Subject: Re: [patch 0/8] unprivileged mount syscall
Posted by serue on Mon, 09 Apr 2007 17:07:43 GMT
View Forum Message <> Reply to Message

Quoting Miklos Szeredi (miklos@szeredi.hu):
> > > > > One thing that is missing from this series is the ability to restrict
> > > > > user mounts to private namespaces. The reason is that private
> > > > > namespaces have still not gained the momentum and support needed for
> > > > > painless user experience. So such a feature would not yet get enough
> > > > > attention and testing. However adding such an optional restriction
> > > > > can be done with minimal changes in the future, once private
> > > > > namespaces have matured.
> > > >
> > > > I suspect the people who developed and maintain nsproxy would disagree ;)
> > >
> > > Well, they better show me some working and simple-to-use userspace
> > > code, because I've not seen anything like that related to mount
> > > namespaces.
> >
> > If you mean to test/exploit them, see
> > http://lxc.sourceforge.net/patches/2.6.20/2.6.20-lxc8/broken-out/tests/
> >
> > Compile the ns_exec.c program and do
> >
> > ns_exec -m /bin/sh
> >
> > to get a shell in a new mounts namespace.
>
> Cool, thanks. This is a very nice utility for testing, but for the
> end user rather useless:

Well that depends on which end-user. Those wanting to create a vserver
or checkpoint-restart job will want this, but clearly we have a long way
to go for that upstream anyway.

>   - user starts up a private namespace in a shell, mounts something
>
>   - then opens app from menu, tries to access mount, but the mount is
>   not there
>
>   - user unhappy
>
> BTW, looking at -mm unshare() on namespace is not privileged any more.
> Why is that? Or rather, what's the reason, that clone() is privileged
> and unshare() is not?

The check is still there - see kernel/nsproxy.c:unshare_nsproxy_namespaces().

> > > pam_namespace.so is one example of a non-working, but probably-not-too-
> > > hard-to-fix one.
> >
> > Non-working? I sure hope the one used for LSPP certification is
> > working... As is the ugly version I wrote 18 mounts ago and use on my
> > laptop.
>
> The one in pam-0.99.6.3-29.1 in opensuse-10.2 is totally broken. Are
> you interested in the details? I can reproduce it, but forgot to note
> down the details of the brokenness.

I don't know how far removed that is from the one being used by redhat,
but assuming it's the same, then redhat-lspp@redhat.com will be
very interested.

> > > I'm just saying this is not yet something that Joe Blow would just
> > > enable by ticking a box in their desktop setup wizard, and it would
> > > all work flawlessly thereafter. There's still a _long_ way towards
> > > that, and mostly in userspace.
> >
> > I'm not sure there's a that long a way to go, but clearly we need to be
> > showing users what they can do, or they'll never work their way towards
> > there.
>
> There _is_ a long way to go. Random things that spring to my mind:
>
> - using /etc/mtab is broken with private namespaces, using
>   /proc/mounts is missing various functionality, that /etc/mtab has,
>   for example the "user" option, which this patchset adds

Agreed those need fixing.

> - need to set up mount propagation from global namespace to private
>   ones, mount(8) does not yet have options to configure propagation

Hmm, I guess I get lost using my own little systems, and just assumed
that shared subtree functionality was making its way up into mount(8).
Ram, have you been working on that?

> - user namespace setup: what if user has multiple sessions?
>
>   1) namespaces are shared? That's tricky because the session needs to
>   be a child of a namespace server, not of login. I'm not sure PAM
>   can handle this
>
>   2) or mounts are copied on login? That's not possible currently,
>   as there's no way to send a mount between namespaces. Also it's
>   tricky to make sure that new mounts are also shared

See toward the end of the 'shared subtrees' OLS paper from last year for
a suggestion on how to let users effectively 'log in to' an existing
private mounts ns.

> > For instance, as you say, a user admin gui with a checkmark and text
> > boxes saying 'enter new namespace on login', 'create private /tmp',
> > and 'create private dmcrypted /home' would be trivial right now.
>
> Trivial modulo the above slightly non-trivial exemptions ;)

Ok, so it can use some very non-trivial fine-tuning...

But I've been using the above - minus the trivial gui - for over a year
without ever worrying about any of these short-comings.

> Miklos

-serge

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

---

Subject: Re: [patch 0/8] unprivileged mount syscall
Posted by Miklos Szeredi on Mon, 09 Apr 2007 20:10:41 GMT
View Forum Message <> Reply to Message

> > The one in pam-0.99.6.3-29.1 in opensuse-10.2 is totally broken. Are
> > you interested in the details? I can reproduce it, but forgot to note
> > down the details of the brokenness.
>
> I don't know how far removed that is from the one being used by redhat,
> but assuming it's the same, then redhat-lspp@redhat.com will be
> very interested.

OK.

> > - user namespace setup: what if user has multiple sessions?
> >
> >   1) namespaces are shared? That's tricky because the session needs to
> >   be a child of a namespace server, not of login. I'm not sure PAM
> >   can handle this
> >
> >   2) or mounts are copied on login? That's not possible currently,
> >   as there's no way to send a mount between namespaces. Also it's
> >   tricky to make sure that new mounts are also shared

>
> See toward the end of the 'shared subtrees' OLS paper from last year for
> a suggestion on how to let users effectively 'log in to' an existing
> private mounts ns.

This?

1. create a new namespace
2. bind /share/$USER to /share
3. for each pair ($who, $what) such that
   /share/$USER/$who/$what exists, look
   in /share/$who/allowed for "peer $what
   $USER" or "slave $what $USER". If the
   former is found, rbind /share/$who/$what
   on /share/$USER/$who/$what; if the
   latter is found, do the same and
   follow with marking subtree under
   /share/$USER/$who/$what as slave.
4. rbind /share/$USER to /share
5. mark subtree under /share as private.
6. umount -l /share

Well, someone please explain using short words, because I don't
understand at all.

Thanks,
Miklos

_____

Subject: Re: [patch 0/8] unprivileged mount syscall
Posted by Ram Pai on Tue, 10 Apr 2007 08:38:04 GMT
View Forum Message <> Reply to Message

On Mon, 2007-04-09 at 22:10 +0200, Miklos Szeredi wrote:
> > > The one in pam-0.99.6.3-29.1 in opensuse-10.2 is totally broken. Are
> > > you interested in the details? I can reproduce it, but forgot to note
> > > down the details of the brokenness.
> >
> > I don't know how far removed that is from the one being used by redhat,
> > but assuming it's the same, then redhat-lspp@redhat.com will be
> > very interested.
>
> OK.
>

> > > - user namespace setup: what if user has multiple sessions?
> > >
> > >   1) namespaces are shared? That's tricky because the session needs to
> > >   be a child of a namespace server, not of login. I'm not sure PAM
> > >   can handle this
> > >
> > >   2) or mounts are copied on login? That's not possible currently,
> > >   as there's no way to send a mount between namespaces. Also it's
> > >   tricky to make sure that new mounts are also shared
> >
> > See toward the end of the 'shared subtrees' OLS paper from last year for
> > a suggestion on how to let users effectively 'log in to' an existing
> > private mounts ns.
>
> This?
>
>   1. create a new namespace
>   2. bind /share/$USER to /share
>   3. for each pair ($who, $what) such that
>   /share/$USER/$who/$what exists, look
>   in /share/$who/allowed for "peer $what
> $USER" or "slave $what $USER". If the
>   former is found, rbind /share/$who/$what
>   on /share/$USER/$who/$what; if the
>   latter is found, do the same and
>   follow with marking subtree under
>   /share/$USER/$who/$what as slave.
>   4. rbind /share/$USER to /share
>   5. mark subtree under /share as private.
>   6. umount -l /share
>
> Well, someone please explain using short words, because I don't
> understand at all.

I am trying to re-construct Viro's thoughts. I think the steps outlined
above; though not accurate, are still insightful.

The idea is -- there is one master namespace, which has
under /share, a replica of the mount tree of namespaces belonging to all
users.

for example if there are two users A and B, then in the master namespace
under /share you will find /share/A and /share/B, each reflecting the
mount tree for the namespaces belonging to user-A and user-B
respectively.

Note: /share is a shared mount-tree, which means it can propagate mount
events.

Everytime the user logs on the machine, a new namespace is created which
is the clone of the master namespace. In this new namespace,
the /share/$user is made the root of the namespace. Also if other
users have allowed part of their namespace available to this user,
than those mounts are also brought under this namespace. And finally the
entire tree under /share is unmounted.

Note, though multiple namespaces can exist simultaneously for the same
user, the user is provided the illusion of per-process-namespace since
all the namespaces look identical.

I am trying to rewrite the steps outlined above, which may or may not
reflect Viro's thoughts, but certainly reflect my reconstruction of
viro's thoughts.

1. clone the master namespace.

2. in the new namespace

```
 move the tree under /share/$me to /
     for each ($user, $what, $how) {
         move /share/$user/$what to /$what
   if ($how == slave) {
             make the mount tree under /$what as slave
         }
     }
```

3. in the new namespace make the tree under
     /share as private and unmount /share

RP

>
> Thanks,
> Miklos

_____

Subject: Re: [patch 0/8] unprivileged mount syscall

> 1. clone the master namespace.
>
> 2. in the new namespace
>
> move the tree under /share/$me to /
>   for each ($user, $what, $how) {
>   move /share/$user/$what to /$what
>   if ($how == slave) {
>   make the mount tree under /$what as slave
>   }
>   }
>
> 3. in the new namespace make the tree under
>   /share as private and unmount /share

Thanks. I get the basic idea now: the namespace itself need not be
shared between the sessions, it is enough if "share" propagation is
set up between the different namespaces of a user.

I don't yet see either in your or Viro's description how the trees
under /share/$USER are initialized. I guess they are recursively
bound from /, and are made slaves.

Miklos

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