WOOLLEMIA INFOSEC

Case Study

DevSecOps Implementation for a Multi-Team Cloud-Native Engineering Organization

Industry	Enterprise Software Development
Engagement Type	DevSecOps Program Design & Implementation
Environment	Cloud-Native Microservices with Multi-Team CI/CD Pipelines (GitLab CI / Jenkins, Kubernetes)
Technology Landscape	SAST, DAST, SCA, Container Image Scanning, Infrastructure as Code Scanning, Secrets Detection, Security Gates in CI/CD
Core Services Delivered	DevSecOps Maturity Assessment, Pipeline Security Tooling Integration, Security Gate & Policy-as-Code Design, Secrets Management Implementation, Developer Enablement & Training

Executive Summary

As engineering organizations scale and adopt continuous delivery practices, security testing performed only at the end of a release cycle can no longer keep pace with the speed of deployment. Embedding security directly into the development lifecycle, commonly known as DevSecOps, has become essential for organizations that need to move quickly without accumulating unmanaged risk.

An enterprise software development organization engaged Woollemia Infosec to design and implement a DevSecOps program across its multi-team, cloud-native engineering environment, integrating automated security testing directly into existing CI/CD pipelines without significantly slowing developer velocity.

Woollemia Infosec conducted a structured maturity assessment of the client's existing development and deployment practices, designed a phased security tooling integration strategy, implemented automated security gates across the pipeline, and supported developer enablement to ensure the new practices were genuinely adopted rather than bypassed.

The engagement enabled the client to shift security testing earlier into the development lifecycle, reduce the volume of vulnerabilities reaching production, and build a sustainable model for ongoing security ownership within engineering teams.

Client Overview

The client is an enterprise software organization operating a cloud-native microservices architecture, with multiple engineering teams independently building and deploying services through shared CI/CD pipelines built on GitLab CI and Jenkins, deploying to a Kubernetes-based production environment.

Prior to this engagement, security testing was largely performed by a separate security team late in the release cycle, creating bottlenecks, delayed feedback to developers, and inconsistent security coverage across different engineering teams and services.

As the organization continued to scale its engineering headcount and release frequency, it recognized the need to embed security testing directly into its development workflows rather than relying solely on periodic manual reviews.

Business Challenge

The client needed to introduce automated security testing across a large number of independently operated pipelines without creating a one-size-fits-all process that would be resisted or bypassed by engineering teams with different technology stacks and release cadences.

The organization also needed to address inconsistent secrets management practices across repositories, a lack of standardized dependency and container image scanning, and limited visibility into security findings across its many microservices.

Because security ownership had previously sat almost entirely with a separate security team, the client needed to build genuine security awareness and ownership within engineering teams themselves, rather than reinforcing a model where security was seen as someone else's responsibility.

Any implementation also needed to avoid introducing excessive pipeline friction that would slow deployment frequency, since rapid, reliable delivery was a core part of the organization's engineering culture.

Our Understanding

Woollemia Infosec recognized that a successful DevSecOps implementation depends as much on developer experience and organizational buy-in as it does on selecting the right tooling. Poorly tuned security gates that generate excessive false positives or block deployments unnecessarily are quickly disabled or bypassed by engineering teams under delivery pressure.

Our team understood that the client needed a phased, pragmatic rollout that started with high-confidence, low-friction checks before progressively introducing stricter security gates, giving engineering teams time to build trust in the new tooling and adapt their workflows accordingly.

Our Solution

DevSecOps Maturity Assessment

The engagement began with a structured assessment of the client's existing CI/CD pipelines, tooling, secrets management practices, and security ownership model across its various engineering teams, establishing a baseline maturity level and prioritized improvement roadmap.

Pipeline Security Tooling Integration

Our consultants designed and implemented integration of static application security testing, software composition analysis, and container image scanning directly into the client's existing GitLab CI and Jenkins pipelines, selecting tooling appropriate to each team's technology stack.

Security Gate and Policy-as-Code Design

The team designed a phased approach to security gates, beginning with informational findings surfaced to developers, then progressively introducing blocking gates for high-confidence, high-severity findings, implemented through policy-as-code to ensure consistent enforcement across pipelines.

Infrastructure as Code and Container Security Scanning

Our consultants integrated infrastructure-as-code scanning and Kubernetes manifest security checks into the deployment pipeline, catching misconfigurations before they reached the production cluster.

Secrets Management Implementation

The engagement implemented automated secrets detection within source repositories and pipelines, alongside migration guidance toward a centralized secrets management solution, reducing the risk of credentials being committed to version control.

Centralized Security Findings Visibility

Our team implemented centralized aggregation and triage of security findings across all pipelines, giving both engineering teams and security leadership consistent visibility into vulnerability trends across the organization's services.

Developer Enablement and Training

Recognizing that tooling alone does not create secure development practices, Woollemia Infosec delivered targeted training sessions for engineering teams on interpreting security findings, secure coding practices relevant to the client's technology stack, and how to work effectively within the new pipeline security gates.

Phased Rollout and Continuous Tuning

The implementation was rolled out in phases across engineering teams, with continuous tuning of tooling configuration based on false positive rates and developer feedback, ensuring the program remained sustainable rather than being perceived as a deployment obstacle.

Business Value Delivered

The DevSecOps implementation enabled the client to identify and remediate a significant proportion of security issues earlier in the development lifecycle, reducing the cost and disruption of late-stage security fixes.

The engagement shifted security ownership meaningfully toward engineering teams themselves, reducing bottlenecks previously created by a centralized, manual security review process.

By carefully phasing the rollout and tuning tooling to minimize false positives, the client achieved meaningful security improvements without a corresponding drop in deployment frequency or developer satisfaction.

Why Woollemia Infosec

Woollemia Infosec designs and implements DevSecOps programs that balance rigorous security testing with genuine developer experience, combining deep expertise in application security, CI/CD architecture, and cloud-native deployment practices.

Rather than imposing rigid, one-size-fits-all pipeline security requirements, we work closely with engineering teams to build a program they will actually sustain and support over time.

Conclusion

This representative engagement demonstrates Woollemia Infosec's ability to design and implement practical, sustainable DevSecOps programs for complex, multi-team cloud-native engineering organizations.

By combining maturity assessment, tooling integration, policy-as-code security gates, and developer enablement, we helped the client shift security left across its development lifecycle while preserving delivery velocity.