



author Andrew Price <anprice@redhat.com> 2022-08-17 13:22:00 +0100
committer Greg Kroah-Hartman <gregkh@linuxfoundation.org> 2022-11-25 17:42:21 +0100
commit [8b6534c9ae9dba5489703a19d8ba6c8f2cfa33c2](#) (patch)
tree [a0e1d9ccede95cb8526bac679e445296918742ac](#)
parent [96760723aae1b45f733f702abb4333137143909f](#) (diff)
download [linux-8b6534c9ae9dba5489703a19d8ba6c8f2cfa33c2.tar.gz](#)

diff options

context: ▼
space: ▼
mode: ▼

gfs2: Check sb_bsize_shift after reading superblock

commit 670f8ce56dd0632dc29a0322e188cc73ce3c6b92 upstream.

Fuzzers like to scribble over sb_bsize_shift but in reality it's very unlikely that this field would be corrupted on its own. Nevertheless it should be checked to avoid the possibility of messy mount errors due to bad calculations. It's always a fixed value based on the block size so we can just check that it's the expected value.

Tested with:

```
mkfs.gfs2 -O -p lock_nolock /dev/vdb
for i in 0 -1 64 65 32 33; do
    gfs2_edit -p sb field sb_bsize_shift $i /dev/vdb
    mount /dev/vdb /mnt/test && umount /mnt/test
done
```

Before this patch we get a withdraw after

```
[ 76.413681] gfs2: fsid=loop0.0: fatal: invalid metadata block
[ 76.413681] bh = 19 (type: exp=5, found=4)
[ 76.413681] function = gfs2_meta_buffer, file = fs/gfs2/meta_io.c, line = 492
```

and with UBSAN configured we also get complaints like

```
[ 76.373395] UBSAN: shift-out-of-bounds in fs/gfs2/ops_fstype.c:295:19
[ 76.373815] shift exponent 4294967287 is too large for 64-bit type 'long unsigned int'
```

After the patch, these complaints don't appear, mount fails immediately and we get an explanation in dmesg.

Reported-by: syzbot+dcf33a7aae997956fe06@syzkaller.appspotmail.com

Signed-off-by: Andrew Price <anprice@redhat.com>

Signed-off-by: Andreas Gruenbacher <agruenba@redhat.com>

Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

Diffstat

```
-rw-r--r-- fs/gfs2/ops_fstype.c 5
```

1 files changed, 4 insertions, 1 deletions

diff --git a/fs/gfs2/ops_fstype.c b/fs/gfs2/ops_fstype.c
index 29b27d769860cd..36aed0fc5650eb 100644

```
--- a/fs/gfs2/ops_fstype.c
+++ b/fs/gfs2/ops_fstype.c
@@ -180,7 +180,10 @@ static int gfs2_check_sb(struct gfs2_sbd *sdp, int silent)
        pr_warn("Invalid superblock size\n");
        return -EINVAL;
    }
-
+    if (sb->sb_bsize_shift != ffs(sb->sb_bsize) - 1) {
+        pr_warn("Invalid block size shift\n");
+        return -EINVAL;
+    }
    return 0;
}
```