Closed · Issue created 2 years ago by A Wireshark GitLab Utility

Buildbot crash output: fuzz-2020-11-19-20476.pcap

Problems have been found with the following capture file:

https://www.wireshark.org/download/automated/captures/fuzz-2020-11-19-20476.pcap

stderr:

```
Input file: /home/wireshark/menagerie/menagerie/xrite-i1displaypro-i1profiler.pcap.gz

Build host information:
Linux build6 4.15.0-122-generic #124-Ubuntu SMP Thu Oct 15 13:03:05 UTC 2020 x86_64 x86_64 x86_64 GNU/Linux
Distributor ID: Ubuntu
Description:    Ubuntu 18.04.5 LTS
Release:        18.04
Codename:       bionic

Buildbot information:
BUILDBOT_WORKERNAME=clang-code-analysis
BUILDBOT_BUILDNUMBER=5344
BUILDBOT_BUILDERNAME=Clang Code Analysis
BUILDBOT_URL=https://buildbot.wireshark.org/wireshark-master/
BUILDBOT_REPOSITORY=git@gitlab.com:wireshark/wireshark.git
BUILDBOT_GOT_REVISION=1d7bc367e943464f912a67ad436fabddb1a61a37

Return value:  0

Dissector bug:  0

Valgrind error count:  1



Latest (but not necessarily the problem) commit:
1d7bc367e9 GSM A Common: Dissect polygon points


Command and args: ./tools/valgrind-wireshark.sh -b /home/wireshark/builders/wireshark-master-fuzz/clangcodeanalysis/install
==10764== Memcheck, a memory error detector
==10764== Copyright (C) 2002-2017, and GNU GPL'd, by Julian Seward et al.
==10764== Using Valgrind-3.13.0 and LibVEX; rerun with -h for copyright info
==10764== Command: /home/wireshark/builders/wireshark-master-fuzz/clangcodeanalysis/install.plain/bin/tshark -Vx -nr /fuzz/
==10764==
==10764==
==10764== HEAP SUMMARY:
==10764==     in use at exit: 597,877 bytes in 8,905 blocks
==10764==   total heap usage: 2,023,536 allocs, 2,014,631 frees, 157,450,596 bytes allocated
==10764==
==10764== LEAK SUMMARY:
==10764==    definitely lost: 556,928 bytes in 8,702 blocks
==10764==    indirectly lost: 0 bytes in 0 blocks
==10764==      possibly lost: 0 bytes in 0 blocks
==10764==    still reachable: 40,154 bytes in 172 blocks
==10764==         suppressed: 795 bytes in 31 blocks
==10764== Rerun with --leak-check=full to see details of leaked memory
==10764==
==10764== For counts of detected and suppressed errors, rerun with: -v
==10764== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
Definitely + indirectly (556928 + 0) exceeds max (102400).
```

no debug trace

To upload designs, you'll need to enable LFS and have an admin enable hashed storage. More information

Tasks 0

No tasks are currently assigned. Use tasks to break down this issue into smaller parts.

Linked items 1

Relates to

Buildbot crash output: fuzz-2020-12-01-896.pcap
#17056

Related merge requests 2

epan: Fix a memory leak.
!1102

epan: Fix a memory leak.
!1107

When these merge requests are accepted, this issue will be closed automatically.

Activity

A Wireshark GitLab Utility added cli tshark scoped label 2 years ago

A Wireshark GitLab Utility added crash label 2 years ago

Gerald Combs @geraldcombs · 2 years ago · Owner

--leak-check=full returns:

```
$ valgrind --tool=memcheck --leak-check=full ./run/tshark -nVx -r /tmp/fuzz-2020-11-19-20476.pcap > /dev/null
==25003== Memcheck, a memory error detector
==25003== Copyright (C) 2002-2017, and GNU GPL'd, by Julian Seward et al.
==25003== Using Valgrind-3.13.0 and LibVEX; rerun with -h for copyright info
==25003== Command: ./run/tshark -nVx -r /tmp/fuzz-2020-11-19-20476.pcap
==25003==
==25003==
==25003== HEAP SUMMARY:
==25003==     in use at exit: 598,157 bytes in 8,911 blocks
==25003==   total heap usage: 699,491 allocs, 690,580 frees, 89,248,635 bytes allocated
==25003==
==25003== 556,928 bytes in 8,702 blocks are definitely lost in loss record 155 of 155
==25003==    at 0x4C2FB0F: malloc (in /usr/lib/valgrind/vgpreload_memcheck-amd64-linux.so)
==25003==    by 0xD56BAB8: g_malloc (in /usr/lib/x86_64-linux-gnu/libglib-2.0.so.0.5600.4)
==25003==    by 0x861D773: wmem_alloc (wmem_core.c:35)
==25003==    by 0x86F269B: tvb_memdup (tvbuff.c:905)
==25003==    by 0x86F4D6A: tvb_get_bits_array (tvbuff.c:1990)
==25003==    by 0x86CFB13: _proto_tree_add_bits_ret_val (proto.c:12302)
==25003==    by 0x86CE4F6: proto_tree_add_bits_ret_val (proto.c:12532)
==25003==    by 0x86CE494: proto_tree_add_bits_item (proto.c:12167)
==25003==    by 0x7BA3734: dissect_usb_hid_data (packet-usb-hid.c:4931)
==25003==    by 0x8693237: call_dissector_through_handle (packet.c:720)
==25003==    by 0x868ED8C: call_dissector_work (packet.c:813)
==25003==    by 0x868EB67: dissector_try_uint_new (packet.c:1413)
==25003==
==25003== LEAK SUMMARY:
==25003==    definitely lost: 556,928 bytes in 8,702 blocks
==25003==    indirectly lost: 0 bytes in 0 blocks
==25003==      possibly lost: 0 bytes in 0 blocks
==25003==    still reachable: 41,229 bytes in 209 blocks
```

```
==25003==          suppressed: 0 bytes in 0 blocks
==25003== Reachable blocks (those to which a pointer was found) are not shown.
==25003== To see them, rerun with: --leak-check=full --show-leak-kinds=all
==25003==
==25003== For counts of detected and suppressed errors, rerun with: -v
==25003== ERROR SUMMARY: 1 errors from 1 contexts (suppressed: 0 from 0)
```

Gerald Combs mentioned in merge request !1102 (merged) 2 years ago

AndersBroman closed via merge request !1102 (merged) 2 years ago

Gerald Combs mentioned in commit 5edf715c 2 years ago

Gerald Combs mentioned in merge request !1107 (merged) 2 years ago

Gerald Combs marked #17056 (closed) as a duplicate of this issue 2 years ago

Gerald Combs marked this issue as related to #17056 (closed) 2 years ago