Subject: Re:  Re: [PATCH 1/2] iptables 32bit compat layer
Posted by dim on Tue, 21 Feb 2006 09:24:27 GMT
View Forum Message <> Reply to Message

On Tuesday 21 February 2006 00:23, Andi Kleen wrote:
> Mishin Dmitry <dim@openvz.org> writes:
> > Hello,
> >
> > This patch set extends current iptables compatibility layer in order to
> > get 32bit iptables to work on 64bit kernel. Current layer is insufficient
> > due to alignment checks both in kernel and user space tools.
> >
> > This patch introduces base compatibility interface for other ip_tables
> > modules
>
> Nice. But some issues with the implementation
>
>
> +#if defined(CONFIG_X86_64)
> +#define is_current_32bits() (current_thread_info()->flags & _TIF_IA32)
>
> This should be is_compat_task(). And we don't do such ifdefs
> in generic code.  And what you actually need here is a
> is_compat_task_with_funny_u64_alignment() (better name sought)
>
> So I would suggest you add macros for that to the ia64 and x86-64
> asm/compat.hs and perhaps a ARCH_HAS_FUNNY_U64_ALIGNMENT #define in there.
agree.

>
> + ret = 0;
> + switch (convert) {
> +  case COMPAT_TO_USER:
> +   pt = (struct ipt_entry_target *)target;
>
> etc. that looks ugly. why can't you just define different functions
> for that?  We don't really need in kernel ioctl
3 functions and the requirement that if defined one, than defined all of them?

>
> +#ifdef CONFIG_COMPAT
> + down(&compat_ipt_mutex);
> +#endif
>
> Why does it need an own lock?
Because it protects only compatibility translation. We spend a lot of time in
these cycles and I don't think that it is a good way to hold ipt_mutex for
this. The only reason of this lock is offset list - in the first iteration I

fill it, in the second - use it. If you know how to implement this better,
let me know.

>
> Overall the implementation looks very complicated. Are you sure
> it wasn't possible to do this simpler?
ughh...
I don't like this code as well. But seems that it is due to iptables code
itself, which was designed with no thoughts about compatibility in minds.

So, I see following approaches:
1) do translation before pass data to original do_replace or get_entries.
Disadvantage of such approach is additional 2 cycles through data.
2) do translation in compat_do_replace and compat_get_entries. Avoidance of
additional cycles, but some code duplication.
3) remove alignment checks in kernel - than we need only first time
translation from kernel to user. But such code will not work with both 32bit
and 64 bit iptables at the same time.

Any suggestions?

>
>
> -Andi
>
--
Thanks,
Dmitry.