PackLinuxElf::canUnpack did not check for ELF input #485

Closed **chibataiki** opened this issue on Apr 6, 2021 · 0 comments

chibataiki commented on Apr 6, 2021 • edited

What's the problem (or question)?

Null pointer dereference was discovered in upx in the latest commit of the devel branch. [2638bee]

During the pointer 'p' points to 0x0 in func get_ne32(). The issue can be triggered by different places, which can cause a denial of service.

ASAN reports:

```
         File size         Ratio      Format      Name
    --------------------   ------   -----------   -----------
p_lx_elf.cpp:2406:54: runtime error: member access within null pointer of type 'const Elf64_Phdr' (aka 'const Phdr<ElfITypes<LE16, LE32, LE64, LE64, LE64>>')
SUMMARY: UndefinedBehaviorSanitizer: undefined-behavior p_lx_elf.cpp:2406:54 in
AddressSanitizer:DEADLYSIGNAL
=================================================================
==3546154==ERROR: AddressSanitizer: SEGV on unknown address 0x000000000000 (pc 0x00000082a541 bp 0x7fffe268e150 sp 0x7fffe268e140 T0)
==3546154==The signal is caused by a READ memory access.
==3546154==Hint: address points to the zero page.
    #0 0x82a541 in get_ne32(void const*) /home/upx/src/./bele.h:48:5
    #1 0x82a541 in get_le32(void const*) /home/upx/src/./bele.h:136:50
    #2 0x82a541 in N_BELE_RTP::LEPolicy::get32(void const*) const /home/upx/src/./bele_policy.h:168:48
    #3 0x58717f in PackLinuxElf64::canUnpack() /home/upx/src/p_lx_elf.cpp:2406:38
    #4 0x79c0e1 in try_unpack(Packer*, void*) /home/upx/src/packmast.cpp:114:20
    #5 0x7955d2 in PackMaster::visitAllPackers(Packer* (*)(Packer*, void*), InputFile*, options_t const*, void*) /home/upx/src/packmast.cpp:194:9
    #6 0x79bdda in PackMaster::getUnpacker(InputFile*) /home/upx/src/packmast.cpp:248:18
    #7 0x79c768 in PackMaster::unpack(OutputFile*) /home/upx/src/packmast.cpp:266:9
    #8 0x82bd8c in do_one_file(char const*, char*) /home/upx/src/work.cpp:157:12
    #9 0x82d684 in do_files(int, int, char**) /home/upx/src/work.cpp:269:13
    #10 0x50e805 in upx_main(int, char**) /home/upx/src/main.cpp:1516:9
    #11 0x510e85 in main /home/upx/src/main.cpp:1584:13
    #12 0x7fbe9660a0b2 in __libc_start_main /build/glibc-eX1tMB/glibc-2.31/csu/../csu/libc-start.c:308:16
    #13 0x41d93d in _start (/home/upx/upx.out+0x41d93d)

AddressSanitizer can not provide additional info.
SUMMARY: AddressSanitizer: SEGV /home/upx/src/./bele.h:48:5 in get_ne32(void const*)
==3546154==ABORTING
```

debug info

```
── source:./bele.h+48 ────
     43       return v;
     44   }
     45
     46   __acc_static_forceinline unsigned get_ne32(const void *p) {
     47       upx_uint32_t v = 0;
              // p=0x00007fffffff9720  →  0x0000000000000000
 →   48       upx_memcpy_inline(&v, p, sizeof(v));
     49       return v;
     50   }
     51
     52   __acc_static_forceinline upx_uint64_t get_ne64(const void *p) {
     53       upx_uint64_t v = 0;

gef➤ bt
[#0] 0x4ff4cf → get_ne32(p=0x0)
[#1] 0x4ff4cf → get_le32(p=0x0)
[#2] 0xa1417c → N_BELE_RTP::LEPolicy::get32(this=0x1704740 <N_BELE_RTP::le_policy>, p=0x0)
[#3] 0x69bdf6 → Packer::get_te32(this=0x61b000000080, p=0x0)
[#4] 0x60123b → PackLinuxElf64::canUnpack(this=0x61b000000080)
[#5] 0x942adb → try_unpack(p=0x61b000000080, user=0x7fffffffbe10)
[#6] 0x93856c → PackMaster::visitAllPackers(func=0x9425c0 <try_unpack(Packer*, void*)>, f=0x7fffffffbe10, o=0x7fffffffc4c8, user=0x7fffffffbe10)
[#7] 0x942428 → PackMaster::getUnpacker(f=0x7fffffffbe10)
[#8] 0x94359b → PackMaster::unpack(this=0x7fffffffc4b0, fo=0x7fffffffbf20)
[#9] 0xa16d11 → do_one_file(iname=0x7fffffffdf7d "poc")

gef➤  p *p
Attempt to dereference a generic pointer.
gef➤  p p
$1 = (const void *) 0x0
```

What should have happened?

Decompress a crafted/suspicious file.

Do you have an idea for a solution?

This bug is coursed by `upx_memcpy_inline(&v, p, sizeof(v));` , the pointer isn't sanitized. Strengthen the sanitize of all pointer used in upx_memcpy_inline may helpful reduce the .

How can we reproduce the issue?

1. compile upx with address-sanitize
2. execute cmd
   upx.out -d $PoC

poc zipped
null_pointer_01_get32.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

jreiser changed the title ~~Null pointer dereference in function get_ne32()~~ **PackLinuxElf::canUnpack did not check for ELF input** on Apr 8, 2021

jreiser added a commit that referenced this issue on Apr 10, 2021

`PackLinuxElf::canUnpack must checkEhdr() for ELF input` ··· ✓ 90279ab

jreiser closed this as completed on Apr 10, 2021

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

`PackLinuxElf::canUnpack must checkEhdr() for ELF input` ··· be05069

Assignees

No one assigned

Labels

None yet

Projects

None yet

Milestone

No milestone

Development

No branches or pull requests

2 participants