Subject: Re: [RFC] L3 network isolation : broadcast
Posted by Daniel Lezcano on Fri, 15 Dec 2006 10:04:14 GMT
View Forum Message <> Reply to Message

Daniel Lezcano wrote:
> Hi all,
>
> I am trying to find a solution to handle the broadcast traffic on the l3
> namespace.
>
> The broadcast issue comes from the l2 isolation:
>
> in udp.c
>
> static inline struct sock *udp_v4_mcast_next(struct sock *sk,
> __be16 loc_port,
> __be32 loc_addr,
> __be16 rmt_port,
> __be32 rmt_addr,
> int dif)
> {
> struct hlist_node *node;
> struct sock *s = sk;
> struct net_namespace *ns = current_net_ns;
> unsigned short hnum = ntohs(loc_port);
>
> sk_for_each_from(s, node) {
> struct inet_sock *inet = inet_sk(s);
>
> if (inet->num != hnum ||
> (inet->daddr && inet->daddr != rmt_addr) ||
> (inet->dport != rmt_port && inet->dport) ||
> (inet->rcv_saddr && inet->rcv_saddr != loc_addr) ||
> ipv6_only_sock(s) ||
> !net_ns_match(sk->sk_net_ns, ns) ||
> (s->sk_bound_dev_if && s->sk_bound_dev_if != dif))
> continue;
> if (!ip_mc_sf_allow(s, loc_addr, rmt_addr, dif))
> continue;
> goto found;
> }
> s = NULL;
> found:
> return s;
> }
>
> This is absolutely correct for l2 namespaces because they share the
> socket hash table. But that is not correct for l3 namespaces because we

> want to deliver the packet to each l3 namespaces which have binded to
> the broadcast address, so we should avoid checking net_ns_match if we
> are in a layer 3 namespace. Doing that we will break the l2 isolation
> because an another l2 namespace could have binded to the same broadcast
> address.
>
> The solution I see here is:
>
> if namespace is l3 then;
> net_ns match any net_ns registered as listening on this address
> else
> net_ns_match
> fi

Finally, I found a more simple solution which does not need extra
registering address and is more efficient.
The packet is to be delivered to the l2 namespace and all l3 childs.
So, instead of doing net_ns_match in the udp_v4_mcast_next, we do
net_ns_sock_is_visible and this function does:

```
  if current_namespace is l3 then;
namespace = current_namespace->parent
  fi

  if socket->namespace is l3 then
return socket->namespace->parent == namespace
  else
return socket->namespace == namespace
  fi
```