Table of Contents

**Type chapter title (level 1)1**

Type chapter title (level 2)2

Type chapter title (level 3)3

**Type chapter title (level 1)4**

Type chapter title (level 2)5

Type chapter title (level 3)6

# Installation

## Standalone

**1. Install python**

apt-get install python

or install from official site: <https://www.python.org/downloads/>

**2. Install git**

apt-get install git

or install from official site: https://git-scm.com/downloads

**3. Download project:**

git clone https://gitlab.com/invuls/pentest-projects/pcf/

**4. Go to folder:**

cd pcf

**5. Install dependencies (for unix-based systems):**

pip3 install -r requirements\_unix.txt

or windows:

pip.exe install -r requirements\_windows.txt

**6. Run initiation script:**

python3 new\_initiation.py

**7. Edit configuration:**

nano configuration/settings.ini

**8. Run:**

old version: python3 app.py

new version: python3 run.py

# Database

If you want to use PostgreSQL instead of SQLite3, change database options inside configuration/settings.ini file. Don't forget to create Postgres database and user!

# Settings

## Configuration File

/configuration/settings.txt - configuration file with major settings.

File Example

[main]

secret = frkagiz3h8a460k2wswi

debug = 0

tmp\_path =./tmp\_storage/

auto\_delete\_poc = 0

delete\_projects = 0

# waitress, flask, fastwsgi

web\_engine = waitress

webdav = 1

[logs]

logging = 1

log\_file = ./backups/console.log

[backup]

db\_backup = 1

db\_backup\_weeks = 1

db\_backup\_days = 0

db\_backup\_hours = 0

db\_backup\_minutes = 0

db\_backup\_seconds = 0

db\_backup\_amount = 3

db\_backup\_folder = ./backups/db/

[security]

basic\_auth = 0

basic\_login = pcf

basic\_password = ojsflijurngrbvijsl1

# lifetime hours (1 week = 24 \* 7 = 168 hours)

session\_lifetime = 168

csrf\_lifetime = 24

proxy\_auth = 0

proxy\_email\_header = X-Forwarded-User

enable\_form\_registration = 1

enable\_form\_login = 1

# filesystem, memory

sessions\_type = filesystem

[speedup]

external\_js = 0

external\_css = 0

external\_img = 0

one\_file\_js = 1

one\_file\_css = 1

[database]

# sqlite3, postgres

type = sqlite3

path = ./configuration/database.sqlite3

host = 0.0.0.0

port = 5432

name = pcf

login = test\_login

password = test\_password

[files]

# 5 MB = 52428800 bytes

files\_max\_size = 52428800

poc\_max\_size = 52428800

template\_max\_size = 52428800

# storage = "filesystem" or "database"

files\_storage = filesystem

poc\_storage = filesystem

template\_storage = filesystem

[ssl]

# only if web\_engine=="flask"

ssl = 0

priv\_key = ./configuration/server.key

cert = ./configuration/server.crt

[network]

host = 0.0.0.0

port = 5000

ngrok = 0

ngrok\_token =

ngrok\_url\_file = ngrok\_url.txt

[bruteforce]

top10k = ./static/files/wordlists/10-million-password-list-top-10000.txt

top1000 = ./static/files/wordlists/10-million-password-list-top-1000.txt

top100 = ./static/files/wordlists/10-million-password-list-top-100.txt

[design]

date\_format\_template = %%d/%%m/%%Y

report\_filename\_date = %%Y-%%m-%%dT%%H:%%M:%%S

[timeouts]

# timeouts in seconds

report\_timeout = 10

regexp\_timeout = 2

[main] parameters

[main]

secret = frkagiz3h8a460k2wswi

debug = 0

tmp\_path = ./tmp\_storage/

auto\_delete\_poc = 0

delete\_projects = 0

# waitress, flask, fastwsgi

web\_engine = waitress

webdav = 1

**secret** - random string, which used as a flask session. It must be changed with new\_initiation.py.

**debug** - turn on/off flask debug mode.

**tmp\_path** - path for tmp files. It can be changed for /tmp/ at Unix systems.

**auto\_delete\_poc** - turn on/off Proof-of-Concept file autoremove.

**delete\_projects** - delete project from database (when click "Delete" button) or just make them invisible to users.

**web\_engine** - engine to run application: "flask", "waitress" or "fastwsgi".

**webdav** - turn on/off WebDAV technology to connect to projects filesystem (files tab)

WebDAV

To get access to WebDAV, you need to open a link "http://pcf-ip:pcf-port/webdav/" using your WebDAV client, for example, OS file manager.

webdav\_username == pcf\_account\_email

webdav\_password == pcf\_account\_password

If you also turned on basic auth:

webdav\_username == basic\_auth\_login + ":::" + basic\_auth\_password

webdav\_password == pcf\_account\_email + ":::" + pcf\_account\_password

Structure:

project-name1\_project-uuid/

file-name1\_filename-uuid.extension

file-name2\_filename-uuid.extension

project-name2\_project-uuid/

file-name3\_filename-uuid.extension

file-name4\_filename-uuid.extension

Restrictions:

1. Move file - works!
2. Edit file - works!
3. Creating empty file - works!
4. Read file - works!
5. Rename files - works, but need a UUID of file at the end of filename (before the extension).
6. Move file from PC to webdav - works (but it in addition creates 3-5 empty files with same name due to webdav feature)
7. Delete file - works, but sometimes file can be locked (I think that fixed it, but there may be more bugs)

[logs] parameters

[logs]

logging = 1

log\_file = ./backups/console.log

**logging** - turn on/off logging to file.

**log\_file** - path to file with logs.

[backup] parameters

[backup]

db\_backup = 1

db\_backup\_weeks = 1

db\_backup\_days = 0

db\_backup\_hours = 0

db\_backup\_minutes = 0

db\_backup\_seconds = 0

db\_backup\_amount = 3

db\_backup\_folder = ./backups/db/

**db\_backup** - turn on/off autobackup.

**db\_backup\_weeks/days/hours/minutes/seconds/** - backup time range.

**db\_backup\_amount** - amount of database backup files.

[security] parameters

[security]

basic\_auth = 0

basic\_login = pcf

basic\_password = 11sflijurngrbvijsl1

# lifetime hours (1 week = 24 \* 7 = 168 hours)

session\_lifetime = 168

csrf\_lifetime = 24

proxy\_auth = 0

proxy\_email\_header = X-Forwarded-User

enable\_form\_registration = 1

enable\_form\_login = 1

# filesystem, memory

sessions\_type = filesystem

**basic\_auth** - turn on/off HTTP basic authorization.

**basic\_login/password** - HTTP basic authorization credentials.

**session\_lifetime** - Flask session lifetime (in hours).

**csrf\_lifetime** - Flask CSRF-protection token lifetime (in hours).

**proxy\_auth** - Turn on/off proxy authorization.

**proxy\_email\_header** - If proxy authorization - this header will be users' identification.

**enable\_form\_registration** - Enable/disable registration form.

**enable\_form\_login** - Enable/disable login form.

**sessions\_type** - Type of session storage. It can be "filesystem"(store all session data at filesystem) or "memory"(standard flask session cookies). Careful! Memory storage is less secure, so you need to set a complex main/secret variable.

[speedup] parameters

[speedup]

external\_js = 0

external\_css = 0

external\_img = 0

one\_file\_js = 1

one\_file\_css = 1

**external\_js** - turn on/off external JS links.

**external\_css** - turn on/off external CSS/fonts links.

**external\_img** - turn on/off external images links.

**one\_file\_js** - turn on/off joining .js files into one.

**one\_file\_css** - turn on/off joining .css files into one.

[database] parameters

[database]

# sqlite3, postgres

type = sqlite3

path = ./configuration/database.sqlite3

host = 0.0.0.0

port = 5432

name = pcf

login = test\_login

password = test\_password

**type** - database type. Now it supports only sqlite3 & postgres.

**path** - (sqlite3) path of database file.

**host** - (postgres) host of database.

**port** - (postgres) port of database.

**name** - (postgres) database name.

**login** - (postgres) database account login.

**password** - (postgres) database account password.

[files] parameters

[files]

# 5 MB = 52428800 bytes

files\_max\_size = 52428800

poc\_max\_size = 52428800

template\_max\_size = 52428800

# storage = "filesystem" or "database"

files\_storage = filesystem

poc\_storage = filesystem

template\_storage = filesystem

**files\_max\_size** - max filesize in bytes for project files.

**poc\_max\_size** - max filesize in bytes for Proof-of-Concept files.

**template\_max\_size** - max report template filesize in bytes.

**files\_storage** - storage of project files ("database" or "filesystem").

**poc\_storage** - storage of Proof-of-Concept files ("database" or "filesystem").

**template\_storage** - storage of report template files ("database" or "filesystem").

[ssl] parameters

[ssl]

# only if web\_engine=="flask"

ssl = 0

priv\_key = ./configuration/server.key

cert = ./configuration/server.crt

**ssl** - turn on/off HTTPS service (!!! Works only if main->web\_engine is "flask" !!!).

**priv\_key** - path to SSL key. Default SSL key is generated by new\_initiation.py script at path ./configuration/server.crt

**cert** - path to SSL certificate. Default SSL certificate is generated by new\_initiation.py script at path ./configuration/server.crt

**️WARNING! Flask runs slow anouph with SSL extention! Better use proxy software for this!**

[network] parameters

[network]

host = 0.0.0.0

port = 5000

ngrok = 0

ngrok\_token =

ngrok\_url\_file = ngrok\_url.txt

**host** - listening interface.

**port** - listening tcp-port.

**ngrok** - turn on/off ngrok startup ( <https://ngrok.com/> )

**ngrok\_token** - token for ngrok integration (need to set!)

**ngrok\_url\_file** - if you running service as a daemon, you will be able to check file for generated ngrok url.

[bruteforce] parameters

[bruteforce]

top10k = ./static/files/wordlists/10-million-password-list-top-10000.txt

top1000 = ./static/files/wordlists/10-million-password-list-top-1000.txt

top100 = ./static/files/wordlists/10-million-password-list-top-100.txt

**top10k** - path to top-10k passwords list.

**top1000** - path to top-1000 passwords list.

**top100** - path to top-100 passwords list.

[design] parameters

[design]

date\_format\_template = %%d/%%m/%%Y

report\_filename\_date = %%Y-%%m-%%dT%%H:%%M:%%S

**date\_format\_template** - python datetime format for projects table.

**report\_filename\_date** - python datetime format for generated report names.

[timeouts] parameters

[timeouts]

# timeouts in seconds

report\_timeout = 10

regexp\_timeout = 2

**report\_timeout** - timeout in seconds for report generation.

**regexp\_timeout** - timeout in seconds for regexp processing (issue replace rules etc).

# Database

/configuration/database.sqlite3 - default path to database.

It can be changed inside Configuration file.

# Robots.txt

/configuration/robots.txt - file which will be returned by web-server for .../robots.txt requests. You need to fill it with discovery rules which will be used by search systems such as Google or Yandex.