

Rudder - Bug #10312

"Download files from the shared folder" behaving badly?

2017-03-01 13:21 - Hamlyn Mootoo

Status: Released	
Priority: N/A	
Assignee: Nicolas CHARLES	
Category: Techniques	
Target version: 3.1.19	
Pull Request: https://github.com/Normation/rudder-techniques/pull/1124	Priority: 0
Severity: Critical - prevents main use of Rudder no workaround data loss security	Name check:
UX impact:	Fix check:
User visibility:	Regression:
Effort required:	
Description	
<p>I was experimenting with this directive and uploaded a hosts file to the share. I then specified that this file should be downloaded to the /tmp directory (screenshot attached). The selected "recursion level of the copy" did not seem to matter, nor did putting a trailing slash after the destination (as in "/tmp/"). What I got was the /tmp directory being renamed to /tmp_1488367735_Wed_Mar__1_06_28_59_2017.cfsaved and the hosts file being renamed to /tmp. On this server /tmp is not a separate partition.</p> <p>Is this behavior expected? It seems a bit dangerous.</p>	
Related issues:	
Related to Rudder - User story #10214: More consistant naming of techniques	Released
Related to Rudder - Bug #10377: When copying files, digest comparison uses ct...	New
Has duplicate Rudder - Bug #4334: Technique "Download From A Shared Folder": ...	Rejected

Associated revisions

Revision 4fb58eb6 - 2017-03-08 18:01 - Alexis Mousset

Fixes #10312: "\"Download files from the shared folder\" behaving badly?

History

#1 - 2017-03-02 17:06 - Vincent MEMBRÉ

- Target version changed from 4.1.0~rc1 to 4.1.0

#2 - 2017-03-06 15:05 - Benoît PECCATTE

The destination must be the absolute file path, not the destination directory.
If you say /tmp, /tmp is considered to be the destination "file".

#3 - 2017-03-07 01:55 - Hamlyn Mootoo

While I understand this now, it is very easy to misinterpret, and getting /tmp renamed was not fun. As a suggestion I would change the wording of the this field to something like "Full path of destination file". This would be somewhat symmetrical with the "Path of file to be copied" field. I understand there is some help text when the mouse hovers over the question mark, but if other sys admins are anything like me, when we read "destination" for a file, we think "directory location" not full path, and would not think help is needed. The reason for this is that when copying a file from one place to another, unless you're renaming the file, not many admins would put in the full path for the destination. For example:

```
cp /etc/hosts /tmp
NOT cp /etc/hosts /tmp/hosts
```

#4 - 2017-03-07 01:59 - Hamlyn Mootoo

Also, when a trailing slash was put in (clearly indicating the intent to put the file into a directory), it behaved the same way, namely overwriting (but luckily renaming first) the directory.

#5 - 2017-03-07 12:11 - Jonathan CLARKE

- Tracker changed from User story to Bug
- Target version changed from 4.1.0 to 3.1.19
- Reproduced set to No

I agree with you 100%, Hamlyn. This is misleading, and you're not the first person to misunderstand this, far from it. Worse than misleading, this could cause data loss (if it wasn't /tmp but /etc, and the .cfsaved backup ended up getting removed... the consequences could be disastrous).

This needs to be reworked. I'm requalifying this as a bug - it needs fixing, at least by clearer wording, and better still by figuring out we probably don't ever want to replace a directory with a single file (for example).

#6 - 2017-03-07 12:11 - Jonathan CLARKE

- Category changed from Agent to Techniques

#7 - 2017-03-07 12:31 - François ARMAND

- Related to User story #10214: More consistant naming of techniques added

#8 - 2017-03-08 12:05 - Jonathan CLARKE

- Severity set to Critical - prevents main use of Rudder | no workaround | data loss | security

#9 - 2017-03-08 14:06 - Alexis Mousset

- Category changed from Techniques to Agent
- Assignee set to Alexis Mousset

Besides the documentation issue, it seems to be a bug in CFEngine. We do not use move_obstructions in this technique, so we should not replace a link or a directory by a file. Furthermore, the behavior seems different according to the file comparison method.

#10 - 2017-03-08 14:08 - Alexis Mousset

May be linked to <https://tracker.mender.io/browse/CFE-1811>.

#11 - 2017-03-08 14:14 - Alexis Mousset

When the dir directory exists on the destination:

Source	Destination	Comparison	Purge	Recursion	Outcome
file	dir	digest	*	*	Does nothing, returns success or replaces the directory, depending on ctime!
file	dir/	digest	*	*	Does nothing, returns success or fails and does not replace the directory, depending on ctime!
file	dir	mtime	*	*	Replaces the directory
file	dir/	mtime	*	*	Creates dir.cfnew but fails and does no replace the directory

The issue with checksum is:

```
verbose: Checksum comparison replaced by ctime: files not regular
```

With mtime:

```

verbose: File '/tmp/dir' copy_from '/tmp/file'
verbose: Destination file '/tmp/dir' already exists
verbose: Image file '/tmp/dir' out of date, should be copy of '/tmp/file'
verbose: Copy of regular file succeeded '/tmp/file' to '/tmp/dir.cfnew'
verbose: No depth search when copying '/tmp/dir.cfsaved' so purging does not apply
info: Cannot move a directory to repository, leaving at '/tmp/dir.cfsaved'
info: Updated '/tmp/dir' from source '/tmp/file' on 'localhost'

```

#12 - 2017-03-08 14:44 - Alexis Mousset

From move_obstruction doc:

If we have promised to make file X a link, but it already exists as a file, or vice-versa, or if a file is blocking the creation of a directory, then normally CFEngine will report an error.

#13 - 2017-03-08 17:23 - Alexis Mousset

We can:

- If (the destination exists and is a directory) and (the destination does not end with a "/") => Add a "/" at the end of destination.

This allows to fail if the source is a file, and avoid removing the destination.

#14 - 2017-03-08 17:24 - Alexis Mousset

- Category changed from Agent to Techniques

#15 - 2017-03-08 17:50 - Alexis Mousset

- Status changed from New to In progress

#16 - 2017-03-08 18:01 - Alexis Mousset

- Status changed from In progress to Pending technical review

- Assignee changed from Alexis Mousset to Nicolas CHARLES

- Pull Request set to <https://github.com/Normation/rudder-techniques/pull/1124>

PR <https://github.com/Normation/rudder-techniques/pull/1124>

#17 - 2017-03-08 18:07 - Alexis Mousset

- Related to Bug #10377: When copying files, digest comparison uses ctime when types are different. added

#18 - 2017-03-09 10:46 - Alexis Mousset

The only case where a directory got replaced by a file in my tests were when the destination did not end with a "/".

The proposed fix is, when the target is an existing directory on the managed node, to enforce that the destination in the directive ends by a "/". This way, the directory will not be replaced, and an error will be reported.

#19 - 2017-03-16 10:50 - Alexis Mousset

- Status changed from Pending technical review to Pending release

Applied in changeset [rudder-techniques|4fb58eb68699892f9821c0c1ef37980d0212286d](https://github.com/Normation/rudder-techniques/commit/4fb58eb68699892f9821c0c1ef37980d0212286d).

#20 - 2017-03-30 12:10 - François ARMAND

- Has duplicate Bug #4334: Technique "Download From A Shared Folder": copy files to directory destination doesn't work added

#21 - 2017-04-19 12:12 - Vincent MEMBRÉ

- Status changed from Pending release to Released

- Priority set to 0

This bug has been fixed in Rudder 3.1.19, 4.0.4 and 4.1.1 which were released today.

- 3.1.19: [Announce Changelog](#)
- 4.0.4: [Announce Changelog](#)
- 4.1.1: [Announce Changelog](#)
- Download: <https://www.rudder-project.org/site/get-rudder/downloads/>

Files

downloadfile.JPG	56 KB	2017-03-01	Hamlyn Mootoo
------------------	-------	------------	---------------