Recon-ng

Table of Contents:

1. Introduction

Recon-ng is a full-featured reconnaissance framework written in Python, designed to provide a powerful environment for gathering open-source intelligence (OSINT). It offers a web-based interface and a command-line interface, making it suitable for users with different preferences. Recon-ng automates the process of collecting data from various sources, providing modules for tasks such as domain name resolution, contact information discovery, and social media profiling. This makes it an invaluable tool for penetration testers and security researchers.

2. Tool Details

- **GitHub Repository: Recon-ng on GitHub**
- **Dependencies:**
  - Python3
  - SQLAlchemy
  - Other specific dependencies based on individual modules
- **Use Cases List:**
  - Domain name resolution
  - IP geolocation
  - WHOIS lookups
  - Social media profiling
  - Gathering email addresses and contact information
  - Conducting DNS lookups
- **Version of the Tool: The version used for this report is Recon-ng 5.1.2.**
- **Operating System Used: Kali Linux 2024.1**
- **Interface: Command-Line Interface (CLI)**
- **More Details:**
  - Recon-ng is modular, with each module designed to perform a specific task. This modularity allows users to customize their reconnaissance activities to their specific needs.
  - The tool maintains a database to store the gathered information, enabling users to manage and query the collected data efficiently.
  - It provides interactive help and auto-completion features, making it user-friendly even for those new to OSINT tools.

3. Installation of the Tool

Installation of the Tool

1. **Update the System**:

   ```
   sudo apt update && sudo apt upgrade -y
   ```

2. **Install Dependencies**:

   ```
   sudo apt install -y python3 python3-pip
   ```

3. **Install Recon-ng**:

   ```
   git clone https://github.com/lanmaster53/recon-ng.git
   cd recon-ng
   pip3 install -r REQUIREMENTS
   ```

4. **Run Recon-ng**:

   ```
   cd recon-ng
   python3 recon-ng
   ```

The tool that I've designed doesn't require any kind of permission but the os should be Kali Linux or any other linux distro for the script to work. As soon as the script runs it first checks for the operating system its executed on and if its an unsupported version, it will throw an error and automatically exit. And if the script is run in an a linux environment then it will check whether rustscan is installed, and if not then it will automatically run

'Sudo apt install recon-ng -y" and install recon-ng on the machine.

Disclaimer-Recon-ng uses multiple repositories and dependencies at the same time so they need to be installed to get the respective desired output. Eg-whois etc

4.Execution

To use Recon-ng with the provided Python script, here is a detailed mapping of the program and tool commands:

1. Environment Preparation
   Since the installation of recon-ng is already done in the previous step, the environment is already prepared for execution.
2. User Input
   The user is then prompted for the input of the target ip address or the target web address.eg- 192.168.0.1,www.google.com
3. Execution of code

Then the code is executed and runs recon-ng in the terminal and gives the output

All of these steps are better explained and visualized in the attached video

General Commands

- **help**: Displays the help menu with a list of available commands.

  ```
  recon-ng> help
  ```

- **exit**: Exits the Recon-ng console.

  ```
  recon-ng> exit
  ```

- **version**: Displays the current version of Recon-ng.

  ```
  recon-ng> version
  ```

- **show [command]**: Displays lists of workspaces, modules, options, keys, etc.

  ```
  recon-ng> show workspaces
  recon-ng> show modules
  recon-ng> show options
  ```

- **info [module]**: Provides detailed information about a specific module.

  ```
  recon-ng> info recon/domains-hosts/google_site_web
  ```

- **use [module]**: Loads a specified module

  ```
  recon-ng> use recon/domains-hosts/google_site_web
  ```

Workspace Commands

- **workspace create [name]**: Creates a new workspace

  ```
  recon-ng> workspace create example_workspace
  ```

- **workspace delete [name]**: Deletes an existing workspace.

  ```
  recon-ng> workspace delete example_workspace
  ```

- **workspace select [name]**: Switches to a specified workspace.

  ```
  recon-ng> workspace select example_workspace
  ```

- **workspace list**: Lists all workspaces.

  ```
  recon-ng> workspace list
  ```

Module Commands

- **modules search [keyword]**: Searches for modules matching the specified keyword.
  ```
  recon-ng> modules search whois
  ```

- **modules load [module]**: Loads a specific module.
  ```
  recon-ng> modules load recon/domains-hosts/google_site_web
  ```

- **modules reload [module]**: Reloads a specific module.
  ```
  recon-ng> modules reload recon/domains-hosts/google_site_web
  ```

- **modules info [module]**: Displays detailed information about a module.
  ```
  recon-ng> modules info recon/domains-hosts/google_site_web
  ```

- **modules unload [module]**: Unloads a specific module.
  ```
  recon-ng> modules unload recon/domains-hosts/google_site_web
  ```

Execution Commands

- **run**: Executes the loaded module.
  ```
  recon-ng> run
  ```

- **set [option] [value]**: Sets an option for the loaded module
  ```
  recon-ng> set SOURCE example.com
  ```

- **show options**: Displays the options for the loaded module.
  ```
  recon-ng> show options
  ```

- **scripts list**: Lists all available scripts.
  ```
  recon-ng> scripts list
  ```

Database Commands

- **db insert [table] [field=value]**: Inserts data into the database.

```
recon-ng> db insert hosts ip=192.168.1.1
```

- **db query [SQL query]**: Executes an SQL query on the database.

```
recon-ng> db query select * from hosts
```

- **db delete [table]**: Deletes data from a specific table.

```
recon-ng> db delete hosts
```

- **db schema**: Displays the database schema.

```
recon-ng> db schema
```

5.Output

- **Setup Workspace**: Creates a directory for the workspace and initializes a SQLite database to store reconnaissance data.
- **Write to File**: Helper function to append content to a file.
- **WHOIS Lookup**: Uses the whois library to retrieve WHOIS information for a given domain.
- **Store WHOIS Data**: Stores WHOIS information in the SQLite database and writes it to a file.
- **IP Geolocation**: Uses the ip-api.com API to get geolocation data for an IP address.
- **Store IP Geolocation**: Stores IP geolocation data in the SQLite database and writes it to a file.
- **DNS Lookup**: Uses the dns.resolver library to perform a DNS lookup for a domain.
- **Store DNS Data**: Stores DNS lookup results in the SQLite database and writes them to a file.
- **Run Recon**: Main function that orchestrates the entire reconnaissance process. It calls the above functions and also executes Recon-ng commands using subprocess.
- **Check and Install Recon-ng**: Checks if Recon-ng is installed and installs it if necessary.

6.Automation

In the run_recon function, the script attempts to run Recon-ng commands using subprocesses. However, not all Recon-ng commands can be easily automated, especially those that require user interaction or complex input/output handling. Additionally, some Recon-ng commands may require specific setup or configuration that cannot be easily replicated through automation.

For commands that cannot be automated, you can provide instructions on how to use the tool manually and what output to expect. For example, if a specific Recon-ng module requires manual interaction or generates output that needs further analysis, you can provide steps for running the module manually and interpreting the results.

Also Recon-ng uses multiple marketplace that needs to be downloaded based on the desired output. One cannot guess what the other person wants while executing the script. Also downloading all the marketplace can solve this problem but this will cause the program to take up a lot of time which the user does not want.

Another problem of automating recon-ng is that this program is highly interactive which can cause a lot of problems because we as a developer cannot guess what desired output the user wants .

7.DeepDive Research

Recon-ng is an open-source tool primarily used for web reconnaissance and information gathering during penetration testing or security assessments. It provides a modular framework that allows users to execute various reconnaissance tasks, such as WHOIS lookups, DNS enumeration, IP geolocation, and more, using pre-built modules.

The provided code demonstrates how to automate some of these reconnaissance tasks using the recon-ng tool and other libraries in Python. Here's a breakdown of how the code interacts with recon-ng and performs reconnaissance:

1. **Setup Workspace:** The code creates a workspace directory and initializes a SQLite database to store reconnaissance data. This is where all the gathered information will be stored for later analysis.
2. **WHOIS Lookup: The whois_lookup function uses the whois library to perform a WHOIS lookup for a given domain. It retrieves information such as the registrar, WHOIS server, creation date, and expiration date of the domain.**
3. **IP Geolocation: The ip_geolocation function uses the requests library to query the ip-api.com API for geolocation information of an IP address. It retrieves details such as the country, region, city, latitude, and longitude of the IP address.**
4. **DNS Lookup**: The dns_lookup function uses the dns.resolver library to perform a DNS lookup for a domain and retrieves the A records associated with the domain, including the IP addresses.
5. **Storing Data: The code stores the retrieved WHOIS, IP geolocation, and DNS lookup data into the SQLite database and also writes it to a text file for easy reference.**
6. **Running recon-ng Commands: The run_recon function attempts to run Recon-ng commands using subprocess. It prepares and executes Recon-ng commands for modules like WHOIS lookup, geolocation, DNS enumeration, etc., and prints the output for analysis.**

Features of Recon-ng

1. **Modular Framework**: Recon-ng is highly modular, with a structure that allows users to easily add, remove, and customize modules. Each module is designed to perform a specific task, such as gathering information from a particular source or analyzing data.
2. **Database Integration**: The tool uses a database to store the results of reconnaissance tasks. This allows users to persist data across sessions, run queries on the collected data, and generate reports.
3. **Command-Line Interface**: Recon-ng features a robust command-line interface that is user-friendly and supports various commands to manage modules, interact with the database, and control the workflow of the reconnaissance process.
4. **API Key Management**: Many Recon-ng modules require API keys to interact with external services. Recon-ng provides a centralized way to manage these keys, making it easier to use the tool without hardcoding keys into the scripts.
5. **Report Generation**: Recon-ng can generate detailed reports of the collected data, including HTML and JSON formats. This is useful for documenting findings and sharing them with stakeholders

Common Commands

- **help**: Displays the help menu with a list of available commands.
- **modules search [keyword]**: Searches for modules related to the specified keyword.
- **modules load [module_name]**: Loads a specific module.
- **modules info [module_name]**: Displays information about a specific module.
- **keys add [name] [value]**: Adds an API key.
- **db insert [table] [field=value]**: Inserts data into the database.
- **report [format] [filename]**: Generates a report in the specified format.

Popular Modules

Recon-ng comes with a variety of modules designed for different tasks. Some popular modules include:

- **recon/domains-hosts/bing_domain_web**: Queries Bing for hostnames within a target domain.
- **recon/domains-hosts/google_site_web**: Queries Google for hostnames within a target domain.
- **recon/domains-contacts/whois_pocs**: Collects contact information from WHOIS records.
- **recon/hosts-hosts/resolve**: Resolves IP addresses for a given list of hostnames.
- **recon/netblocks-hosts/shodan_net**: Uses the Shodan API to find hosts within a netblock.

All of these modules have been integrated into this tool to generate all the possible output of recon-ng.

Advanced Features

1. **Custom Modules**: Users can create custom modules by writing Python scripts that adhere to the Recon-ng module framework. This allows for extending the tool's functionality to meet specific needs.
2. **Marketplace**: Recon-ng includes a module marketplace where users can find and share modules created by the community. This encourages collaboration and the sharing of useful reconnaissance techniques.
3. **Scripting and Automation**: Recon-ng supports scripting and automation through its command-line interface and the ability to run scripts that automate common tasks or complex workflows.

Overall, the code provides a framework for automating reconnaissance tasks using recon-ng and other libraries in Python. However, as mentioned earlier, not all Recon-ng commands can be easily automated, especially those that require manual interaction or complex input/output handling.

8.Conclusion

Recon-ng is a versatile and powerful tool for OSINT and reconnaissance tasks. Its modular architecture and extensive database management capabilities make it an essential tool for security professionals. By automating data collection and analysis processes, Recon-ng helps streamline reconnaissance efforts, allowing users to focus on deeper analysis and decision-making.

The provided code demonstrates how to integrate Recon-ng functionalities with additional custom scripts, showcasing the flexibility and extensibility of the tool. This combination enhances the overall reconnaissance process, providing a comprehensive approach to gathering and managing intelligence data.

9.References

- Recon-ng on GitHub
- Recon-ng Wiki
- pywhois
- dnspython