New issue

Jump to bottom

# Heap buffer overflow in get_le32() #388

Closed  **giantbranch** opened this issue on Jul 22, 2020 · 1 comment

**giantbranch** commented on Jul 22, 2020 • edited

Author: giantbranch of NSFOCUS Security Team

## What's the problem (or question)?

A heap buffer overflow read in the latest commit 4e1ae22a1a07be5135c68b25ff05058ae8ae48e1 of the devel branch

ASAN reports:

```
==6518==ERROR: AddressSanitizer: heap-buffer-overflow on address 0x616000000300 at pc 0x000000756793 bp 0x7ffdf8e9ed60 sp 0x7ffdf8e9ed58
READ of size 4 at 0x616000000300 thread T0
    #0 0x756792 in get_le32(void const*) /src/upx-multi/src/./bele.h:164:12
    #1 0x756792 in N_BELE_RTP::LEPolicy::get32(void const*) const /src/upx-multi/src/./bele_policy.h:192:18
    #2 0x5d0186 in Packer::get_te32(void const*) const /src/upx-multi/src/./packer.h:296:65
    #3 0x5d0186 in PackLinuxElf64::unpack(OutputFile*) /src/upx-multi/src/p_lx_elf.cpp:4643:28
    #4 0x6c7820 in Packer::doUnpack(OutputFile*) /src/upx-multi/src/packer.cpp:107:5
    #5 0x757f59 in do_one_file(char const*, char*) /src/upx-multi/src/work.cpp:160:12
    #6 0x7594b2 in do_files(int, int, char**) /src/upx-multi/src/work.cpp:271:13
    #7 0x555aed in main /src/upx-multi/src/main.cpp:1538:5
    #8 0x7f0daabe483f in __libc_start_main /build/glibc-e6zv40/glibc-2.23/csu/../csu/libc-start.c:291
    #9 0x41ce98 in _start (/out/upx-multi/upx-multi+0x41ce98)

0x616000000300 is located 40 bytes to the right of 600-byte region [0x616000000080,0x6160000002d8)
allocated by thread T0 here:
    #0 0x49519d in malloc (/out/upx-multi/upx-multi+0x49519d)
    #1 0x5697a7 in MemBuffer::alloc(unsigned long long) /src/upx-multi/src/mem.cpp:194:42

SUMMARY: AddressSanitizer: heap-buffer-overflow /src/upx-multi/src/./bele.h:164:12 in get_le32(void const*)
Shadow bytes around the buggy address:
  0x0c2c7fff8010: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x0c2c7fff8020: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x0c2c7fff8030: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x0c2c7fff8040: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x0c2c7fff8050: 00 00 00 00 00 00 00 00 00 00 00 fa fa fa fa fa
=>0x0c2c7fff8060:[fa]fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c2c7fff8070: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c2c7fff8080: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c2c7fff8090: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c2c7fff80a0: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c2c7fff80b0: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
Shadow byte legend (one shadow byte represents 8 application bytes):
  Addressable:           00
  Partially addressable: 01 02 03 04 05 06 07
  Heap left redzone:       fa
  Freed heap region:       fd
  Stack left redzone:      f1
  Stack mid redzone:       f2
  Stack right redzone:     f3
  Stack after return:      f5
  Stack use after scope:   f8
  Global redzone:          f9
  Global init order:       f6
  Poisoned by user:        f7
  Container overflow:      fc
  Array cookie:            ac
  Intra object redzone:    bb
  ASan internal:           fe
  Left alloca redzone:     ca
  Right alloca redzone:    cb
  Shadow gap:              cc
==6518==ABORTING
```

## What should have happened?

Check if the file is normal, exit if abnormal

## Do you have an idea for a solution?

Add more checks

## How can we reproduce the issue?

upx.out -d <poc_filename>

poc:
poc-heap-buffer-overflow-get_le32.tar.gz

## Please tell us details about your environment.

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

```
upx 4.0.0-git-4e1ae22a1a07+
UCL data compression library 1.03
```

```
zlib data compression library 1.2.8
LZMA SDK version 4.43
Copyright (C) 1996-2020 Markus Franz Xaver Johannes Oberhumer
Copyright (C) 1996-2020 Laszlo Molnar
Copyright (C) 2000-2020 John F. Reiser
Copyright (C) 2002-2020 Jens Medoch
Copyright (C) 1995-2005 Jean-loup Gailly and Mark Adler
Copyright (C) 1999-2006 Igor Pavlov
UPX comes with ABSOLUTELY NO WARRANTY; for details type 'upx-multi -L'.
```

- Host Operating System and version: Ubuntu 16.04.2 LTS
- Host CPU architecture: x86_64
- Target Operating System and version: same as Host
- Target CPU architecture: same as Host

**jreiser** added a commit that referenced this issue on Jul 22, 2020

`Unpack: Phdrs must be within expansion of first compressed block` ··· ✗ 87b73e5

**jreiser** commented on Jul 22, 2020 Collaborator

Fixed by above commit on `devel` branch.

```
$ upx -d -o foo poc*
poc-heap-buffer-overflow-get_le32.tar.gz: CantUnpackException: bad compressed e_phnum
```

**giantbranch** closed this as completed on Jul 26, 2020

**markus-oberhumer** pushed a commit that referenced this issue on Aug 17

`Unpack: Phdrs must be within expansion of first compressed block` ··· 0016512

Assignees

No one assigned

Labels

None yet

Projects

None yet

Milestone

No milestone

Development

No branches or pull requests

2 participants