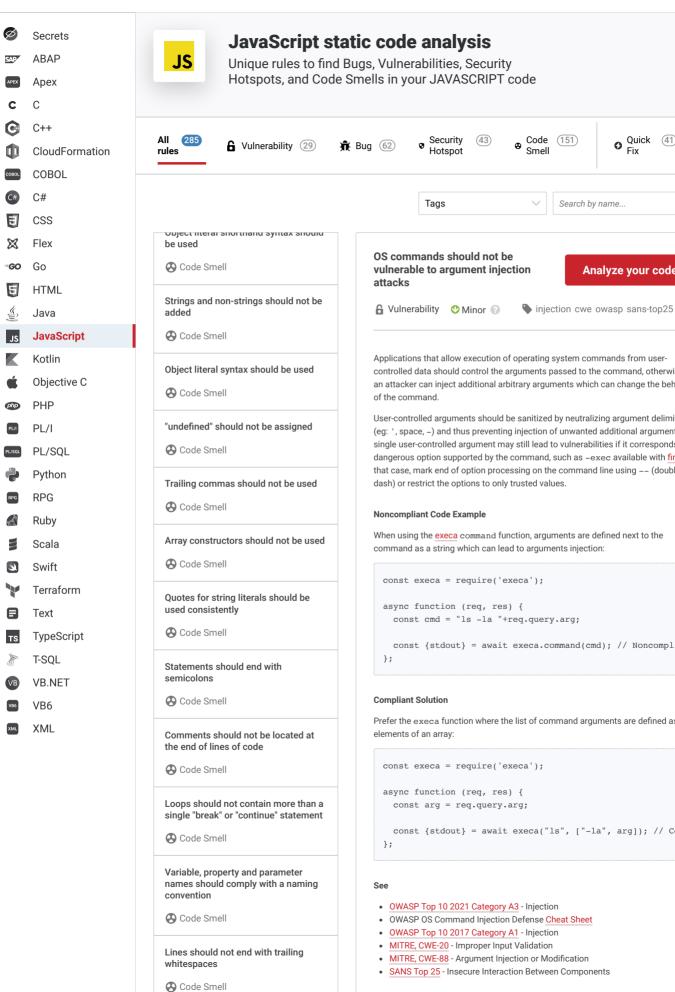


Products >



Analyze your code

O Quick 41 Fix

Applications that allow execution of operating system commands from usercontrolled data should control the arguments passed to the command, otherwise an attacker can inject additional arbitrary arguments which can change the behavior

(151)

Search by name.

User-controlled arguments should be sanitized by neutralizing argument delimiters (eg: ', space, -) and thus preventing injection of unwanted additional arguments. A single user-controlled argument may still lead to vulnerabilities if it corresponds to a dangerous option supported by the command, such as -exec available with find, in that case, mark end of option processing on the command line using -- (doubledash) or restrict the options to only trusted values.

command as a string which can lead to arguments injection:

```
const execa = require('execa');
 const cmd = "ls -la "+req.query.arg;
 const {stdout} = await execa.command(cmd); // Noncompliant
```

Prefer the execa function where the list of command arguments are defined as

```
const {stdout} = await execa("ls", ["-la", arg]); // Compl
```

- OWASP OS Command Injection Defense Cheat Sheet
- MITRE, CWE-20 Improper Input Validation
- SANS Top 25 Insecure Interaction Between Components

Available In:

JavaScript static code analysis: C 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

sonarcloud ⊗ sonarqube 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