Subject: Re: [PATCH 1/2] rcfs core patch
Posted by dev on Sun, 11 Mar 2007 17:09:29 GMT
View Forum Message <> Reply to Message

Herbert,

> sorry, I'm not in the lucky position that I get payed
> for sending patches to LKML, so I have to think twice
> before I invest time in coding up extra patches ...
>
> i.e. you will have to live with my comments for now
looks like you have no better argurments then that...

>>Looks like your main argument is non-intrusive...
>>"working", "secure", "flexible" are not required to
>>people any more? :/
>
>
> well, Linux-VServer is "working", "secure", "flexible"
> _and_ non-intrusive ... it is quite natural that less
> won't work for me ... and regarding patches, there
> will be a 2.2 release soon, with all the patches ...
ok. please check your dcache and slab accounting then
(analyzed according to patch-2.6.20.1-vs2.3.0.11.diff):
Both are full of races and problems. Some of them:
1. Slabs allocated from interrupt context are charged to current context.
   So charged values contain arbitrary mess, since during interrupts
   context can be arbitrary.
2. Due to (1) I guess you do not make any limiting of slabs.
   So there are number of ways how to consume a lot of kernel
   memory from inside container and
   OOM killer will kill arbitrary tasks in case of memory-shortage after that.
   Don't think it is secure... real DoS.
3. Dcache accounting simply doesn't work, since
   charges/uncharges are done on current context (sic!!!), which is arbitrary.
   i.e. lookup can be done in VE context, while dcache shrink can be done
   from another context.
   So the whole problem with dcache DoS is not solved at all, it is just hard to trigger.
4. Dcache accounting is racy, since your checks look like:
   if (atomic_read(de->d_count))
      charge();
   which obviously races with other dput()'s/lookups.
5. Dcache accounting can be hit if someone does `find /` inside container.
   After that it is impossible to open something new,
   since all the dentries for directories in dcache will have d_count > 0
   (due it's children).
   It is a BUG.
6. Counters can be non-zero on container stop due to all of the above.

There are more and more points which arise when such a non-intrusive
accounting is concerned. I'm really suprised, that you don't see them
or try to behave as you don't see them :/
And, please, believe me, I would not suggest so much complicated patches
If everything was so easy and I had no reasons simply to accept vserver code.

> well, as you know, all current solutions use a syscall
> interface to do most of the work, in the OpenVZ/Virtuozzo
> case several, unassigned syscalls are used, while
> FreeVPS and Linux-VServer use a registered and versioned
> (multiplexed) system call, which works quite fine for
> all known purposes ...
>
> I'm quite happy with the extensibility and flexibility
> the versioned syscall interface has, the only thing I'd
> change if I would redesign that interface is, that I
> would add another pointer argument to eliminate 32/64bit
> issues completely (i.e. use 4 args instead of the 3)

Well, I would be happy with syscalls also.
But my guess is that cpuset guys who already use fs approach won't be happy :/
Maybe we can use both?

Thanks,
Kirill