

---

Subject: Re: [PATCH] pidns: Place under CONFIG\_EXPERIMENTAL (take 2)  
Posted by [ebiederm](#) on Mon, 29 Oct 2007 19:11:23 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

Cedric Le Goater <clg@fr.ibm.com> writes:

>> The outstanding issues I can think of off the top of my head:

>> - signal handling for init on secondary pid namespaces.

>> - Properly setting si\_pid on signals that cross namespaces.

>

> these are being addressed by suka patches, and also you with the latest patch

> you sent. right ?

I am just starting to review suka's patches it is a subtle area and tricky.

My signal related patches were aimed at just going through the global list of processes.

>> - The kthread API conversion so we don't get kernel threads

>> trapped in pid namespaces and make them unfreeable.

>

> a lot of work has been done on that part. take a look at it. the clean up

> is really impressive !

Agreed.

> NFS still uses the kernel\_thread() API. the first thing to do on the kthread

> topic is to improve the kthread API.

yes, getting the kthread API up to snuff works.

> I think we can discard the remaining drivers for the moment.

>

>> - At fork time I think we are doing a little bit too much work

>> in setting the session and the pgrp, and removing the controlling

>> tty.

>

> yes probably. this needs to be sorted out. it makes a container init

> process different from the system init process.

Yes. I sent a patch for review that fixes just this one aspect and

I will see what happens.

>> - AF\_unix domain credential passing.

>

> see commit b488893a390edfe027bae7a46e9af8083e740668 which is covering

> UNIX socket credentials and more. Are you thinking we should do more for

> credentials and use a struct pid\* ? This looks scary.

Yes. We need to get the pid in the pid namespace that remove the

credentials. Not the pid in the pid namespace that places the credentials on the socket.

As for scary and delicate I agree. We really need to pass the struct pid, not a pid\_t in the credentials.

```
>> - misc pid vs vpid sorting out (autofs autofs4, coda, arch specific
>>  syscalls, others?)
>
> autofs* is fixed. netlink ?
```

No. autofs compiles builds and works in a single pid namespace. The issues with multiple pid namespaces have not been fully addresses, and autofs4 is in worse shape. Unless there are pending patches I'm not aware of.

Partly it may be Pavel's shift from my intent of having pid\_nr be the pid\_t in current->nsproxy->pid\_namespace. To calling that function pid\_vnr, and having pid\_nr be the pid\_t value in init\_pid\_ns.

```
>> - Removal of task->pid, task->tgid, task->signal->__pgrp,
>>  tsk->signal->__session or some other way to ensure that we have
>>  touched and converted all of the kernel pid handling.
>
> well, __pgrp and __session are pretty well covered with the __deprecated
> attribute. I don't see what else we could do on these. we can't remove
> the task_{session,pgrp}__* routines.
>
> we could apply the same __deprecated technique to task->pid, task->tgid.
> This is going to be a challenge :)
```

Well the point is not to use the pid\_t for anything except passing to user space. It isn't that I object to people using \_\_session directly it is that I object to people using task\_session\_nr because it hides possible bugs. When we work with pids using struct pid pointers it is clear we can't get it wrong. When we use pid\_t values it is quite easy to get things wrong. As the current autofs code demonstrates.

```
> diff --git a/fs/autofs/inode.c b/fs/autofs/inode.c
> index 45f5992..af82143 100644
> --- a/fs/autofs/inode.c
> +++ b/fs/autofs/inode.c
> @@ -80,7 +80,7 @@ static int parse_options(char *options, int *pipefd, uid_t *uid, gid_t *gid,
>
>      *uid = current->uid;
>      *gid = current->gid;
> -      *pgrp = task_pgrp_nr(current);
```

```
> +    *pgrp = task_pgrp_vnr(current);  
>  
>    *minproto = *maxproto = AUTOFS_PROTO_VERSION;  
>
```

>> - flock pid handling.

>

> Pavel again.

Yes. I know he was looking at it.

> The kernel will be protected by a CONFIG\_NAMESPACES option as soon as it  
> gets in. Unfortunately, it didn't make 2.6.24 so this will be 2.6.25  
> material.

Which sounds completely bass akwards. We want the option so we can  
disable things now, when everything is immature, and not really  
doing the right thing. I guess I really should look in Andrews queue  
and see what didn't make it, and see if there are pieces that fixes  
bugs that really should have.

I think the CONFIG\_NAMESPACES work was focused primarily on size  
reduction under CONFIG\_EMBEDDED. Which means from a correctness  
perspective I don't care.

Eric

---

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

---