TIDELIFT

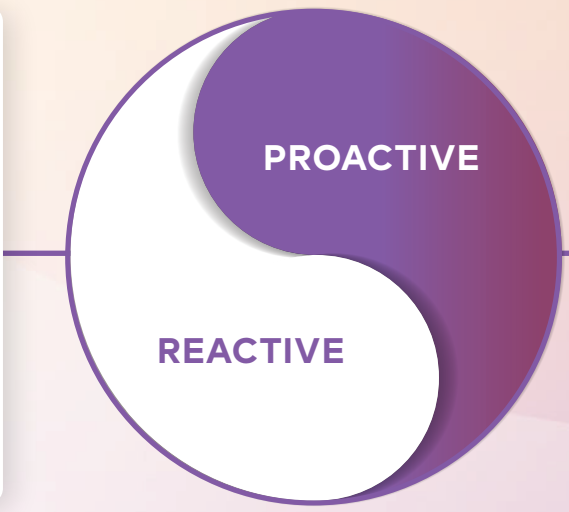
10 questions you should answer before using a new open source project

When it comes to open source software security, many organizations rely heavily on software scanning (often called software composition analysis or SCA) as the primary means of defense. While scanning helps protect against known vulnerabilities *reactively*, leading organizations today are adding *proactive* defenses that help them make better decisions about which open source packages to bring into their supply chain in the first place.

PROTECT AGAINST CURRENT RISK

Identify and resolve known issues and vulnerabilities in the open source packages your organization uses.

BENEFIT
Harden defenses by closing off attack vectors that have already been identified.



PROTECT AGAINST FUTURE ISSUES

Identify which open source packages your organization uses follow secure software development practices.

BENEFIT
Ensuring the work is done minimizes the likelihood of being impacted by issues in the first place.

The easiest way to avoid having to replace problematic open source dependencies is to not bring them in at all.

01 Is the project abandoned or is it actively maintained and receiving fixes?

Using unmaintained projects means potential future issues will not be fixed.

02 Is the project deprecated?

Using deprecated projects means potential future issues will not be fixed.

03 Who are the maintainers and how many maintainers are behind the project?

The more you know about the maintainers and their projects, the better. Knowing there are multiple maintainers is a good sign for continued maintenance.

04 Who has publishing rights on upstream package managers?

This helps ensure only approved maintainers are contributing code.

05 Does the project have security practices such as multi-factor authentication in place?

This provides an extra layer of security to ensure only authorized maintainers are able to access code.

06 Does the project have a history of responding to security and other issues?

This helps you understand the maintainers previous track record responding to security issues.

07 What is the version history and which is the recommended version?

This helps you make good decisions about which version to use and ensure you're using a version that aligns with your organization's policies.

08 Is the project or release impacted by any existing vulnerabilities?

This minimizes new vulnerabilities entering your organization's software development lifecycle.

09 What are the associated project and release dependencies?

Understanding a project's transitive dependencies helps you avoid accidentally bringing in a vulnerability from a package dependency.

10 Is the license compatible with your organization's legal guidelines?

Checking the package's license to ensure it aligns with your organization's policies helps avoid unnecessary legal risk.

Tidelift is the only solution in the market that ensures maintainers meet these standards. Schedule a demo to see the secure software supply chain we're building.