New issue

Jump to bottom

Segmentation fault in PackLinuxElf64::adjABS of p_lx_elf.cpp #396

Closed giantbranch opened this issue on Jul 24, 2020 · 1 comment

giantbranch commented on Jul 24, 2020 • edited

Author: giantbranch of NSFOCUS Security Team

What's the problem (or question)?

Segmentation fault in PackLinuxElf64::adjABS of p_lx_elf.cpp in the latest commit of the devel branch

code:

```
──────────────────────────────────────────────────────────────────────[ REGISTERS ]──────────────────────────────────────────────────────────────────────
 RAX  0x53be80 (abs_symbol_names) ◂— pop    rdi /* '__bss_end__' */
 RBX  0xe8cf
 RCX  0xff
 RDX  0x0
*RDI  0x53be80 (abs_symbol_names) ◂— pop    rdi /* '__bss_end__' */
 RSI  0xff
 R8   0xa9d79
 R9   0x98c32
 R10  0x86f
 R11  0x246
 R12  0xff
 R13  0x7fffffffe440 ◂— 0x3
 R14  0x0
 R15  0x0
 RBP  0x7fffffffd450 —▸ 0x7fffffffd660 —▸ 0x7fffffffd680 —▸ 0x7fffffffd6a0 —▸ 0x7fffffffdea0 ◂— ...
 RSP  0x7fffffffd420 —▸ 0x7ffff7fb506e ◂— 0x40ffffff1
*RIP  0x442e5e ◂— call   0x401fd0
───────────────────────────────────────────────────────────────────────[ DISASM ]───────────────────────────────────────────────────────────────────────
   0x442e47    lea    rax, [rdx*8]
   0x442e4f    sub    rax, rdx
   0x442e52    add    rax, abs_symbol_names <0x53be80>
   0x442e58    mov    rsi, rcx
   0x442e5b    mov    rdi, rax
 ► 0x442e5e    call   strcmp@plt <0x401fd0>
        s1: 0x53be80 (abs_symbol_names) ◂— '__bss_end__'
        s2: 0xff

   0x442e63    test   eax, eax
   0x442e65    sete   al
   0x442e68    test   al, al
   0x442e6a    je     0x442e89

   0x442e6c    mov    eax, dword ptr [rbp - 0x24]
────────────────────────────────────────────────────────────────────[ SOURCE (CODE) ]────────────────────────────────────────────────────────────────────
   3132 int
   3133 PackLinuxElf64::adjABS(Elf64_Sym *sym, unsigned delta)
   3134 {
   3135     for (int j = 0; abs_symbol_names[j][0]; ++j) {
   3136         unsigned st_name = get_te32(&sym->st_name);
 ► 3137         if (!strcmp(abs_symbol_names[j], get_str_name(st_name, (unsigned)-1))) {
   3138             sym->st_value += delta;
   3139             return 1;
   3140         }
   3141     }
   3142     return 0;
──────────────────────────────────────────────────────────────────────[ STACK ]──────────────────────────────────────────────────────────────────────
00:0000│ rsp  0x7fffffffd420 —▸ 0x7ffff7fb506e ◂— 0x40ffffff1
01:0008│      0x7fffffffd428 ◂— 0xffffff00000804fd8
02:0010│      0x7fffffffd430 —▸ 0x7ffff7fb5068 ◂— 0xfff10000000000ff
03:0018│      0x7fffffffd438 —▸ 0x817030 —▸ 0x53e1f8 —▸ 0x43af84 (PackLinuxElf64amd::~PackLinuxElf64amd()) ◂— push   rbp
04:0020│      0x7fffffffd440 —▸ 0x7ffff7fb506e ◂— 0x40ffffff1
05:0028│      0x7fffffffd448 ◂— 0xff00000000
06:0030│ rbp  0x7fffffffd450 —▸ 0x7fffffffd660 —▸ 0x7fffffffd680 —▸ 0x7fffffffd6a0 —▸ 0x7fffffffdea0 ◂— ...
07:0038│      0x7fffffffd458 —▸ 0x449fab (PackLinuxElf64::unpack(OutputFile*)+3259) ◂— add    qword ptr [rbp - 0x158], 0x18
────────────────────────────────────────────────────────────────────[ BACKTRACE ]────────────────────────────────────────────────────────────────────
 ► f 0           442e5e
   f 1           449fab PackLinuxElf64::unpack(OutputFile*)+3259
   f 2           492eb2 Packer::doUnpack(OutputFile*)+90
   f 3           49c5ab PackMaster::unpack(OutputFile*)+109
   f 4           4b946e
   f 5           4b985f
   f 6           42aade main+746
   f 7     7ffff727b840 __libc_start_main+240
```

get_str_name returned an unreadable value and causing crash in strcmp

In this poc, get_str_name return 0xff

ASAN reports:

```
==7880==ERROR: AddressSanitizer: SEGV on unknown address 0x0000000000ff (pc 0x000000430045 bp 0x7fff7a4d5050 sp 0x7fff7a4d47f0 T0)
==7880==The signal is caused by a READ memory access.
==7880==Hint: address points to the zero page.
    #0 0x430045 in strcmp (/out/upx-multi/upx-multi+0x430045)
    #1 0x5b6c98 in PackLinuxElf64::adjABS(N_Elf64::Sym<N_Elf::ElfITypes<LE16, LE32, LE64, LE64, LE64> >*, unsigned int) /src/upx-multi/src/p_lx_elf.cpp:3137:14
    #2 0x5d06a9 in PackLinuxElf64::unpack(OutputFile*) /src/upx-multi/src/p_lx_elf.cpp:4611:25
    #3 0x6c82b0 in Packer::doUnpack(OutputFile*) /src/upx-multi/src/packer.cpp:107:5
    #4 0x7589f8 in do_one_file(char const*, char*) /src/upx-multi/src/work.cpp:160:12
    #5 0x759f42 in do_files(int, int, char**) /src/upx-multi/src/work.cpp:271:13
```

```
    #6 0x555afd in main /src/upx-multi/src/main.cpp:1538:5
    #7 0x7fb3d16be83f in __libc_start_main /build/glibc-e6zv40/glibc-2.23/csu/../csu/libc-start.c:291
    #8 0x41ce98 in _start (/out/upx-multi/upx-multi+0x41ce98)

AddressSanitizer can not provide additional info.
SUMMARY: AddressSanitizer: SEGV (/out/upx-multi/upx-multi+0x430045) in strcmp
==7880==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?

POC:
tests_7bc36b368db6594ef16f8abfd694fc11e4dc9acb_tar.gz

```
$ ./src/upx.out -d ./tests_7bc36b368db6594ef16f8abfd694fc11e4dc9acb_.tar.gz
                       Ultimate Packer for eXecutables
                          Copyright (C) 1996 - 2020
UPX git-8d1d60  Markus Oberhumer, Laszlo Molnar & John Reiser   Jan 24th 2020

        File size         Ratio      Format      Name
   --------------------   ------   -----------   -----------
Segmentation fault
```

Please tell us details about your environment.

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

```
upx 4.0.0-git-8d1d605b3d8c+
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 25, 2020

`Check de-compressed SHT_SYMTAB` ··· ✕ 624eb22

jreiser commented on Jul 25, 2020 Collaborator

Fixed on `devel` branch by above commit.

giantbranch closed this as completed on Jul 27, 2020

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

`Check de-compressed SHT_SYMTAB` ··· 77c914b

Assignees

No one assigned

Labels

None yet

Projects

None yet

Milestone

No milestone

Development

No branches or pull requests

2 participants