Applying Governance to CI/CD

By: Tiffany Jachja . Tech Evangelist

Introduction

In the SolarWinds hack of 2020, FireEye, a cybersecurity vendor, announced an intrusion that resulted in the theft of over 300 proprietary security tools offered by the vendor to their customers. SolarWinds, an IT monitoring vendor, had pushed malicious code to over 18,000 customers, including Fortune 500 companies and large federal agencies. Those affected included organizations such as the U.S. Department of Homeland Security (DHS), Microsoft, and NASA. Following this announcement, security experts launched an investigation into the hack, putting the Software Development Life Cycle (SDLC) and all its processes under review.

This eBook will share how to apply IT Governance principles and practices to reduce and mitigate software risks.

What Role Did CI/CD play in the SolarWinds hack?

In the SolarWinds hack, attackers legitimized their software malware by injecting it into the SolarWinds build process. This build pipeline produced trusted software artifacts, obtaining digital certifications before being released to over 18,000 customers worldwide. After the hack, SolarWinds took security-related action, further restricting access to its build environment and reviewing its build process, but not before it was too late.

In 2021, investigations are still finding the impacts of the SolarWinds hack.

What is CI/CD Governance?

CI/CD is a shortened term for Continuous Integration and Continuous Delivery. CI/CD is the combination of principles, practices, and capabilities that allow for software changes of all kinds to get users in a quick, repeatable, and safe manner. CI/CD Governance goes beyond traditional CI/CD, where IT practitioners simply automate delivery without truly mitigating risk.

> *Automated pipeline governance or CI/CD Governance gives enterprises the ability to attest to the integrity of assets in a delivery pipeline.*

How could CI/CD Governance have helped in the SolarWinds hack?

CI/CD Governance is about putting controls into your CI/CD pipeline. The goal is to detect, approve, and track changes made by any user during your SDLC. Here are some practices to consider when reviewing how your CI/CD pipeline ensures security, control, and standards throughout your SDLC.

Secrets Management

In the case of the SolarWinds hack, secrets management in the CI/CD process would have prevented an entry point for hackers. The attacker(s) gained access to SolarWinds through an exposed secret in its GitHub repository. The build process then baked the malicious code into the product that was shipped to customers.

How could CI/CD Governance have helped in the SolarWinds hack? (Cont...)

Secrets Management (Cont...)

Secrets management not only focuses on obscuring, storing, and encrypting sensitive information such as passwords or personal information. It also extends to solutions and tools that can actively monitor secrets usage and access. CI/CD platforms can provide an additional layer of security, enhancing the experience of using tools like HashiCorp Vault, CyberArk or AWS Secrets Manager.

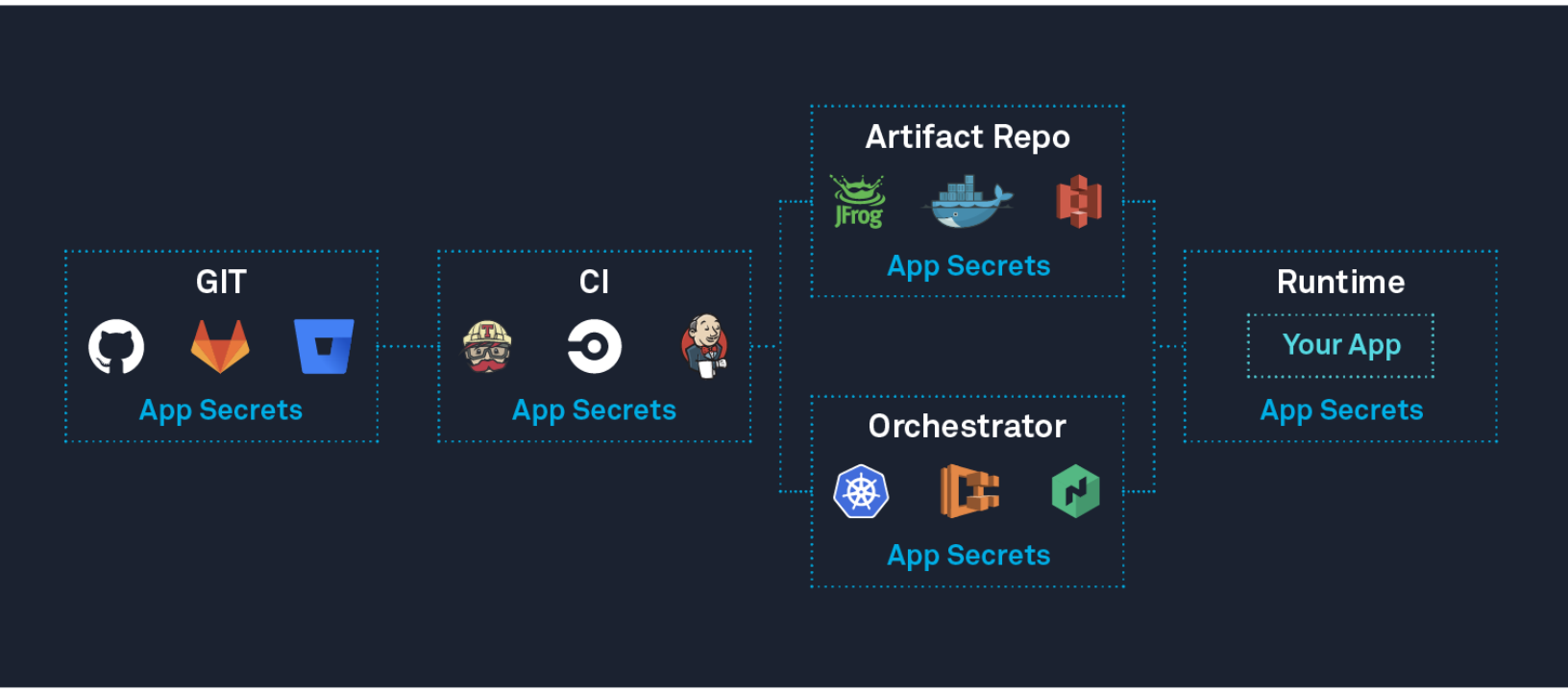
As shown by the illustration below, secrets live in various parts of the SDLC involving solutions, environments, and tooling. To properly secure secrets ensure you cover these foundational areas of your software delivery life cycle. Also consider investing in tools to monitor configuration drift, code integrity, and Git configurations.

Manual Approvals

Some changes can go unmoderated. Change Approval Boards (CAB) mitigate risk by reviewing specific or milestone software changes. A CAB may have prevented the SolarWinds hack or allowed SolarWinds to detect the problem before another customer detected the issue.

Consider defining user groups and users with the ability to approve or reject CI/CD pipeline deployments. This manual quality gate can ensure changes don't go unreviewed. To prevent additional overhead or idle time, approval request notifications and integration with ticketing systems, such as Jira, can provide the needed context for changes.

—
Diagram: Secure Secrets in the Software Delivery Life Cycle



How could CI/CD Governance have helped in the SolarWinds hack? (Cont...)

Role-Based Access Control

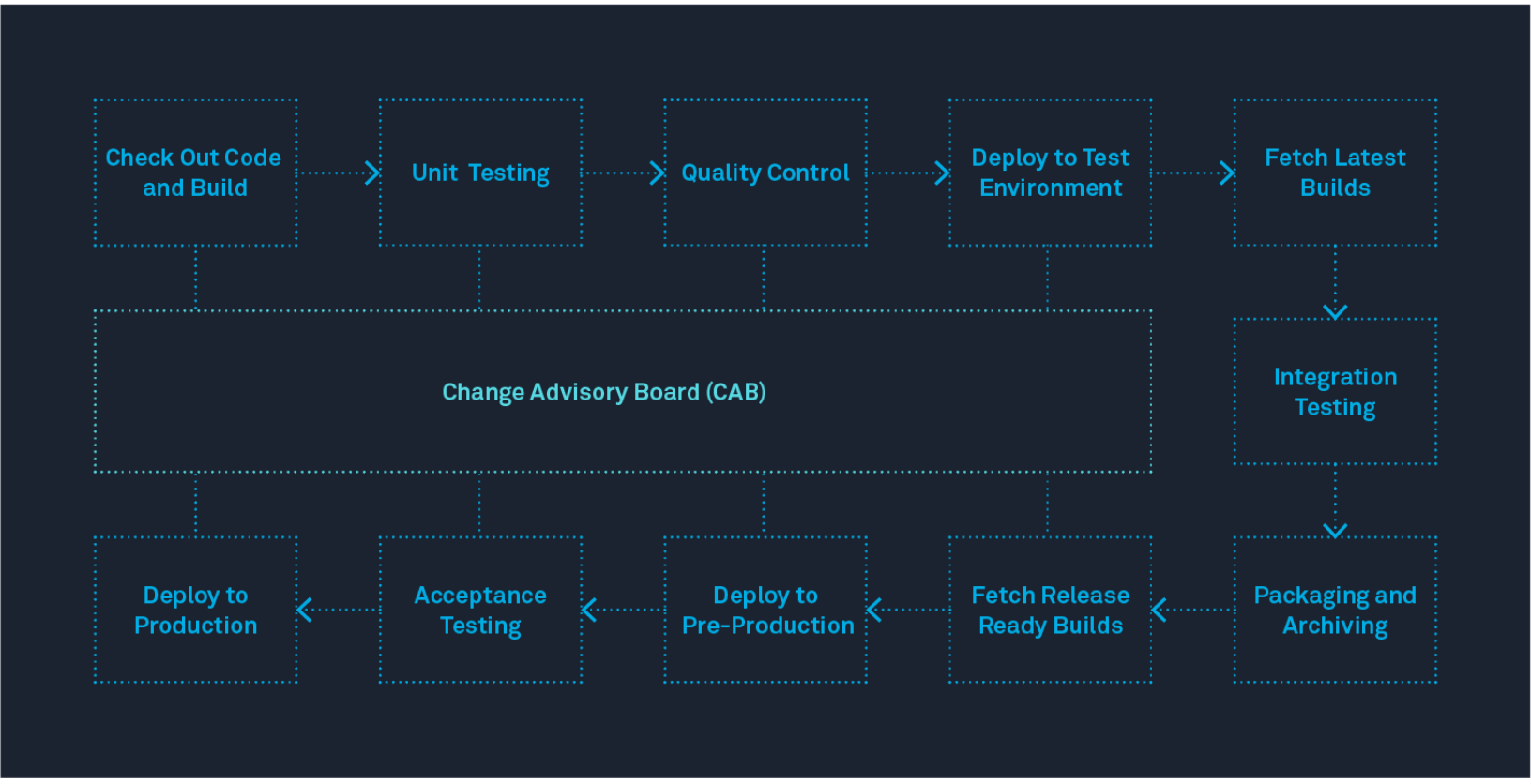
The person who left the shared secret as plain text inside the SolarWinds repository had no intention to cause the SolarWinds hack. However, the results were the same, and attackers exploited the security weakness. This proves that the delivery process has to be secure even from accidents or mistakes.

You'll need to understand and control who has access to your CI/CD pipeline(s) and under what privileges. Role-Based Access Control (RBAC) ensures organizations can manage user and user group permissions. RBAC can apply to granular parts of your CI/CD pipeline, including environments, stages, and workflows. As an example, an organization may decide developers should not have access to production environments. This limits access to clean environments avoiding situations where developers can impact customers utilizing live or running versions of the application.

Vulnerability Scanning

Security vulnerabilities can be present in an application's codebase, its operating system packages, dependencies, and configuration. Common vulnerabilities include missing data encryption, buffer overflows, missing authentication for critical functions, and insecure interactions between software components.

—
Diagram: A Change Approval Process



How could CI/CD Governance have helped in the SolarWinds hack? (Cont...)

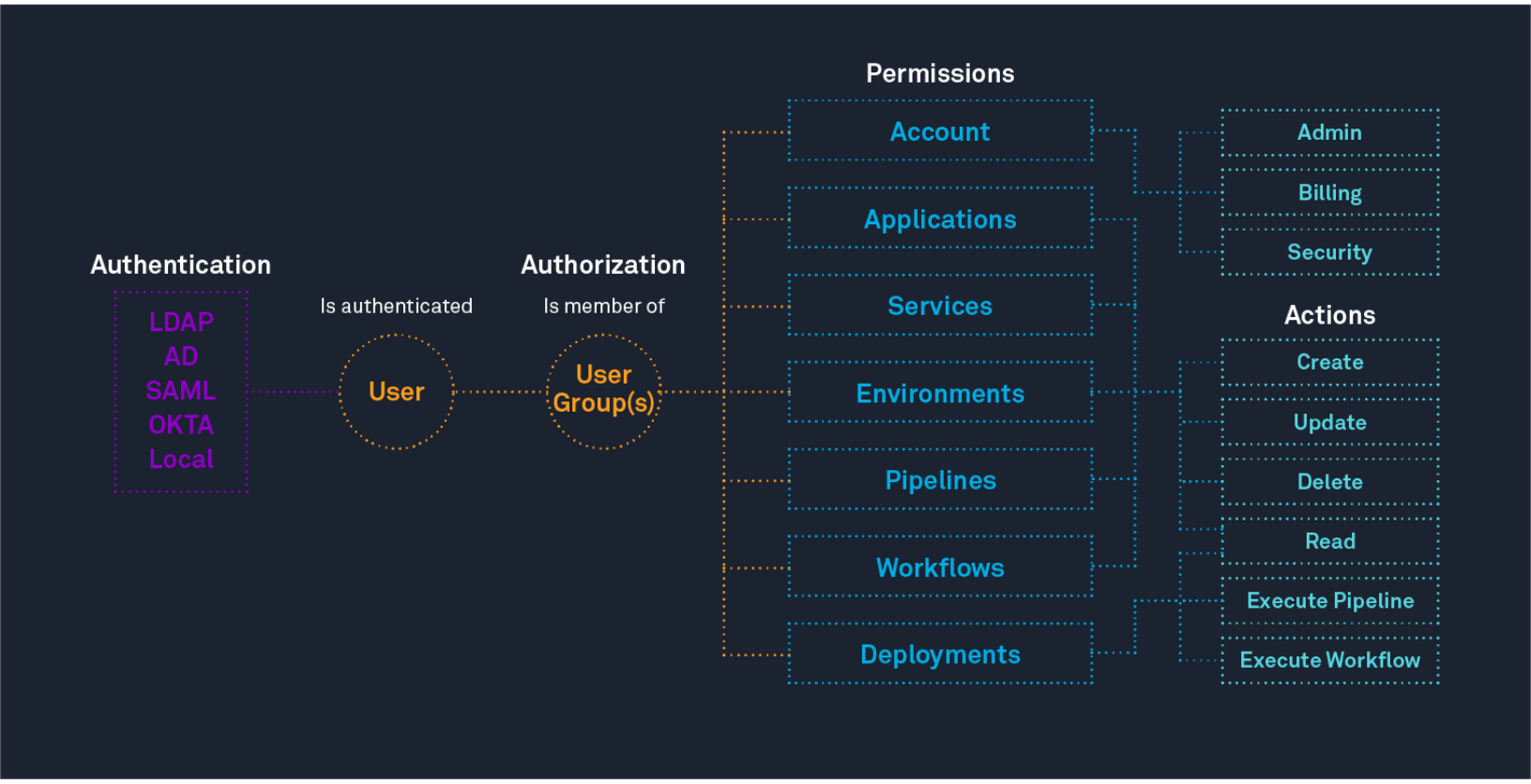
Vulnerability Scanning (Cont...)

There are different risks associated with vulnerabilities. With critical or high-risk vulnerabilities, someone who exploits your software has the potential to impact your organization severely. Risks can involve data breaches that impact not only an organization, but also its customers.

Some vulnerability management techniques include Penetration Testing, Configuration Management, and Application/Artifact scanning.

- **Penetration Testing:** Penetration Testing (or Pen Testing) allows you to identify security vulnerabilities by attempting to break or steal your data through software service.
- **Configuration Management:** Configuration Management involves managing infrastructure configurations and missing patches that leave your software service vulnerable to errors and risks. Some tools provide infrastructure scanning as a service to help detect outdated or misconfigured instances.
- **Container and Application Scanning:** This allows you to detect vulnerabilities in deployable artifacts and running applications. Some tools for container scanning include Twistlock, Clair, and Trivy.

—
Diagram: RBAC on CI/CD Platform and Pipeline Components



How could CI/CD Governance have helped in the SolarWinds hack? (Cont...)

Pipeline Templates

CI/CD pipelines standardize and automate your software delivery process, providing consistency and allowing for repeatable outcomes. Pipeline templates allow you to define one pipeline for any service, infrastructure definitions, and target environment. This pipeline will run for your particular choice of resources by replacing specific configurations with evaluated values through variable functions and expressions.

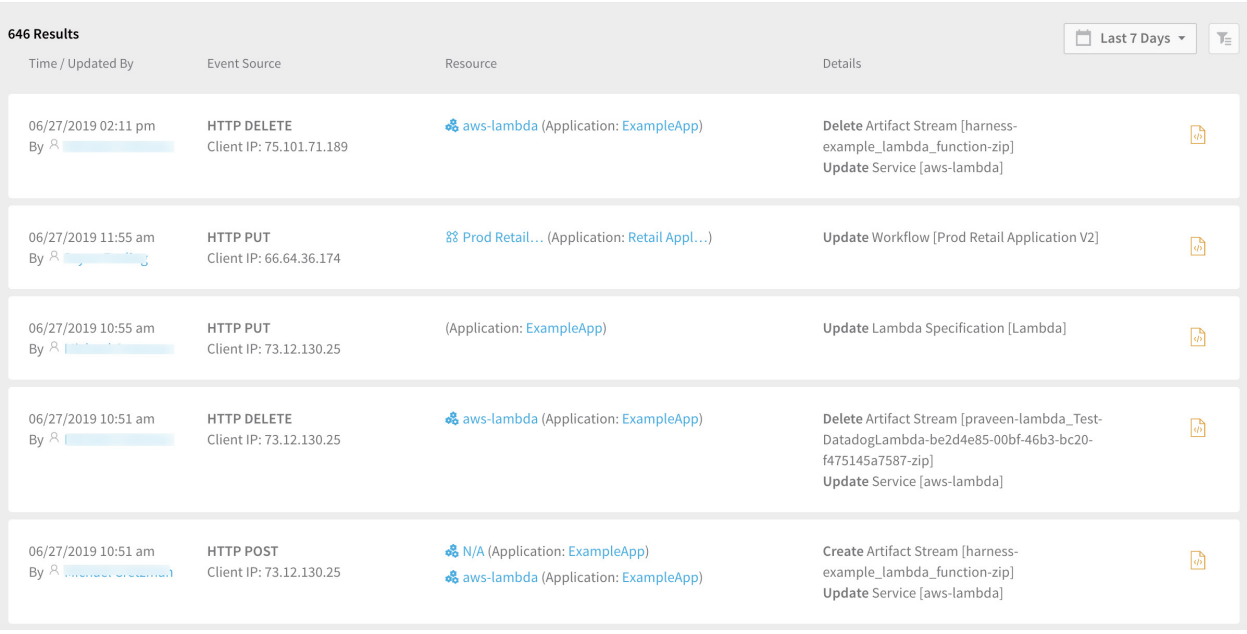
Create standardized, reviewed, and well-understood subcomponents of CI/CD pipelines to scale CI/CD Governance. This ensures that custom CI/CD pipelines consist of compliant, secure, and governed workflows.

Audit Trails

An audit trail provides a catalog of changes and events leading up to a specific date and time. Audits and compliance ensure organizations adhere to a minimum level of capability. CI/CD Governance ensures we can answer who, what, where, when, and why?

Audit trails are a catalog of events related to operational procedures or changes. These events (such as create, update, delete actions) map to software resources like software artifacts and environments. Robust audit trails contain enough information to allow reviewers or auditors to understand what kinds of changes were made to specific applications using pipeline resources, who made those changes, and when.

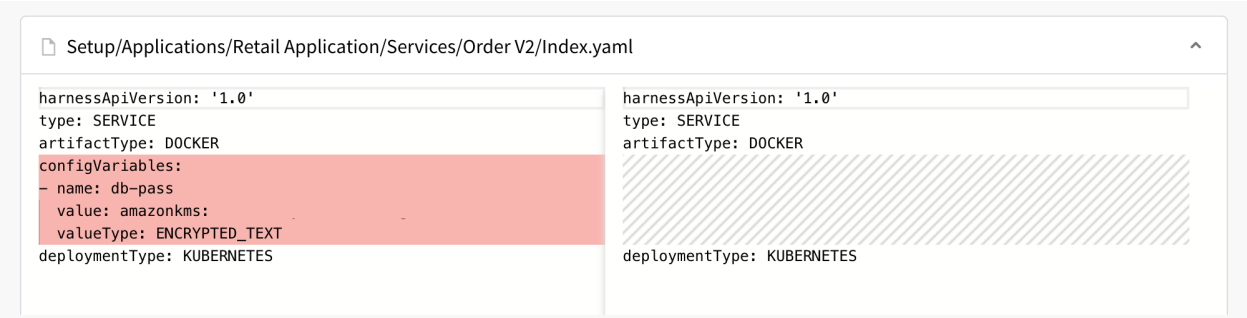
Image: Harness Audit Trail



How could CI/CD Governance have helped in the SolarWinds hack? (Cont...)

In this example, the Harness Audit Trail presents the resulting diff view, indicating deletions and insertions in the underlying pipeline configuration YAML.

—
Image: Leveraging Logs to Review Changes to Service Deployments



Balancing developer speed with control and security?

CI/CD Governance doesn't have to be about losing developer freedom or speed. The practices shared in this ebook highlight the CI/CD principles that help ensure control, security, stability, and quality throughout your organization and your software delivery process. Building, testing, deploying, and verifying your services on demand is easy with Harness, an all-in-one software delivery platform.

Try it for free today.

Author Appendix

Written By:

Tiffany Jachja, Technical Evangelist at Harness
Tiffany advocates for better software delivery by sharing applicable practices, stories, and content around modern technologies. Before joining Harness, Tiffany was a consultant with Red Hat's Consulting practice. There she used her experience to help customers build their software applications living in the cloud.

You can find her on Twitter: @tiffanyjachja

Reviewed By :

Steve Burton, CMO at Harness
Steve is a CD Geek at Harness. Prior to Harness, he did Geek stuff at Moogsoft, AppDynamics, and Glassdoor. He started his career as a Java developer back in 2004 at Sapient. When he's not playing around with tech, he's normally watching F1 or researching cars on the web.

You can find him on Twitter: @BurtonSays