Closed Issue created 10 months ago by **Sharon Brizinov**

Heap overflow - OOB Read in PVFS2 dissector - dissect_pvfs2_getconfig_response

Summary

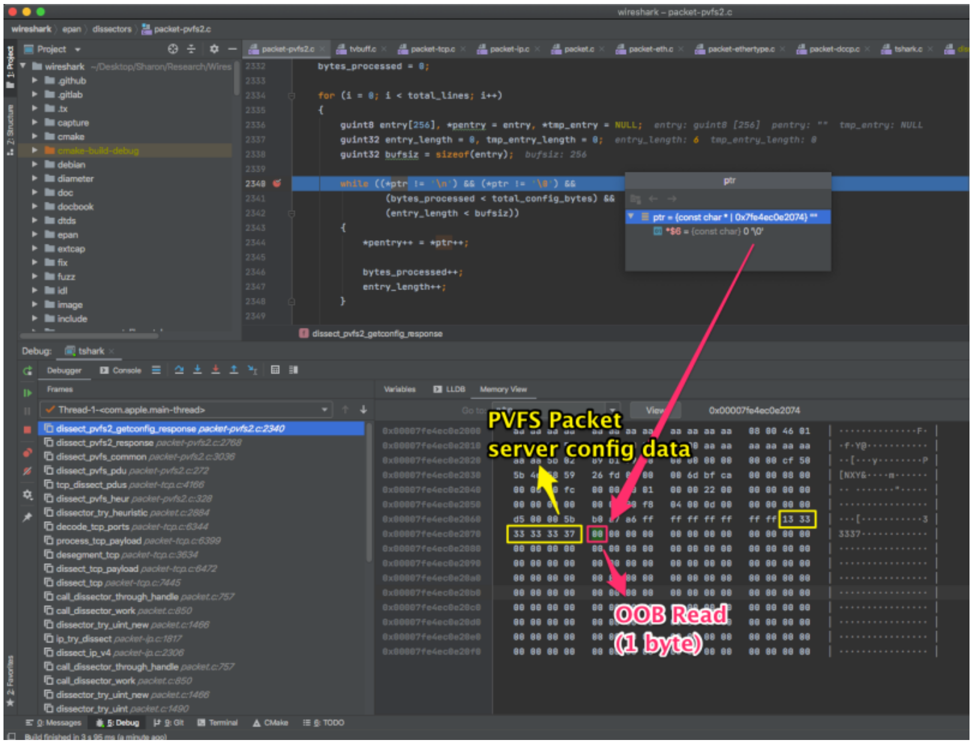
Heap overflow of type out-of-bounds read exists in PVFS dissector in the function `dissect_pvfs2_getconfig_response` . The OOB occurs due to bad use of a pointer which is dereferenced before checking bounds. AFAIK the bug exists since 2005, and while there were some efforts to improve and guard from dangerous edge-cases (see this fix), this specific OOB was left in the code.

Technical Details

Parallel Virtual File System (PVFS) is an open-source parallel file system. Usually its protocol runs over TCP port 3334, but it can also be detected heuristically by identifying the magic sequence 0xbfca0000 and some more attributes related to this protocol.

The protocol has several commands, and one of them is "Get Config". The response to this command should return a textual based config. This response will be parsed by the PVFS2 dissector in the `dissect_pvfs2_getconfig_response` function. However, due to a minor oversight a OOB read is possible when crafting a special PVFS2 packet.

The bug resides in a `for` loop which iteratres over all of the "get-config" response config lines. There is a pointer which is used to read the bytes from the config. The dissector will use this pointer to read the data while checking that there are more bytes to read before the end of the packet. However, it seems that the pointer is first being deference to check for a NEWLINE or NULL byte, before checking if end of buffer has reached. So as long as there are enough `total_lines` , the last cycle of the inner `while` loop will always lead to a heap overflow OOB read of 1 byte.



```
    bytes_processed = 0;

    for (i = 0; i < total_lines; i++)
    {
        guint8 entry[256], *pentry = entry, *tmp_entry = NULL;  entry: guint8 [256]  pentry: ""  tmp_entry: NULL
        guint32 entry_length = 0, tmp_entry_length = 0;  entry_length: 6  tmp_entry_length: 0
        guint32 bufsiz = sizeof(entry);  bufsiz: 256

        while ((*ptr != '\n') && (*ptr != '\0') &&
                (bytes_processed < total_config_bytes) &&
                (entry_length < bufsiz))
        {
            *pentry++ = *ptr++;

            bytes_processed++;
            entry_length++;
        }
```

```
ptr
ptr = {const char * | 0x7fe4ec0e2074} ""
    *$6 = {const char} 0 '\0'
```

```
Thread-1-<com.apple.main-thread>
dissect_pvfs2_getconfig_response packet-pvfs2.c:2340
dissect_pvfs2_response packet-pvfs2.c:2768
dissect_pvfs_common packet-pvfs2.c:3036
dissect_pvfs_pdu packet-pvfs2.c:272
tcp_dissect_pdus packet-tcp.c:4166
dissect_pvfs_heur packet-pvfs2.c:328
dissector_try_heuristic packet.c:2884
decode_tcp_ports packet-tcp.c:6344
process_tcp_payload packet-tcp.c:6399
desegment_tcp packet-tcp.c:3634
dissect_tcp_payload packet-tcp.c:6472
dissect_tcp packet-tcp.c:7445
call_dissector_through_handle packet.c:757
call_dissector_work packet.c:850
dissector_try_uint_new packet.c:1466
ip_try_dissect packet-ip.c:1817
dissect_ip_v4 packet-ip.c:2306
call_dissector_through_handle packet.c:757
call_dissector_work packet.c:850
dissector_try_uint_new packet.c:1466
dissector_try_uint packet.c:1490
```

```
Build finished in 3 s 95 ms (a minute ago)
```

IMO the fix should be simple - Moving the check `(bytes_processed < total_config_bytes)` before dereferencing ptr `(*ptr != '\n') && (*ptr != '\0')`

Steps to reproduce

Use the provided pcap. Run it with address sanitizer or debug tshark manually pvfs_heap_overflow_poc.pcap

What is the current bug behavior?

Heap overflow OOB read of 1 byte

What is the expected correct behavior?

Don't overflow beyond the allocated buffer.

Sample capture file

Attached pvfs_heap_overflow_poc.pcap

Build information

```
TShark (Wireshark) 3.7.0 (v3.7.0rc0-826-gb3215d99cacb)
```

Edited 10 months ago by Sharon Brizinov

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 1

fix PVFS2 oob read dissect_pvfs2_getconfig_response

!5783 ✓

Activity

Sharon Brizinov changed the description 10 months ago ·

Sharon Brizinov @sean007 · 10 months ago Author Contributor

Open a MR !5783 (merged)

Edited by Sharon Brizinov 10 months ago

Sharon Brizinov mentioned in merge request !5783 (merged) 10 months ago

Uli Heilmeier added lib wireshark scoped label 10 months ago

Jaap Keuter closed 10 months ago

Sharon Brizinov @sean007 · 10 months ago Author Contributor

@uhei can you please add `crash` label?

Uli Heilmeier added crash label 10 months ago

Gerald Combs made the issue visible to everyone 9 months ago

Gerald Combs @geraldcombs · 9 months ago Owner

This has been assigned CVE-2022-0583.

Please register or sign in to reply