Subject: ip tunnel in VPS: ioctl: No such device
Posted by demark on Thu, 13 Jul 2006 12:25:46 GMT
View Forum Message <> Reply to Message

hello,

i have the following problem: i need to create a tunnel within a VPS node, like this:

ip tunnel add test mode sit remote 1.2.3.4 local 4.3.2.1 dev eth0

however, i'm getting an error:

ioctl: No such device

so, is there a way to permit ip tunnels within a VPS?

thanks for any input, it's much appreciated here.

Subject: Re: ip tunnel in VPS: ioctl: No such device
Posted by aistis on Thu, 13 Jul 2006 15:19:56 GMT
View Forum Message <> Reply to Message

i believe this should help: VPN via the TUN/TAP device

Subject: Re: ip tunnel in VPS: ioctl: No such device
Posted by demark on Fri, 14 Jul 2006 10:17:29 GMT
View Forum Message <> Reply to Message

sorry, this does not help. i'm not looking for tun/tap, and i don't want to run proprietary solutions like openvpn.

i'm rather looking for ip-over-ip tunnels which are created by iproute and installed into the kernel. like i said, "ip tunnel add"

seems like this is something that's not supported or not documented?!

Subject: Re: ip tunnel in VPS: ioctl: No such device
Posted by dev on Sat, 15 Jul 2006 10:58:05 GMT
View Forum Message <> Reply to Message

openvpn is not proprietary imho. it is open
ip tunneling is not ON by default in kernel.
it is not virtualized so for safety turned off.

feel free to recompile the kernel with tunneling and give it a try.

Subject: Re: ip tunnel in VPS: ioctl: No such device
Posted by demark on Sat, 15 Jul 2006 23:35:00 GMT
View Forum Message <> Reply to Message

i'm running 2.6.16-026test014.4-smp (rpm downloaded from openvz.org). tunneling is actually compiled in - on the hardware node, i have no problems running the "ip tunnel add" command.

i looked at kernel config and there's no obvious option which controls virtualization of tunnels? or did i miss something?

my guess is that openvz doesn't support creating the tunnel device inside a VE ...

Subject: Re: ip tunnel in VPS: ioctl: No such device
Posted by dim on Mon, 17 Jul 2006 07:57:03 GMT
View Forum Message <> Reply to Message

Did you try to create tunnel device on hardware node and moving it to VE after?

Subject: Re: ip tunnel in VPS: ioctl: No such device
Posted by demark on Tue, 18 Jul 2006 12:05:53 GMT
View Forum Message <> Reply to Message

i thought of that, but there doesn't seem to be an actual interface node if you make a tunnel with "ip tunnel add" ... i already searched /dev for that

Subject: Re: ip tunnel in VPS: ioctl: No such device
Posted by aistis on Tue, 18 Jul 2006 14:42:58 GMT
View Forum Message <> Reply to Message

maybe --netdev_add will help?

Subject: Re: ip tunnel in VPS: ioctl: No such device
Posted by l4ndy74 on Tue, 17 Oct 2006 18:13:59 GMT
View Forum Message <> Reply to Message

i have the same problem it's possible add the ipv6 tunnel in the node and moving it in to the VE?!?!

Subject: Re: ip tunnel in VPS: ioctl: No such device
Posted by duswil on Sat, 08 Sep 2007 21:55:25 GMT
View Forum Message <> Reply to Message

I have the exact same problem. Did anyone ever find a solution? I really don't want to define the tunnel on the HN.

Thanks!

Subject: Re: ip tunnel in VPS: ioctl: No such device
Posted by emkravts on Thu, 13 Sep 2007 09:27:55 GMT
View Forum Message <> Reply to Message

Hello.
There are 3 types of tunnels supported by kernel:
ipip (tunl0, tunl1 etc. logical devices) - ipv4 over ipv4 tunnels
sit (sit0, sit1 etc. logical devices) - ipv6 over ipv4 tunnels
gre (gre0, gre1 etc. logical devices) - ipv4 over ipv4 tunnels

Till this moment listed devices are not virtualized in OpenVZ. The only way to setup a tunnel between VE and some node was to grant the network device (for example eth0) from HN to partucular VE and then setup the tunnel using granted device. But seems it was not the best solution. Because any VE should have an opportunity to setup tunnel. So I have virtualized ipip module that provides such an opportunity for ipv4 over ipv4 tunnels.

Attached patch does the following:

1) struct ve_ipip_tunnels containing global variables for virtualization introduced. Global variables are: pointer to per-ve tunneling net_device, storages of tunnels and per-ve lock.

2) Pointer to struct ve_ipip_tunnels added to struct_ve. Related ve_*** variables defined, functions in net/ipv4/ipip.c that use global variables updated. Corresponding start/stop functions that allocate ve_ipip_tunnels struct, per-ve net tunneling devices and initialize them introduced.

3) Hook ipip_ve_hook, initialized by start/stop functions introduced. Hook functions are to be called from do_env_create->ve_hook_iterate_init during start ve and env_cleanup->ve_hook_iterate_fini on stop ve.

4) Feature NETIF_F_VIRTUAL is set to dev->features during net_device initialization to make possible per-ve tunneling net_device creation.

5) Check for capabilities updated in ipip_tunnel_ioctl: check for CAP_VE_NET_ADMIN is added on tunnels adding and deleting. This is necessary for enabling tunneling device's ioctl within VEs.

After applying the patch to 2.6.18-028stab039 OpenVZ kernel, building and rebooting into updated kernel we can carry out some testing how ipip tunnels work. Assume we have 2 VEs (VE 895 and VE 910) running on HN and we are setting up tunnel between them:

On HN:
```
# modprobe ipip
# vzctl start 895
# vzctl start 910
# vzlist
      VEID      NPROC STATUS  IP_ADDR         HOSTNAME
       895          5 running 10.0.43.25      -
       910          5 running 10.0.98.102     -
# vzctl enter 895
```

Within 895:
```
895 # ip tunnel show
tunl0: ip/ip  remote any  local any  ttl inherit  nopmtudisc
895 # ip tunnel add tunl1 mode ipip remote 10.0.98.102 local 10.0.43.25 dev venet0
895 # ip tunnel show
tunl0: ip/ip  remote any  local any  ttl inherit  nopmtudisc
tunl1: ip/ip  remote 10.0.98.102  local 10.0.43.25  dev venet0  ttl inherit
895 # ip addr add 10.0.98.103/28 dev tunl1
895 # ip link set tunl1 up
895 # ip link set tunl1 mtu 1500
895 # ifconfig
lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 b)  TX bytes:0 (0.0 b)

tunl1     Link encap:IPIP Tunnel  HWaddr
          inet addr:10.0.98.103  P-t-P:10.0.98.103  Mask:255.255.255.240
          UP POINTOPOINT RUNNING NOARP  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 b)  TX bytes:0 (0.0 b)

venet0    Link encap:UNSPEC  HWaddr 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00
          inet addr:127.0.0.1  P-t-P:127.0.0.1  Bcast:0.0.0.0 Mask:255.255.255.255
          UP BROADCAST POINTOPOINT RUNNING NOARP  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 b)  TX bytes:0 (0.0 b)

venet0:0  Link encap:UNSPEC  HWaddr 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00
```

```
        inet addr:10.0.43.25  P-t-P:10.0.43.25  Bcast:10.0.43.25 Mask:255.255.255.255
        UP BROADCAST POINTOPOINT RUNNING NOARP  MTU:1500  Metric:1

895 # exit
# vzctl enter 910
Within 910:
910 # ip tunnel show
tunl0: ip/ip  remote any  local any  ttl inherit  nopmtudisc
910 # ip tunnel add tunl1 mode ipip remote 10.0.98.102 local 10.0.43.25 dev venet0
910 # ip tunnel show
tunl0: ip/ip  remote any  local any  ttl inherit  nopmtudisc
tunl1: ip/ip  remote 10.0.98.102  local 10.0.43.25  dev venet0  ttl inherit
910 # ip addr add 10.0.98.103/28 dev tunl1
910 # ip link set tunl1 up
910 # ip link set tunl1 mtu 1500
910 # ifconfig
lo        Link encap:Local Loopback
        inet addr:127.0.0.1  Mask:255.0.0.0
        inet6 addr: ::1/128 Scope:Host
        UP LOOPBACK RUNNING  MTU:16436  Metric:1
        RX packets:0 errors:0 dropped:0 overruns:0 frame:0
        TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:0
        RX bytes:0 (0.0 b)  TX bytes:0 (0.0 b)

tunl1     Link encap:IPIP Tunnel  HWaddr
        inet addr:10.0.98.103  P-t-P:10.0.98.103  Mask:255.255.255.240
        UP POINTOPOINT RUNNING NOARP  MTU:1500  Metric:1
        RX packets:0 errors:0 dropped:0 overruns:0 frame:0
        TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:0
        RX bytes:0 (0.0 b)  TX bytes:0 (0.0 b)

venet0    Link encap:UNSPEC  HWaddr 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00
        inet addr:127.0.0.1  P-t-P:127.0.0.1  Bcast:0.0.0.0 Mask:255.255.255.255
        UP BROADCAST POINTOPOINT RUNNING NOARP  MTU:1500  Metric:1
        RX packets:0 errors:0 dropped:0 overruns:0 frame:0
        TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:0
        RX bytes:0 (0.0 b)  TX bytes:0 (0.0 b)

venet0:0  Link encap:UNSPEC  HWaddr 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00
        inet addr:10.0.43.25  P-t-P:10.0.43.25  Bcast:10.0.43.25 Mask:255.255.255.255
        UP BROADCAST POINTOPOINT RUNNING NOARP  MTU:1500  Metric:1

Ping 895 through tunl1:
910 # ping 10.0.98.103
PING 10.0.98.103 (10.0.98.103) 56(84) bytes of data.
```

64 bytes from 10.0.98.103: icmp_seq=1 ttl=64 time=0.043 ms
64 bytes from 10.0.98.103: icmp_seq=2 ttl=64 time=0.036 ms
64 bytes from 10.0.98.103: icmp_seq=3 ttl=64 time=0.035 ms
64 bytes from 10.0.98.103: icmp_seq=4 ttl=64 time=0.034 ms

--- 10.0.98.103 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3001ms
rtt min/avg/max/mdev = 0.034/0.037/0.043/0.003 ms
910 #

Works. Patch can be succesfully applied also to 2.6.18-rhel5-042 kernel. The same test passes.

Could you please apply the patch and carry out some more testing for ipip tunnels. Thanks.

I suppose the next step is virtualization sit.

File Attachments

1) diff-ipip-tunnel-virtualization-20070913, downloaded 1050 times

---

Subject: Re: ip tunnel in VPS: ioctl: No such device
Posted by duswil on Thu, 13 Sep 2007 12:55:50 GMT
View Forum Message <> Reply to Message

I'm looking to make a sit tunnel to provide an IPv6 tunnel via IPv4. It sounds like this patch doesn't provide that. Please confirm my interpretation. Thanks!

Subject: Re: ip tunnel in VPS: ioctl: No such device
Posted by emkravts on Thu, 13 Sep 2007 12:59:45 GMT
View Forum Message <> Reply to Message

Yes, this is ipip. Virtualization of sit what I am working on this very moment.

Subject: Re: ip tunnel in VPS: ioctl: No such device
Posted by duswil on Thu, 13 Sep 2007 13:04:43 GMT
View Forum Message <> Reply to Message

Thank you. Having the ability to have a sit tunnel in a VPS will be extremely helpful.

On each of my servers, my VPSes have their own bridge and network apart from the HN's network and are connected to other VPS bridges by way of OpenVPN. This is all managed by a VPS on each server called "Gateway". It also manages my incoming IPv4 filtering. I am wanting "Gateway" to provide the IPv6 tunnels (and filtering) as well. Your work will allow this to happen.

Thanks!

Subject: Re: ip tunnel in VPS: ioctl: No such device
Posted by emkravts on Fri, 14 Sep 2007 10:55:21 GMT
View Forum Message <> Reply to Message

Hello. Attached patch diff-sit-tunnel-virtualization-20070913 virtualizes sit device. Now it is possible to create per-ve sets of sit tunnels. Created tunnels can be tested in the same way as described above for ipip. The difference is valid ipv6 address should be assigned to sit devices. And ping6 should be used instead of ping.

File Attachments
1) diff-sit-tunnel-virtualization-20070913, downloaded 902 times

Subject: Re: ip tunnel in VPS: ioctl: No such device//found a solution!
Posted by LucienLu on Mon, 22 Feb 2010 13:59:57 GMT
View Forum Message <> Reply to Message

Hi guys,

I've written a tiny porgram to enable TunnelBroker(6to4) on OpenVZ kernel which compiled without ipv6-over-ip tunnel. It uses the TUN/TAP device.

Detail is at

http://code.google.com/p/tb-tun/

I'm now using that tunnel. See the performance at

http://www.lostriver.net/linux-userspace-6to4-tun/