

WebScripts 2.4.10

`pip install WebScripts`

✓ Latest version

Released: Jan 19, 2022

This tool run scripts and display the result in a Web Interface.

Navigation

Project description

Release history

Download files

Project links

Homepage

Wiki

Documentation

Statistics

GitHub statistics:

★ Stars: 0

🍴 Forks: 0

📌 Open issues/PRs: 2

```
kali@kali: ~  
user@host:/home/user/$ python3 script.py  
  
WebScripts  
  
By Maurice LAMBERT.  
  
[+] 20% | ██████████ | Job1 running ...  
[+] 30% | ██████████ | Job2 running ...  
[+] 60% | ██████████ | Job3 running ...  
[+] 10% | ██████████ | Job4 running ...  
[X] 95% | ██████████ | Job5 error ...  
  
>>> █
```

Content

1. Why WebScripts
2. Usages
3. How to use WebScripts
4. WebScripts administration
5. WebScripts files
6. Features
7. Pre-integrated scripts
8. Ressources
9. Screenshot



Why WebScripts

- Useful for SOC, devops and administrators teams
 - Launch CLI scripts from Web Interface
 - Secure
 - Security tests
 - DAST
 - SAST
 - Hardening
 - Permissions management
- Configure with JSON or INI/CFG files (preconfigured)
- Easy/fast deployment (requirements: python3)
- Scripts/environments share



WebScripts

Usages

- Request APIs
- Backup data/configurations
- Share secrets/files
- Build dynamic configurations for other tools
- Automatic connection to other servers
- Analyse script results and post it on other tools
- Notifies teams on certain actions
- Report (scripts results/server states)
- Manage user access to data/secrets



WebScripts

How to use WebScripts

- Client
 - API
 - [Documentation](#)
 - [Client / Library client](#)
 - [Examples](#)
 - Web Interface
 - Request API
 - [Demonstration](#)
 - `http(s)://<webscript.server>(:8000)/web/`
- Development
 - Script
 - [Add script](#)
 - [Library / Tools](#)
 - Server
 - [Web Customization](#)
 - [Module \(example\)](#)



WebScripts

How to use WebScripts

- Authentication
 - API Key (API only)
 - Basic auth (API only)
 - Script (Web only)
 - Blacklisted after 3 authentications in 30 seconds (configurable)
- Session
 - API token
 - Web Cookie
 - 1 session by user/IP address
 - Expire after 1 hours (configurable)
- CSRF (Web only)
 - Expire after 5 minutes (configurable)
 - Reload page -> new token
 - 1 token by session



WebScripts administration

- Platform
 - Windows (python 3.8, 3.9, 3.10)
 - Linux (python 3.8, 3.9, 3.10)
 - Mac OSX (python 3.8, 3.9, 3.10)
 - Docker
- Installation
 - `pip install WebScripts`
- Upgrade
 1. Backup data directory
 2. `pip install --upgrade WebScripts`
 3. Replace data directory
 4. Restart the server



WebScripts

WebScripts administration

- Server configuration files:
 1. `./config/server.json`
 2. `./config/server.ini`
 3. `<lib path>/config/server.json`
 4. `<lib path>/config/server.ini`
- Minimum configuration (recommendation)
 - Email (notification)
 - `smtp_server`
 - `admin_adresses`
- [Add your scripts \(example\)](#)
- [Add your arguments](#)
- [Check security recommendations](#)
- Check the hardening report



WebScripts

WebScripts files

- Configurations (`config/`)
- Scripts (`scripts/`, configurable)
- Static (`static/`, configurable)
- Modules (`modules/`, configurable)
- Files are searched in
 1. Current directory
 2. Library directory
- `data/` -> only in lib directory
- `doc/` -> only in lib directory



WebScripts files

- System permissions
 - Files `<user>:<user> 400`
 - Directory `root:root 755`
- Exceptions
 - Directories
 - `data/uploads/`
`<user>:<user> 700`
 - `logs/` `<user>:<user> 700`
 - `doc/` `<user>:<user> 700`
 - Files
 - `data/id` `<user>:<user> 600`
 - `data/*.csv` `<user>:<user> 600`



WebScripts

Features

- CLI in Web Interface
- HTML script in Web Interface
- Modules for custom requests/responses
- Auto-generated documentation
- Execution history
- Download script output
- Web interface customization
- Hardening
- Themes
- Predefined values (URL query)
- Advanced arguments



Pre-integrated scripts

- Integrated
 - Account management (user)
 - Account management (administration)
 - Logs
 - Temp passwords share
 - Requests/reports
 - Files share
- In the future
 - Permanently passwords share (group)
 - User passwords manager
 - Supervision (reports and statistics only)
 - Supervision agents management
 - Encrypted file share



WebScripts

Ressources

- Documentations: [readthedocs](#) / [wiki](#)
- [Youtube](#)
- Pydoc documentation: [online](#) / [local](#)
- [Source code](#)
- [Issues](#)
- Contact:
 - mauricelambert434@gmail.com



WebScripts

Screenshot

WebScripts Server

This server can help administrators, SOC teams (Security Operations Center) and devops teams to deploy faster some scripts, to share some "console scripts" with people who have no particular computer knowledge and share environnements of scripts with their teams without install requirements on all computers of the team.



WebScripts presentation

A little Web Server to execute scripts in a Web Interface.

[Documentation on ReadTheDocs.io](#)

Welcome on the WebScripts presentation !

Featured by Maurice LAMBERT <mauricelambert434@gmail.com>, author, developer and maintainer of WebScripts.

Open source	Sources are available on github and Pypi	✓
License	Licensed under the GPL, version 3.	✓
Tools	Python, NGINX, Apache HTTPD, Docker	✓

Start presentation

Is comming...

```
>>> WebScripts_presentation.py  ExitCode: 0  Error: No errors  ExecutionTime: 00:00

This tool run scripts and display the result in a Web Interface.
[+] Run CLI scripts in Web Interface
[+] Share scripts and environnements with your teams, other teams and people without IT knowledge.
[+] Deploy easily, quickly, without requirements and securely.
[+] Customizable, highly configurable and scalable and cross-platform.
```

0

- Source code and contact is [here \(on my github\)](#).
- Documentation is [on readthedocs](#) and [on my github wiki](#).
- License: [GPL-3.0 License](#).

- Pydoc documentation index: [local](#), [online](#).
- Pydoc documentation for WebScripts.py: [local](#), [online](#).