Subject: [PATCH] namespaces: fix race at task exit
Posted by serue on Thu, 25 Jan 2007 15:05:42 GMT
View Forum Message <> Reply to Message

In do_exit(), the exit_task_namespaces() was placed after
exit_notify() because exit_notify ends up using the pid
namespace both to access the reaper, and for detaching the
pid. However, this placement allows an nfs server to reap
the task before exit_task_namespaces() completes.

This patch moves the exit_task_namespaces() into release_task,
below release_thread() which puts the pids(), and just above
the call_rcu(delayed_put_task_struct). I believe this should
solve both problems.

Signed-off-by: Serge E. Hallyn <serue@us.ibm.com>

---

kernel/exit.c | 2 +-
1 files changed, 1 insertions(+), 1 deletions(-)

765277a4170d7bbd1c4613de661ec6ac64d5580a
diff --git a/kernel/exit.c b/kernel/exit.c
index 3540172..ab9ae30 100644
--- a/kernel/exit.c
+++ b/kernel/exit.c
@@ -174,6 +174,7 @@ repeat:
write_unlock_irq(&tasklist_lock);
proc_flush_task(p);
release_thread(p);
+ exit_task_namespaces(p);
call_rcu(&p->rcu, delayed_put_task_struct);

p = leader;
@@ -939,7 +940,6 @@ fastcall NORET_TYPE void do_exit(long co
tsk->exit_code = code;
proc_exit_connector(tsk);
exit_notify(tsk);
- exit_task_namespaces(tsk);
#ifdef CONFIG_NUMA
mpol_free(tsk->mempolicy);
tsk->mempolicy = NULL;
--
1.1.6

_____
Containers mailing list
Containers@lists.osdl.org

Subject: Re: [PATCH] namespaces: fix race at task exit
Posted by Cedric Le Goater on Thu, 25 Jan 2007 15:20:23 GMT
View Forum Message <> Reply to Message

Serge E. Hallyn wrote:
> In do_exit(), the exit_task_namespaces() was placed after
> exit_notify() because exit_notify ends up using the pid
> namespace both to access the reaper, and for detaching the
> pid. However, this placement allows an nfs server to reap
> the task before exit_task_namespaces() completes.
>
> This patch moves the exit_task_namespaces() into release_task,
> below release_thread() which puts the pids(), and just above
> the call_rcu(delayed_put_task_struct). I believe this should
> solve both problems.
>
> Signed-off-by: Serge E. Hallyn <serue@us.ibm.com>

I've run some tests on x86 and x86_64: mounted a NFS share after
having unshare(CLONE_NEWNS) and I didn't reproduce the bug Daniel
had found.

it looks safe.

C.

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

Subject: Re: [PATCH] namespaces: fix race at task exit
Posted by ebiederm on Thu, 25 Jan 2007 16:29:13 GMT
View Forum Message <> Reply to Message

"Serge E. Hallyn" <serue@us.ibm.com> writes:

> In do_exit(), the exit_task_namespaces() was placed after
> exit_notify() because exit_notify ends up using the pid
> namespace both to access the reaper, and for detaching the
> pid. However, this placement allows an nfs server to reap
> the task before exit_task_namespaces() completes.

>
> This patch moves the exit_task_namespaces() into release_task,
> below release_thread() which puts the pids(), and just above
> the call_rcu(delayed_put_task_struct). I believe this should
> solve both problems.

For the pid namespace this seems to be correct placement.
For the mount namespace this would seem to exacerbate the problem
because it now gets called after the task has been reaped!

I'd love to be convinced otherwise but I do not believe we
can safely exit both the mount and the pid namespace at the
same location in the code.

The NFS unmount currently wants a killable thread as it
uses interruptible sleeps. How does starting that process
after the process in which it lives aid this?

But thanks for remembering this. This is a real problem we
do need to solve.

Eric

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

Subject: Re: [PATCH] namespaces: fix race at task exit
Posted by Oleg Nesterov on Thu, 25 Jan 2007 16:39:36 GMT
View Forum Message <> Reply to Message

On 01/25, Serge E. Hallyn wrote:
>
> In do_exit(), the exit_task_namespaces() was placed after
> exit_notify() because exit_notify ends up using the pid
> namespace both to access the reaper, and for detaching the
> pid. However, this placement allows an nfs server to reap
> the task before exit_task_namespaces() completes.
>
> This patch moves the exit_task_namespaces() into release_task,
> below release_thread() which puts the pids(), and just above
> the call_rcu(delayed_put_task_struct). I believe this should
> solve both problems.
>
> Signed-off-by: Serge E. Hallyn <serue@us.ibm.com>
>

> ---
>
> kernel/exit.c | 2 +-
> 1 files changed, 1 insertions(+), 1 deletions(-)
>
> 765277a4170d7bbd1c4613de661ec6ac64d5580a
> diff --git a/kernel/exit.c b/kernel/exit.c
> index 3540172..ab9ae30 100644
> --- a/kernel/exit.c
> +++ b/kernel/exit.c
> @@ -174,6 +174,7 @@ repeat:
>   write_unlock_irq(&tasklist_lock);
>   proc_flush_task(p);
>   release_thread(p);
> + exit_task_namespaces(p);
>   call_rcu(&p->rcu, delayed_put_task_struct);

Probably I missed some other patches in this area, but I can't understand
this fix.

With this change we are doing __put_mnt_ns() when we surely don't have ->sighand,
no? Could you please explain?

Oleg.

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

Subject: Re: [PATCH] namespaces: fix race at task exit
Posted by serue on Thu, 25 Jan 2007 17:35:56 GMT
View Forum Message <> Reply to Message

Quoting Eric W. Biederman (ebiederm@xmission.com):
> "Serge E. Hallyn" <serue@us.ibm.com> writes:
>
> > In do_exit(), the exit_task_namespaces() was placed after
> > exit_notify() because exit_notify ends up using the pid
> > namespace both to access the reaper, and for detaching the
> > pid. However, this placement allows an nfs server to reap
> > the task before exit_task_namespaces() completes.
> >
> > This patch moves the exit_task_namespaces() into release_task,
> > below release_thread() which puts the pids(), and just above
> > the call_rcu(delayed_put_task_struct). I believe this should
> > solve both problems.

>
>
> For the pid namespace this seems to be correct placement.
> For the mount namespace this would seem to exacerbate the problem
> because it now gets called after the task has been reaped!
>
> I'd love to be convinced otherwise but I do not believe we
> can safely exit both the mount and the pid namespace at the
> same location in the code.
>
> The NFS unmount currently wants a killable thread as it
> uses interruptible sleeps. How does starting that process
> after the process in which it lives aid this?

I should have mentioned I'm unable to reproduce the original
oops myself, so i wanted confirmation about whether this fixed
the problem.

I had thought the mount problem was that the nfs server causes
the task_struct to be freed before exit_task_namespaces() completes,
so that exit_task_namespaces() dereferences a bad pointer. If
that were the case, this would fix it by not putting the final
reference to the task_struct (with delayed_put_task_struct())
until after exit_task_namespaces(). It sounds like I misunderstood
the nfs server problem though.

> But thanks for remembering this. This is a real problem we
> do need to solve.

If it is confirmed that my patch is wrong, then I guess we simply
need a two-stage namespace exit, where the first stage happens
above exit_notify() and exits the mounts namespace, and the second
stage can happen in the location I used in this patch.

-serge

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

Subject: Re: [PATCH] namespaces: fix race at task exit
Posted by serue on Thu, 25 Jan 2007 17:36:55 GMT
View Forum Message <> Reply to Message

Quoting Oleg Nesterov (oleg@tv-sign.ru):
> On 01/25, Serge E. Hallyn wrote:
> >

> > In do_exit(), the exit_task_namespaces() was placed after
> > exit_notify() because exit_notify ends up using the pid
> > namespace both to access the reaper, and for detaching the
> > pid. However, this placement allows an nfs server to reap
> > the task before exit_task_namespaces() completes.
> >
> > This patch moves the exit_task_namespaces() into release_task,
> > below release_thread() which puts the pids(), and just above
> > the call_rcu(delayed_put_task_struct). I believe this should
> > solve both problems.
> >
> > Signed-off-by: Serge E. Hallyn <serue@us.ibm.com>
> >
> > ---
> >
> > kernel/exit.c | 2 +-
> > 1 files changed, 1 insertions(+), 1 deletions(-)
> >
> > 765277a4170d7bbd1c4613de661ec6ac64d5580a
> > diff --git a/kernel/exit.c b/kernel/exit.c
> > index 3540172..ab9ae30 100644
> > --- a/kernel/exit.c
> > +++ b/kernel/exit.c
> > @@ -174,6 +174,7 @@ repeat:
> >   write_unlock_irq(&tasklist_lock);
> >   proc_flush_task(p);
> >   release_thread(p);
> > + exit_task_namespaces(p);
> >   call_rcu(&p->rcu, delayed_put_task_struct);
>
> Probably I missed some other patches in this area, but I can't understand
> this fix.
>
> With this change we are doing __put_mnt_ns() when we surely don't have ->sighand,
> no? Could you please explain?

Explanation: it's wrong :)

we'll just need to break exit_task_namespaces() up.

thanks,
-serge

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

Subject: Re: [PATCH] namespaces: fix race at task exit
Posted by serue on Thu, 25 Jan 2007 20:36:44 GMT
View Forum Message <> Reply to Message

Quoting Serge E. Hallyn (serue@us.ibm.com):
> Quoting Eric W. Biederman (ebiederm@xmission.com):
> > "Serge E. Hallyn" <serue@us.ibm.com> writes:
> >
> > > In do_exit(), the exit_task_namespaces() was placed after
> > > exit_notify() because exit_notify ends up using the pid
> > > namespace both to access the reaper, and for detaching the
> > > pid. However, this placement allows an nfs server to reap
> > > the task before exit_task_namespaces() completes.
> > >
> > > This patch moves the exit_task_namespaces() into release_task,
> > > below release_thread() which puts the pids(), and just above
> > > the call_rcu(delayed_put_task_struct). I believe this should
> > > solve both problems.
> >
> >
> > For the pid namespace this seems to be correct placement.
> > For the mount namespace this would seem to exacerbate the problem
> > because it now gets called after the task has been reaped!
> >
> > I'd love to be convinced otherwise but I do not believe we
> > can safely exit both the mount and the pid namespace at the
> > same location in the code.
> >
> > The NFS unmount currently wants a killable thread as it
> > uses interruptible sleeps. How does starting that process
> > after the process in which it lives aid this?
>
> I should have mentioned I'm unable to reproduce the original
> oops myself, so i wanted confirmation about whether this fixed
> the problem.
>
> I had thought the mount problem was that the nfs server causes
> the task_struct to be freed before exit_task_namespaces() completes,
> so that exit_task_namespaces() dereferences a bad pointer. If
> that were the case, this would fix it by not putting the final
> reference to the task_struct (with delayed_put_task_struct())
> until after exit_task_namespaces(). It sounds like I misunderstood
> the nfs server problem though.
>
> > But thanks for remembering this. This is a real problem we
> > do need to solve.
>
> If it is confirmed that my patch is wrong, then I guess we simply
> need a two-stage namespace exit, where the first stage happens

> above exit_notify() and exits the mounts namespace, and the second
> stage can happen in the location I used in this patch.

Of course the problem with this is that the mounts and proc
namespaces now have slightly different lifetimes, and we cannot
use one use count to track both because it's quite possible
that the two last tasks in a namespace could both come to the
release_mounts_namespaces() point at the same time, then both
come to the exit_tasks_namespaces().

So it seems to me we need to either pull one of the two out of
the nsproxy, or add a second use count to the nsproxy. The
second use count looks kludgier, but uses less space and seems
safer to maintain because at least the lifetime management happens
somewhat close to each other, whereas moving moutns namespace back
outside of nsproxy means going back to a completely differnet meaning
of mnt_ns->count.

Opinions, or other ideas?

thanks,
-serge

_____

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