

---

Subject: Re: Re: [PATCH] Fix user struct leakage with locked IPC shmem segment  
Posted by [dev](#) on Tue, 17 Jul 2007 09:06:05 GMT

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

---

Andrew Morton wrote:

> On Mon, 16 Jul 2007 16:24:12 +0400

> Pavel Emelianov <xemul@openvz.org> wrote:

>

>

>>When user locks an ipc shmem segment with SHM\_LOCK ctl and the  
>>segment is already locked the shmem\_lock() function returns 0.

>>After this the subsequent code leaks the existing user struct:

>

>

> I'm curious. For the past few months, people@openvz.org have discovered  
> (and fixed) an ongoing stream of obscure but serious and quite  
> long-standing bugs.

thanks a lot :@)

> How are you discovering these bugs?

Not sure what to answer :) Just trying to do our best.

This bug was thought over by Pavel for about 3 month after a single  
uid leak in container was detected by beancounters' kernel memory accounting...

>>== ipc/shm.c: sys\_shmctl() ==

>> ...

>> err = shmem\_lock(shp->shm\_file, 1, user);

>> if (!err) {

>> shp->shm\_perm.mode |= SHM\_LOCKED;

>> shp->mlock\_user = user;

>> }

>> ...

>>==

>>

>>Other results of this are:

>>1. the new shp->mlock\_user is not get-ed and will point to freed

>> memory when the task dies.

>

>

> That sounds fairly serious - can this lead to memory corruption and crashes?

Yes it can. According to Pavel when the shmem segment is destroyed it  
puts the mlock\_user pointer, which can already be stalled.

Kirill

---