

Fangfrisch

Fangfrisch (German for "freshly caught") is a sibling of the [Clam Anti-Virus](#) freshclam utility. It allows downloading virus definition files that are not official ClamAV canon, e.g. from [Sanesecurity](#) and [URLhaus](#). Fangfrisch was designed with security in mind, to be run by an unprivileged user only.

Table of Contents

1. License	1
2. Update strategy	2
3. Installation	2
3.1. Create home directory	2
3.2. Prepare and activate venv	2
3.3. Install via PyPI	2
4. Installation packages	3
5. Configuration	3
5.1. Default providers	4
5.2. User-defined providers	5
5.3. Semantics	5
5.4. Proxy support	6
6. Preparing the database	6
7. Usage	6
8. Support	7
8.1. Reporting problems	7
8.2. Submitting suggestions	8
Appendix A: Default configuration	8
Appendix B: Effective configuration	8
Appendix C: Database structure	8
C.1. Accessing mappings	8

1. License

Copyright © 2020-2023 Ralph Seichter

This file is part of "Fangfrisch".

Fangfrisch is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

Fangfrisch is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with Fangfrisch. If not, see <https://www.gnu.org/licenses/>.

2. Update strategy

Fangfrisch is expected to run periodically, e.g. using [cron](#). Download attempts are recorded in a database and new attempts are only made after the defined age threshold is reached. Fangfrisch will attempt to download digests first (if available upstream), and only retrieve corresponding virus definition files when their recorded digest changes, minimising transfer volumes.

3. Installation

Fangfrisch requires Python 3.7 or newer. Python 3.6 security support, i.e. its final lifecycle stage, was terminated in 2021 by the Python authors. Please do not ask me to modify Fangfrisch in order to support ancient Python versions, which pose a potential security risk to you.

The recommended installation method is using the [pip command](#) in a virtual Python environment. Here is an example listing of commands for BASH, to be executed as root, assuming that you will be running Fangfrisch as an unprivileged user who is member of the **clamav** group:

3.1. Create home directory

```
mkdir -m 0770 -p /var/lib/fangfrisch
chgrp clamav /var/lib/fangfrisch
```

This will grant group members the necessary write access to create the database (see [Section 6](#)).

3.2. Prepare and activate venv

```
cd /var/lib/fangfrisch
python3 -m venv venv
source venv/bin/activate
```

3.3. Install via [PyPI](#)

```
pip install fangfrisch
```

This step will also create an executable launcher script [venv/bin/fangfrisch](#).

4. Installation packages

As an alternative to pip-based installation, there are packages available for the following Linux distributions:

- Archlinux: [packages/python-fangfrisch](#). Support contact: Archlinux package maintainer.
- Clear Linux: [clearlinux-pkgs/fangfrisch](#). Support contact: Clear Linux package maintainers.
- Gentoo Linux: [app-antivirus/fangfrisch](#). Support contact: Fangfrisch author.

5. Configuration

A configuration file is mandatory, uses an INI-File-like structure and must contain a `db_url` entry. All other settings are optional. However, unless you enable one signature file provider section, Fangfrisch naturally won't do much.

Use the `--conf` command line argument (see [Section 7](#)) to specify the path to your configuration file. Note that there is no default location.

```
# Minimal example configuration, meant for testing.
```

```
[DEFAULT]
db_url = sqlite:///var/lib/fangfrisch/db.sqlite
local_directory = /var/lib/clamav
```

```
[urlhaus]
enabled = yes
```

- **cleanup**: Cleanup method used for provider sections. Default: `automatic`, alternative: `disabled`. In automatic mode, Fangfrisch will attempt to delete obsolete virus definition files whenever you disable a provider section. Should you disable this option, orphaned files will be left behind, and you need to ensure cleanup by different means.
- **db_url**: Database URL in [SQLAlchemy syntax](#). Mandatory, no default. Typically, a local [SQLite](#) database will suffice.
- **enabled**: Scan this section for URLs? Default: `false`.
- **integrity_check**: Mechanism for integrity checks. Default: `sha256`. You can use `disabled` if the signature file provider offers no checksums.
- **interval**: Interval between downloads. Defaults are provider-dependent. Values can be expressed in human-readable form (e.g. `12h` or `45m`). Please respect the limits set by each provider.
- **local_directory**: Downloaded files are stored here. No default, so the current working directory of the Python process is used. As this can vary depending on how you launch Fangfrisch, it is highly recommended to define an absolute path like `/var/lib/clamav` instead. You can override this option in provider sections to separate downloads based on origin.

- **log_format:** See [Formatter class](#) documentation for details. Fangfrisch uses sensible defaults depending on the selected log method.
- **log_level:** Choose one of `DEBUG`, `INFO`, `WARNING` (default), `ERROR` or `FATAL`.
- **log_method:** Either `console` (default, meaning stdout/stderr) or `syslog`. For the latter, you can also specify a **log_target**.
- **log_target:** The `syslog` target address. Typical values are `/dev/log` (local Linux domain socket), `localhost` or `host.domain.tld:udpport`. If no target is specified, `localhost` is assumed. The UDP port number defaults to 514.
- **max_size:** Maximum expected file size. The default is `10MB`, but all predefined providers have individual size limits (see [Appendix A](#)). Values can be expressed in human-readable form (e.g. `250KB` or `3MB`). Fangfrisch attempts to inspect the content length before downloading virus signature files so as not to download files larger than the defined limit. If providers don't respond with content length information, Fangfrisch will log a warning but download the data anyway.
- **on_update_exec:** If any files were downloaded during a pass, a command can be executed in after the pass finishes. No default. A typical value is `clamdscan --reload`.
- **on_update_timeout:** Timeout for the `on_update_exec` command, in seconds. Default: 30.

See [here](#) for details about the configuration parser and extended interpolation. [Section 5.3](#) provides additional information on how configuration options are interpreted.

5.1. Default providers

Fangfrisch contains internal defaults for the following providers (in alphabetical order):

- [Malwarepatrol](#)
- [Sanesecurity](#)
- [SecuriteInfo](#)
- [URLhaus](#)

The internal default values for providers can be used by specifying `enabled = yes` in the desired sections. Some providers require additional configuration as shown in the following example.

```
# Example configuration

[DEFAULT]
db_url = sqlite:///var/lib/fangfrisch/db.sqlite

# The following settings are optional. Other sections inherit
# values from DEFAULT and may also overwrite values.

local_directory = /var/lib/clamav
max_size = 5MB
on_update_exec = clamdscan --reload
on_update_timeout = 42
```

```
[malwarepatrol]
enabled = yes
# Replace with your personal Malwarepatrol receipt
receipt = abcd1234

[sanesecurity]
enabled = yes

[securiteinfo]
enabled = yes
# Replace with your personal SecuriteInfo customer ID
customer_id = abcdef123456

[urlhaus]
enabled = yes
max_size = 2MB
```

5.2. User-defined providers

Fangfrisch is of course not limited to the internal defaults. You can define as many additional virus definition providers as you like. The following defines a fictional provider:

```
[fictionalprovider]
enabled = yes
integrity_check = md5
interval = 90m
prefix = http://fictional-provider.tld/clamav-unofficial/

# Reference the defined prefix in URL definitions. Values in
# other sections can be referenced using ${section:option}.
url_eggs = ${prefix}eggs.ndb
url_spam = ${prefix}spam.hdb

# Override local file name for url_spam
filename_spam = spam_spam_spam_lovely_spam.db

# Execute command after each fresh download from url_eggs
on_update_eggs = echo Fresh eggs in {path}
```

5.3. Semantics

Fangfrisch will scan enabled sections for lines prefixed with `url_` to determine download sources for virus definition files.

- The value of `integrity_check` determines both the expected filename suffix for digests and the hashing mechanism used for verification.

- Local file names will be determined by parsing URLs, but can be manually overridden. To change the file name for `url_xyz`, set `filename_xyz` to the desired value.
- To launch a command after data was downloaded for `url_xyz`, define `on_update_xyz`. The command string may contain a `{path}` placeholder, which will be substituted with the full path of the downloaded file.

You can disable refresh operations for selected URLs by assigning either an empty value or setting it to `url_xyz = disabled`. Note that disabling URLs in this manner does *not* delete any previously downloaded files.

5.4. Proxy support

Fangfrisch relies on the *requests* library to download files, which supports environment variables like `HTTPS_PROXY`. Please refer to [section Advanced Usage](#), [subsection Proxies](#) in the *requests* online documentation for details.

6. Preparing the database

After completing the configuration, make sure to create the database structure by running the `initdb` command in a root shell as shown below. Running `--force initdb` will drop existing database tables. For SQLite, deleting the database file is a viable alternative.

```
sudo -u clamav -- fangfrisch --conf /etc/fangfrisch.conf initdb
```

IMPORTANT

Fangfrisch need never be run as root. Choose an unprivileged user instead (typically **clamav**).

7. Usage

You can display command line arguments as follows:

```
$ fangfrisch --help

usage: fangfrisch [-h] [-c CONF] [-f] [-p PROVIDER]
                {dumpconf,dumpmappings,initdb,refresh}

Update and verify unofficial ClamAV signatures.

positional arguments:
  {dumpconf,dumpmappings,initdb,refresh}

optional arguments:
  -h, --help            show this help message and exit
  -c CONF, --conf CONF  configuration file
  -f, --force            force action (default: False)
```

```
-p PROVIDER, --provider PROVIDER
    provider name filter (regular expression)
```

You can choose among following actions:

- **dumpconf**: Dump the effective configuration to stdout, combining both internal defaults and your own settings. The effective configuration for the example shown in [Section 5](#) is available in [Appendix B](#).
- **dumpmappings**: Dump URL-to-filepath mappings, as recorded in the database refresh log, to stdout. See [Appendix C](#) for details.
- **initdb**: Create the database structure. This needs to be run only once, before the first refresh. Using the **--force** option will drop existing tables from the database.
- **refresh**: Refresh the configured URLs. The **--force** option can be used to override download interval settings.

As stated before, Fangfrisch is typically run using cron. An example crontab looks like this:

```
HOME=/var/lib/fangfrisch
LOG_LEVEL=INFO
# minute hour day-of-month month day-of-week user command
*/15 * * * * clamav venv/bin/fangfrisch --conf /etc/fangfrisch.conf refresh
```

8. Support

The project is hosted on [GitHub](#). Before opening tickets or contacting the author, *always* check [existing issues](#) first, including closed ones. This is not meant to discourage you; it just saves time and effort for all involved. Please contact the author [Ralph Seichter](#) only after having done your "research". Thank you.

8.1. Reporting problems

If you experience problems, please start by trying to figure out underlying issues on your own. Running with DEBUG level logging helps with that. Should your efforts fail, consider filing a GitHub issue. Each issue needs to answer the questions listed below.

If you answer question number 1, 2 or 3 with "no", do not file an issue. Please answer all other questions as detailed as you can, within reason.

1. Have you checked the documentation?
2. Have you checked all existing issues, including closed ones?
3. Have you done your personal best to resolve the issue on your own?
4. What exactly did you do?
5. What did you expect to happen?

6. What happened instead?
7. What was your exact setup (operating system, Python core version, Python module versions)?

8.2. Submitting suggestions

The list of questions is shorter, but important nonetheless:

1. Have you checked the documentation?
2. Have you checked all existing issues, including closed ones?
3. Do you consider the suggested feature helpful for more people than just yourself?

If you answered "yes" for all questions, please explain your idea in a sufficiently thorough manner. Use examples, graphics, and whatever else you think would help others to understand your suggestion.

Appendix A: Default configuration

Fangfrisch contains the following internal configuration settings as defaults. All sections are disabled, and entries with the `!url_` prefix are included for reference only. These represent data sources which either have a high risk of false positives or are not free to use. Enabling a section will not enable these specially prefixed entries.

Appendix B: Effective configuration

The following effective configuration is the result of combining internal defaults (see [Appendix A](#)) with the example settings shown in [Section 5](#).

Appendix C: Database structure

While users can technically access the Fangfrisch backend database directly, its structure and content are considered **private**. They may change at any time, without notice. Related complaints will be filed under [SEP](#).

C.1. Accessing mappings

In contrast to direct database access, the **dumpmappings** action allows accessing selected parts of database entries in a stable manner. Specifically, it returns 3-tuples (provider name, URL, local file path). Elements are separated by horizontal tabulators to facilitate piping the output into `awk` or similar utilities. If specified, the `provider` option is interpreted as a regular expression, and only DB records with matching provider column are returned. That means if you have providers *foo* and

foobar, you need to use anchoring (e.g. `^foo$`) if you only wish to match entries for the former provider. Make sure to use quoting as required by your shell. Example usage:

```
# Print all recorded mappings for the [example] provider section.  
fangfrisch --conf /etc/fangfrisch.conf --provider '^example$' dumpmappings
```

```
# Delete all files that were downloaded by Fangfrisch.  
# DON'T EXECUTE THIS UNLESS YOU REALLY MEAN IT!  
fangfrisch --conf /etc/fangfrisch.conf dumpmappings | awk '{print $3}' | xargs /bin/rm
```