

Re: [PATCH] enable interrupts in user path of page fault.

Re: [PATCH] enable interrupts in user path of page fault.

Source: <http://linux.derkeiler.com/Mailing-Lists/Kernel/2007-06/msg03041.html>

- *From:* Andrew Morton <akpm@xxxxxxxxxxxxxxxxxxxxxx>
 - *Date:* Thu, 7 Jun 2007 18:58:51 -0700
-

On Wed, 06 Jun 2007 23:34:04 -0400

Steven Rostedt <rostedt@xxxxxxxxxxx> wrote:

This is a minor fix, but what is currently there is essentially wrong. In `do_page_fault`, if the faulting address from user code happens to be in kernel address space (`int *p = (int*)-1; p = 0xbed;`) then the `do_page_fault` handler will jump over the `local_irq_enable` with the

```
goto bad_area_nosemaphore;
```

But the first line there sees this is user code and goes through the process of sending a signal to send SIGSEGV to the user task. This whole time interrupts are disabled and the task can not be preempted by a higher priority task.

This patch always enables interrupts in the user path of the `bad_area_nosemaphore`.

Signed-off-by: Steven Rostedt <rostedt@xxxxxxxxxxx>

```
diff --git a/arch/i386/mm/fault.c b/arch/i386/mm/fault.c
index 29d7d61..1ecb3e4 100644
--- a/arch/i386/mm/fault.c
+++ b/arch/i386/mm/fault.c
@@ -458,6 +458,11 @@ bad_area:
bad_area_nosemaphore:
/* User mode accesses just cause a SIGSEGV */
if (error_code & 4) {
+ /*
+ * It's possible to have interrupts off here.
+ */
+ local_irq_enable();
+

```

Interrupts got disabled here because `do_page_fault()` is an interrupt-disabling trap, yes?

Re: [PATCH] enable interrupts in user path of page fault.

Re: [PATCH] enable interrupts in user path of page fault.

The patch looks reasonable to me: a slight reduction in interrupt-off latency when really weird things are happening.

The patch also breaks things, I think: if userspace is running with interrupts disabled and tries to access kernel memory it will presently whizz through the kernel without ever enabling interrupts. With this change, the kernel will now enable interrupts, which is presumably not what the application wanted.

However it's surely already the case that most pagefaults will go and enable interrupts on this process anyway, so no big loss there. I'd expect the kernel to spit piles of might_sleep() warnings when all this happens, so maybe it just doesn't happen for some reason.

—

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

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

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