New issue

Jump to bottom

out-of-bounds read in function write_title() in subs.c #85

Closed **chibataiki** opened this issue on Apr 27, 2021 · 4 comments

chibataiki commented on Apr 27, 2021 • edited

In Version `0cf4a55`
Out-of-bounds read found in function write_title() in subs.c. The flow allows attackers to cause denial of service.

Here didn't check whether `&s->text[2]` is valid .

gdb info:

```
── source:subs.c+1465 ────
   1460  void write_title(struct SYMBOL *s)
   1461  {
   1462          char *p;
   1463          float sz;
   1464
         // s=0x00007fffffffe008  →  0x0000000000433a4b ("K:C"?), p=0x00007fffffffe018  →  [...]  →  0x3a4d14736d616542
 → 1465          p = &s->text[2];
   1466          if (*p == '\0')
   1467                  return;
   1468          if (s == info['T' - 'A']) {
   1469                  sz = cfmt.font_tb[TITLEFONT].size;
   1470                  bskip(cfmt.titlespace + sz);
── threads ────
[#0] Id 1, Name: "abcm2ps", stopped 0x5555555aaad3 in write_title (), reason: SIGSEGV
──── trace ────
[#0] 0x5555555aaad3 → write_title(s=0x433a4b)
[#1] 0x5555555abc2f → write_heading()
[#2] 0x55555559cd23 → get_info(s=0x5555556205c0)
[#3] 0x55555559e658 → do_tune()
[#4] 0x55555555e300 → abc_parse(p=0x55555561e0e0 "", fname=0x5555555fab00 "afl-collect-epsf/s1:id:002115,sig:11,src:019963,time:101664885,op:havoc,rep:4", ln=0x38)
[#5] 0x555555584b9e → txt_add_eos(fname=0x5555555fab00 "afl-collect-epsf/s1:id:002115,sig:11,src:019963,time:101664885,op:havoc,rep:4", linenum=0x38)
[#6] 0x555555585d6e → frontend(s=0x55555561d2b3 "X:1\nT:Beams\024M:C\nK:C\n;\n&[\322-n", 'E' <repeats 11 times>, "\377EEEEEE\n&[B-nEK:\n&[DC\016KX: ?-c,C\275@:\n&[B-nK:\n&)))))))
X:1", ftype=0x0, fname=0x5555555fab00 "afl-collect-epsf/s1:id:002115,sig:11,src:019963,time:101664885,op:havoc,rep:4", linenum=0x38)
[#7] 0x55555555c4ba → treat_file(fn=0x7fffffffe6a4 "afl-collect-epsf/s1:id:002115,sig:11,src:019963,time:101664885,op:havoc,rep:4", ext=0x5555555ba0a6 "abc")
[#8] 0x55555555c5ae → treat_abc_file(fn=0x7fffffffe6a4 "afl-collect-epsf/s1:id:002115,sig:11,src:019963,time:101664885,op:havoc,rep:4")
[#9] 0x55555555dc03 → main(argc=0x0, argv=0x7fffffffe420)
──
gef➤  p &s
$1 = (struct SYMBOL **) 0x7fffffffe008
gef➤  p &s->text
$2 = (char **) 0x433b03
gef➤  p &s->text[2]
Cannot access memory at address 0x433b03
```

reproduce : (poc zipped)

```
unzip  [poc].zip
abcm2ps -E [poc]
```

out-of-bounds-read_subs.c+1465_write_title.zip

reporter: chiba of topsec alphalab

chibataiki mentioned this issue on Apr 27, 2021

try fix issue #85 #86

Closed

moinejf commented on Apr 27, 2021 Collaborator

I could not reproduce the problem on my machine ARM 32 bits.
But, anyway, I wonder how the pointer can be out of bound: the function write_title() is always called when s->text contains a string starting with "T:".
So, may be give me the value of s->text when the problem occurs?

chibataiki commented on Apr 27, 2021 Author

I also could not reproduce the problem in my aarch64 machine.

In my x86_64 machine , here is the values.

```
gef➤  p s
$12 = (struct SYMBOL *) 0x433a4b
gef➤  p s->text
Cannot access memory at address 0x433b03
gef➤  p &s->text
$13 = (char **) 0x433b03
gef➤  p &s->text[2]
Cannot access memory at address 0x433b03
```

moinejf commented on Apr 28, 2021 Collaborator

Just an idea.
Some data in some symbols could be changed on wrong duration in voice overlay. This problem has been fixed by the commit `2f56e11`.
But, as there are voice overlay errors in the ABC file of this issue, may you try it again after applying the last commits?

chibataiki commented on Apr 28, 2021 Author

Seem fix the bug, thanks for your work!

chibataiki closed this as completed on Apr 29, 2021

Assignees

No one assigned

Labels

None yet

Projects

None yet

Milestone

No milestone

Development

No branches or pull requests

2 participants