Subject: Re: [PATCH] namespaces: update some function names
Posted by dev on Wed, 21 Feb 2007 17:21:10 GMT
View Forum Message <> Reply to Message

Acked-By: Kirill Korotaev <dev@openvz.org>

> From: Serge E. Hallyn <serue@us.ibm.com>
> Subject: [PATCH] namespaces: update some function names
>
> The {get,exit}_task_namespaces do not grab references to the individual
> namespaces, only to the nsproxy. Reflect that in the function names.
>
> Not so important right now, but when pid_ns gets pulled out of nsproxy
> (eventually into struct pid_nr) the current naming is wrong.
>
> Signed-off-by: Serge E. Hallyn <serue@us.ibm.com>
>
> ---
>
> include/linux/nsproxy.h | 4 ++--
> kernel/exit.c | 6 +++---
> kernel/fork.c | 2 +-
> kernel/nsproxy.c | 2 +-
> 4 files changed, 7 insertions(+), 7 deletions(-)
>
> 3a423064ce33e801d4e63f6b4198958cdaf68e19
> diff --git a/include/linux/nsproxy.h b/include/linux/nsproxy.h
> index 0b9f0dc..b9d9aae 100644
> --- a/include/linux/nsproxy.h
> +++ b/include/linux/nsproxy.h
> @@ -33,7 +33,7 @@ extern struct nsproxy init_nsproxy;
>
> struct nsproxy *dup_namespaces(struct nsproxy *orig);
> int copy_namespaces(int flags, struct task_struct *tsk);
> -void get_task_namespaces(struct task_struct *tsk);
> +void get_task_nsproxy(struct task_struct *tsk);
> void free_nsproxy(struct nsproxy *ns);
>
> static inline void put_nsproxy(struct nsproxy *ns)
> @@ -43,7 +43,7 @@ static inline void put_nsproxy(struct ns
>   }
> }
>
> -static inline void exit_task_namespaces(struct task_struct *p)
> +static inline void put_task_nsproxy(struct task_struct *p)
> {
>   struct nsproxy *ns = p->nsproxy;
>   if (ns) {

```
> diff --git a/kernel/exit.c b/kernel/exit.c
> index bc71fdf..98f08cf 100644
> --- a/kernel/exit.c
> +++ b/kernel/exit.c
> @@ -395,9 +395,9 @@ void daemonize(const char *name, ...)
>   current->fs = fs;
>   atomic_inc(&fs->count);
>
> - exit_task_namespaces(current);
> + put_task_nsproxy(current);
>   current->nsproxy = init_task.nsproxy;
> - get_task_namespaces(current);
> + get_task_nsproxy(current);
>
>    exit_files(current);
>   current->files = init_task.files;
> @@ -937,7 +937,7 @@ fastcall NORET_TYPE void do_exit(long co
>
>   tsk->exit_code = code;
>   proc_exit_connector(tsk);
> - exit_task_namespaces(tsk);
> + put_task_nsproxy(tsk);
>   exit_notify(tsk);
>  #ifdef CONFIG_NUMA
>   mpol_free(tsk->mempolicy);
> diff --git a/kernel/fork.c b/kernel/fork.c
> index 80284eb..b8b76e5 100644
> --- a/kernel/fork.c
> +++ b/kernel/fork.c
> @@ -1267,7 +1267,7 @@ static struct task_struct *copy_process(
>   return p;
>
>  bad_fork_cleanup_namespaces:
> - exit_task_namespaces(p);
> + put_task_nsproxy(p);
>  bad_fork_cleanup_keys:
>   exit_keys(p);
>  bad_fork_cleanup_mm:
> diff --git a/kernel/nsproxy.c b/kernel/nsproxy.c
> index f5b9ee6..31fa63a 100644
> --- a/kernel/nsproxy.c
> +++ b/kernel/nsproxy.c
> @@ -28,7 +28,7 @@ static inline void get_nsproxy(struct ns
>   atomic_inc(&ns->count);
>  }
>
> -void get_task_namespaces(struct task_struct *tsk)
> +void get_task_nsproxy(struct task_struct *tsk)
```

> {
> struct nsproxy *ns = tsk->nsproxy;
> if (ns) {

_____

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