

New issue

Jump to bottom

Use-after-free (heap) in the SFD_GetFontMetaData() function #4084

Closed fcambus opened this issue on Jan 3, 2020 · 9 comments · Fixed by #4091

Labels

untrusted input

fcambus commented on Jan 3, 2020

Hi,

While fuzzing FontForge with AFL, I found a heap use-after-free in the SFD_GetFontMetaData() function, in sfd.c.

Attaching a reproducer (gzipped so GitHub accepts it): [test01.sfd.gz](#)

Issue can be reproduced in FontForge 20190801 and with latest Git master by running:

fontforge test01.sfd

```
=====
==7418==ERROR: AddressSanitizer: heap-use-after-free on address 0x603000c8068 at pc 0x7f60f1122f2d bp 0x7fff16e6ff70 sp 0x7fff16e6ff78
WRITE of size 48 at 0x603000c8068 thread T0
#0 0x7f60f1122f2c (/usr/lib/x86_64-linux-gnu/libasan.so.5+0x67f2c)
#1 0x7f60ef2c7850 in SFD_GetFontMetaData /home/fcambus/fontforge-20190801/fontforge/sfd.c:8008
#2 0x7f60ef2ccf66 in SFD_GetFont /home/fcambus/fontforge-20190801/fontforge/sfd.c:8502
#3 0x7f60ef2d3e0f in SFD_Read /home/fcambus/fontforge-20190801/fontforge/sfd.c:9077
#4 0x7f60ef2d4360 in _SFDRead /home/fcambus/fontforge-20190801/fontforge/sfd.c:9110
#5 0x7f60ef32d7dd in _ReadSplineFont /home/fcambus/fontforge-20190801/fontforge/splinefont.c:1178
#6 0x7f60ef32f0bf in ReadSplineFont /home/fcambus/fontforge-20190801/fontforge/splinefont.c:1321
#7 0x7f60ef32f591 in LoadSplineFont /home/fcambus/fontforge-20190801/fontforge/splinefont.c:1379
#8 0x7f60eeeee4db in ViewPostScriptFont /home/fcambus/fontforge-20190801/fontforge/fontviewbase.c:1347
#9 0x7f60ef0d3c89e in fontforge_main /home/fcambus/fontforge-20190801/fontforgeexe/startui.c:1392
#10 0x557137b971ec in main /home/fcambus/fontforge-20190801/fontforgeexe/main.c:33
#11 0x7f60f03c81e2 in __libc_start_main (/lib/x86_64-linux-gnu/libc.so.6+0x271e2)
#12 0x557137b9710d in _start (/home/fcambus/fontforge-20190801/fontforgeexe/.libs/fontforge+0x110d)

0x603000c8070 is located 0 bytes to the right of 32-byte region [0x603000c8050,0x603000c8070)
freed by thread T0 here:
#0 0x7f60ef11c86ef in __interceptor_free (/usr/lib/x86_64-linux-gnu/libasan.so.5+0x10d6ef)
#1 0x7f60ee24c014 in _XReply (/usr/lib/x86_64-linux-gnu/libX11.so.6+0x40014)

previously allocated by thread T0 here:
#0 0x7f60ef11c8ae8 in malloc (/usr/lib/x86_64-linux-gnu/libasan.so.5+0x10dae8)
#1 0x7f60ed7a3cfd (/usr/lib/x86_64-linux-gnu/libxcb.so.1+0xdcfd)

SUMMARY: AddressSanitizer: heap-use-after-free (/usr/lib/x86_64-linux-gnu/libasan.so.5+0x67f2c)
Shadow bytes around the buggy address:
 0x0c0680010fb0: fd fd fd fa fa fa fd fd fa fa fa fd fd fd fa
 0x0c0680010fc0: fa fa fd fd fd fa fa fa fd fd fa fa fa fd fd
 0x0c0680010fd0: fd fd fa fa fd fd fd fa fa fa fd fd fa fa fa
 0x0c0680010fe0: fd fd fd fa fa fa fd fd fd fa fa 00 00 04 fa
 0x0c0680010ff0: fa fa 00 00 00 00 fa fa fd fd fd fa fa fa fd fd
=>0x0c0680011000: fd fd fa fa 00 00 00 fa fa fd fd fd[fd]fa fa
0x0c0680011010: 00 00 00 fa fa fa fa fa fa fa fa fa fa fa fa
0x0c0680011020: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
0x0c0680011030: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
0x0c0680011040: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
0x0c0680011050: 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
==7418==ABORTING
```

fcambus commented on Jan 3, 2020

Author

This issue has been assigned [CVE-2020-5395](#).

NicoleG25 commented on Jan 5, 2020

Is there any plan to address this & #4085 ?
Thanks :)

ctrlcctrlv commented on Jan 5, 2020

Member

I have no plans to work on this, but someone else might do so.
See #4086 (comment) for some reasoning

ctrlcctrlv added the **untrusted input** label on Jan 5, 2020

NicoleG25 commented on Jan 5, 2020

@ctrlcctrlv Do you plan to contact MITRE to get this CVE disputed ?
or do you agree with their analysis ?

I have no plans to work on this, but someone else might do so.
See #4086 (comment) for some reasoning

ctrlcctrlv commented on Jan 5, 2020

Member

Sorry, I don't understand. What is there to dispute? What analysis has been made?

NicoleG25 commented on Jan 5, 2020 • edited

Sorry, I don't understand. What is there to dispute? What analysis has been made?

@ctrlcctrlv

Do you agree with the analysis that "FontForge 20190801 has a use-after-free in SFD_GetFontMetaData in sfld.c" according to MITRE ? i.e. vulnerable to use-after-free ? or are you claiming that IT IS vulnerable and you don't plan to address this vulnerability ?
Thanks in advance :)

ctrlcctrlv commented on Jan 5, 2020

Member

It's vulnerable. I have no interest in fixing it, I think doing so would be a misallocation of resources. What I choose to spend my time on doesn't bind any other developer, but it doesn't bode well either because there are only so many of us, and probably others agree with me about how to allocate their time.

NicoleG25 commented on Jan 5, 2020

It's vulnerable. I have no interest in fixing it, I think doing so would be a misallocation of resources. What I choose to spend my time on doesn't bind any other developer, but it doesn't bode well either because there are only so many of us, and probably others agree with me about how to allocate their time.

Thank you for your explanation and patience !
Cheers.

skef commented on Jan 5, 2020

Contributor

@NicoleG25 Just to summarize the explanation @ctrlcctrlv referred to in a somewhat different way:

The primary three ways that software systems present security vulnerabilities, given problems like those reported in those issues (buffer overflows, wild pointers, etc.) are:

1. They take "untrusted" inputs of one sort or another, usually over a network. Those inputs can be carefully constructed to exploit the wild pointer or buffer overflow.
2. The program is itself available over the network in some way, like a mail transport agent, and nefarious people can interact with it directly.
3. The program executes with "elevated privileges", and can be manipulated by a user without those privileges in order to gain them.

Without some connection like this a segfault is "just a segfault" -- it doesn't present a security risk. A user gains nothing by tricking FontForge (for example) into executing carefully constructed instructions because FontForge doesn't run with elevated privileges. And FontForge does present any network interfaces (at least since the collaboration system was removed).

So any viable vulnerabilities would have to be in the form of a carefully constructed font file, which would exploit a buffer overflow or wild pointer to do something when loaded. This scenario is *possible* but *very improbable*. The main problem facing the hacker is that FontForge users tend to either develop their own fonts or open specific existing fonts for reasons very specific to them. The other problem is that FontForge is cross-platform and the file would have to be constructed to work on the specific OS of the targeted user.

Having said all this, I fixed a bunch of these small issues before the last release and I may fix some or all of these.

skef pushed a commit to skef/fontforge that referenced this issue on Jan 6, 2020

Fix for fontforge#4084 Use-after-free (heap) in the SFD_GetFontMetaDa... ..

8da6d56

skef mentioned this issue on Jan 6, 2020

Misc CVE fixes #4091

→ Merged

ctrlcctrlv closed this as completed in #4091 on Jan 6, 2020

ctrlcctrlv pushed a commit that referenced this issue on Jan 6, 2020

Fix for #4084 Use-after-free (heap) in the SFD_GetFontMetaData() func_ ...

048a91e

 skef mentioned this issue on Jan 7, 2020

Heap-based buffer overflow in the Type2NotDefSplines() function #4085


 Closed

 pnemade mentioned this issue on Feb 15, 2020

segfault in SFDDumpUTF7Str #4164

 Closed

 8 tasks

 erictapen added a commit to erictapen/nixpkgs that referenced this issue on May 25, 2020

 fontforge: patch for CVE-2020-5395 and CVE-2020-5496 ...

156fc5f

 Omnikron13 pushed a commit to Omnikron13/fontforge that referenced this issue on May 31


 Fix for fontforge#4084 Use-after-free (heap) in the SFD_GetFontMetaDa_ ...

e780d99

Assignees

No one assigned

Labels

 untrusted input

Projects

None yet

Milestone

No milestone

Development

Successfully merging a pull request may close this issue.

 Misc CVE fixes
skef/fontforge

4 participants

