

Fangfrisch

Fangfrisch (German for "freshly caught") is a sibling to the [Clam Anti-Virus](#) freshclam utility. It allows downloading virus definition files that are not official ClamAV canon, e.g. from [Sanesecurity](#) and [URLhaus](#).

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:

```
# Create venv home directory
mkdir -p /var/lib/fangfrisch
cd /var/lib/fangfrisch

# Prepare and activate venv
python3 -m venv venv
source venv/bin/activate

# Install from PyPI.org
pip install fangfrisch
```

4. Configuration

A configuration file is mandatory and uses an INI File structure. It must contain at least a default section with the options shown below. The suggested path is `/etc/fangfrisch.conf` but the file can be stored elsewhere if you prefer. A description of SQLAlchemy's DB URL syntax is available [here](#). Typically, a local [SQLite](#) database will suffice. Internal default values for Sanesecurity can be used by enabling an optional section as shown below.

```
# Mandatory section.
[DEFAULT]
db_url = sqlite:///var/lib/fangfrisch/fangfrisch.sqlite
# Store downloaded files here:
local_directory = /var/lib/clamav
# Optional: Local data max age (default: 1d)
max_age = 12h
# Optional: Download size limit in bytes per file (default: 10MB).
max_size = 500KB
# Optional: Execute a command if any data was downloaded.
on_update_exec = clamscan --reload
# Optional: Execution timeout in seconds (default: 30)
on_update_timeout = 42

# Optional section: Enable internal defaults for Sanesecurity.
[sanesecurity]
enabled = yes

# Optional section: Enable internal defaults for SecuriteInfo.
[securiteinfo]
enabled = yes
# Specify your personal customer ID here:
customer_id = abcdef123456

# Optional section: Enable internal defaults for URLhaus.
[urlhaus]
enabled = yes
```

Points of note:

- Section names are case-sensitive.
- Max age is specified in minutes.
- Integrity checks can be turned off by using the value `disabled`.

You can also add your own sections for additional virus definition providers. See [here](#) for a detailed description of the configuration file parser. The parser uses extended interpolation, i.e. `${section:option}` style references to options defined elsewhere in the file.

```
[fictional_provider]
enabled = yes
integrity_check = md5
# Download no more than once ever 8 hours
max_age = 8h
prefix = http://fictional-provider.tld/clamav-unofficial/
url_eggs = ${prefix}eggs.ndb
url_spam = ${prefix}spam.hdb
```

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.

5. Prepare the database

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

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

WARNING

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

6. Usage

You can display command line arguments as follows:

```
$ python -m fangfrisch --help

usage: __main__.py [-h] [-c CONF] [-f] {dumpconf,initdb,refresh}

positional arguments:
  {dumpconf,initdb,refresh}
                        Action to perform

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

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 4](#) is available in [Appendix A](#).
- **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, Fanfrisch is typically run using cron. An example crontab looks like this:

```
# minute hour day-of-month month day-of-week user command
*/30 * * * * clamav python -m fangfrisch --conf /etc/fangfrisch.conf refresh
```

7. Contact

If you come across bugs or have suggestions, you should [file an issue](#) on GitHub. To avoid unnecessary effort for all involved parties, please *always* check existing issues first, including closed ones. To contact the author Ralph Seichter directly you can send email to fangfrisch@seichter.de.

Appendix A: Effective configuration

The following effective configuration is the result of combining internal defaults with the example settings shown in [Section 4](#).

```
[DEFAULT]
enabled = no
integrity_check = sha256
max_age = 12h
max_size = 500KB
on_update_exec = clamdscan --reload
on_update_timeout = 42
```

```
db_url = sqlite:///var/lib/fangfrisch/fangfrisch.sqlite
local_directory = /var/lib/clamav
```

```
[sanesecurity]
```

```
max_age = 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
enabled = yes
```

[securiteinfo]

```
customer_id = abcdef123456
max_age = 12h
prefix = https://www.securiteinfo.com/get/signatures/${customer_id}/
!url_old = ${prefix}securiteinfoold.hdb
!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]

```
max_age = 10m
url_urlhaus = https://urlhaus.abuse.ch/downloads/urlhaus.ndb
enabled = yes
```