Subject: [PATCH 0/3] IPv6 start/stop problems
Posted by den on Tue, 18 Mar 2008 14:32:41 GMT
View Forum Message <> Reply to Message

Hello, Dave!

We have faced a several problems with IPv6 start/stop on 2.6.18 RHEL5
kernel in OpenVz. The code in the 2.6.25 does not differ from 2.6.18 in
respect to this.

Regards,
Den

_____
Containers mailing list
Containers@lists.linux-foundation.org
https://lists.linux-foundation.org/mailman/listinfo/containers

Subject: [PATCH 1/3] [IPV6]: Event type in addrconf_ifdown is mis-used.
Posted by den on Tue, 18 Mar 2008 14:35:23 GMT
View Forum Message <> Reply to Message

addrconf_ifdown is broken in respect to the usage of how parameter. This
function is called with (event != NETDEV_DOWN) and (2) on the IPv6 stop.
It the latter case inet6_dev from loopback device should be destroyed.

Signed-off-by: Denis V. Lunev <den@openvz.org>
---
net/ipv6/addrconf.c | 10 +++++-----
1 files changed, 5 insertions(+), 5 deletions(-)

```
diff --git a/net/ipv6/addrconf.c b/net/ipv6/addrconf.c
index 4b86d38..d68e8f5 100644
--- a/net/ipv6/addrconf.c
+++ b/net/ipv6/addrconf.c
@@ -2457,7 +2457,7 @@ static int addrconf_ifdown(struct net_device *dev, int how)
  /* Step 1: remove reference to ipv6 device from parent device.
     Do not dev_put!
   */
- if (how == 1) {
+ if (how) {
   idev->dead = 1;

   /* protected by rtnl_lock */
@@ -2489,12 +2489,12 @@ static int addrconf_ifdown(struct net_device *dev, int how)
  write_lock_bh(&idev->lock);
```

```
  /* Step 3: clear flags for stateless addrconf */
- if (how != 1)
+ if (!how)
   idev->if_flags &= ~(IF_RS_SENT|IF_RA_RCVD|IF_READY);

  /* Step 4: clear address list */
 #ifdef CONFIG_IPV6_PRIVACY
- if (how == 1 && del_timer(&idev->regen_timer))
+ if (how && del_timer(&idev->regen_timer))
   in6_dev_put(idev);

  /* clear tempaddr list */
@@ -2531,7 +2531,7 @@ static int addrconf_ifdown(struct net_device *dev, int how)

  /* Step 5: Discard multicast list */

- if (how == 1)
+ if (how)
   ipv6_mc_destroy_dev(idev);
  else
   ipv6_mc_down(idev);
@@ -2540,7 +2540,7 @@ static int addrconf_ifdown(struct net_device *dev, int how)

  /* Shot the device (if unregistered) */

- if (how == 1) {
+ if (how) {
   addrconf_sysctl_unregister(idev);
   neigh_parms_release(&nd_tbl, idev->nd_parms);
   neigh_ifdown(&nd_tbl, dev);
--
1.5.3.rc5
```

Subject: [PATCH 2/3] [IPV6]: inet6_dev on loopback should be kept until namespace stop.

Posted by den on Tue, 18 Mar 2008 14:35:24 GMT

View Forum Message <> Reply to Message

In the other case it will be destroyed when last address will be removed from lo inside a namespace. This will break IPv6 in several places. The most obvious one is ip6_dst_ifdown.

Signed-off-by: Denis V. Lunev <den@openvz.org>
---
 net/ipv6/addrconf.c | 2 +-
 1 files changed, 1 insertions(+), 1 deletions(-)

diff --git a/net/ipv6/addrconf.c b/net/ipv6/addrconf.c
index d68e8f5..40784ea 100644
--- a/net/ipv6/addrconf.c
+++ b/net/ipv6/addrconf.c
@@ -2444,7 +2444,7 @@ static int addrconf_ifdown(struct net_device *dev, int how)

  ASSERT_RTNL();

- if (dev == init_net.loopback_dev && how == 1)
+ if ((dev->flags & IFF_LOOPBACK) && how == 1)
   how = 0;

  rt6_ifdown(net, dev);
--
1.5.3.rc5

_____
Containers mailing list
Containers@lists.linux-foundation.org
https://lists.linux-foundation.org/mailman/listinfo/containers

---

Subject: [PATCH 3/3] [IPV6]: Fix refcounting for anycast dst entries.
Posted by den on Tue, 18 Mar 2008 14:35:25 GMT
View Forum Message <> Reply to Message

Anycast DST entries allocated inside ipv6_dev_ac_inc are leaked when
network device is stopped without removing IPv6 addresses from it. The
bug has been observed in the reality on 2.6.18-rhel5 kernel.

In the above case addrconf_ifdown marks all entries as obsolete and
ip6_del_rt called from __ipv6_dev_ac_dec returns ENOENT. The referrence is
not dropped.

The fix is simple. DST entry should not keep referrence when stored in the
FIB6 tree.

Signed-off-by: Denis V. Lunev <den@openvz.org>
---
 net/ipv6/anycast.c | 9 ++-------
 1 files changed, 2 insertions(+), 7 deletions(-)

diff --git a/net/ipv6/anycast.c b/net/ipv6/anycast.c
index 96868b9..7bc0469 100644

```
--- a/net/ipv6/anycast.c
+++ b/net/ipv6/anycast.c
@@ -334,9 +334,7 @@ int ipv6_dev_ac_inc(struct net_device *dev, struct in6_addr *addr)
  idev->ac_list = aca;
  write_unlock_bh(&idev->lock);

- dst_hold(&rt->u.dst);
- if (ip6_ins_rt(rt))
-  dst_release(&rt->u.dst);
+ ip6_ins_rt(rt);

  addrconf_join_solict(dev, &aca->aca_addr);

@@ -378,10 +376,7 @@ int __ipv6_dev_ac_dec(struct inet6_dev *idev, struct in6_addr *addr)
  addrconf_leave_solict(idev, &aca->aca_addr);

  dst_hold(&aca->aca_rt->u.dst);
- if (ip6_del_rt(aca->aca_rt))
-  dst_free(&aca->aca_rt->u.dst);
- else
-  dst_release(&aca->aca_rt->u.dst);
+ ip6_del_rt(aca->aca_rt);

  aca_put(aca);
  return 0;
--
1.5.3.rc5
```

Subject: Re: [PATCH 1/3] [IPV6]: Event type in addrconf_ifdown is mis-used.
Posted by davem on Sun, 23 Mar 2008 00:38:43 GMT
View Forum Message <> Reply to Message

From: "Denis V. Lunev" <den@openvz.org>
Date: Tue, 18 Mar 2008 17:35:23 +0300

> addrconf_ifdown is broken in respect to the usage of how parameter. This
> function is called with (event != NETDEV_DOWN) and (2) on the IPv6 stop.
> It the latter case inet6_dev from loopback device should be destroyed.
>
> Signed-off-by: Denis V. Lunev <den@openvz.org>

The code purposefully treats "2" specially because when IPV6 routes

are destroyed they are changed to point to the loopback device's
inet6_dev object.

This allows statistic bumping code to not have to check if it has a
NULL inet6_dev pointer or not, because that's now impossible.

Since ipv6 is not unloadable, addrconf_cleanup(), and thus the
"how == 2" case can only occur when ipv6 fails to load properly.
The only real consequence of this bug is that if ipv6 fails
to load properly, a subsequent successfull load of ipv6 will
leak the loopback device's inet6_dev object, which isn't that
much of a big deal.

I understand that for namespaces you have to deal with multiple
loopback devices, but you'll need to solve that problem while
still handling the wish of the ipv6 stack for inet6_dev objects
of loopback devices to be permanent and guarenteed to always
be around for the sake of statistics bumping.

I thus can't apply any of these patches until those issues are
resolved.

_____
Containers mailing list
Containers@lists.linux-foundation.org
https://lists.linux-foundation.org/mailman/listinfo/containers

---

Subject: Re: [PATCH 1/3] [IPV6]: Event type in addrconf_ifdown is mis-used.
Posted by den on Sun, 23 Mar 2008 08:13:16 GMT
View Forum Message <> Reply to Message

On Sat, 2008-03-22 at 17:38 -0700, David Miller wrote:
> From: "Denis V. Lunev" <den@openvz.org>
> Date: Tue, 18 Mar 2008 17:35:23 +0300
>
> > addrconf_ifdown is broken in respect to the usage of how parameter. This
> > function is called with (event != NETDEV_DOWN) and (2) on the IPv6 stop.
> > It the latter case inet6_dev from loopback device should be destroyed.
> >
> > Signed-off-by: Denis V. Lunev <den@openvz.org>
>
> The code purposefully treats "2" specially because when IPV6 routes
> are destroyed they are changed to point to the loopback device's
> inet6_dev object.
>
> This allows statistic bumping code to not have to check if it has a
> NULL inet6_dev pointer or not, because that's now impossible.
>

> Since ipv6 is not unloadable, addrconf_cleanup(), and thus the
> "how == 2" case can only occur when ipv6 fails to load properly.
> The only real consequence of this bug is that if ipv6 fails
> to load properly, a subsequent successfull load of ipv6 will
> leak the loopback device's inet6_dev object, which isn't that
> much of a big deal.
>
> I understand that for namespaces you have to deal with multiple
> loopback devices, but you'll need to solve that problem while
> still handling the wish of the ipv6 stack for inet6_dev objects
> of loopback devices to be permanent and guarenteed to always
> be around for the sake of statistics bumping.

First, this behaviour is broken for a namespace right now in the 2.6.26 tree. inet6_dev pointer will be NULL for a loopback inside the namespace. The case is simple. Just remove all INET6 addresses from a loopback device inside a VE. This will call

```
  inet6_addr_del
    addrconf_ifdown(dev, 1);
      if (dev == init_net.loopback_dev && how == 1)
              how = 0;
```

the condition will be false and how will not be changed here.

Pls note, that ip6_dst_ifdown deals with a namespace loopback rather than init_net loopback to track referrences of the namespace objects. This allows us to catch refcounting bugs smoothly (see patch 3 in the set).

That's why I have extended a special "2" case to really destroy inet6_dev to have a way to destroy it. Generic code should not suffer from this from my POW.

> I thus can't apply any of these patches until those issues are
> resolved.

IMHO special "2" case was intended to have a stub to unload the module in the future.

Subject: Re: [PATCH 1/3] [IPV6]: Event type in addrconf_ifdown is mis-used.
Posted by davem on Sun, 23 Mar 2008 10:17:24 GMT
View Forum Message <> Reply to Message

From: "Denis V. Lunev" <den@openvz.org>
Date: Sun, 23 Mar 2008 11:13:16 +0300

> First, this behaviour is broken for a namespace right now in the 2.6.26
> tree. inet6_dev pointer will be NULL for a loopback inside the
> namespace. The case is simple. Just remove all INET6 addresses from a
> loopback device inside a VE. This will call
> inet6_addr_del
> addrconf_ifdown(dev, 1);
> if (dev == init_net.loopback_dev && how == 1)
> how = 0;
> the condition will be false and how will not be changed here.

That's a bug.

You can't mark any namespace's loopback device's inet6_dev as NULL
until you know that all routes, devices, and packets referring to such
devices and routes in that namespace are %100 gone and unreferenced.

It is now obviously apparent that there are several severe errors
here.

_____
Containers mailing list
Containers@lists.linux-foundation.org
https://lists.linux-foundation.org/mailman/listinfo/containers

---

Subject: Re: [PATCH 1/3] [IPV6]: Event type in addrconf_ifdown is mis-used.
Posted by den on Sun, 23 Mar 2008 14:34:59 GMT
View Forum Message <> Reply to Message

On Sun, 2008-03-23 at 03:17 -0700, David Miller wrote:
> From: "Denis V. Lunev" <den@openvz.org>
> Date: Sun, 23 Mar 2008 11:13:16 +0300
>
> > First, this behaviour is broken for a namespace right now in the 2.6.26
> > tree. inet6_dev pointer will be NULL for a loopback inside the
> > namespace. The case is simple. Just remove all INET6 addresses from a
> > loopback device inside a VE. This will call
> > inet6_addr_del
> > addrconf_ifdown(dev, 1);
> > if (dev == init_net.loopback_dev && how == 1)
> > how = 0;
> > the condition will be false and how will not be changed here.
>
> That's a bug.
>
> You can't mark any namespace's loopback device's inet6_dev as NULL

> until you know that all routes, devices, and packets referring to such
> devices and routes in that namespace are %100 gone and unreferenced.
>
> It is now obviously apparent that there are several severe errors
> here.

You are perfectly correct and the place in addrconf_cleanup is that
place when we believe that we should destroy all the staff.

You see, it is pretty useless to call addrconf_ifdown(dev, 2) after
addrconf_dev(dev, 0) for a loopback in the current code! No new cleanups
will be performed for 2, pls check :)

_____
Containers mailing list
Containers@lists.linux-foundation.org
https://lists.linux-foundation.org/mailman/listinfo/containers

---

Subject: Re: [PATCH 1/3] [IPV6]: Event type in addrconf_ifdown is mis-used.
Posted by davem on Mon, 24 Mar 2008 05:49:05 GMT
View Forum Message <> Reply to Message

From: "Denis V. Lunev" <den@openvz.org>
Date: Sun, 23 Mar 2008 17:34:59 +0300

> You are perfectly correct and the place in addrconf_cleanup is that
> place when we believe that we should destroy all the staff.
>
> You see, it is pretty useless to call addrconf_ifdown(dev, 2) after
> addrconf_dev(dev, 0) for a loopback in the current code! No new cleanups
> will be performed for 2, pls check :)

Ok, I'll take another close look at this and apply your patches
if I agree with you :-)

_____
Containers mailing list
Containers@lists.linux-foundation.org
https://lists.linux-foundation.org/mailman/listinfo/containers

---

Subject: Re: [PATCH 0/3] IPv6 start/stop problems
Posted by den on Mon, 31 Mar 2008 08:38:01 GMT
View Forum Message <> Reply to Message

On Tue, 2008-03-18 at 17:32 +0300, Denis V. Lunev wrote:
> Hello, Dave!

>
> We have faced a several problems with IPv6 start/stop on 2.6.18 RHEL5
> kernel in OpenVz. The code in the 2.6.25 does not differ from 2.6.18 in
> respect to this.

Hi, Dave!

Have you changed you mind about this?

Regards,
 Den

_____
Containers mailing list
Containers@lists.linux-foundation.org
https://lists.linux-foundation.org/mailman/listinfo/containers

---

Subject: Re: [PATCH 0/3] IPv6 start/stop problems
Posted by davem on Mon, 31 Mar 2008 08:41:01 GMT
View Forum Message <> Reply to Message

From: "Denis V. Lunev" <den@parallels.com>
Date: Mon, 31 Mar 2008 12:38:01 +0400

> On Tue, 2008-03-18 at 17:32 +0300, Denis V. Lunev wrote:
> > Hello, Dave!
> >
> > We have faced a several problems with IPv6 start/stop on 2.6.18 RHEL5
> > kernel in OpenVz. The code in the 2.6.25 does not differ from 2.6.18 in
> > respect to this.
>
> Hi, Dave!
>
> Have you changed you mind about this?

I still haven't gotten around to reviewing this stuff yet.
When I do, I'll be sure to let you know. :-)

_____
Containers mailing list
Containers@lists.linux-foundation.org
https://lists.linux-foundation.org/mailman/listinfo/containers

---

Subject: Re: [PATCH 1/3] [IPV6]: Event type in addrconf_ifdown is mis-used.
Posted by davem on Thu, 03 Apr 2008 20:33:56 GMT
View Forum Message <> Reply to Message

From: "Denis V. Lunev" <den@openvz.org>
Date: Sun, 23 Mar 2008 17:34:59 +0300

> You are perfectly correct and the place in addrconf_cleanup is that
> place when we believe that we should destroy all the staff.
>
> You see, it is pretty useless to call addrconf_ifdown(dev, 2) after
> addrconf_dev(dev, 0) for a loopback in the current code! No new cleanups
> will be performed for 2, pls check :)

I've rereviewed these three patches and I agree with your
assesment.

Therefore, I've applied these three patches to net-2.6 and
will push them out after some build validation.

Thanks!