
Subject: [PATCH RFC 13/31] net: Make device event notification network namespace safe

Posted by [ebiederm](#) on Thu, 25 Jan 2007 19:00:15 GMT

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

From: Eric W. Biederman <ebiederm@xmission.com> - unquoted

Every user of the network device notifiers is either a protocol stack or a pseudo device. If a protocol stack that does not have support for multiple network namespaces receives an event for a device that is not in the initial network namespace it quite possibly can get confused and do the wrong thing.

To avoid problems until all of the protocol stacks are converted this patch modifies all netdev event handlers to ignore events on devices that are not in the initial network namespace.

As the rest of the code is made network namespace aware these checks can be removed.

Signed-off-by: Eric W. Biederman <ebiederm@xmission.com>

```
arch/ia64/hp/sim/simeth.c      | 3 +++
drivers/net/bonding/bond_main.c | 3 +++
drivers/net/hamradio/bpqether.c | 3 +++
drivers/net/pppoe.c             | 3 +++
drivers/net/wan/dlci.c          | 3 +++
drivers/net/wan/hdlc.c          | 3 +++
drivers/net/wan/lapbether.c     | 3 +++
net/8021q/vlan.c               | 4 ++++
net/appletalk/aarp.c           | 3 +++
net/appletalk/ddp.c            | 3 +++
net/atm/clip.c                 | 3 +++
net/atm/mpc.c                  | 4 ++++
net/ax25/af_ax25.c             | 3 +++
net/bridge/br_notify.c         | 4 ++++
net/core/dst.c                 | 4 ++++
net/core/fib_rules.c           | 4 ++++
net/core/pktgen.c              | 3 +++
net/core/rtnetlink.c           | 4 ++++
net/dechnet/af_dechnet.c       | 3 +++
net/econet/af_econet.c         | 3 +++
net/ipv4/arp.c                 | 3 +++
net/ipv4/devinet.c             | 3 +++
net/ipv4/fib_frontend.c        | 3 +++
net/ipv4/ipmr.c                | 7 ++++++-
net/ipv4/multipath_drr.c       | 3 +++
net/ipv4/netfilter/ip_queue.c  | 3 +++
```

```

net/ipv4/netfilter/ipt_MASQUERADE.c | 3 +++
net/ipv6/addrconf.c | 3 +++
net/ipv6/ndisc.c | 3 +++
net/ipv6/netfilter/ip6_queue.c | 3 +++
net/ipx/af_ipx.c | 3 +++
net/netfilter/nfnetlink_queue.c | 3 +++
net/netrom/af_netrom.c | 3 +++
net/packet/af_packet.c | 3 +++
net/rose/af_rose.c | 3 +++
net/tipc/eth_media.c | 3 +++
net/wanrouter/af_wanpipe.c | 3 +++
net/x25/af_x25.c | 3 +++
net/xfrm/xfrm_policy.c | 5 +++++
security/selinux/netif.c | 3 +++
40 files changed, 131 insertions(+), 1 deletions(-)

```

```

diff --git a/arch/ia64/hp/sim/simeth.c b/arch/ia64/hp/sim/simeth.c
index 424e925..1cbaa9e 100644

```

```

--- a/arch/ia64/hp/sim/simeth.c
+++ b/arch/ia64/hp/sim/simeth.c
@@ -300,6 +300,9 @@ simeth_device_event(struct notifier_block *this,unsigned long event, void
*ptr)
    return NOTIFY_DONE;
}

```

```

+ if (!net_eq(dev->nd_net, init_net()))
+ return NOTIFY_DONE;
+
+ if ( event != NETDEV_UP && event != NETDEV_DOWN ) return NOTIFY_DONE;

```

```

/*

```

```

diff --git a/drivers/net/bonding/bond_main.c b/drivers/net/bonding/bond_main.c
index 9c70568..3e04f58 100644

```

```

--- a/drivers/net/bonding/bond_main.c
+++ b/drivers/net/bonding/bond_main.c
@@ -3325,6 +3325,9 @@ static int bond_netdev_event(struct notifier_block *this, unsigned long
event, v
{
    struct net_device *event_dev = (struct net_device *)ptr;

```

```

+ if (!net_eq(event_dev->nd_net, init_net()))
+ return NOTIFY_DONE;
+
+ dprintk("event_dev: %s, event: %lx\n",
+ (event_dev ? event_dev->name : "None"),
+ event);

```

```

diff --git a/drivers/net/hamradio/bpqether.c b/drivers/net/hamradio/bpqether.c
index c513e90..8826a96 100644

```

```

--- a/drivers/net/hamradio/bpqether.c
+++ b/drivers/net/hamradio/bpqether.c
@@ -564,6 +564,9 @@ static int bpq_device_event(struct notifier_block *this,unsigned long
event, voi
{
    struct net_device *dev = (struct net_device *)ptr;

+ if (!net_eq(dev->nd_net, init_net()))
+ return NOTIFY_DONE;
+
    if (!dev_is_ethdev(dev))
        return NOTIFY_DONE;

diff --git a/drivers/net/pppoe.c b/drivers/net/pppoe.c
index caf8ca3..3618862 100644
--- a/drivers/net/pppoe.c
+++ b/drivers/net/pppoe.c
@@ -299,6 +299,9 @@ static int pppoe_device_event(struct notifier_block *this,
{
    struct net_device *dev = (struct net_device *) ptr;

+ if (!net_eq(dev->nd_net, init_net()))
+ return NOTIFY_DONE;
+
    /* Only look at sockets that are using this specific device. */
    switch (event) {
    case NETDEV_CHANGEMTU:
diff --git a/drivers/net/wan/dlci.c b/drivers/net/wan/dlci.c
index 7369875..f826494 100644
--- a/drivers/net/wan/dlci.c
+++ b/drivers/net/wan/dlci.c
@@ -513,6 +513,9 @@ static int dlci_dev_event(struct notifier_block *unused,
{
    struct net_device *dev = (struct net_device *) ptr;

+ if (!net_eq(dev->nd_net, init_net()))
+ return NOTIFY_DONE;
+
    if (event == NETDEV_UNREGISTER) {
        struct dlci_local *dlp;

diff --git a/drivers/net/wan/hdlc.c b/drivers/net/wan/hdlc.c
index f3bf160..e56e0a1 100644
--- a/drivers/net/wan/hdlc.c
+++ b/drivers/net/wan/hdlc.c
@@ -110,6 +110,9 @@ static int hdlc_device_event(struct notifier_block *this, unsigned long
event,
    unsigned long flags;

```

```

int on;

+ if (!net_eq(dev->nd_net, init_net()))
+ return NOTIFY_DONE;
+
+ if (dev->get_stats != hdlc_get_stats)
+ return NOTIFY_DONE; /* not an HDLC device */

diff --git a/drivers/net/wan/lapbether.c b/drivers/net/wan/lapbether.c
index c1de21e..a3560a9 100644
--- a/drivers/net/wan/lapbether.c
+++ b/drivers/net/wan/lapbether.c
@@ -395,6 +395,9 @@ static int lapbeth_device_event(struct notifier_block *this,
    struct lapbethdev *lapbeth;
    struct net_device *dev = ptr;

+ if (!net_eq(dev->nd_net, init_net()))
+ return NOTIFY_DONE;
+
+ if (!dev_is_ethdev(dev))
+ return NOTIFY_DONE;

diff --git a/net/8021q/vlan.c b/net/8021q/vlan.c
index 18fcb9f..f80cfdd 100644
--- a/net/8021q/vlan.c
+++ b/net/8021q/vlan.c
@@ -31,6 +31,7 @@
@@
#include <net/arp.h>
#include <linux/rtnetlink.h>
#include <linux/notifier.h>
+#include <net/net_namespace.h>

#include <linux/if_vlan.h>
#include "vlan.h"
@@ -595,6 +596,9 @@ static int vlan_device_event(struct notifier_block *unused, unsigned long
event,
    int i, flgs;
    struct net_device *vlandev;

+ if (!net_eq(dev->nd_net, init_net()))
+ return NOTIFY_DONE;
+
+ if (!grp)
+ goto out;

diff --git a/net/appletalk/aarp.c b/net/appletalk/aarp.c
index 85c4dbc..6fd58a6 100644
--- a/net/appletalk/aarp.c

```

```

+++ b/net/appletalk/aarp.c
@@ -327,6 +327,9 @@ static int aarp_device_event(struct notifier_block *this, unsigned long
event,
    struct net_device *dev = ptr;
    int ct;

+ if (!net_eq(dev->nd_net, init_net()))
+ return NOTIFY_DONE;
+
    if (event == NETDEV_DOWN) {
        write_lock_bh(&aarp_lock);

diff --git a/net/appletalk/ddp.c b/net/appletalk/ddp.c
index f4ff8aa..61f36b1 100644
--- a/net/appletalk/ddp.c
+++ b/net/appletalk/ddp.c
@@ -649,6 +649,9 @@ static int ddp_device_event(struct notifier_block *this, unsigned long
event,
{
    struct net_device *dev = ptr;

+ if (!net_eq(dev->nd_net, init_net()))
+ return NOTIFY_DONE;
+
    if (event == NETDEV_DOWN)
        /* Discard any use of this */
        atalk_dev_down(dev);
diff --git a/net/atm/clip.c b/net/atm/clip.c
index 5f8a1d2..7d150c2 100644
--- a/net/atm/clip.c
+++ b/net/atm/clip.c
@@ -629,6 +629,9 @@ static int clip_device_event(struct notifier_block *this, unsigned long
event,
{
    struct net_device *dev = arg;

+ if (!net_eq(dev->nd_net, init_net()))
+ return NOTIFY_DONE;
+
    if (event == NETDEV_UNREGISTER) {
        neigh_ifdown(&clip_tbl, dev);
        return NOTIFY_DONE;
diff --git a/net/atm/mpc.c b/net/atm/mpc.c
index c18f737..4fdb1af 100644
--- a/net/atm/mpc.c
+++ b/net/atm/mpc.c
@@ -953,6 +953,10 @@ static int mpoa_event_listener(struct notifier_block *mpoa_notifier,
unsigned lo

```

```

struct lec_priv *priv;

dev = (struct net_device *)dev_ptr;
+
+ if (!net_eq(dev->nd_net, init_net()))
+ return NOTIFY_DONE;
+
if (dev->name == NULL || strncmp(dev->name, "lec", 3))
return NOTIFY_DONE; /* we are only interested in lec:s */

diff --git a/net/ax25/af_ax25.c b/net/ax25/af_ax25.c
index cdbf3f6..8c187a6 100644
--- a/net/ax25/af_ax25.c
+++ b/net/ax25/af_ax25.c
@@ -105,6 +105,9 @@ static int ax25_device_event(struct notifier_block *this, unsigned long
event,
{
    struct net_device *dev = (struct net_device *)ptr;

+ if (!net_eq(dev->nd_net, init_net()))
+ return NOTIFY_DONE;
+
    /* Reject non AX.25 devices */
    if (dev->type != ARPHRD_AX25)
        return NOTIFY_DONE;
diff --git a/net/bridge/br_notify.c b/net/bridge/br_notify.c
index 2027849..0d56bc2 100644
--- a/net/bridge/br_notify.c
+++ b/net/bridge/br_notify.c
@@ -15,6 +15,7 @@

#include <linux/kernel.h>
#include <linux/rtnetlink.h>
+#include <net/net_namespace.h>

#include "br_private.h"

@@ -36,6 +37,9 @@ static int br_device_event(struct notifier_block *unused, unsigned long
event, v
    struct net_bridge_port *p = dev->br_port;
    struct net_bridge *br;

+ if (!net_eq(dev->nd_net, init_net()))
+ return NOTIFY_DONE;
+
    /* not a port of a bridge */
    if (p == NULL)
        return NOTIFY_DONE;

```

```
diff --git a/net/core/dst.c b/net/core/dst.c
```

```
index 836ec66..8c4a272 100644
```

```
--- a/net/core/dst.c
```

```
+++ b/net/core/dst.c
```

```
@ @ -16,6 +16,7 @ @
```

```
#include <linux/skbuff.h>
```

```
#include <linux/string.h>
```

```
#include <linux/types.h>
```

```
+#include <net/net_namespace.h>
```

```
#include <net/dst.h>
```

```
@ @ -256,6 +257,9 @ @ static int dst_dev_event(struct notifier_block *this, unsigned long event, void
```

```
    struct net_device *dev = ptr;
```

```
    struct dst_entry *dst;
```

```
+ if (!net_eq(dev->nd_net, init_net()))
```

```
+ return NOTIFY_DONE;
```

```
+
```

```
    switch (event) {
```

```
        case NETDEV_UNREGISTER:
```

```
        case NETDEV_DOWN:
```

```
diff --git a/net/core/fib_rules.c b/net/core/fib_rules.c
```

```
index 1df6cd4..ffc31c1 100644
```

```
--- a/net/core/fib_rules.c
```

```
+++ b/net/core/fib_rules.c
```

```
@ @ -11,6 +11,7 @ @
```

```
#include <linux/types.h>
```

```
#include <linux/kernel.h>
```

```
#include <linux/list.h>
```

```
+#include <net/net_namespace.h>
```

```
#include <net/fib_rules.h>
```

```
static LIST_HEAD(rules_ops);
```

```
@ @ -441,6 +442,9 @ @ static int fib_rules_event(struct notifier_block *this, unsigned long event,
```

```
    struct net_device *dev = ptr;
```

```
    struct fib_rules_ops *ops;
```

```
+ if (!net_eq(dev->nd_net, init_net()))
```

```
+ return NOTIFY_DONE;
```

```
+
```

```
    ASSERT_RTNL();
```

```
    rcu_read_lock();
```

```
diff --git a/net/core/pktgen.c b/net/core/pktgen.c
```

```
index ab48533..7796b39 100644
```

```
--- a/net/core/pktgen.c
```

```

+++ b/net/core/pktgen.c
@@ -1892,6 +1892,9 @@ static int pktgen_device_event(struct notifier_block *unused,
{
    struct net_device *dev = (struct net_device *) (ptr);

+ if (!net_eq(dev->nd_net, init_net()))
+ return NOTIFY_DONE;
+
    /* It is OK that we do not hold the group lock right now,
     * as we run under the RTNL lock.
     */
diff --git a/net/core/rtnetlink.c b/net/core/rtnetlink.c
index e76539a..7841e89 100644
--- a/net/core/rtnetlink.c
+++ b/net/core/rtnetlink.c
@@ -829,6 +829,10 @@ static struct rtnetlink_link link_rtnetlink_table[RTM_NR_MSGTYPES] =
static int rtnetlink_event(struct notifier_block *this, unsigned long event, void *ptr)
{
    struct net_device *dev = ptr;
+
+ if (!net_eq(dev->nd_net, init_net()))
+ return NOTIFY_DONE;
+
    switch (event) {
    case NETDEV_UNREGISTER:
        rtmsg_ifinfo(RTM_DELLINK, dev, ~0U);
diff --git a/net/decnet/af_decnet.c b/net/decnet/af_decnet.c
index 5e8042f..b27b2ac 100644
--- a/net/decnet/af_decnet.c
+++ b/net/decnet/af_decnet.c
@@ -2086,6 +2086,9 @@ static int dn_device_event(struct notifier_block *this, unsigned long
event,
{
    struct net_device *dev = (struct net_device *) ptr;

+ if (!net_eq(dev->nd_net, init_net()))
+ return NOTIFY_DONE;
+
    switch (event) {
    case NETDEV_UP:
        dn_dev_up(dev);
diff --git a/net/econet/af_econet.c b/net/econet/af_econet.c
index 0baffda..cbf87f4 100644
--- a/net/econet/af_econet.c
+++ b/net/econet/af_econet.c
@@ -1121,6 +1121,9 @@ static int econet_notifier(struct notifier_block *this, unsigned long msg,
void
{
    struct net_device *dev = (struct net_device *) data;

```



```

struct ec_device *edev;

+ if (!net_eq(dev->nd_net, init_net()))
+ return NOTIFY_DONE;
+
+ switch (msg) {
+ case NETDEV_UNREGISTER:
+ /* A device has gone down - kill any data we hold for it. */
diff --git a/net/ipv4/arp.c b/net/ipv4/arp.c
index 95a34c7..0d23fb2 100644
--- a/net/ipv4/arp.c
+++ b/net/ipv4/arp.c
@@ -1206,6 +1206,9 @@ static int arp_netdev_event(struct notifier_block *this, unsigned long
event, void
{
    struct net_device *dev = ptr;

+ if (!net_eq(dev->nd_net, init_net()))
+ return NOTIFY_DONE;
+
+ switch (event) {
+ case NETDEV_CHANGEADDR:
+ neigh_changeaddr(&arp_tbl, dev);
diff --git a/net/ipv4/devinet.c b/net/ipv4/devinet.c
index 216cf2b..a7d991d 100644
--- a/net/ipv4/devinet.c
+++ b/net/ipv4/devinet.c
@@ -1050,6 +1050,9 @@ static int inetdev_event(struct notifier_block *this, unsigned long event,
struct net_device *dev = ptr;
struct in_device *in_dev = __in_dev_get_rtnl(dev);

+ if (!net_eq(dev->nd_net, init_net()))
+ return NOTIFY_DONE;
+
+ ASSERT_RTNL();

+ if (in_dev) {
diff --git a/net/ipv4/fib_frontend.c b/net/ipv4/fib_frontend.c
index d47b72a..049c370 100644
--- a/net/ipv4/fib_frontend.c
+++ b/net/ipv4/fib_frontend.c
@@ -860,6 +860,9 @@ static int fib_netdev_event(struct notifier_block *this, unsigned long
event, void
{
    struct net_device *dev = ptr;
    struct in_device *in_dev = __in_dev_get_rtnl(dev);

+ if (!net_eq(dev->nd_net, init_net()))
+ return NOTIFY_DONE;

```

```

+
if (event == NETDEV_UNREGISTER) {
    fib_disable_ip(dev, 2);
    return NOTIFY_DONE;
diff --git a/net/ipv4/ipmr.c b/net/ipv4/ipmr.c
index af50394..9afaa13 100644
--- a/net/ipv4/ipmr.c
+++ b/net/ipv4/ipmr.c
@@ -1075,13 +1075,18 @@ int ipmr_ioctl(struct sock *sk, int cmd, void __user *arg)

static int ipmr_device_event(struct notifier_block *this, unsigned long event, void *ptr)
{
+ struct net_device *dev = ptr;
    struct vif_device *v;
    int ct;
+
+ if (!net_eq(dev->nd_net, init_net()))
+ return NOTIFY_DONE;
+
    if (event != NETDEV_UNREGISTER)
        return NOTIFY_DONE;
    v=&vif_table[0];
    for(ct=0;ct<maxvif;ct++,v++) {
- if (v->dev==ptr)
+ if (v->dev==dev)
        vif_delete(ct);
    }
    return NOTIFY_DONE;
diff --git a/net/ipv4/multipath_drr.c b/net/ipv4/multipath_drr.c
index 252e837..b14d6ae 100644
--- a/net/ipv4/multipath_drr.c
+++ b/net/ipv4/multipath_drr.c
@@ -87,6 +87,9 @@ static int drr_dev_event(struct notifier_block *this,
    struct net_device *dev = ptr;
    int devidx;

+ if (!net_eq(dev->nd_net, init_net()))
+ return NOTIFY_DONE;
+
    switch (event) {
        case NETDEV_UNREGISTER:
        case NETDEV_DOWN:
diff --git a/net/ipv4/netfilter/ip_queue.c b/net/ipv4/netfilter/ip_queue.c
index aae660c..8650a57 100644
--- a/net/ipv4/netfilter/ip_queue.c
+++ b/net/ipv4/netfilter/ip_queue.c
@@ -567,6 +567,9 @@ ipq_rcv_dev_event(struct notifier_block *this,
{

```

```

    struct net_device *dev = ptr;

+ if (!net_eq(dev->nd_net, init_net()))
+ return NOTIFY_DONE;
+
    /* Drop any packets associated with the downed device */
    if (event == NETDEV_DOWN)
        ipq_dev_drop(dev->ifindex);
diff --git a/net/ipv4/netfilter/ipt_MASQUERADE.c b/net/ipv4/netfilter/ipt_MASQUERADE.c
index d669685..41fe6b5 100644
--- a/net/ipv4/netfilter/ipt_MASQUERADE.c
+++ b/net/ipv4/netfilter/ipt_MASQUERADE.c
@@ -152,6 +152,9 @@ static int masq_device_event(struct notifier_block *this,
{
    struct net_device *dev = ptr;

+ if (!net_eq(dev->nd_net, init_net()))
+ return NOTIFY_DONE;
+
    if (event == NETDEV_DOWN) {
        /* Device was downed. Search entire table for
           conntracks which were associated with that device,
diff --git a/net/ipv6/addrconf.c b/net/ipv6/addrconf.c
index 52bd4dd..7be542f 100644
--- a/net/ipv6/addrconf.c
+++ b/net/ipv6/addrconf.c
@@ -2235,6 +2235,9 @@ static int addrconf_notify(struct notifier_block *this, unsigned long
event,
    struct inet6_dev *idev = __in6_dev_get(dev);
    int run_pending = 0;

+ if (!net_eq(dev->nd_net, init_net()))
+ return NOTIFY_DONE;
+
    switch(event) {
        case NETDEV_UP:
        case NETDEV_CHANGE:
diff --git a/net/ipv6/ndisc.c b/net/ipv6/ndisc.c
index 6a9f616..9b3495f 100644
--- a/net/ipv6/ndisc.c
+++ b/net/ipv6/ndisc.c
@@ -1586,6 +1586,9 @@ static int ndisc_netdev_event(struct notifier_block *this, unsigned long
event,
{
    struct net_device *dev = ptr;

+ if (!net_eq(dev->nd_net, init_net()))
+ return NOTIFY_DONE;

```

```

+
switch (event) {
case NETDEV_CHANGEADDR:
    neigh_changeaddr(&nd_tbl, dev);
diff --git a/net/ipv6/netfilter/ip6_queue.c b/net/ipv6/netfilter/ip6_queue.c
index 45b64a5..f6e108c 100644
--- a/net/ipv6/netfilter/ip6_queue.c
+++ b/net/ipv6/netfilter/ip6_queue.c
@@ -557,6 +557,9 @@ ipq_rcv_dev_event(struct notifier_block *this,
{
    struct net_device *dev = ptr;

+ if (!net_eq(dev->nd_net, init_net()))
+ return NOTIFY_DONE;
+
    /* Drop any packets associated with the downed device */
    if (event == NETDEV_DOWN)
        ipq_dev_drop(dev->ifindex);
diff --git a/net/ipx/af_ipx.c b/net/ipx/af_ipx.c
index 5c5f2cd..f2674fe 100644
--- a/net/ipx/af_ipx.c
+++ b/net/ipx/af_ipx.c
@@ -347,6 +347,9 @@ static int ipxif_device_event(struct notifier_block *notifier,
    struct net_device *dev = ptr;
    struct ipx_interface *i, *tmp;

+ if (!net_eq(dev->nd_net, init_net()))
+ return NOTIFY_DONE;
+
    if (event != NETDEV_DOWN && event != NETDEV_UP)
        goto out;

diff --git a/net/netfilter/nfnetlink_queue.c b/net/netfilter/nfnetlink_queue.c
index a88a017..59bf595 100644
--- a/net/netfilter/nfnetlink_queue.c
+++ b/net/netfilter/nfnetlink_queue.c
@@ -734,6 +734,9 @@ nfqnl_rcv_dev_event(struct notifier_block *this,
{
    struct net_device *dev = ptr;

+ if (!net_eq(dev->nd_net, init_net()))
+ return NOTIFY_DONE;
+
    /* Drop any packets associated with the downed device */
    if (event == NETDEV_DOWN)
        nfqnl_dev_drop(dev->ifindex);
diff --git a/net/netrom/af_netrom.c b/net/netrom/af_netrom.c
index 3fa3f1a..6965a1a 100644

```

```

--- a/net/netrom/af_netrom.c
+++ b/net/netrom/af_netrom.c
@@ -106,6 +106,9 @@ static int nr_device_event(struct notifier_block *this, unsigned long event,
void
{
    struct net_device *dev = (struct net_device *)ptr;

+ if (!net_eq(dev->nd_net, init_net()))
+ return NOTIFY_DONE;
+
    if (event != NETDEV_DOWN)
        return NOTIFY_DONE;

diff --git a/net/packet/af_packet.c b/net/packet/af_packet.c
index aa298c3..6e3b947 100644
--- a/net/packet/af_packet.c
+++ b/net/packet/af_packet.c
@@ -1439,6 +1439,9 @@ static int packet_notifier(struct notifier_block *this, unsigned long msg,
void
{
    struct hlist_node *node;
    struct net_device *dev = (struct net_device *)data;

+ if (!net_eq(dev->nd_net, init_net()))
+ return NOTIFY_DONE;
+
    read_lock(&packet_sklist_lock);
    sk_for_each(sk, node, &packet_sklist) {
        struct packet_sock *po = pkt_sk(sk);
diff --git a/net/rose/af_rose.c b/net/rose/af_rose.c
index 7d5e593..dad50d3 100644
--- a/net/rose/af_rose.c
+++ b/net/rose/af_rose.c
@@ -197,6 +197,9 @@ static int rose_device_event(struct notifier_block *this, unsigned long
event,
{
    struct net_device *dev = (struct net_device *)ptr;

+ if (!net_eq(dev->nd_net, init_net()))
+ return NOTIFY_DONE;
+
    if (event != NETDEV_DOWN)
        return NOTIFY_DONE;

diff --git a/net/tipc/eth_media.c b/net/tipc/eth_media.c
index b181cf9..c6f64de 100644
--- a/net/tipc/eth_media.c
+++ b/net/tipc/eth_media.c
@@ -194,6 +194,9 @@ static int recv_notification(struct notifier_block *nb, unsigned long evt,

```

```

struct eth_bearer *eb_ptr = &eth_bearers[0];
struct eth_bearer *stop = &eth_bearers[MAX_ETH_BEARERS];

+ if (!net_eq(dev->nd_net, init_net()))
+ return NOTIFY_DONE;
+
while ((eb_ptr->dev != dev)) {
    if (++eb_ptr == stop)
        return NOTIFY_DONE; /* couldn't find device */
}
diff --git a/net/wanrouter/af_wanpipe.c b/net/wanrouter/af_wanpipe.c
index 542c737..f9b896c 100644
--- a/net/wanrouter/af_wanpipe.c
+++ b/net/wanrouter/af_wanpipe.c
@@ -1714,6 +1714,9 @@ static int wanpipe_notifier(struct notifier_block *this, unsigned long
msg, void
    hlist_node *node;
    struct net_device *dev = (struct net_device *)data;

+ if (!net_eq(dev->nd_net, init_net()))
+ return NOTIFY_DONE;
+
sk_for_each(sk, node, &wanpipe_sklist) {
    struct wanpipe_opt *po = wp_sk(sk);

diff --git a/net/x25/af_x25.c b/net/x25/af_x25.c
index 6602a34..f4fad10 100644
--- a/net/x25/af_x25.c
+++ b/net/x25/af_x25.c
@@ -190,6 +190,9 @@ static int x25_device_event(struct notifier_block *this, unsigned long
event,
    struct net_device *dev = ptr;
    struct x25_neigh *nb;

+ if (!net_eq(dev->nd_net, init_net()))
+ return NOTIFY_DONE;
+
if (dev->type == ARPHRD_X25
#ifdef CONFIG_LLC
|| defined(CONFIG_LLC_MODULE)
|| dev->type == ARPHRD_ETHER
}
diff --git a/net/xfrm/xfrm_policy.c b/net/xfrm/xfrm_policy.c
index bebd40e..0248343 100644
--- a/net/xfrm/xfrm_policy.c
+++ b/net/xfrm/xfrm_policy.c
@@ -2188,6 +2188,11 @@ static void xfrm_policy_unlock_afinfo(struct xfrm_policy_afinfo
*afinfo)

static int xfrm_dev_event(struct notifier_block *this, unsigned long event, void *ptr)
{

```

```

+ struct net_device *dev = ptr;
+
+ if (!net_eq(dev->nd_net, init_net()))
+ return NOTIFY_DONE;
+
+ switch (event) {
+ case NETDEV_DOWN:
+ xfrm_flush_bundles();
diff --git a/security/selinux/netif.c b/security/selinux/netif.c
index b10c34e..45c422f 100644
--- a/security/selinux/netif.c
+++ b/security/selinux/netif.c
@@ -234,6 +234,9 @@ static int sel_netif_netdev_notifier_handler(struct notifier_block *this,
{
    struct net_device *dev = ptr;

+ if (!net_eq(dev->nd_net, init_net()))
+ return NOTIFY_DONE;
+
+ if (event == NETDEV_DOWN)
+ sel_netif_kill(dev);

--
1.4.4.1.g278f

```

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