SpiderFoot

Spiderfoot is a free and open-source tool available on Github. This tool is a framework written in the python programming language. You must have python installed in your Kali Linux operating system to use this framework. Spiderfoot is used for reconnaissance. Spiderfoot uses different modules for information gathering. Spiderfoot is capable enough to gather information about the target host through active and passive scanning options available on the Spiderfoot framework. In the Spiderfoot framework different scanning options and modules available to set and scan the target host. Spiderfoot is an Open Source Intelligence and Information Gathering Tool. Spiderfoot is capable of doing everything almost you need for reconnaissance as per your need. Spiderfoot works as an open-source tool intelligence tool. It integrates with just about every data source available and utilizes a range of methods for data analysis, making that data easy to navigate. Spiderfoot has an embedded web server for providing an intuitive web-based interface, but you can also do the same using a command-line interface.

Features of Spiderfoot:

- Spiderfoot is a free and open-source tool available on Github.
- Spiderfoot works as a framework cum tool.
- Spiderfoot framework is written in python language.
- Spiderfoot can be used for reconnaissance.

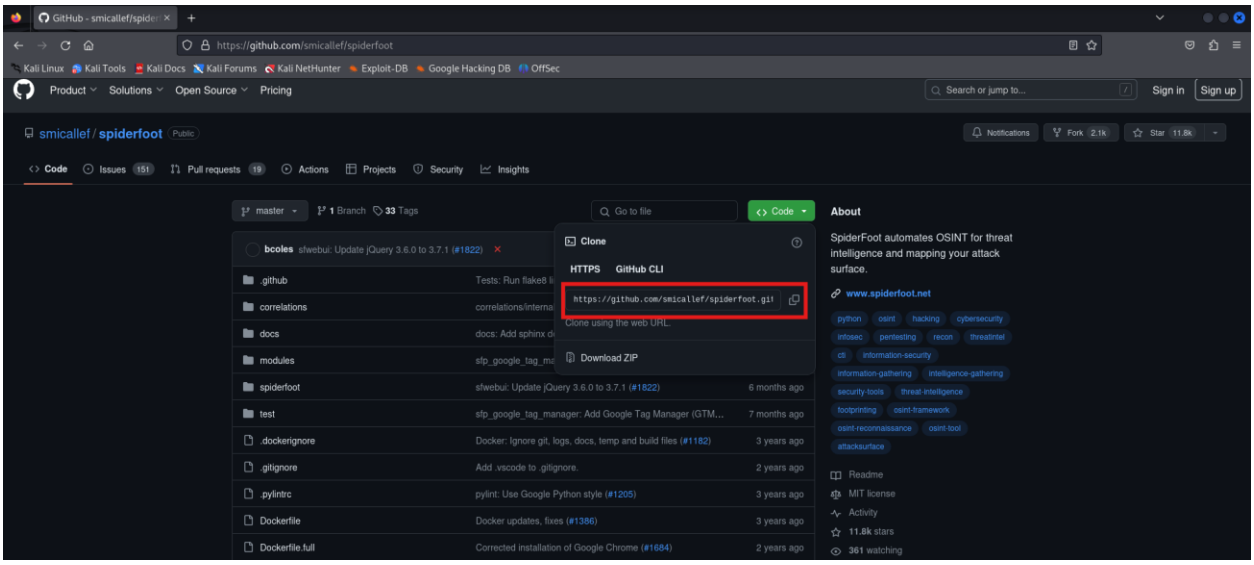
- Spiderfoot contains many modules. As it's a framework that uses modules for information gathering.
- Spiderfoot works on the principles of OSINT.
- Spiderfoot is an automated OSINT Framework.
- Spiderfoot automates the reconnaissance processes.

Uses of Spiderfoot:

- Spiderfoot is used for reconnaissance.
- Spiderfoot is used for information gathering.
- Spiderfoot is working as a scanner for active and passive scanning on target.
- Spiderfoot can be used for domain footprinting.
- Spiderfoot can be used to find the phone numbers, email addresses of the target.
- Spiderfoot can be used to find bitcoin addresses.
- Spiderfoot can be used to save all the information gathering summary.
- Spiderfoot can be used to create graphs of scanning done by Spiderfoot.
- Spiderfoot can be used to automateGitHub all the information gathering processes.

Steps to Download the SpiderFoot:

1.First of all, copy the link of tool from its github page.



2.Now use git clone command in terminal to clone it.

```
┌──(root㉿kali)-[/home/kali]
└─# git clone https://github.com/smicallef/spiderfoot.git
Cloning into 'spiderfoot' ...
remote: Enumerating objects: 26202, done.
remote: Counting objects: 100% (3594/3594), done.
remote: Compressing objects: 100% (272/272), done.
remote: Total 26202 (delta 3427), reused 3353 (delta 3320), pack-reused 22608
Receiving objects: 100% (26202/26202), 16.05 MiB | 8.15 MiB/s, done.
Resolving deltas: 100% (21220/21220), done.
```

3.Now move to the directory of the tool.

```
┌──(root㉿kali)-[/home/kali]
└─# cd spiderfoot

┌──(root㉿kali)-[/home/kali/spiderfoot]
└─# ls
correlations          docker-compose.yml  docs                modules           setup.cfg   sf.py       spiderfoot  VERSION
docker-compose-dev.yml  Dockerfile        generate-certificate  README.md       sfcli.py    sfscan.py   test
docker-compose-full.yml  Dockerfile.full  LICENSE             requirements.txt  sflib.py    sfwebui.py  THANKYOU
```

4.Now we have to install the requirements.txt file by using below shown command.

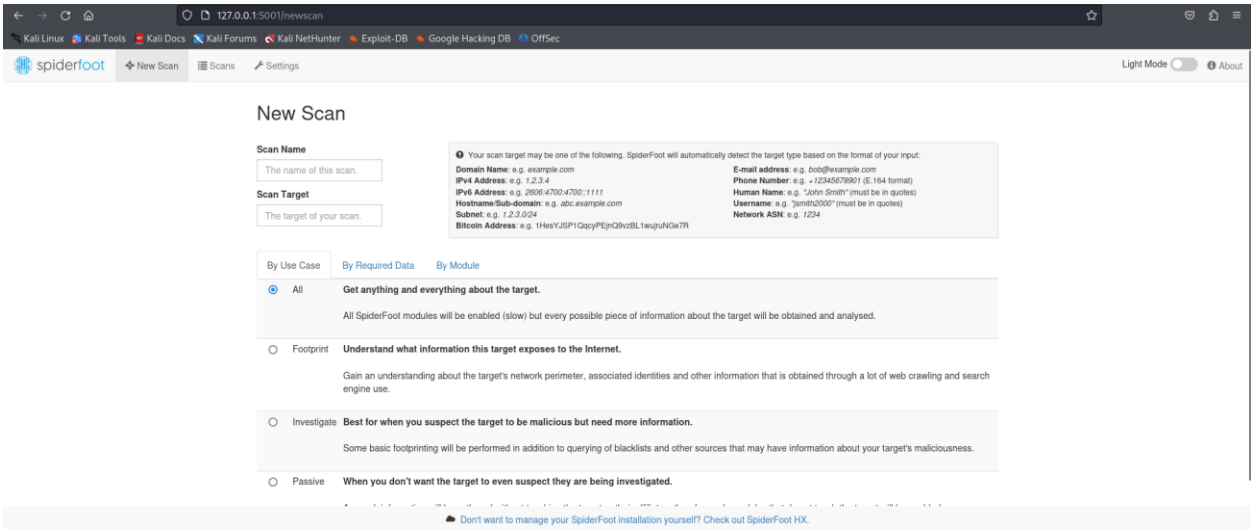
```
┌──(root㉿kali)-[/home/kali/spiderfoot]
└─# pip install -r requirements.txt
Requirement already satisfied: adblockparser<1,≥0.7 in /usr/lib/python3/dist-packages (from -r requirements.txt (line 1)) (0.7)
Requirement already satisfied: dnspython<3,≥2.3.0 in /usr/lib/python3/dist-packages (from -r requirements.txt (line 2)) (2.5.0)
Requirement already satisfied: ExifRead<3,≥2.3.2 in /usr/local/lib/python3.11/dist-packages (from -r requirements.txt (line 3)) (2.3.2)
Requirement already satisfied: CherryPy<19,≥18.8.0 in /usr/lib/python3/dist-packages (from -r requirements.txt (line 4)) (18.9.0)
Requirement already satisfied: cherrypy-cors<2,≥1.6 in /usr/lib/python3/dist-packages (from -r requirements.txt (line 5)) (1.6)
Requirement already satisfied: Mako<2,≥1.2.4 in /usr/lib/python3/dist-packages (from -r requirements.txt (line 6)) (1.3.2.dev0)
Requirement already satisfied: beautifulsoup4<5,≥4.11.2 in /usr/lib/python3/dist-packages (from -r requirements.txt (line 7)) (4.12.3)
Collecting lxml<5,≥4.9.2 (from -r requirements.txt (line 8))
  Using cached lxml-4.9.4-cp311-cp311-manylinux_2_28_x86_64.whl.metadata (3.7 kB)
Requirement already satisfied: netaddr<1,≥0.8.0 in /usr/lib/python3/dist-packages (from -r requirements.txt (line 9)) (0.8.0)
Requirement already satisfied: pysocks<2,≥1.7.1 in /usr/lib/python3/dist-packages (from -r requirements.txt (line 10)) (1.7.1)
Requirement already satisfied: requests<3,≥2.28.2 in /usr/lib/python3/dist-packages (from -r requirements.txt (line 11)) (2.31.0)
Collecting ipwhois<1.2.0,≥1.1.0 (from -r requirements.txt (line 12))
  Using cached ipwhois-1.1.0-py2.py3-none-any.whl.metadata (19 kB)
Requirement already satisfied: ipaddr<3,≥2.2.0 in /usr/local/lib/python3.11/dist-packages (from -r requirements.txt (line 13)) (2.2.0)
Requirement already satisfied: phonenumbers<9,≥8.13.6 in /usr/local/lib/python3.11/dist-packages (from -r requirements.txt (line 14)) (8.13.36)
Requirement already satisfied: pygexf<0.3,≥0.2.2 in /usr/lib/python3/dist-packages (from -r requirements.txt (line 15)) (0.2.2)
Collecting PyPDF2<2,≥1.28.6 (from -r requirements.txt (line 16))
  Using cached PyPDF2-1.28.6-py3-none-any.whl.metadata (5.3 kB)
Collecting python-whois<0.8,≥0.7.3 (from -r requirements.txt (line 17))
  Using cached python_whois-0.7.3-py3-none-any.whl
Requirement already satisfied: secure<0.4.0,≥0.3.0 in /usr/lib/python3/dist-packages (from -r requirements.txt (line 18)) (0.3.0)
Collecting pyOpenSSL<22,≥21.0.0 (from -r requirements.txt (line 19))
  Using cached pyOpenSSL-21.0.0-py2.py3-none-any.whl.metadata (7.4 kB)
Collecting python-docx<0.9,≥0.8.11 (from -r requirements.txt (line 20))
  Using cached python_docx-0.8.11-py3-none-any.whl
Collecting python-pptx<0.7,≥0.6.21 (from -r requirements.txt (line 21))
  Using cached python_pptx-0.6.23-py3-none-any.whl.metadata (18 kB)
```

5.Now after the installation of requirements.txt you can run the tool using the given command.

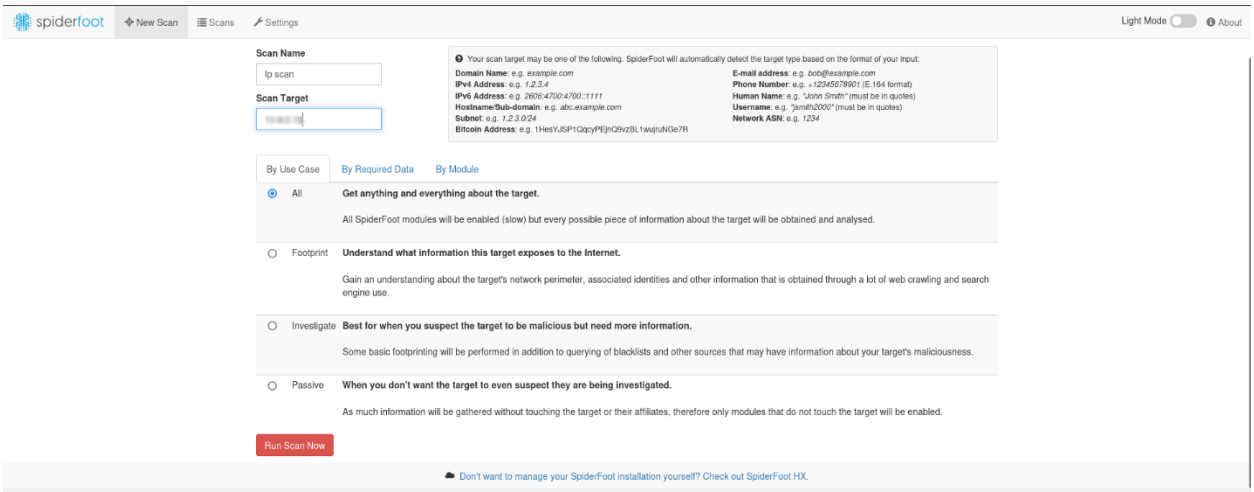
6.Now in order to run spiderfoot's web GUI. We have to run it over local host server. You can do the same using the following command.

```
┌──(root㉿kali)-[/home/kali/spiderfoot]
└─# python3 ./sf.py
SpiderFoot requires -l <ip>:<port> to start the web server. Try --help for guidance.

┌──(root㉿kali)-[/home/kali/spiderfoot]
└─# python3 ./sf.py -l 127.0.0.1:5001

2024-05-08 18:29:03,849 [INFO] sf : Starting web server at 127.0.0.1:5001 ...
2024-05-08 18:29:03,849 [WARNING] sf :
*********************************************************************
Warning: passwd file contains no passwords. Authentication disabled.
Please consider adding authentication to protect this instance!
Refer to https://www.spiderfoot.net/documentation/#security.
*********************************************************************

***************************************************************
 Use SpiderFoot by starting your web browser of choice and
 browse to http://127.0.0.1:5001/
***************************************************************
```

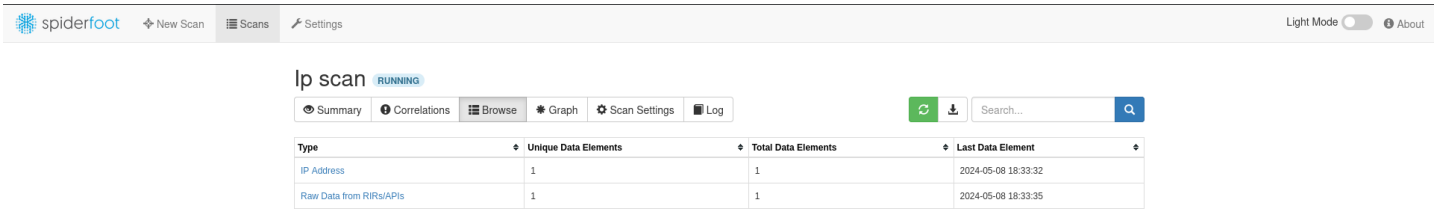
7.Now search the local host server into your browser to run it as a web interface.



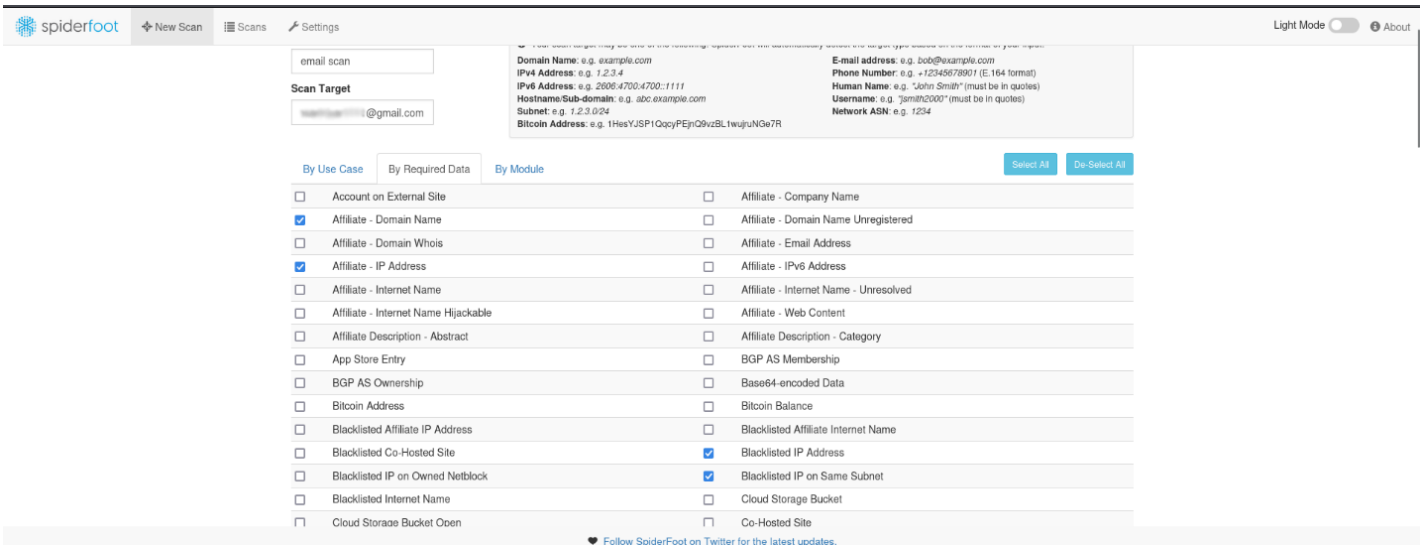
8.Now we will run an IP Scan. We have to give the ip address of the target and hit **Run scan Now** button.



9.This is the result of our scan.



10.Now we will do an email scan. To do so give email to the target field. We can also specify the required information that we are desired to have related our target.



11.This is the result of our email target as per our desired fields.