hyperium / http Public

Code Issues 76 Pull requests 46 Actions Security Insights

New issue

Jump to bottom

Size issue in `HeaderMap::reserve()` #352

Closed **Qwaz** opened this issue on Nov 16, 2019 · 2 comments · Fixed by #360

Labels A-headers E-easy S-bug

Qwaz commented on Nov 16, 2019

http/src/header/map.rs
Lines 622 to 640 in `9c05e39`

```
622     pub fn reserve(&mut self, additional: usize) {
623         // TODO: This can't overflow if done properly... since the max # of
624         // elements is u16::MAX.
625         let cap = self.entries.len()
626             .checked_add(additional)
627             .expect("reserve overflow");
628 
629         if cap > self.indices.len() {
630             let cap = cap.next_power_of_two();
631 
632             if self.entries.len() == 0 {
633                 self.mask = cap - 1;
```

`usize::next_power_of_two()` method silently overflows to 0 in release mode. This makes it possible to shrink the size of the map to 0 with `HeaderMap::reserve()`.

- If the map doesn't contain any entry, it sets the mask value to `usize::MAX` which is inconsistent but doesn't create any immediate harm.
- If the map contains any entry, the code will call `self.grow(0)` and start infinite probing in this line.

Another problem is that the assertion for `MAX_SIZE` doesn't exist here, so it is possible to grow the map larger than `MAX_SIZE`.

Demonstration

Qwaz mentioned this issue on Nov 16, 2019

Audit **`http`** rust-secure-code/safety-dance#37
Open

hawkw commented on Nov 18, 2019 Contributor

@Qwaz Am I correct that the potential vulnerability this presents is that an attacker could cause a denial of service by sending a request which causes this overflow?

Qwaz commented on Nov 18, 2019 Author

@hawkw An attacker can trigger a DoS if they can call `HeaderMap::reserve()` to non-empty `HeaderMap` with a controlled parameter. I believe this scenario is unlikely in practice, but yes, it's theoretically possible.

seanmonstar added A-headers E-easy S-bug labels on Nov 25, 2019

seanmonstar mentioned this issue on Nov 25, 2019

fix capacity overflows in HeaderMap::reserve #360
Merged

seanmonstar closed this as completed in #360 on Nov 25, 2019

Qwaz added a commit to Qwaz/advisory-db that referenced this issue on Jan 9, 2020

`hyperium/http/issues/352` da7383d

Qwaz added a commit to Qwaz/advisory-db that referenced this issue on Jan 9, 2020

`hyperium/http/issues/352` 46a67a2

Qwaz added a commit to Qwaz/advisory-db that referenced this issue on Jan 9, 2020

`hyperium/http/issues/352` 36b8de6

tarcieri added a commit to rustsec/advisory-db that referenced this issue on Jan 9, 2020

`Merge pull request #217 from Qwaz/http1` … 8c9c29b roy-work added a commit to roy-work/advisory-db that referenced this issue on Jan 9, 2020

`Correct affected version range on RUSTSEC-2019-003[34] to patched at …` ··· 200651c

roy-work mentioned this issue on Jan 9, 2020

Correct affected version range on RUSTSEC-2019-003[34] to patched at 0.1.20 rustsec/advisory-db#221

Merged

dfinity-bot added a commit to dfinity/sdk that referenced this issue on Mar 5, 2020

`## Changelog for advisory-db:` ··· cd01476

dfinity-bot mentioned this issue on Mar 5, 2020

niv advisory-db: update 891a872b -> 19196c29 dfinity/sdk#424

Merged

mergify bot added a commit to dfinity/sdk that referenced this issue on Mar 9, 2020

`## Changelog for advisory-db: (#424)` ··· afbd54f

mend-for-github-com bot mentioned this issue on Sep 16, 2020

CVE-2020-25574 (High) detected in http-0.1.16.crate LevyForchh/octobot#26

Open

Assignees

No one assigned

Labels

A-headers E-easy S-bug

Projects

None yet

Milestone

No milestone

Development

Successfully merging a pull request may close this issue.

fix capacity overflows in HeaderMap::reserve
hyperium/http

3 participants