[RHEL6.1] rpc.mountd segfault on NFSv3 server after "umount" followed by "showmount -a" on client system

Issue

- rpc.mountd segfault with NFSv3 server when "umount" followed by "showmount -a" is done on client system
- rpc.mountd segfaults with a message similar to the following, indicating a segfault near the end of the stack:

Aug 7 19:27:22 rhel61f kernel: rpc.mountd[27257]: segfault at 7fff1ab57ff8 ip 00007f7776b596d1 sp 00007fff1ab58000 error 6 in libc-2.12.so[7f7776ad7000+187000]

- The following steps produce the rpc.mountd segfault:

1. Create an NFS export on test system.

2. On some remote system mount the NFS export with

mount -t nfs <ip of test system>:/srv/redhat-dvd -o nfsvers=3,nolock /mnt/nfs.

3. On remote system run showmount -a <ip of test system>

4. On remote system run umount /mnt/nfs

5. On remote system run showmount -a <ip of test system>

6. Segfault occurs

Environment

Red Hat Enterprise Linux 6.1

- nfs-utils-1.2.3-7.el6.x86_64.rpm

Resolution

Root Cause

- This is looking like an infinite loop of function calls, where the stack is exceeded, and a segfault occurs.

Aug 7 19:27:22 rhel61f kernel: rpc.mountd[27257]: segfault at 7fff1ab57ff8 ip 00007f7776b596d1 sp 00007fff1ab58000 error 6 in libc-2.12.so[7f7776ad7000+187000]

Aug 7 19:27:23 rhel61f abrt[27293]: saved core dump of pid 27257 (/usr/sbin/rpc.mountd) to /var/spool/abrt/ccpp-1312759642-27257.new/coredump (13230080 bytes)

Diagnostic Steps

- The issue seems similar to https://bugzilla.redhat.com/show_bug.cgi?id=669065 but that was reported against Fedora.
- Analysis of 'coredump' (attached 8/7/2011 9:37pm) indicates an infinite loop in the code.

<snip>

Loaded symbols for /lib64/ld-2.12.so

Core was generated by `rpc.mountd'.

Program terminated with signal 11, Segmentation fault.

#0 memmove (dest=0x7f7777c42dc8, src=0x7f7777c21cd0, len=12) at memmove.c:46

46 {

Missing separate debuginfos, use: debuginfo-install libblkid-2.17.2-12.el6.x86_64 libgssglue-0.1-11.el6.x86_64 libuuid-2.17.2-12.el6.x86_64 tcp_wrappers-libs-7.6-56.3.el6.x86_64

(gdb) list

41 rettype

42 memmove (a1, a2, len)

43 a1const void *a1;

44 a2const void *a2;

45 size_t len;

46 {

47 unsigned long int dstp = (long int) dest;

48 unsigned long int srcp = (long int) src;

49

50 /* This test makes the forward copying code be used whenever possible.

(gdb) disass memmove

```
Dump of assembler code for function memmove:
   0x00007f7776b596c0 <+0>:     push   %r13
   0x00007f7776b596c2 <+2>:     mov    %rdi,%rax
   0x00007f7776b596c5 <+5>:     sub    %rsi,%rax
   0x00007f7776b596c8 <+8>:     push   %r12
   0x00007f7776b596ca <+10>:    mov    %rdi,%r12
   0x00007f7776b596cd <+13>:    push   %rbp
   0x00007f7776b596ce <+14>:    mov    %rdi,%rbp
=> 0x00007f7776b596d1 <+17>:    push   %rbx
   0x00007f7776b596d2 <+18>:    mov    %rsi,%rbx
(gdb) info reg
rax            0x210f8     135416
rbx            0xc     12
rcx            0xd     13
rdx            0xc     12
rsi            0x7f7777c21cd0      140151087045840
rdi            0x7f7777c42dc8      140151087181256
rbp            0x7f7777c42dc8      0x7f7777c42dc8
rsp            0x7fff1ab58000      0x7fff1ab58000
r8             0x7f7777c21cd0      140151087045840
r9             0x0     0
r10            0x10    16
r11            0x246     582
r12            0x7f7777c42dc8      140151087181256
r13            0x7f7777c21cdc      140151087045852
r14            0x7fff1b756ba0      140733654068128
r15            0x7fff1b756c40      140733654068288
rip            0x7f7776b596d1      0x7f7776b596d1 <memmove+17>
eflags         0x10202     [ IF RF ]
cs             0x33    51
ss             0x2b    43
ds             0x0     0
```

es 0x0 0

fs 0x0 0

gs 0x0 0

(gdb) bt

#0 memmove (dest=0x7f7777c42dc8, src=0x7f7777c21cd0, len=12) at memmove.c:46

#1 0x00007f7776e83c98 in xdrrec_putbytes (xdrs=<value optimized out>, addr=<value optimized out>, len=<value optimized out>)

at /usr/include/bits/string3.h:59

#2 0x00007f7776e82eba in xdr_opaque (xdrs=0x7f7777c344a8, cp=<value optimized out>, cnt=<value optimized out>) at xdr.c:506

#3 0x00007f7776e832ab in xdr_string (xdrs=0x7f7777c344a8, cpp=0x7f7777c34760, maxsize=255) at xdr.c:709

#4 0x00007f77776e87ce in xdr_name (xdrs=<value optimized out>, objp=<value optimized out>) at mount_xdr.c:83

#5 0x00007f77776e88d9 in xdr_mountbody (xdrs=0x7f7777c344a8, objp=0x7f7777c34760) at mount_xdr.c:103

#6 0x00007f7776e845f0 in xdr_reference (xdrs=0x7f7777c344a8, pp=0x7f7777c34770, size=<value optimized out>,

proc=<value optimized out>) at xdr_reference.c:91

#7 0x00007f7776e84731 in xdr_pointer (xdrs=0x7f7777c344a8, objpp=0x7f7777c34770, obj_size=24,

xdr_obj=0x7f77776e88c0 <xdr_mountbody>) at xdr_reference.c:138

#8 0x00007f77776e87a5 in xdr_mountlist (xdrs=<value optimized out>, objp=<value optimized out>) at mount_xdr.c:93

#9 0x00007f77776e890c in xdr_mountbody (xdrs=0x7f7777c344a8, objp=0x7f7777c34760) at mount_xdr.c:107

#10 0x00007f7776e845f0 in xdr_reference (xdrs=0x7f7777c344a8, pp=0x7f7777c34770, size=<value optimized out>,

proc=<value optimized out>) at xdr_reference.c:91

#11 0x00007f7776e84731 in xdr_pointer (xdrs=0x7f7777c344a8, objpp=0x7f7777c34770, obj_size=24,

xdr_obj=0x7f77776e88c0 <xdr_mountbody>) at xdr_reference.c:138

#12 0x00007f77776e87a5 in xdr_mountlist (xdrs=<value optimized out>, objp=<value optimized out>) at mount_xdr.c:93

#13 0x00007f77776e890c in xdr_mountbody (xdrs=0x7f7777c344a8, objp=0x7f7777c34760) at mount_xdr.c:107

#14 0x00007f7776e845f0 in xdr_reference (xdrs=0x7f7777c344a8, pp=0x7f7777c34770, size=<value optimized out>,

proc=<value optimized out>) at xdr_reference.c:91

...

#731 0x00007f7776e84731 in xdr_pointer (xdrs=0x7f7777c344a8, objpp=0x7f7777c34770, obj_size=24,

xdr_obj=0x7f77776e88c0 <xdr_mountbody>) at xdr_reference.c:138

#732 0x00007f77776e87a5 in xdr_mountlist (xdrs=<value optimized out>, objp=<value optimized out>) at mount_xdr.c:93

#733 0x00007f77776e890c in xdr_mountbody (xdrs=0x7f7777c344a8, objp=0x7f7777c34760) at mount_xdr.c:107

#734 0x00007f7776e845f0 in xdr_reference (xdrs=0x7f7777c344a8, pp=0x7f7777c34770, size=<value optimized out>,

proc=<value optimized out>) at xdr_reference.c:91

#735 0x00007f7776e84731 in xdr_pointer (xdrs=0x7f7777c344a8, objpp=0x7f7777c34770, obj_size=24,

xdr_obj=0x7f77776e88c0 <xdr_mountbody>) at xdr_reference.c:138

#736 0x00007f77776e87a5 in xdr_mountlist (xdrs=<value optimized out>, objp=<value optimized out>) at mount_xdr.c:93

#737 0x00007f77776e890c in xdr_mountbody (xdrs=0x7f7777c344a8, objp=0x7f7777c34760) at mount_xdr.c:107

#738 0x00007f7776e845f0 in xdr_reference (xdrs=0x7f7777c344a8, pp=0x7f7777c34770, size=<value optimized out>,

proc=<value optimized out>) at xdr_reference.c:91

#739 0x00007f7776e84731 in xdr_pointer (xdrs=0x7f7777c344a8, objpp=0x7f7777c34770, obj_size=24,

xdr_obj=0x7f77776e88c0 <xdr_mountbody>) at xdr_reference.c:138