Security advisory for the standard library

May 13, 2019 · The Rust Core Team

This is a cross-post of the official security advisory. The official post contains a signed version with our PGP key, as well.

The CVE for this vulnerability is CVE-2019-12083.

The Rust team was recently notified of a security vulnerability affecting manual implementations of `Error::type_id` and their interaction with the `Error::downcast` family of functions in the standard library. If your code does not manually implement `Error::type_id` your code is not affected.

Overview

The `Error::type_id` function in the standard library was stabilized in the 1.34.0 release on 2019-04-11. This function allows acquiring the concrete `TypeId` for the underlying error type to downcast back to the original type. This function has a default implementation in the standard library, but it can also be overridden by downstream crates. For example, the following is currently allowed on Rust 1.34.0 and Rust 1.34.1:

```
struct MyType;

impl Error for MyType {
    fn type_id(&self) -> TypeId {
        // Enable safe casting to `String` by accident.
        TypeId::of::<String>()
```

```
    }
}
```

When combined with the `Error::downcast*` family of methods this can enable safe casting of a type to the wrong type, causing security issues such as out of bounds reads/writes/etc.

Prior to the 1.34.0 release this function was not stable and could not be either implemented or called in stable Rust.

Affected Versions

The `Error::type_id` function was first stabilized in Rust 1.34.0, released on 2019-04-11. The Rust 1.34.1 release, published 2019-04-25, is also affected. The `Error::type_id` function has been present, unstable, for all releases of Rust since 1.0.0 meaning code compiled with nightly may have been affected at any time.

Mitigations

Immediate mitigation of this bug requires removing manual implementations of `Error::type_id`, instead inheriting the default implementation which is correct from a safety perspective. It is not the intention to have `Error::type_id` return `TypeId` instances for other types.

For long term mitigation we are going to destabilize this function. This is unfortunately a breaking change for users calling `Error::type_id` and for users overriding `Error::type_id`. For users overriding it's likely memory unsafe, but users calling `Error::type_id` have only been able to do so on stable for a few weeks since the last 1.34.0 release, so it's thought that the impact will not be too great to overcome.

We will be releasing a 1.34.2 point release on 2019-05-14 (tomorrow) which reverts #58048 and destabilizes the `Error::type_id` function. The upcoming 1.35.0 release along with the beta/nightly channels will also all be updated with a destabilization.

The final fate of the `Error::type_id` API isn't decided upon just yet and is the subject of #60784. No action beyond destabilization is currently planned so nightly code may continue to exhibit this issue. We hope to fully resolve this in the standard library soon.

Timeline of events

- Thu, May 9, 2019 at 14:07 PM - Bug reported to security@rust-lang.org
- Thu, May 9, 2019 at 15:10 PM - Alex reponds, confirming the bug
- Fri, May 10, 2019 - Plan for mitigation developed and implemented

- Mon, May 13, 2019 - PRs posted to GitHub for stable/beta/master branches
- Mon, May 13, 2019 - Security list informed of this issue
- (planned) Tue, May 14, 2019 - Rust 1.34.2 is released with a fix for this issue

Acknowledgements

Thanks to Sean McArthur, who found this bug and reported it to us in accordance with our security policy.