GitLab

Taking the complexity out of compliance frameworks

Authors: Liz Burrows and Corey Oas

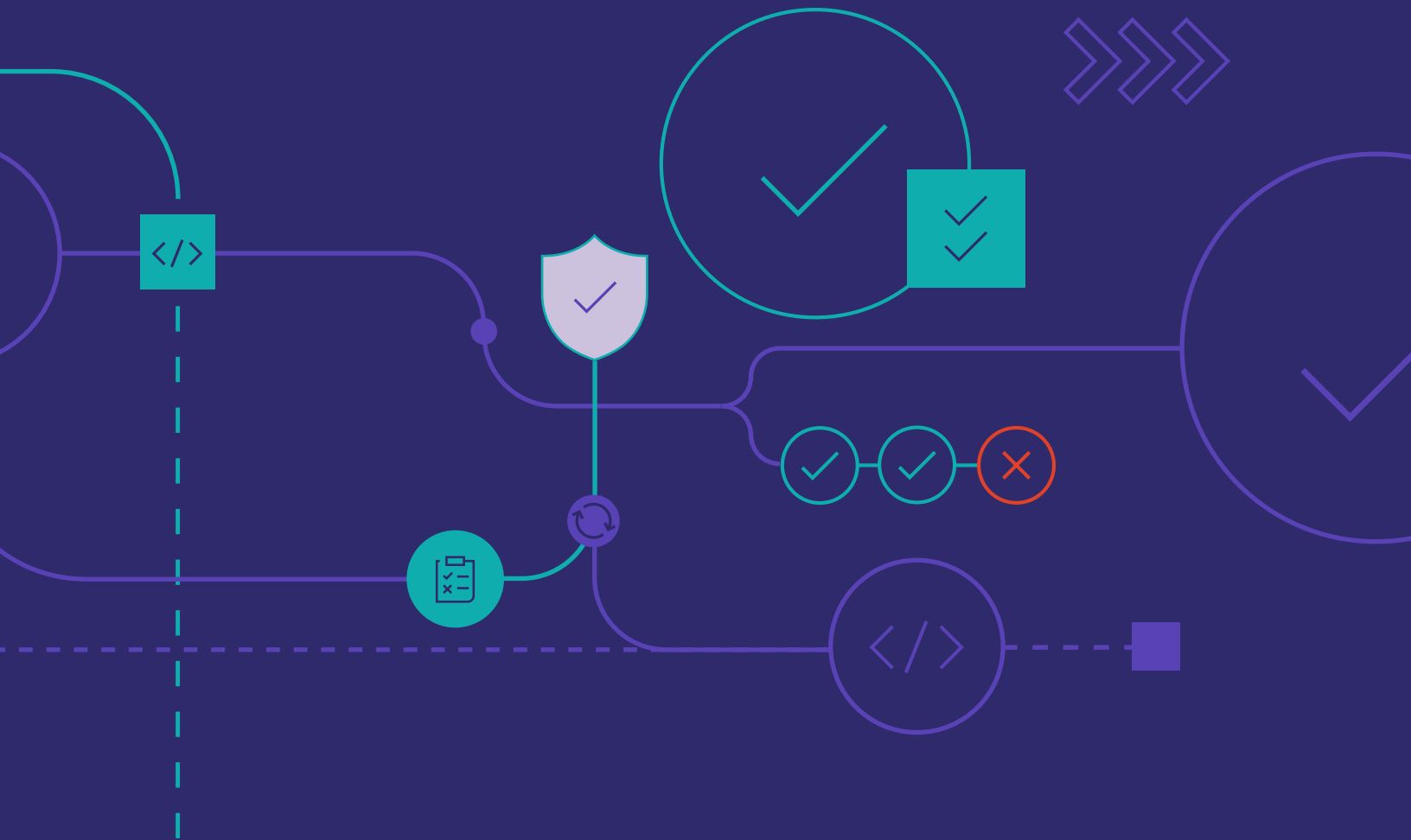
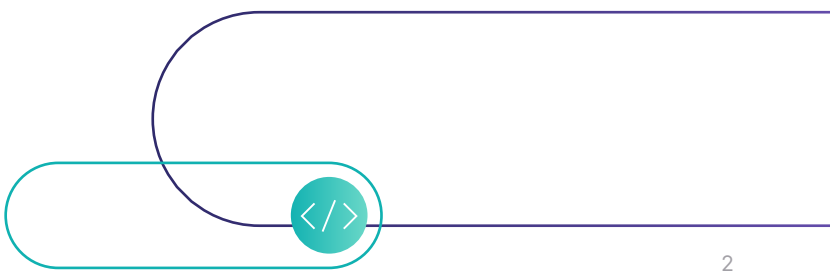


Table of Contents



Executive Summary

In today's landscape, cyber attacks pose a threat to all organizations, particularly for those linked to the U.S. government, where digital vulnerabilities can swiftly escalate to national security issues. Amidst evolving global regulations, organizations face challenges in keeping up with new rules and enforcement trends.

As cyber threats grow more sophisticated, regulatory bodies are intensifying their security mandates, such as the White House Cybersecurity Executive Order 14028. Non-compliance carries hefty penalties, emphasizing the need for specific cybersecurity protocols.

Navigating complex regulatory landscapes demands a strategic, agile approach. Manual methods are no longer sufficient; organizations need an end-to-end DevSecOps platform, like GitLab, that embeds security throughout the software development lifecycle and automates manual tasks. Forward-thinking organizations prioritize proactive compliance, utilizing advanced tools to mitigate risks and embed security practices seamlessly into the development process.

As cyber threats grow more sophisticated, regulatory bodies are intensifying their security mandates

Introduction

Historically, software companies have prioritized speed at the expense of security, leaving vulnerabilities in their products. This trend stems from the demand for rapid releases and has become particularly prominent with the widespread adoption of DevOps practices. **White House Cybersecurity Executive Order 14028** intensified the push for software manufacturers to enhance quality and secure their software supply chains. As an outcome of this mandate, security compliance frameworks, such as the **Secure Software Development Framework**, or NIST SSDF, aim to ease the burden of managing security compliance. However, the responsibility still rests on the industry, rather than regulatory bodies, to shift towards a culture of building secure software from inception. While the goals of most regulatory programs enjoy broad public support, in practice, regulation involves manual and siloed processes that can be costly and complex to navigate. To proactively address this challenge, organizations should embed compliance requirements and standards into the development process from the outset. By codifying these requirements and seamlessly integrating compliance throughout the software development lifecycle, organizations can realize significant time and cost efficiencies.

Decoding federal compliance standards

Decoding government compliance standards can feel like navigating a complex maze, leaving organizations in a state of uncertainty. Even if your organization operates outside the public sector, grasping the foundational principles of the **Federal Information Security Modernization Act** (FISMA), the **Federal Risk and Authorization Management Program** (FedRAMP), and the **National Institute of Standards and Technology** (NIST) is critical. Given the significant purchasing influence of the U.S. government worldwide, alignment with these standards may emerge as a prerequisite for conducting business with the public sector. Furthermore, the attainment of grant funding could be contingent upon adherence to governmental compliance criteria. Federal information security mandates tend to permeate state, local laws, and industry frameworks over time, underlining the widespread impact of governmental guidelines.

How these frameworks relate

NIST, FedRAMP, and FISMA join forces to create a unified framework that steers cybersecurity endeavors, especially for organizations in collaboration with the U.S. federal government. These components work hand in hand, each fulfilling a distinct role in fortifying the cybersecurity stance of federal government contractors and agencies. Picture them as essential components of a system where NIST serves as the cornerstone guideline, bolstering and improving the efficacy of the other elements.

Explaining FISMA

FISMA is federal legislation that defines comprehensive cybersecurity requirements for government agencies to protect their information and information systems. FISMA requires thorough information protection and cybersecurity measures for U.S. government systems, affecting a wide range of entities including federal agencies, state agencies handling federal data, organizations managing federal funds, and private-sector entities involved in federal grants, programs, or contracts. Compliance with FISMA hinges on adhering to NIST SP 800-53

standards, where organizations need to obtain an Authority to Operate (ATO) from each federal agency they engage with, and with security evaluations carried out by either the agency or an approved third-party assessor (3PAO). If you operate distinct systems for different agencies, you'll need separate ATO certifications due to varying System Security Plan requirements based on system type and data. In this regard, FISMA functions on a one-to-one basis, meaning each federal agency is responsible for its own cybersecurity compliance.

Navigating FedRAMP

FedRAMP is a government program that standardizes the security assessment, authorization, and continuous monitoring of cloud products and services used by federal agencies, setting a standard for cloud service providers (CSPs). Compliance with FedRAMP demonstrates that a CSP has met stringent security criteria, enabling access to a specialized government contract marketplace. This standardization removes the necessity for agencies to conduct individual security assessments, simplifying the procurement procedure. Like FISMA, the controls outlined in FedRAMP are based on NIST 800-53. FedRAMP authorization allows agencies to adopt a **do once, use many times** approach, saving the government an estimated 30-40%, reducing redundant agency security assessments, and lowering staffing needs. In contrast to FISMA's requirement for organizations to obtain individual ATOs from each federal agency, a FedRAMP ATO authorizes a cloud service provider to engage with any federal agency. While FISMA operates on a one-to-one basis, FedRAMP follows a many-to-one model.

Laying the groundwork with NIST

The **NIST Cybersecurity Framework**, also known as NIST SP 800-53, helps businesses understand and mitigate cybersecurity risks, laying the groundwork for meeting FISMA and FedRAMP requirements. Aligning cybersecurity practices with NIST's standards is crucial for organizations seeking government contracts, ensuring compliance, and establishing a strong security foundation. Adhering to NIST standards signals commitment to elevated security measures, paving the way for government contracts and demonstrating dedication to robust security standards.

Navigating NIST, FedRAMP, and FISMA with GitLab

Proactively automating processes, as opposed to merely reacting and following a checkbox approach, is critical for attaining compliance success. According to the **KPMG Chief Ethics and Compliance Officer Survey**, 80% of Chief Compliance Officers foresee escalating compliance pressures. Staying ahead of compliance challenges is therefore crucial. In the public sector, adherence to security compliance frameworks can help maintain public trust, protect resources, and avoid legal and financial consequences. By leveraging tools, like GitLab's **NIST 800-53 Configuration Guide** and partnering with GitLab's customer service team, organizations can streamline the development of an automated compliance process. The table below illustrates how GitLab's leading security compliance features align with these essential frameworks.

Compliance frameworks	FedRAMP	FISMA	CMMC	SSDF
Security guardrails				X
Security policies				X
Scan execution policies	X	X		X
Pipeline execution policies				X
Merge request approval policies	X	X	X	X
Branch protections	X	X		X
Compliance frameworks	X	X	X	X
Custom roles	X	X	X	X
Vulnerability scanning and management	X	X	X	X
Static Application Security Testing (SAST)	X	X		X
Secret Detection	X			X
Dynamic Application Security Testing (DAST) / Web App Scanning	X	X	X	X
API Security scanning	X	X		X
Container Scanning	X	X	X	X
Dependency Scanning				X
Infrastructure as Code (IaC) Scanning				X
Software bill of materials (SBOM)				X
Audit events	X	X	X	X
Verify legitimacy and prevent tampering	X	X		X
Two-factor authentication	X	X	X	X
Artifact attestation				X
Container signing	X	X		X
Verified commits	X	X		X

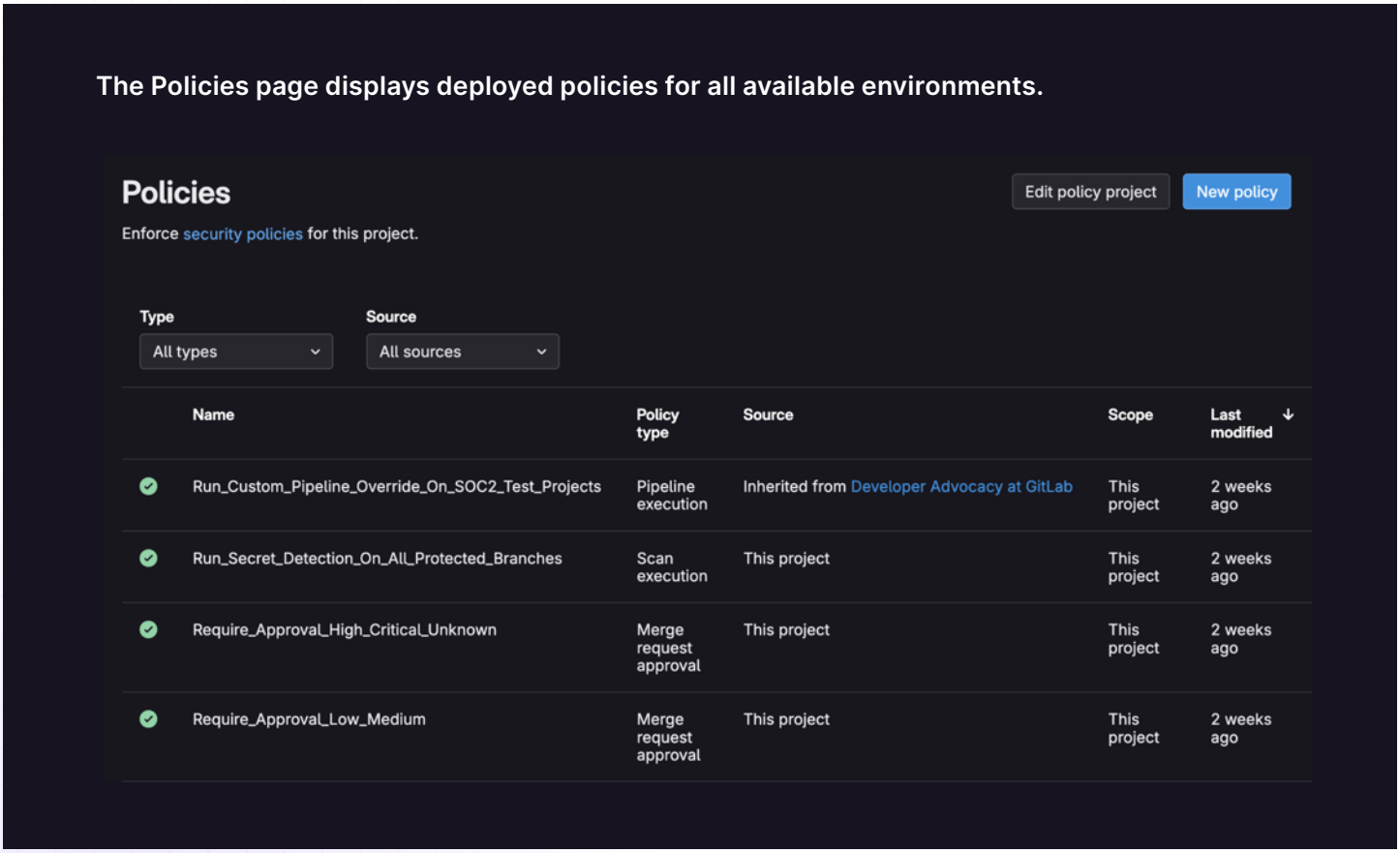
Implementing secure guardrails

Organizations leveraging GitLab can tailor security controls to safeguard critical interests. GitLab's security tools, like security policies, branch protections, and Code Owners, align with industry standards such as NIST, FedRAMP, and FISMA, ensuring code repository control, compliance, and expert oversight for secure software development following federal cybersecurity guidelines.

Security policies enforce **the Principle of Least Privilege** (PoLP) for access control. This principle, developed in the 1970s by the cybersecurity community, advocates granting minimal access to individuals or systems to reduce risks of unauthorized access and potential damage during security breaches. This approach helps enhance overall security posture, minimizes exposure to vulnerabilities, and ensures that each user or system has precisely the permissions required for their designated responsibilities.

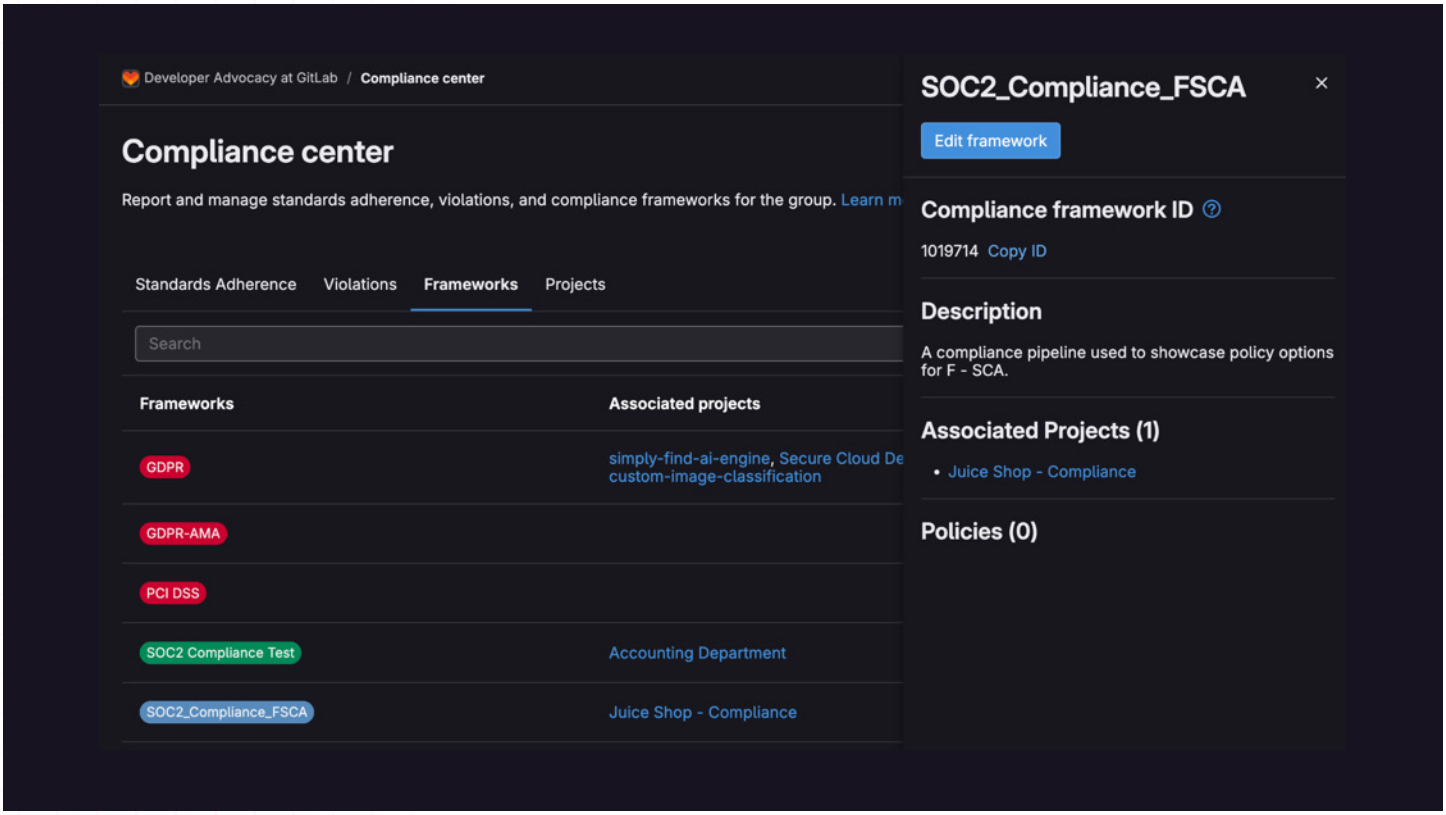
Three critical security policy types, **scan execution policies, merge request approval policies**, and **pipeline execution policies**, empower project administrators to maintain scan integrity and prevent insecure code merges without proper approval. By implementing these policies, organizations can restrict developers from bypassing security measures or merging unsafe code, reinforcing the PoLP. These policies, managed in the Security Policy Project (SPP), ensure distinct permissions and reinforce access control separation. GitLab's branch protections offer additional controls on specific branches by regulating interactions on designated branches, controlling merge access, and pushing capabilities effectively.

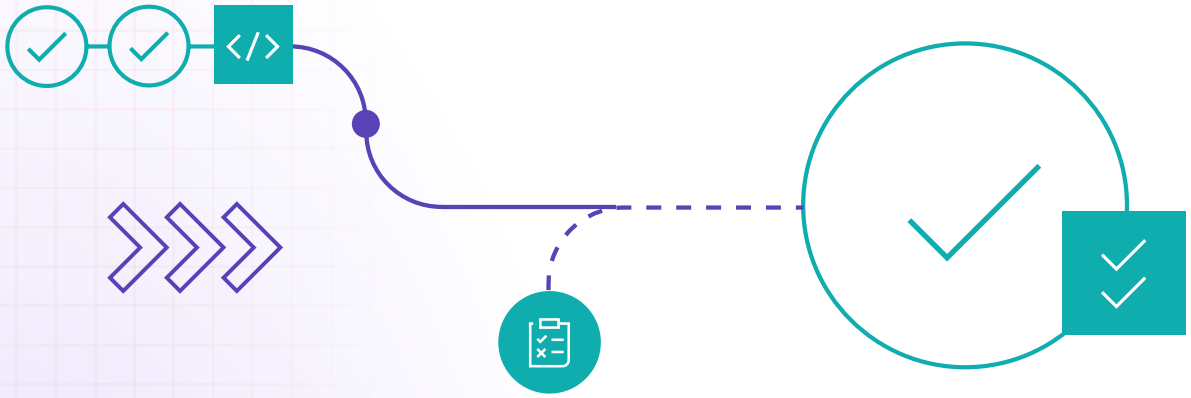
The Policies page displays deployed policies for all available environments.



Enforcing compliance frameworks

GitLab's **pipeline execution policies** offer developers a structured approach to managing compliance requirements, enforcing regulatory standards, and promoting consistency. Projects that must adhere to specific compliance mandates can be identified and monitored using pipeline execution policies and the **GitLab Compliance Center**. This systematic approach guarantees that all code changes undergo thorough automated testing with GitLab's application security features. This enables developers to address security vulnerabilities early in the development cycle, preventing last-minute complications before deployment. By enforcing pipeline execution policies, organizations can rest assured that each code change is rigorously tested with GitLab's **application security features**, which incorporate dependency checks and assessments of vulnerabilities in application and infrastructure configurations.



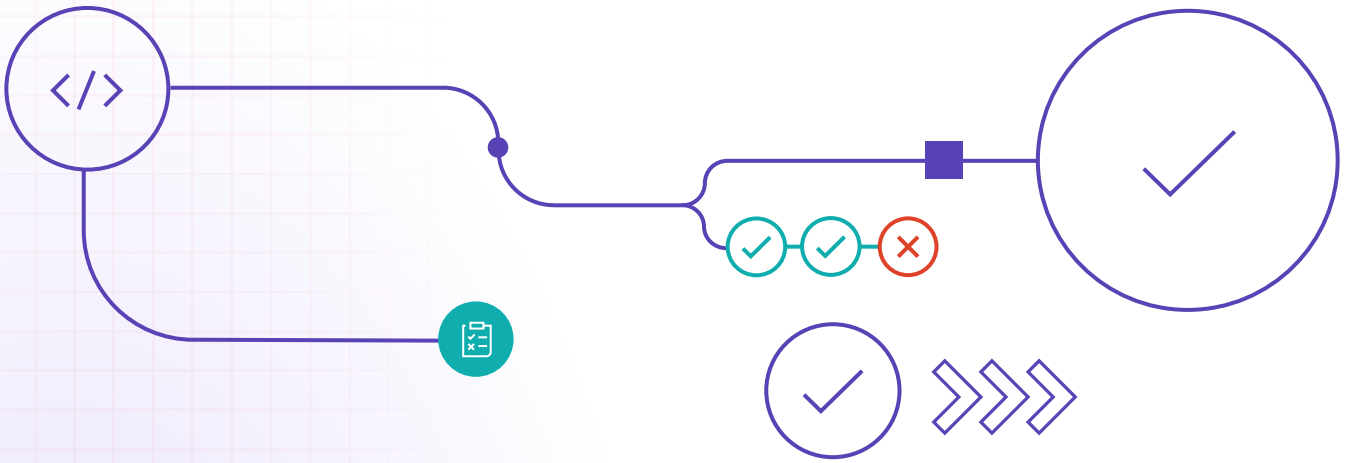


Custom roles and granular permissions

In DevSecOps, role-based access control (RBAC) is a key best practice. Rather than assigning permissions to individual users, this approach involves grouping individuals into roles, each with its designated set of permissions. By structuring access control in this manner, organizations can more easily regulate and monitor who has access to specific resources within the system.

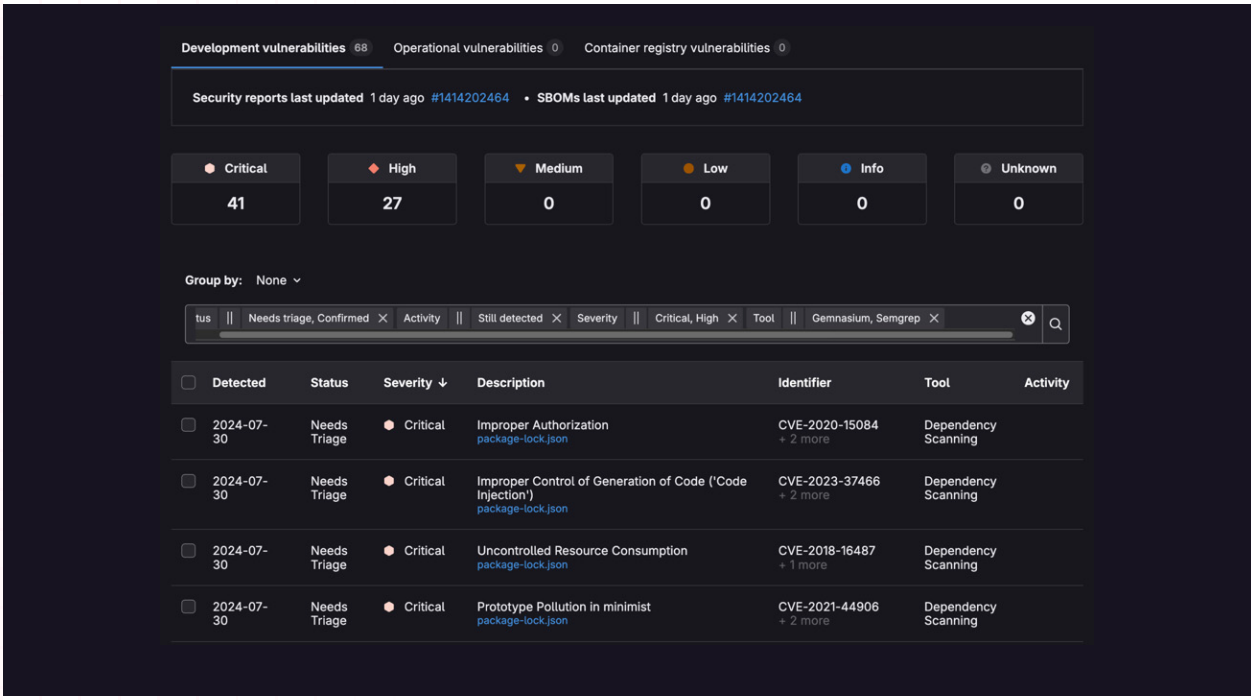
GitLab's **custom roles and permissions** allow an administrator to add a user to a project or group and assign them a specific role. This role dictates the actions the user can perform within GitLab. Different permissions may be applied for **group members**, **project members**, and in **project features**. For instance, a base role can define a user's permissions, with variations for group and project members, as well as project features. The roles eligible to access the vulnerability report are detailed in a table. Notably, the guest role offers limited permissions. For example, a contractor may be assigned a guest role but be granted additional permissions, such as those required for security audits. The administrator can copy that guest role and add additional functionality, like Security Auditor, to it. Once granted, the contractor can review the vulnerability report to assess security risks, though their abilities to interact with the system, like editing code or merging changes, may be restricted.

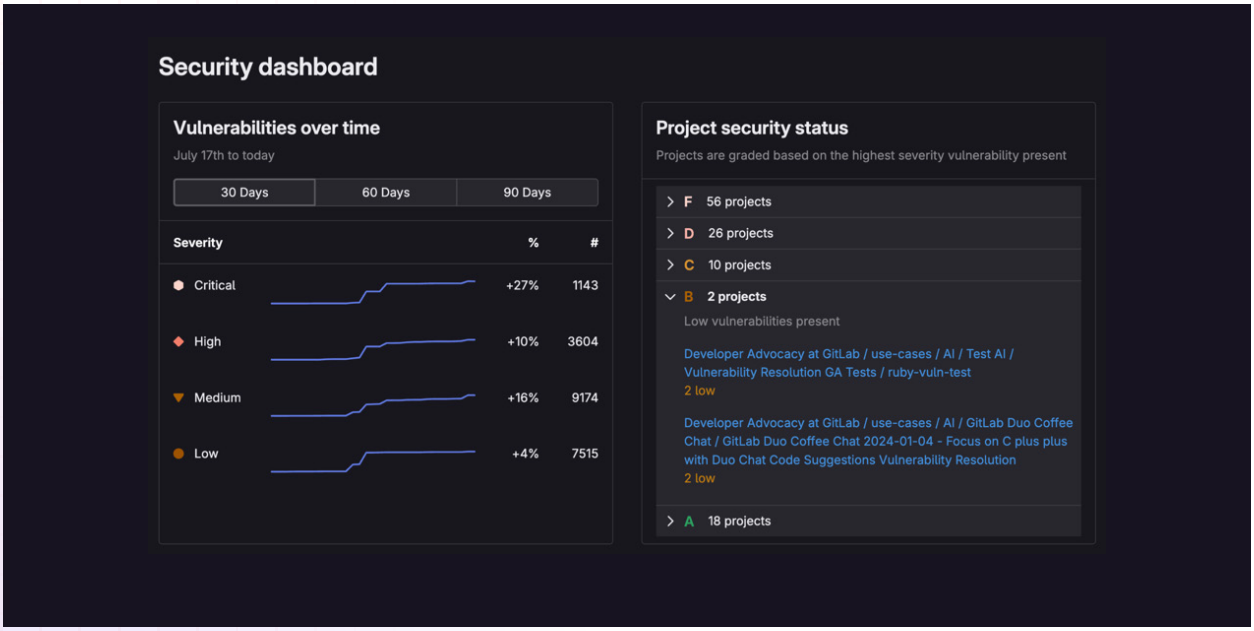
Base role	Can view vulnerability report
Guest	X
Guest + Security Auditor	✓
Reporter	X
Developer	X
Maintainer	✓
Owner	✓



Vulnerability scanning and management

Vulnerability scanning plays a crucial role in supporting compliance frameworks like NIST, FedRAMP, and FISMA by proactively identifying and addressing security weaknesses within systems. By conducting thorough vulnerability scans, organizations can align with the stringent security requirements set forth by these regulatory standards, ensuring robust protection of sensitive data and compliance with industry guidelines. GitLab bolsters supply chain security through comprehensive scanning of source code, containers, dependencies, and running applications for vulnerabilities. Offering a complete security scanner suite (SAST, DAST, container scanning, SCA), GitLab provides strong defense against evolving threats. GitLab has heightened SAST accuracy, minimizing false positives and enhancing proactive risk management.

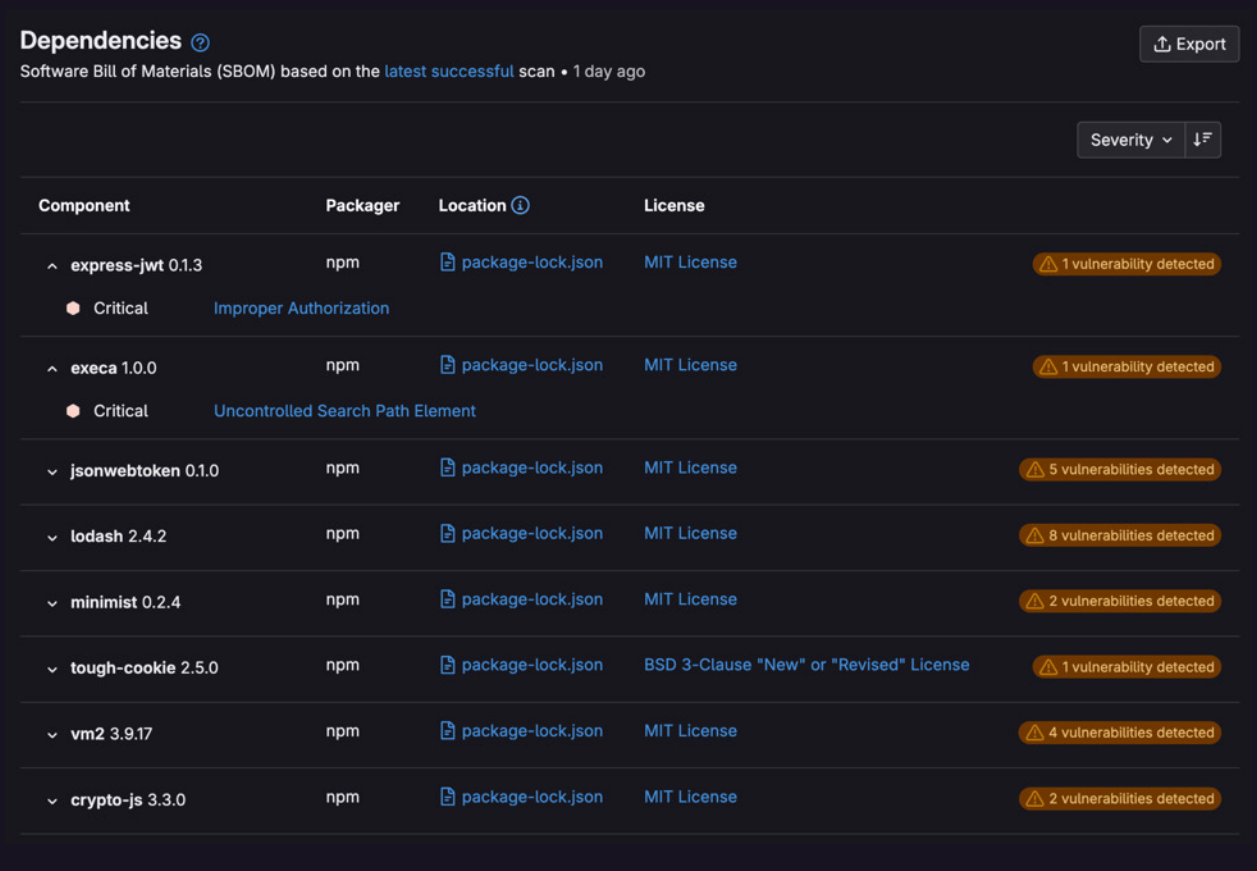




Dynamic SBOMs

Software supply chain visibility is enhanced with GitLab's dynamic software bill of materials (SBOM) capabilities, enabling in-depth security transparency for both proprietary and open-source software. **SBOMs** are intricate, machine-readable inventories detailing the ingredients of software components. Beyond the components, they encompass important details about the libraries, tools, and procedures involved in creating, compiling, and launching a software product. While SBOMs equip DevOps teams with the ability to identify vulnerabilities, assess potential risks, and then mitigate them, a surprisingly low number of organizations report using them today. In **GitLab's 2024 Global DevSecOps Report**, only 21% of organizations reported using SBOMs to enable security in the software development lifecycle, exposing organizations to security risks. To learn more about GitLab's SBOM capabilities, GitLab's **Ultimate Guide to SBOMs** highlights this in greater detail. GitLab has made SBOMs an integral part of the software supply chain roadmap and continues to improve upon its **SBOM capabilities** within the DevSecOps platform.

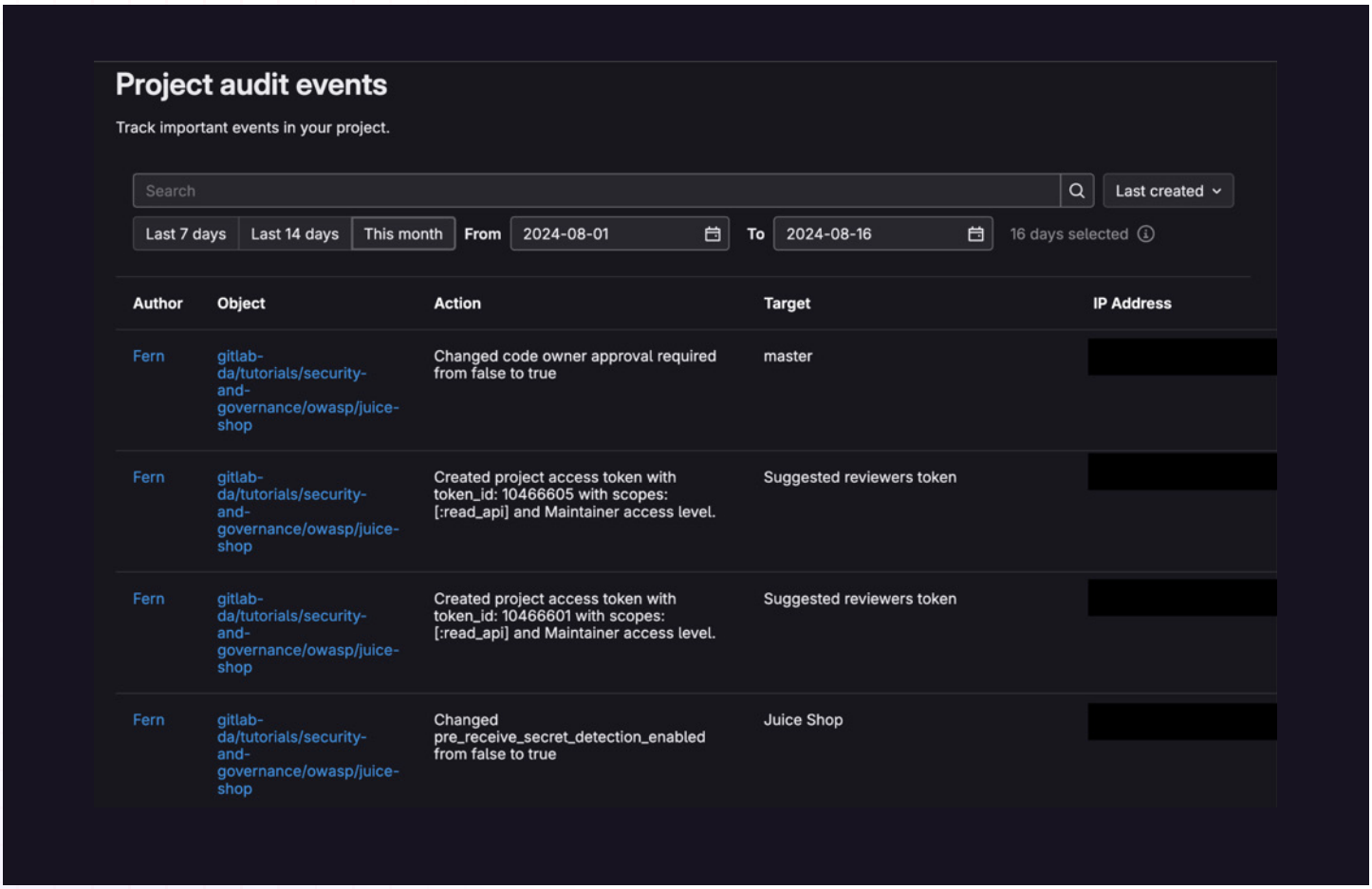
View SBOM dependencies through GitLab's Dependency List



Audit events

Audit logs serve as the vigilant watchdog of organization information systems, documenting all system events, including log-on attempts, file access, and network connections. The ability to trace back the sequence of changes made by a specific user or on a particular day is key to demonstrating adherence to regulatory standards. GitLab's **audit events** enable teams to track every significant event and pinpoint who performed it and when, offering proof of compliance to auditors or regulators. Detailed reports can be generated from audit events using audit reports for showcasing compliance.

The **compliance center** serves as the hub for compliance teams to oversee adherence to standards, report violations, and establish compliance frameworks for their group. The compliance center hosts a range of reports, from the **violations report**, which shows a high-level view of merge request activity for all projects in the group, to the **compliance frameworks report**, which enables you to see the compliance frameworks in the group. All of these reports give an overall view of the organization's compliance posture.



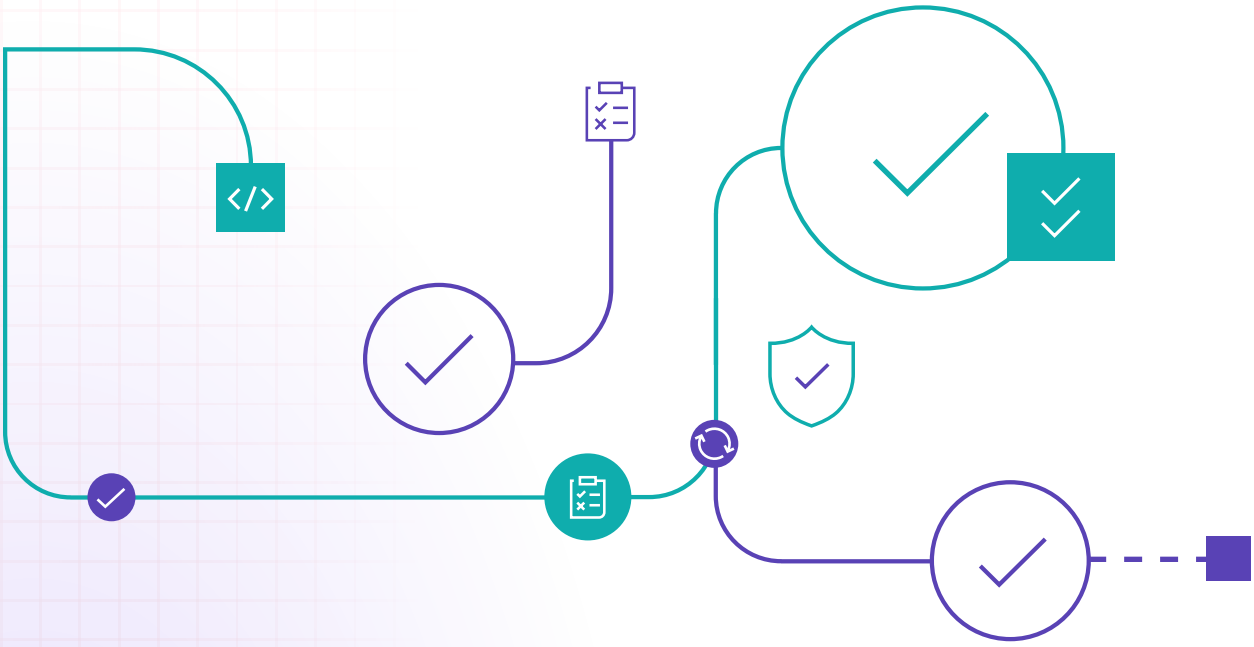
Provenance and signing features to meet SLSA standards

GitLab Continuous Integration and Continuous Delivery (CI/CD) offers a comprehensive suite of features that empower customers to meet stringent compliance and Supply Chain Levels for Software Artifacts (SLSA) standards. By leveraging GitLab's capabilities, organizations can streamline their DevSecOps processes, ensuring that every build, artifact, and release meets regulatory and security requirements.

Key Features:

1. **Build output generation via Sigstore integration:** GitLab CI/CD integrates seamlessly with Sigstore, a trusted platform for signing and verifying software artifacts. This integration ensures that all build outputs are cryptographically signed, providing a verifiable chain of custody for software artifacts.
2. **Provenance metadata generation:** GitLab enables the automatic generation of provenance metadata for every build artifact. This metadata includes detailed information about the build environment, dependencies, and processes, ensuring transparency and traceability in the software supply chain.
3. **Release evidence generation and publishing:** With GitLab CI/CD, generating and publishing release evidence is straightforward. This feature captures critical information about each release, including test results, security scans, and code reviews, providing a comprehensive audit trail that demonstrates compliance with industry standards and regulations.

By utilizing these robust features, GitLab CI/CD enables organizations to maintain a secure and compliant software development lifecycle, meeting SLSA standards and ensuring the integrity and trustworthiness of their software products.



Conclusion

Compliance frameworks set the groundwork for organizations to establish critical security measures, but do not fully address the unique complexities exploited by attackers. While these frameworks provide valuable guidance, they often necessitate a deeper level of specificity to effectively address the vulnerabilities exploited by attackers. NIST even **acknowledges** that compliance controls are the bare minimum of security controls. Agencies that are successful in navigating these obstacles look to unified platforms and data repositories that streamline software development workflows, promote collaboration, and enhance traceability and auditability, ultimately leading to more efficient and secure operations in the ever-evolving landscape of security and compliance. Ensuring compliance isn't just a one-time deal; it's an ongoing effort. By using the right security tools to verify compliance, you can stay ahead of potential risks. It's essential to adapt and improve processes as threats evolve.

Let us Help

GitLab can help you bring AI into your organization responsibly, safely, impactfully, and effectively. This guide scratches the surface of our experience and understanding of working with organizations of all sizes on their AI use cases and needs.

Start a demo today, and see how we can be a partner in your organization's AI journey.


GitLab
Software.
Faster.