Subject: Re: [RFC][PATCH 0/2] user namespace [try #2]
Posted by serue on Thu, 31 Aug 2006 15:59:22 GMT
View Forum Message <> Reply to Message

Quoting Eric W. Biederman (ebiederm@xmission.com):
> "Serge E. Hallyn" <serue@us.ibm.com> writes:
>
> > Quoting Cedric Le Goater (clg@fr.ibm.com):
> >> Cedric Le Goater wrote:
> >> > Hi all,
> >> >
> >> > Here's a second version. It's very close from the first one and takes into
> >> > account some discussions we had with kirill on the topic during OLS. 2
> >> > patches follow, the first introduces the user namespace core and the last
> >> > enables to use it with unshare.
> >> >
> >> > Changes [try #2]
> >> >
> >> > - removed struct user_namespace* argument from find_user()
> >> > - added a root_user per user namespace
> >> >
> >> > execns() syscall is back in the attic for the moment. I'm still maintaining
> >> > it and we'll see if it's of any use when we address the user space API of
> >> > the full conainer. soon, I hope !
> >> >
> >> > This user namespace patchset does not try to address all the issues that
> >> > were raised by the previous thread on the topic, like user mapping per
> >> > namespace, per mount, etc. It tries to solve some simple issues with the
> >> > current implementation of containers in mind. It should be especially
> >> > useful the existing solutions and lay ground basic objects.
> >> >
> >> > thanks for your comments,
> >>
> >> I didn't get much comments on that one. is everybody happy with it ? can we
> >> merge ask andrew to merge in -mm ?
> >>
> >> thanks,
> >
> > Ideally we could collect Acked-by: or Signed-off-by: from Eric, Kir or
> > Kirill, and Herbert or Sam, to show we are all in agreement.
> >
> > Or a NACK :)
>
> Ok for the collection
> Nacked-by: Eric Biederman

Thanks :)

> My gut feel is that this is terribly incomplete, and doesn't
> come with enough description to tell me why it could possibly be complete.
>
> I don't think this addresses any of my primary objections from last
> round.
>
> This doesn't change the kernel to make uid comparisons as (uid_ns, uid)
> tuples or explain why that isn't necessary. It doesn't touch keys.
> it doesn't explain why we are not introducing possibly subtle security problems.
>
> Cedric sorry for not saying so earlier, but I thought that the incompleteness
> was obvious.

(ignoring keys and other uid actions for now)

Here's a stab at semantics for how to handle file access. Should be
pretty simple to implement, but i won't get a chance to implement this
week.

At mount, by default the vfsmount is tagged with a uid_ns.
A new -o uid_ns=<pid> option instead tags the vfsmount with the uid_ns
 belonging to pid <pid>. Since any process in a descendent pid
 namespace should still have a valid pid in the ancestor
 pidspaces, this should work fine.
At vfs_permission, if current->nsproxy->uid_ns != file->f_vfsmnt->uid_ns,
 1. If file is owned by root, then read permission is granted
 2. If file is owned by non-root, no permission is granted
(regardless of process uid)

Does this sound reasonable?

I assume the list of other things we'll need to consider includes
 signals between user namespaces
 keystore
 sys_setpriority and the like
I might argue that all of these should be sufficiently protected
by proper setup by userspace. Can you explain why that is not
the case?

thanks,
-serge

_____
Containers mailing list
Containers@lists.osdl.org
https://lists.osdl.org/mailman/listinfo/containers