Subject: Re: [patch] unprivileged mounts update
Posted by serge on Wed, 25 Apr 2007 17:56:09 GMT
View Forum Message <> Reply to Message

Quoting Eric W. Biederman (ebiederm@xmission.com):
> "Serge E. Hallyn" <serue@us.ibm.com> writes:
>
> > Quoting H. Peter Anvin (hpa@zytor.com):
> >> Miklos Szeredi wrote:
> >> >
> >> > Andrew, please skip this patch, for now.
> >> >
> >> > Serge found a problem with the fsuid approach: setfsuid(nonzero) will
> >> > remove filesystem related capabilities. So even if root is trying to
> >> > set the "user=UID" flag on a mount, access to the target (and in case
> >> > of bind, the source) is checked with user privileges.
> >> >
> >> > Root should be able to set this flag on any mountpoint, _regardless_
> >> > of permissions.
> >> >
> >>
> >> Right, if you're using fsuid != 0, you're not running as root
> >
> > Sure, but what I'm not clear on is why, if I've done a
> > prctl(PR_SET_KEEPCAPS, 1) before the setfsuid, I still lose the
> > CAP_FS_MASK perms. I see the special case handling in
> > cap_task_post_setuid(). I'm sure there was a reason for it, but
> > this is a piece of the capability implementation I don't understand
> > right now.
>
> So we drop CAP_CHOWN, CAP_DAC_OVERRIDE, CAP_DAC_READ_SEARCH,
> CAP_FOWNER, and CAP_FSETID
>
> Since we are checking CAP_SETUID or CAP_SYS_ADMIN how is that
> a problem?
>
> Are there other permission checks that mount is doing that we
> care about.

Not mount itself, but in looking up /share/fa/root/home/fa,
user fa doesn't have the rights to read /share, and by setting
fsuid to fa and dropping CAP_DAC_READ_SEARCH the mount action fails.

But the solution you outlined in your previous post would work around
this perfectly.

> >> (fsuid is
> >> the equivalent to euid for the filesystem.)

> >
> > If it were really the equivalent then I could keep my capabilities :)
> > after changing it.
>
> We drop all capabilities after we change the euid.

Not if we've done prctl(PR_SET_KEEPCAPS, 1)

> >> I fail to see how ruid should have *any* impact on mount(2). That seems
> >> to be a design flaw.
> >
> > May be, but just using fsuid at this point stops me from enabling user
> > mounts under /share if /share is chmod 000 (which it is).
>
> I'm dense today. If we can't work out the details we can always use a flag.
> But what is the problem with fsuid?

See above.

> You are not trying to test this using a non-default security model are you?

Nope, at the moment CONFIG_SECURITY=n so I'm running with capabilities
only.

thanks,
-serge

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