Subject: Re: [PATCH 03/08] allow sethostname in a container
Posted by serge on Fri, 04 Feb 2011 15:56:15 GMT
View Forum Message <> Reply to Message

Quoting Serge E. Hallyn (serge@hallyn.com):
> Quoting Serge E. Hallyn (serge@hallyn.com):
> > Signed-off-by: Serge E. Hallyn <serge.hallyn@canonical.com>
> > ---
> > kernel/sys.c | 2 +-
> > 1 files changed, 1 insertions(+), 1 deletions(-)
> >
> > diff --git a/kernel/sys.c b/kernel/sys.c
> > index 2745dcd..9b9b03b 100644
> > --- a/kernel/sys.c
> > +++ b/kernel/sys.c
> > @@ -1171,7 +1171,7 @@ SYSCALL_DEFINE2(sethostname, char __user *, name, int, len)
> >   int errno;
> >   char tmp[__NEW_UTS_LEN];
> >
> > - if (!capable(CAP_SYS_ADMIN))
> > + if (!ns_capable(current->nsproxy->uts_ns->user_ns, CAP_SYS_ADMIN))
> >   return -EPERM;
> >   if (len < 0 || len > __NEW_UTS_LEN)
> >   return -EINVAL;
> > --
> > 1.7.0.4
>
> An interesting note here is that since the task doing ns_exec (and
> therefore in the init_user_ns) requires CAP_SYS_ADMIN to unshare,
> this check will actually always be true if uts_ns was not unshared.

Noone ever called me on this, so for the sake of posterity reading the
m-l archives: what I said above is not true. If uts_ns was not
unshared, then current->nsproxy->uts_ns->user_ns != current_user_ns(),
so current should not have ns_capable(current->nsproxy->uts_ns->user_ns,
CAP_SYS_ADMIN). So the check will always return false.

> If uts is unshared, then regular capabilities semantics in the
> child user_ns apply (that is, root can do sethostname, unpriv user
> cannot) The intent is that user namespaces will eventually allow
> unprivileged users to unshare, after which this will make much more
> sense.
>
> -serge

_____
Containers mailing list
Containers@lists.linux-foundation.org
https://lists.linux-foundation.org/mailman/listinfo/containe rs