GitHub

Seven questions to ask before using open source software at work

Open source helps enterprise teams build better software, faster—but also comes with unique risks and challenges. Is your organization ready to use open source software securely? Here are seven questions to ask (and answer) to keep your team and your code safe.

1. Is there an approval process to help developers introduce new open source software (OSS) into our organization?

- [ ] Yes
- [ ] No
- [ ] Maybe

To keep up with the volume of OSS code, larger organizations often have open source program offices to help them maintain company policies while using OSS. Work with your organization's legal team to create an official approval process for contributing to OSS projects.

2. Does our organization have a list of pre-approved OSS licenses, so developers know which OSS they can use in proprietary software?

- [ ] Yes
- [ ] No
- [ ] Maybe

Open source licenses set the rules and guidelines for how OSS projects can be viewed, used, modified, and distributed—even in proprietary software. That doesn't mean you can't use any OSS in proprietary code; just confirm the OSS you use has the right license for your application's goals and organization's policies.

3. Can we easily audit dependencies, licenses, and other important information for the open source projects our organization will use?

- [ ] Yes
- [ ] No
- [ ] Maybe

Ignoring your organization's compliance policies only creates future vulnerability or licensing problems. Automate compliance workflows now by building in process gates with required statuses and other code checks.

4. Does my team have a plan and tools for identifying and fixing vulnerable components?

- [ ] Yes
- [ ] No
- [ ] Maybe

Up to 70 percent of new software projects depend on open source code. Code reuse helps everyone build better software faster, but it also puts us all at risk of distributing security vulnerabilities. Make sure your developers have the tools they need to identify vulnerabilities and patch vulnerable code quickly.

5. Does my team use tools like token scanning to make sure highly sensitive credentials aren't accidentally pushed to public repositories?

- [ ] Yes
- [ ] No
- [ ] Maybe

Even the most security-minded organizations eventually make a mistake. Protect your team from accidentally leaking secrets with tools that scan commits as they're shared and proactively invalidate credentials before they can be compromised.

6. Do we have a private registry of software packages that are safe to use within our organization?

- [ ] Yes
- [ ] No
- [ ] Maybe

When you consume an open source package, it's important to trust and understand the code. Use a package registry to quickly find what's been approved for your organization's repositories—and easily store packages in the same secure environment as your source code.

7. Do I have a way to validate users' identities so that only my team is committing code to our projects, including signed commits?

- [ ] Yes
- [ ] No
- [ ] Maybe

Building with open source code means adding thousands of external developers to your team. Whether you're collaborating on public or private projects, work with your identity provider to always verify certain commits and tags as coming from your organization.

Check your score

Did you answer "no" or "maybe" to any of the questions above? Your team may still have a few security changes to make before using open source software.

Need help getting started?

Talk to a GitHub security expert:

github.com/enterprise

experts@github.com