Subject: [PATCH 10/16] net: Make packet reception network namespace safe
Posted by ebiederm on Sat, 08 Sep 2007 21:25:43 GMT
View Forum Message <> Reply to Message

This patch modifies every packet receive function
registered with dev_add_pack() to drop packets if they
are not from the initial network namespace.

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

Signed-off-by: Eric W. Biederman <ebiederm@xmission.com>
---
drivers/block/aoe/aoenet.c | 4 ++++
drivers/net/bonding/bond_3ad.c | 4 ++++
drivers/net/bonding/bond_alb.c | 3 +++
drivers/net/bonding/bond_main.c | 3 +++
drivers/net/hamradio/bpqether.c | 3 +++
drivers/net/pppoe.c | 6 ++++++
drivers/net/wan/hdlc.c | 7 +++++++
drivers/net/wan/lapbether.c | 3 +++
drivers/net/wan/syncppp.c | 6 ++++++
net/8021q/vlan_dev.c | 5 +++++
net/appletalk/aarp.c | 3 +++
net/appletalk/ddp.c | 6 ++++++
net/ax25/ax25_in.c | 5 +++++
net/bridge/br_stp_bpdu.c | 4 ++++
net/decnet/dn_route.c | 3 +++
net/econet/af_econet.c | 3 +++
net/ipv4/arp.c | 3 +++
net/ipv4/ip_input.c | 3 +++
net/ipv4/ipconfig.c | 6 ++++++
net/ipv6/ip6_input.c | 5 +++++
net/ipx/af_ipx.c | 3 +++
net/irda/irlap_frame.c | 3 +++
net/llc/llc_input.c | 4 ++++
net/packet/af_packet.c | 9 +++++++++
net/tipc/eth_media.c | 6 ++++++
net/x25/x25_dev.c | 3 +++
26 files changed, 113 insertions(+), 0 deletions(-)

diff --git a/drivers/block/aoe/aoenet.c b/drivers/block/aoe/aoenet.c
index f9ddfda..4dc0fb7 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 <asm/unaligned.h>
 #include "aoe.h"

@@ -114,6 +115,9 @@ aoenet_rcv(struct sk_buff *skb, struct net_device *ifp, struct packet_type
*pt,
  struct aoe_hdr *h;
  u32 n;

+ if (ifp->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 f829e4a..94bd739 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"

@@ -2448,6 +2449,9 @@ int bond_3ad_lacpdu_recv(struct sk_buff *skb, struct net_device *dev,
struct pac
  struct slave *slave = NULL;
  int ret = NET_RX_DROP;

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

diff --git a/drivers/net/bonding/bond_alb.c b/drivers/net/bonding/bond_alb.c
index 92c3b6f..419a9f8 100644
--- a/drivers/net/bonding/bond_alb.c
+++ b/drivers/net/bonding/bond_alb.c
@@ -345,6 +345,9 @@ static int rlb_arp_recv(struct sk_buff *skb, struct net_device *bond_dev,
struct
  struct arp_pkt *arp = (struct arp_pkt *)skb->data;
```

```
  int res = NET_RX_DROP;

+ if (bond_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 5de648f..e4e5fdc 100644
--- a/drivers/net/bonding/bond_main.c
+++ b/drivers/net/bonding/bond_main.c
@@ -2458,6 +2458,9 @@ static int bond_arp_rcv(struct sk_buff *skb, struct net_device *dev,
struct pack
  unsigned char *arp_ptr;
  u32 sip, tip;

+ if (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 1699d42..85fb8e7 100644
--- a/drivers/net/hamradio/bpqether.c
+++ b/drivers/net/hamradio/bpqether.c
@@ -173,6 +173,9 @@ static int bpq_rcv(struct sk_buff *skb, struct net_device *dev, struct
packet_ty
  struct ethhdr *eth;
  struct bpqdev *bpq;

+ if (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 f8bf5fc..a29ea22 100644
--- a/drivers/net/pppoe.c
+++ b/drivers/net/pppoe.c
@@ -386,6 +386,9 @@ static int pppoe_rcv(struct sk_buff *skb,
  struct pppoe_hdr *ph;
  struct pppox_sock *po;

+ if (dev->nd_net != &init_net)
+  goto drop;
```

```
+
  if (!pskb_may_pull(skb, sizeof(struct pppoe_hdr)))
   goto drop;

@@ -418,6 +421,9 @@ static int pppoe_disc_rcv(struct sk_buff *skb,
  struct pppoe_hdr *ph;
  struct pppox_sock *po;

+ if (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 65ad2e2..3b57350 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 1.21";
@@ -66,6 +67,12 @@ static int hdlc_rcv(struct sk_buff *skb, struct net_device *dev,
       struct packet_type *p, struct net_device *orig_dev)
 {
  struct hdlc_device_desc *desc = dev_to_desc(dev);
+
+ if (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..ca8b3c3 100644
--- a/drivers/net/wan/lapbether.c
+++ b/drivers/net/wan/lapbether.c
@@ -91,6 +91,9 @@ static int lapbeth_rcv(struct sk_buff *skb, struct net_device *dev, struct
packe
  int len, err;
  struct lapbethdev *lapbeth;
```

```
+ if (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 67fc67c..5c71af6 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>
@@ -1445,6 +1446,11 @@ static void sppp_print_bytes (u_char *p, u16 len)

 static int sppp_rcv(struct sk_buff *skb, struct net_device *dev, struct packet_type *p, struct
net_device *orig_dev)
 {
+ if (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);
diff --git a/net/8021q/vlan_dev.c b/net/8021q/vlan_dev.c
index 328759c..6644e8f 100644
--- a/net/8021q/vlan_dev.c
+++ b/net/8021q/vlan_dev.c
@@ -122,6 +122,11 @@ int vlan_skb_recv(struct sk_buff *skb, struct net_device *dev,
  unsigned short vlan_TCI;
  __be16 proto;

+ if (dev->nd_net != &init_net) {
+  kfree_skb(skb);
+  return -1;
+ }
+
  if ((skb = skb_share_check(skb, GFP_ATOMIC)) == NULL)
   return -1;

diff --git a/net/appletalk/aarp.c b/net/appletalk/aarp.c
```

```
index 80b5414..9267f48 100644
--- a/net/appletalk/aarp.c
+++ b/net/appletalk/aarp.c
@@ -713,6 +713,9 @@ static int aarp_rcv(struct sk_buff *skb, struct net_device *dev,
  struct atalk_addr sa, *ma, da;
  struct atalk_iface *ifa;

+ if (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 fd1d52f..c1f1367 100644
--- a/net/appletalk/ddp.c
+++ b/net/appletalk/ddp.c
@@ -1403,6 +1403,9 @@ static int atalk_rcv(struct sk_buff *skb, struct net_device *dev,
  int origlen;
  __u16 len_hops;

+ if (dev->nd_net != &init_net)
+  goto freeit;
+
  /* Don't mangle buffer if shared */
  if (!(skb = skb_share_check(skb, GFP_ATOMIC)))
   goto out;
@@ -1488,6 +1491,9 @@ freeit:
 static int ltalk_rcv(struct sk_buff *skb, struct net_device *dev,
        struct packet_type *pt, struct net_device *orig_dev)
 {
+ if (dev->nd_net != &init_net)
+  goto freeit;
+
  /* Expand any short form frames */
  if (skb_mac_header(skb)[2] == 1) {
   struct ddpehdr *ddp;
diff --git a/net/ax25/ax25_in.c b/net/ax25/ax25_in.c
index 0ddaff0..3b7d172 100644
--- a/net/ax25/ax25_in.c
+++ b/net/ax25/ax25_in.c
@@ -451,6 +451,11 @@ int ax25_kiss_rcv(struct sk_buff *skb, struct net_device *dev,
  skb->sk = NULL;  /* Initially we don't know who it's for */
  skb->destructor = NULL; /* Who initializes this, dammit?! */

+ if (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_stp_bpdu.c b/net/bridge/br_stp_bpdu.c
index 14f0c88..0edbd2a 100644
--- a/net/bridge/br_stp_bpdu.c
+++ b/net/bridge/br_stp_bpdu.c
@@ -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>
@@ -141,6 +142,9 @@ int br_stp_rcv(struct sk_buff *skb, struct net_device *dev,
  struct net_bridge *br;
  const unsigned char *buf;

+ if (dev->nd_net != &init_net)
+  goto err;
+
  if (!p)
   goto err;

diff --git a/net/decnet/dn_route.c b/net/decnet/dn_route.c
index 4cfea95..580e786 100644
--- a/net/decnet/dn_route.c
+++ b/net/decnet/dn_route.c
@@ -584,6 +584,9 @@ int dn_route_rcv(struct sk_buff *skb, struct net_device *dev, struct
packet_type
  struct dn_dev *dn = (struct dn_dev *)dev->dn_ptr;
  unsigned char padlen = 0;

+ if (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 a2429db..7de3006 100644
--- a/net/econet/af_econet.c
+++ b/net/econet/af_econet.c
@@ -1065,6 +1065,9 @@ static int econet_rcv(struct sk_buff *skb, struct net_device *dev, struct
packet
```

```
  struct sock *sk;
  struct ec_device *edev = dev->ec_ptr;

+ if (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 78dd344..bde1297 100644
--- a/net/ipv4/arp.c
+++ b/net/ipv4/arp.c
@@ -932,6 +932,9 @@ static int arp_rcv(struct sk_buff *skb, struct net_device *dev,
 {
  struct arphdr *arp;

+ if (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 9706939..41d8964 100644
--- a/net/ipv4/ip_input.c
+++ b/net/ipv4/ip_input.c
@@ -382,6 +382,9 @@ int ip_rcv(struct sk_buff *skb, struct net_device *dev, struct packet_type
*pt,
  struct iphdr *iph;
  u32 len;

+ if (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.
   */
diff --git a/net/ipv4/ipconfig.c b/net/ipv4/ipconfig.c
index 5ae4849..08ff623 100644
--- a/net/ipv4/ipconfig.c
+++ b/net/ipv4/ipconfig.c
@@ -426,6 +426,9 @@ ic_rarp_recv(struct sk_buff *skb, struct net_device *dev, struct
packet_type *pt
  unsigned char *sha, *tha;  /* s for "source", t for "target" */
  struct ic_device *d;

+ if (dev->nd_net != &init_net)
```

```
+  goto drop;
+
  if ((skb = skb_share_check(skb, GFP_ATOMIC)) == NULL)
   return NET_RX_DROP;

@@ -835,6 +838,9 @@ static int __init ic_bootp_recv(struct sk_buff *skb, struct net_device *dev,
str
  struct ic_device *d;
  int len, ext_len;

+ if (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 30a5cb1..7d18cac 100644
--- a/net/ipv6/ip6_input.c
+++ b/net/ipv6/ip6_input.c
@@ -61,6 +61,11 @@ int ipv6_rcv(struct sk_buff *skb, struct net_device *dev, struct packet_type
*pt
  u32   pkt_len;
  struct inet6_dev *idev;

+ if (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 ee28bab..f7b4d38 100644
--- a/net/ipx/af_ipx.c
+++ b/net/ipx/af_ipx.c
@@ -1647,6 +1647,9 @@ static int ipx_rcv(struct sk_buff *skb, struct net_device *dev, struct
packet_ty
  u16 ipx_pktsize;
  int rc = 0;

+ if (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 25a3444..77ac27e 100644
--- a/net/irda/irlap_frame.c
+++ b/net/irda/irlap_frame.c
@@ -1326,6 +1326,9 @@ int irlap_driver_rcv(struct sk_buff *skb, struct net_device *dev,
  int command;
  __u8 control;

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

diff --git a/net/llc/llc_input.c b/net/llc/llc_input.c
index 099ed8f..c40c9b2 100644
--- a/net/llc/llc_input.c
+++ b/net/llc/llc_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>
@@ -145,6 +146,9 @@ int llc_rcv(struct sk_buff *skb, struct net_device *dev,
  int (*rcv)(struct sk_buff *, struct net_device *,
      struct packet_type *, struct net_device *);

+ if (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.
diff --git a/net/packet/af_packet.c b/net/packet/af_packet.c
index 72ff099..51b0d5c 100644
--- a/net/packet/af_packet.c
+++ b/net/packet/af_packet.c
@@ -252,6 +252,9 @@ static int packet_rcv_spkt(struct sk_buff *skb, struct net_device *dev,
struct
  struct sock *sk;
  struct sockaddr_pkt *spkt;

+ if (dev->nd_net != &init_net)
+  goto out;
+
```

```
  /*
   * When we registered the protocol we saved the socket in the data
   * field for just this event.
@@ -452,6 +455,9 @@ static int packet_rcv(struct sk_buff *skb, struct net_device *dev, struct
packet
  int skb_len = skb->len;
  unsigned int snaplen, res;

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

@@ -568,6 +574,9 @@ static int tpacket_rcv(struct sk_buff *skb, struct net_device *dev, struct
packe
  struct sk_buff *copy_skb = NULL;
  struct timeval tv;

+ if (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 711ca4b..d2ed237 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
@@ -100,6 +101,11 @@ static int recv_msg(struct sk_buff *buf, struct net_device *dev,
  struct eth_bearer *eb_ptr = (struct eth_bearer *)pt->af_packet_priv;
  u32 size;

+ if (dev->nd_net != &init_net) {
+  kfree_skb(buf);
+  return 0;
+ }
+
  if (likely(eb_ptr->bearer)) {
   if (likely(buf->pkt_type <= PACKET_BROADCAST)) {
```

```
    size = msg_size((struct tipc_msg *)buf->data);
diff --git a/net/x25/x25_dev.c b/net/x25/x25_dev.c
index 848a6b6..f0679d2 100644
--- a/net/x25/x25_dev.c
+++ b/net/x25/x25_dev.c
@@ -95,6 +95,9 @@ int x25_lapb_receive_frame(struct sk_buff *skb, struct net_device *dev,
  struct sk_buff *nskb;
  struct x25_neigh *nb;

+ if (dev->nd_net != &init_net)
+  goto drop;
+
  nskb = skb_copy(skb, GFP_ATOMIC);
  if (!nskb)
   goto drop;
--
1.5.3.rc6.17.g1911
```

_____

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