Subject: Re: - merge-sys_clone-sys_unshare-nsproxy-and-namespace.patch removed from -mm tree
Posted by Cedric Le Goater on Mon, 18 Jun 2007 12:25:28 GMT
View Forum Message <> Reply to Message

Oleg Nesterov wrote:
> On 06/18, Cedric Le Goater wrote:
>> Oleg Nesterov wrote:
>>> On 06/17, Oleg Nesterov wrote:
>>>> Let's look at copy_namespaces(), it does the same "get_xxx() in advance", but
>>>> -EPERM forgets to do put_nsproxy(), so we definitely have a leak in copy_process().
>>> Ugh, I am sorry, EPERM does put_nsproxy(). Still I can't understand why
>>> copy_namespaces() does get_nsproxy() unconditionally.
>> well, if you're cloning a new task and not unsharing some of the namespaces
>> you still want to increase the refcount on the nsproxy bc a new task is now
>> referencing it. nop ?
>
> Yes, but copy_namespaces() does get_nsproxy() unconditionally, and then it does
> put_nsproxy() if not unsharing, which is not good imho.
>
> IOW, I think the patch below makes the code a bit better. copy_namespaces()
> doesn't need put_nsproxy() at all.

Indeed it does and also :

nsproxy.c | 23 ++++++++----------------
 1 file changed, 8 insertions(+), 15 deletions(-)

Thanks,

C.

> Oleg.
>
> --- t/kernel/nsproxy.c~ 2007-06-18 16:10:53.000000000 +0400
> +++ t/kernel/nsproxy.c 2007-06-18 16:13:02.000000000 +0400
> @@ -103,31 +103,24 @@ int copy_namespaces(int flags, struct ta
> {
> struct nsproxy *old_ns = tsk->nsproxy;
> struct nsproxy *new_ns;
> - int err = 0;
>
> if (!old_ns)
> return 0;
>
> - get_nsproxy(old_ns);
> -
> - if (!(flags & (CLONE_NEWNS | CLONE_NEWUTS | CLONE_NEWIPC)))

```
> + if (!(flags & (CLONE_NEWNS | CLONE_NEWUTS | CLONE_NEWIPC))) {
> +  get_nsproxy(old_ns);
>     return 0;
> -
> - if (!capable(CAP_SYS_ADMIN)) {
> -  err = -EPERM;
> -  goto out;
>    }
>
> + if (!capable(CAP_SYS_ADMIN))
> +  return -EPERM;
> +
>    new_ns = create_new_namespaces(flags, tsk, tsk->fs);
> - if (IS_ERR(new_ns)) {
> -  err = PTR_ERR(new_ns);
> -  goto out;
> - }
> + if (IS_ERR(new_ns))
> +  return PTR_ERR(new_ns);
>
>    tsk->nsproxy = new_ns;
> -out:
> - put_nsproxy(old_ns);
> - return err;
> + return 0;
>  }
>
>  void free_nsproxy(struct nsproxy *ns)
>
>
```

_____

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