sgInnora / **advisory_agl_v2.md**

Created yesterday

Code Revisions 1

Automotive Grade Linux (AGL) Multiple Vulnerabilities (CVE-2026-37525/37526/37530/37531/37532/42485)

advisory_agl_v2.md

Automotive Grade Linux (AGL) — Multiple Vulnerabilities

Vendor: Automotive Grade Linux (AGL), Linux Foundation **Affected**: agl-service-can-low-level ≤ 17.1.12, app-framework-binder (afb-daemon) ≤ v19.90.0, app-framework-main ≤ 17.1.12 **Reporter**: Feng Ning, Innora Security Research (feng@innora.ai) **Disclosure**: 2026-04-30

CVE	Component	Type	CWE	CVSS
CVE-2026-37525	afb-daemon — supervision socket	Privilege Escalation	CWE-269	7.8
CVE-2026-37526	afb-daemon — privileged supervisord API	Improper Access Control	CWE-862	7.8
CVE-2026-37530	agl-service-can-low-level (uds-c)	Stack Buffer Overflow	CWE-121	9.8
CVE-2026-37531	app-framework-main — widget installer	Zip Slip Path Traversal + RCE	CWE-22	9.8
CVE-2026-37532	agl-service-can-low-level (isotp-c)	Heap Buffer Over-Read	CWE-126	7.5
CVE-2026-42485	agl-service-can-low-level (uds-c)	Stack Buffer Overflow	CWE-121	9.8

CVE-2026-37525 / CVE-2026-37526 — afb-daemon Privilege Escalation

`afb-daemon` exposes a UNIX supervision socket with no peer UID check. Any local process can connect and invoke privileged supervisord API calls directly — there's nothing gating access based on who's on the other end. Sending crafted supervision messages from a low-privilege process is enough to escalate to afb-daemon privileges.

CVE-2026-37530 / CVE-2026-42485 — uds-c Stack Buffer Overflow

In `libs/uds-c/src/uds/uds.c`, `send_diagnostic_request()` copies UDS request payload data into a fixed-size stack buffer. The copy length comes straight from the request structure — no validation against buffer capacity. Feed it a large payload and you get a stack overflow.

CVE-2026-37530 is the `agl-service-can-low-level` packaging angle, reachable over the network via the CAN service API. CVE-2026-42485 tracks the direct library exposure.

CVE-2026-37531 — Widget Installer Zip Slip → RCE

`.wgt` packages are ZIP archives. The AGL widget installer in `app-framework-main` extracts them without sanitizing entry paths, so a crafted widget stuffed with `../` sequences writes files wherever it wants outside the installation directory. Given that installs run in a privileged context, this lands you arbitrary file write and code execution.

Fix: normalize and validate all archive entry paths against the target directory before extraction.

CVE-2026-37532 — isotp-c Heap Buffer Over-Read

ISO-TP multi-frame reassembly in the bundled `isotp-c` library reads past the end of the receive buffer when consecutive frames carry more data than the buffer's declared capacity. The issue is reachable over the CAN bus or through the AGL CAN service API.

Innora Security Research — https://innora.ai