacks

🔒 Vulnerability

Non-empty statements should change control flow or have at least one side-effect

🐛 Bug

Regular expressions with the global flag should be used with caution

🐛 Bug

Replacement strings should reference existing regular expression groups

🐛 Bug

Regular expressions should not contain control characters

🐛 Bug

Alternation in regular expressions should not contain empty alternatives

---

Cryptographic keys should be robust

[Analyze your code]

🔒 Vulnerability   ⬆ Critical ❓   🏷 cwe privacy owasp

Most of cryptographic systems require a sufficient key size to be robust against brute-force attacks.

NIST recommendations will be checked for these use-cases:

Digital Signature Generation and **Verification:**

- $p \geq 2048$ AND $q \geq 224$ for DSA (p is key length and q the modulus length)
- $n \geq 2048$ for RSA (n is the key length)

Key Agreement:

- $p \geq 2048$ AND $q \geq 224$ for DH and MQV
- $n \geq 224$ for ECDH and ECMQV (Examples: `secp192r1` is a non-compliant curve ($n < 224$) but `secp224k1` is compliant ($n \geq 224$))

Symmetric keys:

- key length ≥ 128 bits

This rule will not raise issues for ciphers that are considered weak (no matter the key size) like `DES`, `Blowfish`.

Noncompliant Code Example

`crypto` built-in module:

```
var { privateKey, publicKey } = crypto.generateKeyPairSync('
  modulusLength: 1024,  // Noncompliant
  publicKeyEncoding:  { type: 'spki', format: 'pem' },
  privateKeyEncoding: { type: 'pkcs8', format: 'pem' }
}); // Noncompliant: 1024 bits is too short for a RSA key pa

crypto.generateKeyPair('ec', {
  namedCurve: 'secp112r2',
  publicKeyEncoding:  { type: 'spki', format: 'pem' },
  privateKeyEncoding: { type: 'pkcs8', format: 'pem' }
}, callback); // Noncompliant: secp112r2 curve doesn't provi
```

Compliant Solution

`crypto` built-in module:

```
crypto.generateKeyPair('rsa', {
  modulusLength: 2048,  // Compliant
  publicKeyEncoding:  { type: 'spki', format: 'pem' },
  privateKeyEncoding: { type: 'pkcs8', format: 'pem' }
}, callback); // Compliant

crypto.generateKeyPair('ec', {
  namedCurve: 'secp224k1',
  publicKeyEncoding:  { type: 'spki', format: 'pem' },
```

```
        privateKeyEncoding: { type: 'pkcs8', format: 'pem' }
    }, callback); // compliant
```

Bug

Mocha timeout should be disabled by setting it to "0".

Bug

Unicode Grapheme Clusters should be avoided inside regex character classes

Bug

Assertions should not be given twice the same argument

Bug

Alternatives in regular expressions

See

- OWASP Top 10 2021 Category A2 - Cryptographic Failures
- OWASP Top 10 2017 Category A3 - Sensitive Data Exposure
- OWASP Top 10 2017 Category A6 - Security Misconfiguration
- Mobile AppSec Verification Standard - Cryptography Requirements
- OWASP Mobile Top 10 2016 Category M5 - Insufficient Cryptography
- NIST 800-131A - Recommendation for Transitioning the Use of Cryptographic Algorithms and Key Lengths
- MITRE, CWE-326 - Inadequate Encryption Strength

Available In:

sonarlint | sonarcloud | sonarqube