INTELLIGENCE BRIEFING
Security Command Center
TLP:CLEAR
2026-06-17 08:06 UTC

CVE-2026-54412: LiamBindle MQTT-C through version 1.1.6 contains a heap-based out-of-bounds read and integer underfl...

CVE VULNERABILITY | HIGH | CVSS 8.2

SCC Item ID	SCC-CVE-2026-0319
Type	CVE Vulnerability
CVE ID	CVE-2026-54412
Severity	HIGH
CVSS Base Score	8.2
EPSS Score	0.0041 (32th percentile)
Affected Products	LiamBindle MQTT-C through version 1.1.6
Published	2026-06-14T18:17:20.750
Discovery Source	Nvd

Executive Summary

CVE-2026-54412 is a heap-based out-of-bounds read and integer underflow vulnerability in MQTT-C, an open-source C library used to implement MQTT messaging clients in embedded and IoT systems. A remote attacker controlling or impersonating an MQTT broker can crash any client built on MQTT-C versions through 1.1.6 and potentially read adjacent heap memory, which may contain sensitive application data. Organizations running IoT devices, industrial control systems, or embedded applications that use this library should prioritize identification and patching of affected deployments.

Technical Analysis

CVE-2026-54412 affects LiamBindle MQTT-C through version 1.1.6. The vulnerability resides in mqtt_unpack_publish_response() in src/mqtt.c. Two weaknesses are present: CWE-125 (out-of-bounds read) and CWE-191 (integer underflow). The function validates only that the fixed-header remaining_length is >= 4, then reads a 16-bit topic_name_size field without verifying it fits within remaining_length. When topic_name_size=0xFFFF and remaining_length=7, the parse pointer advances 65,535 bytes past the receive buffer. A subsequent unsigned arithmetic subtraction computes application_message_size near 2^32, which is passed to memmove(), crashing the process. Attack vector requires either control of the MQTT broker or the ability to inject traffic into an unencrypted MQTT session (T1040, network sniffing to position for injection;

T1499.004, application-layer denial of service). Authentication is not required. CVSS base score is 8.2 (High). EPSS score is 0.00407 (32nd percentile), indicating limited exploit activity at this time. No CISA KEV listing. Vendor CVSS vector publication is pending (cvss_vendor=0.0); NVD vector not yet available. Patch status: no patch version is cited in the provided source material; verify with the upstream repository for remediation status.

Action Checklist

1. Step 1: Containment, Audit all software bills of materials, firmware images, and embedded application dependencies for MQTT-C versions 1.1.6 and earlier. Immediately isolate any internet-exposed systems where MQTT-C clients connect to untrusted or externally reachable brokers. Where possible, restrict MQTT broker connectivity to known-good, authenticated, TLS-encrypted endpoints to eliminate the unauthenticated injection path. (NIST AC-4, Information Flow Enforcement; CIS 1.1, Establish and Maintain Detailed Enterprise Asset Inventory)
2. Step 2: Detection, Query your software inventory and SBOM tooling for 'mqtt-c' or 'liambindle/mqtt-c' at versions <= 1.1.6. Monitor MQTT broker logs and network traffic for PUBLISH packets with anomalously large topic_name_size values (0xFF bytes in the topic length field) or unusually short remaining_length fields (e.g., remaining_length=7 with topic_name_size near 0xFFFF). Alert on unexpected client process crashes or restarts on systems running MQTT-C. (NIST AU-2, Event Logging; NIST AU-6, Audit Record Review, Analysis, and Reporting; CIS 8.2, Collect Audit Logs)
3. Step 3: Eradication, Upgrade MQTT-C to a version that addresses CVE-2026-54412. As of this writing, the provided source data does not specify a patched release version; confirm the current remediated release at the upstream repository (github.com/LiamBindle/MQTT-C). Until a patch is applied, enforce mutual TLS authentication on all MQTT sessions to eliminate the unauthenticated broker impersonation and traffic injection vectors. Disable plaintext MQTT (port 1883) where encrypted MQTT (port 8883) is available. (CIS 7.3, Perform Automated Operating System Patch Management; CIS 7.4, Perform Automated Application Patch Management; NIST SI-2, Flaw Remediation; CIS 7.2, Establish and Maintain a Remediation Process)
4. Step 4: Recovery, After patching, verify the updated MQTT-C library version is deployed across all affected devices and containers. Restart affected MQTT client processes and confirm stable operation. Monitor for anomalous crash events or heap-related errors post-deployment. Review broker logs for any evidence of crafted PUBLISH packets sent prior to patching, and assess whether adjacent heap memory contents could have been exfiltrated. (NIST AU-6, Audit Record Review, Analysis, and Reporting; NIST AU-11, Audit Record Retention)
5. Step 5: Post-Incident, Review your SBOM and third-party library tracking processes; this vulnerability class (unvalidated field length in protocol parsing) is common in embedded C libraries. Implement or improve automated SBOM generation for firmware and embedded applications. Enforce TLS and broker authentication as baseline standards for all MQTT deployments. Add MQTT-C and similar embedded messaging libraries to your vulnerability management scanning scope. (CIS 2.1, Establish and Maintain a Software Inventory; CIS 7.1, Establish and Maintain a Vulnerability Management Process; NIST AC-17, Remote Access)

Detection Guidance

Primary detection path: query software inventory, SBOM, and package management tooling for 'liambindle/mqtt-c' or 'mqtt-c' at versions <= 1.1.6 across all embedded firmware, containerized applications, and software packages. On MQTT brokers, capture and inspect PUBLISH packet headers; flag packets where the topic_name_size field (bytes 5-6 of a PUBLISH fixed header) encodes a value near 0xFFFF while remaining_length is 7 or less, this is the specific craft pattern described in CVE-2026-54412. Monitor host-level logs on systems running MQTT-C clients for unexpected process terminations, segmentation faults, or core dumps originating from mqtt.c or functions calling mqtt_unpack_publish_response(). Network-level detection: flag MQTT sessions on port 1883 (plaintext) as a risk indicator, since the injection vector requires unencrypted traffic (aligned with T1040). Defense countermeasures: enforce broker authentication and TLS encryption to eliminate unauthenticated attacker positioning, restrict which processes and accounts can initiate MQTT broker connections via host-level access controls, and monitor MQTT-C binary and configuration for tampering post-deployment.

Framework Mappings

MITRE-ATTACK

- **T1040** — Network Sniffing
- **T1499.004** — Application or System Exploitation

NIST-800-53R5

- **SI-16** — Memory Protection
- **SC-13** — Cryptographic Protection

CIS-V8

- **16.10** — Apply Secure Design Principles in Application Architectures

ISO-27001-2022

- **A.8.8** — Management of technical vulnerabilities
- **A.8.24** — Use of cryptography

HIPAA-SECURITY

- **164.312(e)(1)** — Transmission Security

MITRE ATT&CK Mapping

Technique ID	Technique Name	Tactic
T1040	Network Sniffing	Credential-Access
T1499.004	Application or System Exploitation	Impact

Sources

Source	URL	Tier
nvd	https://nvd.nist.gov/vuln/detail/CVE-2026-54412	T1
CVE-2026-54412 - Vulnerability Details - OpenCVE	https://app.opencve.io/cve/CVE-2026-54412	T3
CVE-2026-54412: MQTT-C Heap-Based Out-of-Bounds Read and ...	https://threat-modeling.com/cve-2026-54412-mqtt-c-heap-oob-read/	T3

Generated 2026-06-17 08:06 UTC by TJS Security Command Center