Subject: insmod problem
Posted by danielcamara on Fri, 25 Apr 2008 08:42:33 GMT
View Forum Message <> Reply to Message

Hi people,

I am having a problem inserting a module in a Virtual Machine here...

I need to install the module in a series of VMs. It is required for each machine to have its own instance of the module installed. However, I am having an "Operation not permitted" problem, though the capability SYS_MODULE is on.

The error I am getting when I try to insert a module, when the module is not in the real machine, is:

```
node_104 # insmod nasmesh.ko nas_is_clusterhead=0
insmod: error inserting 'nasmesh.ko': -1 Operation not permitted
```

The output of the lsmod shows no module inserted, as expected, given the permission error. However, another strange thing is that if I install the module in the real machine, the error inside the VM changes:

```
real_machine # insmod nasmesh.ko nas_is_clusterhead=0
real_machine # vzctl enter 104
VM_node_104 # insmod nasmesh.ko nas_is_clusterhead=0
insmod: error inserting 'nasmesh.ko': -1 File exists
```

Even the output at the lsmod being empty.

I found this discussion from the news (http://forum.openvz.org/index.php?t=msg&goto=25421&), that seems to be more or less the same problem... but I sincerely didn't understand the answer, nor if it is applicable for this case.

Any Ideas, for both problems (operation not permitted, and the File exists one)?!?!? i.e. if they are really two separated problems!

Best regards...

Daniel

Configuration :

OS Host machine: Ubuntu

Kernel Host machine: 2.6.18-028stab053
OpenVZ : kernel-patch-openvz-028.18.3
VM Template: debian-4.0-i386-minimal

lcap on the VM:
---
lcap
Current capabilities: 0x5DCDEEFF
0) *CAP_CHOWN 1) *CAP_DAC_OVERRIDE
2) *CAP_DAC_READ_SEARCH 3) *CAP_FOWNER
4) *CAP_FSETID 5) *CAP_KILL
6) *CAP_SETGID 7) *CAP_SETUID
8) CAP_SETPCAP 9) *CAP_LINUX_IMMUTABLE
10) *CAP_NET_BIND_SERVICE 11) *CAP_NET_BROADCAST
12) CAP_NET_ADMIN 13) *CAP_NET_RAW
14) *CAP_IPC_LOCK 15) *CAP_IPC_OWNER
16) *CAP_SYS_MODULE 17) CAP_SYS_RAWIO
18) *CAP_SYS_CHROOT 19) *CAP_SYS_PTRACE
20) CAP_SYS_PACCT 21) CAP_SYS_ADMIN
22) *CAP_SYS_BOOT 23) *CAP_SYS_NICE
24) *CAP_SYS_RESOURCE 25) CAP_SYS_TIME
26) *CAP_SYS_TTY_CONFIG 27) *CAP_MKNOD
28) *CAP_LEASE 29) CAP_AUDIT_WRITE
30) *CAP_AUDIT_CONTROL
* = Capabilities currently allowed

lsmod VM machine:
---
lsmod
Module Size Used by

lsmod real machine:
---
lsmod
Module Size Used by
nasmesh 34572 0
(.... Other modules)

Subject: Re: insmod problem
Posted by maratrus on Fri, 25 Apr 2008 10:25:37 GMT
View Forum Message <> Reply to Message

Hello,

Loading modules inside VE is prohibited for example because of the secure reasons. Otherwise

you can load bad module inside VE and it may crash the whole HN.

Thank You!

Subject: Re: insmod problem
Posted by adobriyan on Fri, 25 Apr 2008 10:27:37 GMT
View Forum Message <> Reply to Message

There must be some confusion about what kernel modules are,
what they do, what OpenVZ does and what OpenVZ does not.

OpenVZ implements so-called container-style virtualization.
This means only 1 (one) kernel image is in memory and controls
the system and all, let's call them, virtual machines.
The situation is no different from normal unpatches Linux
kernel.

Modules allow administrator to load additional functionality
at live kernel. This can be device driver, allowing operating
system to interact with hardware, or file system driver,
allowing to mount new filesystems and so on.

Module's code is loaded and linked against already existing
kernel image in memory and possibly against other modules.
It's loaded with the very same privileges as the rest of the
kernel which are THE privileges.

Kernel module can access all kernel memory, the rest of the
kernel can access. Kernel module can change many data
structures without problems and all data structures with
minimal workarounds. I'm talking about exported symbols here.

Corollary #1: kernel module can royally screw up the system in
particularly disgusting wayes. It can seamlessly give root
priviledges to any process. Which would be the least of your
troubles, BTW.

Corollary #2: loading of kernel modules from inside
VE/container is not allowed. Root of VE is viewed as untrusted
here, as such, giving him such powerful weapon as
init_module(2) is simply out of question.
If you want to load something, let administrator of VE0 do it.

Corollary #3: if you want containerise some functionality your
module gives to you, you most certainly should change your
module. Depends on functionality it provides.

Other types of virtualization such as KVM allow you to have multiple operating system kernels, as such, you can load different modules into different kernels. In theory and modulo bugs in virtualization layer and in hardware, this can't screw up master kernel. OpenVZ doesn't operate like this by design nor it was ever promised for OpenVZ to operate like this.

So, what your module actually does? In some case problem is fixed by allowing VE to use major:minor pairs and creating device nodes in.

Subject: Re: insmod problem
Posted by danielcamara on Fri, 25 Apr 2008 12:35:42 GMT
View Forum Message <> Reply to Message

Ok, you are right... I didn't pay attention to that sorry, my bad . OK I understand the risks, even though in my specific case all the virtual machines would be trustful, once I would be in the control all of them and no external user would have access to the machines... any way... It is a very peculiar and particular case I guess

So the bottom line is: it is not possible. I should either use a different tool (kde, for example) or redesign the module in a way I could use the module "shared" by all the machines in the VE0. Is that correct?

Just one more stupid question, I am not trying to be a smart ass or any thing alike here, I swear, I am just trying to understand. I tried the insmod because I sow this entry in the Virtuoso manual : " sys_module : Insert and remove kernel modules. Be very careful with setting this capability on for a Virtual Private Server; if a user has the permission of inserting kernel modules, this user has essentially full control over the Hardware Node."

So, this entry is not talking about the insert modules then?!?! Or it is not applicable to this case?

Best regards...

Daniel
