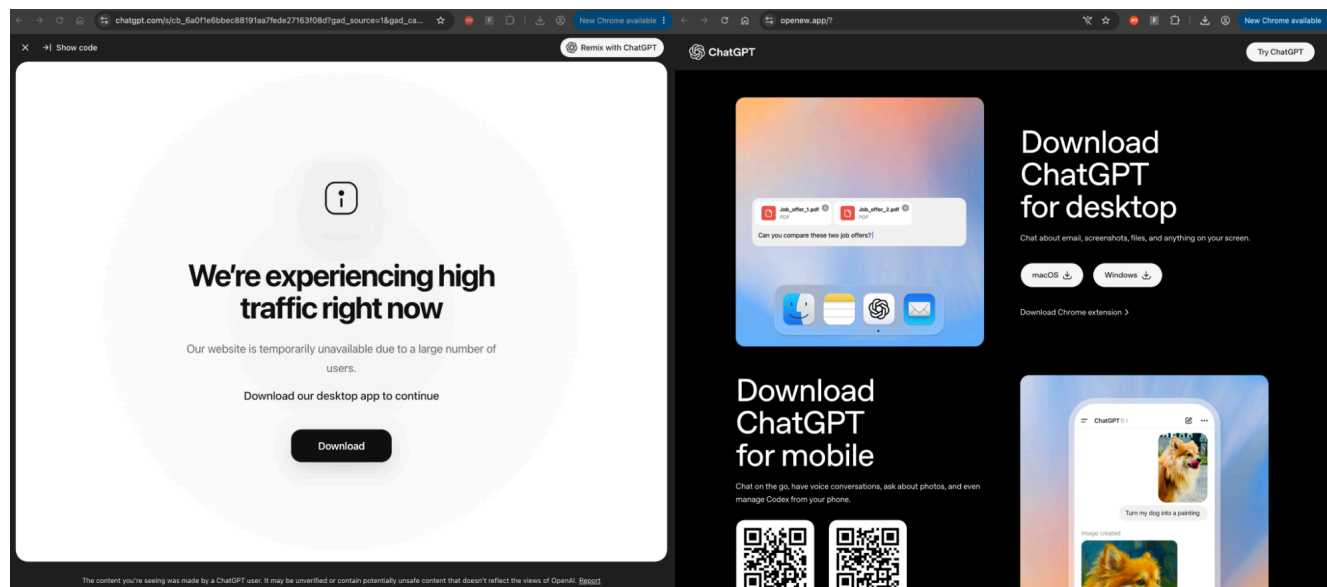


using shared chatbot pages to distribute malware

 pushsecurity.com/blog/llmshare-malvertising-campaign

Keanu Maharaj

May 29, 2026



[Table of contents](#)

[Subscribe to get updates from Push](#)

Enjoying this blog post? Sign up to get notified when we publish new content on topics you're interested in.

[Share this post](#)

Attackers are abusing the shared content features of AI chatbot platforms — ChatGPT and Claude — to deliver malware through pages hosted on legitimate, trusted domains, distributing the malicious links via sponsored malvertising ads on search engines.

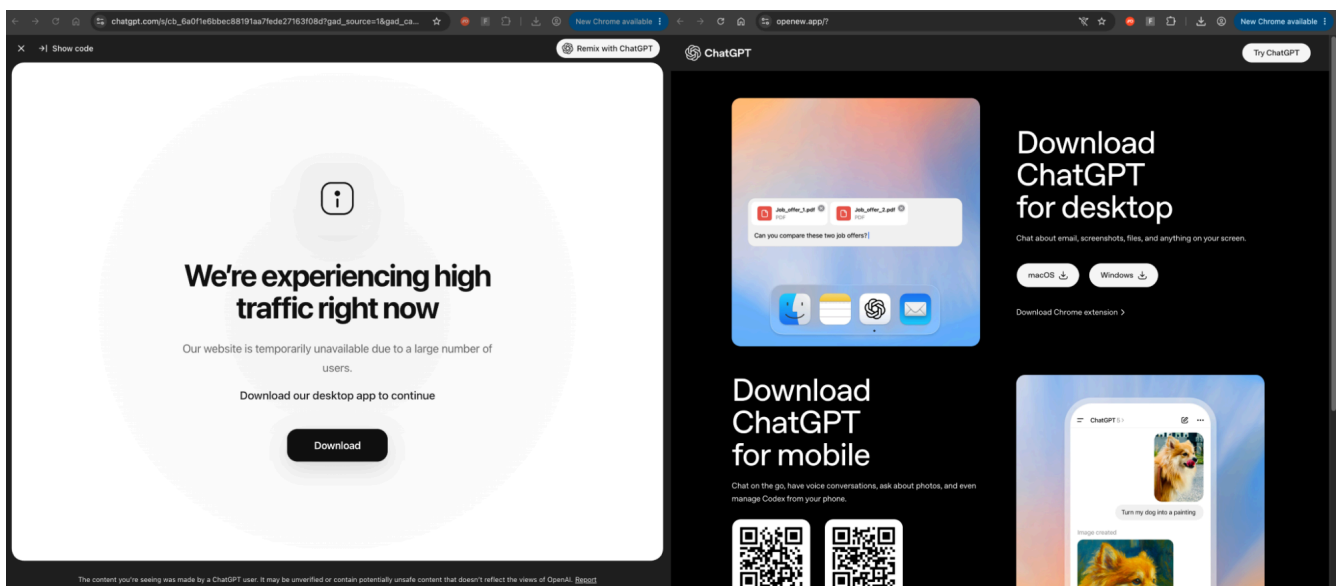
Shared conversations on AI chatbot platforms have become the latest delivery mechanism for malware campaigns targeting macOS and Windows users. Attackers create content on platforms like ChatGPT and Claude that appears to offer installation guidance or service updates, then drive traffic to it via search engine results in the form of malvertising and SEO poisoning.

The content lives on chatgpt.com or claude.ai — domains that users and security tools trust implicitly — so the attack bypasses URL reputation checks before the victim even reaches the

malicious payload.

Several variants of this technique have been [reported over the past few months](#). The earliest examples used shared Claude.ai conversations disguised as installation guides — complete with fake "Apple Support" attribution — that walked users through opening a terminal and pasting a curl command that downloaded and executed an infostealer. [Kaspersky documented a parallel campaign](#) using shared ChatGPT conversations to deliver the AMOS (Atomic macOS Stealer) via the same paste-this-command social engineering pattern.

Push has detected a new variant that goes beyond the previously reported technique of embedding terminal commands in shared conversations: the attacker has used ChatGPT's code rendering feature to build a fully designed fake page that mimics a ChatGPT service disruption, redirecting victims to a convincing clone of ChatGPT's download page that delivers a malicious executable.

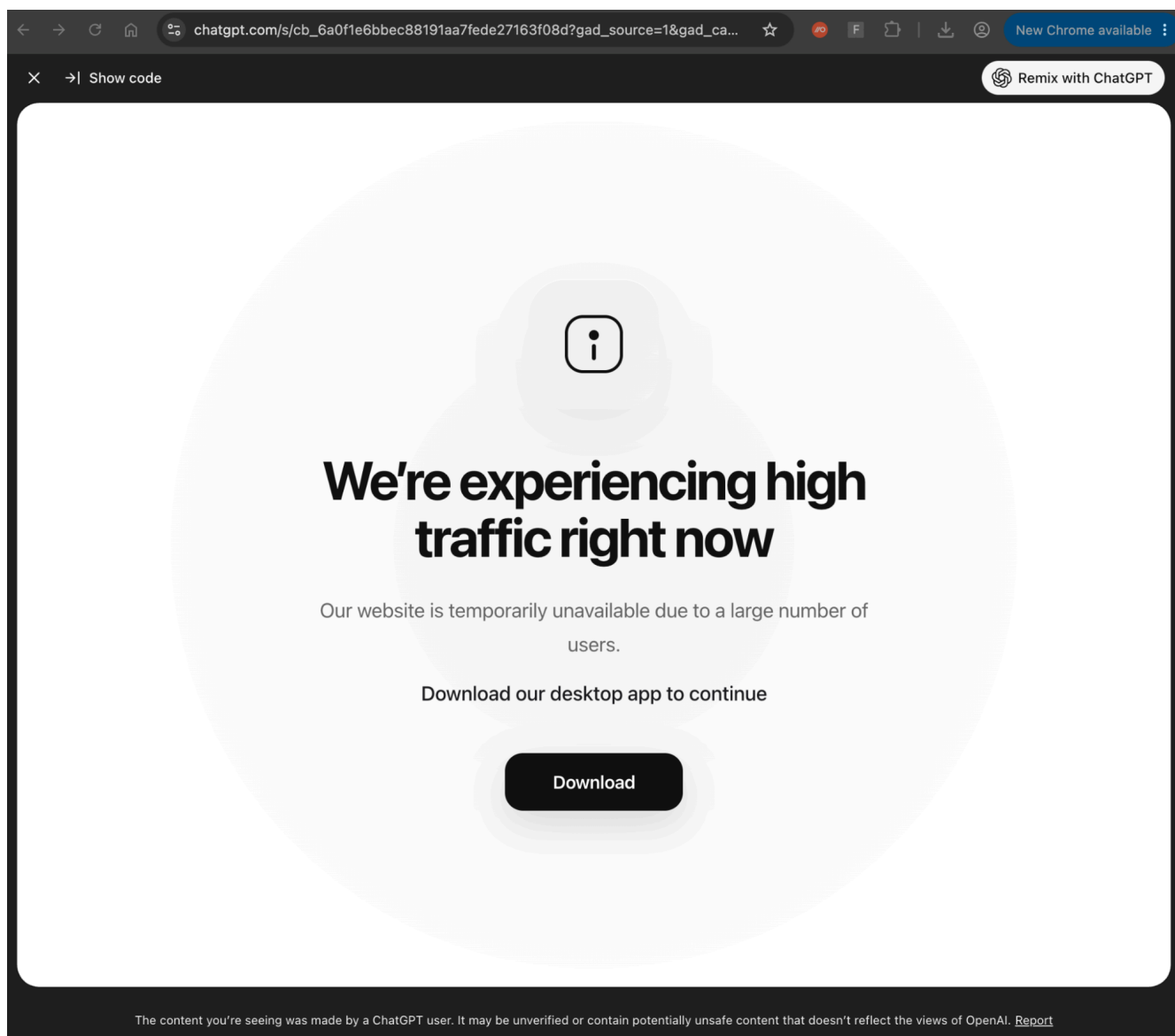


These are essentially InstallFix attacks — a variant of the ClickFix family that [Push documented earlier this year](#) — and they exploit the fact that AI tools have normalized command-line installation workflows for a population of users who lack the experience to distinguish a legitimate terminal command from a malicious one.

This is a live campaign which is still generating detections across our customer base at the time of writing. Push customers are already protected and do not need to take further action. The malicious page URLs can be found at the end of this report but are not exhaustive and are liable to change.

Previously reported variants relied on shared *conversations* — the attacker created a chat that contained step-by-step instructions for the victim to follow, typically involving pasting a command into their terminal. The social engineering was conversational: the "AI assistant" appeared to be helpfully guiding the user through an installation process.

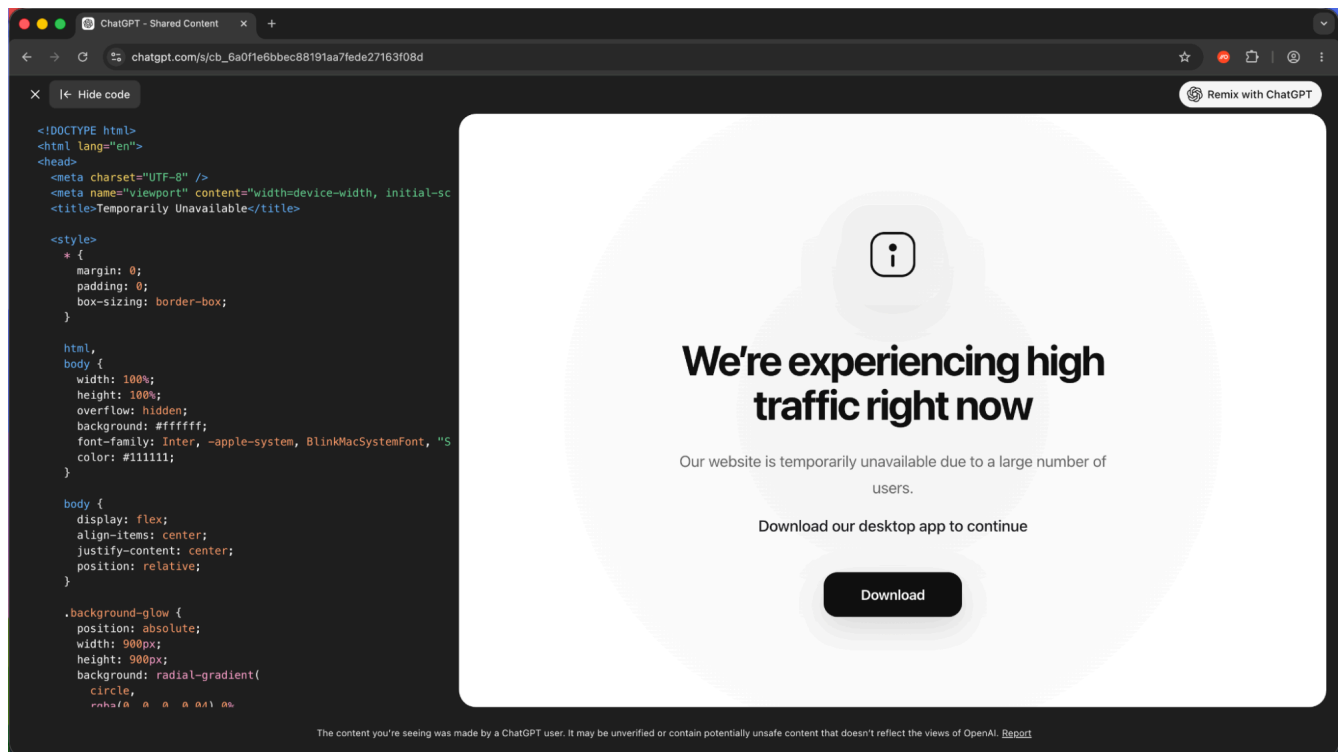
But now, rather than a shared conversation, the attacker has used ChatGPT's code rendering feature to create a fully designed, self-contained web page hosted at a `chatgpt.com/s/` URL. It renders as what appears to be a ChatGPT service disruption notice:



The fake "high traffic" page rendered inside a ChatGPT shared content URL. Note the "Show code" and "Remix with ChatGPT" buttons at the top, which reveal that this is actually rendered HTML/CSS code rather than a real ChatGPT system page.

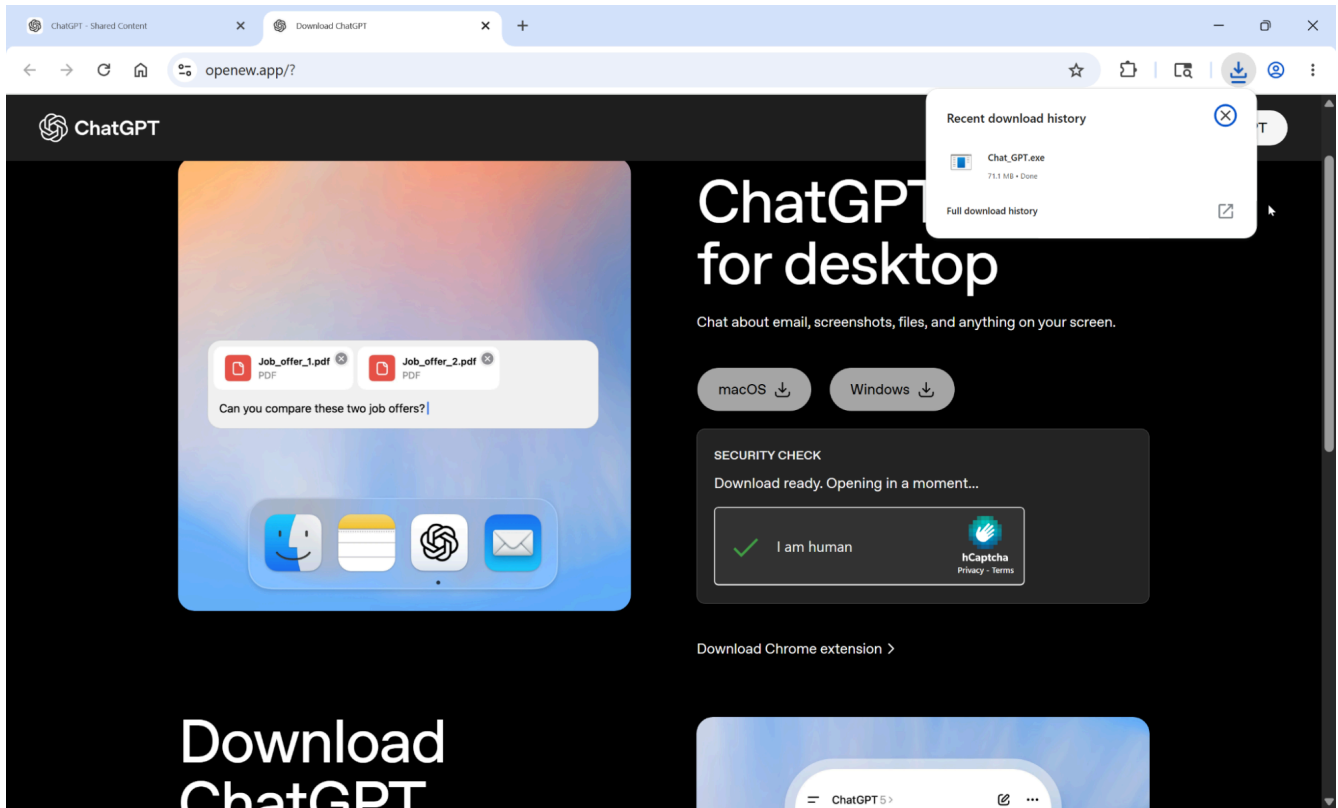
A professional-looking error message reads: "We're experiencing high traffic right now. Our website is temporarily unavailable due to a large number of users. Download our desktop app to continue." A prominent download button sits below.

The "Show code" toggle at the top of the page reveals what's actually happening — the entire thing is custom HTML and CSS, authored to mimic a ChatGPT system notice, rendered using ChatGPT's code output feature. A web page inside a web page, hosted on a domain that every URL reputation system in the world considers safe.

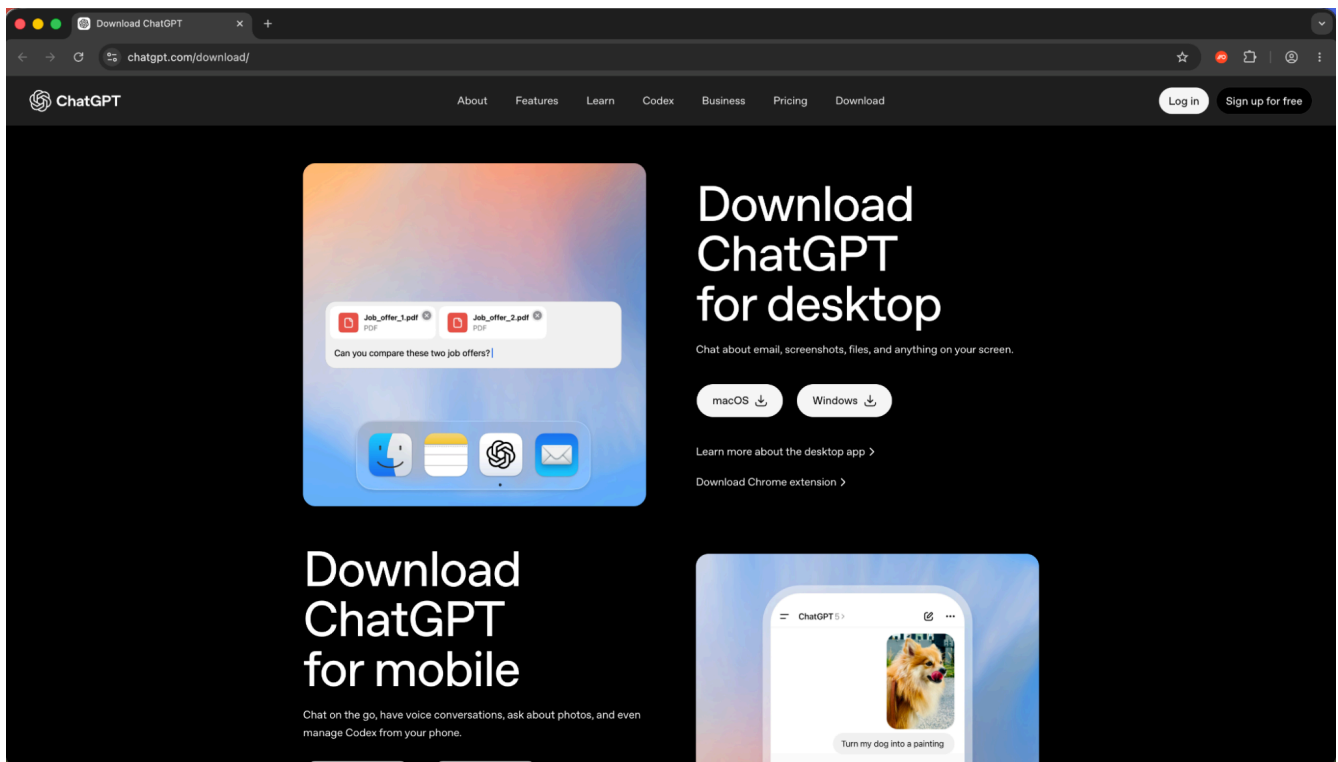


The same page with the code panel open, showing the HTML/CSS source code that generates the fake service disruption notice.

Clicking the download button redirects the user to `openew[.]app`, which presents a convincing clone of ChatGPT's official desktop application download page — complete with OpenAI branding, macOS and Windows download buttons, a Chrome extension link, and a mobile download section.



The fake ChatGPT download page hosted at openew[.]app. The design closely replicates OpenAI's legitimate download page.



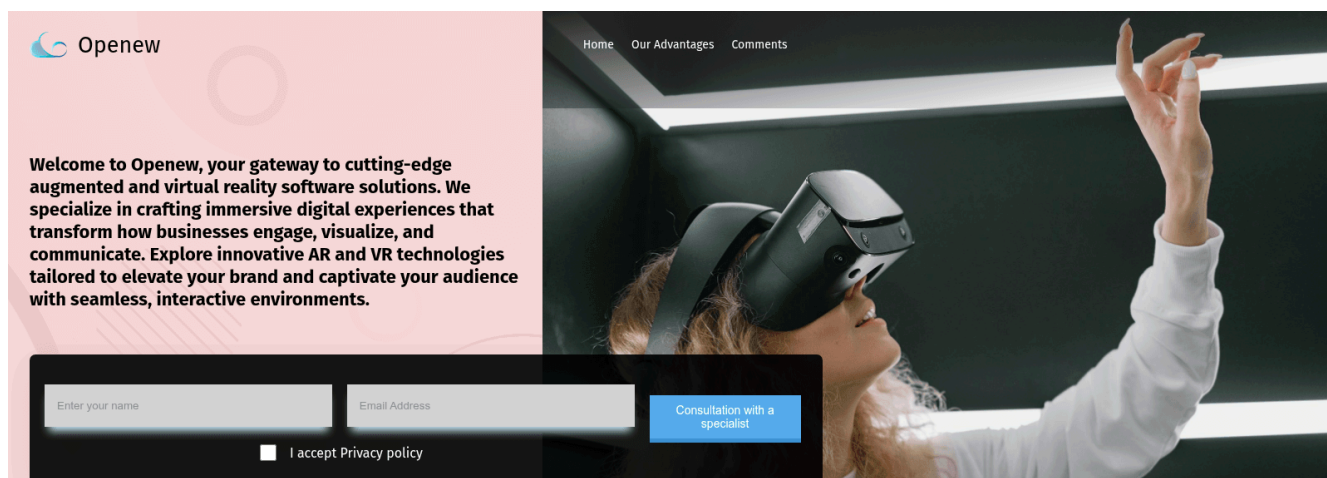
Real ChatGPT download page for comparison chatgpt.com/download.

The site also displays differently depending on who visits it. When Push researchers examined the URL via URLScan, the scanner was redirected to a different page entirely — a

generic AR/VR company website with no obvious connection to ChatGPT.

Real users in a browser see the fake download page; automated scanners and bots see something benign. This kind of conditional rendering is a well-established evasion technique in the malvertising ecosystem, and it makes the malicious infrastructure harder for security teams and threat intelligence services to identify and analyze.

The downloaded executable poses as "ChatGPT for Desktop" and is [flagged on VirusTotal](#).



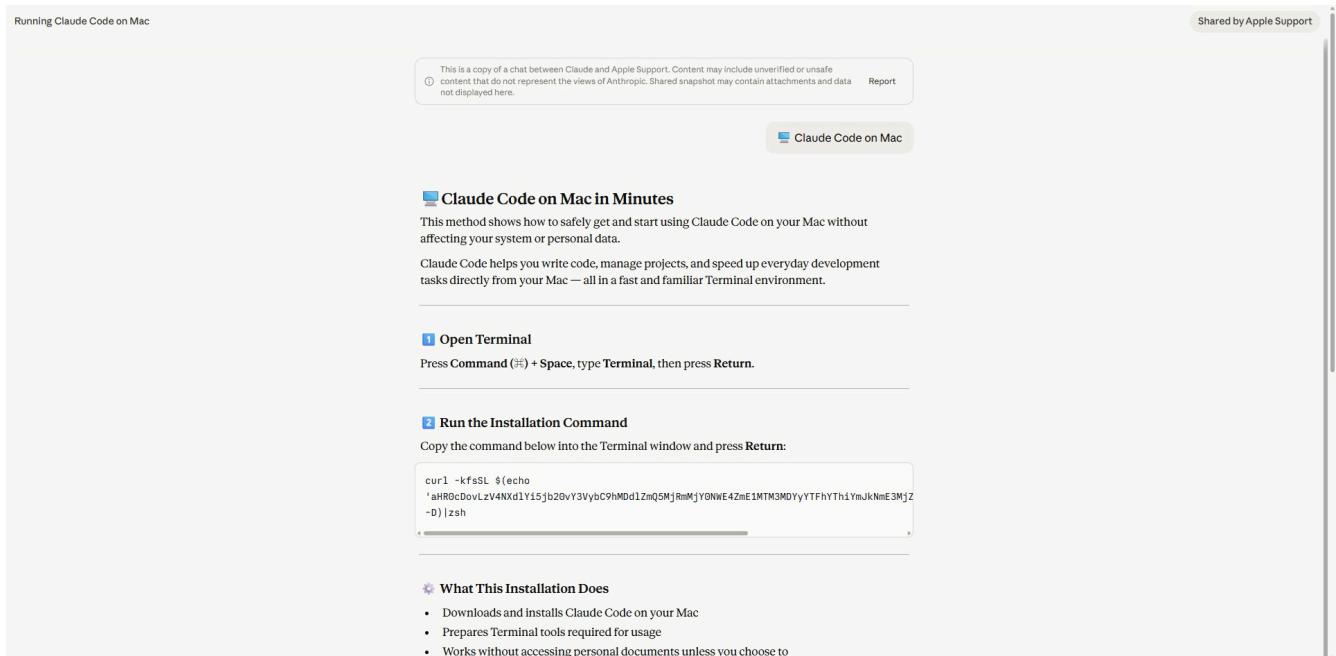
+34714 Questions closed

Why did you contact us:



What URLScan sees when visiting the same openew[.]app URL: a generic "Openew" AR/VR company website with no trace of the ChatGPT impersonation.

Alongside the ChatGPT rendered-page variant, Push has also detected the previously reported style of attack using shared Claude.ai conversations. These follow the pattern documented by [BleepingComputer](#): a shared chat disguised as a "Claude Code on Mac" installation guide, attributed to "Apple Support," containing a curl command that downloads and executes malware.

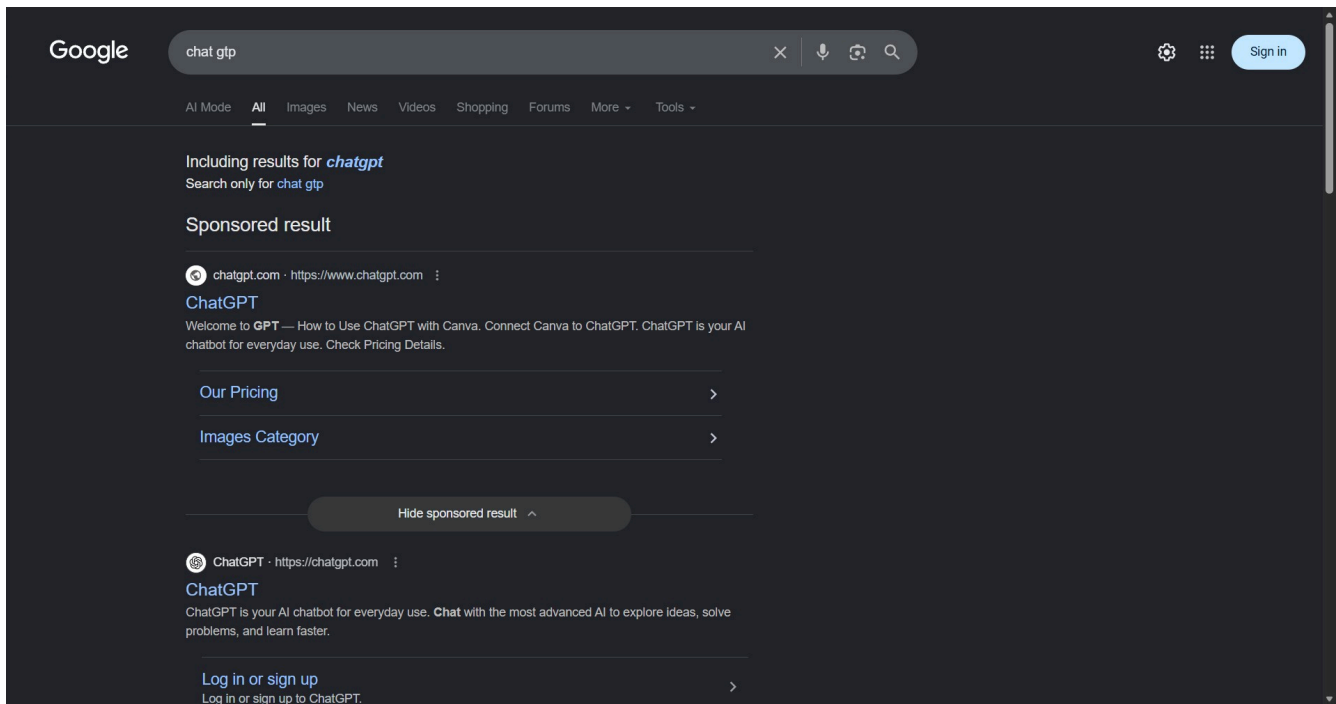


A shared Claude.ai conversation containing malicious installation instructions in the style previously reported by BleepingComputer.

The fact that both the ChatGPT and Claude variants are appearing in Push customer environments suggests a campaign — or at least a shared playbook — that is actively experimenting with different platforms and different social engineering approaches to find what converts best.

Push has detected this variant across multiple customer environments, with users arriving at these shared chat URLs after searching for terms including "chatgpt," "chatgpt free," "chat gpt," and common typos like "chatgo," "chatgot," and "cvhatgpt."

You can see an example of this below: it's incredibly convincing, and uses the real ChatGPT domain — so even users that are paying attention are liable to fall for it.



Although we managed to grab that example, the ads haven't been easy to reproduce. This is because the ads are likely geographically or temporally scoped. It's pretty eye-opening (and creepy) how tightly scoped these kinds of sponsored ads can be across different platforms.

This is one of the key misconceptions people can have about this kind of attack. It's easy to see it as untargeted, when realistically it can be scoped tightly to a desired victim population by role, geography, and so on. We've written about this previously in [our blog](#) on the ad account takeover > malvertising ecosystem.

This fits a pattern Push has tracked extensively. [Search-based delivery is now the dominant channel for malware distribution](#) — our own data shows that ClickFix attacks are reached via search results rather than email in 4 of 5 cases, and Push's own research into [malvertising campaigns impersonating brands like TradingView](#) and [Ahrefs](#) has demonstrated how effectively search ads can funnel victims to malicious pages.

The shared-chat technique adds a new dimension: the destination URL itself is genuine (chatgpt.com, claude.ai), which means even a cautious user who checks the URL before clicking will see nothing suspicious.

This is one example of a much broader pattern that has become one of the defining characteristics of the 2026 threat landscape: attackers systematically abusing legitimate platforms as attack infrastructure. The scale and variety of this abuse in recent months alone is striking, and it spans every stage of the phishing chain.

On the delivery side, attackers have been [weaponizing stolen AWS credentials to send phishing through Amazon SES](#) that passes SPF, DKIM, and DMARC validation because SES is a legitimate Amazon service. A Vietnamese operation dubbed [AccountDumping used Google AppSheet's built-in email capability](#) as a phishing relay to harvest 30,000 Facebook credentials. [Scammers exploited Microsoft's own internal notification pipeline](#) — sending phishing from the same `msonlineserviceteam@microsoftonline.com` address that delivers legitimate 2FA codes — with Spamhaus confirming months of ongoing abuse.

For hosting, the platforms being abused read like a who's who of modern web infrastructure. [Operation HookedWing ran for four years](#) on GitHub Pages and Vercel, compromising 500+ organizations across more than 100 GitHub Pages domains before anyone documented it publicly. Cofense has separately [documented the growing abuse of Vercel](#) for credential phishing hosting. Pixm's Q1 2026 phishing report tracked over 100 unique Azure Blob Storage subdomain variants hosting phishing content that carried Microsoft's own domain reputation, alongside abuse of Cloudflare CDN, Cloudflare Workers, Cloudflare R2, Backblaze B2, and Supabase.

Compromised legitimate sites are also being repurposed at scale. A mass exploitation of a [Ghost CMS vulnerability planted ClickFix pages across 700+ websites](#) including Harvard, Oxford, and DuckDuckGo subdomains. Microsoft recently documented a campaign where [SEO poisoning was combined with AI chatbot recommendation manipulation](#) to deliver GPU mining malware — extending the poisoning from traditional search results into AI-generated software recommendations. And [fake ChatGPT and Claude installers on GitHub and SourceForge](#) have been delivering the DinDoor backdoor and a Deno-based RAT via repositories that mimic legitimate developer tool distributions.

The structural problem is that every one of these platforms is genuinely legitimate, and the security controls that evaluate them — domain reputation, email authentication, URL categorization — confirm them as trusted because they are trusted. This attack extends this pattern into new territory by weaponizing the content-sharing features of AI chatbot platforms specifically, but the underlying principles are the same.

Shared-chat malware delivery exploits a structural property of AI platforms that traditional security controls aren't designed to handle. Domain reputation, URL categorization, and safe browsing databases all treat `chatgpt.com` and `claude.ai` as trusted — because they are. Using these trusted pages to link off to further convincing-looking pages hosting malware allows the attacker to run campaigns that blend in, as well as rotate the phishing delivery pages later in the chain should they ever be flagged, allowing the campaign to continue without interruption (a well known [detection evasion technique](#)).

What makes the rendered-page variant particularly concerning is that it eliminates the most obvious red flag in the earlier attacks. The Claude.ai conversation variants required the victim to recognize that a shared chat instructing them to paste terminal commands might be suspicious — a tall order for many users, but at least the attack surface was visible. The rendered-page variant shows nothing that looks like an attack. It presents what appears to be a routine service disruption with a reasonable call to action: download the desktop app to continue using ChatGPT.

We've aligned our detection logic for this technique under the name **LLMShare** — a technique-level detection that covers shared content abuse across LLM platforms, not tied to any single campaign or set of IOCs.

Because Push sees the full context of how a user arrived at a page and what that page does once it renders, we can identify LLMShare attacks regardless of which AI platform is being abused or what social engineering wrapper the attacker has chosen.

When we identified the initial instances of this campaign, we used our [agentic threat hunting pipeline](#) to hunt for additional examples across our customer telemetry, develop the LLMShare detection, and rapidly deploy it to customers. Push blocks users from interacting with the page before any malicious activity can occur.

Push customers do not need to take any further action.

Push Security is the most powerful AI-native security tool in the browser. Think EDR, but for the browser — high-fidelity telemetry and real-time control across every session, on every device, with no browser migration required.

Security teams use Push to detect and stop advanced browser-based attacks like AiTM phishing, ClickFix, and session hijacking; gain visibility and control over AI tool usage across their workforce; harden identities by surfacing credential reuse, SSO gaps, and shadow IT; and support data loss and insider investigations with browser-layer telemetry that other tools can't see.

[Book a live demo to learn more.](#)

As we always say, short-lived IOCs are of limited value when tackling modern phishing attacks due to the rate at which attackers are able to [quickly spin up and rotate the sites used](#) in the attack chain. IoC-based detections for campaigns like this are of limited value.

At the time of writing, the indicators observed were:

Indicator	Type
hxxps://claude[.]ai/share/8e6401b5-4849-46c4-a3cb-29e1c3c49131	URL
hxxps://chatgpt[.]com/s/cb_6a0f1e6bbec88191aa7fede27163f08d	URL
openew[.]app	Domain
de8c50e8ccd240ef9d10ec26c26eeb37a4d1cad7c1e0edf3bb6e5689ec2dde78	SHA256