



Look up package or ID...

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



RUSTSEC-2020-0027

[History](#) · [Edit](#)

traitobject assumes the layout of fat pointers

**Reported** June 1, 2020

**Issued** October 2, 2020 (last modified: October 19, 2021)

**Package** [traitobject](#) ([crates.io](#))

**Type** INFO Unsound

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

**Aliases** [CVE-2020-35881](#)

**Details** <https://github.com/reem/rust-traitobject/issues/7>

**CVSS Score** 9.8 CRITICAL

**CVSS Details**

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

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

**Patched** no patched versions

**Affected Functions** Version

`traitobject::data`

`traitobject::data_mut`

## Description

This crate gets the data pointer from fat pointers assuming that the first element in a fat pointer is the data pointer. This is currently true, but it may change in a future Rust version, leading to memory corruption.

This has been fixed in the master branch of the crate, but it has not been released into crates.io.