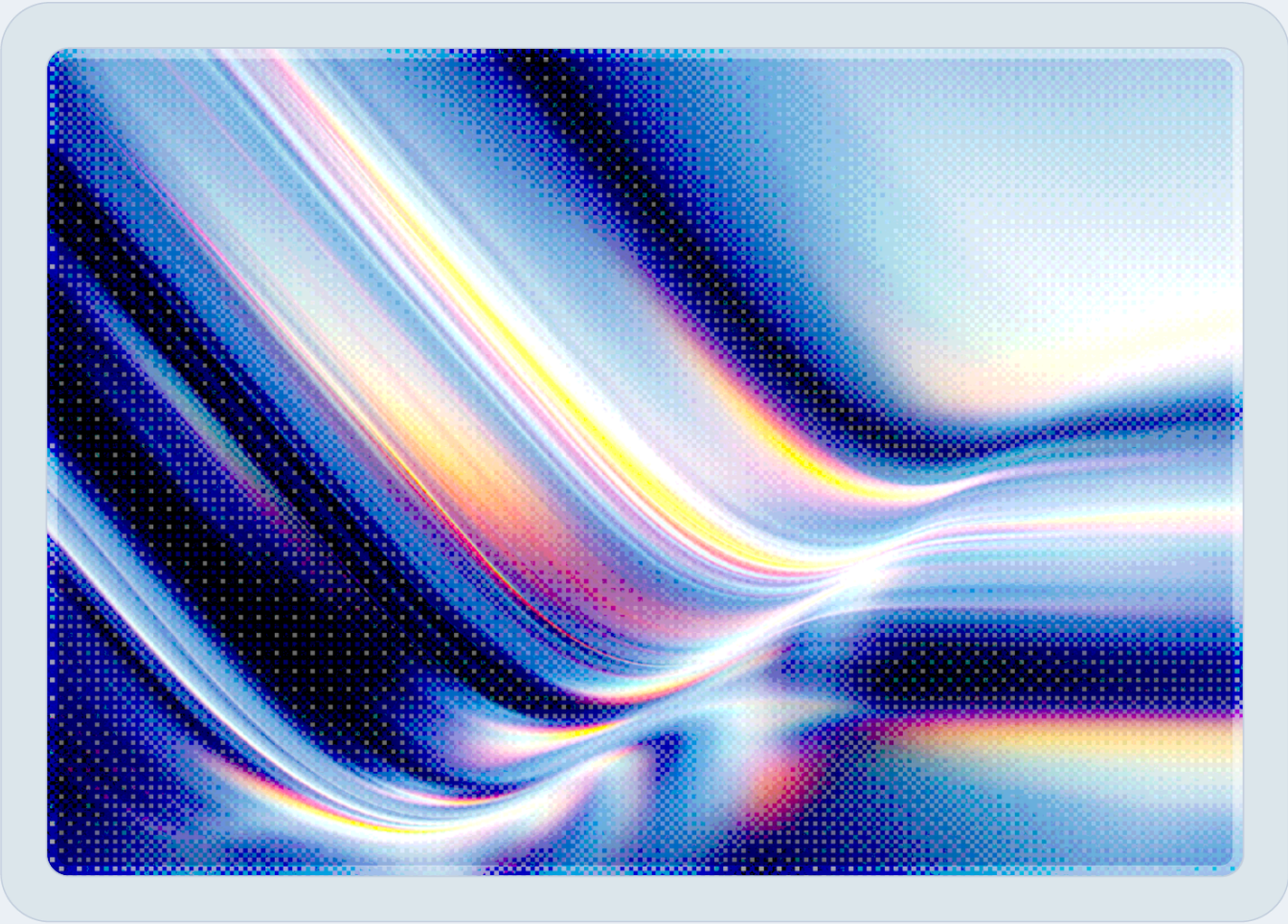
CSAI Foundation | Cloud Security Alliance

Claude for Chrome: Synthetic Click Flaw Lets Extensions Hijack AI Actions

2026-07-18

AI-assisted Rapid Research



CSAI

CSA cloud security alliance®

Key Takeaways

Independent researchers at Manifold Security disclosed that Anthropic's Claude for Chrome browser extension fails to verify whether a click approving an AI-driven task originated from a real user, allowing any co-installed malicious extension to forge the interaction and silently trigger privileged agent workflows [1][2]. The flaw exploits the absence of a check on the browser's native `Event.isTrusted` property, meaning a script running on claude.ai from an unrelated extension can fabricate an approval click in as little as six lines of JavaScript [1]. Because Claude for Chrome ships with built-in access to Gmail, Google Docs, Google Calendar, and connected services such as Salesforce, DoorDash, and Zillow, a successful forgery can cause the agent to read Gmail messages, Google Docs content, or calendar entries, and interact with those connected integrations, without the user ever clicking anything themselves [2][3]. Manifold reported the issue to Anthropic in May 2026; as of version 1.0.80, released July 7, 2026, the underlying code path remains functionally unchanged from the version first reported, even though Anthropic marked the tracking issues as resolved [1][4]. A second, related finding shows that Claude's side panel can be switched into a reduced-confirmation mode through a `skipPermissions=true` URL parameter, a design where a sensitive privilege state depends on an inspectable and potentially forgeable URL value rather than an authenticated user action [2][5]. This case is a useful illustration of a risk worth watching more broadly in agentic browser tools: the mechanisms used to establish "the user asked for this" may be weaker than the privileges the agent is granted once that condition is assumed true.

Background

Claude for Chrome is Anthropic's browser extension that lets the Claude model observe a user's active tab, read page content, and take actions on the user's behalf inside connected web applications. Anthropic has kept the extension in beta and, following an earlier disclosed weakness the security community has referred to as "ClaudeBleed," restricted the agent's autonomous behavior to a fixed menu of nine predefined tasks that a user can approve with a single click: three onboarding prompts, three third-party integrations (DoorDash, Salesforce, and Zillow), and three tasks that read from Google services, covering a user's Gmail messages, the latest Google Doc and its comments, and calendar entries [3]. That restriction was intended to shrink the space of actions an attacker-influenced agent could take; SecurityWeek reported that Anthropic characterized the list of pre-approved tasks as an

initial mitigation for ClaudeBleed pending a complete fix [5]. The redesign addressed *what* the agent was allowed to do but left open a separate question the new findings expose: *who* is authorized to ask the agent to do it.

Manifold Security's research, conducted by Ax Sharma, found that the click handler responsible for launching one of these nine predefined tasks does not check whether the triggering click event carries a legitimate `Event.isTrusted` flag, a boolean the browser itself sets to distinguish a real pointer action from a script-generated one [1][2]. Because Chrome extensions with content-script access to a given origin are common and routinely granted broad host permissions, any extension already present in the browser and able to inject a script into claude.ai can dispatch a synthetic click that Claude's extension processes identically to a genuine one [1]. In practical terms, a second extension installed for an unrelated purpose, one the user trusts for note-taking, ad-blocking, or price comparison, could carry hidden logic that reaches into the Claude tab and fires the approval sequence for a task the user never requested. Manifold demonstrated the exploit in roughly six lines of JavaScript and noted that Anthropic's own proposed remediation, checking `isTrusted` before executing the workflow, was a single added line of code [1][2].

Security Analysis

The core defect is an authentication gap: a UI convenience feature, the single-click approval, that was never paired with verification that the click actually came from a human. Claude for Chrome's design correctly recognizes that letting an AI agent take actions in a user's Gmail or Salesforce account is high-risk enough to require an explicit approval gesture. Where the design fails is in verifying that the gesture came from a human rather than from arbitrary code with DOM access to the same page. This distinction matters because a browser extension's threat model is fundamentally multi-tenant: any tab can, and routinely does, host script execution from several unrelated extensions simultaneously, each operating under whatever host permissions the user granted at install time. An approval mechanism that trusts the DOM event stream without checking its provenance effectively treats every extension on the page as equally authorized to speak for the user, collapsing a security boundary that the browser itself was designed to preserve through the `isTrusted` property.

The nine predefined tasks were chosen, in Anthropic's own account, specifically to bound the blast radius of a compromised approval flow, yet each of the nine still grants access to a real productivity, CRM, or consumer-service account [3]. Reading Gmail messages, reviewing Google Docs comments, or interacting with the Salesforce, DoorDash, and Zillow integrations are not low-sensitivity operations merely because they were pre-approved as a category; they expose personal correspondence, business-sensitive documents, and third-party account data to any co-resident extension capable of forging a

click. A user who has never interacted with Claude for Chrome that session, and who has no reason to suspect anything happened, could have email, documents, or calendar entries read without any visible prompt, since the synthetic click can substitute for the confirmation step entirely rather than merely pre-filling it [1][2]. The attack also does not require compromising Anthropic's infrastructure, the Claude model, or the user's accounts directly; it only requires that one other extension already installed in the browser be malicious or compromised, a considerably lower bar than the compromises that AI agent security assessments often focus on.

The second finding, the `skipPermissions=true` side-panel parameter, compounds this concern architecturally even though researchers and Anthropic agree it is not currently reachable by an external attacker because only the extension itself constructs the relevant URLs [2][5]. The design pattern of encoding a privileged-mode toggle in a URL value, rather than deriving it from an authenticated session state or a signed capability token, is fragile by construction: it works safely only for as long as no other code path, browser extension, or future feature is able to construct or influence that URL. Treating a URL parameter as a security boundary is a known anti-pattern precisely because URLs are frequently logged, cached, shared, and manipulated by code that never anticipated carrying security-relevant state.

Anthropic's handling of the disclosure raises a separate governance concern independent of the technical root cause. According to the researchers, both issues were acknowledged within a day of reporting in May 2026, and the tracking tickets were subsequently marked resolved, yet independent testing found the content-script and side-panel handler code in version 1.0.80, released July 7, 2026, to be functionally identical to the version originally reported eight releases earlier, version 1.0.72 [1][4]. Whether this reflects a substantive disagreement about severity, a mitigation that addressed a narrower internal report without covering the externally reported reproduction path, or a tracking error, the practical effect for defenders is the same: the vulnerability that security researchers can reproduce today is the one enterprises must plan around, regardless of how an internal ticket is labeled. This gap between reported status and observed behavior is a pattern security teams evaluating any AI agent vendor should watch for directly, by re-testing disclosed issues against shipped releases rather than relying on vendor status labels alone.

Recommendations

Immediate Actions

Organizations that have deployed Claude for Chrome should audit which other browser extensions are installed on the same endpoints, since the attack requires a second, co-resident extension capable of script injection into claude.ai, and any unrecognized or unnecessary extension should be removed

regardless of its stated purpose [1]. Security teams should also confirm whether users have enabled any "act without asking" or reduced-confirmation setting in Claude for Chrome and disable it, since removing the human-in-the-loop checkpoint removes the last layer of defense against a forged approval, however weak that layer currently is [4]. Where Claude for Chrome is connected to high-sensitivity accounts, particularly primary email, document repositories, or CRM systems holding customer data, administrators should consider disabling the extension entirely until Anthropic ships an independently verifiable fix, or restrict its use to dedicated, low-privilege browser profiles that are not used for other extension-dependent browsing [3][4].

Short-Term Mitigations

Enterprises that manage browser extensions centrally should extend existing extension allowlisting and permission-review processes to explicitly account for AI browser agents as a new category of high-privilege extension, since the interaction surface between two arbitrary extensions on the same page is not something traditional extension vetting programs were built to evaluate. Security teams piloting or operating agentic browser tools more broadly should require vendors to demonstrate that user-approval mechanisms verify event provenance, for example via `Event.isTrusted` or an equivalent authenticated confirmation channel, rather than assuming that any DOM-level click is user-initiated, and should treat vendor claims of "resolved" status with the same skepticism applied to any other unverified patch claim, retesting against the currently shipped release. Where agent tools expose privileged modes or reduced-confirmation states, organizations should ask vendors whether those states can be triggered by any input other than an authenticated user action, since URL- or query-parameter-driven privilege escalation is a pattern that recurs across agentic tooling generally, not solely in this product.

Strategic Considerations

This disclosure is a specific instance of a broader authorization problem that CSA's Agentic AI Red Teaming Guide catalogs under agent authorization and control hijacking: an autonomous or semi-autonomous agent's actions are only as trustworthy as the mechanism confirming that a legitimate principal requested them, and that mechanism must be evaluated independently of how sensible the agent's resulting actions appear [6]. Organizations building internal governance for agentic browser and desktop tools should require, as a baseline procurement and deployment criterion, that any component accepting a "confirm" or "approve" signal for a high-privilege action be evaluated for spoofability by co-resident or adjacent processes, not merely for whether the resulting workflow itself is appropriately scoped. As enterprises expand adoption of AI agents that integrate directly into browsers, email clients, and productivity suites, the multi-tenant nature of these environments, where an agent extension shares

a runtime with dozens of other unrelated extensions or applications, should be treated as a first-class element of the threat model rather than an edge case, since the attacker does not need to breach the AI vendor at all if a weaker adjacent trust boundary is available to exploit.

CSA Resource Alignment

This incident maps directly to authorization and control-hijacking scenarios addressed in CSA's Agentic AI Red Teaming Guide, which provides structured test procedures for exactly this class of weakness under its "Agent Authorization and Control Hijacking" category, including guidance on validating that agent-triggering signals cannot be spoofed by adjacent, lower-trust code [6]. The guide's emphasis on testing whether an agent correctly distinguishes an authenticated user request from an injected or forged one is directly applicable to any organization evaluating browser-based AI agents such as Claude for Chrome before or during deployment. This case also aligns closely with CSA's Securing LLM Backed Systems: Essential Authorization Practices, which catalogs authorization pitfalls and design patterns specific to LLM-integrated systems and speaks directly to the unverified-click-provenance failure at the center of this disclosure [8]. Because no CSA artifact in the corpus addresses browser-extension-specific AI agent authorization as a named topic, this analysis also draws on the identity and access management guidance within the AI Controls Matrix (AICM) v1.1, which organizations can use to assess whether a given AI tool's authorization and session-integrity controls meet baseline expectations before granting it access to email, document, and CRM systems [7]. Together, these resources give security teams a structured way to move beyond this single disclosure and build a repeatable evaluation process for the next agentic browser tool that requests the same category of access.

References

[1] Sharma, Ax / Manifold Security, reported via BleepingComputer. "Claude Chrome extension flaw lets malicious extensions trigger AI actions." BleepingComputer, July 16, 2026.

[2] CSO Online. "New bugs in Claude for Chrome allow extensions to abuse AI privileges." CSO Online, July 2026.

[3] The Hacker News. "Researchers Say Claude for Chrome Flaw Lets Rogue Extensions Trigger Gmail Reads." The Hacker News, July 2026.

[4] Malwarebytes. "Claude for Chrome flaw could let rogue extensions access your Gmail." Malwarebytes Labs, July 2026.

[5] SecurityWeek. "Unpatched Claude for Chrome Flaw Lets Extensions Read Gmail, Calendar." SecurityWeek, July 14, 2026.

[6] Cloud Security Alliance. "Agentic AI Red Teaming Guide." CSA, May 2025.

[7] Cloud Security Alliance. "AI Controls Matrix (AICM) v1.1." CSA, June 2026.

[8] Cloud Security Alliance. "Securing LLM Backed Systems: Essential Authorization Practices." CSA, August 2024.

