

Trivy ecosystem supply chain temporarily compromised

Archived: 2026-08-06 02:01:15 UTC

Summary

On March 19, 2026, a threat actor used compromised credentials to publish a malicious Trivy v0.69.4 release, force-push 76 of 77 version tags in `aquasecurity/trivy-action` to credential-stealing malware, and replace all 7 tags in `aquasecurity/setup-trivy` with malicious commits.

On March 22, 2026, a threat actor used compromised credentials to publish a malicious Trivy v0.69.5 and v0.69.6 DockerHub images.

Exposure Window

Component	Start (UTC)	End (UTC)	Duration
trivy v0.69.4	2026-03-19 18:22 1	2026-03-19 ~21:42	~3 hours
trivy-action	2026-03-19 ~17:43 2	2026-03-20 ~05:40	~12 hours
setup-trivy	2026-03-19 ~17:43 2	2026-03-19 ~21:44	~4 hours
dockerhub trivy images v0.69.5 and v0.69.6	2026-03-22 15:43	2026-03-23 ~01:40	~10 hours

Affected Components

Note that all malicious components, artifacts, commits, etc have been removed from all sources and destinations (yet they may linger in intermediary caches). Use this information to understand if you have been exposed to the malicious artifacts during the exposure window.

trivy binary and image

You are affected if you used:

1. trivy binaries version v0.69.4 (or latest during the exposure window) distributed via GitHub, Deb, RPM.
2. trivy container images v0.69.4 (or latest during the exposure window) distributed via GHCR, ECR public, Docker Hub.
3. trivy container images v0.69.5 and v0.69.6 (or latest during the exposure window) distributed via Docker Hub.

You are not affected if you used:

1. trivy (binary or image) version v0.69.3 or earlier.
 1. v0.69.3 is protected by GitHub's [immutable releases](#) feature (enabled March 3, before v0.69.3 was published).

2. v0.69.2 predates immutable releases enablement but integrity can be verified via sigstore signatures (see "How to Verify" section below).
2. trivy images referenced by digest.
3. trivy binaries built from source.
 1. The malicious code was not committed to Trivy's main branch. It was fetched and built on the ephemeral runner, and also committed to a v0.70.0 branch but no release or git tag was ever pushed.
4. homebrew from official formula (`brew install trivy`)
 1. The [official homebrew formula](#) is building trivy directly from source.
 2. There's an additional custom [trivy tap](#) which was compromised as part of the v0.69.4 release, but that tap requires special installation and is not even mentioned in the trivy documentation.

`aquasecurity/trivy-action` **GitHub Action**

You are affected if you used:

1. Any tags prior except 0.35.0 (0.0.1 – 0.34.2) to reference the action.
2. the action's `version: latest` parameter explicitly (not the default) during the trivy binary exposure window.
3. SHA pinning to a commit prior to 2025-04-09.
 1. trivy-action started pinning setup-go with pull request [trivy-action#456](#). If you pinned trivy-action to a commit prior to that PR (merged 2025-04-09), then you would get a safe trivy-action but it would get a malicious setup-trivy, if invoked during the setup-trivy exposure window.

You are not affected if you used:

1. 0.35.0 tag
 1. 0.35.0 is protected by GitHub's immutable releases feature (enabled March 4, before 0.35.0 was published) and was not affected by the tag hijacking attack.
2. SHA pinning to a safe commit after 2025-04-09.

`aquasecurity/setup-trivy` **GitHub Action**

You are affected if you used:

1. Any version without pinning.

You are not affected if you used:

1. SHA pinning to a safe commit.

Attack Details

Root Cause

This incident is a continuation of the supply chain attack that began in late February 2026. Following the initial disclosure on March 1, credential rotation was performed but was not atomic (not all credentials were revoked simultaneously). The attacker could have used a valid token to exfiltrate newly rotated secrets during the rotation

window (which lasted a few days). This could have allowed the attacker to retain access and execute the March 19 attack.

Trivy v0.69.4 binary and container images

The attacker created a malicious release by:

1. Pushing a commit (`1885610c`) that swapped the `actions/checkout` reference to an imposter commit (`70379aad`) containing a composite action that downloaded malicious Go source files from a typosquatted domain
2. Adding `--skip=validate` to goreleaser to bypass binary validation
3. Tagging this commit as `v0.69.4` , triggering the release pipeline

The compromised release was distributed across Trivy's regular distribution channels channels: GHCR, ECR Public, Docker Hub (both `0.69.4` and `latest` tags), deb/rpm packages, and `get.trivy.dev` .

The attacker attempted to release a v0.70.0 malicious release but that was stopped prematurely.

trivy-action tag hijacking

The attacker force-pushed 76 of 77 version tags to malicious commits that injected an infostealer into `entrypoint.sh` . The malicious code executes before the legitimate Trivy scan and does the following:

1. Dumps `Runner.Worker` process memory via `/proc/<pid>/mem` to extract secrets. Sweeps 50+ filesystem paths for SSH keys, AWS/GCP/Azure credentials, Kubernetes tokens, Docker configs, `.env` files, database credentials, and cryptocurrency wallets.
2. Encrypts collected data using AES-256-CBC with RSA-4096 hybrid encryption.
3. Transmits to attacker-controlled infrastructure. If exfiltration fails and `INPUT_GITHUB_PAT` is set, creates a public `tcp-docs` repository on the victim's GitHub account and uploads stolen data as a release asset.

setup-trivy release replacement

All 7 existing tags (v0.2.0 – v0.2.6) were force-pushed to malicious commits. The malicious `action.yaml` contained the same infostealer as `trivy-action`, injected as a "Setup environment" step that executes before the legitimate Trivy installation.

We have removed all malicious releases within ~4 hours and re-created v0.2.6 with safe content. Tags v0.2.0 – v0.2.5 were not restored.

Trivy v0.69.5 and v0.69.6 docker image published.

The attacker created `aquasec/trivy:0.69.5` and `aquasec/trivy:0.69.6` with the same C2 domain as the `v0.69.4` payload, and pushed them directly to Docker Hub using separately-compromised Docker Hub credentials (not via GitHub). No corresponding GitHub tags or releases existed.

We have removed all tags related to `0.69.5` and `0.69.6` and restored the latest tag to the safe `0.69.3` tag.

Recommended Actions

Update to Known-Safe Versions

Component	Safe Version
Trivy binary	v0.69.2, v0.69.3
trivy-action	v0.35.0
setup-trivy	v0.2.6

Regarding trivy-action: The original tags (`0.0.1` – `0.34.2`) were deleted during remediation. Because the attacker's force-push caused these tags to be treated as immutable releases by GitHub, they cannot be re-created with the same names. New tags have been published with a `v` prefix (`v0.0.1` – `v0.34.2`) pointing to the original legitimate commits. Three tags: `v0.0.10` , `v0.34.1` , and `v0.34.2` have not yet been restored. If you need to reference a version older than 0.35.0, use the `v` -prefixed tag (e.g., `aquasecurity/trivy-action@v0.34.0` instead of `@0.34.0`).

Rotate All Potentially Exposed Secrets

Based on information shared above, if there is any possibility that a compromised version ran in your environment, all secrets accessible to affected pipelines must be treated as exposed and rotated immediately.

Audit Trivy Versions

Check whether your organization pulled or executed Trivy v0.69.4 from any source. Remove any affected artifacts immediately.

Audit GitHub Action References

Review all workflows using `aquasecurity/trivy-action` or `aquasecurity/setup-trivy` . Check workflow run logs from March 19–20, 2026 for signs of compromise.

Search for Exfiltration Artifacts

Look for repositories named `tcp-docs` in your GitHub organization. The presence of such a repository may indicate that the fallback exfiltration mechanism was triggered and secrets were successfully stolen.

Pin GitHub Actions to Full SHA Hashes

Pin GitHub Actions to full, immutable commit SHA hashes, don't use mutable version tags. As described here: <https://docs.github.com/en/actions/reference/security/secure-use#using-third-party-actions>

How to Verify Existing Installations

Binary verification

```
# Download binary and sigstore bundle
curl -sLO "https://github.com/aquasecurity/trivy/releases/download/v0.69.2/trivy_0.69.2_Linux-64bit.tar.gz"
curl -sLO "https://github.com/aquasecurity/trivy/releases/download/v0.69.2/trivy_0.69.2_Linux-64bit.tar.gz.sigstore.json"

# Verify signature
$ cosign verify-blob \
  --certificate-identity-regexp 'https://github\.com/aquasecurity/' \
  --certificate-oidc-issuer 'https://token.actions.githubusercontent.com' \
  --bundle trivy_0.69.2_Linux-64bit.tar.gz.sigstore.json \
  trivy_0.69.2_Linux-64bit.tar.gz
Verified OK

# Check signing timestamp
$ date -u -d @$(jq -r '.verificationMaterial.tlogEntries[].integratedTime' trivy_0.69.2_Linux-64bit.tar.gz.sigstore.json)
Sat Mar 1 19:11:02 UTC 2026
#  Signed on Mar 1, before the attack on Mar 19
```

Container image verification

```
# Verify signature and get image digest
$ cosign verify \
  --certificate-identity-regexp 'https://github\.com/aquasecurity/' \
  --certificate-oidc-issuer 'https://token.actions.githubusercontent.com' \
  --new-bundle-format \
  ghcr.io/aquasecurity/trivy:0.69.2
Verification for ghcr.io/aquasecurity/trivy:0.69.2 --
The following checks were performed on each of these signatures:
- The cosign claims were validated
- Existence of the claims in the transparency log was verified offline
- The code-signing certificate was verified using trusted certificate authority certificates

# Get digest and check all signing timestamps via Rekor
$ DIGEST=$(cosign verify \
  --certificate-identity-regexp 'https://github\.com/aquasecurity/' \
  --certificate-oidc-issuer 'https://token.actions.githubusercontent.com' \
  --new-bundle-format -o json ghcr.io/aquasecurity/trivy:0.69.2 2>/dev/null | \
  jq -r '.[0].critical.image."docker-manifest-digest"')

$ rekor-cli search --sha "$DIGEST" | grep -v 'Found' | while read uuid; do
  rekor-cli get --uuid "$uuid" | grep IntegratedTime
done
IntegratedTime: 2026-03-01T19:13:52Z
IntegratedTime: 2026-03-01T19:13:47Z
IntegratedTime: 2026-03-01T19:13:57Z
IntegratedTime: 2026-03-01T19:13:54Z
IntegratedTime: 2026-03-01T19:13:46Z
```

IntegratedTime: 2026-03-01T19:13:37Z

 All signed on Mar 1, before the attack on Mar 19

Indicators of Compromise

Executable binaries

SHA256	Filename
c5b16c42dbd2a1494141cd651a406ec9094d5031a421c0aa624c4d139ae81239	trivy_0.69.4_FreeBSD_64bit.tar.gz
cff74e3e9ac0cda2078d31800d8fcad832d7b52c9920b085054d1e96dacff8a3	trivy_0.69.4_Linux-32bit.deb
55047c55a5ceab6d80b13884b4a4e8cd27a0bab7a218a952a00aae9e05f16f80	trivy_0.69.4_Linux-32bit.rpm
ba04ba6a0c028cde17599c8ddaeafb854055c5a23c595e06630732002ea59a76	trivy_0.69.4_Linux-32bit.tar.gz
0ca60dd18178d1c79d59cc06be12c540c121a4ea467484244667131aa13c311	trivy_0.69.4_Linux-64bit.deb
a5696321a6c93071f46c8bb8cbd0a8d2bce6d1860cc3c109247a4e8b64ebd317	trivy_0.69.4_Linux-64bit.rpm
385d498d18a3a7c67878ca7322716f9da25683eb1a4bf9e9592da0d5f2ab09f6	trivy_0.69.4_Linux-64bit.tar.gz
8f0c7b92b251c61cbca2add06c676dd21fde8fbb2d0cd6616383fae29b21756a	trivy_0.69.4_Linux-ARM.deb
c5df9d1bc6275711b2884a9ed4aacfe4e10dbe3c8f6c79df59126fd0e6dcd83f	trivy_0.69.4_Linux-ARM.rpm
f7a9bbfec8add36c548add4d875848b8b57c21fabe236d115f1c49113d12b332	trivy_0.69.4_Linux-ARM.tar.gz
9a833d68a49ec6d44bc50fb9ff3b184bafb0edc913e1293daebe51d334676a70	trivy_0.69.4_Linux-ARM64.deb
451ce0c4deb620894d07a2f4a37c8ea3b7a4f9b6d111651b4ac3bcc737b0fac0	trivy_0.69.4_Linux-ARM64.rpm
e401ae1e6d2442fa9a0c79dc0f3b0457ecfebf74a9c0a920159c49437f663aef	trivy_0.69.4_Linux-ARM64.tar.gz
284622577cf6a7c58704de60194205f765fcef432934c200b462ef0290aa5f57	trivy_0.69.4_Linux-PPC64LE.deb
5fac89e66d70cadec5c0e30c0b0cf8bf38c145cbf06422d40d076985195e1dd6	trivy_0.69.4_Linux-PPC64LE.rpm
52518d441fd6dd25fa5126683a330592d3be80d5ce3fb9e0b1becb806ff4f857	trivy_0.69.4_Linux-PPC64LE.tar.gz
62585efcdc7767f3fe0b9ae2897fe03bf331934492fd7a5da46f14fd7bf705c8	trivy_0.69.4_Linux-s390x.deb
107be2081bdc3ddad2889ae037ab2ad6bbd214fb9a43eaa25390d00411d1c7dd	trivy_0.69.4_Linux-s390x.rpm
16c855c398a8b185a907790054b70164358844a893bf9965651b88d6967c7c0a	trivy_0.69.4_Linux-s390x.tar.gz
90d61cf37355b89fae9ff84867100e1721c1876007ef1771e465ce5a721141ad	trivy_0.69.4_macOS-64bit.tar.gz
1dc871b02cd7a1fd80babb1b8762a2fd9cc2b735d4d3759d012626de3ccc7a5b	trivy_0.69.4_macOS-ARM64.tar.gz
0376b98064636c30f5fbe60fb3b1225516e23e88dd7e909937f81d9265292e7d	trivy_0.69.4_windows_64bit.zip

SHA256	Filename
822dd269ec10459572dfaafe163dae693c344249a0161953f0d5cdd110bd2a0	trivy_0.69.4_linux_amd64
e64e152afe2c722d750f10259626f357cdea40420c5eedae37969fbf13abbecf	trivy_0.69.4_linux_arm64
d5edd791021b966fb6af0ace09319ace7b97d6642363ef27b3d5056ca654a94c	trivy_0.69.4_s390x
ecce7ae5ffc9f57bb70efd3ea136a2923f701334a8cd47d4fbf01a97fd22859c	trivy_0.69.4_ppc64le

Container images (v0.69.4)

Digest	Tag
sha256:27f446230c60bbf0b70e008db798bd4f33b7826f9f76f756606f5417100beef3	0.69.4
sha256:12c702212dee1cbec9471e9261501a3335963321fe76e60e5a715b5acd3c40a2	0.69.4-linux/amd64
sha256:2d7cee41048988eec27615412e7c6e2e21046f2b5faa888c24e11ca6764058ed	0.69.4-linux/arm64
sha256:ae3494bd6ae860d7727116681bd09fc7b20dc994ec7a8105738f0a623ea93427	0.69.4-linux/ppc64le
sha256:43f46547efd488e56dcf862ed4d7cc342730a803f8d5bec5cac443028fefabef	0.69.4-linux/s390x
sha256:cc464a3961e1dbe145c75343b55c2f446e08b821782ec993728c4222b0d85589	0.69.4-signature
sha256:5aaa1d7cfa9ca4649d6ffad165435c519dc836fa6e21b729a2174ad10b057d2b	0.69.5
sha256:95ff680103570179feb0c6667a9b9b2d98c53fa5a9a451265036810390bbe70a	0.69.5-linux/arm64
sha256:4f7a06bb51714713ab308d2f8125f3b09ee1c3ffbbba1a5ffd0cc80da95fbb6cc	0.69.5-linux/ppc64le
sha256:edef8e5816eced552a909b878ff262c0c47776d3297bcc23796ad4cce1e85414	0.69.5-linux/s390x
sha256:425cd3e1a2846ac73944e891250377d2b03653e6f028833e30fc00c1abbc6d33	0.69.6
sha256:dd8beb3b40df080b3fd7f9a0f5a1b02f3692f65c68980f46da8328ce8bb788ef	0.69.6-linux/amd64
sha256:4b22cedea58780ff76735c3e08b9ee8cb5d06c908ffa868152f11d45349eb696	0.69.6-linux/arm64
sha256:9efd59534d2b6b81b8b7a0eeb3ad0e74015f358650e24b9dab00c900d3118593	0.69.6-linux/ppc64le
sha256:5e5fb53cf4ce5555171ff5206302ba2f4f66f5381bbf673c354c87a925473f07	0.69.6-linux/s390x

Network

C2/sinks:

- scan.aquasecurity.org
- 45.148.10.212

GitHub Repositories

Public repo on victim's GitHub account with `tpcp-docs-` prefix.

Stolen data uploaded as a release asset with tag `data-<timestamp>` .

References

- <https://www.aquasec.com/blog/trivy-supply-chain-attack-what-you-need-to-know/>
- [#10425](#)

Footnotes

1. Time when v0.69.4 release artifacts became publicly available. The malicious tag was pushed at ~17:43 UTC, triggering the release pipeline. [↩](#)
2. Earliest suspicious activity observed in our audit log. [↩](#) [↩²](#)

Source: <https://github.com/aquasecurity/trivy/security/advisories/GHSA-69fq-xp46-6x23>