

>> hmm, I'm wondering how this is going to work for a process which  
>> would have unshared its device (pts) namespace. How are we going  
>> to link the pts living in different namespaces if the stdios of the  
>> hijacked process is using them ? like in the case of a shell, which  
>> is certainly something we would like to hijacked.

>>

>> it looks like a challenge for me. maybe I'm wrong.

>

> Might be a problem, but tough to address that until we actually  
> have a dev ns or devpts ns and established semantics.

>

> Note the filestruct comes from current, not the hijack target, so  
> presumably we can work around the tty issue in any case by  
> keeping an open file across the hijack?

>

> For instance, use the attached modified version of hijack.c  
> which puts a writeable fd for /tmp/helloworld in fd 5, then  
> does hijack, then from the resulting shell do

>

> echo ab >&5

>

> So we should easily be able to work around it.

yes. it should.

> Or am i missing something?

I guess we need to work a little more on the pts/device namespace  
to see how it interacts.

>>> The effect is a sort of namespace enter. The following program  
>>> uses sys\_hijack to 'enter' all namespaces of the specified pid.

>>> For instance in one terminal, do

>>>

>>> mount -t cgroup -ons /cgroup

>>> hostname

>>> qemu

>>> ns\_exec -u /bin/sh

>>> hostname serge

>>> echo \$\$

>>> 1073

>>> cat /proc/\$\$/cgroup

>>> ns:/node\_1073

>> Is there a reason to have the 'node\_' prefix ? couldn't we just

>> use \$pid ?

>

> Good question. It's just how the ns-cgroup does it... If you want to

> send in a patch to change that, I'll ack it.

just below.

I gave a quick look to the ns subsystem and didn't see how the node\_\$pid was destroyed. do we have to do a rmdir ?

Thanks,

C.

Signed-off-by: Cedric Le Goater <clg@fr.ibm.com>

---

kernel/cgroup.c | 2 +-  
1 file changed, 1 insertion(+), 1 deletion(-)

Index: 2.6.23-mm1/kernel/cgroup.c

=====

--- 2.6.23-mm1.orig/kernel/cgroup.c

+++ 2.6.23-mm1/kernel/cgroup.c

@ @ -2604,7 +2604,7 @ @ int cgroup\_clone(struct task\_struct \*tsk  
cg = tsk->cgroups;  
parent = task\_cgroup(tsk, subsys->subsys\_id);

- snprintf(nodename, MAX\_CGROUP\_TYPE\_NAMELEN, "node\_%d", tsk->pid);

+ snprintf(nodename, MAX\_CGROUP\_TYPE\_NAMELEN, "%d", tsk->pid);

/\* Pin the hierarchy \*/

atomic\_inc(&parent->root->sb->s\_active);

---

Containers mailing list

Containers@lists.linux-foundation.org

<https://lists.linux-foundation.org/mailman/listinfo/containers>

---