Towards Measuring Vulnerabilities and Exposures in Open-Source Packages

Tobias Dam, Sebastian Neumaier
What is Open Source?

Open-source software is software with publicly available code. This means that anyone can not only inspect the source code but also modify it and enhance it.
# The Open Source Packages Landscape

<table>
<thead>
<tr>
<th>Project Manager</th>
<th>Projects</th>
<th>Perc.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Go</td>
<td>1,818,666</td>
<td>39.45</td>
</tr>
<tr>
<td>NPM</td>
<td>1,275,082</td>
<td>27.66</td>
</tr>
<tr>
<td>Packagist</td>
<td>313,278</td>
<td>6.80</td>
</tr>
<tr>
<td>Pypi</td>
<td>231,690</td>
<td>5.03</td>
</tr>
<tr>
<td>NuGet</td>
<td>199,447</td>
<td>4.33</td>
</tr>
<tr>
<td>Maven</td>
<td>184,871</td>
<td>4.01</td>
</tr>
<tr>
<td>Ruby</td>
<td>161,608</td>
<td>3.51</td>
</tr>
<tr>
<td>Others</td>
<td>425,020</td>
<td>9.22</td>
</tr>
</tbody>
</table>

![Graph showing the growth of various project managers over years](image)
Open Source is everywhere

- 96% of scanned codebases contained open source
- 76% of code in codebases was open source

“Open Source Security and Risk Analysis”, Synopsys, 2023
Dependencies...

https://xkcd.com/2347/
For instance: Heartbleed (2014)

- Vulnerability in OpenSSL
- Large proportion of online services, websites, VoIP phones, routers, etc. were vulnerable
- Was implemented by the only permanent developer

Security Risk is prevalent

- 84% of codebases contained at least one vulnerability
- 48% of codebases contained high-risk vulnerabilities

"Open Source Security and Risk Analysis", Synopsys, 2023
Common Vulnerability and Exposures (CVE)

CVE-Search / CVE-2021-44631

<table>
<thead>
<tr>
<th>ID</th>
<th>CVE-2021-44631</th>
</tr>
</thead>
<tbody>
<tr>
<td>Summary</td>
<td>A Buffer Overflow vulnerability exists in TP-LINK WR-886N 20190826 2.3.8 in the /cloud_config/router_post/reset_cloud_pwd feature, which allows malicious users to execute arbitrary code on the system via a crafted post request.</td>
</tr>
<tr>
<td>References</td>
<td><a href="https://github.com/Yu3H0/IoT_CVE/tree/main/886N/resetCloudPwdRegister">https://github.com/Yu3H0/IoT_CVE/tree/main/886N/resetCloudPwdRegister</a></td>
</tr>
<tr>
<td>Vulnerable Configurations</td>
<td>cpe:2.3:o:tp-link:tl-wr886n_firmware:20190826_2.3.8:<em>:</em>:<em>:</em>:*</td>
</tr>
<tr>
<td>CVSS</td>
<td>Base: 10.0 (as of 12-03-2022 - 04:23)</td>
</tr>
<tr>
<td>CWE</td>
<td>CWE-120</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>6,602</td>
<td>6,520</td>
<td>18,161</td>
<td>18,213</td>
<td>19,095</td>
<td>20,516</td>
<td>22,226</td>
</tr>
</tbody>
</table>
Research Goal

- Find corresponding Software repositories, packages, and CVE entries
  - Mapping of CVE entries to open-source projects
  - Quantitative analysis of the frequency and distribution of CVE
  - Identify challenges and quality issues wrt. the mapping
Mapping Approaches

- Open Data dump of the Libraries.io database
  - Includes metadata such as package manager, repository (e.g., GitHub) links, owner and keywords

- Evaluation of different CVE mapping approaches
  - Based on matching repository names, fuzzy string matching, and URLs
Findings

- Insights regarding number of vulnerabilities for certain package managers
- Noticeable upwards trend in vulnerabilities in recent years
- Improved approach evaluation is necessary
Identified Mapping Challenges

1. CVE entries do not contain clear identifiers of the code base
2. Incomplete CVE entries and potentially wrong attribute values
3. Multiple relevant packages to the CVE entry
4. No complete and reliable index of open source projects
Is this information sufficient..?

Are the reported bugs and vulnerabilities sufficient to make statements about the health of open source software projects?
Operational Risks

no development activity, no user updates, no feature upgrades or code improvements

- 89% were more than 4 years out of date
- 91% contained components that weren’t the current version
- 91% had received no development activity in the last 2 years
- 88% contained components with no activity in the last 2 years and contained components that weren’t the latest version

“Open Source Security and Risk Analysis”, Synopsys, 2023
Towards a Critical Open-Source Software Database

Goal

- Assess the "health" of (especially: critical) OSS projects
- "Health" is to be defined by various metrics
  - Security, stability, resilience, compliance, etc.
- Comprehensive analysis and ongoing evaluation on large software repositories
Metrics

Quantitative & qualitative metrics are to be applied

- **Quantitative & automated:**
  Crawling of meta-information (e.g. contributors, commits, activity), analysis of dependencies

- **Qualitative:**
  sustainability/funding, activity indicators, security policies
Metrics (Cont‘d)

Categories

- *Relevance metrics*: such as retrieval numbers, map the popularity and visibility of projects
- *Activity metrics*: highlight critical points, e.g. development failure or incipient inactivity
- *Security metrics*: e.g. known CVE vulnerabilities can show security issues
Platform Architecture

Open-Source Software
- Repositories
- GHTorrent
- Libraries.io
- Other Sources

Integration & UI
- IDE
- CLI Tools
- Build Processes
- CI/CD
- Notifications

Metrics

Update

Project Health Database

Analyse

Utilise
Next Steps

- Database of continuous assessment of metrics
  - Relevance and criticality become visible
  - Identify which support is target-oriented, efficient and effective (financially, working time, know-how, ...)

- Web platform displaying the results
  - Support for developers in software selection and continuous safety checks
  - Basis for making informed decisions about funding and support
Sebastian Neumaier
St. Pölten University of Applied Sciences
Austria

sebastian.neumaier@fhstp.ac.at
https://sebneumaier.wordpress.com/
https://twitter.com/sebneum