Can't access file storage after user restart

+1   Hotlists (4)   Mark as Duplicate

Comments (7)   Dependencies   Duplicates (0)   Blocking (0)   Resources (12)

Assigned   Bug   P3   + Add Hotlist   [AOSP] assigned

👥 **STATUS UPDATE** No update yet.   Edit

📄 **DESCRIPTION** ro...@romanr.info created issue #1

In multiuser setup when a user is switched back the file storage is not accessible.

Steps to reproduce

1. Set 🔗config_multiuserMaxRunningUsers to "2" (1 + Owner) via 🔗RRO (🔗sample)
    1. Set 🔗config_multiuserMaximumUsers to a value greater "2" if necessary.
    2. Also take a look at 🔗config_enableMultiUserUI.
2. Create users "1" and "2"
3. Switch to user "1"
4. Switch to user "2"
5. Switch back to user "1"
6. Try to open file storage in a File manager.

What happened

The file manager says "Can't load content at this time". Some exception in the logcat log.

What you think the correct behavior should be

The user is restarted, file storage is mounted, permissions are set, file storage is accessible in a file manager.

Setup

Lineage 18.1 (lineage-18.1-20221018-nightly-z3c, AOSP android-11.0.0_r46) for Sony Xperia Z3 Compact (z3c).

Bug artifacts

- 🔗logcat
- 🔗gdrive

Could you please fix it and backport the patch to the android-11.0.0 branch. Thanks.

✓ Links (11)

🔗 **Links (11)**

"Set 🔗config_multiuserMaxRunningUsers to "2" (1 + Owner) via RRO (sample) Set config_multiuserMaximumUsers to a value greater "2" if necessary. Also take a look at config_enableMultiUserUI."

"Set config_multiuserMaxRunningUsers to "2" (1 + Owner) via 🔗RRO (sample) Set config_multiuserMaximumUsers to a value greater "2" if necessary. Also take a look at config_enableMultiUserUI."

"Set config_multiuserMaxRunningUsers to "2" (1 + Owner) via RRO (🔗sample) Set config_multiuserMaximumUsers to a value greater "2" if necessary. Also take a look at config_enableMultiUserUI."

"Set 🔗config_multiuserMaximumUsers to a value greater "2" if necessary."

"Also take a look at 🔗config_enableMultiUserUI ."

See all related links

COMMENTS

All cor

○ **vi...@google.com** <vi...@google.com> #2

Assigned to vi...@google.com.

We have shared this with our product and engineering team and will update this issue with more information as it becomes available.

○ **ro...@romanr.info** <ro...@romanr.info> #3

Another way to reproduce this:

1. Set 🔗fw.max_users to 3.
    This will work since 🔗config_multiuserMaxRunningUsers is set to 3 by default.
    ```
    $ setprop fw.max_users 3
    ```
2. Create 3 extra users.
3. Switch to "1", then to "2", then to "3" user.
4. Switch back to the "1" user.

ro...@romanr.info <ro...@romanr.info> #4

I've noticed in the attached logcat.log that `fscrypt_unlock_user_key` is issued at start and `fscrypt_lock_user_key` is issued when the user is stopped but no `fscrypt_unlock_user_ke` restarted. Can this be the root cause?

---

vi...@google.com <vi...@google.com> #5

Thanks for the information, we have shared it with our engineering team.

---

ro...@romanr.info <ro...@romanr.info> #6

I have noticed that on the second switch to the user there is an extra fuse mount:

```
/dev/fuse on /storage/emulated type fuse (rw,nosuid,nodev,noexec,noatime,user_id=0,group_id=0,allow_other)
/data/media on /storage/emulated/10/Android/data type sdcardfs (rw,nosuid,nodev,noexec,noatime,fsuid=1023,fsgid=1023,gid=1015,multiuser,mask=6,derive_gid,
/data/media on /storage/emulated/10/Android/obb type sdcardfs (rw,nosuid,nodev,noexec,noatime,fsuid=1023,fsgid=1023,gid=1015,multiuser,mask=6,derive_gid,d
/dev/fuse on /storage/emulated type fuse (rw,nosuid,nodev,noexec,noatime,user_id=0,group_id=0,allow_other)
/data/media on /storage/emulated/10/Android/data type sdcardfs (rw,nosuid,nodev,noexec,noatime,fsuid=1023,fsgid=1023,gid=1015,multiuser,mask=6,derive_gid,
/data/media on /storage/emulated/10/Android/data type sdcardfs (rw,nosuid,nodev,noexec,noatime,fsuid=1023,fsgid=1023,gid=1015,multiuser,mask=6,derive_gid,
/data/media on /storage/emulated/10/Android/obb type sdcardfs (rw,nosuid,nodev,noexec,noatime,fsuid=1023,fsgid=1023,gid=1015,multiuser,mask=6,derive_gid,d
/data/media on /storage/emulated/10/Android/obb type sdcardfs (rw,nosuid,nodev,noexec,noatime,fsuid=1023,fsgid=1023,gid=1015,multiuser,mask=6,derive_gid,d
```

I can imagine 2 possible cases:

1. The previous mount was not cleanly unmounted
2. The subsequent mount was mounted twice

To debug this I've applied this ⊝change: adding a seq number to a mount entry in a void-kind parameter.

Now I see this in the /proc/5522/mounts:

```
/dev/fuse on /storage/emulated type fuse (rw,nosuid,nodev,noexec,noatime,user_id=0,group_id=0,allow_other,max_read=20000)
/data/media on /storage/emulated/10/Android/data type sdcardfs (rw,nosuid,nodev,noexec,noatime,fsuid=1023,fsgid=1023,gid=1015,multiuser,mask=6,derive_gid,
/data/media on /storage/emulated/10/Android/obb type sdcardfs (rw,nosuid,nodev,noexec,noatime,fsuid=1023,fsgid=1023,gid=1015,multiuser,mask=6,derive_gid,d
/dev/fuse on /storage/emulated type fuse (rw,nosuid,nodev,noexec,noatime,user_id=0,group_id=0,allow_other,max_read=50000)
/data/media on /storage/emulated/10/Android/data type sdcardfs (rw,nosuid,nodev,noexec,noatime,fsuid=1023,fsgid=1023,gid=1015,multiuser,mask=6,derive_gid,
/data/media on /storage/emulated/10/Android/data type sdcardfs (rw,nosuid,nodev,noexec,noatime,fsuid=1023,fsgid=1023,gid=1015,multiuser,mask=6,derive_gid,
/data/media on /storage/emulated/10/Android/obb type sdcardfs (rw,nosuid,nodev,noexec,noatime,fsuid=1023,fsgid=1023,gid=1015,multiuser,mask=6,derive_gid,d
/data/media on /storage/emulated/10/Android/obb type sdcardfs (rw,nosuid,nodev,noexec,noatime,fsuid=1023,fsgid=1023,gid=1015,multiuser,mask=6,derive_gid,d
```

Please note that the user mounts are not visible for root user. You need to choose an appropriate pid of a user process.

max_read=20000 vs **max_read=50000**, so there is a case #1 - the previous mount was not unmounted.

Note there are several places where paths are bind-mounted, e.g. in ⊝vold and ⊝zygote. Perhaps there is a race condition.

---

ro...@romanr.info <ro...@romanr.info> #7

For now I've set **persist.sys.fflag.override.settings_fuse**=false as a workaround (Android 11).

```
# setprop persist.sys.fflag.override.settings_fuse false
```

Please note the setting **persist.sys.fuse**=false does not work (Android 11).