Subject: Re: [PATCH 11/11] protect architectures where THREAD_SIZE >= PAGE_SIZE against fork bombs
Posted by Frederic Weisbecker on Tue, 26 Jun 2012 13:38:41 GMT
View Forum Message <> Reply to Message

On Tue, Jun 26, 2012 at 04:48:08PM +0400, Glauber Costa wrote:
> On 06/25/2012 10:38 PM, 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?
> >
>
> A note: Frederic may confirm, but I think he doesn't even need
> the slab accounting to follow to achieve that goal.

Limiting is enough. But that requires internal accounting.