Invalid free() call with oneofs and PB_ENABLE_MALLOC

Moderate **PetteriAimonen** published **GHSA-7mv5-5mxh-qg88** on Mar 20, 2021

Package

nanopb

Affected versions

0.3.2 to 0.3.9.7, 0.4.0 to 0.4.4

Patched versions

0.3.9.8, 0.4.5

Description

Impact

Decoding a specifically formed message can cause invalid `free()` or `realloc()` calls if the message type contains an `oneof` field, and the `oneof` directly contains both a pointer field and a non-pointer field. If the message data first contains the non-pointer field and then the pointer field, the data of the non-pointer field is incorrectly treated as if it was a pointer value. Such message data rarely occurs in normal messages, but it is a concern when untrusted data is parsed.

Patches

Preliminary patch is available on git for 0.4.x and 0.3.x branches. The fix will be released in versions 0.3.9.8 and 0.4.5 once testing has been completed.

Workarounds

Following workarounds are available:

- Set the option `no_unions` for the oneof field. This will generate fields as separate instead of C union, and avoids triggering the problematic code.
- Set the type of all fields inside the oneof to `FT_POINTER`. This ensures that the data contained inside the `union` is always a valid pointer.
- Heap implementations that guard against invalid `free()` provide a partial mitigation. Depending on the message type, the pointer value may be attacker controlled and can be used to bypass heap protections.

References

Bug report: #647

For more information

If you have any questions or comments about this advisory, comment on the bug report linked above.

Severity

Moderate

CVE ID

CVE-2021-21401

Weaknesses

CWE-763