RUSTSEC-2020-0082

ordered_float:NotNan may contain NaN after panic in assignment operators

Reported	December 6, 2020
Issued	December 6, 2020 (last modified: October 19, 2021)
Package	ordered-float (crates.io)
Type	Vulnerability
Keywords	#unwind
Aliases	CVE-2020-35923
Details	https://github.com/reem/rust-ordered-float/pull/71
CVSS Score	5.5 MEDIUM
CVSS Details	
Attack vector	Local
Attack complexity	Low
Privileges required	Low
User interaction	None
Scope	Unchanged
Confidentiality	None
Integrity	None
Availability	High
CVSS Vector	CVSS:3.1/AV:L/AC:L/PR:L/UI:N/S:U/C:N/I:N/A:H
Patched	`^1.1.1` `>=2.0.1`
Unaffected	`<0.2.2`

Description

After using an assignment operators such as `NotNan::add_assign`, `NotNan::mul_assign`, etc., it was possible for the resulting `NotNan` value to contain a `NaN`. This could cause undefined behavior in safe code, because the safe `NotNan::cmp` method contains internal unsafe code that assumes the value is never `NaN`. (It could also cause undefined behavior in third-party unsafe code that makes the same assumption, as well as logic errors in safe code.)

This was mitigated starting in version 0.4.0, by panicking if the assigned value is NaN. However, in affected versions from 0.4.0 onward, code that uses the `NotNan` value during unwinding, or that continues after catching the panic, could still observe the invalid value and trigger undefined behavior.

The flaw is fully corrected in versions 1.1.1 and 2.0.1, by ensuring that the assignment operators panic without modifying the operand, if the result would be `NaN`.