Subject: + remove-the-likelypid-check-in-copy_process.patch added to -mm tree
Posted by akpm on Thu, 15 Mar 2007 19:54:42 GMT
View Forum Message <> Reply to Message

The patch titled
    Remove the likely(pid) check in copy_process
has been added to the -mm tree. Its filename is
    remove-the-likelypid-check-in-copy_process.patch

*** Remember to use Documentation/SubmitChecklist when testing your code ***

See http://www.zip.com.au/~akpm/linux/patches/stuff/added-to-mm.txt to find
out what to do about this

Subject: Remove the likely(pid) check in copy_process
From: Sukadev Bhattiprolu <sukadev@us.ibm.com>

Now that we pass in a struct pid parameter to copy_process() and even the
swapper (pid_t == 0) has a valid struct pid, we no longer need this check.

Changelog:
 Per Eric Biederman's comments, moved this out to a separate
 patch for easier review.

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: <containers@lists.osdl.org>
Acked-by: Eric W. Biederman <ebiederm@xmission.com>

Signed-off-by: Andrew Morton <akpm@linux-foundation.org>
---

 kernel/fork.c | 34 ++++++++++++++++++----------------
 1 file changed, 16 insertions(+), 18 deletions(-)

diff -puN kernel/fork.c~remove-the-likelypid-check-in-copy_process kernel/fork.c
--- a/kernel/fork.c~remove-the-likelypid-check-in-copy_process
+++ a/kernel/fork.c
@@ -1237,26 +1237,24 @@ static struct task_struct *copy_process(
   }
  }

- if (likely(p->pid)) {
- add_parent(p);
- tracehook_init_task(p);

```
-
-  if (thread_group_leader(p)) {
-   pid_t pgid = process_group(current);
-   pid_t sid = process_session(current);
-
-   p->signal->tty = current->signal->tty;
-   p->signal->pgrp = pgid;
-   set_signal_session(p->signal, process_session(current));
-   attach_pid(p, PIDTYPE_PGID, find_pid(pgid));
-   attach_pid(p, PIDTYPE_SID, find_pid(sid));
+ add_parent(p);
+ tracehook_init_task(p);

-   list_add_tail_rcu(&p->tasks, &init_task.tasks);
-   __get_cpu_var(process_counts)++;
-  }
-  attach_pid(p, PIDTYPE_PID, pid);
-  nr_threads++;
+ if (thread_group_leader(p)) {
+  pid_t pgid = process_group(current);
+  pid_t sid = process_session(current);
+
+  p->signal->tty = current->signal->tty;
+  p->signal->pgrp = pgid;
+  set_signal_session(p->signal, process_session(current));
+  attach_pid(p, PIDTYPE_PGID, find_pid(pgid));
+  attach_pid(p, PIDTYPE_SID, find_pid(sid));
+
+  list_add_tail_rcu(&p->tasks, &init_task.tasks);
+  __get_cpu_var(process_counts)++;
   }
+ attach_pid(p, PIDTYPE_PID, pid);
+ nr_threads++;

  total_forks++;
  spin_unlock(&current->sighand->siglock);
_
```

Patches currently in -mm which might be from sukadev@us.ibm.com are

attach_pid-with-struct-pid-parameter.patch
statically-initialize-struct-pid-for-swapper.patch
explicitly-set-pgid-and-sid-of-init-process.patch
use-struct-pid-parameter-in-copy_process.patch
remove-the-likelypid-check-in-copy_process.patch
use-task_pgrp-task_session-in-copy_process.patch
kill-unused-sesssion-and-group-values-in-rocket-driver.patch
fix-some-coding-style-errors-in-autofs.patch

replace-pid_t-in-autofs-with-struct-pid-reference.patch

_____

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

Subject: Re: + remove-the-likelypid-check-in-copy_process.patch added to -mm tree
Posted by Oleg Nesterov on Fri, 16 Mar 2007 17:02:55 GMT
View Forum Message <> Reply to Message

Sukadev Bhattiprolu wrote:

> @@ -1237,26 +1237,24 @@ static struct task_struct *copy_process(
> }
> }
>
> - if (likely(p->pid)) {
> - add_parent(p);
> - tracehook_init_task(p);
> -
> - if (thread_group_leader(p)) {
> - pid_t pgid = process_group(current);
> - pid_t sid = process_session(current);
> -
> - p->signal->tty = current->signal->tty;
> - p->signal->pgrp = pgid;
> - set_signal_session(p->signal, process_session(current));
> - attach_pid(p, PIDTYPE_PGID, find_pid(pgid));
> - attach_pid(p, PIDTYPE_SID, find_pid(sid));
> + add_parent(p);

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?

Oleg.

_____

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

Subject: Re: + remove-the-likelypid-check-in-copy_process.patch added to -mm tree
Posted by ebiederm on Fri, 16 Mar 2007 18:27:15 GMT
View Forum Message <> Reply to Message

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.

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.

Eric

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

Subject: Re: + remove-the-likelypid-check-in-copy_process.patch added to -mm tree
Posted by Oleg Nesterov on Sat, 17 Mar 2007 13:02:17 GMT
View Forum Message <> Reply to Message

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.

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?

Oleg.

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

---

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

---

Subject: Re: + remove-the-likelypid-check-in-copy_process.patch added to -mm tree
Posted by Oleg Nesterov on Sat, 17 Mar 2007 15:09:49 GMT
View Forum Message <> Reply to Message

On 03/17, Eric W. Biederman wrote:
>
> Oleg Nesterov <oleg@tv-sign.ru> writes:
>
> >   --- 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 far this is the only reason to have init_struct_pid. Because the
boot CPU (swapper) forks, right?

> 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.

Yes, yes, I see. But my (very unclear, sorry) question was: shouldn't we
change INIT_SIGNALS then? /sbin/init inherits ->pgrp == ->_session == 1,
in that case __set_special_pids(1,1) does nothing.

> 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.

Yes. Note also that the parent doing fork_idle() is not always swapper,
it is just wrong to do attach_pid(PIDTYPE_PGID/PIDTYPE_SID) in this case.
example: arch/x86_64/kernel/smpboot.c:do_boot_cpu()

>   So while we are much better we
> still have some use for the if (likely(p->pid)) special case.

Yes, I think this change should be dropped for now.

> Is that enough to bring you up to speed?

Thanks for your explanations!

Oleg.

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

Subject: Re: + remove-the-likelypid-check-in-copy_process.patch added to -mm tree
Posted by Oleg Nesterov on Sat, 17 Mar 2007 15:24:37 GMT
View Forum Message <> Reply to Message

On 03/17, Oleg Nesterov wrote:
>
> > 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 far this is the only reason to have init_struct_pid. Because the
> boot CPU (swapper) forks, right?

Damn. I am afraid I was not clear again :) Not init_struct_pid, but

```
+       .pids = {                                               \
+               [PIDTYPE_PID]  = INIT_PID_LINK(PIDTYPE_PID),            \
+               [PIDTYPE_PGID] = INIT_PID_LINK(PIDTYPE_PGID),           \
+               [PIDTYPE_SID]  = INIT_PID_LINK(PIDTYPE_SID),            \
+       },                                                      \
```

for INIT_TASK().

> > 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.
>
> Yes, yes, I see. But my (very unclear, sorry) question was: shouldn't we
> change INIT_SIGNALS then? /sbin/init inherits ->pgrp == ->_session == 1,
> in that case __set_special_pids(1,1) does nothing.

... and thus /sbin/init remains attached to the .pids above, no?

Oleg.

_____
Containers mailing list
Containers@lists.linux-foundation.org

Subject: Re: + remove-the-likelypid-check-in-copy_process.patch added to -mm tree
Posted by ebiederm on Sat, 17 Mar 2007 17:01:50 GMT
View Forum Message <> Reply to Message

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

> On 03/17, Oleg Nesterov wrote:
>>
>> > 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 far this is the only reason to have init_struct_pid. Because the
>> boot CPU (swapper) forks, right?
>
> Damn. I am afraid I was not clear again :) Not init_struct_pid, but
>
> + .pids = { \
> + [PIDTYPE_PID] = INIT_PID_LINK(PIDTYPE_PID), \
> + [PIDTYPE_PGID] = INIT_PID_LINK(PIDTYPE_PGID), \
> + [PIDTYPE_SID] = INIT_PID_LINK(PIDTYPE_SID), \
> + }, \
>
> for INIT_TASK().
>
>> > 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.
>>
>> Yes, yes, I see. But my (very unclear, sorry) question was: shouldn't we
>> change INIT_SIGNALS then? /sbin/init inherits ->pgrp == ->_session == 1,
>> in that case __set_special_pids(1,1) does nothing.
>
> ... and thus /sbin/init remains attached to the .pids above, no?

The problem is that we dynamically allocate the struct pid for
pid_t == 1 when we fork init.

Which means we don't have access to it at compile time so we can
no longer make INIT_SIGNALS set ->gprp == ->session == 1.

Eric

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

---

Subject: Re: + remove-the-likelypid-check-in-copy_process.patch added to -mm tree
Posted by Oleg Nesterov on Sat, 17 Mar 2007 17:17:47 GMT
View Forum Message <> Reply to Message

On 03/17, Eric W. Biederman wrote:
>
> Oleg Nesterov <oleg@tv-sign.ru> writes:
>
> > On 03/17, Oleg Nesterov wrote:
> >>
> >> > 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 far this is the only reason to have init_struct_pid. Because the
> >> boot CPU (swapper) forks, right?
> >
> > Damn. I am afraid I was not clear again :) Not init_struct_pid, but
> >
> > + .pids = { \
> > + [PIDTYPE_PID] = INIT_PID_LINK(PIDTYPE_PID), \
> > + [PIDTYPE_PGID] = INIT_PID_LINK(PIDTYPE_PGID), \
> > + [PIDTYPE_SID] = INIT_PID_LINK(PIDTYPE_SID), \
> > + }, \
> >
> > for INIT_TASK().
> >
> >> > 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.
> >>
> >> Yes, yes, I see. But my (very unclear, sorry) question was: shouldn't we
> >> change INIT_SIGNALS then? /sbin/init inherits ->pgrp == ->_session == 1,
> >> in that case __set_special_pids(1,1) does nothing.
> >
> > ... and thus /sbin/init remains attached to the .pids above, no?
>
> The problem is that we dynamically allocate the struct pid for
> pid_t == 1 when we fork init.
>
> Which means we don't have access to it at compile time so we can
> no longer make INIT_SIGNALS set ->gprp == ->session == 1.

Yes! I meant we should change INIT_SIGNALS(), currently it does

```
 #define INIT_SIGNALS(sig) {
  ...
  .pgrp          = 1,
  { .__session      = 1},
```

and this confuses (I think) set_special_pids(1,1) above. Because
__set_special_pids() still deals with pid_t, not "struct pid".

Unless I missed something, we should kill these 2 initializations
above.

Oleg.

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

---

Subject: Re: + remove-the-likelypid-check-in-copy_process.patch added to -mm tree
Posted by ebiederm on Sat, 17 Mar 2007 18:54:11 GMT
View Forum Message <> Reply to Message

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

> Yes! I meant we should change INIT_SIGNALS(), currently it does
>
> #define INIT_SIGNALS(sig) {
>   ...
>   .pgrp   = 1,
>   { .__session   = 1},
>
> and this confuses (I think) set_special_pids(1,1) above. Because
> __set_special_pids() still deals with pid_t, not "struct pid".
>
> Unless I missed something, we should kill these 2 initializations
> above.

Got it. I agree we should initialize those fields to 0.

Sukadev you want to get that?

Eric

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

---

Subject: Re: + remove-the-likelypid-check-in-copy_process.patch added to -mm tree
Posted by Sukadev Bhattiprolu on Sun, 18 Mar 2007 06:50:36 GMT
View Forum Message <> Reply to Message

Eric W. Biederman [ebiederm@xmission.com] wrote:
| Oleg Nesterov <oleg@tv-sign.ru> writes:
|
| > Yes! I meant we should change INIT_SIGNALS(), currently it does
| >
| > #define INIT_SIGNALS(sig) {
| >   ...
| >   .pgrp   = 1,
| >   { .__session   = 1},
| >
| > and this confuses (I think) set_special_pids(1,1) above. Because
| > __set_special_pids() still deals with pid_t, not "struct pid".
| >
| > Unless I missed something, we should kill these 2 initializations
| > above.
|
| Got it. I agree we should initialize those fields to 0.
|
| Sukadev you want to get that?

Sure. Will do that.

Thanks Oleg for your detailed review/comments.

Suka

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