

## **CSN 190 – Assignment 6.3 Tutorial**

**Project Name:** OWASP Nettacker

**Project Type:** Automated Penetration Testing / Security Testing Tool

### **Introduction**

This tutorial demonstrates the setup and use of OWASP Nettacker, an open-source automated penetration testing framework developed by OWASP. The objective of this exercise is to successfully initialize the tool on a Windows system, verify correct installation, and safely test it using a localhost target while observing platform limitations.

### **Prerequisites**

- Windows 10 or 11
- Python 3.x installed and added to PATH
- Git for Windows
- Visual Studio Code
- Internet connection

### **Step-by-Step Instructions**

1. Open Visual Studio Code and navigate to the Nettacker project folder.

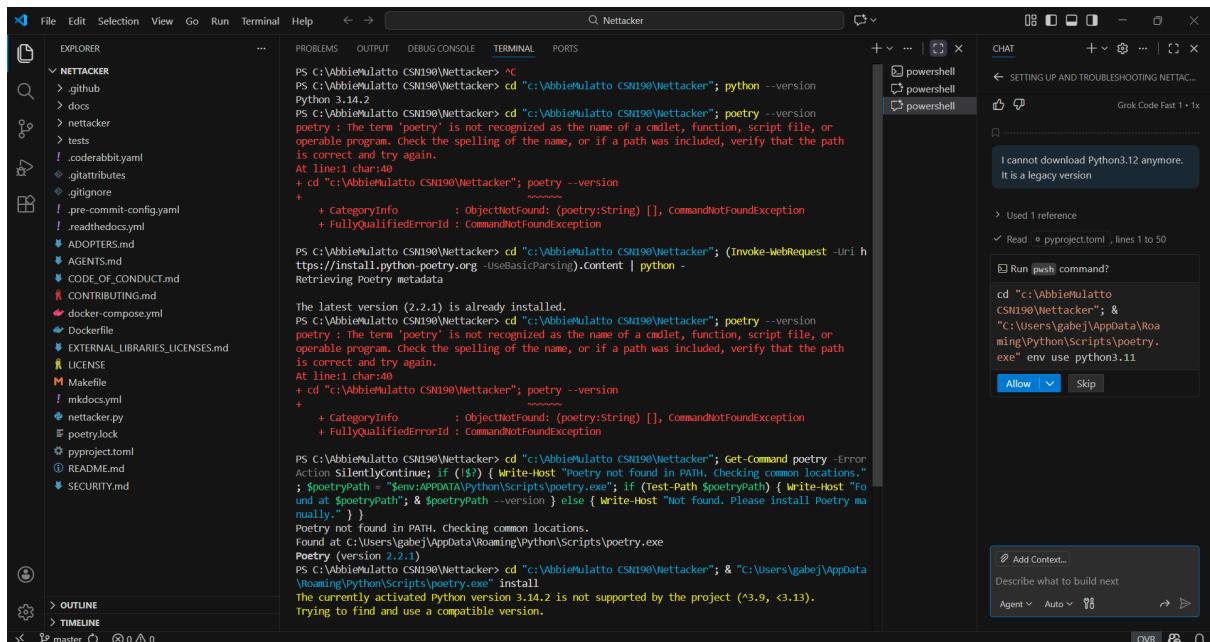
Command:

```
cd C:\AbbieMulatto CSN190\Nettacker
```

2. Verify Python installation.

Command:

```
python --version
```



```
PS C:\AbbieMuliatto CSN190\Nettacker> python --version
Python 3.14.2
PS C:\AbbieMuliatto CSN190\Nettacker> cd "c:\AbbieMuliatto CSN190\Nettacker"; poetry --version
poetry : The term 'poetry' is not recognized as the name of a cmdlet, function, script file, or
operable program. Check the spelling of the name, or if a path was included, verify that the path
is correct and try again.
At line:1 char:40
+ cd "c:\AbbieMuliatto CSN190\Nettacker"; poetry --version
+ ~~~~~
+ CategoryInfo          : ObjectNotFound: (poetry:String) [], CommandNotFoundException
+ FullyQualifiedErrorId : CommandNotFoundException

PS C:\AbbieMuliatto CSN190\Nettacker> cd "c:\AbbieMuliatto CSN190\Nettacker"; Invoke-WebRequest -Uri https://install.python.org -UseBasicParsing).Content | python -
Retrieving Poetry metadata

The latest version (2.2.1) is already installed.

PS C:\AbbieMuliatto CSN190\Nettacker> cd "c:\AbbieMuliatto CSN190\Nettacker"; poetry --version
poetry : The term 'poetry' is not recognized as the name of a cmdlet, function, script file, or
operable program. Check the spelling of the name, or if a path was included, verify that the path
is correct and try again.
At line:1 char:40
+ cd "c:\AbbieMuliatto CSN190\Nettacker"; poetry --version
+ ~~~~~
+ CategoryInfo          : ObjectNotFound: (poetry:String) [], CommandNotFoundException
+ FullyQualifiedErrorId : CommandNotFoundException

PS C:\AbbieMuliatto CSN190\Nettacker> cd "c:\AbbieMuliatto CSN190\Nettacker"; Get-Command poetry -ErrorAction SilentlyContinue; if (!$?) { Write-Host "Poetry not found in PATH. Checking common locations." ; $poetryPath = "$env:APPDATA\Python\Scripts\poetry.exe"; if (Test-Path $poetryPath) { Write-Host "Found at $poetryPath" ; & $poetryPath --version } else { Write-Host "Not found. Please install Poetry manually." } }
Poetry not found in PATH. Checking common locations.
Found at C:\Users\gabej\AppData\Roaming\Python\Scripts\poetry.exe
Poetry (version 2.2.1)
PS C:\AbbieMuliatto CSN190\Nettacker> cd "c:\AbbieMuliatto CSN190\Nettacker"; & "C:\Users\gabej\AppData\Roaming\Python\Scripts\poetry.exe" install
The currently activated Python version 3.14.2 is not supported by the project (>3.9, <3.13).
Trying to find and use a compatible version.
```

(Python was downloaded - Copilot wanted use of a legacy version \*Problem #1\*)

### 3. Install missing dependencies as errors appear.

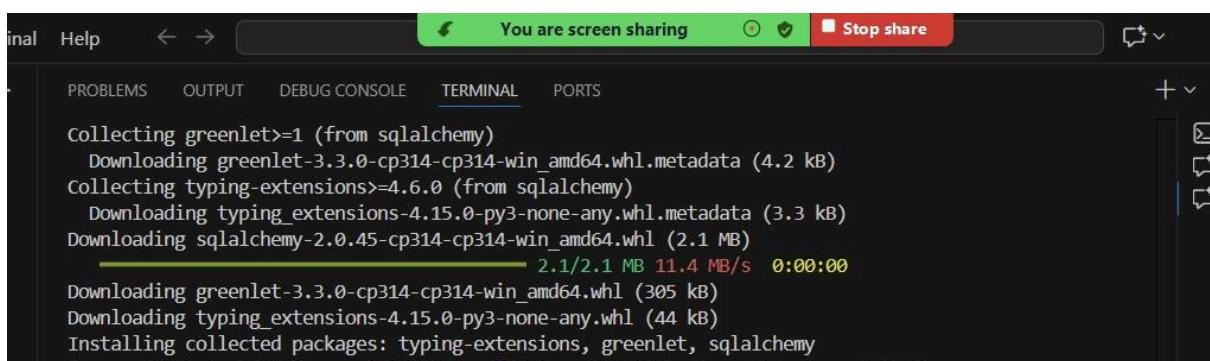
Commands used included:

```
pip install multiprocessing
```

```
pip install pyyaml
```

```
pip install netaddr
```

```
pip install sqlalchemy
```



```
Collecting greenlet>=1 (from sqlalchemy)
  Downloading greenlet-3.3.0-cp314-cp314-win_amd64.whl.metadata (4.2 kB)
Collecting typing-extensions>=4.6.0 (from sqlalchemy)
  Downloading typing_extensions-4.15.0-py3-none-any.whl.metadata (3.3 kB)
  Downloading sqlalchemy-2.0.45-cp314-cp314-win_amd64.whl (2.1 MB)
  2.1/2.1 MB 11.4 MB/s 0:00:00
  Downloading greenlet-3.3.0-cp314-cp314-win_amd64.whl (305 kB)
  Downloading typing_extensions-4.15.0-py3-none-any.whl (44 kB)
  Installing collected packages: typing-extensions, greenlet, sqlalchemy
  
```

(Installation of missing dependencies - \*Problem #2\*)

### 4. Run Nettacker help command.

### Command:

```
python nettacker.py --help
```

A screenshot of a terminal window titled 'AbbieMulatto CSN190\Nettacker>'. The command 'python nettacker.py --help' is run, displaying the program's logo and version information. The logo consists of a stylized tree or character made of line art, with the text 'Version 0.4.0' and 'QUIN' below it. The word 'OWASP' is visible on the left side of the screen.

5. Attempt a safe localhost scan.

### Command:

```
python nettacker.py -i 127.0.0.1
```

The screenshot shows the Microsoft Visual Studio Code interface with three terminal windows open. The left sidebar displays the project structure for 'NETTACKER' with files like .github, .venv, docs, nettacker, tests, .coderabbit.yaml, .gitattributes, .gitignore, .pre-commit-config.yaml, .readthedocs.yaml, .ADOPTERS.md, AGENTS.md, CODE\_OF\_CONDUCT.md, CONTRIBUTING.md, docker-compose.yaml, Dockerfile, and EXTERNAL\_LIBRARIES\_LICENSES.md. The bottom status bar indicates the code is in 'master' branch with 0 changes.

**Terminal 1:**

```
[2025-12-12 18:34:41][X] Unfortunately, this version of the software can run on Linux/darwin
(.venv) PS C:\Abbie\Julatto\CS190\Nettacker> python nettacker.py -i 192.168.1.13
```

**Terminal 2:**

```
[2025-12-12 18:35:15][X] Unfortunately, this version of the software can run on Linux/darwin
(.venv) PS C:\Abbie\Julatto\CS190\Nettacker> python nettacker.py -i 127.0.0.1
```

**Terminal 3:**

```
[2025-12-12 18:35:38][X] Unfortunately, this version of the software can run on Linux/darwin
(.venv) PS C:\Abbie\Julatto\CS190\Nettacker>
```

**Right Panel:**

- RECENT SESSIONS:
  - Setting up and troubleshooting Nettacker ... (1 file + 2)
  - Setting up OWASP Nettacker on Windows
- CHAT: Local + 1 hr
- Build with Agent: AI responses may be inaccurate.

(Attempt at scanning local host - Attempt failed due to environment limitations - Expected Result)

## Expected Results

OWASP Nettacker successfully launches, displays its ASCII banner and version number, and reports that active scanning is supported only on Linux and macOS systems. This confirms that the tool is installed correctly and functioning as designed on Windows.

## Troubleshooting

During setup, several missing Python modules caused errors. These were resolved by carefully reading error messages and installing the required dependencies using pip. The platform limitation message is expected behavior and does not indicate failure.

## Conclusion

This exercise provided hands-on experience working with a real-world cybersecurity tool. I learned how to troubleshoot dependency issues, understand platform limitations, and ethically initialize penetration testing software. Future work could involve running OWASP Nettacker on a Linux-based environment for full functionality.