Securing GitLab's Supply Chain

Software supply chains are now the center of attention of most companies. With the increased reliance on SaaS software and complex relationships between vendors, a security incident can very quickly trickle down to a massive knock-down effect similar to the Solarwinds attack, with about 18,000 companies affected.

Software supply chain attacks occur when the materials or processes of producing software are themselves compromised, resulting in vulnerabilities targeting downstream consumers of the software produced. These attacks often have systemic effects on potentially hundreds of companies.

GitLab is *the One DevOps platform.* As such, security is central to our value proposition. Customers are looking for unified solutions that embed security every step of the way.

To achieve this, we provide our customers with a variety of automated tools, such as Static Application Security Testing (SAST), Dynamic Application Security Testing (DAST), dependency scanning, container scanning and license compliance capabilities.

Those tools, also known as analyzers, are the backbone of our security automation capabilities. Those solutions are also leveraged internally. We use them to automatically verify, secure and protect our own source code.

How can I be sure that the Open Source Software that GitLab uses is secure?

GitLab leverages numerous open source software packages in order to power our security tool suite. Open source software ("OSS") benefits from crowd sourced security where any contributors can identify potential vulnerabilities. All of our Open Source code is fully visible and anyone can contribute. We pioneer the approach of security through transparency. Some of the benefits include:

- Many more people can review, analyze and check the source code
- Vulnerabilities are identified quicker
- Remediation can be collegial and include numerous stakeholders

Nonetheless, GitLab follows an additional rigorous security vetting process that is detailed below. This will greatly help in mitigating risk of vulnerable software being embedded into our DevOps platform. This also ensures that all software in use is compliant with our strong internal security standards.

GitLab releases a new version every single month on the 22nd. For each new release, we check if any of the Open Source software we leverage has been updated.

Current Analyzers

As GitLab is offered as a unified solution, we check for any differences between the latest version of the analyzer and the one currently packaged within the product. Our goal is to improve the security of the product with each iteration, while upholding our internal security standards.

We perform numerous checks during this process. The following list is a sample of what our security team reviews:

Review type	Review process	Periodicity
Breaking changes	Any changes that are impacting the analyzer's performance	Each release cycle (monthly)
Rules changes	Any changes to the analyzer's security rules, new rules added or any current rules deprecated	Each release cycle (monthly)
Behavioral changes	Any changes leading to unexpected results from analyzer	Each release cycle (monthly)
Compatibility changes	Any changes that might need the analyzer to be updated to the latest version to work properly	Each release cycle (monthly)
Vulnerability assessment	Any critical or high vulnerabilities discovered during the review process	Each release cycle (monthly)

We mirror copies of the most critical OSS projects we leverage as part of the product. In order to first verify if the new version is safe to be integrated, we push it to the mirrored copy. It will go through our CI/CD pipeline which includes all of the security testing (SAST, DAST, Container Scanner, Dependency Scanning, etc.). Once it is completed, we will review any vulnerabilities that were discovered during the tests.

Analyzers will not be updated if the new version contains a critical or high vulnerability. If discovered by GitLab during the review, team members will assist the maintainers of the project in the remediation process. The updates to the GitLab product will be withheld until the relevant dependencies have been patched.

If a *breaking change* is discovered, an issue will be created internally and the discovery work will be prioritized.

For any new code pushed to the analyzer's repository, 3 individuals will review the merge request, enforcing the separation of duties principle:

1. *Submitter* reviews for any relevant upstream changes
2. *Reviewer* reviews the changes before merger and release
3. *Maintainer* will review the changes before merger and release

New Analyzers

We may need to add new capabilities to the product, to support new languages and upgrade our testing suite. If we develop a new analyzer, it will undergo a full security review.

This includes everything mentioned above for the *current analyzers* as well as the following:

- Offline execution (no Internet access)
- Ability to be configured to use custom proxies and/or CA certificates
- Launchable via a Command Line Interface
- Bundle-able dependencies to be packaged as a Docker image in order to go through our container security scanning

Conclusion

In addition to the specific controls and security measures highlighted above, GitLab has a strong commitment to protect its supply chain holistically.

Our security initiatives and controls are further referenced in the GitLab Security Handbook.