RUSTSEC-2020-0009

`read_scalar` and `read_scalar_at` allow transmuting values without `unsafe` blocks

Reported	April 11, 2020
Issued	October 2, 2020 (last modified: October 19, 2021)
Package	flatbuffers (crates.io)
Type	Vulnerability
Aliases	CVE-2020-35864
Details	https://github.com/google/flatbuffers/issues/5825
CVSS Score	7.5 HIGH
CVSS Details	

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

CVSS Vector	CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:N/A:H
Patched	>=2.0.0
Unaffected	<0.4.0

Affected Functions	Version
`flatbuffers::read_scalar`	>=0.4.0
`flatbuffers::read_scalar_at`	>=0.4.0

Description

The `read_scalar` and `read_scalar_at` functions are unsound because they allow transmuting values without `unsafe` blocks.

The following example shows how to create a dangling reference:

```
fn main() {
    #[derive(Copy, Clone, PartialEq, Debug)]
    struct S(&'static str);
    impl flatbuffers::EndianScalar for S {
        fn to_little_endian(self) -> Self { self }
        fn from_little_endian(self) -> Self { self }
    }
    println!("{:?}", flatbuffers::read_scalar::<S>(&[1; std::mem::size_of::<S>()]));
}
```