

## Re: PATCH: cdrecord: avoiding scsi device numbering for ide devices

*Source:* <http://linux.derkeiler.com/Mailing-Lists/Kernel/2004-08/6457.html>

---

**From:** Pascal Schmidt (*der.eremit\_at\_email.de*)

**Date:** 08/22/04

Date: Sun, 22 Aug 2004 15:13:37 +0200 (CEST)  
To: Joerg Schilling <schilling@fokus.fraunhofer.de>

On Sun, 22 Aug 2004, Joerg Schilling wrote:

> *Not checking for Write access permissions at this place is a typical  
> mistake made by novice programmers, so I never thought this could be in  
> Linux.....*

People will find this kind of language inflammatory. ;) However, exactly because it is such a bad mistake did Linus put out what he deemed a correct fix *\*immediately\**.

> *If Linux still does not check for write permissions, I would consider  
> there is still a bug.*

The open question is whether write permission really is meaningful enough to allow arbitrary SCSI commands. I personally think "being able to wipe the drive firmware" is too much, and since filtering of vendor commands is generally impossible to do right, sending SG\_IO should require CAP\_SYS\_RAWIO capability.

> *If there is a list of "arently safe" SCSI commands that are allowed to  
> be executed, then there is another bug in Linux. The only SCSI command  
> that could be called safe if Test Unit Ready and even this only if not  
> send more then once every few seconds.*

Currently (2.6.8.1), there is a list in the kernel. I agree that it doesn't make sense. I would think read permission means to be able to read from the device, write means you can write. I would even go as far as *\*not\** to have that mean "you can also read/write via SG\_IO", because for normal uses of the device, read(2) and write(2) should be enough.

BTW, there are a number of people on the kernel list who believe a filter list is bad and generally unmaintainable.

> *There are several SCSI commands that look safe but would result in coasters*

> *if issued while a CD or DVD is written.*

Good point.

> *The best immediate fix for the problem is to just check for read & write permissions on the file descriptor and otherwise revert to how it has been before 2.6.8.*

I don't think that's going to happen. You already said you'd be okay with `uid==0` being required for burning, if only the transition period were longer. So if people complain to you that `cdrecord` is broken with 2.6.8, you will have to tell them burning requires root for the moment. Then in your next release, change your startup code not to drop the `CAP_SYS_RAWIO` capability when you drop root privileges.

Alternatively, provide a patch that changes the current code to just require write permission or `CAP_SYS_RAWIO` to be able to send arbitrary commands. Then, after a transition period, submit a patch that changes it to just `CAP_SYS_RAWIO`. The patch would look like the one below (untested).

Jens, since this seems to be your code, what do you think?

```
--- scsi_ioctl.c 2004-08-14 18:26:17.000000000 +0200
+++ scsi_ioctl.c.new 2004-08-22 15:08:36.000000000 +0200
@@ -105,70 +105,12 @@ static int sg_emulated_host(request_queue_t *q)
     return put_user(1, p);
 }

-#define CMD_READ_SAFE 0x01
-#define CMD_WRITE_SAFE 0x02
-#define safe_for_read(cmd) [cmd] = CMD_READ_SAFE
-#define safe_for_write(cmd) [cmd] = CMD_WRITE_SAFE
-
-static int verify_command(struct file *file, unsigned char *cmd)
+static int verify_command(struct file *file)
 {
- static const unsigned char cmd_type[256] = {
-
- /* Basic read-only commands */
- safe_for_read(TEST_UNIT_READY),
- safe_for_read(REQUEST_SENSE),
- safe_for_read(READ_6),
- safe_for_read(READ_10),
- safe_for_read(READ_12),
- safe_for_read(READ_16),
- safe_for_read(READ_BUFFER),
- safe_for_read(READ_LONG),
- safe_for_read(INQUIRY),
- safe_for_read(MODE_SENSE),

```

## Linux–Kernel: Re: PATCH: cdrecord: avoiding scsi device numbering for ide devices

```
– safe_for_read(MODE_SENSE_10),
– safe_for_read(START_STOP),
–
– /* Audio CD commands */
– safe_for_read(GPCMD_PLAY_CD),
– safe_for_read(GPCMD_PLAY_AUDIO_10),
– safe_for_read(GPCMD_PLAY_AUDIO_MSF),
– safe_for_read(GPCMD_PLAY_AUDIO_TI),
–
– /* CD/DVD data reading */
– safe_for_read(GPCMD_READ_CD),
– safe_for_read(GPCMD_READ_CD_MSF),
– safe_for_read(GPCMD_READ_DISC_INFO),
– safe_for_read(GPCMD_READ_CDVD_CAPACITY),
– safe_for_read(GPCMD_READ_DVD_STRUCTURE),
– safe_for_read(GPCMD_READ_HEADER),
– safe_for_read(GPCMD_READ_TRACK_RZONE_INFO),
– safe_for_read(GPCMD_READ_SUBCHANNEL),
– safe_for_read(GPCMD_READ_TOC_PMA_ATIP),
– safe_for_read(GPCMD_REPORT_KEY),
– safe_for_read(GPCMD_SCAN),
–
– /* Basic writing commands */
– safe_for_write(WRITE_6),
– safe_for_write(WRITE_10),
– safe_for_write(WRITE_VERIFY),
– safe_for_write(WRITE_12),
– safe_for_write(WRITE_VERIFY_12),
– safe_for_write(WRITE_16),
– safe_for_write(WRITE_BUFFER),
– safe_for_write(WRITE_LONG),
– };
– unsigned char type = cmd_type[cmd[0]];
–
– /* Anybody who can open the device can do a read–safe command */
– if (type & CMD_READ_SAFE)
+ /* write access means being able to send any command (for now) */
+ if (file->f_mode & FMODE_WRITE)
    return 0;

– /* Write–safe commands just require a writable open.. */
– if (type & CMD_WRITE_SAFE) {
– if (file->f_mode & FMODE_WRITE)
– return 0;
– }
–
    /* And root can do any command.. */
    if (capable(CAP_SYS_RAWIO))
        return 0;
@@ –181,7 +123,7 @@ static int sg_io(struct file *file, requ
    struct gendisk *bd_disk, struct sg_io_hdr *hdr)
```

Linux-Kernel: Re: PATCH: cdrecord: avoiding scsi device numbering for ide devices

```
{
    unsigned long start_time;
- int reading, writing;
+ int reading, writing, res;
    struct request *rq;
    struct bio *bio;
    char sense[SCSI_SENSE_BUFFERSIZE];
@@ -193,8 +135,8 @@ static int sg_io(struct file *file, requ
    return -EINVAL;
    if (copy_from_user(cmd, hdr->cmdp, hdr->cmd_len))
        return -EFAULT;
- if (verify_command(file, cmd))
- return -EPERM;
+ if (res = verify_command(file))
+ return res;

    /*
     * we'll do that later

--
Ciao,
Pascal
-
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/
```