
Subject: Re: [RFC][PATCH 1/2] add user namespace [try #2]
Posted by [Cedric Le Goater](#) on Mon, 11 Sep 2006 08:35:45 GMT
[View Forum Message](#) <> [Reply to Message](#)

Kirill Korotaev wrote:

[...]

```
>> +static struct user_namespace *clone_user_ns(struct user_namespace *old_ns)
>> +{
>> + struct user_namespace *ns;
>> +
>> + ns = kmalloc(sizeof(struct user_namespace), GFP_KERNEL);
>> + if (ns) {
> <<<< can you remake this function please so that normal case would go without indentation,
i.e.:
> if (!ns) {
>     error path;
> }
> normal path
```

right, the error path has been growing too much. Will do.

```
>> + int n;
>> + struct user_struct *new_user;
>> +
>> + kref_init(&ns->kref);
>> +
>> + for(n = 0; n < UIDHASH_SZ; ++n)
>> + INIT_LIST_HEAD(ns->uidhash_table + n);
>> +
>> + /* Insert new root user. */
>> + ns->root_user = alloc_uid(ns, 0);
>> + if (!ns->root_user) {
>> + kfree(ns);
>> + return NULL;
>> + }
>> +
>> + /* Reset current->user with a new one */
>> + new_user = alloc_uid(ns, current->uid);
>> + if (!new_user) {
>> + kfree(ns);
>> + return NULL;
>> + }
>> +
>> + switch_uid(new_user);
> <<<< I see switch_uid in this function, but you don't change nsproxy->user_ns here...
> <<<< it is done much later, in sys_unshare()... This looks a bit inconsistent for me...
```

I would say it is safe but i agree with you that it would be better to switch user namespace before than switching user_struct.

> <<<< But I'm unsure whether it can be fixed... :/

he, may be an argument for the clone_ns() syscall ?

```
>> + return ns;
>> +}
>> +
>> +/*
>> + * unshare the current process' user namespace.
>> + */
>> +int unshare_user_ns(unsigned long unshare_flags,
>> +    struct user_namespace **new_user)
>> +{
>> + if (unshare_flags & CLONE_NEWUSER) {
>> + if (!capable(CAP_SYS_ADMIN))
>> + return -EPERM;
> <<<< such checks for CAP_SYS_ADMIN mean that we can't use copy_xxx/clone_xxx functions
directly
> <<<< from OpenVZ code, since VE creation is done with dropped capabilities already.
```

OK.

> <<<< (user level tools decide which capabilities should be granted to VE, so CAP_SYS_ADMIN
> <<<< is not normally granted :))
> <<<< Can we move capability checks into some more logical place which deals with user, e.g.
sys_unshare()?

hmm, another argument for a new syscall() ?

C.
