

Fangfrisch

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	2
5. Configuration	3
5.1. Default providers	3
5.2. User-defined providers	4
5.3. Semantics	5
6. Preparing the database	5
7. Usage	5
8. Contact	6
Appendix A: Default configuration	6
Appendix B: Effective configuration	8
Appendix C: Database structure	10
C.1. Accessing mappings	11

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.

1. License

Copyright © 2020 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. The recommended installation method is using the [pip command](#) in a virtual Python environment. Here is an example listing of commands for BASH:

3.1. Create home directory

```
mkdir -p /var/lib/fangfrisch
cd /var/lib/fangfrisch
```

3.2. Prepare and activate venv

```
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 user repository: [python-fangfrisch](#)
- Gentoo Linux official repository: [app-antivirus/fangfrisch](#)

Should you need support for these packages, please contact the respective Linux distribution maintainers.

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.

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

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

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

- **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.
- **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 `clamscan --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 = clamscan --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.

6. Preparing the database

After completing the configuration, make sure to create the database structure by running the `initdb` command.

```
fangfrisch --conf /etc/fangfrisch.conf initdb
```

WARNING

Fangfrisch should never be run as `root`, but as an unprivileged user like `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}

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.
- **refresh**: Refresh the configured URLs. The **force** switch can be set to force downloads regardless of local file age.

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. Contact

The project is hosted on [GitHub](#). If you experience problems or have suggestions, you should [file an issue](#). When doing so, please always check existing issues first, including closed ones. This saves time and effort for all involved. To contact the author Ralph Seichter directly you can send email to fangfrisch@seichter.de.

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.

```
[DEFAULT]
enabled = false
integrity_check = sha256
max_size = 10MB

[malwarepatrol]
interval = 1d
integrity_check = disabled
product = 8
receipt = you_forgot_to_configure_receipt
prefix =
https://lists.malwarepatrol.net/cgi/getfile?product=${product}&receipt=${receipt}&list
=
url_clamav_basic = ${prefix}clamav_basic
filename_clamav_basic = malwarepatrol.db

[sanesecurity]
interval = 2h
prefix = http://ftp.swin.edu.au/sanesecurity/
!url_foxhole_all_cdb = ${prefix}foxhole_all.cdb
!url_foxhole_all_ndb = ${prefix}foxhole_all.ndb
!url_foxhole_mail = ${prefix}foxhole_mail.cdb
!url_winnow_phish_complete = ${prefix}winnow_phish_complete.ndb
url_badmacro = ${prefix}badmacro.ndb
url_blurl = ${prefix}blurl.ndb
url_bofhland_cracked_url = ${prefix}bofhland_cracked_URL.ndb
url_bofhland_malware_attach = ${prefix}bofhland_malware_attach.hdb
url_bofhland_malware_url = ${prefix}bofhland_malware_URL.ndb
url_bofhland_phishing_url = ${prefix}bofhland_phishing_URL.ndb
url_crdfam_clamav = ${prefix}crdfam.clamav.hdb
url_doppelstern_hdb = ${prefix}doppelstern.hdb
url_doppelstern_ndb = ${prefix}doppelstern.ndb
url_doppelstern_phishtank = ${prefix}doppelstern-phishtank.ndb
url_foxhole_filename = ${prefix}foxhole_filename.cdb
url_foxhole_generic = ${prefix}foxhole_generic.cdb
url_foxhole_js_cdb = ${prefix}foxhole_js.cdb
url_foxhole_js_ndb = ${prefix}foxhole_js.ndb
url_hackingteam = ${prefix}hackingteam.hsb
url_junk = ${prefix}junk.ndb
url_jurlbl = ${prefix}jurlbl.ndb
url_jurlbla = ${prefix}jurlbla.ndb
url_lott = ${prefix}lott.ndb
url_malwareexpert_fp = ${prefix}malware.expert.fp
url_malwareexpert_hdb = ${prefix}malware.expert.hdb
url_malwareexpert_ldb = ${prefix}malware.expert.ldb
url_malwareexpert_ndb = ${prefix}malware.expert.ndb
url_malwarehash = ${prefix}malwarehash.hsb
```

```
url_phish = ${prefix}phish.ndb
url_phishtank = ${prefix}phishtank.ndb
url_porcupine = ${prefix}porcupine.ndb
url_rogue = ${prefix}rogue.hdb
url_scam = ${prefix}scam.ndb
url_scannailer = ${prefix}scannailer.ndb
url_shelter = ${prefix}shelter.ldb
url_spamattach = ${prefix}spamattach.hdb
url_spamimg = ${prefix}spamimg.hdb
url_spear = ${prefix}spear.ndb
url_spearl = ${prefix}spearl.ndb
url_winnow_attachments = ${prefix}winnow.attachments.hdb
url_winnow_bad_cw = ${prefix}winnow_bad_cw.hdb
url_winnow_extended_malware = ${prefix}winnow_extended_malware.hdb
url_winnow_extended_malware_links = ${prefix}winnow_extended_malware_links.ndb
url_winnow_malware = ${prefix}winnow_malware.hdb
url_winnow_malware_links = ${prefix}winnow_malware_links.ndb
url_winnow_phish_complete_url = ${prefix}winnow_phish_complete_url.ndb
url_winnow_spam_complete = ${prefix}winnow_spam_complete.ndb
```

```
[securiteinfo]
customer_id = you_forgot_to_configure_customer_id
interval = 1h
max_size = 20MB
prefix = https://www.securiteinfo.com/get/signatures/${customer_id}/
!url_0hour = ${prefix}securiteinfo0hour.hdb
!url_old = ${prefix}securiteinfoold.hdb
!url_securiteinfo_mdb = ${prefix}securiteinfo.mdb
!url_spam_marketing = ${prefix}spam_marketing.ndb
url_android = ${prefix}securiteinfoandroid.hdb
url_ascii = ${prefix}securiteinfoascii.hdb
url_html = ${prefix}securiteinfohtml.hdb
url_javascript = ${prefix}javascript.ndb
url_pdf = ${prefix}securiteinfopdf.hdb
url_securiteinfo = ${prefix}securiteinfo.hdb
url_securiteinfo_ign2 = ${prefix}securiteinfo.ign2
```

```
[urlhaus]
interval = 10m
url_urlhaus = https://urlhaus.abuse.ch/downloads/urlhaus.ndb
```

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](#).

```
[DEFAULT]
enabled = false
integrity_check = sha256
```



```
max_size = 5MB
db_url = sqlite:///var/lib/fangfrisch/db.sqlite
local_directory = /var/lib/clamav
on_update_exec = clamdscan --reload
on_update_timeout = 42

[malwarepatrol]
interval = 1d
integrity_check = disabled
product = 8
receipt = abcd1234
prefix =
https://lists.malwarepatrol.net/cgi/getfile?product=${product}&receipt=${receipt}&list
=
url_clamav_basic = ${prefix}clamav_basic
filename_clamav_basic = malwarepatrol.db
enabled = yes

[sanesecurity]
interval = 2h
prefix = http://ftp.swin.edu.au/sanesecurity/
!url_foxhole_all_cdb = ${prefix}foxhole_all.cdb
!url_foxhole_all_ndb = ${prefix}foxhole_all.ndb
!url_foxhole_mail = ${prefix}foxhole_mail.cdb
!url_winnow_phish_complete = ${prefix}winnow_phish_complete.ndb
url_badmacro = ${prefix}badmacro.ndb
url_blurl = ${prefix}blurl.ndb
url_bofhland_cracked_url = ${prefix}bofhland_cracked_URL.ndb
url_bofhland_malware_attach = ${prefix}bofhland_malware_attach.hdb
url_bofhland_malware_url = ${prefix}bofhland_malware_URL.ndb
url_bofhland_phishing_url = ${prefix}bofhland_phishing_URL.ndb
url_crdfam_clamav = ${prefix}crdfam.clamav.hdb
url_doppelstern_hdb = ${prefix}doppelstern.hdb
url_doppelstern_ndb = ${prefix}doppelstern.ndb
url_doppelstern_phishtank = ${prefix}doppelstern-phishtank.ndb
url_foxhole_filename = ${prefix}foxhole_filename.cdb
url_foxhole_generic = ${prefix}foxhole_generic.cdb
url_foxhole_js_cdb = ${prefix}foxhole_js.cdb
url_foxhole_js_ndb = ${prefix}foxhole_js.ndb
url_hackingteam = ${prefix}hackingteam.hsb
url_junk = ${prefix}junk.ndb
url_jurlbl = ${prefix}jurlbl.ndb
url_jurlbla = ${prefix}jurlbla.ndb
url_lott = ${prefix}lott.ndb
url_malwareexpert_fp = ${prefix}malware.expert.fp
url_malwareexpert_hdb = ${prefix}malware.expert.hdb
url_malwareexpert_ldb = ${prefix}malware.expert.ldb
url_malwareexpert_ndb = ${prefix}malware.expert.ndb
url_malwarehash = ${prefix}malwarehash.hsb
url_phish = ${prefix}phish.ndb
url_phishtank = ${prefix}phishtank.ndb
```

```
url_porcupine = ${prefix}porcupine.ndb
url_rogue = ${prefix}rogue.hdb
url_scam = ${prefix}scam.ndb
url_scamnailer = ${prefix}scamnailer.ndb
url_shelter = ${prefix}shelter.ldb
url_spamattach = ${prefix}spamattach.hdb
url_spamimg = ${prefix}spamimg.hdb
url_spear = ${prefix}spear.ndb
url_spearl = ${prefix}spearl.ndb
url_winnow_attachments = ${prefix}winnow.attachments.hdb
url_winnow_bad_cw = ${prefix}winnow_bad_cw.hdb
url_winnow_extended_malware = ${prefix}winnow_extended_malware.hdb
url_winnow_extended_malware_links = ${prefix}winnow_extended_malware_links.ndb
url_winnow_malware = ${prefix}winnow_malware.hdb
url_winnow_malware_links = ${prefix}winnow_malware_links.ndb
url_winnow_phish_complete_url = ${prefix}winnow_phish_complete_url.ndb
url_winnow_spam_complete = ${prefix}winnow_spam_complete.ndb
enabled = yes
```

```
[securiteinfo]
customer_id = abcdef123456
interval = 1h
max_size = 20MB
prefix = https://www.securiteinfo.com/get/signatures/${customer_id}/
!url_0hour = ${prefix}securiteinfo0hour.hdb
!url_old = ${prefix}securiteinfoold.hdb
!url_securiteinfo_mdb = ${prefix}securiteinfo.mdb
!url_spam_marketing = ${prefix}spam_marketing.ndb
url_android = ${prefix}securiteinfoandroid.hdb
url_ascii = ${prefix}securiteinfoascii.hdb
url_html = ${prefix}securiteinfohtml.hdb
url_javascript = ${prefix}javascript.ndb
url_pdf = ${prefix}securiteinfopdf.hdb
url_securiteinfo = ${prefix}securiteinfo.hdb
url_securiteinfo_ign2 = ${prefix}securiteinfo.ign2
enabled = yes
```

```
[urlhaus]
interval = 10m
url_urlhaus = https://urlhaus.abuse.ch/downloads/urlhaus.ndb
enabled = yes
max_size = 2MB
```

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](#), and possibly ridiculed.

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
```