

---

Subject: Re: [PATCH] Teach cifs about network namespaces (take 2)

Posted by [Rob Landley](#) on Wed, 12 Jan 2011 13:57:36 GMT

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

---

On 01/11/2011 03:30 PM, Jeff Layton wrote:

> I've got a patch queued that rearranges some fields in TCP\_Server\_Info  
> according to pahole's recommendations. You may want to base this patch  
> on that.

Queued where?

> It might also be good to run pahole on this after your patch to see if  
> it might be better placed...

I put it right after the network address fields affected by it. I can  
stick it between:

```
char *hostname; /* hostname portion of UNC string */  
struct socket *ssocket;
```

So that two consecutive pointers become three consecutive pointers. But  
again, I haven't seen what you did to the structure yet...

```
>> spin_lock(&cifs_tcp_ses_lock);  
>> list_for_each_entry(server, &cifs_tcp_ses_list, tcp_ses_list) {  
>> + if (HAVE_NET_NS &&  
>> +   cifs_net_ns(server) != current->nsproxy->net_ns)  
>> +   continue;  
>> +  
>> +  
>
```

> This HAVE\_NET\_NS thing is pretty ugly.

It's a compile time constant optimized away by the compiler, and cleanly  
taking out the entire dependent clause with it when it's false (because  
&& only evaluates the right side when the left side is true, meaning the  
if can never trigger, thus dead code elimination).

How is that ugly? It does exactly what it says. I'm not sure what the  
criteria are here. (Was it better when it looked like a function rather  
than a constant? I'd use the CONFIG\_NET\_NS symbol directly if it was 0  
or 1 rather than undefined or 1, but it isn't. It's designed for use  
with #ifdefs only, not from C code.)

> This is not a high-performance  
> codepath. My vote would be to get rid of this and just have the useless  
> test when CONFIG\_NET\_NS isn't set.  
>  
> Alternately, you could turn the comparison into a static inline or

> macro, and simply have that compile down to nothing when CONFIG\_NET\_NS  
> isn't set.

I tried returning a constant from a static inline, it doesn't propagate  
the constant far enough to do compile time dead code elimination based  
on the return value under gcc 4.4.3.

As for a macro, the constant already is a macro. So adding more #ifdefs  
to the headers to make the code in the function look like it's doing  
something other than it actually is somehow improves matters? (I'm not  
understanding the aesthetic criteria here at all.)

I think I'll just ignore the bloat. Make allnoconfig is already 800k  
compressed, what's a few more bytes...

```
>> @@ -1754,6 +1762,8 @@ cifs_get_tcp_session(struct smb_vol *volume_info)
>> out_err_crypto_release:
>> cifs_crypto_shash_release(tcp_ses);
>>
>> + put_net(cifs_net_ns(tcp_ses));
>> +
> ^^^
> This looks like it will oops if you end up doing the "goto
> out_err_crypto_release" after the extract_hostname call.
```

Hmmm... Yup, failure case when CONFIG\_NET\_NS enabled will dereference  
the null pointer. I need to move the initialization up a few lines.  
Thanks.

Rob

---

Containers mailing list  
Containers@lists.linux-foundation.org  
<https://lists.linux-foundation.org/mailman/listinfo/containers>

---