OSINT Automation Lab

Advanced Osint

M Sameer Malik

OSINT  m.sameer.malik113@gmail.com

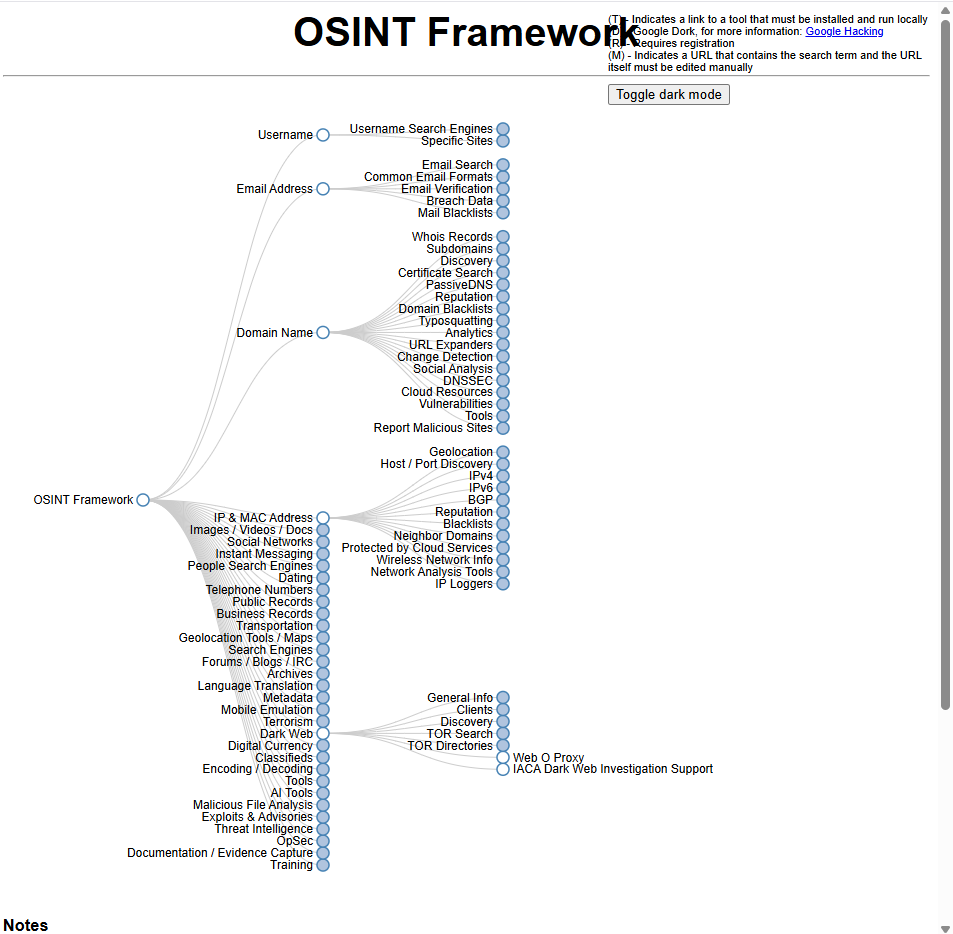
## **Instructions**

## **Part 1: Examine OSINT Resources**

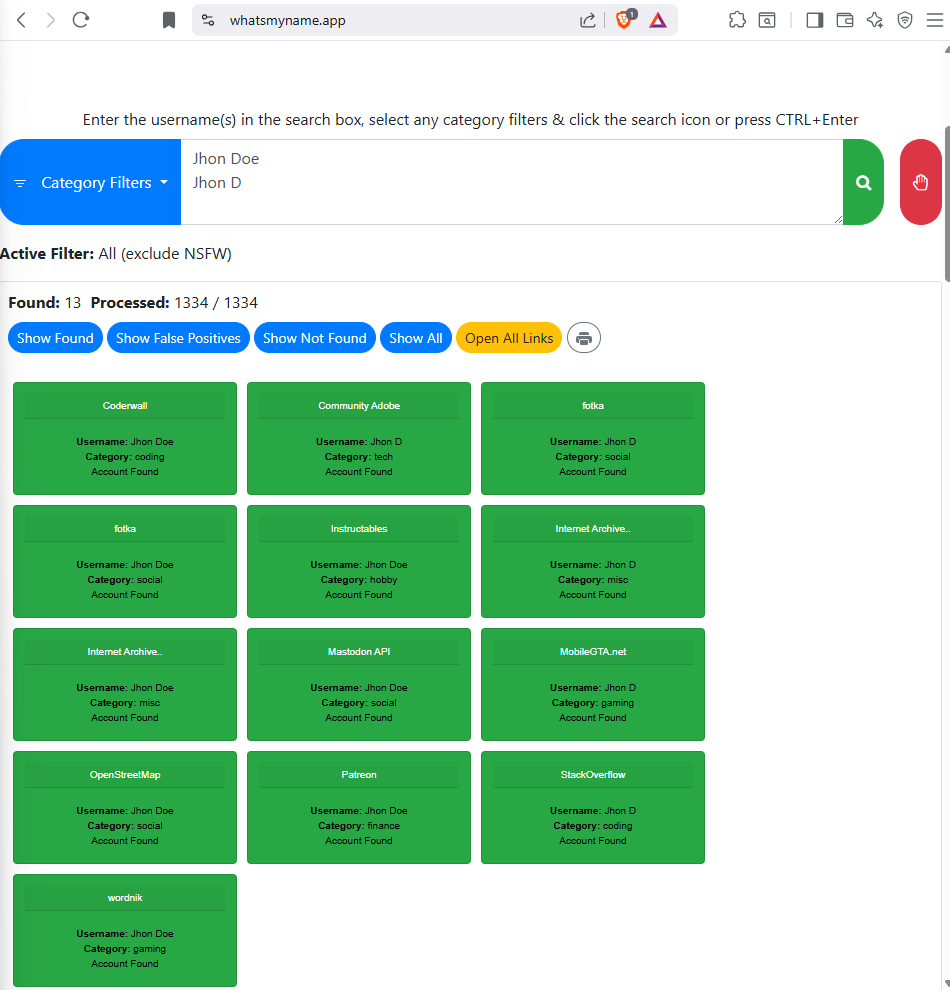
## **Step 1: Access the OSINT Framework**

The OSINT Framework is a useful way to visualize the OSINT tools and resources that are available. Unfortunately, it has become a bit outdated, with some resources no longer available. It is still valuable to help you understand the types of tools available and their uses. In many cases, the links are still good.

1. Go to the OSINT Framework site at <https://osintframework.com/>.



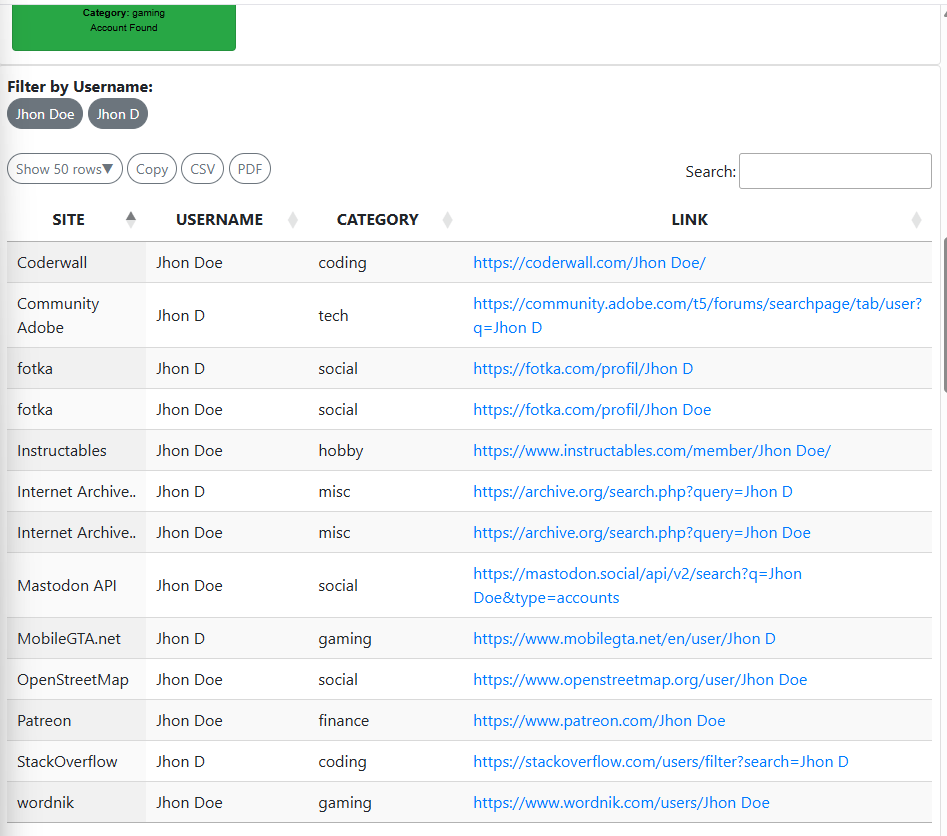
1. Under **Username Search Engines**, click **"WhatsMyName(T)"**.



1. The link takes you to a Git repository for the WhatsMyName project. In the **README.md** content for the tool, the various sites that implement WhatsMyName are listed. Feel free to explore these, but we will click the first link <https://whatsmyname.app/> to visit a free website that implements WhatsMyName.

The parent organization for the site, <https://www.osintcombine.com/>, has several interesting free tools available.

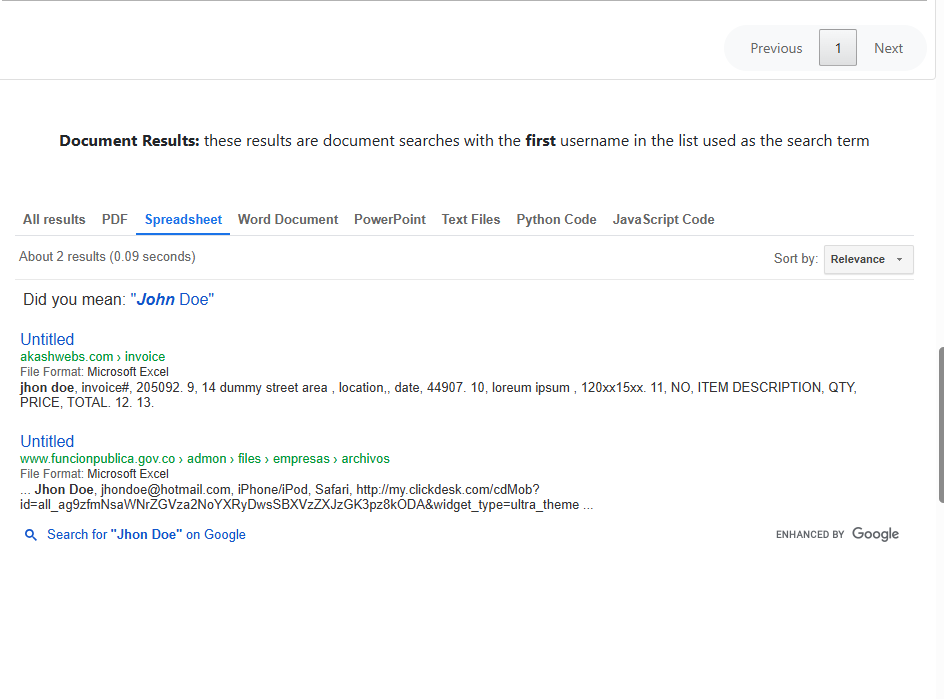
1. In the search box, type in a few usernames, each on a separate line. Use your own usernames or others that you find. Try searching the internet for **common username wordlist** for other potential search terms. You can filter the results based on the category filters, but for now, just click the green magnifying glass button to start the search.



In a pentest, you would use another tool, such as **SpiderFoot** (below) to find usernames in email addresses that are associated with a company or domain.

Investigate the results. You can open the links to the accounts either from the green rectangles or the table of results.

WhatsMyName provides a very flexible report of the results. The results table can be sorted by column, and you can export the results as CSV or PDF for reporting purposes. In addition, you can easily filter by username and search within the results. Finally, you get links for the profile pages for the users at many different sites.



**Part 2: Use SpiderFoot**

SpiderFoot is an automated OSINT scanner. It is included with Kali. SpiderFoot queries over 1000 open-information sources and presents the results in an easy-to-use GUI. SpiderFoot can also be run from a console. SpiderFoot seeds its scan with one of the following:

* Domain names
* IP addresses
* Subnet addresses
* Autonomous System Numbers (ASN)
* Email addresses
* Phone numbers
* Personal names

SpiderFoot offers the option of choosing scans based on use case, required data, and by SpiderFoot module. The use cases are:

* All – Get every possible piece of information about the target. This use case can take a very long time to complete.
* Footprint – Understand the target’s network perimeter, associated identities and other information that is yielded by extensive web crawling and search engine use.
* Investigate – This is or targets that you suspect of malicious behavior. Footprinting, blacklist lookups, and other sources that report on malicious sites will be returned.
* Passive – This type of scan is used if it is undesirable for the target to suspect that it is being scanned. This is a form of passive OSINT.

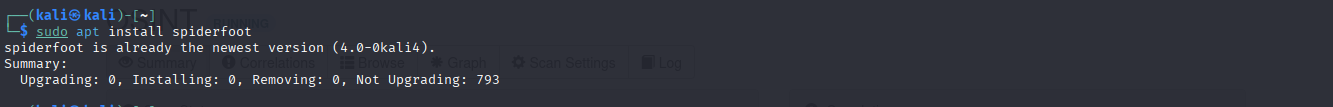
**Step 1: Start and run SpiderFoot.**

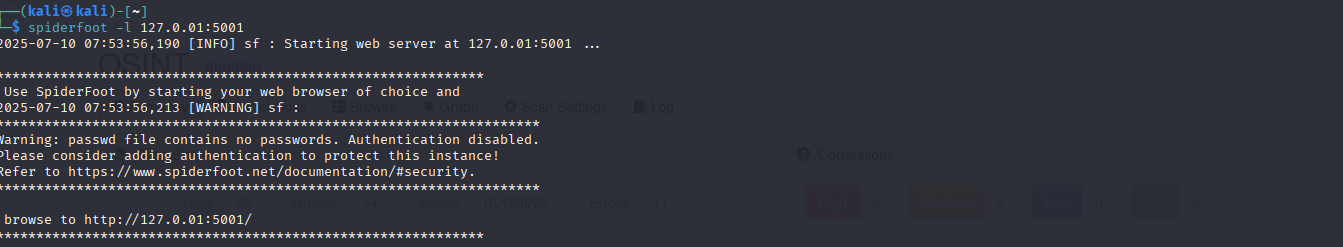
In a terminal, enter the following command:

┌──(kali㉿Kali)-[~]

└─$ **spiderfoot -l 127.0.0.1:5001**

The command should run without errors. Open a browser and enter the IP address and port for the SpiderFoot GUI. You will see the SpiderFoot interface appear. If this is the first time that SpiderFoot has been opened in this VM, you will see the Scans screen.

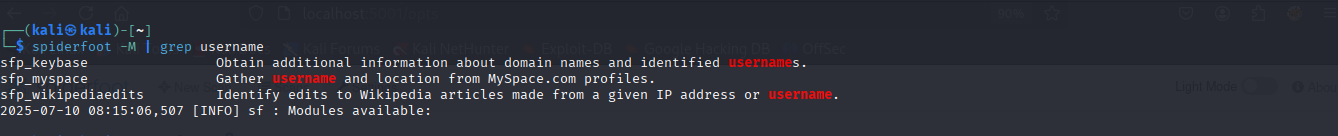




**Step 2: Explore SpiderFoot.**

┌──(kali㉿Kali)-[~]

└─$ spiderfoot -M | grep *[search term]*

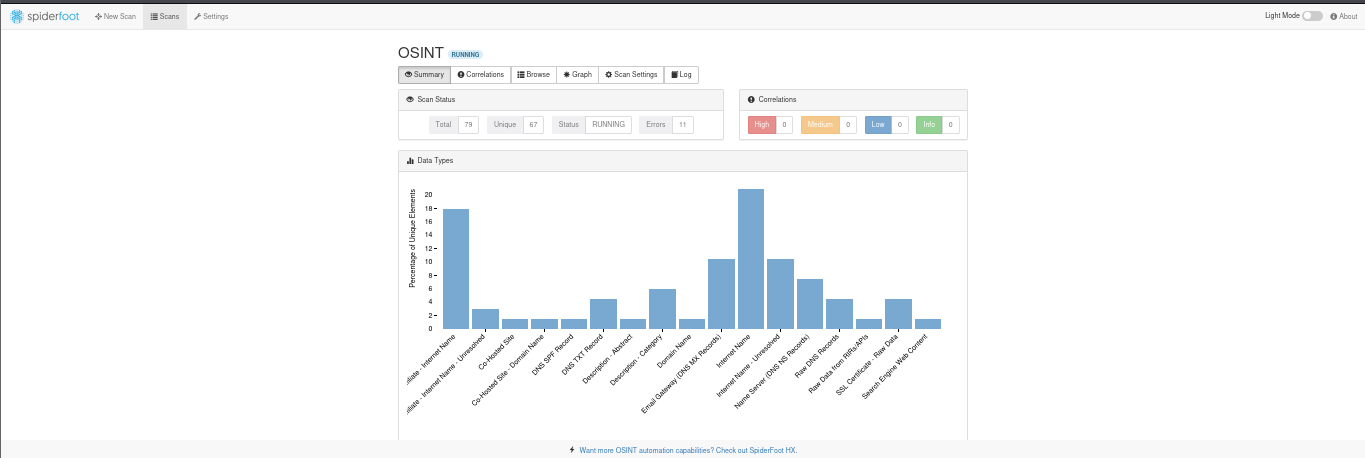


**Step 3: Run a SpiderFoot Scan**

1. Click the **New Scan** tab in the GUI.
2. Enter a name for the scan and select a target. In this case, we will use **hackthissite.org**.
3. We will scan by use case. Note that you can also scan by the type of information required or by selecting the individual scanner modules that you would like to use. By executing narrower scans, you can learn more about the modules and information that can be gathered.

**Note**: The **All** use case scan may use active scanning. Unless you have permission to scan the target, you should avoid this setting. To be completely safe, the Passive use case should avoid any problems with unauthorized scanning.

1. Click the **Run Scan Now** button.
2. You should see a bar graph appear. The scan statistics will start to increment, and new bars will appear in the graph as new results are obtained. Mouse over the bars for a summary of the findings for that data type.





**Part 3: Investigate Recon-ng**

Recon-ng is an OSINT framework that is similar to the Metasploit exploitation framework or the Social-Engineering Tooklit (SET). If consists of a series of modules that can be run in their own workspaces. The modules can be configured to run with option settings that are specific to the module. This simplifies running Recon-ng at the command line because options for the modules are independently set within the workspace. When you run the module, it uses these settings to perform its searches.

As the name suggests, Recon-ng is used to perform a wide range of reconnaissance activities on different settings that you provide. Some modules are available with the Kali installation and others are available for download and installation in the Recon-ng modules marketplace.

**Step 1: Create a workspace.**

Recon-ng has auto complete. Press the tab button to complete commands and command options. Use the tab key twice to list the available commands and options at different places in the command line. This is very handy.

1. To run Recon-ng, open a new terminal window and enter **recon-ng**. You can also start the program by going to the Kali tools menu, searching for the app, and clicking the icon.



1. Note that the terminal prompt changes to indicate that you are working within the Recon-ng framework. Enter **help** to get a sense of the commands that are available.



1. Recon-ng uses workspaces to isolate investigations from one another. Workspaces can be created for different parts of a test or different customers for example. Type **workspaces help** to view options for the workspaces command.
2. Create a workspace named **test** by entering **workspaces create** followed by the workspace name. Note that the prompt has changed to indicate that you are in this workspace.



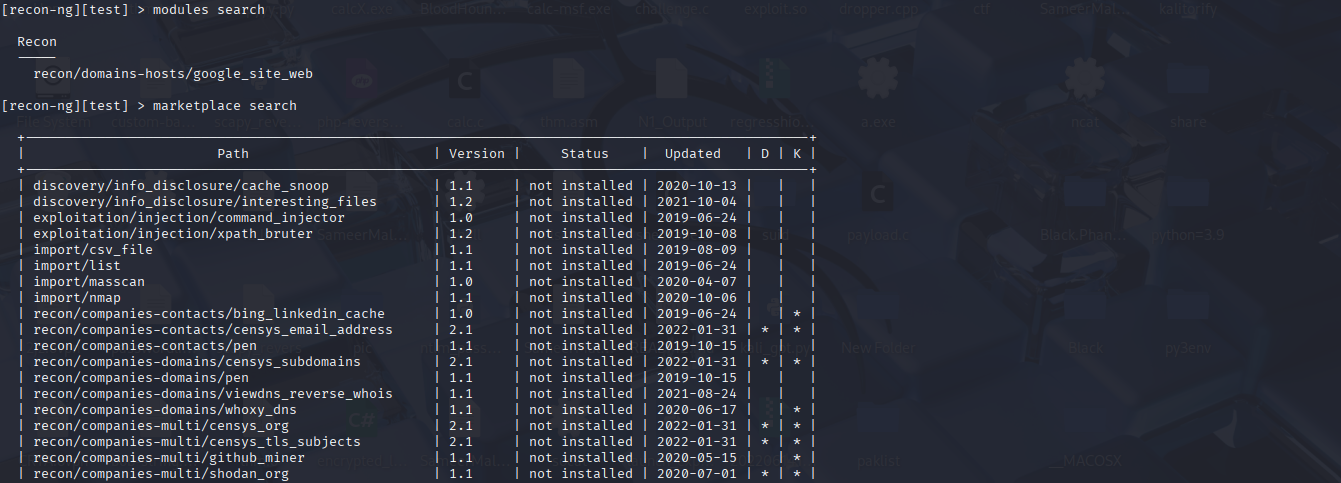
1. Type**help** to see the commands that are available within workspaces.



**Step 2: Investigate modules.**

Recon-ng is a modular framework. Modules are Python programs with different functions. They are stored in an external marketplace that permits developers to create their own modules and contribute them for use by others.

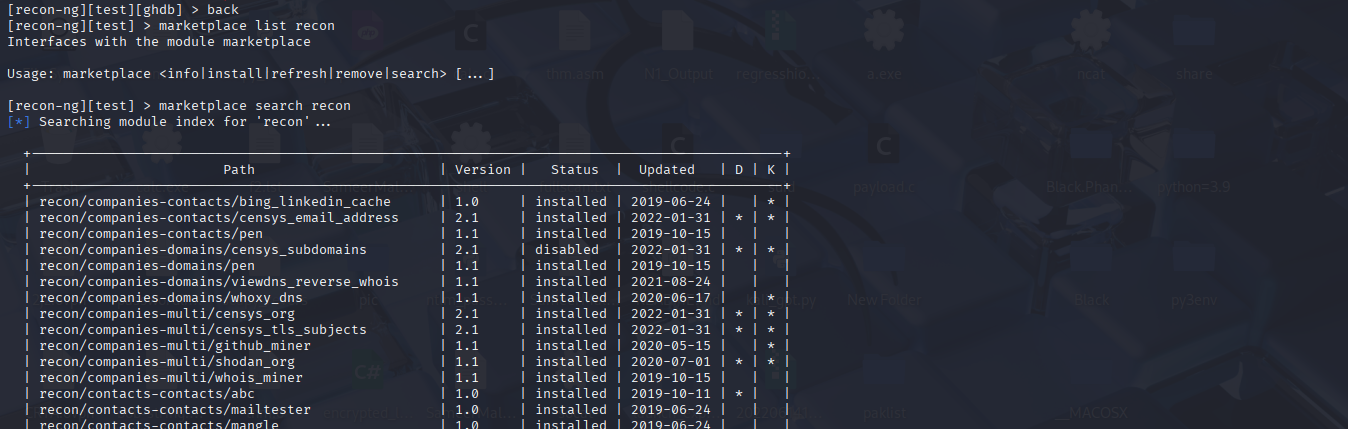
Return to the Recon-ng prompt. Enter the **modules search** command. This will display the currently installed modules.



**Step 3: Investigate the module marketplace.**

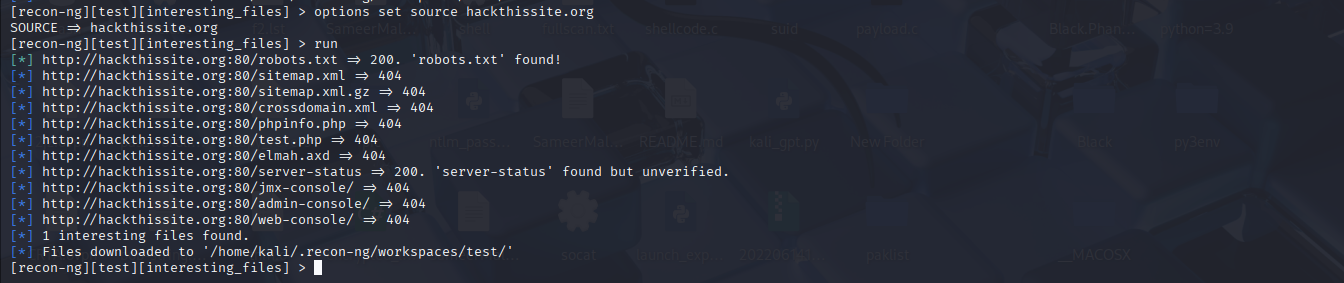
Recon-ng will not function without modules. In this step, we will install modules from the Recon-ng marketplace. The module marketplace is a GitHub public repository. Search the web for **recon-ng-marketplace** to view the repository. Explore the folders to learn more about the modules.

1. In the terminal, view help for the **marketplace** command. Use the **search** option to list all the modules that are currently available.

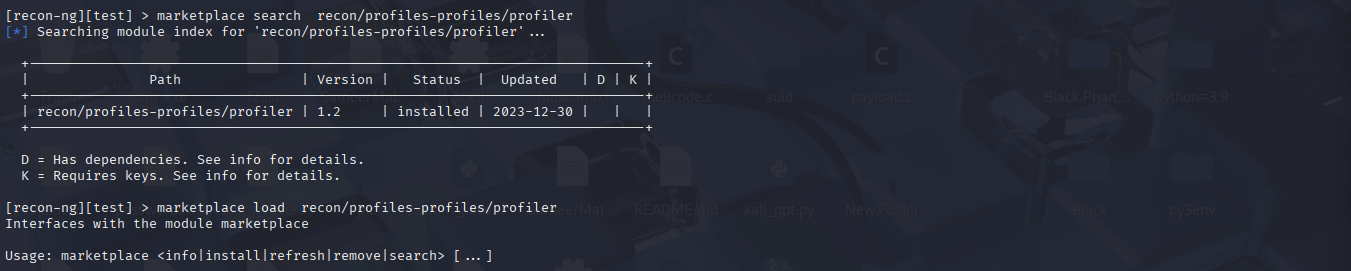


Note that the modules are organized by their category and type. This appears as a path prepended to the name of the module. You can filter the output by adding a search term to the marketplace search command. Try a few different search terms that are related to OSINT information to get a sense of the modules that are available.

1. Instead of passing options at the command line, in Recon-ng you set the options and then enter a simple command to execute the module. Use the **options set source** command to set the only option for this module. Complete the command by specifying the target as **hackthissite.org**.



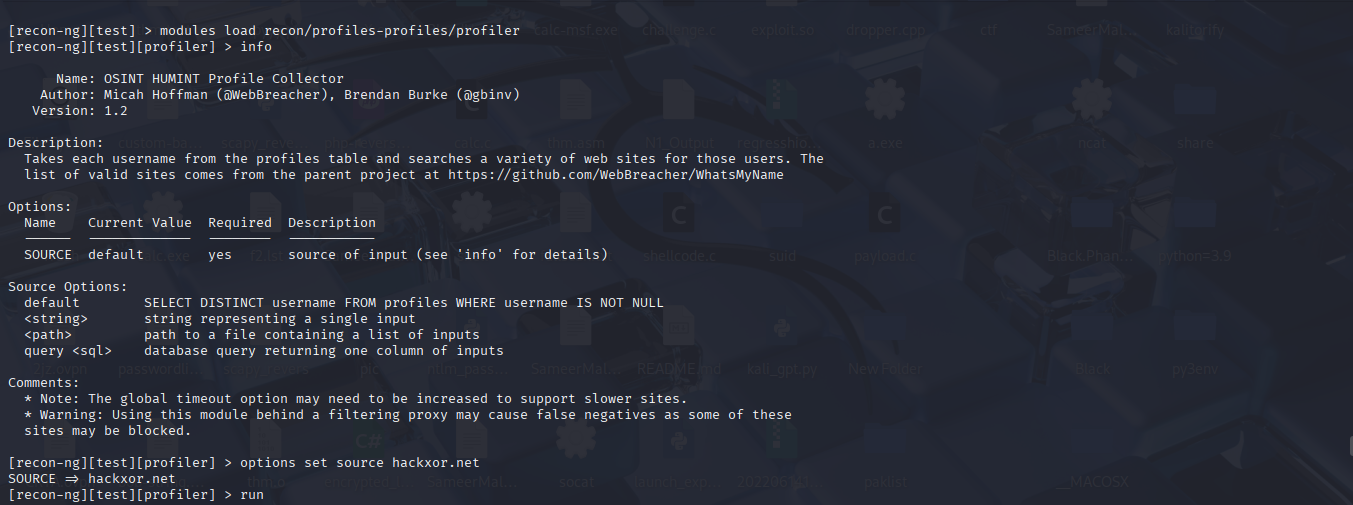
Now for the practice purposes we will repeat the same steps with another module, I choose the *recon/profiles-profiles/profiler.*



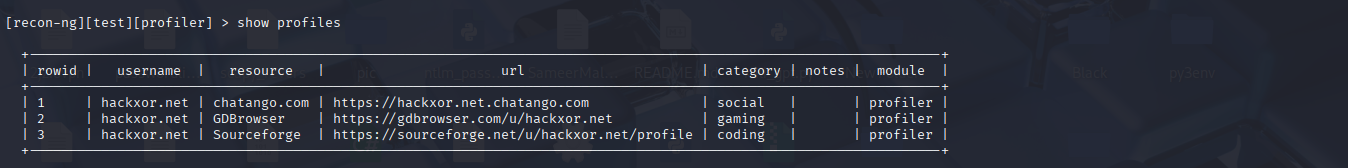
**Info** command is used to gather general information about a specific module.



The ***load module*** command is used to load a specific module into the memory upon which we can operate and execute commands this plays a vital role as it is necessary to load the module in the memory to operate upon it.



After running this module we can see that it managed to gather 3 profiles related to the source and each one can viewed separately.

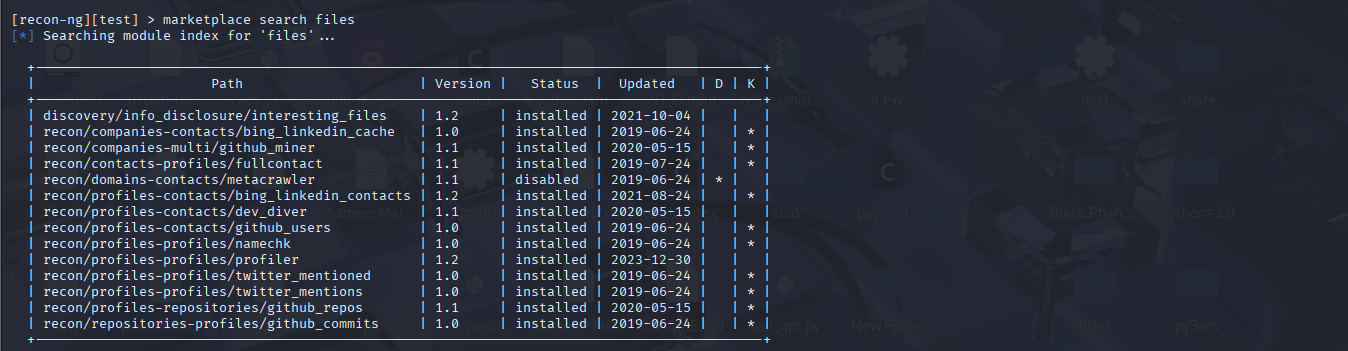


**Find Interesting Files with Recon-ng**

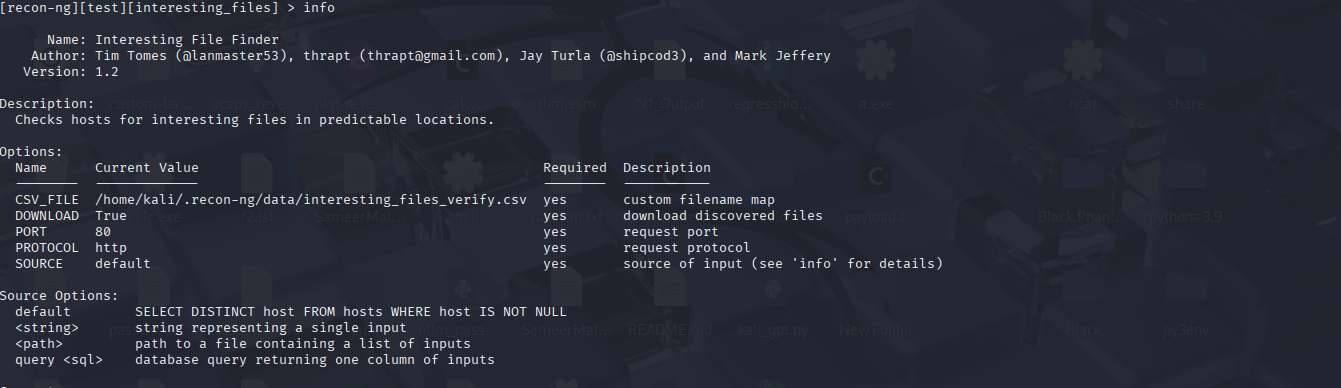
In this part of the lab, we will install and use another plugin.

**Step 1: Install another module.**

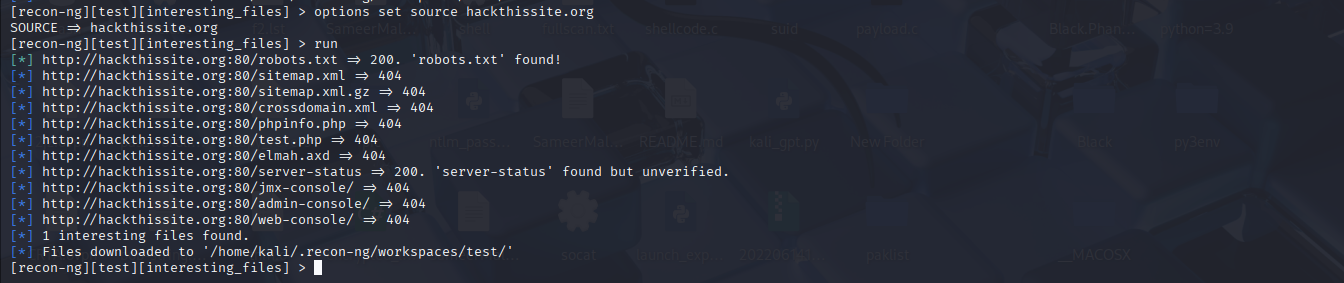
1. Search the marketplace for a module that will discover interesting files in a domain. The plugin that you use should have no dependencies or key requirements.



Set the source option to **hackthisite.org** or another location of your choice. (Please comply with the terms of the course when choosing a domain.) The h4cker.org website is interesting also.



Now we will set the source to hackthissite.org and run the module. As we can see as the module completes it is automatically saved under a directory!



**Conclusion:**

The following lab demonstrated the use of OSINT tools with a special focus on ***OSINT FRAMEWORK,*** Spiderfoot and Recon-ng. These automated tools saves a ton of time and empowers to perform deep inspection on the target. Thus leveraging these tools one can boost his efficiency and focus on the work that cant be automated!