Subject: [PATCH 0/6 net-2.6.25] Provide correct namespace on IPv4 packet input path.
Posted by den on Mon, 21 Jan 2008 14:49:50 GMT
View Forum Message <> Reply to Message

This patchset sequentially adds namespace parameter to fib_lookup and
inetdev_by_index. After that it is possible to pass network namespace
from input packet to routing engine.

Output path is much more intrusive and will be sent separately.

Signed-off-by: Denis V. Lunev <den@openvz.org>

Subject: [PATCH 1/6 net-2.6.25] [NETNS] Add netns parameter to fib_lookup.
Posted by den on Mon, 21 Jan 2008 14:50:41 GMT
View Forum Message <> Reply to Message

Signed-off-by: Denis V. Lunev <den@openvz.org>
---
 include/net/ip_fib.h     |    9 +++++----
 net/ipv4/fib_frontend.c  |    4 ++--
 net/ipv4/fib_rules.c     |    4 ++--
 net/ipv4/fib_semantics.c |    2 +-
 net/ipv4/route.c         |    6 +++---
 5 files changed, 13 insertions(+), 12 deletions(-)

```
diff --git a/include/net/ip_fib.h b/include/net/ip_fib.h
index 08ebb1e..9daa60b 100644
--- a/include/net/ip_fib.h
+++ b/include/net/ip_fib.h
@@ -178,15 +178,16 @@ static inline struct fib_table *fib_new_table(struct net *net, u32 id)
  return fib_get_table(net, id);
 }

-static inline int fib_lookup(const struct flowi *flp, struct fib_result *res)
+static inline int fib_lookup(struct net *net, const struct flowi *flp,
+        struct fib_result *res)
 {
  struct fib_table *table;

- table = fib_get_table(&init_net, RT_TABLE_LOCAL);
+ table = fib_get_table(net, RT_TABLE_LOCAL);
  if (!table->tb_lookup(table, flp, res))
   return 0;

- table = fib_get_table(&init_net, RT_TABLE_MAIN);
+ table = fib_get_table(net, RT_TABLE_MAIN);
```

```
  if (!table->tb_lookup(table, flp, res))
   return 0;
  return -ENETUNREACH;
@@ -200,7 +201,7 @@ extern void __net_exit fib4_rules_exit(struct net *net);
 extern u32 fib_rules_tclass(struct fib_result *res);
 #endif

-extern int fib_lookup(struct flowi *flp, struct fib_result *res);
+extern int fib_lookup(struct net *n, struct flowi *flp, struct fib_result *res);

 extern struct fib_table *fib_new_table(struct net *net, u32 id);
 extern struct fib_table *fib_get_table(struct net *net, u32 id);
diff --git a/net/ipv4/fib_frontend.c b/net/ipv4/fib_frontend.c
index 8c0081c..dcd3a28 100644
--- a/net/ipv4/fib_frontend.c
+++ b/net/ipv4/fib_frontend.c
@@ -256,7 +256,7 @@ int fib_validate_source(__be32 src, __be32 dst, u8 tos, int oif,
  if (in_dev == NULL)
   goto e_inval;

- if (fib_lookup(&fl, &res))
+ if (fib_lookup(&init_net, &fl, &res))
   goto last_resort;
  if (res.type != RTN_UNICAST)
   goto e_inval_res;
@@ -280,7 +280,7 @@ int fib_validate_source(__be32 src, __be32 dst, u8 tos, int oif,
  fl.oif = dev->ifindex;

  ret = 0;
- if (fib_lookup(&fl, &res) == 0) {
+ if (fib_lookup(&init_net, &fl, &res) == 0) {
   if (res.type == RTN_UNICAST) {
    *spec_dst = FIB_RES_PREFSRC(res);
    ret = FIB_RES_NH(res).nh_scope >= RT_SCOPE_HOST;
diff --git a/net/ipv4/fib_rules.c b/net/ipv4/fib_rules.c
index 2b43002..19274d0 100644
--- a/net/ipv4/fib_rules.c
+++ b/net/ipv4/fib_rules.c
@@ -54,14 +54,14 @@ u32 fib_rules_tclass(struct fib_result *res)
 }
 #endif

-int fib_lookup(struct flowi *flp, struct fib_result *res)
+int fib_lookup(struct net *net, struct flowi *flp, struct fib_result *res)
 {
  struct fib_lookup_arg arg = {
   .result = res,
  };
```

```
  int err;

- err = fib_rules_lookup(init_net.ipv4.rules_ops, flp, 0, &arg);
+ err = fib_rules_lookup(net->ipv4.rules_ops, flp, 0, &arg);
  res->r = arg.rule;

  return err;
diff --git a/net/ipv4/fib_semantics.c b/net/ipv4/fib_semantics.c
index 0e08df4..ecd91c6 100644
--- a/net/ipv4/fib_semantics.c
+++ b/net/ipv4/fib_semantics.c
@@ -559,7 +559,7 @@ static int fib_check_nh(struct fib_config *cfg, struct fib_info *fi,
    /* It is not necessary, but requires a bit of thinking */
    if (fl.fl4_scope < RT_SCOPE_LINK)
     fl.fl4_scope = RT_SCOPE_LINK;
-   if ((err = fib_lookup(&fl, &res)) != 0)
+   if ((err = fib_lookup(&init_net, &fl, &res)) != 0)
     return err;
   }
   err = -EINVAL;
diff --git a/net/ipv4/route.c b/net/ipv4/route.c
index 162e738..c107bc3 100644
--- a/net/ipv4/route.c
+++ b/net/ipv4/route.c
@@ -1559,7 +1559,7 @@ void ip_rt_get_source(u8 *addr, struct rtable *rt)

  if (rt->fl.iif == 0)
   src = rt->rt_src;
- else if (fib_lookup(&rt->fl, &res) == 0) {
+ else if (fib_lookup(&init_net, &rt->fl, &res) == 0) {
   src = FIB_RES_PREFSRC(res);
   fib_res_put(&res);
  } else
@@ -1911,7 +1911,7 @@ static int ip_route_input_slow(struct sk_buff *skb, __be32 daddr,
__be32 saddr,
  /*
   * Now we are ready to route packet.
   */
- if ((err = fib_lookup(&fl, &res)) != 0) {
+ if ((err = fib_lookup(&init_net, &fl, &res)) != 0) {
   if (!IN_DEV_FORWARD(in_dev))
    goto e_hostunreach;
   goto no_route;
@@ -2363,7 +2363,7 @@ static int ip_route_output_slow(struct rtable **rp, const struct flowi
*oldflp)
   goto make_route;
  }
```

```
- if (fib_lookup(&fl, &res)) {
+ if (fib_lookup(&init_net, &fl, &res)) {
   res.fi = NULL;
   if (oldflp->oif) {
    /* Apparently, routing tables are wrong. Assume,
--
1.5.3.rc5
```

---

Subject: [PATCH 2/6 net-2.6.25] [NETNS] Add netns parameter to inetdev_by_index.
Posted by den on Mon, 21 Jan 2008 14:50:42 GMT
View Forum Message <> Reply to Message

```
Signed-off-by: Denis V. Lunev <den@openvz.org>
---
 include/linux/inetdevice.h |    2 +-
 net/ipv4/devinet.c         |    6 ++++---
 net/ipv4/fib_semantics.c   |    2 +-
 net/ipv4/igmp.c            |    4 ++--
 net/ipv4/ip_gre.c          |    3 ++-
 5 files changed, 9 insertions(+), 8 deletions(-)

diff --git a/include/linux/inetdevice.h b/include/linux/inetdevice.h
index 45f3731..e74a2ee 100644
--- a/include/linux/inetdevice.h
+++ b/include/linux/inetdevice.h
@@ -133,7 +133,7 @@ extern struct net_device  *ip_dev_find(__be32 addr);
 extern int  inet_addr_onlink(struct in_device *in_dev, __be32 a, __be32 b);
 extern int  devinet_ioctl(unsigned int cmd, void __user *);
 extern void  devinet_init(void);
-extern struct in_device *inetdev_by_index(int);
+extern struct in_device *inetdev_by_index(struct net *, int);
 extern __be32  inet_select_addr(const struct net_device *dev, __be32 dst, int scope);
 extern __be32  inet_confirm_addr(struct in_device *in_dev, __be32 dst, __be32 local, int scope);
 extern struct in_ifaddr *inet_ifa_byprefix(struct in_device *in_dev, __be32 prefix, __be32 mask);
diff --git a/net/ipv4/devinet.c b/net/ipv4/devinet.c
index e381edb..21f71bf 100644
--- a/net/ipv4/devinet.c
+++ b/net/ipv4/devinet.c
@@ -409,12 +409,12 @@ static int inet_set_ifa(struct net_device *dev, struct in_ifaddr *ifa)
  return inet_insert_ifa(ifa);
 }

-struct in_device *inetdev_by_index(int ifindex)
+struct in_device *inetdev_by_index(struct net *net, int ifindex)
 {
  struct net_device *dev;
```

```
  struct in_device *in_dev = NULL;
  read_lock(&dev_base_lock);
- dev = __dev_get_by_index(&init_net, ifindex);
+ dev = __dev_get_by_index(net, ifindex);
  if (dev)
   in_dev = in_dev_get(dev);
  read_unlock(&dev_base_lock);
@@ -454,7 +454,7 @@ static int inet_rtm_deladdr(struct sk_buff *skb, struct nlmsghdr *nlh, void
*arg
   goto errout;

  ifm = nlmsg_data(nlh);
- in_dev = inetdev_by_index(ifm->ifa_index);
+ in_dev = inetdev_by_index(net, ifm->ifa_index);
  if (in_dev == NULL) {
   err = -ENODEV;
   goto errout;
diff --git a/net/ipv4/fib_semantics.c b/net/ipv4/fib_semantics.c
index ecd91c6..8b47e11 100644
--- a/net/ipv4/fib_semantics.c
+++ b/net/ipv4/fib_semantics.c
@@ -583,7 +583,7 @@ out:
   if (nh->nh_flags&(RTNH_F_PERVASIVE|RTNH_F_ONLINK))
    return -EINVAL;

-  in_dev = inetdev_by_index(nh->nh_oif);
+  in_dev = inetdev_by_index(&init_net, nh->nh_oif);
   if (in_dev == NULL)
    return -ENODEV;
   if (!(in_dev->dev->flags&IFF_UP)) {
diff --git a/net/ipv4/igmp.c b/net/ipv4/igmp.c
index 285d262..b4df39a 100644
--- a/net/ipv4/igmp.c
+++ b/net/ipv4/igmp.c
@@ -1389,7 +1389,7 @@ static struct in_device * ip_mc_find_dev(struct ip_mreqn *imr)
  struct in_device *idev = NULL;

  if (imr->imr_ifindex) {
-  idev = inetdev_by_index(imr->imr_ifindex);
+  idev = inetdev_by_index(&init_net, imr->imr_ifindex);
   if (idev)
    __in_dev_put(idev);
   return idev;
@@ -2222,7 +2222,7 @@ void ip_mc_drop_socket(struct sock *sk)
   struct in_device *in_dev;
   inet->mc_list = iml->next;

-  in_dev = inetdev_by_index(iml->multi.imr_ifindex);
```

```
+  in_dev = inetdev_by_index(&init_net, iml->multi.imr_ifindex);
   (void) ip_mc_leave_src(sk, iml, in_dev);
   if (in_dev != NULL) {
    ip_mc_dec_group(in_dev, iml->multi.imr_multiaddr.s_addr);
diff --git a/net/ipv4/ip_gre.c b/net/ipv4/ip_gre.c
index 8b81deb..a74983d 100644
--- a/net/ipv4/ip_gre.c
+++ b/net/ipv4/ip_gre.c
@@ -1193,7 +1193,8 @@ static int ipgre_close(struct net_device *dev)
 {
  struct ip_tunnel *t = netdev_priv(dev);
  if (ipv4_is_multicast(t->parms.iph.daddr) && t->mlink) {
-  struct in_device *in_dev = inetdev_by_index(t->mlink);
+  struct in_device *in_dev;
+  in_dev = inetdev_by_index(dev->nd_net, t->mlink);
   if (in_dev) {
    ip_mc_dec_group(in_dev, t->parms.iph.daddr);
    in_dev_put(in_dev);
--
1.5.3.rc5
```

---

## Subject: [PATCH 3/6 net-2.6.25] [NETNS] Pass correct namespace in fib_validate_source.

Posted by den on Mon, 21 Jan 2008 14:50:43 GMT

View Forum Message <> Reply to Message

Correct network namespace is available inside fib_validate_source. It can be obtained from the device passed in. The device is not NULL as in_device is obtained from it just above.

Signed-off-by: Denis V. Lunev <den@openvz.org>

```
---
 net/ipv4/fib_frontend.c |    6 ++++--
 1 files changed, 4 insertions(+), 2 deletions(-)

diff --git a/net/ipv4/fib_frontend.c b/net/ipv4/fib_frontend.c
index dcd3a28..39b8b35 100644
--- a/net/ipv4/fib_frontend.c
+++ b/net/ipv4/fib_frontend.c
@@ -243,6 +243,7 @@ int fib_validate_source(__be32 src, __be32 dst, u8 tos, int oif,
  struct fib_result res;
  int no_addr, rpf;
  int ret;
+ struct net *net;

  no_addr = rpf = 0;
  rcu_read_lock();
```

```
@@ -256,7 +257,8 @@ int fib_validate_source(__be32 src, __be32 dst, u8 tos, int oif,
  if (in_dev == NULL)
   goto e_inval;

- if (fib_lookup(&init_net, &fl, &res))
+ net = dev->nd_net;
+ if (fib_lookup(net, &fl, &res))
   goto last_resort;
  if (res.type != RTN_UNICAST)
   goto e_inval_res;
@@ -280,7 +282,7 @@ int fib_validate_source(__be32 src, __be32 dst, u8 tos, int oif,
  fl.oif = dev->ifindex;

  ret = 0;
- if (fib_lookup(&init_net, &fl, &res) == 0) {
+ if (fib_lookup(net, &fl, &res) == 0) {
   if (res.type == RTN_UNICAST) {
    *spec_dst = FIB_RES_PREFSRC(res);
    ret = FIB_RES_NH(res).nh_scope >= RT_SCOPE_HOST;
--
1.5.3.rc5
```

---

## Subject: [PATCH 4/6 net-2.6.25] [NETNS] Pass correct namespace in context fib_check_nh.

Posted by den on Mon, 21 Jan 2008 14:50:44 GMT

View Forum Message <> Reply to Message

Correct network namespace is already used in fib_check_nh. Re-work its usage
for better readability and pass into fib_lookup & inetdev_by_index.

Signed-off-by: Denis V. Lunev <den@openvz.org>

```
---
 net/ipv4/fib_semantics.c |   12 ++++++------
 1 files changed, 6 insertions(+), 6 deletions(-)

diff --git a/net/ipv4/fib_semantics.c b/net/ipv4/fib_semantics.c
index 8b47e11..c791286 100644
--- a/net/ipv4/fib_semantics.c
+++ b/net/ipv4/fib_semantics.c
@@ -519,7 +519,9 @@ static int fib_check_nh(struct fib_config *cfg, struct fib_info *fi,
    struct fib_nh *nh)
 {
  int err;
+ struct net *net;

+ net = cfg->fc_nlinfo.nl_net;
  if (nh->nh_gw) {
```

```
   struct fib_result res;

@@ -532,11 +534,9 @@ static int fib_check_nh(struct fib_config *cfg, struct fib_info *fi,

    if (cfg->fc_scope >= RT_SCOPE_LINK)
     return -EINVAL;
-   if (inet_addr_type(cfg->fc_nlinfo.nl_net,
-       nh->nh_gw) != RTN_UNICAST)
+   if (inet_addr_type(net, nh->nh_gw) != RTN_UNICAST)
     return -EINVAL;
-   if ((dev = __dev_get_by_index(cfg->fc_nlinfo.nl_net,
-           nh->nh_oif)) == NULL)
+   if ((dev = __dev_get_by_index(net, nh->nh_oif)) == NULL)
     return -ENODEV;
    if (!(dev->flags&IFF_UP))
     return -ENETDOWN;
@@ -559,7 +559,7 @@ static int fib_check_nh(struct fib_config *cfg, struct fib_info *fi,
    /* It is not necessary, but requires a bit of thinking */
    if (fl.fl4_scope < RT_SCOPE_LINK)
     fl.fl4_scope = RT_SCOPE_LINK;
-   if ((err = fib_lookup(&init_net, &fl, &res)) != 0)
+   if ((err = fib_lookup(net, &fl, &res)) != 0)
     return err;
   }
   err = -EINVAL;
@@ -583,7 +583,7 @@ out:
   if (nh->nh_flags&(RTNH_F_PERVASIVE|RTNH_F_ONLINK))
    return -EINVAL;

-  in_dev = inetdev_by_index(&init_net, nh->nh_oif);
+  in_dev = inetdev_by_index(net, nh->nh_oif);
   if (in_dev == NULL)
    return -ENODEV;
   if (!(in_dev->dev->flags&IFF_UP)) {
--
1.5.3.rc5
```

---

## Subject: [PATCH 5/6 net-2.6.25] [NETNS] Pass correct namespace in ip_route_input_slow.

Posted by den on Mon, 21 Jan 2008 14:50:45 GMT

View Forum Message <> Reply to Message

The packet on the input path always has a referrence to an input network device it is passed from. Extract network namespace from it.

Signed-off-by: Denis V. Lunev <den@openvz.org>

---

```
 net/ipv4/route.c |    7 ++++---
 1 files changed, 4 insertions(+), 3 deletions(-)

diff --git a/net/ipv4/route.c b/net/ipv4/route.c
index c107bc3..b3c6122 100644
--- a/net/ipv4/route.c
+++ b/net/ipv4/route.c
@@ -1881,6 +1881,7 @@ static int ip_route_input_slow(struct sk_buff *skb, __be32 daddr,
__be32 saddr,
  __be32  spec_dst;
  int  err = -EINVAL;
  int  free_res = 0;
+ struct net    * net = dev->nd_net;

  /* IP on this device is disabled. */

@@ -1911,7 +1912,7 @@ static int ip_route_input_slow(struct sk_buff *skb, __be32 daddr,
__be32 saddr,
  /*
   * Now we are ready to route packet.
   */
- if ((err = fib_lookup(&init_net, &fl, &res)) != 0) {
+ if ((err = fib_lookup(net, &fl, &res)) != 0) {
   if (!IN_DEV_FORWARD(in_dev))
    goto e_hostunreach;
   goto no_route;
@@ -1926,7 +1927,7 @@ static int ip_route_input_slow(struct sk_buff *skb, __be32 daddr,
__be32 saddr,
  if (res.type == RTN_LOCAL) {
   int result;
   result = fib_validate_source(saddr, daddr, tos,
-          init_net.loopback_dev->ifindex,
+          net->loopback_dev->ifindex,
           dev, &spec_dst, &itag);
   if (result < 0)
    goto martian_source;
@@ -1988,7 +1989,7 @@ local_input:
 #endif
  rth->rt_iif =
  rth->fl.iif = dev->ifindex;
- rth->u.dst.dev = init_net.loopback_dev;
+ rth->u.dst.dev = net->loopback_dev;
  dev_hold(rth->u.dst.dev);
  rth->idev = in_dev_get(rth->u.dst.dev);
  rth->rt_gateway = daddr;
--
1.5.3.rc5
```

Subject: [PATCH 6/6 net-2.6.25] [NETNS] Pass correct namespace in ip_rt_get_source.
Posted by den on Mon, 21 Jan 2008 14:50:46 GMT
View Forum Message <> Reply to Message

ip_rt_get_source is the infamous place for which dst_ifdown kludges have
been implemented. This means that rt->u.dst.dev can be safely dereferrenced
obtain nd_net.

Signed-off-by: Denis V. Lunev <den@openvz.org>
---
 net/ipv4/route.c |    2 +-
 1 files changed, 1 insertions(+), 1 deletions(-)

diff --git a/net/ipv4/route.c b/net/ipv4/route.c
index b3c6122..ede0571 100644
--- a/net/ipv4/route.c
+++ b/net/ipv4/route.c
@@ -1559,7 +1559,7 @@ void ip_rt_get_source(u8 *addr, struct rtable *rt)

  if (rt->fl.iif == 0)
   src = rt->rt_src;
- else if (fib_lookup(&init_net, &rt->fl, &res) == 0) {
+ else if (fib_lookup(rt->u.dst.dev->nd_net, &rt->fl, &res) == 0) {
   src = FIB_RES_PREFSRC(res);
   fib_res_put(&res);
  } else
--
1.5.3.rc5

---

Subject: Re: [PATCH 0/6 net-2.6.25] Provide correct namespace on IPv4 packet input path.
Posted by davem on Tue, 22 Jan 2008 01:35:19 GMT
View Forum Message <> Reply to Message

From: "Denis V. Lunev" <den@sw.ru>
Date: Mon, 21 Jan 2008 17:49:50 +0300

> This patchset sequentially adds namespace parameter to fib_lookup and
> inetdev_by_index. After that it is possible to pass network namespace
> from input packet to routing engine.
>
> Output path is much more intrusive and will be sent separately.
>
> Signed-off-by: Denis V. Lunev <den@openvz.org>

All 6 patches applied, thanks.

---