OSS Risk Management

Petr Špaček, Victoria Risk – NANOG 91, June 11, 2024
Internet Systems Consortium, Inc.
Open source developer of BIND 9, ISC DHCP and Kea DHCP
(NOT ISC2)

https://www.isc.org
Government experts are here to help …

- NIST Secure Software Development Framework (SSDF)
- EO14028 – Securing Critical Software
- NTIA Software BOM requirements
- CISA Secure by Design Pledge
- EU Cyber Resilience Act
- White House EO on Zero Trust (encryption for DNS and HTTP)
How do YOU assess OSS quality?

- We created a survey, and sent it to:
  - RIPE OSS working group
  - DNS-Ops IETF mailing list
  - Posted on ISC’s social media
  - We got 71 responses
Survey

https://ec.europa.eu/eusurvey/publication/RIPE88OpenSourceWGSurvey
<table>
<thead>
<tr>
<th>When selecting an open source system to use for a critical application, how do you build confidence in the software project?</th>
<th>Answers</th>
<th>Ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>documentation</td>
<td>45</td>
<td>63.380%</td>
</tr>
<tr>
<td>the user mailing list or forum is active and helpful</td>
<td>40</td>
<td>56.338%</td>
</tr>
<tr>
<td>releases are frequent/recent enough</td>
<td>39</td>
<td>54.930%</td>
</tr>
<tr>
<td>the software is already familiar to me</td>
<td>34</td>
<td>47.887%</td>
</tr>
<tr>
<td>versions are maintained for long enough</td>
<td>31</td>
<td>43.662%</td>
</tr>
<tr>
<td>there is more than one regular committer</td>
<td>25</td>
<td>35.211%</td>
</tr>
<tr>
<td>we conduct thorough testing of the software</td>
<td>25</td>
<td>35.211%</td>
</tr>
<tr>
<td>number of open, unresolved issues</td>
<td>16</td>
<td>22.535%</td>
</tr>
<tr>
<td>history of CVEs</td>
<td>16</td>
<td>22.535%</td>
</tr>
<tr>
<td>popularity (e.g. stars on GitHub)</td>
<td>14</td>
<td>19.718%</td>
</tr>
<tr>
<td>financial sponsors of the project are identified</td>
<td>12</td>
<td>16.901%</td>
</tr>
<tr>
<td>project test suite</td>
<td>9</td>
<td>12.676%</td>
</tr>
<tr>
<td>adequate packaging options</td>
<td>9</td>
<td>12.676%</td>
</tr>
<tr>
<td>software development process</td>
<td>8</td>
<td>11.268%</td>
</tr>
<tr>
<td>published roadmap</td>
<td>5</td>
<td>7.042%</td>
</tr>
<tr>
<td>badges on the project's homepage (example)</td>
<td>1</td>
<td>1.408%</td>
</tr>
</tbody>
</table>
A DNS server – example OSS project

https://gitlab.isc.org/isc-projects/bind9
BIND 9 key quality processes

- Development process (commit requirements, coding standards, peer review)
- Test suite, test coverage, automated tools
- Ad-hoc and performance testing (real data)
- Investigation of all reported security issues (very time-consuming)
Continuous integration

For main

Scheduled 112 jobs 69 minutes 53 seconds, queued for 14 seconds

Pipeline Needs Jobs 112 Failed Jobs 1 Tests 6321
BIND 9 Release process

Release Checklist

Before the Code Freeze

- [ ] QA: Review Patches on top of current integration versions. git branch -D bind-9.8.1-bf as git branch origin/bind-9.8.1
- [ ] QA: Inform Support and Marketing of impending release (and get estimated release dates).
- [ ] QA: Ensure there are no permanent test failures on any platform. Check public and private scheduled pipelines.
- [ ] QA: Check whether all issues assigned to the release reviewer are resolved.
- [ ] QA: Ensure that there are no outstanding feature requests in the private repository (Subscription Editor only).
- [ ] QA: Ensure all large requests marked for backporting have beeninned backporting.
- [ ] QA: Announce (on Mathtool) that the code freeze is in effect.

Before the Tagging Deadline

- [ ] QA: Inspect the current output of the check-version-conf-tests job to verify that no unexpected backward-incompatible change was introduced in the current release cycle.
- [ ] QA: Ensure release rules are correct, ask support and Marketing to check them as well. Example.
- [ ] QA: Add a release matrix to CI/CD. Example: Examples 9.15, 9.16.
- [ ] QA: Add a release matrix to CI/CD. Example: Subscription Editor only. Example.
- [ ] QA: Update BIND 9 version in configuration at fi.tan or version 0.10.
- [ ] QA: Reduce configure using Autobot on OpenBSD. Example: Example.
- [ ] QA: Update smaller settings for all maintained branches to allow merging to them again (public, private).
- [ ] QA: Tag the releases in the private repository (git tag -s -a "BIND 9-x" v.*.v.y).

Before the ASN Deadline (for ASN Releases) or the Public Release Date (for Regular Releases)

- [ ] QA: Check that the formatting is correct for the HTML version of release notes.
- [ ] QA: Check that the formatting of the generated man pages is correct.
- [ ] QA: Verify BIND 9 OOT results for the logs created and sign off on the releases to be published.
- [ ] QA: Update OOTlib settings for all maintained branches to allow merging to them again (public, private).
- [ ] QA: Update htdocs version, htdocs, and make sure NAV is meeting the release notes and updating the version string for each maintained branch.
- [ ] QA: Prepare the Subscription Editor branches (including recent releases) prep commits on top of the open source branches with updated version strings.
- [ ] QA: Announce (on Mathtool) that the code freeze is over.
- [ ] QA: Request signatures for the tags, providing their location and checksums. Ask signers on Mathtool.
- [ ] Include or simulate actual tags as identica.
- [ ] Signoff and validate that tags are identic.
- [ ] QA: Validate that tags are identic, cmp, and signature is unique.
- [ ] QA: Verify that tags are modified and checksums are unique. Run addition bind-ock on the branch to pre-publish.
- [ ] QA: Prepare the patches/subdirectory for each security release if applicable.
- [ ] QA: Pre-parchs ASN and/or Subscription Editor patches if branch packages can be built.
- [ ] QA: Build and test ASN and/or Subscription Editor packages (in standalone branch to private repo). Example.
- [ ] Marketing: Prepare and send out ASN emails as outlined in the CVE checklist if applicable.

On the Day of Public Release

- [ ] QA: Visit for clearance from Security Officer to proceed with the public release (if applicable).
- [ ] QA: Place patches in public announc on FTP site.
- [ ] QA: Inform Marketing of the release, providing FTP links for the published tarballs.
- [ ] QA: Use the Printing Press project to prepare a release announcement email.
- [ ] Marketing: Update the BIND 9 Information document in the SIG portal if any stable versions were released.
- [ ] Marketing: Send the release announcement emails to the announce mailing list (and to bind-users if a major release - example).
- [ ] Marketing: Announce release on social media sites.
- [ ] Marketing: Update Wikipedia entry for BIND.
- [ ] QA: Add the new patches to the vulnerability matrix in the Knowledge Base.
- [ ] QA: Send patch tickets to release support.
- [ ] QA: Build public Debian/Ubuntu packages.
- [ ] QA: Clone BIND Git with make sure patches are propagated to BIND.
- [ ] QA: Ensure all new patches are annotated and signed.
- [ ] QA: Update the public repositories to the public repository.
- [ ] QA: Build public Debian/Ubuntu.
- [ ] QA: Use BIND 9.15.12 and merge published release tags back into the their relevant development/maintenance branches.
- [ ] QA: Ensure public patches: true is removed from the cp_publish-config-tests job if it was set during the current release cycle.
- [ ] QA: Announce any issues that are assigned to the current release milestone and do not list a security vulnerability; make them public.
- [ ] QA: Announce any issues that are assigned to older release milestones and describe security vulnerabilities; make them public.
- [ ] QA: Build public Debian/Ubuntu with the relevant Deisl (e.g. Debian, Ubuntu, Fedora) by importing the relevant Debian/Ubuntu.
- [ ] QA: Run a pipeline to rebuild all images used in the Debian CI.
- [ ] QA: Update metadata files with the upcoming release information.
What makes a project trustworthy?

<table>
<thead>
<tr>
<th>What makes a project trustworthy?</th>
<th>Answers</th>
<th>Ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>documentation</td>
<td>45</td>
<td>63.380%</td>
</tr>
<tr>
<td>the user mailing list or forum is active and helpful</td>
<td>40</td>
<td>56.338%</td>
</tr>
<tr>
<td>releases are frequent/recent enough</td>
<td>39</td>
<td>54.930%</td>
</tr>
<tr>
<td>the software is already familiar to me</td>
<td>34</td>
<td>47.887%</td>
</tr>
<tr>
<td>versions are maintained for long enough</td>
<td>31</td>
<td>43.662%</td>
</tr>
<tr>
<td>there is more than one regular committer</td>
<td>25</td>
<td>35.211%</td>
</tr>
<tr>
<td>we conduct thorough testing of the software</td>
<td>25</td>
<td>35.211%</td>
</tr>
<tr>
<td>number of open, unresolved issues</td>
<td>16</td>
<td>22.535%</td>
</tr>
<tr>
<td>history of CVEs</td>
<td>16</td>
<td>22.535%</td>
</tr>
<tr>
<td>popularity (e.g., stars on GitHub)</td>
<td>14</td>
<td>19.718%</td>
</tr>
<tr>
<td>financial sponsors of the project are identified</td>
<td>12</td>
<td>16.901%</td>
</tr>
<tr>
<td>project test suite</td>
<td>9</td>
<td>12.676%</td>
</tr>
<tr>
<td>adequate packaging options</td>
<td>9</td>
<td>12.676%</td>
</tr>
<tr>
<td>software development process</td>
<td>8</td>
<td>11.268%</td>
</tr>
<tr>
<td>published roadmap</td>
<td>5</td>
<td>7.042%</td>
</tr>
<tr>
<td>badges on the project's homepage (example)</td>
<td>1</td>
<td>1.408%</td>
</tr>
<tr>
<td>No Answer</td>
<td>0</td>
<td>0.000%</td>
</tr>
</tbody>
</table>
Self-imposed policies

- Coding & review procedures
- OpenSSF software quality badge
  - Lots of specific quality process requirements, many reflected in the new government requirements
- ISC software defect and security vulnerability disclosure
- ISC CVSS scoring guidelines

A lot of invisible work, but well-aligned with BCP
## BIND9 practices vs. survey

<table>
<thead>
<tr>
<th></th>
<th>Our priority</th>
<th>Survey priority</th>
</tr>
</thead>
<tbody>
<tr>
<td>CI &amp; automated tests</td>
<td>#1</td>
<td># 14</td>
</tr>
<tr>
<td>code reviews &amp; standards</td>
<td>#1</td>
<td># 13</td>
</tr>
<tr>
<td>Documentation</td>
<td>#?</td>
<td># 1</td>
</tr>
</tbody>
</table>
Conclusion

• Are users, or the experts, wrong?
• Are users assessing results, whereas experts are focused on processes?
• Is all the preoccupation with software security missing a more fundamental problem?
• Is this some kind of learned helplessness on the part of users, who may just be overwhelmed?
• Or ....
Thank you

https://www.isc.org
vicky@isc.org