Quote by Devroop Dhar, Co-Founder and India CEO, Primus Partners

Published in CSO
June 18th, 2026

Attackers abuse Google Ads, GitLab, and Claude to deliver malware

Researchers tracked a seven-week campaign that leveraged trusted platforms and AI-generated trust to trick users into executing malicious commands and exposing enterprise credentials.

Authored by Nidhi Singal

Read on: Attackers abuse Google Ads, GitLab, and Claude to deliver malware | AI security, cybersecurity, AI trust, digital safety, AI risks

Article Content:

Threat actors are abusing trusted platforms, including Google Ads, GitLab pages, and Claude's shared chat feature, to trick users into executing malicious commands on their systems.

Disguised as popular AI developer tools, the threat actors used ClickFix social engineering attacks, where victims were tricked into manually executing malicious commands. Typically, this involved copying and pasting PowerShell or terminal commands, noted researchers at TrendAI.

The campaign funnelled more than 2,000 victims from sponsored Google search results for popular AI developer tools to malicious download pages before leveraging the claude.ai shared-chat feature as another stage in the attack chain.

The campaign demonstrates how threat actors are exploiting trust in widely used AI platforms to make social-engineering attacks more convincing and harder to detect.

Inside the six-wave campaign:

Unlike traditional malware campaigns that rely on suspicious domains or fake download websites, this attack chain was built almost entirely on legitimate services. The threat actors used 92 unique malicious hostnames across GitLab pages, impersonated legitimate brand names including ChatGPT Codex, Perplexity, Cursor IDE, JetBrains, Claude AI, and claude.ai, and simultaneously ran Mac utility scam lures.

The campaign was spread across seven weeks, where weekly campaigns introduced new pages and keywords.

The first wave of the campaign launched between April 8-13, with claude-code-app.gitlab[.]io as the primary lure, supported by claudeapp.gitlab[.]io. Simultaneously, Mac utility-themed lures (mac-clean-storage.gitlab[.]io, mac-guide-tool.gitlab[.]io) were also found to have been deployed. During this wave, a single Google Ads campaign ID (23736589328) resulted in driving the majority of the traffic.

During the next wave spanning April 14-21, the campaign was diversified with the new Claude-themed variants, including gitlab.io domain (claude-tool-app, claud-desktop-app, claudesktop, claude-desktop-apps) alongside expanded Mac utility lures (macsupp-group, macsupp-usb, jetbrains-apps-group).

The brand impersonation was expanded during the third wave with the introduction of perplexity-platform.gitlab.io and chatgpt-codex.gitlab[.]io, while also creating claude-desktop-lm.gitlab[.]io and cladesktop.gitlab[.]io.

During the fourth wave between April 29 and May 5, the operators pivoted significantly toward ChatGPT and Codex branding with codexgpt.gitlab[.]io , chatgpt-codex-app.gitlab[.]io, and chatgpt-codex-lm.gitlab[.]io, while the Claude-themed attacks continued.

In the fifth wave, spanning May 6-14, the threat actors moved their campaign from self-hosted GitLab Pages to abusing claude.ai's legitimate shared chat feature. For this, claude.ai's "share" feature was leveraged to create persistent, publicly accessible URLs on a fully trusted domain and then used Google Ads to direct victims to these weaponized pages, claimed the research.

During the sixth wave, between May 21 and June 14, the threat actors had completely shifted to claude.ai's shared chat feature.

The campaign appears to have been designed primarily to target developers and technical users, say experts.

"An interaction with a Claude can be perceived as reliable because the users have become accustomed to considering AI tools as sources of productivity tips and technical advice. In this scenario, when users are provided with harmful instructions via an AI platform, there is a good chance that they will comply with them automatically," explained Devroop Dhar, co-founder and India CEO at Primus Partners.

Reputation-based defenses fell short:

Security experts say the campaign's success stemmed from its ability to leverage trusted platforms at every stage of the attack chain, making malicious activity appear like normal user behavior.

"What makes this attack chain particularly effective is that it does not ask the victim to trust something obviously suspicious. Instead, it borrows trust from familiar brands, legitimate ad infrastructure, reputable hosting, and an AI platform that many developers already use in their daily workflow. This reduced the psychological friction that normally makes users pause before clicking or executing something," said Amit Jaju, senior managing director at Ankura Consulting.

This is also a strong example of trust stacking. Each layer looks individually legitimate, so the full chain appears safer than it really is, added Jaju.

By leveraging platforms that organizations routinely allow and trust, the attackers were able to blend malicious activity into normal user workflows, making detection significantly more difficult.

Dhar added that in most cases, access to applications such as Google, GitLab, and AI applications is not blocked, as this could hinder operations within an organization. The reputation-based security systems cannot work efficiently here since the domain in question is seen as a reputable one, meaning security personnel will have to dig deeper into behaviour and user actions.

Breaking the attack chain:

If a developer falls victim, the blast radius can be much larger than a normal user compromise. Jaju warned that a developer machine often contains browser session cookies, SSO tokens, SSH keys, Git credentials, source code, cloud CLI tokens, package

manager credentials, secrets stored in local files, and access to internal documentation or collaboration platforms.

From there, attackers can move into code repositories, CI/CD pipelines, cloud environments, container registries, ticketing systems, and enterprise messaging platforms. In some cases, they may not need to steal passwords at all because session tokens or authenticated browser sessions are enough to bypass part of the security stack.

While the campaign relied heavily on trusted platforms, organizations can still disrupt attacks at multiple points.

Dhar noted the first thing to understand here is that not all cyberattacks necessarily require malicious software. Nowadays, more and more attackers try to persuade victims into performing actions by themselves. Hence, one solution might be limiting unnecessary administrative privileges, monitoring shell and PowerShell executions, and detecting any suspicious behaviour. Additionally, developer PCs might need to be monitored because of the high level of access they have.

From a control standpoint, enterprises should restrict local admin rights and enforce least privilege on developer endpoints. They should also segment developer environments and separate high-risk browsing from privileged engineering workflows, where feasible, added Jaju.