ruby2ßhell

A one-byte heap overflow to remote code execution in Ruby's string case conversion. Reachable without authentication in the default configuration of common Ruby web frameworks and self-hosted software.

Severity
Critical

Fixed in
3.3.14, 3.4.13, 4.0.10

Access needed
Network, no authentication

Advisory
GHSA-34m4-m9fh-gqp5
CVE pending

Affected software

Ruby versions

Ruby series	Affected	Fixed in
4.0	4.0.0 to 4.0.9	4.0.10
3.4	3.4.0 to 3.4.12	3.4.13
3.3	3.3.0 to 3.3.13	3.3.14
3.2 and older	All releases back to 2.4.0	EOL

Your deployment

The bug is reachable from the following frameworks in their default configuration. Other frameworks are not exposed by default, but **application code that upcases strings received in a legacy encoding may also be vulnerable.**

Framework	Status on an affected Ruby
Rails	Affected
Sinatra (classic style)	Affected
Other frameworks and plain Rack	Not exposed by default

Self-hosted software

The latest version of the following popular software bundles an affected Ruby.

Software	Bundled Ruby	Status
GitLab 19.6.0	3.3.13	Affected
GitHub Enterprise Server 3.22.5	4.0.9	Affected
Mastodon 4.7.5	4.0.9	Affected
OpenProject 17.9.7	4.0.9	Affected
Discourse 2026.10.1	3.4.7	Affected
Chatwoot 4.20.0	3.4.4	Affected

Updated 19 November 2026 at 09:29 UTC.

Remediation

Your deployment

Upgrade Ruby to a fixed release: 3.3.14, 3.4.13 or 4.0.10.

Self-hosted software

Install the vendor's fixed release. See Self-hosted software for fixed versions of popular products.

Technical details

Root cause

Most of Ruby's single-byte encodings have no capital `ß`. Upcasing it writes two bytes, `SS`, but the loop only bounds-checks one (`to < to_end`).

```
while (*pp < end && to < to_end) {    /* room for one byte */
    code = *(*pp)++;
    if (code == SHARP_s) {
        if (flags & ONIGENC_CASE_UPCASE) {
            *to++ = 'S';              /* first byte */
            code = 'S';
        }
        /* ... */
    }
    /* ... */
    *to++ = code;                     /* second byte, unchecked */
}
```

The output buffer is allocated as the input length plus 20 bytes. Each `ß` adds one byte to the output, so an input with 21 of them overflows the buffer by one byte.

The same loop is in several ISO-8859 and Windows-125x encodings and has been there since Ruby 2.4.0. It is reached by `upcase`, `swapcase`, `downcase(:fold)` and `casecmp?`.

UTF-8 strings take a different code path and are not affected.

Fixed upstream in `ruby/ruby@d582f8f`.

Primitive

A one-byte heap overflow. The byte written is `S` (`0x53`) when upcasing or `s` (`0x73`) when case-folding, and it lands directly after the output buffer. The attacker controls the size of the buffer through the input length.

Copying the result out of the buffer then reads one byte out of bounds.

Proof of concept

We'll be publishing a proof of concept in a week, once people have had time to upgrade.

Resources

Blog

Case studies

Company

Get a demo

Contact

© 2026 Veria Labs, Inc.