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...

Return values from functions without side effects should not be ignored
Bug

Special identifiers should not be bound or assigned
Bug

Values should not be uselessly incremented
Bug

Related "if/else if" statements should not have the same condition
Bug

Objects should not be created to be dropped immediately without being used
Bug

Identical expressions should not be used on both sides of a binary operator
Bug

All code should be reachable
Bug

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

Using pseudorandom number generators (PRNGs) is security-sensitive

Analyze your code

Security Hotspot Critical cwe owasp

Using pseudorandom number generators (PRNGs) is security-sensitive. For example, it has led in the past to the following vulnerabilities:

- CVE-2013-6386
- CVE-2006-3419
- CVE-2008-4102

When software generates predictable values in a context requiring unpredictability, it may be possible for an attacker to guess the next value that will be generated, and use this guess to impersonate another user or access sensitive information.

As the `Math.random()` function relies on a weak pseudorandom number generator, this function should not be used for security-critical applications or for protecting sensitive data. In such context, a cryptographically strong pseudorandom number generator (CSPRNG) should be used instead.

Ask Yourself Whether

- the code using the generated value requires it to be unpredictable. It is the case for all encryption mechanisms or when a secret value, such as a password, is hashed.
- the function you use generates a value which can be predicted (pseudo-random).
- the generated value is used multiple times.
- an attacker can access the generated value.

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

Recommended Secure Coding Practices

- Use a cryptographically strong pseudorandom number generator (CSPRNG) like `crypto.getRandomValues()`.
- Use the generated random values only once.
- You should not expose the generated random value. If you have to store it, make sure that the database or file is secure.

Sensitive Code Example

```
const val = Math.random(); // Sensitive
// Check if val is used in a security context.
```

Compliant Solution

```
// === Client side ===
const crypto = window.crypto || window.msCrypto;
var array = new Uint32Array(1);
crypto.getRandomValues(array); // Compliant for security-sen

// === Server side ===
```

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

```
const crypto = require('crypto');
const buf = crypto.randomBytes(1); // Compliant for security
```

See

- OWASP Top 10 2021 Category A2 - Cryptographic Failures
- OWASP Top 10 2017 Category A3 - Sensitive Data Exposure
- Mobile AppSec Verification Standard - Cryptography Requirements
- OWASP Mobile Top 10 2016 Category M5 - Insufficient Cryptography
- MITRE, CWE-338 - Use of Cryptographically Weak Pseudo-Random Number Generator (PRNG)
- MITRE, CWE-330 - Use of Insufficiently Random Values
- MITRE, CWE-326 - Inadequate Encryption Strength
- MITRE, CWE-1241 - Use of Predictable Algorithm in Random Number Generator
- Derived from FindSecBugs rule Predictable Pseudo Random Number Generator

Available In:

sonarcloud sonarqube