RUSTSEC-2020-0061

futures_task::noop_waker_ref can segfault due to dereferencing a NULL pointer

Reported	May 3, 2020
Issued	October 31, 2020 (last modified: October 19, 2021)
Package	futures-task (crates.io)
Type	Vulnerability
Categories	denial-of-service
Keywords	#NULL-pointer-dereference #memory-management
Aliases	CVE-2020-35907
Details	https://github.com/rust-lang/futures-rs/issues/2091
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	>=0.3.5

Affected Functions	Version
`futures_task::noop_waker_ref`	>=0.3.0

Description

Affected versions of the crate used a `UnsafeCell` in thread-local storage to return a noop waker reference, assuming that the reference would never be returned from another thread.

This resulted in a segmentation fault crash if `Waker::wake_by_ref()` was called on a waker returned from another thread due to it attempting to dereference a pointer that wasn't accessible from the main thread.

Reproduction Example (from issue):

```
use futures_task::noop_waker_ref;
fn main() {
    let waker = std::thread::spawn(|| noop_waker_ref()).join().unwrap();
    waker.wake_by_ref();
}
```

The flaw was corrected by using a `OnceCell::Lazy<>` wrapper around the noop waker instead of thread-local storage.