

New issue

[Jump to bottom](#)

Side effect handling in specialized zip implementation causes buffer overflow #82282

Closed Qwaz opened this issue on Feb 18, 2021 · 2 comments · Fixed by #82289

Labels C-bug I-unsound P-critical T-libs

Qwaz commented on Feb 18, 2021

Contributor

[rust/library/core/src/iter/adapters/zip.rs](#)
Lines 200 to 208 in #148b97

```
200 } else if A::may_have_side_effect() && self.index < self.a.size() {
201     let i = self.index;
202     self.index += 1;
203     // match the base implementation's potential side effects
204     // SAFETY: we just checked that 'i' < 'self.a.len()'
205     unsafe {
206         self.a.__iterator_get_unchecked(i);
207     }
208     None
```

[rust/library/core/src/iter/adapters/zip.rs](#)
Lines 214 to 218 in #148b97

```
214 #[inline]
215 fn size_hint(&self) -> (usize, Option<usize>) {
216     let len = self.len - self.index;
217     (len, Some(len))
218 }
```

`self.index` can be set to a value greater than `self.len` in this branch. This causes integer overflow in `size_hint()` and lead to a buffer overflow.

[Playground Link](#) that demonstrates segfault with safe Rust code.

 Qwaz added the C-bug label on Feb 18, 2021



Qwaz commented on Feb 18, 2021

Contributor Author


For the context, this causes a buffer overflow by violating the safety requirement of `TrustedRandomAccess` trait.

[rust/library/core/src/iter/adapters/zip.rs](#)
Lines 384 to 406 in #148b97

```
384 /// An iterator whose items are random-accessible efficiently
385 ///
386 /// # Safety
387 ///
388 /// The iterator's `size_hint` must be exact and cheap to call.
389 ///
390 /// `size` may not be overridden.
391 ///
392 /// `<Self as Iterator>::__iterator_get_unchecked` must be safe to call
393 /// provided the following conditions are met.
394 ///
395 /// 1. `0 <= idx` and `idx < self.size()`.
```

 GuillaumeGomez added the I-unsound label on Feb 19, 2021 rustbot added the I-prioritize label on Feb 19, 2021 SkiFire13 mentioned this issue on Feb 19, 2021


Fix underflow in specialized ZipImpl::size_hint #82289

Merged jonas-schievink added the T-libs label on Feb 19, 2021

hameerabbasi commented on Feb 19, 2021

Contributor

Assigning P-critical as part of the [WG-prioritization discussion on Zulip](#).

 hameerabbasi added P-critical and removed I-prioritize labels on Feb 19, 2021 m-ou-se added a commit to m-ou-se/rust that referenced this issue on Mar 5, 2021

 bors closed this as completed in ee796c6 on Mar 5, 2021

  the8472 mentioned this issue on Jan 28

Use TrustedRandomAccess for loop desugaring #93243

 Closed

  the8472 mentioned this issue on Feb 22

Stacked borrows fails on {ChunksMut, ChunksExactMut}::__iterator_get_unchecked() #94231

 Closed

Assignees

No one assigned

Labels

C-bug I-unsound P-critical T-libs

Projects


None yet

Milestone

No milestone

Development

Successfully merging a pull request may close this issue.

 Fix underflow in specialized ZipImpl::size_hint
SkiFire13/rust

5 participants

