Hidden prompt injection on GitHub: what a study of 145 million items found

27 November 2026 · 5 min read · blogs.iictp.org/hidden-prompt-injection-github-coding-agents-study

Short answer: hidden prompt injection on GitHub is AI-directed text that a human reader does not see but a coding agent receives in full, and Gabi Nakibly and Zvika Babo published a large measurement of it on arXiv on 21 November 2026. The authors report that outright attacks are rare, while projects using hidden markers to steer their own agents are common and growing. It matters because anyone who can comment can imitate those markers.

TL;DR The short version

- Hidden prompt injection on GitHub is AI-directed text a human reader cannot see but a coding agent receives in full, measured across 145 million items in an arXiv study of 21 November 2026.
- Outright attacks were rare, but projects using hidden markers to steer their own agents are common and growing.
- Anyone who can comment can imitate those markers, so treat repository text as untrusted input for agents and limit what they can do.

Read online, listen to it, or change the text size: https://blogs.iictp.org/hidden-prompt-injection-github-coding-agents-study

What is hidden prompt injection on GitHub?

Indirect prompt injection means instructions placed in untrusted content that steer a model which later processes it. On GitHub, agents triage issues, review pull requests and pick up tasks, and the text they read can come from any account. The study looks only at text that GitHub does not render for people, such as content inside HTML comments, but that an agent reading the raw source still receives.

How the measurement worked

1. Collect roughly 145 million issues, pull requests, comments and README files.
2. Detect text that GitHub hides from the rendered page.
3. Filter it with a two-stage language-model classification.
4. Check whether the text comes from a repeated template.
5. Review every remaining item by hand under written rules.

What is new compared with lab demonstrations

Earlier work showed that attacks on GitHub agents are possible in controlled settings. This paper asks how often they occur in public, who places them and why. The authors also separate attackers from legitimate users of the same technique, which matters for anyone writing a filter.

Results reported by the authors

- 2,281 validated findings in issues, pull requests and comments, which the authors put at 16 per million items, plus AI-directed hidden text in the READMEs of 107 repositories.
- Four kinds of use: attacks on other parties' agents, defensive text that restricts agents, content override that steers what agents write, and orchestration of a project's own agents.
- Attacks were about one in 3.3 million items, aimed at maintainers' triage, review and continuous-integration agents.
- Orchestration made up two thirds of findings in issues, pull requests and comments, and grew fivefold in four months.
- Replayed offline against five models, the attacks succeeded against one small, fast model in 19% of cases, and against none once HTML comments were stripped.

This is an arXiv preprint or workshop paper. Treat its numbers as the authors' own results, not as independently confirmed findings.

Limitations and open questions

- Only hidden text is counted. Injections written in plain view, in commit messages, in code or in agent instruction files are outside the census, so the authors call their counts lower bounds.
- Comment coverage came from a sample that held only a small share of activity, so orchestration is likely undercounted.
- Manual review removes false positives but cannot recover items the classifiers missed.
- Telling a security probe from an attack involves judgment. The authors publish a review note for each finding.
- An offline replay against five models does not show how a given agent with tools and permissions would behave.

Sources were read on 27 November 2026. Products, rules and data change, so check the linked originals before you rely on any detail.

How to protect your repository agents

1. Give an issue-reading agent no secrets and no write access it does not need.
2. Set workflow token permissions explicitly and keep cloud credentials out of jobs that an outsider's comment can trigger.
3. Never let a model's verdict open a security gate on its own.
4. Check who wrote a marker before acting on it, using an allowlist or the author's association with the project.
5. Filter hidden text by author: strip what outsiders write and keep what members and your own bots write.
6. For approvals, prefer visible signals that only maintainers can set, such as labels or check status.

The authors released their detection tool and findings at github.com/rdwr-gabin/github-ipi-in-the-wild. Review any third-party tool before running it.

What it means for developers and platform teams

Treat repository text as untrusted input to a program that has credentials. The useful control is not a smarter prompt but a narrower job: limited tokens, a verified author for anything that grants permission, and a human decision for merges. How the Codex sandbox limits files and network access covers the machine-level half of the same problem.

Frequently asked questions

- **What is hidden prompt injection on GitHub?** Text aimed at AI agents that GitHub does not show to people but that agents can read.
- **Are such attacks common?** The authors found them rare, about one in 3.3 million items among those they could detect.
- **Does stripping HTML comments solve it?** In their offline replay it stopped the attacks tested, but it also breaks legitimate project workflows.
- **Who is targeted?** Mainly maintainers' own triage, review and continuous-integration agents.

Related reading

- What is prompt injection?
- Codex sandbox for AI coding agents
- What is least privilege?
- AI Practitioner certification

> Related credential: AI Practitioner. Explore the exam facts and syllabus.

Sources and further reading

1. Gabi Nakibly and Zvika Babo. Who Is Prompt-Injecting Coding Agents? An Empirical Study of Hidden AI-Directed Text on GitHub. arXiv 2611.28763. 21 November 2026 (preprint). arxiv.org/abs/2611.28763

Published by IICTP, an independent technical credential body, on 27 November 2026. The current version of this post is at https://blogs.iictp.org/hidden-prompt-injection-github-coding-agents-study. More guides: blogs.iictp.org.