I was able to trigger an easily reproducible crash through the D-Bus interface. The org.atheme.audacious.SongTuple method does not valid the requested tuple in `do_song_tuple` before calling `get_value_type`. This triggers an assert in `get_value_type`, but could potentially still crash otherwise. Instead of a crash, I'd expect either an empty string or an error to be returned.

The crash can be reproduced with:

dbus-send --session --print-reply --dest=org.atheme.audacious /org/atheme/audacious org.atheme.audacious.SongTuple uint32:0 string:x

It crashes with the following backtrace:

```
#0 0x00007fff638afeb in raise () at /lib64/libc.so.6
#1 0x00007fff63755c1 in abort () at /lib64/libc.so.6
#2 0x00007fff6375491 in _ni_load_domain.cold.0 () at /lib64/libc.so.6
#3 0x00007fff6383752 in () at /lib64/libc.so.6
#4 0x00007fff7bb1d26 in Tuple::get_value_type(Tuple::Field) const (this=0x7fffffffc970, field=field@entry=Tuple::Invalid) at tuple.cc:450
#5 0x000055555555f66d in do_song_tuple(_ObjAudacious*, _DBusMethodInvocation*, unsigned int, char const*) (obj=0x55555557a4ce0, invoc=0x5555555cf360, pos=0, key=<optimized out>) at dbus-server.cc:644
#6 0x00007fff55ae90e3 in ffi_call_unix64 () at /lib64/libffi.so.6
#7 0x00007fff55a59dfb in ffi_call () at /lib64/libffi.so.6
#8 0x00007fff73795a5 in g_closure_marshal_generic () at /lib64/libgobject-2.0.so.0
#9 0x00007fff7378add in g_closure_invoke () at /lib64/libgobject-2.0.so.0
#10 0x00007fff738b43 in signal_emit_unlocked_R () at /lib64/libgobject-2.0.so.0
#11 0x00007fff739419f in g_signal_emitv () at /lib64/libgobject-2.0.so.0
#12 0x00007fff55555556a01a in _obj_audacious_skeleton_handle_method_call (connection=0x7fffd401f80, obj=0x7fffd40193a0 "SongTuple", parameters=<optimized out>, invocation=0x5555555cf360, user_data=0x55555557a4ce0) at aud-dbus.c:15178
#13 0x00007fff76970f6 in g_dbus_interface_method_dispatch_helper () at /lib64/libgio-2.0.so.0
#14 0x00007fff7676e50 in call_in_idle_ch () at /lib64/libgio-2.0.so.0
#15 0x00007fff709a1cb in _idle_dispatch () at /lib64/libgio-2.0.so.0
#16 0x00007fff7098d3d in g_main_context_dispatch () at /lib64/libgio-2.0.so.0
#17 0x00007fff709d8f7 in g_main_context_iterate.isra () at /lib64/libgio-2.0.so.0
#18 0x00007fff709da2f in g_main_loop_run () at /lib64/libgio-2.0.so.0
#19 0x00007fff632695f in gtk_main () at /lib64/libgtktk-x11-2.0.so.0
#20 0x00007fff7bcb65e in interface_run() () at interface.cc:163
#21 0x00007fff7baf6d in aud_run() () at runtime.cc:323
#22 0x00007fff632695f in gtk_main () at /lib64/libgtktk-x11-2.0.so.0
#23 0x00007fff632695f in __libc_start_main () at /lib64/libc.so.6
#24 0x00007fff55555555ed7a in __start () at main.cc:410
```
I'm running into this with Audacious 3.10-beta1 (from https://copr.fedorainfracloud.org/coprs/mschwendt/audacious-next/).

## History

### #1 - August 19, 2018 18:17 - John Lindgren
- % Done changed from 0 to 100
- Target version changed from 3.10 to 3.10.1
- Status changed from New to Closed
- Category set to core

Fixed:
https://github.com/audacious-media-player/audacious/commit/1528e780825b1fc49639ea16d332c5752051dab7

### #2 - August 19, 2018 18:19 - John Lindgren
- Affects version 3.10 added