Subject: Re: [patch 0/8] unprivileged mount syscall
Posted by serue on Fri, 13 Apr 2007 13:28:32 GMT
View Forum Message <> Reply to Message

Quoting Miklos Szeredi (miklos@szeredi.hu):
> > On Wed, 2007-04-11 at 12:44 +0200, Miklos Szeredi wrote:
> > > > 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/$USER are initialized. I guess they are recursively
> > > bound from /, and are made slaves.
> >
> > yes. I suppose, when a userid is created one of the steps would be
> >
> > mount --rbind / /share/$USER
> > mount --make-rslave /share/$USER
> > mount --make-rshared /share/$USER
>
> Thinking a bit more about this, I'm quite sure most users wouldn't
> even want private namespaces. It would be enough to
>
>   chroot /share/$USER
>
> and be done with it.
>
> Private namespaces are only good for keeping a bunch of mounts
> referenced by a group of processes. But my guess is, that the natural
> behavior for users is to see a persistent set of mounts.
>
> If for example they mount something on a remote machine, then log out
> from the ssh session and later log back in, they would want to see

> their previous mount still there.
>
> Miklos

Agreed on desired behavior, but not on chroot sufficing. It actually
sounds like you want exactly what was outlined in the OLS paper.

Users still need to be in a different mounts namespace from the admin
user so long as we consider the deluser and backup problems to be
legitimate problems (well, so long as user mounts are allowed). So,
when they log in, pam gives them a new namespace and chroots them into
/share/$USER.

Assuming I'm thinking clearly :)

-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 Fri, 13 Apr 2007 14:05:16 GMT
View Forum Message <> Reply to Message

> > Thinking a bit more about this, I'm quite sure most users wouldn't
> > even want private namespaces. It would be enough to
> >
> > chroot /share/$USER
> >
> > and be done with it.
> >
> > Private namespaces are only good for keeping a bunch of mounts
> > referenced by a group of processes. But my guess is, that the natural
> > behavior for users is to see a persistent set of mounts.
> >
> > If for example they mount something on a remote machine, then log out
> > from the ssh session and later log back in, they would want to see
> > their previous mount still there.
> >
> > Miklos
>
> Agreed on desired behavior, but not on chroot sufficing. It actually
> sounds like you want exactly what was outlined in the OLS paper.
>
> Users still need to be in a different mounts namespace from the admin
> user so long as we consider the deluser and backup problems

I don't think it matters, because /share/$USER duplicates a part or
the whole of the user's namespace.

So backup would have to be taught about /share anyway, and deluser
operates on /home/$USER and not on /share/*, so there shouldn't be any
problem.

There's actually very little difference between rbind+chroot, and
CLONE_NEWNS. In a private namespace:

1) when no more processes reference the namespace, the tree will be
disbanded

2) the mount tree won't be accessible from outside the namespace

Wanting a persistent namespace contradicts 1).

Wanting a per-user (as opposed to per-session) namespace contradicts
2). The namespace _has_ to be accessible from outside, so that a new
session can access/copy it.

So both requirements point to the rbind/chroot solution.

Miklos

_____
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 serue on Fri, 13 Apr 2007 21:44:15 GMT
View Forum Message <> Reply to Message

Quoting Miklos Szeredi (miklos@szeredi.hu):
> > > Thinking a bit more about this, I'm quite sure most users wouldn't
> > > even want private namespaces. It would be enough to
> > >
> > >   chroot /share/$USER
> > >
> > > and be done with it.
> > >
> > > Private namespaces are only good for keeping a bunch of mounts
> > > referenced by a group of processes. But my guess is, that the natural
> > > behavior for users is to see a persistent set of mounts.
> > >
> > > If for example they mount something on a remote machine, then log out

> > > from the ssh session and later log back in, they would want to see
> > > their previous mount still there.
> > >
> > > Miklos
> >
> > Agreed on desired behavior, but not on chroot sufficing. It actually
> > sounds like you want exactly what was outlined in the OLS paper.
> >
> > Users still need to be in a different mounts namespace from the admin
> > user so long as we consider the deluser and backup problems
>
> I don't think it matters, because /share/$USER duplicates a part or
> the whole of the user's namespace.
>
> So backup would have to be taught about /share anyway, and deluser
> operates on /home/$USER and not on /share/*, so there shouldn't be any
> problem.

In what I was thinking of, /share/$USER is bind mounted to
~$USER/share, so it would have to be done in a private namespace in
order for deluser to not be tricked.

> There's actually very little difference between rbind+chroot, and
> CLONE_NEWNS. In a private namespace:
>
>   1) when no more processes reference the namespace, the tree will be
>   disbanded
>
>   2) the mount tree won't be accessible from outside the namespace

But it *can* be, if properly set up. That's part of the point of the
example in the OLS paper. When a user logs in, sshd clones a new
namespace, then bind-mounts /share/$USER into ~$USER/share. So assuming
that /share/$USER was --make-shared'd, it and ~$USER are now in the
same peer group, and any changes made by the user under ~$USER will
be reflected back into /share/$USER.

> Wanting a persistent namespace contradicts 1).

Not necessarily, see above.

> Wanting a per-user (as opposed to per-session) namespace contradicts
> 2). The namespace _has_ to be accessible from outside, so that a new
> session can access/copy it.

Again, I *think* you are wrong that private namespace contradicts this
requirement.

> So both requirements point to the rbind/chroot solution.

It all points to a combination of the two :-)

-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 Sun, 15 Apr 2007 20:39:40 GMT
View Forum Message <> Reply to Message

> > > Agreed on desired behavior, but not on chroot sufficing. It actually
> > > sounds like you want exactly what was outlined in the OLS paper.
> > >
> > > Users still need to be in a different mounts namespace from the admin
> > > user so long as we consider the deluser and backup problems
> > >
> > I don't think it matters, because /share/$USER duplicates a part or
> > the whole of the user's namespace.
> >
> > So backup would have to be taught about /share anyway, and deluser
> > operates on /home/$USER and not on /share/*, so there shouldn't be any
> > problem.
>
> In what I was thinking of, /share/$USER is bind mounted to
> ~$USER/share, so it would have to be done in a private namespace in
> order for deluser to not be tricked.

But /share/$USER is surely not bind mounted to ~$USER/share in the
global namespace, is it? I can't see any sense in that.

> > There's actually very little difference between rbind+chroot, and
> > CLONE_NEWNS. In a private namespace:
> >
> > 1) when no more processes reference the namespace, the tree will be
> > disbanded
> >
> > 2) the mount tree won't be accessible from outside the namespace
>
> But it *can* be, if properly set up. That's part of the point of the
> example in the OLS paper. When a user logs in, sshd clones a new
> namespace, then bind-mounts /share/$USER into ~$USER/share. So assuming
> that /share/$USER was --make-shared'd, it and ~$USER are now in the
> same peer group, and any changes made by the user under ~$USER will

> be reflected back into /share/$USER.

I acknowledge, that it can be done. My point was that it can be done
more simply _without_ using CLONE_NS.

> > Wanting a persistent namespace contradicts 1).
>
> Not necessarily, see above.
>
> > Wanting a per-user (as opposed to per-session) namespace contradicts
> > 2). The namespace _has_ to be accessible from outside, so that a new
> > session can access/copy it.
>
> Again, I *think* you are wrong that private namespace contradicts this
> requirement.

I'm not saying there's any contradiction, I'm saying rbind+chroot is a
better fit.

I haven't yet heard a single reason why a per-session namespace with
parts shared per-user is better than just a per-user namespace.

Miklos

Subject: Re: [patch 0/8] unprivileged mount syscall
Posted by Ram Pai on Mon, 16 Apr 2007 08:18:29 GMT
View Forum Message <> Reply to Message

On Fri, 2007-04-13 at 16:05 +0200, Miklos Szeredi wrote:
> > > Thinking a bit more about this, I'm quite sure most users wouldn't
> > > even want private namespaces. It would be enough to
> > >
> > > chroot /share/$USER
> > >
> > > and be done with it.
> > >
> > > Private namespaces are only good for keeping a bunch of mounts
> > > referenced by a group of processes. But my guess is, that the natural
> > > behavior for users is to see a persistent set of mounts.
> > >
> > > If for example they mount something on a remote machine, then log out
> > > from the ssh session and later log back in, they would want to see
> > > their previous mount still there.

> > >
> > > Miklos
> >
> > Agreed on desired behavior, but not on chroot sufficing. It actually
> > sounds like you want exactly what was outlined in the OLS paper.
> >
> > Users still need to be in a different mounts namespace from the admin
> > user so long as we consider the deluser and backup problems
>
> I don't think it matters, because /share/$USER duplicates a part or
> the whole of the user's namespace.
>
> So backup would have to be taught about /share anyway, and deluser
> operates on /home/$USER and not on /share/*, so there shouldn't be any
> problem.
>
> There's actually very little difference between rbind+chroot, and
> CLONE_NEWNS. In a private namespace:
>
>   1) when no more processes reference the namespace, the tree will be
>   disbanded
>
>   2) the mount tree won't be accessible from outside the namespace
>
> Wanting a persistent namespace contradicts 1).
>
> Wanting a per-user (as opposed to per-session) namespace contradicts
> 2). The namespace _has_ to be accessible from outside, so that a new
> session can access/copy it.

As i mentioned in the previous mail, disbanding all the namespaces of a user will not disband his mount tree, because a mirror of the mount tree still continues to exist in /share/$USER in the admin namespace.

And a new user session can always use this copy to create a namespace that looks identical to that which existed earlier.

>
> So both requirements point to the rbind/chroot solution.

Arn't there ways to escape chroot jails? Serge had pointed me to a URL which showed chroots can be escaped. And if that is true than having all user's private mount tree in the same namespace can be a security issue?

RP

>

> Miklos

_____

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, 16 Apr 2007 09:27:45 GMT
View Forum Message <> Reply to Message

> Arn't there ways to escape chroot jails? Serge had pointed me to a URL
> which showed chroots can be escaped. And if that is true than having all
> user's private mount tree in the same namespace can be a security issue?

No. In fact chrooting the user into /share/$USER will actually
grant a privilege to the user, instead of taking it away. It allows
the user to modify it's root namespace, which it wouldn't be able to
in the initial namespace.

So even if the user could escape from the chroot (which I doubt), s/he
would not be able to do any harm, since unprivileged mounting would be
restricted to /share. Also /share/$USER should only have read/search
permission for $USER or no permissions at all, which would mean, that
other users' namespaces would be safe from tampering as well.

Miklos

_____

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

---