QTBUG-91507

Out of bounds read in function `QRadialFetchSimd<QSimdSse2>::fetch` when input craft svg file

Details

Type:	Bug	Status:	CLOSED
Priority:	P1: Critical	Resolution:	Done
Affects Version/s:	5.15.1, (3)	Fix Version/s:	5.12.11, (4)
Component/s:	GUI: Painting, SVG Support		
Labels:	None		
Platform/s:	Linux/Other display system		
Commits:	bfd6ee0d8cf34b63d32adf10ed93daa0086b359f (qt/qtsvg/dev) 0fa522904d65b73d48d5fadf690131e9ebb58d2a (qt/qtsvg/6.0) 9f7ccbfc68d20d0dc2ddc1e7dee5572dcf7dcd48 (qt/qtsvg/6.1) 7bbf88403fd2d1fe79fab7c8e469f8aeafeb7372 (qt/tqtc-qtsvg/tqtc/lts-5.15)		

Description

To Reproduce

```
./qtsvg_svg_qsvgrenderer_render ./1.svg
```

Debug Info

```
# ./qtsvg_svg_qsvgrenderer_render ./1.svg
INFO: Seed: 3360833592
./qtsvg_svg_qsvgrenderer_render: Running 1 inputs 1 time(s) each.
Running: ./1.svg
UndefinedBehaviorSanitizer:DEADLYSIGNAL
==12881==ERROR: UndefinedBehaviorSanitizer: SEGV on unknown address 0xffffffe01e5cbd0 (pc 0x00000086cd75 bp 0x7fffffff7bb0 sp 0x7fffffff7a30 T12881)
==12881==The signal is caused by a READ memory access.
    #0 0x86cd75 in QRadialFetchSimd<QSimdSse2>::fetch(unsigned int*, unsigned int*, Operator const*, QSpanData const*, double, double, double, double, double)
/src/qt/qtbase/src/gui/painting/qdrawhelper_p.h:601:13
    #1 0x86c140 in unsigned int const* qt_fetch_radial_gradient_template<QRadialFetchSimd<QSimdSse2>, unsigned int>(unsigned int*, Operator const*, QSpanData const*, int, int, int)
/src/qt/qtbase/src/gui/painting/qdrawhelper_p.h:469:9
    #2 0x86beed in qt_fetch_radial_gradient_sse2(unsigned int*, Operator const*, QSpanData const*, int, int, int) /src/qt/qtbase/src/gui/painting/qdrawhelper_sse2.cpp:515:12
    #3 0x81a1e6 in BlendSrcGeneric::fetch(int, int, int) /src/qt/qtbase/src/gui/painting/qdrawhelper.cpp:3141:16
    #4 0x81a09d in void handleSpans<BlendSrcGeneric>(int, QT_FT_Span_ const*, QSpanData const*, BlendSrcGeneric&) /src/qt/qtbase/src/gui/painting/qdrawhelper.cpp:3085:56
    #5 0x807b6a in blend_src_generic(int, QT_FT_Span_ const*, void*) /src/qt/qtbase/src/gui/painting/qdrawhelper.cpp:3193:5
    #6 0x807a0d in qBlendGradient(int, QT_FT_Span_ const*, void*) /src/qt/qtbase/src/gui/painting/qdrawhelper.cpp
    #7 0x6645a4 in qt_span_fill_clipRect(int, QT_FT_Span_ const*, void*) /src/qt/qtbase/src/gui/painting/qpaintengine_raster.cpp:4166:9
    #8 0x83f79b in drawPixel(QCosmeticStroker*, int, int, int) /src/qt/qtbase/src/gui/painting/qcosmeticstroker.cpp:165:13
    #9 0x842a6f in bool drawLineAA<&(drawPixel(QCosmeticStroker*, int, int, int)), (anonymous namespace)::NoDasher>(QCosmeticStroker*, double, double, double, double, int)
/src/qt/qtbase/src/gui/painting/qcosmeticstroker.cpp:1070:21
    #10 0x83fda3 in QCosmeticStroker::drawPath(QVectorPath const&) /src/qt/qtbase/src/gui/painting/qcosmeticstroker.cpp:575:21
    #11 0x65c310 in QRasterPaintEngine::stroke(QVectorPath const&, QPen const&) /src/qt/qtbase/src/gui/painting/qpaintengine_raster.cpp:1622:17
    #12 0x84432e in QEmulationPaintEngine::stroke(QVectorPath const&, QPen const&) /src/qt/qtbase/src/gui/painting/qemulationpaintengine.cpp
```

I think the invalid sign extension on 0x86cd6d causes integer overflow to be the root cause of this vulnerability

```
[----------------------------------registers-----------------------------------]
RAX: 0x0
RBX: 0xffffffff
RCX: 0xffffffff
RDX: 0x1e5cea0 --> 0xffff0000ffff0000
RSI: 0xffffffff80000000
     ^^^^^^^^^^^^^^^^^^
RDI: 0x7ff
RBP: 0x7fffffff7b50 --> 0x7fffffff7be0 --> 0x7fffffff7c00 --> 0x7fffffff7c40 --> 0x7fffffff7cc0 --> 0x7fffffffbe30 (--> ...)
RSP: 0x7fffffff79d0 --> 0x7ff000007ff
RIP: 0x86cd75 (<QRadialFetchSimd<QSimdSse2>::fetch(unsigned int*, unsigned int*, Operator const*, QSpanData const*, double, double, double, double, double)+725>:       and    ecx,DWORD
PTR [rdx+rsi*4])
R8 : 0x0
R9 : 0x1
R10: 0x7fffffffc0a8 --> 0xbff0000000000000
R11: 0x1
R12: 0x7fffffff7ce0 --> 0x3fbeb85100000003
R13: 0x7fffffff9d78 --> 0x0
R14: 0x7fffffff9d7c --> 0x0
R15: 0x1e545c8 --> 0x1e4d310 --> 0xf6d8cb00
EFLAGS: 0x286 (carry PARITY adjust zero SIGN trap INTERRUPT direction overflow)
[-------------------------------------code-------------------------------------]
   0x86cd64 <QRadialFetchSimd<QSimdSse2>::fetch(unsigned int*, unsigned int*, Operator const*, QSpanData const*, double, double, double, double, double)+708>:  or     ecx,ebx
```

Environment:

- version : Qt branch 6.0.0 master (4e43ed1f939a797a8562361145713be8a0780365)
- OS: Ubuntu 16.04
- clang version: 11

Additional context

compile argument:

https://github.com/google/oss-fuzz/tree/master/projects/qt

Credit: 1vanChen of NSFOCUS Security Team

Attachments

1.svg
03 Mar '21 00:52 2 kB

1-1.svg
02 Nov '22 06:30 2 kB

Gerrit Reviews

No reviews matched the request. Check your Options in the drop-down menu of this sections header.

Activity

Allan Sandfeld Jensen added a comment - 03 Mar '21 15:16

Can't reproduce that with current dev branch

Robert Löhning added a comment - 03 Mar '21 16:22

First of all I'd like to thank you for running my scripts and reporting your findings here.

I can reproduce the issue with Qt 6.0.0, 6.0.1 and with the dev branch from February 15th. I'll try with more recent versions but I need to build them first.

Robert Löhning added a comment - 03 Mar '21 16:31

I uploaded the input to oss-fuzz which can reproduce the issue in today's dev branch.

Allan Sandfeld Jensen added a comment - 03 Mar '21 16:44

Anyway the only interesting detail about the SVG is an absurdly large r for the radialGradiant. If it isn't already fixed, it likely just needs to be sanitized befor ebeing used.

Robert Löhning added a comment - 04 Mar '21 12:06

For reference, Google now also assigned an issue number to this: https://bugs.chromium.org/p/oss-fuzz/issues/detail?id=31668
That report will be published in 90 days, the latest.

Robert Löhning added a comment - 09 May '22 18:19

No Gerrit Bot, this is *not* fixed in Qt 5.15.3.

People

Assignee:

Allan Sandfeld Jensen

Reporter:

chen ivan

Votes:

0 Vote for this issue

Watchers:

4 Start watching this issue

Dates

Created:

03 Mar '21 00:53

Updated:

02 Nov '22 06:30

Resolved:

04 Mar '21 22:10

Gerrit Reviews

There are no open Gerrit changes

There are 5 closed Gerrit changes

There are 5 closed Gerrit changes

Clamp parsed doubles to float representable values

Clamp parsed doubles to float representable values

Clamp parsed doubles to float representable values

Clamp parsed doubles to float representable values

Clamp parsed doubles to float representable values