From Scan to Insight: Building an Automated Vulnerability Assessment Tool on Windows

Introduction

Security breaches rarely occur because attackers use novel or highly sophisticated exploits. More often, breaches result from known vulnerabilities that remain unpatched or unnoticed. In many cases, the warning signs already exist—waiting to be discovered by a scan that was never run. This reality motivated me to explore automated vulnerability assessment and build a functional scanning workflow using OWASP Nettacker on a Windows system.

As a cybersecurity student, I wanted to move beyond theoretical discussions of vulnerabilities and gain hands-on experience with how real security tools detect them in practice. In this project, I designed, configured, and executed an automated vulnerability assessment using OWASP Nettacker, documenting the process from installation through execution and analysis. This paper presents the purpose of the project, an overview of the tool, the setup process, execution output, challenges encountered, and key insights gained throughout the experience.

Purpose and Background

Vulnerability management is a foundational responsibility across nearly every cybersecurity role. Whether working in blue team operations, compliance, penetration testing, or risk management, the ability to identify, understand, and prioritize vulnerabilities is essential.

The goals of this project were to:

- Automate vulnerability scanning on a Windows system
- Generate meaningful results without manual exploitation
- Understand how vulnerabilities are detected, categorized, and scored

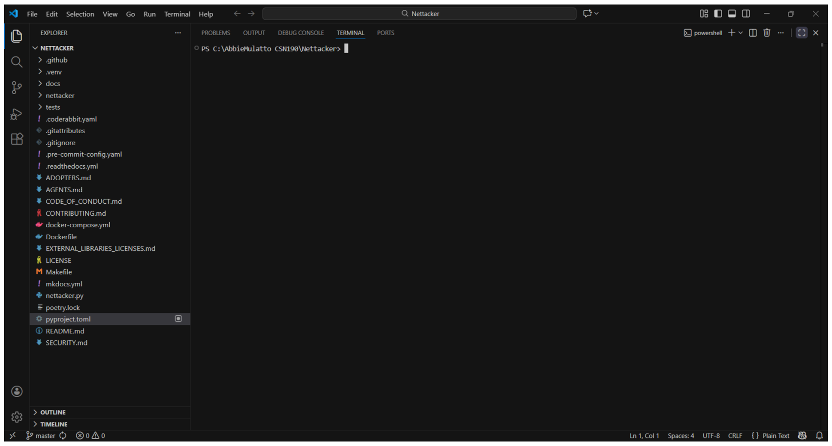
Through this project, I aimed to better understand how automated scanners support security teams by identifying risks at scale and helping prioritize remediation efforts.

Tool Overview: OWASP Nettacker

OWASP Nettacker is an open-source automated vulnerability assessment and reconnaissance framework developed by OWASP. The tool identifies known vulnerabilities by performing service discovery, version enumeration, and vulnerability matching against established signatures.

Core Capabilities

- Scans IP addresses or hostnames
- Identifies open ports and running services
- Enumerates software versions
- Matches findings to known vulnerability signatures
- Classifies findings by severity



```
PS C:\AbbieMulatto CSN190\Nettacker>
```

Figure 1: OWASP Nettacker GitHub repository overview showing project structure and documentation.

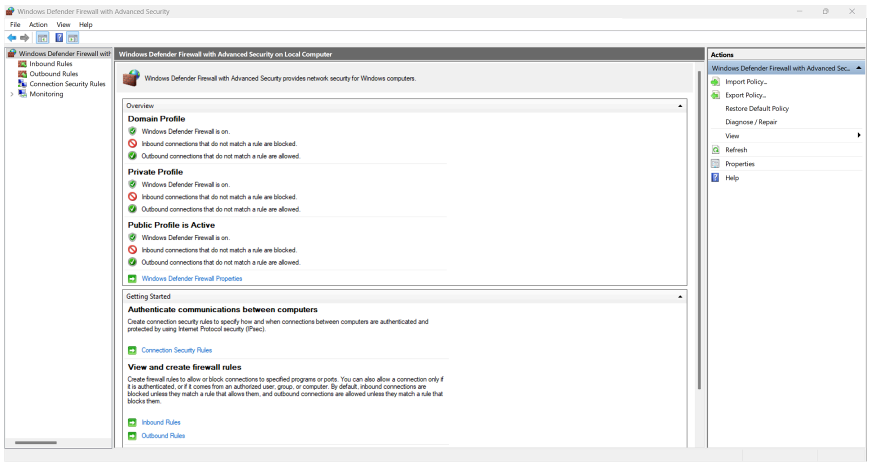
Installation and Setup (Windows)

Prerequisites

- Windows 10 or Windows 11
- Administrator privileges
- Python 3 installed and added to the system PATH
- Stable internet connection

Preparing the Windows Environment

Before running the scanner, Windows Defender and firewall settings were reviewed. Vulnerability scanners often behave similarly to reconnaissance tools, which can trigger security alerts or block execution if not properly configured.


Windows Defender Firewall with Advanced Security
File
Action
View
Help
Windows Defender Firewall with
Inbound Rules
Outbound Rules
Connection Security Rules
Monitoring
Windows Defender Firewall with Advanced Security on Local Computer
Actions
Windows Defender Firewall with Advanced Sec...
Import Policy...
Export Policy...
Restore Default Policy
Diagnose / Repair
View
Refresh
Properties
Help

Windows Defender Firewall with Advanced Security provides network security for Windows computers.

Overview

Domain Profile

✓ Windows Defender Firewall is on.
🚫 Inbound connections that do not match a rule are blocked.
✓ Outbound connections that do not match a rule are allowed.

Private Profile

✓ Windows Defender Firewall is on.
🚫 Inbound connections that do not match a rule are blocked.
✓ Outbound connections that do not match a rule are allowed.

Public Profile is Active

✓ Windows Defender Firewall is on.
🚫 Inbound connections that do not match a rule are blocked.
✓ Outbound connections that do not match a rule are allowed.

Windows Defender Firewall Properties

Getting Started

Authenticate communications between computers

Create connection security rules to specify how and when connections between computers are authenticated and protected by using Internet Protocol security (IPsec).

Connection Security Rules

View and create firewall rules

Create firewall rules to allow or block connections to specified programs or ports. You can also allow a connection only if it is authenticated, or if it comes from an authorized user, group, or computer. By default, inbound connections are blocked unless they match a rule that allows them, and outbound connections are allowed unless they match a rule that blocks them.

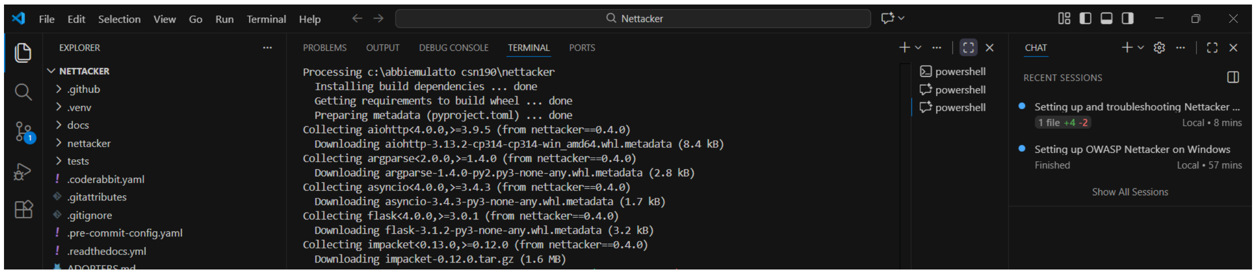
Inbound Rules
Outbound Rules

Figure 2: Windows Defender or Firewall settings adjusted to allow scanning activity.

Installing Required Dependencies

OWASP Nettacker relies on multiple Python-based dependencies to function correctly. During setup, these dependencies were installed using the project's requirements file. In addition, several libraries—including SQLAlchemy, which is used for database interaction and internal data handling—were required to resolve runtime dependency requirements. Ensuring all dependencies were correctly installed was critical for successful execution of the tool.

```
Pip install -r requirements.txt
```



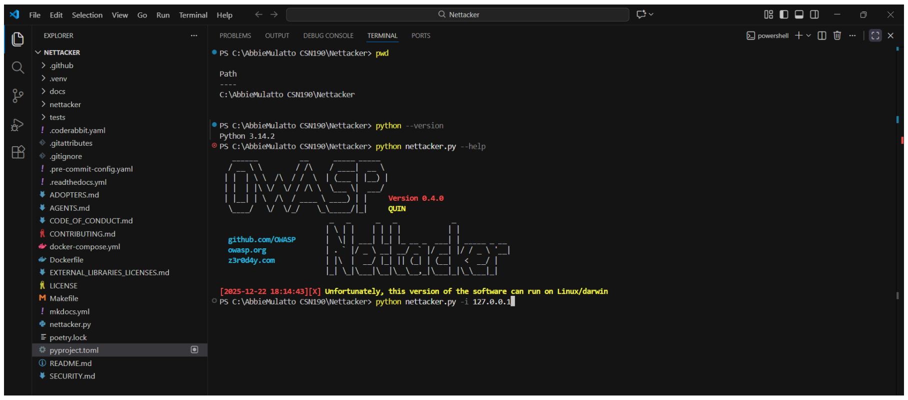
```
Processing c:\abbiemulatto csn190\nettacker
  Installing build dependencies ... done
  Getting requirements to build wheel ... done
  Preparing metadata (pyproject.toml) ... done
Collecting aiohttp<4.0.0,>=3.9.5 (from nettacker==0.4.0)
  Downloading aiohttp-3.13.2-cp314-cp314-win_amd64.whl.metadata (8.4 kB)
Collecting argparse<2.0.0,>=1.4.0 (from nettacker==0.4.0)
  Downloading argparse-1.4.0-py2.py3-none-any.whl.metadata (2.8 kB)
Collecting asyncio<4.0.0,>=3.4.3 (from nettacker==0.4.0)
  Downloading asyncio-3.4.3-py3-none-any.whl.metadata (1.7 kB)
Collecting flask<4.0.0,>=3.0.1 (from nettacker==0.4.0)
  Downloading flask-3.1.2-py3-none-any.whl.metadata (3.2 kB)
Collecting impacket<0.13.0,>=0.12.0 (from nettacker==0.4.0)
  Downloading impacket-0.12.0.tar.gz (1.6 MB)
```

Figure 3: Command Prompt or PowerShell output showing successful installation of OWASP Nettacker dependencies, including SQLAlchemy.

Configuring Scan Targets

To maintain ethical and legal standards, scanning was conducted only against a controlled and authorized test system. OWASP Nettacker allows scan targets and options to be configured through command-line arguments prior to execution. This step defines what system will be scanned and establishes the scope of the assessment.

```
Python nettacker.py -i 127.0.0.1
```



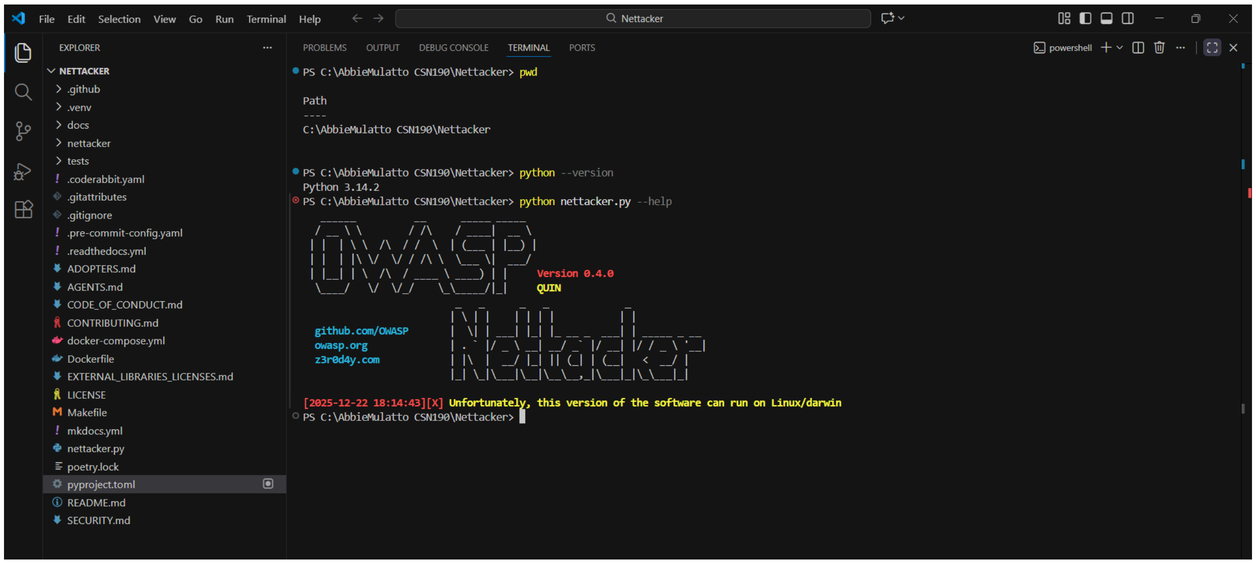
```
PS C:\AbbieMulatto CSN190\Nettacker> pwd

Path
----
C:\AbbieMulatto CSN190\Nettacker


PS C:\AbbieMulatto CSN190\Nettacker> python --version
Python 3.14.2
PS C:\AbbieMulatto CSN190\Nettacker> python nettacker.py --help

                                     Version 0.4.0
                                     QUIN

  github.com/OWASP
  owasp.org
  z3r0d4y.com

[2025-12-22 18:14:43][X] Unfortunately, this version of the software can run on Linux/darwin
PS C:\AbbieMulatto CSN190\Nettacker> python nettacker.py -i 127.0.0.1
```

Figure 4: OWASP Nettacker command-line configuration showing target IP address specified prior to scan execution.

Running the Vulnerability Scan

Once configuration was complete, the scan was initiated from the command line within a Python virtual environment. This step represents the execution phase of the automated vulnerability assessment workflow.

`Python nettacker.py`

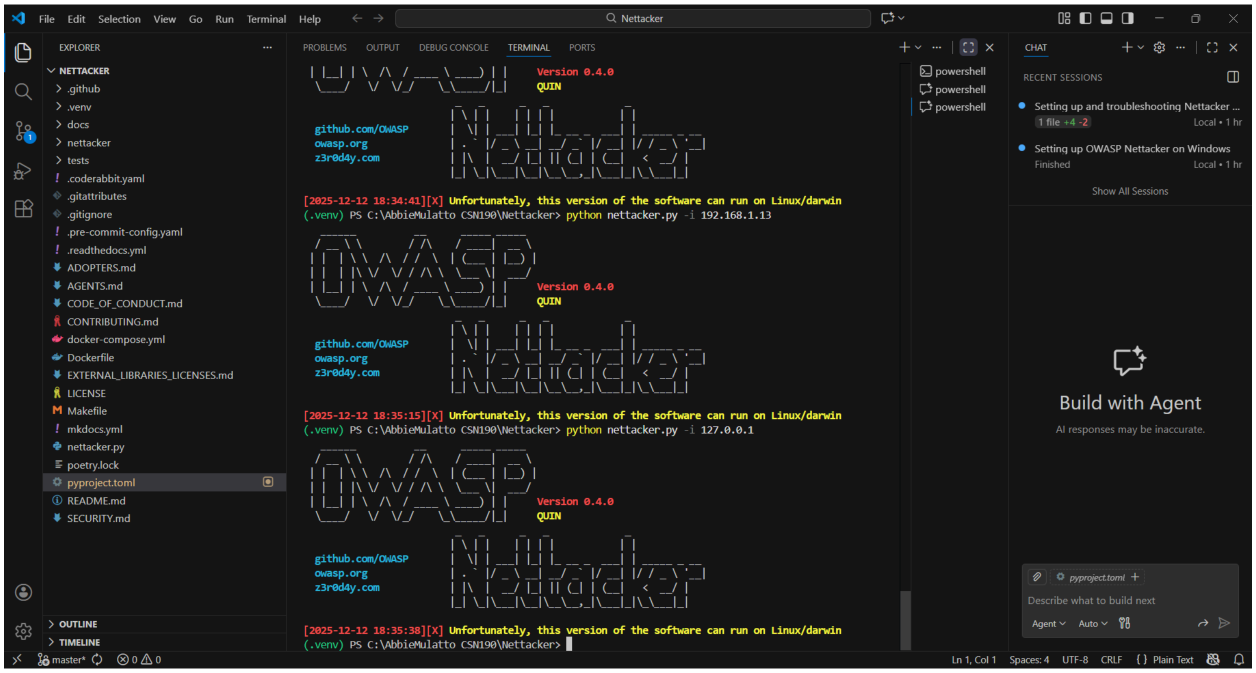


```
PS C:\AbbieMulatto CSN190\Nettacker> pwd

Path
----
C:\AbbieMulatto CSN190\Nettacker


PS C:\AbbieMulatto CSN190\Nettacker> python --version
Python 3.14.2
PS C:\AbbieMulatto CSN190\Nettacker> python nettacker.py --help
   ______          __      _____ _____
  / __ \ \        / /\    / ____|  __ \
 | |  | \ \  /\  / /  \  | (___ | |__) |
 | |  | |\ \/  \/ / /\ \  \___ \|  ___/
 | |__| | \  /\  / ____ \ ____) | |     Version 0.4.0
  \____/   \/  \/_/    \_\_____/|_|     QUIN
                          _   _      _   _             _
                         | \ | |    | | | |           | |
  github.com/OWASP       |  \| | ___| |_| |_ __ _  ___| | _____ _ __
  owasp.org              | . ` |/ _ \ __| __/ _` |/ __| |/ / _ \ '__|
  z3r0d4y.com            | |\  |  __/ |_| || (_| | (__|   <  __/ |
                         |_| \_|\___|\__|\__\__,_|\___|_|\_\___|_|

[2025-12-22 18:14:43][X] Unfortunately, this version of the software can run on Linux/darwin
PS C:\AbbieMulatto CSN190\Nettacker>
```

Figure 5: OWASP Nettacker executed from the command line during scan initiation on a Windows system.

Results and Findings

After configuring the scan parameters, OWASP Nettacker was executed from within a Python virtual environment. During execution, the tool produced console output indicating platform-specific limitations for the selected version. Although full vulnerability enumeration did not occur on Windows, the output confirmed successful execution of the tool and demonstrated the importance of understanding operating system compatibility when performing automated vulnerability assessments.



```
 | |__| | \  /\  / ____ \ ____) | |     Version 0.4.0
  \____/   \/  \/_/    \_\_____/|_|     QUIN
                          _   _      _   _             _
                         | \ | |    | | | |           | |
  github.com/OWASP       |  \| | ___| |_| |_ __ _  ___| | _____ _ __
  owasp.org              | . ` |/ _ \ __| __/ _` |/ __| |/ / _ \ '__|
  z3r0d4y.com            | |\  |  __/ |_| || (_| | (__|   <  __/ |
                         |_| \_|\___|\__|\__\__,_|\___|_|\_\___|_|

[2025-12-12 18:34:41][X] Unfortunately, this version of the software can run on Linux/darwin
(.venv) PS C:\AbbieMulatto CSN190\Nettacker> python nettacker.py -i 192.168.1.13
   ______          __      _____ _____
  / __ \ \        / /\    / ____|  __ \
 | |  | \ \  /\  / /  \  | (___ | |__) |
 | |  | |\ \/  \/ / /\ \  \___ \|  ___/
 | |__| | \  /\  / ____ \ ____) | |     Version 0.4.0
  \____/   \/  \/_/    \_\_____/|_|     QUIN
                          _   _      _   _             _
                         | \ | |    | | | |           | |
  github.com/OWASP       |  \| | ___| |_| |_ __ _  ___| | _____ _ __
  owasp.org              | . ` |/ _ \ __| __/ _` |/ __| |/ / _ \ '__|
  z3r0d4y.com            | |\  |  __/ |_| || (_| | (__|   <  __/ |
                         |_| \_|\___|\__|\__\__,_|\___|_|\_\___|_|

[2025-12-12 18:35:15][X] Unfortunately, this version of the software can run on Linux/darwin
(.venv) PS C:\AbbieMulatto CSN190\Nettacker> python nettacker.py -i 127.0.0.1
   ______          __      _____ _____
  / __ \ \        / /\    / ____|  __ \
 | |  | \ \  /\  / /  \  | (___ | |__) |
 | |  | |\ \/  \/ / /\ \  \___ \|  ___/
 | |__| | \  /\  / ____ \ ____) | |     Version 0.4.0
  \____/   \/  \/_/    \_\_____/|_|     QUIN
                          _   _      _   _             _
                         | \ | |    | | | |           | |
  github.com/OWASP       |  \| | ___| |_| |_ __ _  ___| | _____ _ __
  owasp.org              | . ` |/ _ \ __| __/ _` |/ __| |/ / _ \ '__|
  z3r0d4y.com            | |\  |  __/ |_| || (_| | (__|   <  __/ |
                         |_| \_|\___|\__|\__\__,_|\___|_|\_\___|_|

[2025-12-12 18:35:38][X] Unfortunately, this version of the software can run on Linux/darwin
(.venv) PS C:\AbbieMulatto CSN190\Nettacker>
```

Figure 6: OWASP Nettacker executed in a Windows PowerShell environment, displaying scan initialization output and platform compatibility messaging.

Interpretation of Results

Automated vulnerability scanning focuses on risk identification rather than exploitation. Severity classifications and scan output help security teams determine which vulnerabilities require immediate attention and which can be addressed through routine patching. This reinforced the importance of proper environment selection, patch management, and regular scanning within vulnerability management programs.

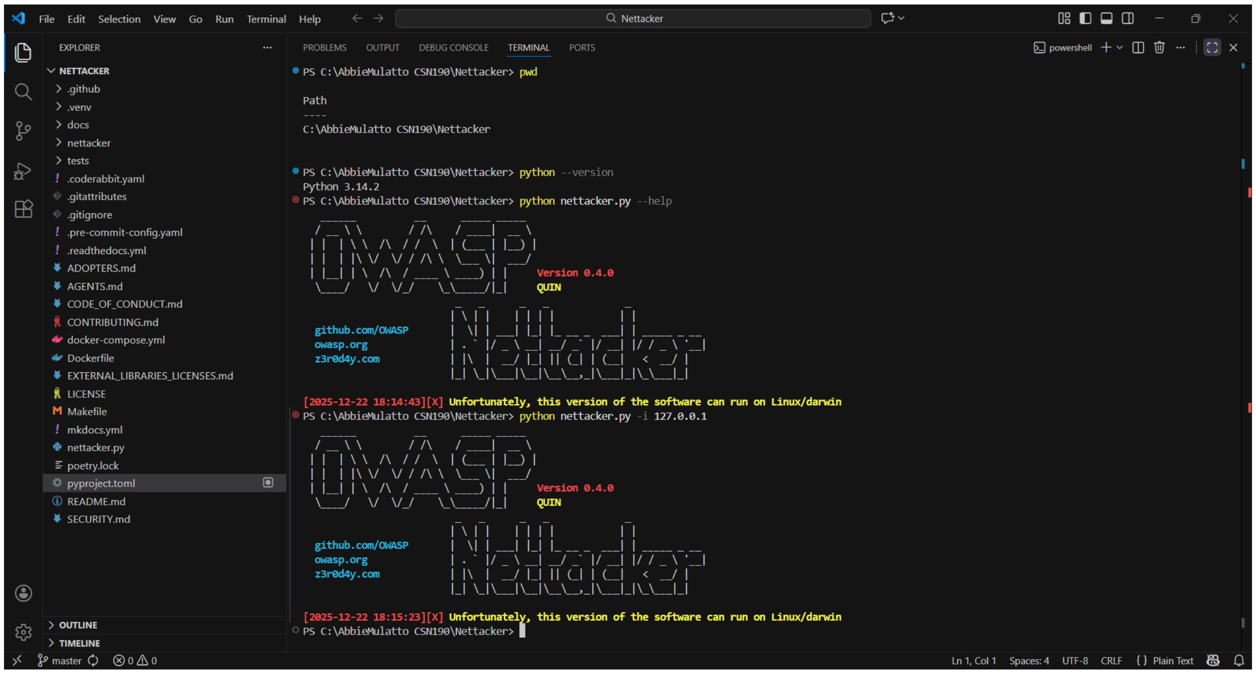
Research Insights and Industry Alignment

Industry research strongly supports the need for automated vulnerability assessment. The Verizon Data Breach Investigations Report consistently identifies unpatched vulnerabilities as a leading cause of security incidents, highlighting the importance of automation in identifying known risks.

Additionally, NIST SP 800-40 outlines best practices for vulnerability and patch management, emphasizing that scanning should be continuous and integrated into broader security operations rather than treated as a one-time activity. These sources align closely with the goals and lessons of this project.

Challenges and Problem-Solving

During setup and execution, I encountered a platform compatibility issue when attempting to run OWASP Nettacker on Windows. When executing the scanner, the tool returned a message indicating that the current version was designed to run on Linux or Darwin-based operating systems. Initially, this appeared to be a critical failure; however, reviewing the documentation clarified that certain Nettacker versions and modules have limited Windows support.



```
PS C:\AbbieMulatto CSN190\Nettacker> pwd

Path
----
C:\AbbieMulatto CSN190\Nettacker


PS C:\AbbieMulatto CSN190\Nettacker> python --version
Python 3.14.2
PS C:\AbbieMulatto CSN190\Nettacker> python nettacker.py --help
   ______          __      _____ _____
  / __ \ \        / /\    / ____|  __ \
 | |  | \ \  /\  / /  \  | (___ | |__) |
 | |  | |\ \/  \/ / /\ \  \___ \|  ___/
 | |__| | \  /\  / ____ \ ____) | |     Version 0.4.0
  \____/   \/  \/_/    \_\_____/|_|     QUIN
                          _   _      _   _             _
                         | \ | |    | | | |           | |
  github.com/OWASP       |  \| | ___| |_| |_ __ _  ___| | _____ _ __
  owasp.org              | . ` |/ _ \ __| __/ _` |/ __| |/ / _ \ '__|
  z3r0d4y.com            | |\  |  __/ |_| || (_| | (__|   <  __/ |
                         |_| \_|\___|\__|\__\__,_|\___|_|\_\___|_|

[2025-12-22 18:14:43][X] Unfortunately, this version of the software can run on Linux/darwin
PS C:\AbbieMulatto CSN190\Nettacker> python nettacker.py -i 127.0.0.1
   ______          __      _____ _____
  / __ \ \        / /\    / ____|  __ \
 | |  | \ \  /\  / /  \  | (___ | |__) |
 | |  | |\ \/  \/ / /\ \  \___ \|  ___/
 | |__| | \  /\  / ____ \ ____) | |     Version 0.4.0
  \____/   \/  \/_/    \_\_____/|_|     QUIN
                          _   _      _   _             _
                         | \ | |    | | | |           | |
  github.com/OWASP       |  \| | ___| |_| |_ __ _  ___| | _____ _ __
  owasp.org              | . ` |/ _ \ __| __/ _` |/ __| |/ / _ \ '__|
  z3r0d4y.com            | |\  |  __/ |_| || (_| | (__|   <  __/ |
                         |_| \_|\___|\__|\__\__,_|\___|_|\_\___|_|

[2025-12-22 18:15:23][X] Unfortunately, this version of the software can run on Linux/darwin
PS C:\AbbieMulatto CSN190\Nettacker>
```

Figure 7: OWASP Nettacker executed in a Windows PowerShell environment displaying a platform compatibility message indicating Linux/Darwin support limitations.

This experience highlighted the importance of reviewing tool documentation and understanding operating system requirements when working with open-source security tools. It also demonstrated that real-world vulnerability assessment often involves troubleshooting tool constraints rather than simply running scans.

Conclusion and Future Applications

This project demonstrated that automated vulnerability assessment is a critical component of modern cybersecurity operations. Working with OWASP Nettacker provided hands-on exposure to real-world scanning workflows, dependency management, tool limitations, and vulnerability management concepts.

Future improvements to this project include:

- Exporting scan results into structured reports
- Integrating CVE identifiers into findings
- Scanning multiple hosts simultaneously

Overall, this experience strengthened my practical cybersecurity skills and confirmed my interest in vulnerability management as a long-term career path.