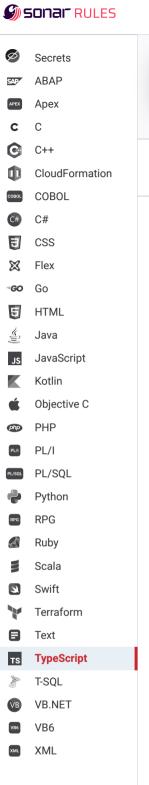
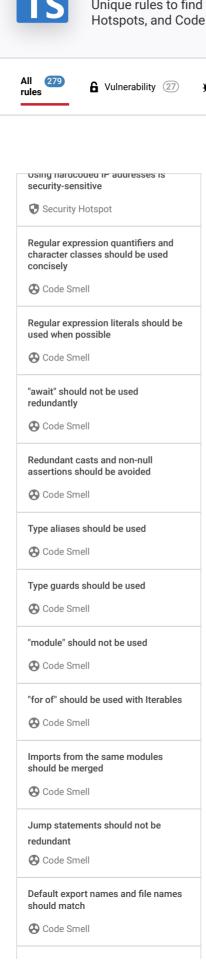
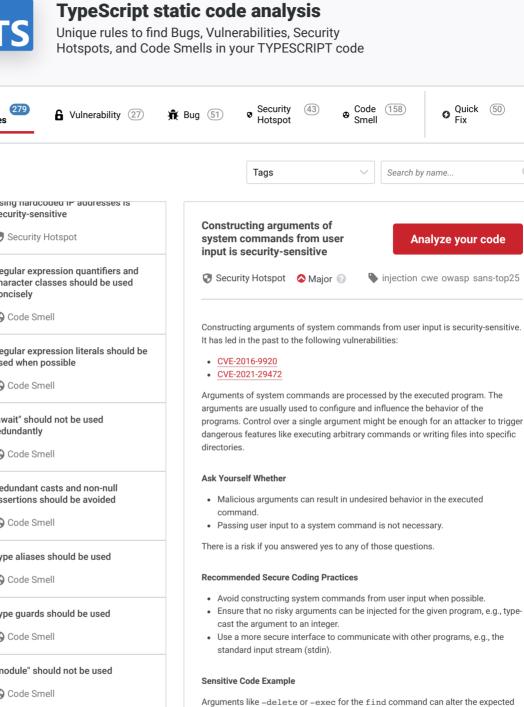


Products >







Arguments like -delete or -exec for the find command can alter the expected behavior and result in vulnerabilities:

```
const { spawn } = require("child_process");
const input = req.query.input;
const proc = spawn("/usr/bin/find", [input]); // Sensitive
```

Compliant Solution

Use an allow-list to restrict the arguments to trusted values:

```
const { spawn } = require("child_process");
const input = req.query.input;
if (allowed.includes(input)) {
  const proc = spawn("/usr/bin/find", [input]);
```

See

TypeScript static code analysis: Constructing arguments of system commands from user input is security-sensitive

TypeScript static code analysis: Constr
The global "this" object should not be used

Code Smell

Catch" clauses should do more than rethrow

Code Smell

Boolean checks should not be inverted

Code Smell

Deprecated APIs should not be used

Code Smell

Wrapper objects should not be used

OWASP Top 10 2021 Category A3 - Injection
OWASP Top 10 2017 Category A1 - Injection
MITRE, CWE-88 - Argument Injection or Modification
SANS Top 25 - Insecure Interaction Between Components
CVE-2021-29472 - PHP Supply Chain Attack on Composer

Available In:

sonarcloud Sonarqube Developer Edition

© 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