RUSTSEC-2020-0006

Flaw in `realloc` allows reading unknown memory

Reported	March 24, 2020
Issued	October 2, 2020 (last modified: October 19, 2021)
Package	bumpalo (crates.io)
Type	Vulnerability
Categories	memory-exposure
Aliases	CVE-2020-35861
Details	https://github.com/fitzgen/bumpalo/issues/69
CVSS Score	7.5 HIGH
CVSS Details	
CVSS Vector	CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:N/A:N
Patched	`>=3.2.1`
Unaffected	`<3.0.0`

CVSS Details

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

Description

When `realloc`ing, if we allocate new space, we need to copy the old allocation's bytes into the new space. There are `old_size` number of bytes in the old allocation, but we were accidentally copying `new_size` number of bytes, which could lead to copying bytes into the realloc'd space from past the chunk that we're bump allocating out of, from unknown memory.

If an attacker can cause `realloc`s, and can read the `realoc`ed data back, this could allow them to read things from other regions of memory that they shouldn't be able to. For example, if some crypto keys happened to live in memory right after a chunk we were bump allocating out of, this could allow the attacker to read the crypto keys.

Beyond just fixing the bug and adding a regression test, I've also taken two additional steps:

1. While we were already running the testsuite under `valgrind` in CI, because `valgrind` exits with the same code that the program did, if there are invalid reads/writes that happen not to trigger a segfault, the program can still exit OK and we will be none the wiser. I've enabled the `--error-exitcode=1` flag for `valgrind` in CI so that tests eagerly fail in these scenarios.
2. I've written a quickcheck test to exercise `realloc`. Without the bug fix in this patch, this quickcheck immediately triggers invalid reads when run under `valgrind`. We didn't previously have quickchecks that exercised `realloc` because `realloc` isn't publicly exposed directly, and instead can only be indirectly called. This new quickcheck test exercises `realloc` via `bumpalo::collections::Vec::resize` and `bumpalo::collections::Vec::shrink_to_fit` calls.