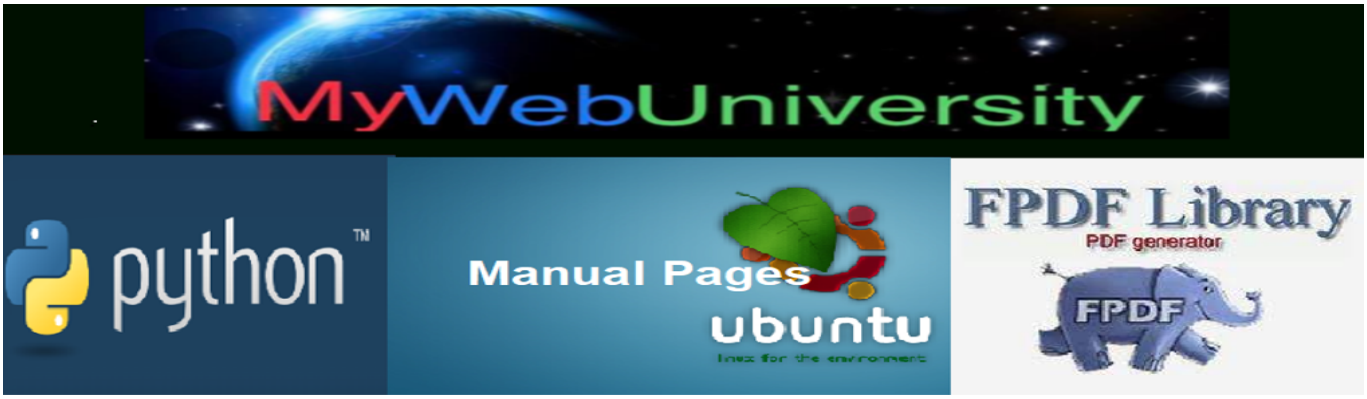

Rocky Enterprise Linux 9.2 Manual Pages on command 'rrsync.1'

$ man rrsync.1

rrsync(1) User Commands rrsync(1)

NAME

rrsync - a script to setup restricted rsync users via ssh logins

SYNOPSIS

rrsync [-ro|-rw] [-munge] [-no-del] [-no-lock] DIR

The single non-option argument specifies the restricted DIR to use. It can be relative to the user's home directory or an absolute path.

The online version of this manpage (that includes cross-linking of topics) is available at https://download.samba.org/pub/rsync/rrsync.1.

DESCRIPTION

A user's ssh login can be restricted to only allow the running of an rsync transfer in one of two easy ways:

- o forcing the running of the rrsync script
- o forcing the running of an rsync daemon-over-ssh command.

Both of these setups use a feature of ssh that allows a command to be forced to run in? stead of an interactive shell. However, if the user's home shell is bash, please see BASH SECURITY ISSUE for a potential issue.

To use the rrsync script, edit the user's ~/.ssh/authorized_keys file and add a prefix like one of the following (followed by a space) in front of each ssh-key line that should be restricted:

command="rrsync DIR"

command="rrsync -ro DIR"

command="rrsync -munge -no-del DIR"

Then, ensure that the rrsync script has your desired option restrictions. You may want to copy the script to a local bin dir with a unique name if you want to have multiple configurations. One or more rrsync options can be specified prior to the DIR if you want to further restrict the transfer.

To use an rsync daemon setup, edit the user's ~/.ssh/authorized_keys file and add a prefix like one of the following (followed by a space) in front of each ssh-key line that should be restricted:

```
command="rsync --server --daemon ."
command="rsync --server --daemon --config=/PATH/TO/rsyncd.conf ."
```

Then, ensure that the rsyncd.conf file is created with one or more module names with the appropriate path and option restrictions. If rsync's --config option is omitted, it defaults to ~/rsyncd.conf. See the rsyncd.conf(5) manpage for details of how to configure an rsync daemon.

When using rrsync, there can be just one restricted dir per authorized key. A daemon setup, on the other hand, allows multiple module names inside the config file, each one with its own path setting.

The remainder of this manpage is dedicated to using the rrsync script.

OPTIONS

-ro Allow only reading from the DIR. Implies -no-del and -no-lock.

-wo Allow only writing to the DIR.

-munge Enable rsync's --munge-links on the server side.

-no-del

Disable rsync's --delete* and --remove* options.

-no-lock

Avoid the single-run (per-user) lock check. Useful with -munge.

-help, -h

Output this help message and exit.

SECURITY RESTRICTIONS

The rrsync script validates the path arguments it is sent to try to restrict them to staying within the specified DIR.

The rrsync script rejects rsync's --copy-links option (by default) so that a copy cannot dereference a symlink within the DIR to get to a file outside the DIR.

The rrsync script rejects rsync's --protect-args (-s) option because it would allow op?

tions to be sent to the server-side that the script cannot check. If you want to support --protect-args, use a daemon-over-ssh setup.

The rrsync script accepts just a subset of rsync's options that the real rsync uses when running the server command. A few extra convenience options are also included to help it to interact with BackupPC and accept some convenient user overrides.

The script (or a copy of it) can be manually edited if you want it to customize the option handling.

BASH SECURITY ISSUE

If your users have bash set as their home shell, bash may try to be overly helpful and en? sure that the user's login bashrc files are run prior to executing the forced command.

This can be a problem if the user can somehow update their home bashrc files, perhaps via the restricted copy, a shared home directory, or something similar.

One simple way to avoid the issue is to switch the user to a simpler shell, such as dash.

When choosing the new home shell, make sure that you're not choosing bash in disguise, as it is unclear if it avoids the security issue.

Another potential fix is to ensure that the user's home directory is not a shared mount and that they have no means of copying files outside of their restricted directories.

This may require you to force the enabling of symlink munging on the server side.

A future version of openssh may have a change to the handling of forced commands that al? lows it to avoid using the user's home shell.

EXAMPLES

The ~/.ssh/authorized_keys file might have lines in it like this:

```
command="rrsync client/logs" ssh-rsa AAAAB3NzaC1yc2EAAAABIwAAAIEAzG...
command="rrsync -ro results" ssh-rsa AAAAB3NzaC1yc2EAAAABIwAAAIEAmk...
```

FILES

~/.ssh/authorized_keys

SEE ALSO

rsync(1), rsyncd.conf(5)

VERSION

This manpage is current for version 3.2.7 of rsync.

CREDITS

rsync is distributed under the GNU General Public License. See the file COPYING for de? tails.

An rsync web site is available at https://rsync.samba.org/ and its github project is https://github.com/WayneD/rsync.

AUTHOR

The original rrsync perl script was written by Joe Smith. Many people have later contributed to it. The python version was created by Wayne Davison.