Twistlock vs GitLab

GitLab compared to other DevOps tools

Twistlock provides lifecycle security for containerized environments, â€œfrom pipeline to perimeterâ€. Twistlock capabilities include runtime defense, vulnerability management, cloud native firewalls, pre-built compliance templates for HIPAA, PCI, GDPR, and NIST SP 800-190. It can be integrated into your CI/CD pipeline. Automated and custom policies can block builds or deployments based on vulnerabilities or compliance requirements. Runtime capabilities are limited to containerized applications.

Twistlockâ€™s container security features are robust, but they do not offer the breadth of GitLabâ€™s security scans. GitLab Ultimate automatically includes broad security scanning with every code commit including Static and Dynamic Application Security Testing, along with dependency scanning, container scanning, and license management.

FEATURES		
Static Application Security Testing GitLab allows easily running Static Application Security Testing (SAST) in CI/CD pipelines; checking for vulnerable source code or well known security bugs in the libraries that are included by the application. Results are then shown in the Merge Request and in the Pipeline view. This feature is available as part of [Auto DevOps](https://docs.gitlab.com/ee/topics/autodevops/#auto-sast) to provide security-by-default. Learn more about Static Application Security Testing	✗	✓
Dependency Scanning GitLab automatically detects well known security bugs in the libraries that are included by the application, protecting your application from vulnerabilities that affect dependencies that are used dynamically. Results are then shown in the Merge Request and in the Pipeline view. This feature is available as part of [Auto DevOps](https://docs.gitlab.com/ee/topics/autodevops/#auto-dependency-scanning) to provide security-by-default. Learn more about Dependency Scanning	✗	✓
Container Scanning When building a Docker image for your application, GitLab can run a security scan to ensure it does not have any known vulnerability in the environment where your code is shipped. Results are then shown in the Merge Request and in the Pipeline view. This feature is available as part of [Auto DevOps](https://docs.gitlab.com/ee/topics/autodevops/#auto-container-scanning) to provide security-by-default. Learn more about container scanning	✓	✓

Dynamic Application Security Testing

Once your application is online, GitLab allows running Dynamic Application Security Testing (DAST) in CI/CD pipelines; your application will be scanned to ensure threats like XSS or broken authentication flaws are not affecting it. Results are then shown in the Merge Request and in the Pipeline view. This feature is available as part of [Auto DevOps] (https://docs.gitlab.com/ee/topics/autodevops/#auto-sast) to provide security-by-default.

Learn more about application security for containers


Interactive Application Security Testing

[IAST](https://blogs.gartner.com/neil_macdonald/2012/01/30/interactive-application-security-testing/) combines elements of static and dynamic application security testing methods to improve the overall quality of the results. IAST typically uses an agent to instrument the application to monitor library calls and more. GitLab does not yet offer this feature.


Runtime Application Security Testing

RASP uses an agent to instrument the application to monitor library calls as the application is running in production. Unlike other security tools, RASP can take action to block threats in real-time, similar to a Web Application Firewall but from within the app's runtime environment rather than at the network layer. GitLab does not yet offer this feature.