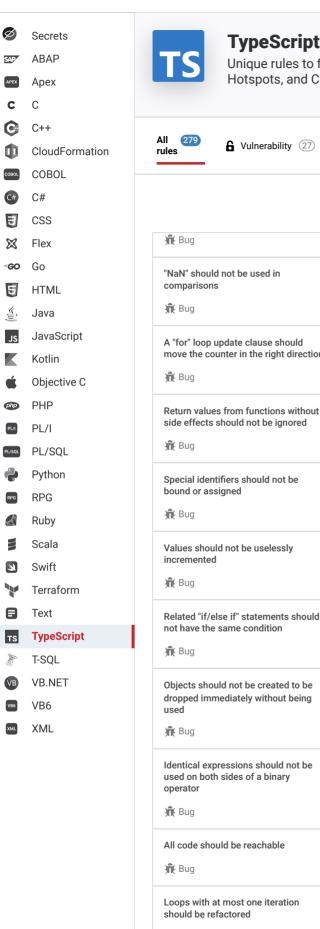
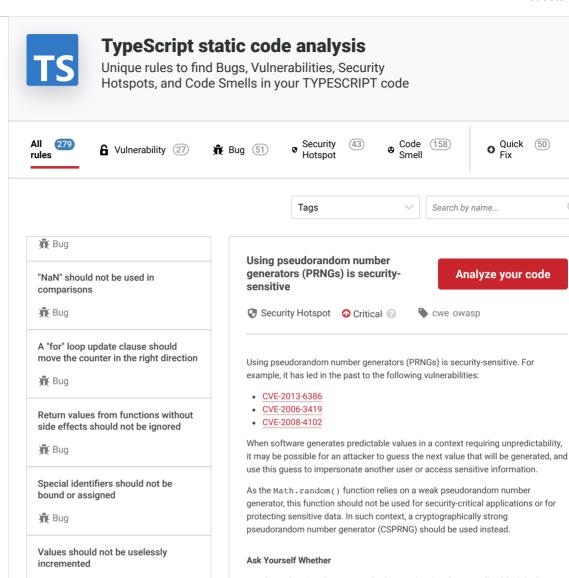


Products ✓





- 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 (pseudorandom).
- 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 ===
```

Rug

Variables should not be self-assigned

TypeScript static code analysis: Us

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

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:

© 2008-2022 SonarSource S.A., Switzerland. All content is copyright protected. SONAR, SONARSOURCE, SONARLINT, SONARQUBE and SONARCLOUD are trademarks of SonarSource S.A. All other trademarks and copyrights are the property of their respective owners. All rights are expressly reserved.

Privacy Policy

sonarcloud 🐼 | sonarqube