



Home > Submit > 796352

Submit #796352: musl libc musl 0.8.0 - 1.2.6 Inefficient Algorithmic Complexity

Title musl libc musl 0.8.0 - 1.2.6 Inefficient Algorithmic Complexity

Description An algorithmic complexity DoS vulnerability has been discovered in musl 1.2.6.

The GB18030 4-byte decoder in musl libc's iconv() implementation contains a gap-skipping loop that performs a full linear scan of the gb18030126 lookup table (23,940 entries) on each iteration of an outer loop whose iteration count is input-dependent. For 4-byte sequences whose linear index falls just below the dense CJK Unified Ideographs range, the outer loop executes approximately 20,905 times, resulting in approximately 500 million comparisons per input character.

This allows a remote attacker to cause denial of service via CPU exhaustion by sending a crafted GB18030 payload to any network service that uses musl's iconv() for character encoding conversion. Measured on musl 1.2.6 and 1.2.5: a single 4-byte input character (bytes 0x82 0x35 0x8F 0x33) takes approximately 260ms to decode, compared to approximately 13 microseconds for a benign character — a 19,000x slowdown. A payload of 40kB will take the server ~43 minutes process/decode.

Source <https://www.openwall.com/lists/oss-security/2026/04/03/2>

User 0x001 (UID 97056)

Submission 04/03/2026 01:15 PM (7 days ago)

Moderation 04/09/2026 07:34 PM (6 days later)

Status Accepted

VulDB entry VUL-2026-0001 [musl libc up to 1.2.6 GB18030 4-byte Decoder src/locate/iconv.c iconv algorithmic complexity]

Points 20

Community Content

Submissions are made by [VulDB community users](#). VulDB is *not responsible* for their content nor the links to external sources.

Please use the raw information shown and the links listed *with caution*. They might contain malicious and harmful actions, code or data.

The corresponding VulDB entries contain the moderated, verified, and normalized information provided within the raw submission.

Documentation

- [Submission Policy](#)
- [Data Processing](#)
- [CVE Handling](#)