OSINT Tool Guide & Demonstration: MASTO

🦣 About Masto -

Masto provides information/intelligence on Mastodon.social users and fediverse instances (servers).
Masto OSINT Tool has been added as a Python package on PyPI --> https://pypi.org/project/masto/

- Latest version --> https://pypi.org/project/masto/2.0.5/

🚀 Masto capabilities -

Masto OSINT Tool helps to:

- Find user ID
- Find exact username match across instances (the tool currently pulls many accounts with the username **OSINT**, whereas the mastodon.social (browser search bar) returns one result, as well as returning unreliable results, such as accounts that only start with osint

- Find all accounts belonging to a user without logging in to Mastodon (**Mastodon requires users to log in and after 5 results you get**: 401 Search queries pagination is not supported without authentication
- Find username correlation (can't be found by browser)
- Check if the user is a bot
- Check if the account is a group
- Check if the account is locked
- Check if the user opted to be listed on the profile directory
- Get avatar link with an **additional choice** of opening the avatar within your browser
- Get profile creation date
- Get number of followers & following
- Get number of posts
- Get user last status date
- Get user's bio

Additional instance (server) feature -

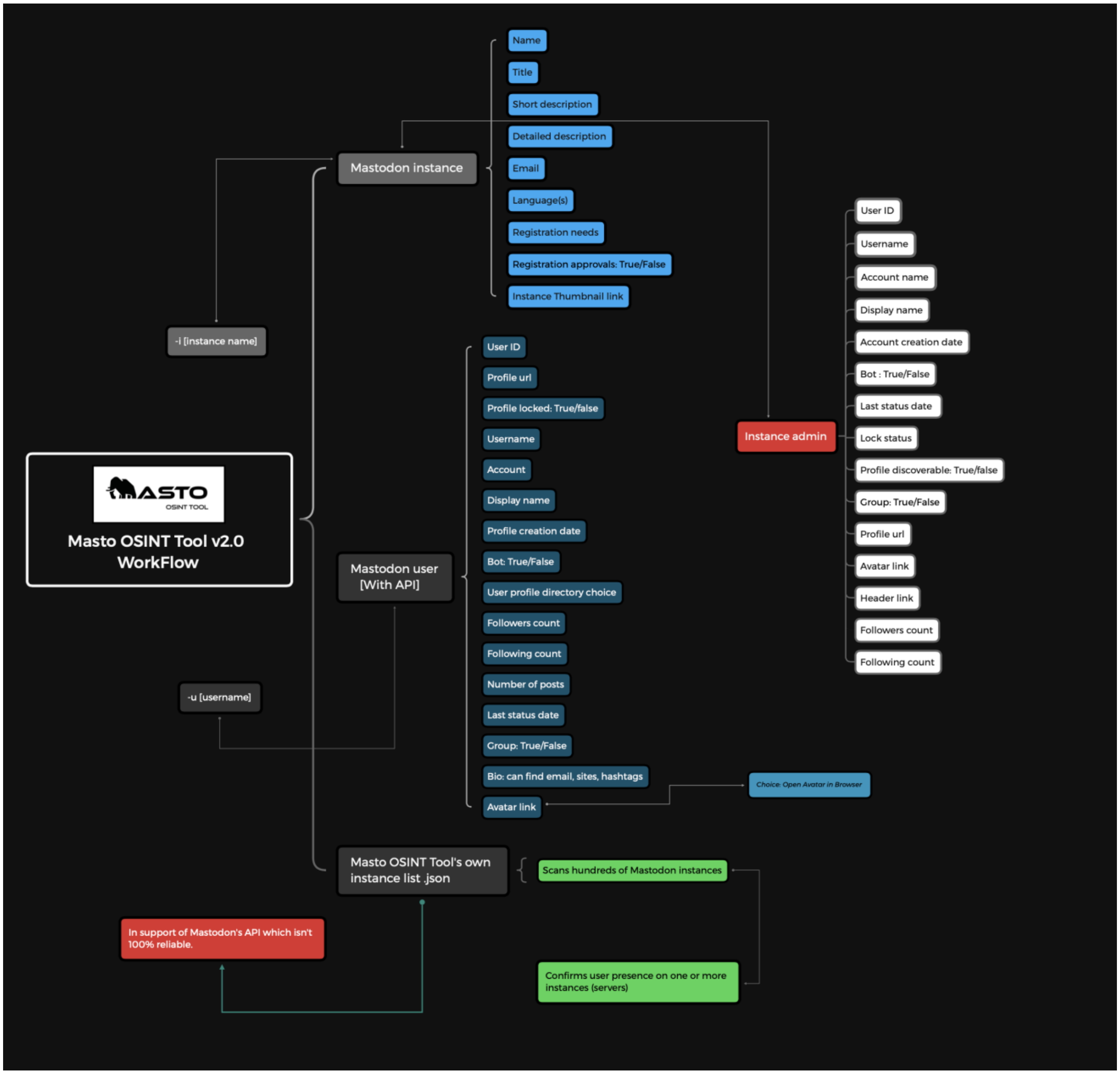
This is a nice feature, if you type social.network.europa.eu on Mastodon.social , you won't get a result as the instance is set to not discoverable.

This function helps to:

- Get information on an instance
- Get instance Admin ID
- Get instance email
- Get a short description

- Get server thumbnail link
- Get instance creation date
- Get instance language used
- Get instance admin count of followers and following
- Get instance admin last status date
- Get header image link and avatar link
- Get instance display name
- Get admin url
- Get admin avatar
- Check if instance admin account is locked
- Check if registration is required and if the admin needs to approve the request
- Check if the admin is a bot

Masto WorkFlow -



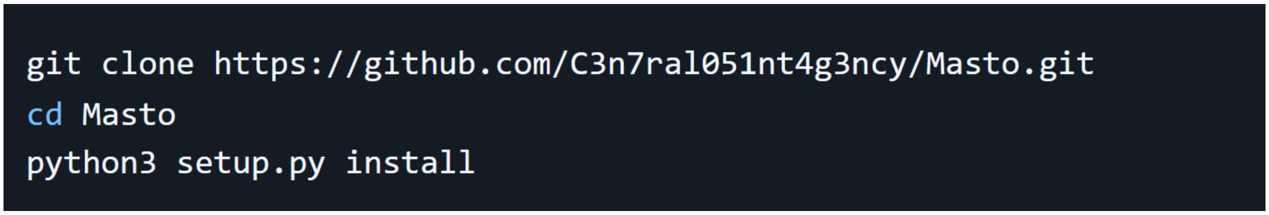
Installation -

1. Using PyPI

https://pypi.org/project/masto/

pip install masto==2.0.5

2. **Using GitHub**



```
git clone https://github.com/C3n7ral051nt4g3ncy/Masto.git
cd Masto
python3 setup.py install
```

🧑‍💻 Usage -

- Help: masto -h
- Search for user masto -user {username}
- Search for instance masto -instance {instance_name}

☆ Tool use cases -

Use case 1	Searching for a user and bypassing the profile directory opt-out

- Tried searching via browser both terms Webbreacher and @Webbreacher **1 result** --> @Webbreacher@mastodon.social
- Searched Webbreacher on **Masto**: **3 results** --> ☑ 3 accounts found
- On the counter.social profile, @Webbreacher's settings are --> user opted to be on the profile directory = False, this is why the browser search didn't find the counter.social profile!

□ **Masto successful outcome**: **Masto found all 3 accounts**.

Use case 2	Searching without getting a 401 error

- Many people don't want an account on Mastodon, and if you don't have an account, you can search on Mastodon, but you will only get 5 results.

- Clicking on load more will give you a 401 error and request for the user to log in.

□ **Masto successful outcome**: **You can use Masto without logging in to Mastodon**, you won't get a 401 error.

Use case 3	Getting information on locked instances:

- Tried searching for the instance 0sint.social, there isn't much information via a browser search because it's locked.

□ **Masto successful outcome**: **Masto found more information on the instance and on the admin, including email address.**

Use case 4	Conducted a username search for Defcon:

- Conducted a search with Masto for the username defcon, the Mastodon API returned 2 user accounts.

□ **Masto successful outcome**: **Masto OSINT Tool picked up after the initial API search by doing a full scan and found 4 accounts.**

🐘 Mastodon.social understanding -

The **same username** can be found across different instances(servers):

- example: @osint@mastodon.social | @osint@mstdn.social | @osint@ counter.social

- Finding the same username on different instances does not prove it's the same person behind each account.

- Each instance can only have **one unique username** in the server. Tip: verify your account with the <a rel="me" attribute which confirms you are behind the account, and will help avoid or detect impersonators.

👩 Testing on known users and instances -

- For a username test, try: python3 masto.py -u Gargron, the founder of [Mastodon.social](), this pulls a wopping 11 accounts!!! (keep in mind that the same username doesn't prove the 11 accounts belong to @Gargron {Gargron is the Mastodon Dev}).
- For an instance test, try: python3 masto.py -i social.network.europa.eu

⭕ Mastodon API reliability issues -

- You may know of a valid user & have the link to the user's profile, you input the username on Masto but get no result.
- I asked the Mastodon Team about this api issue, they replied:

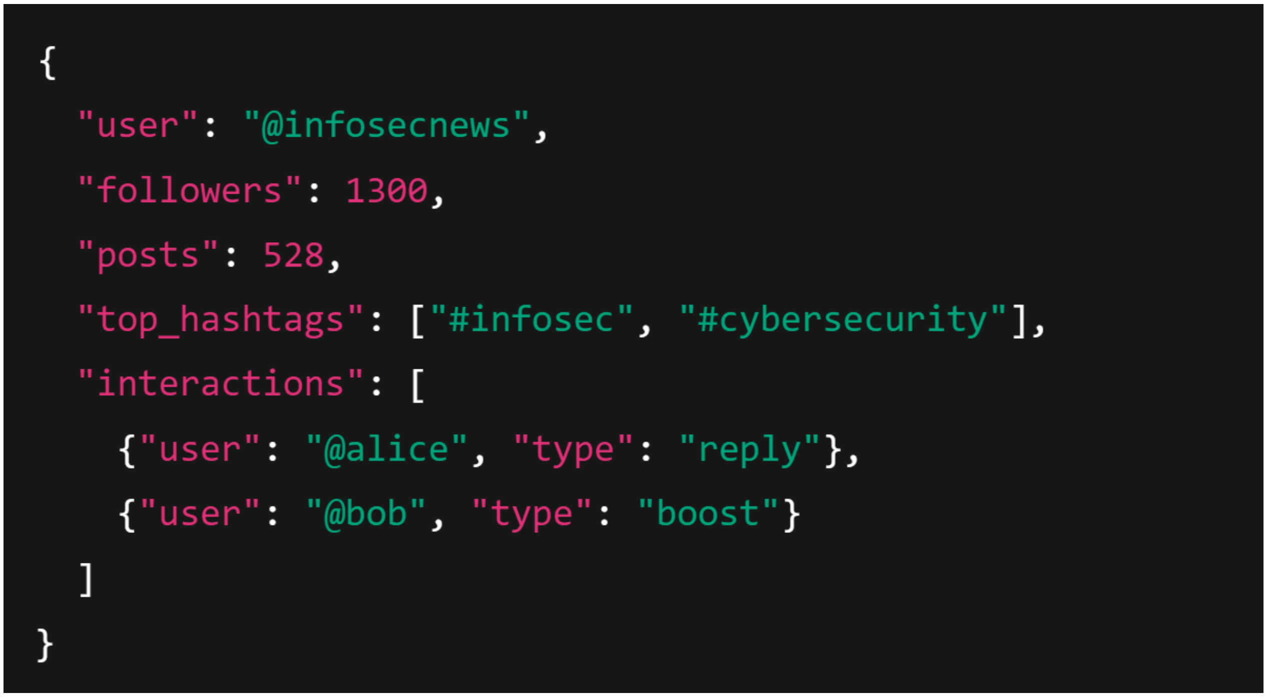
There is no global search, the server will reply with what it knows about. If it has not encountered the account, it will not return it in search results.

- □ **Masto v2.0 fixes this**, the scan of Masto's own json instances list comes in support of Mastodon's API and picks up on things the API missed.
- **v2.0 is 100% reliable** if the server is listed in the Masto fediverse_instances.json file.
- This fix is thanks to @Webbreacher who suggested this feature.

Results Achieved by Some Users who used the tool (Sample Output) -

When analyzing a public account like @infosecnews@mastodon.social, the tool was able to retrieve:
- Over 500 public posts.
- Interaction data with over 300 distinct users.
- Trends in hashtags such as #infosec and #cybersecurity.
- Visual graphs representing the most frequent interaction nodes.

Sample JSON output:



```
{
  "user": "@infosecnews",
  "followers": 1300,
  "posts": 528,
  "top_hashtags": ["#infosec", "#cybersecurity"],
  "interactions": [
    {"user": "@alice", "type": "reply"},
    {"user": "@bob", "type": "boost"}
  ]
}
```

Educational Use Cases, Limitations & Ethical Considerations

Educational Use Cases -

- Cybersecurity training: Demonstrates how threat actors can be tracked through open networks.
- Digital literacy: Teaches students about their digital footprint and how public data can be analyzed.
- Research: Enables sociologists and digital ethnographers to study communication trends in federated networks.

Limitations -

- Limited to publicly accessible content; private posts and DMs cannot be accessed.
- API rate limits and server restrictions may affect data retrieval.
- Not all Mastodon instances share the same architecture, which may cause inconsistencies.

Ethical Considerations -

- Always respect user privacy and only use data that is publicly available.
- Avoid targeting or profiling individuals unfairly.
- Disclose usage of such tools in research and educational projects.
- Stay compliant with platform terms of service and applicable privacy laws.

Submitted by – Deeptansh Nagar

Team – 2 (Keshav Goyal & Deeptansh Nagar)
Mail – deeptansh.121033@stu.upes.ac.in
Contact - +91-9257417905