Subject: Re: [PATCH] namespaces: introduce sys_hijack (v4)
Posted by serue on Wed, 10 Oct 2007 18:32:34 GMT
View Forum Message <> Reply to Message

Quoting Cedric Le Goater (clg@fr.ibm.com):
> Serge E. Hallyn wrote:
> >>From 945fe66259cd0cfdc2fe846287b7821e329a558c Mon Sep 17 00:00:00 2001
> > From: sergeh@us.ibm.com <hallyn@kernel.(none)>
> > Date: Tue, 9 Oct 2007 08:30:30 -0700
> > Subject: [PATCH] namespaces: introduce sys_hijack (v4)
> >
> > Move most of do_fork() into a new do_fork_task() which acts on
> > a new argument, task, rather than on current. do_fork() becomes
> > a call to do_fork_task(current, ...).
> >
> > Introduce sys_hijack (for x86 only so far). It is like clone, but
> > in place of a stack pointer (which is assumed null) it accepts a
> > pid. The process identified by that pid is the one which is
> > actually cloned. Some state - include the file table, the signals
> > and sighand (and hence tty), and the ->parent are taken from the
> > calling process.
>
> 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.

Or am i missing something?

> > 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.

> > In another terminal then do
> >
> > hostname
> >   qemu
> > cat /proc/$$/cgroup
> >   ns:/
> > hijack 1073
> >   hostname
> >   serge
> >   cat /proc/$$/cgroup
> >   ns:/node_1073
> >
> > sys_hijack is arch-dependent and is only implemented for i386 so far.
>
> and worked on my qemu.
>
> Thanks !

Cool. Thanks for testing.

-serge

File Attachments

1) duphijack.c, downloaded 281 times