
Subject: L3 network isolation

Posted by [Daniel Lezcano](#) on Wed, 06 Dec 2006 23:25:45 GMT

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

Hi all,

Dmitry and I, we thought about a possible implementation allowing the I2/I3 to coexists.

The idea is assuming the I3 network namespaces are the leaf in the I2 namespace hierarchy tree. By default, init process is I2 namespace. From a layer 3, it is impossible to do a new network namespace unshare.

All the configuration is done into the I2 namespace. When a I3 is created a new IP address should be created into the I2 namespace and "pushed" into the I3. When the I3 dies, the IP is pulled to its parent, aka the I2. In order to ensure security into the I3, the NET_ADMIN capability is lost when doing unsharing for I3.

There is no extra code for socket virtualization. It is a common part.

How to setup a I3 namespace ?

- 1 - setup a new IP address in I2 namespace
- 2 - create a I3 namespace
- 3 - specific socket ioctl to "push" the IP address from the I2 namespace to the newly created I3 namespace

The I2 lose visibility on the IP address and I3 gains visibility on the IP address. A ifconfig or a ip command shows only the IP address assigned to the namespace. Loopback address is always visible.

How to handle outgoing traffic ?

The bind must be checked with the IP addresses belonging to the I3 namespace and with all the derivative addresses (multicast, broadcast, zero net, loopback, ...).

The IP addresses will rely on aliased IP address. The source address must be filled with the IP address belonging the I3 namespace when not set. This is a trivial operation, because we know which IP addresses are assigned to the I3 namespace.

When the route are resolved, the I3 namespace switch the its parent, that is to say the I2 namespace, and the virtualization follows its normal path.

How to handle incoming traffic ?

Because we can have several sockets listening on the same INADDR_ANY:port, we must find the network namespace associated with the destination IP address.

For unicast, this is a trivial operation, because that can be checked with the assigned IP address again. For broadcast and multicast, some extra work should be done in order to store the namespaces which are listening on a broadcast address. As soon as the namespace is found, we switch to it. This can be done with netfilters.

Routes and co.

- Routes: they are not isolated, each I3 namespace can see all the routes from the other namespaces. That allows the routing engine to see all the routes and choose the loopback when two network namespaces in the same host try to communicate.

- Cache: the routing cache must be isolated, otherwise the socket isolation will not work. The I3 namespace code does not impact the I2 namespace code and route cache isolation is a common part if the I3 namespace switching is done in the right place.

Dmitry has posted the I2 namespace relying on the net namespace empty framework, I will post the I3 namespace relying on the I2 namespace today or tomorrow.

-- Daniel

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

Subject: Re: L3 network isolation
Posted by [Herbert Poetzel](#) on Thu, 07 Dec 2006 19:43:25 GMT
[View Forum Message](#) <> [Reply to Message](#)

On Thu, Dec 07, 2006 at 12:25:45AM +0100, Daniel Lezcano wrote:

> Hi all,
>
> Dmitry and I, we thought about a possible implementation allowing the
> I2/I3 to coexists.
>

- > The idea is assuming the I3 network namespaces are the leaf in the I2 namespace hierarchy tree. By default, init process is I2 namespace. From a layer 3, it is impossible to do a new network namespace unshare.
- >
- > All the configuration is done into the I2 namespace. When a I3 is created a new IP address should be created into the I2 namespace and "pushed" into the I3. When the I3 dies, the IP is pulled to its parent, aka the I2. In order to ensure security into the I3, the NET_ADMIN capability is lost when doing unsharing for I3.
- > There is no extra code for socket virtualization. It is a common part.
- >
- > How to setup a I3 namespace ?
- > -----
- >
- > 1 - setup a new IP address in I2 namespace
- > 2 - create a I3 namespace
- > 3 - specific socket ioctl to "push" the IP address from the I2 namespace to the newly created I3 namespace
- >
- > The I2 lose visibility on the IP address and I3 gains visibility on the IP address.

why that?

I consider visibility of the IP addresses on the host (what you call I2 space) a feature ...

- > A ifconfig or a ip command shows only the IP address assigned to the namespace.

that is okay though ...

- > Loopback address is always visible.

is it also bindable?

- > How to handle outgoing traffic ?
- > -----
- >
- > The bind must be checked with the IP addresses belonging to the I3 namespace and with all the derivative addresses (multicast, broadcast, zero net, loopback, ...).
- >
- > The IP addresses will rely on aliased IP address.

hmm? please elaborate ...

- > The source address must be filled with the IP address belonging the I3 namespace when not set. This is a trivial operation, because we know

> which IP addresses are assigned to the I3 namespace.
>
> When the route are resolved, the I3 namespace switch the its parent,
> that is to say the I2 namespace, and the virtualization follows its
> normal path.
>
> How to handle incoming traffic ?
> -----
>
> Because we can have several sockets listening on the same
> INADDR_ANY:port, we must find the network namespace associated
> with the destination IP address.
> For unicast, this is a trivial operation, because that can be checked
> with the assigned IP address again. For broadcast and multicast, some
> extra work should be done in order to store the namespaces which are
> listening on a broadcast address. As soon as the namespace is found, we
> switch to it. This can be done with netfilters.

okay ...

> Routes and co.
> -----
>
> - Routes: they are not isolated, each I3 namespace can see all the
> routes from the other namespaces. That allows the routing engine to see
> all the routes and choose the loopback when two network namespaces in
> the same host try to communicate.
>
> - Cache: the routing cache must be isolated, otherwise the socket
> isolation will not work. The I3 namespace code does not impact the I2
> namespace code and route cache isolation is a common part if the I3
> namespace switching is done in the right place.
>
> Dmitry has posted the I2 namespace relying on the net namespace empty
> framework, I will post the I3 namespace relying on the I2 namespace
> today or tomorrow.

looking forward to it ...

best,
Herbert

> -- Daniel
>
>
> _____
> Containers mailing list
> Containers@lists.osdl.org
> <https://lists.osdl.org/mailman/listinfo/containers>

Subject: Re: L3 network isolation
Posted by [Vlad Yasevich](#) on Thu, 07 Dec 2006 21:33:27 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi Daniel

> Hi all,
>
> Dmitry and I, we thought about a possible implementation allowing the
> I2/I3 to coexists.
>
> The idea is assuming the I3 network namespaces are the leaf in the I2
> namespace hierarchy tree. By default, init process is I2 namespace. From
> a layer 3, it is impossible to do a new network namespace unshare.
>
> All the configuration is done into the I2 namespace. When a I3 is
> created a new IP address should be created into the I2 namespace and
> "pushed" into the I3. When the I3 dies, the IP is pulled to its parent,
> aka the I2. In order to ensure security into the I3, the NET_ADMIN
> capability is lost when doing unsharing for I3.
> There is no extra code for socket virtualization. It is a common part.
>
> How to setup a I3 namespace ?
> -----
>
> 1 - setup a new IP address in I2 namespace
> 2 - create a I3 namespace
> 3 - specific socket ioctl to "push" the IP address from the I2
> namespace to the newly created I3 namespace

This means that there is some kind of identifier for the I3 namespace, right?

>
> The I2 lose visibility on the IP address and I3 gains visibility on the
> IP address. A ifconfig or a ip command shows only the IP address
> assigned to the namespace. Loopback address is always visible.

Hmm.... I've been thinking about this, and I think this OK from the sockets point of view, i.e. binds() in I2 lose visibility to the new I3 address. There is a concern for a potential race here though.

However, it would be really nice to be able to see I3 namespace addresses in

the parent I2 tagged in some way.

- >
- > How to handle outgoing traffic ?
- > -----
- >
- > The bind must be checked with the IP addresses belonging to the I3
- > namespace and with all the derivative addresses (multicast, broadcast,
- > zero net, loopback, ...).
- >
- > The IP addresses will rely on aliased IP address. The source address
- > must be filled with the IP address belonging the I3 namespace when not
- > set. This is a trivial operation, because we know which IP addresses are
- > assigned to the I3 namespace.

Can you provide a little more info?

- >
- > When the route are resolved, the I3 namespace switch the its parent,
- > that is to say the I2 namespace, and the virtualization follows its
- > normal path.
- >
- > How to handle incoming traffic ?
- > -----
- >
- > Because we can have several sockets listening on the same
- > INADDR_ANY:port, we must find the network namespace associated with the
- > destination IP address.
- > For unicast, this is a trivial operation, because that can be checked
- > with the assigned IP address again. For broadcast and multicast, some
- > extra work should be done in order to store the namespaces which are
- > listening on a broadcast address. As soon as the namespace is found, we
- > switch to it. This can be done with netfilters.

The problem is with multicasts. Multicast groups are joined on the interface bases. Every socket that bound *:multicast_port will receive multicast traffic once a single app joined the group. Since I3 namespaces don't have share the conceptual interface, theoretically, all I3 namespaces should receive multicast traffic.

- >
- > Routes and co.
- > -----
- >
- > - Routes: they are not isolated, each I3 namespace can see all the
- > routes from the other namespaces. That allows the routing engine to see
- > all the routes and choose the loopback when two network namespaces in
- > the same host try to communicate.

>
> - Cache: the routing cache must be isolated, otherwise the socket
> isolation will not work. The I3 namespace code does not impact the I2
> namespace code and route cache isolation is a common part if the I3
> namespace switching is done in the right place.
>
>
> Dmitry has posted the I2 namespace relying on the net namespace empty
> framework, I will post the I3 namespace relying on the I2 namespace
> today or tomorrow.
>

Looking forward to it.

Thanks
-vlad

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

Subject: Re: L3 network isolation
Posted by [Daniel Lezcano](#) on Thu, 07 Dec 2006 22:08:05 GMT
[View Forum Message](#) <> [Reply to Message](#)

Herbert Poetzi wrote:

> On Thu, Dec 07, 2006 at 12:25:45AM +0100, Daniel Lezcano wrote:
>> Hi all,
>>
>> Dmitry and I, we thought about a possible implementation allowing the
>> I2/I3 to coexists.
>>
>> The idea is assuming the I3 network namespaces are the leaf in the I2
>> namespace hierarchy tree. By default, init process is I2 namespace. From
>> a layer 3, it is impossible to do a new network namespace unshare.
>>
>> All the configuration is done into the I2 namespace. When a I3 is
>> created a new IP address should be created into the I2 namespace and
>> "pushed" into the I3. When the I3 dies, the IP is pulled to its parent,
>> aka the I2. In order to ensure security into the I3, the NET_ADMIN
>> capability is lost when doing unsharing for I3.
>> There is no extra code for socket virtualization. It is a common part.
>>
>> How to setup a I3 namespace ?
>> -----
>>

```
>> 1 - setup a new IP address in I2 namespace
>> 2 - create a I3 namespace
>> 3 - specific socket ioctl to "push" the IP address from the I2
>> namespace to the newly created I3 namespace
>>
>> The I2 lose visibility on the IP address and I3 gains visibility on
>> the IP address.
>
> why that?
> I consider visibility of the IP addresses on the host
> (what you call I2 space) a feature ...
```

Perhaps the sentence is malformed. I mean, you set an IP address in the layer 2, you do `ifconfig/ip` => you see it. The IP is pushed to I3, you do again `ifconfig/ip` in the I2 namespace and you do not see it. This is related to the section below.

```
>
>> A ifconfig or a ip command shows only the IP address
>> assigned to the namespace.
>
> that is okay though ...
>
>> Loopback address is always visible.
>
> is it also bindable?
```

Yes, bindable, usable, isolated. I think the loopback isolation should be enabled/disabled by configuration in order to let the application to communicate with portmap.

```
>
>> How to handle outgoing traffic ?
>> -----
>>
>> The bind must be checked with the IP addresses belonging to the I3
>> namespace and with all the derivative addresses (multicast, broadcast,
>> zero net, loopback, ...).
>>
>> The IP addresses will rely on aliased IP address.
>
> hmm? please elaborate ...
```

If you create 5 IP address, `1.2.3.[1-5]/24`, the IP `1.2.3.1` will be the primary address and `1.2.3.[2-4]` will be secondaries IP addresses. You create five I3 namespaces and assign each IP to each namespace. So we have:

```
namespace 1 -> 1.2.3.1/24
namespace 2 -> 1.2.3.2/24
```

```
....
```

If namespace 2 connects to 1.2.3.100 for example, the routing engine will choose the primary address as source address if it was not specified by a bind, which is the usual case for a connection. The peer 1.2.3.100 will answer to 1.2.3.1 instead of 1.2.3.2 => RST

```
>
>> The source address must be filled with the IP address belonging the I3
>> namespace when not set. This is a trivial operation, because we know
>> which IP addresses are assigned to the I3 namespace.
>>
>> When the route are resolved, the I3 namespace switch the its parent,
>> that is to say the I2 namespace, and the virtualization follows its
>> normal path.
>>
>> How to handle incoming traffic ?
>> -----
>>
>> Because we can have several sockets listening on the same
>> INADDR_ANY:port, we must find the network namespace associated
>> with the destination IP address.
>> For unicast, this is a trivial operation, because that can be checked
>> with the assigned IP address again. For broadcast and multicast, some
>> extra work should be done in order to store the namespaces which are
>> listening on a broadcast address. As soon as the namespace is found, we
>> switch to it. This can be done with netfilters.
>
> okay ...
>
>> Routes and co.
>> -----
>>
>> - Routes: they are not isolated, each I3 namespace can see all the
>> routes from the other namespaces. That allows the routing engine to see
>> all the routes and choose the loopback when two network namespaces in
>> the same host try to communicate.
>>
>> - Cache: the routing cache must be isolated, otherwise the socket
>> isolation will not work. The I3 namespace code does not impact the I2
>> namespace code and route cache isolation is a common part if the I3
>> namespace switching is done in the right place.
>>
>> Dmitry has posted the I2 namespace relying on the net namespace empty
>> framework, I will post the I3 namespace relying on the I2 namespace
>> today or tomorrow.
>
> looking forward to it ...
>
```

> best,
> Herbert
>
>> -- Daniel
>>
>>
>> _____
>> 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: L3 network isolation
Posted by [Daniel Lezcano](#) on Thu, 07 Dec 2006 22:33:30 GMT
[View Forum Message](#) <> [Reply to Message](#)

Vlad Yasevich wrote:

> Hi Daniel
>
>> Hi all,
>>
>> Dmitry and I, we thought about a possible implementation allowing the
>> I2/I3 to coexists.
>>
>> The idea is assuming the I3 network namespaces are the leaf in the I2
>> namespace hierarchy tree. By default, init process is I2 namespace. From
>> a layer 3, it is impossible to do a new network namespace unshare.
>>
>> All the configuration is done into the I2 namespace. When a I3 is
>> created a new IP address should be created into the I2 namespace and
>> "pushed" into the I3. When the I3 dies, the IP is pulled to its parent,
>> aka the I2. In order to ensure security into the I3, the NET_ADMIN
>> capability is lost when doing unsharing for I3.
>> There is no extra code for socket virtualization. It is a common part.
>>
>> How to setup a I3 namespace ?
>> -----
>>
>> 1 - setup a new IP address in I2 namespace
>> 2 - create a I3 namespace
>> 3 - specific socket ioctl to "push" the IP address from the I2
>> namespace to the newly created I3 namespace
>
> This means that there is some kind of identifier for the I3 namespace, right?

Not exactly. The `bind_ns` allows to assign an identifier to a namespace. The namespace is an aggregation of the different namespace resources (`ipc`, `pid`, `net`, `utsname`, ...). But the result is the same, we use the namespace identifier instead of a `l3` namespace identifier.

>
>> The `l2` lose visibility on the IP address and `l3` gains visibility on the
>> IP address. A `ifconfig` or a `ip` command shows only the IP address
>> assigned to the namespace. Loopback address is always visible.
>
> Hmm.... I've been thinking about this, and I think this OK from the sockets point
> of view, i.e. `binds()` in `l2` lose visibility to the new `l3` address. There is
> a concern for a potential race here though.

Do you mean, someone in the `l2` namespace can use the IP address before pushing it the `l3` namespace ? That is right, perhaps the call should be done in one shoot (`set address + pushing it to l3`)

> However, it would be really nice to be able to see `l3` namespace addresses in
> the parent `l2` tagged in some way.

>
>> How to handle outgoing traffic ?
>> -----
>>
>> The `bind` must be checked with the IP addresses belonging to the `l3`
>> namespace and with all the derivative addresses (multicast, broadcast,
>> zero net, loopback, ...).
>>
>> The IP addresses will rely on aliased IP address. The source address
>> must be filled with the IP address belonging the `l3` namespace when not
>> set. This is a trivial operation, because we know which IP addresses are
>> assigned to the `l3` namespace.
>
> Can you provide a little more info?

I think I already answered this question in the previous email. I am afraid this paragraph is not very clear ... ;)

>
>> When the route are resolved, the `l3` namespace switch the its parent,
>> that is to say the `l2` namespace, and the virtualization follows its
>> normal path.
>>
>> How to handle incoming traffic ?
>> -----

>>
>> Because we can have several sockets listening on the same
>> INADDR_ANY:port, we must find the network namespace associated with the
>> destination IP address.
>> For unicast, this is a trivial operation, because that can be checked
>> with the assigned IP address again. For broadcast and multicast, some
>> extra work should be done in order to store the namespaces which are
>> listening on a broadcast address. As soon as the namespace is found, we
>> switch to it. This can be done with netfilters.
>
> The problem is with multicasts. Multicast groups are joined on the interface
> bases. Every socket that bound *:multicast_port will receive multicast
> traffic once a single app joined the group. Since I3 namespaces don't have
> share the conceptual interface, theoretically, all I3 namespaces should receive
> multicast traffic.

Right. You sunk my battleship :)
Need to be thought...

>
>> Routes and co.
>> -----
>>
>> - Routes: they are not isolated, each I3 namespace can see all the
>> routes from the other namespaces. That allows the routing engine to see
>> all the routes and choose the loopback when two network namespaces in
>> the same host try to communicate.
>>
>> - Cache: the routing cache must be isolated, otherwise the socket
>> isolation will not work. The I3 namespace code does not impact the I2
>> namespace code and route cache isolation is a common part if the I3
>> namespace switching is done in the right place.
>>
>>
>> Dmitry has posted the I2 namespace relying on the net namespace empty
>> framework, I will post the I3 namespace relying on the I2 namespace
>> today or tomorrow.
>>
>
> Looking forward to it.

Fixing a kref problem...

Thanks for all your comments.

-- Daniel

Containers mailing list

Subject: [RFC] L3 network isolation : broadcast
Posted by [Daniel Lezcano](#) on Wed, 13 Dec 2006 20:43:22 GMT
[View Forum Message](#) <> [Reply to Message](#)

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 I2 namespaces because they share the socket hash table. But that is not correct for I3 namespaces because we want to deliver the packet to each I3 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 I2 isolation because an another I2 namespace could have binded to the same broadcast address.

The solution I see here is:

```
if namespace is I3 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 I3 namespaces. This will add more overhead for sure. Does anyone have comments on that or perhaps a better solution ?

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