
Subject: [PATCH 8/12 net-2.6.25] [NETNS]: Add namespace parameter to ip_route_output_flow.

Posted by [den](#) on Tue, 22 Jan 2008 15:59:25 GMT

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

Needed to propagate it down to the __ip_route_output_key.

Signed_off_by: Denis V. Lunev <den@openvz.org>

```
drivers/infiniband/hw/cxgb3/iwch_cm.c |  2 ++
include/net/route.h                  |  6 +-----
net/dccp/ipv4.c                     |  2 +-+
net/ipv4/af_inet.c                  |  2 +-+
net/ipv4/inet_connection_sock.c     |  2 +-+
net/ipv4/ip_output.c                |  2 +-+
net/ipv4/raw.c                      |  2 +-+
net/ipv4/route.c                   |  7 +++++-
net/ipv4/udp.c                      |  2 +-+
9 files changed, 14 insertions(+), 13 deletions(-)

diff --git a/drivers/infiniband/hw/cxgb3/iwch_cm.c b/drivers/infiniband/hw/cxgb3/iwch_cm.c
index 20ba372..ff3dee4 100644
--- a/drivers/infiniband/hw/cxgb3/iwch_cm.c
+++ b/drivers/infiniband/hw/cxgb3/iwch_cm.c
@@ -332,7 +332,7 @@ static struct rtable *find_route(struct t3cdev *dev, __be32 local_ip,
    }
};

- if (ip_route_output_flow(&rt, &fl, NULL, 0))
+ if (ip_route_output_flow(&init_net, &rt, &fl, NULL, 0))
    return NULL;
    return rt;
}

diff --git a/include/net/route.h b/include/net/route.h
index 3e3b14e..6b970d7 100644
--- a/include/net/route.h
+++ b/include/net/route.h
@@ -112,7 +112,7 @@ extern void ip_rt_redirect(__be32 old_gw, __be32 dst, __be32 new_gw,
extern void rt_cache_flush(int how);
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_output_flow(struct net *, 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);
extern unsigned short ip_rt_frag_needed(struct iphdr *iph, unsigned short new_mtu);
extern void ip_rt_send_redirect(struct sk_buff *skb);
```

```

@@ -167,7 +167,7 @@ static inline int ip_route_connect(struct rtable **rp, __be32 dst,
    *rp = NULL;
}
security_sk_classify_flow(sk, &fl);
- return ip_route_output_flow(rp, &fl, sk, flags);
+ return ip_route_output_flow(&init_net, rp, &fl, sk, flags);
}

static inline int ip_route_newports(struct rtable **rp, u8 protocol,
@@ -184,7 +184,7 @@ static inline int ip_route_newports(struct rtable **rp, u8 protocol,
    ip_rt_put(*rp);
    *rp = NULL;
    security_sk_classify_flow(sk, &fl);
- return ip_route_output_flow(rp, &fl, sk, 0);
+ return ip_route_output_flow(&init_net, rp, &fl, sk, 0);
}
return 0;
}
diff --git a/net/dccp/ipv4.c b/net/dccp/ipv4.c
index f450df2..9e38b0d 100644
--- a/net/dccp/ipv4.c
+++ b/net/dccp/ipv4.c
@@ -469,7 +469,7 @@ static struct dst_entry* dccp_v4_route_skb(struct sock *sk,
};

security_skb_classify_flow(skb, &fl);
- if (ip_route_output_flow(&rt, &fl, sk, 0)) {
+ if (ip_route_output_flow(&init_net, &rt, &fl, sk, 0)) {
    IP_INC_STATS_BH(IPSTATS_MIB_OUTNOROUTES);
    return NULL;
}
diff --git a/net/ipv4/af_inet.c b/net/ipv4/af_inet.c
index bcf8c8a..09ca529 100644
--- a/net/ipv4/af_inet.c
+++ b/net/ipv4/af_inet.c
@@ -1113,7 +1113,7 @@ int inet_sk_rebuild_header(struct sock *sk)
};

security_sk_classify_flow(sk, &fl);
- err = ip_route_output_flow(&rt, &fl, sk, 0);
+ err = ip_route_output_flow(&init_net, &rt, &fl, sk, 0);
}
if (!err)
    sk_setup_caps(sk, &rt->u.dst);
diff --git a/net/ipv4/inet_connection_sock.c b/net/ipv4/inet_connection_sock.c
index 1c2a32f..7801cce 100644
--- a/net/ipv4/inet_connection_sock.c
+++ b/net/ipv4/inet_connection_sock.c

```

```

@@ -333,7 +333,7 @@ struct dst_entry* inet_csk_route_req(struct sock *sk,
    .dport = ireq->rmt_port } } };

security_req_classify_flow(req, &fl);
- if (ip_route_output_flow(&rt, &fl, sk, 0)) {
+ if (ip_route_output_flow(&init_net, &rt, &fl, sk, 0)) {
    IP_INC_STATS_BH(IPSTATS_MIB_OUTNOROUTES);
    return NULL;
}
diff --git a/net/ipv4/ip_output.c b/net/ipv4/ip_output.c
index e57de0f..dc56e40 100644
--- a/net/ipv4/ip_output.c
+++ b/net/ipv4/ip_output.c
@@ -350,7 +350,7 @@ int ip_queue_xmit(struct sk_buff *skb, int ipfragok)
    * itself out.
 */
security_sk_classify_flow(sk, &fl);
- if (ip_route_output_flow(&rt, &fl, sk, 0))
+ if (ip_route_output_flow(&init_net, &rt, &fl, sk, 0))
    goto no_route;
}
sk_setup_caps(sk, &rt->u.dst);
diff --git a/net/ipv4/raw.c b/net/ipv4/raw.c
index 91a5218..85c0869 100644
--- a/net/ipv4/raw.c
+++ b/net/ipv4/raw.c
@@ -558,7 +558,7 @@ static int raw_sendmsg(struct kiocb *iocb, struct sock *sk, struct msghdr
*msg,
}

security_sk_classify_flow(sk, &fl);
- err = ip_route_output_flow(&rt, &fl, sk, 1);
+ err = ip_route_output_flow(&init_net, &rt, &fl, sk, 1);
}
if (err)
    goto done;
diff --git a/net/ipv4/route.c b/net/ipv4/route.c
index cb035cc..58ad12a 100644
--- a/net/ipv4/route.c
+++ b/net/ipv4/route.c
@@ -2532,11 +2532,12 @@ static int ipv4_dst_blackhole(struct rtable **rp, struct flowi *flp, struct
sock
    return (rt ? 0 : -ENOMEM);
}

-int ip_route_output_flow(struct rtable **rp, struct flowi *flp, struct sock *sk, int flags)
+int ip_route_output_flow(struct net *net, struct rtable **rp, struct flowi *flp,
+    struct sock *sk, int flags)

```

```

{
int err;

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

    if (flp->proto) {
@@ -2559,7 +2560,7 @@ EXPORT_SYMBOL_GPL(ip_route_output_flow);

int ip_route_output_key(struct rtable **rp, struct flowi *flp)
{
- return ip_route_output_flow(rp, flp, NULL, 0);
+ return ip_route_output_flow(&init_net, rp, flp, NULL, 0);
}

static int rt_fill_info(struct sk_buff *skb, u32 pid, u32 seq, int event,
diff --git a/net/ipv4/udp.c b/net/ipv4/udp.c
index cb2411c..9092d80 100644
--- a/net/ipv4/udp.c
+++ b/net/ipv4/udp.c
@@ -660,7 +660,7 @@ int udp_sendmsg(struct kiocb *iocb, struct sock *sk, struct msghdr *msg,
    { .sport = inet->sport,
      .dport = dport } } );
    security_sk_classify_flow(sk, &fl);
- err = ip_route_output_flow(&rt, &fl, sk, 1);
+ err = ip_route_output_flow(&init_net, &rt, &fl, sk, 1);
    if (err) {
        if (err == -ENETUNREACH)
            IP_INC_STATS_BH(IPSTATS_MIB_OUTNOROUTES);
    }
--
```

1.5.3.rc5
