
Subject: [PATCH 3/11 net-2.6.26] [NETNS]: Add namespace parameter to ip_options_compile.

Posted by [den](#) on Mon, 24 Mar 2008 14:36:04 GMT

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

ip_options_compile uses inet_addr_type which requires a namespace. The packet argument is optional, so parameter is the only way to obtain it. Pass the init_net there for now.

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

```
include/net/ip.h      | 3 +-
net/ipv4/ip_input.c   | 2 +-
net/ipv4/ip_options.c | 7 +++++-
3 files changed, 7 insertions(+), 5 deletions(-)
```

```
diff --git a/include/net/ip.h b/include/net/ip.h
```

```
index 9f50d4f..bcc3afa 100644
```

```
--- a/include/net/ip.h
```

```
+++ b/include/net/ip.h
```

```
@@ -347,7 +347,8 @@ extern int ip_forward(struct sk_buff *skb);
extern void ip_options_build(struct sk_buff *skb, struct ip_options *opt, __be32 daddr, struct
rtable *rt, int is_frag);
extern int ip_options_echo(struct ip_options *dopt, struct sk_buff *skb);
extern void ip_options_fragment(struct sk_buff *skb);
-extern int ip_options_compile(struct ip_options *opt, struct sk_buff *skb);
+extern int ip_options_compile(struct net *net,
+ struct ip_options *opt, struct sk_buff *skb);
extern int ip_options_get(struct ip_options **optp,
unsigned char *data, int optlen);
extern int ip_options_get_from_user(struct ip_options **optp,
```

```
diff --git a/net/ipv4/ip_input.c b/net/ipv4/ip_input.c
```

```
index e3a0c78..f3a7a08 100644
```

```
--- a/net/ipv4/ip_input.c
```

```
+++ b/net/ipv4/ip_input.c
```

```
@@ -286,7 +286,7 @@ static inline int ip_rcv_options(struct sk_buff *skb)
opt = &(IPCB(skb)->opt);
opt->optlen = iph->ihl*4 - sizeof(struct iphdr);
```

```
- if (ip_options_compile(opt, skb)) {
+ if (ip_options_compile(&init_net, opt, skb)) {
IP_INC_STATS_BH(IPSTATS_MIB_INHDRERRORS);
goto drop;
}
```

```
diff --git a/net/ipv4/ip_options.c b/net/ipv4/ip_options.c
```

```
index aeed4e5..f0949b4 100644
```

```
--- a/net/ipv4/ip_options.c
```

```
+++ b/net/ipv4/ip_options.c
```

```

@@ -248,7 +248,8 @@ void ip_options_fragment(struct sk_buff * skb)
 * If opt == NULL, then skb->data should point to IP header.
 */

-int ip_options_compile(struct ip_options * opt, struct sk_buff * skb)
+int ip_options_compile(struct net *net,
+ struct ip_options * opt, struct sk_buff * skb)
{
    int l;
    unsigned char * iph;
@@ -389,7 +390,7 @@ int ip_options_compile(struct ip_options * opt, struct sk_buff * skb)
{
    __be32 addr;
    memcpy(&addr, &optptr[optptr[2]-1], 4);
- if (inet_addr_type(&init_net, addr) == RTN_UNICAST)
+ if (inet_addr_type(net, addr) == RTN_UNICAST)
    break;
    if (skb)
        timeptr = (__be32*)&optptr[optptr[2]+3];
@@ -512,7 +513,7 @@ static int ip_options_get_finish(struct ip_options **optp,
while (optlen & 3)
    opt->__data[optlen++] = IPOPT_END;
    opt->optlen = optlen;
- if (optlen && ip_options_compile(opt, NULL)) {
+ if (optlen && ip_options_compile(&init_net, opt, NULL)) {
    kfree(opt);
    return -EINVAL;
}
--
1.5.3.rc5

```
