TypeScript static code analysis

Unique rules to find Bugs, Vulnerabilities, Security Hotspots, and Code Smells in your TYPESCRIPT code

"indexOf" checks should not be for positive numbers

Analyze your code

Code Smell | Major | suspicious

Most checks against an `indexOf` call against an array compare it with -1 because 0 is a valid index. Any checks which look for values >0 ignore the first element, which is likely a bug. If you're merely checking the presence of the element, consider using `includes` instead. Before using `includes` method make sure that your browser version is supporting it.

Noncompliant Code Example

```
var color = "blue";
var name = "ishmael";

var arr = [color, name];

if (arr.indexOf("blue") > 0) { // Noncompliant
  // ...
}
```

Compliant Solution

```
var color = "blue";
var name = "ishmael";

var arr = [color, name];

if (arr.indexOf("blue") >= 0) {
  // ...
}
if (arr.includes("blue")) {
  // ...
}
```

See

Array.prototype.includes() documentation at MDN

Available In:

sonarlint | sonarcloud | sonarqube

Standard outputs should not be used directly to log anything

Code Smell

Files should not have too many lines of code

Code Smell

Lines should not be too long

Code Smell

Debugger statements should not be used

Vulnerability