VulSpy

Project Process & Project Documentation

By

MohammadKazem Bolandnazar

January, 2023

**Introduction**

This project developed by MohammadKazem BolandNazar for Hamravesh company.

NOTE: VulSpy is just a cool name I used for this project.

**Description**

The purpose of this program is to get a domain name with API from user and find it’s subdomains then check if they have vulnerabilities. I used Django REST framework to deploy

web app.

**RELEVANT BACKGROUND INFORMATION AND PRE-REQUISITS**

All things a user needs to know , is an API platform to send and get data using API’s. I recommend [Postman](https://www.postman.com/) because it’s so simple and practical.

For software developers they should be familiar with Python, Django, REST framework, Celery, RabbitMQ and after all docker to further understand the code.

**How It works**

Every API contains an authorize Token that’s unique for every user.

To get make a token first we make a user with `django.contrib.auth.models`

Then we give every user a token with `rest\_framework\_authtoken`

After authorization header we get request domain (as POST) in body of API.

Then we make a request object with specific uuid `ScanRequest.objects.creat()`

In tasks.py I used Python subprocess to run commands.

We give subfinder the domain and it start to work. With every subdomain we find we give it to nuclei in next function then if there was a vulnerability in app we save it in that UUID.

Ok so now we have a uuid with subdomains and vulnerabilities that is specific for user (every user have some uuids but every uuid belongs to only one user).

After that to manage requests I used celery. With celery we can offload work in python web app. I used celery to queue tasks and distributes them to the workers and with RabbitMQ massage broker, we response a status for all tasks.

**How to use run app**

There are 3 ways to run this program:

1. Clone project in your machine then install these apps:

* Python
* Pip install Django && restframework && celery && pyamqp
* Subfinder
* Nuclei
* Celery
* RabbitMQ

After these you should run:

$ rabbitmq-server

$ celery -A hamravesh worker -l info

$ python manage.py runserver 0.0.0.0:8000

1. You can clone it in your machine and build image

➜ ~ docker compose up --build

1. Or you can pull docker image:

➜ ~ docker pull kaxem/hamravesh

➜ ~ docker run -rm kaxem/hamravesh

**Challenges**

There were some challenges developing this project, such as:

Instead of copying web app to run celery in docker compose I used shell script to run all

of that commands in web container.

Subfinder and nuclei were developed with go-lang so I installed a kali-linux to install them and use them in subprocess. I know there was a better way, pulling their docker images, and runningimage in docker compose but I installed kali to use more tools like nmap and tools like that to adding them to the project in future.

there’s an issue using subfinder in project in bigger order:

it is too slow. It takes long time to find all subdomains. instead of it, I suggest [massdns](https://github.com/blechschmidt/massdns) it’s much faster than subfinder.