Subject: Re: [lxc-dev] [BUG][cryo]: underflow in semundo_release() ?
Posted by serue on Fri, 13 Jun 2008 14:36:32 GMT
View Forum Message <> Reply to Message

Quoting sukadev@us.ibm.com (sukadev@us.ibm.com):
> Serge E. Hallyn [serue@us.ibm.com] wrote:
> | Quoting sukadev@us.ibm.com (sukadev@us.ibm.com):
> | > Serge E. Hallyn [serue@us.ibm.com] wrote:
> | > | > The last few messages on stdout of the restart are:
> | > | >
> | > | > DEBUG (cr.c::1141) next memseg_t is: start bfe5c000 end bfe71000 prot 3 flag 50 offset 3221139456 fnam [stack]
> | > | > DEBUG (cr.c::1187) Delete segment bfa35000 - bfa4a000
> | > | > DEBUG (cr.c::1189) Restore segment bfe5c000 - bfe71000
> | > |
> | > | The stack segments are not the same. How can that be? Did you turn off
> | > | stack randomization?
> | >
> | > Grr, I did not. After I randomized, it seems to work on -lxc4 as well.
> | >
> | > Rather than warn, can we have cryo fail if stack is not randomized ?
> | > (its almost sure to fail anyway).
> |
> | I guess we may as well, because I guess the error message shows up at
> | the top of a long list of output so you'll never see it.
> |
> | Will change it.
>
> I run into this bug (twice so far) even with randomize_va_space set to 0.
>
> I run the attached pipe2.c program, ckpt and restart. The restart started
> out fine (printed "i is 10") then I hit CTRL-C.
>
> Last commit in my git tree:
>
> commit 84d005031a8a17bdca62dc541c296a3bea74658c
> (which adds 'exit(1)' to Dave's following commit)
>   96bb0ed3351c2e4268dade4416e1acbff7dab152
>
> qemu login: BUG: atomic counter underflow at:
>
> BUG: atomic counter underflow at:
> Pid: 2252, comm: pipe2 Not tainted 2.6.26-rc2-mm1-lxc4 #2
> Pid: 2252, comm: pipe2 Not tainted 2.6.26-rc2-mm1-lxc4 #2
>   [<c01d9d7b>] [<c01d9d7b>] semundo_release+0x28/0x4a
> semundo_release+0x28/0x4a
>   [<c01514c5>] [<c01514c5>] __fput+0x93/0x13b
> __fput+0x93/0x13b

> [<c0151827>] [<c0151827>] fput+0x2d/0x32
> fput+0x2d/0x32
> [<c014f044>] [<c014f044>] filp_close+0x50/0x5a
> filp_close+0x50/0x5a
> [<c01146e0>] [<c01146e0>] put_files_struct+0x7c/0xbe
> put_files_struct+0x7c/0xbe
> [<c0114759>] [<c0114759>] exit_files+0x37/0x3c
> exit_files+0x37/0x3c
> [<c01156ec>] [<c01156ec>] do_exit+0x1e4/0x589
> do_exit+0x1e4/0x589
> [<c0115aef>] [<c0115aef>] do_group_exit+0x5e/0x86
> do_group_exit+0x5e/0x86
> [<c011d126>] [<c011d126>] get_signal_to_deliver+0x2e0/0x31e
> get_signal_to_deliver+0x2e0/0x31e
> [<c0102215>] [<c0102215>] do_notify_resume+0x91/0x6dd
> do_notify_resume+0x91/0x6dd
> [<c0126d31>] [<c0126d31>] ? ? getnstimeofday+0x37/0xb7
> getnstimeofday+0x37/0xb7
> [<c01f63a8>] [<c01f63a8>] ? ? copy_to_user+0x2a/0x36
> copy_to_user+0x2a/0x36
> [<c012490d>] [<c012490d>] ? ? update_rmtp+0x49/0x5b
> update_rmtp+0x49/0x5b
> [<c0124d0d>] [<c0124d0d>] ? ? hrtimer_nanosleep+0x57/0x95
> hrtimer_nanosleep+0x57/0x95
> [<c012491f>] [<c012491f>] ? ? hrtimer_wakeup+0x0/0x1c
> hrtimer_wakeup+0x0/0x1c
> [<c0124d8d>] [<c0124d8d>] ? ? sys_nanosleep+0x42/0x53
> sys_nanosleep+0x42/0x53
> [<c0102c16>] [<c0102c16>] work_notifysig+0x13/0x19
> work_notifysig+0x13/0x19
> [<c02f0000>] [<c02f0000>] ? ? serial_pci_guess_board+0xb0/0x141
> serial_pci_guess_board+0xb0/0x141

Suka,

yeah I get this with that kernel too (though mine looks very different).
But I don't with a more recent -mm. Certainly seems like either a
exit-vs-signal or exit-vs-semundo bug. So there are three
possibilities:

1. it was a -mm bug which Oleg squashed - in which case it's
   fixed.
2. it was a semundo bug which Manfred fixed with his recent
   semundo rcu-ification
2. it was a Pierre bug with the semundo-rcu for c/r. In which
   case, Nadia is having to rework that set anyway on top of
   Manfred's patch. So let's make sure to do this test when
   Kathy releases the next -lxc with Nadia's new version of

the semundo-rcu patchset.

thanks,
-serge

_____
Containers mailing list
Containers@lists.linux-foundation.org
https://lists.linux-foundation.org/mailman/listinfo/containers