ESPILON
SECURITY RESEARCH

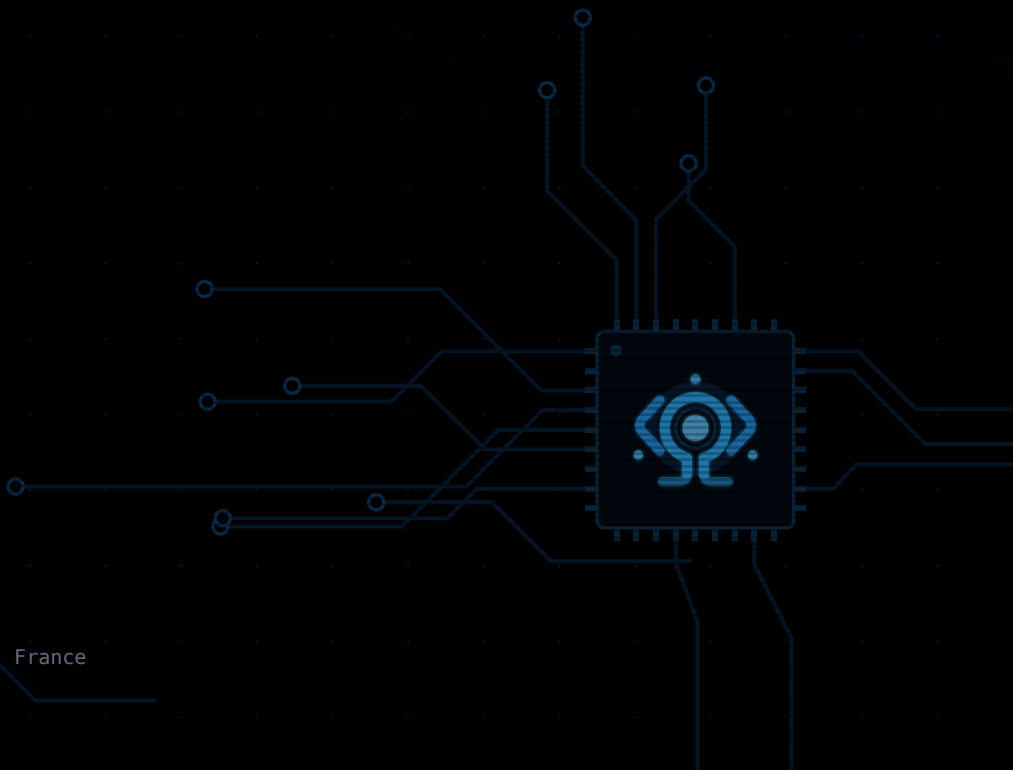
SECURITY ADVISORY

Out-of-bounds write in the LibOSDP control-panel reply parser

CVE-2026-77273 CVSS 8.1 · HIGH

LibOSDP · libosdp

ADVISORY PUBLISHED **30 Aug 2026**



Marouane El Moufid (Eun0us)

espilon.net · Association loi 1901 · France

01 Vulnerability

8.1

HIGH

CVSS 3.1 BASE SCORE

AV:A / AC:L / PR:N / UI:N / S:U / C:N / I:H / A:H

CVE	CVE-2026-77273
Component	LibOSDP, control panel reply parser `cp_decode_response` (`src/osdp_cp.c`)
Affected	LibOSDP >= 1.3.0
Fixed	LibOSDP 3.2.4
Type	Stack out-of-bounds write (CWE-787)
Advisory	GHSA-vcf2-q3rm-gfpq
Credit	Espilon · Marouane El Moufid (Eun0us)

Issue

OSDP (IEC 60839-11-5) is how an access-control control panel (CP), inside the building, talks to the card readers and keypads (peripheral devices, PD) at the door, over a two-wire RS-485 bus. The reader outside is the untrusted end; the panel inside is what unlocks the door. The control panel parses every PD reply in `cp_decode_response()`. Two reply handlers, `REPLY_KEYPAD` and `REPLY_RAW`, copy a length taken off the wire into a fixed 64-byte field of a 160-byte stack event object without clamping it to that field. A single crafted reply frame, with attacker-controlled length and content, overruns the panel's stack. On a build without stack protection, the common case for this hardware, we took that overflow end to end to control of the program counter (x86-64 host build with embedded-like flags; demonstrated after the advisory was published).

Timeline

DATE	EVENT
2026-07-10	LibOSDP v3.2.4 released with the fix
2026-08-30	Advisory GHSA-vcf2-q3rm-gfpq published

Mitigation

Update to LibOSDP 3.2.4, which adds the missing upper clamp at both write sinks. Beyond the patch, enforce Secure Channel (`OSDP_FLAG_ENFORCE_SECURE`): it does not fix the parser, but it removes the pre-authentication reachability that makes these frames trivial to inject on the bus.

02 Technical analysis

`cp_decode_response()` decodes each PD reply into a fixed-size `struct osdp_event` (160 bytes) on the CP's stack. Two reply types carry a variable-length payload into a fixed 64-byte field and both trust a length from the wire without checking it against that field. The sibling handler for `REPLY_MFGREP` does the clamp correctly, which is the tell:

• src/osdp_cp.c, cp_decode_response() in LibOSDP 3.2.3 (vulnerable, abridged)

```
case REPLY_KEYPAD:
    event.keypress.length = buf[pos++];          // length byte off the wire, 0..255
    if ((len - REPLY_KEYPAD_DATA_LEN) != event.keypress.length) break;  // consistency only
    memcpy(event.keypress.data, buf + pos, event.keypress.length);
    // ^ no clamp to OSDP_EVENT_KEYPRESS_MAX_DATALEN (the 64-byte field)
case REPLY_RAW:
    event.cardread.length = bread_u16_le(buf, &pos);   // 16-bit length off the wire
    t = (event.cardread.length + 7) / 8;
    if (t != (len - REPLY_RAW_DATA_LEN)) break;        // consistency only
    memcpy(event.cardread.data, buf + pos, t);         // no clamp to the 64-byte field
case REPLY_MFGREP:                                     // the sibling, done right
    event.mfgrep.length = len - REPLY_MFGREP_LEN;
    if (event.mfgrep.length > OSDP_EVENT_MFGREP_MAX_DATALEN) break;   // clamps first
    memcpy(event.mfgrep.data, buf + pos, event.mfgrep.length);
```

A frame declaring more than 64 payload bytes passes the consistency check and drives a `memcpy` of well over 100 bytes into the 64-byte field, past the end of the 160-byte event object and into adjacent stack, with fully attacker-controlled content, null bytes included.

From missing clamp to a hijacked control panel

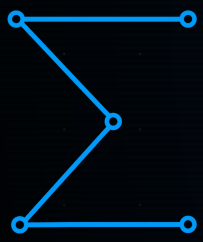
Against our own x86-64 build of v3.2.3, compiled without stack protector or PIE to match a typical embedded CP, the overflow was taken end to end through the genuine receive path: a real panel stood up with `osdp_cp_setup` / `osdp_cp_refresh`, driven by a fake PD on the channel's `recv` callback, with nothing hand-called. A single legal-looking KEYPAD frame, valid CRC and sequence, redirected the panel's program counter to a chosen address. The saved return address sits a fixed distance past the sink, within the single-byte length ceiling, on every ABI we measured (the AArch64 and ARMv7 offsets were read from the generated frames, not run). GHSA-vcf2-q3rm-gfpq predates this work and states that program-counter control was not demonstrated at the time.

BUILD POSTURE	RESULT
x86-64 build: no canary, no PIE, no ASLR	RCE-FEASIBLE: clean controlled transfer to a chosen address, proven
host, -fstack-protector-strong	DoS-only: canary sits before the saved RA, aborts before ret
host, -D_FORTIFY_SOURCE=2	DoS-only, earlier: `__memcpy_chk` knows the 64-byte dest
ASLR, no leak	NEEDS-LEAK; not required on no-ASLR embedded flash

On a hardened glibc host this is a denial of service. On the realistic OSDP controller, a no-MMU microcontroller with no canary, no FORTIFY and fixed flash addresses, the same primitive is expected to give a control-flow hijack of the device that unlocks the door; that is the common build for this class of hardware, but it was not run on such a device.

Reachability

`cp_decode_response()` runs on every reply during the normal poll cycle, before Secure Channel when it is not enforced (`OSDP_FLAG_ENFORCE_SECURE` unset, the default). The attacker is on the RS-485 bus: a rogue or cloned PD, a tapped reader wire, or an in-line serial-to-IP bridge injecting a CRC-valid frame. No pairing, key or prior handshake is needed.



ESPILON

SECURITY RESEARCH

Embedded, IoT and OT security research

espilon.net

eun0us@espilon.net

Association loi 1901 · France