



author Filipe Manana <fdmanana@suse.com> 2022-11-01 16:15:39 +0000
committer Greg Kroah-Hartman <gregkh@linuxfoundation.org> 2022-11-10 17:46:53 +0100
commit 203204798831c35d855ecc6417d98267d2d2184b (patch)
tree 15c447982a680bc1da5a3991284e13b79c324df8
parent cded2c89774b99b67c98147ae103ea878c92a206 (diff)
download linux-203204798831c35d855ecc6417d98267d2d2184b.tar.gz

diff options

context: 3 ▼
space: include ▼
mode: unified ▼

btrfs: fix ulist leaks in error paths of qgroup self tests

[Upstream commit d37de92b38932d40e4a251e876cc388f9aee5f42]

In the test_no_shared_qgroup() and test_multiple_refs() qgroup self tests, if we fail to add the tree ref, remove the extent item or remove the extent ref, we are returning from the test function without freeing the "old_roots" ulist that was allocated by the previous calls to btrfs_find_all_roots(). Fix that by calling ulist_free() before returning.

Fixes: 442244c96332 ("btrfs: qgroup: Switch self test to extent-oriented qgroup mechanism.")

Signed-off-by: Filipe Manana <fdmanana@suse.com>

Signed-off-by: David Sterba <dsterba@suse.com>

Signed-off-by: Sasha Levin <sashal@kernel.org>

Diffstat

-rw-r--r--	fs/btrfs/tests/qgroup-tests.c	20
------------	-------------------------------	----

1 files changed, 15 insertions, 5 deletions

diff --git a/fs/btrfs/tests/qgroup-tests.c b/fs/btrfs/tests/qgroup-tests.c

index d07dd26194b15d..24a61e52486117 100644

--- a/fs/btrfs/tests/qgroup-tests.c

+++ b/fs/btrfs/tests/qgroup-tests.c

@@ -237,8 +237,10 @@ static int test_no_shared_qgroup(struct btrfs_root *root,

ret = insert_normal_tree_ref(root, nodesize, nodesize, 0,
BTRFS_FS_TREE_OBJECTID);

- if (ret)
+ if (ret) {
+ ulist_free(old_roots);
return ret;
+ }

ret = btrfs_find_all_roots(&trans, fs_info, nodesize, 0, &new_roots,
false);

@@ -273,8 +275,10 @@ static int test_no_shared_qgroup(struct btrfs_root *root,
}

ret = remove_extent_item(root, nodesize, nodesize);

- if (ret)
+ if (ret) {
+ ulist_free(old_roots);

```

        return -EINVAL;
+    }

    ret = btrfs_find_all_roots(&trans, fs_info, nodesize, 0, &new_roots,
                              false);
@@ -338,8 +342,10 @@ static int test_multiple_refs(struct btrfs_root *root,

    ret = insert_normal_tree_ref(root, nodesize, nodesize, 0,
                                BTRFS_FS_TREE_OBJECTID);
-    if (ret)
+    if (ret) {
+        ulist_free(old_roots);
        return ret;
+    }

    ret = btrfs_find_all_roots(&trans, fs_info, nodesize, 0, &new_roots,
                              false);
@@ -373,8 +379,10 @@ static int test_multiple_refs(struct btrfs_root *root,

    ret = add_tree_ref(root, nodesize, nodesize, 0,
                      BTRFS_FIRST_FREE_OBJECTID);
-    if (ret)
+    if (ret) {
+        ulist_free(old_roots);
        return ret;
+    }

    ret = btrfs_find_all_roots(&trans, fs_info, nodesize, 0, &new_roots,
                              false);
@@ -414,8 +422,10 @@ static int test_multiple_refs(struct btrfs_root *root,

    ret = remove_extent_ref(root, nodesize, nodesize, 0,
                          BTRFS_FIRST_FREE_OBJECTID);
-    if (ret)
+    if (ret) {
+        ulist_free(old_roots);
        return ret;
+    }

    ret = btrfs_find_all_roots(&trans, fs_info, nodesize, 0, &new_roots,
                              false);

```