
Subject: [patch 10/20] [Network namespace] ioctl to push ifa to net_ns l3

Posted by [Daniel Lezcano](#) on Sun, 10 Dec 2006 21:58:27 GMT

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

New ioctl to "push" ifaddr to a container. Actually, the push is done from the current namespace, so the right word is "pull". That will be changed to move ifaddr from l2 network namespace to l3.

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

```
include/linux/net_namespace.h | 4 +
include/linux/sockios.h      | 4 +
net/core/net_namespace.c     | 97 +++++
net/ipv4/af_inet.c           | 4 +
4 files changed, 108 insertions(+), 1 deletion(-)
```

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

=====

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

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

@@ -122,6 +122,10 @@

#define SIOCBRADDIF 0x89a2 /* add interface to bridge */

#define SIOCBRDELIF 0x89a3 /* remove interface from bridge */

/* Container calls */

+#define SIOCNETNSPUSHIF 0x89b0 /* add ifaddr to namespace */

+#define SIOCNETNSPULLIF 0x89b1 /* remove ifaddr to namespace */

+

/* Device private ioctl calls */

/*

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

=====

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

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

@@ -789,6 +789,10 @@ int inet_ioctl(struct socket *sock, unsi

case SIOCSIFFLAGS:

err = devinet_ioctl(cmd, (void __user *)arg);

break;

+ case SIOCNETNSPUSHIF:

+ case SIOCNETNSPULLIF:

+ err = net_ns_ioctl(cmd, (void __user *)arg);

+ break;

default:

if (sk->sk_prot->ioctl)

err = sk->sk_prot->ioctl(sk, cmd, arg);

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

=====

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

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

@@ -86,6 +86,8 @@ extern struct net_namespace *find_net_ns

extern int hlist_dev_name_init(struct net_namespace *net_ns);

+extern int net_ns_ioctl(unsigned int cmd, void __user *arg);

+

#else /* CONFIG_NET_NS */

#define INIT_NET_NS(net_ns)

@@ -134,7 +136,7 @@ static inline struct net_namespace *find

static inline int net_ns_ioctl(unsigned int cmd, void __user *arg)

{

- return 0;

+ return -ENOSYS;

}

#endif /* !CONFIG_NET_NS */

Index: 2.6.19-rc6-mm2/net/core/net_namespace.c

=====

--- 2.6.19-rc6-mm2.orig/net/core/net_namespace.c

+++ 2.6.19-rc6-mm2/net/core/net_namespace.c

@@ -153,6 +153,28 @@ int copy_net_ns(int flags, struct task_s

return err;

}

+static void release_ifa_to_parent(const struct net_namespace* net_ns)

+{

+ struct net_device *dev;

+ struct in_device *in_dev;

+

+ read_lock(&dev_base_lock);

+ rcu_read_lock();

+ for (dev = dev_base; dev; dev = dev->next) {

+ in_dev = __in_dev_get_rcu(dev);

+ if (!in_dev)

+ continue;

+

+ for_ifa(in_dev) {

+ if (ifa->ifa_net_ns != net_ns)

+ continue;

+ ifa->ifa_net_ns = net_ns->parent;

+ } endfor_ifa(in_dev);

+ }

```

+ read_unlock(&dev_base_lock);
+ rcu_read_unlock();
+}
+
void free_net_ns(struct kref *kref)
{
    struct net_namespace *ns;
@@ -175,6 +197,10 @@ void free_net_ns(struct kref *kref)
}
}

+ if (ns->level == NET_NS_LEVEL3) {
+ release_ifa_to_parent(ns);
+ }
+
    printk(KERN_DEBUG "NET_NS: net namespace %p (%u) destroyed\n",
           ns, ns->id);

@@ -183,4 +209,75 @@ void free_net_ns(struct kref *kref)
/* because of put_net_ns() */
EXPORT_SYMBOL(free_net_ns);

+int net_ns_ioctl(unsigned int cmd, void __user *arg)
+{
+ struct ifreq ifr;
+ struct sockaddr_in *sin = (struct sockaddr_in *)&ifr.ifr_addr;
+ struct net_namespace *net_ns = current_net_ns;
+ struct net_device *dev;
+ struct in_device *in_dev;
+ struct in_ifaddr **ifap = NULL;
+ struct in_ifaddr *ifa = NULL;
+ char *colon;
+ int err;
+
+ if (!capable(CAP_NET_ADMIN))
+ return -EPERM;
+
+ if (net_ns->level != NET_NS_LEVEL3)
+ return -EPERM;
+
+ if (copy_from_user(&ifr, arg, sizeof(struct ifreq)))
+ return -EFAULT;
+
+ ifr.ifr_name[IFNAMSIZ - 1] = 0;
+
+ colon = strchr(ifr.ifr_name, ':');
+ if (colon)

```

```

+ *colon = 0;
+
+ rtnl_lock();
+
+ err = -ENODEV;
+ dev = __dev_get_by_name(ifr.ifr_name);
+ if (!dev)
+ goto out;
+
+ if (colon)
+ *colon = ':';
+
+ err = -EADDRNOTAVAIL;
+ in_dev = __in_dev_get_rtnl(dev);
+ if (!in_dev)
+ goto out;
+
+ for (ifap = &in_dev->ifa_list; (ifa = *ifap) != NULL;
+      ifap = &ifa->ifa_next)
+ if (!strcmp(ifr.ifr_name, ifa->ifa_label) &&
+     sin->sin_addr.s_addr == ifa->ifa_local)
+ break;
+ if (!ifa)
+ goto out;
+
+ err = -EINVAL;
+ switch(cmd) {
+
+ case SIOCNETHNSPUSHIF:
+ ifa->ifa_net_ns = net_ns;
+ break;
+
+ case SIOCNETHNSPULLIF:
+ ifa->ifa_net_ns = net_ns->parent;
+ break;
+ default:
+ goto out;
+ }
+
+ err = 0;
+out:
+ rtnl_unlock();
+ return err;
+}
+
+ #endif /* CONFIG_NET_NS */
+
--

```

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