author	Jan Kara <jack@suse.cz>	2023-01-26 12:22:21 +0100
committer	Greg Kroah-Hartman <gregkh@linuxfoundation.org>	2023-03-17 08:50:21 +0100
commit	b0bb13612292ca90fa4c2a7e425375649bc50d3e (patch)	
tree	d088ab94bc86ee4d97cdb238c48b1ed41776e838	
parent	17e98a5ede81b7696bec421f7afa2dfe467f5e6b (diff)	
download	linux-b0bb13612292ca90fa4c2a7e425375649bc50d3e.tar.gz	

diff options

context: 3

space: include

mode: unified

ext4: Fix possible corruption when moving a directory

[Upstream commit 0813299c586b175d7edb25f56412c54b812d0379]

When we are renaming a directory to a different directory, we need to
update '..' entry in the moved directory. However nothing prevents moved
directory from being modified and even converted from the inline format
to the normal format. When such race happens the rename code gets
confused and we crash. Fix the problem by locking the moved directory.

CC: stable@vger.kernel.org
Fixes: 32f7f22c0b52 ("ext4: let ext4_rename handle inline dir")
Signed-off-by: Jan Kara <jack@suse.cz>
Link: https://lore.kernel.org/r/20230126112221.11866-1-jack@suse.cz
Signed-off-by: Theodore Ts'o <tytso@mit.edu>
Signed-off-by: Sasha Levin <sashal@kernel.org>

Diffstat

-rw-r--r-- fs/ext4/namei.c 11

1 files changed, 10 insertions, 1 deletions

```
diff --git a/fs/ext4/namei.c b/fs/ext4/namei.c
index 9799ed2fdbc09e..dc8f8a435a7ea4 100644
--- a/fs/ext4/namei.c
+++ b/fs/ext4/namei.c
@@ -3873,9 +3873,16 @@ static int ext4_rename(struct user_namespace *mnt_userns, struct inode *old_dir,
                        if (new.dir != old.dir && EXT4_DIR_LINK_MAX(new.dir))
                                goto end_rename;
                }
+               /*
+                * We need to protect against old.inode directory getting
+                * converted from inline directory format into a normal one.
+                */
+               inode_lock_nested(old.inode, I_MUTEX_NONDIR2);
                retval = ext4_rename_dir_prepare(handle, &old);
-               if (retval)
+               if (retval) {
+                       inode_unlock(old.inode);
                        goto end_rename;
+               }
        }
        /*
         * If we're renaming a file within an inline_data dir and adding or
@@ -4007,6 +4014,8 @@ end_rename:
        } else {
                ext4_journal_stop(handle);
        }
```

```
+       if (old.dir_bh)
+               inode_unlock(old.inode);
 release_bh:
        brelse(old.dir_bh);
        brelse(old.bh);
```