**Use example**

This is a short POC of the abilities that we have when using this newly created tool, in this example we will fetch from the AIP all the URLs that are related to the Qbot campaign.

We will start by creating a regex that will match the known URLs that qbot is reported in on, the regex will be:

^(?:https?:\/\/?)?(?:[^.\/\s+]+\.)+[^\/]+\/[^-\/\\_0-9]+[^\_\-\/]+\/[^.\/\-\\_0-9][a-z]+.\-\d{0,12}

Now that we have a regex expression ready, we will create a command for the tool.

python main.py -P T --tag qbot -O qbotURL.json -R "^(?:https?:\/\/?)?(?:[^.\/\s+]+\.)+[^\/]+\/[^-\/\\_0-9]+[^\_\-\/]+\/[^.\/\-\\_0-9][a-z]+.\-\d{0,12}" --fetch url

we will dissect the command to farther the understanding of the request.

**-P** – create a POST request to the API.

**T --tag qbot** – adding a filter of type tag to the query, in this case it is “qbot”.

**-O qbotURL.json –** outputting the data that is retrieved to a file of type json, many more types are available, please disclose the manual for more information.

**-R <The regex expression>** - filtering the data that was already retrieved by the POST request with the regex expression that was provided.

**--fetch url** – get the value of every key named “url” from the json that we have after all the filtering that were applied.

**data representation**

In this section we will see the data that is retrieved from the POST call, the data after the filtering, and the data in the file.

First we will pass the command line to the tool, I the image we will be able to see the arguments that were passed to the tool (image 1).

Text, application

Description automatically generated

Image 1: the parameters that are used in this command.

Here we can see the number of entries that were returned by our query to the API, only the tag “qbot” was sent to the API (image 2).

Graphical user interface, text, application

Description automatically generated

Image 2: the number of entries that the API call returned to us

Here we can see the data that we wanted to fetch from all the entries that matches our regex expression (image 3).

Text

Description automatically generated

Image 3: the URLs that passed the regex filtering.

We have outputted the data that was fetched to a file (image 4).

Graphical user interface

Description automatically generated with medium confidence

Image 4: all the data that passed the filtering stored in a file.

After retrieving all the data, we can use JSON beautifier the make it presentation ready (image 5).

A screenshot of a computer

Description automatically generated with medium confidence

Image 5: the data presented beautifully 😊.

Just to make sure that the data that was given matches out expectations we will use the tool again but now on one of the URLs that we got, we will use the first URL in the list.

Graphical user interface, application

Description automatically generatedWe will query the first URL that we got ("http://aayom.biz/eavoluptates/oditbeatae-4758271") and we will output the data to a file named (qbotURLcheck.json) (image 6), we will use JSON butifiere to make the data presentation ready (image 7).

Graphical user interface, text, application, email

Description automatically generated

Image 6: the data regarding the URL is stored in a file.