

Subject: [patch 11/20] [Network namespace] Check the bind address is allowed. It must match ifaddr assigned to

Posted by [Daniel Lezcano](#) on Sun, 10 Dec 2006 21:58:28 GMT

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

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

```
include/linux/net_namespace.h | 7 +++++++
net/core/net_namespace.c      | 40 +++++++++++++++++++++++++++++++++++++
net/ipv4/af_inet.c            | 2 ++
net/ipv4/raw.c                | 3 +++
4 files changed, 52 insertions(+)
```

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
@@ -433,6 +433,8 @@ int inet_bind(struct socket *sock, struct
 * is temporarily down)
 */
err = -EADDRNOTAVAIL;
+ if (net_ns_check_bind(chk_addr_ret, addr->sin_addr.s_addr))
+ goto out;
if (!sysctl_ip_nonlocal_bind &&
    !inet->freebind &&
    addr->sin_addr.s_addr != INADDR_ANY &&
```

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

```
=====
--- 2.6.19-rc6-mm2.orig/net/ipv4/raw.c
+++ 2.6.19-rc6-mm2/net/ipv4/raw.c
@@ -559,7 +559,10 @@ static int raw_bind(struct sock *sk, struct
if (sk->sk_state != TCP_CLOSE || addr_len < sizeof(struct sockaddr_in))
goto out;
chk_addr_ret = inet_addr_type(addr->sin_addr.s_addr);
+
ret = -EADDRNOTAVAIL;
+ if (net_ns_check_bind(chk_addr_ret, addr->sin_addr.s_addr))
+ goto out;
if (addr->sin_addr.s_addr && chk_addr_ret != RTN_LOCAL &&
    chk_addr_ret != RTN_MULTICAST && chk_addr_ret != RTN_BROADCAST)
goto out;
```

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
@@ -88,6 +88,8 @@ extern int hlist_dev_name_init(struct ne
```

```

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

+extern int net_ns_check_bind(int addr_type, u32 addr);
+
+else /* CONFIG_NET_NS */

#define INIT_NET_NS(net_ns)
@@ -139,6 +141,11 @@ static inline int net_ns_ioctl(unsigned
    return -ENOSYS;
}

+static inline int net_ns_check_bind(int addr_type, u32 addr)
+{
+ return 0;
+}
+
#endif /* !CONFIG_NET_NS */

```

```

#endif /* _LINUX_NET_NAMESPACE_H */

```

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

```

```

@@ -280,4 +280,44 @@ out:
    return err;
}

```

```

+int net_ns_check_bind(int addr_type, u32 addr)
+{
+ int ret = -EPERM;
+     struct net_device *dev;
+     struct in_device *in_dev;
+ struct net_namespace *net_ns = current_net_ns;
+
+ if (LOOPBACK(addr) ||
+     MULTICAST(addr) ||
+     INADDR_ANY == addr ||
+     INADDR_BROADCAST == addr)
+ return 0;
+
+     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;
+             if (addr == ifa->ifa_local ||
+                 addr == ifa->ifa_broadcast ||
+                 addr == (ifa->ifa_local & ifa->ifa_mask) ||
+                 addr == ((ifa->ifa_address & ifa->ifa_mask) |
+                     ~ifa->ifa_mask)) {
+                 ret = 0;
+                 goto out;
+             }
+         } endfor_ifa(in_dev);
+     }
+out:
+     read_unlock(&dev_base_lock);
+     rcu_read_unlock();
+
+     return ret;
+}
+
+endif /* CONFIG_NET_NS */

```

--

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