
Subject: [PATCH 7/12 net-2.6.25] [NETNS]: Add namespace parameter to
__ip_route_output_key.

Posted by [den](#) on Tue, 22 Jan 2008 15:59:24 GMT

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

This is only required to propagate it down to the ip_route_output_slow.

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

```
include/net/route.h    | 4 ++--
net/ipv4/icmp.c        | 4 ++--
net/ipv4/route.c       | 7 ++++---
net/ipv4/xfrm4_policy.c | 2 +-
4 files changed, 9 insertions(+), 8 deletions(-)
```

diff --git a/include/net/route.h b/include/net/route.h

index 5847e6f..3e3b14e 100644

--- a/include/net/route.h

+++ b/include/net/route.h

```
@@ -110,7 +110,7 @@ extern int ip_rt_init(void);
extern void ip_rt_redirect(__be32 old_gw, __be32 dst, __be32 new_gw,
                          __be32 src, struct net_device *dev);
extern void rt_cache_flush(int how);
-extern int __ip_route_output_key(struct rtable **, const struct flowi *flp);
+extern int __ip_route_output_key(struct net *, struct rtable **, const struct flowi *flp);
extern int ip_route_output_key(struct rtable **, struct flowi *flp);
extern int ip_route_output_flow(struct rtable **rp, struct flowi *flp, struct sock *sk, int flags);
extern int ip_route_input(struct sk_buff*, __be32 dst, __be32 src, u8 tos, struct net_device
*devin);
@@ -158,7 +158,7 @@ static inline int ip_route_connect(struct rtable **rp, __be32 dst,
```

```
int err;
if (!dst || !src) {
- err = __ip_route_output_key(rp, &fl);
+ err = __ip_route_output_key(&init_net, rp, &fl);
if (err)
return err;
fl.fl4_dst = (*rp)->rt_dst;
```

diff --git a/net/ipv4/icmp.c b/net/ipv4/icmp.c

index 7ed8c50..21422bf 100644

--- a/net/ipv4/icmp.c

+++ b/net/ipv4/icmp.c

```
@@ -569,7 +569,7 @@ void icmp_send(struct sk_buff *skb_in, int type, int code, __be32 info)
struct rtable *rt2;
```

```
security_skb_classify_flow(skb_in, &fl);
- if (__ip_route_output_key(&rt, &fl))
+ if (__ip_route_output_key(&init_net, &rt, &fl))
```

```

goto out_unlock;

/* No need to clone since we're just using its address. */
@@ -592,7 +592,7 @@ void icmp_send(struct sk_buff *skb_in, int type, int code, __be32 info)
    goto out_unlock;

    if (inet_addr_type(&init_net, fl.fl4_src) == RTN_LOCAL)
-   err = __ip_route_output_key(&rt2, &fl);
+   err = __ip_route_output_key(&init_net, &rt2, &fl);
    else {
        struct flowi fl2 = {};
        struct dst_entry *odst;
diff --git a/net/ipv4/route.c b/net/ipv4/route.c
index c1f9950..cb035cc 100644
--- a/net/ipv4/route.c
+++ b/net/ipv4/route.c
@@ -2442,7 +2442,8 @@ make_route:
out: return err;
}

-int __ip_route_output_key(struct rtable **rp, const struct flowi *flp)
+int __ip_route_output_key(struct net *net, struct rtable **rp,
+    const struct flowi *flp)
{
    unsigned hash;
    struct rtable *rth;
@@ -2469,7 +2470,7 @@ int __ip_route_output_key(struct rtable **rp, const struct flowi *flp)
}
rcu_read_unlock_bh();

- return ip_route_output_slow(&init_net, rp, flp);
+ return ip_route_output_slow(net, rp, flp);
}

EXPORT_SYMBOL_GPL(__ip_route_output_key);
@@ -2535,7 +2536,7 @@ int ip_route_output_flow(struct rtable **rp, struct flowi *flp, struct sock
*sk,
{
    int err;

- if ((err = __ip_route_output_key(rp, flp)) != 0)
+ if ((err = __ip_route_output_key(&init_net, rp, flp)) != 0)
    return err;

    if (flp->proto) {
diff --git a/net/ipv4/xfrm4_policy.c b/net/ipv4/xfrm4_policy.c
index f04516c..3783e3e 100644
--- a/net/ipv4/xfrm4_policy.c

```

+++ b/net/ipv4/xfrm4_policy.c

```
@@ -36,7 +36,7 @@ static struct dst_entry *xfrm4_dst_lookup(int tos, xfrm_address_t *saddr,  
    if (saddr)  
        fl.fl4_src = saddr->a4;
```

```
- err = __ip_route_output_key(&rt, &fl);
```

```
+ err = __ip_route_output_key(&init_net, &rt, &fl);
```

```
    dst = &rt->u.dst;
```

```
    if (err)
```

```
        dst = ERR_PTR(err);
```

```
--
```

1.5.3.rc5
