Supply Chain Codebook [anonymized]

V0.4

Changelog

V0.4 - Merged all iterations again for replication package
V0.3 - Add coder table
V0.2 - Split into iterations
V0.1 - Initial Draft

Coder Table

[anonymized]

All Iterations

C1 Project Demographics

- **C1.1** Project
  - **C1.1.1** Project type
  - **C1.1.2** Project age
  - **C1.1.3** Team size
    - **C1.1.3-0** Not mentioned
    - **C1.1.3-1** 1
    - **C1.1.3-2** 2-5
    - **C1.1.3-3** 5+

        - **C1.1.4** Team structure
            - **C1.1.4-0** Not mentioned
            - **C1.1.4-1** Consultant / Freelancer (alone)
            - **C1.1.4-2** Single (developer) team
            - **C1.1.4-3** Multiple teams (including stuff like SRE, QA, etc.)
        - **C1.1.5** Number of Projects
            - **C1.1.5-0** Not mentioned
            - **C1.1.5-1** One
            - **C1.1.5-2** Multiple (>1)
- **C1.2** Project Relation
    - **C1.2.1** Participant joined the project
    - **C1.2.2** Participant role
- **C1.3** Project Setup
    - **C1.3.1** Project-specific tools
    - **C1.3.2** Project stages
    - **C1.3.3** Review frequency
    - **C1.3.4** Security roles.
        - **C1.3.4-0** Not mentioned
        - **C1.3.4-1** Yes
        - **C1.3.4-2** No
        - **C1.3.4-3** Other

C2 Usage OSCs

- **C2.1** OSC Components
    - **C2.1.1** OSCs included
        - **C2.1.1-0** No
        - **C2.1.1-1** Yes
    - **C2.1.2** Specific OSCs
    - **C2.1.3** Roles that interact with OSCs
- **C2.2** OSC Selection Metrics
    - **C2.2.1** Metrics for selecting OSCs [Select all that apply, feel free to extend]
        - **C2.2.1-0** None mentioned
        - **C2.2.1-1** Popularity (like Github stars or downloads)
        - **C2.2.1-2** Sponsorship (by trusted entity)
        - **C2.2.1-3** Activity (like commit frequency or releases)
        - **C2.2.1-4** Quality (e.g., commit quality)
        - **C2.2.1-5** Recommendations (by friends, blogs, communities, …)
        - **C2.2.1-6** License (must allow usage, etc.)
        - **C2.2.1-7** No fix rules (each developer doing as they think)
        - **C2.2.1-8** Features (needs to have the needed features)
        - **C2.2.1-9** Security history (e.g., past incidents or CVEs)
        - **C2.2.1-10** Ease of use (for developers, not including documentation)
        - **C2.2.1-11** Community (e.g., to be large, or active)
        - **C2.2.1-12** Minimize number of dependencies

- - - **C2.2.1-13** Dependencies predefined (e.g., customer requirements)
    - **C2.2.1-14** Code Inspection (by developer before use)
    - **C2.2.1-15** Maturity (of the whole project)
    - **C2.2.1-16** Documentation (easy to read/understand/apply, helpful, etc.)
  - **C2.2.2** Exclusion criteria for OSCs [Select all that apply, feel free to extend]
    - **C2.2.2-0** None mentioned
    - **C2.2.2-1** Previous vulnerability
    - **C2.2.2-2** Inactive maintainer / project
    - **C2.2.2-3** Avoid specific organizations (companies, vendors, etc.)
    - **C2.2.2-4** Minimum star limit
    - **C2.2.2-5** Company Policies (e.g., not to use any 3rd party code at all, license restrictions, etc.)
    - **C2.2.2-6** Obviously malicious code/vulnerabilities
    - **C2.2.2-7** Single/low number of contributors
    - **C2.2.2-8** Bad documentation
    - **C2.2.2-9** Bad code quality
  - **C2.2.3** Personal wishlist metrics
  - **C2.2.4** Awareness of existing metrics
    - **C2.2.4-0** No
    - **C2.2.4-1** OpenSSF Scorecards
    - **C2.2.4-2** socket.dev
    - **C2.2.4-3** Other
- **C2.3** How are OSCs pulled in?
  - **C2.3.1** How are OSCs pulled in
  - **C2.3.2** Process for including new OSCs
  - **C2.3.3** Using internal mirrors.
    - **C2.3.3-0** No
    - **C2.3.3-1** Yes
    - **C2.3.3-2** Other
- **C2.4** OSC in other infrastructure?
- **C2.5** Contribute back to OSS?
  - **C2.5-0** Did not contribute back
  - **C2.5-1** Did contribute back
  - **C2.5-2** Would like to contribute back

C3 Policies and Guidance

- **C3.1** What security policies?
  - **C3.1.1** Security policies for external code
    - **C3.1.1-0** No
    - **C3.1.1-1** Yes
    - **C3.1.1-2** Other
  - **C3.1.2** Content of security policies
  - **C3.1.3** Applicability / Awareness
- **C3.2** How are incidents in components handled?

  - **C3.2.1** Security incident handling
  - **C3.2.2** Incident by what policy
  - **C3.2.3** Incident by whom
  - **C3.2.4** Specific security team
    - **C3.2.4-0** Not mentioned
    - **C3.2.4-1** Yes
    - **C3.2.4-2** No
    - **C3.2.4-3** Other
  - **C3.2.5** Incident process
  - **C3.2.6** Incident process history
  - **C3.2.7** Disclosure policies
    - **C3.2.7-0** Not mentioned
    - **C3.2.7-1** Yes
    - **C3.2.7-2** No
    - **C3.2.7-3** Other
- **C3.3** Project provides documentation for including external code?
  - **C3.3.1** Documentation
    - **C3.3.1-0** Not mentioned
    - **C3.3.1-1** Yes
    - **C3.3.1-2** No
    - **C3.3.1-3** Other
  - **C3.3.2** Documentation Opinion

C4 Experiences OSCs

- **C4.1** Developer experience using components?
  - **C4.1.2** Development experience
    - **C4.1.2-0** No opinion
    - **C4.1.2-1** Mostly Negative
    - **C4.1.2-2** Neutral
    - **C4.1.2-3** Mostly Positive
  - **C4.1.1** Did customize OSC in the past?
    - **C4.1.1-0** Not mentioned
    - **C4.1.1-1** Yes
    - **C4.1.1-2** No
    - **C4.1-1-3** Other
- **C4.2** How are components kept up-to-date?
  - **C4.5** OSC update
  - **C4.6** OSC update responsible
  - **C4.7** OSC update version
  - **C4.8** OSC update checks
  - **C4.9** OSC update metrics
- **C4.3** Would you use the same components again?
  - **C4.3-0** Not mentioned
  - **C4.3-1** Mostly Yes

    - **C4.3-2** Mostly No
    - **C4.3-3** Other
- **C4.4** How are releases and updates handled?
    - **C4.11** Release process
    - **C4.12** Release decision
    - **C4.13** Release secured
    - **C4.14** Release update system
    - **C4.15** Release deprecation
    - **C4.16** Release dependencies

C5 Challenges and Incidents

- **C5.1** Opinion of incident
    - **C5.1-0** No opinion
    - **C5.1-1** Mostly Negative
    - **C5.1-2** Neutral
    - **C5.1-3** Mostly Positive
- **C5.2** General trust strategy
    - **C5.2-0** Handling similar incident
    - **C5.2-1** Trust strategy
    - **C5.2-2** Identify untrustworthy
    - **C5.2-3** Exclude components
- **C5.3** Past security challenges / inconveniences
    - **C5.3.1** Past Challenges encountered?
        - **C5.3.1-0** Not mentioned
        - **C5.3.1-1** Yes
        - **C5.3.1-2** No
        - **C5.3.1-3** Other
    - **C5.3.2** Past Inconveniences

C6 Problems and Improvements

- **C6.1** Opinions
    - **C6.1.1** Internal opinion
        - **C6.1.1-0** No opinion
        - **C6.1.1-1** Mostly Negative
        - **C6.1.1-2** Neutral
        - **C6.1.1-3** Mostly Positive
    - **C6.1.2** External opinion
        - **C6.1.2-0** No opinion
        - **C6.1.2-1** Mostly Negative
        - **C6.1.2-2** Neutral
        - **C6.1.2-3** Mostly Positive
- **C6.2** Improvements to [Select all that apply, feel free to extend]
    - **C6.2-0** More developer hours
    - **C6.2-1** Better documentation / guidance

- **C6.2-2** Static analysis and similar tooling
- **C6.2-3** Audit external components (on introduction and updates)
- **C6.2-4** Tust processes between oss and third party devs (TLS etc)
- **C6.2-5** Resources for security implications (mailings lists)
- **C6.2-6** (certificate) updates for long lifecycle devices
- **C6.2-7** Automated alerts for dep. updates (CI)
- **C6.2-8** Contribute back to dependencies
- **C6.2-9** Make transportation more secure
- **C6.2-10** Regular pentests
- **C6.2-11** Build security in from the start
- **C6.2-12** Dedicated security expert for project
- **C6.2-13** More/better quality assurance
- **C6.2-14** Use security software (e.g., proxy)
- **C6.2-15** Better security education for devs
- **C6.2-16** Incentives/ monetary rewards
- **C6.2-17** Rewrite deps in-house