Commit 244f3ee

Théophane Hufsch… committed on Mar 7, 2024

```
Copy the output of fixed-output derivations before registering them

It is possible to exfiltrate a file descriptor out of the build sandbox
of FODs, and use it to modify the store path after it has been
registered.
To avoid that issue, don't register the output of the build, but a copy
of it (that will be free of any leaked file descriptor).
```

2.20-maintenance · 2.20.9 ••• 2.20.5

1 parent 4645652 commit 244f3ee

3 files changed +18 -0 lines changed

Top

Filter files…

- src
  - libstore/build
    - local-derivation-goal.cc
  - libutil
    - file-system.cc
    - file-system.hh

3 files changed +18 -0 lines changed

Search within code

src/libstore/build/local-derivation-goal.cc

```
@@ -2527,6 +2527,12 @@ SingleDrvOutputs
LocalDerivationGoal::registerOutputs()
2527  2527              [&](const DerivationOutput::CAFixed & dof) {
2528  2528                  auto & wanted = dof.ca.hash;
```

```
2529  2529
      2530  +                 // Replace the output by a fresh copy of itself to make sure
      2531  +                 // that there's no stale file descriptor pointing to it
      2532  +                 Path tmpOutput = actualPath + ".tmp";
      2533  +                 copyFile(actualPath, tmpOutput, true);
      2534  +                 renameFile(tmpOutput, actualPath);
      2535  +
2530  2536                    auto newInfo0 = newInfoFromCA(DerivationOutput::CAFloating {
2531  2537                        .method = dof.ca.method,
2532  2538                        .hashAlgo = wanted.algo,
```

src/libutil/file-system.cc

```
                  @@ -628,6 +628,11 @@ void copy(const fs::directory_entry & from, const
                  fs::path & to, bool andDelete)
628   628             }
629   629         }
630   630
      631   + void copyFile(const Path & oldPath, const Path & newPath, bool andDelete)
      632   + {
      633   +     return copy(fs::directory_entry(fs::path(oldPath)), fs::path(newPath),
                  andDelete);
      634   + }
      635   +
631   636     void renameFile(const Path & oldName, const Path & newName)
632   637     {
633   638         fs::rename(oldName, newName);
```

src/libutil/file-system.hh

```
                  @@ -186,6 +186,13 @@ void renameFile(const Path & src, const Path & dst);
186   186      */
187   187     void moveFile(const Path & src, const Path & dst);
188   188
      189   + /**
      190   +  * Recursively copy the content of `oldPath` to `newPath`. If `andDelete` is
      191   +  * `true`, then also remove `oldPath` (making this equivalent to `moveFile`,
                  but
      192   +  * with the guaranty that the destination will be “fresh”, with no stale inode
```

```
193 +  * or file descriptor pointing to it).
194 +  */
195 + void copyFile(const Path & oldPath, const Path & newPath, bool andDelete);
189 196
190 197    /**
191 198     * Automatic cleanup of resources.
```

Comments 0

Please sign in to comment.