RUSTSEC-2020-0127

SyncRef's clone() and debug() allow data races

Reported	December 18, 2020
Issued	January 30, 2021 (last modified: October 19, 2021)
Package	v9 (crates.io)
Type	Vulnerability
Categories	memory-corruption thread-safety
Aliases	CVE-2020-36447
Details	https://github.com/purpleposeidon/v9/issues/1
CVSS Score	8.1 HIGH
CVSS Details	

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

CVSS Vector	CVSS:3.1/AV:N/AC:H/PR:N/UI:N/S:U/C:H/I:H/A:H
Patched	no patched versions

Description

Affected versions of this crate unconditionally implement `Sync` for `SyncRef<T>`. This definition allows data races if `&T` is accessible through `&SyncRef`.

`SyncRef<T>` derives `Clone` and `Debug`, and the default implementations of those traits access `&T` by invoking `T::clone()` & `T::fmt()`. It is possible to create data races & undefined behavior by concurrently invoking `SyncRef<T>::clone()` or `SyncRef<T>::fmt()` from multiple threads with `T: !Sync`.