Subject: Re: Re: [RFC][PATCH 0/2] user namespace [try #2]
Posted by dev on Thu, 07 Sep 2006 15:37:01 GMT
View Forum Message <> Reply to Message

> Here's a stab at semantics for how to handle file access. Should be
> pretty simple to implement, but i won't get a chance to implement this
> week.
>
> At mount, by default the vfsmount is tagged with a uid_ns.
> A new -o uid_ns=<pid> option instead tags the vfsmount with the uid_ns
> belonging to pid <pid>. Since any process in a descendent pid
> namespace should still have a valid pid in the ancestor
> pidspaces, this should work fine.
> At vfs_permission, if current->nsproxy->uid_ns != file->f_vfsmnt->uid_ns,
> 1. If file is owned by root, then read permission is granted
> 2. If file is owned by non-root, no permission is granted
> (regardless of process uid)
>
> Does this sound reasonable?
imho this in acceptable for OpenVZ as makes VE files to be inaccessiable from
host. At least this is how I understand your idea...
Am I correct?

> I assume the list of other things we'll need to consider includes
> signals between user namespaces
> keystore
> sys_setpriority and the like
> I might argue that all of these should be sufficiently protected
> by proper setup by userspace. Can you explain why that is not
> the case?
The same requirement (ability to send signals from host to VE)
is also applicable to signals.

Thanks,
Kirill

Subject: Re: Re: [RFC][PATCH 0/2] user namespace [try #2]
Posted by Herbert Poetzl on Thu, 07 Sep 2006 15:48:57 GMT
View Forum Message <> Reply to Message

On Thu, Sep 07, 2006 at 07:40:23PM +0400, Kirill Korotaev wrote:
> > Here's a stab at semantics for how to handle file access. Should be
> > pretty simple to implement, but i won't get a chance to implement this
> > week.
> >
> > At mount, by default the vfsmount is tagged with a uid_ns.
> > A new -o uid_ns=<pid> option instead tags the vfsmount with the uid_ns

> > belonging to pid <pid>. Since any process in a descendent pid
> > namespace should still have a valid pid in the ancestor
> > pidspaces, this should work fine.
> > At vfs_permission, if current->nsproxy->uid_ns != file->f_vfsmnt->uid_ns,
> > 1. If file is owned by root, then read permission is granted
> > 2. If file is owned by non-root, no permission is granted
> > (regardless of process uid)
> >
> > Does this sound reasonable?

> imho this in acceptable for OpenVZ as makes VE files to be
> inaccessiable from host. At least this is how I understand your
> idea... Am I correct?
>
> > I assume the list of other things we'll need to consider includes
> > signals between user namespaces
> > keystore
> > sys_setpriority and the like
> > I might argue that all of these should be sufficiently protected
> > by proper setup by userspace. Can you explain why that is not
> > the case?

> The same requirement (ability to send signals from host to VE)
> is also applicable to signals.

at some point, we tried to move all cross context
signalling (from the host to the guests) into a special
context, but later on we moved away from that, because
it was much simpler and more intuitive to handle the
signalling with a separate syscall command

what I want to point out here is, that things like
sending signals across namespaces is something which
is not required to make this work

best,
Herbert

> Thanks,
> Kirill
>
> _____
> Containers mailing list
> Containers@lists.osdl.org
> https://lists.osdl.org/mailman/listinfo/containers

Subject: Re: Re: [RFC][PATCH 0/2] user namespace [try #2]
Posted by serue on Thu, 07 Sep 2006 15:53:37 GMT
View Forum Message <> Reply to Message

Quoting Kirill Korotaev (dev@sw.ru):
> > Here's a stab at semantics for how to handle file access. Should be
> > pretty simple to implement, but i won't get a chance to implement this
> > week.
> >
> > At mount, by default the vfsmount is tagged with a uid_ns.
> > A new -o uid_ns=<pid> option instead tags the vfsmount with the uid_ns
> > belonging to pid <pid>. Since any process in a descendent pid
> > namespace should still have a valid pid in the ancestor
> > pidspaces, this should work fine.
> > At vfs_permission, if current->nsproxy->uid_ns != file->f_vfsmnt->uid_ns,
> > 1. If file is owned by root, then read permission is granted
> > 2. If file is owned by non-root, no permission is granted
> > (regardless of process uid)
> >
> > Does this sound reasonable?
> imho this in acceptable for OpenVZ as makes VE files to be inaccessiable from
> host. At least this is how I understand your idea...
> Am I correct?

Only if the host did the setup correctly. Either it could do

 mount -o uid_ns=<pid> /dev/hdc1 /mnt/guest/root/5

right off the bat, or it could simply

 mount -o uid_ns=<pid> --bind /mnt/guest/root/5 /mnt/guest/root/5

since after that, any access under /mnt/guest/root/5 would be looked up
with the vfsmount belonging to the guest's uid namespace.

> > I assume the list of other things we'll need to consider includes
> > signals between user namespaces
> > keystore
> > sys_setpriority and the like
> > I might argue that all of these should be sufficiently protected
> > by proper setup by userspace. Can you explain why that is not
> > the case?
> The same requirement (ability to send signals from host to VE)
> is also applicable to signals.

This property should be inherent to the use of a pid_ns. Let's say the
host is in pid_ns one, and creates a new pid_ns 2. pid_ns 2 has a
process known as (pid_ns 2, pid 22). There will be another 'struct pid'

pointing to the same task_struct, calling it (pid_ns 1, pid 578).

So a process in pid_ns 1 can signal (pid_ns 2, pid 22) by sending a
signal to pid 578.

A proces in pid_ns 2 has no reference to any process in pid_ns 1 (and
not in pid_ns 2), therefore cannot signal those processes.

-serge

Subject: Re: Re: [RFC][PATCH 0/2] user namespace [try #2]
Posted by dev on Thu, 07 Sep 2006 16:05:58 GMT
View Forum Message <> Reply to Message

>>imho this in acceptable for OpenVZ as makes VE files to be
>>inaccessiable from host. At least this is how I understand your
>>idea... Am I correct?
>>
>>
>>>I assume the list of other things we'll need to consider includes
>>> signals between user namespaces
>>> keystore
>>> sys_setpriority and the like
>>>I might argue that all of these should be sufficiently protected
>>>by proper setup by userspace. Can you explain why that is not
>>>the case?
>
>
>>The same requirement (ability to send signals from host to VE)
>>is also applicable to signals.
>
>
> at some point, we tried to move all cross context
> signalling (from the host to the guests) into a special
> context, but later on we moved away from that, because
> it was much simpler and more intuitive to handle the
> signalling with a separate syscall command
I'm not sure what a separate context is for, but a separate syscall
is definetely not a good idea.

> what I want to point out here is, that things like
> sending signals across namespaces is something which
> is not required to make this work
well, people have different requirements...

Kirill

Subject: Re: Re: [RFC][PATCH 0/2] user namespace [try #2]
Posted by Herbert Poetzl on Thu, 07 Sep 2006 17:55:20 GMT
View Forum Message <> Reply to Message

On Thu, Sep 07, 2006 at 08:09:38PM +0400, Kirill Korotaev wrote:
> >>imho this in acceptable for OpenVZ as makes VE files to be
> >>inaccessiable from host. At least this is how I understand your
> >>idea... Am I correct?
> >>
> >>
> >>>I assume the list of other things we'll need to consider includes
> >>> signals between user namespaces
> >>> keystore
> >>> sys_setpriority and the like
> >>>I might argue that all of these should be sufficiently protected
> >>>by proper setup by userspace. Can you explain why that is not
> >>>the case?
> >
> >
> >>The same requirement (ability to send signals from host to VE)
> >>is also applicable to signals.
> >
> >
> >at some point, we tried to move all cross context
> >signalling (from the host to the guests) into a special
> >context, but later on we moved away from that, because
> >it was much simpler and more intuitive to handle the
> >signalling with a separate syscall command

> I'm not sure what a separate context is for, but a separate syscall
> is definetely not a good idea.

care to explain _why_ you think so?

> >what I want to point out here is, that things like
> >sending signals across namespaces is something which
> >is not required to make this work

> well, people have different requirements...

of course, it's all about 'different' requirements ...

TIA,
Herbert

> Kirill

Subject: Re: Re: [RFC][PATCH 0/2] user namespace [try #2]
Posted by dev on Tue, 12 Sep 2006 13:48:46 GMT
View Forum Message <> Reply to Message

Herbert Poetzl wrote:
> On Thu, Sep 07, 2006 at 08:09:38PM +0400, Kirill Korotaev wrote:
>
>>>>imho this in acceptable for OpenVZ as makes VE files to be
>>>>inaccessiable from host. At least this is how I understand your
>>>>idea... Am I correct?
>>>>
>>>>
>>>>
>>>>I assume the list of other things we'll need to consider includes
>>>> signals between user namespaces
>>>> keystore
>>>> sys_setpriority and the like
>>>>I might argue that all of these should be sufficiently protected
>>>>by proper setup by userspace. Can you explain why that is not
>>>>the case?
>>>
>>>
>>>>The same requirement (ability to send signals from host to VE)
>>>>is also applicable to signals.
>>>
>>>
>>>at some point, we tried to move all cross context
>>>signalling (from the host to the guests) into a special
>>>context, but later on we moved away from that, because
>>>it was much simpler and more intuitive to handle the
>>>signalling with a separate syscall command
>
>
>>I'm not sure what a separate context is for, but a separate syscall
>>is definetely not a good idea.
>
>
> care to explain _why_ you think so?
cause duplicating syscalls with the same meaning but just working in a bit
different situations doesn't look good.

Kirill

Subject: Re: Re: [RFC][PATCH 0/2] user namespace [try #2]
Posted by Herbert Poetzl on Tue, 12 Sep 2006 14:07:08 GMT
View Forum Message <> Reply to Message

On Tue, Sep 12, 2006 at 05:52:40PM +0400, Kirill Korotaev wrote:
> Herbert Poetzl wrote:
> > On Thu, Sep 07, 2006 at 08:09:38PM +0400, Kirill Korotaev wrote:

> >>>>imho this in acceptable for OpenVZ as makes VE files to be
> >>>>inaccessiable from host. At least this is how I understand your
> >>>>idea... Am I correct?

> >>>>I assume the list of other things we'll need to consider includes
> >>>> signals between user namespaces
> >>>> keystore
> >>>> sys_setpriority and the like
> >>>>I might argue that all of these should be sufficiently protected
> >>>>by proper setup by userspace. Can you explain why that is not the
> >>>>case?

> >>>>The same requirement (ability to send signals from host to VE)
> >>>>is also applicable to signals.

> >>>at some point, we tried to move all cross context signalling
> >>>(from the host to the guests) into a special context, but later
> >>>on we moved away from that, because it was much simpler and more
> >>>intuitive to handle the signalling with a separate syscall command

> >>I'm not sure what a separate context is for, but a separate syscall
> >>is definetely not a good idea.

> > care to explain _why_ you think so?
> cause duplicating syscalls with the same meaning but just working in a
> bit different situations doesn't look good.

hmm ... well, I guess the kernel doesn't look too good then :)

```
 .long sys_setuid16
 .long sys_getuid16
 .long sys_geteuid16
 .long sys_setreuid16 /* 70 */
 .long sys_setfsuid16
 .long sys_setresuid16
 .long sys_getresuid16 /* 165 */
 .long sys_getuid
 .long sys_geteuid
 .long sys_setreuid
 .long sys_setresuid
 .long sys_getresuid
 .long sys_setuid
 .long sys_setfsuid /* 215 */
```

```
.long sys_umount /* recycled never used phys() */
.long sys_oldumount

.long sys_olduname
.long sys_uname
.long sys_newuname

.long sys_old_getrlimit
.long sys_getrlimit
```

best,
Herbert

> Kirill
