Scorecard Tech Talk

Building a Stronger Open Source Ecosystem: OpenSSF Scorecard

March 13th, 10AM PT/1PM ET

OpenSSF

OPEN SOURCE SECURITY FOUNDATION

Welcome!

- Thank you for joining us today! We will begin at 10:02am PT.
- While we wait for everyone to join, please take a moment to do one (or more) of the following:
  - Please add questions using the Zoom Q&A feature
  - Follow us on Twitter: @openssf, Mastodon: social.lfx.dev/@openssf, & LinkedIn: OpenSSF
  - Visit our website: https://openssf.org
  - Sign up for training: https://openssf.org/training/courses/
- This Tech Talk is being recorded

OpenSSF
OPEN SOURCE SECURITY FOUNDATION

Agenda

- Housekeeping
- Panelist Introductions
- Introduction to OpenSSF Scorecard
- Understanding the OpenSSF Scorecard
- Insights from Implementing Organizations
- Panel Discussion: Member Organizations’ Experiences
- Q&A from the Audience

OpenSSF
OPEN SOURCE SECURITY FOUNDATION

Antitrust Policy Notice

Linux Foundation meetings **involve participation by industry competitors**, and it is the intention of the Linux Foundation to conduct all of its activities in accordance with applicable antitrust and competition laws. It is therefore extremely important that attendees adhere to meeting agendas, and be aware of, and not participate in, any activities that are prohibited under applicable US state, federal or foreign antitrust and competition laws.

Examples of types of actions that are prohibited at Linux Foundation meetings and in connection with Linux Foundation activities are described in the Linux Foundation Antitrust Policy available at **http://www.linuxfoundation.org/antitrust-policy**. If you have questions about these matters, please contact your company counsel, or if you are a member of the Linux Foundation, feel free to contact Andrew Updegrove of the firm of Gesmer Updegrove LLP, which provides legal counsel to the Linux Foundation.

Code of Conduct

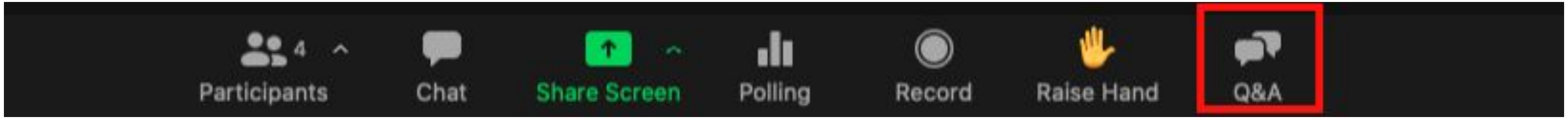
- The Linux Foundation and its project communities are **dedicated to providing a harassment-free experience** for participants at all of our events, whether they are held in person or virtually.

- All event participants, whether they are attending an in-person event or a virtual event, **are expected to behave in accordance with professional standards**, with both this Code of Conduct as well as their respective employer's policies governing appropriate workplace behavior and applicable laws.

- https://openssf.org/community/code-of-conduct/

Housekeeping

Please submit your questions during the meeting by using the Q&A feature on Zoom.



Thank you!

Panelists

Caroline Lee, Security Engineer, IBM

Caroline is based out of Boston, Massachusetts, and works as a Security Engineer at IBM in CISO Remediation. She holds a Masters in Computer Science with a Specialization in Cybersecurity. Previously, she has worked on CICD, Application Security, and Cloud Security initiatives in the government sector. She is currently involved in projects in Application Security, DevSecOps, and more.

OpenSSF
OPEN SOURCE SECURITY FOUNDATION

OpenSSF
OPEN SOURCE SECURITY FOUNDATION

Chris Swan - Engineer, Atsign

Chris Swan is an Engineer at Atsign, building the atPlatform, a Networking 2.0 technology that is putting people in control of their data and removing the frictions and surveillance associated with today’s Internet. He was previously a Fellow at DXC Technology where he held various CTO roles. Before that he held CTO and Director of R&D roles at Cohesive Networks, UBS, Capital SCF and Credit Suisse, where he worked on app servers, compute grids, security, mobile, cloud, networking and containers. Chris is an InfoQ Editor writing about cloud, DevOps and security, he co-hosts the Tech Debt Burndown Podcast and is a Dart Google Developer Expert (GDE).

OpenSSF
OPEN SOURCE SECURITY FOUNDATION

Melba Lopez - Senior Technical Staff Member (STSM), IBM

Melba Lopez is a seasoned cybersecurity professional currently serving as a Senior Technical Staff Member (STSM) at the IBM Office of CISO. With a primary focus on the strategy and delivery of enterprise software supply chain security, Melba plays a pivotal role in safeguarding critical assets against emerging supply chain threats.

In addition to her role at IBM, Melba is deeply involved in industry initiatives aimed at fortifying software supply chains. She serves as an OWASP Dependency Track maintainer, demonstrating her commitment to advancing open-source security solutions. Previously, she held leadership positions within the Open Source Security Foundation (OpenSSF), including co-lead of the Supply Chain Integrity Working Group and Lead of the Positioning Special Interest Group.

Melba’s expertise spans over 18 years, covering a diverse range of domains such as application development, cloud computing, networking, and security. Her multidisciplinary background equips her with a comprehensive understanding of the intricate landscape of cybersecurity challenges. With a Master’s degree in Cybersecurity & Information Assurance, Melba is passionate about leveraging her knowledge and experience to drive impactful changes in the cybersecurity ecosystem.

Laurent Simon - Security Engineer, Google

Laurent is a security engineer in the Open Source Security Team (GOSST) at Google. His team works in collaboration with the open-source community and the OpenSSF on novel security solutions, such as Scorecards, Allstar, Sigstore, SLSA, OSS-Fuzz, OSV, etc.

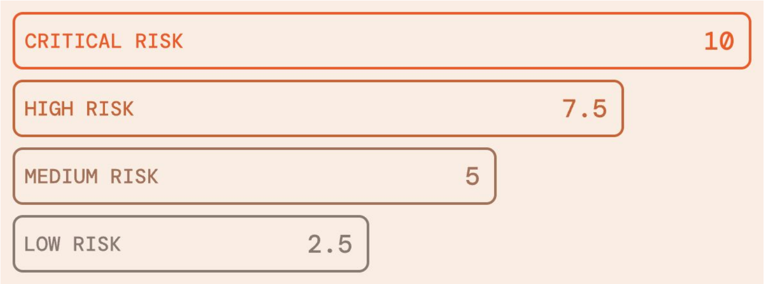
OpenSSF
OPEN SOURCE SECURITY FOUNDATION

Introduction to OpenSSF Scorecard

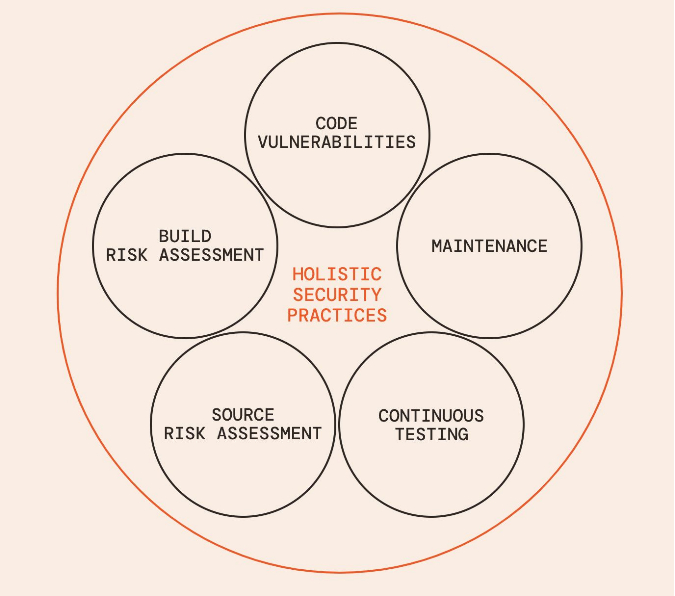
Caroline Lee

Introduction to OpenSSF Scorecard

- 18 checks affecting different aspect of the software supply chain (shown right)
- Each automated check returns a score out of 10 and a risk level
- The risk level adds weighting to the score (shown below)
- The weighted value of all checks are compiled into a single, aggregate score



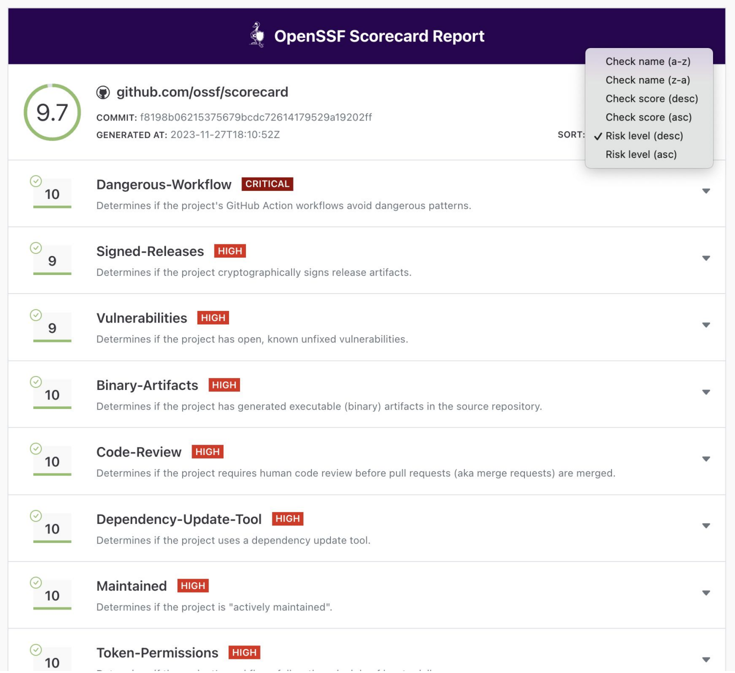
CRITICAL RISK	10
HIGH RISK	7.5
MEDIUM RISK	5
LOW RISK	2.5



https://github.com/ossf/scorecard#scorecard-checks

Introduction to OpenSSF Scorecard

- Running the Checks
  - Run automatically using the GitHub Action
  - Run manually via the Command Line Interface
- Viewing Scores
  - Use the Webviewer to see score reports for regularly scanned projects (shown right)
  - Use the REST API to query pre-calculated scores of OSS projects
  - Weekly scan of *1.2M+* critical OSS projects with results published in a BigQuery public dataset
- Details at https://github.com/ossf/scorecard



OpenSSF Scorecard Report

9.7

github.com/ossf/scorecard

COMMIT: f8198b06215375679bcdc72614179529a19202ff

GENERATED AT: 2023-11-27T18:10:52Z

SORT: Check name (a-z) / Check name (z-a) / Check score (desc) / Check score (asc) / ✓ Risk level (desc) / Risk level (asc)

10 Dangerous-Workflow CRITICAL
Determines if the project's GitHub Action workflows avoid dangerous patterns.

9 Signed-Releases HIGH
Determines if the project cryptographically signs release artifacts.

9 Vulnerabilities HIGH
Determines if the project has open, known unfixed vulnerabilities.

10 Binary-Artifacts HIGH
Determines if the project has generated executable (binary) artifacts in the source repository.

10 Code-Review HIGH
Determines if the project requires human code review before pull requests (aka merge requests) are merged.

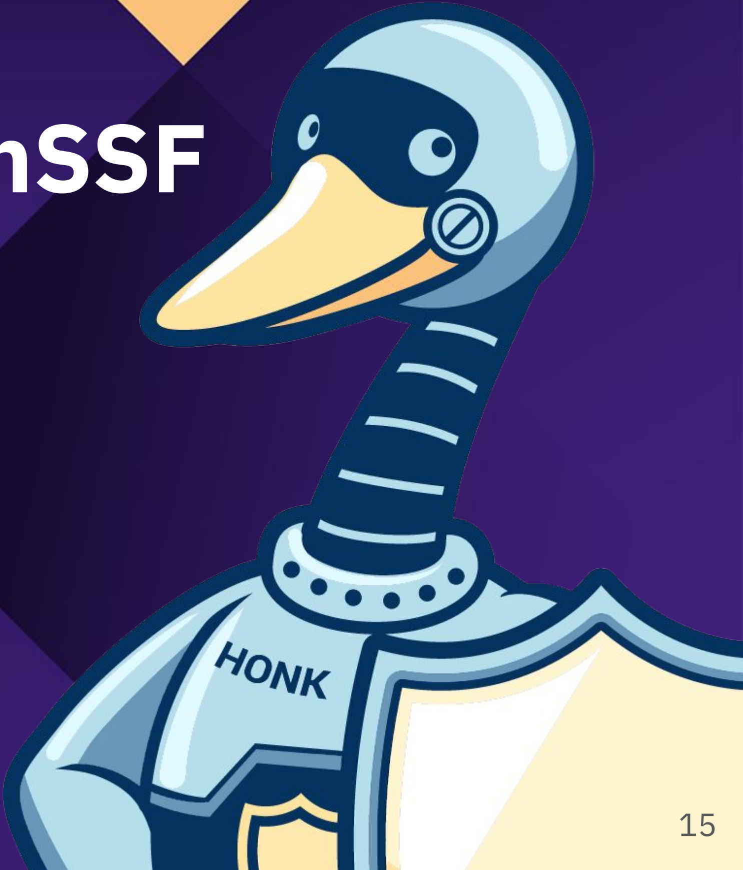
10 Dependency-Update-Tool HIGH
Determines if the project uses a dependency update tool.

10 Maintained HIGH
Determines if the project is "actively maintained".

10 Token-Permissions HIGH

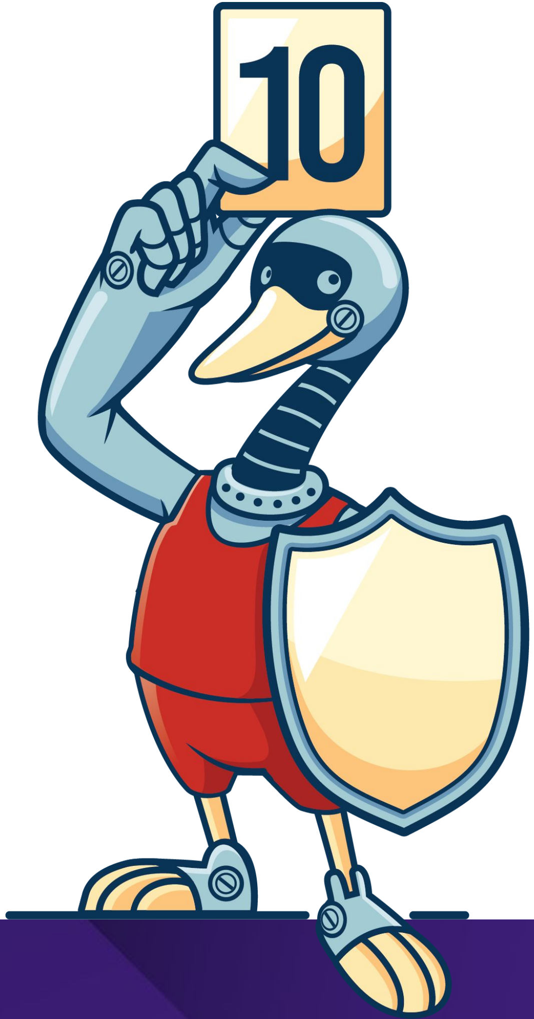
Understanding the OpenSSF Scorecard

Laurent Simon



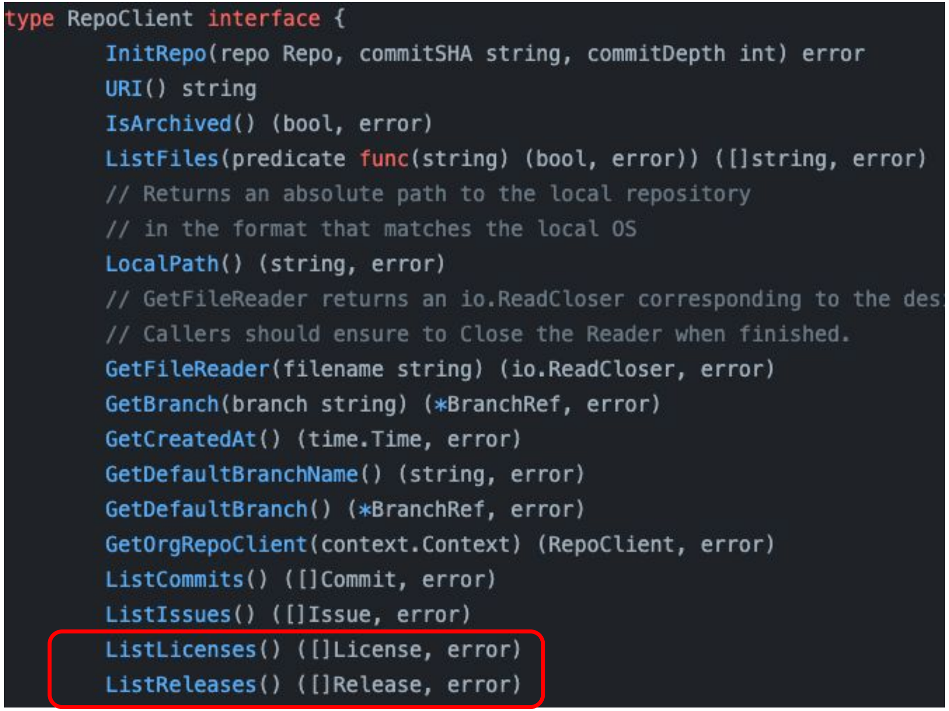
Main components of the codebase

- Repository client
- Checks
  - Raw part layer
  - Probe layer
  - Evaluation layer
- Scorecard run API



Repository client

- Golang interface defining a CVS
- Abstracts away details of each CVS (GitHub, GitLab, etc)



```go
type RepoClient interface {
        InitRepo(repo Repo, commitSHA string, commitDepth int) error
        URI() string
        IsArchived() (bool, error)
        ListFiles(predicate func(string) (bool, error)) ([]string, error)
        // Returns an absolute path to the local repository
        // in the format that matches the local OS
        LocalPath() (string, error)
        // GetFileReader returns an io.ReadCloser corresponding to the des
        // Callers should ensure to Close the Reader when finished.
        GetFileReader(filename string) (io.ReadCloser, error)
        GetBranch(branch string) (*BranchRef, error)
        GetCreatedAt() (time.Time, error)
        GetDefaultBranchName() (string, error)
        GetDefaultBranch() (*BranchRef, error)
        GetOrgRepoClient(context.Context) (RepoClient, error)
        ListCommits() ([]Commit, error)
        ListIssues() ([]Issue, error)
        ListLicenses() ([]License, error)
        ListReleases() ([]Release, error)
```

https://github.com/ossf/scorecard/blob/main/clients/repo_client.go

Repository client implementations

- Uses platform-specific APIs
- GitHub
  - https://github.com/ossf/scorecard/tree/main/clients/githubrepo
  - GraphQL APIs https://docs.github.com/en/graphql
  - RESTful APIs https://docs.github.com/en/rest
- GitLab
  - https://github.com/ossf/scorecard/tree/main/clients/gitlabrepo

Checks[1] (1)

- Three sub components
  - Raw result layer
  - Probe layer
  - Evaluation layer
- Raw result layer[2]
  - Uses the **repository client** to **gather** all information required by a check and **caches** it, like a snapshot
  - Branch protection check: settings, CODEOWNER file, etc

[1] https://github.com/ossf/scorecard/tree/main/checks

[2] https://github.com/ossf/scorecard/blob/main/checker/raw_result.go

Checks[1] (2)

- Probes (release soon-ish!)
  - Takes as input the raw result cache data
  - Output a low-level **claim** about a project. Will enable **fine-grained policies**[2] on results!
  - A check is made up of multiple probes, each with a description and a clear remediation
  - Example: **requiresCodeOwnersReview**[3] probe determines if code owners reviews are required, based on settings and presence of Codeowner file (raw result cache)
- Evaluation
  - Reads a set of probe results and computes a score[4]

[1] https://github.com/ossf/scorecard/tree/main/checks

[2] https://events.linuxfoundation.org/open-source-summit-north-america/program/schedule/

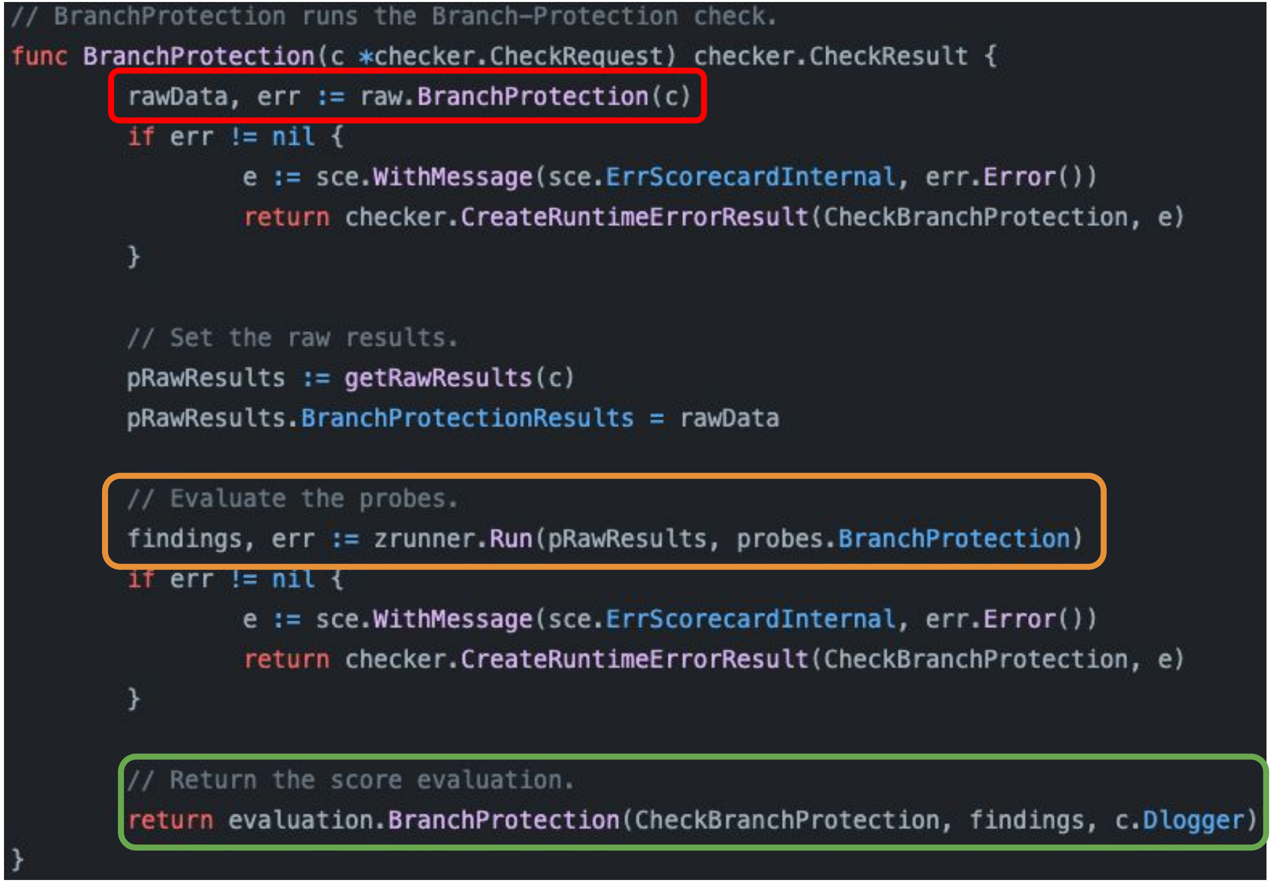
[3] https://github.com/ossf/scorecard/tree/main/probes/requiresCodeOwnersReview

[4] https://github.com/ossf/scorecard/blob/main/checks/evaluation/branch_protection.go

Raw results

Probes

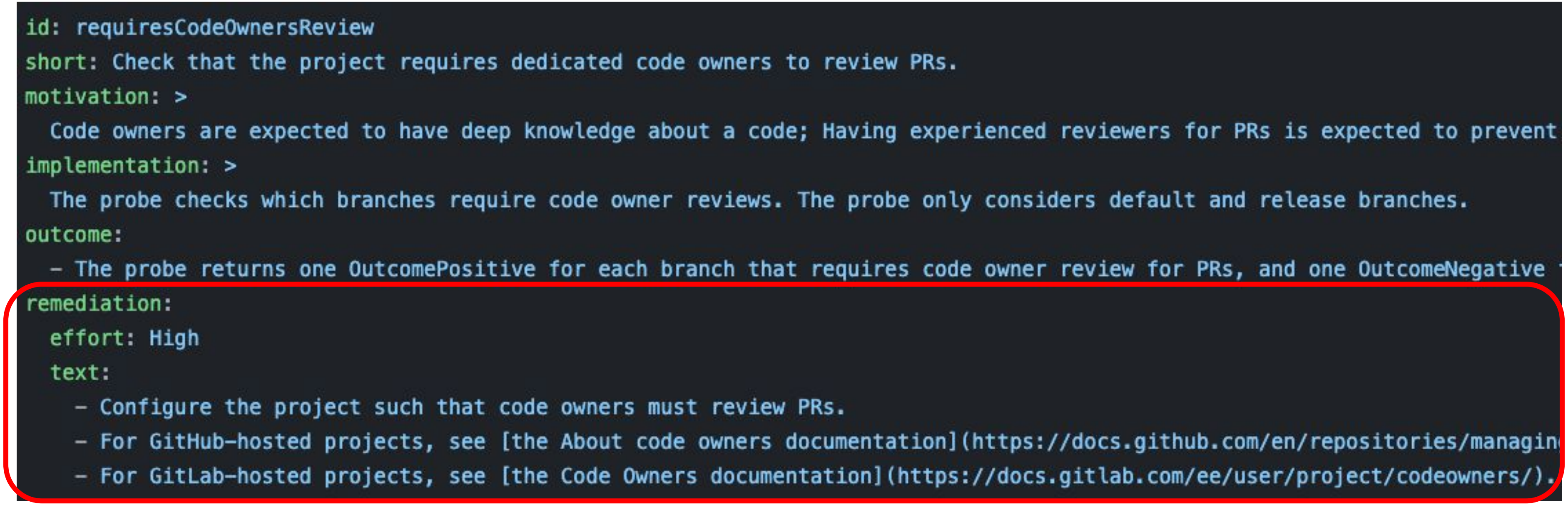
Scoring



```go
// BranchProtection runs the Branch-Protection check.
func BranchProtection(c *checker.CheckRequest) checker.CheckResult {
	rawData, err := raw.BranchProtection(c)
	if err != nil {
		e := sce.WithMessage(sce.ErrScorecardInternal, err.Error())
		return checker.CreateRuntimeErrorResult(CheckBranchProtection, e)
	}

	// Set the raw results.
	pRawResults := getRawResults(c)
	pRawResults.BranchProtectionResults = rawData

	// Evaluate the probes.
	findings, err := zrunner.Run(pRawResults, probes.BranchProtection)
	if err != nil {
		e := sce.WithMessage(sce.ErrScorecardInternal, err.Error())
		return checker.CreateRuntimeErrorResult(CheckBranchProtection, e)
	}

	// Return the score evaluation.
	return evaluation.BranchProtection(CheckBranchProtection, findings, c.Dlogger)
}
```

https://github.com/ossf/scorecard/blob/main/checks/branch_protection.go

requiresCodeOwnersReview probe



```
id: requiresCodeOwnersReview
short: Check that the project requires dedicated code owners to review PRs.
motivation: >
  Code owners are expected to have deep knowledge about a code; Having experienced reviewers for PRs is expected to prevent
implementation: >
  The probe checks which branches require code owner reviews. The probe only considers default and release branches.
outcome:
  - The probe returns one OutcomePositive for each branch that requires code owner review for PRs, and one OutcomeNegative
remediation:
  effort: High
  text:
    - Configure the project such that code owners must review PRs.
    - For GitHub-hosted projects, see [the About code owners documentation](https://docs.github.com/en/repositories/managin
    - For GitLab-hosted projects, see [the Code Owners documentation](https://docs.gitlab.com/ee/user/project/codeowners/).
```

https://github.com/ossf/scorecard/blob/main/probes/requiresCodeOwnersReview

Run API

- https://github.com/ossf/scorecard/blob/main/pkg/scorecard.go
- RunScorecard() Golang API
- Launched each check in a Go routine

Collaborative aspects

- Weekly public meetings. ~100 contributors, ~5 triagers, ~3 active maintainers
  - Upcoming Contributor Workshop at OSS NA[5]
- Collaboration with GitHub and Google since the inception of the project
  - GitHub Action[1,2]
  - Private vulnerability reporting API[3]
- Probe implementation available soon-ish!
  - Several months of work with AdaLogics (vendor)[4]
  - Funded by AWS donation!
  - Feedback from IBM, nodejs security WG, Google, CNCF, etc
- GOSST Upstream Team
  - Open 455 PRs in 2023, with 320 merged, 41 rejected (others are still "in flight")

[1] https://github.blog/2022-01-19-reducing-security-risk-oss-actions-opensff-scorecards-v4/

[2] https://security.googleblog.com/2021/07/measuring-security-risks-in-open-source.html

[3] https://github.blog/changelog/2024-03-08-check-if-private-vulnerability-reporting-is-enabled-via-rest-api/

[4] https://events.linuxfoundation.org/open-source-summit-north-america/program/schedule/

[5] https://openssf.org/blog/2024/02/26/soss-community-day-north-america-na-agenda-live/

Recognition of scorecard in the community (1)

- Engagements with government
  - Google promotes Scorecard through consultations with White House and RFI[1,2]
  - The OpenSSF promotes scorecard to US Office of the National Cyber Director (ONCD)[3,4]
- Research
  - Sonatype shows Scorecard results correlate with project security[5]
  - NCSU using scorecard effectiveness and check relevance to ecosystems[6,7]

[1] https://blog.google/technology/safety-security/shared-success-in-building-a-safer-open-source-community/

[2] https://www.regulations.gov/comment/ONCD-2023-0002-0074

[3] https://www.whitehouse.gov/wp-content/uploads/2024/02/Final-ONCD-Technical-Report.pdf

[4] https://openssf.org/blog/2024/02/26/openssf-supports-efforts-to-build-more-secure-and-measurable-software/

[5] https://openssf.org/blog/2022/10/20/report-finds-openssf-scorecards-are-highly-effective-measures-to-assess-project-security

[6] https://arxiv.org/abs/2208.03412, [7] https://arxiv.org/abs/2210.14884

Recognition of scorecard in the community (2)

- Scorecard monitor[1]
  - Used by Node.js Security Working Group and CISCO OSPO team
  - Donation to Scorecard project under way[2]
- CNCF CLOMonitor[3,4]
  - Used by CNCF to monitor their projects
  - Used in several of their security slams[5,6,7]

[1] https://github.com/UlisesGascon/openssf-scorecard-monitor, [2] https://github.com/ossf/scorecard/issues/3204
[3] https://github.com/cncf/clomonitor, [4] https://clomonitor.io, [5] https://www.cncf.io/reports/security-slam-2023/
[6] https://www.cncf.io/reports/lightning-round-at-security-slam-2023/, [7] https://www.cncf.io/reports/security-slam-north-america-2022/

Insights from Implementing Organizations

Chris Swan, Melba Lopez

Implementing Scorecard - Atsign

- An OpenSSF Scorecard can show you care about security.
- Allstar provides a good starting point.
- Pick a first repo to get a sense of what’s needed.
- Then automate across the rest of the organisation.
- 20% of the effort to get 80% of the score. Uphill from there.
- Scorecards create ongoing toil that needs to be minimised.

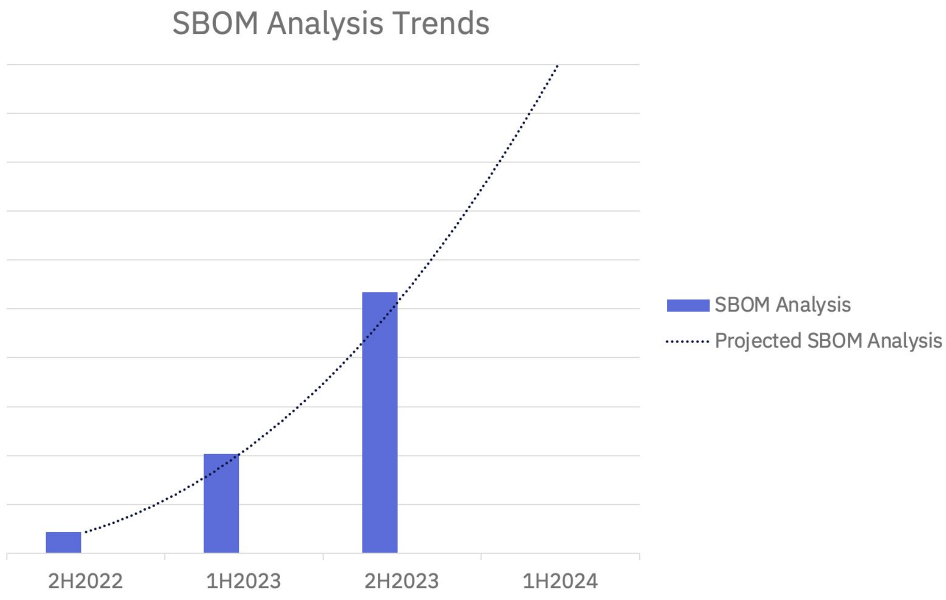
Implementing Scorecard – IBM CISO

Business Value:

- 2,327% increase YoY in SBOM assessments
- Increased awareness of 3rd party open-source hygiene

Key Areas:

- Third Party Security Risk Management (Suppliers, Vendors, OEMs)
- Mergers & Acquisitions
- IBM Product Security
- Open Source Security

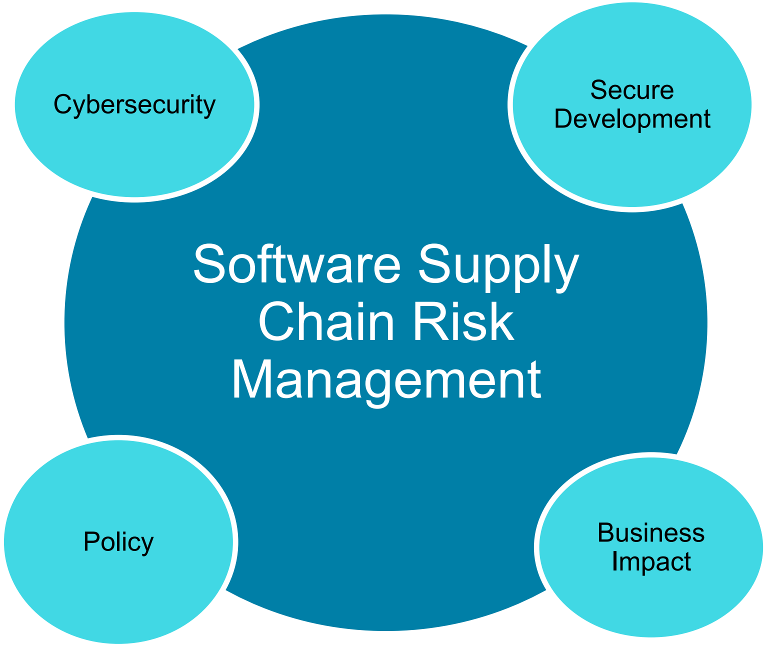


Implementing Scorecard – IBM CISO

- OpenSSF Scorecard is a valuable tool in your Supply Chain Security Toolkit

- OpenSSF Scorecard can be used to assess risk beyond vulnerabilities
  - Community Health
  - Secure Development
  - Legal Risk

- OpenSSF Scorecard can be used for internal GitHub Enterprise



Panel Discussion: Member Organizations' Experiences

Experiences, challenges, and successes in adopting OpenSSF Scorecard

Audience Q&A

OpenSSF
OPEN SOURCE SECURITY FOUNDATION

Take our quick Tech Talk Survey

https://forms.gle/bzyoowDMAHbN4rAZ8

OpenSSF Scorecard User Survey

https://forms.gle/aqxZwmVQzWJkNuso8

Securing Projects with OpenSSF Scorecard Course

This FREE course: Securing Projects with OpenSSF Scorecard (LFEL1006) is available on the Linux Foundation Training & Certification platform and is designed with end users of Scorecard tooling in mind.

This course will cover how to integrate the OpenSSF Scorecard into your software development life cycle.

EXPRESS LEARNING COURSE

Securing Projects with OpenSSF Scorecard (LFEL1006)

Integrate the OpenSSF Scorecard into your software development life cycle.

OpenSSF
OPEN SOURCE SECURITY FOUNDATION

Upcoming Events

LF Open Source Summit North America 2024

When: April 16-18, 2024

Where: Seattle, Washington, USA

- April 15: OpenSSF Scorecard New Contributor Workshop [Pre-Registration Required]

- April 17: Structured Scorecard Results: Tailor Your Own Supply-Chain Security Policies - Adam Korczynski & David Korczynski, Ada Logics

OpenSSF
OPEN SOURCE SECURITY FOUNDATION

Ways to Participate

- Join a Working Group/Project
- Come to a Meeting (see Public Calendar)
- Collaborate on Slack
- Contribute on GitHub
- Become an Organizational Member
- Keep up to date by subscribing to the OpenSSF Mailing List

OpenSSF
OPEN SOURCE SECURITY FOUNDATION

Engage with us on social media

X
@openssf

LinkedIn
OpenSSF

Mastodon
social.lfx.dev/@openssf

YouTube
OpenSSF

Facebook
OpenSSF

OpenSSF
OPEN SOURCE SECURITY FOUNDATION

Subscribe to our mailing list

openssf.org/sign-up

Is your organization a member?

Questions? Contact membership@openssf.org

openssf.org/join

Thank You

Legal Notice