Security

Hotspot

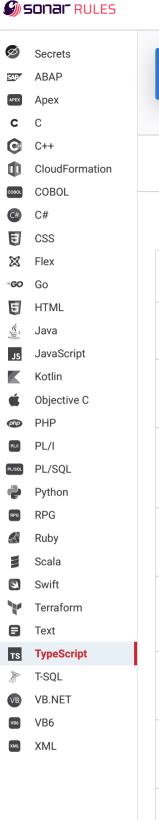
TypeScript static code analysis

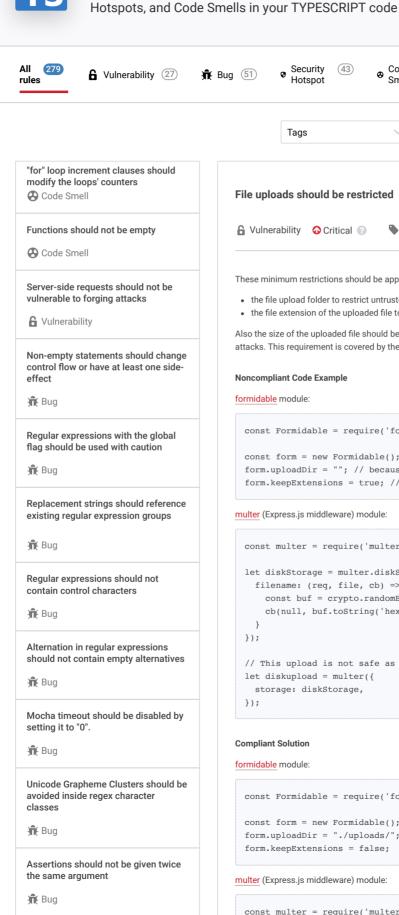
Unique rules to find Bugs, Vulnerabilities, Security



Products ✓

O Quick 50 Fix





Alternatives in regular expressions

should be arouped when used with

```
Search by name.
           Tags
                                             Analyze your code
File uploads should be restricted
cwe sans-top25 owasp express.js
These minimum restrictions should be applied when handling file uploads:

    the file upload folder to restrict untrusted files to a specific folder.

 • the file extension of the uploaded file to prevent remote code execution.
Also the size of the uploaded file should be limited to prevent denial of service
attacks. This requirement is covered by the rule {rule:javascript:S5693}.
Noncompliant Code Example
formidable module:
 const Formidable = require('formidable');
 const form = new Formidable(); // Noncompliant, this form is
 form.uploadDir = ""; // because upload dir is not defined (b
 form.keepExtensions = true; // and file extensions are kept
multer (Express.js middleware) module:
 const multer = require('multer');
 let diskStorage = multer.diskStorage({ // Noncompliant: no d
    filename: (req, file, cb) => {
      const buf = crypto.randomBytes(20);
      cb(null, buf.toString('hex'))
 });
 // This upload is not safe as no destination specified, /var
 let diskupload = multer({
   storage: diskStorage,
 });
Compliant Solution
formidable module:
 const Formidable = require('formidable');
 const form = new Formidable(); // Compliant
 form.uploadDir = "./uploads/";
 form.keepExtensions = false;
multer (Express.js middleware) module:
 const multer = require('multer');
```

let diskStorage = multer.diskStorage({ // Compliant

(158)

anchors

Bug

Promise rejections should not be caught by 'try' block

Bug

Collection elements should not be replaced unconditionally

Bug

Constructors should not be declared inside interfaces

```
filename: (req, file, cb) => {
  const buf = crypto.randomBytes(20);
  cb(null, buf.toString('hex'))
},
  destination: (req, file, cb) => {
    cb(null, './uploads/')
}
});

let diskupload = multer({
  storage: diskStorage,
});
```

## See

- OWASP Top 10 2021 Category A4 Insecure Design
- MITRE, CWE-434 Unrestricted Upload of File with Dangerous Type
- MITRE, CWE-400 Uncontrolled Resource Consumption
- OWASP Unrestricted File Upload Unrestricted File Upload
- SANS Top 25 Insecure Interaction Between Components

Available In:

sonarlint ⊖ | 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