Subject: Re: [PATCH -mm 1/2] user namespace : add unshare
Posted by akpm on Fri, 08 Jun 2007 19:22:15 GMT
View Forum Message <> Reply to Message

On Fri, 08 Jun 2007 17:14:07 +0200
Cedric Le Goater <clg@fr.ibm.com> wrote:

> Basically, it will allow a process to unshare its user_struct table, resetting
> at the same time its own user_struct and all the associated accounting.
>
> A new root user (uid == 0) is added to the user namespace upon creation. Such
> root users have full privileges and it seems that theses privileges should be
> controlled through some means (process capabilities ?)

This second paragraph is distressingly indecisive. How much thought has
gone into this??

For a start, it seems wrong for the kernel to hardwire knowledge about UID
0 in this fashion.

I'd have thought that a better model for user-namespace unsharing would be
to do a copy-by-value of the entire namespace, then permit a
suitably-privileged application to go through and kill off any unwanted
users from the now-unshared user namespace.

Or maybe just remove that "Insert new root user" altogether? What would
then go wrong?

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

Subject: Re: [PATCH -mm 1/2] user namespace : add unshare
Posted by serue on Mon, 11 Jun 2007 15:33:25 GMT
View Forum Message <> Reply to Message

Quoting Andrew Morton (akpm@linux-foundation.org):
> On Fri, 08 Jun 2007 17:14:07 +0200
> Cedric Le Goater <clg@fr.ibm.com> wrote:
>
> > Basically, it will allow a process to unshare its user_struct table, resetting
> > at the same time its own user_struct and all the associated accounting.
> >
> > A new root user (uid == 0) is added to the user namespace upon creation. Such
> > root users have full privileges and it seems that theses privileges should be
> > controlled through some means (process capabilities ?)

>
> This second paragraph is distressingly indecisive. How much thought has
> gone into this??

Quite a lot of thought, very little in the way of decisions.

The idea with going with just these two patches for now is that the uid
0 in the child namespace can be contained using selinux anyway. The
program which does the clone(CLONE_NEWUSER) is of type newuserns_exec_t,
causing a domain transition to newuserns_user_t, which is denied rights
to all but the files in the filesystem into which it has been chrooted,
let's say newuserns_chroot_t, and to all process not of type
newuserns_user_t.

> For a start, it seems wrong for the kernel to hardwire knowledge about UID
> 0 in this fashion.

Here's what I'd like to see happen eventually:

Some user, let's say apache, or it could be root, spawns a new user
namespace. The current process retains the original user's rights in
the original namespace, and becomes uid 0 in the new namespace, with
full capabilities to the new namespace. It can then set up the new user
namespace as it likes.

That is where we would be heading with the roadmap I laid out in my
intro msg to the longer userns patchset.

Unfortunately that's a long way and a lot of intrusive code away from
reality atm.

> I'd have thought that a better model for user-namespace unsharing would be
> to do a copy-by-value of the entire namespace, then permit a
> suitably-privileged application to go through and kill off any unwanted
> users from the now-unshared user namespace.

Hmm, I generally go under the impression that user namespaces would be
primarily useful for vservers, not checkpoint/restart jobs, and so any
vserver would have completely different set of users from the host
system.

> Or maybe just remove that "Insert new root user" altogether? What would
> then go wrong?

It would prevent the new root user from administrating users in the
virtual server, I guess.

Again the intent is that the 'new root user' would be root in the new

user namespace, but only have the capabilities of the user who did the
user namespace unshare in the original user namespace. So once that is
possible, we would definately want to start with a root user.

thanks,
-serge

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