Security

Hotspot

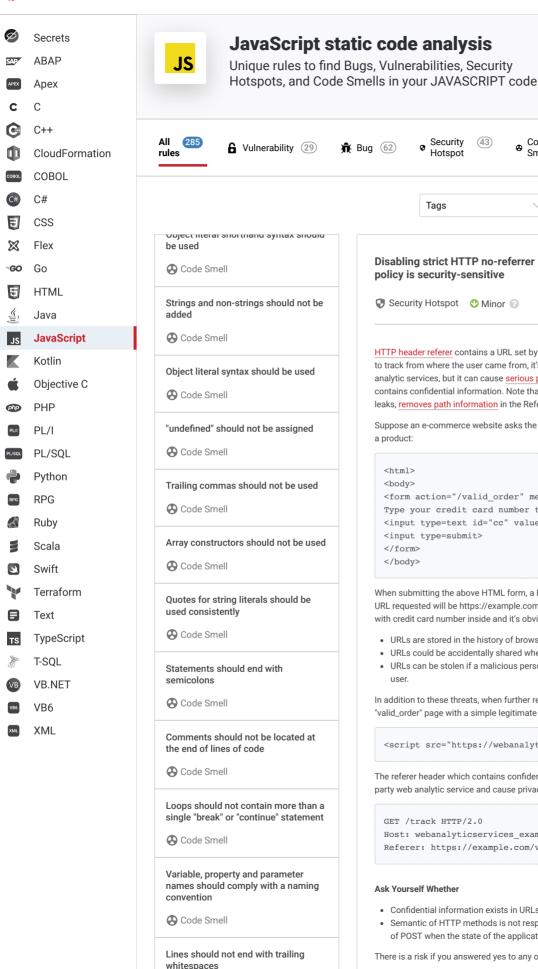
Tags

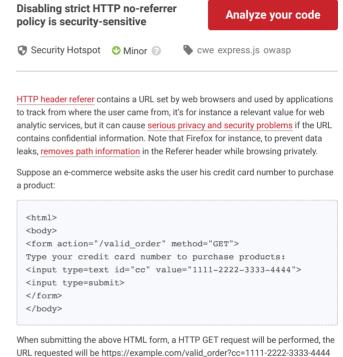
(43)



Products >

O Quick 41 Fix





⊗ Code

Smell

Search by name.

- URLs are stored in the history of browsers.
- URLs could be accidentally shared when doing copy/paste actions.
- URLs can be stolen if a malicious person looks at the computer screen of an

with credit card number inside and it's obviously not secure for these reasons:

In addition to these threats, when further requests will be performed from the "valid_order" page with a simple legitimate embedded script like that:

<script src="https://webanalyticservices_example.com/track">

The referer header which contains confidential information will be send to a third party web analytic service and cause privacy issue:

GET /track HTTP/2.0 Host: webanalyticservices_example.com Referer: https://example.com/valid_order?cc=1111-2222-3333-4

Ask Yourself Whether

- · Confidential information exists in URLs.
- Semantic of HTTP methods is not respected (eg: use of a GET method instead of POST when the state of the application is changed).

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

Recommended Secure Coding Practices

Code Smell

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

Confidential information should not be set inside URLs (GET requests) of the application and a safe (ie: different from unsafe-url or no-referrer-when-downgrade) referrer-Policy header, to control how much information is included in the referer header, should be used.

Sensitive Code Example

In Express.js application the code is sensitive if the helmet referrerPolicy middleware is disabled or used with no-referrer-when-downgrade or unsafe-url:

```
const express = require('express');
const helmet = require('helmet');

app.use(
  helmet.referrerPolicy({
    policy: 'no-referrer-when-downgrade' // Sensitive: no-re
  })
);
```

Compliant Solution

In Express,js application a secure solution is to user the $\frac{\text{helmet}}{\text{melmet}}$ referrer policy middleware set to no-referrer:

```
const express = require('express');
const helmet = require('helmet');

let app = express();

app.use(
  helmet.referrerPolicy({
    policy: 'no-referrer' // Compliant
  })
);
```

See

- OWASP Top 10 2021 Category A1 Broken Access Control
- OWASP Top 10 2017 Category A3 Sensitive Data Exposure
- developer.mozilla.org Referrer-Policy
- developer.mozilla.org Referer header: privacy and security concerns
- MITRE, CWE-200 Exposure of Sensitive Information to an Unauthorized Actor

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