Subject: Re: [PATCH 1/6] user namespace : add the framework
Posted by Herbert Poetzl on Wed, 18 Jul 2007 00:11:35 GMT
View Forum Message <> Reply to Message

On Mon, Jul 16, 2007 at 10:08:00AM -0500, Serge E. Hallyn wrote:
> Quoting Kirill Korotaev (dev@sw.ru):
> > Serge E. Hallyn wrote:
> > > Quoting Andrew Morton (akpm@linux-foundation.org):
> > >
> > >>On Mon, 4 Jun 2007 14:40:24 -0500 "Serge E. Hallyn" <serue@us.ibm.com> wrote:
> > >>
> > >>
> > >>>Add the user namespace struct and framework
> > >>>
> > >>>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 ?)
> > >>
> > >>The whole magical-uid-0-user thing in this patch seem just wrong
> > >>to me.
> > >>
> > >>I'll merge it anyway, mainly because I want to merge _something_
> > >>(why oh why do the git-tree guys leave everything to the last
> > >>minute?) but it strikes me that there's something fundamentally
> > >>wrong whenever the kernel starts "knowing" about the significance
> > >>of UIDs in this fashion.
> > >
> > >
> > > $(&(%
> > >
> > > I thought I disagreed, but now I'm pretty sure I completely agree.
> > >
> > > 'root_user' exists in the kernel right now, but the root_user
> > > checks which exist (in fork.c and sys.c) shouldn't actually be
> > > applied for root in a container, since the container may not be
> > > trusted.
> >
> > This rlimit check doesn't help *untrusted* containers, so your logic
> > is wrong here. Instead, it allows root of the container to operate
> > in any situation.
>
> And I'm not sure that should be the case.
>

> In my view, root of a container is equivalent to a normal user on the
> host system, just like root in a qemu process.
>
> > E.g. consider root user hit the limit. After that you won't be able
> > to login/ssh to fix anything.
>
> That's fine in the container.
>
> > NOTE: container root can have no CAP_SYS_RESOURCE and CAP_SYS_ADMIN
> > as it is in OpenVZ.

> And eventually we'll want that to be the default in upstream containers.
> But it's not the case upstream right now. Before we can do that, we
> need an answer to per-container capabilities.
>
> Do you (either you specifically, or anyone at openvz) have plans to
> address the per-container capabilities problem? Herbert? Eric?

it is already addressed in Linux-VServer and OpenVZ
Linux-VServer adds a so called 'capabilitiy mask',
which is applied to the 'normal' capability system,
thus a guest cannot utilize/exercise capabilities
not included in that mask (which makes the guest
root 'secure')

> I'm interested, but would like to get the file capabilites squared
> away before I consider coding on it.
>
> > But in general I'm not against the patch, since in OpenVZ we can
> > replace the check with another capability we use for VE admin -
> > CAP_VE_SYS_ADMIN.
>
> If that truly sufficies then great. If not, then in order to support
> openvz in the meantime I say we drop my patch, but we remember that
> when we straighten out the security issues this will need to be
> addressed.

I'm not very fond of handling guest or host root special
and I think the capability system was designed to exactly
handle the guest root case properly ...

will look through the patches shortly and comment ...

best,
Herbert

> thanks,
> -serge

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

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