
Subject: Re: design of user namespaces
Posted by [serue](#) on Sat, 21 Jun 2008 19:05:32 GMT
[View Forum Message](#) <> [Reply to Message](#)

Quoting Eric W. Biederman (ebiederm@xmission.com):

> "Serge E. Hallyn" <serue@us.ibm.com> writes:
>
> > Quoting Serge E. Hallyn (serue@us.ibm.com):
> >> > Just skimming through your patch I don't expect we will need the list
> >> > of children, and not having should reduce our locking burden.
> >>
> >> Hmm, that's true. I can't see a reason for that. Thanks!
> >
> > BTW here is the new, slightly smaller patch:
> >
> >> From d17fbd87d97f64a0e879a7efbe5e1835fc573eae Mon Sep 17 00:00:00 2001
> > From: Serge Hallyn <serge@us.ibm.com>
> > Date: Thu, 19 Jun 2008 20:18:17 -0500
> > Subject: [PATCH 1/1] user namespaces: introduce user_struct->user_namespace
> > relationship
> >
> > When a task does clone(CLONE_NEWNS), the task's user is the 'creator' of the
> > new user_namespace, and the user_namespace is tacked onto a list of those
> > created by this user.
> >
> > When we create or put a user in a namespace, we also do so for all creator
> > users up the creator chain.
> >
> > Changelog:
> > Jun 20: Eric Biederman pointed out the sibling/child_user_ns
> > list is unnecessary!
> >
> > Signed-off-by: Serge Hallyn <serge@us.ibm.com>
> > ---
> > include/linux/sched.h | 1 +
> > include/linux/user_namespace.h | 1 +
> > kernel/user.c | 66 ++++++-----
> > kernel/user_namespace.c | 15 +++-----
> > 4 files changed, 72 insertions(+), 11 deletions(-)
> >
> > diff --git a/include/linux/sched.h b/include/linux/sched.h
> > index 799bbdd..da1bcc6 100644
> > --- a/include/linux/sched.h
> > +++ b/include/linux/sched.h
> > @@ -604,6 +604,7 @@ struct user_struct {
> > /* Hash table maintenance information */
> > struct hlist_node uidhash_node;
> > uid_t uid;

```

>> + struct user_namespace *user_namespace;
>>
>> #ifdef CONFIG_USER_SCHED
>> struct task_group *tg;
>> diff --git a/include/linux/user_namespace.h b/include/linux/user_namespace.h
>> index b5f41d4..f9477c3 100644
>> --- a/include/linux/user_namespace.h
>> +++ b/include/linux/user_namespace.h
>> @@ -13,6 +13,7 @@ struct user_namespace {
>> struct kref kref;
>> struct hlist_head uidhash_table[UIDHASH_SZ];
>> struct user_struct *root_user;
>> + struct user_struct *creator;
>> };
>>
>> extern struct user_namespace init_user_ns;
>> diff --git a/kernel/user.c b/kernel/user.c
>> index 865ecf5..e583be4 100644
>> --- a/kernel/user.c
>> +++ b/kernel/user.c
>> @@ -21,6 +21,7 @@ struct user_namespace init_user_ns = {
>> .kref = {
>> .refcount = ATOMIC_INIT(2),
>> },
>> + .creator = &root_user,
>> .root_user = &root_user,
>> };
>> EXPORT_SYMBOL_GPL(init_user_ns);
>> @@ -53,6 +54,7 @@ struct user_struct root_user = {
>> .files = ATOMIC_INIT(0),
>> .sigpending = ATOMIC_INIT(0),
>> .locked_shm = 0,
>> + .user_namespace = &init_user_ns,
>> #ifdef CONFIG_USER_SCHED
>> .tg = &init_task_group,
>> #endif
>> @@ -71,6 +73,18 @@ static void uid_hash_remove(struct user_struct *up)
>> hlist_del_init(&up->uidhash_node);
>> }
>>
>> +void inc_user_and_creators(struct user_struct *user)
>> +{
>> + struct user_namespace *ns = user->user_namespace;
>> + while (user) {
>> + atomic_inc(&user->__count);
>> + if (ns == ns->creator->user_namespace)
>> + return;
>> + user = ns->creator;

```

```
> > + ns = user->user_namespace;
> > + }
> > +}
> > +
>
> This functionality appears unnecessary. Holding a count on the user
> and the user holding a count on it's user_namespace and the user_namespace
> holding a count on it's creator should be sufficient.
>
> Or am I missing something?
```

Argh. No I don't think you're missing anything. You're absolutely right.

thanks,
-serge

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