Subject: [PATCH] Routing table change in vps-functions for complex setups
Posted by Christian Hofstaedtle on Sat, 08 Mar 2008 11:57:54 GMT
View Forum Message <> Reply to Message

Hello!

I'd like to propose a change to vps-functions, to allow for more complex routing setups (with multiple VLANs bound on VE0, etc.).

The change would modify vzaddrouting and vzdelrouting to always add the VE0 source routing to the "local" table. This way, all routing decisions regarding _local_ VEs will always be done at the very top in the routing stack.
Therefore you can do other routing decisions, which would affect the reachability of the local VEs lower in the routing stack, without affecting the local VEs.
Now this all sounds very complicated, but the patch is very simple, and it should not affect "normal" setups.

I'm attaching the patch which we are currently running in production on 5 HNs.

Everything tested with IPv4 only, though; I'm also not so sure that modifying the "local" table is the best choice -- OTOH the VEs are local to the HN.

Because of the iproute table usage, the kernel needs to have 'Advanced Routing' set, but I'd think the OpenVZ kernels have this on / this is not a new requirement.

- Christian

------- example setup & further explanations -------

Example setup (done on a Debian etch host, vzctl 3.0.22,
kernel 2.6.18-028stab053, custom config):

VE0 has got multiple VLAN devices:
eth0.110 -> 10.10.110.62/24 (this is used for management of VE0)
eth0.150 -> 10.10.150.249/24 (used for VEs)
eth0.152 -> 10.10.152.249/24 (used for VEs)

Please note that VLAN150 + 152 are not dedicated to this HN, other nodes also run VEs in these VLANs.
The VLANs are connected together by a single router, which does strict source IP filtering (i.e. packets from 10.10.110.0/24 are not allowed to come from VLAN110).

Main routing table on HN looks like this:

Destination	Gateway	Iface
10.10.152.0	0.0.0.0	eth0.152
10.10.150.0	0.0.0.0	eth0.150
10.10.110.0	0.0.0.0	eth0.110
0.0.0.0	10.10.110.1	eth0.110

Routing rules on HN:

```
# ip rule ls
0:      from all lookup 255
32763:  from 10.10.152.0/24 lookup 152
32764:  from 10.10.150.0/24 lookup 150
32765:  from 10.10.110.0/24 lookup 110
32766:  from all lookup main
32767:  from all lookup default

# ip route ls table 150
10.10.150.0/24 dev eth0.150  scope link
default via 10.10.150.1 dev eth0.150
```

Example VE2:

```
cat /etc/vz/conf/2.conf | grep IP_
IP_ADDRESS="10.10.150.244"
```

On VE2 startup, with the original vps-functions, source routes will be configured in the "main" routing table. The "main" routing table will not be considered in this setup, because table 150 will be used, which already contains a (correct) default gateway. This also implies that Proxy ARP requests for VE2 will not be handled, because the kernel does not find the IP address of VE2 in its routing table.

With the patched vps-functions, the source route will be added to the local table instead, and Proxy ARP requests can be handled, because the kernel will see the IP address of VE2. The rules for 10.10.150.0/24 will be ignored during Proxy ARP (lookup can be fulfilled already in the "local" table), but outgoing packets will still use the rules for 10.10.150.0/24.

------- end of example -------

--
christian hofstaedtler

```
--- vps-functions 2008-03-05 15:42:02.000000000 +0100
+++ vps-functions 2008-03-05 16:30:03.000000000 +0100
@@ -193,14 +193,14 @@
    vzerror "Unable to get source ip [${VE_ROUTE_SRC_DEV}]" $VZ_CANT_ADDIP
   src_addr="src $src_addr"
  fi
- ${IP_CMD} route add "$1" dev venet0 $src_addr ||
-  vzerror "Unable to add route ${IP_CMD} route add $1 dev venet0 $src_addr"
$VZ_CANT_ADDIP
+ ${IP_CMD} route add "$1" dev venet0 $src_addr table local ||
+  vzerror "Unable to add route ${IP_CMD} route add $1 dev venet0 $src_addr table local"
$VZ_CANT_ADDIP
 }

 vzaddrouting6()
 {
- ${IP_CMD} route add "$1" dev venet0 ||
-  vzerror "Unable to add route ${IP_CMD} route add $1 dev venet0" $VZ_CANT_ADDIP
+ ${IP_CMD} route add "$1" dev venet0 table local ||
+  vzerror "Unable to add route ${IP_CMD} route add $1 dev venet0 table local"
$VZ_CANT_ADDIP
 }

 # Sets VE0 source routing for given IP
@@ -228,9 +228,9 @@
  local arg

  if [ "${1%%:*}" = "$1" ]; then
-  arg="route del $1 dev venet0"
+  arg="route del $1 dev venet0 table local"
  else
-  arg="-6 route flush $1 dev venet0"
+  arg="-6 route flush $1 dev venet0 table local"
  fi
  ${IP_CMD} $arg ||
   vzwarning "vzdelrouting: ${IP_CMD} $arg failed"
```

File Attachments

1) vps-functions.diff-3.0.22, downloaded 427 times

Subject: Re: [PATCH] Routing table change in vps-functions for complex setups
Posted by kir on Wed, 12 Mar 2008 16:13:57 GMT

Hi Chris,

Sorry for the long time to reply. This is the comment from our network
expert Alexey Kuznetsov, regarding your patch.

>
> This is legal. This makes sense. I would not do this, because local
> table was not supposed to be used to hardwire some routes except for
> truly local ones.
> I am not quite sure what problem it solves. It looks like it reduces flexibility instead of increasing it.
>
> The first question: if we create one more table and one more rule with priority only a bit less than priority of local sure, sort of:
>
> SPECIAL=250
> ip rule add from any to any table $SPECIAL pref 1
>
> and add all the routes for VE addresses there. Would not it be the same?
>
> If it would, then such option can be added.

So, if using a separate table helps, would you please implement it (with
some global parameter making it optional, i.e. only then this param is set).

Christian Hofstaedtler wrote:
> Hello!
>
> I'd like to propose a change to vps-functions, to allow for more
> complex routing setups (with multiple VLANs bound on VE0, etc.).
>
> The change would modify vzaddrouting and vzdelrouting to always add
> the VE0 source routing to the "local" table. This way, all routing
> decisions regarding _local_ VEs will always be done at the very top
> in the routing stack.
> Therefore you can do other routing decisions, which would affect the
> reachability of the local VEs lower in the routing stack, without
> affecting the local VEs.
> Now this all sounds very complicated, but the patch is very simple,
> and it should not affect "normal" setups.
>
>
> I'm attaching the patch which we are currently running in production
> on 5 HNs.
>
> Everything tested with IPv4 only, though; I'm also not so sure that

> modifying the "local" table is the best choice -- OTOH the VEs are
> local to the HN.
>
> Because of the iproute table usage, the kernel needs to have
> 'Advanced Routing' set, but I'd think the OpenVZ kernels have this
> on / this is not a new requirement.
>
>
> - Christian
>
>
>
>
>
> ------- example setup & further explanations -------
>
> Example setup (done on a Debian etch host, vzctl 3.0.22,
> kernel 2.6.18-028stab053, custom config):
>
> VE0 has got multiple VLAN devices:
> eth0.110 -> 10.10.110.62/24 (this is used for management of VE0)
> eth0.150 -> 10.10.150.249/24 (used for VEs)
> eth0.152 -> 10.10.152.249/24 (used for VEs)
>
> Please note that VLAN150 + 152 are not dedicated to this HN, other
> nodes also run VEs in these VLANs.
> The VLANs are connected together by a single router, which does
> strict source IP filtering (i.e. packets from 10.10.110.0/24 are not
> allowed to come from VLAN110).
>
> Main routing table on HN looks like this:
> Destination   Gateway   Iface
> 10.10.152.0   0.0.0.0   eth0.152
> 10.10.150.0   0.0.0.0   eth0.150
> 10.10.110.0   0.0.0.0   eth0.110
> 0.0.0.0   10.10.110.1 eth0.110
>
> Routing rules on HN:
> # ip rule ls
> 0:   from all lookup 255
> 32763: from 10.10.152.0/24 lookup 152
> 32764: from 10.10.150.0/24 lookup 150
> 32765: from 10.10.110.0/24 lookup 110
> 32766: from all lookup main
> 32767: from all lookup default
>
> # ip route ls table 150
> 10.10.150.0/24 dev eth0.150 scope link

> default via 10.10.150.1 dev eth0.150
>
>
> Example VE2:
> cat /etc/vz/conf/2.conf | grep IP_
> IP_ADDRESS="10.10.150.244"
>
>
> On VE2 startup, with the original vps-functions, source routes will
> be configured in the "main" routing table. The "main" routing table
> will not be considered in this setup, because table 150 will be
> used, which already contains a (correct) default gateway. This also
> implies that Proxy ARP requests for VE2 will not be handled, because
> the kernel does not find the IP address of VE2 in its routing table.
>
>
> With the patched vps-functions, the source route will be added to
> the local table instead, and Proxy ARP requests can be handled,
> because the kernel will see the IP address of VE2. The rules for
> 10.10.150.0/24 will be ignored during Proxy ARP (lookup can be
> fulfilled already in the "local" table), but outgoing packets will
> still use the rules for 10.10.150.0/24.
>
> ------- end of example -------
>
>
> ---
>

Subject: Re: [PATCH] Routing table change in vps-functions for complex setups
Posted by Christian Hofstaedtle[1] on Thu, 13 Mar 2008 21:07:32 GMT
View Forum Message <> Reply to Message

Kir,
Alexey,

Thanks for your comments.

> >This is legal. This makes sense. I would not do this, because local
> >table was not supposed to be used to hardwire some routes except for
> >truly local ones.
> >I am not quite sure what problem it solves. It looks like it reduces
> >flexibility instead of increasing it.

It solves this problem:
If you add routing rules for multiple local subnets which are shared
with the VEs, the VEs will be unreachable, because the subnet

routing rules would come before the main table (and therefore before
the VE routes).

> >The first question: if we create one more table and one more rule with
> >priority only a bit less than priority of local sure, sort of:
> >
> >SPECIAL=250
> >ip rule add from any to any table $SPECIAL pref 1
> >and add all the routes for VE addresses there. Would not it be the same?
> >
> >If it would, then such option can be added.
>
> So, if using a separate table helps, would you please implement it (with
> some global parameter making it optional, i.e. only then this param is set).

Yes, having a seperate routing table will work, too. One needs to
create the routing rule (ip rule, as specified by Alexey), though.
This leads to the question, where/when to create this rule.

I've had this working with a seperate table already, and chose to
do the "ip route add" in the vz-initscript, but this seemed quite
fragile.

I'll incorporate this.

- Christian

--