

Re: Out of tree module using LSM

Source: <http://linux.derkeiler.com/Mailing-Lists/Kernel/2007-11/msg09482.html>

- *From:* James Morris <jmorris@xxxxxxxxxx>
 - *Date:* Thu, 29 Nov 2007 11:12:40 +1100 (EST)
-

On Wed, 28 Nov 2007, tvrtko.ursulin@xxxxxxxxxx wrote:

So as there is no question the current code does some ugly things it is even more true that we would be even more happy to use an official API.

How about becoming involved in creating that official API ?

"A person will stand on the top of a hill for a very long time with their mouth open before a roast duck will fly in"

LSM was that and we were happily using it which we won't be able to do if it abruptly goes away. Yes it is not a perfect match but until it is modified to be better, or until something appropriate is designed and implemented, it would be very nice if it could stay.

So, what you're asking for is for us to maintain infrastructure in the kernel, for your out of tree code, which we could not have even known about until it was dumped on sourceforge a couple of days ago ?

And you're expecting us to modify it to be "better", for you, but without us possibly knowing what you want ?

How is this sustainable? Every time we make a change in the kernel code, do we have to wait for months while all of the (unknown) out of tree users discover their code is broken and then complain to Linus ?

Do you really think we should be supporting developers who make zero effort to engage with the kernel community, and instead abuse kernel APIs and ship out of tree code to customers?

You are essentially demanding that we provide you with a stable kernel API. I suggest you review Greg KH's excellent document on the issue: <http://os.cqu.edu.au/docs/kernel-doc/Documentation/stable_api_nonsense.txt>

Also, your users are likely unaware of the risks associated with these

Re: Out of tree module using LSM

techniques, such as, that they are not actually running "linux" any more, and that their kernel is made unportable by both the community and their OS vendors when code like this is being used:

```
/*
 * hidden vsmnt_lock handling
 */
#ifdef TALPA_USE_VFSMOUNT_LOCK
static inline void talpa_vfsmount_lock(void)
{
spinlock_t* talpa_vfsmount_lock_addr = (spinlock_t *)TALPA_VFSMOUNT_LOCK_ADDR;

spin_lock(talpa_vfsmount_lock_addr);
}
```

Code quality and correctness issues are definitely relevant, and in fact are directly linked to the fact that out of tree code receives no community peer review, and is not able to be considered in the general kernel design & development process.

Hacks like this effectively void support from the community, which is another way of saying that the user's kernel becomes proprietary once the module containing it is loaded.

Actually, it's worse than that, as not only is the kernel no longer "open source", but also lacks even the level of support provided by proprietary OS vendors (who provide stable interfaces for such purposes).

If you want to resolve this properly, you'll need to do more than complain to Linus when the upstream APIs (which you are abusing) change.

You'll need to become productively engaged in designing, developing and maintaining appropriate APIs, which suit not only your specific needs, but likely those of others, and submit your code for review and upstream inclusion.

What I'd suggest, is that you create a public mailing list, get the other AV projects involved, and then work on a set of requirements so that the general problem can be understood. Then, propose a solution to the problem and have it reviewed by core kernel folk (cc it to lkml), so that it can be evaluated in terms of e.g. VFS correctness, maintainability etc.

That would be at least a useful first step in taking this issue seriously.

Thanks,

– James

—

James Morris

<jmorris@xxxxxxxxxx>

Re: Out of tree module using LSM

Re: Out of tree module using LSM

—

To unsubscribe from this list: send the line "unsubscribe linux-kernel" in the body of a message to majordomo@xxxxxxxxxxxxxxxxx

More majordomo info at <http://vger.kernel.org/majordomo-info.html>

Please read the FAQ at <http://www.tux.org/lkml/>