Security Advisory for Cargo (CVE-2026-5223)

May 25, 2026 · Rust Security Response Team

The Rust Security Response Team was notified that Cargo incorrectly handled symlinks inside of crate tarballs downloaded from third-party registries, allowing a malicious crate to override the source code of another crate from the same registry.

This vulnerability is tracked as CVE-2026-5223. The severity of the vulnerability is **medium** for users of third-party registries. Users of crates.io are **not affected**, as crates.io forbids uploading crates containing any symlink.

Overview

When building a crate, Cargo extracts its source code in a local cache (stored within `~/.cargo`), reusing it for any future build. Cargo includes protections to prevent any file from being extracted outside of the crate's own cache directory.

It was discovered that it's possible to craft a malicious tarball able to extract files one level below the crate's own cache directory. With the way the cache is structured, that allowed the malicious crate to override the cache of other crates belonging to the same registry.

Mitigations

Rust 1.96.0, to be released on May 28th, 2026, will update Cargo to reject extracting *any* symlink within crate tarballs, regardless of whether they come from crates.io (which already forbids them) or third-party

registries. Note that Cargo never added symlinks when running `cargo package` or `cargo publish`, so the impact of this should be minimal.

Users who are not able to upgrade to the most recent Rust version are recommended to audit the contents of their registry for the presence of any symlink, and to configure their registry to reject symlink (if such option is available).

Affected versions

All versions of Cargo shipped before Rust 1.96.0 are affected.

Acknowledgements

We'd like to thank Christos Papakonstantinou for reporting this to us according to the Rust security policy.

We also want to thank the members of the Rust project who helped us address the vulnerability: Josh Triplett for developing the fix; Arlo Siemsen for reviewing the fix; Emily Albini for writing this advisory; Emily Albini, Josh Stone and Manish Goregaokar for coordinating the disclosure; Ed Page and Eric Huss for advising during the disclosure.