



author Edward Adam Davis <eadavis@qq.com> 2025-02-20 19:13:21 +0800
committer Greg Kroah-Hartman <gregkh@linuxfoundation.org> 2025-04-20 10:17:43 +0200
commit 994787341358816d91b2fded288ecb7f129f2b27 (patch)
tree 25104d69646266015b5e797cfd5cb442b8d4d1a4
parent 211ed8f5e39e61f9e4d18edd64ce8005a67a1b2a (diff)
download [linux-994787341358816d91b2fded288ecb7f129f2b27.tar.gz](#)

diff options

context: 3
space: include
mode: unified

jfs: Prevent copying of nlink with value 0 from disk inode

[Upstream commit b61e69bb1c049cf507e3c654fa3dc1568231bd07]

syzbot report a deadlock in diFree. [1]

When calling "ioctl\$LOOP_SET_STATUS64", the offset value passed in is 4, which does not match the mounted loop device, causing the mapping of the mounted loop device to be invalidated.

When creating the directory and creating the inode of iag in diReadSpecial(), read the page of fixed disk inode (AIT) in raw mode in read_metapage(), the metapage data it returns is corrupted, which causes the nlink value of 0 to be assigned to the iag inode when executing copy_from_dinode(), which ultimately causes a deadlock when entering diFree().

To avoid this, first check the nlink value of dinode before setting iag inode.

[1]

WARNING: possible recursive locking detected
6.12.0-rc7-syzkaller-00212-g4a5df3796467 #0 Not tainted

syz-executor301/5309 is trying to acquire lock:
ffff888044548920 (&(imap->im_aglock[index])){+..}-{3:3}, at: diFree+0x37c/0x2fb0 fs/jfs/jfs_imap.c:889

but task is already holding lock:
ffff888044548920 (&(imap->im_aglock[index])){+..}-{3:3}, at: diAlloc+0x1b6/0x1630

other info that might help us debug this:
Possible unsafe locking scenario:

CPU0

```
lock(&(imap->im_aglock[index]));  
lock(&(imap->im_aglock[index]));
```

*** DEADLOCK ***

May be due to missing lock nesting notation

5 locks held by syz-executor301/5309:

```
#0: ffff8880422a4420 (sb_writers#9){+..}-{0:0}, at: mnt_want_write+0x3f/0x90 fs/namespace.c:515  
#1: ffff88804755b390 (&type->i_mutex_dir_key#6/1){+..}-{3:3}, at: inode_lock_nested include/linux/fs.h:850 [inline]  
#1: ffff88804755b390 (&type->i_mutex_dir_key#6/1){+..}-{3:3}, at: filename_create+0x260/0x540 fs/namei.c:4026  
#2: ffff888044548920 (&(imap->im_aglock[index])){+..}-{3:3}, at: diAlloc+0x1b6/0x1630  
#3: ffff888044548890 (&imap->im_freelock){+..}-{3:3}, at: diNewIAG fs/jfs/jfs_imap.c:2460 [inline]  
#3: ffff888044548890 (&imap->im_freelock){+..}-{3:3}, at: diAllocExt fs/jfs/jfs_imap.c:1905 [inline]  
#3: ffff888044548890 (&imap->im_freelock){+..}-{3:3}, at: diAllocAG+0x4b7/0x1e50 fs/jfs/jfs_imap.c:1669  
#4: ffff88804755a618 (&jfs_ip->rdwrlock/1){++++}-{3:3}, at: diNewIAG fs/jfs/jfs_imap.c:2477 [inline]  
#4: ffff88804755a618 (&jfs_ip->rdwrlock/1){++++}-{3:3}, at: diAllocExt fs/jfs/jfs_imap.c:1905 [inline]  
#4: ffff88804755a618 (&jfs_ip->rdwrlock/1){++++}-{3:3}, at: diAllocAG+0x869/0x1e50 fs/jfs/jfs_imap.c:1669
```

stack backtrace:

CPU: 0 UID: 0 PID: 5309 Comm: syz-executor301 Not tainted 6.12.0-rc7-syzkaller-00212-g4a5df3796467 #0
Hardware name: QEMU Standard PC (Q35 + ICH9, 2009), BIOS 1.16.3-debian-1.16.3-2~bpo12+1 04/01/2014
Call Trace:

```
<TASK>
__dump_stack lib/dump_stack.c:94 [inline]
dump_stack_lvl+0x241/0x360 lib/dump_stack.c:120
print_deadlock_bug+0x483/0x620 kernel/locking/lockdep.c:3037
check_deadlock kernel/locking/lockdep.c:3089 [inline]
validate_chain+0x15e2/0x5920 kernel/locking/lockdep.c:3891
__lock_acquire+0x1384/0x2050 kernel/locking/lockdep.c:5202
lock_acquire+0x1ed/0x550 kernel/locking/lockdep.c:5825
__mutex_lock_common kernel/locking/mutex.c:608 [inline]
__mutex_lock+0x136/0xd70 kernel/locking/mutex.c:752
diFree+0x37c/0x2fb0 fs/jfs/jfs_imap.c:889
jfs_evict_inode+0x32d/0x440 fs/jfs/inode.c:156
evict+0x4e8/0x9b0 fs/inode.c:725
diFreeSpecial fs/jfs/jfs_imap.c:552 [inline]
duplicateIXtree+0x3c6/0x550 fs/jfs/jfs_imap.c:3022
diNewIAG fs/jfs/jfs_imap.c:2597 [inline]
diAllocExt fs/jfs/jfs_imap.c:1905 [inline]
diAllocAG+0x17dc/0x1e50 fs/jfs/jfs_imap.c:1669
diAlloc+0x1d2/0x1630 fs/jfs/jfs_imap.c:1590
ialloc+0x8f/0x900 fs/jfs/jfs_inode.c:56
jfs_mkdir+0x1c5/0xba0 fs/jfs/namei.c:225
vfs_mkdir+0x2f9/0x4f0 fs/namei.c:4257
do_mkdirat+0x264/0x3a0 fs/namei.c:4280
__do_sys_mkdirat fs/namei.c:4295 [inline]
__se_sys_mkdirat fs/namei.c:4293 [inline]
__x64_sys_mkdirat+0x87/0xa0 fs/namei.c:4293
do_syscall_x64 arch/x86/entry/common.c:52 [inline]
do_syscall_64+0xf3/0x230 arch/x86/entry/common.c:83
entry_SYSCALL_64_after_hwframe+0x77/0x7f
```

Reported-by: syzbot+355da3b3a74881008e8f@syzkaller.appspotmail.com
Closes: <https://syzkaller.appspot.com/bug?extid=355da3b3a74881008e8f>
Signed-off-by: Edward Adam Davis <eadavis@qq.com>
Signed-off-by: Dave Kleikamp <dave.kleikamp@oracle.com>
Signed-off-by: Sasha Levin <sashal@kernel.org>

Diffstat

```
-rw-r--r-- fs/jfs/jfs_imap.c 2
```

1 files changed, 1 insertions, 1 deletions

```
diff --git a/fs/jfs/jfs_imap.c b/fs/jfs/jfs_imap.c
index cf16655cd26ba9..8ddc14c56501ac 100644
--- a/fs/jfs/jfs_imap.c
+++ b/fs/jfs/jfs_imap.c
@@ -456,7 +456,7 @@ struct inode *diReadSpecial(struct super_block *sb, ino_t inum, int secondary)
     dp += inum % 8;          /* 8 inodes per 4K page */

     /* copy on-disk inode to in-memory inode */
-    if ((copy_from_dinode(dp, ip)) != 0) {
+    if ((copy_from_dinode(dp, ip)) != 0) || (ip->i_nlink == 0) {
         /* handle bad return by returning NULL for ip */
         set_nlink(ip, 1);      /* Don't want iput() deleting it */
         iput(ip);
```