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...

Jump statements should not occur in "finally" blocks
Bug

Using slow regular expressions is security-sensitive
Security Hotspot

Using publicly writable directories is security-sensitive
Security Hotspot

Using clear-text protocols is security-sensitive
Security Hotspot

Expanding archive files without controlling resource consumption is security-sensitive
Security Hotspot

Using weak hashing algorithms is security-sensitive
Security Hotspot

Disabling CSRF protections is security-sensitive
Security Hotspot

Using pseudorandom number generators (PRNGs) is security-sensitive
Security Hotspot

Dynamically executing code is security-sensitive
Security Hotspot

Equality operators should not be used in "for" loop termination conditions
Code Smell

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

"default" clauses should be last

A new session should be created during user authentication

Analyze your code

Vulnerability | Critical | cwe owasp

Session fixation attacks occur when an attacker can force a legitimate user to use a session ID that he knows. To avoid fixation attacks, it’s a good practice to generate a new session each time a user authenticates and delete/invalidate the existing session (the one possibly known by the attacker).

Noncompliant Code Example

For Passport.js:

```
app.post('/login',
  passport.authenticate('local', { failureRedirect: '/login'
  function(req, res) {
    // Sensitive - no session.regenerate after login
    res.redirect('/');
  });
```

Compliant Solution

For Passport.js:

```
app.post('/login',
  passport.authenticate('local', { failureRedirect: '/login'
  function(req, res) {
    let prevSession = req.session;
    req.session.regenerate((err) => {  // Compliant
      Object.assign(req.session, prevSession);
      res.redirect('/');
    });
  });
```

See

- OWASP Top 10 2021 Category A7 - Identification and Authentication Failures
- OWASP Top 10 2017 Category A2 - Broken Authentication
- OWASP Sesssion Fixation
- MITRE, CWE-384 - Session Fixation

Available In:

sonarlint | sonarcloud | sonarqube

"default" clauses should be last

Code Smell

"await" should only be used with promises

Code Smell

A conditionally executed single line should be denoted by indentation

Code Smell

Conditionals should start on new lines

Code Smell

Cognitive Complexity of functions should not be too high