RUSTSEC-2020-0141

History · Edit

MvccRwLock allows data races & aliasing violations

Reported	December 10, 2020
Issued	February 1, 2021 (last modified: October 19, 2021)
Package	noise_search (crates.io)
Type	Vulnerability
Categories	memory-corruption thread-safety
Aliases	CVE-2020-36461
Details	https://github.com/pipedown/noise/issues/72
CVSS Score	8.1 HIGH
CVSS Details	Attack vector: Network Attack complexity: High Privileges required: None User interaction: None Scope: Unchanged Confidentiality: High Integrity: High Availability: High
CVSS Vector	CVSS:3.1/AV:N/AC:H/PR:N/UI:N/S:U/C:H/I:H/A:H
Patched	no patched versions

Description

Affected versions of this crate unconditionally implement Send/Sync for `MvccRwLock`. This can lead to data races when types that are either `!Send` or `!Sync` (e.g. `Rc<T>`, `Arc<Cell<_>>`) are contained inside `MvccRwLock` and sent across thread boundaries. The data races can potentially lead to memory corruption (as demonstrated in the PoC from the original report issue).

Also, safe APIs of `MvccRwLock` allow aliasing violations by allowing `&T` and `LockResult<MutexGuard<Box<T>>>` to co-exist in conflicting lifetime regions. The APIs of `MvccRwLock` should either be marked as `unsafe` or `MbccRwLock` should be changed to private or pub(crate).