RUSTSEC-2021-0012

History · Edit

Reading uninitialized memory can cause UB (`Deserializer::read_vec`)

Reported	January 2, 2021
Issued	January 24, 2021 (last modified: October 19, 2021)
Package	cdr (crates.io)
Type	Vulnerability
Categories	memory-exposure
Aliases	CVE-2021-26305
Details	https://github.com/hrektts/cdr-rs/issues/10
CVSS Score	9.8 CRITICAL
CVSS Details	

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

CVSS Vector	CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H
Patched	`>=0.2.4`

Description

`Deserializer::read_vec()` created an uninitialized buffer and passes it to a user-provided `Read` implementation (`Deserializer.reader.read_exact()`).

Passing an uninitialized buffer to an arbitrary `Read` implementation is currently defined as undefined behavior in Rust. Official documentation for the `Read` trait explains the following: "It is your responsibility to make sure that buf is initialized before calling read. Calling read with an uninitialized buf (of the kind one obtains via MaybeUninit) is not safe, and can lead to undefined behavior."

The flaw was corrected in commit ce310f7 by zero-initializing the newly allocated buffer before handing it to `Deserializer.reader.read_exact()` .