Passive Check Runner

Release 1.0

Maarten

Apr 01, 2023

CONTENTS

Pcrunner (Passive Checks Runner is a daemon and service that periodically runs Nagios / Icinga checks paralell. The results are posted via HTTPS to a NSCAweb server.

CHAPTER

ONE

FEATURES

- Runs as a daemon on Linux.
- Runs as a service on win32.
- Command line interface for single test runs and/or cron use.
- Parallel execution of check commands.
- Posts check results external commands.
- Termniation of check commands if maximum time exceeds.
- Configuration in YAML.
- Command definition in YAML or text format.

CHAPTER

TWO

CREDITS

This package was created with Cookiecutter and the audreyr/cookiecutter-pypackage project template.

2.1 Contents:

2.1.1 Installation

Virtualenv

- Create a virtual environment:

```
$ python -m venv venv
```

- Activate virtual environment:

```
$ source venv/bin/activate
```

- Install *pcrunner*:

```
(venv)$ pip install pcrunner
```

2.1.2 Usage

pcrunner can run as a daemon on Linux, as a service on win32 and has a command.

pcrunner's has sensible defaults which can be overridden by the configuration file. Most of the options in the configuration file can be overriden by command line options.

pcrunner command line arguments and options:

```
$ pcrunner --help
usage: pcrunner [-h] [-c CONFIG_FILE] [-n NSCA_WEB_URL] [-u NSCA_WEB_USERNAME]
                [-p NSCA_WEB_PASSWORD] [-o COMMAND_FILE] [-H HOSTNAME]
                [-i INTERVAL] [-m MAX_PROCS] [-e LINES_PER_POST]
                [-r RESULT_FILE] [-d RESULT_DIR] [-f PID_FILE]
                [-t HTTP_TIMEOUT] [-s MAX_LINE_SIZE] [-l LOG_FILE] [-a] [-v]
                [--version]
                [{start,stop}]
```

(continues on next page)

(continued from previous page)

```
Passive Command Runner.

positional arguments:
  {start,stop}          Start or stop pcrunner runloop

optional arguments:
  -h, --help            show this help message and exit
  -c CONFIG_FILE, --config-file CONFIG_FILE
                        Configuration file
  -n NSCA_WEB_URL, --nsca_web_url NSCA_WEB_URL
                        NSCA server url.
  -u NSCA_WEB_USERNAME, --nsca-web-username NSCA_WEB_USERNAME
                        NSCA Web username.
  -p NSCA_WEB_PASSWORD, --nsca-web-password NSCA_WEB_PASSWORD
                        NSCA Web password.
  -o COMMAND_FILE, --command-file COMMAND_FILE
                        Command file.
  -H HOSTNAME, --hostname HOSTNAME
                        Hostname excpected by Nagios/Icinga.
  -i INTERVAL, --interval INTERVAL
                        Time interval between checks in seconds.
  -m MAX_PROCS, --max-procs MAX_PROCS
                        Max processes to run simultaneously.
  -e LINES_PER_POST, --lines-per-post LINES_PER_POST
                        number of lines per HTTP post
  -r RESULT_FILE, --result-file RESULT_FILE
                        File to where results are written to when NSCA
                        webserver is not reachable.
  -d RESULT_DIR, --result-dir RESULT_DIR
                        Directory for results from external commands.
  -f PID_FILE, --pid-file PID_FILE
                        PID file
  -t HTTP_TIMEOUT, --http-timeout HTTP_TIMEOUT
                        Max secs to timeout when posting results to NSCA
                        webserver
  -s MAX_LINE_SIZE, --max-line-size MAX_LINE_SIZE
                        Maximum result data to post to NSCA webserver in bytes
                        per line.
  -l LOG_FILE, --log-file LOG_FILE
                        log file
  -a, --no-daemon       Run pcrunner in foreground
  -v, --verbose         Show verbose info (level DEBUG).
  --version             Show version
```

example config:

```
---
# NSCW Web url
# Default: http://localhost:5668/queue
nsca_web_url: http://localhost:5668/queue

# NSCW Web username
```

(continues on next page)

(continued from previous page)

```
# Default: default
nsca_web_username: default

# NSCW Web password
# Default: changeme
nsca_web_password: changeme

# hostname of local host (host that is being checked)
# Default: <gethostname>
hostname: host.example.com

# pid file
# Default: /var/run/pcrunner.pid as root
# or <OS tempdir>/pcrunner.pid'
pid_file: /var/run/pcrunner.pid

# log file, when configured don't use syslog
# Default: null
log_file: null

# Verbose logging
# Default: False
verbose: False

# File with check commands
# Default win32: <python_site-packages_dir>/pcrunner/etc/commands.yml
# Default POSIX: /etc/pcrunner/commands.yml
command_file: /etc/pcrunner/commands.yml

# Directory for results from external commands
# Must be writable for external commands and pcrunner
# Example: /var/spool/pcrunner/results
# Default: null
result_dir: /var/spool/pcrunner/results

# Temp file for results not yet uploaded to NSCA Web
# Default win32: <python_site-packages_dir>/pcrunner/data/pcrunner.dat
# Default POSIX: /var/spool/pcrunner/pcrunner.dat
result_file: /var/spool/pcrunner/pcrunner.dat

# Number of maximum process to run concurrent
# Default: CPU count
max_procs: 2

# Time interval between checks in seconds
# Default: 60
interval: 60

# Max secs to timeout when posting results to NSCA webserver
# Default: 3
http_timeout: 3
```

(continues on next page)

(continued from previous page)

```
# FQDN Syslog server
# Default: null
syslog_server: null

# Syslog server port
# Default: 514
syslog_port: 514

# Number of lines per HTTP post
# Default: 400
lines_per_post: 400

# Maximum result data to post to NSCA webserver in bytes per line.
# Default: 8192 bytes (max length of an external command)
max_line_size: 8192
```

2.1.3 pcrunner

pcrunner package

Submodules

pcrunner.configuration module

pcrunner.configuration

Global configuration handling

class pcrunner.configuration.**Config**(*args=None*, ***kwargs*)

Bases: dict

subset(**keys*, ***kwargs*)

Return a sub set of Config dict of keys if kwargs also update the returned dictionary.

update_yaml()

Read Yaml config

pcrunner.configuration.**read_check_commands**(*command_filename*)

pcrunner.configuration.**read_check_commands_txt**(*fd*)

pcrunner.configuration.**read_check_commands_yaml**(*fd*)

pcrunner.daemon module

pcrunner.daemon

Generic linux daemon base class for python 3.x.

class pcrunner.daemon.**Daemon**(*pid_file*)

Bases: object

A generic daemon class. Usage: subclass the daemon class and override the run() method.

daemonize()

Daemonize class. UNIX double fork mechanism.

delpid()

Remove pid file.

run()

You should override this method when you subclass Daemon.

It will be called after the process has been daemonized by start()

start()

Start the daemon.

stop()

Stop the daemon.

pcrunner.exception module

pcrunner.exceptions

All exceptions used in the PassiveCheckRunner code base are defined here.

exception pcrunner.exception.**PassiveCheckRunnerException**

Bases: Exception

Base exception class. All PassiveCheckRunner specific exceptions should subclass this class.

exception pcrunner.exception.**PostFailed**

Bases: *PassiveCheckRunnerException*

Raised when an error occurs when posting results:

- An error occurs while posting.
- A non 200 HTTP return code.

exception pcrunner.exception.**PostResultTooBig**

Bases: *PassiveCheckRunnerException*

Raised when post result are bigger then max_post_size.

pcrunner.main module

pcrunner.main

Main entry point for the pcrunner command.

class pcrunner.main.**Check**(*result_type*, *name*, *command*, *hostname*)

Bases: object

property duration

property elapsed

end()

property plugin_output

Checks (loosely) if performance data is form of: rx_errors=0;;;0;tx_errors=0;;;0; Otherwise remove '|' and everything after.

run()

Run the command and saves excection data

start()

terminate()

Terminates check if still running.

class pcrunner.main.**CheckRun**(*hostname*)

Bases: object

class pcrunner.main.**PassiveCheckRunner**(*nsca_web_url*, *nsca_web_username*, *nsca_web_password*, *hostname*, *command_file*, *result_file*, *result_dir*, *max_procs*, *interval*, *lines_per_post*, *pid_file*, *http_timeout*, *max_line_size*)

Bases: object

check_pcrunner_end()

check_results_from_finished_queue()

get_checks()

kill_running_checks()

property number_of_checks_finished

post(*lines*)

post_results()

post_results_previous_run()

If a previous result file exists post the results that are found in this file in chunks of number of lines per post. If post fails save failed checks in `self.results_post_failed`.

read_results_from_spool_dir()

run()

start()

Get checks, put them on start_queue and start threads. When max time reached kill all running processes.

stop()

write_failed_results()

class pcrunner.main.**PassiveCheckRunnerDaemon**(*pcrunner*)

Bases: *Daemon*

run()

You should override this method when you subclass Daemon.

It will be called after the process has been daemonized by start()

pcrunner.main.**get_syslog_socket_or_win32**()

pcrunner.main.**is_socket**(*path*)

pcrunner.main.**main**()

Entry point for the package, as defined in setup.py.

pcrunner.main.**parse_pcrunner_args**(*args*)

Parse the command-line arguments to pcrunner.

pcrunner.main.**remove_root_logger_handlers**()

pcrunner.main.**run_process**(*start_queue*, *run_queue*, *finished_queue*)

Function to be started as thread. Runs checks from start_queue, puts them on end_queue

pcrunner.main.**setup_logging**(*log_file=None*, *verbose=False*, *console=False*)

pcrunner.main.**setup_logging_with_config_opts**(*no_daemon*, *log_file*, *verbose*, *syslog_server*, *syslog_port*)

pcrunner.main.**slice_up_file**(*fd*, *number_of_lines*)

return lists

pcrunner.version module

Get the version string for the named package.

param distribution_name

The name of the distribution package to query.

return

The version string for the package as defined in the package's "Version" metadata key.

pcrunner.windows_service module

pcrunner.windows_service

Entry poing for Passive Check Runner as Windows Service

class pcrunner.windows_service.**PassiveCheckRunnerService**(**args: Any*, ***kwargs: Any*)

Bases: ServiceFramework

Passive Check Runner as Windows Service

SvcDoRun()

SvcStop()

Module contents

pcrunner

Main package for Passive Check Runner

2.1.4 History

0.4.12 (2021-10-21)

- Last python 2.7 version.

0.4.11 (2021-09-12)

- readthedocs config

0.4.10 (2021-09-11)

- Updated documentation.
- Updated deployment.

0.4.9 (2020-03-21)

- Fix #97 AttributeError: 'Popen' object has no attribute 'status_code'
- Fix #98 Python 3 TypeError: sequence item 0: expected str instance, bytes found

0.4.8 (2020-03-20)

- Fix #96 passive host check results seem to processed as service check results
- Update requirements.

0.4.7 (2019-10-26)

- Security update: Bump pyyaml from 3.12 to 5.1
- Update requirements.
- No tests for python 3.4

0.4.6 (2018-11-30)

- Better logging for invalid perf data.
- Update dev requirements.

0.4.5 (2018-11-16)

- Pypi metadata fix

0.4.3 (2018-11-16)

- Real Fix bug in logging.warning: wrong placeholder.

0.4.2 (2018-11-12)

- Fix bug in logging.warning.
- Update Python package metadata.

0.4.1 (2018-11-03)

- Have (result) data for urllib *urlencode* utf-8 encoded before (PY2) and after (PY3).
- Warn when performance data not validates (and gets removed).

0.4.0 (2018-11-03)

- Legitimately, truly and undoubtedly fixed issue #94 (we assume™, for now).
- Unicode all the way (like we never unicode before).
- No hopes on Python < 2.7 compatibility

0.3.11 (2018-10-12)

- Fix issue #94 Performance data 'sanitized' NSCAweb won't hang.

0.3.10 (2018-07-17)

- Fix RHEL 6 RPM build (make initrddir).

0.3.10 (2018-07-17)

- Fix RHEL 6 RPM build (make initrddir).

0.3.9 (2018-07-14)

- Added systemd service file for Fedora >=18 Centos >=7

0.3.8 (2018-02-09)

- Fix: issue #83

0.3.7 (2017-11-17)

- Fix: quotes in commands.txt and commands.txt seem to get ignored #82

0.3.6 (2017-11-17)

- dev requirements updates

0.3.5 (2016-12-09)

- dev requirements updates
- docs usage

0.3.4 (2016-11-18)

- dev requirements updates

0.3.3 (2016-11-11)

- dev requirements updates
- docs: download from *GitHub*

0.3.2 (2016-10-14)

- dev requirements updates

0.3.1 (2016-09-30)

- dev requirements updates

0.3.0 (2016-09-09)

- Added *–no-daemon* option for starting pcrunner's run loop in foreground.
- dev requirements updates

0.2.10 (2016-08-26)

- tox.ini updated
- removed specific version for package requirements from setup.py.
- readthedocs theme for local docs build.
- OS-X and vim files in .gitignore
- Update requirements: pytest -> 3.0.1

0.2.8 (2016-08-20)

- Updated docs

0.2.7 (2016-08-20)

- Updated project links.

0.2.6 (2016-08-20)

- Fixed ISSUE#4: commands file with extra white lines.

0.2.5 (2016-08-20)

- Updated Python installation documentation with new versions.

0.2.4 (2016-08-13)

- xrange -> range for python3 compatibility.

0.2.3 (2016-08-13)

- Travis/tox fix

0.2.2 (2016-08-13)

- ISC License

0.2.1 (2016-08-13)

- Documentation RPM build updated.

0.2.0 (2016-08-12)

- First release on PyPI.

2.2 Feedback

If you have any suggestions or questions about **Passive Check Runner** feel free to mail me ikmaarten@gmail.com

If you encounter any errors or problems with **Passive Check Runner**, please let me know! Open an Issue at the GitHub main repository.

PYTHON MODULE INDEX

p

INDEX

R

S

T

U

W