
Subject: [patch 09/20] [Network namespace] Isolate the inet device. ip and ifconfig commands will not show ip

Posted by [Daniel Lezcano](#) on Sun, 10 Dec 2006 21:58:26 GMT

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

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

```
include/linux/inetdevice.h | 4 ++++
net/ipv4/devinet.c          | 30 ++++++-----
2 files changed, 28 insertions(+), 6 deletions(-)
```

Index: 2.6.19-rc6-mm2/include/linux/inetdevice.h

=====

--- 2.6.19-rc6-mm2.orig/include/linux/inetdevice.h

+++ 2.6.19-rc6-mm2/include/linux/inetdevice.h

@@ -99,6 +99,7 @@ struct in_ifaddr

unsigned char ifa_flags;

unsigned char ifa_prefixlen;

char ifa_label[IFNAMSIZ];

+ struct net_namespace *ifa_net_ns;

};

extern int register_inetaddr_notifier(struct notifier_block *nb);

@@ -115,6 +116,9 @@ extern __be32 inet_confirm_addr(const s

extern struct in_ifaddr *inet_ifa_byprefix(struct in_device *in_dev, __be32 prefix, __be32 mask);

extern void inet_forward_change(void);

+extern void inet_del_ifa(struct in_device *in_dev, struct in_ifaddr **ifap, int destroy);

+extern void inet_free_ifa(struct in_ifaddr *ifa);

+

static __inline__ int inet_ifa_match(__be32 addr, struct in_ifaddr *ifa)

{

return !((addr^ifa->ifa_address)&ifa->ifa_mask);

Index: 2.6.19-rc6-mm2/net/ipv4/devinet.c

=====

--- 2.6.19-rc6-mm2.orig/net/ipv4/devinet.c

+++ 2.6.19-rc6-mm2/net/ipv4/devinet.c

@@ -54,6 +54,7 @@

#include <linux/notifier.h>

#include <linux/inetdevice.h>

#include <linux/igmp.h>

+#include <linux/net_namespace.h>

#ifdef CONFIG_SYSCTL

#include <linux/sysctl.h>

#endif

@@ -91,8 +92,6 @@ static struct nla_policy ifa_ipv4_policy

```

static void rtmsg_ifa(int event, struct in_ifaddr *, struct nlmsg_hdr *, u32);

static BLOCKING_NOTIFIER_HEAD(inetaddr_chain);
-static void inet_del_ifa(struct in_device *in_dev, struct in_ifaddr **ifap,
- int destroy);
#ifdef CONFIG_SYSCTL
static void devinet_sysctl_register(struct in_device *in_dev,
    struct ipv4_devconf *p);
@@ -120,7 +119,7 @@ static void inet_rcu_free_ifa(struct rcu
    kfree(ifa);
}

-static inline void inet_free_ifa(struct in_ifaddr *ifa)
+void inet_free_ifa(struct in_ifaddr *ifa)
{
    call_rcu(&ifa->rcu_head, inet_rcu_free_ifa);
}
@@ -268,6 +267,7 @@ static void __inet_del_ifa(struct in_dev

    if (!(ifa->ifa_flags & IFA_F_SECONDARY) ||
        ifa1->ifa_mask != ifa->ifa_mask ||
+    !net_ns_match(ifa->ifa_net_ns, ifa1->ifa_net_ns) ||
    !inet_ifa_match(ifa1->ifa_address, ifa)) {
        ifap1 = &ifa->ifa_next;
        prev_prom = ifa;
@@ -333,8 +333,8 @@ static void __inet_del_ifa(struct in_dev
    }
}

-static void inet_del_ifa(struct in_device *in_dev, struct in_ifaddr **ifap,
- int destroy)
+void inet_del_ifa(struct in_device *in_dev, struct in_ifaddr **ifap,
+ int destroy)
{
    __inet_del_ifa(in_dev, ifap, destroy, NULL, 0);
}
@@ -470,6 +470,9 @@ static int inet_rtm_deladdr(struct sk_buff

    for (ifap = &in_dev->ifa_list; (ifa = *ifap) != NULL;
        ifap = &ifa->ifa_next) {
+    if (!net_ns_match(ifa->ifa_net_ns, current_net_ns))
+        continue;
+
    if (tb[IFA_LOCAL] &&
        ifa->ifa_local != nla_get_be32(tb[IFA_LOCAL]))
        continue;
@@ -543,6 +546,7 @@ static struct in_ifaddr *rtm_to_ifaddr(s
    ifa->ifa_flags = ifm->ifa_flags;

```

```

    ifa->ifa_scope = ifm->ifa_scope;
    ifa->ifa_dev = in_dev;
+ ifa->ifa_net_ns = current_net_ns;

    ifa->ifa_local = nla_get_be32(tb[IFA_LOCAL]);
    ifa->ifa_address = nla_get_be32(tb[IFA_ADDRESS]);
@@ -688,6 +692,8 @@ int devinet_ioctl(unsigned int cmd, void
    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) &&
        sin_orig.sin_addr.s_addr ==
        ifa->ifa_address) {
            break; /* found */
@@ -700,11 +706,16 @@ int devinet_ioctl(unsigned int cmd, void
    if (!ifa) {
        for (ifap = &in_dev->ifa_list; (ifa = *ifap) != NULL;
             ifap = &ifa->ifa_next)
-        if (!strcmp(ifr.ifr_name, ifa->ifa_label))
+        if (!strcmp(ifr.ifr_name, ifa->ifa_label) &&
+            net_ns_match(ifa->ifa_net_ns,
+            current_net_ns))
            break;
    }
}

+ if (ifa && !net_ns_match(ifa->ifa_net_ns, current_net_ns))
+ goto done;
+
    ret = -EADDRNOTAVAIL;
    if (!ifa && cmd != SIOCSIFADDR && cmd != SIOCSIFFLAGS)
        goto done;
@@ -748,6 +759,8 @@ int devinet_ioctl(unsigned int cmd, void
    ret = -ENOBUFS;
    if ((ifa = inet_alloc_ifa()) == NULL)
        break;
+
+ ifa->ifa_net_ns = current_net_ns;
    if (colon)
        memcpy(ifa->ifa_label, ifr.ifr_name, IFNAMSIZ);
    else
@@ -852,6 +865,8 @@ static int inet_gifconf(struct net_devic
    goto out;

    for (; ifa; ifa = ifa->ifa_next) {
+ if (!net_ns_match(ifa->ifa_net_ns, current_net_ns))
+ continue;

```

```

    if (!buf) {
        done += sizeof(ifr);
        continue;
@@ -1085,6 +1100,7 @@ static int inetdev_event(struct notifier
    in_dev_hold(in_dev);
    ifa->ifa_dev = in_dev;
    ifa->ifa_scope = RT_SCOPE_HOST;
+   ifa->ifa_net_ns = current_net_ns;
    memcpy(ifa->ifa_label, dev->name, IFNAMSIZ);
    inet_insert_ifa(ifa);
    }
@@ -1197,6 +1213,8 @@ static int inet_dump_ifaddr(struct sk_buff

    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))
+   continue;
    if (ip_idx < s_ip_idx)
        continue;
    if (inet_fill_ifaddr(skb, ifa, NETLINK_CB(cb->skb).pid,

```

--

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