New issue

Jump to bottom

SEGV in function elf::section::as_strtab at elf/elf.cc:284 #49

Open **xiaoxiongwang** opened this issue on Aug 15, 2020 · 1 comment

xiaoxiongwang commented on Aug 15, 2020 • edited

Tested in Ubuntu 16.04, 64bit.

The tested program is the example program dump-syms.

The testcase is dump_syms_segv.

I use the following command:

```
/path-to-libelfin/examples/dump-syms dump_syms_segv
```

and get:

```
Segmentation fault (core dumped)
```

I use **valgrind** to analysis the bug and get the below information (absolute path information omitted):

```
valgrind /path-to-libelfin/examples/dump-syms dump_syms_segv
==13575== Memcheck, a memory error detector
==13575== Copyright (C) 2002-2015, and GNU GPL'd, by Julian Seward et al.
==13575== Using Valgrind-3.11.0 and LibVEX; rerun with -h for copyright info
==13575== Command: /path-to-libelfin/examples/dump-syms dump_syms_segv
==13575==
==13575== Invalid read of size 4
==13575==    at 0x40A8A3: elf::section::as_strtab() const (elf.cc:284)
==13575==    by 0x40BD91: elf::section::as_symtab() const (elf.cc:295)
==13575==    by 0x401FD8: main (dump-syms.cc:32)
==13575==  Address 0x14 is not stack'd, malloc'd or (recently) free'd
==13575==
==13575==
==13575== Process terminating with default action of signal 11 (SIGSEGV)
==13575==  Access not within mapped region at address 0x14
==13575==    at 0x40A8A3: elf::section::as_strtab() const (elf.cc:284)
==13575==    by 0x40BD91: elf::section::as_symtab() const (elf.cc:295)
==13575==    by 0x401FD8: main (dump-syms.cc:32)
==13575==  If you believe this happened as a result of a stack
==13575==  overflow in your program's main thread (unlikely but
==13575==  possible), you can try to increase the size of the
==13575==  main thread stack using the --main-stacksize= flag.
==13575==  The main thread stack size used in this run was 8388608.
Symbol table '.dynsym':
   Num: Value            Size  Type    Binding Index Name
==13575==
==13575== HEAP SUMMARY:
==13575==     in use at exit: 79,384 bytes in 50 blocks
==13575==   total heap usage: 62 allocs, 12 frees, 84,776 bytes allocated
==13575==
==13575== LEAK SUMMARY:
==13575==    definitely lost: 0 bytes in 0 blocks
==13575==    indirectly lost: 0 bytes in 0 blocks
==13575==      possibly lost: 0 bytes in 0 blocks
==13575==    still reachable: 79,384 bytes in 50 blocks
==13575==         suppressed: 0 bytes in 0 blocks
==13575== Rerun with --leak-check=full to see details of leaked memory
==13575==
==13575== For counts of detected and suppressed errors, rerun with: -v
==13575== ERROR SUMMARY: 1 errors from 1 contexts (suppressed: 0 from 0)
Segmentation fault (core dumped)
```

I use **AddressSanitizer** to build ffjpeg and running it with the following command:

```
/path-to-libelfin/examples/dump-syms dump_syms_segv
```

This is the ASAN information (absolute path information omitted):

```
/path-to-libelfin-address/examples/dump-syms dump_syms_segv
ASAN:SIGSEGV
=================================================================
==13619==ERROR: AddressSanitizer: SEGV on unknown address 0x000000000014 (pc 0x000000409050 bp 0x7ffdc34bbe50 sp 0x7ffdc34bbe10 T0)
    #0 0x40904f in elf::section::as_strtab() const /path-to-libelfin-address/elf/elf.cc:284
    #1 0x4099f5 in elf::section::as_symtab() const /path-to-libelfin-address/elf/elf.cc:295
    #2 0x4023fa in main /path-to-libelfin-address/examples/dump-syms.cc:32
    #3 0x7fae884aa82f in __libc_start_main (/lib/x86_64-linux-gnu/libc.so.6+0x2082f)
    #4 0x403728 in _start (/path-to-libelfin-address/examples/dump-syms+0x403728)

AddressSanitizer can not provide additional info.
SUMMARY: AddressSanitizer: SEGV /path-to-libelfin-address/elf/elf.cc:284 elf::section::as_strtab() const
==13619==ABORTING
```

An attacker can exploit this vulnerability by submitting a malicious elf file that exploits this bug which will result in a Denial of Service (DoS).

1

fgeek commented on Aug 6, 2021

CVE-2020-24826 has been assigned for this issue.

Assignees

No one assigned

Labels

None yet

Projects

None yet

Milestone

No milestone

Development

No branches or pull requests

2 participants