
Subject: [PATCH 11/16] net: Make device event notification network namespace safe

Posted by [ebiederm](#) on Sat, 08 Sep 2007 21:27:11 GMT

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

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/decnnet/af_decnnet.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/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/x25/af_x25.c                  | 3 +++
net/xfrm/xfrm_policy.c            | 5 +++++
security/selinux/netif.c          | 4 +++++
38 files changed, 126 insertions(+), 1 deletions(-)

```

```

diff --git a/arch/ia64/hp/sim/simeth.c b/arch/ia64/hp/sim/simeth.c
index f26077a..93d6004 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 (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 e4e5fdc..cf97d8a 100644
--- a/drivers/net/bonding/bond_main.c
+++ b/drivers/net/bonding/bond_main.c
@@ -3299,6 +3299,9 @@ static int bond_netdev_event(struct notifier_block *this, unsigned long
event, v
{
    struct net_device *event_dev = (struct net_device *)ptr;

```

```

+ if (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 85fb8e7..df09210 100644
--- a/drivers/net/hamradio/bpqether.c
+++ b/drivers/net/hamradio/bpqether.c
@@ -563,6 +563,9 @@ static int bpq_device_event(struct notifier_block *this, unsigned long
event, voi

```

```

{
    struct net_device *dev = (struct net_device *)ptr;

+ if (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 a29ea22..b4ba584 100644
--- a/drivers/net/pppoe.c
+++ b/drivers/net/pppoe.c
@@ -301,6 +301,9 @@ static int pppoe_device_event(struct notifier_block *this,
{
    struct net_device *dev = (struct net_device *) ptr;

+ if (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 66be20c..61041d5 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 (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 3b57350..ee23b91 100644
--- a/drivers/net/wan/hdlc.c
+++ b/drivers/net/wan/hdlc.c
@@ -109,6 +109,9 @@ static int hdlc_device_event(struct notifier_block *this, unsigned long
event,
    unsigned long flags;
    int on;

+ if (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 ca8b3c3..699b934 100644
--- a/drivers/net/wan/lapbether.c
+++ b/drivers/net/wan/lapbether.c
@@ -394,6 +394,9 @@ static int lapbeth_device_event(struct notifier_block *this,
    struct lapbethdev *lapbeth;
    struct net_device *dev = ptr;

+ if (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 1583c5e..55f2dbb 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"
@@ -605,6 +606,9 @@ static int vlan_device_event(struct notifier_block *unused, unsigned long
event,
    int i, flgs;
    struct net_device *vlandev;

+ if (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 9267f48..e9a51a6 100644
--- a/net/appletalk/aarp.c
+++ b/net/appletalk/aarp.c
@@ -333,6 +333,9 @@ static int aarp_device_event(struct notifier_block *this, unsigned long
event,
    struct net_device *dev = ptr;

```

```

int ct;

+ if (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 c1f1367..36fcdbf 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 (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 806ea98..741742f 100644
--- a/net/atm/clip.c
+++ b/net/atm/clip.c
@@ -612,6 +612,9 @@ static int clip_device_event(struct notifier_block *this, unsigned long
event,
{
    struct net_device *dev = arg;

+ if (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 7c85aa5..0968430 100644
--- a/net/atm/mpc.c
+++ b/net/atm/mpc.c
@@ -956,6 +956,10 @@ static int mpoa_event_listener(struct notifier_block *mpoa_notifier,
unsigned lo
    struct lec_priv *priv;

    dev = (struct net_device *)dev_ptr;
+

```

```

+ if (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 def6c42..8d13a8b 100644
--- a/net/ax25/af_ax25.c
+++ b/net/ax25/af_ax25.c
@@ -104,6 +104,9 @@ static int ax25_device_event(struct notifier_block *this, unsigned long
event,
{
    struct net_device *dev = (struct net_device *)ptr;

+ if (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 c8451d3..07ac3ae 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 (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 c6a0587..32267a1 100644
--- a/net/core/dst.c
+++ b/net/core/dst.c

```

```
@@ -15,6 +15,7 @@
#include <linux/skbuff.h>
#include <linux/string.h>
#include <linux/types.h>
+#include <net/net_namespace.h>
```

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

```
@@ -252,6 +253,9 @@ static int dst_dev_event(struct notifier_block *this, unsigned long event,
void
    struct net_device *dev = ptr;
    struct dst_entry *dst;
```

```
+ if (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 8c5474e..9eabe1a 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);
```

```
@@ -596,6 +597,9 @@ static int fib_rules_event(struct notifier_block *this, unsigned long event,
    struct net_device *dev = ptr;
    struct fib_rules_ops *ops;
```

```
+ if (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 fa15cc7..21ae955 100644
```

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

```
@@ -1966,6 +1966,9 @@ static int pktgen_device_event(struct notifier_block *unused,
{
    struct net_device *dev = ptr;
```

```

+ if (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 dca9e80..4185950 100644
--- a/net/core/rtnetlink.c
+++ b/net/core/rtnetlink.c
@@ -1286,6 +1286,10 @@ static void rtnetlink_rcv(struct sock *sk, int len)
static int rtnetlink_event(struct notifier_block *this, unsigned long event, void *ptr)
{
    struct net_device *dev = ptr;
+
+ if (dev->nd_net != &init_net)
+ return NOTIFY_DONE;
+
    switch (event) {
    case NETDEV_UNREGISTER:
        rtmsg_ifinfo(RTM_DELLINK, dev, ~0U);
diff --git a/net/decnnet/af_decnnet.c b/net/decnnet/af_decnnet.c
index aca4c49..83398da 100644
--- a/net/decnnet/af_decnnet.c
+++ b/net/decnnet/af_decnnet.c
@@ -2089,6 +2089,9 @@ static int dn_device_event(struct notifier_block *this, unsigned long
event,
{
    struct net_device *dev = (struct net_device *)ptr;

+ if (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 7de3006..f877f3b 100644
--- a/net/econet/af_econet.c
+++ b/net/econet/af_econet.c
@@ -1122,6 +1122,9 @@ 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 (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 bde1297..a11e7a5 100644
--- a/net/ipv4/arp.c
+++ b/net/ipv4/arp.c
@@ -1205,6 +1205,9 @@ static int arp_netdev_event(struct notifier_block *this, unsigned long
event, vo
{
struct net_device *dev = ptr;

+ if (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 5b77bda..9996b57 100644
--- a/net/ipv4/devinet.c
+++ b/net/ipv4/devinet.c
@@ -1051,6 +1051,9 @@ 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 (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 eff6bce..cefb55e 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, vo
struct net_device *dev = ptr;
struct in_device *in_dev = __in_dev_get_rtnl(dev);

+ if (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 35683e1..0365988 100644
--- a/net/ipv4/ipmr.c
+++ b/net/ipv4/ipmr.c
@@ -1083,13 +1083,18 @@ 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 (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/netfilter/ip_queue.c b/net/ipv4/netfilter/ip_queue.c
index cb5e61a..d918560 100644
--- a/net/ipv4/netfilter/ip_queue.c
+++ b/net/ipv4/netfilter/ip_queue.c
@@ -557,6 +557,9 @@ ipq_rcv_dev_event(struct notifier_block *this,
{
+ struct net_device *dev = ptr;

+ if (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 7c4e4be..3e0b562 100644
--- a/net/ipv4/netfilter/ipt_MASQUERADE.c
+++ b/net/ipv4/netfilter/ipt_MASQUERADE.c
@@ -125,6 +125,9 @@ static int masq_device_event(struct notifier_block *this,
{
+ const struct net_device *dev = ptr;

+ if (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 5e62ee5..5b191a3 100644
--- a/net/ipv6/addrconf.c
+++ b/net/ipv6/addrconf.c
@@ -2259,6 +2259,9 @@ static int addrconf_notify(struct notifier_block *this, unsigned long
event,
    int run_pending = 0;
    int err;

+ if (dev->nd_net != &init_net)
+ return NOTIFY_DONE;
+
    switch(event) {
    case NETDEV_REGISTER:
        if (!idev && dev->mtu >= IPV6_MIN_MTU) {
diff --git a/net/ipv6/ndisc.c b/net/ipv6/ndisc.c
index 0358e60..4d03d35 100644
--- a/net/ipv6/ndisc.c
+++ b/net/ipv6/ndisc.c
@@ -1524,6 +1524,9 @@ static int ndisc_netdev_event(struct notifier_block *this, unsigned long
event,
{
    struct net_device *dev = ptr;

+ if (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 dfc58fb..64536a3 100644
--- a/net/ipv6/netfilter/ip6_queue.c
+++ b/net/ipv6/netfilter/ip6_queue.c
@@ -547,6 +547,9 @@ ipq_rcv_dev_event(struct notifier_block *this,
{
    struct net_device *dev = ptr;

+ if (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 f7b4d38..24921f1 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 (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 bb65a38..5a8e8ff 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 (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 e969d1b..3a4d479 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 (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 51b0d5c..30f2143 100644
--- a/net/packet/af_packet.c
+++ b/net/packet/af_packet.c
@@ -1478,6 +1478,9 @@ static int packet_notifier(struct notifier_block *this, unsigned long msg,

```

```

void
    struct hlist_node *node;
    struct net_device *dev = data;

+ if (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 67e06ab..509defe 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 (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 d2ed237..406f0d2 100644
--- a/net/tipc/eth_media.c
+++ b/net/tipc/eth_media.c
@@ -198,6 +198,9 @@ 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 (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/x25/af_x25.c b/net/x25/af_x25.c
index 2e99315..fc416f9 100644
--- a/net/x25/af_x25.c
+++ b/net/x25/af_x25.c
@@ -191,6 +191,9 @@ static int x25_device_event(struct notifier_block *this, unsigned long
event,
    struct net_device *dev = ptr;
    struct x25_neigh *nb;

```

```

+ if (dev->nd_net != &init_net)
+ return NOTIFY_DONE;
+
+ if (dev->type == ARPHRD_X25
+ #if defined(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 6ab81b1..bc316bc 100644
--- a/net/xfrm/xfrm_policy.c
+++ b/net/xfrm/xfrm_policy.c
@@ -2358,6 +2358,11 @@ 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 (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..e87ab94 100644
--- a/security/selinux/netif.c
+++ b/security/selinux/netif.c
@@ -20,6 +20,7 @@
#include <linux/notifier.h>
#include <linux/netdevice.h>
#include <linux/rcupdate.h>
+#include <net/net_namespace.h>

#include "security.h"
#include "objsec.h"
@@ -234,6 +235,9 @@ static int sel_netif_netdev_notifier_handler(struct notifier_block *this,
{
+ struct net_device *dev = ptr;

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

--
1.5.3.rc6.17.g1911

```

Containers mailing list
Containers@lists.linux-foundation.org
<https://lists.linux-foundation.org/mailman/listinfo/containers>
