FreeCiv - Bug #858212
Qt heap-use-after-free entering Pick Nation screen (2.6.1+)
2020-02-02 07:33 AM - Chippo Elder

Status: Closed
Priority: Normal
Assignee: Marko Lindqvist
Category: gui-qt
Sprint/Milestone: 2.6.2

Description

This BT/dump/thingy has appeared in another ticket #858069, but that ticket has 3 different dumps for between 3 and 5 different bugs and a completely inappropriate title.

To trigger it you need to compile with -fsanitze=address and I used Ipvlm, but it should be the same with gcc. Then, sadly, you have to play for a while. Load a savegame, do a turn or two, start a New Game with a new ruleset, play a turn. Then when it's primed, Leave Game, Start New Game, don't touch anything (ruleset says default) except you click Pick Nation, and it dumps (which is quite pretty, with colours and all).

I can reliably reproduce it, but any exact written instructions never seem to work more than once.

chippo@chippo-Aspire-V3-731:~$ ASAN_OPTIONS="detect_leaks=0" freeciv-qt-26 -F
2: Loading tileset "amplio2".
2: Loading tileset "delta2".
2: Loading tileset "amplio2".
QSocketNotifier: Invalid socket 41 and type 'Read', disabling...
2: last message repeated 2 times
2: Loading tileset "delta2". 

2826ERROR: AddressSanitizer: heap-use-after-free on address 0x6030027b16b8 at pc 0x00000099c5e4 bp 0x7ffe6f90c350 sp 0x7ffe6f90c348

READ of size 1 at 0x6030027b16b8 thread T0
#0 0x99c5e3 in skip_intl_qualifier_prefix /home/chippo/Downloads/git_clones/freeciv/utility/fcintl.c:48:7
#1 0x845c3c in nation_set_by_rule_name /home/chippo/Downloads/git_clones/freeciv/common/nation.c:766:23
#2 0x72f040 in races_dialog::nationset_changed(int) /home/chippo/Downloads/git_clones/freeciv/client/gui-qt/dialogs.cpp:984:10
#3 0x59f240 in racesdialogs::qt_static_metacall(QObject*, QMetaObject::Call, int, void**) /home/chippo/Downloads/git_clones/freeciv/client/gui-qt/meta_dialogs.cpp:369:21
#4 0x7efc52f577d3 in QMetaObject::activate(QObject*, int, int, void**) (/usr/lib/x86_64-linux-gnu/libQt5Core.so.5+0x2b1467)
#5 0x7efc52f570fc4 in QComboBox::currentIndexChanged(int) (/usr/lib/x86_64-linux-gnu/libQt5Widgets.so.5+0x270c74)
#6 0x7efc52f5712075 (/usr/lib/x86_64-linux-gnu/libQt5Widgets.so.5+0x273075)
#7 0x7efc52f5714893 (/usr/lib/x86_64-linux-gnu/libQt5Widgets.so.5+0x27589c)
#8 0x7efc52f571af2 in QComboBox::setCurrentIndex(int) (/usr/lib/x86_64-linux-gnu/libQt5Widgets.so.5+0x275af2)
#9 0x7efc52f571f8 in QComboBox::currentIndexChanged(int) (/usr/lib/x86_64-linux-gnu/libQt5Widgets.so.5+0x275f8)
#10 0x7efc52f571f2 in QComboBox::setCurrentIndex(int) (/usr/lib/x86_64-linux-gnu/libQt5Widgets.so.5+0x275f2)
#11 0x7efc52f571f0 in QComboBox::currentIndexChanged(int) (/usr/lib/x86_64-linux-gnu/libQt5Widgets.so.5+0x275f0)
#12 0x7efc52f571f0 in QComboBox::currentIndexChanged(int) (/usr/lib/x86_64-linux-gnu/libQt5Widgets.so.5+0x275f0)
#13 0x7efc52f571f0 in QComboBox::currentIndexChanged(int) (/usr/lib/x86_64-linux-gnu/libQt5Widgets.so.5+0x275f0)
#14 0x7efc52f571f0 in QComboBox::currentIndexChanged(int) (/usr/lib/x86_64-linux-gnu/libQt5Widgets.so.5+0x275f0)
#15 0x7efc52f571f0 in QComboBox::currentIndexChanged(int) (/usr/lib/x86_64-linux-gnu/libQt5Widgets.so.5+0x275f0)
#16 0x7efc52f571f0 in QComboBox::currentIndexChanged(int) (/usr/lib/x86_64-linux-gnu/libQt5Widgets.so.5+0x275f0)
#17 0x7efc52f571f0 in QComboBox::currentIndexChanged(int) (/usr/lib/x86_64-linux-gnu/libQt5Widgets.so.5+0x275f0)
#18 0x7efc52f571f0 in QComboBox::currentIndexChanged(int) (/usr/lib/x86_64-linux-gnu/libQt5Widgets.so.5+0x275f0)
#19 0x7efc52f571f0 in QComboBox::currentIndexChanged(int) (/usr/lib/x86_64-linux-gnu/libQt5Widgets.so.5+0x275f0)
#20 0x7efc52f571f0 in QComboBox::currentIndexChanged(int) (/usr/lib/x86_64-linux-gnu/libQt5Widgets.so.5+0x275f0)

2021-08-19 1/10
Shadow bytes around the buggy address:
0x0c06804ee280: fd fd fd fd fa 00 00 00 00 fa 00 00 07 fa
0x0c06804ee290: fa fd fd fd fa fd fd fa fd fd fa fd fd fa
0x0c06804ee2a0: fd fd fa 00 00 00 00 fa fd fd fd fa fd fd fa
0x0c06804ee2b0: 00 00 00 00 fa 00 00 00 00 fa fa fa fa
0x0c06804ee2c0: fa fd fd fa fd fd fa fd fd fd fa fd fd fa
=>0x0c06804ee2d0: fd fd fa fd fd fd [fd]fa fd fd fd fa
0x0c06804ee2e0: fd fd fd fd fd fd fa fd fd fd fa fd fd fa
0x0c06804ee2f0: fd fd fd fd fd fd fd fd fd fd fd fd fd fd
0x0c06804ee300: fd fd fd fd fd fd fd fd fd fd fd fd fd fd
0x0c06804ee310: fd fd fd fd fd fd fa fa fa fa fa fa 00 00 05 fa
0x0c06804ee320: fa fd fd fa fa fa 00 00 00 00 fa fa fd fd
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
Fully redzone:
2021-08-19
Stack left redzone: f1
Stack mid redzone: f2
Stack right redzone: f3
Stack after return: f5
Stack use after scope: f7
Global redzone: f9
Global init order: f6
Poisoned by user: ac
Container overflow: fc
Intra object redzone: bb
ASan internal: fe
Left alloca redzone: ca
Right alloca redzone: cb
Shadow gap: cc

Related issues:
Related to Freeciv - Bug #859139: Qt: Temporary QByteArrays often used after ...

History

#1 - 2020-02-02 09:20 PM - Marko Lindqvist
Which commit this was with? gui-qt/dialogs.cpp has changed several times recently (within a week) and I suspect those line numbers in your trace are not the same as in current S2_6 HEAD.

#2 - 2020-02-02 09:37 PM - Marko Lindqvist
Chippo Elder wrote:

Load a savegame, do a turn or two, start a New Game with a new ruleset, play a turn. Then when it's primed, Leave Game, Start New Game, don't touch anything (ruleset says default) except you click Pick Nation, and it dumps (which is quite pretty, with colours and all).

Do you use "Pick Nation" when starting the earlier games, or do you go by the random nation selection that is the default?

#3 - 2020-02-03 12:36 AM - Chippo Elder
Marko Lindqvist wrote:

Which commit this was with?

The dump I'm going to show you below is from 2.6.1+ (modified ec0f40dcb), gui-qt client.

Marko Lindqvist wrote:

Do you use "Pick Nation" when starting the earlier games, or do you go by the random nation selection that is the default?

The latter. This is the first time during this program run that you touch the Pick Nation button.

So I followed my own instructions above and get this:

chippo@chippo-Aspire-V3-731:~$ ASAN_OPTIONS="abort_on_error=1:disable_coredump=0:unmap_shadow_on_exit=1" freeciv-qt-26
2: Loading tileset "amplio2".
2: Loading tileset "delta2".
QSocketNotifier: Invalid socket 41 and type 'Read', disabling...
2: Loading tileset "amplio2".
2: Loading tileset "delta2".
QSocketNotifier: Invalid socket 41 and type 'Read', disabling...
2: Loading tileset "delta2".
2: Loading tileset "amplio2".
2: Loading tileset "delta2".
2: Loading tileset "delta2".
=================================================================
==28888==ERROR: AddressSanitizer: heap-use-after-free on address 0x6030002b8598 thread T0
#0 0x9758f3 in skip_intl_qualifier_prefix /home/chippo/Downloads/git_clones/freeciv/utility/fcintl.c:48:7
#1 0x1082f3cc in nation_set_by_rule_name /home/chippo/Downloads/git_clones/freeciv/common/nation.c:766:23
#2 0x719750 in races_dialog::nationset_changed(int) /home/chippo/Downloads/git_clones/freeciv/client/gui-qt/dialogs.cpp:984:10
#3 0x95052450 in races_dialog::qt_static_metacall(QObject*, QMetaObject::Call, int, void**) /home/chippo/Downloads/git_clones/freeciv/client/gui-qt/meta_dialogs.cpp:369:21

2021-08-19
Shadow byte around the buggy address:
0x0608040060: fa fa fa fa fa fa fa fd fd fd fa fa fa
0x0608040070: fa fa fa fa fa fa fa fa fa fa fa fa fa
0x0608040080: fa fa fa fa fa fa fa fa fa fa fa fa fa
0x0608040090: fa 00 00 00 00 00 00 fa fa fa fa fa fa
0x06080400a0: fd fd fa fa fd fd fa fa fa fa fa fa
0x06080400b0: =0x06080400b0: fd fd fd fd fd fa fa fa fa fa fa
0x06080400c0: fa 00 00 00 00 00 00 fa fa fa fa fa fa
0x06080400d0: fa fa fa fa fa fa fa fa fa fa fa fa fa
0x06080400e0: 00 00 00 00 fa fa fa fa fa fa fa fa
0x06080400f0: fa fa fa fa fa fa fa fa fa fa fa fa fa
Shadow byte legend (one shadow byte represents 8 application bytes):
Addressable: 0
Partially addressable: 0
2021-08-19
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 @==28888==ABORTING
AddressSanitizer:DEADLYSIGNAL
Aborted (core dumped)

And since I worked out my ASAN core dump troubles, I can get a full backtrace:

Core was generated by `freeciv-qt-26'.
Program terminated with signal SIGABRT, Aborted.
#0  __GI_raise (sig=sig@entry=6) at ../sysdeps/unix/sysv/linux/raise.c:50
   ../sysdeps/unix/sysv/linux/raise.c: No such file or directory.
   [Current thread is 1 (Thread 0x7f1a4e3baac0 (LWP 28888))]
   (gdb) bt full
   #0  __GI_raise (sig=sig@entry=6) at ../sysdeps/unix/sysv/linux/raise.c:50
   set = {val = {0, 5106555, 895, 0, 0, 281470681751461, 0, 0, 0, 0, 0, 0, 0, 206158430216, 140737362021568}}
   pid = <optimized out>
   tid = <optimized out>
   #1 0x000007f1a5301c899 in __GI_abort () at abort.c:79
   save_stage = 1
   act = {__sigaction_handler = {sa_handler = 0x3fe6b5ce50b7821a, sa_sigaction = 0x3fe6b5ce50b7821a}, sa_mask = {__val = {0, 1397516071578154, 13731440, 2, 2, 0, 0, 0, 0, 3, 0, 0}}, sa_flags = 0, sa_restarter = 0xc85e68 <__asan::error_message_buf_mutex>}
   sigs = {{val = {32, 0 <repeats 15 times>}}
   #2 0x00000000004f8e27 in ()
   #3 0x00000000004f7801 in ()
   #4 0x00000000004df389 in ()
   #5 0x00000000004e0aff in __asan::ReportGenericError(unsigned long, unsigned long, unsigned long, unsigned long, bool, unsigned long, int, bool) ()
   #6 0x00000000004e11d8 in __asan_report_load1 ()
   #7 0x00000000009758f4 in skip_intl_qualifier_prefix (str=0x6030002b8598 "all") at fcintl.c:48
   ptr = <optimized out>
   #10 0x0000000000082f3cd in nation_set_by_rule_name (name=0x2 <error: Cannot access memory at address 0x2>) at nation.c:766
   rule_name = {static null = {<No data fields>}, d = 0x6030087bcb0}
   poption = 0x627000f7adb0
   rn = 0x6030002b8598 "all"
   #10 0x00000000000592451 in races_dialog::nationset_changed(int) (this=<optimized out>, index=<optimized out>) at dialogs.cpp:984
   rule_name = {static null = {<No data fields>}, d = 0x6030087bcb0}
   poption = 0x627000f7adb0
   #17 0x0000000000592451 in races_dialog::qt_static_metacall(QObject*, QMetaObject::Call, int, void**)
   (_o=0x2, _c=<optimized out>, _id=<optimized out>) at meta_dialogs.cpp:369
   _t = 0x2
   #11 0x000007f1a537d0468 in QMetaObject::activate(QObject*, int, int, void**) () at /usr/lib/x86_64-linux-gnu/libQt5Core.so.5
   #12 0x000007f1a5369e6c75 in QComboBox::currentIndexChanged(int) () at /usr/lib/x86_64-linux-gnu/libQt5Widgets.so.5
   #13 0x000007f1a53699076 in () at /usr/lib/x86_64-linux-gnu/libQt5Widgets.so.5
   #14 0x000007f1a5369b9bd in () at /usr/lib/x86_64-linux-gnu/libQt5Widgets.so.5
   #15 0x000007f1a5369b9bd in () at /usr/lib/x86_64-linux-gnu/libQt5Widgets.so.5
   #16 0x000007f1a536a1525 in () at /usr/lib/x86_64-linux-gnu/libQt5Widgets.so.5
   #17 0x000007f1a536a1525 in () at /usr/lib/x86_64-linux-gnu/libQt5Widgets.so.5
   #18 0x000007f1a5369de6d in QComboBoxPrivateContainer::itemSelected(QModelIndex const&) () at /usr/lib/x86_64-linux-gnu/libQt5Widgets.so.5
   #19 0x000007f1a536974fa in QComboBoxBoxPrivateContainer::eventFilter(QObject*, QEvent*) () at /usr/lib/x86_64-linux-gnu/libQt5Widgets.so.5
   #20 0x000007f1a536d447ab in QCOREAPPLICATIONPRIVATE::sendThroughObjectEventFilters(QObject*, QEvent*) () at /usr/lib/x86_64-linux-gnu/libQt5Widgets.so.5
   #21 0x000007f1a535b0a75 in QCOREAPPLICATIONPRIVATE::notify_helper(QObject*, QEvent*) () at /usr/lib/x86_64-linux-gnu/libQt5Widgets.so.5
   #22 0x000007f1a535b0a53 in QApplication::notify(QObject*, QEvent*) () at /usr/lib/x86_64-linux-gnu/libQt5Widgets.so.5
   #23 0x000007f1a5344a9a in QCOREAPPLICATION::notifyInternal2(QObject*, QEvent*) () at /usr/lib/x86_64-linux-gnu/libQt5Core.so.5
   #24 0x000007f1a53599157 in QCOREAPPLICATIONPRIVATE::sendMouseEvent(QWidget*, QMouseEvent*, QWindow*, QWindow*, QWindow*, QWindow**, QPointer<QWindow>>&), bool, bool) () at /usr/lib/x86_64-linux-gnu/libQt5Widgets.so.5
   #25 0x000007f1a535e2f44 in () at /usr/lib/x86_64-linux-gnu/libQt5Widgets.so.5
   #26 0x000007f1a535f16dc in () at /usr/lib/x86_64-linux-gnu/libQt5Widgets.so.5
   #27 0x000007f1a5359a86 in QCOREAPPLICATIONPRIVATE::notify_helper(QObject*, QEvent*) () at /usr/lib/x86_64-linux-gnu/libQt5Widgets.so.5
I still don’t fully understand what happens here. If we’re lucky, attached wild experiment helps, but I’m not surprised if it doesn’t.

If the problem is in the lifetime of the temporary QByteArray when using construct like 'char *buf = str.toLocal8Bit().data()', the bad news is that Qt-client and ruledit code has lots of such usage. And I now think it’s a bug in principle, even if this one is not about it.

grep -r "data()" client/qui-qt tools/ruledit

I still don’t fully understand what happens here. If we’re lucky, attached wild experiment helps, but I’m not surprised if it doesn’t.

If the problem is in the lifetime of the temporary QByteArray when using construct like 'char *buf = str.toLocal8Bit().data()', the bad news is that Qt-client and ruledit code has lots of such usage. And I now think it’s a bug in principle, even if this one is not about it.

grep -r "data()" client/qui-qt tools/ruledit

I still don’t fully understand what happens here. If we’re lucky, attached wild experiment helps, but I’m not surprised if it doesn’t.

I can't trigger it anymore. I tried a lot. The combination of the two patches (0007-Do-not-unqueue-mapview-updates-when-tileset-is-not-f.patch and QByteArrayLifetime.patch) has done the trick. Close the ticket.

In my efforts to discover why I no longer have a fixed-ghost-seen tree (and for simpler repeatability reasons), I dropped the QByteArrayLifetime.patch from my workflow. Now I'm being hit by use-after-free regularly. So I'm putting it back in.

This last one happened under quite different circumstances than earlier reported. I had been playing around in several savegames during that session in different rulesets etc. I had just started a New Game, had configured ruleset, players and difficulty. I had not gone into game options yet, 'cos that's where the Save is, and I want that ruleset to remember that I always want the Extended set of Nations. I was already in the choose-your-nation screen, fiddling with moving that window around on my screen and clicking all around on it, looking for I-dunno-what. Then I wanted to switch to Extended and clicked on the dropdown. It appeared in the correct place. I paused a good while, to take a screenshot, file it, and some other stuff. Then I clicked on Extended and the dump happened.

This BT looks very similar to the last, but there are some largish differences like the "set = {__val = {" values, so I'll include it here in case.
Core was generated by `freeciv-qt-26 -P'.
Program terminated with signal SIGABRT, Aborted.
#0  __GI_raise (sig=sig@entry=6) at ../sysdeps/unix/sysv/linux/raise.c:50
      ../sysdeps/unix/sysv/linux/raise.c: No such file or directory.
      [Current thread is 1 (Thread 0x7fd803b0eb80 (LWP 1282733))]
(gdb) bt full
#0  __GI_raise (sig=sig@entry=6) at ../sysdeps/unix/sysv/linux/raise.c:50
    set = 
      __val = {0, 281470681751461, 140724658498656, 140724658498656, 5200464, 20615843
           0216, 140724658499104, 5111467, 140724658499296, 14072465849992}
    pid = <optimized out>
    tid = <optimized out>
#1  0x000007f808e64899 in __GI_abort () at abort.c:79
    save_stage = 1
    act = 
      __sigaction_handler = {sa_handler = 0x400, sa_sigaction = 0x400}, sa_mask = {__val = {13850928, 140
           565878323744, 13850928, 2, 2, 140724658499064, 140565878323840, 4686603, 2, 0, 0, 0, 0, 0, 0}}, sa_flags = 0
      sa_restorer = 0xca3128 <__asan::error_message_buf_mutex>
    sigs = {__val = {32, 0 <repeats 15 times>}}
#2  0x00000000004fa157 in ()
#3  0x00000000004f8b31 in ()
#4  0x00000000004e06b9 in ()
#5  0x00000000004e1e2f in __asan::ReportGenericError(unsigned long, unsigned long, unsigned long, unsigned lon
      g, bool, unsigned long, unsigned int, bool) ()
#6  0x00000000004e2508 in __asan_report_load1 ()
#7  0x00000000009510a4 in skip_intl_qualifier_prefix (str=0x603000723058 "all") at fcintl.c:48
    ptr = <optimized out>
#8  0x00000000007fa6ed in nation_set_by_rule_name (name=0x2 <error: Cannot access memory at address 0x2>) at n
      ation.c:766
    qname = <optimized out>
#9  0x00000000007028531 in races_dialog::nationset_changed(int) (this=0x00000000007028531, index=<optimized out>) at dialogs.cpp:984
    rule_name = {static null = {<No data fields>}, d = 0x62700003fdb0}
    poption = 0x62700003fdb0
    rn = 0x603000723058 "all"
#10 0x0000000000593a11 in races_dialog::qt_static_metacall(QObject*, QMetaObject::Call, int, void**) at meta_dialogs.cpp:369
    _t = 0x2
#11 0x00007fd809ca8468 in QMetaObject::activate(QObject*, int, int, void**) at /usr/lib/x86_64-linux-gnu/libQt5Core.so.5
#12 0x00007fd8095cede6 in QComboBoxPrivateContainer::itemSelected(QModelIndex const&) at /usr/lib/x86_64-linux-gnu/libQt5Widgets.so.5
#13 0x00007fd8095cf4fa in QComboBoxPrivateContainer::eventFilter(QObject*, QEvent*) at /usr/lib/x86_64-linux-gnu/libQt5Widgets.so.5
#14 0x00007fd8095c9ed6 in QComboBox::currentIndexChanged(int) at /usr/lib/x86_64-linux-gnu/libQt5Widgets.so.5
#15 0x00007fd8095d5b9d in () at /usr/lib/x86_64-linux-gnu/libQt5Widgets.so.5
#16 0x00007fd8095d5b9d in () at /usr/lib/x86_64-linux-gnu/libQt5Widgets.so.5
#17 0x00007fd8095d5b9d in () at /usr/lib/x86_64-linux-gnu/libQt5Widgets.so.5
#18 0x00007fd8095d5b9d in () at /usr/lib/x86_64-linux-gnu/libQt5Widgets.so.5
#19 0x00007fd8095d5b9d in () at /usr/lib/x86_64-linux-gnu/libQt5Widgets.so.5
#20 0x00007fd8095d5b9d in () at /usr/lib/x86_64-linux-gnu/libQt5Widgets.so.5
--Type <RET> for more, q to quit, c to continue without paging--

2021-08-19 7/10
I betcha that with the QByteArrayLifetime.patch applied, I don't see this again.
I'm going to try the QByteArrayLifetime.patch and see if that helps.

#9 - 2020-02-06 04:20 PM - Chippo Elder
It applies cleanly in master and seems to do the job. I haven't managed to trigger this bug in master since applying it. I recommend that it gets committed.

#10 - 2020-02-07 07:07 PM - Marko Lindqvist
- Status changed from New to In Progress
- Assignee set to Marko Lindqvist
- Sprint/Milestone set to 2.6.2

Marko Lindqvist wrote:

... the lifetime of the temporary QByteArray when using construct like 'char *buf = str.toLocal8Bit().data()', the bad news is that Qt-client and ruledit code has lots of such usage.

-> Bug #859139

#11 - 2020-02-07 07:31 PM - Marko Lindqvist
- File 0008-Qt-Fix-heap-use-after-free-in-nations-dialog.patch added
- Status changed from In Progress to Resolved

Patch with commit message. No functional changes.

#12 - 2020-02-09 09:21 AM - Marko Lindqvist
- Status changed from Resolved to Closed

#13 - 2020-02-09 01:55 PM - Jacob Nevins
- Related to Bug #859139: Qt: Temporary QByteArrays often used after their lifetime has ended added

Files
<table>
<thead>
<tr>
<th>File Name</th>
<th>Size</th>
<th>Date</th>
<th>Author</th>
</tr>
</thead>
<tbody>
<tr>
<td>QByteArrayLifetime.patch</td>
<td>891 Bytes</td>
<td>2020-02-03</td>
<td>Marko Lindqvist</td>
</tr>
<tr>
<td>0008-Qt-Fix-heap-use-after-free-in-nations-dialog.patch</td>
<td>1.28 KB</td>
<td>2020-02-07</td>
<td>Marko Lindqvist</td>
</tr>
</tbody>
</table>