



## Bug 2459181 (CVE-2026-6507) - CVE-2026-6507 dnsmasq: dnsmasq: Denial of Service due to out-of-bounds write in DHCP BOOTREPLY processing

**Keywords:**

**Reported:** 2026-04-17 11:39 UTC by OSIDB Bzimport

**Status:** NEW

**Modified:** 2026-04-20 09:22 UTC ([History](#))

**Alias:** CVE-2026-6507

**CC List:** 1 user ([show](#))

**Product:** Security Response

**Fixed In Version:**

**Component:** vulnerability

**Clone Of:**

**Environment:**

**Version:** unspecified

**Last Closed:**

**Embargoed:**

**Hardware:** All

**OS:** Linux

**Priority:** medium

**Severity:** medium

**Target Milestone:** ---

**Assignee:** Product Security DevOps Team

**QA Contact:**

**Docs Contact:**

**URL:**

**Whiteboard:**

**Depends On:** [2459196](#) [2459195](#)

**Blocks:**

**TreeView+** [depends on](#) / [blocked](#)

**Attachments** ([Terms of Use](#))

OSIDB Bzimport 2026-04-17 11:39:54 UTC

[Description](#)

On dnsmasq 2.92, a server-facing BOOTREPLY processed under `--dhcp-split-relay` can place `OPTION_AGENT_ID` at the end of a 552-byte packet and make dnsmasq zero one byte past the receive buffer. In my PoC, a benign 552-byte BOOTREPLY leaves the daemon alive, while the malicious variant followed by one oversized BOOTREPLY aborts the normal build with `malloc(): invalid next size (unsorted)`. The attached PoC also reproduces the direct AddressSanitizer report at `src/rfc2131.c:3251`.

Details

The bug is in `src/rfc2131.c:3249-3251`. After finding `OPTION_AGENT_ID`, dnsmasq sets `*opt = OPTION_END` and then executes `memset(opt + 1, 0, option_len(opt) + 2)`. The attached

PoC uses the smallest RFC 3046-conformant Agent Information option: OPTION\_AGENT\_ID, length 2, followed by one zero-length sub-option (01 00). When that option starts at byte 548 of a 552-byte BOOTREPLY, memset(opt + 1, 0, 4) clears bytes 549..552, so the last byte is still written one byte past the end of the packet buffer. The buffer is exact-size because recv\_dhcp\_packet() grows the receive iovec to the packet length in src/dhcp-common.c:54-64, and expand\_buf() reallocates that exact size in src/util.c:703-716.

---

**Note**

You need to [log in](#) before you can comment on or make changes to this bug.

