Subject: [PATCH 7/11 net-2.6.26] [NETNS]: Process INET socket layer in the correct namespace.
Posted by den on Mon, 24 Mar 2008 14:36:08 GMT
View Forum Message <> Reply to Message

Replace all the reast of the init_net with a proper net on the socket layer.

Signed-off-by: Denis V. Lunev <den@openvz.org>
---
 net/ipv4/af_inet.c | 4 ++--
 net/ipv4/inet_connection_sock.c | 2 +-
 net/ipv4/raw.c | 4 ++--
 net/ipv4/udp.c | 4 ++--
 4 files changed, 7 insertions(+), 7 deletions(-)

```
diff --git a/net/ipv4/af_inet.c b/net/ipv4/af_inet.c
index 44f5ce1..a9e241c 100644
--- a/net/ipv4/af_inet.c
+++ b/net/ipv4/af_inet.c
@@ -446,7 +446,7 @@ int inet_bind(struct socket *sock, struct sockaddr *uaddr, int addr_len)
  if (addr_len < sizeof(struct sockaddr_in))
   goto out;

- chk_addr_ret = inet_addr_type(&init_net, addr->sin_addr.s_addr);
+ chk_addr_ret = inet_addr_type(sk->sk_net, addr->sin_addr.s_addr);

  /* Not specified by any standard per-se, however it breaks too
   * many applications when removed.  It is unfortunate since
@@ -1114,7 +1114,7 @@ int inet_sk_rebuild_header(struct sock *sk)
  };

  security_sk_classify_flow(sk, &fl);
- err = ip_route_output_flow(&init_net, &rt, &fl, sk, 0);
+ err = ip_route_output_flow(sk->sk_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 f9c5c4d..d13c5f1 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(&init_net, &rt, &fl, sk, 0)) {
+ if (ip_route_output_flow(sk->sk_net, &rt, &fl, sk, 0)) {
   IP_INC_STATS_BH(IPSTATS_MIB_OUTNOROUTES);
```

```
   return NULL;
  }
diff --git a/net/ipv4/raw.c b/net/ipv4/raw.c
index 7d29a05..3f68a93 100644
--- a/net/ipv4/raw.c
+++ b/net/ipv4/raw.c
@@ -499,7 +499,7 @@ static int raw_sendmsg(struct kiocb *iocb, struct sock *sk, struct msghdr
*msg,
   ipc.oif = sk->sk_bound_dev_if;

   if (msg->msg_controllen) {
-   err = ip_cmsg_send(&init_net, msg, &ipc);
+   err = ip_cmsg_send(sk->sk_net, msg, &ipc);
    if (err)
     goto out;
    if (ipc.opt)
@@ -553,7 +553,7 @@ static int raw_sendmsg(struct kiocb *iocb, struct sock *sk, struct msghdr
*msg,
    }

    security_sk_classify_flow(sk, &fl);
-   err = ip_route_output_flow(&init_net, &rt, &fl, sk, 1);
+   err = ip_route_output_flow(sk->sk_net, &rt, &fl, sk, 1);
   }
   if (err)
    goto done;
diff --git a/net/ipv4/udp.c b/net/ipv4/udp.c
index 9d69e9d..1f48687 100644
--- a/net/ipv4/udp.c
+++ b/net/ipv4/udp.c
@@ -607,7 +607,7 @@ int udp_sendmsg(struct kiocb *iocb, struct sock *sk, struct msghdr *msg,

   ipc.oif = sk->sk_bound_dev_if;
   if (msg->msg_controllen) {
-   err = ip_cmsg_send(&init_net, msg, &ipc);
+   err = ip_cmsg_send(sk->sk_net, msg, &ipc);
    if (err)
     return err;
    if (ipc.opt)
@@ -656,7 +656,7 @@ 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(&init_net, &rt, &fl, sk, 1);
+   err = ip_route_output_flow(sk->sk_net, &rt, &fl, sk, 1);
    if (err) {
     if (err == -ENETUNREACH)
      IP_INC_STATS_BH(IPSTATS_MIB_OUTNOROUTES);
```

--
1.5.3.rc5