Subject: Re: [patch -mm 10/17] nsproxy: add unshare_ns and bind_ns syscalls
Posted by Herbert Poetzl on Sat, 09 Dec 2006 08:22:02 GMT
View Forum Message <> Reply to Message

On Sat, Dec 09, 2006 at 12:40:03AM -0700, Eric W. Biederman wrote:
> Herbert Poetzl <herbert@13thfloor.at> writes:
>
> > On Fri, Dec 08, 2006 at 12:26:49PM -0700, Eric W. Biederman wrote:
> >> clg@fr.ibm.com writes:
> >>
> >> > From: Cedric Le Goater <clg@fr.ibm.com>
> >> >
> >> > The following patch defines 2 new syscalls specific to nsproxy and
> >> > namespaces :
> >> >
> >> > * unshare_ns :
> >> >
> >> > enables a process to unshare one or more namespaces. this
> >> > duplicates the unshare syscall for the moment but we
> >> > expect to diverge when the number of namespaces increases
> >>
> >> Are we out of clone flags yet? If not this is premature.
> >
> > no, but a different nevertheless related question:
> > does anybody, except for 'us' use the unshare() syscall?
>
> The pam_namespace module if I have looked at things properly.
> I believe that is what it was added to support.

hmm, too bad then ...

> > because if not, then why not simply extend that one
> > to 64bit and be done, we probably won't need a clone64()
> > but if we find we do (at some point) adding that with
> > the new flags would be trivial ...
> >
> > OTOH, we could also just add an unshare64() too
> >
> > anyway, we _will_ run out of flags in the near future
>
> Agreed. Please let's cross that bridge when we come to it.

well, personally I'd prefer to 'know' that the
interfaces we introduce now to handle the 'new'
namespaces will still work in a few months (or
maybe years?) and not require _another_ change

let me give an example here:

we now change our userspace tools to support
new clone() flags, we can do that with a little
support from the kernel quite easily without
changing the tools with every kernel release and
more important, without breaking backwards
compatibility in the Linux-VServer ABI (and API)

if we decide to switch to a completely different
API in, lets say six month from now, 'we' have
two options:

- add _another_ compatibility layer to handle
  the 'old' (i.e. now introduced) ABI
- accept that older tools will fail and/or
  produce strange results because 'our' ABI
  isn't supported anymore ...

so naturally, I'm not very excited to introduce
and/or utilize and interface which we all _know_
is not able to satisfy the upcoming demands ...

nevertheless, that isn't considered a major issue
here, so feel free to ignore my personal feelings

> >> I'm also worried about the security implications of switching
> >> namespaces on a process.
> >> That is something that needs to be looked at very closely.
> >
> > Linux-VServer currently uses a capability to prevent
> > changing between namespaces (a very generic one) but
> > it probably makes sense to add something like that
> > in general ... btw, did I mention that the capability
> > flags are running out too?
>
> I think they have run out. Not that sys_capability needs a
> revision but it appears the format of the data does which
> is likely just as bad.

gladly not in 2.6.19, as we absolutely need to
add one capability (CAP_CONTEXT) in Linux-VServer
(the one allowing to utilize sys_vserver)

but once that is taken by mainline, we have to
add another workaround ...

sidenote: I made several suggestions to extend
the capability system (in number of caps and

functionality) of which none was even considered

> >> These two changes certainly don't belong in a single patch,
> >> and they certainly use a bit more explanation.
> >> syscalls are not something to add lightly.
> >
> > well, and they will take ages to get into mainline
> > for all archs, or has that changed sine we reserved
> > sys_vserver()?
>
> I think it is likely a little better. I'm not certain what
> your definition of ages is.

took slightly more than a year, IIRC ...

> >> Because they must be supported forever.
> >
> > I'm not sure about that, most archs 'reuse' syscalls
> > when there is no user left ...
>
> I haven't seen that on i386. Except for experimental
> syscalls I have a hard time believing we have any syscalls
> that have had all of their users disappear.

okay, so be it ...

> Reusing syscall numbers is in a lot of ways completely
> irresponsible once you start supporting a binary interface.
> Even if you remove the syscall because there are no users
> or it makes absolutely no sense any more.

okay, no problem with keeping them around ...

best,
Herbert

> Eric

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