
Subject: Re: [PATCH v2] SUNRPC: check current nsproxy before set of node name on client creation

Posted by [Myklebust, Trond](#) on Thu, 13 Sep 2012 13:30:25 GMT

[View Forum Message](#) <> [Reply to Message](#)

On Thu, 2012-09-13 at 16:11 +0400, Stanislav Kinsbursky wrote:

> > On Mon, 2012-09-10 at 19:37 +0400, Stanislav Kinsbursky wrote:

> >> Hi, Trond.

> >> So, if I understand you right, we can create rpc client (or increase usage

> >> counter) on NSMPROC_MON call and destroy (or decrease usage counter) on

> >> NSMPROC_UNMON call.

> >> Will this solution works?

> >

> > The rpc client(s) will need to be per-net-namespace, which complicates

> > matters a little bit, but yes, creation at NSMPROC_MON, and destruction

> > at NSMPROC_UNMON should work.

> >

>

> Hi, Trond.

> With reference-counted NSM client I got this:

>

> BUG: unable to handle kernel NULL pointer dereference at 0000000000000008

> IP: [<fffffffa00c0d7f>] encode_mon_id+0x2e/0x64 [lockd]

> PGD 0

> Oops: 0000 [#1] SMP DEBUG_PAGEALLOC

> Modules linked in: nfsv3 nfs_acl nfs lockd veth sunrpc bridge stp llc i2c_piix4

> i2c_core virtio_blk virtio_net floppy virtio_pci virtio_ring virtio

> CPU 1

> Pid: 1174, comm: bash Not tainted 3.5.0-kitchensink+ #44 Bochs Bochs

> RIP: 0010:[<fffffffa00c0d7f>] [<fffffffa00c0d7f>] encode_mon_id+0x2e/0x64 [lockd]

> RSP: 0018:ffff88001d0f1a08 EFLAGS: 00010246

> RAX: 0000000000000000 RBX: ffff88001d0f1c38 RCX: 0000000000000000

> RDX: ffff88001c85803f RSI: ffff88001d23b204 RDI: ffff88001d0f1a48

> RBP: ffff88001d0f1a18 R08: ffff88001c858040 R09: 0000000000000003

> R10: ffff88001a5aba10 R11: ffff88001d0f1ac8 R12: ffff88001d0f1a48

> R13: ffff88001a8a3138 R14: ffffffffa008d580 R15: ffffffffa00c0db5

> FS: 00007f0d60eea700(0000) GS:ffff88001f700000(0000) knlGS:0000000000000000

> CS: 0010 DS: 0000 ES: 0000 CR0: 000000008005003b

> CR2: 0000000000000008 CR3: 000000001db3d000 CR4: 000000000000006e0

> DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000

> DR3: 0000000000000000 DR6: 00000000ffff0ff0 DR7: 00000000000000400

> Process bash (pid: 1174, threadinfo ffff88001d0f0000, task ffff88001d1f4160)

> Stack:

> ffff88001d0f1a48 ffff88001c858030 ffff88001d0f1a28 ffffffffa00c0dc9

> ffff88001d0f1ab8 ffffffffa00731a0 ffff88001a5aba10 ffff88001d0f1c38

> ffff88001c858040 ffff88001a8a3140 ffff88001c858854 ffff88001a8a3140

> Call Trace:

```

> [<fffffffa00c0dc9>] nsm_xdr_enc_unmon+0x14/0x16 [lockd]
> [<fffffffa00731a0>] rpcauth_wrap_req+0xa1/0xb2 [sunrpc]
> [<fffffffa006b83f>] ? xprt_prepare_transmit+0x83/0x93 [sunrpc]
> [<fffffffa0068e54>] call_transmit+0x1e4/0x263 [sunrpc]
> [<fffffffa00728e2>] __rpc_execute+0xe7/0x313 [sunrpc]
> [<fffffffa0068c70>] ? call_transmit_status+0xb8/0xb8 [sunrpc]
> [<ffffff81055ed9>] ? wake_up_bit+0x25/0x2a
> [<fffffffa0072be0>] rpc_execute+0x9d/0xa5 [sunrpc]
> [<fffffffa006a6ae>] rpc_run_task+0x7e/0x86 [sunrpc]
> [<fffffffa006a7a3>] rpc_call_sync+0x44/0x65 [sunrpc]
> [<fffffffa00c0883>] nsm_mon_unmon+0x81/0xad [lockd]
> [<fffffffa00c0931>] nsm_unmonitor+0x82/0x13a [lockd]
> [<fffffffa00bc251>] nlm_destroy_host_locked+0x93/0xc9 [lockd]
> [<fffffffa00bc33a>] nlmclnt_release_host+0xb3/0xc3 [lockd]
> [<fffffffa00ba09f>] nlmclnt_done+0x1a/0x26 [lockd]
> [<fffffffa00d583e>] nfs_destroy_server+0x24/0x26 [nfs]
> [<fffffffa00d5d85>] nfs_free_server+0xad/0x134 [nfs]
> [<fffffffa00dd5ff>] nfs_kill_super+0x22/0x26 [nfs]
> [<ffffff8112b278>] deactivate_locked_super+0x26/0x57
> [<ffffff8112bd89>] deactivate_super+0x45/0x4c
> [<ffffff811423ec>] mntput_no_expire+0x110/0x119
> [<ffffff8114241f>] mntput+0x2a/0x2c
> [<ffffff81142605>] release_mounts+0x72/0x84
> [<ffffff811427cf>] put_mnt_ns+0x6f/0x7e
> [<ffffff8105a3db>] free_nsproxy+0x1f/0x87
> [<ffffff8105a49f>] switch_task_namespaces+0x5c/0x65
> [<ffffff8105a4b8>] exit_task_namespaces+0x10/0x12
> [<ffffff8103c232>] do_exit+0x69b/0x84f
> [<ffffff8103c469>] do_group_exit+0x83/0xae
> [<ffffff8103c4ab>] sys_exit_group+0x17/0x1b
> [<ffffff814657e9>] system_call_fastpath+0x16/0x1b
> Code: e5 41 54 53 66 66 66 66 90 48 89 f3 49 89 fc 48 8b 76 18 e8 93 ff ff ff 4c
> 89 e7 65 48 8b 04 25 c0 b9 00 00 48 8b 80 00 05 00 00 <48> 8b 70 08 48 83 c6 45
> e8 73 ff ff ff 4c 89 e7 be 0c 00 00 00
> RIP [<fffffffa00c0d7f>] encode_mon_id+0x2e/0x64 [lockd]
>
>
> There are more places, where NFS and Lockd code dereference utsname().
> In XDR layer, for instance.
>
> So, probably, we have to tie NFS to utsns as well as to netns.
> Add one more ns to xprt? What do you think?
>

```

We've already saved the utsname in the rpc_client as clnt->cl_nodename.
All XDR users can be trivially converted to use that.

Cheers

Trond

--

Trond Myklebust
Linux NFS client maintainer

NetApp
Trond.Myklebust@netapp.com
www.netapp.com
