Closed   Issue created 1 year ago by A Wireshark GitLab Utility

Buildbot crash output: fuzz-2021-04-03-1597129.pcap

Problems have been found with the following capture file:

https://www.wireshark.org/download/automated/captures/fuzz-2021-04-03-1597129.pcap

stderr:

```
Input file: /home/wireshark/menagerie/menagerie/13761-capture-search-t0-win8-win12server.pcapng

Build host information:
Linux build1 5.4.0-70-generic #78-Ubuntu SMP Fri Mar 19 13:29:52 UTC 2021 x86_64 x86_64 x86_64 GNU/Linux
Distributor ID: Ubuntu
Description:    Ubuntu 20.04.2 LTS
Release:        20.04
Codename:       focal

Buildbot information:
BUILDBOT_REPOSITORY=git@gitlab.com:wireshark/wireshark.git
BUILDBOT_WORKERNAME=fuzz-test
BUILDBOT_URL=https://buildbot.wireshark.org/wireshark-3.4/
BUILDBOT_BUILDNUMBER=73
BUILDBOT_BUILDERNAME=Fuzz Test
BUILDBOT_GOT_REVISION=4a7ddb6b1a5e413f924758374408d9824e0df1ea

Return value:  0

Dissector bug:  0

Valgrind error count:  0



Git commit
commit 4a7ddb6b1a5e413f924758374408d9824e0df1ea
Author: Guy Harris <gharris@sonic.net>
Date:   Mon Mar 29 00:55:23 2021 +0000

    tvbuff_subset: fix its implementation of string scanning.

    Both subset_find_guint8() and subset_pbrk_guint8() pass the parent
    tvbuff to tvb_find_guint8()/tvb_ws_mempbrk_pattern_guint8(), along with
    the offset in that tvbuff.

    That means that the offset they get back is relative to that tvbuff, so
    it must be adjusted to be relative to the tvbuff *they* were handed.

    For subsets of frame and "real data" tvbuffs, there's a single lump of
    data containing the content of the subset tvbuff, so they go through the
    "fast path" and get the offset correct, bypassing the broken code;
    that's the vast majority of calls to those routines.

    For subsets of *composite* tvbuffs, however, they don't go through the
    "fast path", and this bug shows up.

    This causes both crashes and misdissection of HTTP if the link-layer is
    PPP with Van Jacobson compression, as the decompression uses composite
    tvbuffs.

    Fixes #17254 and its many soon-to-be-duplicates.


    (cherry picked from commit 2ba52cdc0e4216dafdfc32498fc0210c99449ec9)


Command and args: /home/wireshark/builders/wireshark-3.4-fuzz/fuzztest/install.asan/bin/tshark  -nVxr
=================================================================
==3805788==ERROR: AddressSanitizer: allocator is out of memory trying to allocate 0x3c6000127c bytes
    #0 0x55c2ab93949d in malloc (/home/wireshark/builders/wireshark-3.4-fuzz/fuzztest/install.asan/bin/tshark+0xd649d)
    #1 0x7fd725994e98 in g_malloc (/lib/x86_64-linux-gnu/libglib-2.0.so.0+0x57e98)
    #2 0x7fd73230145b in wmem_strict_alloc /home/wireshark/builders/wireshark-3.4-fuzz/fuzztest/build/cmbuild/../epan/wmem/
    #3 0x7fd7322f7a30 in wmem_alloc /home/wireshark/builders/wireshark-3.4-fuzz/fuzztest/build/cmbuild/../epan/wmem/wmem_cc
    #4 0x7fd7304cc9f2 in dissect_CPMSetBindings /home/wireshark/builders/wireshark-3.4-fuzz/fuzztest/build/cmbuild/../epan/
    #5 0x7fd7304c9b74 in dissect_mswsp /home/wireshark/builders/wireshark-3.4-fuzz/fuzztest/build/cmbuild/../epan/dissector
    #6 0x7fd7304c8d6f in dissect_mswsp_smb2 /home/wireshark/builders/wireshark-3.4-fuzz/fuzztest/build/cmbuild/../epan/diss
    #7 0x7fd7324258b5 in dissector_try_heuristic /home/wireshark/builders/wireshark-3.4-fuzz/fuzztest/build/cmbuild/../epar
    #8 0x7fd730c9f278 in dissect_file_data_smb2_pipe /home/wireshark/builders/wireshark-3.4-fuzz/fuzztest/build/cmbuild/../
    #9 0x7fd730c979d5 in dissect_smb2_FSCTL_PIPE_TRANSCEIVE /home/wireshark/builders/wireshark-3.4-fuzz/fuzztest/build/cmbu
    #10 0x7fd730c965c2 in dissect_smb2_ioctl_data /home/wireshark/builders/wireshark-3.4-fuzz/fuzztest/build/cmbuild/../epa
    #11 0x7fd730cc2db9 in dissect_smb2_ioctl_data_in /home/wireshark/builders/wireshark-3.4-fuzz/fuzztest/build/cmbuild/../
    #12 0x7fd730cbe981 in dissect_smb2_olb_buffer /home/wireshark/builders/wireshark-3.4-fuzz/fuzztest/build/cmbuild/../epa
    #13 0x7fd730cb82d9 in dissect_smb2_ioctl_request /home/wireshark/builders/wireshark-3.4-fuzz/fuzztest/build/cmbuild/../
    #14 0x7fd730cab2e1 in dissect_smb2_command /home/wireshark/builders/wireshark-3.4-fuzz/fuzztest/build/cmbuild/../epan/c
    #15 0x7fd730ca87c2 in dissect_smb2 /home/wireshark/builders/wireshark-3.4-fuzz/fuzztest/build/cmbuild/../epan/dissector
    #16 0x7fd730c9e46f in dissect_smb2_heur /home/wireshark/builders/wireshark-3.4-fuzz/fuzztest/build/cmbuild/../epan/diss
    #17 0x7fd7324258b5 in dissector_try_heuristic /home/wireshark/builders/wireshark-3.4-fuzz/fuzztest/build/cmbuild/../epa
    #18 0x7fd730554e5e in dissect_netbios_payload /home/wireshark/builders/wireshark-3.4-fuzz/fuzztest/build/cmbuild/../epa
    #19 0x7fd730555b92 in dissect_nbss_packet /home/wireshark/builders/wireshark-3.4-fuzz/fuzztest/build/cmbuild/../epan/di
    #20 0x7fd73055161c in dissect_nbss /home/wireshark/builders/wireshark-3.4-fuzz/fuzztest/build/cmbuild/../epan/dissector
    #21 0x7fd73242a5a1 in call_dissector_through_handle /home/wireshark/builders/wireshark-3.4-fuzz/fuzztest/build/cmbuild/
    #22 0x7fd73241f550 in call_dissector_work /home/wireshark/builders/wireshark-3.4-fuzz/fuzztest/build/cmbuild/../epan/pa
    #23 0x7fd73241ee69 in dissector_try_uint_new /home/wireshark/builders/wireshark-3.4-fuzz/fuzztest/build/cmbuild/../epar
    #24 0x7fd730dcc920 in decode_tcp_ports /home/wireshark/builders/wireshark-3.4-fuzz/fuzztest/build/cmbuild/../epan/disse
    #25 0x7fd730dd354b in process_tcp_payload /home/wireshark/builders/wireshark-3.4-fuzz/fuzztest/build/cmbuild/../epan/di
    #26 0x7fd730dd0ca3 in desegment_tcp /home/wireshark/builders/wireshark-3.4-fuzz/fuzztest/build/cmbuild/../epan/dissecto
    #27 0x7fd730dce7e4 in dissect_tcp_payload /home/wireshark/builders/wireshark-3.4-fuzz/fuzztest/build/cmbuild/../epan/di
    #28 0x7fd730de0060 in dissect_tcp /home/wireshark/builders/wireshark-3.4-fuzz/fuzztest/build/cmbuild/../epan/dissectors
    #29 0x7fd73242a5a1 in call_dissector_through_handle /home/wireshark/builders/wireshark-3.4-fuzz/fuzztest/build/cmbuild/

==3805788==HINT: if you don't care about these errors you may set allocator_may_return_null=1
SUMMARY: AddressSanitizer: out-of-memory (/home/wireshark/builders/wireshark-3.4-fuzz/fuzztest/install.asan/bin/tshark+0xd6
==3805788==ABORTING
```

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 0

Link issues together to show that they're related or that one is blocking others. Learn more.

Related merge requests 3

MS-WSP: Don't allocate huge amounts of memory.

!2766

MS-WSP: Don't allocate huge amounts of memory.

!2781

MS-WSP: Don't allocate huge amounts of memory.

!2782

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

Activity

A Wireshark GitLab Utility added cli tshark scoped label 1 year ago

A Wireshark GitLab Utility added crash label 1 year ago

Gerald Combs made the issue visible to everyone 1 year ago

Gerald Combs mentioned in merge request !2766 (merged) 1 year ago

A Wireshark GitLab Utility closed via merge request !2766 (merged) 1 year ago

Gerald Combs mentioned in commit `04f9d3e0` 1 year ago

Gerald Combs mentioned in merge request !2781 (merged) 1 year ago

Gerald Combs mentioned in commit `01c31e7a` 1 year ago

Gerald Combs mentioned in merge request !2782 (merged) 1 year ago

Gerald Combs @geraldcombs · 1 year ago Owner

This has been assigned CVE-2021-22207.

Please register or sign in to reply