New issue

Jump to bottom

A Segmentation fault in libxsmm_gemm_generator #398

Closed seviezhou opened this issue on Aug 2, 2020 · 3 comments

Assignees

seviezhou commented on Aug 2, 2020

System info

Ubuntu X64, gcc (Ubuntu 5.5.0-12ubuntu1), libxsmm_gemm_generator (latest master ea905d0)

Command line

This input is the project testcases `right_sparse_test_csc.mtx`

./bin/libxsmm_gemm_generator sparse foo.c foo 16 16 16 32 0 32 1 1 1 1 hsw nopf DP ./SEGV-gemm_generator

Output

```
Segmentation fault
```

AddressSanitizer output

```
ASAN:SIGSEGV
=================================================================
==70432==ERROR: AddressSanitizer: SEGV on unknown address 0x608000010000 (pc 0x00000042a33a bp 0x7ffcf3a8c680 sp 0x7ffcf3a8c300 T0)
    #0 0x42a339  (/home/seviezhou/test/libxsmm_gemm_generator+0x42a339)
    #1 0x40912d  (/home/seviezhou/test/libxsmm_gemm_generator+0x40912d)
    #2 0x402920  (/home/seviezhou/test/libxsmm_gemm_generator+0x402920)
    #3 0x7fe3a865383f in __libc_start_main (/lib/x86_64-linux-gnu/libc.so.6+0x2083f)
    #4 0x4035a8  (/home/seviezhou/test/libxsmm_gemm_generator+0x4035a8)

AddressSanitizer can not provide additional info.
SUMMARY: AddressSanitizer: SEGV ??:0 ??
==70432==ABORTING
```

POC

SEGV-gemm_generator.zip

hfp self-assigned this on Aug 3, 2020

hfp commented on Aug 4, 2020 Collaborator

Thank you for reporting this issue!

(Just FYI, we make no promise regarding health of our master [although we continuously test our code even beyond the prominent Travis tests]. We treat our master-branch like a development branch and create releases at reasonable sync-points and with [hopefully] acceptable cadence. Let us know if this is not meeting your expectation so that we can rethink our QA.)

For our own records: are you relying on features in our master revision? It would be interesting to know! The main differentiation point of master vs. v1.16..1 at the moment are Intel Advanced Matrix Extensions.

hfp added a commit that referenced this issue on Aug 7, 2020

`Issue #398, #399, #400, #401, and #402: account for cases where reque…` ... d698491

hfp commented on Aug 7, 2020 Collaborator

You have used our static code generation and there were two cases of errors: (1) the requested kernel-shape did not match the given sparse input-data, and (2) the given input data was plain invalid/malformed (matrix market file header did not match data records). Our static code generation is "legacy functionality" and we do not intent to carry it forward. We completely embrace JIT-code generation.

Regarding the errors: we designed and implemented our code generation to deliver what the user requests ("WYSIWYG" with different perspective) and our API is not meant to perform deep validation and sanitation of user input. If you walk-in with fuzzed data, you can expect an error message at best. Also, LIBXSMM is quiet and intents to deliver error messages only when enabled (LIBXSMM_VERBOSE).

However, we strive to support our users to incorporate LIBXSMM and to deliver a reasonable amount of runtime error handling, which is the reason we fixed the reported issue. We would also like to learn about your application if any of the above statements made you nervous. Thank you for your report and your dedicated contribution!

hfp closed this as completed on Aug 7, 2020

hfp commented on Sep 28, 2021 • edited Collaborator

The associated changes for this issue are supposed to fix CVE-2021-39535 (see #513).

hfp mentioned this issue on Sep 29, 2021

FYI: CVEs #513

Closed

Assignees

hfp

Labels

None yet

Projects

None yet

Milestone

No milestone

Development

No branches or pull requests

2 participants