backupchecker Documentation

Release 1.9

Carl Chenet <chaica@backupcheckerproject.org>

Apr 13, 2017

Contents

You’ll find below anything you need to install, configure or run Backup Checker.

CHAPTER 1

Guide

How to install Backup Checker

From PyPI

> $ pip3 install backupchecker

From sources

- You need at least Python 3.4.
- Untar the tarball and go to the source directory with the following commands:

```
$ tar zxvf backupchecker-1.9.tar.gz
$ cd backupchecker
```

- Next, to install Backup Checker on your computer, type the following command with the root user:

```
$ python3 setup.py install
$ # or
$ python3 setup.py install --install-scripts=/usr/bin
```

Configure Backup Checker

You need two files in order to use Backup Checker, a file offering the configuration of the archive and another file giving the detail of what's inside the archive, let's call it the list of files. But don't worry, the option -G allows to generate both files from a given archive. The next sections offer the details of what parameter these files contain.

Configuration of the archive

The first one contains general information about the backup checking session. It is mandatory your configuration file uses the .conf extension. Here is an example with all the currently supported parameters:

```
[main]
name=mybackup-checking-session
type=archive
path=tests/expected_mode/foos.tar.gz
files_list=tests/expected_mode/files-list
delimiter=|
sha512=87d3325d3bb844734c1b011fb0f12a3ae47676153a8b05102a5e1b5347a86096d85b1b239752c3fdc10a8a2226928
```

- **[main]** is mandatory.
- **name** is the name of your backup checking session. If you have several backup checking sessions, they MUST use a different name.
- **type** is the type of your backup. Currently you only have archive (tar, tar.gz, tar.bz2, zip files) and tree for a tree of directories and files.
- **path** is the path to the archive or the top directory of your files tree. Relative or absolute paths are accepted.
- **files_list** is the path to the file containing the information about the archive, the tree or the files inside your backups. Relative or absolute paths are accepted.
- **delimiter** (optional field) is the string to use in the list of files to mark the end of the key and the beginning of the value. Default is | (pipe).
- **sha512** (optional field) provides the sha512 hash sum of the list of files, in order to check if this file is the expected one.

Placeholders

The **path** and **files_list** values both support some placeholders. Using a value like **/backups/backup-%d-%m-%Y.tar.gz** will see **%m** being replaced by the current monthday, **%m** by the current month number and **%Y** being replace by the current year. The available placeholders are:

- **%Y** for current year
- **%y** for current two-digit year (2015->15)
- **%m** for current month number (1..12)
- **%d** for current monthday (1..31)
- **%w** for current weekday, monday first (1..7)
- **%H** for current hour (00..23)
- **%M** for current minute (00..59)
- **%S** for current second (00..59)
- **%i** for the highest integer found in a filename (/backups/backup.%i.tar.gz will return /backups/backup.2.tar.gz if you have backup.1.tar.gz and backup.2.tar.gz in the /backups directory.

Understanding the parameters of the list of files

The second file you need is the list containing the information about the archive, the tree or the files inside your backups. Here is an example with the full list of the parameters available for now:

```
[archive]
size| <5m
mode| 755
uid| 5000
gid| 5001
owner| chaica
group| sysadmin
sha1| e0f58dcc57caad2182f701eb63f0c81f347d3fe5
mtime| 1425890843.0
outdated| 2 months

[files]
foos/foo|
foos/foo1| >105k type|f mode|755 uid|5022 gid|5023 owner|chaica group|sysadmin unexpected md5|3718422a0bf93f7fc46cff6b5e660ff8
```

- **[archive]** section hosts the parameter for the archive itself. This section is not mandatory if you do not need it.
- **size** defines what the archive size should be. You can specify <,> or =. Default value is expressed in bytes, also available are (k)ilo, (m)ega, (g)iga, (p)eta,(e)xa, (z)etta and (y)ottabyte.
- **mode** is for the expected mode of the archive.
- **uid** is for the expected uid of the archive.
- **gid** is for the expected gid of the archive.
- **owner** is the name of the owner of this file in the archive.
- **group** is the name of the owner group of this file in the archive.
- **sha1** is for the expected md5 hash sum of the archive. Also available is sha1, sha224, sha256, sha384, sha512.
- **mtime** it is the posix timestamp of the last modification of the archive. It is usually automatically generated.
- **outdated** takes a duration starting from the mtime of the archive. Afther this duration, a warning is triggered to warn that the archive is outdated.
- **[files]** section stands for the files inside the archive or the tree of directories and files. This section is not mandatory if you do not need it.
- **foos/foo|** means this file has to exist in the backup, whatever it is.
- **foos/foo1| >105k** defines that the file size of foos/foo1 in the archive should be strictly bigger than 105 kilobytes. You can specify <,> or =. Default value is expressed in bytes, also available are (k)ilo, (m)ega, (g)iga, (p)eta,(e)xa, (z)etta and (y)ottabyte.
- **foos/foo1| type|f** means the file foos/foo1 is expected to be of type f. Several types are allowed : f for files, d for directory, s for symbolic link, k for socket, b for block, c for character.
- **foos/foo1| mode|755** means the file foos/foo1 is expected to have the mode 755 (meaning read, write and execute for the owner, read and execute for the group owner, read and execute for the others). All values respecting this octal representation (including values with setuid bit on four digits) is allowed.
- **foos/foo1| uid|5022** means the file foos/foo1 is expected to have a uid of 5022.
- **foos/foo1| gid|5023** means the file foos/foo1 is expected to have a gid of 5023.

- **foos/foo1| owner|chaica** means the file foos/foo1 is expected to be owned by the user with the name chaica.
- **foos/foo1| group|sysadmin** means the file foos/foo1 is expected to be owned by the owner group with the name sysadmin.
- **foo/bar| unexpected** means that foo/bar is unexpected in this archive of tree of directories and files and that an alert should be launched about it.
- **foos/foo1| md5|hashsum** means the file foos/foo1 is expected to have a md5 hash sum of "hashsum". Also available is sha1, sha224, sha256, sha384, sha512.

Use Backup Checker

Two uses of Backup Checker are available:

- generate a description of what's inside the archive
- scan the content of an archive to compare with the associated description

Generate a list of files within a backup

Generate the configuration files and the list of files inside for a given archive

Starting from 0.4, Backup Checker is able to generate the configuration of a backup and the associated list of files within this backup.

Use the following command to generate the list of files:

```
$ backupchecker -G mybackup.tar.gz
$ ls
mybackup.tar.gz mybackup.list mybackup.conf
```

- **mybackup.conf** is the configuration file associated to your archive. See *Configure Backup Checker* section for more details.
- **mybackup.list** is the list of files inside your archive. See *Configure Backup Checker* section for more details.

Generate the configuration files through SSH for a remote archive

Generate the configuration files for an archive located on a remote server through SSH:

```
$ ssh -q server "cat /tmp/backup.tar.gz" | ./backupchecker -G -
```

Again, don't forget the last - character in order to trigger the stream mode. By the very nature of the Unix stream, some options are not available using the stream mode. The most annoying one is the feature allowing to compute the hash sums of files inside an archive.

While generating, compute the hash sums of all files in the archive

Backup Checker is able to compute the hash sums of all files inside an archive. That was the default from the start of the project to the version 0.9. Given the fact this behaviour heavily loads the computer backupchecker runs on and that the final list of files is protected by a sha512 hash sum written in the associated configuration file (e.g yourbackup.conf), it is safe to make this behaviour optional starting from the version 0.10. The associated options is `--hashes` or `-H`:

```
$ backupchecker --hashes -G mybackup.tar.gz
$ # or
$ backupchecker -H -G mybackup.tar.gz
```

Specify that backupchecker need to compute the hash sums of some files inside the archive

Backup Checker starting from the version 0.10 by default does not compute any more the hash sum of every files inside an archive except if you use the `--hashes` option (heavy compute time for big archives). But you can specify to compute the hash sums of some files - either using the path or a glob syntax - in a list of files you provide thanks to the `--exceptions-file` option:

```
$ cat archive-exceptions.list
[files]
archive/foo| sha1
archive/bar/*.txt| sha256
$ backupchecker --exceptions-file archive-exceptions.list -G archive.tar.gz
```

The result of this command will be two files : the usual configuration file and the list of files inside the archive where only archive/foo and archive/bar/*.txt will have a hash sum.

Switch the delimiter of fields in the list of files

You can also modifiy the default delimiter (‘|’) that backupchecker uses and specify your own with the `-d` or `--delimiter` option:

```
$ backupchecker -d @ -G myarchive.tar.bz
$ # or
$ backupchecker  --delimiter @ -c myconfs/myconf.conf
```

We use @ as the default delimiter for the two commands just above.

Specify the names of configuration files

By default, while generating the configuration files of an archive, the names of the generated files are given from the name of the archive. If you need to change these name, use the `--configuration-name` option:

```
$ backupchecker --configuration-name backup -G archive.tar.gz
$ ls *.conf *.list
$ backup.conf backup.list
```

Scan the content of an archive to compare with the associated description

Common use case

You launch the scan mode of Backup Checker from the command line with the following command:

```
$ backupchecker -c myconfs/
```

The option `-c` or `--configpath` specifies a path to a directory where your Backup Checker configurations are stored. If any alert is triggered, it will appear in the your current working directory in a file named a.out. Relative or absolute paths are accepted.

You can also specify your own output file:

```
$ backupchecker -c myconfs/ -l output/backupchecker.log
```

The option `-l` or `--log` specifies your own output file.

Verify a remote archive

Verify an archive located on a remote server from a FTP server:

```
$ wget --quiet -O - ftp://user:pass@server/backup.tar.gz | ./backupchecker -c /path/
↪to/conf/dir -
```

Don't forget the last - character, triggering the stream mode of Backup Checker.

Change the path to the configuration file and the list of files for a given archive

By default, the files containing the different parameters of the content of the archive and the configuration file are created in the same directory as the archive itself. From Backup Checker 0.9, you can specifiy a custom directory for the configuration file (the `-C` option), for the list of files (the `-L` option) or both with the `-O` option:

```
$ backupchecker -c /etc/backupchecker/ -l /var/log/backupchecker.log -C /etc/
↪backupchecker/confs/ -L /etc/backupchecker/lists/
```

The example above indicates a /etc/backupchecker/confs directory to store the configuration files of Backup Checker and a /etc/backupchecker/lists/ directory to store the list of files of Backup Checker.

Unsupported parameters for a given kind of archive

Given the very nature of the different kind of archive formats, some parameters are not supported for a given archive type (e.g for a bzip2 file, original rights and mode of the file inside the archive are not saved). An explicit warning will appear in the backup checker log file if you are using an unsupported feature for a given type of archive.

Unsupported parameters for the remote archive

By the nature of Unix streams, some options commonly available while using Backup Checker from a local host are not available from a remote host. The most annoying one is that computing the hash sums of files inside an archive is not possible for a stream.

License

This software comes under the terms of the **GPLv3+**. See the LICENSE file for the complete text of the license.

Authors

Carl Chenet <chaica@backupchecker.com>

CHAPTER 2

Indices and tables

- genindex
- modindex
- search