

# 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.

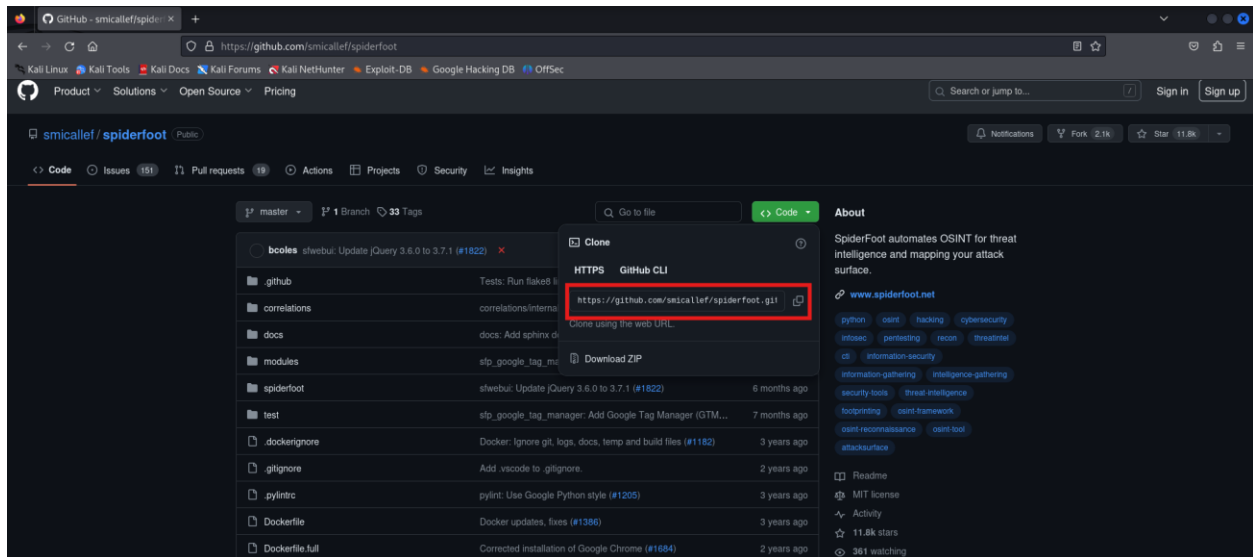
- Spiderfoot contains many modules. As it's a framework that uses modules for information gathering.
- Spiderfoot works on the principles of OSINT.
- Spiderfoot is an automated OSINT Framework.
- Spiderfoot automates the reconnaissance processes.

### **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 automate 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: BeautifulSoup<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: pygeof<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.

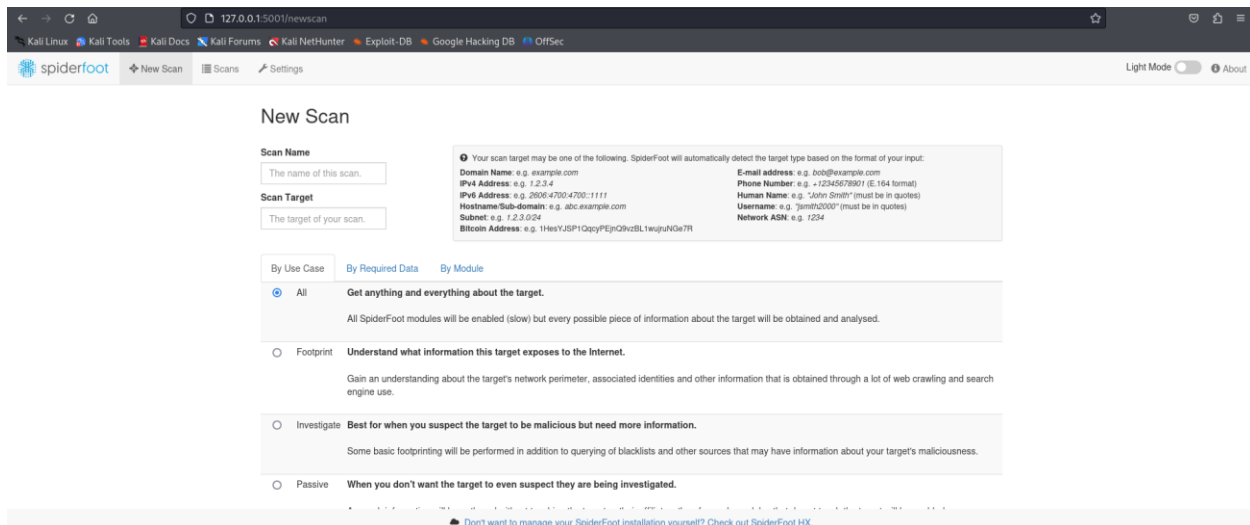
```
(root@kali)~[/home/kali/spiderfoot]
# python3 ./sf.py
SpiderFoot requires -l <ip>:<port> to start the web server. Try --help for guidance.

(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.

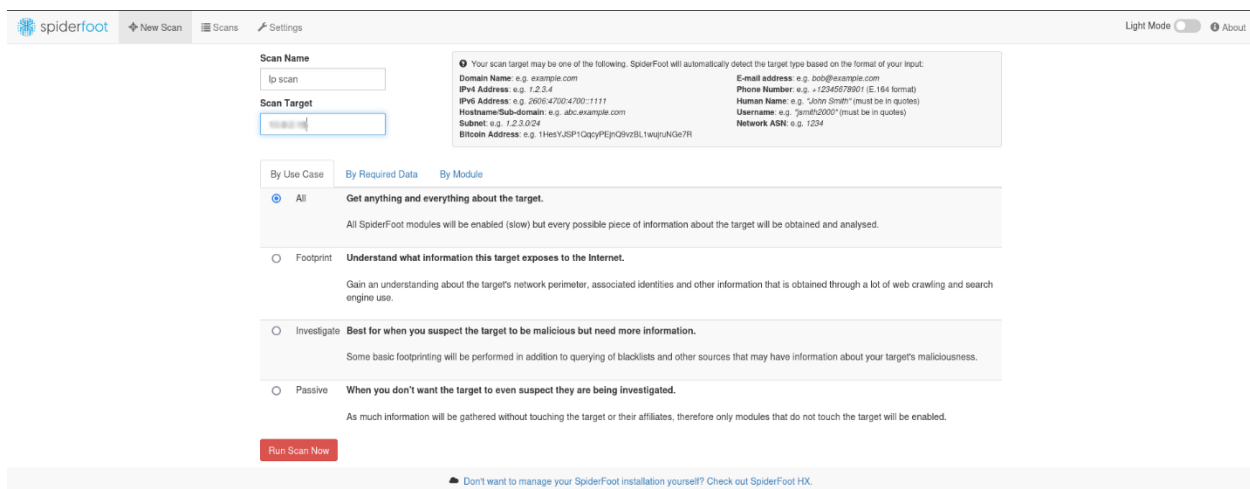


The screenshot shows the SpiderFoot web interface in a browser window. The address bar shows the URL `127.0.0.1:5001/newscan`. The interface has a navigation bar with links for 'New Scan', 'Scans', and 'Settings'. The 'New Scan' form is displayed with the following fields and options:

- Scan Name:** A text input field with the placeholder 'The name of this scan.'
- Scan Target:** A text input field with the placeholder 'The target of your scan.'
- Target Type Examples:** A box listing various target types: Domain Name (e.g., example.com), IPv4 Address (e.g., 1.2.3.4), IPv6 Address (e.g., 2001:4700:4700::1111), Hostname/Sub-domain (e.g., abc.example.com), Subnet (e.g., 1.2.3.0/24), Bitcoin Address (e.g., 1HesYJSP1QqyPEjyQhZBL1wjuH9k7R), E-mail address (e.g., bob@example.com), Phone Number (e.g., +12345678901 (E.164 format)), Human Name (e.g., 'John Smith' (must be in quotes)), Username (e.g., 'jsmith2000' (must be in quotes)), and Network ASN (e.g., 1234).
- By Use Case:** Three radio button options:
  - All:** Selected. Description: 'Get anything and everything about the target. All SpiderFoot modules will be enabled (slow) but every possible piece of information about the target will be obtained and analysed.'
  - Footprint:** Description: 'Understand what information this target exposes to the Internet. Gain an understanding about the target's network perimeter, associated identities and other information that is obtained through a lot of web crawling and search engine use.'
  - Investigate:** Description: 'Best for when you suspect the target to be malicious but need more information. Some basic footprinting will be performed in addition to querying of blacklists and other sources that may have information about your target's maliciousness.'
  - Passive:** Description: 'When you don't want the target to even suspect they are being investigated. As much information will be gathered without touching the target or their affiliates, therefore only modules that do not touch the target will be enabled.'

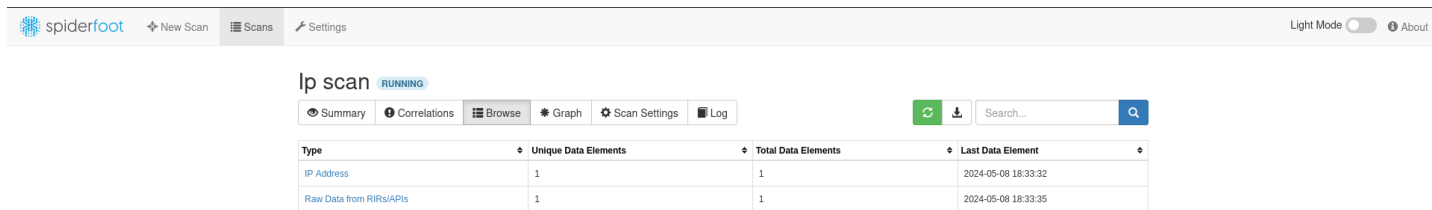
At the bottom of the form, there is a red button labeled 'Run Scan Now' and a link: 'Don't want to manage your SpiderFoot installation yourself? Check out SpiderFoot HX.'

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



This screenshot shows the same SpiderFoot web interface, but with the 'Scan Name' field filled with the text 'ip scan'. The 'Scan Target' field is empty. The 'By Use Case' section remains the same, with 'All' selected. The 'Run Scan Now' button is now visible at the bottom of the form. The same link 'Don't want to manage your SpiderFoot installation yourself? Check out SpiderFoot HX.' is present at the very bottom.

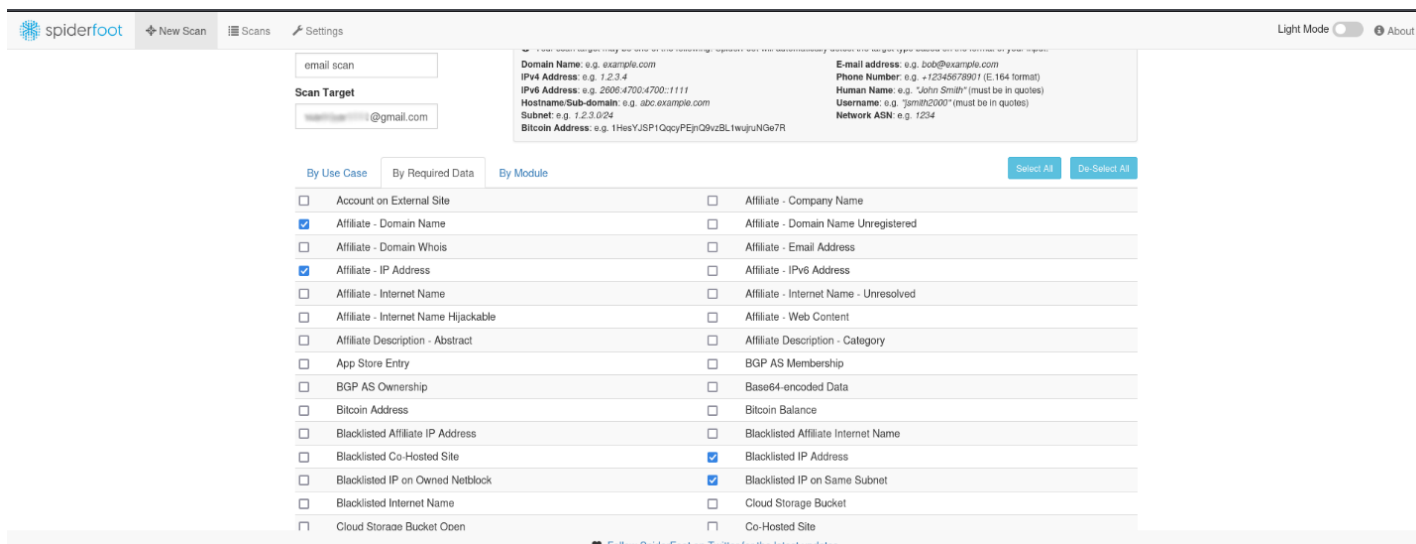
## 9.This is the result of our scan.



The screenshot shows the SpiderFoot web interface. At the top, there's a navigation bar with 'spiderfoot', 'New Scan', 'Scans', and 'Settings'. On the right, there are 'Light Mode' and 'About' links. Below the navigation bar, the main heading is 'Ip scan' with a 'RUNNING' status indicator. Underneath, there are tabs for 'Summary', 'Correlations', 'Browse', 'Graph', 'Scan Settings', and 'Log'. To the right of these tabs are icons for refresh, download, and a search bar. The main content area displays a table with the following data:


Type	Unique Data Elements	Total Data Elements	Last Data Element
IP Address	1	1	2024-05-08 18:33:32
Raw Data from RIRs/APIs	1	1	2024-05-08 18:33:35

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.



The screenshot shows the SpiderFoot web interface for configuring an email scan. The 'email scan' is selected in the 'Scan Target' field. Below this, there's a 'Scan Target' field with a placeholder email address. To the right, there's a list of target types with example values: Domain Name, IPv4 Address, IPv6 Address, Hostname/Sub-domain, Subnet, Bitcoin Address, E-mail address, Phone Number, Human Name, Username, and Network ASN. Below this, there are three tabs: 'By Use Case', 'By Required Data', and 'By Module'. The 'By Use Case' tab is active, showing a list of checkboxes for various scan modules. The 'By Required Data' tab is also visible, showing a list of checkboxes for various data types. The 'By Module' tab is also visible, showing a list of checkboxes for various modules. At the bottom, there are 'Select All' and 'De-Select All' buttons. Below the list of checkboxes, there's a footer with a heart icon and the text 'Follow SpiderFoot on Twitter for the latest updates.'

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

New ScanScansSettings

Light Mode

email scanRUNNING

SummaryCorrelationsBrowseGraphScan SettingsLog

RefreshDownload

Search...

Search

Type	Unique Data Elements	Total Data Elements	Last Data Element
Email Address	1	1	2024-05-08 18:35:53