GlassWorm Returns: New Wave Strikes as We Expose Attacker Infrastructure

By Idan Dardikman, Yuval Ronen, Lotan Sery

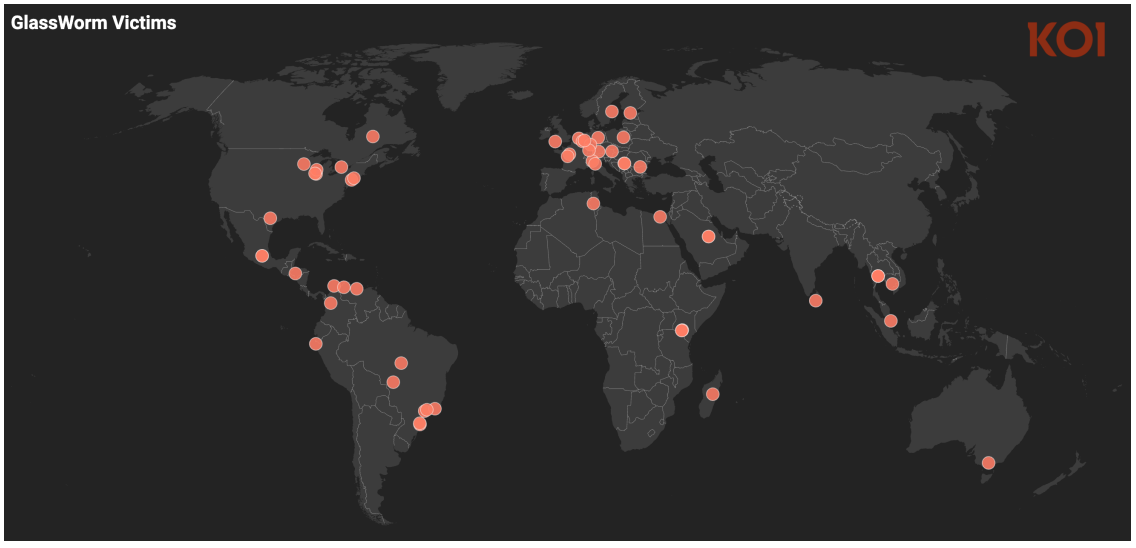
Published: 2025-11-06 · Archived: 2026-04-29 02:07:33 UTC

Almost three weeks ago, we disclosed GlassWorm - the first self-propagating worm targeting VS Code extensions, using invisible Unicode characters to hide malicious code that literally disappears from code editors.

On October 21, 2025, OpenVSX declared the incident "fully contained and closed."

But on November 6, 2025 - sixteen days later - we detected a new wave of GlassWorm infections. Three more extensions compromised. A fresh Solana blockchain transaction providing new C2 endpoints. Same attacker infrastructure, still fully operational.

But here's where this story gets more serious. We managed to access the attacker's server. What we found inside confirmed the real-world impact: a partial list of victims from around the world - the US, South America, Europe, Asia - including a major government entity from the Middle East.



Wold map of GlassWorm victims

This isn't just about compromised extensions anymore. This is about real victims, critical infrastructure at risk, and a worm that's doing exactly what we warned it would do: spreading through the developer ecosystem.

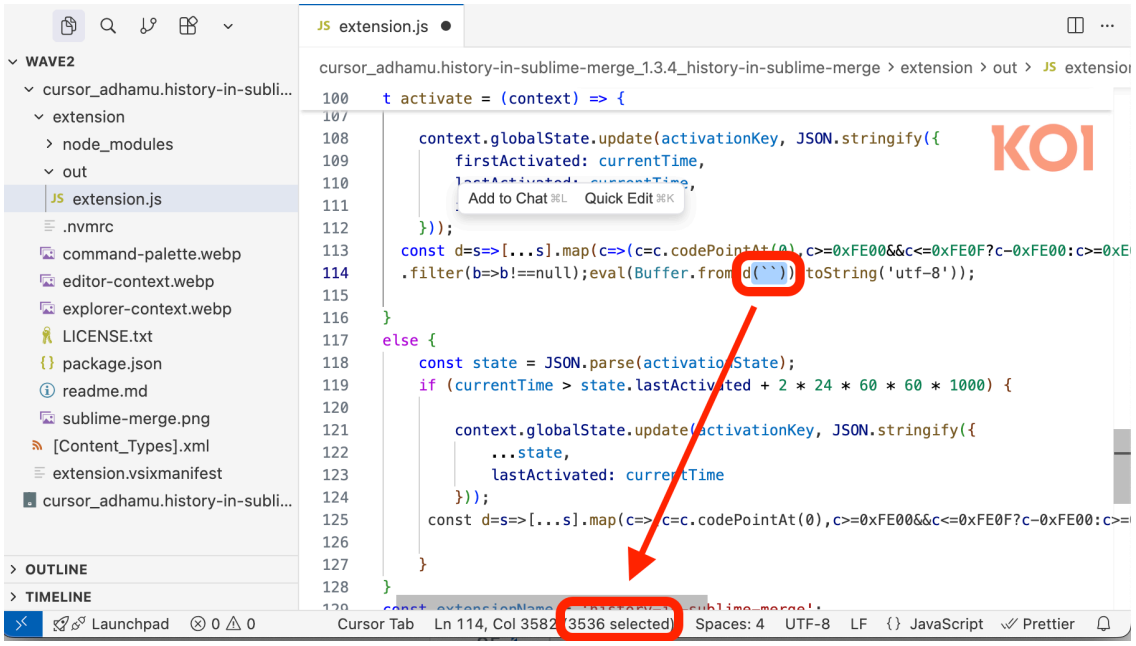
And it's not just OpenVSX. Developers have reported that GlassWorm has jumped to GitHub repositories, using AI-generated commits to hide its invisible payloads in what looks like legitimate code changes.

The New Wave: Three More Extensions Fall

On November 6, 2025, our risk engine flagged three more OpenVSX extensions showing the exact GlassWorm signature:

- **ai-driven-dev.ai-driven-dev** (3,300 downloads)
- **adhamu.history-in-sublime-merge** (4,000 downloads)
- **yasuyuky.transient-emacs** (2,400 downloads)

Total impact from this wave alone: approximately 10,000 additional infections.



The invisible payload in the new wave of GlassWorm

All three extensions contain invisible Unicode malware very similar to what we documented in our original analysis. The malicious code is still literally invisible in code editors - encoded in unprintable Unicode characters that render as blank space to human eyes but execute as JavaScript to the interpreter.

The attacker has posted a fresh transaction to the Solana blockchain, providing an updated C2 endpoint for downloading the next-stage payload. This demonstrates the resilience of blockchain-based C2 infrastructure - even if payload servers are taken down, the attacker can post a new transaction for a fraction of a cent, and all infected machines automatically fetch the new location.

Notably, while the Solana transaction is fresh, the C2 and exfiltration servers remain unchanged from our original analysis:

- **199.247.10.166** (primary C2 server)
- **199.247.13.106:80/wall** (exfiltration endpoint)

The infrastructure we documented a month ago is still operational. Still serving payloads. Still collecting stolen credentials.

We Got Inside: What the Attacker's Server Revealed

Here's where our investigation took an unexpected turn.

Following a tip from an independent security researcher who preferred to remain anonymous, we discovered that the attacker had inadvertently left an endpoint exposed on their server. We leveraged this opening to exfiltrate data from the attacker's infrastructure.

And that's when we found it: a partial list of victims.

We can't share specific names or identifying details - both for victim privacy and because this is now part of an active law enforcement investigation. But we can tell you what we saw:

- Victims spanning the US, South-America, Europe, and Asia
- A government entity from the Middle East
- Dozens of individual developers and organizations

These aren't hypothetical victims. These are real organizations and real people whose credentials have been harvested, whose machines may be serving as criminal proxy infrastructure, whose internal networks may already be compromised.

But the server held something else: the attacker's own keylogger data. Whether from testing infrastructure or operational oversight, we obtained intelligence that significantly advances attribution efforts:

- The attacker is Russian-speaking
- They use RedExt, an open-source browser extension C2 framework, as part of their infrastructure
- We have their user IDs for multiple cryptocurrency exchanges and messaging platforms


```
  3   "data_by_type": {
 91     "history": [
 92       {
 96         "payload": {
 98           "entries": [
295             {
296               "url": "https://www.bybit.com/ru-RU/dashboard",
297               "title": "Авторизация | Bybit",
298               "visitCount": 4,
299               "lastVisit": "2025-09-02T12:16:00.230Z",
300               "typedCount": 0
301             },
302             {
303               "url": "http://localhost:3000/urlwatch",
304               "title": "RedExt C2 Dashboard",
305               "visitCount": 1,
306               "lastVisit": "2025-09-01T14:52:48.556Z",
307               "typedCount": 0
308             },
309             {
```


Attacker's data extracted from the C2 server

This intelligence has been shared with law enforcement agencies and provides investigators with concrete leads for attribution.

We're currently working with law enforcement agencies to notify affected victims and coordinate efforts to take down the attacker's infrastructure. But the reality is sobering: this campaign has been running for over a month, and it continues to spread.

The victims we found represent only a partial snapshot - what we could extract from one exposed endpoint. The real scale of compromise is likely much larger.

GlassWorm Spreads to GitHub

On October 31, 2025, security researchers at Aikido Security published findings showing that GlassWorm has jumped to GitHub repositories. Developers contacted Aikido after discovering their own repositories had been compromised with seemingly legitimate commits - project-specific code changes that appear to be AI-generated to blend in with normal development activity. Hidden within these commits: the same invisible Unicode malware pattern, using Private Use Area encoding to conceal malicious payloads. The decoded payloads use the same Solana blockchain delivery mechanism we documented, confirming this is GlassWorm. Most significantly, stolen GitHub credentials are being used to push malicious commits to additional repositories - proving the self-propagating worm behavior we warned about in our original analysis.

IOCs

Compromised Extensions

OpenVSX (November 2025 wave):

- adhamu.history-in-sublime-merge@1.3.4
- yasuyuky.transient-emacs@0.23.1
- ai-driven-dev.ai-driven-dev@0.4.11

Final Thoughts

This writeup was authored by the research team at Koi Security, with gratitude to our partners in the security research community and a commitment to a safer open-source ecosystem.

GlassWorm demonstrates why visibility and governance across the entire software supply chain is no longer optional. When malware can be literally invisible, when worms can self-propagate through stolen credentials, when attack infrastructure can't be taken down - traditional security tools aren't enough.

We've built Koi to meet this moment. Our platform helps discover, assess, and govern everything your teams pull from marketplaces like the Chrome Web Store, VSCode, Hugging Face, Homebrew, GitHub, and beyond. Trusted by Fortune 50 organizations, BFSIs, and some of the largest tech companies in the world, Koi automates the security processes needed to gain visibility, establish governance, and proactively reduce risk across this sprawling attack surface.

Book a demo to see how Koi closes the gaps that legacy tools miss.

Stay paranoid out there. Because in a world where malware can be invisible and worms can propagate themselves, paranoia isn't a bug - it's a feature.
