Table of Contents

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.