
Subject: [patch 08/12] net namespace : find namespace by addr
Posted by [Daniel Lezcano](#) on Fri, 19 Jan 2007 15:47:22 GMT
[View Forum Message](#) <> [Reply to Message](#)

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

Switch to the the I3 namespace using the destination address.

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

```
include/linux/net_namespace.h | 7 ++++++
net/core/net_namespace.c      | 35 ++++++++++++++++++++++++++++++++++++++
net/ipv4/ip_input.c           | 16 ++++++++
3 files changed, 57 insertions(+), 1 deletion(-)
```

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

=====

--- 2.6.20-rc4-mm1.orig/net/ipv4/ip_input.c

+++ 2.6.20-rc4-mm1/net/ipv4/ip_input.c

@@ -374,6 +374,9 @@

```
{
    struct iphdr *iph;
    u32 len;
+ int err;
+ struct net_namespace *net_ns = current_net_ns;
+ struct net_namespace *dst_net_ns = NULL;
```

```
/* When the interface is in promisc. mode, drop all the crap
 * that it receives, do not try to analyse it.
```

@@ -393,6 +396,9 @@

```
iph = skb->nh.iph;
```

```
+ dst_net_ns = net_ns_find_from_dest_addr(iph->daddr);
+ if (dst_net_ns && !net_ns_match(net_ns, dst_net_ns))
+ push_net_ns(dst_net_ns);
/*
 * RFC1122: 3.1.2.2 MUST silently discard any IP frame that fails the checksum.
 *
```

@@ -431,10 +437,18 @@

```
/* Remove any debris in the socket control block */
memset(IPCB(skb), 0, sizeof(struct inet_skb_parm));
```

```
- return NF_HOOK(PF_INET, NF_IP_PRE_ROUTING, skb, dev, NULL,
+ err = NF_HOOK(PF_INET, NF_IP_PRE_ROUTING, skb, dev, NULL,
                ip_rcv_finish);
```

```

+ if (dst_net_ns && !net_ns_match(net_ns, dst_net_ns))
+ pop_net_ns(net_ns);
+
+ return err;
+
+ inhdr_error:
+ if (dst_net_ns && !net_ns_match(net_ns, dst_net_ns))
+ pop_net_ns(net_ns);
+
+ IP_INC_STATS_BH(IPSTATS_MIB_INHDRERRORS);
drop:
    kfree_skb(skb);
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
@@ -99,6 +99,8 @@
extern __be32 net_ns_select_source_address(const struct net_device *dev,
    u32 dst, int scope);

+extern struct net_namespace *net_ns_find_from_dest_addr(u32 daddr);
+
#define SELECT_SRC_ADDR net_ns_select_source_address

#else /* CONFIG_NET_NS */
@@ -167,6 +169,11 @@
    return 0;
}

+static inline struct net_namespace *net_ns_find_from_dest_addr(u32 daddr)
+{
+ return NULL;
+}
+
#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
@@ -385,4 +385,39 @@
out:
    return addr;
}
+
+/*
+ * This function finds the network namespace destination deduced from

```

```

+ * the destination address. The network namespace is retrieved from
+ * the ifaddr owned by a network namespace
+ * @daddr : destination
+ * Returns : the network namespace destination or NULL if not found
+ */
+struct net_namespace *net_ns_find_from_dest_addr(u32 daddr)
+{
+ struct net_namespace *net_ns = NULL;
+ struct net_device *dev;
+ struct in_device *in_dev;
+
+ if (LOOPBACK(daddr))
+ return current_net_ns;
+
+ read_lock(&dev_base_lock);
+ rcu_read_lock();
+ for (dev = dev_base; dev; dev = dev->next) {
+ if ((in_dev = __in_dev_get_rcu(dev)) == NULL)
+ continue;
+ for_ifa(in_dev) {
+ if (ifa->ifa_local == daddr) {
+ net_ns = ifa->ifa_net_ns;
+ goto out_unlock_both;
+ }
+ } endfor_ifa(in_dev);
+ }
+out_unlock_both:
+ read_unlock(&dev_base_lock);
+ rcu_read_unlock();
+
+ return net_ns;
+}
#endif /* CONFIG_NET_NS */

--

```

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