Subject: Re: [RFC PATCH] namespaces: document unshare security implications
Posted by ebiederm on Thu, 15 Nov 2007 18:45:15 GMT
View Forum Message <> Reply to Message

"Serge E. Hallyn" <serue@us.ibm.com> writes:

> Quoting Eric W. Biederman (ebiederm@xmission.com):
>> "Serge E. Hallyn" <serue@us.ibm.com> writes:
>
> Thanks for responding, Eric. Good points.
>
>> > Ok, the following isn't meant so much as a patch as for discussion.
>> > However, this may be a change we want to think about for awhile and
>> > collect opinions and facts. So having this file sitting in the
>> > kernel tree (updated with the results of any discussion we have in the
>> > meantime) may be useful.
>> >
>> > So what do people think? Are we ok using CAP_SYS_ADMIN? Do we
>> > authorize unsharing of each resource using the capability required
>> > to administrate the resource? Do we introduce CAP_NS_UNSHARE? Do
>> > we add CAP_SYS_USHARE, CAP_NET_UNSHARE, and CAP_USER_UNSHARE? Or
>> > do we allow unprivileged users to unshare, trusting that the actual
>> > administration is properly authorized?
>>
>> Well if we ant to sit this in the kernel we need to remove mention
>> of CAP_NS_UNSHARE.
>
> Mostly for now I wanted it to sit in the mailing list :)
>
>> However even there the document below is only an ok first stab at
>> documenting things.
>>
>> The big big big problem are suid executables. If we don't have suid
>> executables and the namespaces only apply to our children we can
>> unshare them all day long and no one cares. If we do have suid executables
>> any messing up of their context that they are not prepared to deal with
>> is a potential security violation.
>
> Ok let's look just at the mounts namespace since that one is complete.
>
> Unsharing the mounts namespace makes no change in mounts context, and
> does not provide the user any additional privilege to make such change.
> So suid executable are safe.

Generally. However we get a stale view, of the mount namespace (which
isn't a big deal because changing mounts is rare) and a stale view
is the equivalent of making a specific change to a namespace from
a suid executables perspective.

I guess I could change /etc/mtab to not reflect the mount namespace
of the initial mount namespace (by running mount), possibly I could
run with an older copy of /dev.

Further you pin filesystems making unmounts difficult and untrackable
with fuser.

I'm not good at coming up with exploits but stale data and state manipulations
that don't manipulate what you think you are manipulating feels like something
a creative person could use to generate an exploit.

A general problem with stale namespaces is that if someone ever does
something stupid you can open the window of vulnerability from just a
moment to years.

>> So I think CAP_SYS_ADMIN is a good starting place. It is trivial verifiable
>> that it is safe. So starting there allows us to work on other aspects
>> of the problem for now.
>
> It was a good starting place, but at this point I have two concerns with
> sticking with CAP_SYS_ADMIN:
>
> 1. now that file capabilities are upstream, people may want to
> add just the requisite capability in fP for an unsharing helper
> program. Cedric had mentioned wanting to do that.
> If we are going to switch to unprivileged unshares, then doing
> so later is ok. But if we're going to switch to a custom
> capability later, then that could be seen as an API change
> since users will have to switch the capability on all the
> unsharing programs.

Ugh. The only capabilities I see that make sense to switch to are
the capabilities needed to make deep changes to a namespace say
(CAP_NET_ADMIN and CAP_NET_RAW be required for the network namespace).

It is a good point that we should work to get the capabilities correct.

> 2. As I pointed out a few times, we can cleanly separate
> unsharing namespace and actually manipulating the resources.
> By requiring CAP_SYS_ADMIN for both unsharing a mounts namespace
> and for performing privileged mounts, any program given the
> authority to unshare is automatically given the authority to
> also completely manipulate the mounts, both in the new private
> namespace and the original namespace (by just not unsharing).
>

> It's even worse with the net namespace, since the privilege
> needed to unshare the namespace authorizes you to update
> *other* namespaces in the system, but *not* network devices!
> But like you say let's stick with established namespaces.
Yes.

> So while I started the original email just wanting some discussion, now
> I'm actually thinking that we should consider the appended patch soon.
>
>> I would like to remove the restrictions from creating new namespaces
>> however we will either have to have restrictions like the current
>> unprivileged mount patches, so we don't surprised root.
>
> Yes, exactly, we need to understand exactly how the resources being
> unshared can be updated and how separate the unsharing and updating
> really are semantically (per namespace). That's why I wanted to start
> floating the document now. I don't think it's something one person can
> sit down and write out in one sitting, bc something will be overlooked.

Definitely. Especially the nasties with suid executables.

>> Or we figure out
>> how to ensure we don't have suid applications.
>
> ... "how to ensure we don't have suid applications" seems the wrong
> level to think at. Rather, for each namespace, what do the tools which
> will be privileged to perform updates depend upon?

Totally. It isn't the unix way to kill suid applications.
suid applications are the Achilles heel in a lot of ways for
applications that want to enhance the unix API as they make it much
much harder. Plan9 gained tremendous support when it dropped the
ability. I'm not at all certain what I think of fine grained
filesystem capabilities. I honestly think that is moving in the wrong
direction. Generally we can solve the same problem in user space with
servers on a system that start out with privileges have a narrow
channel for communicating with the outside world (reducing their
amount of vulnerability), and that drop privileges as they go.

If we try to limit ourselves to what the privileged tools actually
depend on (instead of what they can depend on) we are setting
ourselves up for a world of hurt when we miss something.

> An obvious example is depending on a file location, i.e. /etc/fstab.
> The proper implementation of user mounts solves that. /etc/resolv.conf
> is another example, except in this case we have the updating of the
> network namespace (kinda) depending on proper updates of the mounts

> namespace (and user namespace).
>
>> Given my intuitive understanding of a complete uid namespace it
>> fundamentally prohibits suid executables from executing because those
>> users simply do not exist in the new namespace. So my hunch is we can
>> drop the requirement for CAP_SYS_ADMIN on namespace creation in
>> concert with a uid namespace creation.
>>
>> So my feeling at the moment is that we need to flesh out and complete
>> the namespaces we have user, net, pid, user and then come back and
>> see what we can do.
>
> I think you're saying "what we're doing is at least safe, so let's wait
> to change things."

Largely. Safe at least from the well understood perspective.

> My assertion is that the current approach is not the safest bc we have
> to give unneeded extra authority to a mounts unsharing helper.

True. Safest and most flexible is to figure out how to remove the
need for capabilities at all.

>> I don't think it makes sense to document a snapshot in time of the
>> discussion with unresolved issues in the Documentation directory.
>
> 1. The basics aren't going to change, updating your network namespace
> will require CAP_NET_ADMIN.

Agreed.

> 2. These things should really be considered now, bc resulting
> implications may have effects on the namespace design. As an example,
> the interaction of network namespaces, pid namespaces, netlink sockets,
> and audit daemons makes me uneasy.

Yes. I think that sounds like a fun one to sort through.

>>From 0e04048d0a22cfd9507487a09ca8d7aa500be1c2 Mon Sep 17 00:00:00 2001
> From: Serge E. Hallyn <serue@us.ibm.com>
> Date: Thu, 15 Nov 2007 10:47:32 -0500
> Subject: [PATCH 1/1] namespaces: introduce CAP_NS_UNSHARE for some namespaces
>
> (purely for comment at this point)

I'm not at all ready to consider the implications of a new capability

at this point.

Eric

_____

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