

**NAME**

**filedaemon** – Invoke another program on files matching a glob pattern

**SYNOPSIS**

```
filedaemon --in INPUT_GLOB [--out OUTPUT_DIR]
           --nextdir PROCESSED_INPUT_DIRECTORY
           --faildir FAILED_INPUT_DIRECTORY
           [--extension OUTPUT_EXTENSION]
           [--poll POLLING_DELAY] [--lock]
           [--pidfile PID_FILE] [--no-daemon]
           [--log LOG_SPECIFIER] [--loglevel LOG_LEVEL]
           [--verbose] [--version]
           -- PROGRAM [PROGRAM_ARGS]
```

**DESCRIPTION**

**filedaemon** operates on input files matching a *glob* (3) pattern. Each matching input file is passed to a child program (specified in the PROGRAM argument) which is expected to read input data from standard input and write the results to standard output. **filedaemon**'s job is simply to handle the mechanics of directory polling, file globbing, and routing of input and output files on behalf of the child program.

Use of two dashes (--) after all filedaemon command-line switches allows PROGRAM\_ARGS to be interpreted by the PROGRAM rather than **filedaemon** itself. While they are not strictly required if you do not need to pass arguments to PROGRAM, they should be used for consistency.

**OPTIONS****I/O Options**

These options control the file locations for input and output files.

**--in** *INPUT\_GLOB*

*INPUT\_GLOB* is a file glob pattern, which must be escaped or quoted to prevent the shell expansion. Files that match this pattern will be processed by filedaemon. This option is required.

**--out** *OUTPUT\_DIR*

*OUTPUT\_DIR* is a directory in which to place output files. The directory must exist prior to invoking **filedaemon**, and any files in the directory that match the names of output files will be overwritten. If not specified, the current working directory will be used.

**Daemon Options**

These options control what is done with processed input files, file locking, and other options to facilitate operation as a file daemon.

**--nextdir** *PROCESSED\_INPUT\_DIRECTORY*

When reading from files, if this option is present, input files will be moved to *PROCESSED\_INPUT\_DIRECTORY* after they are successfully processed. The special string **delete** will cause successfully processed input to be removed instead. This option is required.

**--faildir** *FAILED\_INPUT\_DIRECTORY*

When reading from files, if this option is present, input files will be moved to *FAILED\_INPUT\_DIRECTORY* if processing failed. The special string **delete** will cause failed input to be removed instead. This option is required.

**--extension** *OUTPUT\_EXTENSION*

Replace the input file's extension with *OUTPUT\_EXTENSION*. For example, if an input file is named "foo.txt", and this option is "out", then the output file will be named "foo.out". If the input file has no extension, then this option's value will be appended to the filename. If this option is not specified, output files will have the same name as input files, except in the case when the **--out** option is also not specified, in which case output files will be given a .out extension to avoid clobbering input files.

**--poll** *POLLING\_DELAY*

*POLLING\_DELAY* is the polling delay in seconds; how long filedaemon will wait for new input when none is available. The default is 30 seconds.

**--lock**

Use lockfiles for concurrent file access protection. filedaemon will not process an input file for which a lock file exists, but will do so when the lock file is removed. Lock files are written to the same directory as the input file, and the filename is the input filename (including any extensions) with “.lock” appended.

**--pidfile=***PIDFILE*

Write the process identifier of the filedaemon process to *PIDFILE*. This option exists to facilitate the termination of the forked filedaemon process by shutdown scripts.

**--no-daemon**

Do not actually daemonize. Mainly useful for testing/debugging.

**Logging Options**

These options are used to specify how log messages are routed. filedaemon can log to standard error, regular files, or the UNIX syslog facility.

**--log** *LOG\_SPECIFIER*

Specifies destination for log messages. *LOG\_SPECIFIER* can be a *syslog* (3) facility name, the special value **stderr** for standard error, or the *absolute* path to a file for file logging. Standard error logging is only available in **--daemon** mode if **--foreground** is present. The default log specifier is **stderr** if available, **user** otherwise.

**--loglevel** *LOG\_LEVEL*

Specify minimum level for logged messages. In increasing levels of verbosity, the supported log levels are **quiet**, **error**, **critical**, **warning**, **message**, **info**, and **debug**. The default logging level is **warning**.

**--verbose**

Equivalent to **--loglevel debug**.

**--version**

If present, print version and copyright information to standard error and exit.

**EXAMPLES**

The following will invoke “yaf” on .pcap files in the /in directory, writing results to the /out directory with a .yaf extension. Processed input files will be moved to the /next directory, and failed input files will be moved to the /fail directory.

```
filedaemon -i "/in/*.txt" -o /out -e yaf \
--nextdir /next--faildir /fail -- yaf
```

The same as the first example, but with all input files deleted after processing:

```
filedaemon -i "/in/*.txt" -o /out -e yaf \
--nextdir delete --faildir delete -- yaf
```

The same as the first example, but with a polling delay of 10 seconds (instead of the default 30) and an additional **--mac** parameter passed to yaf:

```
filedaemon -i "/in/*.txt" -o /out -e yaf -p 10 \
--nextdir /next --faildir /fail -- yaf --mac
```

**BUGS**

Known issues are listed in the **README** file in the Airframe source distribution. Note that Airframe should be considered alpha-quality software; not every conceivable input and aggregation is exhaustively tested at each release, and specific features may be completely untested. Please be mindful of this before deploying Airframe in production environments. Bug reports and feature requests may be sent directly to the author, Tony Cebzanov, via email at <tonyc@cert.org>.

**AUTHORS**

Tony Cebzanov <tonyc@cert.org> and Brian Trammell <bht@cert.org>, for the CERT Network Situational Awareness Group, <http://www.cert.org/netsa>.

**SEE ALSO**

*glob* (3), *airdaemon* (1)