t-defence

CVE-2026-6861

GNU Emacs — Off-by-one Heap Buffer Overflow in SVG CSS Handling

#TDefenceBusiness

Malware Lab

Summary

Our Malware Lab

T-Defence Malware Lab daily performs dissection of malware with the aim of timely understanding the technological evolutions of attacks, consolidating the knowledge of necessary to make more effective and faster the process of incidents responding, contributing to spreading information about emerging threats into the expert's community and among its clients.

Malware Lab analysts are continuously engaged in searching and experimenting new analysis tools, for increasing accuracy and scope of action with regard to the proliferation of new evasion and anti-analysis techniques adopted by malware.

The Malware Lab is also committed to the development of proprietary tools for malware analysis and supporting the management and response of incidents.

Besides malware analysis, Malware Lab ideated and implemented an automatic process of extraction of **Indicators of Compromise (IOC)** that is daily run on dozens of new malwares, intercepted in the wide for populating our Knowledge Base.

Executive Summary

A single flaw in the SVG CSS handling code of src/image.c in GNU Emacs produces two observable memory-safety violations that share the same root cause and are resolved by the same one-line patch:

1. **Off-by-one heap buffer overflow** — a null terminator is written one byte past an `xmalloc`-allocated buffer (poison null byte primitive).
2. **Adjacent use of uninitialized heap memory** — `strncpy()` does not null-terminate the destination when the source length equals the bound, and the off-by-one terminator falls outside the allocation; a subsequent `strlen()` call reads the byte left uninitialized inside the buffer and propagates an incorrect length to `rsvg_handle_set_stylesheet()`.

Per MITRE Counting Rules (INC3 — not independently fixable), a single CVE was assigned covering both manifestations.

Autori:

- Gaetano Zappulla: CISO

Affected component

- **Product:** GNU Emacs
- **File / function:** `src/image.c, svg_load_image()` (around lines 12053–12055 on branch `emacs-30`)
- **Introduced in:** Emacs 28.1 (when CSS support for SVG rendering was added)
- **Affected versions:** 28.1, 28.2, 29.1, 29.2, 29.3, 29.4, 30.1, 30.2
- **Master /** `emacs-31` **branch:** not affected (the vulnerable code path exists only on the release branch)

Root cause

Original vulnerable code:

```
css = xmalloc (SBYTES (lcss) + 1);
strncpy (css, SSDATA (lcss), SBYTES (lcss));
*(css + SBYTES (lcss) + 1) = 0;   /* writes one byte past allocation */
```

The allocation holds `SBYTES(lcss) + 1` bytes; valid indices are `0 .. SBYTES(lcss)`. The original code writes the null terminator at index `SBYTES(lcss) + 1`, one byte past the end of the buffer.

Separately, because `strncpy()` does not null-terminate when the source length equals the bound, and the (incorrect) terminator landed outside the buffer, the byte at index `SBYTES(lcss)` is left uninitialized inside the allocation. This byte is subsequently read by a `strlen()` call further down in the same function, and the resulting length is passed to `rsvg_handle_set_stylesheet()`.

The two issues share root cause and fix, but at runtime they manifest as two distinct sanitizer reports against the same input.

Trigger

Evaluation of Emacs Lisp invoking `create-image` with a `:css` keyword argument, or any code path that causes such an invocation (for example `eval-buffer` on attacker-influenced Lisp input, or rendering paths that auto-invoke `create-image` with CSS processing on user-controlled buffers).

The trigger is ***not*** simply opening a `.svg` file in Emacs. It requires the SVG-with-CSS rendering path to be reached with attacker-influenced CSS content. Standard `find-file` on a `.svg` is insufficient unless a configured hook or auto-render rule funnels the buffer through `create-image :css`.

Proof of vulnerability

Confirmed via AddressSanitizer on macOS arm64, 2026-04-17. Minimal reproduction:

3. Build Emacs (any version in the affected range) with `--with-rsvg` and ASan instrumentation.
4. Construct an Elisp wrapper that calls `create-image` with a `:css` argument whose byte length equals `SBYTES(lcss)` with no internal null terminator, so that `strncpy` writes exactly `SBYTES(lcss)` bytes and leaves the buffer un-terminated.
5. Trigger image rendering on that buffer.

ASan reports (extract, paths and pointers redacted):

```
==[PID]==ERROR: AddressSanitizer: heap-buffer-overflow on address 0x...
WRITE of size 1 at 0x... thread T0
    #0 svg_load_image src/image.c:12055
    #1 lookup_image src/image.c:...
    [...]
0x... is located 0 bytes after N-byte region [0x..., 0x...)
allocated by thread T0 here:
    #0 xmalloc src/alloc.c:...
    #1 svg_load_image src/image.c:12053

==[PID]==ERROR: AddressSanitizer: use-of-uninitialized-value
READ of size 1 at 0x... thread T0
    #0 strlen
    #1 svg_load_image src/image.c:...
```

A complete weaponizable proof-of-concept is intentionally not provided. The information above, combined with the upstream commit diff, is sufficient for any qualified reader to reproduce and verify the issue independently.

Impact

- **Availability — High:** the off-by-one write corrupts allocator metadata immediately adjacent to the `css` allocation. In practice this manifests as an Emacs crash through allocator abort or SIGSEGV on the next allocation or free cycle.
- **Integrity — Low:** the uninitialized byte read propagates an incorrect length to `rsvg_handle_set_stylesheet`, which may cause downstream parsing on out-of-range input. No path to controlled writes of attacker data was identified by the reporter.
- **Confidentiality — None (per CNA scoring):** the uninitialized data is consumed internally as a length parameter and is not exposed back to the attacker through any observable channel.

Arbitrary code execution has not been demonstrated. Off-by-one null-terminator primitives of this shape have historically been leveraged against certain glibc allocator versions for heap-metadata attacks; whether this is achievable against modern allocators on the target platforms is left as an open question.

Fix

Fixed upstream by Eli Zaretskii on the emacs-30 maintenance branch, commit [8f535370b9efbc91673b20c6987a5cae4f6dc562](), 2026-04-18:

```
-     *(css + SBYTES (lcss) + 1) = 0;
+     *(css + SBYTES (lcss)) = 0;
```

The fix removes the off-by-one offset. This single change resolves both manifestations simultaneously: the terminator now lands at index `SBYTES(lcss)` inside the allocation, fixing the out-of-bounds write and leaving no uninitialized byte for the subsequent `strlen()` to read.

The upstream commit message references only the off-by-one issue; the dual-manifestation analysis is documented in this advisory.

Workaround (pre-patch)

For systems that cannot upgrade immediately:

- Do not evaluate untrusted Emacs Lisp that may invoke `create-image` with a `:css` argument.
- Avoid auto-rendering SVG content from untrusted sources where the rendering path includes CSS processing (the default for any Emacs ≥28.1 built with librsvg support).
- Where SVG rendering is not required, build or configure Emacs without rsvg support, or remove `svg` from the relevant image-type variables.

Disclosure timeline

Date (UTC)	Event
2026-04-17	AddressSanitizer confirmation on macOS arm64.
2026-04-18	Fix committed upstream by Eli Zaretskii on the emacs-30 branch.
2026-04-22	CVE-2026-6861 published by Red Hat (CNA).
2026-05-20	This public advisory released.

Discovery and report dates above to be finalized by the author before publication.

References

- CVE record: https://www.cve.org/CVERecord?id=CVE-2026-6861
- NVD: https://nvd.nist.gov/vuln/detail/CVE-2026-6861
- Red Hat advisory: https://access.redhat.com/security/cve/CVE-2026-6861
- Red Hat Bugzilla #2459992: https://bugzilla.redhat.com/show_bug.cgi?id=2459992
- Fix commit (Savannah cgit): https://git.savannah.gnu.org/cgit/emacs.git/commit/?h=emacs-30&id=8f535370b9efbc91673b20c6987a5cae4f6dc562

Credits

Vulnerability discovered, analyzed and reported by **Gaetano Zappulla**, CISO at Defence Tech SpA.

Upstream fix by **Eli Zaretskii** (GNU Emacs maintainer). CVE coordination by **Red Hat Product Security** (CNA).

t-defence

Next | Donexit | Foramil | Innodesi

Via Giacomo Peroni, 452 – 00131 Roma
tel. 06.45752720 – info@defencetech.it
www.tinextadefence.it

#TDefenceBusiness