New issue

Jump to bottom

MemBuffer assertions again in function MemBuffer::alloc() #486

Closed **chibataiki** opened this issue on Apr 7, 2021 · 0 comments

chibataiki commented on Apr 7, 2021 • edited

What's the problem (or question)?

Same problem like issue 448but not fix all the bug position. MemBuffer is attempted to be allocated with 0 bytes, failing an assertion in mem.cpp.

asan

```
./upx.out -d abort01
                       Ultimate Packer for eXecutables
                          Copyright (C) 1996 - 2021
UPX git-2638be+ Markus Oberhumer, Laszlo Molnar & John Reiser    Jan 1st 2021

        File size         Ratio      Format      Name
   --------------------   ------   -----------   -----------
upx.out: mem.cpp:216: void MemBuffer::alloc(upx_uint64_t): Assertion `size > 0' failed.
[1]    1517141 abort      ./upx.out -d abort01
```

gdb

```
gef➤  bt
#0  __GI_raise (sig=sig@entry=0x6) at ../sysdeps/unix/sysv/linux/raise.c:50
#1  0x00007ffff7a38859 in __GI_abort () at abort.c:79
#2  0x00007ffff7a38729 in __assert_fail_base (fmt=0x7ffff7bce588 "%s%s%s:%u: %s%sAssertion `%s' failed.\n%n", assertion=0xa54aa0 <str> "size > 0", file=0xa544b0 "mem.cpp",
line=0xd8, function=<optimized out>) at assert.c:92
#3  0x00007ffff7a49f36 in __GI___assert_fail (assertion=0xa54aa0 <str> "size > 0", file=0xa544b0 "mem.cpp", line=0xd8, function=0xa54a40 <__PRETTY_FUNCTION__._ZN9MemBuffer5allocEy>
"void MemBuffer::alloc(upx_uint64_t)") at assert.c:101
#4  0x0000000000550503 in MemBuffer::alloc (this=0x61a000001458, size=0x0) at mem.cpp:216
#5  0x00000000007833c1 in PackMachBase<N_Mach::MachClass_64<N_BELE_CTP::LEPolicy> >::canUnpack (this=0x61a000001280) at p_mach.cpp:1555
#6  0x0000000000942adb in try_unpack (p=0x61a000001280, user=0x7fffffffbe30) at packmast.cpp:114
#7  0x00000000009407ce in PackMaster::visitAllPackers (func=0x9425c0 <try_unpack(Packer*, void*)>, f=0x7fffffffbe30, o=0x7fffffffc4e8, user=0x7fffffffbe30) at packmast.cpp:225
#8  0x0000000000942428 in PackMaster::getUnpacker (f=0x7fffffffbe30) at packmast.cpp:248
#9  0x000000000094359b in PackMaster::unpack (this=0x7fffffffc4d0, fo=0x7fffffffbf40) at packmast.cpp:266
#10 0x0000000000a16d11 in do_one_file (iname=0x7fffffffdfb7 "abort_01", oname=0x7fffffffd0a0 "abort_01.007") at work.cpp:157
#11 0x0000000000a18c16 in do_files (i=0x2, argc=0x3, argv=0x7fffffffdbd8) at work.cpp:269
#12 0x00000000005359b3 in upx_main (argc=0x3, argv=0x7fffffffdbd8) at main.cpp:1516
#13 0x0000000000539d77 in main (argc=0x3, argv=0x7fffffffdbd8) at main.cpp:1584

gef➤  frame 4
#4  0x0000000000550503 in MemBuffer::alloc (this=0x61a000001458, size=0x0) at mem.cpp:216
216         assert(size > 0);

gef➤  frame 5
#5  0x00000000007833c1 in PackMachBase<N_Mach::MachClass_64<N_BELE_CTP::LEPolicy> >::canUnpack (this=0x61a000001280) at p_mach.cpp:1555
1555        rawmseg_buf.alloc(mhdri.sizeofcmds);

gef➤  p mhdri.sizeofcmds
$7 = {
  d = "\000\000\000"
}
```

What should have happened?

No failed assertions.

Do you have an idea for a solution?

Either remove the assertion (probably easier), or add logic to check that allocations are > 0 bytes.

or like the
c55b570 add some sanitize code.

How can we reproduce the issue?

1.Build UPX

```
export BUILD_TYPE_DEBUG=1
make all
```

2. Run

```
./src/upx.out -d poc
```

zipped poc :
alloc_abort01.zip

Please tell us details about your environment.

- UPX version used (`upx --version`):

```
./upx.out --version
upx 4.0.0-git-2638bee3c0f7+
UCL data compression library 1.03
zlib data compression library 1.2.11
LZMA SDK version 4.43
```

- Host Operating System and version:
  `OS: Ubuntu 20.04.2 LTS x86_64`
- Host CPU architecture:
  `CPU: Intel i5-4590 (4) @ 3.700GHz`
- Target Operating System and version:
  `same as Host`
- Target CPU architecture:
  `same as Host`

reporter: chiba of Topsec alphalab

chibataiki mentioned this issue on Apr 7, 2021

try fix issue 486 #487

Merged

jreiser closed this as completed on Apr 10, 2021

Assignees

No one assigned

Labels

None yet

Projects

None yet

Milestone

No milestone

Development

No branches or pull requests

2 participants