Repositories & Tools

Vulnerable: OWASP Juice Shop https://github.com/Marwan123-web/juice-shop

Clean: Sports-Web (tournaments/booking backend) https://github.com/Marwan123-web/sports-web

SAST: SonarCloud | SCA: OWASP Dependency-Check

Juice Shop Overview

Deliberately insecure Node.js e-commerce (OWASP Top 10).

Why: Known vulns (injections, XSS, auth bypass); benchmark for tool accuracy.

Sports-Web Overview

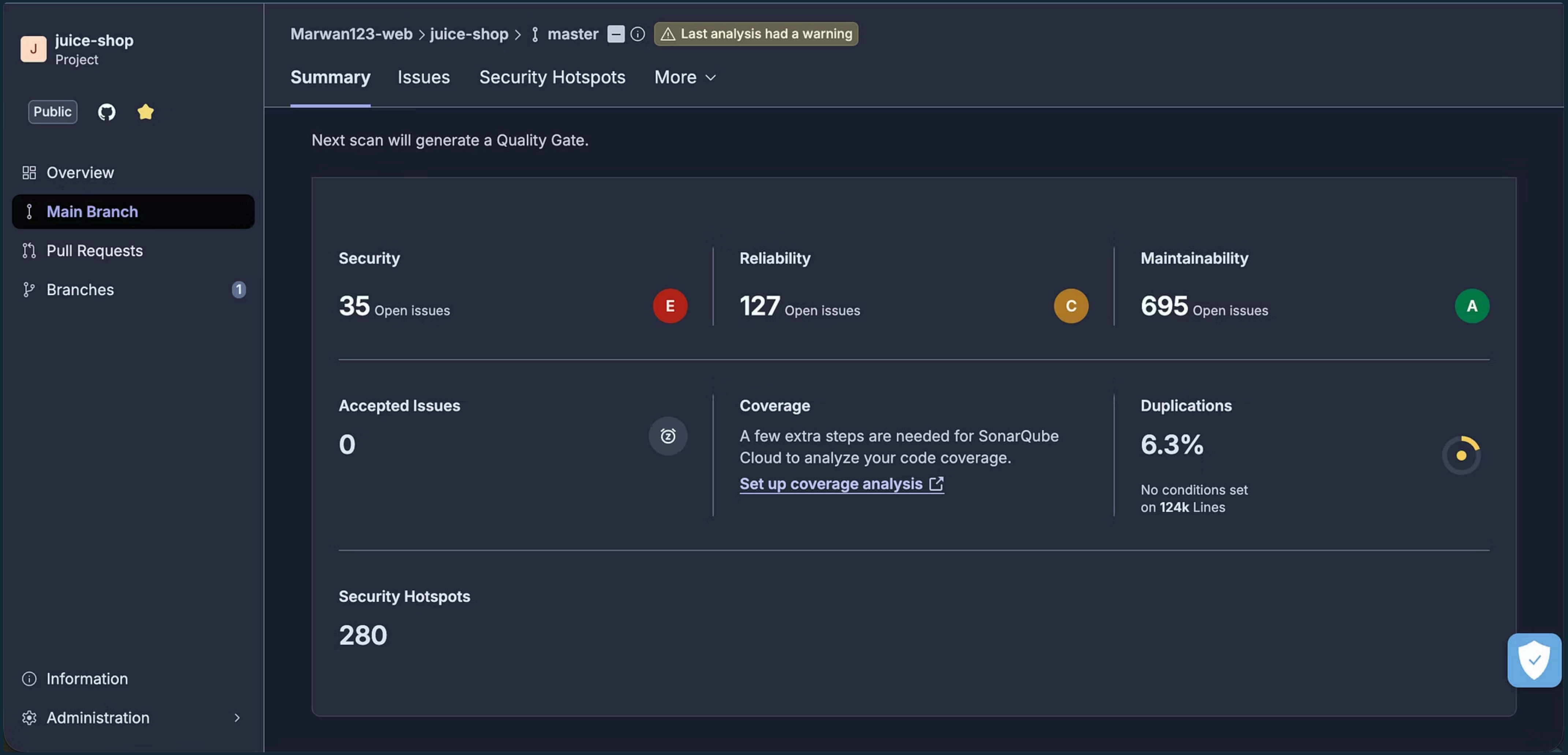
Personal Node.js/Express backend for sports app (tournaments, fields booking).

Why: No public advisories/CVEs; tests real-world clean code.

Juice Shop SAST

35 Security Issues

280 Hotspots



Next scan will generate a Quality Gate.

Security

35 Open issues E

Reliability

127 Open issues C

Maintainability

695 Open issues A

Accepted Issues

0

Coverage

A few extra steps are needed for SonarQube Cloud to analyze your code coverage.

Set up coverage analysis

Duplications

6.3%

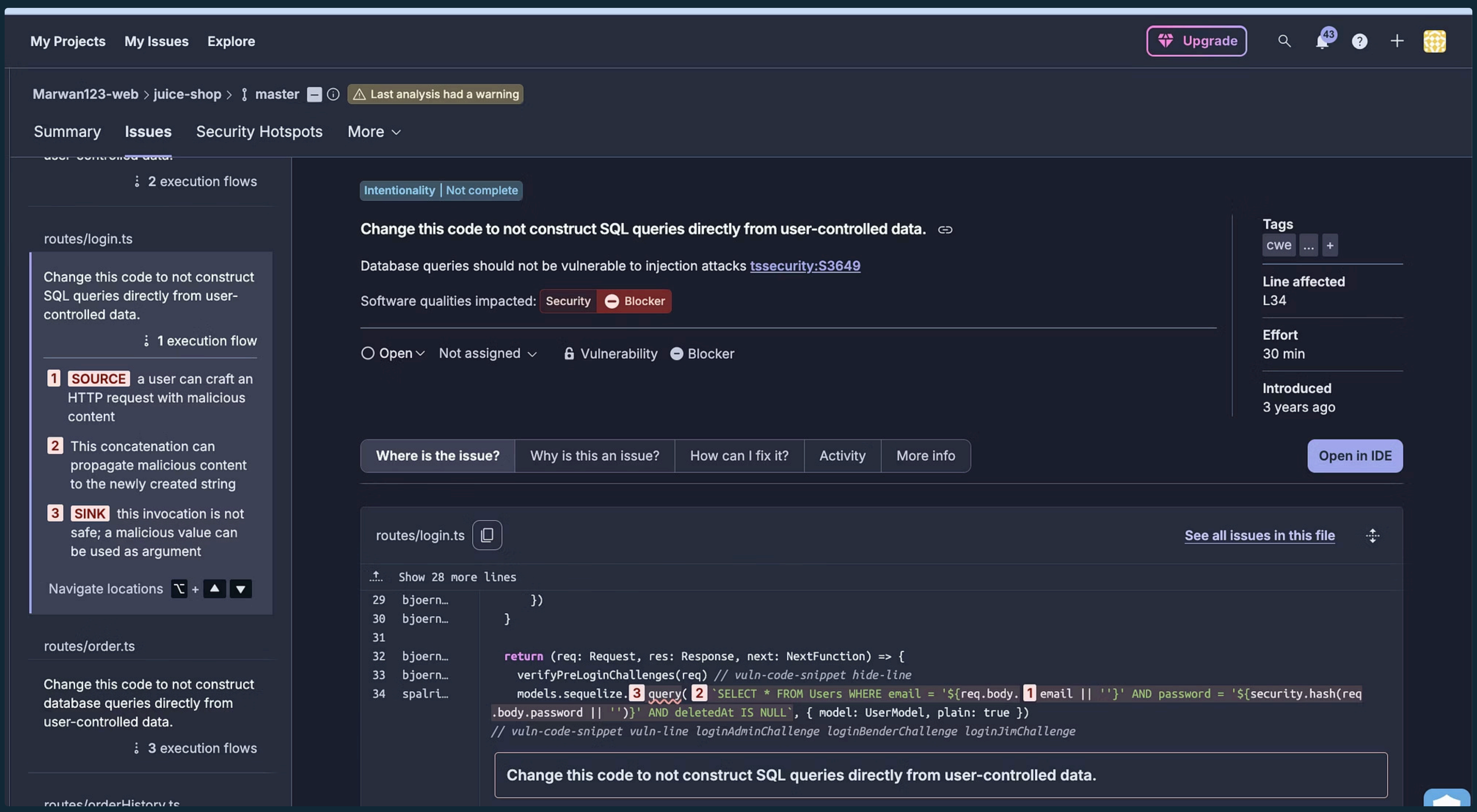
No conditions set on **124k** Lines

Security Hotspots

280

Juice Shop SAST: Key Vulnerabilities

SonarCloud identified critical security flaws, particularly SQL Injection and Cross-Site Scripting, showcasing the importance of SAST for high-risk applications.



2 execution flows

routes/login.ts

Change this code to not construct SQL queries directly from user-controlled data.

1 execution flow

1 SOURCE a user can craft an HTTP request with malicious content

2 This concatenation can propagate malicious content to the newly created string

3 SINK this invocation is not safe; a malicious value can be used as argument

Navigate locations ⌥ + ▲ ▼

routes/order.ts

Change this code to not construct database queries directly from user-controlled data.

3 execution flows

routes/orderHistory.ts

Intentionality | Not complete

Change this code to not construct SQL queries directly from user-controlled data.

Database queries should not be vulnerable to injection attacks tssecurity:S3649

Software qualities impacted: Security Blocker

○ Open · Not assigned · Vulnerability · Blocker

Tags: cwe ... +

Line affected: L34

Effort: 30 min

Introduced: 3 years ago

```
bjoern…          })
bjoern…        }

bjoern…        return (req: Request, res: Response, next: NextFunction) => {
bjoern…          verifyPreLoginChallenges(req) // vuln-code-snippet hide-line
spalri…          models.sequelize.3 query( 2 `SELECT * FROM Users WHERE email = '${req.body. 1 email || ''}' AND password = '${security.hash(req
.body.password || '')}' AND deletedAt IS NULL`, { model: UserModel, plain: true })
// vuln-code-snippet vuln-line loginAdminChallenge loginBenderChallenge loginJimChallenge
```

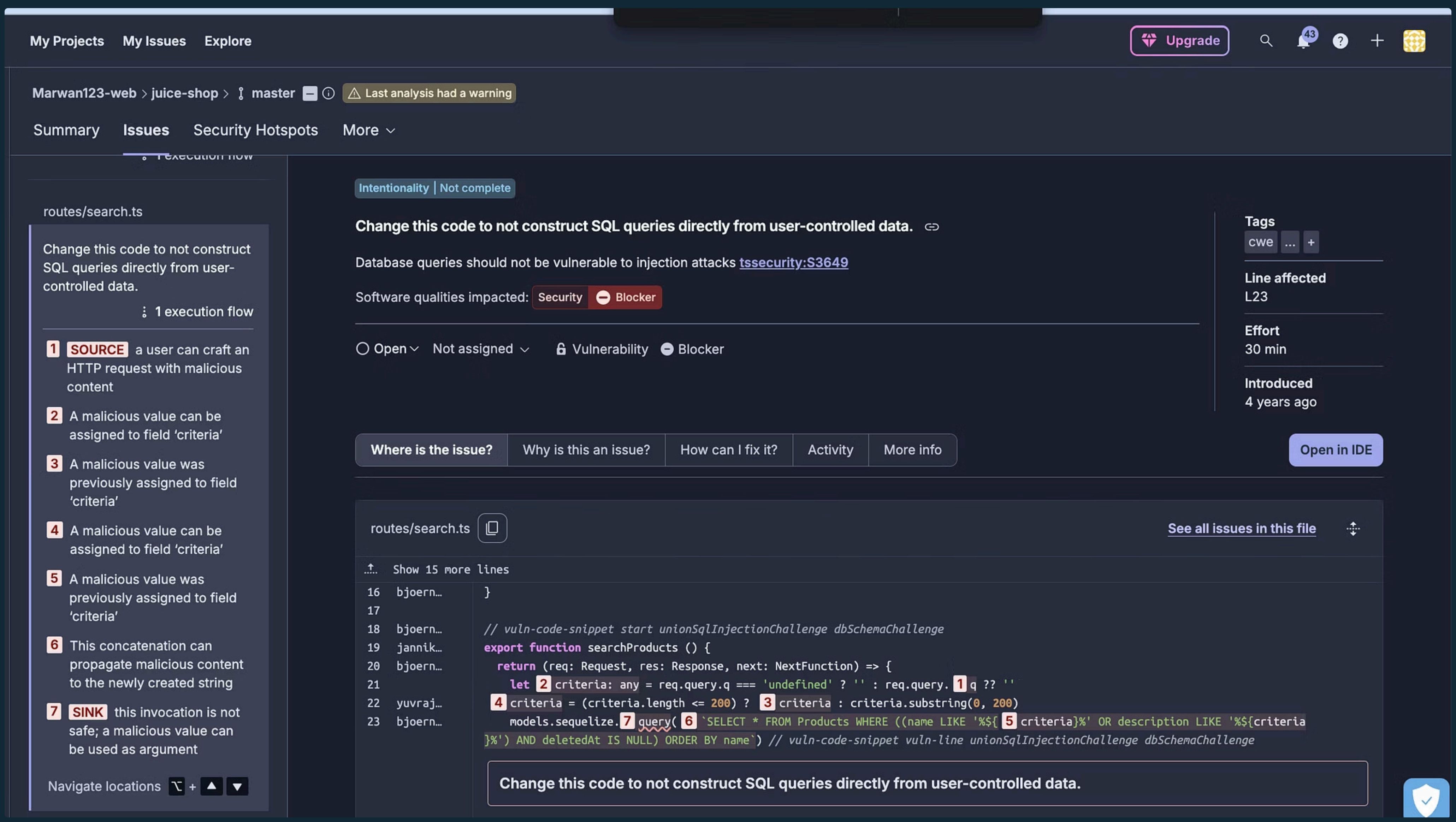
Change this code to not construct SQL queries directly from user-controlled data.

Login Bypass - SQL Injection

```
query(`SELECT * FROM Users WHERE email = '${req.body.email || ''}' AND password = '${security.hash(req.body.password || '')}'`)
```

FIX:(Parameterized Query)

```
const query = 'SELECT * FROM Users WHERE email = ? AND password = ?'; query(query, [email, hashedPassword])
```



routes/search.ts

Change this code to not construct SQL queries directly from user-controlled data.

1 execution flow

1 SOURCE a user can craft an HTTP request with malicious content

2 A malicious value can be assigned to field 'criteria'

3 A malicious value was previously assigned to field 'criteria'

4 A malicious value can be assigned to field 'criteria'

5 A malicious value was previously assigned to field 'criteria'

6 This concatenation can propagate malicious content to the newly created string

7 SINK this invocation is not safe; a malicious value can be used as argument

Navigate locations ⌥ + ▲ ▼

Intentionality | Not complete

Change this code to not construct SQL queries directly from user-controlled data.

Database queries should not be vulnerable to injection attacks tssecurity:S3649

Software qualities impacted: Security Blocker

○ Open · Not assigned · Vulnerability · Blocker

Tags: cwe ... +

Line affected: L23

Effort: 30 min

Introduced: 4 years ago

```
bjoern…      }

bjoern…      // vuln-code-snippet start unionSqlInjectionChallenge dbSchemaChallenge
jannik…      export function searchProducts () {
bjoern…        return (req: Request, res: Response, next: NextFunction) => {
               let 2 criteria: any = req.query.q === 'undefined' ? '' : req.query. 1 q ?? ''
yuvraj…       4 criteria = (criteria.length <= 200) ? 3 criteria : criteria.substring(0, 200)
bjoern…          models.sequelize.7 query( 6 `SELECT * FROM Products WHERE ((name LIKE '%${ 5 criteria}%' OR description LIKE '%${criteria
}%') AND deletedAt IS NULL) ORDER BY name`) // vuln-code-snippet vuln-line unionSqlInjectionChallenge dbSchemaChallenge
```

Change this code to not construct SQL queries directly from user-controlled data.

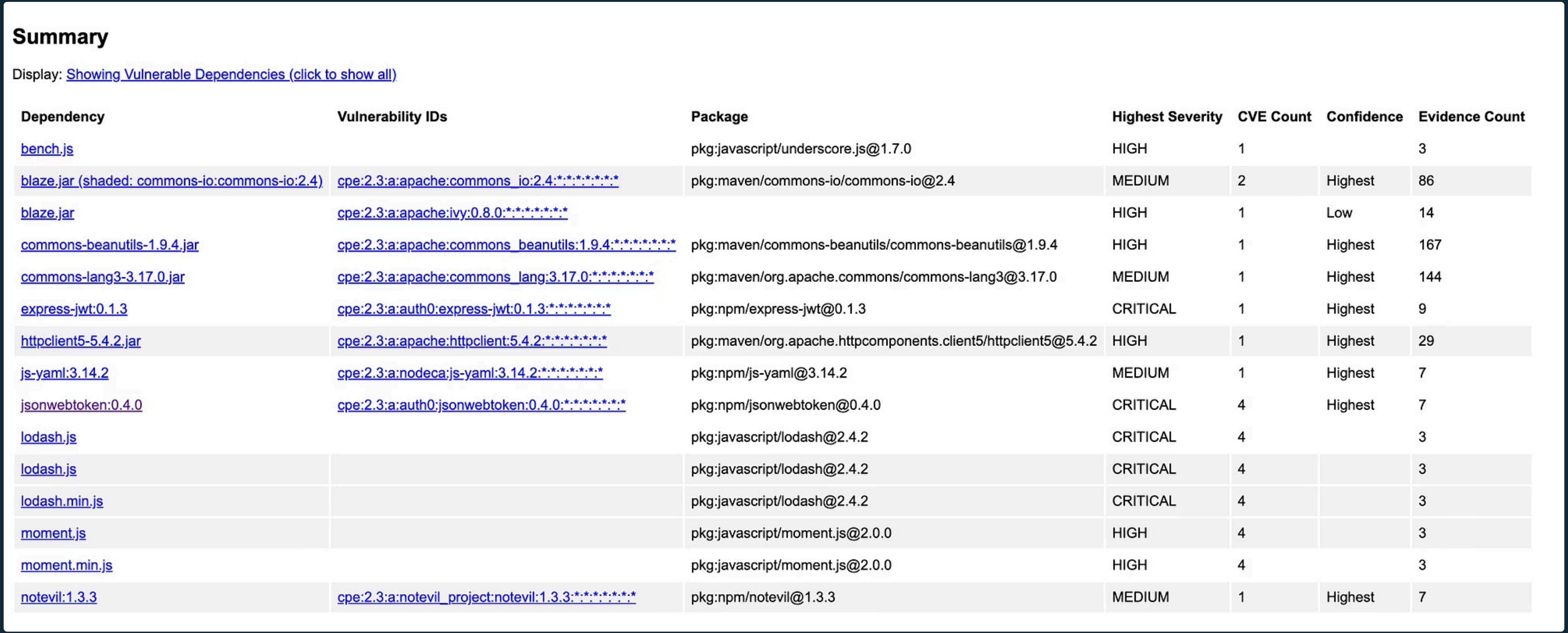
Search Function - Extracting data

```
query(`SELECT * FROM Products WHERE ((name LIKE '%${criteria}%' OR description LIKE '%${criteria}%')`)
```

FIX:(Parameterized Query)

```
query('SELECT * FROM Products WHERE name LIKE ? OR description LIKE ?', [`%${criteria}%`, `%${criteria}%`])
```

Juice Shop SCA



Summary

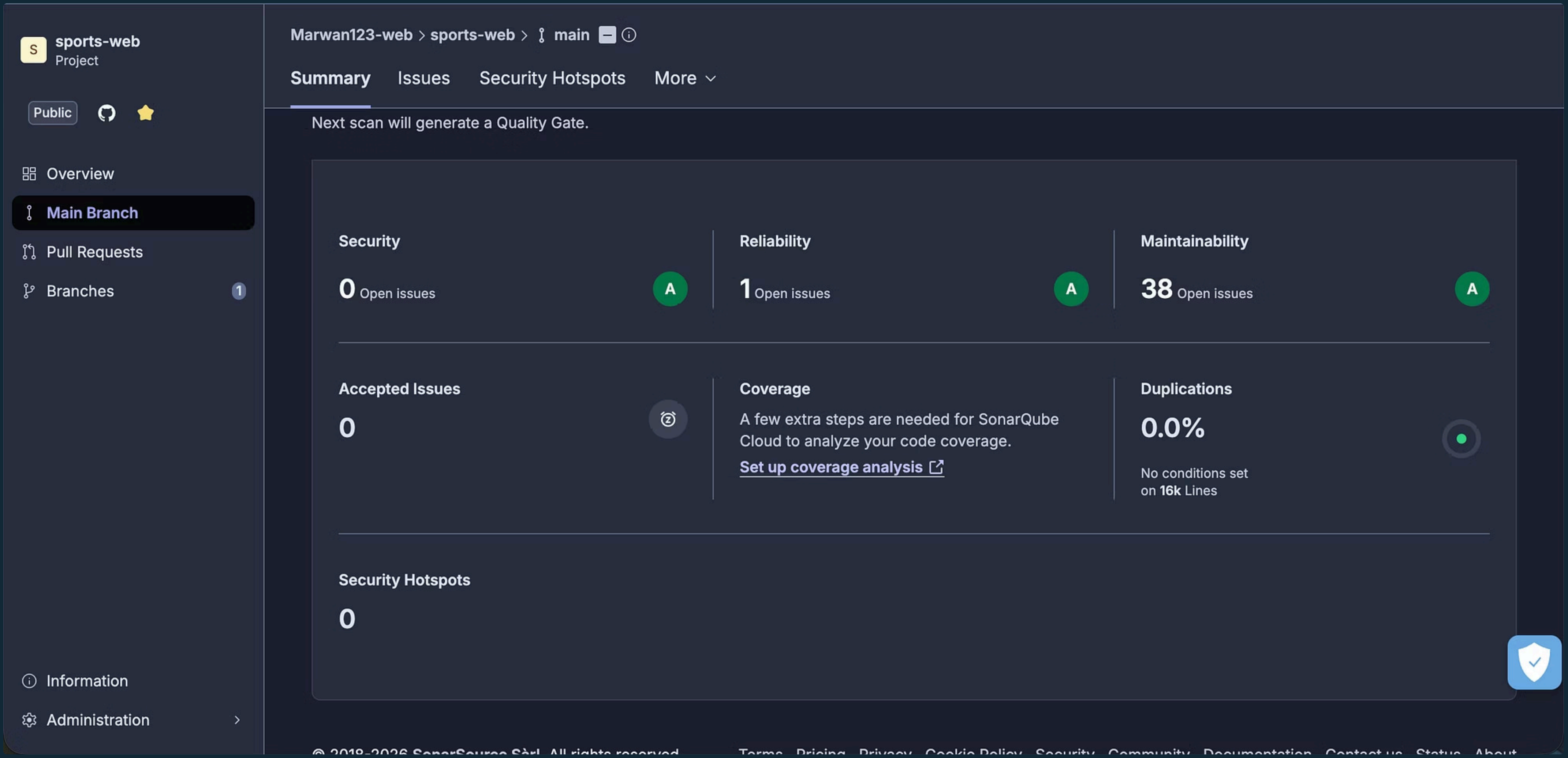
Display: Showing Vulnerable Dependencies (click to show all)

Dependency	Vulnerability IDs	Package	Highest Severity	CVE Count	Confidence	Evidence Count
bench.js		pkg:javascript/underscore.js@1.7.0	HIGH	1		3
blaze.jar (shaded: commons-io:commons-io:2.4)	cpe:2.3:a:apache:commons_io:2.4:*:*:*:*:*:*:*	pkg:maven/commons-io/commons-io@2.4	MEDIUM	2	Highest	86
blaze.jar	cpe:2.3:a:apache:ivy:0.8.0:*:*:*:*:*:*:*		HIGH	1	Low	14
commons-beanutils-1.9.4.jar	cpe:2.3:a:apache:commons_beanutils:1.9.4:*:*:*:*:*:*:*	pkg:maven/commons-beanutils/commons-beanutils@1.9.4	HIGH	1	Highest	167
commons-lang3-3.17.0.jar	cpe:2.3:a:apache:commons_lang:3.17.0:*:*:*:*:*:*:*	pkg:maven/org.apache.commons/commons-lang3@3.17.0	MEDIUM	1	Highest	144
express-jwt:0.1.3	cpe:2.3:a:auth0:express-jwt:0.1.3:*:*:*:*:*:*:*	pkg:npm/express-jwt@0.1.3	CRITICAL	1	Highest	9
httpclient5-5.4.2.jar	cpe:2.3:a:apache:httpclient:5.4.2:*:*:*:*:*:*:*	pkg:maven/org.apache.httpcomponents.client5/httpclient5@5.4.2	HIGH	1	Highest	29
js-yaml:3.14.2	cpe:2.3:a:nodeca:js-yaml:3.14.2:*:*:*:*:*:*:*	pkg:npm/js-yaml@3.14.2	MEDIUM	1	Highest	7
jsonwebtoken:0.4.0	cpe:2.3:a:auth0:jsonwebtoken:0.4.0:*:*:*:*:*:*:*	pkg:npm/jsonwebtoken@0.4.0	CRITICAL	4	Highest	7
lodash.js		pkg:javascript/lodash@2.4.2	CRITICAL	4		3
lodash.js		pkg:javascript/lodash@2.4.2	CRITICAL	4		3
lodash.min.js		pkg:javascript/lodash@2.4.2	CRITICAL	4		3
moment.js		pkg:javascript/moment.js@2.0.0	HIGH	4		3
moment.min.js		pkg:javascript/moment.js@2.0.0	HIGH	4		3
notevil:1.3.3	cpe:2.3:a:notevil_project:notevil:1.3.3:*:*:*:*:*:*:*	pkg:npm/notevil@1.3.3	MEDIUM	1	Highest	7

Sports-Web SAST

0 Security Issues/Hotspots ✅

38 Maintainability (minor dupls/unchanged funcs).



sports-web
Project

Public

Overview

Main Branch

Pull Requests

Branches 1

Information

Administration

Marwan123-web > sports-web > main

Summary | Issues | Security Hotspots | More

Next scan will generate a Quality Gate.

Security	Reliability	Maintainability
0 Open issues A	1 Open issues A	38 Open issues A

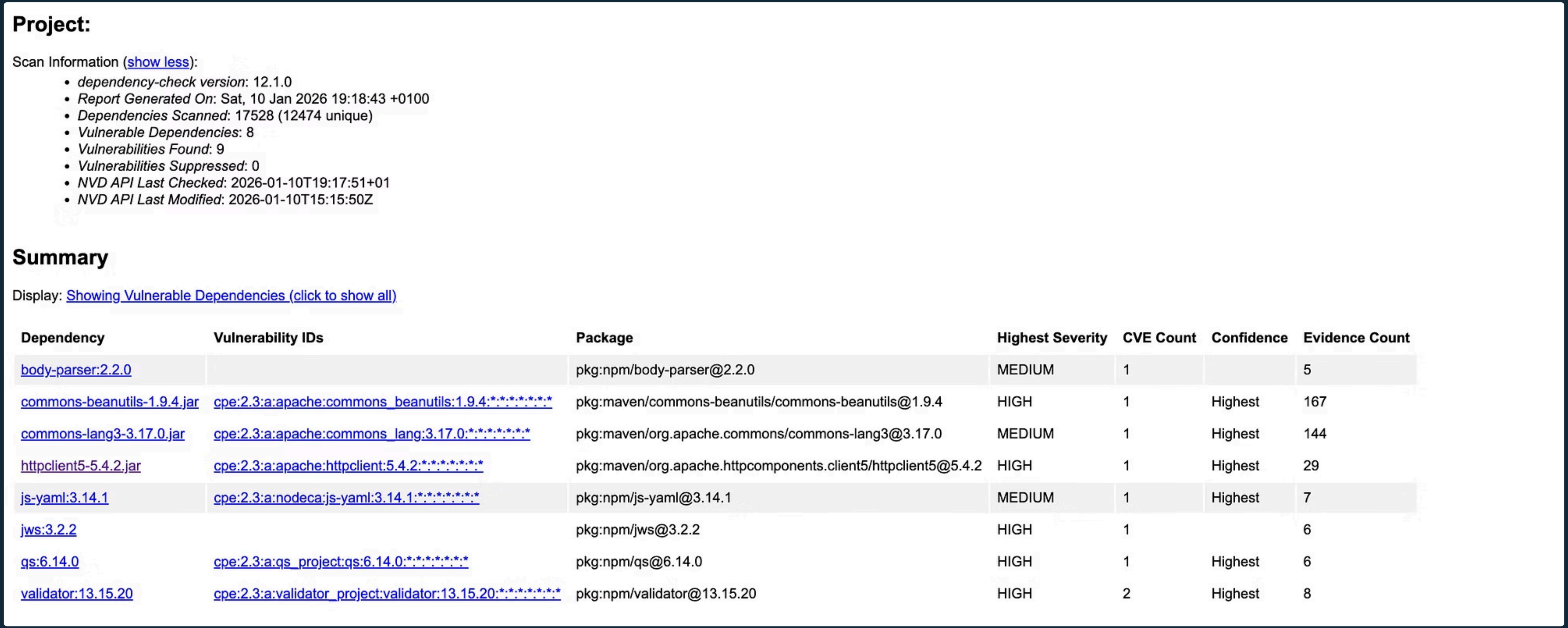
Accepted Issues	Coverage	Duplications
0	A few extra steps are needed for SonarQube Cloud to analyze your code coverage. Set up coverage analysis	0.0% No conditions set on 16k Lines

Security Hotspots

0

© 2018-2026 SonarSource Sàrl. All rights reserved. Terms Pricing Privacy Cookie Policy Security Community Documentation Contact us Status About

Sports-Web SCA



Project:

Scan Information (show less):

- *dependency-check version*: 12.1.0
- *Report Generated On*: Sat, 10 Jan 2026 19:18:43 +0100
- *Dependencies Scanned*: 17528 (12474 unique)
- *Vulnerable Dependencies*: 8
- *Vulnerabilities Found*: 9
- *Vulnerabilities Suppressed*: 0
- *NVD API Last Checked*: 2026-01-10T19:17:51+01
- *NVD API Last Modified*: 2026-01-10T15:15:50Z

Summary

Display: Showing Vulnerable Dependencies (click to show all)

Dependency	Vulnerability IDs	Package	Highest Severity	CVE Count	Confidence	Evidence Count
body-parser:2.2.0		pkg:npm/body-parser@2.2.0	MEDIUM	1		5
commons-beanutils-1.9.4.jar	cpe:2.3:a:apache:commons_beanutils:1.9.4:*:*:*:*:*:*:*	pkg:maven/commons-beanutils/commons-beanutils@1.9.4	HIGH	1	Highest	167
commons-lang3-3.17.0.jar	cpe:2.3:a:apache:commons_lang:3.17.0:*:*:*:*:*:*:*	pkg:maven/org.apache.commons/commons-lang3@3.17.0	MEDIUM	1	Highest	144
httpclient5-5.4.2.jar	cpe:2.3:a:apache:httpclient:5.4.2:*:*:*:*:*:*:*	pkg:maven/org.apache.httpcomponents.client5/httpclient5@5.4.2	HIGH	1	Highest	29
js-yaml:3.14.1	cpe:2.3:a:nodeca:js-yaml:3.14.1:*:*:*:*:*:*:*	pkg:npm/js-yaml@3.14.1	MEDIUM	1	Highest	7
jws:3.2.2		pkg:npm/jws@3.2.2	HIGH	1		6
qs:6.14.0	cpe:2.3:a:qs_project:qs:6.14.0:*:*:*:*:*:*:*	pkg:npm/qs@6.14.0	HIGH	1	Highest	6
validator:13.15.20	cpe:2.3:a:validator_project:validator:13.15.20:*:*:*:*:*:*:*	pkg:npm/validator@13.15.20	HIGH	2	Highest	8

Dependency Chain Analysis "Sports-Web Clean" → Why 8 Vulnerable Dependencies?

THE PROBLEM: ✅ Clean CODE (0 SAST issues) ❌ 8 Vulnerable DEPENDENCIES

SIMPLE EXPLANATION: My app → Direct deps → TRANSITIVE deps (hidden!)

KEY LESSON: Clean code ≠ Clean dependencies 90% vulnerabilities = TRANSITIVE

TAKEAWAY: Always run SCA + SAST Update direct deps → fixes transitive