

Secrets

ABAP

Apex

C

C++

CloudFormation

COBOL

C#

CSS

Flex

Go

HTML

Java

JavaScript

Kotlin

Objective C

PHP

PL/I

PL/SQL

Python

RPG

Ruby

Scala

Swift

Terraform

Text

TypeScript

T-SQL

VB.NET

VB6

XML

JavaScript static code analysis

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

All rules285

Vulnerability29

Bug62

Security Hotspot43

Code Smell151

Quick Fix41

Tags

Search by name...

Object literal shorthand syntax should be used

Code Smell

Strings and non-strings should not be added

Code Smell

Object literal syntax should be used

Code Smell

"undefined" should not be assigned

Code Smell

Trailing commas should not be used

Code Smell

Array constructors should not be used

Code Smell

Quotes for string literals should be used consistently

Code Smell

Statements should end with semicolons

Code Smell

Comments should not be located at the end of lines of code

Code Smell

Loops should not contain more than a single "break" or "continue" statement

Code Smell

Variable, property and parameter names should comply with a naming convention

Code Smell

Lines should not end with trailing whitespaces

Code Smell

Creating cookies without the "secure" flag is security-sensitive

Analyze your code

Security Hotspot

Minor

cwe privacy sans-top25 express.js owasp

When a cookie is protected with the `secure` attribute set to `true` it will not be send by the browser over an unencrypted HTTP request and thus cannot be observed by an unauthorized person during a man-in-the-middle attack.

Ask Yourself Whether

- the cookie is for instance a *session-cookie* not designed to be sent over non-HTTPS communication.
- it's not sure that the website contains **mixed content** or not (ie HTTPS everywhere or not)

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

Recommended Secure Coding Practices

- It is recommended to use HTTPs everywhere so setting the `secure` flag to `true` should be the default behaviour when creating cookies.
- Set the `secure` flag to `true` for session-cookies.

Sensitive Code Example

cookie-session module:

```
let session = cookieSession({
  secure: false, // Sensitive
}); // Sensitive
```

express-session module:

```
const express = require('express');
const session = require('express-session');

let app = express();
app.use(session({
  cookie: {
    secure: false // Sensitive
  }
}));
```





cookies module:

```
let cookies = new Cookies(req, res, { keys: keys });

cookies.set('LastVisit', new Date().toISOString(), {
  secure: false // Sensitive
}); // Sensitive
```

https://rules.sonarsource.com/javascript/RSPEC-2092

1/2

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

[csrf](#) module:

```
const cookieParser = require('cookie-parser');
const csrf = require('csrf');
const express = require('express');

let csrfProtection = csrf({ cookie: { secure: false }}); //
```

Compliant Solution

[cookie-session](#) module:

```
let session = cookieSession({
  secure: true, // Compliant
}); // Compliant
```

[express-session](#) module:

```
const express = require('express');
const session = require('express-session');

let app = express();
app.use(session({
  cookie: {
    {
      secure: true // Compliant
    }
  }
}));
```

[cookies](#) module:

```
let cookies = new Cookies(req, res, { keys: keys });

cookies.set('LastVisit', new Date().toISOString(), {
  secure: true // Compliant
}); // Compliant
```

[csrf](#) module:

```
const cookieParser = require('cookie-parser');
const csrf = require('csrf');
const express = require('express');

let csrfProtection = csrf({ cookie: { secure: true }}); // C
```

See

- [OWASP Top 10 2021 Category A4](#) - Insecure Design
- [OWASP Top 10 2021 Category A5](#) - Security Misconfiguration
- [OWASP Top 10 2017 Category A3](#) - Sensitive Data Exposure
- [MITRE, CWE-311](#) - Missing Encryption of Sensitive Data
- [MITRE, CWE-315](#) - Cleartext Storage of Sensitive Information in a Cookie
- [MITRE, CWE-614](#) - Sensitive Cookie in HTTPS Session Without 'Secure' Attribute
- [SANS Top 25](#) - Porous Defenses

Available In:

