

A night with threads and gdb

Source: <http://unix.derkeiler.com/Mailing-Lists/FreeBSD/questions/2004-03/4030.html>

From: Andrea Venturoli (andrea.venturoli_at_netfence.it)

Date: 03/30/04

Date: Tue, 30 Mar 2004 17:42:41 +0100

To: freebsd-questions@freebsd.org

A night with threads and gdb

or

How I began to wonder whether 5.2.1 works
or thread support is really broken

It all started on Saturday 2004/3/27: the spring sun was shining hot and I was struggling in the effort to get apache working decently on a 5.2.1p3/i386 (more on this later). While portupgrading mod_php4, the system suddenly stopped working properly: no more "make install", no more "install", even "ls -l" would dump core!!! I wondered what could have caused this and thought that any changes to installed ports should not affect the stability of binaries from the base system; I tried moving /usr/local/lib out of the way and "ls -l" would work again. Logic or intuition lead me to blame nss_ldap, so I disabled it and everything would work fine again.

To make it clear: with nss_ldap enabled, everything that accessed the user database would crash: so "ls -l", "id" and so on (but not, e.g., "ls" without "-l").

I recompiled ls and libc with -ggdb3 and found out that the problem was in nsdispatch.c, and precisely in the last line of the following function:

```
nss_atexit(void)
{
    (void)_pthread_rwlock_wrlock(&nss_lock);
    vector_free((void **)&nsmmap, &nsmmapsize, sizeof(*_nsmmap),
        (vector_free_elem)ns_dbt_free);
    vector_free((void **)&nsmmod, &nsmmodsize, sizeof(*_nsmmod),
        (vector_free_elem)ns_mod_free);
    (void)_pthread_rwlock_unlock(&nss_lock);
}
```

freebsd-questions: A night with threads and gdb

Once again Google turned out to be man's best friend, by providing me the following link:

http://groups.google.it/groups?q=vector_free+nss_atexit&hl=it&lr=&ie=UTF-8&oe=UTF-8&selm=1080344625.821

Apart from the psychological help derived from knowing I'm not alone, this suggested to patch that file to look like:

```
nss_atexit(void)
{
    if (__isthreaded) (void)_pthread_rwlock_wrlock(&nss_lock);
    vector_free((void **)&_nsmmap, &_nsmmapsize, sizeof(*_nsmmap),
        (vector_free_elem)ns_dbt_free);
    vector_free((void **)&_nsmod, &_nsmodsize, sizeof(*_nsmod),
        (vector_free_elem)ns_mod_free);
    if (__isthreaded) (void)_pthread_rwlock_unlock(&nss_lock);
}
```

I did, and did similarly for other pthread calls in that file, declaring `__isthreaded` as:

```
extern int __isthreaded;
```

That was one step ahead: now `"ls -l /bin"` would crash no more, but `"ls -l /home"` would still be problematic. Obviously the difference between the two is that in `/bin` everything is owned by system accounts, while listing `/home` would imply searching for users in the ldap database.

I guessed the problem was that upgrading php had upgraded openldap too, so I looked at freshport and found out that the main difference was in the makefile, where `"-with-threads"` had been replaced with `"-with-threads=posix"`.

I decided to try the three alternatives:

- a) `-without-threads` would not do, as it would cause slapd to crash when `ldapsearching` with a filter (i.e. `"ldapsearch -b 'dc=mydomain'"` works fine, but `"ldapsearch -b 'dc=mydomain' (objectClass=posixAccount)"` not);
- b) `-with-threads=posix` would exhibit the above mentioned problem with `ls`;
- c) `-with-threads` would work best.

Now I could even `"ls -l /home"` and see the correct usernames. However, I could not login or su anymore. (This forced me to go and ask for the keys to the server room and wait until Sunday). I ended up finding out (again by 'gdb su') that now using `nss_ldap` hampers the ability of a process to read from stdin.

I can even provide this demonstrative program:

```
#include <stdio.h>
int main(int argc, char** argv)
{
    char ch;
```

```
getpwent();
while (1)
{
    ch=getchar();
    putchar(ch);
}
}
```

If I want it to work, I'll either need to comment the call to `getpwent()` or "ldap" in `/etc/nsswitch.conf`.
ktracing su showed "resource temporarily unavailable" when it tried to read from descriptor 0.
Also, telnetting to localhost:pop3 had qpopper say "I/O error".

Afternoon was over, darkness was coming and the machine had to be up again before morning, so I decided to leave
nss_ldap and migrate the user accounts to the system password files.
This will not do in the long run, since it prevents web management, but has allow several mail domains to be up again before any message was lost!
However, I was forced to increase the username length limit (`MAXLOGNAME` to 65 in `/usr/src/sys/sys/param` and `UT_NAMESIZE=64` in `utmp.h`). This is a deviation from a standard system which I'd like to avoid, but it is needed until the day I can get nss_ldap back up.

(Long base system recompile).

Now I had pop3 back up, time to think about smtp.

I tried recompiling `/usr/ports/mail/sendmail-ldap` but it hangs on `t-event` test, after the message:

```
./t-event
```

This test may hang. If there is no output within twelve seconds, abort it and recompile with `-DSM_CONF_SETITIMER=0`

I tried make `-DSM_CONF_SETITIMER=0`, but it makes no difference.
This test calls `sleep(1)` and program flow never gets out of it; if I use `gdb` and interrupt it, I see it's in `poll()`; if I single step into that function with `gdb`, it works fine, instead. Looks a lot like PR kern/56339, which is rather old (freebsd 4.8), but still open. I'm not sure however if it's really the same problem.

Being already a little suspicious on ldap I tried `/usr/ports/mail/sendmail` instead and it doesn't exhibit this problem. It fails however on the test about `shminit`, but the suggested workaround does its job. I'm not so sure it should be needed, anyway.

freebsd-questions: A night with threads and gdb

So, I also converted my sendmail maps to files and abandoned ldap completely for now.

Later on I realized that sendmail wasn't using authentication, so I deinstalled sendmail and installed sendmail-sasl, instead: no problem at all this time (!!!).

In the end, after a 40 km ride, a sleepless night, 20 consecutive hours of work and a couple pizzas, I finally managed to get my system up again, albeit with some more handicaps than before.

As for apache, I hoped removing LDAP from PHP would help, but unfortunately nothing has changed:

_ apache 1.3 will core dump on startup if php module mnogosearch is used (and I need it);

_ apache 2.0 with default prefork MPM will start, but will chew up all cpu time after a while; using "httpd -DSSL -X" shows that the server dies when nocc is used to forward a mail; no need to say that it's a problem with threads, the exact message being

Fatal error 'Unable to read from thread kernel pipe' at line 1100 in file /usr/src/lib/libc_r/uthread/uthread_kern.c (errno = 0)

I guess that when started up without -X, one process dies and the manager httpd will not cope correctly (and start eating up every cycle).

_ when using perchild MPM (and recompiling mod_php in a thread-safe manner) httpd doesn't die in the above case, but is very unstable anyway;

_ worker MPM seems to be the best, but, although no process dies, often apache will stop responding all the same; furthermore SSL is painfully slow, the difference with plain http being more than tenfold.

I have also verified that this same behaviour shows up on another 5.2.1 machine.

From all the above, there are only two possible conclusions I can draw: either there is something really obvious that I'm so blindly missing or the beast is very broken down to the bones!

This is at the same time my SOS to the world and an offer to provide the community with any small help I can give in improving this software's stability. If anyone has any hints, please tell me, and if anyone wants core dumps, ktraces or any other test result just ask!

Please, HELP!!!

A night with threads and gdb

freebsd-questions: A night with threads and gdb

bye

av.

Ceterum censeo SpamCop delendum esse

freebsd-questions@freebsd.org mailing list

<http://lists.freebsd.org/mailman/listinfo/freebsd-questions>

To unsubscribe, send any mail to "freebsd-questions-unsubscribe@freebsd.org"