Developing a Security Mindset for Your Open Source Project

Authors: *Georgia Bullen, Robert P. Davey, Beth Duckles, Jonah Duckles, Eriol Fox, Kate Hertweck, Dan Sholler, David Swenson, Kirstie Whitaker*

Published: August 12, 2025

To use this resource, Make a Copy to your own Google Drive.

Introduction

This worksheet is designed to help your open source project team have meaningful conversations about security practices. By working through these questions together, you'll identify potential security gaps and develop a more security-conscious mindset across your team.*

Setting aside a one to two hour conversation with your team would be a good start at bringing more of a security mindset. The below worksheet can walk a team through security considerations and ways in which the team feels vulnerable / exposed.

Comments, contributions and questions can be shared with the editorial team using GitHub Issues on the repository used for managing and maintaining this work: eoss-om-communitycalls/2025-05-22-security-mindsets-worksheet

Or reach out to the team at Organizational Mycology by email at: info@orgmycology.com They can provide services to facilitate these discussions across larger teams and help your community build accountability frameworks for these topics.

**This resource was developed based on conversation with attendees at the May 22, 2025 Chan-Zuckerberg Initiative community call for the Essential Open Source Software Program on the topic of Security in Open Source. Comments welcome in this document.*

Part 0: Preparation for Your Discussion

Before hosting the workshop, ask yourself:

1. Are the right people in the room to have this discussion?
2. Have you set aside enough time for your project's security concerns?
3. Have you experienced any security incidents in the past? What happened and how did you respond?
4. Do you have a clear sense of how actions and next-steps will be shared by the group?
5. Looking at the lists below, does everyone who will be in the room have a working understanding of the concepts? If not, consider ways to bring them up to speed during the conversation or with external resources

Enter your answers below:

Part 1: Security Landscape Assessment

What security threats is your project most vulnerable to? (Check all that apply)

- ☐ Supply chain vulnerabilities (compromised dependencies)
- ☐ Contributors that might cause collateral damage
- ☐ Data exfiltration
- ☐ CI/CD system compromise
- ☐ Namespace confusion/squatting
- ☐ Unmaintained dependencies
- ☐ AI-generated code with hidden vulnerabilities
- ☐ Lack of capacity to manage security audits, issues and solutions
- ☐ Other: _____

Discussion Questions:

1. Which of these threats concerns you most and why?
2. What specific assets or components of your project need the most protection?
3. Which of these threats is least likely to happen but most devastating to your project if it did? Can you play out/model out this scenario?
4. What are your blindspots, who could you engage to help you see them better?

Enter your answers below:

Part 2: Contributor Management

How does your project currently and continually, manage and vet contributors? (Check all that apply)

- ☐ Review of first PR only
- ☐ Progressive trust model (limited access initially)
- ☐ Background verification for critical contributors
- ☐ Require 2+ reviewers for all PRs
- ☐ Limit CI/CD access for external contributors
- ☐ An existing person 'knows' a person and has met them in person
- ☐ No formal vetting process
- ☐ Other: _____

Discussion Questions:

1. What is your process when a new contributor submits their first PR?
2. How do you balance being welcoming to new contributors while protecting your project?
3. Who has administrative access to your source control and CI/CD systems?
4. Who or what organization "owns" those accounts, and deployment pipelines and are they still actively involved in the project?
5. What process do you use to give contributor elevated permissions?
6. What is your process for offboarding, or managing inactive, contributors and maintainers?

Enter your answers below:

Part 3: Dependency Management

How does your project manage dependencies? (Check all that apply)

- ☐ Regular dependency audits
- ☐ Automated vulnerability scanning (e.g., DependaBot)
- ☐ Software Bill of Materials (SBOM) generation
- ☐ Pinned dependency versions
- ☐ Verification of dependency maintainer activity
- ☐ No formal dependency management
- ☐ Other: _____

Discussion Questions:

1. How often do you review and update dependencies?
2. How would you know, and what would you do if a critical dependency was found to be compromised?
3. Do you know which of your dependencies are actively maintained?

Enter your answers below:

Part 4: Security Responsibilities

Basic Security

- ☐ 2 Factor authentication for all critical systems
- ☐ Passwords managers used for sharing passwords
- ☐ Regular access control audits
- ☐ Secure communications for project leaders
- ☐ Logging / monitoring
- ☐ Other: _____

Who handles security in your project? (Check all that apply)

- ☐ Dedicated security team/working group
- ☐ All core maintainers share responsibility
- ☐ Single security point person
- ☐ Ad-hoc as issues arise
- ☐ Outsourced externally
- ☐ No one specifically
- ☐ Other: _____

Discussion Questions:

1. How are security responsibilities distributed among team members?
2. What security skills or knowledge gaps exist in your team?
3. How do you train team members on security practices?
4. How do you ensure security doesn't become siloed or overlooked?
5. How do you ensure you keep up to date with wider security issues and practices?

Enter your answers below:

Part 5: Action Planning

Based on your discussions, what are your top 3 security priorities?

1. _____
2. _____
3. _____

If these are your priorities, what other work will you stop doing, or deprioritize to address them?

How will you track progress toward addressing these priorities?

References and Resources

Tools & Services

- **FOSSA** : Vulnerability scanning and license compliance
- **DependaBot**: Automated dependency updates
- **GitHub Security Fund**: https://resources.github.com/github-secure-open-source-fund/
- **Superbloom**: UI/UX evaluation for security improvements
- **Convocation Design:** Security Audits, secure UI/UX and policy
- **Arachne Digital**: Open-source tool for identifying and mapping cyber threat actors
- To add more resources, please contribute to this GitHub Issue

Community Resources

- **Scientific Python Security Specs** : https://scientific-python.org/specs/spec-0006 and https://scientific-python.org/specs/spec-0008
- **CHAOSS Security Guide**: https://chaoss.community/practitioner-guide-security/
- **SAFE OSE Program**: https://www.nsf.gov/funding/opportunities/safe-ose-safety-security-privacy-open-source-ecosystems
- **Arachne Digital**: https://www.arachne.digital/ (security consulting)

Security Concepts to Explore

- Software Bill of Materials (SBOM) and Supply chain security
- Slopsquatting (https://en.wikipedia.org/wiki/Slopsquatting)
- Progressive trust models for contributors
- Security through usability

Remember: Security is not a one-time effort but an ongoing process. Regular conversations about security help build a security mindset throughout your team.