**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**Факультет прикладной математики – процессов управления**

**отчет**

**по лабораторной работе**

**по дисциплине «Системное программирование в Linux»**

**на тему «Разработка сетевого сканера»**

|  |  |  |
| --- | --- | --- |
| Студент гр. 22.Б15 |  | Агишев А.Б. |
| Преподаватель |  | Киямов Ж.У. |

**Санкт-Петербург**

**2023 г.**

**Содержание**

[1. Цель работы 2](#_Toc154235591)

[2. Задача 2](#_Toc154235592)

[3. Теоретическая часть 2](#_Toc154235593)

[4. Алгоритм метода 3](#_Toc154235594)

[5. Описание программы 4](#_Toc154235595)

[6. Рекомендации пользователю 5](#_Toc154235596)

[7. Рекомендации программисту 5](#_Toc154235597)

[8. Контрольный пример 6](#_Toc154235598)

[9. Заключение 6](#_Toc154235599)

[10. Литература 6](#_Toc154235600)

# **Цель работы**

Создать инструмент для сканирования и анализа сетевой активности.

# **Задача**

1. Реализовать функционал для сканирования портов на заданном диапазоне IP-адресов.
2. По результатам сканирования определять активные сервисы и службы на каждом обнаруженном узле.
3. Предоставить детализированную информацию об обнаруженных хостах, включая IP-адрес, MAC-адрес, страну, провайдера и др.
4. Разработать графический интерфейс для удобного взаимодействия с приложением.
5. Предоставить интуитивно понятный и информативный вывод результатов.

# **Теоретическая часть**

*IP (Интернет-протокол)* является ключевым протоколом сети Интернет, который обеспечивает маршрутизацию пакетов данных между узлами сети. Каждый узел в сети IP идентифицируется уникальным IP-адресом, который может состоять из 32 бит или 128 бит (IPv6). IP-адрес используется для направления пакетов данных по сети.

*Порт* в компьютерных сетях представляет собой числовой адрес, который служит для идентификации конкретного процесса или службы в сети. Протоколы TCP и UDP используют порты для организации коммуникации между устройствами.

*Сетевой сканер* — это инструмент, который позволяет анализировать сеть, обнаруживать устройства, открытые порты и службы, а также выполнять различные виды анализа безопасности сети.

*Scapy* — это мощная библиотека Python, предоставляющая программистам возможность создавать, отправлять, перехватывать и анализировать сетевые пакеты. Метод sr1 в библиотеке Scapy используется для отправки сетевых запросов (например, ICMP, ARP, DNS) и получения одиночного ответа. Это удобный инструмент для отправки настроенных сетевых пакетов и анализа ответов.

*ARP (Протокол разрешения адресов)* — это протокол канального уровня, который используется для отображения IP-адресов в физические MAC-адреса в локальных сетях. ARP запросы используются для определения MAC-адреса устройства по его IP-адресу в локальной сети.

*Библиотека Requests в Python* — это простой в использовании инструмент для выполнения HTTP-запросов. Она предоставляет удобный интерфейс для отправки запросов на серверы, обработки ответов и управления сеансами.

# **Алгоритм метода**

Программа состоит из следующих частей:

1. Инициализация графического интерфейса.
2. Получение IP-адресов и портов.
3. Сканирование IP-адресов и портов.
4. Обработка каждого IP-адреса и его открытых портов, а также получение дополнительной информации.
5. Отображение результатов сканирования в интерфейс, а также сохранение в файл.

# **Описание программы**

Программа состоит из класса Interface, который отвечает за интерфейс программы. Описание функций класса представлено в таблице 5.1. Также программа содержит функции, описание которых представлено в таблице 5.2.

*Таблица 5.1 Описание функций класса «Interface»*

|  |  |  |
| --- | --- | --- |
| Входные данные | Имя | Описание |
| - | \_\_init\_\_ | Инициализация интерфейса |
| - | create\_interface | Создание элементов интерфейса |
| Событие | prevent\_typing | Макрос чтобы не было возможности записывать текст в окно |
| - | set\_event | Изменение состояния флага скана портов |
| - | scan | Запуск сканирования |

*Таблица 5.2. Описание функций программы «main.py»*

|  |  |  |
| --- | --- | --- |
| Входные данные | Имя | Описание |
| - | get\_services | Получение данных с сайта сервисов портов |
| Данные о сервисах портов | parse\_services | Анализ и сохранение сервисов портов |
| IP-адреса, порты | scan\_ports | Сканирование IP-адресов с выбранными портами |
| IP-адрес | is\_on\_server | Проверка доступности IP-адреса |
| Порты, сервисы | name\_services | Связывание портов со службами |
| IP-адрес | get\_mac | Получение MAC-адреса IP-адреса |
| IP-адреса | get\_host | Получение информации об IP-адреса |
| IP-адреса, порты, сервисы портов, статистика | IP\_scan | Сканирование IP-адреса |
| - | main | Запуск программы и интерфейса |

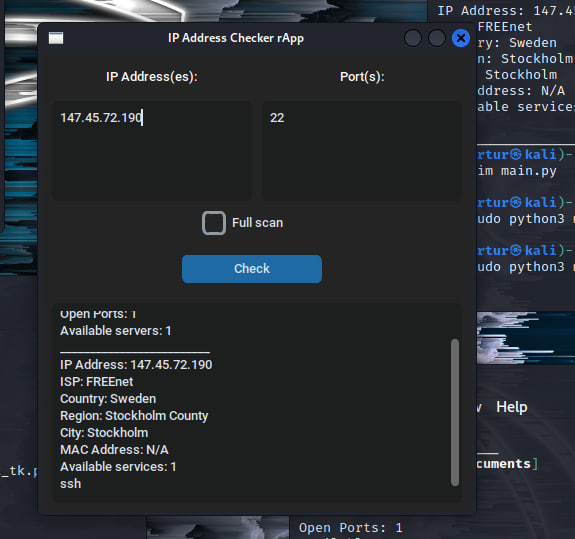
# **Рекомендации пользователю**

Запустите программу, далее укажите IP-адреса и порты по отдельности на каждой строчке. Для IP-адреса можно использовать маску, к примеру 182.101.45.\*[0-4] это будет значит, что программа переберет все адреса в данном диапазоне. Также можно сделать и для портов, только убрав \*. Если необходим полный перебор всех портов, то нужно поставить галочку рядом с пунктом “Full scan”. Для начала сканирования нажмите на кнопку “Check”. Результаты сканирования будут доступны внизу, а также в файле “results.txt” в той же директории, где была запущена программа.

# **Рекомендации программисту**

Для запуска программы необходим Python 3.11 и выше. Необходимо установить следующие библиотеки: scapy, requests, customtkinter.

# **Контрольный пример**

м

*Рисунок 8.1 Пример работы программы.*

# **Заключение**

В ходе выполнения этих задач были изучены средства Python для взаимодействия с сетью (Scapy), работа с внешними API (запросы через библиотеку requests). Полученный результат представляет собой функциональное и информативное приложение для работы с сетью, позволяющее выполнять сканирование и анализ узлов сети.

# **Литература**

1. <https://github.com/finstape/system-programming-in-linux/tree/main/NetScanner>
2. <https://scapy.readthedocs.io/en/latest/usage.html>