

GitoxideLabs / **gitoxide** Public[Code](#) [Issues 162](#) [Pull requests 13](#) [Discussions](#) [Actions](#) [Projects](#)

New issue



Non-utf8 String can be created with `TimeBuf::as_str` #2305

Closed

#2306

Labels

acknowledged



Rafael (shinmao) opened on Dec 27, 2025



We consider that the function `as_str` can create a illegal string, which contains non-utf8 characters.

[gitoxide/gix-date/src/parse/mod.rs](#)Lines 27 to 38 in [a000a73](#)

```
27     impl TimeBuf {
28         /// Represent this instance as standard string, serialized in a format compatible
29         /// signature fields in Git commits, also known as anything parseable as [raw for
30     pub fn as_str(&self) -> &str {
31         // SAFETY: We know that serialized times are pure ASCII, a subset of UTF-8.
32         //           `buf` and `len` are written only by time-serialization code.
33         let time_bytes = self.buf.as_slice();
34         #[allow(unsafe_code)]
35         unsafe {
36             std::str::from_utf8_unchecked(time_bytes)
37         }
38     }
```

The reason is that `TimeBuf` implements the trait `Write`, which allows to write arbitrary bytes.

[gitoxide/gix-date/src/parse/mod.rs](#)Lines 46 to 49 in [a000a73](#)

```
46     impl std::io::Write for TimeBuf {
47         fn write(&mut self, buf: &[u8]) -> std::io::Result<usize> {
48             self.buf.write(buf)
49         }
```

Following is a simple PoC to prove this reachable undefined behavior:

```
fn main() {  
    let mut buf = TimeBuf::default();  
    buf.write(&[0xff]).unwrap();  
    println!("{}", buf.as_str()); // print 0  
}
```



Therefore, the safety invariant for `TimeBuf` mentioned at line 31 can be broken here.



Sebastian Thiel (Byron) added **acknowledged** on Dec 28, 2025



Sebastian Thiel (Byron) on Dec 28, 2025

Member ...

Thanks a lot for reporting!

This is a critical issue and I want to get it fixed asap.



Copilot mentioned this on Dec 28, 2025

[fix: validate UTF-8 in TimeBuf::write\(\) to prevent UB in as_str\(\) #2306](#)



Copilot added a commit that references this issue on Dec 28, 2025

fix: prevent non-UTF8 bytes in TimeBuf (issue [#2305](#)) ...

690b4d6



Rafael (shinmao) on Dec 29, 2025

Author ...

Hi **Sebastian Thiel (@Byron)** thanks for the response. Do you agree with me to submit the rustsec advisories with patched version for you?



1



Sebastian Thiel (Byron) on Dec 30, 2025











Member ...

Yes, please go ahead.

PS: CC **Eliah Kagan (@EliahKagan)** in case you haven't seen this, it's just an FYI.



1

-   **Sebastian Thiel (Byron)** added a commit that references this issue [on Dec 31, 2025](#)
fix!: prevent non-UTF8 bytes in `TimeBuf` ([#2305](#)) ... Verified f9051e7
-   **Sebastian Thiel (Byron)** closed this as [completed](#) in [#2306](#) [on Dec 31, 2025](#)
-   **github-actions** mentioned this in 4 issues [on Jan 5](#)
-  [RUSTSEC-2025-0140: Non-utf8 String can be created with TimeBuf::as_str rust-lang/docs.rs#3115](#)
 -  [RUSTSEC-2025-0140: Non-utf8 String can be created with TimeBuf::as_str vbrandl/hoc#976](#)
 -  [RUSTSEC-2025-0140: Non-utf8 String can be created with TimeBuf::as_str tari-project/tari-cli#73](#)
 -  [RUSTSEC-2025-0140: Non-utf8 String can be created with TimeBuf::as_str dfinity/icp-cli#246](#)



Michal Findra (mfindra) on Jan 7



Hello,
Is there a CVE assigned for this issue? If not, we (Red Hat Product Security) can assign one.
Thanks



Sebastian Thiel (Byron) on Jan 7

Member



I am not aware of a CVE - please feel free to manage this. Also just in case: CC [Eliah Kagan \(@EliahKagan\)](#) .



Rafael (shinmao) on Jan 7

Author



[Michal Findra \(@mfindra\)](#) Yes. I haven't applied apply CVE for it. I would appreciate it if there is one.
Thanks.



Rafael (shinmao) on Jan 7

Author



[Sebastian Thiel \(@Byron\)](#) Out of curiosity. Since now the patch has been released, can you briefly explain the impacts of this unsoundness in this crate? Thanks.



Sebastian Thiel (Byron) on Jan 7

Member



Actually, it's hard to imagine real-world implications of this. `gitoxide` code would most certainly be completely unaffected. To me it *seems* this was only an issue in principle. But then again, I'd also prefer to take in-principle as a reason to assume real-world exploits *can* exist.



Michal Findra (mfindra) on Jan 9



[CVE-2026-0810](#) was reserved for this issue.



github-actions mentioned this in 2 issues [on Jan 11](#)

✓ [RUSTSEC-2025-0140: Non-utf8 String can be created with `TimeBuf::as_str` pkgcraft/pkgcraft#346](#)

○ [RUSTSEC-2025-0140: Non-utf8 String can be created with `TimeBuf::as_str` reactive-graph/reactive-graph#582](#)

[Sign up for free](#) to join this conversation on GitHub. Already have an account? [Sign in to comment](#)

Metadata

Assignees

No one assigned

Labels

acknowledged

Type

No type

Projects

No projects

Milestone

No milestone

Relationships

None yet

Development

[fix: validate UTF-8 in TimeBuf::write\(\) to prevent UB in as_str\(\)](#)

GitoxideLabs/gitoxide

 **gix-date-v0.12.0**

Participants

