Own Private Search Engine in Linux Will Save Our Privacy

12 May 2020 Kali Linux Home Configure Custom Tools For Kali Linux ○ size ○

Surfing the internet has become a passion for everyone but most of the search engines keep the log of our search behavior and track us. They can share our searching data with the government. Recently Forbes warned us that some company sends our data to China in **this article**.

They make a huge database using our data and make category of consumers. We have noticed that if we search for “Adidas football” on Google, then after sometime we got Adidas football's ads on the internet. They use our search behavior to show us ads. This might be not against Google's terms and conditions but against our privacy.

Not only showing ads, they keep an extra eye on us. They always monitor our search results and using them.

Some privacy search engines (such as **Duckduckgo**, **Privado**) claim that **they did not store any log**, but we don’t know what is running on their back end.

It will be better that we create our own search engine and use it. This will be the best option to keep our privacy.

In this detailed tutorial we create our own search engine and surf the internet in a very private way in our Kali Linux or any Debian based Linux distributions like Ubuntu Parrot and other.

To do this we use the **Searx** tool published on GitHub. This is an python2 based open-source privacy search engine. Open-Source means we can read the codes in it and we know what Searx is doing in the background, it is open for all.

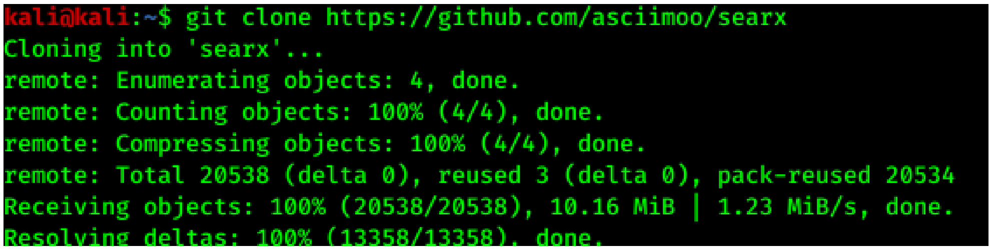
We can host this search engine on our localhost and use it to surf the internet. We need to have Python2 installed in our machine. Python comes pre-installed in Kali Linux.

First we open the terminal window and type the following command and hit enter.

```
git clone https://github.com/asciimoo/searx
```

This command will clone the Searx tool in our machine.

Now we need to go inside the searx directory by using cd command:



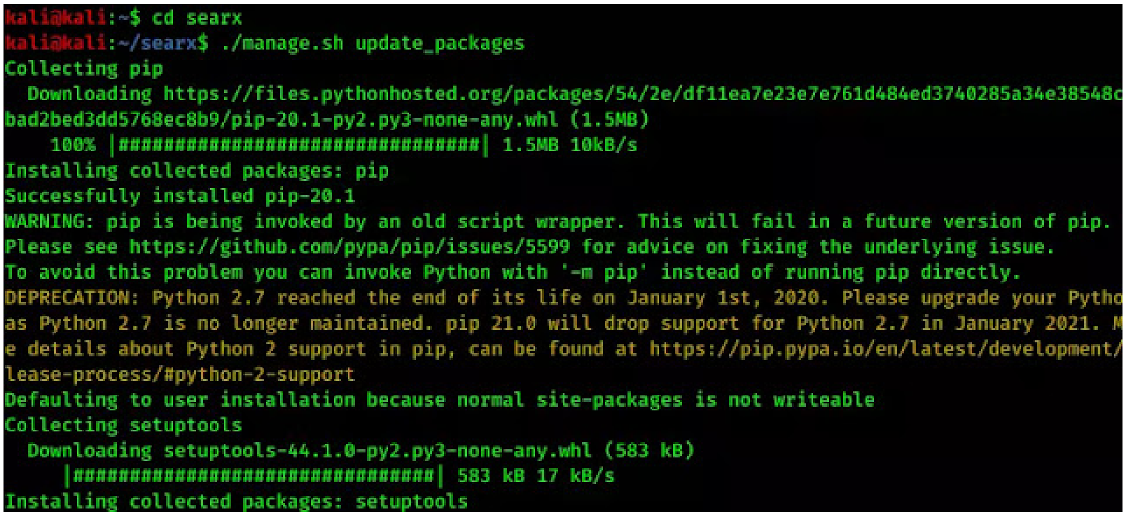
```
kali@kali:~$ git clone https://github.com/asciimoo/searx
Cloning into 'searx'...
remote: Enumerating objects: 4, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 20538 (delta 0), reused 3 (delta 0), pack-reused 20534
Receiving objects: 100% (20538/20538), 10.16 MiB | 1.23 MiB/s, done.
Resolving deltas: 100% (13358/13358), done.
```

```
cd searx
```

Here we need to install dependencies to run searx. To do this we apply following command:

```
./manage.sh update_packages
```

Then it will install the dependencies or update them. This may take some time depending on the internet speed and machine configuration. The screenshot is following:

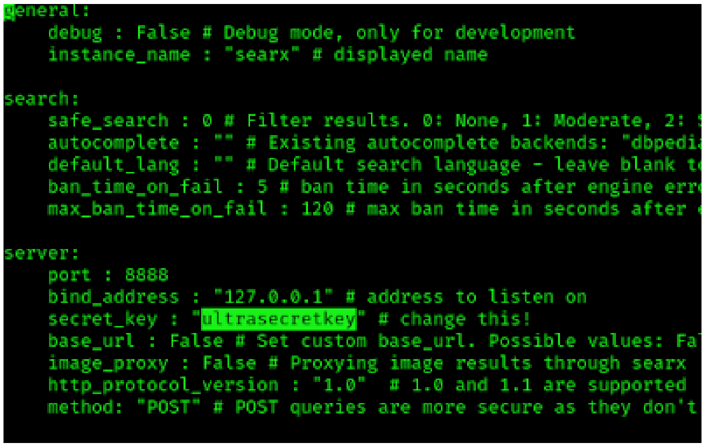


```
kali@kali:~$ cd searx
kali@kali:~/searx$ ./manage.sh update_packages
Collecting pip
  Downloading https://files.pythonhosted.org/packages/54/2e/df11ea7e23e7e761d484ed3740285a34e38548c
bad2bed3dd5768ec8b9/pip-20.1-py2.py3-none-any.whl (1.5MB)
    100% |################################| 1.5MB 10kB/s
Installing collected packages: pip
Successfully installed pip-20.1
WARNING: pip is being invoked by an old script wrapper. This will fail in a future version of pip.
Please see https://github.com/pypa/pip/issues/5599 for advice on fixing the underlying issue.
To avoid this problem you can invoke Python with '-m pip' instead of running pip directly.
DEPRECATION: Python 2.7 reached the end of its life on January 1st, 2020. Please upgrade your Pytho
as Python 2.7 is no longer maintained. pip 21.0 will drop support for Python 2.7 in January 2021. M
e details about Python 2 support in pip, can be found at https://pip.pypa.io/en/latest/development/
lease-process/#python-2-support
Defaulting to user installation because normal site-packages is not writeable
Collecting setuptools
  Downloading setuptools-44.1.0-py2.py3-none-any.whl (583 kB)
     |################################| 583 kB 17 kB/s
Installing collected packages: setuptools
```

Here we need to change our secret key inside /searx/settings.yml. To do this we use following command:

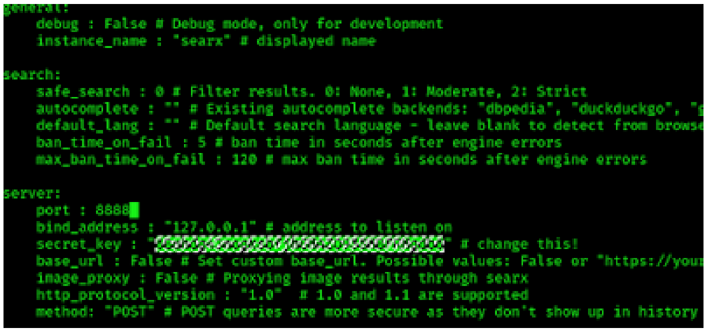
```
nano searx/settings.yml
```

The screenshot is following.



```
general:
    debug : False # Debug mode, only for development
    instance_name : "searx" # displayed name

search:
    safe_search : 0 # Filter results. 0: None, 1: Moderate, 2: S
    autocomplete : "" # Existing autocomplete backends: "dbpedi
    default_lang : "" # Default search language - leave blank t
    ban_time_on_fail : 5 # ban time in seconds after engine err
    max_ban_time_on_fail : 120 # max ban time in seconds after

server:
    port : 8888
    bind_address : "127.0.0.1" # address to listen on
    secret_key : "ultrasecretkey" # change this!
    base_url : False # Set custom base_url. Possible values: Fa
    image_proxy : False # Proxying image results through searx
    http_protocol_version : "1.0"  # 1.0 and 1.1 are supported
    method: "POST" # POST queries are more secure as they don't
```

Here we need to change this key to whatever we want(any thing like 2gvjyGDH68gtplaHuvrty45), inside the quotation mark. What we did shown in the following screenshot:



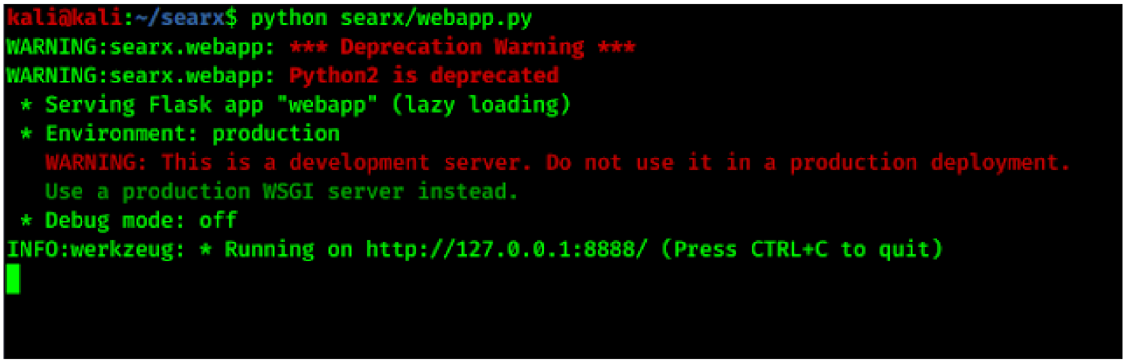
```
general:
    debug : False # Debug mode, only for development
    instance_name : "searx" # displayed name

search:
    safe_search : 0 # Filter results. 0: None, 1: Moderate, 2: Strict
    autocomplete : "" # Existing autocomplete backends: "dbpedia", "duckduckgo", "g
    default_lang : "" # Default search language - leave blank to detect from browse
    ban_time_on_fail : 5 # ban time in seconds after engine errors
    max_ban_time_on_fail : 120 # max ban time in seconds after engine errors

server:
    port : 8888
    bind_address : "127.0.0.1" # address to listen on
    secret_key : "[redacted]" # change this!
    base_url : False # Set custom base_url. Possible values: False or "https://your
    image_proxy : False # Proxying image results through searx
    http_protocol_version : "1.0"  # 1.0 and 1.1 are supported
    method: "POST" # POST queries are more secure as they don't show up in history
```

We should not publish or share it anywhere. After assign the secret_key We just save and close it by pressing CTRL+X and then pressing Y, then ENTER.

Now we are ready to rock. We can run this on our localhost server by applying following command:

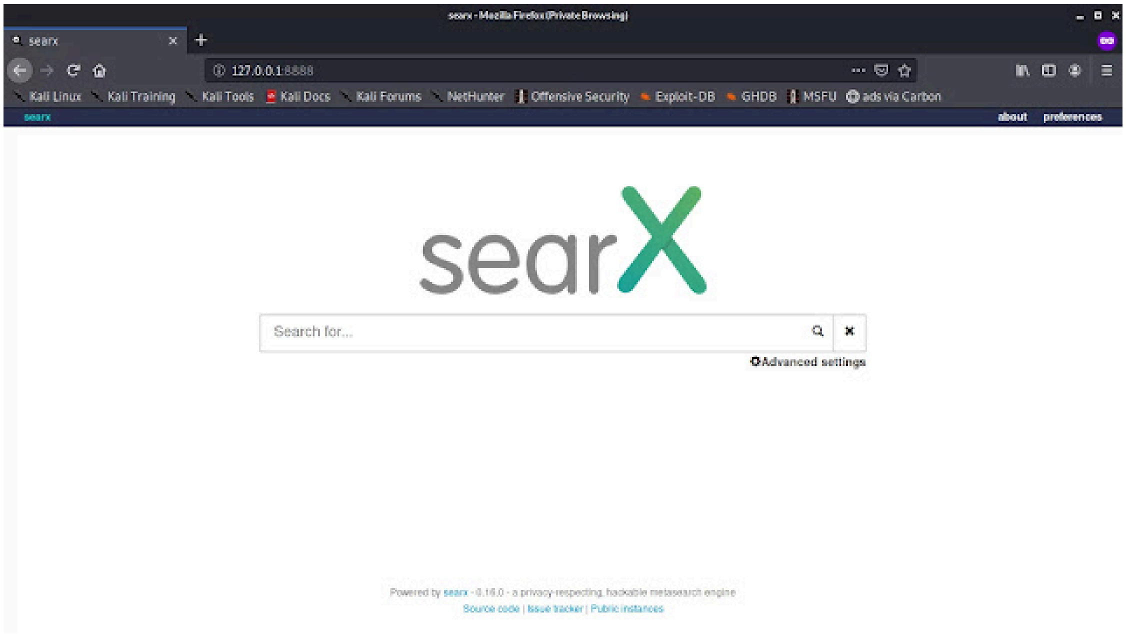
```
python3 searx/webapp.py
```

The output of the preceding command shows in the following screenshot:



```
kali@kali:~/searx$ python searx/webapp.py
WARNING:searx.webapp: *** Deprecation Warning ***
WARNING:searx.webapp: Python2 is deprecated
 * Serving Flask app "webapp" (lazy loading)
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployment.
   Use a production WSGI server instead.
 * Debug mode: off
INFO:werkzeug: * Running on http://127.0.0.1:8888/ (Press CTRL+C to quit)
```

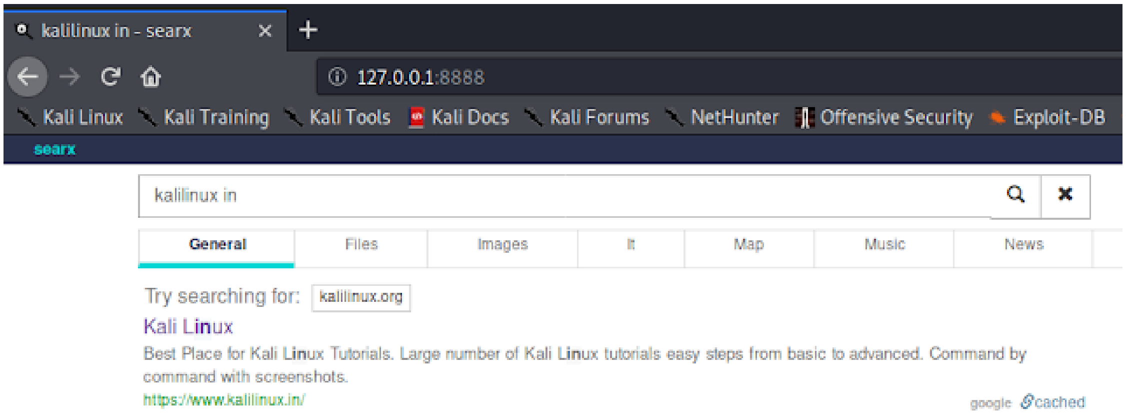
We can see that Searx is running on our localhost. Now we can open this link on our browser and our own private search engine is ready.



Powered by searx - 0.16.0 - a privacy-respecting, hackable metasearch engine
Source code | Issue tracker | Public instances

own private search engine hosted on localhost

Here we search for our website and we got search results as shown in the following screenshot:



Try searching for: kalilinux.org

Kali Linux

Best Place for Kali Linux Tutorials. Large number of Kali Linux tutorials easy steps from basic to advanced. Command by command with screenshots.

https://www.kalilinux.in/ google cached

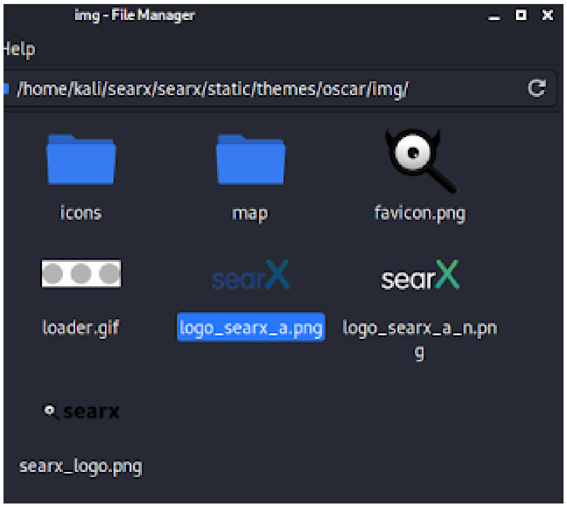
Search result of private search engine

The search results are come from various popular search engines like Google, Bing, Duckduckgo, Yahoo and more. But this search engine didn't compromise our privacy.

Some Customization

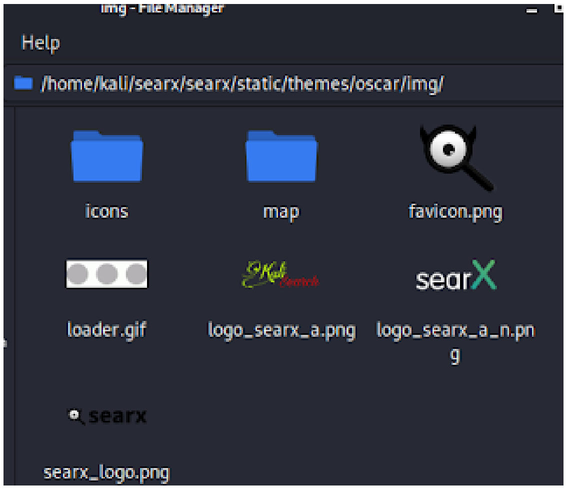
How it becomes own? It uses Searx branding in search engine homepage. Well it is an open-source search engine that means we can customize it as our own. For an example we change the logo of Searx in the homepage.

From file manager we need to go to this directory */home/kali/searx/searx/static/themes/oscar/img/* , here we got the icons of Searx search engine. The screenshot is following:



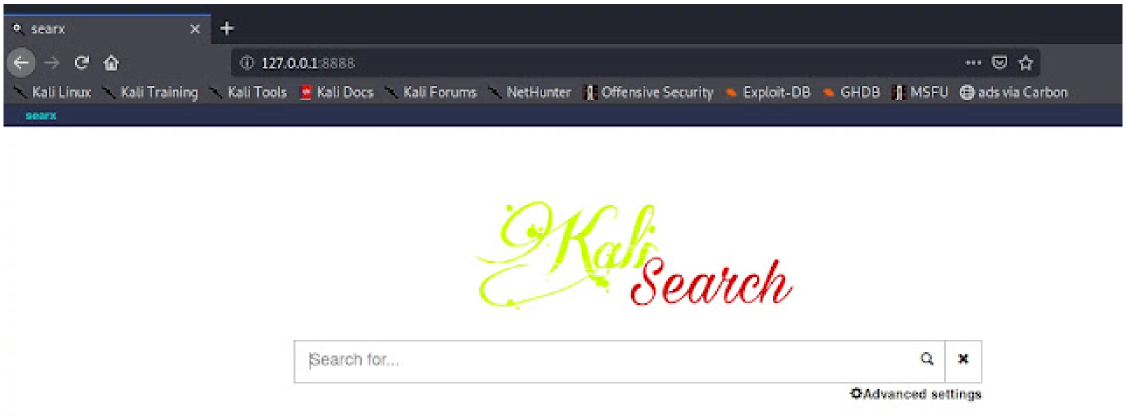
We need to change the highlighted one

We change the highlighted png image file with our custom png logo file. To do this we need a little image editing knowledge(same size will provide a perfect result). Keep in mind that the name of our custom image file should be the same(logo_searx_a.png).



custom logo for Searx search engine

Then we can see that our logo in the search engine has been changed.



personalized Private Search Engine

For More Privacy

Searx is hosted on our localhost but it collects search results from the internet (other search engines). When Searx does this other search engines could get our IP address. In simple words the IP address that Searx passes to other search engines is the same as the outgoing IP address of our device.

To prevent this we need to **link our searx search engine with a proxy server**. Then the requests made to other search engines are made via the proxy server, rather than wherever the Searx instance is running. We also can use **Tor** or **VPN** to do the same thing. **This is how we can create our own private search engine easily that maintains our privacy.**

If you liked our tutorial then share this with privacy concern friends. For more tutorials visit our site regularly and for a quick update follow us on **Twitter** and **Medium**.

For any kind of questions or problems feel free to comment down below. We always reply.