Subject: Re: [RFC] L3 network isolation : broadcast
Posted by Vlad Yasevich on Wed, 13 Dec 2006 21:41:46 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.

A question, if you will... I am still digesting the l2 changes, and I can't
remember/find if the broadcasts will be replicated across multiple l2 or not.

Example:
A system has 2 interfaces eth0 and eth1 connected to the same lan/link.
Each NIC was isolated to it's own L2 space. Each L2 space configures
the its nic with unique IP but in the same subnet. Will both L2s receive
a subnet broadcast packet?

If yes, then below approach will work. If no, then we'll need something else
since both L2s should get the packet in their own right.

>
> 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
>
> The registered network namespace is a list shared between brothers l3
> namespaces. This will add more overhead for sure. Does anyone have
> comments on that or perhaps a better solution ?

-vlad

Subject: Re: [RFC] L3 network isolation : broadcast
Posted by Daniel Lezcano on Wed, 13 Dec 2006 23:08:11 GMT
View Forum Message <> Reply to Message

Vlad Yasevich wrote:
> 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.
>
> A question, if you will... I am still digesting the l2 changes, and I can't
> remember/find if the broadcasts will be replicated across multiple l2 or not.

Well ... I am not sure (never tested it) but as far as I remember, it is
the bridge which should duplicate the packets because it acts as a "hub".

```
eth0 --- br0 ---- veth0--|ns l2]--eth0
            |
             -- veth1--|ns l2]--eth0
            |
             -- veth2--[ns l2]--eth0
```

When a packet is received on eth0, it is forwarded to br0 (the bridge)
and this one will send the packet to veth0, veth1 and veth2. The packets
will follow the normal incoming path for each namespace. So I think the
answer is yes, the broadcast is replicated to each l2 namespace.

Dmitry can give more information on that I think.

>
> Example:
> A system has 2 interfaces eth0 and eth1 connected to the same lan/link.
> Each NIC was isolated to it's own L2 space. Each L2 space configures
> the its nic with unique IP but in the same subnet.   Will both L2s receive
> a subnet broadcast packet?

Depending on the bridge configuration, I am inclined to say yes if eth0
and eth1 are attached to the bridge, no if they are not attached.

Not attached

```
eth0 --- br0 ---- veth0--|ns l2]--eth0

eth1 --- br1 ---- veth1--|ns l2]--eth0
```

Attached

```
eth0 ---          ---- veth0--|ns l2]--eth0
        |       |
         -- br0 --
        |       |
eth1 ---          ---- veth1--|ns l2]--eth0
```

But again, I am not sure.

>
> If yes, then below approach will work. If no, then we'll need something else

> since both L2s should get the packet in their own right.

It is a critical path for broadcast and multicast incoming traffic,
should I implement this approach and we try to optimize that later ?

>> 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
>>
>> The registered network namespace is a list shared between brothers l3
>> namespaces. This will add more overhead for sure. Does anyone have
>> comments on that or perhaps a better solution ?
>
> -vlad
>
> _____
> Containers mailing list
> Containers@lists.osdl.org
> https://lists.osdl.org/mailman/listinfo/containers

_____
Containers mailing list
Containers@lists.osdl.org
https://lists.osdl.org/mailman/listinfo/containers

Subject: Re: [RFC] L3 network isolation : broadcast
Posted by Daniel Lezcano on Wed, 13 Dec 2006 23:16:21 GMT
View Forum Message <> Reply to Message

Daniel Lezcano wrote:
> Vlad Yasevich wrote:
>> 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.
>> A question, if you will...  I am still digesting the l2 changes, and I can't
>> remember/find if the broadcasts will be replicated across multiple l2 or not.
>
> Well ... I am not sure (never tested it) but as far as I remember, it is
> the bridge which should duplicate the packets because it acts as a "hub".
>
> eth0 --- br0 ---- veth0--|ns l2]--eth0
>              |
>               -- veth1--|ns l2]--eth0
>              |
>               -- veth2--[ns l2]--eth0
```

>
> When a packet is received on eth0, it is forwarded to br0 (the bridge)
> and this one will send the packet to veth0, veth1 and veth2. The packets
> will follow the normal incoming path for each namespace. So I think the
> answer is yes, the broadcast is replicated to each l2 namespace.
>
> Dmitry can give more information on that I think.
>
>> Example:
>> A system has 2 interfaces eth0 and eth1 connected to the same lan/link.
>> Each NIC was isolated to it's own L2 space. Each L2 space configures
>> the its nic with unique IP but in the same subnet. Will both L2s receive
>> a subnet broadcast packet?
>
> Depending on the bridge configuration, I am inclined to say yes if eth0
> and eth1 are attached to the bridge, no if they are not attached.

Sorry I missed "eth0 and eth1 connected to the **same** lan/link"

So yes, each l2 namespace should receive the broadcast packets.

Subject: Re: [RFC] L3 network isolation : broadcast
Posted by Vlad Yasevich on Thu, 14 Dec 2006 15:14:32 GMT
View Forum Message <> Reply to Message

Daniel

Thanks. I think I just now found all this in the code (too much code to look at :)

Daniel Lezcano wrote:
[snip]
>
>>
>> If yes, then below approach will work. If no, then we'll need
>> something else
>> since both L2s should get the packet in their own right.
>
> It is a critical path for broadcast and multicast incoming traffic,
> should I implement this approach and we try to optimize that later ?
>
>>> 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
>>>
>>> The registered network namespace is a list shared between brothers l3
>>> namespaces. This will add more overhead for sure. Does anyone have
>>> comments on that or perhaps a better solution ?

In the above solution, you'll need to account for wildcard binds as well.

So, for l3, a match is (!rcv_saddr || rcv_saddr == loc_addr).

Should be easy enough with the right arguments. My suggestion is to
make broadcast case use a different match macro/function.

-vlad

_____

Containers mailing list
Containers@lists.osdl.org
https://lists.osdl.org/mailman/listinfo/containers