

Trivy Compromised: Everything You Need to Know about the Latest Supply Chain Attack

By Rami McCarthy

Published: 2026-03-20 · Archived: 2026-08-06 02:01:06 UTC

Update March 23, 17:40 UTC: Wiz Research has [identified a parallel compromise](#) of `kics-github-action`

On March 19, 2026, threat actors compromised Aqua Security's [Trivy](#) vulnerability scanner, injecting credential-stealing malware into official releases and GitHub Actions. While [Aqua reports](#) they have since removed the malicious releases, organizations using Trivy should audit their environments immediately.

Update March 22, 13:15 UTC: Wiz Research continues to track TeamPCP activity following the initial Trivy compromise. The threat actor has expanded operations to the npm ecosystem via a worm ("[CanisterWorm](#)") leveraging stolen publish tokens. Additionally, the ICP-hosted fallback C2 (`tdtqy-oyaaa-aaaae-af2dq-cai`) is now actively serving an iteratively developed payload (`kamikaze.sh`). Aqua has published a [blog post](#) and a [GitHub Security Advisory](#).

Update March 22, 21:40 UTC: ~16:00 UTC, attackers were able to publish malicious images of Trivy (`0.69.5` , `0.69.6`) to Docker Hub. The attacker has also demonstrated continued access to Aqua by publishing internal Aqua repositories publicly on GitHub. As of 21:31 UTC, the IPC Canister has been made "Unavailable Due to Policy Violation." We continue to monitor the situation.

Update March 23, 19:26 UTC: Aqua's blog post has been [updated with additional details](#).

Note: this incident is distinct from the previous instance earlier this month, where `MegaGame10418` exploited a PWN request, that was also later flagged by `hackerbot-claw` . Customers can refer to [the Threat Center Advisory on the previous incident](#).

What happened?

Wiz Research, in concert with other industry parties, identified a multi-faceted supply chain attack targeting Aqua Security's Trivy. The attack **compromised multiple components of the Trivy project: the core scanner, the `trivy-action` GitHub Action, and the `setup-trivy` GitHub Action**.

The attack was conducted with [access retained following incomplete containment of the earlier incident](#).

The threat actor, [self-identifying as TeamPCP](#), made imposter commits that were pushed to `actions/checkout` (while spoofing user `rauchg`) and to `aquasecurity/trivy` (while spoofing user `DmitriyLewen`). At 17:43:37 UTC, the Trivy repository's `v0.69.4` tag was pushed, triggering a release. This resulted in a malicious checkout that fetched **credential stealer code** from a typosquatted domain (`scan.aquasecurtiy[.]org` , resolving to

45.148.10.212), and backdoored binaries being published to GitHub Releases, Docker Hub, GHCR, and ECR. The maintainers have since removed these malicious artifacts.

The attacker also compromised the `aqua-bot` service account and then abused their access push malicious workflows to `tfsec` , `traceeshark` , and `trivy-action` and steal additional credentials from Aqua (including GPG keys and credentials for Docker Hub, Twitter, and Slack). These secrets were exfiltrated to a Cloudflare Tunnel C2 (`plug-tab-protective-relay.trycloudflare.com`). Furthermore, **75 out of 76 `trivy-action` tags were force-pushed to malicious versions, and 7 `setup-trivy` tags were force-pushed as well.**

The malicious versions of these Actions run a tool self-described as "TeamPCP Cloud stealer", which dumps `Runner.Worker` process memory, harvests SSH, cloud, and K8s secrets, encrypts the data (using AES-256+RSA-4096), and exfiltrates it to a remote server. The malware is configured to create a repository named `tpcp-docs` in the victim's GitHub account as a fallback method.

What did the malicious GitHub workflows do?

The attacker injected the same three-stage payload into both `trivy-action` (in `entrypoint.sh`) and `setup-trivy` (in `action.yaml`). When a downstream workflow executed either action, the payload ran on the victim's GitHub Actions runner:

1. **Collection:** The malware scraped memory from the `Runner.Worker` process by reading `/proc/<pid>/mem` and searching for the pattern `{"value":"<secret>","isSecret":true}` . It also swept the filesystem for SSH keys, cloud credentials (AWS, GCP, Azure), Kubernetes tokens, and cryptocurrency wallets across 50+ sensitive file paths.
2. **Encryption:** Harvested credentials were encrypted using AES-256-CBC with RSA-4096 hybrid encryption and bundled into `tpcp.tar.gz`
3. **Exfiltration:** The malware transmitted the encrypted bundle to a typosquatted domain (`scan.aquasecurity[.]org`). As a fallback, it could create a `tpcp-docs` repository in the victim's GitHub account and upload the stolen credentials as a release asset.

What did the malicious Trivy binary (v0.69.4) do?

When the malicious binary is executed it starts both the legitimate trivy service and the malicious code in parallel. In our initial analysis the malicious code exfiltrates secrets with a primary and backup mechanism. If it detects it is on a developer machine it additionally writes a base64 encoded python dropper for persistence.

Data Theft

The malicious process gathers environmental variables, sweeps the system for any credentials stored in the filesystem and enumerates the network interface. It then compresses and encrypts the data and attempts to exfiltrate it via post request to `https://scan.aquasecurity[.]org` . If that fails (returns a non-2xx status code), the malware then attempts to use a stolen `GITHUB_TOKEN` to create a repo `tpcp-docs` and post the data there.

Set Persistence

The malicious code attempts to identify if it is on a developer machine by checking if the environmental variable `GITHUB_ACTIONS != "true"` . If that check passes, the malware attempts to create persistence by writing a python script as `~/.config/systemd/user/sysmon.py` and creating a systemd unit that persistently runs the python script.

`sysmon.py` initially sleeps for five minutes and then polls `https://tdtqy-oyaaa-aaaae-af2dq-cai.raw.icp0.io/` . If a non-youtube link is found there the dropper downloads to `/tmp/pglog` and executes it in a new session.

At the time of analysis (10:30 AM EST/15:30 UTC) the malicious site was set to a RickRoll, but it could be changed by the attacker at any time.

Which actions should security teams take?

1. **Audit Trivy versions:** Check whether your organization pulled or executed Trivy `v0.69.4` from any source (GitHub Releases, container registries, etc.). Remove any affected artifacts immediately.
2. **Audit GitHub Action references:** Review workflows using `aquasecurity/trivy-action` or `aquasecurity/setup-trivy` . If you referenced a version tag rather than a SHA, check workflow run logs from March 19-20 for signs of compromise. Specifically, you can look in the `Run Trivy` step of `trivy-action` and the `Setup environment` step of `setup-trivy` .
3. **Search for exfiltration artifacts:** Look for repositories named `tpcp-docs` in your GitHub organization, which may indicate successful exfiltration via the fallback mechanism. Hunt based on the IOCs provided below.

Long-term hardening: Pin GitHub Actions to full SHA hashes, not version tags. Version tags can be moved to point at malicious commits, as demonstrated in this attack.

How Wiz can help?

Wiz customers should refer to and monitor the advisory in the [Wiz Threat Center](#) for ongoing guidance, pre-built queries, and references to relevant detections they can use to assess the risk in their environment.

Worried you've been impacted? [Connect with the Wiz Incident Response team](#).

Appendix

SITF diagram

Learn more about SITF [here](#).

Indicators of compromise

Network Indicators

Indicator	Notes
scan.aquasecurity.org	Typosquatted C2
45.148.10.212	TECHOFF SRV LIMITED, Amsterdam
tdtqy-oyaaa-aaaae-af2dq-cai.raw.icp0.io	ICP-hosted fallback within malicious Trivy binary
plug-tab-protective-relay.trycloudflare.com	Used within GitHub Actions for exfiltration

Malicious Artifacts

Type	Value	Details
IOC (Hash)	887e1f5b5b50162a60bd03b66269e0ae545d0aef0583c1c5b00972152ad7e073	FreeBSD-64bit
IOC (Hash)	f7084b0229dce605ccc5506b14acd4d954a496da4b6134a294844ca8d601970d	Linux-32bit
IOC (Hash)	822dd269ec10459572dfaafe163dae693c344249a0161953f0d5cdd110bd2a0	Linux-64bit
IOC (Hash)	bef7e2c5a92c4fa4af17791efc1e46311c0f304796f1172fce192f5efc40f5d7	Linux-ARM
IOC (Hash)	e64e152afe2c722d750f10259626f357cdea40420c5eedae37969fbf13abbecf	Linux-ARM64 (unconfirmed)
IOC (Hash)	ecce7ae5ffc9f57bb70efd3ea136a2923f701334a8cd47d4fbf01a97fd22859c	Linux-PPC64LE
IOC (Hash)	d5edd791021b966fb6af0ace09319ace7b97d6642363ef27b3d5056ca654a94c	Linux-s390x
IOC (Hash)	e6310d8a003d7ac101a6b1cd39ff6c6a88ee454b767c1bdce143e04bc1113243	macOS-64bit
IOC (Hash)	6328a34b26a63423b555a61f89a6a0525a534e9c88584c815d937910f1ddd538	macOS-ARM64
IOC (Hash)	0880819ef821cff918960a39c1c1aada55a5593c61c608ea9215da858a86e349	Windows-64bit

Malicious Workflows

Credit to Socket for compiling this data and making it easily available at <https://socket.dev/supply-chain-attacks/trivy-github-actions-compromise>

Action	Hash
setup-trivy	8afa9b9f9183b4e00c46e2b82d34047e3c177bd0
setup-trivy	386c0f18ac3d7f2ed33e2d884761119f4024ff8a
setup-trivy	384add36b52014a0f99c0ab3a3d58bd47e53d00f
setup-trivy	7a4b6f31edb8db48cc22a1d41e298b38c4a6417e
setup-trivy	6d8d730153d6151e03549f276faca0275ed9c7b2
setup-trivy	99b93c070aac11b52dfc3e41a55cbb24a331ae75
setup-trivy	f4436225d8a5fd1715d3c2290d8a50643e726031
trivy-action	f4f1785be270ae13f36f6a8cfbf6faaae50e660a
trivy-action	0891663bc55073747be0eb864fbec3727840945d
trivy-action	2e7964d59cd24d1fd2aa4d6a5f93b7f09ea96947
trivy-action	ddb9da4475c1cef7d5389062bdfdfbdbc1394648
trivy-action	4209dcadeaea6a7df69262fef1beeda940881d4d
trivy-action	f5c9fd927027beaa3760d2a84daa8b00e6e5ee21
trivy-action	18f01febc4c3cd70ce6b94b70e69ab866fc033f5
trivy-action	bb75a9059c2d5803db49e6ed6c6f7e0b367f96be
trivy-action	d488f4388ff4aa268906e25c2144f1433a4edec2
trivy-action	3c615ac0f29e743eda8863377f9776619fd2db76
trivy-action	a9bc513ea7989e3234b395cafb8ed5ccc3755636
trivy-action	8519037888b189f13047371758f7aed2283c6b58
trivy-action	8cfb9c31cc944da57458555aa398bb99336d5a1f
trivy-action	9092287c0339a8102f91c5a257a7e27625d9d029
trivy-action	7b955a5ece1e1b085c12dac7ac10e0eb1f5b0d4d
trivy-action	19851bef764b57ff95b35e66589f31949eeb229d
trivy-action	61fbe20b7589e6b61eedcd5fe1e958e1a95fbd13
trivy-action	fa78e67c0df002c509bcdea88677fb5e2fe6a9b1
trivy-action	b7befdc106c600585d3eec87d7e98e1c136839ae

Action	Hash
trivy-action	7f6f0ce52a59bdfc5757c3982aac2353b58f4c73
trivy-action	ddb6697447a97198bdef9bae00215059eb5e8bc2
trivy-action	3dffed04dc90cf1c548f40577d642c52241ec76c
trivy-action	ad623e14ebdfe82b9627811d57b9a39e283d6128
trivy-action	848d665ed24dc1a41f6b4b7c7ffac7693d6b37be
trivy-action	ddb94181dcbc723d96ffc07fddd14d97e4849016
trivy-action	b7252377a3d82c73d497bfafa3eabe84de1d02c4
trivy-action	fa4209b6182a4c1609ce34d40b67f5cfd7f00f53
trivy-action	2b1dac84ff12ba56158b3a97e2941a587cb20da9
trivy-action	66c90331c8b991e7895d37796ac712b5895dda3b
trivy-action	fd429cf86db999572f3d9ca7c54561fdf7d388a4
trivy-action	8ae5a08aec3013ee8f6132b2a9012b45002f8eaa
trivy-action	2a51c5c5bb1fd1f0e134c9754f1702cfa359c3dd
trivy-action	9c000ba9d482773cbbc2c3544d61b109bc9eb832
trivy-action	91e7c2c36dcad14149d8e455b960af62a2ffb275
trivy-action	4bdcc5d9ef3ddb42ccc9126e6c07faa3df2807e3
trivy-action	9e8968cb83234f0de0217aa8c934a68a317ee518
trivy-action	c5967f85626795f647d4bf6eb67227f9b79e02f5
trivy-action	b745a35bad072d93a9b83080e9920ec52c6b5a27
trivy-action	38623bf26706d51c45647909dcfb669825442804
trivy-action	555e7ad4c895c558c7214496df1cd56d1390c516
trivy-action	2297a1b967ecc05ba2285eb6af56ab4da554ecae
trivy-action	820428afeb64484d311211658383ce7f79d31a0a
trivy-action	f77738448eec70113cf711656914b61905b3bd47
trivy-action	252554b0e1130467f4301ba65c55a9c373508e35
trivy-action	22e864e71155122e2834eb0c10d0e7e0b8f65aa3

Action	Hash
trivy-action	405e91f329294fb696f55793203abf1f6aba9b40
trivy-action	506d7ff06abc509692c600b5b69b4dc6ceaa4b15
trivy-action	276ca9680f6df9016db12f7c48571e5c4639451d
trivy-action	aa3c46a9643b18125abb8aefc13219014e9c4be8
trivy-action	ea56cd31d82b853932d50f1144e95b21817e52cf
trivy-action	0d49ceb356f7d4735c63bd0d5c7e67665ec7f80c
trivy-action	7550f14b64c1c724035a075b36e71423719a1f30
trivy-action	da73ae0790e458e878b300b57ceb5f81ac573b46
trivy-action	6ec7aaf336b7d2593d980908be9bc4fed6d407c6
trivy-action	cf19d27c8a7fb7a8bbf1e1000e9318749bcd82cf
trivy-action	ef3a510e3f94df3ea9fcd01621155ca5f2c3bf5b
trivy-action	6fc874a1f9d65052d4c67a314da1dae914f1daff
trivy-action	b9faa60f85f6f780a34b8d0faaf45b3e3966fdda
trivy-action	ab6606b76e5a054be08cab3d07da323e90e751e8
trivy-action	a5b4818debf2adbaba872aaffd6a0f64a26449fa
trivy-action	e53b0483d08da44da9dfe8a84bf2837e5163699b
trivy-action	8aa8af3ea1de8e968a3e49a40afb063692ab8eae
trivy-action	91d5e0a13afab54533a95f8019dd7530bd38a071
trivy-action	794b6d99daefd5e27ecb33e12691c4026739bf98
trivy-action	9ba3c3cd3b23d033cd91253a9e61a4bf59c8a670
trivy-action	e0198fd2b6e1679e36d32933941182d9afa82f6f
trivy-action	9738180dd24427b8824445dbbc23c30ffc1cb0d8
trivy-action	3201ddddd69a1419c6f1511a14c5945ba3217126
trivy-action	985447b035c447c1ed45f38fad7ca7a4254cb668
trivy-action	3d1b5be1589a83fc98b82781c263708b2eb3b47b
trivy-action	fd090040b5f584f4fcbe466878cb204d0735dcf4

Action	Hash
trivy-action	85cb72f1e8ee5e6e44488cd6cbdbca94722f96ed
trivy-action	cf1692a1fc7a47120e6508309765db7e33477946
trivy-action	1d74e4cf63b7cf083cf92bf5923cf037f7011c6b
trivy-action	c19401b2f58dc6d2632cb473d44be98dd8292a93

References

- [Trivy Under Attack Again: Widespread GitHub Actions Tag Compromise Exposes CI/CD Secrets](#)
- [20 Days Later: Trivy Compromise, Act II](#)
- [Trivy Compromised a Second Time - Malicious v0.69.4 Release, aquasecurity/setup-trivy, aquasecurity/trivy-action GitHub Actions Compromised](#)
- [Trivy Security incident 2026-03-19](#)

[Watch the Webinar](#)

Source: <https://www.wiz.io/blog/trivy-compromised-teampcp-supply-chain-attack>