Subject: Re: [PATCH 0/4] Fix race between sk_filter reassign and sk_clone()
Posted by Olof Johansson on Fri, 19 Oct 2007 02:29:47 GMT
View Forum Message <> Reply to Message

On Wed, Oct 17, 2007 at 09:23:02PM -0700, David Miller wrote:
[...]
> > The same problem exists for detaching filter (SO_DETACH_FILTER).
> >
> > The proposed fix consists of 3 preparation patches and the fix itself.
> >
> > Signed-off-by: Pavel Emelyanov <xemul@openvz.org>
>
> Looks good, applied.
>
> Thanks for fixing this bug Pavel!

Looks like this might be causing problems, at least for me on ppc. This
happened during a normal boot, right around first interface config/dhcp
run..

```
cpu 0x0: Vector: 300 (Data Access) at [c00000000147b820]
    pc: c000000000435e5c: .sk_filter_delayed_uncharge+0x1c/0x60
    lr: c0000000004360d0: .sk_attach_filter+0x170/0x180
    sp: c00000000147baa0
   msr: 9000000000009032
   dar: 4
 dsisr: 40000000
  current = 0xc000000004780fa0
  paca    = 0xc000000000650480
    pid   = 1295, comm = dhclient3
0:mon> t
[c00000000147bb20] c0000000004360d0 .sk_attach_filter+0x170/0x180
[c00000000147bbd0] c000000000418988 .sock_setsockopt+0x788/0x7f0
[c00000000147bcb0] c000000000438a74 .compat_sys_setsockopt+0x4e4/0x5a0
[c00000000147bd90] c00000000043955c .compat_sys_socketcall+0x25c/0x2b0
[c00000000147be30] c000000000007508 syscall_exit+0x0/0x40
--- Exception: c01 (System Call) at 000000000ff618d8
SP (fffdf040) is in userspace
0:mon>
```

I.e. null pointer deref at sk_filter_delayed_uncharge+0x1c:

```
0:mon> di $.sk_filter_delayed_uncharge
c000000000435e40  7c0802a6      mflr    r0
c000000000435e44  fbc1fff0      std     r30,-16(r1)
c000000000435e48  7c8b2378      mr      r11,r4
c000000000435e4c  ebc2cdd0      ld      r30,-12848(r2)
c000000000435e50  f8010010      std     r0,16(r1)
```

```
c000000000435e54  f821ff81      stdu    r1,-128(r1)
c000000000435e58  380300a4      addi    r0,r3,164
c000000000435e5c  81240004      lwz     r9,4(r4)
```

That's the deref of fp:

```
static void sk_filter_delayed_uncharge(struct sock *sk, struct sk_filter *fp)
{
        unsigned int size = sk_filter_len(fp);
...
```

That is called from sk_attach_filter():

```
...
        rcu_read_lock_bh();
        old_fp = rcu_dereference(sk->sk_filter);
        rcu_assign_pointer(sk->sk_filter, fp);
        rcu_read_unlock_bh();

        sk_filter_delayed_uncharge(sk, old_fp);
        return 0;
...
```

So, looks like rcu_dereference() returned NULL. I don't know the
filter code at all, but it seems like it might be a valid case?
sk_detach_filter() seems to handle a NULL sk_filter, at least.

So, this needs review by someone who knows the filter, but it fixes the
problem for me:

Signed-off-by: Olof Johansson <olof@lixom.net>

```
diff --git a/net/core/filter.c b/net/core/filter.c
index 1f0068e..e0a0694 100644
--- a/net/core/filter.c
+++ b/net/core/filter.c
@@ -447,7 +447,8 @@ int sk_attach_filter(struct sock_fprog *fprog, struct sock *sk)
  rcu_assign_pointer(sk->sk_filter, fp);
  rcu_read_unlock_bh();

- sk_filter_delayed_uncharge(sk, old_fp);
+ if (old_fp)
+  sk_filter_delayed_uncharge(sk, old_fp);
  return 0;
 }
```