

# Solution brief Vulcan Cyber and GitLab

# The challenge

In the modern enterprise landscape, the adoption of Continuous Integration/Continuous Deployment (CI/CD) methodologies is paramount for business growth. However, a significant challenge emerges as engineering teams are tasked with not only writing code but also securing it, which often falls lower on their list of priorities. Naturally, engineers typically prefer working within their native tools, focusing solely on coding rather than integrating security measures. To overcome this, security teams need to maintain comprehensive asset inventories and ensure all assets are surfaced and continually monitored for potential software risks.

- Over 75% of applications have at least one flaw
- About 20% of cyber attacks target vulnerabilities in web applications

# Vulcan Cyber and GitLab - better together

The Vulcan Cyber and GitLab integration enables the ingestion of code project inventories that correlate with other third-party security findings, ensuring comprehensive monitoring for potential risks related to their software assets. This integration empowers organizations to enhance their application security posture by proactively identifying vulnerabilities in their codebases and dependencies, and communicating remediation guidelines to developers and engineers effectively. For GitLab Ultimate users, this integration goes a step beyond by injecting both static and dynamic application security testing capabilities, along with dependency scanning functionalities.

# Joint solution key features

- **Risk exception:** Streamline exception handling by communicating it directly to engineers, ultimately reducing operational burdens and shortening remediation cycles.
- **Reporting:** Access advanced reporting capabilities including historical trends, policy compliance tracking, and Root Cause Analysis (RCA) insights.

• **Incorporate code projects into your organization's holistic view:** Enable structured organization hierarchy for business units, departments, GEOs, and applications.

#### **Use cases**

- Effective Software Bill of Materials (SBOM) management: Identify vulnerabilities within code libraries and mitigate associated risks
- **Dev security collaboration:** Accelerate software risk remediation by fostering collaboration between engineering and security teams, allowing each to work within their native tools.
- **Comprehensive visibility of application risks:** Get a centralized view of all code and application assets and their associated risks (alongside other types of risks).
- **SLA compliance:** Ensure stakeholder alignment within the organization to adhere to company software risk policies.

🥶 GitLab.org > 🚥 > 👰 Tests > webgoat > Vulnerability Report > 41857351	Explain this vulnerability × This response is generated by AI.				
Needs triage Detected · Apr 19, 2022, 7:02 PM in pipeline 520134833					
Improper Restriction of XML External Entity Reference ('XXE')	Improper Restriction of XML				
Description	('XXE') Vulnerability				
XML External Entity (XXE) attacks can occur when an XML parser supports XML entities while processing XML rece	(7012) Furiorability				
Severity:  Critical	Explanation				
Project: GitLab.org / security-products / Tests / webgoat	The vulnerability "Improper Restriction of XML External Entity Reference ('XXE')" occurs when an application processes XML input from untrusted sources, allowing an attacker to read files, execute				
Tool: SAST					
Scanner: Semgrep					
Location	remote code, or perform other malicious actions on the server. The vulnerability is caused by the				
File: webgoat-lessons/xxe/src/main/java/org/owasp/webgoat/plugin/Comments.java:70	application failing to properly restrict the use of external entities in XML input.				
Identifiers	In the provided code snippet, the vulnerability is				
• CWE-611	present in the parseXML method, where an XML input is passed to the createXMLStreamReader				
find_sec_bugs.XXE_XMLSTREAMREADER-1	method without proper validation. This allows an attacker to include external entities in the XML input, which can be used to read files or execute remote				
Find Security Bugs-XXE_XMLSTREAMREADER					
Explain this vulnerability and how to mitigate it with AI     Experiment	code.				
This is an experimental feature that uses AI to explain the vulnerability and provide recommendations. Use the Please provide your feedback and ideas in this issue.	Exploitation				
Try it out	To exploit this vulnerability, an attacker can craft a				
	malicious XML input that includes an external entity				
Linked items D 0	example:				
Link issues together to show that they're related. Learn more.	xml version="1.0" encoding="UTF-8</td				
	ELEMENT foo ANY				

VULCAN:	Hosts (0) C	ode Projects (16)	Websites (0)	Images (0)	Cloue	d Resources (0)					
ulÎ Dashboard	All Business Group	5 ~	≡ Filter 1						٩		
÷	Showing 8		Select Saved Filt	er ~							
Vulnerabilities	Project	Project		Asset > Connector	~ ] [	is ~	GitLab ×		~ :		
Campaigns	vulcan-int-lab/Connector Lab		+ Add Rule	+ Add Rule C							
Ø	vulcan-int-lab/Aweso	vulcan-int-lab/Awesome Sca Main							Apply -in Pre		
Exceptions	yaelvulcanlab/aviv_t	est	4	Jan 7, 2024	0	0	0	Compliant	Group: yaelvulca		
Automation	yaelvulcanlab/javavu	ıln	4	Jan 7, 2024	0	0	0	Compliant	Group: yaelvulca		
ф,	yaelvulcanlab/vulnp	/thon	4	Jan 7, 2024	0	0	0	Compliant	Group: yaelvulca) Pr		
Assets	yaelvulcanlab/conne	yaelvulcanlab/connector-lab yaelvulcanlab/Log4j		Jan 7, 2024	0	0	0	Compliant	Group: yaelvulca		
Remedies	yaelvulcanlab/Log4j			Jan 7, 2024	0	0	0	Compliant	Group: yaelvulca Pr		
	yaelvulcanlab/aweso	me-sca-main	¥	Jan 7, 2024	0	0	0	Compliant	Group: yaelvulca Pr		
Analytics											

VULCAN:	Hosts (0)	Code Projects (16)	Websites (0	)	Images (0)	Cloud Resources (0)					/ 🕮 💮 🛛 🗤 vulcan
ull Dashboard	vulcan-int-lab/Aw	Max Risk: 58	Created: Dec 28, 202	8 L	ast Seen: Jan 7, 2024						~ Take Action
Vulnerabilities	6 Vulnerabilities	Codebase	Libraries		Details	Activity				Language: U Status: A Sources	Unknown Active
Campaigns	Name				Components	Risk $\downarrow$	Sources	SLA Status	Status	Last Report 🛛 J	an 7, 2024, 12:49 PM
Exceptions	Unintended leak	of Proxy-Authorization head	er in requests		requests: 2.25.1	58	4	Compliant (80 Days)	Vulnerable	Citi ali Tarr	
0-⊕	Insufficient Verifi	ication of Data Authenticity			certifi: 2020.12.5	45	\$	Compliant (80 Days)	Vulnerable	Group: vulcan-int	lab Project: Awesome Sca Main
	Uncontrolled Re	Uncontrolled Resource Consumption			urllib3: 1.26.4	32	4	Compliant (170 Days)	Vulnerable	Visibility: private	
Assets	`Cookie` HTTP P	neader isn't stripped on cross	-origin redirects		urllib3: 1.26.4	32	4	Compliant (170 Days)	Vulnerable		
ije.	Insufficient Verifi	ication of Data Authenticity			certifi: 2020.12.5	32	4	Compliant (170 Days)	Vulnerable		
Remedies	urllib3's request changes request	body not stripped after redir method to GET	ect from 303 status		urllib3: 1.26.4	23	÷	Compliant (170 Days)	Vulnerable		
Analytics											
Connectors											

#### **About Vulcan Cyber**

Vulcan Cyber has developed the market-leading Exposure operating system (ExposureOS) to provide information security teams with one platform to prioritize, orchestrate, and mitigate exposure risk at scale throughout the entire attack surface.

#### **About GitLab**

GitLab, The DevOps Platform, empowers organizations to maximize the overall return on software development by delivering software faster and more efficiently while strengthening security and compliance.