copy: avoid useless comparison for non-directory entries #6656

New issue

Closed rhertzog wants to merge 1 commit into `systemd:master` from `rhertzog:overlayfs-fixup`

Conversation 22 · Commits 1 · Files changed 1 · +4 −5

rhertzog commented on Aug 22

The comparison of the device number is meant to not follow directories across multiple filesystems, so the test is only relevant when the entry is a directory.

This also fixes the test-copy test when /tmp is an overlayfs with a kernel older than Linux 4.12 since overlayfs reported inconsistent devices numbers. See https://bugs.debian.org/854400 for some details.

`copy: avoid useless comparison for non-directory entries` (sic) ✓ fba31f5

evverx reviewed 29 days ago — View changes

src/basic/copy.c

```
394     -               else if (S_ISDIR(buf.st_mode))
        394 +           else if (S_ISDIR(buf.st_mode)) {
        395 +               if (buf.st_dev != original_device)
        396 +                   continue;
```

evverx 29 days ago • edited — Member

Thank you. I'm not sure where this `if` should be placed. `fd_copy_directory` stops copying as soon as it detects a filesystem boundary, while `cp --one-file-system` creates an empty directory before stopping copying.

rhertzog 29 days ago

Right. I made my change to be minimal while keeping the current behaviour. But when you come over such a mount point, it's true that the underlying filesystem has at least an empty directory for the mount to be possible. So arguably the behaviour of `cp --one-file-system` is more logical. Shall I update my patch to behave like cp ?

evverx 29 days ago — Member

@rhertzog, I think that the patch fixes the bug and should be updated, but I'd rather wait for someone else to confirm that skipping device nodes and bind-mounted files is not intentional.

rhertzog 29 days ago

The stat call on device nodes does not report the device number of the device itself but of the filesystem hosting the device node, so this one is fine. You are right though that this changes the behaviour for bind-mounted files which would be copied with my patch while they are skipped currently. That said copy_tree is only used by `systemd-tmpfiles` to setup the temporary files and by `machinectl` to copy files between host and containers. In both cases, I don't believe that we have any specific requirement to ignore bind mounted files.

👍

evverx 29 days ago — Member

I'm sorry. Indeed, I was wrong about device nodes. I should have used `mknod` to check the behaviour of `stat`.

> I don't believe that we have any specific requirement to ignore bind mounted files.

Neither do I.

poettering commented 24 days ago — Owner

> The comparison of the device number is meant to not follow directories across multiple filesystems, so the test is only relevant when the entry is a directory.

I don't follow? On Linux mounts may be on either files and directories, and often are. Hence limiting any such checks to directories will make us miss any file mounts. This patch hence looks not OK.

poettering commented 24 days ago — Owner

On traditional UNIX the way to detect file system boundaries is via comparing st_dev. This is implemented in numerous tools, and as a fallback in systemd too. overlayfs breaks with that UNIX API if it changes st_dev within the same file system, but the right place to fix that really appears to be overlayfs, instead of making all the numerous tools work around overlayfs' peculiarities etc. I am not sure what overlayfs' strategy on this is, but it appears to be quite a steep compat break on their side...

poettering added needs-discussion util-lib labels 24 days ago

poettering commented 24 days ago — Owner

(oh, and we have similar checks in path_is_mount_point() and various other places iirc. Any such patch if accepted would need to cover all cases not just one specific one — but again I am not convinced this is really the right way to go...)

rhertzog commented 24 days ago

@poettering What can you mount on a file except another file?

In a copy_tree() function, you want to avoid infinite copying through (directory) mount points loops, so it's a safety measure to skip the mounted directories. But a mounted file does not pose any risk, does it?

That's why I believe it's OK to do the change here. But it would not make sense to change the more generic `path_is_mount_point()`.

overlayfs improved already in Linux 4.12 to have fewer such edge cases but the vast majority of the code out there does not care about st_dev on files. `cp -rx` does not for instance.

poettering commented 24 days ago — Owner

> **@poettering** What can you mount on a file except another file?

IIRC the semantics on Linux are that:

1. dirs can be only mounted on dirs, and
2. symlinks not at all and
3. everything else on everything else (i.e. also device nodes on files, and files on device nodes, and any other weird shit)

poettering commented 24 days ago — Owner

> In a copy_tree() function, you want to avoid infinite copying through (directory) mount points loops, so it's a safety measure to skip the mounted directories. But a mounted file does not pose any risk, does it?

Well, while "same-fs" checks are useful to avoid bind mount loops they have other uses too. For example, they often are conceptual boundaries. For example, you copy a fully set up OS image, then /proc, /sys/, ... and so on, are all mounts, which you want to avoid, but everything else should be copied just fine. and if somebody mounts /proc/core to /root/core, then it should also be avoided.

I am still not convinced that adding such a work-around for one specific fs that departs so strongly from accepted UNIX behaviour is the right way to go...

It's weird being on the side of arguing for UNIX here, instead of the other side, but here I am ;-)

poettering commented 24 days ago — Owner

> overlayfs improved already in Linux 4.12 to have fewer such edge cases but the vast majority of the code out there does not care about st_dev on files. cp -rx does not for instance.

What precisely changed in 4.12? Do you have a link or so?

What about rsync, find, tar and all those tools? What do they do?

rhertzog commented 24 days ago • edited

> What precisely changed in 4.12? Do you have a link or so?

I'm not sure that the relevant change were added in 4.12, I just know that 4.9 was still exhibiting the test-copy failure while 4.12 no longer does. I think the changes have been done in 4.10 and merged by Linus in commit ff0f962cac3d823962f9a70e7eea27abfbb979c3.

The documentation of overlayfs (https://github.com/torvalds/linux/blob/master/Documentation/filesystems/overlayfs.txt) now states:

> In the special case of all overlay layers on the same underlying filesystem, all objects will report an st_dev from the overlay filesystem and st_ino from the underlying filesystem. This will make the overlay mount more compliant with filesystem scanners and overlay objects will be distinguishable from the corresponding objects in the original filesystem.

Now about your other question:

> What about rsync, find, tar and all those tools? What do they do?

rsync skips only mount-point directories: https://git.samba.org/?p=rsync.git;a=blob;f=flist.c;h=28553fc3dab1683b953a085c2528aceeef9f9204;hb=HEAD#l1205

Looks like find is doing the same: http://git.savannah.gnu.org/cgit/gnulib.git/tree/lib/fts.c#n912 (FTS_D stands for directory)

And same for tar: http://git.savannah.gnu.org/cgit/tar.git/tree/src/create.c#n1197 (it's in a function dump_dir0 that only handles directories, no occurrence anywhere else)

evverx commented 23 days ago — Member

> But a mounted file does not pose any risk, does it?

`fd_copy_directory` might end up copying huge `/proc/*/pagemap` , which can be considered as an infinite loop. `rsync` seems to avoid that issue by copying at most `st_size` bytes.

poettering commented 23 days ago — Owner

hmm, interesting to know that those tools only do this check for directories. But it still feels hackish to special case dirs for that I must say... not sure what to do on this... I am still tempted to say that the overlayfs people really need to pass out valid st_dev values, and evrything else is compat breakage... And it appears they are even aware of the issue...

Maybe a more digestable version would be to check the fs magic value of the relevant subdir, to special case this. It's awful, but then at least we do the broken logic only on broken overlayfs...

rhertzog commented 23 days ago

If the "fix/work-around" is not desired, as far as I am concerned, I would be also happy if the failing test was just skipped when we detect overlayfs (and maybe Linux < 3.10). I'm not sure how to achieve this though.

evverx commented 22 days ago • edited — Member

I'm not sure that people who use `C` in `tmpfiles.d/*.conf` expect `systemd-tmpfiles` to work in a way that is different from `cp -r -x` or `rsync -r --one-file-system` . In addition, not crossing file system boundaries is not documented, so that might probably be surprising.

Also, I think that not creating empty files and directories is still a bug.

evverx commented 22 days ago — Member

> Also, I think that not creating empty files and directories is still a bug.

On the other hand, `rsync` doesn't create empty directories when `--one-file-system` is repeated, so it seems that some people might expect `systemd-tmpfiles` to do the same when `C` is used.

poettering commented 22 days ago — Owner

> If the "fix/work-around" is not desired, as far as I am concerned, I would be also happy if the failing test was just skipped when we detect overlayfs (and maybe Linux < 3.10). I'm not sure how to achieve this though.

wouldn't it be better to not use overlayfs for /tmp on those test systems? It's a pretty poor choice for /tmp for all its semantics. Most apps expect a fully featured, fast fs there, and overlayfs isn't really that at all, it has all kinds of weird shortcomings and performance weirdnesses (since copy-up needs to happen all the time)

rhertzog commented 21 days ago

The build bots are using overlayfs in order to be able to easily discard changes from the initial clean chroot in which the build is triggered. Arguably we could put a tmpfs on /tmp but then we could hit ENOSPC for packages storing/generating large amount of data on /tmp. The build itself happens in bind mount with a directory outside of the build chroot which is thus not under the control of overlayfs... so the last solution is to not use /tmp but a temporary directory withing the build-tree itself.

But I think this is a bit off-topic, I know how to avoid the problem... I just thought that it would best addressed in systemd.

poettering commented 21 days ago — Owner

Ok, then let's close this here, if that's OK. I am not convinced that adding explicit work-arounds for overlayfs is really the best option for now, and maybe we can revisit this later, should the issue come up again in some form.

poettering closed this 21 days ago

evverx commented 20 days ago — Member

I think it would be great if `systemd-tmpfiles` copied files in a similar way to `rsync` . On the other hand, nobody appears to have ever pointed `systemd-tmpfiles` to `/proc` or noticed missing empty files and directories, so everything seems to be fine :-)

Reviewers: evverx
Assignees: No one assigned
Labels:
Projects: None yet
Milestone: No milestone
3 participants