Subject: Re: + remove-the-likelypid-check-in-copy_process.patch added to -mm tree
Posted by ebiederm on Sat, 17 Mar 2007 14:04:16 GMT
View Forum Message <> Reply to Message

Oleg Nesterov <oleg@tv-sign.ru> writes:

> On 03/16, Eric W. Biederman wrote:
>>
>> Oleg Nesterov <oleg@tv-sign.ru> writes:
>>
>> > Sukadev Bhattiprolu wrote:
>> >
>> > This means that idle threads (except "swapper") are visible to
>> > for_each_process()
>> > and do_each_thread(). Looks dangerous and somewhat strange to me.
>> >
>> > Could you explain this change?
>>
>> Good catch. I've been so busy pounding reviewing this patches into
>> something that made sense that I missed the fact that we care about
>> this for more than just the NULL pointer that would occur if we didn't
>> do this.

Err. I meant NULL pointer dereference.

> Why it is bad to have a NULL pointer for idle thread? (Sorry for stupid
> question, I can't track the code changes these days).

>
>> Still it would be good if we could find a way to remove this rare
>> special case.
>>
>> Any chance we can undo what we don't want done for_idle, or create
>> a factor of copy_process that only does as much as fork_idle should do,
>> and make copy_process a wrapper that does the rest.
>>
>> I doubt it is significant anywhere but it would be nice to remove a
>> branch that except at boot up never happens.
>
> ... or at cpu-hotplug. Probably you are right, but I am not sure.
>
> The "if (p->pid)" check in essence implements CLONE_UNHASHED flag,
> it may be useful.
>
> Btw. Looking at http://marc.theaimsgroup.com/?l=linux-mm-commits,
>
>   Subject: Explicitly set pgid and sid of init process

> From: Sukadev Bhattiprolu <sukadev@us.ibm.com>
>
> Explicitly set pgid and sid of init process to 1.
>
> Signed-off-by: Sukadev Bhattiprolu <sukadev@us.ibm.com>
> Cc: Cedric Le Goater <clg@fr.ibm.com>
> Cc: Dave Hansen <haveblue@us.ibm.com>
> Cc: Serge Hallyn <serue@us.ibm.com>
> Cc: Eric Biederman <ebiederm@xmission.com>
> Cc: Herbert Poetzl <herbert@13thfloor.at>
> Cc: <containers@lists.osdl.org>
> Acked-by: Eric W. Biederman <ebiederm@xmission.com>
> Signed-off-by: Andrew Morton <akpm@linux-foundation.org>
> ---
>
> init/main.c | 1 +
> 1 file changed, 1 insertion(+)
>
> diff -puN init/main.c~explicitly-set-pgid-and-sid-of-init-process
> init/main.c
> --- a/init/main.c~explicitly-set-pgid-and-sid-of-init-process
> +++ a/init/main.c
> @@ -783,6 +783,7 @@ static int __init init(void * unused)
> */
> init_pid_ns.child_reaper = current;
>
> + __set_special_pids(1, 1);
> cad_pid = task_pid(current);
>
> smp_prepare_cpus(max_cpus);
>
> Nice changelog :)
>
> The patch looks good, except __set_special_pids(1, 1) should be no-op.
> This is a child forked by swapper. copy_process() was changed by
> use-task_pgrp-task_session-in-copy_process.patch
> , but signal->{pgrp,_session} get its value from INIT_SIGNALS ?
>
> Could you explain this as well? Some other changes I missed?

As I recall the patch series started with modifying attach_pid
to take a struct pid pointer instead of a pid_t value. It means
fewer hash table looks ups and it should help in implementing the pid
namespace.

Well the initial kernel process does not have a struct pid so when
it's children start doing:
attach_pid(p, PIDTYPE_PGID, task_group(p));

attach_pid(p, PIDTYPE_SID, task_session(p));
We will get an oops.

So a dummy unhashed struct pid was added for the idle threads.
Allowing several special cases in the code to be removed.

With that chance the previous special case to force the idle thread
init session 1 pgrp 1 no longer works because attach_pid no longer
looks at the pid value but instead at the struct pid pointers.

So we had to add the __set_special_pids() to continue to keep init
in session 1 pgrp 1. Since /sbin/init calls setsid() that our setting
the sid and the pgrp may not be strictly necessary. Still is better
to not take any chances.

Anyway the point of removing the likely(pid) check was that it didn't
look necessary any longer. But as you have correctly pointed putting
it on the task list and incrementing the process count for the idle
threads is probably still a problem. So while we are much better we
still have some use for the if (likely(p->pid)) special case.

Is that enough to bring you up to speed?

Eric

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