

---

Subject: [PATCH RFC 12/31] net: Make packet reception network namespace safe  
Posted by [ebiederm](#) on Thu, 25 Jan 2007 19:00:14 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

From: Eric W. Biederman <[ebiederm@xmission.com](mailto:ebiederm@xmission.com)> - unquoted

This patch modifies every packet receive function registered with `dev_add_pack()` to drop packets if they are not from the initial network namespace, in addition to ensure consistency of argument passing the unnecessary device parameter is removed.

This should ensure that the various network stacks do not receive packets in anything but the initial network namespace until the code has been converted and is ready for them.

Anything I may have missed will generate a compiler error, as the function prototype has changed, preventing us from overlooking something by accident.

Signed-off-by: Eric W. Biederman <[ebiederm@xmission.com](mailto:ebiederm@xmission.com)>

---

```
drivers/block/aoe/aoenet.c    | 7 ++++++-
drivers/net/bonding/bond_3ad.c | 7 ++++++-
drivers/net/bonding/bond_3ad.h | 2 +-
drivers/net/bonding/bond_alb.c | 6 +++++-
drivers/net/bonding/bond_main.c | 6 +++++-
drivers/net/hamradio/bpqether.c | 8 ++++++-
drivers/net/pppoe.c           | 8 ++++++-
drivers/net/wan/hdlc.c        | 10 ++++++++-
drivers/net/wan/lapbether.c   | 6 +++++-
drivers/net/wan/syncppp.c     | 14 +++++++-----
include/linux/netdevice.h     | 1 -
include/net/ax25.h            | 2 +-
include/net/datalink.h        | 2 +-
include/net/ip.h              | 2 +-
include/net/ipv6.h            | 1 -
include/net/lc.h              | 4 +---
include/net/p8022.h           | 1 -
include/net/psnap.h           | 2 +-
include/net/x25.h             | 2 +-
net/802/p8022.c               | 1 -
net/802/psnap.c               | 5 +----
net/8021q/vlan.h              | 2 +-
net/8021q/vlan_dev.c          | 8 ++++++-
net/appletalk/aarp.c          | 6 +++++-
net/appletalk/ddp.c           | 15 +++++++-----
```

```

net/ax25/ax25_in.c      | 8 ++++++--
net/bridge/br_private.h | 2 +-
net/bridge/br_stp_bpdu.c | 8 ++++++--
net/core/dev.c          | 6 +++---
net/deccnet/af_deccnet.c | 2 +-
net/deccnet/dn_route.c  | 6 +++++-
net/econet/af_econet.c  | 6 +++++-
net/ipv4/arp.c          | 6 +++++-
net/ipv4/ip_input.c     | 7 +++++--
net/ipv4/ipconfig.c     | 16 ++++++-----
net/ipv6/ip6_input.c    | 8 ++++++--
net/ipx/af_ipx.c        | 6 +++++-
net/irda/irlap_frame.c  | 7 +++++--
net/irda/irmod.c        | 2 +-
net/lc/lc_core.c        | 1 -
net/lc/lc_input.c       | 10 ++++++---
net/packet/af_packet.c  | 18 ++++++-----
net/tipc/eth_media.c    | 9 ++++++--
net/x25/x25_dev.c       | 6 +++++-
44 files changed, 195 insertions(+), 67 deletions(-)

```

```
diff --git a/drivers/block/aoe/aoenet.c b/drivers/block/aoe/aoenet.c
```

```
index 9626e0f..9b72a58 100644
```

```
--- a/drivers/block/aoe/aoenet.c
```

```
+++ b/drivers/block/aoe/aoenet.c
```

```
@@ -8,6 +8,7 @@
```

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

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

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

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

```
#include "aoe.h"
```

```
#define NECODES 5
```

```
@@ -108,11 +109,15 @@ aoenet_xmit(struct sk_buff *sl)
```

```
* (1) len doesn't include the header by default. I want this.
```

```
*/
```

```
static int
```

```
-aoenet_rcv(struct sk_buff *skb, struct net_device *ifp, struct packet_type *pt, struct net_device
*orig_dev)
```

```
+aoenet_rcv(struct sk_buff *skb, struct packet_type *pt, struct net_device *orig_dev)
```

```
{
```

```
+ struct net_device *ifp = skb->dev;
```

```
struct aoe_hdr *h;
```

```
u32 n;
```

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

```
+ goto exit;
```

```
+
```

```

    skb = skb_share_check(skb, GFP_ATOMIC);
    if (skb == NULL)
        return 0;
diff --git a/drivers/net/bonding/bond_3ad.c b/drivers/net/bonding/bond_3ad.c
index 3fb354d..eea4f11 100644
--- a/drivers/net/bonding/bond_3ad.c
+++ b/drivers/net/bonding/bond_3ad.c
@@ -29,6 +29,7 @@
#include <linux/ethtool.h>
#include <linux/if_bonding.h>
#include <linux/pkt_sched.h>
+#include <net/net_namespace.h>
#include "bonding.h"
#include "bond_3ad.h"

@@ -2443,12 +2444,16 @@ out:
    return 0;
}

-int bond_3ad_lacpdu_rcv(struct sk_buff *skb, struct net_device *dev, struct packet_type* ptype,
struct net_device *orig_dev)
+int bond_3ad_lacpdu_rcv(struct sk_buff *skb, struct packet_type* ptype, struct net_device
*orig_dev)
{
+ struct net_device *dev = skb->dev;
    struct bonding *bond = dev->priv;
    struct slave *slave = NULL;
    int ret = NET_RX_DROP;

+ if (!net_eq(skb->dev->nd_net, init_net()))
+     goto out;
+
    if (!(dev->flags & IFF_MASTER))
        goto out;

diff --git a/drivers/net/bonding/bond_3ad.h b/drivers/net/bonding/bond_3ad.h
index 6ad5ad6..1f2d7d2 100644
--- a/drivers/net/bonding/bond_3ad.h
+++ b/drivers/net/bonding/bond_3ad.h
@@ -282,7 +282,7 @@
void bond_3ad_adapter_duplex_changed(struct slave *slave);
void bond_3ad_handle_link_change(struct slave *slave, char link);
int bond_3ad_get_active_agg_info(struct bonding *bond, struct ad_info *ad_info);
int bond_3ad_xmit_xor(struct sk_buff *skb, struct net_device *dev);
-int bond_3ad_lacpdu_rcv(struct sk_buff *skb, struct net_device *dev, struct packet_type* ptype,
struct net_device *orig_dev);
+int bond_3ad_lacpdu_rcv(struct sk_buff *skb, struct packet_type* ptype, struct net_device
*orig_dev);
int bond_3ad_set_carrier(struct bonding *bond);

```

```
#endif // __BOND_3AD_H__
```

```
diff --git a/drivers/net/bonding/bond_alb.c b/drivers/net/bonding/bond_alb.c
```

```
index 3292316..be780a8 100644
```

```
--- a/drivers/net/bonding/bond_alb.c
```

```
+++ b/drivers/net/bonding/bond_alb.c
```

```
@@ -336,12 +336,16 @@ static void rlb_update_entry_from_arp(struct bonding *bond, struct  
arp_pkt *arp)  
    _unlock_rx_hashtbl(bond);  
}
```

```
-static int rlb_arp_rcv(struct sk_buff *skb, struct net_device *bond_dev, struct packet_type  
*ptype, struct net_device *orig_dev)
```

```
+static int rlb_arp_rcv(struct sk_buff *skb, struct packet_type *ptype, struct net_device *orig_dev)  
{  
+ struct net_device *bond_dev = skb->dev;  
+ struct bonding *bond = bond_dev->priv;  
+ struct arp_pkt *arp = (struct arp_pkt *)skb->data;  
+ int res = NET_RX_DROP;
```

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

```
+ goto out;
```

```
+
```

```
if (!(bond_dev->flags & IFF_MASTER))  
goto out;
```

```
diff --git a/drivers/net/bonding/bond_main.c b/drivers/net/bonding/bond_main.c
```

```
index 9b3bf4e..9c70568 100644
```

```
--- a/drivers/net/bonding/bond_main.c
```

```
+++ b/drivers/net/bonding/bond_main.c
```

```
@@ -2475,14 +2475,18 @@ static void bond_validate_arp(struct bonding *bond, struct slave  
*slave, u32 sip  
}  
}
```

```
-static int bond_arp_rcv(struct sk_buff *skb, struct net_device *dev, struct packet_type *pt, struct  
net_device *orig_dev)
```

```
+static int bond_arp_rcv(struct sk_buff *skb, struct packet_type *pt, struct net_device *orig_dev)  
{  
+ struct net_device *dev = skb->dev;  
+ struct arphdr *arp;  
+ struct slave *slave;  
+ struct bonding *bond;  
+ unsigned char *arp_ptr;  
+ u32 sip, tip;
```

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

```
+ goto out;
```

```

+
if (!(dev->priv_flags & IFF_BONDING) || !(dev->flags & IFF_MASTER))
    goto out;

diff --git a/drivers/net/hamradio/bpqether.c b/drivers/net/hamradio/bpqether.c
index 9fc92ad..c513e90 100644
--- a/drivers/net/hamradio/bpqether.c
+++ b/drivers/net/hamradio/bpqether.c
@@ -93,7 +93,7 @@ static char bcast_addr[6]={0xFF,0xFF,0xFF,0xFF,0xFF,0xFF};

static char bpq_eth_addr[6];

-static int bpq_rcv(struct sk_buff *, struct net_device *, struct packet_type *, struct net_device *);
+static int bpq_rcv(struct sk_buff *, struct packet_type *, struct net_device *);
static int bpq_device_event(struct notifier_block *, unsigned long, void *);
static const char *bpq_print_ethaddr(const unsigned char *);

@@ -166,13 +166,17 @@ static inline int dev_is_ethdev(struct net_device *dev)
/*
 * Receive an AX.25 frame via an ethernet interface.
 */
-static int bpq_rcv(struct sk_buff *skb, struct net_device *dev, struct packet_type *ptype, struct
net_device *orig_dev)
+static int bpq_rcv(struct sk_buff *skb, struct packet_type *ptype, struct net_device *orig_dev)
{
+ struct net_device *dev = skb->dev;
    int len;
    char * ptr;
    struct ethhdr *eth;
    struct bpqdev *bpq;

+ if (!net_eq(skb->dev->nd_net, init_net()))
+     goto drop;
+
    if ((skb = skb_share_check(skb, GFP_ATOMIC)) == NULL)
        return NET_RX_DROP;

diff --git a/drivers/net/pppoe.c b/drivers/net/pppoe.c
index d09334d..caf8ca3 100644
--- a/drivers/net/pppoe.c
+++ b/drivers/net/pppoe.c
@@ -376,7 +376,6 @@ abort_kfree:
 *
*****/
static int pppoe_rcv(struct sk_buff *skb,
- struct net_device *dev,
  struct packet_type *pt,
  struct net_device *orig_dev)

```

```
@@ -384,6 +383,9 @@ static int pppoe_rcv(struct sk_buff *skb,
    struct pppoe_hdr *ph;
    struct pppox_sock *po;
```

```
+ if (!net_eq(skb->dev->nd_net, init_net()))
+ goto drop;
+
+ if (!pskb_may_pull(skb, sizeof(struct pppoe_hdr)))
+ goto drop;
```

```
@@ -408,7 +410,6 @@ out:
```

```
*
```

```
*****/
```

```
static int pppoe_disc_rcv(struct sk_buff *skb,
- struct net_device *dev,
  struct packet_type *pt,
  struct net_device *orig_dev)
```

```
@@ -416,6 +417,9 @@ static int pppoe_disc_rcv(struct sk_buff *skb,
    struct pppoe_hdr *ph;
    struct pppox_sock *po;
```

```
+ if (!net_eq(skb->dev->nd_net, init_net()))
+ goto abort;
+
+ if (!pskb_may_pull(skb, sizeof(struct pppoe_hdr)))
+ goto abort;
```

```
diff --git a/drivers/net/wan/hdlc.c b/drivers/net/wan/hdlc.c
```

```
index db354e0..f3bf160 100644
```

```
--- a/drivers/net/wan/hdlc.c
```

```
+++ b/drivers/net/wan/hdlc.c
```

```
@@ -36,6 +36,7 @@
```

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

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

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

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

```
static const char* version = "HDLC support module revision 1.20";
```

```
@@ -62,10 +63,17 @@ static struct net_device_stats *hdlc_get_stats(struct net_device *dev)
```

```
-static int hdlc_rcv(struct sk_buff *skb, struct net_device *dev,
+static int hdlc_rcv(struct sk_buff *skb,
    struct packet_type *p, struct net_device *orig_dev)
```

```

{
+ struct net_device *dev = skb->dev;
  struct hdlc_device_desc *desc = dev_to_desc(dev);
+
+ if (!net_eq(skb->dev->nd_net, init_net())) {
+   kfree_skb(skb);
+   return 0;
+ }
+
  if (desc->netif_rx)
    return desc->netif_rx(skb);

```

diff --git a/drivers/net/wan/lapbether.c b/drivers/net/wan/lapbether.c

index 6c302e9..c1de21e 100644

--- a/drivers/net/wan/lapbether.c

+++ b/drivers/net/wan/lapbether.c

@@ -86,11 +86,15 @@ static \_\_inline\_\_ int dev\_is\_ethdev(struct net\_device \*dev)

/\*

\* Receive a LAPB frame via an ethernet interface.

\*/

-static int lapbeth\_rcv(struct sk\_buff \*skb, struct net\_device \*dev, struct packet\_type \*ptype, struct net\_device \*orig\_dev)

+static int lapbeth\_rcv(struct sk\_buff \*skb, struct packet\_type \*ptype, struct net\_device \*orig\_dev)

{

+ struct net\_device \*dev = skb->dev;

int len, err;

struct lapbethdev \*lapbeth;

+ if (!net\_eq(skb->dev->nd\_net, init\_net()))

+ goto drop;

+

if ((skb = skb\_share\_check(skb, GFP\_ATOMIC)) == NULL)

return NET\_RX\_DROP;

diff --git a/drivers/net/wan/syncppp.c b/drivers/net/wan/syncppp.c

index 218f7b5..25137fc 100644

--- a/drivers/net/wan/syncppp.c

+++ b/drivers/net/wan/syncppp.c

@@ -51,6 +51,7 @@

#include <linux/spinlock.h>

#include <linux/rcupdate.h>

+#include <net/net\_namespace.h>

#include <net/syncppp.h>

#include <asm/byteorder.h>

@@ -220,13 +221,13 @@ static void sPPP\_clear\_timeout(struct sPPP \*p)

\* here.

```

*/

-static void sPPP_input (struct net_device *dev, struct sk_buff *skb)
+static void sPPP_input (struct sk_buff *skb)
{
+ struct net_device *dev = skb->dev;
  struct PPP_header *h;
  struct sPPP *sp = (struct sPPP *)sPPP_of(dev);
  unsigned long flags;

- skb->dev=dev;
  skb->mac.raw=skb->data;

  if (dev->flags & IFF_RUNNING)
@@ -1443,11 +1444,16 @@ static void sPPP_print_bytes (u_char *p, u16 len)
  * after interrupt servicing to process frames queued via netif_rx.
*/

-static int sPPP_rcv(struct sk_buff *skb, struct net_device *dev, struct packet_type *p, struct
net_device *orig_dev)
+static int sPPP_rcv(struct sk_buff *skb, struct packet_type *p, struct net_device *orig_dev)
{
+ if (!net_eq(skb->dev->nd_net, init_net())) {
+ kfree_skb(skb);
+ return 0;
+ }
+
  if ((skb = skb_share_check(skb, GFP_ATOMIC)) == NULL)
    return NET_RX_DROP;
- sPPP_input(dev,skb);
+ sPPP_input(skb);
  return 0;
}

diff --git a/include/linux/netdevice.h b/include/linux/netdevice.h
index 6a1579d..9e28671 100644
--- a/include/linux/netdevice.h
+++ b/include/linux/netdevice.h
@@ -558,7 +558,6 @@ struct packet_type {
  __be16 type; /* This is really htons(ether_type). */
  struct net_device *dev; /* NULL is wildcarded here */
  int (*func) (struct sk_buff *,
- struct net_device *,
  struct packet_type *,
  struct net_device *);
  struct sk_buff *(*gso_segment)(struct sk_buff *skb,
diff --git a/include/net/ax25.h b/include/net/ax25.h
index 5ae10dd..a2ad59a 100644

```



```

--- a/include/net/ax25.h
+++ b/include/net/ax25.h
@@ -360,7 +360,7 @@ extern int ax25_protocol_is_registered(unsigned int);

/* ax25_in.c */
extern int ax25_rx_iframe(ax25_cb *, struct sk_buff *);
-extern int ax25_kiss_rcv(struct sk_buff *, struct net_device *, struct packet_type *, struct
net_device *);
+extern int ax25_kiss_rcv(struct sk_buff *, struct packet_type *, struct net_device *);

/* ax25_ip.c */
extern int ax25_hard_header(struct sk_buff *, struct net_device *, unsigned short, void *, void *,
unsigned int);
diff --git a/include/net/datalink.h b/include/net/datalink.h
index deb7ca7..133d55e 100644
--- a/include/net/datalink.h
+++ b/include/net/datalink.h
@@ -8,7 +8,7 @@ struct datalink_proto {

    unsigned short header_length;

-   int (*rcvfunc)(struct sk_buff *, struct net_device *,
+   int (*rcvfunc)(struct sk_buff *,
                        struct packet_type *, struct net_device *);
    int (*request)(struct datalink_proto *, struct sk_buff *,
                    unsigned char *);
diff --git a/include/net/ip.h b/include/net/ip.h
index 053f02b..c0c0dfd 100644
--- a/include/net/ip.h
+++ b/include/net/ip.h
@@ -88,7 +88,7 @@ extern int igmp_mc_proc_init(void);
extern int ip_build_and_send_pkt(struct sk_buff *skb, struct sock *sk,
    __be32 saddr, __be32 daddr,
    struct ip_options *opt);
-extern int ip_rcv(struct sk_buff *skb, struct net_device *dev,
+extern int ip_rcv(struct sk_buff *skb,
    struct packet_type *pt, struct net_device *orig_dev);
extern int ip_local_deliver(struct sk_buff *skb);
extern int ip_mr_input(struct sk_buff *skb);
diff --git a/include/net/ipv6.h b/include/net/ipv6.h
index 00328b7..0b1d1a9 100644
--- a/include/net/ipv6.h
+++ b/include/net/ipv6.h
@@ -438,7 +438,6 @@ static inline int ipv6_addr_diff(const struct in6_addr *a1, const struct
in6_addr
    */

extern int ipv6_rcv(struct sk_buff *skb,

```

```
- struct net_device *dev,
  struct packet_type *pt,
  struct net_device *orig_dev);
```

```
diff --git a/include/net/llc.h b/include/net/llc.h
```

```
index f502458..dae09b9 100644
```

```
--- a/include/net/llc.h
```

```
+++ b/include/net/llc.h
```

```
@@ -48,7 +48,6 @@ struct llc_sap {
```

```
  unsigned char f_bit;
  atomic_t      refcnt;
  int  (*rcv_func)(struct sk_buff *skb,
-      struct net_device *dev,
        struct packet_type *pt,
        struct net_device *orig_dev);
  struct llc_addr laddr;
```

```
@@ -67,7 +66,7 @@ extern struct list_head llc_sap_list;
```

```
extern rwlock_t llc_sap_list_lock;
```

```
extern unsigned char llc_station_mac_sa[ETH_ALEN];
```

```
-extern int llc_rcv(struct sk_buff *skb, struct net_device *dev,
```

```
+extern int llc_rcv(struct sk_buff *skb,
  struct packet_type *pt, struct net_device *orig_dev);
```

```
extern int llc_mac_hdr_init(struct sk_buff *skb,
```

```
@@ -81,7 +80,6 @@ extern void llc_set_station_handler(void (*handler)(struct sk_buff *skb));
```

```
extern struct llc_sap *llc_sap_open(unsigned char lsap,
```

```
  int (*rcv)(struct sk_buff *skb,
-      struct net_device *dev,
        struct packet_type *pt,
        struct net_device *orig_dev));
```

```
static inline void llc_sap_hold(struct llc_sap *sap)
```

```
diff --git a/include/net/p8022.h b/include/net/p8022.h
```

```
index 42e9fac..545c15e 100644
```

```
--- a/include/net/p8022.h
```

```
+++ b/include/net/p8022.h
```

```
@@ -3,7 +3,6 @@
```

```
extern struct datalink_proto *
  register_8022_client(unsigned char type,
  int (*func)(struct sk_buff *skb,
-      struct net_device *dev,
        struct packet_type *pt,
        struct net_device *orig_dev));
```

```
extern void unregister_8022_client(struct datalink_proto *proto);
```

```
diff --git a/include/net/psnap.h b/include/net/psnap.h
```

```
index b2e01cc..e935d50 100644
```

```
--- a/include/net/psnap.h
```

```

+++ b/include/net/psnap.h
@@ -1,7 +1,7 @@
#ifdef _NET_PSNAP_H
#define _NET_PSNAP_H

-extern struct datalink_proto *register_snap_client(unsigned char *desc, int (*rcvfunc)(struct
sk_buff *, struct net_device *, struct packet_type *, struct net_device *orig_dev));
+extern struct datalink_proto *register_snap_client(unsigned char *desc, int (*rcvfunc)(struct
sk_buff *, struct packet_type *, struct net_device *orig_dev));
extern void unregister_snap_client(struct datalink_proto *proto);

#endif
diff --git a/include/net/x25.h b/include/net/x25.h
index e47fe44..e3d4cfb 100644
--- a/include/net/x25.h
+++ b/include/net/x25.h
@@ -184,7 +184,7 @@ extern void x25_kill_by_neigh(struct x25_neigh *);

/* x25_dev.c */
extern void x25_send_frame(struct sk_buff *, struct x25_neigh *);
-extern int x25_lapb_receive_frame(struct sk_buff *, struct net_device *, struct packet_type *,
struct net_device *);
+extern int x25_lapb_receive_frame(struct sk_buff *, struct packet_type *, struct net_device *);
extern void x25_establish_link(struct x25_neigh *);
extern void x25_terminate_link(struct x25_neigh *);

diff --git a/net/802/p8022.c b/net/802/p8022.c
index 2530f35..1c7022d 100644
--- a/net/802/p8022.c
+++ b/net/802/p8022.c
@@ -34,7 +34,6 @@ static int p8022_request(struct datalink_proto *dl, struct sk_buff *skb,

struct datalink_proto *register_8022_client(unsigned char type,
int (*func)(struct sk_buff *skb,
- struct net_device *dev,
struct packet_type *pt,
struct net_device *orig_dev))
{
diff --git a/net/802/psnap.c b/net/802/psnap.c
index 270b9d2..59ac0c5 100644
--- a/net/802/psnap.c
+++ b/net/802/psnap.c
@@ -46,7 +46,7 @@ static struct datalink_proto *find_snap_client(unsigned char *desc)
/*
 * A SNAP packet has arrived
 */
-static int snap_rcv(struct sk_buff *skb, struct net_device *dev,
+static int snap_rcv(struct sk_buff *skb,

```

```

    struct packet_type *pt, struct net_device *orig_dev)
{
    int rc = 1;
@@ -61,7 +61,7 @@ static int snap_rcv(struct sk_buff *skb, struct net_device *dev,
    /* Pass the frame on. */
    skb->h.raw += 5;
    skb_pull_rcsum(skb, 5);
- rc = proto->rcvfunc(skb, dev, &snap_packet_type, orig_dev);
+ rc = proto->rcvfunc(skb, &snap_packet_type, orig_dev);
} else {
    skb->sk = NULL;
    kfree_skb(skb);
@@ -117,7 +117,6 @@ module_exit(snap_exit);
*/

struct datalink_proto *register_snap_client(unsigned char *desc,
    int (*rcvfunc)(struct sk_buff *,
-    struct net_device *,
    struct packet_type *,
    struct net_device *))
{
diff --git a/net/8021q/vlan.h b/net/8021q/vlan.h
index 9ae3a14..9207999 100644
--- a/net/8021q/vlan.h
+++ b/net/8021q/vlan.h
@@ -50,7 +50,7 @@ struct net_device *__find_vlan_dev(struct net_device* real_dev,

/* found in vlan_dev.c */
int vlan_dev_rebuild_header(struct sk_buff *skb);
-int vlan_skb_rcv(struct sk_buff *skb, struct net_device *dev,
+int vlan_skb_rcv(struct sk_buff *skb,
    struct packet_type *ptype, struct net_device *orig_dev);
int vlan_dev_hard_header(struct sk_buff *skb, struct net_device *dev,
    unsigned short type, void *daddr, void *saddr,
diff --git a/net/8021q/vlan_dev.c b/net/8021q/vlan_dev.c
index 60a508e..9fce3a8 100644
--- a/net/8021q/vlan_dev.c
+++ b/net/8021q/vlan_dev.c
@@ -112,9 +112,10 @@ static inline struct sk_buff *vlan_check_reorder_header(struct sk_buff
*skb)
*
*      been commented out now... --Ben
*
*/
-int vlan_skb_rcv(struct sk_buff *skb, struct net_device *dev,
+int vlan_skb_rcv(struct sk_buff *skb,
    struct packet_type * ptype, struct net_device *orig_dev)
{
+ struct net_device *dev = skb->dev;
    unsigned char *rawp = NULL;

```

```

    struct vlan_hdr *vhdr = (struct vlan_hdr *) (skb->data);
    unsigned short vid;
@@ -122,6 +123,11 @@ int vlan_skb_rcv(struct sk_buff *skb, struct net_device *dev,
    unsigned short vlan_TCI;
    __be16 proto;

+ if (!net_eq(skb->dev->nd_net, init_net())) {
+ kfree_skb(skb);
+ return 0;
+ }
+
/* vlan_TCI = ntohs(get_unaligned(&vhdr->h_vlan_TCI)); */
vlan_TCI = ntohs(vhdr->h_vlan_TCI);

diff --git a/net/appletalk/aarp.c b/net/appletalk/aarp.c
index b51a010..85c4dbc 100644
--- a/net/appletalk/aarp.c
+++ b/net/appletalk/aarp.c
@@ -697,9 +697,10 @@ static void __aarp_resolved(struct aarp_entry **list, struct aarp_entry
*a,
 * This is called by the SNAP driver whenever we see an AARP SNAP
 * frame. We currently only support Ethernet.
 */
-static int aarp_rcv(struct sk_buff *skb, struct net_device *dev,
+static int aarp_rcv(struct sk_buff *skb,
    struct packet_type *pt, struct net_device *orig_dev)
{
+ struct net_device *dev = skb->dev;
    struct elapaarp *ea = aarp_hdr(skb);
    int hash, ret = 0;
    __u16 function;
@@ -707,6 +708,9 @@ static int aarp_rcv(struct sk_buff *skb, struct net_device *dev,
    struct atalk_addr sa, *ma, da;
    struct atalk_iface *ifa;

+ if (!net_eq(skb->dev->nd_net, init_net()))
+ goto out0;
+
/* We only do Ethernet SNAP AARP. */
if (dev->type != ARPHRD_ETHER)
    goto out0;
diff --git a/net/appletalk/ddp.c b/net/appletalk/ddp.c
index e08367b..f4ff8aa 100644
--- a/net/appletalk/ddp.c
+++ b/net/appletalk/ddp.c
@@ -1393,9 +1393,10 @@ free_it:
 * extracted. PPP should probably pass frames marked as for this layer.
 * [ie ARPHRD_ETHERTALK]

```

```

*/
-static int atalk_rcv(struct sk_buff *skb, struct net_device *dev,
+static int atalk_rcv(struct sk_buff *skb,
    struct packet_type *pt, struct net_device *orig_dev)
{
+ struct net_device *dev = skb->dev;
    struct ddpehdr *ddp;
    struct sock *sock;
    struct atalk_iface *atif;
@@ -1403,6 +1404,9 @@ static int atalk_rcv(struct sk_buff *skb, struct net_device *dev,
    int origlen;
    __u16 len_hops;

+ if (!net_eq(skb->dev->nd_net, init_net()))
+ goto freeit;
+
    /* Don't mangle buffer if shared */
    if (!(skb = skb_share_check(skb, GFP_ATOMIC)))
        goto out;
@@ -1482,9 +1486,14 @@ freeit:
    * Caller must provide enough headroom on the packet to pull the short
    * header and append a long one.
    */
-static int ltalk_rcv(struct sk_buff *skb, struct net_device *dev,
+static int ltalk_rcv(struct sk_buff *skb,
    struct packet_type *pt, struct net_device *orig_dev)
{
+ struct net_device *dev = skb->dev;
+
+ if (!net_eq(skb->dev->nd_net, init_net()))
+ goto freeit;
+
    /* Expand any short form frames */
    if (skb->mac.raw[2] == 1) {
        struct ddpehdr *ddp;
@@ -1526,7 +1535,7 @@ static int ltalk_rcv(struct sk_buff *skb, struct net_device *dev,
    }
    skb->h.raw = skb->data;

- return atalk_rcv(skb, dev, pt, orig_dev);
+ return atalk_rcv(skb, pt, orig_dev);
freeit:
    kfree_skb(skb);
    return 0;
diff --git a/net/ax25/ax25_in.c b/net/ax25/ax25_in.c
index e9d9429..8c9b0dd 100644
--- a/net/ax25/ax25_in.c
+++ b/net/ax25/ax25_in.c

```

```

@@ -444,12 +444,18 @@ static int ax25_rcv(struct sk_buff *skb, struct net_device *dev,
/*
 * Receive an AX.25 frame via a SLIP interface.
 */
-int ax25_kiss_rcv(struct sk_buff *skb, struct net_device *dev,
+int ax25_kiss_rcv(struct sk_buff *skb,
+    struct packet_type *ptype, struct net_device *orig_dev)
{
+ struct net_device *dev = skb->dev;
+    skb->sk = NULL; /* Initially we don't know who it's for */
+    skb->destructor = NULL; /* Who initializes this, dammit?! */

+ if (!net_eq(skb->dev->nd_net, init_net())) {
+     kfree_skb(skb);
+     return 0;
+ }
+
+ if ((*skb->data & 0x0F) != 0) {
+     kfree_skb(skb); /* Not a KISS data frame */
+     return 0;
diff --git a/net/bridge/br_private.h b/net/bridge/br_private.h
index 3a534e9..f1712b9 100644
--- a/net/bridge/br_private.h
+++ b/net/bridge/br_private.h
@@ -223,7 +223,7 @@ extern void br_stp_set_path_cost(struct net_bridge_port *p,
extern ssize_t br_show_bridge_id(char *buf, const struct bridge_id *id);

/* br_stp_bpdu.c */
-extern int br_stp_rcv(struct sk_buff *skb, struct net_device *dev,
+extern int br_stp_rcv(struct sk_buff *skb,
+    struct packet_type *pt, struct net_device *orig_dev);

/* br_stp_timer.c */
diff --git a/net/bridge/br_stp_bpdu.c b/net/bridge/br_stp_bpdu.c
index 068d8af..7f9f8b4 100644
--- a/net/bridge/br_stp_bpdu.c
+++ b/net/bridge/br_stp_bpdu.c
@@ -17,6 +17,7 @@
@@ -17,6 +17,7 @@
#include <linux/netfilter_bridge.h>
#include <linux/etherdevice.h>
#include <linux/llc.h>
+#include <net/net_namespace.h>
#include <net/llc.h>
#include <net/llc_pdu.h>
#include <asm/unaligned.h>
@@ -129,15 +130,18 @@ void br_send_tcn_bpdu(struct net_bridge_port *p)
*
 * NO locks, but rcu_read_lock (preempt_disabled)

```

```

*/
-int br_stp_rcv(struct sk_buff *skb, struct net_device *dev,
+int br_stp_rcv(struct sk_buff *skb,
    struct packet_type *pt, struct net_device *orig_dev)
{
    const struct llc_pdu_un *pdu = llc_pdu_un_hdr(skb);
    const unsigned char *dest = eth_hdr(skb)->h_dest;
- struct net_bridge_port *p = rcu_dereference(dev->br_port);
+ struct net_bridge_port *p = rcu_dereference(skb->dev->br_port);
    struct net_bridge *br;
    const unsigned char *buf;

+ if (!net_eq(skb->dev->nd_net, init_net()))
+ goto err;
+
    if (!p)
        goto err;

diff --git a/net/core/dev.c b/net/core/dev.c
index a3ee150..d8aa534 100644
--- a/net/core/dev.c
+++ b/net/core/dev.c
@@ -1094,7 +1094,7 @@ static void dev_queue_xmit_nit(struct sk_buff *skb, struct net_device
 *dev)

    skb2->h.raw = skb2->nh.raw;
    skb2->pkt_type = PACKET_OUTGOING;
- ptype->func(skb2, skb->dev, ptype, skb->dev);
+ ptype->func(skb2, ptype, skb->dev);
}
}
rcu_read_unlock();
@@ -1693,7 +1693,7 @@ static __inline__ int deliver_skb(struct sk_buff *skb,
    struct net_device *orig_dev)
{
    atomic_inc(&skb->users);
- return pt_prev->func(skb, skb->dev, pt_prev, orig_dev);
+ return pt_prev->func(skb, pt_prev, orig_dev);
}

#ifdef CONFIG_BRIDGE || defined (CONFIG_BRIDGE_MODULE)
@@ -1841,7 +1841,7 @@ ncls:
}

if (pt_prev) {
- ret = pt_prev->func(skb, skb->dev, pt_prev, orig_dev);
+ ret = pt_prev->func(skb, pt_prev, orig_dev);
} else {

```



```

    kfree_skb(skb);
    /* Jamal, now you will not be able to escape explaining
diff --git a/net/decnet/af_decnet.c b/net/decnet/af_decnet.c
index f1553fa..5e8042f 100644
--- a/net/decnet/af_decnet.c
+++ b/net/decnet/af_decnet.c
@@ -2104,7 +2104,7 @@ static struct notifier_block dn_dev_notifier = {
    .notifier_call = dn_device_event,
};

-extern int dn_route_rcv(struct sk_buff *, struct net_device *, struct packet_type *, struct
net_device *);
+extern int dn_route_rcv(struct sk_buff *, struct packet_type *, struct net_device *);

static struct packet_type dn_dix_packet_type = {
    .type = __constant_htons(ETH_P_DNA_RT),
diff --git a/net/decnet/dn_route.c b/net/decnet/dn_route.c
index 0d657eb..4263cd9 100644
--- a/net/decnet/dn_route.c
+++ b/net/decnet/dn_route.c
@@ -575,14 +575,18 @@ static int dn_route_ptp_hello(struct sk_buff *skb)
    return NET_RX_SUCCESS;
}

-int dn_route_rcv(struct sk_buff *skb, struct net_device *dev, struct packet_type *pt, struct
net_device *orig_dev)
+int dn_route_rcv(struct sk_buff *skb, struct packet_type *pt, struct net_device *orig_dev)
{
+ struct net_device *dev = skb->dev;
+ struct dn_skb_cb *cb;
+ unsigned char flags = 0;
+ __u16 len = dn_ntohs((__le16 *)skb->data);
+ struct dn_dev *dn = (struct dn_dev *)dev->dn_ptr;
+ unsigned char padlen = 0;

+ if (!net_eq(skb->dev->nd_net, init_net()))
+ goto dump_it;
+
+ if (dn == NULL)
+ goto dump_it;

diff --git a/net/econet/af_econet.c b/net/econet/af_econet.c
index a0b3fc5..0baffda 100644
--- a/net/econet/af_econet.c
+++ b/net/econet/af_econet.c
@@ -1057,12 +1057,16 @@ release:
    * Receive an Econet frame from a device.
    */

```

```

-static int econet_rcv(struct sk_buff *skb, struct net_device *dev, struct packet_type *pt, struct
net_device *orig_dev)
+static int econet_rcv(struct sk_buff *skb, struct packet_type *pt, struct net_device *orig_dev)
{
+ struct net_device *dev = skb->dev;
  struct ec_framehdr *hdr;
  struct sock *sk;
  struct ec_device *edev = dev->ec_ptr;

+ if (!net_eq(skb->dev->nd_net, init_net()))
+ goto drop;
+
  if (skb->pkt_type == PACKET_OTHERHOST)
    goto drop;

```

```

diff --git a/net/ipv4/arp.c b/net/ipv4/arp.c
index e3b89a7..95a34c7 100644
--- a/net/ipv4/arp.c
+++ b/net/ipv4/arp.c
@@ -928,11 +928,15 @@ static void parp_redo(struct sk_buff *skb)
 * Receive an arp request from the device layer.
 */

```

```

-static int arp_rcv(struct sk_buff *skb, struct net_device *dev,
+static int arp_rcv(struct sk_buff *skb,
  struct packet_type *pt, struct net_device *orig_dev)
{
+ struct net_device *dev = skb->dev;
  struct arphdr *arp;

+ if (!net_eq(skb->dev->nd_net, init_net()))
+ goto freeskb;
+
  /* ARP header, plus 2 device addresses, plus 2 IP addresses. */
  if (!pskb_may_pull(skb, (sizeof(struct arphdr) +
    (2 * dev->addr_len) +

```

```

diff --git a/net/ipv4/ip_input.c b/net/ipv4/ip_input.c
index 212734c..77dddce 100644
--- a/net/ipv4/ip_input.c
+++ b/net/ipv4/ip_input.c
@@ -370,11 +370,14 @@ drop:
 /*
 * Main IP Receive routine.
 */

```

```

-int ip_rcv(struct sk_buff *skb, struct net_device *dev, struct packet_type *pt, struct net_device
*orig_dev)
+int ip_rcv(struct sk_buff *skb, struct packet_type *pt, struct net_device *orig_dev)

```

```

{
    struct iphdr *iph;
    u32 len;

+ if (!net_eq(skb->dev->nd_net, init_net()))
+ goto drop;
+
    /* When the interface is in promisc. mode, drop all the crap
     * that it receives, do not try to analyse it.
     */
@@ -431,7 +434,7 @@ int ip_rcv(struct sk_buff *skb, struct net_device *dev, struct packet_type
*pt,
    /* 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,
+ return NF_HOOK(PF_INET, NF_IP_PRE_ROUTING, skb, skb->dev, NULL,
    ip_rcv_finish);

inhdr_error:
diff --git a/net/ipv4/ipconfig.c b/net/ipv4/ipconfig.c
index 8b649c5..91b5729 100644
--- a/net/ipv4/ipconfig.c
+++ b/net/ipv4/ipconfig.c
@@ -397,7 +397,7 @@ static int __init ic_defaults(void)

#ifdef IPCONFIG_RARP

-static int ic_rarp_rcv(struct sk_buff *skb, struct net_device *dev, struct packet_type *pt, struct
net_device *orig_dev);
+static int ic_rarp_rcv(struct sk_buff *skb, struct packet_type *pt, struct net_device *orig_dev);

static struct packet_type rarp_packet_type __initdata = {
    .type = __constant_htons(ETH_P_RARP),
@@ -418,14 +418,18 @@ static inline void ic_rarp_cleanup(void)
    * Process received RARP packet.
    */
static int __init
-ic_rarp_rcv(struct sk_buff *skb, struct net_device *dev, struct packet_type *pt, struct net_device
*orig_dev)
+ic_rarp_rcv(struct sk_buff *skb, struct packet_type *pt, struct net_device *orig_dev)
{
+ struct net_device *dev = skb->dev;
    struct arphdr *rarp;
    unsigned char *rarp_ptr;
    __be32 sip, tip;
    unsigned char *sha, *tha; /* s for "source", t for "target" */
    struct ic_device *d;

```

```

+ if (!net_eq(skb->dev->nd_net, init_net()))
+ goto drop;
+
+ if ((skb = skb_share_check(skb, GFP_ATOMIC)) == NULL)
+ return NET_RX_DROP;

@@ -559,7 +563,7 @@ struct bootp_pkt { /* BOOTP packet format */
#define DHCPRELEASE 7
#define DHCPINFORM 8

-static int ic_bootp_rcv(struct sk_buff *skb, struct net_device *dev, struct packet_type *pt, struct
net_device *orig_dev);
+static int ic_bootp_rcv(struct sk_buff *skb, struct packet_type *pt, struct net_device *orig_dev);

static struct packet_type bootp_packet_type __initdata = {
.type = __constant_htons(ETH_P_IP),
@@ -827,13 +831,17 @@ static void __init ic_do_bootp_ext(u8 *ext)
/*
 * Receive BOOTP reply.
 */
-static int __init ic_bootp_rcv(struct sk_buff *skb, struct net_device *dev, struct packet_type *pt,
struct net_device *orig_dev)
+static int __init ic_bootp_rcv(struct sk_buff *skb, struct packet_type *pt, struct net_device
*orig_dev)
{
+ struct net_device *dev = skb->dev;
+ struct bootp_pkt *b;
+ struct iphdr *h;
+ struct ic_device *d;
+ int len, ext_len;

+ if (!net_eq(skb->dev->nd_net, init_net()))
+ goto drop;
+
+ /* Perform verifications before taking the lock. */
+ if (skb->pkt_type == PACKET_OTHERHOST)
+ goto drop;
diff --git a/net/ipv6/ip6_input.c b/net/ipv6/ip6_input.c
index ad0b8ab..ac366b9 100644
--- a/net/ipv6/ip6_input.c
+++ b/net/ipv6/ip6_input.c
@@ -56,12 +56,18 @@ inline int ip6_rcv_finish( struct sk_buff *skb)
return dst_input(skb);
}

-int ipv6_rcv(struct sk_buff *skb, struct net_device *dev, struct packet_type *pt, struct net_device
*orig_dev)

```

```
+int ipv6_rcv(struct sk_buff *skb, struct packet_type *pt, struct net_device *orig_dev)
{
+ struct net_device *dev = skb->dev;
  struct ipv6hdr *hdr;
  u32 pkt_len;
  struct inet6_dev *idev;
```

```
+ if (!net_eq(skb->dev->nd_net, init_net())) {
+ kfree_skb(skb);
+ return 0;
+ }
```

```
+
  if (skb->pkt_type == PACKET_OTHERHOST) {
    kfree_skb(skb);
    return 0;
```

```
diff --git a/net/ipx/af_ipx.c b/net/ipx/af_ipx.c
index 2ec4a3c..5c5f2cd 100644
```

```
--- a/net/ipx/af_ipx.c
```

```
+++ b/net/ipx/af_ipx.c
```

```
@ @ -1637,14 +1637,18 @ @ out:
  return rc;
}
```

```
-static int ipx_rcv(struct sk_buff *skb, struct net_device *dev, struct packet_type *pt, struct
net_device *orig_dev)
```

```
+static int ipx_rcv(struct sk_buff *skb, struct packet_type *pt, struct net_device *orig_dev)
{
+ struct net_device *dev = skb->dev;
  /* NULL here for pt means the packet was looped back */
  struct ipx_interface *intrfc;
  struct ipxhdr *ipx;
  u16 ipx_pktsize;
  int rc = 0;
```

```
+ if (!net_eq(skb->dev->nd_net, init_net()))
+ goto drop;
+
```

```
  /* Not ours */
  if (skb->pkt_type == PACKET_OTHERHOST)
    goto drop;
```

```
diff --git a/net/irda/irlap_frame.c b/net/irda/irlap_frame.c
index dba349c..3252be7 100644
```

```
--- a/net/irda/irlap_frame.c
```

```
+++ b/net/irda/irlap_frame.c
```

```
@ @ -1306,7 +1306,7 @ @ static void irlap_rcv_test_frame(struct irlap_cb *self, struct sk_buff
*skb,
  * LMP level in irlmp.c.
  * Jean II
```

```

*/
-int irlap_driver_rcv(struct sk_buff *skb, struct net_device *dev,
+int irlap_driver_rcv(struct sk_buff *skb,
    struct packet_type *ptype, struct net_device *orig_dev)
{
    struct irlap_info info;
@@ -1314,8 +1314,11 @@ int irlap_driver_rcv(struct sk_buff *skb, struct net_device *dev,
    int command;
    __u8 control;

+ if (!net_eq(skb->dev->nd_net, init_net()))
+ goto out;
+
    /* FIXME: should we get our own field? */
- self = (struct irlap_cb *) dev->atalk_ptr;
+ self = (struct irlap_cb *) skb->dev->atalk_ptr;

    /* If the net device is down, then IrLAP is gone! */
    if (!self || self->magic != LAP_MAGIC) {
diff --git a/net/irda/irmod.c b/net/irda/irmod.c
index 2869b16..6b1989c 100644
--- a/net/irda/irmod.c
+++ b/net/irda/irmod.c
@@ -52,7 +52,7 @@ extern void irda_sysctl_unregister(void);
extern int irsock_init(void);
extern void irsock_cleanup(void);
/* irlap_frame.c */
-extern int irlap_driver_rcv(struct sk_buff *, struct net_device *,
+extern int irlap_driver_rcv(struct sk_buff *,
    struct packet_type *, struct net_device *);

/*
diff --git a/net/lc/lc_core.c b/net/lc/lc_core.c
index d12413c..f438c38 100644
--- a/net/lc/lc_core.c
+++ b/net/lc/lc_core.c
@@ -112,7 +112,6 @@ struct lc_sap *lc_sap_find(unsigned char sap_value)
*/
struct lc_sap *lc_sap_open(unsigned char lsap,
    int (*func)(struct sk_buff *skb,
- struct net_device *dev,
    struct packet_type *pt,
    struct net_device *orig_dev))
{
diff --git a/net/lc/lc_input.c b/net/lc/lc_input.c
index db82aff..cecb4a9 100644
--- a/net/lc/lc_input.c
+++ b/net/lc/lc_input.c

```

```

@@ -12,6 +12,7 @@
 * See the GNU General Public License for more details.
 */
#include <linux/netdevice.h>
+#include <net/net_namespace.h>
#include <net/llc.h>
#include <net/llc_pdu.h>
#include <net/llc_sap.h>
@@ -136,15 +137,18 @@ static inline int llc_fixup_skb(struct sk_buff *skb)
 * the frame is related to a busy connection (a connection is sending
 * data now), it queues this frame in the connection's backlog.
 */
-int llc_rcv(struct sk_buff *skb, struct net_device *dev,
+int llc_rcv(struct sk_buff *skb,
      struct packet_type *pt, struct net_device *orig_dev)
{
    struct llc_sap *sap;
    struct llc_pdu_sn *pdu;
    int dest;
- int (*rcv)(struct sk_buff *, struct net_device *,
+ int (*rcv)(struct sk_buff *,
      struct packet_type *, struct net_device *);

+ if (!net_eq(skb->dev->nd_net, init_net()))
+ goto drop;
+
    /*
     * When the interface is in promisc. mode, drop all the crap that it
     * receives, do not try to analyse it.
@@ -175,7 +179,7 @@ int llc_rcv(struct sk_buff *skb, struct net_device *dev,
    if (rcv) {
        struct sk_buff *cskb = skb_clone(skb, GFP_ATOMIC);
        if (cskb)
- rcv(cskb, dev, pt, orig_dev);
+ rcv(cskb, pt, orig_dev);
    }
    dest = llc_pdu_type(skb);
    if (unlikely(!dest || !llc_type_handlers[dest - 1]))
diff --git a/net/packet/af_packet.c b/net/packet/af_packet.c
index ca371ea..aa298c3 100644
--- a/net/packet/af_packet.c
+++ b/net/packet/af_packet.c
@@ -258,11 +258,15 @@ static const struct proto_ops packet_ops;
#ifdef CONFIG_SOCK_PACKET
static const struct proto_ops packet_ops_spkt;

-static int packet_rcv_spkt(struct sk_buff *skb, struct net_device *dev, struct packet_type *pt,
struct net_device *orig_dev)

```

```

+static int packet_rcv_spkt(struct sk_buff *skb, struct packet_type *pt, struct net_device *orig_dev)
{
+ struct net_device *dev = skb->dev;
  struct sock *sk;
  struct sockaddr_pkt *spkt;

+ if (!net_eq(dev->nd_net, init_net()))
+ goto out;
+
  /*
   * When we registered the protocol we saved the socket in the data
   * field for just this event.
@@ -461,8 +465,9 @@ static inline int run_filter(struct sk_buff *skb, struct sock *sk,
   we will not harm anyone.
  */

-static int packet_rcv(struct sk_buff *skb, struct net_device *dev, struct packet_type *pt, struct
net_device *orig_dev)
+static int packet_rcv(struct sk_buff *skb, struct packet_type *pt, struct net_device *orig_dev)
{
+ struct net_device *dev = skb->dev;
  struct sock *sk;
  struct sockaddr_ll *sll;
  struct packet_sock *po;
@@ -470,6 +475,9 @@ static int packet_rcv(struct sk_buff *skb, struct net_device *dev, struct
packet
  int skb_len = skb->len;
  unsigned snaplen;

+ if (!net_eq(dev->nd_net, init_net()))
+ goto drop;
+
  if (skb->pkt_type == PACKET_LOOPBACK)
    goto drop;

@@ -561,8 +569,9 @@ drop:
}

#ifdef CONFIG_PACKET_MMAP
-static int tpacket_rcv(struct sk_buff *skb, struct net_device *dev, struct packet_type *pt, struct
net_device *orig_dev)
+static int tpacket_rcv(struct sk_buff *skb, struct packet_type *pt, struct net_device *orig_dev)
{
+ struct net_device *dev = skb->dev;
  struct sock *sk;
  struct packet_sock *po;
  struct sockaddr_ll *sll;
@@ -574,6 +583,9 @@ static int tpacket_rcv(struct sk_buff *skb, struct net_device *dev, struct

```



```

packe
    unsigned short macoff, netoff;
    struct sk_buff *copy_skb = NULL;

+ if (!net_eq(dev->nd_net, init_net()))
+ goto drop;
+
    if (skb->pkt_type == PACKET_LOOPBACK)
        goto drop;

diff --git a/net/tipc/eth_media.c b/net/tipc/eth_media.c
index 682da4a..b181cf9 100644
--- a/net/tipc/eth_media.c
+++ b/net/tipc/eth_media.c
@@ -38,6 +38,7 @@
#include <net/tipc/tipc_bearer.h>
#include <net/tipc/tipc_msg.h>
#include <linux/netdevice.h>
+#include <net/net_namespace.h>

#define MAX_ETH_BEARERS 2
#define ETH_LINK_PRIORITY TIPC_DEF_LINK_PRI
@@ -91,12 +92,18 @@ static int send_msg(struct sk_buff *buf, struct tipc_bearer *tb_ptr,
    * and ensures message size matches actual length
    */

-static int recv_msg(struct sk_buff *buf, struct net_device *dev,
+static int recv_msg(struct sk_buff *buf,
    struct packet_type *pt, struct net_device *orig_dev)
{
+ struct net_device *dev = buf->dev;
    struct eth_bearer *eb_ptr = (struct eth_bearer *)pt->af_packet_priv;
    u32 size;

+ if (!net_eq(buf->dev->nd_net, init_net())) {
+ kfree_skb(buf);
+ return 0;
+ }
+
    if (likely(eb_ptr->bearer)) {
        if (likely(!dev->promiscuity) ||
            !memcmp(buf->mac.raw, dev->dev_addr, ETH_ALEN) ||
diff --git a/net/x25/x25_dev.c b/net/x25/x25_dev.c
index 47b68a3..0f63415 100644
--- a/net/x25/x25_dev.c
+++ b/net/x25/x25_dev.c
@@ -79,12 +79,16 @@ static int x25_receive_data(struct sk_buff *skb, struct x25_neigh *nb)
    return 0;

```

```
}
```

```
-int x25_lapb_receive_frame(struct sk_buff *skb, struct net_device *dev,  
+int x25_lapb_receive_frame(struct sk_buff *skb,  
    struct packet_type *ptype, struct net_device *orig_dev)  
{  
+ struct net_device *dev = skb->dev;  
    struct sk_buff *nskb;  
    struct x25_neigh *nb;  
  
+ if (!net_eq(skb->dev->nd_net, init_net()))  
+ goto drop;  
+  
    nskb = skb_copy(skb, GFP_ATOMIC);  
    if (!nskb)  
        goto drop;  
--  
1.4.4.1.g278f
```

---

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

---