Subject: [patch 04/38][IPV6] inet6_addr - check ipv6 address per namespace
Posted by Daniel Lezcano on Mon, 03 Dec 2007 16:16:40 GMT
View Forum Message <> Reply to Message

When a new address is added, we must check if the new address does
not already exists.
This patch makes this check to be aware of a network namespace,
so the check will look if the address already exists for the specified
network namespace. While the addresses are browsed, the addresses
which do not belong to the namespace are discarded.

The modifications are trivial here and they do not touch the
callers of the function ipv6_chk_addr.

Signed-off-by: Daniel Lezcano <dlezcano@fr.ibm.com>
Signed-off-by: Benjamin Thery <benjamin.thery@bull.net>
---
 include/net/addrconf.h | 11 ++++++++++-
 net/ipv6/addrconf.c | 7 +++++--
 2 files changed, 15 insertions(+), 3 deletions(-)

Index: linux-2.6-netns/include/net/addrconf.h
===
--- linux-2.6-netns.orig/include/net/addrconf.h
+++ linux-2.6-netns/include/net/addrconf.h
@@ -58,9 +58,18 @@ extern int   addrconf_add_ifaddr(void __
 extern int   addrconf_del_ifaddr(void __user *arg);
 extern int   addrconf_set_dstaddr(void __user *arg);

-extern int   ipv6_chk_addr(struct in6_addr *addr,
+extern int   __ipv6_chk_addr(struct net *net,
+   struct in6_addr *addr,
           struct net_device *dev,
           int strict);
+
+static inline int ipv6_chk_addr(struct in6_addr *addr,
+   struct net_device *dev,
+   int strict)
+{
+ return __ipv6_chk_addr(&init_net, addr, dev, strict);
+}
+
 #if defined(CONFIG_IPV6_MIP6) || defined(CONFIG_IPV6_MIP6_MODULE)
 extern int   ipv6_chk_home_addr(struct in6_addr *addr);
 #endif
Index: linux-2.6-netns/net/ipv6/addrconf.c
===
--- linux-2.6-netns.orig/net/ipv6/addrconf.c

```
+++ linux-2.6-netns/net/ipv6/addrconf.c
@@ -1181,13 +1181,16 @@ static int ipv6_count_addresses(struct i
  return cnt;
 }

-int ipv6_chk_addr(struct in6_addr *addr, struct net_device *dev, int strict)
+int __ipv6_chk_addr(struct net *net, struct in6_addr *addr,
+     struct net_device *dev, int strict)
 {
  struct inet6_ifaddr * ifp;
  u8 hash = ipv6_addr_hash(addr);

  read_lock_bh(&addrconf_hash_lock);
  for(ifp = inet6_addr_lst[hash]; ifp; ifp=ifp->lst_next) {
+  if (ifp->idev->dev->nd_net != net)
+   continue;
    if (ipv6_addr_equal(&ifp->addr, addr) &&
        !(ifp->flags&IFA_F_TENTATIVE)) {
     if (dev == NULL || ifp->idev->dev == dev ||
@@ -1199,7 +1202,7 @@ int ipv6_chk_addr(struct in6_addr *addr,
  return ifp != NULL;
 }

-EXPORT_SYMBOL(ipv6_chk_addr);
+EXPORT_SYMBOL(__ipv6_chk_addr);

 static
 int ipv6_chk_same_addr(const struct in6_addr *addr, struct net_device *dev)

--
```


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