

Popular telnyx package compromised on PyPI by TeamPCP

By Charlie Eriksen

Published: 2026-03-27 · Archived: 2026-08-06 02:01:55 UTC

This morning's [telnyx](#) compromise is the latest move in what is now a weeks-long TeamPCP supply chain campaign crossing multiple ecosystems. Trivy. Checkmarx. LiteLLM. And now Telnyx on PyPI, uploaded hours ago at 03:51 UTC on March 27.

The pattern is consistent: steal credentials from a trusted security tool, use those credentials to push malicious versions of whatever that tool had access to, collect whatever's running in the next environment, repeat.

Where This Fits in the Campaign

A quick recap of what TeamPCP has done over the past two weeks:

March 19: Trivy compromised. Aqua Security's open source vulnerability scanner was backdoored, resulting in CVE-2026-33634 (CVSS 9.4). Attackers exfiltrated credentials from every CI/CD pipeline running Trivy without version pinning. 44 Aqua Security GitHub repositories were renamed with the prefix `tpcp-docs-` and the description "TeamPCP Owns Aqua Security."

March 20: CanisterWorm hits npm. Using stolen tokens from Trivy users, TeamPCP published the CanisterWorm backdoor across 46+ npm packages including scopes like `@EmilGroup` and `@opengov`. The worm automated token-to-compromise: given one stolen npm token, it enumerated all publishable packages, bumped versions, and published across the entire scope in under 60 seconds.

March 22: I first observed TeamPCP using WAV steganography to deliver payloads in their Kubernetes wiper variant. [I flagged it on Twitter at the time](#): "TeamPCP is now embedding their malware in .wav files."

March 23: Checkmarx. The `kics-github-action` and `ast-github-action` GitHub Actions were [compromised](#), along with two OpenVSX extensions (`cx-dev-assist 1.7.0` and `ast-results 2.53.0`). The payload used a new C2 domain, `checkmarx[.]zone`, impersonating the Checkmarx brand. 35 tags were hijacked between 12:58 and 16:50 UTC; malicious code was removed three hours later.

March 24: LiteLLM. Versions 1.82.7 and 1.82.8 of the LiteLLM PyPI package [were published](#) using credentials stolen from LiteLLM's CI/CD pipeline, which ran unpinned Trivy. LiteLLM serves roughly 95 million downloads per month and is increasingly deployed as a centralized LLM gateway with access to credentials for OpenAI, Anthropic, AWS Bedrock, GCP VertexAI, and more. PyPI quarantined the packages after about three hours. The C2 was `models[.]littellm[.]cloud`.

March 27 (today): [Telnyx](#). Two malicious versions of the official Telnyx Python SDK hit PyPI this morning. Telnyx has been downloaded 742k times over the last month.

The Telnyx Payload

The injection is in `telnyx/_client.py`, which runs at import time. No install hook to disable, no postinstall to block. Just `import telnyx` and the malware runs.

Two paths depending on OS:

Windows: Downloads `hangup.wav` from `83[.]142[.]209[.]203:8080`, decodes an XOR-obfuscated executable from the audio frames, drops it as `msbuild.exe` in the Windows Startup folder. Runs silently on every login with a 12-hour re-drop cooldown enforced by a hidden `.lock` file.

Linux/Mac: A complete second-stage Python script is hardcoded as a base64 blob in `_client.py` at line 459. It fetches `ringtone.wav` from the same C2, decodes a third-stage collector script from the WAV frames using the same XOR technique, runs it via `sys.executable -` piped to stdin, then encrypts the output with AES-256-CBC and exfiltrates it. The session key is wrapped with an attacker RSA-4096 public key (OAEP), so only the attacker can decrypt what was stolen.

The exfil bundle is named `tpcp.tar.gz`, sent with header `X-Filename: tpcp.tar.gz`.

The WAV Trick

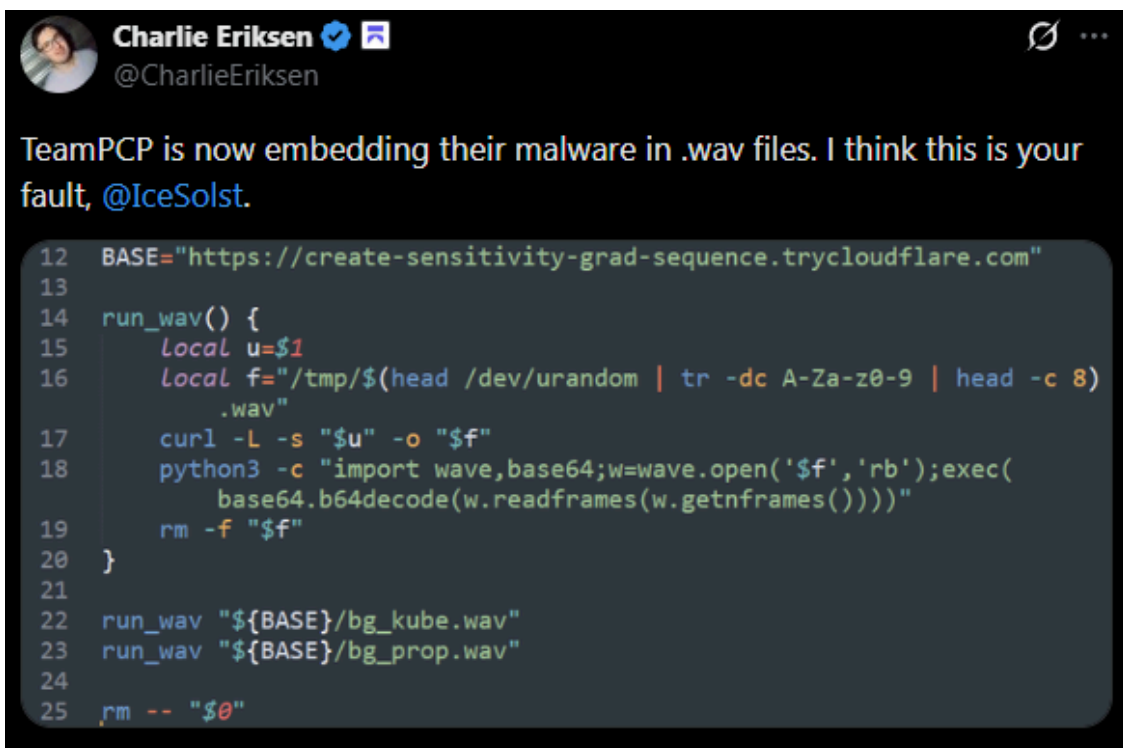
This is the part worth dwelling on. The payload isn't delivered as a raw binary or a Python file. It's disguised as a `.wav` audio file. Here's what happens when the malware fetches `hangup.wav` (Windows) or `ringtone.wav` (Linux) from the C2:

```
with wave.open(wf, 'rb') as w:
    b = base64.b64decode(w.readframes(w.getnframes()))
    s, m = b[:8], b[8:]
    payload = bytes([m[i] ^ s[i % len(s)] for i in range(len(m))])
```

The WAV file is a valid audio file. It passes MIME-type checks. But the audio frame data contains a base64-encoded payload. Decode the frames, take the first 8 bytes as the XOR key, XOR the rest, and you have your executable or Python script.

Content-based filtering won't catch it. A URL allowing `.wav` fetches won't block it. The file looks like audio because it is structured as audio. The malicious content is just hiding in the frame data.

We first saw this technique in TeamPCP's version 3.3 Kubernetes payload on March 22. Five days later it's in the telnyx PyPI package, carrying both the Windows dropper and the Linux infostealer. They liked it enough to keep it.



What to Do

Remove `telnyx>=4.87.1` immediately and pin to `telnyx==4.87.0`.

If you installed either malicious version, treat the environment as compromised: rotate API keys, database credentials, SSH keys, and any secrets accessible from that machine. On Windows, check for `msbuild.exe` in `%APPDATA%\Microsoft\Windows\Start Menu\Programs\Startup\` and delete it. Monitor for outbound HTTP to `83[.]142[.]209[.]203:8080`.

IOCs

Malicious telnyx versions:

- `telnyx==4.87.1` (SHA256: `7321caa303fe96ded0492c747d2f353c4f7d17185656fe292ab0a59e2bd0b8d9`)
- `telnyx==4.87.2` (SHA256: `cd08115806662469bbedec4b03f8427b97c8a4b3bc1442dc18b72b4e19395fe3`)

Network:

- `83[.]142[.]209[.]203:8080` (C2)
- `hxxp://83[.]142[.]209[.]203:8080/hangup.wav` (Windows payload)
- `hxxp://83[.]142[.]209[.]203:8080/ringtone.wav` (Linux payload)
- `hxxp://83[.]142[.]209[.]203:8080/` (exfil POST)
- Exfil header: `X-Filename: tpcp.tar.gz`

Windows persistence:

- `%APPDATA%\Microsoft\Windows\Start Menu\Programs\Startup\msbuild.exe`

- `%APPDATA%\Microsoft\Windows\Start Menu\Programs\Startup\msbuild.exe.lock`

Developing story... Stay tuned for updates.

Source: <https://www.aikido.dev/blog/telnix-pypi-compromised-teampcp-canisterworm>