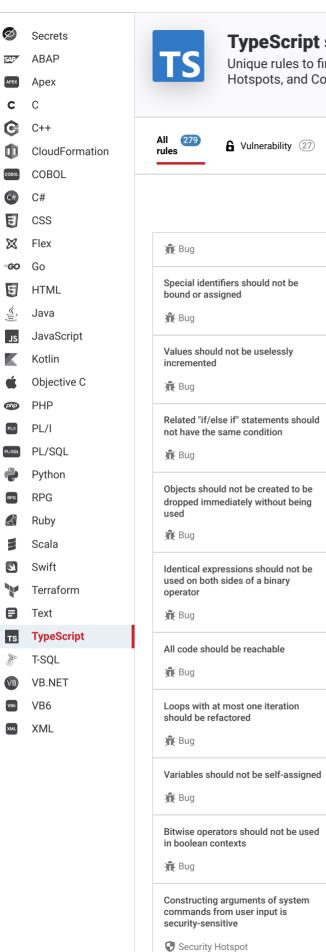
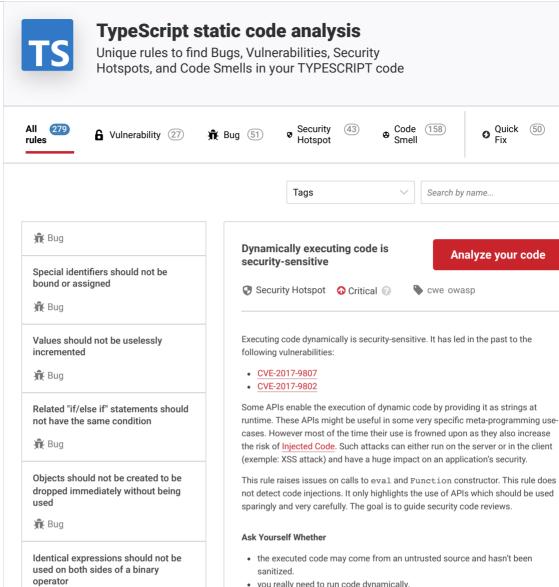


Products ✓





- · you really need to run code dynamically.

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

Recommended Secure Coding Practices

Regarding the execution of unknown code, the best solution is to not run code provided by an untrusted source. If you really need to do it, run the code in a sandboxed environment. Use jails, firewalls and whatever means your operating system and programming language provide (example: Security Managers in java, iframes and same-origin policy for javascript in a web browser).

Do not try to create a blacklist of dangerous code. It is impossible to cover all attacks that way.

Avoid using dynamic code APIs whenever possible. Hard-coded code is always

Sensitive Code Example

```
let value = eval('obj.' + propName); // Sensitive
let func = Function('obj' + propName); // Sensitive
```

Exceptions

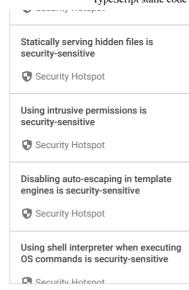
This rule will not raise an issue when the argument of the eval or $\operatorname{Function}$ is a literal string as it is reasonably safe.

- OWASP Top 10 2021 Category A3 Injection
- OWASP Top 10 2017 Category A1 Injection

Allowing requests with excessive

Security Hotspot

content length is security-sensitive



MITRE, CWE-95 - Improper Neutralization of Directives in Dynamically Evaluated Code ('Eval Injection')

Available In:

sonarcloud sonarqube

© 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