linuxfoundation.org

Linux Kernel Bug Fixing Mentorship Hackathon 2023
Team - syzBuzz

Atul Raut, Rajeshwar Shinde, Vincent Palazzo

14th – 21st of August 2023

Bug - UBSAN: array-index-out-of-bounds in truncate_inode_pages_final

Link - https://syzkaller.appspot.com/bug?extid=e295147e14b474e4ad70

Bug - UBSAN: array-index-out-of-bounds in truncate_inode_pages_final

Sample crash report:

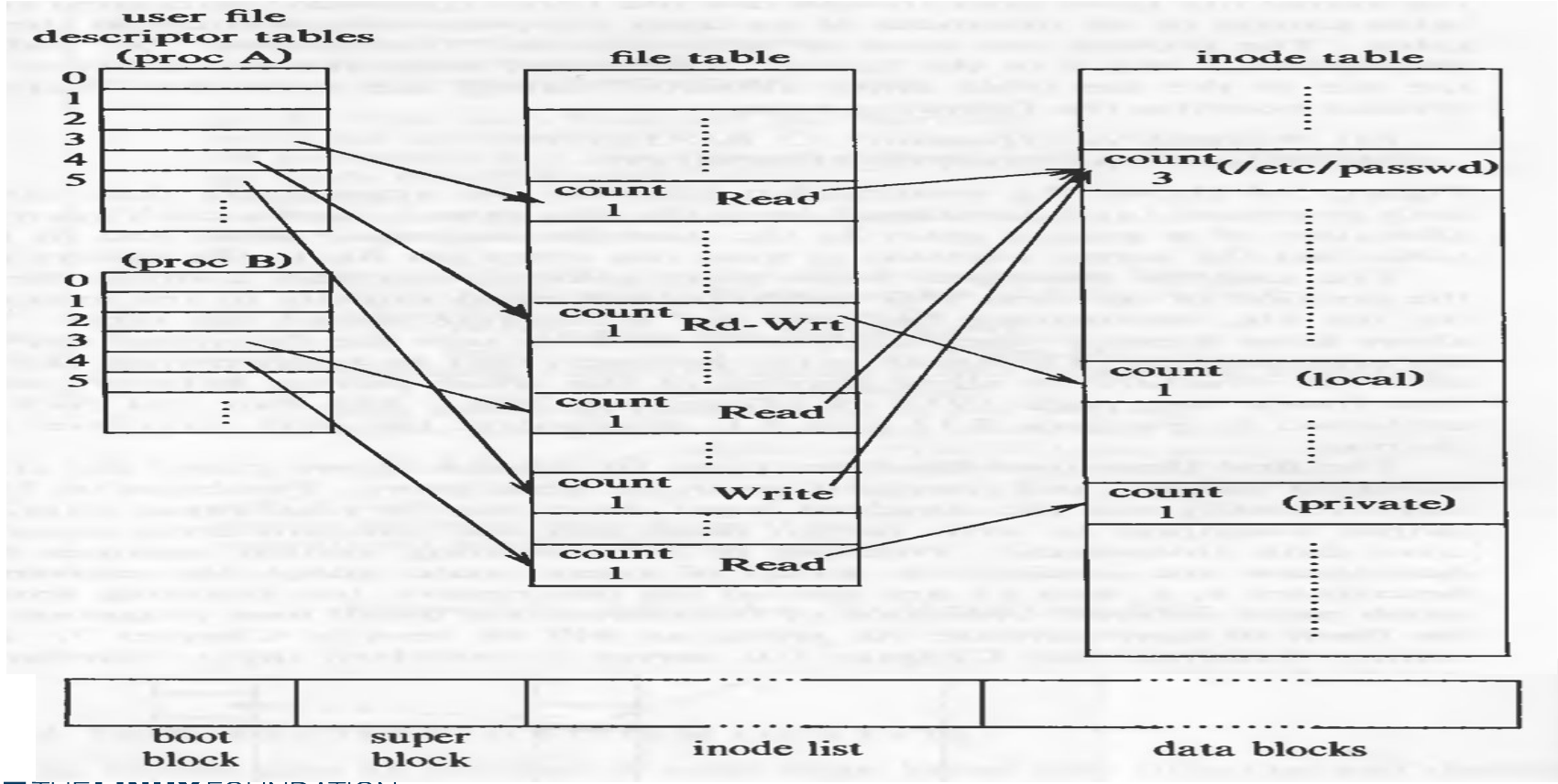
```
================================================================================
UBSAN: array-index-out-of-bounds in ./include/linux/pagevec.h:74:2
index 255 is out of range for type 'struct folio *[15]'
CPU: 1 PID: 12841 Comm: syz-executor402 Not tainted 6.5.0-rc7-syzkaller-g35e2132122ba #0
Hardware name: Google Google Compute Engine/Google Compute Engine, BIOS Google 07/26/2023
Call trace:
 dump_backtrace+0x1b8/0x1e4 arch/arm64/kernel/stacktrace.c:233
 show_stack+0x2c/0x44 arch/arm64/kernel/stacktrace.c:240
 __dump_stack lib/dump_stack.c:88 [inline]
 dump_stack_lvl+0xd0/0x124 lib/dump_stack.c:106
 dump_stack+0x1c/0x28 lib/dump_stack.c:113
 ubsan_epilogue lib/ubsan.c:217 [inline]
 __ubsan_handle_out_of_bounds+0xfc/0x148 lib/ubsan.c:348
 folio_batch_add include/linux/pagevec.h:74 [inline]
 find_lock_entries+0x8fc/0xd84 mm/filemap.c:2089
 truncate_inode_pages_range+0x1b0/0xf74 mm/truncate.c:364
 truncate_inode_pages mm/truncate.c:449 [inline]
 truncate_inode_pages_final+0x90/0xc0 mm/truncate.c:484
 ntfs_evict_inode+0x20/0x48 fs/ntfs3/inode.c:1790
 evict+0x260/0x68c fs/inode.c:664
 iput_final fs/inode.c:1788 [inline]
 iput+0x734/0x818 fs/inode.c:1814
 ntfs_fill_super+0x3648/0x3f90 fs/ntfs3/super.c:1420
 get_tree_bdev+0x378/0x570 fs/super.c:1318
 ntfs_fs_get_tree+0x28/0x38 fs/ntfs3/super.c:1647
```

Bug - UBSAN: array-index-out-of-bounds in truncate_inode_pages_final

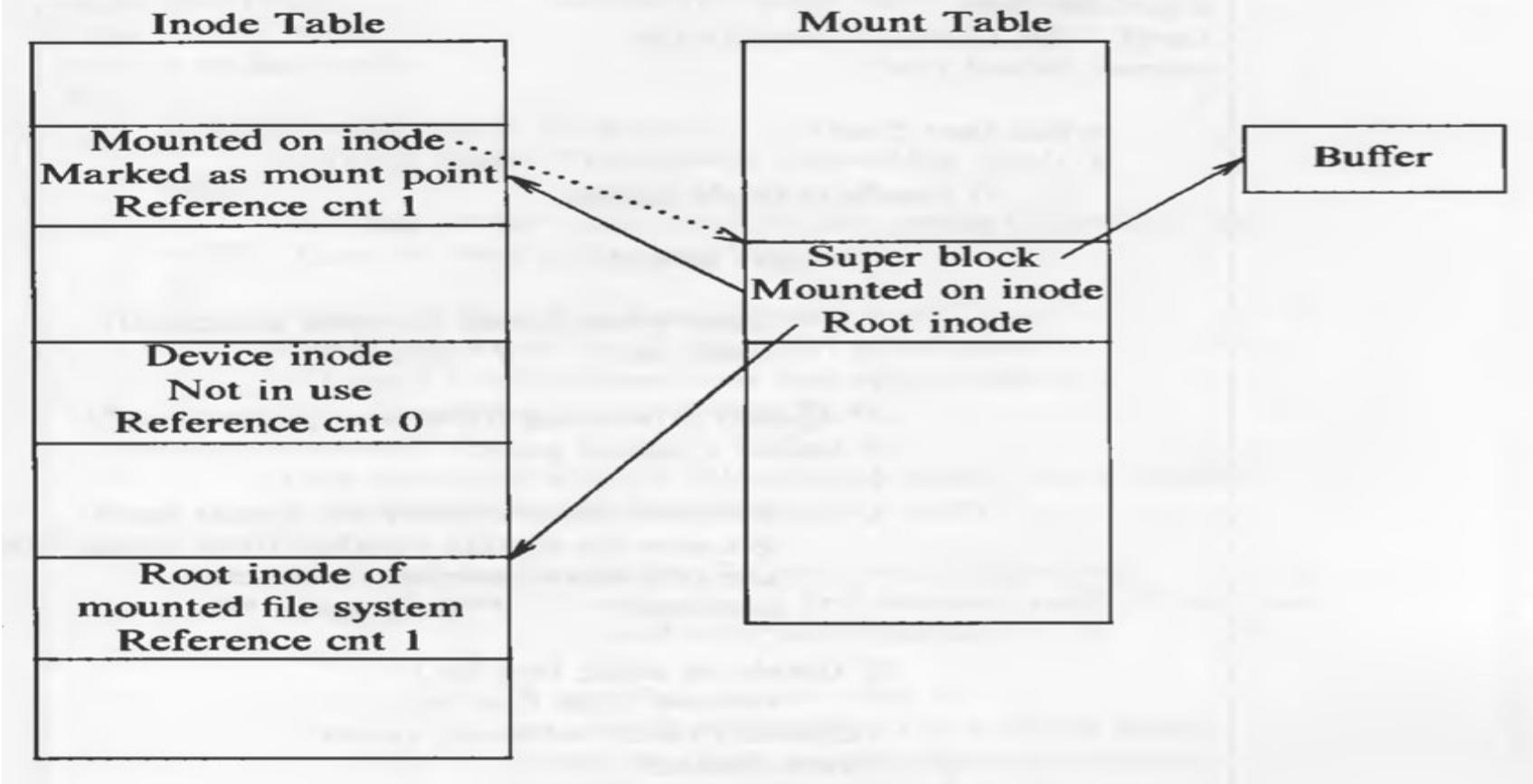
Debugging techniques employed -

1. Recognize the subsystem introduction issue
2. Recognize broader changes surrounding the issue.
3. Find out what changes cause the bug.

Bug - UBSAN: array-index-out-of-bounds in truncate_inode_pages_final



Bug - UBSAN: array-index-out-of-bounds in truncate_inode_pages_final



Bug - UBSAN: array-index-out-of-bounds in truncate_inode_pages_final

Background about mm/filemap.c -

- This file handles the generic file mmap semantics used by most "*normal*" filesystems.
- The Implementation in this file, used by the filesystems and the page cache to manage memory in larger chunks than PAGE_SIZE
- Introduce 5.15 onwards, Clarifying memory management with page folios
- The folio type itself is defined as a simple wrapper structure:

```
struct folio {
    struct page page;
};
```

Bug - UBSAN: array-index-out-of-bounds in truncate_inode_pages_final

Background about struct folio - why ?

- Our type system does not currently distinguish between tail pages and head or single pages. This is a problem because we call compound_head() multiple times (and the compiler cannot optimise it out), bloating the kernel.
- It also makes programming hard as it is often unclear whether a function operates on an individual page, or an entire compound page.
- This patch series introduces the struct folio, which is a type that represents an entire compound page.
- This initial set reduces the kernel size by approximately 6kB, although its real purpose is adding infrastructure to enable further use of the folio.
- The big correctness proof that exists in this patch series is that we never lock or wait for writeback on a tail page. This is important as we would miss wakeups due to being on the wrong page waitqueue if we ever did.

Bug - UBSAN: array-index-out-of-bounds in truncate_inode_pages_final

References - Clarifying memory management with page folios

https://lwn.net/Articles/849538/

https://lwn.net/ml/linux-kernel/20210305041901.2396498-1-willy@infradead.org/

Bug - UBSAN: array-index-out-of-bounds in truncate_inode_pages_final

!Fix -

Bug - UBSAN: array-index-out-of-bounds in truncate_inode_pages_final

SyzKaller Reproducer References -

https://syzkaller.appspot.com/text?tag=ReproC&x=12224553a80000 - C code

https://syzkaller.appspot.com/x/repro.syz?x=101c2da4a80000

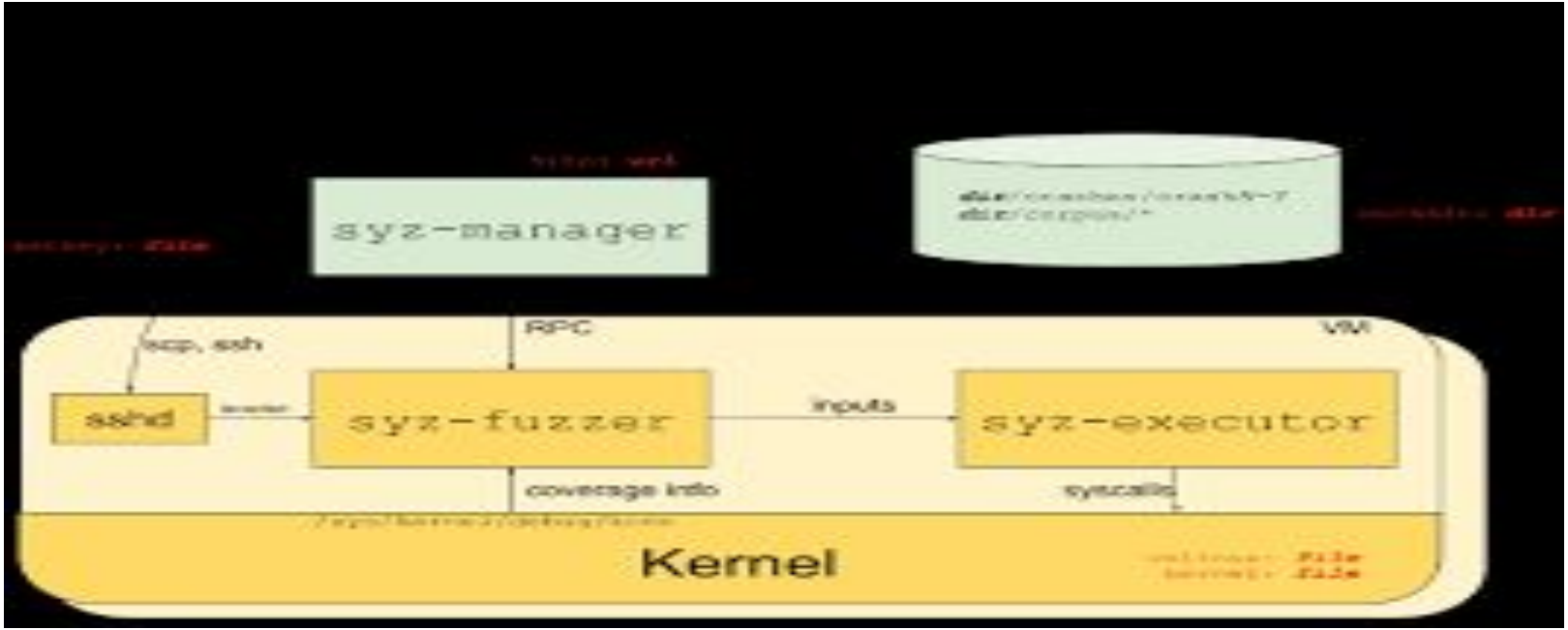
https://github.com/google/syzkaller/blob/master/docs/syzbot.md#syzkaller-reproducers

https://github.com/google/syzkaller/blob/49be837e029feccab241a98641b01a146890b66f/executor/common_linux.h#L3007

https://github.com/search?q=repo%3Agoogle%2Fsyzkaller%20syz_mount_image&type=code

https://github.com/google/syzkaller/blob/49be837e029feccab241a98641b01a146890b66f/pkg/subsystem/linux/rules.go#L51

Syzkaller Learning



syzkaller Learning - 1

Issue - memory leak in skb_copy (2)

Ref - https://syzkaller.appspot.com/bug?extid=6eb09d75211863f15e3e

syzkaller Learning - 1

Sample crash report:

```
BUG: memory leak
unreferenced object 0xffff88811fff5e00 (size 240):
  comm "kworker/u4:0", pid 10, jiffies 4294989700 (age 28.220s)
  hex dump (first 32 bytes):
    00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
    00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
  backtrace:
    [<ffffffff83e1c0bd>] __alloc_skb+0x1fd/0x230 net/core/skbuff.c:634
    [<ffffffff83e1efcf>] skb_copy+0x6f/0x180 net/core/skbuff.c:1925
    [<ffffffff82c3526f>] virtual_nci_send+0x3f/0xb0 drivers/nfc/virtual_ncidev.c:58
    [<ffffffff84990da9>] nci_send_frame+0x69/0xb0 net/nfc/nci/core.c:1347
    [<ffffffff84990e82>] nci_cmd_work+0x92/0xc0 net/nfc/nci/core.c:1567
    [<ffffffff812b19e4>] process_one_work+0x2c4/0x620 kernel/workqueue.c:2597
    [<ffffffff812b233d>] worker_thread+0x5d/0x5c0 kernel/workqueue.c:2748
    [<ffffffff812bbde3>] kthread+0x133/0x180 kernel/kthread.c:389
    [<ffffffff81002b5f>] ret_from_fork+0x1f/0x30 arch/x86/entry/entry_64.S:308

BUG: memory leak
unreferenced object 0xffff88810d74e500 (size 640):
```

syzkaller Learning - 1

Issue - memory leak in skb_copy (2)

```
48  static int virtual_nci_send(struct nci_dev *ndev, struct sk_buff *skb)
49  {
50          struct virtual_nci_dev *vdev = nci_get_drvdata(ndev);
51
52          mutex_lock(&vdev->mtx);
53          if (vdev->send_buff) {
54                  mutex_unlock(&vdev->mtx);
55                  kfree_skb(skb);
56                  return -1;
57          }
58          vdev->send_buff = skb_copy(skb, GFP_KERNEL);
59          if (!vdev->send_buff) {
60                  mutex_unlock(&vdev->mtx);
61                  kfree_skb(skb);
62                  return -1;
63          }
64          mutex_unlock(&vdev->mtx);
65          wake_up_interruptible(&vdev->wq);
66          consume_skb(skb);
67
68          return 0;
69  }
```

syzkaller Learning - 2

A] Issue - Failed to ssh into qemu

Why - Didn't created ssh key

Fix - generate an SSH key pair, using ssh-keygen command

B] Issue - Failed to compile syzkaller

After firing make command, it used to exit in a minute.

Fix - Was using some older laptop/machine, switch to latest hardware.

syzkaller Learning - 3

Issue - Missing reproducer

BUG: unable to handle kernel NULL pointer dereference in __io_remove_buffers

e.g. https://syzkaller.appspot.com/bug?extid=70de24bf68bee5f644e3

Some of issues that syzkaller generated dont have reproducer

If we try to fix such issues, syzkaller failed to verify.

BUG: unable to handle kernel NULL pointer dereference in __io_remove_buffers

Status: moderation: reported on 2023/05/30 07:17
Labels: io-uring (incorrect?)
Reported-by: syzbot+70de24bf68bee5f644e3@syzkaller.appspotmail.com
First crash: 91d, last: 91d

► Similar bugs (1)

Sample crash report:

```
8<--- cut here ---
Unable to handle kernel NULL pointer dereference at virtual address 0000000e when read
[0000000e] *pgd=80000080004003, *pmd=00000000
Internal error: Oops: 207 [#1] PREEMPT SMP ARM
Modules linked in:
CPU: 0 PID: 20796 Comm: kworker/u4:0 Not tainted 6.4.0-rc3-syzkaller #0
Hardware name: ARM-Versatile Express
Workqueue: events_unbound io_ring_exit_work
PC is at __io_remove_buffers io_uring/kbuf.c:219 [inline]
PC is at __io_remove_buffers+0x38/0x184 io_uring/kbuf.c:209
LR is at io_destroy_buffers+0x40/0x134 io_uring/kbuf.c:268
pc : [<807aeeb8>]    lr : [<807af454>]    psr: 20000113
sp : dfb61e48  ip : dfb61e78  fp : dfb61e74
```

BUG 2
Sample crash report:

```
================================================================================
UBSAN: shift-out-of-bounds in drivers/media/usb/gspca/cpia1.c:1031:27
shift exponent 245 is too large for 32-bit type 'int'
CPU: 1 PID: 25 Comm: kworker/1:1 Not tainted 6.5.0-rc4-syzkaller-00118-g55c3e571d2a0 #0
Hardware name: Google Google Compute Engine/Google Compute Engine, BIOS Google 07/26/2023
Workqueue: usb_hub_wq hub_event
Call Trace:
 <TASK>
 __dump_stack lib/dump_stack.c:88 [inline]
 dump_stack_lvl+0x125/0x1b0 lib/dump_stack.c:106
 ubsan_epilogue lib/ubsan.c:217 [inline]
 __ubsan_handle_shift_out_of_bounds+0x27a/0x600 lib/ubsan.c:387
 set_flicker.cold+0x1b/0x20 drivers/media/usb/gspca/cpia1.c:1031
 sd_s_ctrl+0x2c6/0xbf0 drivers/media/usb/gspca/cpia1.c:1782
 __v4l2_ctrl_handler_setup+0x511/0x710 drivers/media/v4l2-core/v4l2-ctrls-core.c:2481
 v4l2_ctrl_handler_setup drivers/media/v4l2-core/v4l2-ctrls-core.c:2498 [inline]
 v4l2_ctrl_handler_setup+0x50/0xa0 drivers/media/v4l2-core/v4l2-ctrls-core.c:2490
 gspca_set_default_mode drivers/media/usb/gspca/gspca.c:908 [inline]
 gspca_dev_probe2+0xdd6/0x1b20 drivers/media/usb/gspca/gspca.c:1541
 gspca_dev_probe+0x18b/0x270 drivers/media/usb/gspca/gspca.c:1610
 usb_probe_interface+0x307/0x930 drivers/usb/core/driver.c:396
 kthread+0x33a/0x430 kernel/kthread.c:389
 ret_from_fork+0x2c/0x70 arch/x86/kernel/process.c:145
 ret_from_fork_asm+0x11/0x20 arch/x86/entry/entry_64.S:304
 </TASK>
================================================================================
```

What caused patch?

```
if (sd->params.exposure.expMode != 2) {
                        sd->params.exposure.expMode = 2;
                        sd->exposure_status = EXPOSURE_NORMAL;
                }
                currentexp = currentexp << sd->params.exposure.gain;
                sd->params.exposure.gain = 0;
                /* round down current exposure to nearest value */
```

How I fixed it.

```
--- a/drivers/media/usb/gspca/cpia1.c
+++ b/drivers/media/usb/gspca/cpia1.c
@@ -1028,6 +1028,8 @@ static int set_flicker(struct gspca_dev *gspca_dev, int on, int apply)
                        sd->params.exposure.expMode = 2;
                        sd->exposure_status = EXPOSURE_NORMAL;
                }
+               if (sd->params.exposure.gain > 31)
+                       return -EINVAL;
        currentexp = currentexp << sd->params.exposure.gain;
```

Thank You!