

---

Subject: Veth mac generation

Posted by [dietmar](#) on Wed, 11 Jun 2008 10:46:31 GMT

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

---

Hi all,

The code to generate mac addresses for veth (generate\_mac in veth.c) uses the Constant SW\_OUI for upper 3 bytes, and random values for lower 3 bytes. Thus giving 2<sup>24</sup> possible values.

Isn't it better to use random numbers for all 6 bytes, like the code in the linux kernel:

```
static inline void random_ether_addr(u8 *addr)
{
    get_random_bytes (addr, ETH_ALEN);
    addr [0] &= 0xfe;    /* clear multicast bit */
    addr [0] |= 0x02;    /* set local assignment bit (IEEE802) */
}
```

That would make conflict less likely.

- Dietmar

---

---

Subject: Re: Veth mac generation

Posted by [Kirill Korotaev](#) on Wed, 11 Jun 2008 13:46:49 GMT

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

---

and yes and no.

These upper 3 bytes are reserved for our company, so selecting them you will never conflict with other devices in network infrastructure. i.e. the worst what can happen 2 veths will conflict.

On the other hand - you are right, 6 bytes are better :)

Kirill

Dietmar Maurer wrote:

> Hi all,

>

> The code to generate mac addresses for veth (generate\_mac in veth.c)

> uses the

> Constant SW\_OUI for upper 3 bytes, and random values for lower 3 bytes.

> Thus

> giving 2<sup>24</sup> possible values.

>  
> Isn't it better to use random numbers for all 6 bytes, like the code  
> in the linux kernel:  
>  
> static inline void random\_ether\_addr(u8 \*addr)  
> {  
> get\_random\_bytes (addr, ETH\_ALEN);  
> addr [0] &= 0xfe; /\* clear multicast bit \*/  
> addr [0] |= 0x02; /\* set local assignment bit (IEEE802) \*/  
> }  
>  
> That would make conflict less likely.  
>  
> - Dietmar  
>  
>  
>

---

Subject: AW: Veth mac generation  
Posted by [dietmar](#) on Thu, 12 Jun 2008 05:40:03 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Why I asked is because of that bridge problem:

[http://forum.openvz.org/index.php?t=msg&th=5291&#msg\\_26576](http://forum.openvz.org/index.php?t=msg&th=5291&#msg_26576)

A bridge always select the lowest mac address.

This patch solves the problem, but i am not sure if there are side effects.

<https://lists.linux-foundation.org/pipermail/bridge/2008-June/005895.html>

The SWSOFT OID is quite 'low', so the problem occurs frequently.

- Dietmar

> -----Urspr  
> Von: [users-bounces@openvz.org](mailto:users-bounces@openvz.org)  
> [<mailto:users-bounces@openvz.org>] Im Auftrag von Kirill Korotaev  
> Gesendet: Mittwoch, 11. Juni 2008 15:47  
> An: [users@openvz.org](mailto:users@openvz.org)  
> Betreff: Re: [Users] Veth mac generation  
>  
> and yes and no.  
> These upper 3 bytes are reserved for our company, so  
> selecting them you will never conflict with other devices in

> network infrastructure.  
> i.e. the worst what can happen 2 veths will conflict.  
>  
> On the other hand - you are right, 6 bytes are better :)

---

---

Subject: Re: AW: Veth mac generation  
Posted by [Kirill Korotaev](#) on Fri, 13 Jun 2008 00:21:19 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Do I understand correctly that you actually experience the following problem:  
1. veth MAC address is lower then your ethX MAC.  
2. so brX is assigned min(vethX-MAC, ethX-MAC) which is vethX-MAC.  
3. and what is the your problem with that? that host system MAC changes dynamically and networking breaks or what?

I just can't see how fully random 6 bytes MAC can help. Because sometimes it will be "low" enough as well and you will hit the problem anyway.

If I got your problem right then I can advise you a possible solution - in RHEL5 kernel we have a functionality called "via\_phys\_dev" (triggered by BRCTL\_SET\_VIA\_ORIG\_DEV ioctl). This forces kernel to work with original interface ethX (first added to bridge) and pass the traffic to it. This allows to add ethX to bridge w/o need to propogate it's netfilter rules and other settings to brX.

Thanks,  
Kirill

Dietmar Maurer wrote:

> Why I asked is because of that bridge problem:  
>  
> [http://forum.openvz.org/index.php?t=msg&th=5291&#msg\\_26576](http://forum.openvz.org/index.php?t=msg&th=5291&#msg_26576)  
>  
> A bridge always select the lowest mac address.  
>  
> This patch solves the problem, but i am not sure if there are side effects.  
>  
> <https://lists.linux-foundation.org/pipermail/bridge/2008-June/005895.html>  
>  
> The SWSOFT OID is quite 'low', so the problem occurs frequently.  
>  
> - Dietmar  
>  
>

>> -----Urspr  
>> Von: users-bounces@openvz.org  
>> [mailto:users-bounces@openvz.org] Im Auftrag von Kirill Korotaev  
>> Gesendet: Mittwoch, 11. Juni 2008 15:47  
>> An: users@openvz.org  
>> Betreff: Re: [Users] Veth mac generation  
>>  
>> and yes and no.  
>> These upper 3 bytes are reserved for our company, so  
>> selecting them you will never conflict with other devices in  
>> network infrastructure.  
>> i.e. the worst what can happen 2 veths will conflict.  
>>  
>> On the other hand - you are right, 6 bytes are better :)  
>  
>

---

Subject: AW: AW: Veth mac generation  
Posted by [dietmar](#) on Fri, 13 Jun 2008 06:52:34 GMT  
[View Forum Message](#) <> [Reply to Message](#)

> Do I understand correctly that you actually experience the  
> following problem:  
> 1. veth MAC address is lower then your ethX MAC.  
> 2. so brX is assigned min(vethX-MAC, ethX-MAC) which is vethX-MAC.  
> 3. and what is the your problem with that? that host system  
> MAC changes dynamically and networking breaks or what?

The host gets unreachable for about 20 seconds (because ARP mappings are wrong now)

> I just can't see how fully random 6 bytes MAC can help.  
> Because sometimes it will be "low" enough as well and you  
> will hit the problem anyway.

Well one idea was to assign macs starting with 'fe' - but ist not a good idea anyways.

> If I got your problem right then I can advise you a possible  
> solution - in RHEL5 kernel we have a functionality called  
> "via\_phys\_dev" (triggered by BRCTL\_SET\_VIA\_ORIG\_DEV ioctl).  
> This forces kernel to work with original interface ethX  
> (first added to bridge) and pass the traffic to it. This  
> allows to add ethX to bridge w/o need to propogate it's  
> netfilter rules and other settings to brX.

Interesting - thanks for that hint.

- Dietmar

---

---

Subject: AW: AW: Veth mac generation  
Posted by [dietmar](#) on Fri, 13 Jun 2008 09:43:33 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

> If I got your problem right then I can advise you a possible  
> solution - in RHEL5 kernel we have a functionality called  
> "via\_phys\_dev" (triggered by BRCTL\_SET\_VIA\_ORIG\_DEV ioctl).  
> This forces kernel to work with original interface ethX  
> (first added to bridge) and pass the traffic to it. This  
> allows to add ethX to bridge w/o need to propagate it's  
> netfilter rules and other settings to brX.

Looking at the openvz-2.6.24 sources that via\_phys\_dev is  
already there. I just wonder how to use it:

```
brctl addbr vmbr0  
echo 1 > /sys/class/net/vmbr0/bridge/via_phys_dev  
brctl addif eth0
```

But looks like ip setting of eth0 gets lost (or what is  
meant by 'and other settings?'). Maybe you have further info  
How to use that feature?

- Dietmar

---

---

Subject: Re: AW: AW: Veth mac generation  
Posted by [Kirill Korotaev](#) on Fri, 13 Jun 2008 17:03:38 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Dietmar Maurer wrote:

>  
>> If I got your problem right then I can advise you a possible  
>> solution - in RHEL5 kernel we have a functionality called  
>> "via\_phys\_dev" (triggered by BRCTL\_SET\_VIA\_ORIG\_DEV ioctl).  
>> This forces kernel to work with original interface ethX  
>> (first added to bridge) and pass the traffic to it. This  
>> allows to add ethX to bridge w/o need to propagate it's  
>> netfilter rules and other settings to brX.  
>  
> Looking at the openvz-2.6.24 sources that via\_phys\_dev is  
> already there. I just wonder how to use it:  
>

```
> brctl addbr vobr0
> echo 1 > /sys/class/net/vobr0/bridge/via_phys_dev
> brctl addif eth0
>
> But looks like ip setting of eth0 gets lost (or what is
> meant by 'and other settings?'). Maybe you have further info
> How to use that feature?
```

Sorry, I can't check right now. So can only say what I remember.  
The idea was really to make traffic going through original ethX device  
so you don't need to reconfigure anything in the host after bridge creation.

in general command look right. I guess you also need to make vobr0 interface up before adding  
eth0.  
plz check.

Thanks,  
Kirill

---