Subject: Re: [RFD] L2 Network namespace infrastructure
Posted by ebiederm on Sun, 24 Jun 2007 12:58:54 GMT

View Forum Message <> Reply to Message

David Miller <davem@davemloft.net> writes:

> From: ebiederm@xmission.com (Eric W. Biederman)
> Date: Sat, 23 Jun 2007 15:41:16 -0600
>
>> If you want the argument to compile out. That is not a problem at all.
>> I dropped that part from my patch because it makes infrastructure more
>> complicated and there appeared to be no gain. However having a type
>> that you can pass that the compiler can optimize away is not a
>> problem. Basically you just make the argument:
>>
>> typedef struct {} you_can_compile_me_out; /* when you don't want it. */
>> typedef void * you_can_compile_me_out;   /* when you do want it. */
>>
>> And gcc will generate no code to pass the argument when you compile
>> it out.
>
> I don't want to have to see or be aware of the types or the
> fact that we support namespaces when I work on the networking
> code.
>
> This is why I like the security layer in the kernel we have,
> I can disable it and it's completely not there. And I can
> be completely ignorant of it's existence when I work on the
> networking stack.

That is a mostly reasonable objection. I certainly see minimal
impact to people working on the code long term is an important goal.

My impression of the current linux security subsystem is a non-flexible
brittle iptables that works between processes. I'm rather jaded in
that regard because I have fixed several things and broken the security
subsystem, and people yell. So I'm not certain if it is good for anything
or if we can actually ignore it right now. However for most of networking
we can ignore it.

What makes namespaces interesting to me is that they enable us to do things
that we can't do with linux today. Sorry you seem to have a security
mono-focus so this bears repeating.

I'm not trying to hide network namespaces because I think that is
potentially a much more serious maintenance issue.

During the merge and enabling of this we have to audit everything or not be responsible programmers. So that is going to be slightly intrusive.

I am convinced I can keep network namespaces something that is so trivial and obvious to get right you won't have to pay attention to them.

Mostly it isn't the core paths in the network that are a challenge to work with but the little out of the way paths that don't take a network device or a socket, and things like the initialization code which is almost never touched.

I don't actually thing the core fast paths of the network stack even make namespaces visible. I will go back an reexamine that.

You are asking for something much more challenging though. You are asking for absolutely no foot-print. Something that is a completely a side-car. Given the thousands of global variables in the networking code I don't know if that is possible to implement.

Eric