1 April 2026

TLP: WHITE

EMERGING THREATS

TeamPCP Supply Chain Attacks

The TeamPCP supply chain compromise has recently emerged as a highly impactful campaign targeting widely used open-source ecosystems, specifically abusing trust in package repositories such as npm and PyPI. This activity has been linked to compromises involving popular developer tooling and libraries, including Trivy, LiteLLM, and Checkmarx KICS, where malicious code was introduced into legitimate packages and distributed downstream to unsuspecting users. Researchers observed that TeamPCP leveraged these trusted packages to execute malicious payloads during installation or runtime, effectively turning legitimate software into a delivery mechanism for credential theft and environment compromise. It is worthy to note that the compromise of LiteLLM in PyPI and Trivy-related npm packages significantly increased the potential blast radius due to their widespread adoption across cloud-native, DevOps, and AI-driven environments

THREAT SUMMARY

The TeamPCP supply chain compromise has recently emerged as a highly impactful campaign targeting widely used open-source ecosystems, specifically abusing trust in package repositories such as npm and PyPI. This activity has been linked to compromises involving popular developer tooling and libraries, including Trivy, LiteLLM, and Checkmarx KICS, where malicious code was introduced into legitimate packages and distributed downstream to unsuspecting users. Researchers observed that TeamPCP leveraged these trusted packages to execute malicious payloads during installation or runtime, effectively turning legitimate software into a delivery mechanism for credential theft and environment compromise. It is worthy to note that the compromise of LiteLLM in PyPI and Trivy-related npm packages significantly increased the potential blast radius due to their widespread adoption across cloud-native, DevOps, and AI-driven environments.

Over the past several months, TeamPCP has evolved from isolated package tampering into a more coordinated and worm-like supply chain operation capable of propagating across development environments. They have been observed to have targeted developers and organizations globally, particularly those operating in technology, cloud infrastructure, and software development sectors, where automated dependency installation pipelines are common. The impact of these compromises includes unauthorized access to developer environments, exposure of API keys and credentials, and the potential compromise of downstream applications and production systems. By abusing CI/CD workflows and dependency chains, TeamPCP enables malicious actors to execute code within trusted environments, exfiltrate sensitive data, and potentially pivot into broader enterprise infrastructure. Given the scale and trust associated with affected packages, it is important to assess, understand, and prepare for this type of supply chain threat as it continues to evolve.

Verity471 References:

- SITREP 26.1: TeamPCP threat group conducts supply chain attack via Trivy vulnerability scanner
- SITREP 26.2: TeamPCP threat group conducts supply chain attack via Trivy vulnerability scanner

THREAT SYNOPSIS

Initially, the TeamPCP campaign has been observed compromising legitimate packages within both npm and PyPI ecosystems, embedding malicious logic directly into package code that is executed during installation or runtime. In the Trivy-related npm compromise, malicious scripts were injected into packages that execute automatically when developers install dependencies, allowing the attacker to run arbitrary commands within the victim environment. Similarly, in the LiteLLM PyPI compromise, attackers modified the package to include credential harvesting functionality, which extracts sensitive information such as API keys, tokens, and environment variables from the host system. These actions occur transparently within trusted workflows, making detection difficult as the execution appears to originate from legitimate development tools.

The campaign further demonstrates worm-like propagation characteristics, where compromised environments can be leveraged to introduce malicious modifications into additional repositories or packages. This is achieved by harvesting credentials such as GitHub tokens or cloud access keys, which can then be used to push malicious updates or access additional systems. Once executed, the malicious code can enumerate environment variables, access configuration files, and exfiltrate collected data to attacker-controlled infrastructure. In CI/CD environments, this allows attackers to compromise, build pipelines and inject malicious code into downstream applications, effectively extending the attack surface beyond the initial victim. By combining trusted package distribution, automated execution, and credential harvesting, TeamPCP enables attackers to gain persistent access to developer ecosystems and expand compromise across software supply chains at scale.

TeamPCP HUNT COLLECTION

https://hunter.cyborgsecurity.io/research/search?state=(compatible:!f,filters:(),library:!(cyborg_collections),page:0,size:10,sort:last_updated_desc,term:!(%275b4ca401-a8e0-4dae-9e38-6d1450cecef2%27),touched:!t)

RELATED HUNT PACKAGES

Suspicious NPM Auth Token Retrieval via Encoded Python

https://hunter.cyborgsecurity.io/research/hunt-package/bb07e860-1c4d-444d-ac94-0bb33ebd73fa

User Context systemctl Invocation by Node.js - Suspicious Service Activity

https://hunter.cyborgsecurity.io/research/hunt-package/fe009be0-7a2e-42e3-87d9-797514a03728

Usage of chmod to Enable Execution - Potential Payload Staging

https://hunter.cyborgsecurity.io/research/hunt-package/dfbdc565-a37c-472b-a4c7-6c0e5325b255

Suspicious NPM Auth Token Retrieval via npm config get

https://hunter.cyborgsecurity.io/research/hunt-package/11ee7199-f55b-429a-b5ee-52e353c14e55

Python Executing from Non-Standard Directory

https://hunter.cyborgsecurity.io/research/hunt-package/eebb7f80-07f3-4d37-aab7-51c1ad31f1b3

MITRE CONTEXT

- MITRE Technique IDs:
  - T1059.006
  - T1222.002
  - T1195.002
  - T1552
  - T1552.001
  - T1543.002
- MITRE Technique Names:
  - Credentials In Files
  - Unsecured Credentials
  - Compromise Software Supply Chain
  - Linux and Mac File and Directory Permissions Modification
  - Systemd Service
  - Python
- MITRE Tactic Names:
  - Execution
  - Credential Access
  - Privilege Escalation
  - Persistence
  - Defense Evasion
  - Initial Access

REFERENCES

1. https://www.aikido.dev/blog/teampcp-deploys-worm-npm-trivy-compromise
2. https://futuresearch.ai/blog/litellm-pypi-supply-chain-attack/\u2028
3. https://www.bleepingcomputer.com/news/security/popular-litellm-pypi-package-compromised-in-teampcp-supply-chain-attack/
4. https://www.endorlabs.com/learn/teampcp-isnt-done