**Ongoing Monitoring of CI/CD: Automatically Finding Threats**

In our last reading about cybersecurity incident detection methods, you explored ways to discover threats. You learned about tools like intrusion detection systems (IDS) and security information and event management (SIEM). This reading focuses on ongoing monitoring specifically for your Continuous Integration and Continuous Delivery/Deployment (CI/CD) pipelines. Monitoring your CI/CD pipeline helps protect your software supply chain.and there are special tools that can automatically find unusual activity and help you pinpoint Indicators of Compromise (IoCs).

**Automation for Finding Threats**

CI/CD pipelines help you release software faster, but they can also open up new vulnerabilities for attackers. If someone breaks into your pipeline, they could add code, steal private information, or stop your software from working. So, ongoing monitoring that automatically finds unusual pipeline activity is critical. Effective CI/CD monitoring uses automation to do more than just collect logs. It uses monitoring tools to automatically find unusual things happening in build processes, code, or deployment steps that may indicate potential security threats. When these threats are found, security teams can respond quickly and limit the damage. This automated threat detection is a main goal of strong CI/CD security.

**Common Indicators of Compromise (IoCs) in CI/CD Pipelines**

Understanding common CI/CD IoCs helps you monitor effectively and quickly find security incidents. Here are some examples:

* **Unauthorized Code Changes:**
  + Code changes from people who shouldn't be making changes.
  + Code changes made at unusual times or from unexpected locations.
  + Code changes that look suspicious, like confusing code, very large deletions without a good reason, or code that doesn't follow coding rules.
* **Suspicious Deployment Patterns:**
  + Deployments to unusual or unapproved systems (for example, production deployments started directly from developer branches).
  + Deployments happening at unexpected times or too often (deployments outside of planned release times).
  + Deployments started by unusual user accounts or automated accounts that shouldn't be releasing to production.
* **Compromised Dependencies:**
  + Finding known vulnerabilities (CVEs) in dependencies during automated checks in the CI/CD pipeline.
  + Suddenly adding new, unexpected dependencies to build settings.
  + Attempts to download dependencies from unofficial or untrusted sources.
* **Unusual Pipeline Execution:**
  + Pipeline steps that normally work fine suddenly failing.
  + Pipelines takeing much longer to run for no clear reason.
  + Changes in the order or way pipeline steps run without approved changes being made.
* **Secrets Exposure Attempts:**
  + Logs showing attempts to get to secrets from unapproved places in the pipeline.
  + Finding private secrets hardcoded in code changes (ideally prevented earlier, but monitoring can catch mistakes).

**Proactive Security Through Monitoring for IoCs**

Ongoing monitoring of CI/CD pipelines, focusing on automated anomaly detection and finding IoCs, makes your security stronger and more proactive. By using monitoring tools to continuously check pipeline activity for these indicators before serious damage occurs, you can:

* **Respond to Incidents Quickly:** Finding IoCs early helps security teams respond rapidly to potential attacks, stopping problems before attackers reach their goals.
* **Limit the Damage:** Responding quickly based on IoC detection reduces the possible impact of a security issue by limiting how long attackers are in the pipeline.
* **Improve Threat Knowledge:** Checking IoCs gives valuable information about how attackers are targeting your CI/CD, which helps improve security and threat hunting in the future.

**Using Automation to Find Anomalies and IoCs**

To monitor CI/CD pipelines and automatically find threats, you can use these methods:

**Comprehensive Logging and Auditing**

Detailed logs are the bases of monitoring. Logs provide the raw data that monitoring tools check for unusual activity and potential Indicators of Compromise (IoCs). The most common logs for finding anomalies include:

* **Pipeline Execution Logs:** To effectively leverage pipeline execution logs for security monitoring, specialized tools employ automated baselining techniques. These tools analyze logs from successful, typical CI/CD pipeline runs to establish a profile of normal operation. This baseline encompasses key performance indicators such as the standard duration of each pipeline stage and expected success and failure rates. By continuously monitoring execution logs and comparing them against this established baseline, the tools can automatically detect anomalous activities. Deviations from the norm, including pipeline steps exceeding typical execution times, unexpected error occurrences, or alterations in the usual step order, are flagged as potential Indicators of Compromise (IoCs), warranting further security scrutiny.
* **Code Commit Logs:** Keep track of code changes for each pipeline run. Unusual code changes, such as changes from people who shouldn't be making changes, changes made late at night, or changes with suspicious content (like very large deletions or confusing code), are important IoCs to monitor.
* **Access Logs:** Monitoring tools can learn who usually accesses CI/CD. Unusual logins, like logins from different countries, failed login attempts followed by a successful login, or login attempts to change important pipeline settings, are strong indicators of compromise.
* **Deployment Logs:** Tools can learn how often deployments usually happen and what those deployments look like. Unusual deployments, such as deployments at odd times or deployments to unexpected places, can be IoCs.

**Security Information and Event Management (SIEM) Integration**

Connecting your CI/CD logs to a SIEM tool can help  automatically find anomalies at a large scale. SIEM platforms are made to:

* **Automatically Find Anomalies:** SIEMs use machine learning and analytics to automatically find unusual patterns in CI/CD logs, which are  possible IoCs to investigate.
* **Use Rules to Alert for Known IoCs:** You can set up specific rules in the SIEM to find known CI/CD IoCs. For example, rules can send alerts when:
  + Detection of specific malicious file hashes (related to known CI/CD attacks) are found in build results.
  + CI/CD servers connect to known malicious command and control (C2) servers (using threat intelligence data).
  + Someone tries to download or access private secrets outside of approved pipeline steps.

**Real-time Alerting and Notifications**

Automated alerts make sure security teams are notified right away about unusual activity and possible IoCs, so they can respond quickly. Alerts should be set up for:

* **Unusual Build Failures:** Pipeline steps failing repeatedly, especially after code changes that shouldn't cause failures.
* **Suspicious Code Changes (Based on Anomalies):** Alerts sent by code analysis tools that find highly unusual code changes based on size, author, or confusing content.
* **Attempts to Expose Secrets:** Alerts sent by security tools when someone tries to access or steal secrets from unapproved parts of the pipeline.
* **Unusual Network Traffic:** Alerts for unusual network traffic from CI/CD servers, especially traffic going out to unknown or suspicious locations.

**Performance Monitoring to Find IoAs and Discover IoCs**

Performance monitoring, while mainly used to make sure things are running smoothly, can also indirectly help find IoCs. Performance issues (Indicators of Attack - IoAs) like sudden slowdowns or CI/CD servers running out of resources can lead to deeper checks that may uncover IoCs.

**Continuous Vulnerability Scanning**

Regularly checking the CI/CD infrastructure for weaknesses can proactively find vulnerable parts. This includes Common Vulnerabilities and Exposures (CVEs) in CI/CD tools, plugins, and containers. These weaknesses are potential IoCs. They highlight areas that need to be patched right away to prevent attacks and possible pipeline compromise.

**Key Takeaways**

Using automation to monitor your CI/CD pipelines, you can find unusual activity and identify important Indicators of Compromise. With this information, you can secure your software pipeline and respond quickly to threats. This enables engineers to develop, test, and deploy code with confidence and resilience against threats. By building security into your CI/CD, you empower your team to release features, improvements, and critical security updates rapidly and reliably. This ensures software is not only delivered efficiently but also with the highest level of security, proactively protecting your organization and your customers.

**Resources:**

1. Optimizing logs for a more effective CI/CD pipeline [Best Practices]. <https://coralogix.com/blog/optimizing-logs-for-a-more-effective-ci-cd-pipeline/>

  Streamline Your CI/CD: Hand-on Anomaly Detection with AI. <https://www.latesttechinsights.com/2024/04/streamline-your-cicd-hands-on-anomaly.html>

  What is CI/CD? - Continuous Integration, Delivery, and Deployment. <https://www.threatintelligence.com/blog/continuous-integration-continuous-delivery>

  CI/CD & DevOps Pipelines: An Introduction. <https://www.splunk.com/en_us/blog/learn/ci-cd-devops-pipeline.html>

* MK is a Director in the Office of the CISO at Google Cloud, focused on both protecting Google Cloud and helping customers meet their own security goals.
* He has a 32-year background in U.S. government service, including 22 years as an FBI special agent, later transitioning into cybersecurity.
* The cybersecurity industry lacks the agility that adversaries possess—attackers adapt quickly and persistently exploit what works until blocked, then pivot easily.
* The field is constantly evolving; defenders must remain flexible, think critically, and create new solutions on the fly.
* Agility and comfort with uncertainty are essential traits for cybersecurity professionals.
* Zero Trust is a growing trend and, in some regions, a requirement—it's a shift from traditional perimeter-based security.
* In simple terms, Zero Trust verifies both user identity and device security before granting access, regardless of location (e.g., a business traveler accessing data from a hotel).
* Continued investment in Zero Trust is seen as a strong foundational strategy, but the future of cybersecurity remains unpredictable.
* Ongoing learning and exposure to new areas of the field are necessary to stay prepared for emerging threats.