

---

Subject: Re: [PATCH 11/11] protect architectures where THREAD\_SIZE >= PAGE\_SIZE against fork bombs

Posted by [Frederic Weisbecker](#) on Mon, 25 Jun 2012 20:57:51 GMT

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

---

On Mon, Jun 25, 2012 at 11:38:18AM -0700, Tejun Heo wrote:

> On Mon, Jun 25, 2012 at 06:55:35PM +0200, Frederic Weisbecker wrote:

> > On 06/25/2012 04:15 PM, Glauber Costa wrote:

> >

> > > Because those architectures will draw their stacks directly from  
> > > the page allocator, rather than the slab cache, we can directly  
> > > pass \_\_GFP\_KMEMCG flag, and issue the corresponding free\_pages.

> > >

> > > This code path is taken when the architecture doesn't define  
> > > CONFIG\_ARCH\_THREAD\_INFO\_ALLOCATOR (only ia64 seems to), and has  
> > > THREAD\_SIZE >= PAGE\_SIZE. Luckily, most - if not all - of the  
> > > remaining architectures fall in this category.

> > >

> > > This will guarantee that every stack page is accounted to the memcg  
> > > the process currently lives on, and will have the allocations to fail  
> > > if they go over limit.

> > >

> > > For the time being, I am defining a new variant of THREADINFO\_GFP, not  
> > > to mess with the other path. Once the slab is also tracked by memcg,  
> > > we can get rid of that flag.

> > >

> > > Tested to successfully protect against :(){ :|:& };;

> > >

> > > Signed-off-by: Glauber Costa <glommer@parallels.com>

> > > CC: Christoph Lameter <cl@linux.com>

> > > CC: Pekka Enberg <penberg@cs.helsinki.fi>

> > > CC: Michal Hocko <mhocko@suse.cz>

> > > CC: Kamezawa Hiroyuki <kamezawa.hiroyu@jp.fujitsu.com>

> > > CC: Johannes Weiner <hannes@cmpxchg.org>

> > > CC: Suleiman Souhlal <suleiman@google.com>

> >

> >

> > Acked-by: Frederic Weisbecker <fweisbec@redhat.com>

>

> Frederic, does this (with proper slab accounting added later) achieve  
> what you wanted with the task counter?

I think so yeah. Relying on general kernel memory accounting should do the trick for us. And if we need more finegrained limitation on kernel stack accounting we can still add it incrementally. But I believe global limitation can be enough.

Thanks.

---