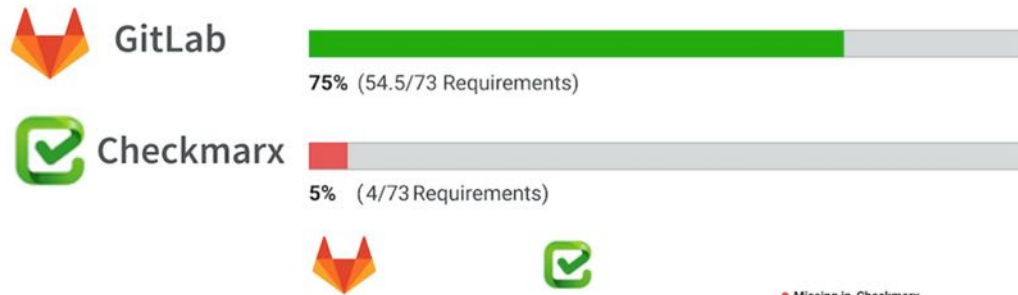


# GitLab vs Checkmarx

## Decision Kit



Category	GitLab Score	Checkmarx Score	Missing in Checkmarx
Manage	5.5/8	1/8	<ul style="list-style-type: none"> <li>Subgroups</li> <li>Audit Events</li> <li>Audit Reports</li> <li>Compliance Management</li> <li>Code Analytics</li> <li>DevOps Reports</li> <li>Value Stream Management</li> <li>Insights</li> </ul>
Plan	6/8		<ul style="list-style-type: none"> <li>Issue Tracking</li> <li>Kanban Boards</li> <li>Time Tracking</li> <li>Epics</li> <li>Roadmaps</li> <li>Service Desk</li> <li>Requirements Management</li> <li>Quality Management</li> </ul>
Create	7.5/8		<ul style="list-style-type: none"> <li>Source Code Management</li> <li>Code Review</li> <li>Wiki</li> <li>Static Site Editor</li> <li>Web IDE</li> <li>Live Preview</li> <li>Snippets</li> <li>Design Management</li> </ul>
Verify	6/8		<ul style="list-style-type: none"> <li>Continuous Integration</li> <li>Code Quality</li> <li>Code Testing and Coverage</li> <li>Load Testing</li> <li>Web Performance</li> <li>Usability Testing</li> <li>Accessibility Testing</li> <li>Merge Trains</li> </ul>
Package	4.5/6		<ul style="list-style-type: none"> <li>Package Registry</li> <li>Container Registry</li> <li>Helm Chart Registry</li> <li>Dependency Proxy</li> <li>Jupyter Notebooks</li> <li>Git LFS</li> <li>Dependency Firewall</li> </ul>
Secure	7/8	3/8	<ul style="list-style-type: none"> <li>SAST</li> <li>DAST</li> <li>Fuzz Testing</li> <li>Dependency Scanning</li> <li>Container Scanning</li> <li>License Compliance</li> <li>Secret Detection</li> <li>Vulnerability Management</li> </ul>
Release	7/8		<ul style="list-style-type: none"> <li>Continuous Delivery</li> <li>Pages</li> <li>Review Apps</li> <li>Advanced Deployments</li> <li>Feature Flags</li> <li>Release Orchestration</li> <li>Release Evidence</li> <li>Secrets Management</li> </ul>
Configure	4.5/7		<ul style="list-style-type: none"> <li>Auto DevOps</li> <li>Kubernetes Configuration</li> <li>ChatOps</li> <li>Runbooks</li> <li>Serverless</li> <li>Infrastructure as Code</li> <li>Cluster Cost Optimization</li> </ul>
Monitor	5/8		<ul style="list-style-type: none"> <li>Metrics</li> <li>Alert Management</li> <li>Incident Management</li> <li>Logging</li> <li>Tracing</li> <li>Error Tracking</li> <li>Product Analytics</li> <li>Synthetic Monitoring</li> </ul>
Defend	1.5/3		<ul style="list-style-type: none"> <li>Web Application Firewall</li> <li>Container Host Security</li> <li>Container Network Security</li> </ul>

## Summary

Checkmarx is a long-standing company with their roots in SAST. They are recognized as a Leader in the Gartner Application Security Testing Magic Quadrant.

## Comparison to GitLab

Although Checkmarx has a more mature SAST offering, GitLab offers a much broader range of security testing capabilities, including DAST and Fuzz Testing. GitLab's capabilities come integrated with the rest of GitLab out-of-the-box and do not require any special integration to shift the workflow left to the development team. GitLab customers report that GitLab generally has a better false positive rate than Checkmarx, which saves time when trying to find true vulnerabilities that really matter. Checkmarx's established position in the security market and deep SAST capabilities are offset by GitLab's lower price point and tighter integration with the rest of the software development lifecycle.

The Checkmarx vision is closest to GitLab among the AppSec vendors, but because they must integrate into the rest of the SDLC via APIs, their path

toward execution is more limited. Also, like the other AppSec vendors, Checkmarx is expensive. It is priced per developer with a rough estimate of 12 Developers for \$59k USD per year or 50 Developers for \$99k USD per year. Checkmarx uses Whitesource for dependency scanning and charges an extra \$12k USD per year for this open source scanning.

Checkmarx excels in that they are context aware, meaning they can mark what is not exploitable based on path. GitLab lacks this capability. On the other hand, GitLab automatically includes broad security scanning with every code commit including Static and Dynamic Application Security Testing, along with dependency scanning, container scanning, and license compliance. All of this is part of the single GitLab Ultimate application.

## Security Scanning

### Strengths and Weaknesses

	GitLab	Checkmarx
<b>Strengths</b>	<ul style="list-style-type: none"> <li>Cost is significantly less expensive than Checkmarx</li> <li>Tight integration with developer workflow</li> <li>Complete range of application testing types (SAST, DAST, etc.) are included by default</li> <li>Comparatively low false positive rates</li> </ul>	<ul style="list-style-type: none"> <li>Strong offering across scanning types</li> <li>Good integration with IDEs and local developer environments</li> <li>Well known, market-leading SAST offering</li> </ul>
<b>Weaknesses</b>	<ul style="list-style-type: none"> <li>GitLab's SAST offering only scans code repositories today and cannot scan compiled binaries</li> </ul>	<ul style="list-style-type: none"> <li>SCA is essentially a brand new product and only available as an add-on to their SAST product</li> <li>DAST is only available as a managed service via a partnership</li> <li>Fuzz testing is not offered</li> <li>Each kind of testing is a separate piece of software that must be licensed, managed, and integrated with the DevOps lifecycle separately</li> <li>Operating system support to run the Checkmarx software is limited to Windows</li> <li>Significant tuning is required to reduce false positives</li> </ul>

### Feature Lineup

	GitLab	Checkmarx
SAST	✓	✓
DAST	✓	managed service only
IAST		✓
SCA: Vulnerability Scanning	✓	✓
SCA: Open Source Audit	✓	✓
Fuzz Testing	✓	

## Feature Comparison

### 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 to provide security-by-default.

[Learn more about Static Application Security Testing](#)

#### Secret Detection



supports 18 languages

GitLab allows you to perform Secret Detection in CI/CD pipelines; checking for unintentionally committed secrets and credentials. Results are then shown in the Merge Request and in the Pipeline view. This feature is available as part of [Auto DevOps](#) to provide security-by-default.



[Learn more about Secret Detection](#)

### 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](#) 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](#) 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](#) to provide security-by-default.



[Learn more about application security for containers](#)

### Interactive Application Security Testing



IAST 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.



### License Compliance



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



[Learn more about License Compliance](#)

### On-demand Dynamic Application Security Testing



"There's no reason to wait for the next CI pipeline run to find out if your site is vulnerable or to reproduce a previously found vulnerability. GitLab offers scanning your running application with On-demand Dynamic Application Security Testing (DAST), independent of code changes or merge requests."



[Learn more about On-demand DAST](#)



## Why GitLab?

- [Product](#)
- [Solutions](#)
- [Services](#)
- [DevOps lifecycle](#)
- [DevOps tools](#)
- [Is it any good?](#)
- [Releases](#)
- [Pricing](#)
- [Get started](#)

## Resources

- [All resources](#)
- [All-Remote](#)
- [Blog](#)
- [Newsletter](#)
- [Events](#)
- [Webcasts](#)
- [Topics](#)
- [Training](#)
- [Docs](#)
- [Install](#)

## Community

- [Customers](#)
- [Contribute](#)
- [Partners](#)
- [Channel Partners](#)
- [Explore repositories](#)
- [Source code](#)
- [Shop](#)
- [Direction](#)
- [Contributors](#)
- [Core Team](#)
- [Hall of fame](#)
- [Community Forum](#)

## Support

- [Get help](#)
- [Contact Sales](#)
- [Contact Support](#)
- [Support options](#)
- [Status](#)
- [Customers portal](#)

## Company

- [About](#)
- [What is GitLab?](#)
- [Jobs](#)
- [Culture](#)
- [Team](#)
- [Press](#)
- [Analysts](#)
- [Handbook](#)
- [Security](#)
- [Contact](#)
- [Terms](#)
- [Privacy](#)
- [Trademark](#)

Git is a trademark of Software Freedom Conservancy and our use of 'GitLab' is under license