

Re: dlopen(), atexit() crash on FreeBSD (testcase included)

Re: dlopen(), atexit() crash on FreeBSD (testcase included)

Source: <http://unix.derkeiler.com/Mailing-Lists/FreeBSD/hackers/2007-12/msg00289.html>

- *From:* Alexander Kabaev <kabaev@xxxxxxxxxx>
 - *Date:* Mon, 31 Dec 2007 14:26:20 -0500
-

On Mon, 31 Dec 2007 17:35:10 +0100

"Markus Hoenicka" <markus.hoenicka@xxxxxxxxxxxxxx> wrote:

Hi,

I've been redirected by Giorgos Keramidas to this list after reporting a problem on the freebsd-questions list. I'd greatly appreciate if you could have a look at the following problem. Apparently programs are doomed to segfault on FreeBSD if dlopen(ed) modules install exit handlers via atexit(). Similar problem reports have cropped up before, see e.g.

<http://www.imagemagick.org/pipermail/magick-developers/2006-March/002523.html>

My system runs:

```
FreeBSD yeti.mininet 6.1-RELEASE FreeBSD 6.1-RELEASE #1: Mon Aug 28
22:24:48 CEST 2006
markus@xxxxxxxxxxxxxx:/usr/src/sys/i386/compile/YETI i386
```

I'm one of the developers of libdbi, a database abstraction layer for C, see <http://libdbi.sourceforge.net>.

libdbi is a library for programs which are supposed to be able to access different database engines with a unified API. libdbi essentially maps generic API calls to the specific database client library calls of a particular database engine. To do this, libdbi loads available database drivers at runtime via dlopen() calls. Each of these drivers is linked against one database client library. E.g. the Firebird driver is linked against libfbclient.so. When libdbi is properly shut down, it unloads all loaded drivers by calling dlclose() on each of them.

This design works well on all supported platforms and with all supported database engines, with one exception: the Firebird driver on FreeBSD invariably causes a segfault when the application linked against libdbi exits:

Re: dlopen(), atexit() crash on FreeBSD (testcase included)

Re: dlopen(), atexit() crash on FreeBSD (testcase included)

```
#0 0x28514fe4 in ?? ()
#1 0x281507c3 in __cxa_finalize () from /lib/libc.so.6
#2 0x281503fe in exit () from /lib/libc.so.6
#3 0x0804a40f in main (argc=1, argv=0xbfbfe754) at test_dbi.c:419
```

The reason appears to be that the Firebird client libraries install exit handlers via atexit(). Remember that due to libdbi's design to load all available drivers whether or not they are used later, libdbi will cause a crash even if no Firebird database is accessed – it is sufficient that the driver has been loaded. As per Giorgos' suggestion it is simple to circumvent this segfault by avoiding the call to dlclose() before exiting, but I wonder whether there is a more robust solution for this problem.

The attached minimal testcase is sufficient to illustrate the problem. atexitmod.c defines a module which is loaded by datest.c Make sure to fix the hardcoded path in datest.c before building the app. To build the test program and watch it crash, do the following:

```
gcc -shared -o atexitmod.so atexitmod.c
gcc -o datest datest.c
./datest
```

Commenting out either the atexit() call in atexitmod.c or the dlclose() call in datest.c prevent the segfault.

If you find some solution, please cc me as I'm not subscribed to freebsd-hackers.

regards,
Markus

As designed. atexit should not be used by shared objects that do not expect themselves to live until actual exit() happens. ELF provides proper _init/_fini sections to support shared object initialization/destruction.

--

Alexander Kabaev

Attachment: signature.asc

Description: PGP signature