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...

Loops with at most one iteration should be refactored
Bug

Variables should not be self-assigned
Bug

Function argument names should be unique
Bug

Property names should not be duplicated within a class or object literal
Bug

Bitwise operators should not be used in boolean contexts
Bug

Constructing arguments of system commands from user input is security-sensitive
Security Hotspot

Allowing requests with excessive content length is security-sensitive
Security Hotspot

Statically serving hidden files is security-sensitive
Security Hotspot

Using intrusive permissions is security-sensitive
Security Hotspot

Disabling auto-escaping in template engines is security-sensitive
Security Hotspot

Using shell interpreter when executing OS commands is security-sensitive
Security Hotspot

Setting loose POSIX file permissions is security-sensitive

Tests should not execute any code after "done()" is called

Analyze your code

Code Smell | Critical | tests unpredictable mocha

The done callback is used to inform Mocha when an asynchronous test ends. Exceptions thrown after done (with or without parameters) is called are not handled in a consistent manner. Sometimes they will be correctly handled, but they might as well be assigned to a different test, no test at all, or even be completely ignored. Even when it works as expected this will be a source of confusion for other developers. Thus no code should be executed after done is called.

This rule raises an issue when some code is executed after a call to done.

Noncompliant Code Example

```
const expect = require("chai").expect;
const fs = require("fs");

describe("Code is executed after Done", function() {
    it("Has asserts after done()", function(done) {
        try {
            expect(1).toEqual(2);
        } catch (err) {
            done();
            // This assertion will be ignored and the test w
            expect(err).to.be.an.instanceof(RangeError);  //
        }
    });

    it("Throws an error some time after done()", function(do
        fs.readFile("/etc/bashrc", 'utf8', function(err, dat
            done();
            setTimeout(() => {  // Noncompliant
                // This assertion error will not be assigned
                // Developers will have to guess which test
                expect(data).to.match(/some expected string/
            }, 3000);
        });
    });

    it("Has code after done(err)", function(done) {
        try {
            throw Error("An error");
        } catch (err) {
            done(err);
        }
        fs.readFile("/etc/bashrc", 'utf8', function(err, dat
            // This assertion error will be assigned to "Oth
            expect(data).to.match(/some expected string/);
        });
    });

    it("Other test", function(done) {
        done()
```

is security-sensitive

Security Hotspot

Formatting SQL queries is security-sensitive

Security Hotspot

Comma operator should not be used

Code Smell

Regular expressions should not contain empty groups

Code Smell

Regular expressions should not contain multiple spaces

```
    });
});
```

Compliant Solution

```
const expect = require("chai").expect;
const fs = require("fs");

describe("Code is executed after Done", function() {
    it("Has asserts after done()", function(done) {
        try {
            expect(1).toEqual(2);
        } catch (err) {
            expect(err).to.be.an.instanceof(RangeError);
            done();
        }
    });

    it("Throws an error some time after done()", function(do
        fs.readFile("/etc/bashrc", 'utf8', function(err, dat
            setTimeout(() => {
                expect(data).to.match(/some expected string/
                done();
            }, 3000);
        });
    });

    it("Has code after done(err)", function(done) {
        try {
            throw Error("An error");
        } catch (err) {
            return done(err);
        }
        fs.readFile("/etc/bashrc", 'utf8', function(err, dat
            // This assertion error will be assigned to "Oth
            expect(data).to.match(/some expected string/);
            done();
        });
    });

    it("Other test", function(done) {
        done()
    });
});
```

Available In:

sonarlint | sonarcloud | sonarqube