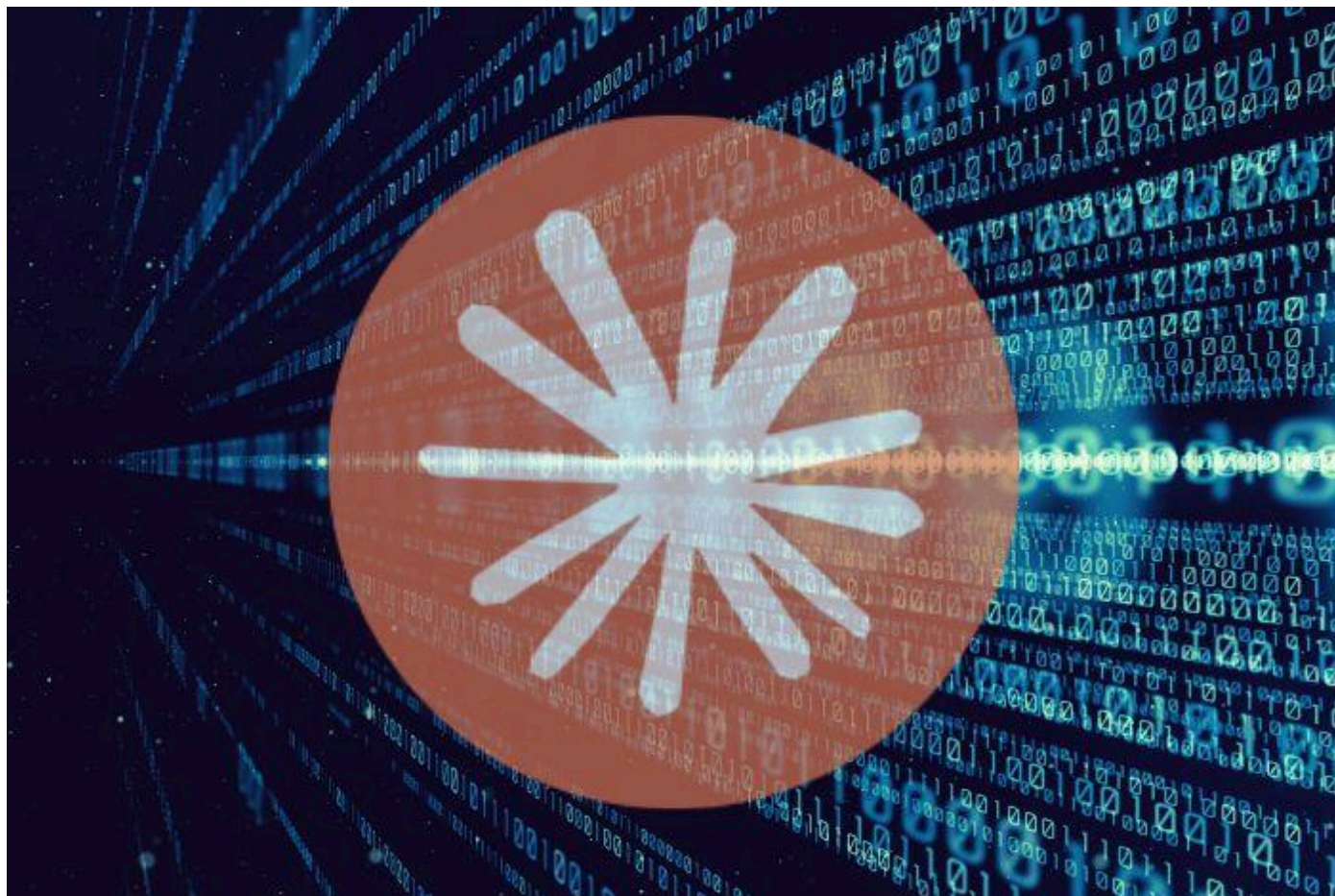


Anthropic Claude Code Leak

zscaler.com/blogs/security-research/anthropic-claude-code-leak

Manisha Ramcharan Prajapati, Jithin Prajeev Nair, Avinash Kumar, Mallikarjun Piddannavar

April 1, 2026



[Security Research](#)

Introduction

On March 31, 2026, Anthropic accidentally exposed the full source code of Claude Code (its flagship terminal-based AI coding agent) through a 59.8 MB JavaScript source map (.map) file bundled in the public npm package `@anthropic-ai/claude-code` version 2.1.88. A security researcher, [Chaofan Shou \(@Fried_rice\)](#), publicly disclosed Anthropic's leak on X which triggered an immediate viral response.

The leaked file contained approximately 513,000 lines of unobfuscated TypeScript across 1,906 files, revealing the complete client-side agent harness, according to online [publications](#). Within hours, the codebase was downloaded from Anthropic's own Cloudflare R2 bucket, mirrored to

GitHub, and forked tens of thousands of times. Thousands of developers, researchers, and threat actors are actively analyzing, forking, porting to Rust/Python and redistributing it. Some of the GitHub repositories have gained over 84,000 stars and 82,000 forks. Anthropic has issued Digital Millennium Copyright Act (DMCA) notices on some mirrors, but the code is now available across hundreds of public repositories.

In addition to discussing the Anthropic leak, this blog post also covers a “Claude Code leak” lure delivering Vidar and Ghostsocks malware that was discovered and analyzed by the Zscaler ThreatLabz team.

Recommendations

- Implement Zero Trust architecture and prioritize segmenting mission critical application access.
- Do not download, fork, build, or run code from any GitHub repository claiming to be the “leaked Claude Code.” Verify every source against Anthropic’s official channels only.
- Educate developers that leaked code is not “open source”. It remains proprietary and dangerous to run unmodified.
- Avoid running AI agents with local shell/tool access on untrusted codebases.
- Monitor for anomalous telemetry or outbound connections from developer workstations.
- Use official channels and signed binaries only.
- Scan local environments and Git clones for suspicious processes, modified hooks, or unexpected `npm` packages, and wait for a cool down period before using the latest `npm` packages.
- Watch for Anthropic patches addressing newly exposed paths.

Background

Claude Code is Anthropic’s official AI-powered coding CLI/agent that delegates tasks directly in the terminal, using hooks, background agents, autonomous daemons, and local execution capabilities. The leak stemmed from a packaging error where `Bun` (the runtime used) generated a full source map by default, and `*.map` was not excluded in `.npmignore` or the `files` field of `package.json`. The map file referenced a complete ZIP of the original TypeScript sources hosted on Anthropic’s infrastructure.

Components Exposed

- **Agent orchestration:** LLM API calls, streaming, tool-call loops, retry logic, thinking/review modes, multi-agent coordination.
- **Permission and execution layer:** Claude Code hooks (auto-executing shell commands/scripts), Model Context Protocol (MCP) integrations, environment variable handling, project-load flows.
- **Memory and state:** Persistent memory systems, background agents/autonomous daemons.

- **Security-related internals:** Telemetry analysis, encryption tools, inter-process communication (IPC), OAuth flows, permission logic.
- **Hidden/restricted features:** [44 feature flags](#) (20+ unshipped), internal API design, system prompts.
- **Build and dependency details:** Exact npm handling, local execution paths.

Not exposed: Model weights, safety pipelines, or user data.

Potential Misuse and Security Risks

The heavy sharing on GitHub (thousands of forks, stars, and mirrors by developers worldwide) turns this into a vector for abuse. Key risks include:

1. **Supply chain attacks via malicious forks and mirrors:** Thousands of repositories now host the leaked code or derivatives. Threat actors can (and already are) seeding trojanized versions with backdoors, data exfiltrators, or cryptominers. Unsuspecting users cloning “official-looking” forks risks immediate compromise.
2. **Amplified exploitation of known vulnerabilities and discovery of new vulnerabilities:** Pre-existing flaws (e.g., CVE-2025-59536, CVE-2026-21852, RCE and API key exfiltration via malicious repo configs, hooks, MCP servers, and env vars) are now far easier to weaponize. Threat actors with full source visibility can craft precise malicious repositories or project files that trigger arbitrary shell execution or credential theft simply by cloning/opening an untrusted repo. The exposed hook and permission logic makes silent device takeover more reliable.
3. **Local environment and developer workstation compromise:** Users building or running the leaked code locally introduce unvetted dependencies and execution paths. The leak coincided exactly with a separate malicious Axios [npm](#) supply chain attack (RATs published March 31, 00:21–03:29 UTC), creating a perfect storm for anyone updating Claude Code via npm that day.

ThreatLabz discovers “Claude Code leak” lure that distributes Vidar and GhostSocks

While monitoring GitHub for threats, ThreatLabz came across a “Claude Code leak” repository published by idbzoomh (links located in the IOC section). The repository looks like it’s trying to pass itself off as leaked TypeScript source code for Anthropic’s Claude Code CLI. The README file even claims the code was exposed through a [.map](#) file in the npm package and then rebuilt into a working fork with “unlocked” enterprise features and no message limits.

The repository link appears near the top of Google results for searches like “leaked Claude Code,” which makes it easy for curious users to encounter, as shown in the figure below.

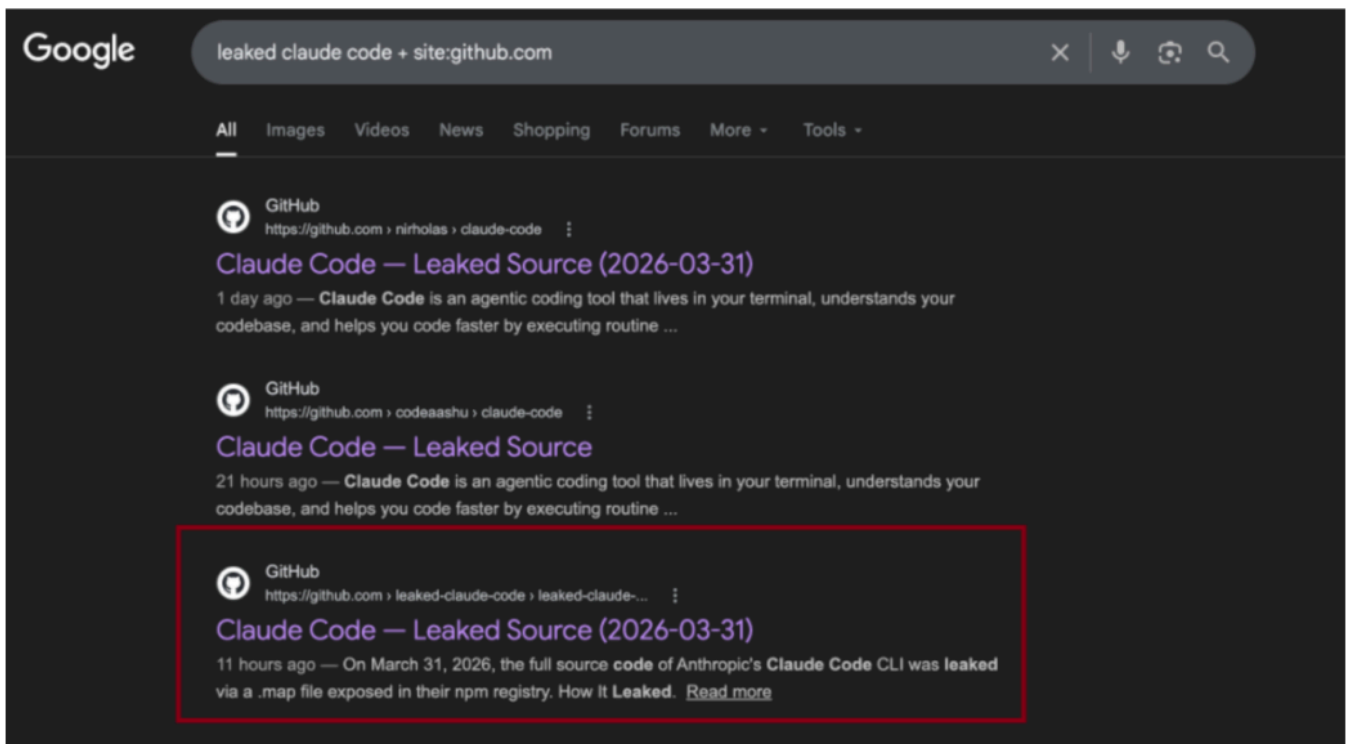


Figure 1: Google search results for leaked Claude Code on GitHub returning a malicious repository.

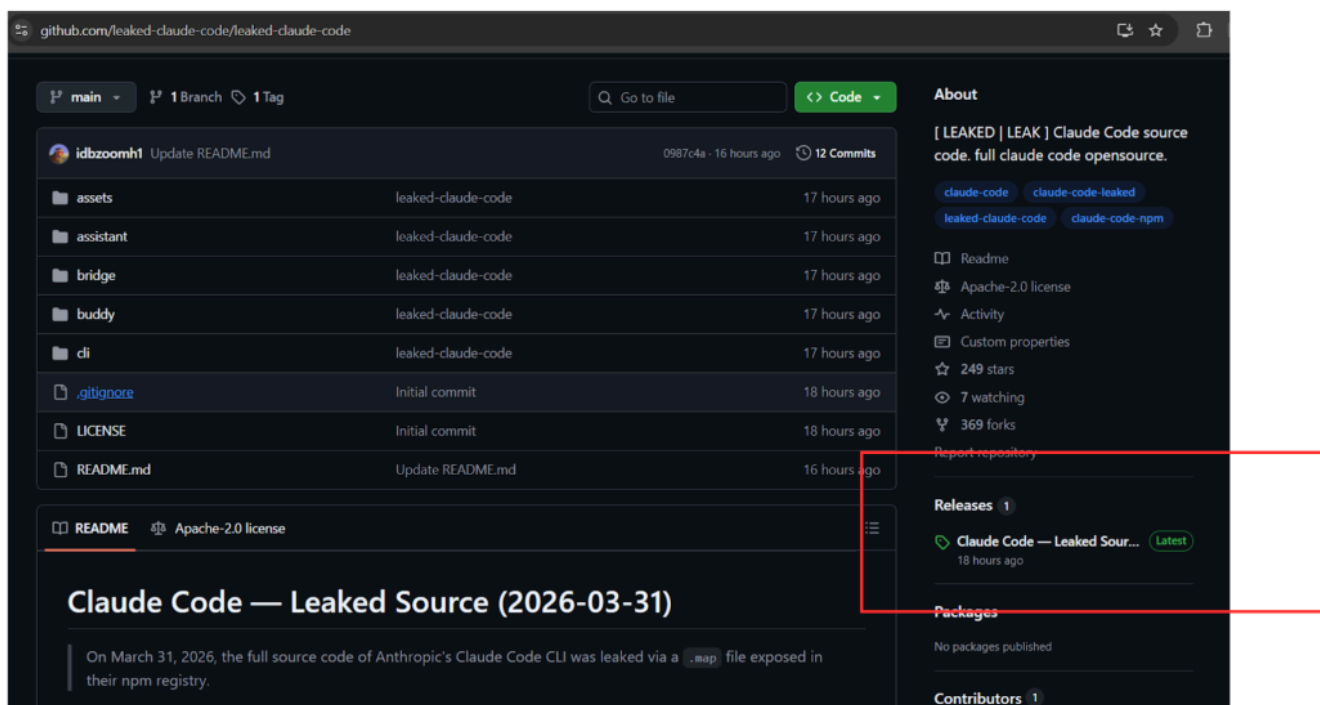


Figure 2: Malicious GitHub repository using the leaked Claude Code source as a lure.

The malicious ZIP archive in the repository's releases section is named *Claude Code - Leaked Source Code (.7z)*. The archive includes *ClaudeCode_x64.exe*, a Rust-based dropper. On execution, the *ClaudeCode_x64.exe* drops Vidar v18.7 and GhostSocks. [Vidar](#) is an information stealer and [GhostSocks](#) is used to proxy network traffic. In early March, another [security vendor](#) reported a similar campaign where GitHub was being used to deliver the same payload.

The threat actor keeps updating the malicious ZIP archive in short intervals. At the time of analysis, ThreatLabz observed that there were two ZIP archives updated in the releases section in a short timeframe. The figure below shows the first ZIP archive ThreatLabz encountered which was updated about 13 hours ago.

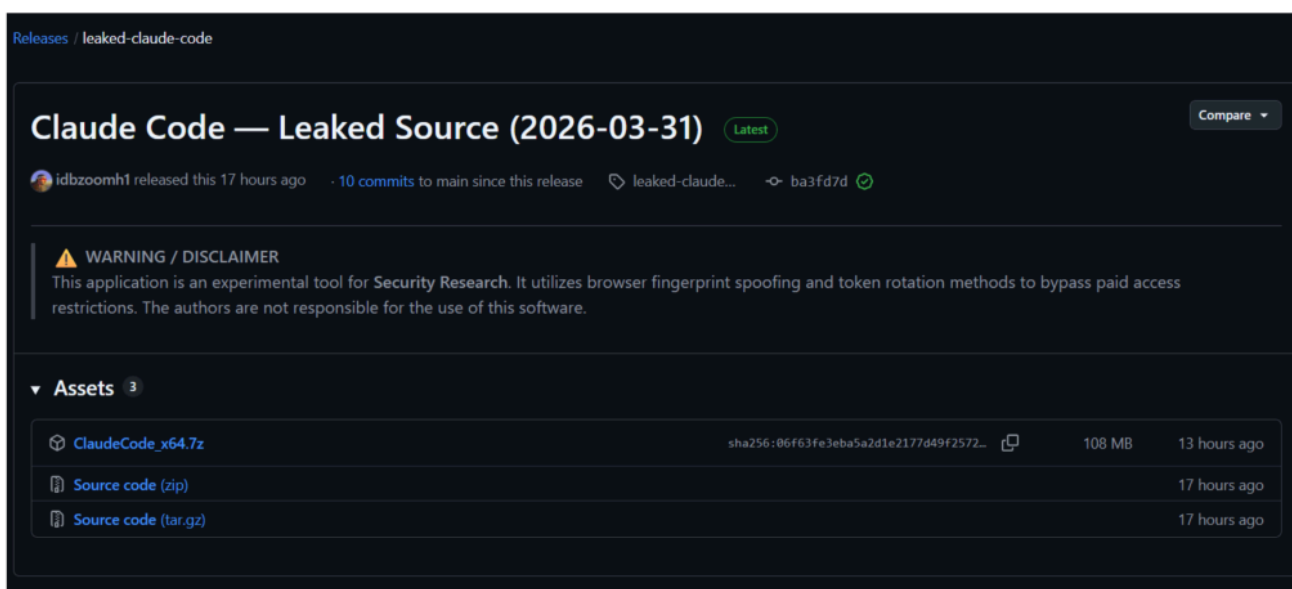


Figure 3: GitHub repository using the Claude Code leak as a lure to distribute malicious ZIP archives.

ThreatLabz also identified the same GitHub repository hosted under another account (located in the IOC section) that contains identical code and appears to be committed by the same threat actor, idbzoomh1.

Unlike the earlier repository, this one does not include a releases section. The README file displays a prominent “Download ZIP” button. However, it does not link to any compiled binary or an archive and was non-functional at the time of analysis. The figure below shows the repository and non-functional button.

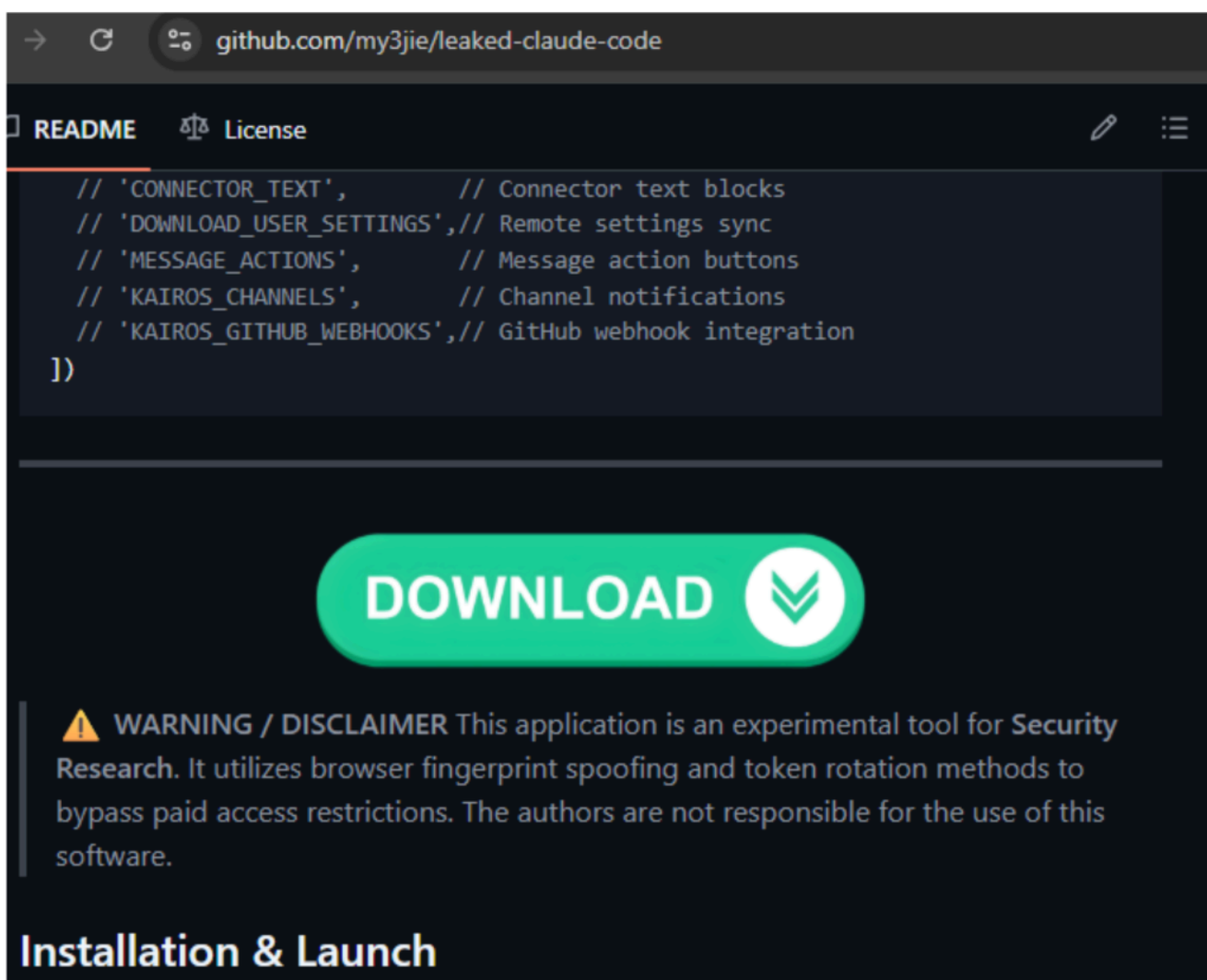


Figure 4: Additional GitHub repository hosting the same Claude Code leak lure with a “Download ZIP” button.

Conclusion

Threat actors are actively leveraging the recent Claude Code leak as a social engineering lure to distribute malicious payloads with GitHub serving as a delivery channel. Threat actors move quickly to take advantage of a publicized incident. That kind of rapid movement increases the chance of opportunistic compromise, especially through trojanized repositories.

Organizations must prioritize the implementation of Zero Trust architecture to minimize the impact from a shadow AI instance of a trojanized Claude agent, as well as potential vulnerability exploit attempts against legitimate Claude agents stemming from this code leak.

Zscaler Coverage

Zscaler has ensured coverage for the threats associated with the trojanized version of the Claude source code repository, ensuring detection with the following threat names.

Advanced Threat Protection

- [Win64.Downloader.TradeDownloader](#)
- [Win32.PWS.Vidar](#)
- [Win32.Trojan.GHOSTSOCKS](#)

Indicators Of Compromise (IOCs)

Hash	Description
d8256fbc62e85dae85eb8d4b49613774	Initial archive file
8660646bbc6bb7dc8f59a764e25fe1fd	Initial archive file
77c73bd5e7625b7f691bc00a1b561a0f	Dropper EXE file for payload
81fb210ba148fd39e999ee9cdc085dfc	Dropper EXE file for payload
9a6ea91491ccb1068b0592402029527f	Vidar v18.7
3388b415610f4ae018d124ea4dc99189	GhostSocks
https://steamcommunity[.]com/profiles/76561198721263282	Vidar DDR (Dead Drop Resolvers)
https://telegram[.]me/g1n3sss	Vidar DDR
https://rti.cargomanbd[.]com	Vidar C2
https://147.45.197[.]92:443	GhostSocks C2
https://94.228.161[.]88:443	GhostSocks C2

Hash	Description
https://github.com/leaked-claude-code/leaked-claude-code	Trojanized Claude Code source leak
https://github.com/my3jie/leaked-claude-code	Trojanized Claude Code source leak
https://github.com/idbzoomh1	Trojanized repository publisher



Thank you for reading

Was this post useful?

[Yes, very!](#)[Not really.](#)

Disclaimer: This blog post has been created by Zscaler for informational purposes only and is provided "as is" without any guarantees of accuracy, completeness or reliability. Zscaler assumes no responsibility for any errors or omissions or for any actions taken based on the information provided. Any third-party websites or resources linked in this blog post are provided for convenience only, and Zscaler is not responsible for their content or practices. All content is subject to change without notice. By accessing this blog, you agree to these terms and acknowledge your sole responsibility to verify and use the information as appropriate for your needs.

Explore more Zscaler blogs



Malicious NPM Packages Deliver NodeCordRAT

[Read post](#)



Mitigating Risks from the Shai-Hulud NPM Worm

[Read post](#)



Shai-Hulud V2 Poses Risk to NPM Supply Chain

[Read post](#)



Malicious PyPI Packages Deliver SilentSync RAT

[Read post](#)

01 / 02

Get the latest Zscaler blog updates in your inbox

*

By submitting the form, you are agreeing to our [privacy policy](#).