Subject: Re: [RFC][PATCH] Add child reaper to struct pspace
Posted by dev on Fri, 08 Sep 2006 17:32:27 GMT
View Forum Message <> Reply to Message

Sukadev Bhattiprolu wrote:
> Cedric Le Goater [clg@fr.ibm.com] wrote:
> |
> | <snip>
> |
> | > */
> | > static void
> | > forget_original_parent(struct task_struct *father, struct list_head *to_release)
> | > @@ -669,7 +670,7 @@ forget_original_parent(struct task_struc
> | > do {
> | > reaper = next_thread(reaper);
> | > if (reaper == father) {
> | > - reaper = child_reaper;
> | > + reaper = father->pspace->child_reaper;
> | > break;
> | > }
> | > } while (reaper->exit_state);
> | > @@ -857,7 +858,7 @@ fastcall NORET_TYPE void do_exit(long co
> |
> | what about killing all the task in that pid space if child_reaper == init
> | dies ?
> |
>
> We probably need that for instance when a process in the parent pspace
> kills the init of a child pspace, we should destroy the child pspace
> by killing all the tasks in the child pspace including the child reaper.
exactly. the situation you described is how we do handle it.
you can check do_initproc_exit() function in OpenVZ
to check how it can be done and probably save some of your time.
(http://git.openvz.org/?p=linux-2.6-openvz;a=summary)

> I guess we need to maintain a list of task_structs in the pspace and walk
> that list. Will work on that as a separate patch.
wait. we either need to have a list of _pids_ or it
should be called task_namespace, not pid, since we are adding more
code related to tasks.

Kirill

Subject: Re: [RFC][PATCH] Add child reaper to struct pspace
Posted by ebiederm on Sat, 09 Sep 2006 04:54:56 GMT
View Forum Message <> Reply to Message

Kirill Korotaev <dev@sw.ru> writes:

>> I guess we need to maintain a list of task_structs in the pspace and walk
>> that list. Will work on that as a separate patch.
> wait. we either need to have a list of _pids_ or it
> should be called task_namespace, not pid, since we are adding more
> code related to tasks.

There will be a way to iterate through all of the pids.

It will probably be through a linked list of struct pid, but
it may be an in order traversal of some pid related data structure.

I was hoping for a moment I might be able to only implement one
struct pid, but I need some method to perform a reverse lookup
from struct pid to a (struct pid_namespace, pid_t) pair to
properly implement pid_nr(). A linked list of struct pid entries
is the obvious implementation.

Eric

Subject: Re: [RFC][PATCH] Add child reaper to struct pspace
Posted by dev on Tue, 12 Sep 2006 14:41:45 GMT
View Forum Message <> Reply to Message

Eric W. Biederman wrote:
> Kirill Korotaev <dev@sw.ru> writes:
>
>
>>>I guess we need to maintain a list of task_structs in the pspace and walk
>>>that list. Will work on that as a separate patch.
>>
>>wait. we either need to have a list of _pids_ or it
>>should be called task_namespace, not pid, since we are adding more
>>code related to tasks.
>
>
> There will be a way to iterate through all of the pids.
>
> It will probably be through a linked list of struct pid, but
> it may be an in order traversal of some pid related data structure.
>
> I was hoping for a moment I might be able to only implement one
> struct pid, but I need some method to perform a reverse lookup
> from struct pid to a (struct pid_namespace, pid_t) pair to
> properly implement pid_nr(). A linked list of struct pid entries
> is the obvious implementation.

I guess there will be a need of list of tasks... not of pids only...
many of loops like do_each_thread()/while_each_thread() has nothing to do with pids
and should be narrowed down to loop through the container.

Does this logic belong to pid_ns? if yes, then it definetely should be called
task_ns.

Kirill

Subject: Re: [RFC][PATCH] Add child reaper to struct pspace
Posted by ebiederm on Tue, 12 Sep 2006 15:43:19 GMT
View Forum Message <> Reply to Message

Kirill Korotaev <dev@sw.ru> writes:

> I guess there will be a need of list of tasks... not of pids only...
> many of loops like do_each_thread()/while_each_thread() has nothing to do with
> pids
> and should be narrowed down to loop through the container.
>
> Does this logic belong to pid_ns? if yes, then it definetely should be called
> task_ns.

Just skimming through I see one or two instances. Where the existing
loop uses do_each_thread()/while_each_thread() that we need to change.

kernel/capabilities.c cap_set_all() is an example.

However what we are trying to achieve there is to iterate through
the same list that kill(-1,) uses. So we need to replace
do_each_thread()/while_each_thread() with something that will
iterate through everything in the pid namespace.

Most instances of do_each_thread()/while_each_thread() the kernel
really does need a global view, and need to be left unchanged.

Basically the current kernel is short the concept of a process
group of all processes, and uses the concept of a list of all processes
instead.

Since the two concepts of a list of all tasks, and a list of all processes
I can see diverge when we have multiple pid namespaces we need to add
an additional concept, in the kernel.

Do you know an example in that we need to change to implement a pid
namespace that goes beyond iterating through the list of processes

that kill(-1,) uses?

Eric

Subject: Re: [RFC][PATCH] Add child reaper to struct pspace
Posted by dev on Sat, 16 Sep 2006 11:55:04 GMT
View Forum Message <> Reply to Message

Eric W. Biederman wrote:
> Kirill Korotaev <dev@sw.ru> writes:
>
>
>
>>I guess there will be a need of list of tasks... not of pids only...
>>many of loops like do_each_thread()/while_each_thread() has nothing to do with
>>pids
>>and should be narrowed down to loop through the container.
>>
>>Does this logic belong to pid_ns? if yes, then it definetely should be called
>>task_ns.
>
>
> Just skimming through I see one or two instances. Where the existing
> loop uses do_each_thread()/while_each_thread() that we need to change.
>
> kernel/capabilities.c cap_set_all() is an example.
>
> However what we are trying to achieve there is to iterate through
> the same list that kill(-1,) uses. So we need to replace
> do_each_thread()/while_each_thread() with something that will
> iterate through everything in the pid namespace.
>
> Most instances of do_each_thread()/while_each_thread() the kernel
> really does need a global view, and need to be left unchanged.
>
> Basically the current kernel is short the concept of a process
> group of all processes, and uses the concept of a list of all processes
> instead.
>
> Since the two concepts of a list of all tasks, and a list of all processes
> I can see diverge when we have multiple pid namespaces we need to add
> an additional concept, in the kernel.
>
> Do you know an example in that we need to change to implement a pid
> namespace that goes beyond iterating through the list of processes
> that kill(-1,) uses?

from OVZ patches:

do_each_thread_ve()
  elf_core_dump() (need pid namespace list?)
  zap_threads (need pid namespace list?)
  chroot_fs_refs
  cap_set_all (need pid namespace list?)
  cpt functions (need to freeze VE processes, pid namespace list?)
  sys_setpriority (needs task list for user namespace!)
  sys_getpriority (the same)
  sys_ioprio_set (the same)
  sys_ioprio_get (the same)
  selinux_setprocattr (should be changed with the check for thread_group_empty()???)

for_each_process_ve()
  asids_proc_info (need pid namespace list? in host should print all?)
  kill_something_info (I suppose you changed it already?)

some of these are optimizations which are natural for containers and are good
for scalability (as zap_threads, elf_core_dump etc.).

Thanks,
Kirill
P.S. Sorry for not always replying quickly...
