RUSTSEC-2020-0073

History · Edit

Mutable reference with immutable provenance

Reported	November 12, 2020
Issued	November 20, 2020 (last modified: October 19, 2021)
Package	image (crates.io)
Type	INFO Unsound
Keywords	#pointer #cast #provenance
Aliases	CVE-2020-35916
Details	https://github.com/image-rs/image/issues/1357
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	>=0.23.12

Affected Functions	Version
`image::Bgr::from_slice_mut`	<0.23.12
`image::Bgra::from_slice_mut`	<0.23.12
`image::Luma::from_slice_mut`	<0.23.12
`image::LumaA::from_slice_mut`	<0.23.12
`image::Rgb::from_slice_mut`	<0.23.12
`image::Rgba::from_slice_mut`	<0.23.12

Description

A mutable reference to a struct was constructed by dereferencing a pointer obtained from `slice::as_ptr`. Instead, `slice::as_mut_ptr` should have been called on the mutable slice argument. The former performs an implicit reborrow as an immutable shared reference which does not allow writing through the derived pointer.

There is no evidence for miscompilation, exploitable or otherwise, caused by this bug. Further investigation on Zulip suggests that the unoptimized generated LLVM IR does not contain any UB itself, effectively mitigating further effects.