RUSTSEC-2021-0031

split_at allows obtaining multiple mutable references to the same data

Reported	January 31, 2021
Issued	March 2, 2021 (last modified: October 19, 2021)
Package	nano_arena (crates.io)
Type	Vulnerability
Categories	memory-corruption
Keywords	#memory-safety #aliasing #unsound
Aliases	CVE-2021-28032
Details	https://github.com/bennetthardwick/nano-arena/issues/1
CVSS Score	9.8 CRITICAL
CVSS Details	
CVSS Vector	CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H
Patched	>=0.5.2

CVSS Details

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

Affected Functions	Version
`nano_arena::Arena::split_at`	<0.5.2
`nano_arena::ArenaSplit::split_at`	<0.5.2

Description

Affected versions of this crate assumed that `Borrow<Idx>` was guaranteed to return the same value on `.borrow()`. The borrowed index value was used to retrieve a mutable reference to a value.

If the `Borrow<Idx>` implementation returned a different index, the split arena would allow retrieving the index as a mutable reference creating two mutable references to the same element. This violates Rust's aliasing rules and allows for memory safety issues such as writing out of bounds and use-after-frees.

The flaw was corrected in commit `6b83f9d` by storing the `.borrow()` value in a temporary variable.