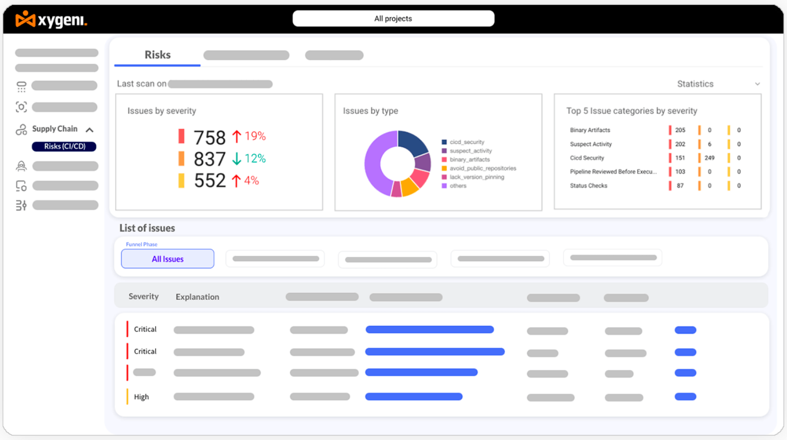
Software Supply Chain

Solution Brief

Secure Your Software Delivery

Xygeni's Software Supply Chain (SSC) Security platform enhances your CI/CD pipelines and infrastructure by integrating robust security measures that protect software workflows from start to finish. It ensures compliance and secures software artifacts against tampering, making your software delivery faster and more secure.



Top 5 Issue categories by severity

Binary Artifacts	205	0	0
Suspect Activity	202	6	0
Cicd Security	151	249	0
Pipeline Reviewed Before Execu...	103	0	0
Status Checks	87	0	0

About Company

Xygeni specializes in enhancing software development security and efficiency with our Application Security Posture Management (ASPM) platform. We offer complete control over application risks, a unified security view from code to cloud, and eliminate noise to prioritize risks effectively. Our advanced malware detection and early warning system makes Xygeni a leader in protecting applications from emerging threats, ensuring rapid and secure software delivery.

The frequency and impact of software supply chain attacks have surged, emphasizing the need for stringent CI/CD security. Recent statistics reveal a staggering **742% increase in such attacks from 2019 to 2022, with forecasts suggesting that 45% of organizations will be affected by 2025**. The financial toll is also expected to rise sharply, with projected annual costs reaching $138 billion by 2031. This escalating threat landscape underscores the critical importance of implementing robust CI/CD security measures.

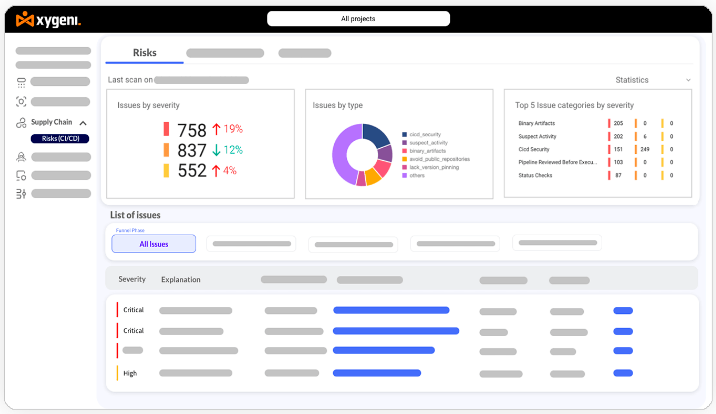
These attacks often leverage vulnerabilities identified in the OWASP Top 10 CI/CD security risks, which are detailed in NIST SP 800-204D. This document provides guidelines for integrating software supply chain security into DevSecOps CI/CD pipelines and emphasizes mitigating risks such as unauthorized code injections, dependency chain abuses, inadequate access controls, and compromised artifacts.

Xygeni incorporates security measures that align with industry standards like OWASP and NIST SP 800-204D to ensure that each CI/CD pipeline phase adheres to the highest security standards and best practices.

60%
of supply chain attacks "took advantage" of customer trust in their supplier

ENISA

Enhance CI/CD Pipeline Security and Coverage



Top 5 Issue categories by severity

Binary Artifacts	205	0	0
Suspect Activity	202	6	0
Cicd Security	151	249	0
Pipeline Reviewed Before Execu...	103	0	0
Status Checks	87	0	0

Xygeni's Misconfiguration Detectors protect your CI/CD pipelines by scanning configuration files, build scripts, and CI job definitions. These detectors identify deviations from security best practices and standards, providing immediate alerts on potential misconfigurations that could lead to unauthorized access or code or pipeline execution compromises. With a robust set of rules based on the latest security advisories, Xygeni ensures every component of your pipeline adheres to the highest security protocols.

Detected issues may include improper settings in package managers, insecure build file or infrastructure configurations, or risky CI jobs or plugins, all of which are notified for rapid correction to maintain the integrity and safety of your software delivery processes.

Automated DevOps Security Scanning

Xygeni enhances DevOps security by easily and flexibly integrating continuous scanning within your CI/CD workflows. This process identifies and addresses potential misconfigurations and vulnerabilities before they affect production. Here's how you can implement Xygeni for automated continuous security scanning:

1. Git Hooks for Immediate Scanning: Integrate Xygeni scanners directly into your Git workflow using pre-commit hooks. This setup scans commits for misconfigurations or sensitive data before they're pushed to the repository. The commit is blocked if critical issues are detected, ensuring that only secure code progresses through your pipelines.

2. Using Pre-commit Frameworks: For a more standardized approach, use frameworks like pre-commit to manage and execute scanning scripts. These frameworks facilitate the installation and updating of hooks, making it easier to maintain and distribute scanning tasks across multiple projects.

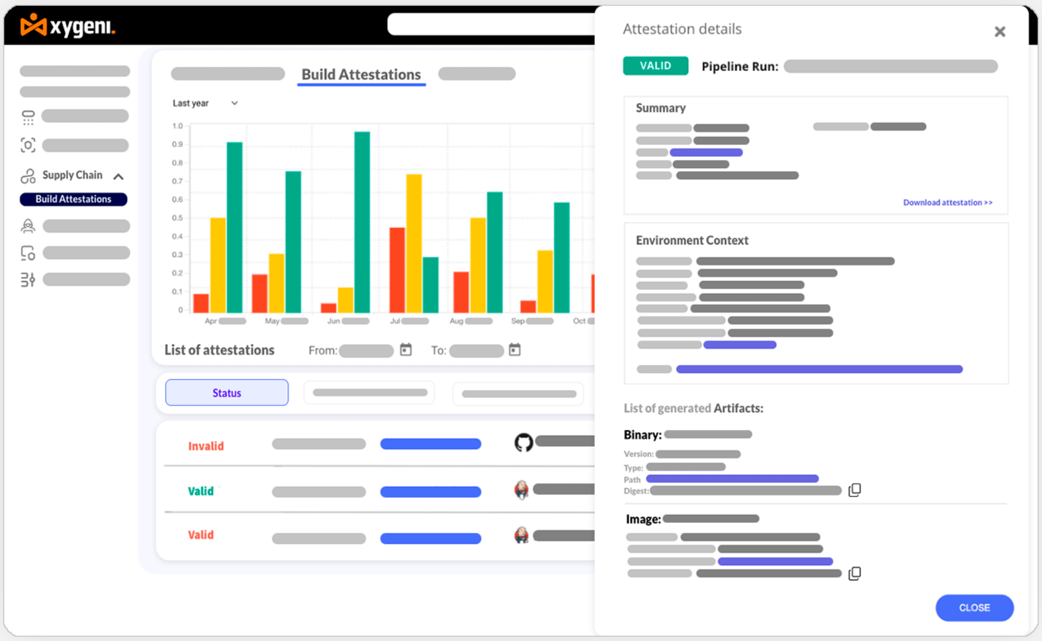
3. Customizable Scanning with Fail Safeguards: Configure Xygeni to align with your team's risk tolerance using the --fail-on option. Set this to 'critical' to halt the CI/CD process when severe threats are detected, or use --fail-on=never to ensure continuous delivery without interruptions, even when issues are found.

3. Implement Guardrails in CI Steps: Incorporate Xygeni scans into CI steps using tools like GitHub Actions. Configure the fail_on parameter to control how builds are affected by detected issues. For stricter enforcement, set fail_on to react to issues according to your custom rules, ensuring only secure builds are deployed.

Secure Build Attestations with SLSA Compliance

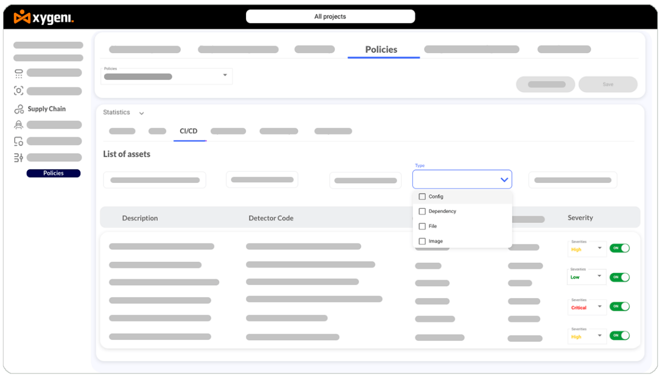
Xygeni improves the integrity of your software builds by generating and managing SLSA provenance and in-toto build attestations. This capability is crucial for ensuring the authenticity of source code, container images, and other software artifacts. Attestations are implemented with a different CLI: SALT.

Xygeni's Software Attestations Layer for Trust (SALT) is a dedicated CLI for creating, storing, and verifying build attestations. This feature aligns with robust security frameworks and provides comprehensive tools for the entire lifecycle of software attestation:



- **Attestation Generation:** Automatically generate attestations during the build process to certify the integrity and origin of software artifacts.
- **Verification and Management:** Easily verify attestations against their respective software artifacts through the Xygeni platform. Manage all attestations centrally to ensure they meet compliance and security standards.
- **Integration and Accessibility**: SALT integrates smoothly into existing CI/CD pipelines and is accessible via a command-line interface (CLI), making it versatile for different operational environments.
- **Compliance with Standards:** Comply with emerging standards like SLSA and NIST SP 800-204D to ensure your builds are secure and verifiable.

Customizable Policies for CI/CD Security



Xygeni's security platform allows you to customize security policies specifically for your organization's needs using a customer-defined YAML file. By specifying the --custom-detectors-dir option when running the xygeni misconf command, you can direct the scanner to use your customized security policies stored in the specified directory. This flexibility ensures that your CI/CD pipelines are protected according to both the general security best practices and the specific requirements of your business environment.

This approach not only tailors security measures to fit unique corporate landscapes but also adapts dynamically to various regulatory environments, ensuring thorough compliance and optimal security management.

Demonstrate Compliance with SSC Standards

Demonstrate Compliance with SSC Standards: Xygeni improves your software supply chain's security posture by ensuring your development processes adhere to leading industry standards for compliance. With Xygeni, you can:

- **Automate Compliance Assessments:** Quickly evaluate your projects against comprehensive standards like CIS, OWASP, OpenSSF, or ESF using Xygeni's automated tools. Detailed compliance reports help you understand and address gaps efficiently.
- **Continuous Monitoring and Validation:** Xygeni monitors your development activities, ensuring ongoing compliance and providing real-time insights into your security status.
- **Customizable Compliance Frameworks:** Tailor compliance checks to fit your business needs and regulatory requirements. Xygeni allows for the integration of custom standards, making it flexible enough to adapt to customer-specific compliance demands.

Enforce Least Privilege Approach

Xygeni enhances supply chain security by enforcing least privilege policies and integrating comprehensive Collaborator Analysis. This approach ensures that access rights within the development and CI/CD ecosystems are strictly necessary, reducing insider threats and unauthorized access risks.

Xygeni conducts detailed reviews of the permissions and activity of all SCM user accounts, groups, and git users without SCM accounts who contribute to repositories. This analysis helps identify inactive or overprivileged users and assesses the risks they may introduce based on their access levels and activities.

Our Health Check section lists these overprivileged and inactive users for quick identification. This section allows security teams to raise tickets with the pre-filled necessary information for prompt analysis and remediation, streamlining the process of enforcing security measures within your organization.

Summary of CI/CD detectors

Here's an integration of the supported systems into the summary of key misconfiguration detectors for Xygeni:

CI/CD Security Detectors:

- Enforce appropriate permissions and secure configurations across CI/CD tools and workflows, specifically tailored for platforms like GitHub, GitLab, Azure DevOps, Bitbucket, CircleCI, and Jenkins.
- Verify the integrity of build processes and prevent unauthorized code or pipeline changes, with specific checks for each platform to ensure compliance and security.
- Monitor for unusual activities and ensure encrypted storage and handling of secrets across all supported CI/CD platforms.

Container and Dependency Management:

- Apply best practices in container configurations, such as avoiding running as root and ensuring secure file operations across various CI environments, including Jenkins and CircleCI.
- Maintain secure connections by enforcing HTTPS for remote repository access and ensuring proper version control on platforms like GitHub, GitLab, and Azure DevOps.

Compliance and SCM Detectors:

- Support compliance with key standards like CIS, NIST, and OpenSSF, ensuring security policies are adhered to across all integrated platforms.
- Secure source code management practices, including enforcing MFA, signed commits, and code review protocols, particularly on platforms like GitHub, GitLab, and Bitbucket.

General Security Practices:

- Detect and prevent insecure webhook configurations, unprotected branches, and insecure dependencies across all supported systems.
- Monitor and validate security policies, ensuring continuous compliance and signed releases within the integrated ecosystem.

Book Your Demo Now - Transform Your Approach to Cybersecurity!