Subject: restrict netfilter to HN only

Posted by gm77 on Sat, 21 Oct 2006 20:50:40 GMT

View Forum Message <> Reply to Message

Hello,

I want to configure the following system:

1. kernel should be a monolithic one (no modules).

2. hardware node should have the powers of netfilter (i.e. all iptables rules are configured only on the hardware node).

3. VPS shouldn't be able to mess with any netfilter functionality (i.e. it should look like there are no support in the kernel).

I started with step 3 and undefined CONFIG_VE_IPTABLES (AFAIK, this is the support for iptables in virtual environments and I don't need any). Then I found that if I didn't enable modules iptables cannot be properly initialised at startup, so any 'iptables ...' command produce an error message that there are no netfilter modules available. The fix was trivial - there is a bug in net/ipv4/netfilter/ip_tables.c: when CONFIG_VE_IPTABLES isn't defined the following code in the init_iptables() function returns -EEXIST immediately (it's me who surrounded the block with #ifdef):

```
#ifdef CONFIG_VE_IPTABLES
	if (ve_ipt_standard_target != NULL)
		return -EEXIST;
#endif
```

Once this block of code is surrounded with #ifdef I'm able to use iptables on the hardware node (at least a can filter packets), my VPS users are unable to use iptables (and this is what I want),

10 minutes later:

I was wrong with the VPS users. :(
Disabling CONFIG_VE_IPTABLES gives them access to HN's tables 8o.
It's horrible! I was expecting that disabling the netfilter
virtualisation just disables iptables in VPSes, but it actually
just _disable_ the virtualisation :-/.

Oh, God, there a lot of work to be done tonight :-D

but I encountered another problem. Now, when I define a VPS with the rfc1918 IP (e.g. 192.168.0.1) and try to masquerade it on the hardware node using

```
iptables -t nat -A POSTROUTING -o eth0 -s 192.168.0.1 -j MASQUERADE
```

nothing happens.

Well, I just replaced this rule with the following:

```
iptables -t nat -A POSTROUTING -j LOG
```

and tried to reach some host in the wild, so I've got the following in my dmesg output:

```
IN= OUT=venet0 SRC=192.168.0.1 DST=3.3.3.3 LEN=84 TOS=0x00 PREC=0x00 TTL=64 ID=0
DF PROTO=ICMP TYPE=8 CODE=0 ID=27669 SEQ=1
```

I was quite surprised that the output interface is venet0 and the input interface is empty (I expected IN=venet0 OUT=eth0). If I enable CONFIG_VE_IPTABLES then POSTROUTING will work as expected, but my users will get access to the virtualised iptables and I don't want this.

So the question is have this configuration ever been tested? I guess not . But it's a quite comfortable setup so I think it's good to test monolithic kernels as well (I found a lot of issues with networking code if modules are disabled -- fortunately, I'm not using much of network virtualisation)).

P.S. I may be wrong with all this stuff, but I wanted to point out that there could be some issues and, perhaps, several bugs in the networking code. The only thing I need to solve now is to repair routing from VPS to the wild when CONFIG_VE_IPTABLES are disabled. Any help is much appreciated!

Thanks!