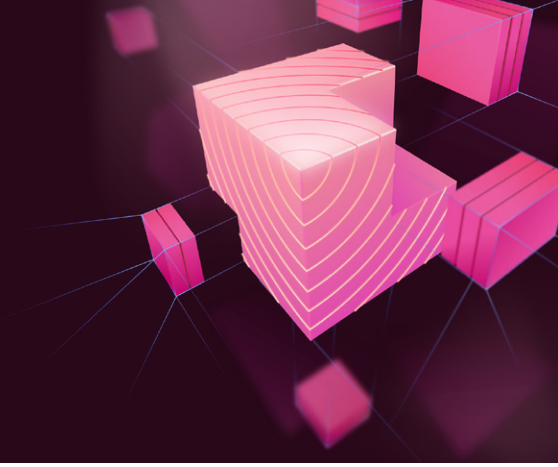
cycode

Code Tampering Prevention

SOLUTION BRIEF



Software is the lifeblood of modern companies, providing the means to interact with customers, deliver value, and transact business. The source code from which that software is created forms the foundation of the enterprise. When source code is tampered with, it not only can interfere with revenue but can also be weaponized to adversely affect customers. Organizations such as Solarwinds, Kaseya, and an increasing number of others have been affected by code tampering, and the consequences are often front page news. Attackers find code tampering and other software supply chain attacks attractive because they are a force multiplier that provides access to not only the software company that is breached, but also their customers.

Code tampering is a multifaceted problem that is forcing organizations to push beyond the limitations of existing security and development practices to eliminate silos and harden the entire software development process. Today's software supply chains, which include code, dependencies, development and runtime infrastructures, have attack surfaces that are so vast and interconnected that point solutions and siloed approaches simply cannot provide comprehensive protection.

Establish a Foundation of Trust

Good security hygiene creates a foundation of trust in the development process itself. For example, by establishing policies such as mandatory MFA to ensure actors are who they claim to be, along with mandatory commit and artifact signing, then ensuring these are enforced consistently and automatically across all of the tools used in the SDLC, teams can gain trusted visibility into code and artifact provenance.

Recent attacks like SolarWinds have demonstrated the importance of verifying not only that components are properly signed, but that the signing authority properly validates the provenance of the component being signed. Otherwise, it is possible for attackers to tamper with code as it is built, and leverage automated signing processes to hide their malware.

Cycode orchestrates a suite of tools, working in concert across each phase of the SDLC to enforce consistent security policies and establish governance to build a foundation of trust.

Validate Integrity in Every Step of the SDLC

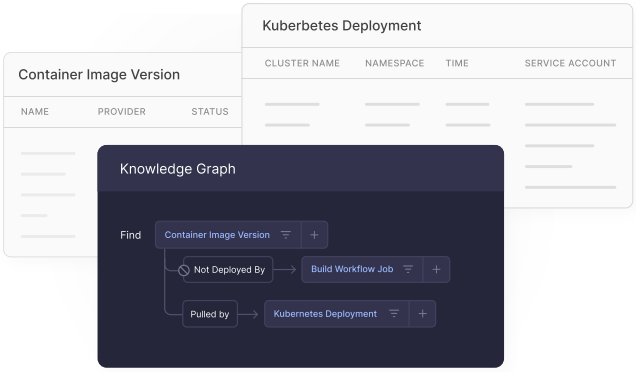
It is also important to validate that code and components are flowing through the development process as intended; that inputs and outputs match across all the interconnections within your software delivery pipeline. This helps ensure code has not been maliciously altered along the way.

Cycode can ensure your repository requires signed commits, help you trace code and artifacts through the SDLC, identify when the actual configuration of production infrastructure has drifted away from infrastructure as code (IaC), and validate many other

handshakes across your SDLC. These capabilities enable users to recognize when code tampering has occurred.

It is important to validate these handshakes between all phases of the SDLC to ensure a complete chain of custody for software artifacts. Any gaps in that chain present an opportunity for attackers to tamper with the code or artifact without being detected. Validating the chain of handshakes includes validating that each handshake is successful, and that all the necessary handshakes were actually performed.

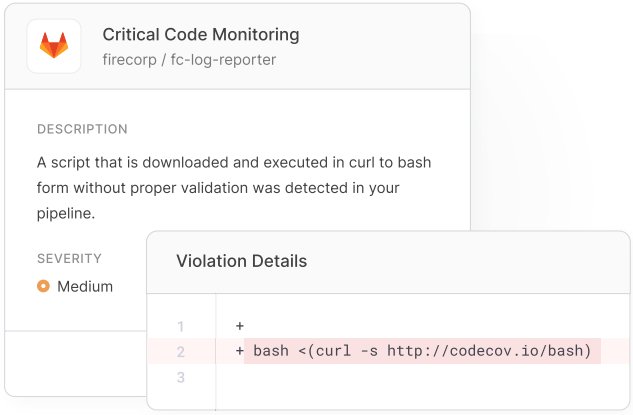
Cycode leverages its Knowledge Graph to consolidate metadata that emerges out of the SDLC such as who committed which code, how components are built, which handshakes were performed, their results, which users triggered each action, and more. The Knowledge Graph can serve as an audit trail of the development process so that users can validate integrity in real time and also understand the downstream effects of decisions made earlier in the development process.



Monitor Critical Code and Security Controls

Certain types of code such as build rules, branch protection rules, CI/CD settings and infrastructure as code should always be changed deliberately and with an extra level of scrutiny given the security, integrity and governance implications. Unsanctioned changes to SDLC tooling such as altering security controls, may be a sign of nefarious activity.

Cycode monitors critical areas of code such as those that control security policy and pipeline configurations, and upon any change delivers automated alerts to designated team members so that malicious changes and unintended consequences can't slip through.



Detect Anomalies

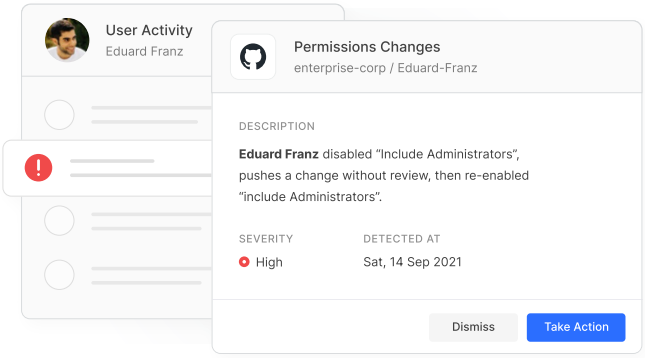
Protecting against the misuse of authorized user credentials, either by malicious insiders or attackers impersonating legitimate users, is notoriously difficult. Often, such misuse is only discovered after the damage is done.

To address these threats, Cycode learns the normal patterns of your SDLC systems, processes, and user behavior, so it can highlight anomalies in order to prevent accidents, foil attackers, and stymie insider threats.

Anomalies may be as simple as a suspicious repository configuration change, unusual clone or commit activity, or bypassed branch protection or status checks. Or as complicated as multiple compromised developer accounts being used to peer review pull requests.

Not all anomalous behavior is malicious, so it is also important to recognize unusual but legitimate behavior in order to avoid false alarms. Cycode leverages the metadata in the Knowledge Graph to understand the broader context of activity and events happening in the

SDLC to focus on anomalous behavior that increases security or compliance risk.



Complete Software Supply Chain Security

Unlike traditional AppSec, which focuses on code, preventing code tampering must consider events and behavior across the entire development process. Cycode's Knowledge Graph connects data points about code integrity, user activity, and events across the SDLC to draw attention to anomalies and prevent code tampering.

Preventing code tampering is a necessary step toward securing your software supply chain but only addresses one dimension of risk. Software supply chain attack surfaces are so vast and interconnected that organizations need a comprehensive solution that covers all dimensions of SDLC risk.

The Cycode platform includes a collection of scanning engines that identify many important dimensions of software supply chain risk such as hardcoded secrets, code leaks, IaC misconfigurations and more. These engines augment the Knowledge Graph with a rich understanding of context, normal user activity and events, enabling Cycode to prioritize risk, find anomalies, and effectively secure the software supply chain.

Cycode integrates with all software delivery pipeline tools and infrastructure providers to implement consistent governance and security policies, harden security posture, and provide visibility across the entire SDLC. Pre-built integrations are easily deployed. With just a few simple clicks, organizations are able to realize immediate value and maximize their agility as new tools are added to the SDLC.

Cycode is the only end-to-end software supply chain security solution that provides visibility, security, and integrity across all phases of the SDLC to effectively prevent code tampering.

Cycode is a complete software supply chain security solution that provides visibility, security, and integrity across all phases of the SDLC. Cycode integrates with DevOps tools and infrastructure providers, hardens their security postures by implementing consistent governance, and reduces the risk of breaches with a series of scanning engines that look for issues like hardcoded secrets, infrastructure as code misconfigurations, code leaks and more. Cycode's knowledge graph tracks code integrity, user activity, and events across the SDLC to prioritize risk, find anomalies, and prevent code tampering.

More solution briefs at Cycode.com