Subject: Re: [PATCH 0/13] Pid namespaces (OpenVZ view)
Posted by Pavel Emelianov on Fri, 25 May 2007 07:06:49 GMT
View Forum Message <> Reply to Message

Serge E. Hallyn wrote:
> Quoting Pavel Emelianov (xemul@sw.ru):
>> Serge E. Hallyn wrote:
>>> Quoting Pavel Emelianov (xemul@openvz.org):
>>>> That's how OpenVZ sees the pid namespaces.
>>>>
>>>> The main idea is that kernel keeps operating with tasks pid
>>>> as it did before, but each task obtains one more pid for each
>>>> pid type - the virtual pid. When putting the pid to user or
>>>> getting the pid from it kernel operates with the virtual ones.
>>>>
>>>> E.g. virtual pid is returned from getpid(), virtual pgid -
>>>> from getpgid() and so on. Getting virtual pid from user is
>>>> performed in setpgid(), setsid() and kill() mainly and in some
>>>> other places.
>>>>
>>>> As far as the namespace are concerned I propose the following
>>>> scheme. The namespace can be created from unshare syscall only.
>>>> This makes fork() code look easier. Of course task must be
>>> So is your main reason for posting this as a counter to Suka's patchset
>>> the concern of overhead at clone?
>> No, that's just a coincidence that I worked on the same problem.
>> What I propose is another way to make pid namespaces. It has its
>> advantages over Suka's approach. Main are:
>>
>> 1. Lighter exporting of pid to userspace and performance issues
>>   on the whole - as you have noticed at least fork() is
>>   lighter and many syscalls that return task pids are;
>> 2. Kernel logic of tracking pids is kept - virtual pids are
>>   used on kernel-user boundary only;
>
> On the other hand I've really learned to like the consistency of "there
> is always a single active pid ns for the task from which it sees all
> other tasks; it is seen in every pid ns for which it has a struct
> upid."
>
>> 3. Cleaner logic for namespace migration: with this approach
>>   one need to save the virtual pid and let global one change;
>>   with Suka's logic this is not clear how to migrate the level
>>   2 namespace (concerning init to be level 0).
>
> This is a very good point.
>
> How *would* we migrate the pids at the second level?

I would like to know how you migrate a *part* of a virtual
server? What happens with pids, IPC ids, network connections?

There are many entities in VS that are not bound to task, but to
VS and if you migrate only half of them you're risking in loosing
the integrity of the VS. If you don't care it - why do you need
namespaces at all?

> -serge
>
