



Look up package or ID...

[About](#) [Advisories](#) [Report Vulnerabilities](#)



## RUSTSEC-2020-0114

[History](#) · [Edit](#)

**Demuxer** can carry non-Send types across thread boundaries

**Reported** December 22, 2020

**Issued** January 21, 2021 (last modified: October 19, 2021)

**Package** [va-ts](#) ([crates.io](#))

**Type** Vulnerability

**Categories** [memory-corruption](#)  
[thread-safety](#)

**Aliases** [CVE-2020-36220](#)

**Details** <https://github.com/video-audio/va-ts/issues/4>

**CVSS Score** 5.9 MEDIUM

### CVSS Details

<b>Attack vector</b>	Network
<b>Attack complexity</b>	High
<b>Privileges required</b>	None
<b>User interaction</b>	None
<b>Scope</b>	Unchanged
<b>Confidentiality</b>	None
<b>Integrity</b>	None
<b>Availability</b>	High

**CVSS Vector** [CVSS:3.1/AV:N/AC:H/PR:N/UI:N/S:U/C:N/I:N/A:H](#)

**Patched** `>=0.0.4`

### Description

In the affected versions of this crate, `Demuxer<T>` unconditionally implemented `Send` with no trait bounds on `T`.

This allows sending a non-Send type `T` across thread boundaries, which can cause undefined behavior like unlocking a mutex from a thread that didn't lock the mutex, or memory corruption from data race.

The flaw was corrected in commit 0562cbf by adding a `T: Send` bound to the `Send` impl for `Demuxer<T>`.