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...

Object literal shorthand syntax should be used
Code Smell

Strings and non-strings should not be added
Code Smell

Object literal syntax should be used
Code Smell

"undefined" should not be assigned
Code Smell

Trailing commas should not be used
Code Smell

Array constructors should not be used
Code Smell

Quotes for string literals should be used consistently
Code Smell

Statements should end with semicolons
Code Smell

Comments should not be located at the end of lines of code
Code Smell

Loops should not contain more than a single "break" or "continue" statement
Code Smell

Variable, property and parameter names should comply with a naming convention
Code Smell

Lines should not end with trailing whitespaces
Code Smell

Disclosing fingerprints from web application technologies is security-sensitive

Analyze your code

Security Hotspot   Minor   cwe owasp express.js

Disclosing technology fingerprints allows an attacker to gather information about the technologies used to develop the web application and to perform relevant security assessments more quickly (like the identification of known vulnerable components).

Ask Yourself Whether

- The `x-powered-by` HTTP header or similar is used by the application.
- Technologies used by the application are confidential and should not be easily guessed.

There is a risk if you answered yes to any of these questions.

Recommended Secure Coding Practices

It's recommended to not disclose technologies used on a website, with `x-powered-by` HTTP header for example.

In addition, it's better to completely disable this HTTP header rather than setting it a random value.

Sensitive Code Example

Express.js name is disclosed by default into the `x-powered-by` HTTP header:

```
let express = require('express');
let app = express(); // Sensitive

app.get('/', function (req, res) {
  res.send('hello')
});
```

Compliant Solution

`x-powered-by` HTTP header should be disabled in Express.js with `app.disable` or with helmet hidePoweredBy middleware:

```
let express = require('express');

let app1 = express();  // Compliant
app1.disable("x-powered-by");

let helmet = require("helmet");
let app2 = express(); // Compliant
app2.use(helmet.hidePoweredBy());
```

See

- OWASP Top 10 2021 Category A5 - Security Misconfiguration

Files should contain an empty newline at the end

Code Smell

An open curly brace should be located at the end of a line

Code Smell

Tabulation characters should not be used

Code Smell

Function and method names should comply with a naming convention

Code Smell

- OWASP Testing Guide - OTG-INFO-008 - Fingerprint Web Application Framework
- OWASP Top 10 2017 Category A6 - Security Misconfiguration
- MITRE, CWE-200 - Information Exposure

Available In:

sonarcloud sonarqube