Санкт-Петербургский государственный университет

Факультет прикладной математики – процессов управления

**Лабораторная работа №4**

**Отчет**

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

**Разработка Сетевого Сканера**

Автор работы: Дацык Р.В.

Группа: 22.Б15-пу

Преподаватель: Дик А.Г.

Санкт-Петербург, 2023

**Оглавление**

[**1.** **Цель работы** 3](#_Toc151824592)

[**2.** **Задачи** 3](#_Toc151824593)

[**3.** **Введение** 3](#_Toc151824594)

[**4.** **Теоретическая часть** 3](#_Toc151824595)

[**5.** **Алгоритм метода** 4](#_Toc151824596)

[**6.** **Описание программы** 5](#_Toc151824597)

[**7.** **Рекомендации пользователю** 7](#_Toc151824598)

[**8.** **Рекомендации программисту** 7](#_Toc151824599)

[**9.** **Контрольный пример** 7](#_Toc151824600)

[**10.** **Вывод** 9](#_Toc151824601)

[**11.** **Список использованной литературы** 9](#_Toc151824602)

# **Цель работы**

Целью проекта является создание полнофункционального сетевого сканера с графическим интерфейсом, обеспечивающего удобное взаимодействие с пользователем.

# **Задачи**

* Разработка функционала для сканирования портов на заданном диапазоне IP-адресов.
* Определение активных сервисов и служб на каждом обнаруженном узле.
* Предоставление детализированной информации об обнаруженных хостах, включая IP-адрес, MAC-адрес, географические данные и другие.

# **Введение**

В данной лабораторной работе разрабатывается инструмент для сканирования и анализа сетевой активности. Сетевой сканер предназначен для определения открытых портов, активных сервисов и сбора информации о хостах в заданном диапазоне IP-адресов.

# **Теоретическая часть**

Сетевые сканеры являются важным инструментом для администраторов систем и специалистов по информационной безопасности. Они используются для анализа состояния и безопасности сетей, выявления уязвимостей, а также диагностики сетевых узлов. Основной задачей сканеров является определение активных узлов, открытых портов и запущенных служб на целевых машинах.

#### 4.1. Типы сканирования портов

**Активное сканирование** включает в себя отправку сетевых запросов на целевые узлы с целью выявления активных портов. Это включает в себя отправку TCP или UDP пакетов для определения того, какие порты открыты или закрыты. Однако такой вид сканирования может быть обнаружен средствами межсетевой безопасности.

**Пассивное сканирование** основывается на анализе сетевого трафика без отправки активных запросов. Это позволяет избежать обнаружения, но требует доступа к сетевому трафику.

#### 4.2. Библиотека Scapy

Scapy — мощная библиотека на языке Python для работы с пакетами в сети. Она предоставляет инструменты для создания, отправки и приема сетевых пакетов. Scapy может использоваться для реализации различных видов сетевых сканеров, включая сканирование портов и анализ трафика.

#### 4.3. Графический интерфейс Tkinter

Tkinter — встроенная библиотека для создания графического интерфейса пользователя в Python. Она обеспечивает набор инструментов для разработки оконных приложений, включая кнопки, текстовые поля и многое другое. Tkinter удобен для создания простых и интуитивно понятных интерфейсов.

#### 4.4. Безопасность и Этика

При разработке сетевых сканеров важно соблюдать этические стандарты и соблюдать законы. Активное сканирование может быть воспринято как атака, поэтому необходимо получить соответствующие разрешения перед его использованием. Также, при использовании аутентификации, важно убедиться в соблюдении приватности и безопасности передаваемых учетных данных.

# **Алгоритм метода**

1. **Сканирование портов:**

* Определение диапазона IP-адресов.
* Использование сокетов для проверки открытости портов на каждом узле.

1. **Определение сервисов:**

* После обнаружения открытых портов, определение сервисов с использованием стандартной библиотеки Python.

1. **Сбор информации о хостах:**

* Получение данных о хостах, таких как географические данные и MAC-адреса.

# **Описание программы**

Программа написана на языке программирования Python с использованием библиотеки Tkinter для графического интерфейса и Scapy для работы с сетью. Она позволяет пользователю указывать целевые IP-адреса, диапазон портов для сканирования, а также предоставляет информацию об открытых портах, сервисах и хостах.

Таблица 6.1. Описание функций

|  |  |
| --- | --- |
| Название функции | Назначение функции |
| \_\_init\_\_(self, root) | Инициализация графического интерфейса (GUI) и установка начальных параметров приложения. |
| scan\_network(self) | Метод, запускающий процесс сканирования в отдельном потоке при нажатии на кнопку "Scan". |
| perform\_scan\_and\_display(self) | Метод, который выполняет сканирование и отображает результаты в графическом интерфейсе. |
| perform\_scan(self, target\_ips) | Метод, выполняющий сканирование для заданных IP-адресов. Собирает информацию о хосте, открытых портах и предоставляет данные о сервисах. |
| create\_packet(self, target\_ip, port) | Создает сетевой пакет для отправки на указанный IP и порт. |
| send\_packet(self, packet) | Отправляет созданный пакет и получает ответ. Используется для определения открытых портов. |
| get\_geo\_info(self, ip\_address) | Получает географическую информацию для заданного IP-адреса, используя внешний сервис. |
| get\_service\_info(self, port) | Определяет название сервиса по номеру порта. |
| get\_host\_info(self, ip\_address) | Получает информацию о хосте по его IP-адресу. |
| display\_results(self, result) | Отображает результаты сканирования в графическом интерфейсе. |
| save\_results(self) | Сохраняет результаты сканирования в файл, предоставляя пользователю выбор формата (Text, JSON, CSV). |
| if \_\_name\_\_ == "\_\_main\_\_" | Блок кода, который запускает приложение при запуске скрипта. Создает экземпляр класса NetworkScanner и запускает главный цикл событий. |

# **Рекомендации пользователю**

Для успешного запуска программы необходимо устройство с операционной системой Linux, macOS или Windows, а также среда разработки, поддерживающей запуск python 3.10

Для запуска программы введите команду sudo python3 Scan.py. В открывшемся окне введите интересующий вас ip-адрес и порт, нажмите кнопку «Scan». Если вы хотите сохранить вывод выберите интересующее расширение файла и нажмите кнопку «Save Results»

# **Рекомендации программисту**

Для запуска программы необходима 64-битная операционная система Windows, Linux или macOS. Для работы с кодом необходима среда разработки, совместимая с компилятором С++ и библиотеки <iostream>, <ctime>, <cstdlib> и другие.

Исходный код программы и доступен по ссылке:

<https://github.com/CapTopGrade/Linux/blob/main/Scan/Scan.py>

# **Контрольный пример**

В данном разделе представлен контрольный пример, демонстрирующий работу программы.

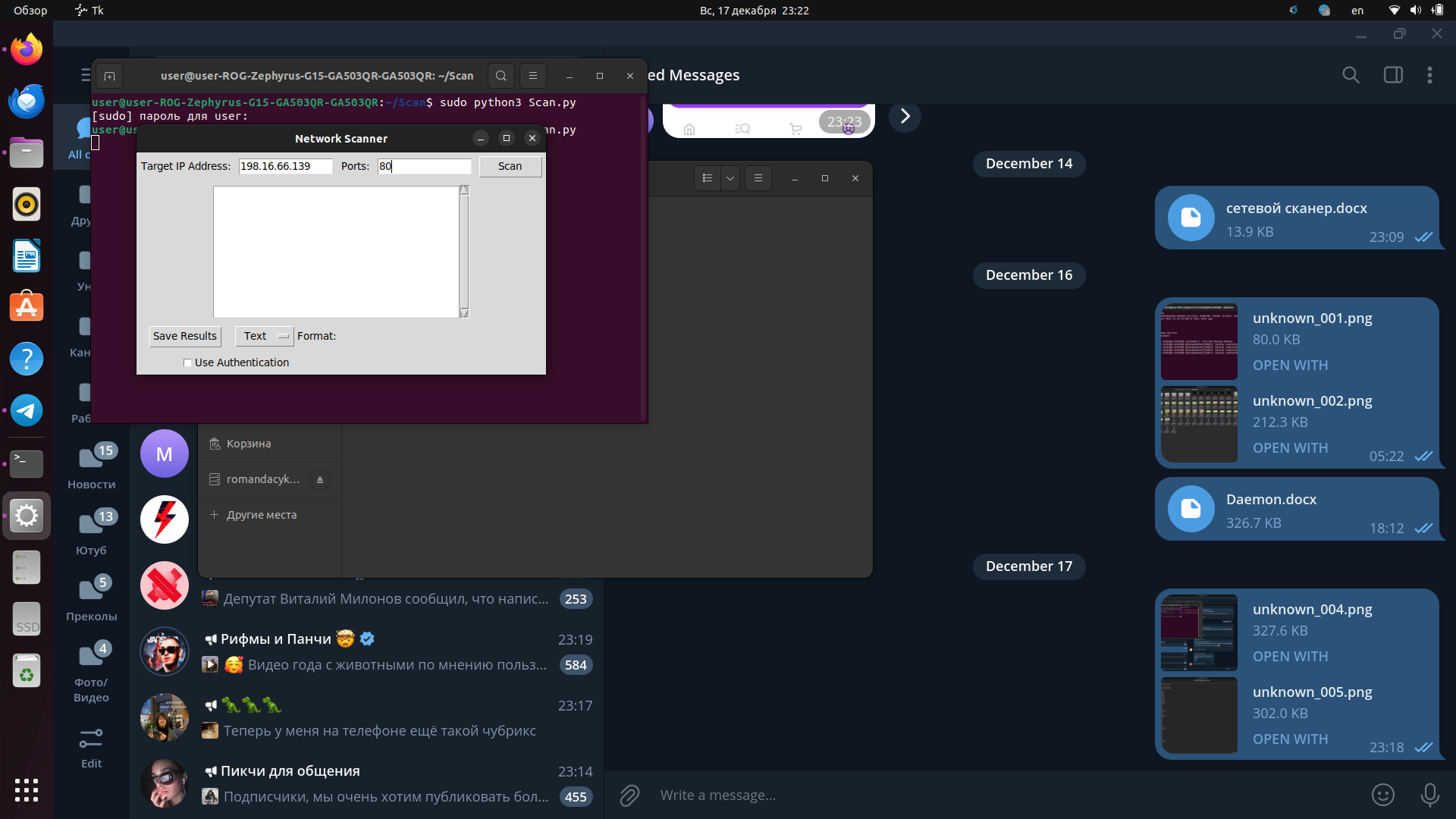


Рис. 9.1 Демонстрация интерфейса программы

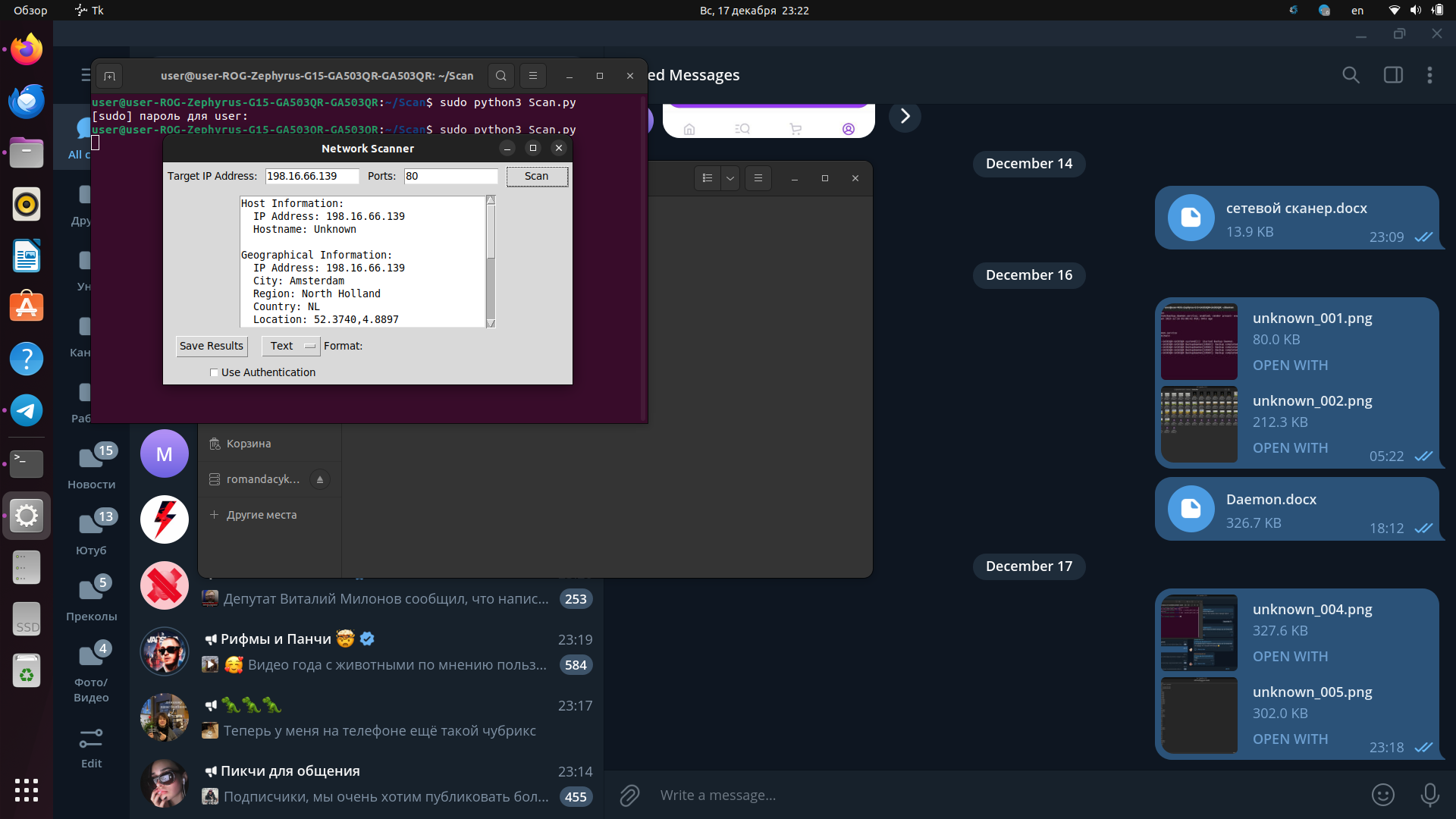


Рис. 9.2 Пример вывода программы 1

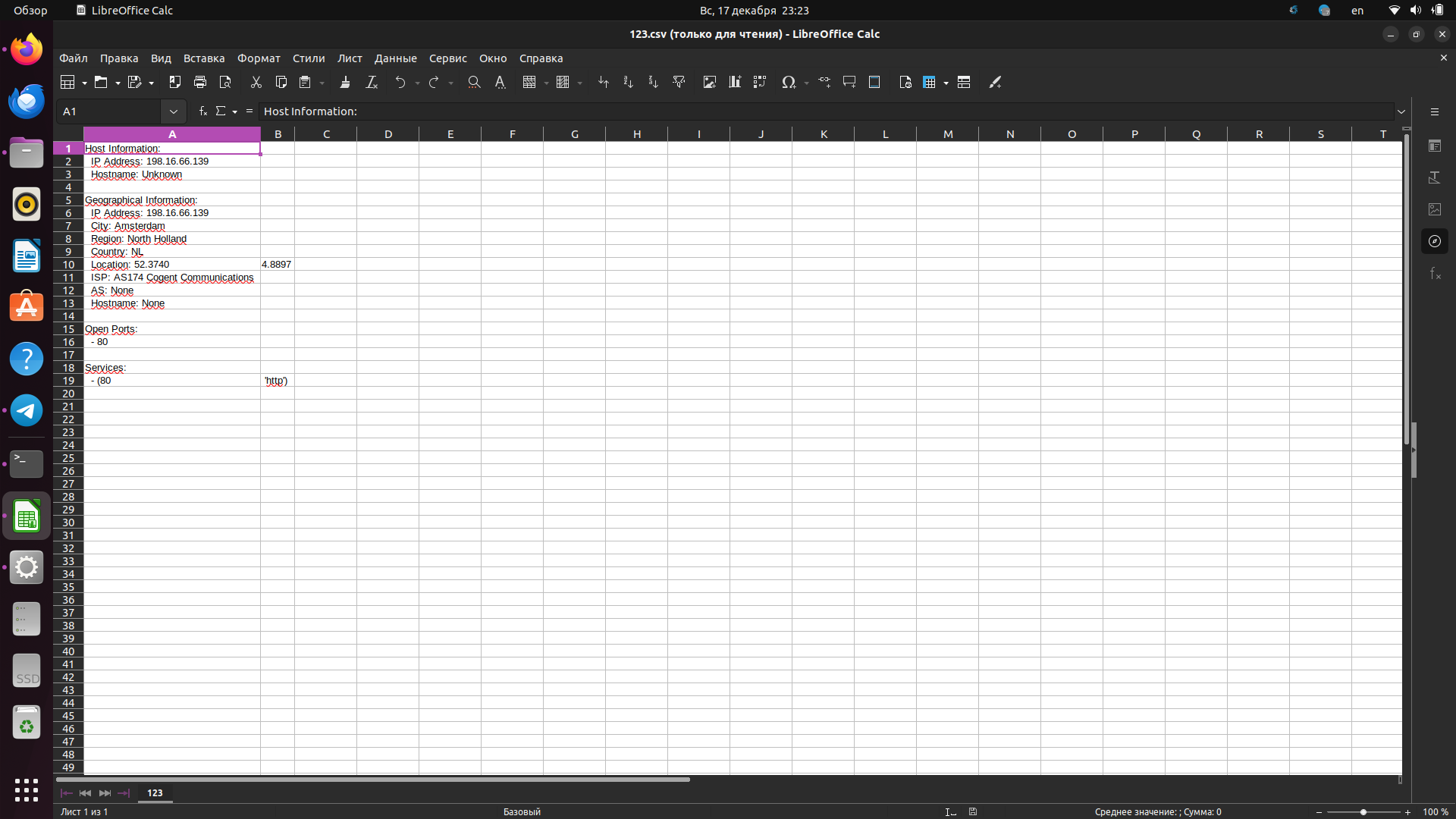


Рис. 9.3 Пример вывода программы 2

# **Вывод**

Лабораторная работа позволила разработать эффективный сетевой сканер с графическим интерфейсом, способный проводить сканирование портов и предоставлять подробную информацию о хостах в сети. Реализация алгоритма метода и использование современных библиотек сделали приложение удобным в использовании и мощным инструментом для анализа сетевой безопасности.

# **Список использованной литературы**

[1] Документация по работе с С++: <https://ru.cppreference.com/w/>

[2] Рекомендации по улучшению кода: <https://github.com/isocpp/CppCoreGuidelines/blob/master/CppCoreGuidelines.md>