Subject: Re: [patch i2o 5/6] i2o_proc files permission
Posted by vaverin on Tue, 15 May 2007 12:59:49 GMT

View Forum Message <> Reply to Message

I would add:
I've reported about this issue some time ago to security@kernel.org
How this lockup can be reproduced:

- boot the kernel,
- load i2o_proc module
- login as user and read all entries in /proc/i2o/ directory

My testnode hangs when I try to read any file from /proc/i2o/iop0/030/ directory: I have the shell prompt and even can try to start any new command which hangs due exec is not works.
Node is pingable, but I cannot login to it nor via ssh neither from local console. Magic Sysrq keys are works. Kernel space software watchdog module works OK. But all the new commnds hangs, looks like i2o controller is in coma.

Greg KH wrote:
And I'd classify this a "low" security issue, as you have to be root to load the i2o_proc module, and I doubt that the distros automatically load it.

Markus Lidel:
The problem is not really driver related, IIRC. The driver properly send a I2O command to the controller, but this one couldn't handle it and instead of just aborting the command "panic's". IIRC it's only Adaptec related, the Promise controllers doesn't show this behaviour.

Thank you,
 Vasily Averin

Vasily Averin wrote:

```
> Reading from some i2o related proc files can lead to the i2o controller hang due
> unknown reasons. As a workaround this patch changes the permission of these
> files to root-only accessible.
>
> Signed-off-by: Vasily Averin <vvs@sw.ru>
>
> --- lk2.6/drivers/message/i2o/i2o_proc.c
> +++ lk2.6/drivers/message/i2o/i2o_proc.c
> @@ -1855,17 +1855,17 @@ static i2o_proc_entry i2o_proc_generic_i
>   * Device specific entries
>   */
>  static i2o_proc_entry generic_dev_entries[] = {
> - {"groups", S_IFREG | S_IRUGO, &i2o_seq_fops_groups},
> - {"phys_dev", S_IFREG | S_IRUGO, &i2o_seq_fops_phys_device},
> - {"claimed", S_IFREG | S_IRUGO, &i2o_seq_fops_claimed},
> - {"users", S_IFREG | S_IRUGO, &i2o_seq_fops_users},
> - {"priv_msgs", S_IFREG | S_IRUGO, &i2o_seq_fops_priv_msgs},
> - {"authorized_users", S_IFREG | S_IRUGO, &i2o_seq_fops_authorized_users},
> - {"dev_identity", S_IFREG | S_IRUGO, &i2o_seq_fops_dev_identity},
> - {"ddm_identity", S_IFREG | S_IRUGO, &i2o_seq_fops_ddm_identity},
> - {"user_info", S_IFREG | S_IRUGO, &i2o_seq_fops_uinfo},
> - {"sgl_limits", S_IFREG | S_IRUGO, &i2o_seq_fops_sgl_limits},
> - {"sensors", S_IFREG | S_IRUGO, &i2o_seq_fops_sensors},
> + {"groups", S_IFREG | S_IRUSR, &i2o_seq_fops_groups},
> + {"phys_dev", S_IFREG | S_IRUSR, &i2o_seq_fops_phys_device},
> + {"claimed", S_IFREG | S_IRUSR, &i2o_seq_fops_claimed},
> + {"users", S_IFREG | S_IRUSR, &i2o_seq_fops_users},
> + {"priv_msgs", S_IFREG | S_IRUSR, &i2o_seq_fops_priv_msgs},
> + {"authorized_users", S_IFREG | S_IRUSR, &i2o_seq_fops_authorized_users},
> + {"dev_identity", S_IFREG | S_IRUSR, &i2o_seq_fops_dev_identity},
> + {"ddm_identity", S_IFREG | S_IRUSR, &i2o_seq_fops_ddm_identity},
> + {"user_info", S_IFREG | S_IRUSR, &i2o_seq_fops_uinfo},
> + {"sgl_limits", S_IFREG | S_IRUSR, &i2o_seq_fops_sgl_limits},
> + {"sensors", S_IFREG | S_IRUSR, &i2o_seq_fops_sensors},
>   {NULL, 0, NULL}
>  };
>
```