Searcher

Searcher is a customized search engine developed as part of a school assignment by ***Deborah Ezekiel, Kenneth Burwell,*** and ***Oluwadara Dina***.

Original Project - Searxng

The foundation of Searcher is built upon the open-source project Searxng, which is a privacy-respecting, hackable metasearch engine. We express our gratitude to the contributors of Searxng for providing the groundwork for our project.

Modifications

We have made several modifications and enhancements to adapt the original Searxng to our specific project requirements. These changes include:

Project Renaming:

The original project, SearXNG, has been renamed to "Searcher" for the purpose of this assignment. This includes changes to website index names, user names, and references within the codebase.

Project Theme:

The theme of the project has been altered to align with the specific requirements of our school assignment. This may involve adjustments to the user interface, color schemes, or other design elements to suit the desired theme.

Secret Key Modification:

To enhance security, the original secret keys within the project have been changed. This includes the server's secret key, which is a crucial element for ensuring secure communication and data integrity.

Installation of Missing Dependencies:

During the setup process, it was identified that certain dependencies were missing. These dependencies have been installed to ensure the proper functioning of the Searcher instance.

These requirements included:

- *certifi==2023.7.22*
- *babel==2.13.1*
- *flask-babel==4.0.0*
- *flask==3.0.0*
- *jinja2==3.1.2*
- *lxml==4.9.3*
- *pygments==2.16.1*
- *python-dateutil==2.8.2*
- *pyyaml==6.0.1*
- *httpx[http2]==0.24.1*
- *Brotli==1.1.0*
- *uvloop==0.19.0*
- *httpx-socks[asyncio]==0.7.7*
- *setproctitle==1.3.3*
- *redis==4.6.0*
- *markdown-it-py==3.0.0*
- *typing_extensions==4.8.0*
- *fasttext-predict==0.9.2.1*
- *pytomlpp==1.0.13*

Logo Design and Favicon Changes:

The project's logo design has been modified to better reflect the intended theme or branding for the assignment. Additionally, the favicon, the small icon displayed in a browser tab, has been updated to align with the new logo design.

Server Port Modification:

The default server port, where the Searcher instance is hosted, has been changed. This adjustment may be made for various reasons, such as avoiding conflicts with other services or adhering to specific project requirements.

Search method:

The search method has been changed to "GET" for improved functionality.

Autocomplete:

Autocomplete has been set to use DuckDuckGo for a more enhanced search experience.

Footer Editing:

The footer of the Searcher instance has been edited to include relevant information or branding specific to the school assignment. This includes names of authors and any other information deemed important.

Addition of Cursor CSS:

A custom cursor style, defined by CSS (Cascading Style Sheets), has been added to enhance the visual experience of the Searcher instance. This could be a unique cursor design that aligns with the project theme.

Importance of Searcher

Using Searcher offers several security benefits, making it an attractive choice, especially when prioritizing privacy and control over your search queries. Here are key points highlighting the importance of using Searcher from a security perspective:

Decentralized Control:

- Hosting Searcher locally or on a private server gives users decentralized control. This means that individuals or organizations have the power to manage and secure their search infrastructure without relying on external entities.

Enhanced Privacy Measures:

- Searcher is designed with a focus on privacy. It minimizes the collection of user data, reducing the risk of exposure to third-party entities. This commitment to privacy is crucial in an era where online activities are increasingly scrutinized.

Open-Source Transparency:

- Searcher being open-source allows users to inspect the source code, ensuring transparency in its operations. Security-conscious users can audit the code, identify vulnerabilities, and contribute to the improvement of the project's security.

Reduced Tracking and Profiling:

- By avoiding the use of external trackers and analytics, Searcher helps reduce the risk of user tracking and profiling. Users can conduct searches without being subject to the extensive tracking mechanisms employed by some commercial search engines.

User-Defined Filters:

- Searcher's customizable nature extends to user-defined filters, enabling users to implement additional security measures. This could include content filtering, blacklisting specific domains, or incorporating custom security modules as needed.

No Third-Party Dependencies:

- Searcher minimizes reliance on external services or APIs for search functionality. This reduces the attack surface and potential vulnerabilities associated with third-party integrations, contributing to a more secure overall environment.

Community-Driven Security Updates:

- The open-source nature of Searcher encourages a community of contributors who can actively address security concerns. This collaborative approach to development ensures that security updates and patches are promptly implemented.

Encrypted Search Connections:

- Searcher supports encrypted search connections (HTTPS), providing a secure communication channel between the user and the search engine. This prevents potential eavesdropping and man-in-the-middle attacks during the search process.

No Profiling for Targeted Ads:

- Unlike some commercial search engines that use search data for targeted advertising, Searcher does not engage in such practices. This protects users from personalized advertising based on their search history.

User Education and Empowerment:

- Searcher places emphasis on user education, empowering individuals to make informed decisions about their online security and privacy. The platform encourages users to take an active role in understanding and managing their digital footprint.

In summary, Searcher empowers users with a secure and customizable search experience, offering the ability to host locally, customize features, obfuscate IP addresses, search images without external redirects, and maintain control over search queries. These security features collectively contribute to a more private and secure online search experience, aligning with the growing emphasis on user privacy in the digital age.

Contributors

- Deborah Ezekiel - [deeze2@morgan.edu]
- Kenneth Burwell - [kebur17@morgan.edu]
- Oluwadara Dina - [oldin1@morgan.edu]

Usage

How to Use Searcher:

Clone the Repository:

- Start by cloning the Searcher repository to your local machine.
- Use the following command in your terminal or command prompt:

```
$git clone [repository_url]\"/usr/local/searxng/searxng-src"
```

Install Packages:

```
$ sudo -H apt-get install -y \
    python3-dev python3-babel python3-venv \
    uwsgi uwsgi-plugin-python3 \
    git build-essential libxslt-dev zlib1g-dev libffi-dev libssl-dev
```

Create User:

```
$ sudo -H useradd --shell /bin/bash --system \
    --home-dir "/usr/local/searxng" \
    --comment 'Privacy-respecting metasearch engine' \
    searxng

$ sudo -H mkdir "/usr/local/searxng"
$ sudo -H chown -R "searxng:searxng" "/usr/local/searxng"
```

Install Dependencies:

```
$ sudo -H -u searxng –i
(searxng)$ python3 -m venv "/usr/local/searxng/searx-pyenv"
(searxng)$ echo ". /usr/local/searxng/searx-pyenv/bin/activate" \
                >> "/usr/local/searxng/.profile"
```

Exit bash and open a new one.

```
$ sudo -H -u searxng -i

(searxng)$ command -v python && python --version
/usr/local/searxng/searx-pyenv/bin/python
Python 3.8.1

# update pip's boilerplate ..
pip install -U pip
pip install -U setuptools
pip install -U wheel
pip install -U pyyaml
```


```
# jump to SearXNG's working tree and install SearXNG into virtualenv
(searxng)$ cd "/usr/local/searxng/searxng-src"
(searxng)$ pip install -e .
```

Set secret key:

```
$ sudo -H mkdir -p "/etc/searxng"
$ sudo -H cp "/usr/local/searxng/searxng-src/utils/templates/etc/searxng/settings.yml" \
             "/etc/searxng/settings.yml"
```

Edit settings:

```
# Searcher settings

use_default_settings: true

general:
  debug: false
  instance_name: "SearXNG"

search:
  safe_search: 2
  autocomplete: 'duckduckgo'

server:
  # Is overwritten by ${SEARXNG_SECRET}
  secret_key: "setsecretkey"
  limiter: true
  image_proxy: true
  # public URL of the instance, to ensure correct inbound links. Is overwritten
  # by ${SEARXNG_URL}.
  # base_url: http://example.com/location

redis:
  # URL to connect redis database. Is overwritten by ${SEARXNG_REDIS_URL}.
  url: unix:///usr/local/searxng-redis/run/redis.sock?db=0

ui:
  static_use_hash: true

# preferences:
#   lock:
#     - autocomplete
#     - method
```

```
enabled_plugins:
  - 'Hash plugin'
  - 'Self Informations'
  - 'Tracker URL remover'
  - 'Ahmia blacklist'
  # - 'Hostname replace'  # see hostname_replace configuration below
  # - 'Open Access DOI rewrite'

# plugins:
#   - only_show_green_results
```

Test run:

```
# enable debug ..
$ sudo -H sed -i -e "s/debug : False/debug : True/g"
"/etc/searxng/settings.yml"

# start webapp
$ sudo -H -u searxng -i
(searxng)$ cd /usr/local/searxng/searxng-src
(searxng)$ export SEARXNG_SETTINGS_PATH="/etc/searxng/settings.yml"

 Edit secret key again under: "/etc/searxng/settings.yml"

(searxng)$ python searx/webapp.py

# disable debug
$ sudo -H sed -i -e "s/debug : True/debug : False/g" "/etc/searxng/settings.yml"
```

Navigate to Project Directory:

- Change your current working directory to the Searcher project directory:

  cd searcher

Activate Virtual Environment:

- If you haven't activated the virtual environment, do so using the following command:

  source /path/to/your/searcher-pyenv/bin/activate

- Replace "/path/to/your/searcher-pyenv/" with the actual path to your virtual environment.

Start the Searcher Webapp:

- Once the virtual environment is activated, start the Searcher web application.

  Use the following commands:

  cd /path/to/your/searcher-src
  python searcher/webapp.py

- Access the web app by navigating to *http://127.0.0.1:8989* in your web browser.

Explore Searcher:

- The Searcher web interface should now be accessible. You can enter search queries in the search bar and hit enter to get results.

Customize Settings (Optional):

- If you want to customize Searcher further, explore the configuration files in "/etc/searcher/" and modify the settings.yml file.

Shutdown the Webapp:

- When you're done using Searcher, you can shut down the webapp. Return to the terminal where it's running and press Ctrl + C to stop the process.

Deactivate Virtual Environment (Optional):

- If you activated a virtual environment, you can deactivate it using the following command:

  deactivate

That's it! You've successfully set up and used the Searcher project. Feel free to explore the different features and configurations based on your preferences and project requirements. If you encounter any issues or have specific tasks in mind, refer to the documentation or reach out for further assistance. Happy searching! 🚀

Issues and Feedback

If you encounter any issues or have feedback, please feel free to reach out to us via email.

License