

## parport\_pc superio chip fixes

**Source:** <http://linux.derkeiler.com/Mailing-Lists/Kernel/2004-09/8987.html>

---

**From:** Andrea Arcangeli ([andrea\\_at\\_novell.com](mailto:andrea_at_novell.com))

**Date:** 09/30/04

Date: Thu, 30 Sep 2004 08:08:51 +0200

To: Andrew Morton <[akpm@osdl.org](mailto:akpm@osdl.org)>, Tim Waugh <[twbaugh@redhat.com](mailto:twbaugh@redhat.com)>

This patch fixes some troubles that somebody reported me with the superio chips.

In short `rmmod parport_pc && cat /proc/iomem` was good enough for crashing the box hard on some machine (and `hwscan --printer` was doing just that). The way the oops triggers is that `iomem` tries to `vsprintf` the `p->name`, but the `p->name` was a static string in the module address (now unloaded).

The reason is that the superio chip scanning leaves up to two persistent ranges claimed. But the second (legacy) pass has no way to notice the resources are already reclaimed. Plus if the `superio->io` was different than the "io" variable (the range to scan for superio chips) the "io" range would generate a leak of the original "io" range too.

I simply make sure to always release the requested space during the superio scan, and I make sure not to instantiate new ranges in the `p->base` that would cause the later `parport` scan to fail too (plus leaving up to leaked resources).

The previous code that was returning values and was leaving garbage in there made no sense to me. My best guess (assuming I didn't misread it ;) is that probably somebody added the `request_region` without realizing they're pointing to the very same address that would be requested later (and nobody does accesses on those ranges until later, so it was very safe to claim it later).

Disclaimer: I don't have the specs of the winbond and smsc at hand, I just guessed what they do from the code (nothing checks `superio->io` except `get_superio_dma` `get_superio_irq`, which made the thing enough self explanatory to fix it without specs)

Signed-off-by: Andrea Arcangeli <[andrea@novell.com](mailto:andrea@novell.com)>

--- sles/drivers/parport/parport\_pc.c~1~ 2004-09-30 05:56:36.000000000 +0200

+++ sles/drivers/parport/parport\_pc.c 2004-09-30 07:54:56.469113152 +0200

@@ -1194,7 +1194,7 @@ struct parport\_operations parport\_pc\_ops

## Linux-Kernel: parport\_pc superio chip fixes

```
#ifndef CONFIG_PARPORT_PC_SUPERIO
/* Super-IO chipset detection, Winbond, SMSC */
-static int __devinit show_parconfig_smsc37c669(int io, int key)
+static void __devinit show_parconfig_smsc37c669(int io, int key)
{
    int cr1,cr4,cra,cr23,cr26,cr27,i=0;
    static const char *modes[]={ "SPP and Bidirectional (PS/2)",
@@ -1261,26 +1261,17 @@ static int __devinit show_parconfig_smsc
        superios[i].io = 0x278;
        superios[i].irq = 5;
    }
- if (io != superios[i].io) {
- /* how many bytes? */
- if (!request_region(superios[i].io, 3, "smsc parport")) {
- superios[i].io = 0;
- return 0;
- }
- }
        d=(cr26 &0x0f);
        if((d==1) || (d==3))
            superios[i].dma= d;
        else
            superios[i].dma= PARPORT_DMA_NONE;
- return 1;
    }
}
- return 0;
}

-static int __devinit show_parconfig_winbond(int io, int key)
+static void __devinit show_parconfig_winbond(int io, int key)
{
    int cr30,cr60,cr61,cr70,cr74,crf0,i=0;
    static const char *modes[] = {
@@ -1336,23 +1327,14 @@ static int __devinit show_parconfig_winb
        printk(KERN_INFO "Super-IO: too many chips!\n");
        else {
            superios[i].io = (cr60<<8)|cr61;
- if (io != superios[i].io) {
- /* how many bytes? */
- if (!request_region(superios[i].io, 3, "winbond parport")) {
- superios[i].io = 0;
- return 0;
- }
- }
            superios[i].irq = cr70&0x0f;
            superios[i].dma = (((cr74 & 0x07) > 3) ?
                PARPORT_DMA_NONE : (cr74 & 0x07));
- return 1;
}

```

## Linux-Kernel: parport\_pc superio chip fixes

```
    }  
  }  
- return 0;  
}  
  
-static int __devinit decode_winbond(int efer, int key, int devid, int devrev, int oldid)  
+static void __devinit decode_winbond(int efer, int key, int devid, int devrev, int oldid)  
{  
    const char *type = "unknown";  
    int id, progif=2;  
@@ -1360,7 +1342,7 @@ static int __devinit decode_winbond(int  
    if (devid == devrev)  
        /* simple heuristics, we happened to read some  
        non-winbond register */  
- return 0;  
+ return;  
  
    id=(devid<<8) | devrev;  
  
@@ -1385,20 +1367,19 @@ static int __devinit decode_winbond(int  
    efer, key, devid, devrev, oldid, type);  
  
    if (progif == 2)  
- return show_parconfig_winbond(efer, key);  
- return 0;  
+ show_parconfig_winbond(efer, key);  
}  
  
-static int __devinit decode_smsc(int efer, int key, int devid, int devrev)  
+static void __devinit decode_smsc(int efer, int key, int devid, int devrev)  
{  
    const char *type = "unknown";  
- int (*func)(int io, int key);  
+ void (*func)(int io, int key);  
    int id;  
  
    if (devid == devrev)  
        /* simple heuristics, we happened to read some  
        non-smisc register */  
- return 0;  
+ return;  
  
    func=NULL;  
    id=(devid<<8) | devrev;  
@@ -1414,8 +1395,7 @@ static int __devinit decode_smsc(int efe  
    efer, key, devid, devrev, type);  
  
    if (func)  
- return func(efer, key);  
- return 0;  
+ func(efer, key);  
  
parport_pc superio chip fixes
```

## Linux-Kernel: parport\_pc superio chip fixes

```
}
```

```
@@ -1448,8 +1428,7 @@ static void __devinit winbond_check(int  
    if ((x_devid == devid) && (x_devrev == devrev) && (x_oldid == oldid))  
        goto out; /* protection against false positives */
```

```
- if (decode_winbond(io,key,devid,devrev,oldid));
```

```
- return;
```

```
+ decode_winbond(io,key,devid,devrev,oldid);
```

```
out:
```

```
    release_region(io, 3);
```

```
}
```

```
@@ -1482,8 +1461,7 @@ static void __devinit winbond_check2(int  
    if ((x_devid == devid) && (x_devrev == devrev) && (x_oldid == oldid))  
        goto out; /* protection against false positives */
```

```
- if (decode_winbond(io,key,devid,devrev,oldid))
```

```
- return;
```

```
+ decode_winbond(io,key,devid,devrev,oldid);
```

```
out:
```

```
    release_region(io, 3);
```

```
}
```

```
@@ -1522,8 +1500,7 @@ static void __devinit smsc_check(int io,  
    (x_oldid == oldid) && (x_rev == rev))  
        goto out; /* protection against false positives */
```

```
- if (decode_smsc(io,key,oldid,oldrev))
```

```
- return;
```

```
+ decode_smsc(io,key,oldid,oldrev);
```

```
out:
```

```
    release_region(io, 3);
```

```
}
```

```
-
```

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/>