BlackDuck vs GitLab

GitLab compared to other DevOps tools

BlackDuck offers security scanning of open source components, container scanning, and license management. BlackDuck was purchased by Synopsis. Synopsis also offers separate products for IAST and SAST, but those will be compared separately as they are purchased separately.

BlackDuck maintains an Inventory of all open source code and its vulnerabilities and makes it available in the CI/CD pipeline via APIs. Unlike GitLab, it can detect more granular components beyond library use. Like GitLab, BlackDuck also uses a national vuln database but they add their own proprietary research.

When a new vulnerability is announced, BlackDuck knows immediately what code is affected. BlackDuck provides remediation advise, scoring, and CWE vuln type. The dashboard shows projects affected, by vulnerability, by project app or container, including a risk view. You can also pivot to view vulns by project and projects affected by vulns. BlackDuck covers 81 languages.

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. While BlackDuck can integrate with GitHub and other tools, that approach will require multiple software licenses and integration / maintenance effort.

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.


License Management

Check that licenses of your dependencies are compatible with your application, and approve or blacklist them. Results are then shown in the Merge Request and in the Pipeline view.

Learn more about License Management