
Subject: Re: [PATCH 1/2] signal checkpoint: define /proc/pid/sig/
Posted by [serue](#) on Tue, 12 Jun 2007 16:12:56 GMT
[View Forum Message](#) <> [Reply to Message](#)

Quoting Carl-Daniel Hailfinger (c-d.hailfinger.devel.2006@gmx.net):

> On 11.06.2007 19:05, Serge E. Hallyn wrote:
> > Quoting Cedric Le Goater (clg@fr.ibm.com):
> >
> >> should we continue to use /proc ? or switch to some other mechanisms
> >> like getnetlink (taskstats) to map kernel structures.
> >
> > We want to avoid 'map'ping kernel structures, though, right? We can
> > dump the data in a more generic fashion through netlink, dunno what we
> > prefer. But this is very definately process information :), so /proc
> > does seem appropriate.
>
> While I agree that /proc seems appropriate, I see a few benefits of
> dumping the data through netlink:

Good points, thanks.

> * Speed. IIRC there were benchmarks showing an advantage of netlink
> over /proc when communicating with userspace. Sorry, no idea where
> I read that.

I don't think we're dumping large amounts of data (the largest amounts, process memory, we're looking at doing just by forcing dump to swap), so I'm not sure how much it matters.

Still,

> * Versioning. While we strive to have the perfect interface on the
> first try, changes might be necessary. I see no way to handle
> multiple versions of an interface in /proc without big headaches.

Good point, this kind of offsets my major point against netlink, that we'd likely inherently end up versioning the interface by being tempted to dump kernel structures verbatim. I doubt anyone would claim that we'll never need to update the /proc interface, so that may make using /proc a nonstarter.

> * Conformity. With /proc, people often see a file, take a look at
> it and try to infer the structure of the file from what they see.
> This has led to multiple problems in the past when the content of
> some files in /proc changed slightly and tools broke. With
> netlink, implementers have to look at the spec to achieve anything
> useful.

Ok, so presumably we'd want some 'start a checkpoint' or 'start a restore' command (through syscall or whatever) to create the netlink socket and pass that to the various kernel dump/restore pieces?

Is there some better alternative people prefer to a syscall? If not, Daniel, would you mind adding that to the front of your patchset, and having your udp socket checkpoint/restore use that socket?

thanks,
-serge

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