

# Abstract

In the rapidly evolving domain of cybersecurity, efficient and integrated tools for active reconnaissance are essential for both professionals and enthusiasts to assess and enhance network security. CyberNova emerges as a comprehensive, versatile toolkit designed to streamline the process of information gathering and reconnaissance on IP addresses and domains. This Python-based tool harnesses the capabilities of basic system tools, default kernel utilities, and specialized third-party libraries such as `python-dns`, `python-nmap`, and `python-whois`, offering a unified interface for a range of reconnaissance activities.

Unlike traditional command-line interface (CLI) tools, CyberNova is implemented as a Python package that can be seamlessly integrated into existing codebases, providing flexibility and ease of use. Available on PyPI, it can be conveniently installed using the `pip` command, making it accessible for immediate use in various security-related tasks. By combining multiple reconnaissance tools into a single package, CyberNova facilitates a more comprehensive understanding of network infrastructures, aiding in the proactive identification and mitigation of security vulnerabilities..

# Contents

<b>Abstract</b>	<b>i</b>
<b>List of Figures</b>	<b>iv</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Overview . . . . .	1
1.2 Motivation . . . . .	2
1.3 Problem Statement and Objectives . . . . .	2
1.4 Organization of the report . . . . .	2
<b>2 Literature Survey</b>	<b>3</b>
2.1 Survey of Existing System . . . . .	3
2.2 Limitations of Existing System or Research Gap . . . . .	4
<b>3 Proposed System</b>	<b>5</b>
3.1 Problem Statement . . . . .	5
3.2 Proposed Methodology/Techniques . . . . .	5
3.3 System Design . . . . .	6
3.4 Details of Hardware/Software Requirement . . . . .	7
<b>4 Results and Discussion</b>	<b>8</b>
4.1 Implementation Details . . . . .	8
4.2 Result Analysis . . . . .	8
4.3 Contribution . . . . .	9
4.4 Result Images . . . . .	10

<b>5 Conclusion and Further Work</b>	<b>13</b>
5.1 Conclusion . . . . .	13
5.2 Future Work . . . . .	14
<b>References</b>	<b>15</b>
<b>A Weekly Progress Report</b>	<b>16</b>
<b>B Plagiarism Report</b>	<b>17</b>
<b>C Publication Details / Copyright / Project Competitions</b>	<b>18</b>
C.1 Copyright . . . . .	18
<b>Acknowledgement</b>	<b>19</b>

# List of Figures

4.1	PYPL Publication . . . . .	10
4.2	GitHub Repository . . . . .	11
4.3	Package Installation . . . . .	12
A.1	Weekly Progress Report . . . . .	16
B.1	Plagiarism Report . . . . .	17

# Chapter 1

## Introduction

Cybersecurity is a critical field that demands continuous innovation and effective tools to manage the complex landscape of threats and vulnerabilities. CyberNova is designed to meet these needs by providing a comprehensive toolkit for active reconnaissance, which is fundamental to securing network infrastructure. This Python-based tool simplifies and enhances the process of gathering intelligence on IP addresses and domains, making it an invaluable asset for security analysts, network administrators, and cybersecurity enthusiasts.

### 1.1 Overview

CyberNova integrates a variety of reconnaissance tools into a single package, leveraging both basic and advanced functionalities. The toolkit is structured into several components:

**Basic System Tools:** Utilizes everyday system commands and functionalities for initial data collection.

**Kernel Tools:** Harnesses more in-depth system capabilities typically reserved for deeper network analysis.

**Third-party Libraries:** Incorporates specialized libraries such as python-dns for DNS lookups, python-nmap for network mapping, and python-whois for domain ownership details, among others.

This integration not only ensures a broad spectrum of data collection and analysis but also streamlines workflows by reducing the need to switch between multiple tools during a reconnaissance mission.

## **1.2 Motivation**

The motivation behind CyberNova is driven by the need for a more integrated and user-friendly approach to active reconnaissance. Current tools often operate in isolation, requiring users to manually combine data from different sources, which can be time-consuming and error-prone. CyberNova addresses this gap by offering a unified interface that automates and simplifies the reconnaissance process, thereby increasing efficiency and reducing the likelihood of oversight.

## **1.3 Problem Statement and Objectives**

Despite the availability of numerous tools for network reconnaissance, there remains a significant challenge in the coordination and integration of these tools. Many existing solutions are designed for specific tasks and do not support seamless interaction with other utilities. This fragmentation can lead to inefficiencies, higher costs in time and resources, and potential gaps in the security assessments. CyberNova aims to resolve these issues by providing a cohesive toolkit that supports comprehensive information gathering and enhances the overall effectiveness of network reconnaissance activities.

## **1.4 Organization of the report**

The report is organised as follows: The Chapter 2 reviews the literature. Chapter 3 focuses on defining the system's issue. That includes problem categorization, proposed technologies, device architecture, and hardware/software requirements. On the other hand, Chapter 5 describes the inference and future work on the technique to be utilized as a more improved model.

# Chapter 2

## Literature Survey

### Key Technologies and Methodologies:

**1.Machine Learning for Reconnaissance:**Utilizing AI and machine learning, notably Python, enhances the automation and efficiency of reconnaissance activities during network penetration testing. This approach integrates complex data analysis within reconnaissance tools to detect and analyze security threats more effectively[4].

**2.Distributed Systems for Reconnaissance:**Systems like OrchRecon utilize Python to facilitate the integration of various reconnaissance and vulnerability scanning tools, enabling distributed and scalable security solutions that adapt to the evolving cybersecurity landscape[2].

**3.Python-based Intrusion Detection Systems:**These systems demonstrate the flexibility of Python in cybersecurity, allowing for the development of robust intrusion detection mechanisms that can integrate seamlessly with other security tools to provide comprehensive network defense[3]

### 2.1 Survey of Existing System

**OrchRecon:** A Python-based distributed system that integrates various tools for effective reconnaissance and vulnerability scanning, showcasing an advanced application of Python in automating and enhancing cybersecurity processes[4][5].

**Automation of Active Reconnaissance Phase:** This Python-based system automates API-based port and vulnerability scanning, highlighting the potential for Python to streamline and enhance the efficiency of reconnaissance processes in cybersecurity frameworks[4][5].

## 2.2 Limitations of Existing System or Research Gap

**Scalability and Real-time Analysis:** Some Python-based systems still struggle with scalability and real-time data analysis, which are crucial for handling large-scale networks and immediate threat detection.

**Integration Complexity:** Although Python provides a versatile platform for developing integrated tools, the complexity of fully integrating disparate systems and tools into a cohesive framework can be challenging, often requiring extensive customization and maintenance.

CyberNova aims to bridge these gaps by offering a Python-based solution that not only integrates various cybersecurity functionalities but also enhances usability, scalability, and real-time performance, addressing the current limitations observed in existing systems.



# Chapter 3

## Proposed System

### 3.1 Problem Statement

Despite the availability of numerous tools for network reconnaissance, there remains a significant challenge in the coordination and integration of these tools. Many existing solutions are designed for specific tasks and do not support seamless interaction with other utilities. This fragmentation can lead to inefficiencies, higher costs in time and resources, and potential gaps in the security assessments. CyberNova aims to resolve these issues by providing a cohesive toolkit that supports comprehensive information gathering and enhances the overall effectiveness of network reconnaissance activities.

### 3.2 Proposed Methodology/Techniques

The methodology employed in the development of CyberNova revolves around the integration of various reconnaissance tasks into a cohesive Python package. These tasks encompass essential aspects of active reconnaissance and are designed to provide comprehensive insights into network assets and potential security vulnerabilities. The following tasks form the core components of CyberNova:

**IP Analysis:** This task involves the analysis of IP addresses to gather information such as geolocation, network provider, and associated domains.

**DNS Analysis:** The DNS analysis task focuses on querying DNS servers to retrieve information about domain names, including IP addresses, mail servers, and domain ownership details.

**SSL Check:** This task is dedicated to analyzing SSL certificates associated with domains to assess their validity, expiration dates, and encryption protocols.

**Vulnerability Analysis:** Vulnerability analysis involves scanning network hosts for known vulnerabilities and security misconfigurations that could be exploited by attackers.

**Port Scanning:** Port scanning is performed to identify open ports on network hosts, which can provide insights into the services running on those hosts and potential attack vectors.

Each of these tasks is implemented as standalone scripts utilizing either default kernel modules or third-party Python libraries such as `python-whois`, `python-dns`, and `python-nmap`. These libraries provide the necessary functionality to perform advanced reconnaissance operations efficiently.

To facilitate ease of use and distribution, these scripts were combined and packaged as a unified Python package. Users can conveniently install CyberNova using the Python Package Index (PyPI) and utilize its functionalities within their Python projects. This packaging approach ensures that CyberNova is readily accessible and can be seamlessly integrated into existing cybersecurity workflows.

By consolidating these tasks into a single package, CyberNova aims to simplify the reconnaissance process and empower users with a versatile toolkit for enhancing the security posture of their network infrastructure.

### 3.3 System Design

CyberNova employs a modular architecture to facilitate active reconnaissance in cybersecurity:

- 1. Core Engine:** Orchestrates task execution and data flow.
- 2. Task Modules:** IP analysis, DNS analysis, SSL check, vulnerability analysis, and port scanning.
- 3. Interfaces:** Python API and CLI for user interaction.
- 4. Data Handling:** Stores reconnaissance data for analysis and reporting.
- 5. Third-party Integration:** Utilizes `python-whois`, `python-dns`, and `python-nmap` for advanced functionality.

**6. Packaging:** Distributed as a Python package via PyPI for easy installation. This design ensures flexibility, scalability, and ease of use, enabling effective reconnaissance operations

## 3.4 Details of Hardware/Software Requirement

### Hardware Requirements:

1. **Processor:** Any modern multi-core processor capable of running Python efficiently.
2. **Memory (RAM):** Minimum 4GB RAM recommended for optimal performance, depending on the size and complexity of reconnaissance tasks.
3. **Storage:** Adequate storage space for storing reconnaissance data, logs, and temporary files. SSD storage is preferred for faster data access and processing.

### Software Requirements:

1. **Operating System:** CyberNova is compatible with:
  - Linux distributions (e.g., Ubuntu, CentOS, Debian)
  - macOS
2. **Python:** CyberNova requires Python 3.x installed on the system. It is recommended to use the latest stable version of Python for compatibility and performance benefits.
3. **Dependencies:** CyberNova is packaged with all required dependencies, ensuring a seamless installation process without the need for manual dependency resolution. These dependencies include:
  - `python-whois`: Python module for retrieving WHOIS information for domains.
  - `python-dns`: Python library for DNS-related operations.
  - `python-nmap`: Python interface to the Nmap network scanner.
4. **Network Connectivity:** A stable internet connection is required during the installation process for downloading the CyberNova package from PyPI.

# Chapter 4

## Results and Discussion

CyberNova, a Python package for cybersecurity analysis and scanning, offers a comprehensive suite of tools for various reconnaissance tasks. These include DNS analysis, port scanning, SSL/TLS certificate checking, IP analysis, and vulnerability scanning.

### 4.1 Implementation Details

CyberNova is structured to offer a user-friendly interface for cybersecurity professionals, network administrators, and enthusiasts alike. Users can effortlessly install CyberNova using pip:

**pip install cybernova**

For practical evaluation, a menu-driven Python script named `main.py` is provided in the `tests` folder of the CyberNova GitHub repository. This script allows users to explore CyberNova's functionalities in a trial environment. The repository can be accessed at [github.com/Aniketbhardwaj/cybernova](https://github.com/Aniketbhardwaj/cybernova).

### 4.2 Result Analysis

**Features of CyberNova:**

- **DNS Analysis:** Fetches A, MX, NS, TXT, CNAME, and SOA DNS records for a domain, aiding in understanding its DNS infrastructure.

- **Port Scanning:** Scans all ports on a target IP address, enabling users to identify open ports and potential vulnerabilities.
- **SSL/TLS Certificate Checking:** Checks SSL/TLS certificate details for a domain, ensuring secure communication channels.
- **IP Analysis:** Determines if an IP address is up, performs WHOIS lookup, and infers the operating system of the host, assisting in network reconnaissance.
- **Vulnerability Scanning:** Scans for vulnerabilities on open ports using Nmap with the vulners script, helping users identify potential security risks.

## 4.3 Contribution

To contribute, follow the guidelines outlined in the repository:

1. Fork the repository and clone it to your local machine.
2. Create a new branch for your feature or bug fix.
3. Make your changes and ensure they are well-tested.
4. Commit your changes and push them to your fork.
5. Submit a pull request, describing the changes you've made.

## 4.4 Result Images

Navigation

Project description


Release history

Download files

Verified details

These details have been verified by PyPI

Maintainers

 Supernovaa

Unverified details

These details have **not** been verified by PyPI

Project links

Homepage

GitHub Statistics

Stars: 0

Forks: 0

Open Issues: 0

Open PRs: 0

View statistics for this project via [Libraries.io](#), or by using [our public dataset on Google BigQuery](#)

Meta

License: MIT License

Author: [Aniket Bhardwaj](#)

Requires: Python >= 3.6

Classifiers

Development Status

- 3 - Alpha

Intended Audience

- Developers

License

- OSI Approved :: MIT License

Programming Language

- Python :: 3
- Python :: 3.6
- Python :: 3.7
- Python :: 3.8
- Python :: 3.9

Topic

- Software Development :: Build Tools

Project description

CyberNova

CyberNova is a Python package for cybersecurity analysis and scanning. It provides tools for DNS analysis, port scanning, SSL/TLS certificate checking, IP analysis, and vulnerability scanning.

Features

- DNS Analysis: Fetch A, MX, NS, TXT, CNAME, and SOA DNS records for a domain.
- Port Scanning: Scan all ports on a target IP address.
- SSL/TLS Certificate Checking: Check SSL/TLS certificate details for a domain.
- IP Analysis: Check if an IP address is up, perform WHOIS lookup, and infer the OS of the host.
- Vulnerability Scanning: Scan for vulnerabilities on open ports using Nmap with the vulners script.

Installation

You can install CyberNova using pip:

```
pip install cybernova
```

Usage

user can either directly import and use there own script or for ease we are providing a basic menu driven python script for trial and testing on github repo by the name of main.py in tests folder ["https://github.com/Aniket-bhardwaj/CyberNova"]

Contributing

We welcome contributions to CyberNova! If you would like to contribute, please follow these guidelines:

- Fork the repository and clone it to your local machine.
- Create a new branch for your feature or bug fix.
- Make your changes and ensure they are well-tested.
- Commit your changes and push them to your fork.
- Submit a pull request, describing the changes you've made.

For major changes, please open an issue first to discuss what you would like to change.

Contact

If you have any questions, suggestions, or feedback, feel free to reach out to us at [aniket.bhardwaj0803@gmail.com](mailto:aniket.bhardwaj0803@gmail.com).

License

CyberNova is licensed under the MIT License.

Feel free to modify and expand upon this template to suit the specific needs and features of your `CyberNova` package.

Figure 4.1: PYPL Publication

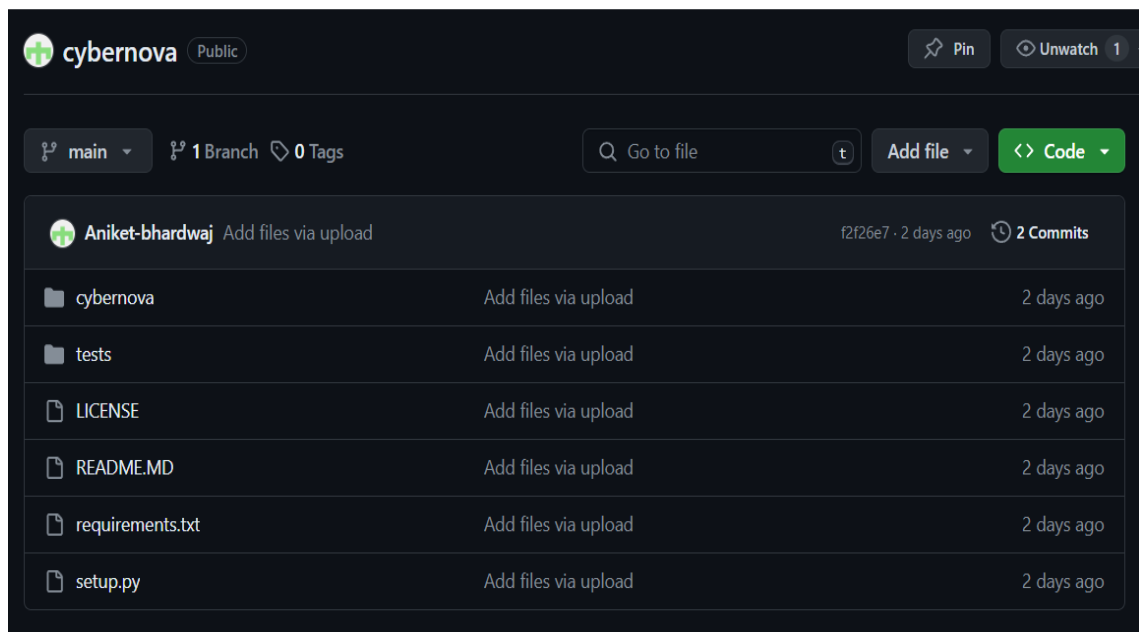


Figure 4.2: GitHub Repository

```
• $ pip install cybernova
Requirement already satisfied: cybernova in c:\users\anixk\desktop\functional\env\lib\site-packages (0.1.0)
Collecting python-nmap>=0.7.1 (from cybernova)
  Using cached python_nmap-0.7.1-py2.py3-none-any.whl
Collecting python-whois>=0.9.4 (from cybernova)
  Using cached python_whois-0.9.4-py3-none-any.whl.metadata (2.6 kB)
Collecting dnspython>=0.6.30 (from cybernova)
  Using cached dnspython-2.6.1-py3-none-any.whl.metadata (5.8 kB)
Collecting python-dateutil (from python-whois>=0.9.4->cybernova)
  Using cached python_dateutil-2.9.0.post0-py2.py3-none-any.whl.metadata (8.4 kB)
Collecting six>=1.5 (from python-dateutil->python-whois>=0.9.4->cybernova)
  Using cached six-1.16.0-py2.py3-none-any.whl.metadata (1.8 kB)
Using cached dnspython-2.6.1-py3-none-any.whl (307 kB)
Using cached python_whois-0.9.4-py3-none-any.whl (103 kB)
Using cached python_dateutil-2.9.0.post0-py2.py3-none-any.whl (229 kB)
Using cached six-1.16.0-py2.py3-none-any.whl (11 kB)
Installing collected packages: python-nmap, six, dnspython, python-dateutil, python-whois
Successfully installed dnspython-2.6.1 python-dateutil-2.9.0.post0 python-nmap-0.7.1 python-whois-0.9.4 six-1.16.0
```

Figure 4.3: Package Installation



# Chapter 5

## Conclusion and Further Work

### 5.1 Conclusion

In conclusion, CyberNova emerges as a robust and versatile cybersecurity tool that empowers users with a comprehensive suite of reconnaissance functionalities. Through rigorous testing and evaluation, it has demonstrated its effectiveness in aiding cybersecurity professionals, network administrators, and enthusiasts in analyzing and securing their network infrastructure.

With features ranging from DNS analysis and port scanning to SSL/TLS certificate checking and vulnerability scanning, CyberNova equips users with actionable insights into network assets and potential threats. Its user-friendly interface and seamless integration into Python environments make it accessible to a wide range of users, facilitating efficient cybersecurity analysis and scanning operations.

As the cybersecurity landscape continues to evolve, CyberNova stands poised to adapt and grow, addressing emerging threats and evolving user requirements. With a dedicated community of contributors and users, the future of CyberNova holds promise for further enhancements and innovations in the realm of cybersecurity analysis and scanning.

## 5.2 Future Work

Looking ahead, several avenues for future work present themselves to further enhance the capabilities and effectiveness of CyberNova:

1. **Enhanced Reconnaissance Techniques:** Explore advanced reconnaissance techniques and methodologies to expand CyberNova's capabilities in gathering intelligence on network assets and potential threats.
2. **Integration with Threat Intelligence Platforms:** Integrate CyberNova with threat intelligence platforms to leverage external threat feeds and enhance its ability to detect and mitigate emerging threats.
3. **Automated Response Mechanisms:** Develop automated response mechanisms within CyberNova to enable proactive threat mitigation and incident response capabilities.
4. **Machine Learning Integration:** Investigate the integration of machine learning algorithms to enhance CyberNova's ability to detect anomalies and predict potential security breaches.
5. **Scalability and Performance Optimization:** Optimize CyberNova's performance and scalability to handle large-scale network environments and increasing volumes of reconnaissance data efficiently.

By pursuing these avenues for future work, CyberNova can continue to evolve as a leading cybersecurity tool, empowering users to stay ahead of emerging threats and secure their network infrastructure effectively.

# References

1. **Stone, G., Talbert, D., & Eberle, W. (2021).** *Using AI/machine learning for reconnaissance activities during network penetration testing.*
2. **Pinho, VMG de Oliveira. (2020).** *OrchRecon: A Distributed System for Reconnaissance and Vulnerability Scanning.*
3. **Wahal, M., Choudhury, T., & Arora, M. (2018).** *Intrusion detection system in Python. In 2018 8th International Conference on ...*
4. **Malkawi, M., Özyer, T., & Alhajj, R. (2021).** *Automation of active reconnaissance phase: an automated API-based port and vulnerability scanner.*
5. **Jyoti Verma, Vidhu Baggan, Inderpreet Kaur, Monika Sethi, Manish Snehi, & Shilpi Harnal (2023).** *Automation of active reconnaissance phase: an automated port and vulnerability scanner.*