COMMUNITY CONTENT

Home › Submit › 780561

Submit #780561: nothings stb (stb_vorbis.c) ≤ 1.22 Out-of-bounds Write, Integer Overflow

Title nothings stb (stb_vorbis.c) ≤ 1.22 Out-of-bounds Write, Integer Overflow

Description

A heap buffer overflow (out-of-bounds write) vulnerability exists in `start_decoder()` in stb_vorbis.c v1.22 and earlier, caused by an integer overflow in the comment list allocation.

The `comment_list_length` field is read from untrusted Vorbis comment header data via `get32_packet()` at line 3660. At line 3664, the allocation size is computed as `sizeof(char*) * comment_list_length`. When `comment_list_length` is set to a value such as `0x20000002`, the multiplication `8 * 0x20000002 = 0x100000010` overflows the `int sz` parameter of `setup_malloc()`, truncating to 16 bytes. The subsequent loop at lines 3668-3670 then writes `comment_list_length` (536,870,914) pointer-sized entries into a 2-entry (16-byte) buffer, resulting in massive sequential heap corruption.

Each out-of-bounds write stores a heap pointer returned by `setup_malloc(f, len+1)`, where `len` is also attacker-controlled from the packet. The allocated buffers pointed to by these pointers contain fully attacker-controlled data (comment string bytes from the Vorbis packet). This creates a heap corruption primitive where:

1. **Written values**: heap pointers (partially influenced by attacker-controlled allocation sizes)
2. **Pointed-to data**: fully attacker-controlled comment string bytes
3. **Write pattern**: sequential 8-byte writes at stride 8, corrupting adjacent heap chunks

Exploitation potential:
- **Tcache poisoning**: OOB writes corrupt adjacent free chunk `fd` pointers, enabling arbitrary address return from subsequent `malloc()` calls
- **Chunk overlap**: corrupting adjacent chunk size fields causes `free()` to misplace chunks, overlapping with live data
- **Data pointer corruption**: overwriting internal decoder buffer pointers (codebooks, floor configs) redirects later writes to attacker-influenced locations

This vulnerability is triggered via `stb_vorbis_decode_memory()` or `stb_vorbis_open_memory()` when decoding untrusted Ogg Vorbis audio. As a heap-based out-of-bounds write with attacker-controlled data in the overwriting buffers, it may lead to arbitrary code execution, denial of service, or heap corruption.

ASAN output:

```
ERROR: AddressSanitizer: heap-buffer-overflow on address 0x602000000040
WRITE of size 8 at 0x602000000040
```

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

```
#0 start_decoder stb_vorbis.c:3670
#1 stb_vorbis_open_memory stb_vorbis.c:5112
#2 stb_vorbis_decode_memory stb_vorbis.c:5390
0x6020000000040 is located 0 bytes to the right of 16-byte region
allocated by: setup_malloc → stb_vorbis.c:960
```

Source	https://gist.github.com/d0razi/2ff8a0e812f74dd6fe7f2843931bb90c
User	d0razi (UID 96474)
Submission	03/15/2025 01:17 AM (17 days ago)
Moderation	04/01/2026 02:40 PM (17 days later)
Status	Published
VulDB entry	[Nothings stb up to 1.22 stb_vorbis.c start_decoder out-of-bounds write]
Points	20