

Re: PROBLEM: null pointer dereference in cfq_dispatch_requests (2.6.21-rc2 and 2.6.20)

Source: <http://linux.derkeiler.com/Mailing-Lists/Kernel/2007-02/msg10163.html>

- *From:* Chuck Ebbert <cebbert@xxxxxxxxxx>
 - *Date:* Wed, 28 Feb 2007 13:49:02 -0500
-

Dan Williams wrote:

I can reliably reproduce a null pointer dereference on 2.6.20 and 2.6.21-rc2. I will keep digging to find the kernel version where this last worked, but wanted to see if there were any immediate experiments I should try.

The failure is caused by running tiobench on a MD raid6 array with 6 out of 8 disks available. The commands I issued to reproduce this are:

```
mdadm -A /dev/md0 /dev/sd[bcddefg]
mount /dev/md0 /mnt/raid
tiobench --numruns 5 --size 2048 --dir /mnt/raid
```

The filesystem is ext3. The controller is an LSI 1068. Here are the two BUG messages first 2.6.21-rc2 followed by 2.6.20. I will reply to this message with the config.

Kernel 2.6.20 on an i686

```
[ 177.299787] BUG: unable to handle kernel NULL pointer dereference at virtual address
0000005c
[ 177.308526] printing eip:
[ 177.311287] c01de510
[ 177.313521] *pde = 34d40001
[ 177.316353] Oops: 0000 [#1]
[ 177.319202] SMP
[ 177.321107] Modules linked in: raid456 xor nfsd exportfs lockd nfs_acl sunrpc autofs4 hidp
l2cap bluetooth iptable_raw xt_policy xt_multiport ipt_ULOG ipt_TTL ipt_ttl ipt_TOS
ipt_tos ipt_SAME ipt_REJECT ipt_REDIRECT ipt_recent ipt_owner ipt_NETMAP
ipt_MASQUERADE ipt_LOG ipt_iprange ipt_ECN ipt_ecn ipt_CLUSTERIP ipt_ah
ipt_addrtype xt_tcpmss xt_pkttype xt_physdev xt_NFQUEUE xt_MARK xt_mark xt_mac
xt_limit xt_length xt_helper xt_dccp xt_contrack xt_CONNMARK xt_connmark
xt_CLASSIFY xt_tcpudp xt_state iptable_nat nf_nat nf_contrack_ipv4 nf_contrack
iptable_mangle nfnetlink iptable_filter ip_tables x_tables video sbs i2c_ec dock button
battery asus_acpi ac radeon drm ipv6 lp parport_pc parport e1000 uhci_hcd floppy mptsas
mptscsih mptbase sg ehci_hcd scsi_transport_sas i2c_i801 i2c_core pcpkr dm_snapshot
dm_zero dm_mirror dm_mod ata_piix ata_generic libata sd_mod scsi_mod ext3 jbd
[ 177.402252] CPU: 2
```

Re: PROBLEM: null pointer dereference in cfq_dispatch_requests (2.6.21-rc2 and 2.6.20)

```
[ 177.402253] EIP: 0060:[<c01de510>] Not tainted VLI
[ 177.402253] EFLAGS: 00210016 (2.6.20 #5)
[ 177.414194] EIP is at cfq_dispatch_insert+0xb/0x53
[ 177.419056] eax: f7773ec0 ebx: 00000000 ecx: f7773cc0 edx: 00000000
[ 177.425982] esi: f70abae0 edi: f7773cc0 ebp: 00000000 esp: f34dbcbc
[ 177.432953] ds: 007b es: 007b ss: 0068
[ 177.437127] Process tiotest (pid: 5405, ti=f34db000 task=f7efc030 task.ti=f34db000)
[ 177.444763] Stack: 00000049 f77d3b9c f7773cc0 00000000 c01de6ce c014041e f8a26806
00000082
[ 177.453456] f7efc030 fffe22d6 00000000 00000000 00000000 00000004 f7efc030
f7773cc0
[ 177.462121] 00000000 00000000 00000000 f70abae0 f7cd5800 f70abae0 c01d4fcc
00000001
[ 177.470798] Call Trace:
[ 177.473503] [<c01de6ce>] cfq_dispatch_requests+0x12d/0x466
[ 177.479223] [<c014041e>] __lock_acquire+0x9e9/0xa72
[ 177.484285] [<f8a26806>] scsi_request_fn+0x286/0x336 [scsi_mod]
[ 177.490485] [<c01d4fcc>] elv_next_request+0x1a2/0x1b2
[ 177.495766] [<f8a26806>] scsi_request_fn+0x286/0x336 [scsi_mod]
[ 177.501912] [<c0315ba8>] _spin_lock_irq+0x38/0x43
[ 177.506840] [<f8a265d9>] scsi_request_fn+0x59/0x336 [scsi_mod]
[ 177.512981] [<c01d7e7d>] blk_remove_plug+0x5a/0x66
[ 177.517983] [<c01d7ea6>] __generic_unplug_device+0x1d/0x1f
[ 177.523705] [<c01d8278>] generic_unplug_device+0x15/0x21
[ 177.529272] [<f97ee054>] unplug_slaves+0x54/0x88 [raid456]
[ 177.535013] [<c01d997a>] blk_backing_dev_unplug+0x73/0x7b
[ 177.540657] [<c0315d82>] _spin_unlock_irqrestore+0x3e/0x4d
[ 177.546382] [<c0154b26>] sync_page+0x0/0x3b
[ 177.550774] [<c013f5f4>] trace_hardirqs_on+0x12e/0x158
[ 177.556108] [<c0154b26>] sync_page+0x0/0x3b
[ 177.560471] [<c018caa5>] block_sync_page+0x31/0x32
[ 177.565449] [<c0154b59>] sync_page+0x33/0x3b
[ 177.569916] [<c0313d9e>] __wait_on_bit_lock+0x2a/0x52
[ 177.575201] [<c0154b18>] __lock_page+0x58/0x5e
[ 177.579810] [<c0139612>] wake_bit_function+0x0/0x3c
[ 177.584905] [<c0155228>] do_generic_mapping_read+0x1db/0x44f
[ 177.590911] [<c01570cb>] generic_file_aio_read+0x173/0x1a4
[ 177.596617] [<c0154930>] file_read_actor+0x0/0xdb
[ 177.601525] [<c0171b47>] do_sync_read+0xc7/0x10a
[ 177.606365] [<c01395dd>] autoremove_wake_function+0x0/0x35
[ 177.612130] [<c0171a80>] do_sync_read+0x0/0x10a
[ 177.616867] [<c01723ce>] vfs_read+0xa6/0x152
[ 177.621362] [<c0172830>] sys_read+0x41/0x67
[ 177.625794] [<c0103e24>] syscall_call+0x7/0xb
[ 177.630403] =====
[ 177.634031] Code: da 11 3b c0 c7 04 24 51 9d 39 c0 e8 c9 a1 f4 ff e8 ca 6e f2 ff ff 4f 34 83
c4 18 5b 5e 5f 5d c3 55 57 56 89 c6 53 8b 40 0c 89 d3 <8b> 7a 5c 8b 68 04 89 d0 e8 b5 fe ff
ff 8b 43 14 89 da 25 01 80
[ 177.654378] EIP: [<c01de510>] cfq_dispatch_insert+0xb/0x53 SS:ESP 0068:f34dbcbc
```

Re: PROBLEM: null pointer dereference in cfq_dispatch_requests (2.6.21-rc2 and 2.6.20)

Re: PROBLEM: null pointer dereference in cfq_dispatch_requests (2.6.21-rc2 and 2.6.20)

cfq_dispatch_requests() has called cfq_dispatch_insert() with a NULL second argument (struct request *rq)

There are two patches for raid5/6 out there that might fix this. I'll attach them (the second just fixes a minor bug in the first one.)

From: Neil Brown <neilb@xxxxxxx>

On Sunday February 11, marcm@xxxxxxxxxxxxxxxxx wrote:

Greetings,

I've been running md on my server for some time now and a few days ago one of the (3) drives in the raid5 array starting giving read errors. The result was usually system hangs and this was with kernel 2.6.17.13. I upgraded to the latest production 2.6.20 kernel and experienced the same behaviour.

System hangs suggest a problem with the drive controller. However this "kernel BUG" is something newly introduced in 2.6.20 which should be fixed in 2.6.20.1. Patch is below.

If you still get hangs with this patch installed, then please report detail, and probably copy to linux-ide@xxxxxxxxxxxxxxxxx

NeilBrown

Fix various bugs with aligned reads in RAID5.

It is possible for raid5 to be sent a bio that is too big for an underlying device. So if it is a READ that we pass straight down to a device, it will fail and confuse RAID5.

So in 'chunk_aligned_read' we check that the bio fits within the parameters for the target device and if it doesn't fit, fall back on reading through the stripe cache and making lots of one-page requests.

Note that this is the earliest time we can check against the device because earlier we don't have a lock on the device, so it could change underneath us.

Also, the code for handling a retry through the cache when a read fails has not been tested and was badly broken. This patch fixes that code.

Signed-off-by: Neil Brown <neilb@xxxxxxx>

Re: PROBLEM: null pointer dereference in cfq_dispatch_requests (2.6.21-rc2 and 2.6.20)

Diffstat output

./drivers/md/raid5.c | 42 ++++++-----
1 file changed, 39 insertions(+), 3 deletions(-)

diff .prev/drivers/md/raid5.c ./drivers/md/raid5.c

Index: linux-2.6.20.noarch/drivers/md/raid5.c

```
=====
--- linux-2.6.20.noarch.orig/drivers/md/raid5.c 2007-02-04 13:44:54.000000000 -0500
+++ linux-2.6.20.noarch/drivers/md/raid5.c 2007-02-18 18:57:04.000000000 -0500
@@ -2620,7 +2620,7 @@
 }
 bi = conf->retry_read_aligned_list;
 if(bi) {
- conf->retry_read_aligned = bi->bi_next;
+ conf->retry_read_aligned_list = bi->bi_next;
 bi->bi_next = NULL;
 bi->bi_phys_segments = 1; /* biased count of active stripes */
 bi->bi_hw_segments = 0; /* count of processed stripes */
@@ -2669,6 +2669,27 @@
 return 0;
 }

+static int bio_fits_rdev(struct bio *bi)
+{
+ request_queue_t *q = bdev_get_queue(bi->bi_bdev);
+
+ if ((bi->bi_size >> 9) > q->max_sectors)
+ return 0;
+ blk_recount_segments(q, bi);
+ if (bi->bi_phys_segments > q->max_phys_segments ||
+ bi->bi_hw_segments > q->max_hw_segments)
+ return 0;
+
+ if (q->merge_bvec_fn)
+ /* it's too hard to apply the merge_bvec_fn at this stage,
+ * just just give up
+ */
+ return 0;
+
+ return 1;
+}
+
+static int chunk_aligned_read(request_queue_t *q, struct bio *raid_bio)
+{
+ mddev_t *mddev = q->queuedata;
@@ -2715,6 +2736,13 @@
 align_bi->bi_flags &= ~(1 << BIO_SEG_VALID);
 align_bi->bi_sector += rdev->data_offset;
```

Re: PROBLEM: null pointer dereference in cfq_dispatch_requests (2.6.21-rc2 and 2.6.20)

```
+ if (!bio_fits_rdev(align_bi)) {
+ /* too big in some way */
+ bio_put(align_bi);
+ rdev_dec_pending(rdev, mddev);
+ return 0;
+ }
+
spin_lock_irq(&conf->device_lock);
wait_event_lock_irq(conf->wait_for_stripe,
conf->quiesce == 0,
@@ -3107,7 +3135,9 @@
last_sector = raid_bio->bi_sector + (raid_bio->bi_size>>9);

for (; logical_sector < last_sector;
- logical_sector += STRIPE_SECTORS, scnt++) {
+ logical_sector += STRIPE_SECTORS,
+ sector += STRIPE_SECTORS,
+ scnt++) {

if (scnt < raid_bio->bi_hw_segments)
/* already done this stripe */
@@ -3123,7 +3153,13 @@
}

set_bit(R5_ReadError, &sh->dev[dd_idx].flags);
- add_stripe_bio(sh, raid_bio, dd_idx, 0);
+ if (!add_stripe_bio(sh, raid_bio, dd_idx, 0)) {
+ release_stripe(sh);
+ raid_bio->bi_hw_segments = scnt;
+ conf->retry_read_aligned = raid_bio;
+ return handled;
+ }
+
handle_stripe(sh, NULL);
release_stripe(sh);
handled++;
From: Neil Brown <neilb@xxxxxxx>
```

On Monday February 12, marcm@xxxxxxxxxxxxxxxxxxx wrote:

Thanks for the quick response Neil unfortunately the kernel doesn't build with this patch due to a missing symbol:

WARNING: "blk_recount_segments" [drivers/md/raid456.ko] undefined!

Is that in another file that needs patching or within raid5.c?

Yes. I keep forgetting about that bit. Sorry.

Re: PROBLEM: null pointer dereference in cfq_dispatch_requests (2.6.21-rc2 and 2.6.20)

Re: PROBLEM: null pointer dereference in cfq_dispatch_requests (2.6.21-rc2 and 2.6.20)

Signed-off-by: Neil Brown <neilb@xxxxxxx>

Index: linux-2.6.20.noarch/block/l1_rw_blk.c

=====
--- linux-2.6.20.noarch.orig/block/l1_rw_blk.c 2007-02-04 13:44:54.000000000 -0500

+++ linux-2.6.20.noarch/block/l1_rw_blk.c 2007-02-18 18:57:04.000000000 -0500

@@ -1264,7 +1264,7 @@

bio->bi_hw_segments = nr_hw_segs;

bio->bi_flags |= (1 << BIO_SEG_VALID);

}

-

+EXPORT_SYMBOL(blk_recount_segments);

static int blk_phys_contig_segment(request_queue_t *q, struct bio *bio,
struct bio *nxt)