

---

Subject: Re: User namespaces and keys

Posted by [Casey Schaufler](#) on Wed, 23 Feb 2011 19:24:20 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

On 2/23/2011 7:53 AM, Serge E. Hallyn wrote:

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

>> David Howells <dhowells@redhat.com> writes:

>>>

>>> Serge E. Hallyn <serge@hallyn.com> wrote:

>>>>

>>>>> I guess we need to look at how to mix keys and namespaces again.

>>>> From strictly kernel pov, at the moment, keys are strictly usable only

>>>> by the user in your own user namespace.

>>> I'm not sure that's currently completely true. Key quota maintenance is

>>> namespaced, and the key's owner UID/GID belong to that namespace, so that's

>>> okay, but:

>>>>

>>>> (\*) key\_task\_permission() does not distinguish UIDs and GIDs from different namespaces.

>>>>

>>>> (\*) A key can be referred to by its serial number, no matter whose namespace it is in, and will yield up its given UID/GID, even if these aren't actually meaningful in your namespace.

>>>>

>>>> This means request\_key() can successfully upcall at the moment.

>>>>

>>>> I wonder if I should make the following changes:

>>>>>

>>>>> (1) If the key and the accessor are in different user namespaces, then skip the UID and GID comparisons in key\_task\_permission(). That means that to be able to access the key you'd have to possess the key and the key would have to grant you Possessor access, or the key would have to grant you Other access.

>>>>>

>>>>> (2) If the key and someone viewing the key description are in different namespaces, then indicate that the UID and the GID are -1, irrespective of the actual values.

>>>>>

>>>>> (3) When an upcall is attempting to instantiate a key, it is allowed to access the keys of requestor using the requestor's credentials (UID, GID, groups, security label). Ensure that this will be done in the requestor's user namespace.

>>>>>

>>>>> Nothing should need to be done here, since search\_process\_keyrings() switches to the requestor's creds.

>>>>>

>>>>> Oh, and are security labels user-namespaced?

>> Not at this time. The user namespace as currently merged is little more

>> than a place holder for a proper implementation. Serge is busily  
>> fleshing out that proper implementation.  
>>  
>> Until we reach the point where all checks that have historically been  
>> "if (uid1 == uid2)" become "if ((uidns1 == uidns2) && (uid1 == uid2))"  
>> there will be problems.  
>>  
>> The security labels and probably lsm's in general need to be per user  
>> namespace but we simply have not gotten that far. For the short term I  
>> will be happy when we get a minimally usable user namespace.  
> Note also that when Eric brought this up at the LSM mini-conf two or three  
> years ago, there was pretty general, strong objection to the idea.  
>  
> Like Eric says, I think that'll have to wait. In the meantime, isolating  
> user namespace sandboxes (or containers) using simple LSM configurations  
> is a very good idea.

I confess that I remain less well educated on namespaces than  
I probably should be, but with what I do know it seems that the  
relationships between user namespaces and LSMs are bound to be  
strained from the beginning. Some LSMs (SELinux and Smack) are  
providing similar sandbox capabilities to what you get from user  
namespaces, but from different directions and with different  
use cases.

> -serge  
> --  
> To unsubscribe from this list: send the line "unsubscribe linux-kernel" in  
> the body of a message to majordomo@vger.kernel.org  
> More majordomo info at <http://vger.kernel.org/majordomo-info.html>  
> Please read the FAQ at <http://www.tux.org/lkml/>  
>  
>

---

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

---