Using OSINT Tools

Introduction

When performing information gathering activities, passive reconnaissance uses open, publicly accessible data to guide active reconnaissance efforts and to gather information about the enterprise and employees. In OSINT, it is the data that is open source. OSINT tools may or may not be open source. OSINT commonly uses data sources that are available to any hacker, so part of the PenTesting effort is to report on sensitive information that is commonly available in order to evaluate vulnerabilities that it may cause. The objectives of OSINT are:

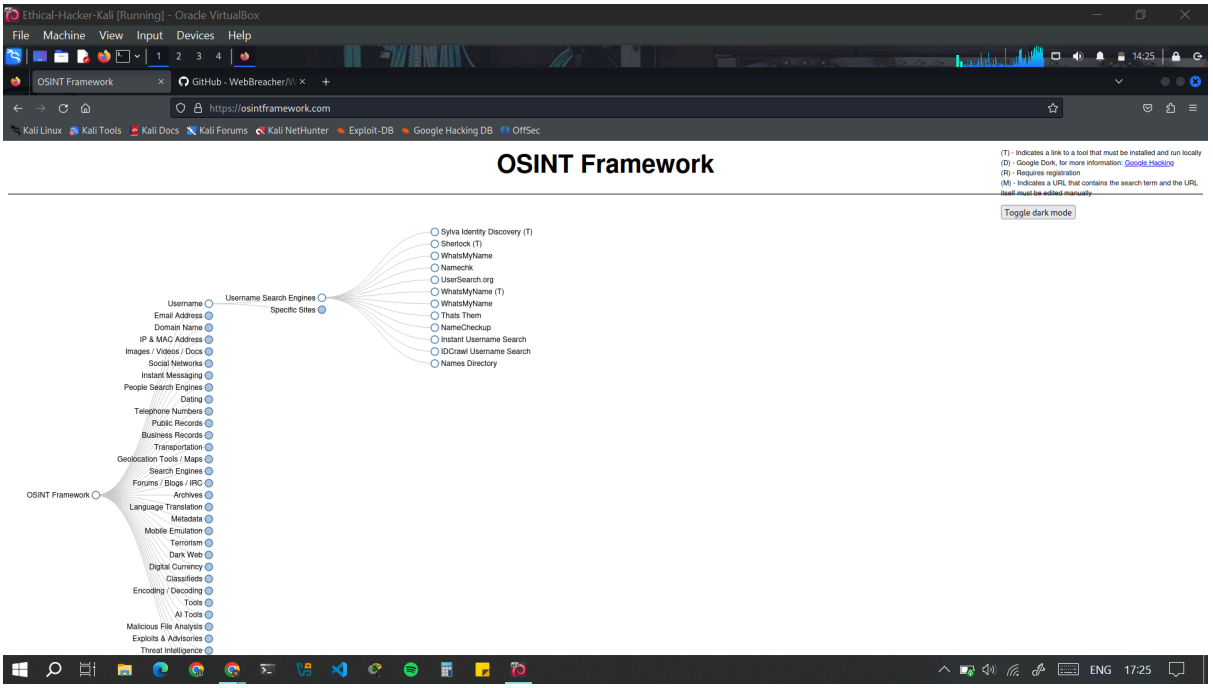
- To determine the digital footprint of the organization.
- Determine what data about the organization is available to cyber criminals.

Part 1: Examine OSINT Resources

Step 1: Access the OSINT Framework

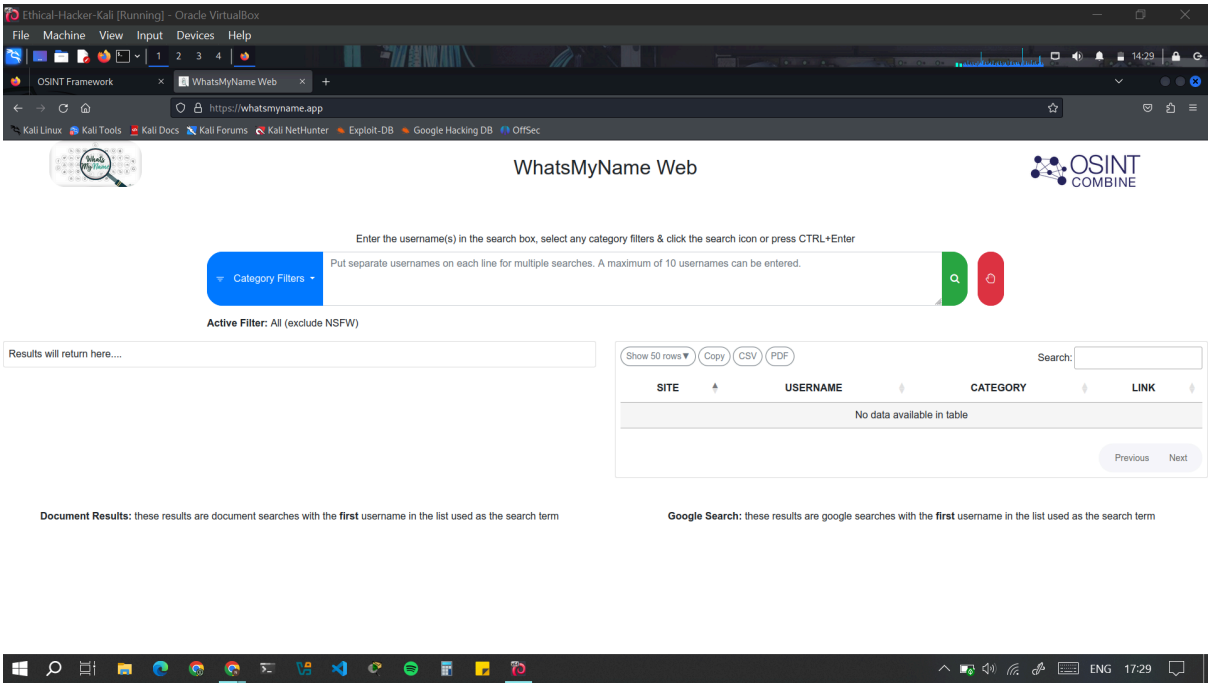
The OSINT Framework is a useful way to visualize the OSINT tools and resources that are available at https://osintframework.com/.

I clicked **Username** at the top of the tree and under **Username Search Engines,** clicked *WhatsMyName(T)* tool. The T Indicates a link to the tool that must be installed and run locally

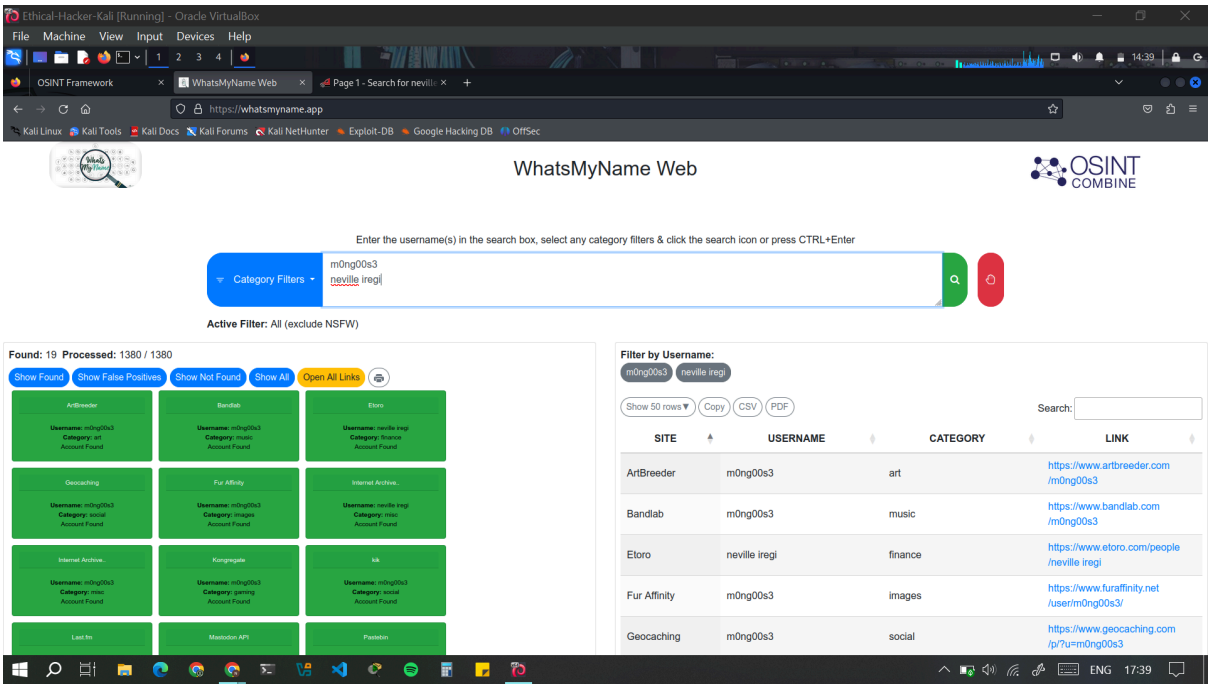


The link takes you to a Git repository for the WhatsMyName project. In the README.md

content for the tool, the various sites that implement WhatsMyName are listed. The tool has a web interface at https://whatsmyname.app/



In the search box, I typed in a some usernames, each on a separate line.



- 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.

The value of doing username searches and account enumeration include:

1. Username searching can identify accounts that important enterprise personnel may have on various sites. Because other sites can be vulnerable, it is possible that

hackers could gain access to personnel information from those accounts, including passwords, addresses, and telephone numbers.
2. The types of sites that personnel have registered for can also provide details of their lives and interests. These details could be used in **social engineering attacks.**

Step 2: Investigate SMART - Start Me Aggregated Resource Tool.

The start.me web service is a popular bookmark manager and productivity tool. The people at My OSINT Training (MOT) have set up a search system that finds all OSINT-related links that people have bookmarked and shared on start.me. You can enter OSINT-relevant search terms to find links to related resources.

At the time of writing, this project has been shut down.

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:

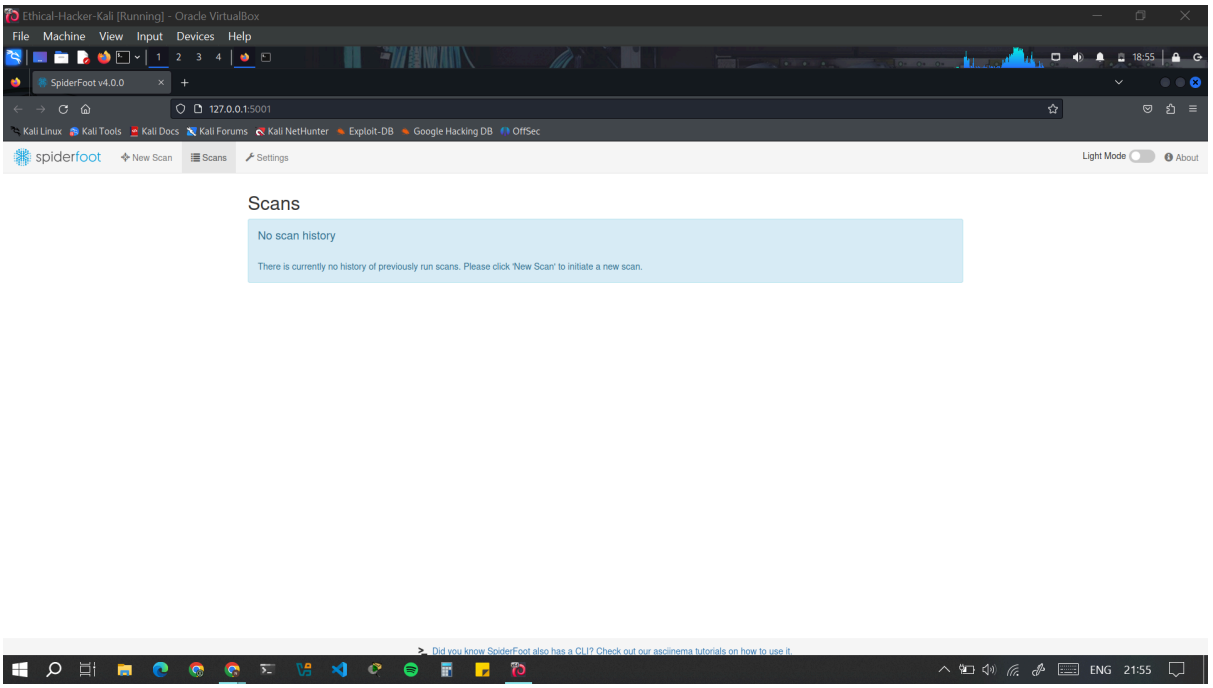
- Domain names
- IP addresses
- Subnet addresses
- Autonomous System Numbers (ASN) - a unique identifier (16-bit or 32-bit integer) assigned to a network, such as an ISP or large organization, to enable routing information exchange across the internet using the Border Gateway Protocol (BGP)
- Email addresses
- 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:

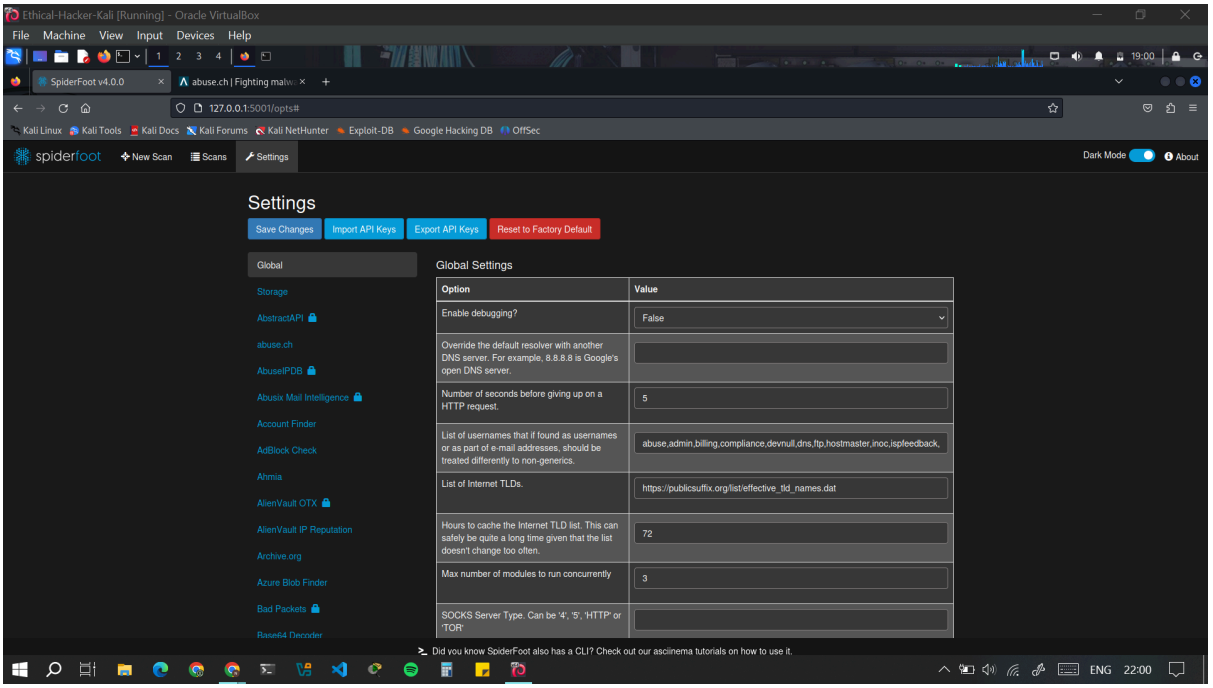
1. All – Get every possible piece of information about the target. This use case can take a very long time to complete.
2. Footprint – Understand the target's network perimeter, associated identities and other information that is yielded by extensive web crawling and search engine use.
3. Investigate – This is for targets that you suspect of malicious behavior. Footprinting, blacklist lookups, and other sources that report on malicious sites will be returned.
4. 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.

Using SpiderFoot in a terminal: `└─$ spiderfoot -l 127.0.0.1:5001`

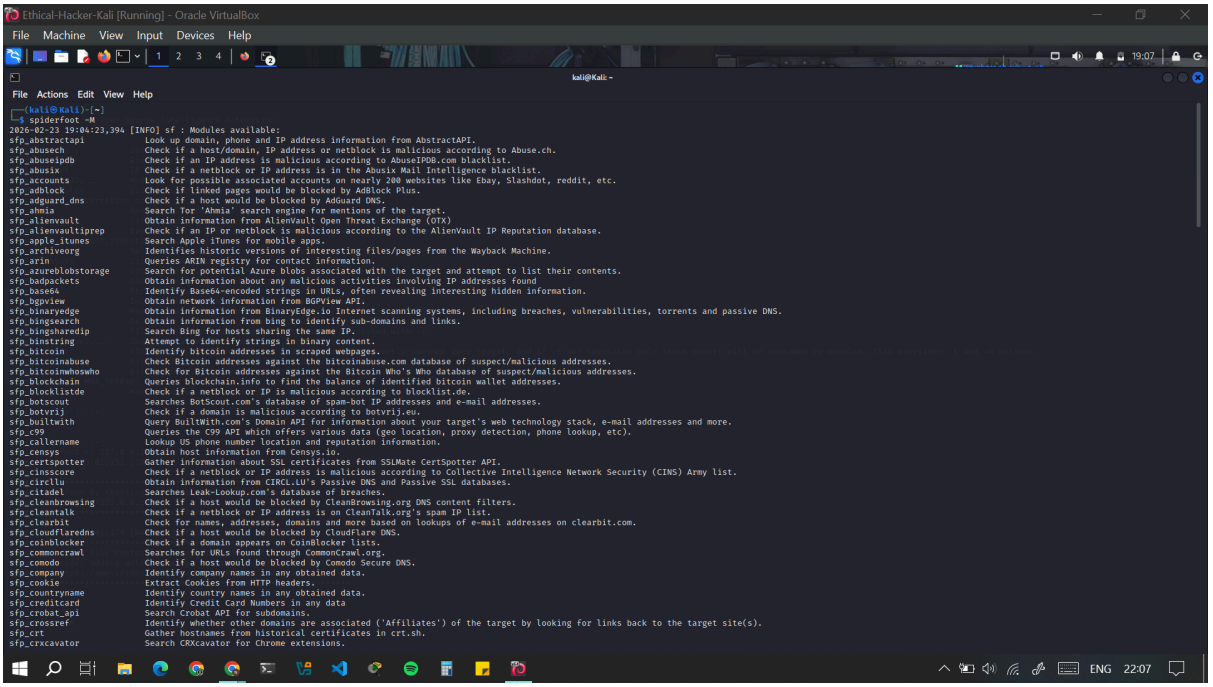
- -l indicates the ip and port to listen on
- Running the command opens a SpiderFoot web interface that displays a list of all the scans ran recently.



SpiderFoot uses over 200 scanners to build its reports. The Scanner name is in the settings menu. The module name appears in the details for the scanner. All SpiderFoot modules are referred to as sfp_[module name].

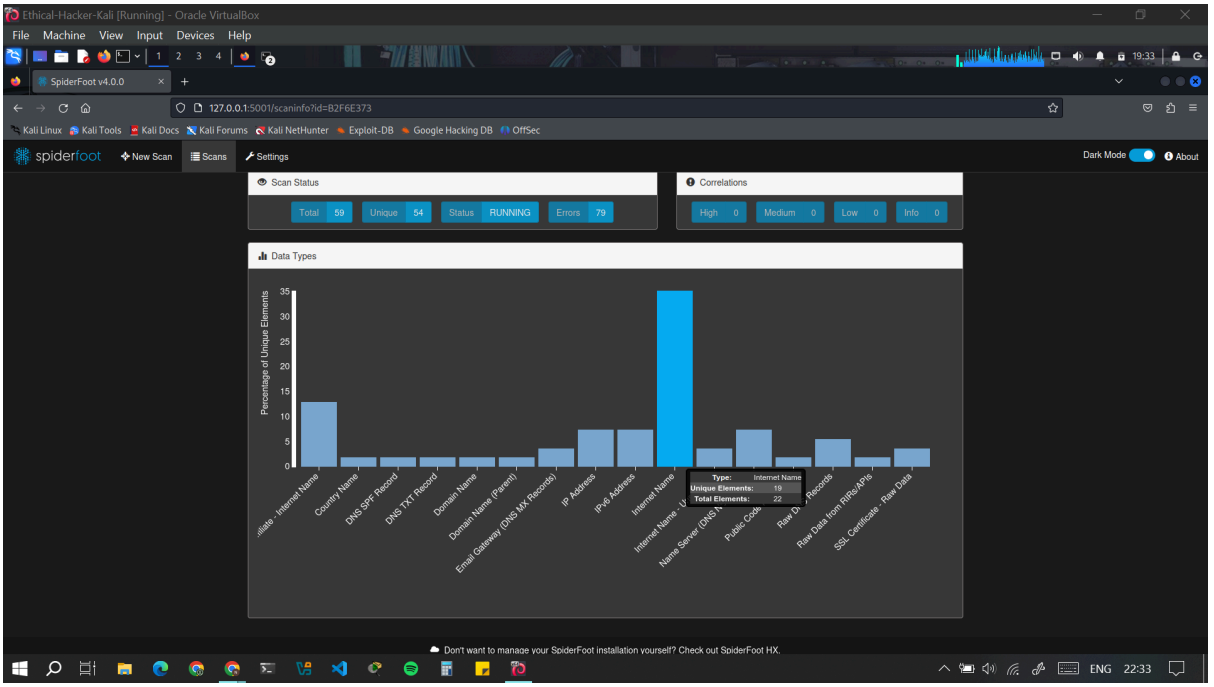


- scanners with a lock next to them indicate an API key is necessary. Further information regarding the key requirements is provided in the details for the scanner. Click the “?” icon next to the API Settings option.
- You can display all the modules that are available in SpiderFoot using spiderfoot -M



A typical SpiderFoot scan for a domain in the GUI e.g h4cker.org requires one to select ‘new scan’ and enter the scan name and scan target as well choosing which method to scan by (use case/required data/individual module)

- Using the use case method, I selected ‘Footprint’
  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.
- On running the scan, a bar graph appears. 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.
- SpiderFoot scans are very detailed and can take a very long time. Give this scan at least 30 minutes so that there is a nice collection of information. To get the most details, a scan could take hours. While the scan is running, you can browse the results.

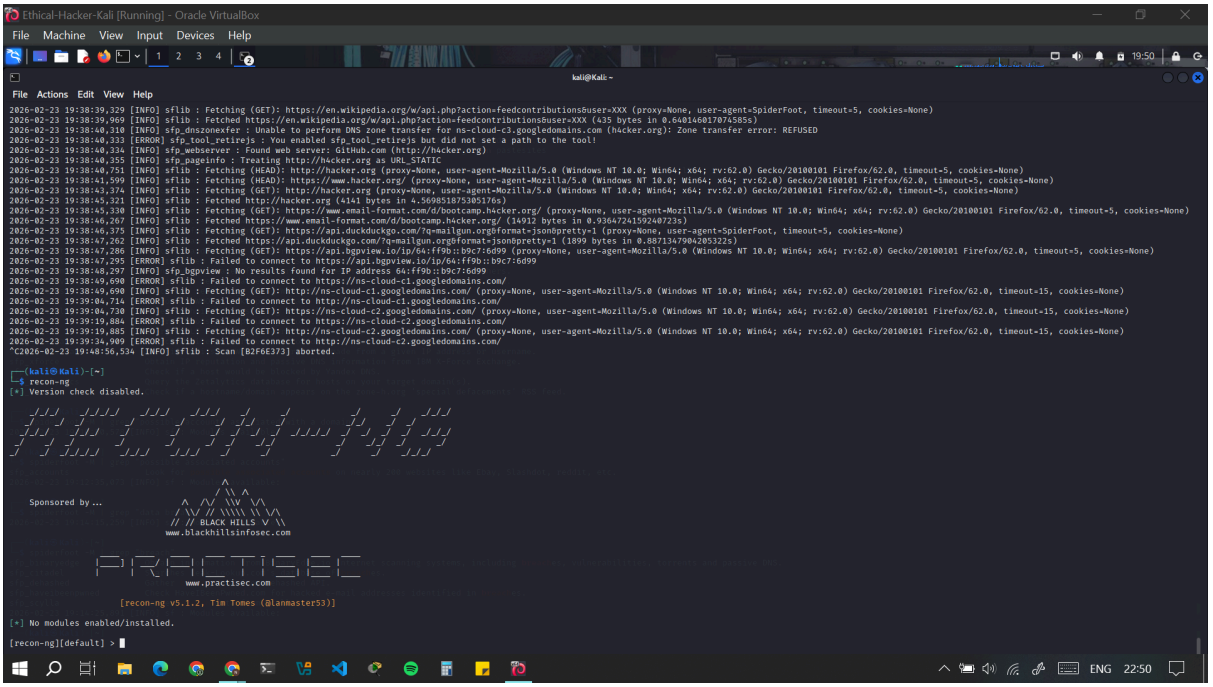


Part 3: Investigate Recon-ng

Recon-ng is an OSINT framework that is similar to the Metasploit exploitation framework or the Social-Engineering Toolkit (SET). It 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.

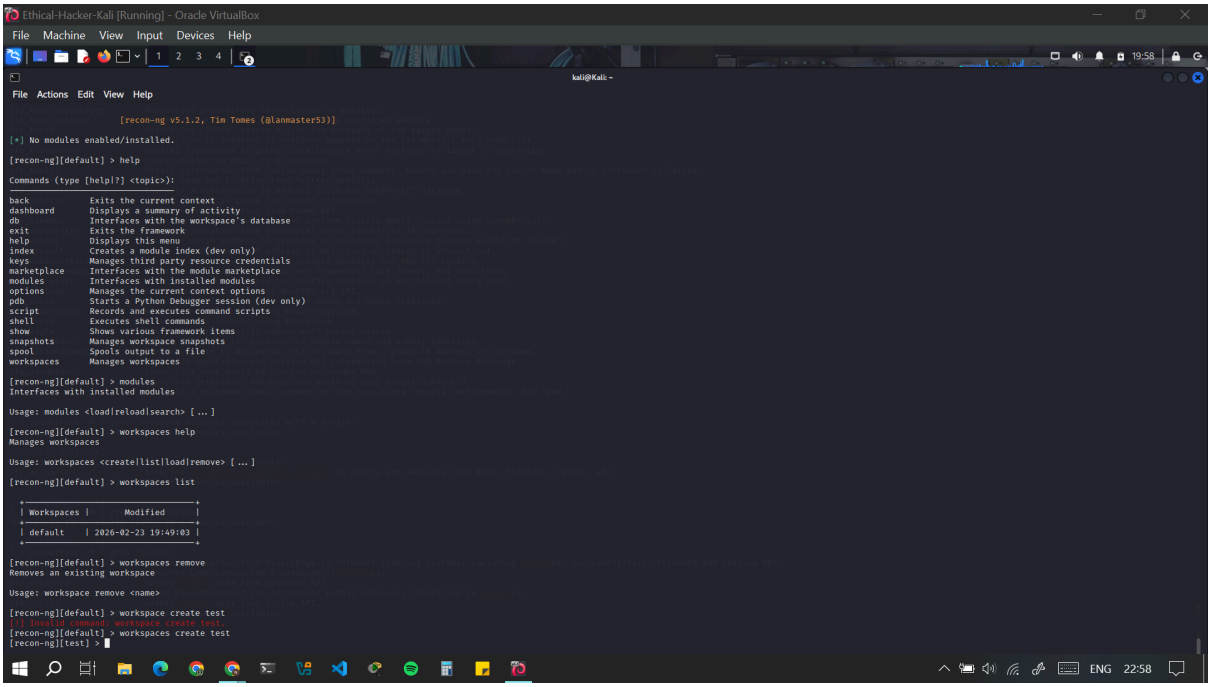
To run Recon-ng, open a new terminal window and enter **recon-ng**. You can also start the program by going to the Kali tools menu, searching for the app, and clicking the icon.



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. The workspaces feature preserves the installed module and their configured options. Workspaces also have their own results data. Workspaces can be configured for individual tests or for an entire PenTesting engagement.

- One can type ‘workspaces help’ to view options for the workspaces command. ‘Workspaces list’ displays the available workspaces.

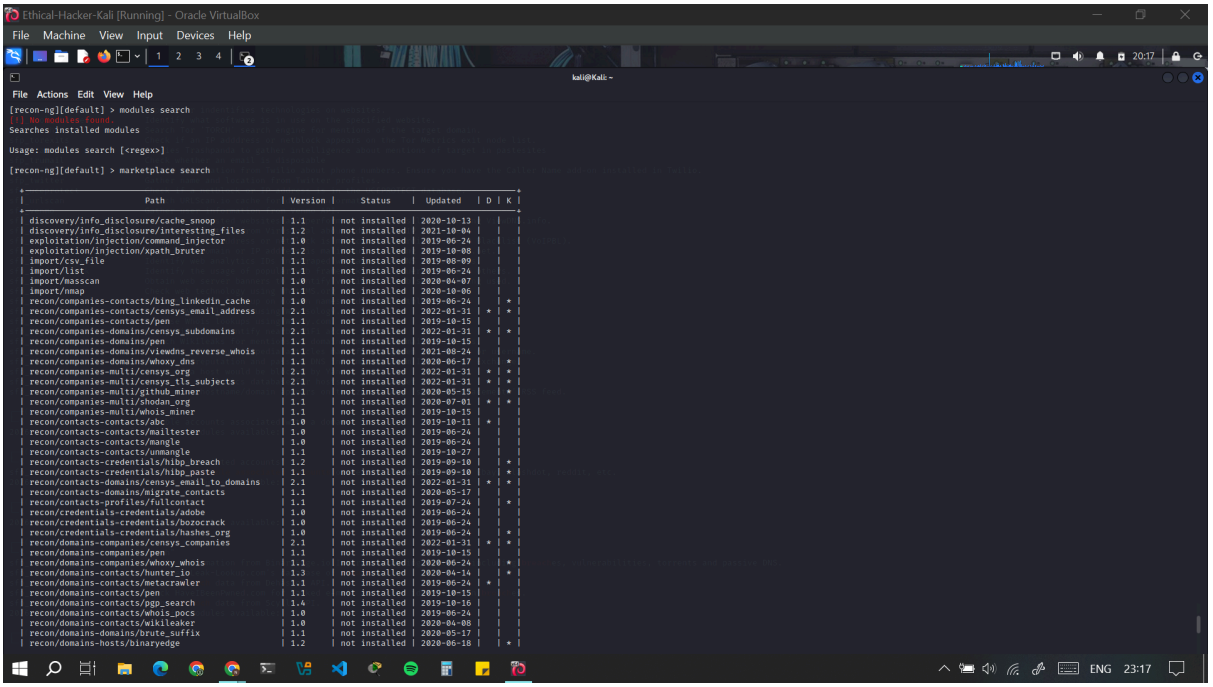
I created a workspace named ‘test’ by entering ‘workspaces create’ followed by the workspace name. The prompt changed from default to test to indicate that you are in this workspace.



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.

- Command ‘modules search’ displays the currently installed modules

Recon-ng cannot function without modules which can be installed from the recon-ng marketplace (a github public repo). Using ‘marketplace search’ once can list all the modules currently available.

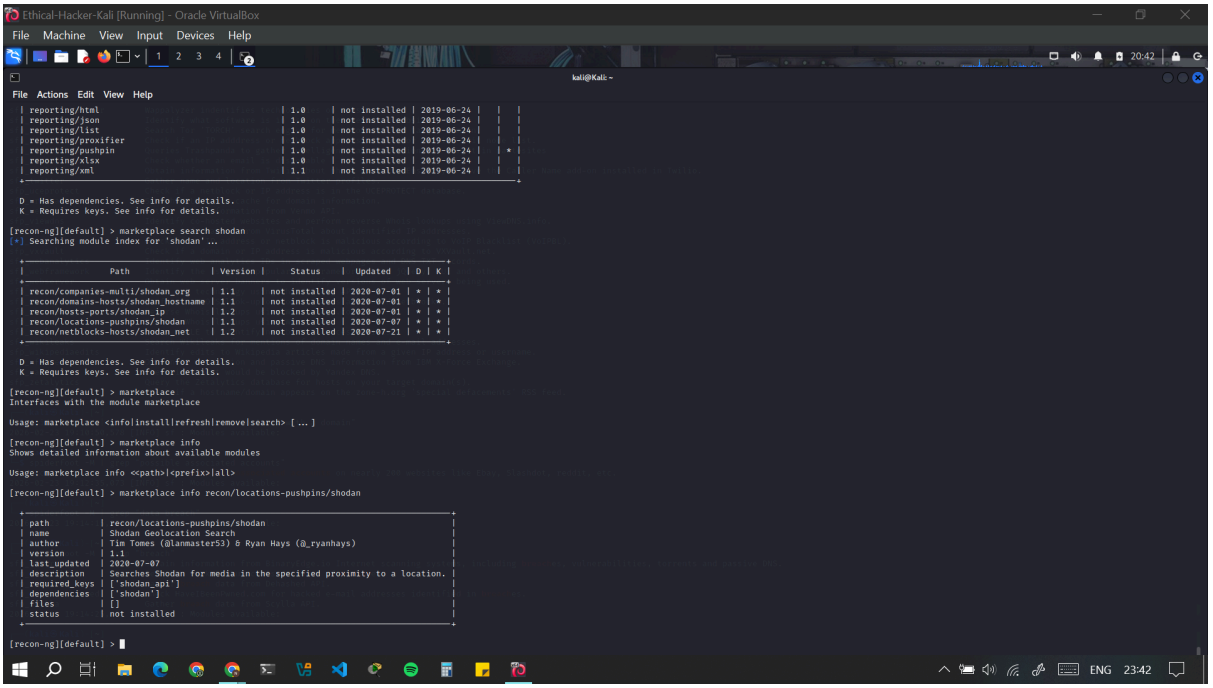


The module tables have columns for D and K.

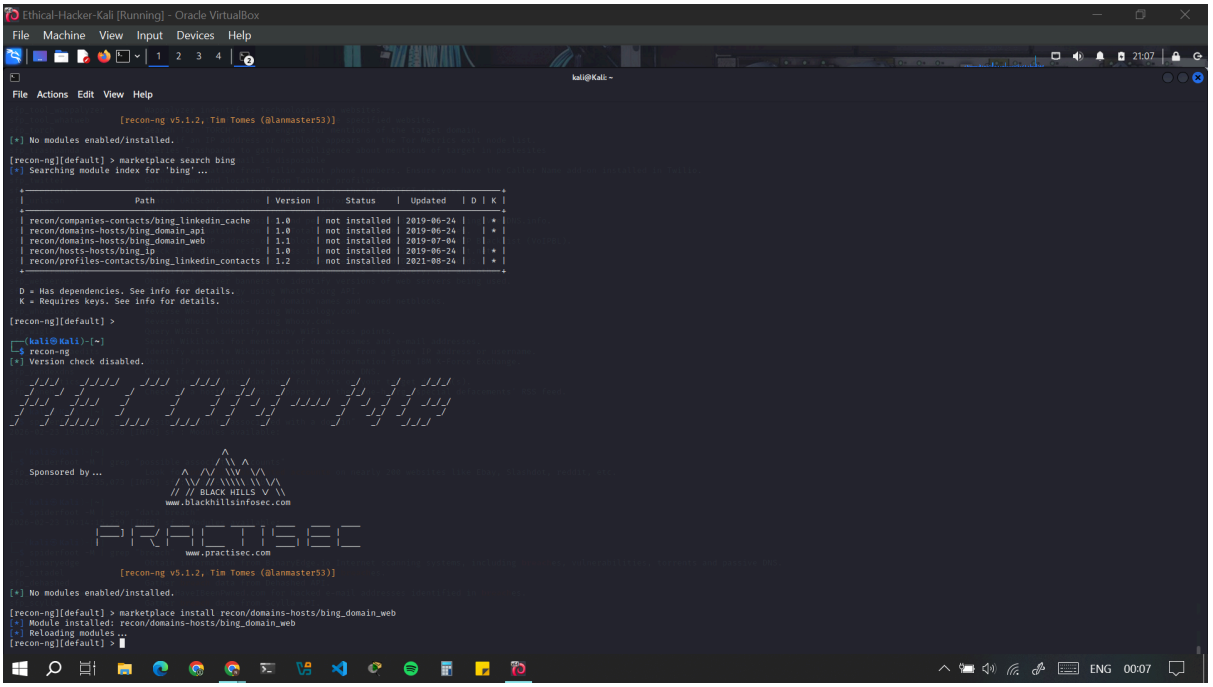
- D = Has dependencies.

- K = Requires keys.

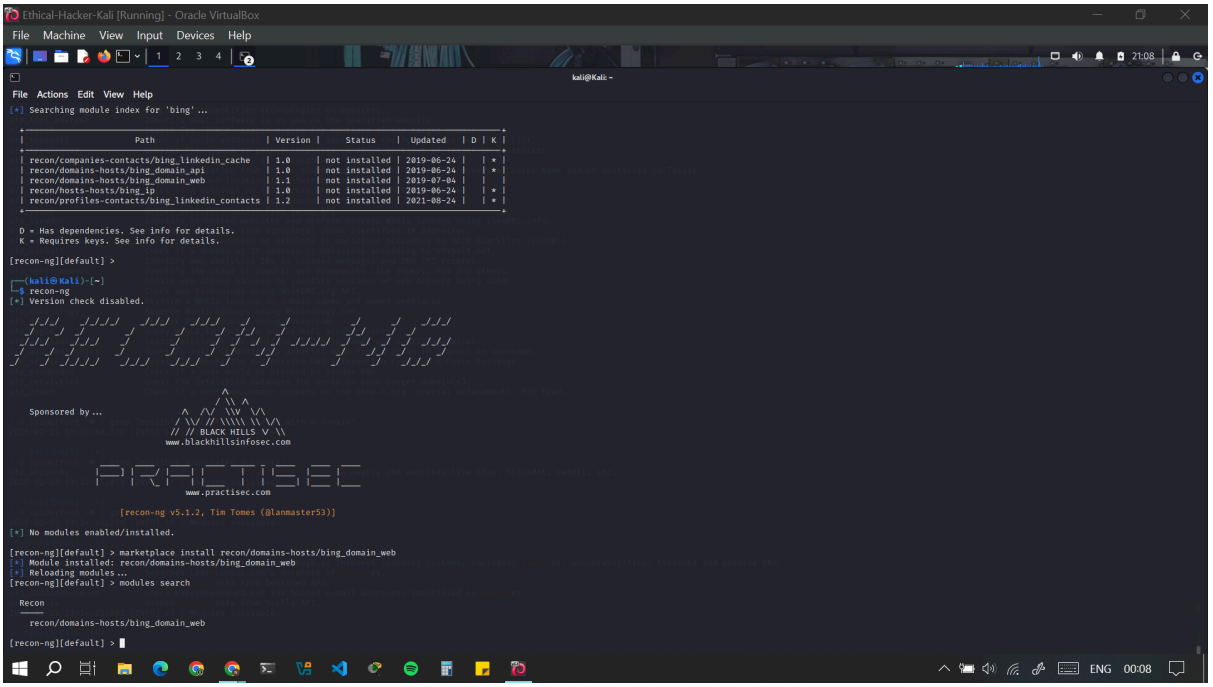
The command ‘marketplace info <<pathname of module>>’ tells more about individual module. For example:



Recon-ng accesses modules from the Github repository and downloads them to Kali when they are installed. To install a module, one uses ‘*marketplace install <<full name of module>>*’

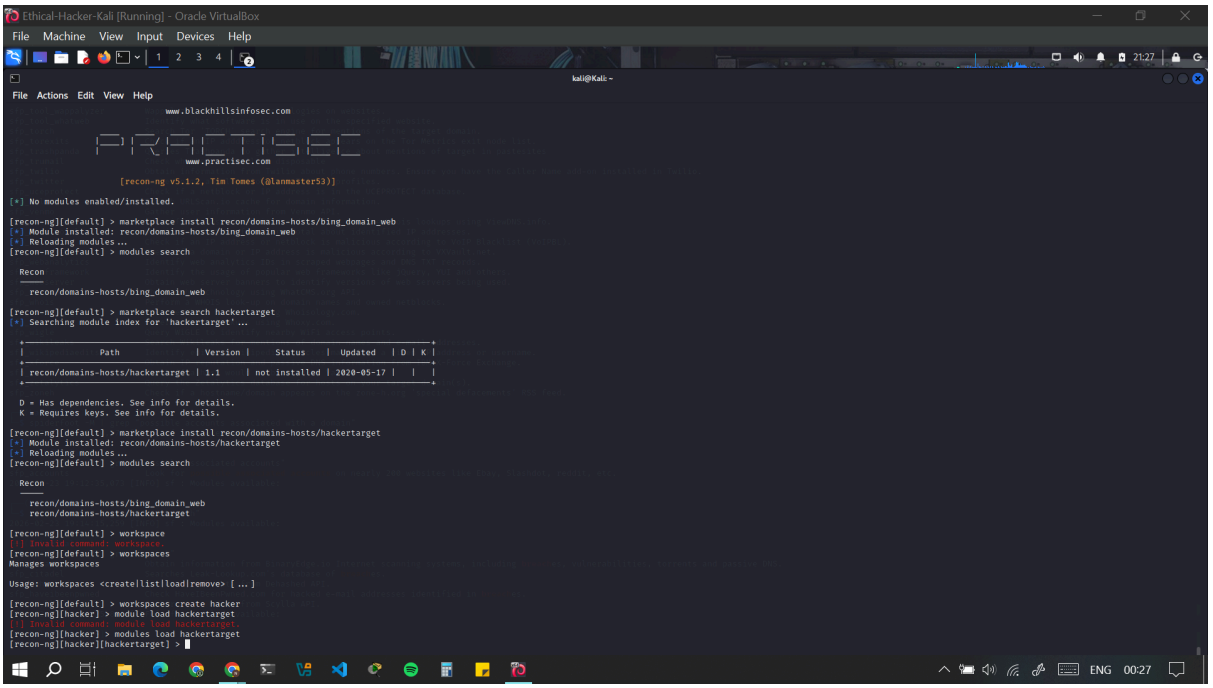


- ‘Modules search’ command should confirm the module is available

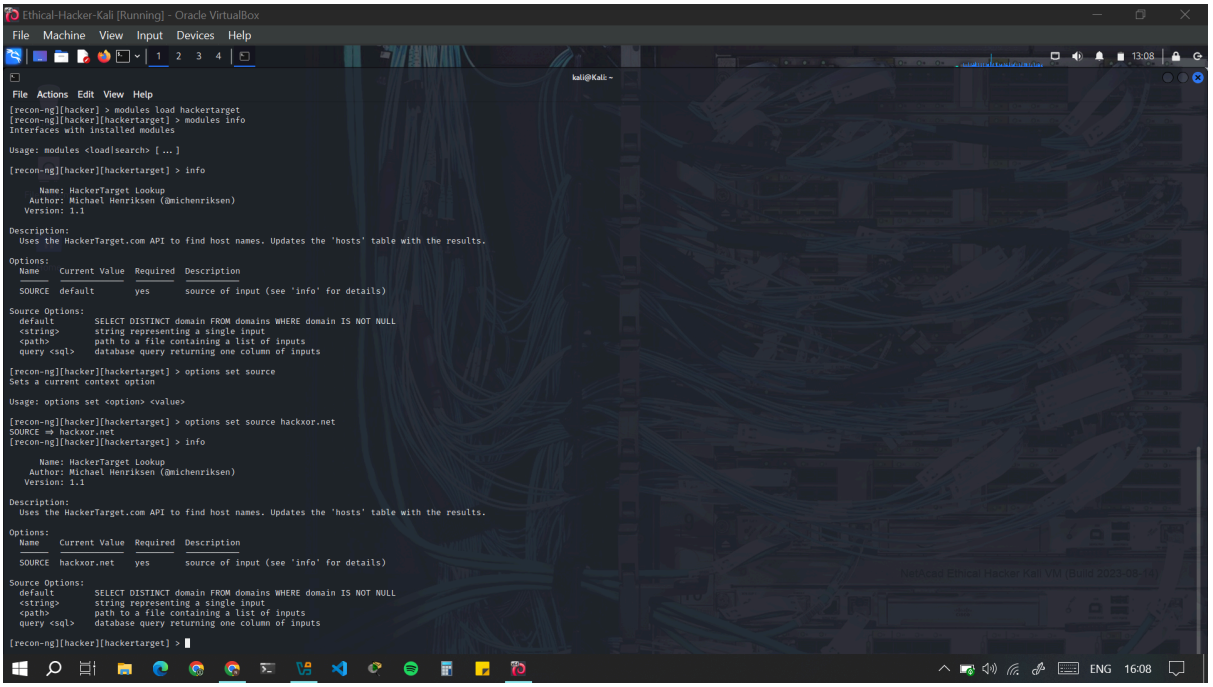


To start working with a module, it must be initialized.Command ‘modules load <modulename>’ to begin working with the module. Note that the prompt changes to reflect the loaded module.

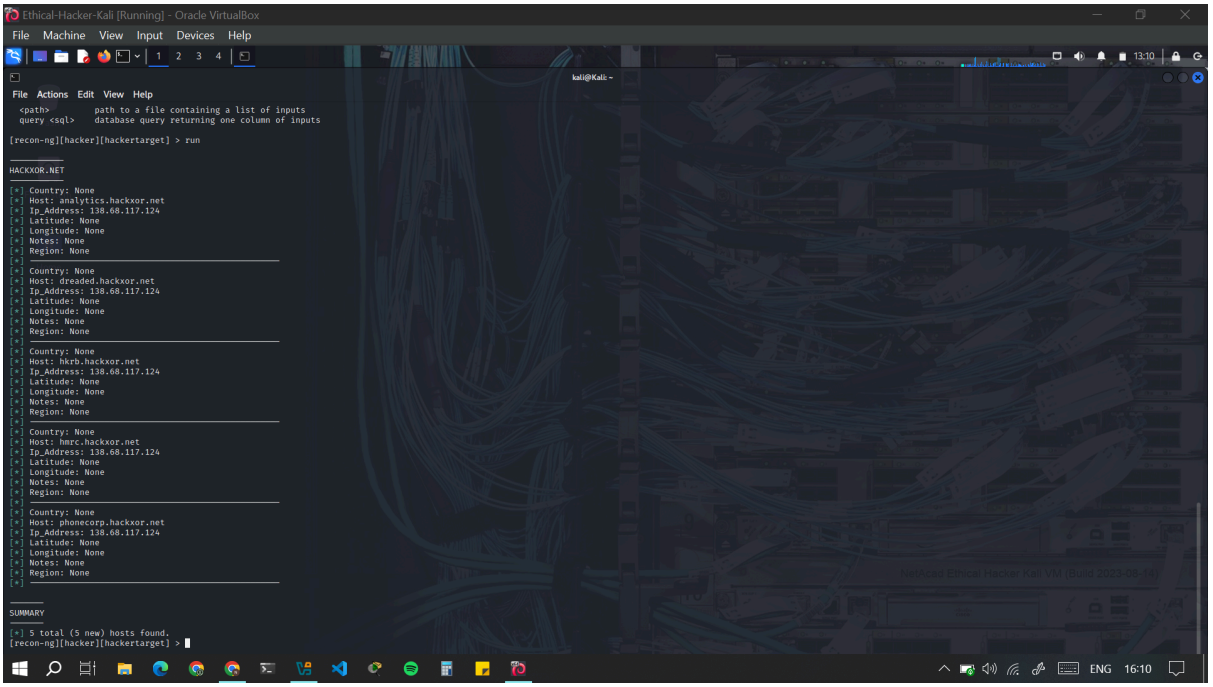
- Each module is its own environment. The developers of recon-ng have taken care to keep the framework consistent, so the same commands are available for each module. However, the options can vary. Command ‘info’ at the module prompt to view important details about the module.
- Modules simplify using Recon-ng because module options can be configured with the module. Instead of needing to specify a long list of options in the command line, execution of the module requires only one simple command.



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. I used the *'options set source'* command to set the only option for this module. I then completed the command by specifying the target as hackxor.net and verified the option setting with the info command.

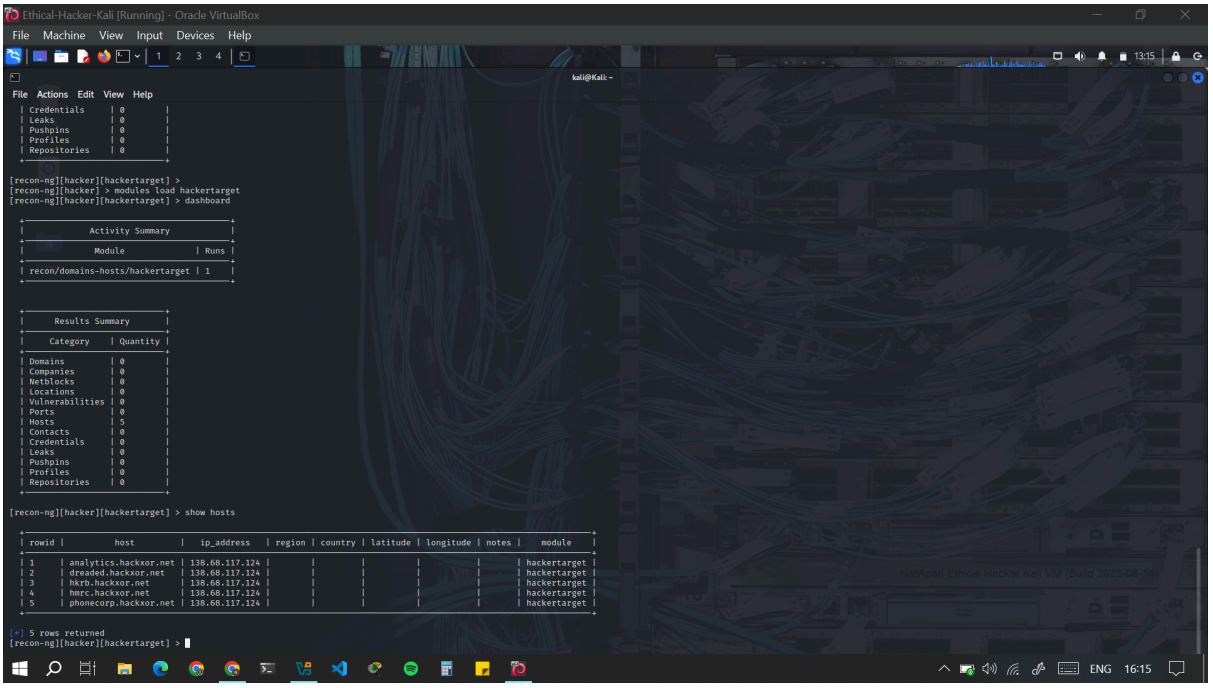


I used the '*run*' command to execute the module



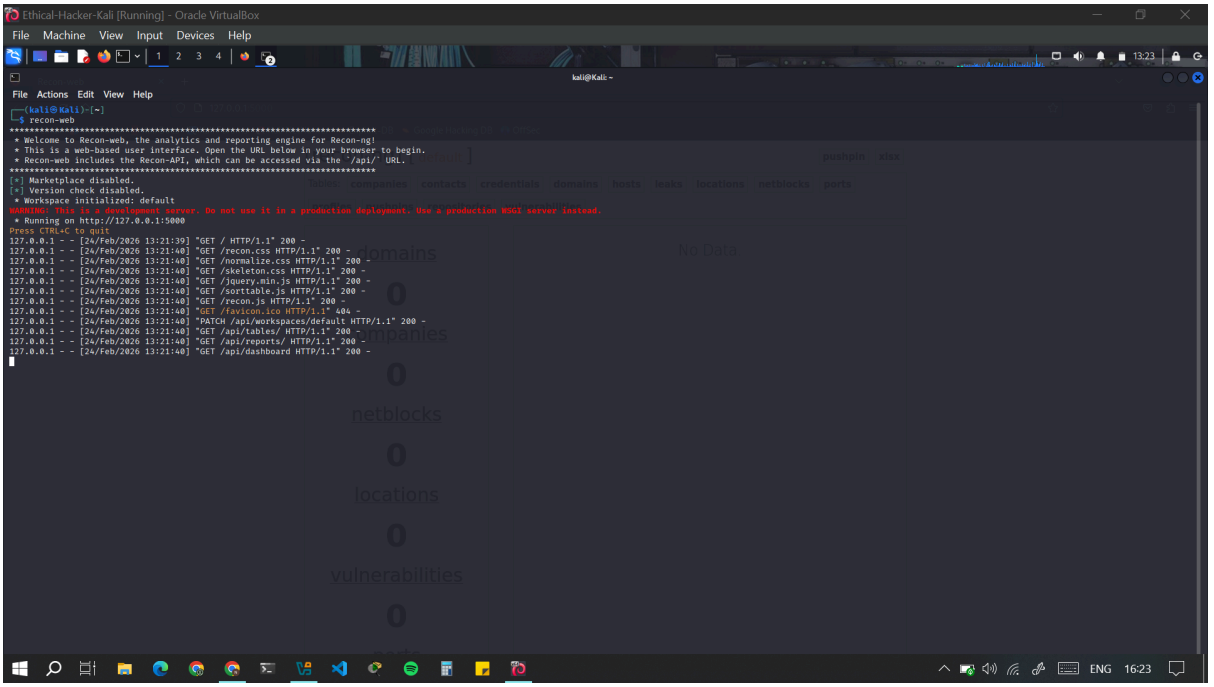
- The output is stored in a database so you can refer to it later. The data that is stored is specific to the workplace in which it was gathered.
- One can also enter the *dashboard* command. This queries the Recon-ng database and provides a summary of the information that has been gathered. It is specific to this workspace.

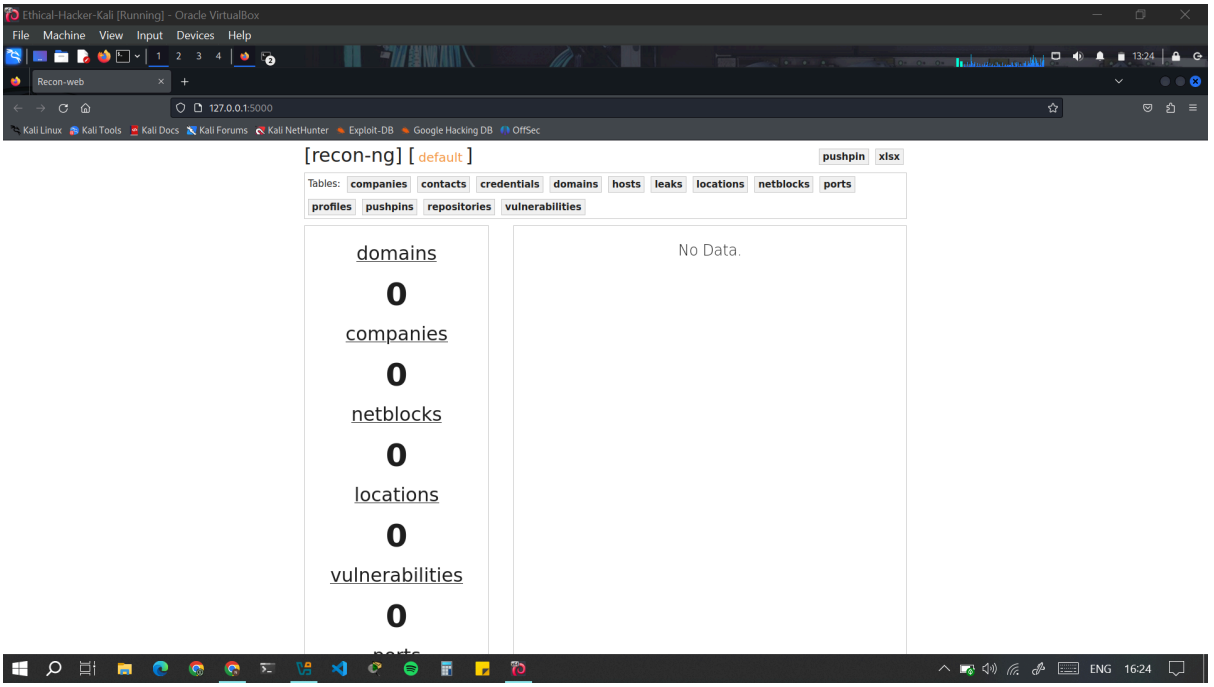
- The show command displays the data for specific categories. I entered the show hosts command to display the list of hosts that were discovered.



Recon-ng also has a web interface that simplifies and improves viewing results that are stored in Recon-ng databases. It also allows easy export of the results tables for reporting purposes. To use it:

1. I opened a new terminal and entered the recon-web command to start the Recon-ng server process.
2. I was then able to access the webpage using the URL information provided in the output.




Ethical-Hacker-Kali [Running] - Oracle VirtualBox
File Machine View Input Devices Help
Recon-web
127.0.0.1:5000
Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec
[recon-ng] [default]
pushpin xlsx
Tables: companies contacts credentials domains hosts leaks locations netblocks ports
profiles pushpins repositories vulnerabilities
domains
0
companies
0
netblocks
0
locations
0
vulnerabilities
0
No Data.
ENG 16:24