OpenSSF
OPEN SOURCE SECURITY FOUNDATION

Implementing the OpenSSF Best Practices Badge & Scorecard into your project

OSS-NA 2023

OpenSSF
OPEN SOURCE SECURITY FOUNDATION

David A. Wheeler

Director of Open Source Supply Chain Security, The Linux Foundation, PhD IT, MS CS, BS EE, CISSP, IEEE SM

CRob, n, adj, and v

Pronunciation: U.S. (K-rowb)

42nd level Dungeon Master

25th level Securityologist

Pirate-enthusiast & hat-owner

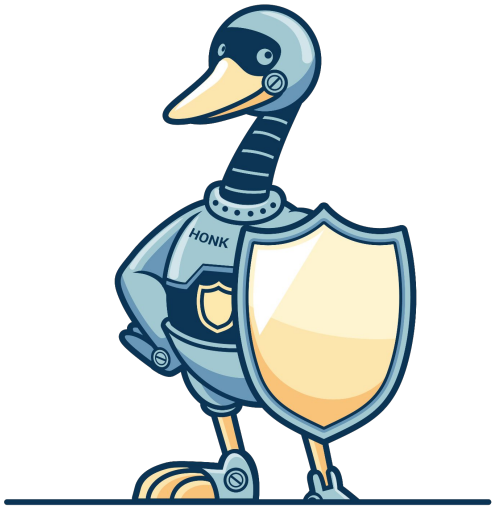
BA, MA, CISSP, CSSLP, ITIL, TOGAF

Open Source Security Foundation (OpenSSF)

Established by the Linux Foundation in 2020

A **global initiative** securing investment, resources, and expertise

“Committed to collaboration and working [with] communities to advance open source security for all.”

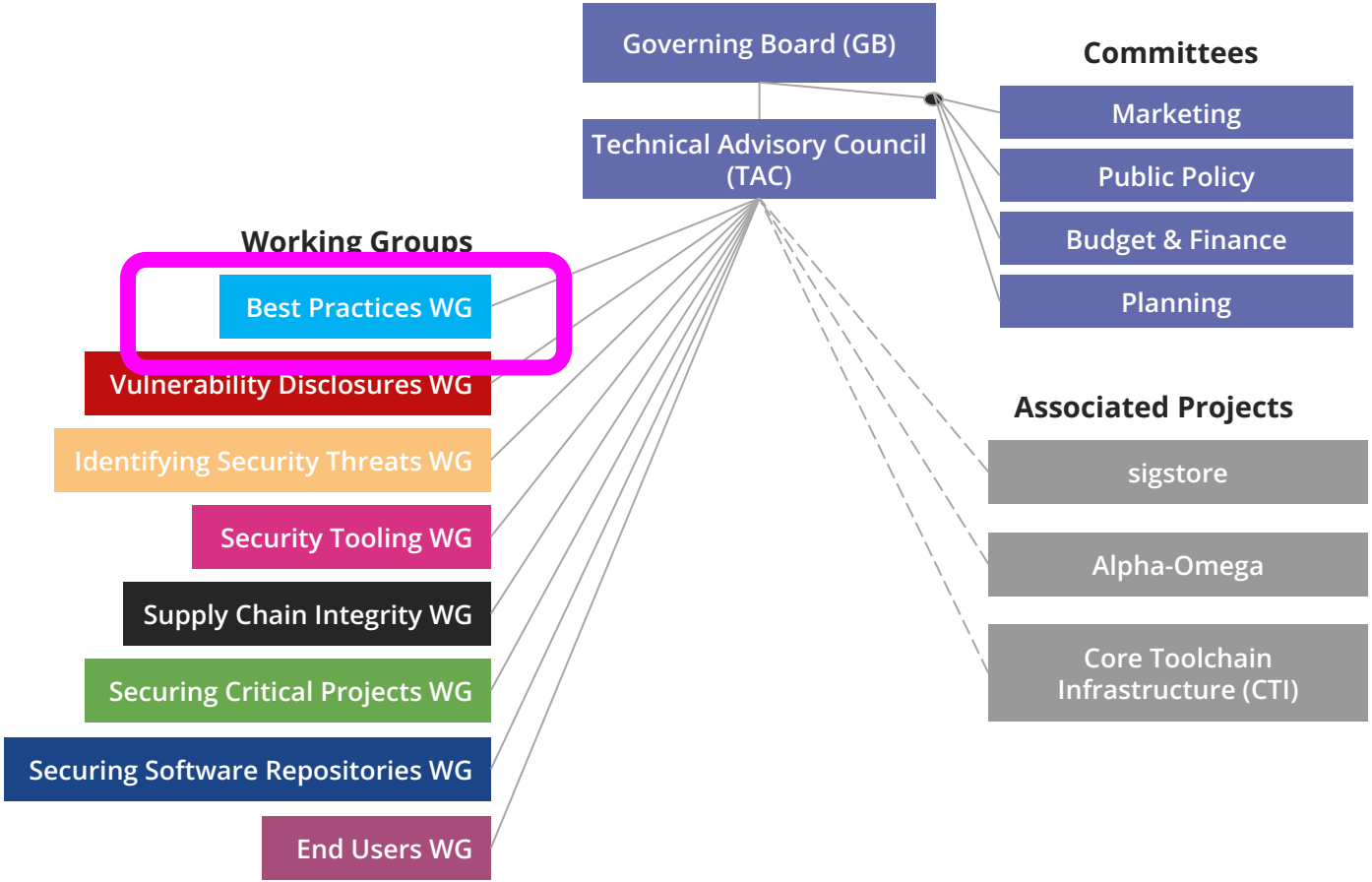


OpenSSF

OPEN SOURCE SECURITY FOUNDATION

OpenSSF
OPEN SOURCE SECURITY FOUNDATION

Open Source Security Foundation (OpenSSF)



OpenSSF
OPEN SOURCE SECURITY FOUNDATION

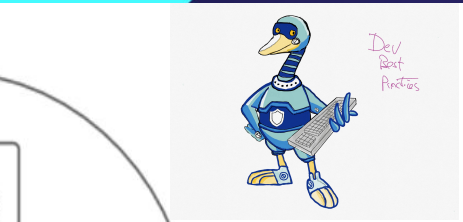
OpenSSF
OPEN SOURCE SECURITY FOUNDATION

OpenSSF Developer BEST Practices Working Group

Our objective is to provide open source developers with best practices recommendations, and with an easy way to learn and apply them.

Our vision is to make it easy for developers to adopt these best practices, thanks to:

- **Identifying** good practices, requirements, and tools that help open source developers create and maintain more secure software
- Helping maintainers **Learn** to write secure software
- Provide tools to help developers **Adopt** these good practices into their daily work



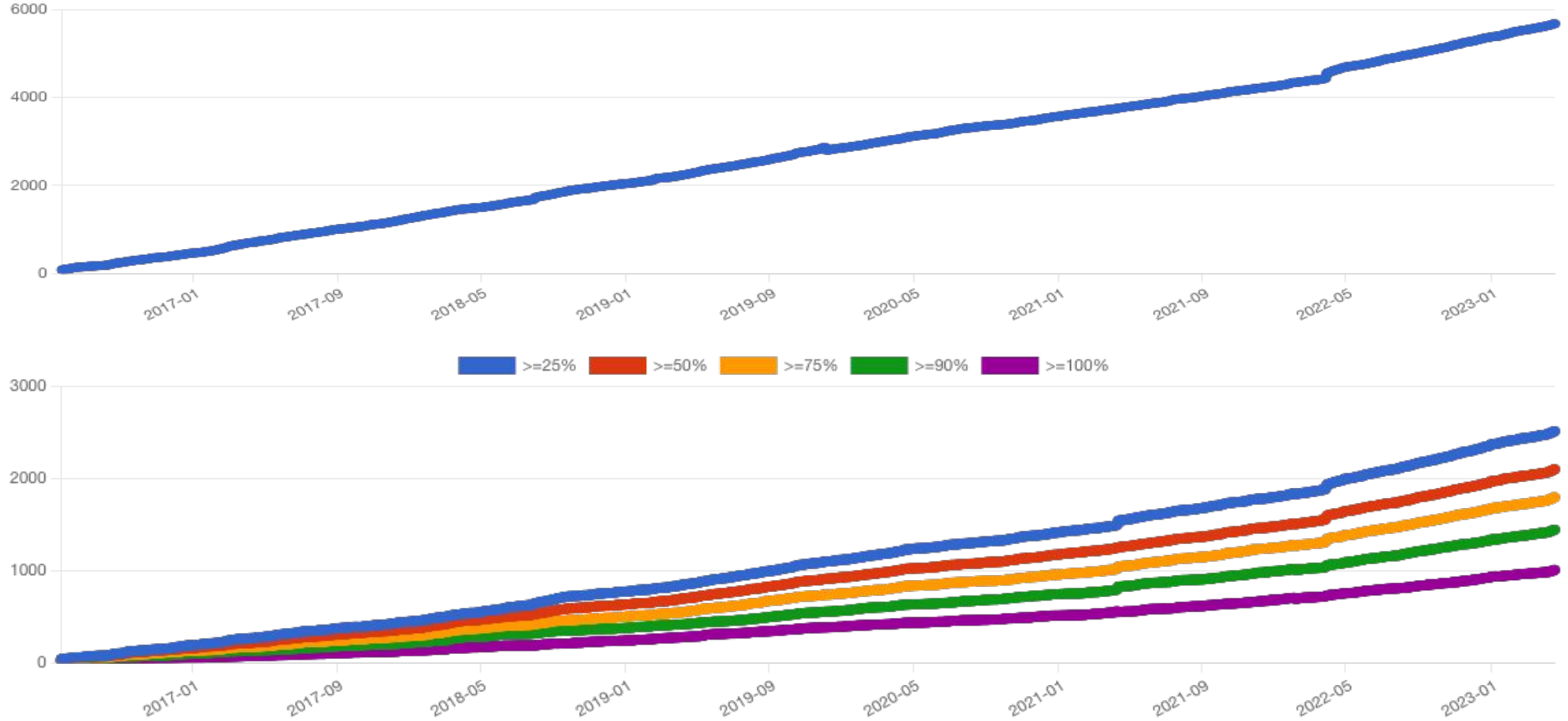
https://github.com/ossf/wg-best-practices-os-developers

OpenSSF
OPEN SOURCE SECURITY FOUNDATION

OpenSSF Best Practices Badge

- Identifies best practices for OSS projects
  - Based on practices of well-run OSS projects
- If OSS project meets certain criteria, it earns a best practices badge
  - Enables projects & potential users know current status & where it can improve
  - Combination of self-certification, automated checks, spot checks, public accountability - some automation, *not* full automation
  - Three badge levels: passing, silver, gold
  - Available in English, Chinese, French, German, Japanese, & Russian
- Participation widespread & continuing to grow
  - >5,600 participating projects, >1,000 passing+ (April 2023)

Growing Number of Participating & Passing Projects



OpenSSF
OPEN SOURCE SECURITY FOUNDATION

How did we judge the criteria to include?

- Must be Relevant, attainable, & clear
- Consensus of developers & users (worked with many OSS projects to test this)
- Does NOT require any specific technology, product, or service
- NEVER requires proprietary software or service (you *may* use/depend on it)
- Does NOT cost anything
- Does NOT “take over your project”
- Does NOT require doing everything immediately

WORDS ARE IMPORTANT

"MUST (NOT)", "SHOULD (NOT)", "SUGGESTED", and "MAY"

The key words "MUST", "MUST NOT", "SHOULD", "SHOULD NOT", and "MAY" are to be interpreted as described in RFC 2119. The additional term SUGGESTED is added. In summary, these key words have the following meanings:

- The term MUST is an absolute requirement, and MUST NOT is an absolute prohibition.
- The term SHOULD indicates a criterion that is normally required, but there may exist valid reasons in particular circumstances to ignore it. However, the full implications must be understood and carefully weighed before choosing a different course. The valid reason be explained.
- The term SUGGESTED is used instead of SHOULD when the criterion must be considered, but the valid reasons to not do so are even more common than for SHOULD.
- The term MAY provides one way something can be done, e.g., to make it clear that the described implementation is acceptable.

Often a criterion is stated as something that SHOULD be done, or is SUGGESTED, because **it may be difficult to implement** or **the costs to do so may be high**.

https://bestpractices.coreinfrastructure.org/en/criteria_discussion

What is the Best Practice Badge?

3 increasingly rigorous Levels to achieve:

- **Passing** - focuses on best practices that well-run FLOSS projects typically already follow (“basics’). This is an achievement (often 1 missing); currently ~18% of projects pursuing a badge achieve this.
- **Silver** - more stringent set of criteria than passing but is expected to be achievable by small and single-organization projects.
- **Gold** - even more stringent than silver and includes criteria that are not achievable by small or single-organization projects.

Image

Image

Image

Passing Criteria: Categories & Sample Criteria

Basics	• The software MUST be released as FLOSS [floss_license] • The project website MUST succinctly describe what the software does… [description_good]
Change Control	• The project MUST have a version-controlled source repository that is publicly readable and has a URL. [repo_public]
Reporting	• The project MUST publish the process for reporting vulnerabilities on the project site. [vulnerability_report_process]
Quality	• The project MUST use at least one automated test suite that is publicly released as FLOSS (this test suite may be maintained as a separate FLOSS project). [test] • The project MUST have a general policy … that as major new functionality is added to the software produced by the project, tests… should be added to an automated test suite. [test_policy]
Security	• At least one of the project's primary developers MUST know of common kinds of errors that lead to vulnerabilities in this kind of software, as well as at least one method to counter or mitigate each of them. [know_common_errors]
Analysis	• At least one static code analysis tool … MUST be applied to any proposed major production release of the software before its release, if there is at least one FLOSS tool that implements this criterion in the selected language. [static_analysis]

OpenSSF
OPEN SOURCE SECURITY FOUNDATION

Silver Criteria (Sample)

- The project MUST achieve a passing level badge. [achieve_passing]
- The project MUST be able to continue with minimal interruption if any one person dies, is incapacitated, or is otherwise unable or unwilling to continue support of the project. … This MAY be done by ensuring someone else has any necessary keys, passwords, and legal rights to continue the project. … [access_continuity]
- The project results MUST check all inputs from potentially untrusted sources to ensure they are valid (an *allowlist*), and reject invalid inputs, if there are any restrictions on the data at all. {N/A justification} [input_validation]

OpenSSF
OPEN SOURCE SECURITY FOUNDATION

Gold Criteria (Sample)

- The project MUST achieve a silver level badge. [achieve_silver]
- The project MUST have at least two unassociated significant contributors. [contributors_unassociated]
- The project MUST require two-factor authentication (2FA) for developers for changing a central repository or accessing sensitive data (such as private vulnerability reports). … [require_2FA]
- Hardening mechanisms MUST be used in the software produced by the project so that software defects are less likely to result in security vulnerabilities. [hardenin

Most OSS projects are single-maintainer projects
https://anchore.com/blog/open-source-is-bigger-than-you-imagine/ &
https://github.com/ossf/tac/issues/101

Testimonials

OWASP ZAP (web app scanner)
- Simon Bennetts: “[it] helped us improve ZAP quality… [it] helped us focus on [areas] that needed most improvement.”
- Change: Significantly improved automated testing

CommonMark (Markdown in PHP) changes:
- TLS for the website (& links from repository to it)
- Publishing the process for reporting vulnerabilities

JSON for Modern C++
- “I really appreciate some formalized [QA] which even hobby projects can follow.”
- Change: Added explicit mention of how to privately report errors
- Change: Added a static analysis check to continuous integration script

Develop OSS? First, Start Getting Your OpenSSF Best Practices Badge Now!

- Start here: https://bestpractices.coreinfrastructure.org
- Click on “Get your badge”, enter repo & home page URL
- System will analyze & auto-fill data where it can
- Focus on *passing* first; don’t worry about silver/gold yet
- Questions? Click on “details” of a criterion to learn more
- Expect to do things incrementally - “Unmet” @ start ok

Tips for Getting a Best Practices Badge (1)

- “The project website MUST succinctly describe what the software does…” [description_good]
  - Does your README explain, early, what problem it solves? Limit jargon!
- “The project MUST publish the process for reporting vulnerabilities on the project site.” [vulnerability_report_process]
  - Decide on email address(es) OR enable GitHub private reporting
  - Create SECURITY.md with this info, make README point to SECURITY.md
- “At least one of the project's primary developers MUST know of common kinds of errors that lead to vulnerabilities [and how] to counter… them.” [know_common_errors]
  - Take free OpenSSF course: https://openssf.org/training/courses/

OpenSSF
OPEN SOURCE SECURITY FOUNDATION

Tips for Getting a Best Practices Badge (2)

- "The project MUST use at least one automated test suite…" [test] & commit to improve
  - *Pick* a test framework, create tests, commit to improvement over time
  - Enable a CI/CD pipeline that runs those tests
- "The project MUST have a general policy … that as major new functionality is added … tests… should be added to an automated test suite". [test_policy]
  - Don't need perfect testing, just some testing & a public commitment to improvement - *ˆtrajectory* is what matters
- "At least one static code analysis tool … MUST be applied to any proposed major production release of the software before its release…" [static_analysis]
  - Pick a tool useful for your ecosystem. Many available; often you want to use many
  - If new (greenfield): Maximally enable all the warnings, so you know them early
  - If not new (brownfield): Start by enabling few warnings, then slowly add warnings
    Insert in CD/CI, focus on getting & staying warning-free (marking false+s as such)

OpenSSF
OPEN SOURCE SECURITY FOUNDATION

Future directions

- Move main site to bestpractices.dev
- Move repository under OpenSSF's repository
- More automation
- Always looking for improved criteria (at most annual update of real changes)
- Support for more human languages
- Various code cleanups

OpenSSF
OPEN SOURCE SECURITY FOUNDATION

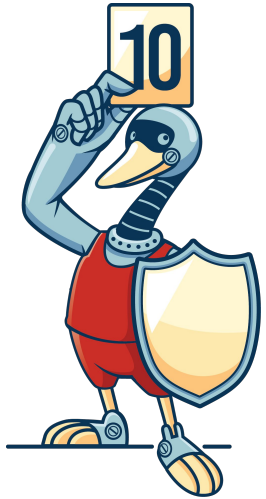
OpenSSF Scorecard

Scorecard takes a different approach, leaning into automation and externally-observable facts it can determine, as opposed to the interview-style approach the **Best Practices Badges** takes.

-

OpenSSF
OPEN SOURCE SECURITY FOUNDATION

OpenSSF Scorecard



- *Automatically* scores OSS projects on security heuristics ("checks")
  - Each related to security, scored 0-10, weighted average computed
  - Can use to evaluate your own or others' projects (they don't need to cooperate)
  - Currently only works on projects hosted on GitHub (not fundamental)
- Gives:
  - Aggregate (weighted) score
  - Scores for different areas
  - Ideas for improvement
- Newer features: scorecards badge, REST API
- Officially it's ("OpenSSF Scorecard") but we often use the plural
- https://github.com/ossf/scorecard

OpenSSF
OPEN SOURCE SECURITY FOUNDATION

Scorecard Checks (1 of 2)

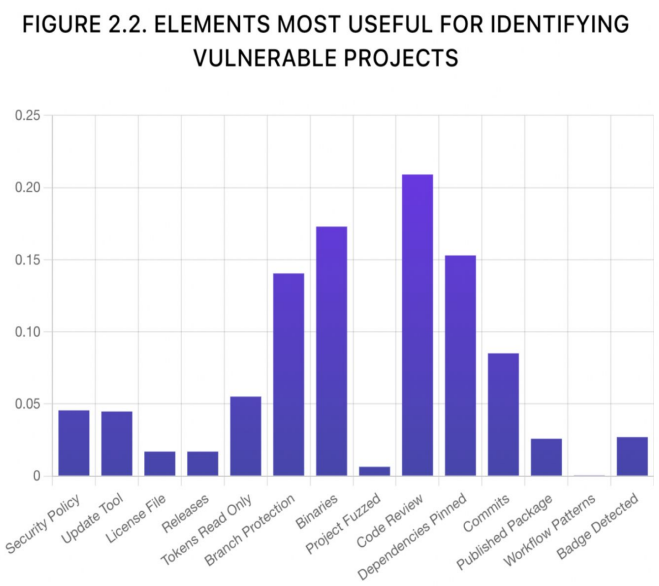
- Source risk assessment
  - Binary-Artifacts: Is the project free of checked-in binaries?
  - Branch-Protection: Does the project use Branch Protection ?
  - Code-Review: Does the project practice code review before code is merged?
  - Contributors: Does the project have contributors from at least two different organizations?
  - Dangerous-Workflow: Does the project avoid dangerous coding patterns in … workflows?
- Maintenance
  - Dependency-Update-Tool: Does the project use tools to help update its dependencies?
  - License: Does the project declare a license?
  - Maintained: Is the project at least 90 days old, and maintained?
  - Security-Policy: Does the project contain a security policy?
- Continuous Verification (including Testing)
  - CI-Tests: Does the project run tests in CI, e.g. GitHub Actions, Prow?
  - Fuzzing: Does the project use fuzzing tools, e.g. OSS-Fuzz?
  - SAST: Does the project use static code analysis tools, e.g. CodeQL, SonarCloud?

OpenSSF
OPEN SOURCE SECURITY FOUNDATION

Scorecard Checks (2 of 2)

- Build Risk Assessment
  - Pinned-Dependencies: Does the project declare and pin dependencies?
  - Packaging: Does the project build and publish official packages from CI/CD, e.g. GitHub Publishing ?
  - Signed-Releases: Does the project cryptographically sign releases?
  - Token-Permissions: Does the project declare GitHub workflow tokens as read only?
  - Webhooks: Does the webhook defined in the repository have a token configured to authenticate the origins of requests? (Experimental)
- Code Vulnerabilities
  - Vulnerabilities: Does the project have unfixed vulnerabilities? Uses the **OSV** service.
- Other risk analysis
  - CII-Best-Practices: Does the project have an OpenSSF (formerly CII) **Best Practices Badge**? (Expected rename to OpenSSF-Best-Practices)

Sonatype Analysis of Scorecard Heuristics



Sonatype's eighth annual "***State of the Software Supply Chain Report***" examined Scorecard

Found some Scorecard heuristics especially helpful at predicting projects with vulnerabilities:

1. Code Review (lack of) - peer review is a *powerful* against vulnerabilities
2. Checked in Binaries - an attack path, decrease transparency, and reduce auditability
3. Pinning dependencies - forces correct ones
4. Branch protection - enables review process

https://www.sonatype.com/state-of-the-software-supply-chain/project-quality-metrics
https://openssf.org/blog/2022/10/20/report-finds-openssf-scorecards-are-highly-effective-measures-to-assess-project-security/

OpenSSF
OPEN SOURCE SECURITY FOUNDATION

Getting Scorecard Results

- Run automatically on your project's code using a GitHub Action (if on GitHub)
  - Immediately updates to current results
  - Requires 1-time setup; instructions at https://securityscorecards.dev/
- Run manually via command line (works on any project Scorecard works on)
  - Measuring "webhooks" criterion requires special project privilege ("maintainer PAT")
- Scorecards API <https://api.securityscorecards.dev/> - REST API, ~1.2M OSS projects
  - Covers projects' GitHub Action results (if project shares) + weekly "Cron" results
  - Cron covers ~1M top OSS projects (criticality score) + Best Practices badge + special
  - Cron excludes 3 resource-hungry checks: CI-Tests, Contributors, and Dependency-Update-Tool
    - Trying to re-enable Dependency-Update-Tool (issue #2845)
- BigQuery public dataset (if you want all that data for wide analysis)
- deps.dev - includes Scorecards results <https://deps.dev/>, starts from *package* ID

OpenSSF
OPEN SOURCE SECURITY FOUNDATION

Running an OSS project? Integrate Scorecard!

- If on GitHub, add GitHub Action to run Scorecard:
  - 1-time setup; instructions at https://securityscorecards.dev/
  - Be sure to let it share its results with “cron”
- Modify your README to link to the Scorecard results, e.g., in markdown:
  - [![openssf scorecards] (https://api.securityscorecards.dev/projects/github.com/ORG/REPO/badge)] (https://api.securityscorecards.dev/projects/github.com/ORG/REPO)
- Check out your results - fix what problems it finds
  - If Scorecard doesn’t detect something you’re doing, consider filing an issue so we can improve its automation
  - What Scorecard is trying to detect is sometimes *hard* to do automatically!
    - In particular, there are *many* ways to do something

OpenSSF
OPEN SOURCE SECURITY FOUNDATION

Upcoming Scorecard work

- Adding GitLab support (many projects use GitLab) - Lockheed working on this
- Improving automation
  - Better CI pipeline support (currently GitHub Actions; need CircleCI, Jenkins, etc.)
  - Better tool detection (currently many static/dynamic tools not detected)
  - Our thanks to Amazon for funding much of this work
- Potential new metrics, e.g., mean time to update (MTTU) dependencies
  - Suggested by Sonatype analysis
- Cleanups

OpenSSF
OPEN SOURCE SECURITY FOUNDATION

Badges (BP Badge & Scorecard) have an Impact

- Per Master's thesis paper / survey released 2023-05-10…
- "Consensus [across] Badging Consumers is that an **OSS badge** provides a **positive** or negative **indication** of the **health** of an OSS project [and] most focused on the positive"
- "Badging Contributors also mentioned that an OSS badge can be used to **improve the health** of an OSS project by suggesting updates… OSS project maintainers seem just as interested…"
- "Badging Consumers expressed significant interest in the ways an OSS badge could **affect** an **OSS project's potential for success** [and] often focused on how a badge can change perceptions of an OSS project."

Source: *Open Source Software Badging: The Proof is in the Project*, Matt Snell, 2023-05-10, University of Nebraska

Considering using OSS (as a Dependency)? Look at BP Badge & Scorecard

- More generally, see OpenSSF "***Concise Guide for Evaluating Open Source Software***"
- Make sure you're evaluating the right thing (counter typosquatting)
- **Best Practices Badge** - does it have a badge, or at least working on one?
  - If it is, that's a great sign!
  - However, there are millions of OSS & thousands of badges, so that's rarified
- **Scorecard** - what are its results?
  - Scorecards results are available for >1M OSS projects, providing some information
  - All tools, including Scorecard, have false+ and false- … double-check if important
- Future: Working on OSS dashboard to include BP Badge, Scorecard, & other data

We also suggest checking out our Concise Guide to Evaluating Open Source Software

Scorecard & BP badge work well together

Scorecard	BP Badge
• Project participation unneeded • Quick auto results on most OSS projects	• Requires project participation • Requires human ~20 minutes/project

• Like all tools, false +s and false –s
  • Many tools & CI systems not handled
  • File presence ≠ Doing something
• Only GitHub today (GItLab coming)
• Can only report on what's automatable
	• Human analysis counters false results
	  • Handles any tool & CI system
	  • Not fooled by file presence
	• Works with any forge (hosting site)
	• Can report on what's important

One of the Scorecard heuristics is "working on/achieved a BP badge"

They work together, like chocolate & peanut butter (yum!)

Image by Stacy

Thank You!

dwheeler @ linuxfoundation . org (preferred)

@drdavidawheeler

@davidawheeler@infosec.exchange

https://github.com/david-a-wheeler

https://www.youtube.com/c/DavidAWheeler

https://www.linkedin.com/in/david-a-wheeler-27798688/

CRob_at_Intel_dot_com

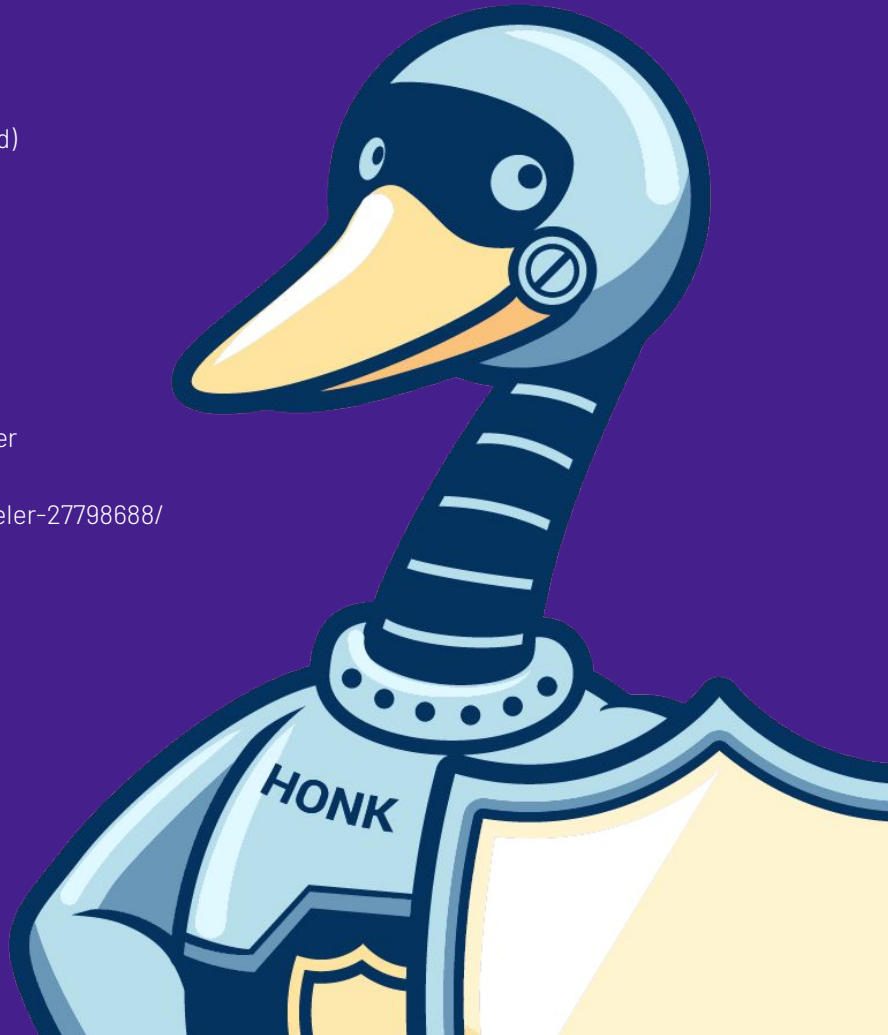
@SecurityCRob

@SecurityCRob@infosec.exchange

https://github.com/SecurityCRob

The Security Unhappy Hour,
Chips & Salsa

https://www.linkedin.com/in/darthcrob/



This presentation released under the CC-BY-4.0 license