

Re: [PATCH] Add TPM hardware enablement driver

Source: <http://linux.derkeiler.com/Mailing-Lists/Kernel/2005-04/1309.html>

From: Kylene Jo Hall (kjhall_at_us.ibm.com)

Date: 04/05/05

To: Greg KH <greg@kroah.com>

Date: Tue, 05 Apr 2005 11:14:49 -0500

On Thu, 2005-03-24 at 13:33 -0800, Greg KH wrote:

> On Thu, Mar 24, 2005 at 04:04:25PM -0500, Jeff Garzik wrote:

>> Greg KH wrote:

>>> On Tue, Mar 22, 2005 at 09:02:24PM -0500, Jeff Garzik wrote:

>>>

>>>> Kylene Hall wrote:

>>>>

>>>>> what is the purpose of this `pci_dev_get/put`? attempting to prevent

>>>>> hotplug or

>>>>> something?

>>>>

>>>>

>>>>> Seems that since there is a reference to the device in the chip structure

>>>>> and I am making the file private data pointer point to that chip

>>>>> structure this is another reference that must be accounted for. If you

>>>>> remove it with it open and attempt read or write bad things will happen.

>>>>> This isn't really hotpluggable either as the TPM is on the motherboard.

>>>>

>>>>> My point was that there will always be a reference -anyway-, AFAICS.

>>>>> There is a `pci_dev` reference assigned to the `pci_driver` when the PCI

>>>>> driver is loaded, and all uses by the TPM generic code of this pointer

>>>>> are -inside- the `pci_driver's pci_dev` object lifetime.

>>>>

>>>>

>>>>> Think of the following situation:

>>>>> - driver is bound to device.

>>>>> - userspace opens char dev node.

>>>>> - device is removed from the system (using `fakephp` I can do this

>>>>> to `_any_ pci` device, even if it is on the motherboard.)

>>>>> - userspace writes to char dev node

>>>>> - driver attempts to access `pci` device structure that is no

>>>>> longer present in memory.

>>>>

>>>>> Because of this open needs to get a reference to the `pci` device to

>>>>> prevent oopses, or the driver needs to be aware of "device is now gone"

>>>>> in some other manner.

>>>>

Linux-Kernel: Re: [PATCH] Add TPM hardware enablement driver

> > *Thanks for explaining; agreed.*
> >
> > *However, there appear to still be massive bugs in this area:*
> >
> > *Consider the behavior of the chrdev if a PCI device has been*
> > *unplugged. It's still actively messing with the non-existent*
> > *hardware, and never checks for dead h/w AFAICS.*
>
> *I agree, the driver should be fixed to handle this properly.*
>

I have now played with the fakephp driver and have a better understanding of these interactions, but I still have questions. With the current structure there is a problem because everything is "cleaned-up" with the tpm_remove function even if userspace has the device open when the tpm's slot is removed and then there are problems on subsequent reads/writes. The get/put didn't really stop this from happening. Is it right to fix this by cleaning mostly up and placing a flag in the read/write path to check for this condition?

This problem actually becomes more complicated. Since the TPM lives on the LPC bus and does not have it's own id we were in the process of converting the driver to not use a pci_driver structure at all like the example in drivers/char/watchdog/i8xx_tco.c. This is desirable so that the driver does not claim the id and other drivers can still find their devices that also live on the LPC bus and thus share the same ID. Without a pci_driver structure there is no probe or remove functions and thus the driver is not alerted of the loss of hardware. Any recommendations of how to handle this situation?

Thanks,
Kylie

-

To unsubscribe from this list: send the line "unsubscribe linux-kernel" in the body of a message to majordomo@vger.kernel.org
More majordomo info at <http://vger.kernel.org/majordomo-info.html>
Please read the FAQ at <http://www.tux.org/lkml/>