Subject: Re: [PATCH v2 11/11] protect architectures where THREAD_SIZE >= PAGE_SIZE against fork bombs
Posted by Glauber Costa on Tue, 21 Aug 2012 09:40:45 GMT
View Forum Message <> Reply to Message

On 08/21/2012 01:35 PM, Michal Hocko wrote:
> On Thu 09-08-12 17:01:19, 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.
>
> quick git grep "define *THREAD_SIZE\>" arch says that there is no such
> architecture.
>
>> 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 :(){ :|:& };:
>
> I guess there were no other tasks in the same group (except for the
> parent shell), right?

Yes.

> I am asking because this should trigger memcg-oom
> but that one will usually pick up something else than the fork bomb
> which would have a small memory footprint. But that needs to be handled
> on the oom level obviously.
>
Sure, but keep in mind that the main protection is against tasks *not*
in this memcg.