
Subject: [patch 09/12] net namespace : make loopback address always visible

Posted by [Daniel Lezcano](#) on Fri, 19 Jan 2007 15:47:23 GMT

[View Forum Message](#) <> [Reply to Message](#)

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

Add a specific condition when doing inet interface listing in order to see always the loopback address.

Signed-off-by: Daniel Lezcano <dlezcano@fr.ibm.com>

include/linux/net_namespace.h | 9 ++++++++
net/core/net_namespace.c | 22 ++++++++
net/ipv4/devinet.c | 12 +++++-----
3 files changed, 36 insertions(+), 7 deletions(-)

Index: 2.6.20-rc4-mm1/net/ipv4/devinet.c

```
=====
--- 2.6.20-rc4-mm1.orig/net/ipv4/devinet.c
+++ 2.6.20-rc4-mm1/net/ipv4/devinet.c
@@ -695,8 +695,7 @@
     for (ifap = &in_dev->ifa_list; (ifa = *ifap) != NULL;
         ifap = &ifa->ifa_next) {
     if (!strcmp(ifr.ifr_name, ifa->ifa_label) &&
-        net_ns_match(ifa->ifa_net_ns,
-        current_net_ns) &&
+        net_ns_ifa_is_visible(ifa) &&
        sin_orig.sin_addr.s_addr ==
        ifa->ifa_address) {
        break; /* found */
@@ -710,13 +709,12 @@
     for (ifap = &in_dev->ifa_list; (ifa = *ifap) != NULL;
         ifap = &ifa->ifa_next)
     if (!strcmp(ifr.ifr_name, ifa->ifa_label) &&
-        net_ns_match(ifa->ifa_net_ns,
-        current_net_ns))
+        net_ns_ifa_is_visible(ifa))
     break;
     }
 }

- if (ifa && !net_ns_match(ifa->ifa_net_ns, current_net_ns))
+ if (ifa && !net_ns_ifa_is_visible(ifa))
     goto done;

     ret = -EADDRNOTAVAIL;
@@ -868,7 +866,7 @@
```

```

goto out;

for (; ifa; ifa = ifa->ifa_next) {
- if (!net_ns_match(ifa->ifa_net_ns, current_net_ns))
+ if (!net_ns_ifa_is_visible(ifa))
    continue;
    if (!buf) {
        done += sizeof(ifr);
@@ -1216,7 +1214,7 @@

    for (ifa = in_dev->ifa_list, ip_idx = 0; ifa;
        ifa = ifa->ifa_next, ip_idx++) {
- if (!net_ns_match(ifa->ifa_net_ns, current_net_ns))
+ if (!net_ns_ifa_is_visible(ifa))
    continue;
    if (ip_idx < s_ip_idx)
        continue;
Index: 2.6.20-rc4-mm1/include/linux/net_namespace.h
=====
--- 2.6.20-rc4-mm1.orig/include/linux/net_namespace.h
+++ 2.6.20-rc4-mm1/include/linux/net_namespace.h
@@ -7,6 +7,8 @@
#include <linux/errno.h>
#include <linux/types.h>

+struct in_ifaddr;
+
struct net_namespace {
    struct kref kref;
    struct net_device *dev_base_p, **dev_tail_p;
@@ -101,6 +103,8 @@

extern struct net_namespace *net_ns_find_from_dest_addr(u32 daddr);

+extern int net_ns_ifa_is_visible(const struct in_ifaddr *ifa);
+
#define SELECT_SRC_ADDR net_ns_select_source_address

#else /* CONFIG_NET_NS */
@@ -174,6 +178,11 @@
    return NULL;
}

+static inline int net_ns_ifa_is_visible(const struct in_ifaddr *ifa)
+{
+ return 1;
+}
+

```

```

#define SELECT_SRC_ADDR inet_select_addr

#endif /* !CONFIG_NET_NS */
Index: 2.6.20-rc4-mm1/net/core/net_namespace.c
=====
--- 2.6.20-rc4-mm1.orig/net/core/net_namespace.c
+++ 2.6.20-rc4-mm1/net/core/net_namespace.c
@@ -420,4 +420,26 @@

    return net_ns;
}
+
+/*
+ * This function checks if the ifaddr is visible from the
+ * current network namespace. This is true if the ifaddr is
+ * the loopback address or if the ifaddr is owned by the network
+ * namespace.
+ * @ifa : the ifaddr
+ * Returns : 1 if visible, 0 otherwise
+ */
+int net_ns_ifa_is_visible(const struct in_ifaddr *ifa)
+{
+ struct net_namespace *net_ns = current_net_ns;
+
+ if (LOOPBACK(ifa->ifa_local))
+ return 1;
+
+ if (net_ns_match(ifa->ifa_net_ns, net_ns))
+ return 1;
+
+ return 0;
+}
+
#endif /* CONFIG_NET_NS */

--

```

Containers mailing list
Containers@lists.osdl.org
<https://lists.osdl.org/mailman/listinfo/containers>
