Subject: [patch 2/2][NETNS45][V3] remove timewait sockets at cleanup
Posted by Daniel Lezcano on Fri, 28 Sep 2007 09:51:10 GMT
View Forum Message <> Reply to Message

From: Daniel Lezcano <dlezcano@fr.ibm.com>

Denis Lunev spotted that if we take a reference to the network namespace
with the timewait sockets, we will need to wait for their expiration to
have the network namespace freed. This is a waste of time, the timewait
sockets are for avoiding to receive a duplicate packet from the network,
if the network namespace is freed, the network stack is removed, so no
chance to receive any packets from the outside world.

This patchset remove/destroy the timewait sockets when the
network namespace is freed.

The exit method registered by netns_register_subsys is put in the tcp.c
file and not in inet_timewait_sock.c. The reasons are we browse the tcp
established hash table and I don't want to add references to tcp in inet
timewait sockets and, furthermore, dccp protocol uses the inet timewait
sock too. IMHO, if we status to cleanup dccp timewait too, we should add
a exit method in dccp file.

Signed-off-by: Daniel Lezcano <dlezcano@fr.ibm.com>
---
 net/ipv4/tcp.c | 41 ++
 1 file changed, 41 insertions(+)

Index: linux-2.6-netns/net/ipv4/tcp.c
===
--- linux-2.6-netns.orig/net/ipv4/tcp.c
+++ linux-2.6-netns/net/ipv4/tcp.c
@@ -2432,8 +2432,49 @@ static int tcp_net_init(struct net *net)
  return 0;
 }

+/*
+ * Wipeout tcp timewait sockets, they are no longer needed
+ * because we destroy the network namespace, so no risk to
+ * have duplicate packet coming from the network
+ */
+static void tcp_net_exit(struct net *net)
+{
+ struct inet_timewait_sock *tw;
+ struct sock *sk;
+ struct hlist_node *node;
+ int h;
+

```
+ local_bh_disable();
+
+       /* Browse the the established hash table */
+ for (h = 0; h < (tcp_hashinfo.ehash_size); h++) {
+  struct inet_ehash_bucket *head =
+   inet_ehash_bucket(&tcp_hashinfo, h);
+ restart:
+  write_lock(&head->lock);
+  sk_for_each(sk, node, &head->twchain) {
+
+   tw = inet_twsk(sk);
+   if (tw->tw_net != net)
+    continue;
+   sock_hold(sk);
+
+   write_unlock(&head->lock);
+
+   inet_twsk_deschedule(tw, &tcp_death_row);
+   inet_twsk_put(tw);
+
+   goto restart;
+  }
+  write_unlock(&head->lock);
+ }
+
+ local_bh_enable();
+}
+
 static struct pernet_operations tcp_net_ops = {
   .init = tcp_net_init,
+ .exit = tcp_net_exit,
 };

 void __init tcp_init(void)

--
```

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