Subject: Re: [RFC] ns containers (v2): namespace entering
Posted by ebiederm on Thu, 22 Feb 2007 20:42:46 GMT
View Forum Message <> Reply to Message

"Paul Menage" <menage@google.com> writes:

> When I implemented a virtual server solution at my previous job, we
> solved the problem of leaking capabilities into the virtualized
> environment by allowing a process to enter the virtual server in a
> "privileged" mode, in which it didn't appear in the virtualized /proc,
> and hence none of its resources were accessible to processes in the
> VS. Children of a privileged process were by default regular members
> of the virtual server.
>
> With the nsproxy model, maybe you could implement that by having two
> pid namespaces. When you create the VS, you create an entire set of
> new namespaces; then before forking init you create a new pid_ns. So
> by entering the outer nsproxy you'd get access to the same environment
> that the VS sees, but your process wouldn't be visible to processes in
> the VS. If you wanted to create a regular process, you'd just enter
> the inner pid_ns too. I think that doing the appropriate process
> "sanitization" to avoid leaking capabilities would be easier from the
> "twilight zone" intermediate nsproxy than from the machine top-level.

And this is why I keep coming back to ptrace. It doesn't leak and
it does seem to get the job done. If you have to because you can
run system calls you can implement exec by hand. ptrace also allows
many of the intermediate enter scenarios.

ptrace is clumsy and there does have to be a better way but until I find
something that isn't clumsy I will stick with ptrace.

Now it is at least worth investigating if you can leak things if you don't
enter the pid namespace. If you can not leak things that potentially
simplifies big chunks of the problem, and we probably don't need the
intermediate pid namespace, of your suggestion.

Still I'm drawn back to the simplicity of the proof of the ptrace
situation.

On another note Serge there is another possibility for dealing with
fs_struct. You can chdir(/proc/<someotherpidinthenamespace>/) and
get your working directory changed after the unshare.

All I know for certain is that it really sucks designing things with
new semantics that interact closely with all of the old existing
rules. It is really easy to come up with something that doesn't

work well, or is a major hazard when people use it in unexpected ways.

So I'm generally in favor of not rushing things and taking the big
things like enter in as small a step as we can. Because whatever
we merge we will likely need to be supported for the rest our lives.

Eric

_____

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