Subject: Re: Cannot add VLAN devices to guest with new kernel
Posted by Andrey Mirkin on Tue, 19 Dec 2006 12:28:18 GMT
View Forum Message <> Reply to Message

Hello Benny,

I have prepared a patch. I have tested it on several configurations. Please
try it too.
BTW, if you are moving VLAN from VE0 to VE then you will not be able to reach
VE by this VLAN from VE0 (because there are no any device in VE0 which can
set VLAN tag on packet). This behaviour is existing on all kernels. If you
want to be able to connect to VE via VLAN from VE0 then please create VLAN in
VE0 and VE.

Thanks for your patience and help!

Andrey

On Tuesday 19 December 2006 11:54 Benny Amorsen wrote:
> I opened bug 395 because of what I considered a regression in test007
> compared to test005. The original message:
>
> [root@router01 ~]# uname -r
> 2.6.18-ovz028test007.1-smp
> [root@router01 ~]# vzctl start 114
> Starting VE ...
> VE is mounted
> Setting CPU units: 1000
> VE start in progress...
> [root@router01 ~]# vzctl set 114 --netdev_add eth0.114 --save
> Unable to add netdev eth0.114: Operation not permitted
> Saved parameters for VE 114
>
> It works with test005.
>
> There was quite a lot of back and forth between Andrey Mirkin and me.
> So far it ended with this comment from Andrey Mirkin:
>
> "It is not secure to use VLANs from VE0 (by --netdev_add) inside VE as
> they will operate from VE0 context."
>
> It confuses me that it is possible to use --netdev_add for physical
> ethernet devices, but not for VLAN devices.
>
> The alternative solution is to create veth devices and bridge them to
> the VLAN devices in VE0. This is not very attractive, because there
> will be hundreds of them in my setup.
>

```
> Good ideas and explanations welcome...
>
>
> /Benny
>
>
--- ./net/8021q/vlan_dev.c.vlan 2006-10-25 11:31:25.000000000 +0400
+++ ./net/8021q/vlan_dev.c 2006-12-19 14:01:32.000000000 +0300
@@ -436,6 +436,7 @@ int vlan_dev_hard_header(struct sk_buff

 int vlan_dev_hard_start_xmit(struct sk_buff *skb, struct net_device *dev)
 {
+ struct ve_struct *env;
  struct net_device_stats *stats = vlan_dev_get_stats(dev);
  struct vlan_ethhdr *veth = (struct vlan_ethhdr *)(skb->data);

@@ -489,13 +490,17 @@ int vlan_dev_hard_start_xmit(struct sk_b
  stats->tx_bytes += skb->len;

  skb->dev = VLAN_DEV_INFO(dev)->real_dev;
+ skb->owner_env = skb->dev->owner_env;
+ env = set_exec_env(skb->owner_env);
  dev_queue_xmit(skb);
+ set_exec_env(env);

  return 0;
 }

 int vlan_dev_hwaccel_hard_start_xmit(struct sk_buff *skb, struct net_device *dev)
 {
+ struct ve_struct *env;
  struct net_device_stats *stats = vlan_dev_get_stats(dev);
  unsigned short veth_TCI;

@@ -513,7 +518,10 @@ int vlan_dev_hwaccel_hard_start_xmit(str
  stats->tx_bytes += skb->len;

  skb->dev = VLAN_DEV_INFO(dev)->real_dev;
+ skb->owner_env = skb->dev->owner_env;
+ env = set_exec_env(skb->owner_env);
  dev_queue_xmit(skb);
+ set_exec_env(env);

  return 0;
 }
--- ./net/8021q/vlan.c.vlan 2006-11-22 19:40:37.000000000 +0300
+++ ./net/8021q/vlan.c 2006-12-19 14:01:34.000000000 +0300
@@ -378,7 +378,8 @@ static void vlan_setup(struct net_device
```

```
  new_dev->set_multicast_list = vlan_dev_set_multicast_list;
  new_dev->destructor = free_netdev;
  new_dev->do_ioctl = vlan_dev_ioctl;
- new_dev->features |= NETIF_F_VIRTUAL;
+ if (!ve_is_super(get_exec_env()))
+  new_dev->features |= NETIF_F_VIRTUAL;
 }

 static void vlan_transfer_operstate(const struct net_device *dev, struct net_device *vlandev)
@@ -636,6 +637,7 @@ static int vlan_device_event(struct noti
  struct vlan_group *grp;
  int i, flgs;
  struct net_device *vlandev;
+ struct ve_struct *env;

  grp = __vlan_find_group(dev->ifindex, dev->owner_env);
  if (!grp)
@@ -699,7 +701,9 @@ static int vlan_device_event(struct noti
    ret = unregister_vlan_dev(dev,
        VLAN_DEV_INFO(vlandev)->vlan_id);

+   env = set_exec_env(vlandev->owner_env);
    unregister_netdevice(vlandev);
+   set_exec_env(env);

    /* Group was destroyed? */
    if (ret == 1)
```

File Attachments

1) diff-net-vlan-virt-fix, downloaded 429 times