Subject: [PATCH][SCTP]: IPv4 vs IPv6 addresses mess in sctp_inet[6]addr_event.
Posted by Pavel Emelianov on Wed, 09 Apr 2008 14:22:17 GMT
View Forum Message <> Reply to Message

All IP addresses that are present in a system are duplicated on
struct sctp_sockaddr_entry. They are linked in the global list
called sctp_local_addr_list. And this struct unions IPv4 and IPv6
addresses.

So, there can be rare case, when a sockaddr_in.sin_addr coincides
with the corresponding part of the sockaddr_in6 and the notifier
for IPv4 will carry away an IPv6 entry.

The fix is to check the family before comparing the addresses.

Signed-off-by: Pavel Emelyanov <xemul@openvz.org>

```
diff --git a/net/sctp/ipv6.c b/net/sctp/ipv6.c
index b1e05d7..85f1495 100644
--- a/net/sctp/ipv6.c
+++ b/net/sctp/ipv6.c
@@ -110,8 +110,9 @@ static int sctp_inet6addr_event(struct notifier_block *this, unsigned long
ev,
   spin_lock_bh(&sctp_local_addr_lock);
   list_for_each_entry_safe(addr, temp,
      &sctp_local_addr_list, list) {
-   if (ipv6_addr_equal(&addr->a.v6.sin6_addr,
-         &ifa->addr)) {
+   if (addr->a.sa.sa_family == AF_INET6 &&
+     ipv6_addr_equal(&addr->a.v6.sin6_addr,
+      &ifa->addr)) {
     found = 1;
     addr->valid = 0;
     list_del_rcu(&addr->list);
diff --git a/net/sctp/protocol.c b/net/sctp/protocol.c
index f90091a..c2dd65d 100644
--- a/net/sctp/protocol.c
+++ b/net/sctp/protocol.c
@@ -647,7 +647,9 @@ static int sctp_inetaddr_event(struct notifier_block *this, unsigned long
ev,
   spin_lock_bh(&sctp_local_addr_lock);
   list_for_each_entry_safe(addr, temp,
      &sctp_local_addr_list, list) {
-   if (addr->a.v4.sin_addr.s_addr == ifa->ifa_local) {
+   if (addr->a.sa.sa_family == AF_INET &&
+     addr->a.v4.sin_addr.s_addr ==
```

```
+     ifa->ifa_local) {
      found = 1;
      addr->valid = 0;
      list_del_rcu(&addr->list);
```