Subject: Re: [patch -mm 08/17] nsproxy: add hashtable
Posted by serue on Mon, 11 Dec 2006 20:03:01 GMT
View Forum Message <> Reply to Message

Quoting Eric W. Biederman (ebiederm@xmission.com):
> "Serge E. Hallyn" <serue@us.ibm.com> writes:
>
> > Quoting Serge E. Hallyn (serue@us.ibm.com):
> >> Quoting Eric W. Biederman (ebiederm@xmission.com):
> >> > Herbert Poetzl <herbert@13thfloor.at> writes:
> >> > >> Beyond that yes it seems to make sense to let user space
> >> > >> maintain any mapping of containers to ids.
> >> > >
> >> > > I agree with that, but we need something to move
> >> > > around between the various spaces ...
> >> >
> >> > If you have CAP_SYS_PTRACE or you have a child process
> >> > in a container you can create another with ptrace.
> >> >
> >> > Now I don't mind optimizing that case, with something like
> >> > the proposed bind_ns syscall. But we need to be darn certain
> >> > why it is safe, and does not change the security model that
> >> > we currently have.
> >>
> >> Sigh, and that's going to have to be a discussion per namespace.
> >
> > Well, assuming that we're using pids as identifiers, that means
> > we can only enter decendent namespaces, which means 'we' must
> > have created them. So anything we could do by entering the ns,
> > we could have done by creating it as well, right?
>
> It isn't strict descendents who we can see. i.e. init can create
> the thing, and we could have just logged into the network but init
> and us still share the same pid namespace.
>
> But yes it would be we can only enter descendent namespaces, for
> some definition of enter.
>
> There are two issues.
> 1) We may have a namespace we want to create and then remove the ability
> for the sysadmin to fiddle with, so it can play with encrypted data or
> something like that safely. Not quite unix but it is certainly worth
> considering.

Yeah, that occurred to me, but it doesn't seem like we can possibly make
sufficient guarantees to the client to make this worthwhile.

I'd love to be wrong about that, but if nothing else we can't prove to

the client that they're running on an unhacked host. So the host admin
will always have to be trusted.

> 2) When we only partially enter a namespace it is very easy for additional
>   properties to enter that namespace. For example we enter the pid
>   namespace and the mount namespace, but keep our current working directory
>   in the previous namespace. Then a process in the restricted namespace
>   can get out by cd into /proc/<?>/cwd.

Yup, entering existing namespaces should be all-or-nothing.

> If someones permissions to various objects does not depend on the namespace
> they are in quite possibly this is a non-issue. If we actually depend on
> the isolation to keep things secure enter is a setup for a first rate escape.

I don't believe the isolation can be effective between two namespaces
where one is an ancestor of another. It can be so long as one isn't
the ancestor of another, but then we're not allowing either to enter
the other namespace. So it's not a problem.

The bind_ns() proposed by Cedric is stricter, only allowing nsid 0 to
switch namespaces. So it may be overly restrictive, and does introduce
a new global namespace, but it is safe.

-serge

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