OWASP Nettacker is an automated tool designed to perform vulnerability scanning and information gathering. Implementing OWASP Nettacker involves several steps, including installation, configuration, and running the tool. Here’s a guide to help you achieve this:

**1. Installation**

**Prerequisites**

* **Python 3.x**: Ensure you have Python 3.x installed on your system.
* **Git**: You need Git to clone the OWASP Nettacker repository.

**Steps**

1. Pull Ubuntu’s image i.e. ‘ubuntu:24.04’ on docker desktop
2. Run the image using the following command;
   1. docker run -it ubuntu:24.04
3. Update the apt-get and install pre-requisite packages using the commands;
   1. apt-get update
   2. apt-get install python3
   3. apt-get install pip
   4. apt install python3.12-venv
4. Activate the virtual environment;
   1. Python -m venv venv
   2. source venv/bin/activate
5. Clone the Repository (directly on Linux or import the downloaded code through docker desktop):
   1. git clone <https://github.com/OWASP/Nettacker.git>
   2. git clone <https://github.com/username/Netacker.git>
6. Install Dependencies:
   1. python3 -m pip install -r docs/requirements.txt

Following are the packages not installed through requirements.txt

* 1. Multiprocess
  2. Numpy
  3. Netaddr
  4. Texttable
  5. aiohttp
  6. Sqlalchemy

Install them manually and check if installation is successful using the command; ‘python nettacker.py’

**How to Run:**

Using OWASP Nettacker involves running it through the command line to perform various types of scans, such as network scanning, information gathering, and vulnerability detection. Here's how to use OWASP Nettacker:

Scan option: <https://github.com/OWASP/Nettacker/wiki/Modules#scan-modules>

Port Scan: <https://github.com/OWASP/Nettacker/wiki/Modules#ports-scanned-by-nettacker>

Vulnerabilities options: <https://github.com/OWASP/Nettacker/wiki/Modules#vuln-modules>

Brute Modules: <https://github.com/OWASP/Nettacker/wiki/Modules#brute-modules>

**1. Basic Command Structure**

Nettacker is run through the command line. The basic command structure is:

* python nettacker.py -i <target> -m <module> -o <output\_file> -f <format>

**2. Commonly Used Options**

* -i, --target: Define the target IP address, domain name, or CIDR range.
* -m, --module: Specify the module to use (e.g., port\_scan, http\_brute, subdomain\_scan, etc.).
* -o, --output: Specify the output file name.
* -f, --format: Set the output format (e.g., html, json, xml, etc.).
* -l, --list: Display all available modules.
* -v, --verbose: Increase output verbosity.

**3. Running a Basic Scan**

For example, to scan a target IP with the port\_scan module and save the output in HTML format:

* python nettacker.py -i <https://pulse.10pearls.com/v2/home> -m http\_status\_scan -o report.html

**4. Listing Available Modules**

To list all the available modules that you can use for scanning:

* python nettacker.py -l

### 5. ****Multiple Targets****

You can scan multiple targets by specifying a file containing a list of targets:

* python nettacker.py -i targets.txt -m port\_scan -o multi\_scan\_report.html -f html

**6. Advanced Usage**

* **Brute Force Attack**: To perform a brute-force attack on HTTP services:
  + python nettacker.py -i <https://pulse.10pearls.com/v2/home>-m http\_brute -o brute\_force\_report.html -f html
* **Subdomain Scan**: To discover subdomains of a target domain:
  + python nettacker.py -i https://pulse.10pearls.com/v2/home -m subdomain\_scan-o subdomain\_report.html

**7. Scheduling Scans**

You can schedule scans to run at specific times using cron jobs or task schedulers on your operating system, allowing you to automate regular security assessments.

**8. Output and Reporting**

Review the generated reports in the specified format (HTML, JSON, XML) for detailed results of your scan.

**9. Best Practices**

* Regularly update Nettacker to ensure you're using the latest modules and vulnerability checks.
* Use Nettacker responsibly and only on systems you have permission to test.

**10. Help and Documentation**

For more detailed usage instructions and options, you can access the help documentation by running:

* python nettacker.py -h

This will provide you with all available options and their descriptions for fine-tuning your scans.