NAME

vfs_shadow_copy2 - Expose snapshots to Windows clients as shadow copies.

SYNOPSIS

vfs objects = shadow_copy2

DESCRIPTION

This VFS module is part of the **samba**(7) suite.

The vfs_shadow_copy2 VFS module offers a functionality similar to Microsoft Shadow Copy services. When set up properly, this module allows Microsoft Shadow Copy clients to browse through file system snapshots as "shadow copies" on Samba shares.

This is a second implementation of a shadow copy module which has the following additional features (compared to the original **shadow_copy**(8) module):

1.
is no need any more to populate your share’s root directory with symlinks to the snapshots if the file system stores the snapshots elsewhere. Instead, you can flexibly configure the module where to look for the file system snapshots. This can be very important when you have thousands of shares, or use [homes].

2.
directories need not be in one fixed central place but can be located anywhere in the directory tree. This mode helps to support file systems that offer snapshotting of particular subtrees, for example the GPFS independent file sets.

3.
naming for snapshots: snapshots can be named in any format compatible with str[fp]time conversions.

4.
can be represented in localtime rather than UTC.

5.
inode number of the files can optionally be altered to be different from the original. This fixes the ’restore’ button in the Windows GUI to work without a sharing violation when serving from file systems, like GPFS, that return the same device and inode number for the snapshot file and the original.

6.

copy results are by default sorted before being sent to the client. This is beneficial for filesystems that don't read directories alphabetically (the default unix). Sort ordering can be configured and sorting can be turned off completely if the file system sorts its directory listing.

This module is stackable.

CONFIGURATION

vfs_shadow_copy2 relies on a filesystem snapshot implementation. Many common filesystems have native support for this.

Filesystem snapshots must be available under specially named directories in order to be recognized by vfs_shadow_copy2. These snapshot directory is typically a direct subdirectory of the share root's mountpoint but there are other modes that can be configured with the parameters described in detail below.

The snapshot at a given point in time is expected in a subdirectory of the snapshot directory where the snapshot's directory is expected to be a formatted version of the snapshot time. The default format which can be changed with the shadow:format option is @GMT-YYYY.MM.DD-hh.mm.ss, where:

- is the 4 digit year
- is the 2 digit month
- is the 2 digit day
- is the 2 digit hour
- is the 2 digit minute
- is the 2 digit second.

The vfs_shadow_copy2 snapshot naming convention can be produced with the following **date**(1) command:

```
TZ=GMT date +@GMT-%Y.%m.%d-%H.%M.%S
```

OPTIONS

shadow:mountpoint = MOUNTPOINT

With this parameter, one can specify the mount point of the filesystem that contains the share path. Usually this mount point is automatically detected. But for some constellations, in particular tests, it can be convenient to be able to specify it.

Example: shadow:mountpoint = /path/to/filesystem

Default: shadow:mountpoint = NOT SPECIFIED

shadow:snapdir = SNAPDIR

Path to the directory where the file system of the share keeps its snapshots. If an absolute path is specified, it is used as-is. If a relative path is specified, then it is taken relative to the mount point of the filesystem of the share root. (See shadow:mountpoint.)

Note that shadow:snapdirseverywhere depends on this parameter and needs a relative path. Setting an absolute path disables shadow:snapdirseverywhere.

Note that the shadow:crossmountpoints option also requires a relative snapdir. Setting an absolute path disables shadow:crossmountpoints.

Example: shadow:snapdir = /some/absolute/path

Default: shadow:snapdir = .snapshots

shadow:basedir = BASEDIR

The basedir option allows one to specify a directory between the share’s mount point and the share root, relative to which the file system’s snapshots are taken.

For example, if

[illegible]
= mountpoint/rel_basedir

-
  = basedir/rel_share_root

-
  = mountpoint/snapdir

  or snapshot_path = snapdir if snapdir is absolute

then the snapshot of a file = mountpoint/rel_basedir/rel_share_root/rel_file at a time TIME will be found under snapshot_path/FS_GMT_TOKEN(TIME)/rel_share_root/rel_file, where FS_GMT_TOKEN(TIME) is the timestamp string belonging to TIME in the format required by the file system. (See shadow:format.)

The default for the basedir is the mount point of the file system of the share root (see shadow:mountpoint).

Note that the shadow:snapdirseverywhere and shadow:crossmountpoints options are incompatible with shadow:basedir and disable the basedir setting.

shadow:snapsharepath = SNAPSHAREPATH

With this parameter, one can specify the path of the share's root directory in snapshots, relative to the snapshot's root directory. It is an alternative method to shadow:basedir, allowing greater control.

For example, if within each snapshot the files of the share have a path/to/share/ prefix, then shadow:snapsharepath can be set to path/to/share.

With this parameter, it is no longer assumed that a snapshot represents an image of the original file system or a portion of it. For example, a system could perform backups of only files contained in shares, and then expose the backup files in a logical structure:

-
-
-

Note that the shadow:snapdirseverywhere and the shadow:basedir options are incompatible with shadow:snapsharepath and disable shadow:snapsharepath setting.

Example: shadow:snapsharepath = path/to/share

Default: shadow:snapsharepath = NOT SPECIFIED

shadow:sort = asc/desc

By default, this module sorts the shadow copy data alphabetically before sending it to the client. With this parameter, one can specify the sort order. Possible known values are desc (descending, the default) and asc (ascending). If the file system lists directories alphabetically sorted, one can turn off sorting in this module by specifying any other value.

Example: shadow:sort = asc

Example: shadow:sort = none

Default: shadow:sort = desc

shadow:localtime = yes/no

This is an optional parameter that indicates whether the snapshot names are in UTC/GMT or in local time. If it is disabled then UTC/GMT is expected.

shadow:localtime = no

shadow:format = format specification for snapshot names

This is an optional parameter that specifies the format specification for the naming of snapshots in the file system. The format must be compatible with the conversion specifications recognized by str[fp]time.

Default: shadow:format = "@GMT-%Y.%m.%d-%H.%M.%S"

shadow:sscanf = yes/no

This parameter can be used to specify that the time in format string is given as an unsigned long integer (%lu) rather than a time strptime() can parse. The result must be a unix time_t time.

Default: shadow:sscanf = no

shadow:fixinodes = yes/no

If you enable shadow:fixinodes then this module will modify the apparent inode number of files in the snapshot directories using a hash of the files path. This is needed for snapshot systems where the snapshots have the same device:inode number as the original files (such as happens with GPFS snapshots). If you don’t set this option then the ’restore’ button in the shadow copy UI will fail with a sharing violation.

Default: shadow:fixinodes = no

shadow:snapdirseverywhere = yes/no

If you enable shadow:snapdirseverywhere then this module will look out for snapshot directories in the current working directory and all parent directories, stopping at the mount point by default. But see shadow:crossmountpoints how to change that behaviour.

An example where this is needed are independent filesets in IBM’s GPFS, but other filesystems might support snapshotting only particular subtrees of the filesystem as well.

Note that shadow:snapdirseverywhere depends on shadow:snapdir and needs it to be a relative path. Setting an absolute snapdir path disables shadow:snapdirseverywhere.

Note that this option is incompatible with the shadow:basedir option and removes the shadow:basedir setting by itself.

Example: shadow:snapdirseverywhere = yes

Default: shadow:snapdirseverywhere = no

shadow:crossmountpoints = yes/no

This option is effective in the case of shadow:snapdirseverywhere = yes. Setting this option makes the module not stop at the first mount point encountered when looking for snapdirs, but lets it search potentially all through the path instead.

An example where this is needed are independent filesets in IBM’s GPFS, but other filesystems might support snapshotting only particular subtrees of the filesystem as well.

Note that shadow:crossmountpoints depends on shadow:snapdir and needs it to be a relative path. Setting an absolute snapdir path disables shadow:crossmountpoints.

Note that this option is incompatible with the shadow:basedir option and removes the shadow:basedir setting by itself.

Example: shadow:crossmountpoints = yes

Default: shadow:crossmountpoints = no

shadow:snapprefix

With growing number of snapshots file-systems need some mechanism to differentiate one set of snapshots from other, e.g. monthly, weekly, manual, special events, etc. Therefore these file-systems provide different ways to tag snapshots, e.g. provide a configurable way to name snapshots, which is not just based on time. With only shadow:format it is very difficult to filter these snapshots. With this optional parameter, one can specify a variable prefix component for names of the snapshot directories in the file-system. If this parameter is set, together with the shadow:format and shadow:delimiter parameters it determines the possible names of snapshot directories in the file-system. The option only supports Basic Regular Expression (BRE).

shadow:delimiter

This optional parameter is used as a delimiter between shadow:snapprefix and shadow:format. This parameter is used only when shadow:snapprefix is set.

Default: shadow:delimiter = "_GMT"

EXAMPLES

Add shadow copy support to user home directories:

```
[homes]
    vfs objects = shadow_copy2
    shadow:snapdir = /data/snapshots
    shadow:basedir = /data/home
    shadow:sort = desc
```

CAVEATS

This is not a backup, archival, or version control solution.

With Samba or Windows servers, vfs_shadow_copy2 is designed to be an end-user tool only. It does not replace or enhance your backup and archival solutions and should in no way be considered as such. Additionally, if you need version control, implement a version control system.

VERSION

This man page is part of version 4.13.17 of the Samba suite.

AUTHOR

The original Samba software and related utilities were created by Andrew Tridgell. Samba is now developed by the Samba Team as an Open Source project similar to the way the Linux kernel is developed.