Capstone1

Risk Insights

For Key Open-Source Node.js Projects

Danilo Briceno

Analyzing SBOM data, OSV reports, outdated libraries, and unauthored libraries to assess risk in key Node.js open-source projects.

TOPIC

1. Analysis of software libraries used by key open-source projects in node.js
2. Correlation between libraries with vulnerabilities, lacking authorship and not updated.
3. Identify high-risk open-source projects based on key risk indicators.

RISK IDENTIFICATION PROCESS

Data Collection

- Clone open-source projects
- Generate SBOM (Software Bill of Materials)
- Create vulnerability reports

- Git clone
- CycloneDX
- OSV Scanner

Data Generation

- Libraries: Name, version, author, license
- Vulnerabilities: Name, version, CVE, severity
- Versioning: Current vs. latest version

- Python Scripts
- npmjs registry

Data Analysis

- Projects with most libraries
- Most-used libraries across OSS
- Top vulnerable libraries
- Projects by vulnerability & severity
- Risk quadrant mapping

- Google Colab
- Pandas
- Bokeh

Open-Source Software (OSS) Projects

- socketio.libraries.csv
- mongoose.libraries.csv
- jest.libraries.csv
- all_libraries.csv
- axios.libraries.csv
- dotenv.libraries.csv
- chalk.libraries.csv
- lodash.libraries.csv
- bcrypt.libraries.csv
- tailwindcss.libraries.csv
- lodash.vuln.csv
- passport.libraries.csv

```
Project Name
socket.io      1373
jest           1227
mongoose       1103
axios           844
dotenv          773
chalk           750
lodash          699
bcrypt          293
tailwindcss     237
passport         39
Name: count, dtype: int64
```

Libraries

Project Name	Project Version	Project Author	Project License	Library Name	Library Version	Library Type	Library Author	Library License
socket.io	4.8.1	NA	MIT	core	7.24.7	library	The Babel Team	MIT
socket.io	4.8.1	NA	MIT	plugin-transform-object-assign	7.24.7	library	The Babel Team	MIT
socket.io	4.8.1	NA	MIT	preset-env	7.24.7	library	The Babel Team	MIT
socket.io	4.8.1	NA	MIT	register	7.24.6	library	The Babel Team	MIT
socket.io	4.8.1	NA	MIT	webtransport-transport-http3-quiche	1.1.4	library	Marten Richter	BSD-3-Clause
socket.io	4.8.1	NA	MIT	webtransport	1.1.4	library	Marten Richter	BSD-3-Clause

Library Name	Total_Usage	Used_by_Projects
semver	60	8
ansi-styles	58	9
chalk	46	8
strip-ansi	46	9
minimatch	45	10
debug	44	10
supports-color	42	9
type-fest	42	8
brace-expansion	41	9
string-width	41	9

Vulnerabilities

name	version	fixed	cvss	cve	severity
axios	1.7.7	1.8.2	7.7	CVE-2025-27152	HIGH
body-parser	1.20.2	1.20.3	8.7	CVE-2024-45590	HIGH
brace-expansion	1.1.11	2.0.2	3.1	CVE-2025-5889	LOW

- bcrypt.vuln.csv
- dotenv.vuln.csv
- passport.vuln.csv
- jest.vuln.csv
- axios.vuln.csv
- tailwindcss.vuln.csv
- chalk.vuln.csv
- lodash.vuln.csv
- mongoose.vuln.csv

Library Name	Total_Usage	Unique_Projects	Vuln_Count
debug	44	10	2
lru-cache	35	10	0
glob	37	10	0
minimatch	45	10	3
ms	34	10	1
wrap-ansi	30	9	0
shebang-command	9	9	0
cross-spawn	9	9	2
isexe	18	9	0
shebang-regex	9	9	0

Licenses

License	Permissive?	Copyleft?	Patent Protection
MIT	✅ Yes	❌ No	❌ No
ISC	✅ Yes	❌ No	❌ No
Apache-2.0	✅ Yes	❌ No	✅ Yes
BSD-3-Clause	✅ Yes	❌ No	❌ No
BSD-2-Clause	✅ Yes	❌ No	❌ No
BlueOak-1.0.0	✅ Yes	❌ No	✅ Yes
BSD (unspec)	✅ Yes	❌ No	❌ No
Python-2.0	✅ Yes	❌ No	❌ No
CC0-1.0	✅ Public Domain	❌ No	❌ No
CC-BY-4.0	❌ No	⚠️ Weak (Attribution)	❌ No

Library Name	Total_Usage	Unique_Libraries
MIT	5735	1903
ISC	807	168
Apache-2.0	209	79
BSD-3-Clause	164	43
BSD-2-Clause	135	34
BlueOak-1.0.0	61	37
BSD	18	16
Python-2.0	10	1
CC0-1.0	7	2
CC-BY-4.0	7	2
0BSD	6	1
Apache 2.0	6	4
CC-BY-3.0	6	1
WTFPL	5	5
Unlicense	4	4
BSD-like	4	3
MIT/X11	3	3

Authors

Library Author Name	Total_Usage	Unique_Libraries
Sindre Sorhus	1249	223
Isaac Z. Schlueter	388	81
The Babel Team	345	112
Jordan Harband	231	99
GitHub Inc.	183	56
Jon Schlinkert	139	55
Ben Coe	111	14
Titus Wormer	102	62
TJ Holowaychuk	88	21
Mathias Bynens	87	14

Outdated Libraries

Library	Current Version	Latest Version	Status
ajv	8.17.1	8.17.1	Up-to-date
assert-browserify	2.0.0	2.0.0	Up-to-date
babel-loader	8.2.5	10.0.0	Update available (10.0.0)
broken-link-checker	0.7.8	0.7.8	Up-to-date
bson	6.10.4	6.10.4	Up-to-date
buffer	5.7.1	6.0.3	Update available (6.0.3)
cheerio	1.1.2	1.1.2	Up-to-date
parse5	7.3.0	8.0.0	Update available (8.0.0)

```python
def get_latest_npm_version(package_name):
    url = f'https://registry.npmjs.org/{package_name}'
```

```
Library Name                    Current      Latest
devtools-protocol               0.0.1120988  0.0.1512837
devtools-protocol               0.0.1147663  0.0.1512837
devtools-protocol               0.0.1302984  0.0.1512837
api-extractor                   7.52.8       99.99.99
electron                        0.4.1        38.0.0
jest                            2.1.1        30.1.3
expect                          2.1.9        30.1.2
pretty-format                   2.1.9        30.0.5
node                            0.16.6       20.19.5
react                           2.0.0        19.1.1

Total outdated libraries: 1799 out of 3870
```

RISK Assessment by OSS Project

Count of CVE (Critical, High, Moderate, Low)

Count of outdated libraries

Count of author missing

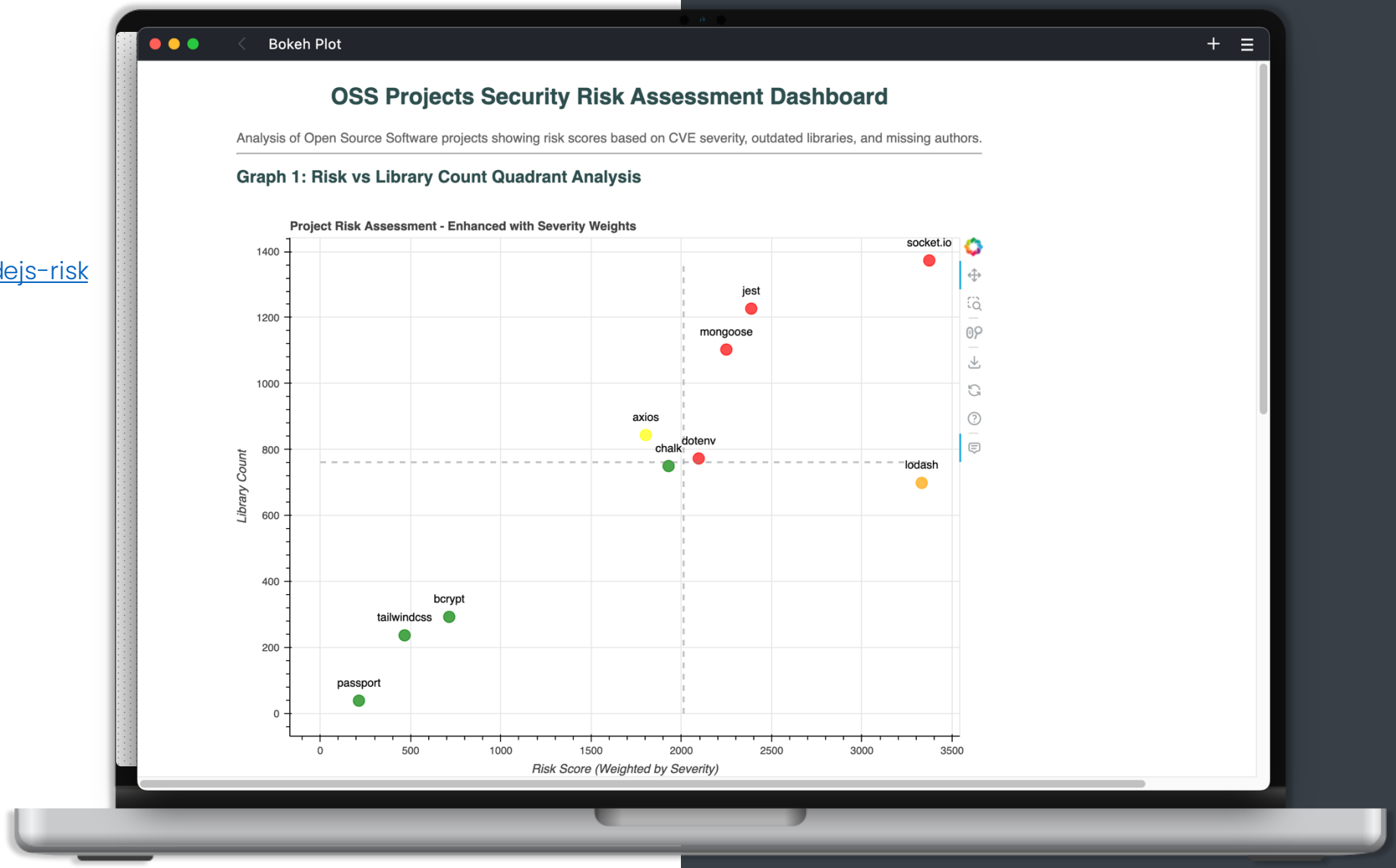
Vulnerability Score = (Critical × 20) + (High × 10) + (Moderate × 5) + (Low × 2)

Risk Score = Vulnerability Score + (Outdated Libraries × 3) + (Missing Authors × 1)

Demo

https://github.com/dan-breu/oss-nodejs-risk

https://oss-nodejs-risk.netlify.app/



Conclusion

- One library can depend on many others, and this can quickly grow into hundreds of libraries in open-source projects. SBOM reports helps us react quickly to software supply chain attacks and give important about open-source software.
- Open-source projects often share libraries, authors, and licenses, which can increase risk. Using multiple package managers makes it even harder to track and manage these risks.
- Socket.io and Lodash are open-source projects with higher risk scores. This is expe cted, as the risk grows with the number of libraries they use. Understanding the me trics that drive these risk scores is important so that proper measures can be take n to prevent potential security issues
- Debug, minimatch, and ms are libraries with known vulnerabilities that are used by all of the selected open-source projects. Analyzing these patterns helps us make better decisions and respond proactively.

Thank You

danbreu.com