

Создание и настройка FTP-сервера для локальной сети на Python

Болтовский Гавриил Александрович

Приамурский государственный университет им. Шолом-Алейхема

Студент

Аннотация

Цель статьи – демонстрация процесса разработки и настройки FTP-сервера для локальной сети с использованием Python. Рассматриваются этапы создания программы, решения возможных проблем, а также анализируются аспекты безопасности и сравнение с альтернативными протоколами. В результате представлены рекомендации по настройке доступа для различных устройств, включая смартфоны и ПК.

Ключевые слова: FTP, LAN, Python

Creating and configuring FTP server for local network in Python

Boltovsky Gavriil Alexandrovich

Sholom-Aleichem Priamursky State University

Student

Abstract

The purpose of this article is to demonstrate the process of developing and configuring an FTP server for a local network using Python. The stages of creating the program, solving possible problems, publishing on GitHub are considered, and security aspects and comparison with alternative protocols are analyzed. As a result, recommendations for configuring access for various devices, including smartphones and PCs, are presented.

Keywords: FTP, LAN, Python

1 Введение

1.1 Актуальность исследования

В условиях стремительного развития информационных технологий и увеличения объемов данных, необходимость в эффективных инструментах передачи информации становится все более очевидной. Многие организации и частные пользователи сталкиваются с задачей обеспечения быстрого и безопасного доступа к данным, хранящимся на серверах.

Применение языка программирования Python для разработки FTP-сервера предоставляет возможность создания гибкого и легко настраиваемого решения, способного удовлетворить специфические потребности различных пользователей. Более того, Python обладает широкими возможностями по интеграции с другими технологиями и сервисами.

1.2 Обзор исследований

В работе Д.В. Конобрицкого [1] анализируются ключевые аспекты выбора и настройки FTP-серверов на платформе Linux, включая установку и конфигурацию vsFTPD. Рассматриваются аспекты лицензирования и отмечается эволюция протокола FTP, повлиявшая на его востребованность.

Исследование А.В. Иванова [2] акцентирует внимание на уязвимостях FTP-протокола, связанных со сбором информации о сети и атак методом перебора, обусловленных особенностями ответов сервера. Отмечается отсутствие механизмов ограничения попыток аутентификации, что требует дополнительных мер защиты.

Работа А.А. Акулова [3] посвящена разработке системы управления доступом в корпоративной сети путём настройки маршрутизатора, в том числе настраивается доступ к FTP серверу компании.

Практические аспекты применения FTP-серверов для резервного копирования рассматриваются в исследовании [4], где анализируются подходы к хранению данных с использованием vsFTPD, с учётом безопасности, эффективности и масштабируемости.

1.3 Цель исследования

Цель статьи – демонстрация процесса разработки и настройки FTP-сервера для локальной сети с использованием Python.

1.4 Постановка задачи

Основными задачами исследования являются разработка и настройка FTP-сервера на Python, анализ особенностей доступа с различных устройств, а также сравнение FTP с другими протоколами передачи данных.

2 Методы исследования

Протокол FTP (File Transfer Protocol) исторически является одним из первых стандартов для передачи файлов между устройствами в сети. Однако с развитием технологий появились альтернативные протоколы, обеспечивающие более высокий уровень безопасности, производительности и удобства использования. NFS и SMB предоставляют сетевой доступ к файлам с возможностью совместной работы, интеграции с операционными системами и управления доступом. HTTP(S) используется для передачи данных в веб-среде благодаря высокой скорости и безопасности. AFS обеспечивает масштабируемость и надежность в распределенных системах. Эти протоколы отличаются архитектурой и подходом, но объединены общей целью – эффективной передачи данных в сети.

Протокол FTP изначально не предусматривает встроенной поддержки списков контроля доступа (ACL, Access Control List) в своей спецификации. Однако управление доступом может быть реализовано на уровне файловой системы сервера. Современные FTP-серверы, такие как ProFTPD или vsftpd, могут интегрироваться с файловыми системами, поддерживающими ACL, или

использовать расширенные механизмы контроля доступа, предоставляемые операционной системой.

В рамках исследования рассматривается разработка FTP-сервера для обмена данными между устройствами, объединенными в локальную сеть. Для реализации выбран язык программирования Python, а графический интерфейс создается с использованием библиотеки Tkinter. Основу серверной части составляет библиотека `pyftplib`, обеспечивающая гибкость и надежность в работе с протоколом FTP.

Программа функционирует следующим образом. После запуска пользователь определяет директорию, которая станет доступной для других устройств в локальной сети. Затем приложение автоматически определяет IP-адрес компьютера, на котором оно выполняется, и использует его для запуска FTP-сервера. Такой подход обеспечивает простоту настройки и гарантирует корректную интеграцию с локальной сетью, исключая необходимость ручного задания сетевых параметров.

В описываемом случае карта сети выглядит как на схеме (рис. 1). Схема построена с использованием веб-приложения Smartdraw [5]

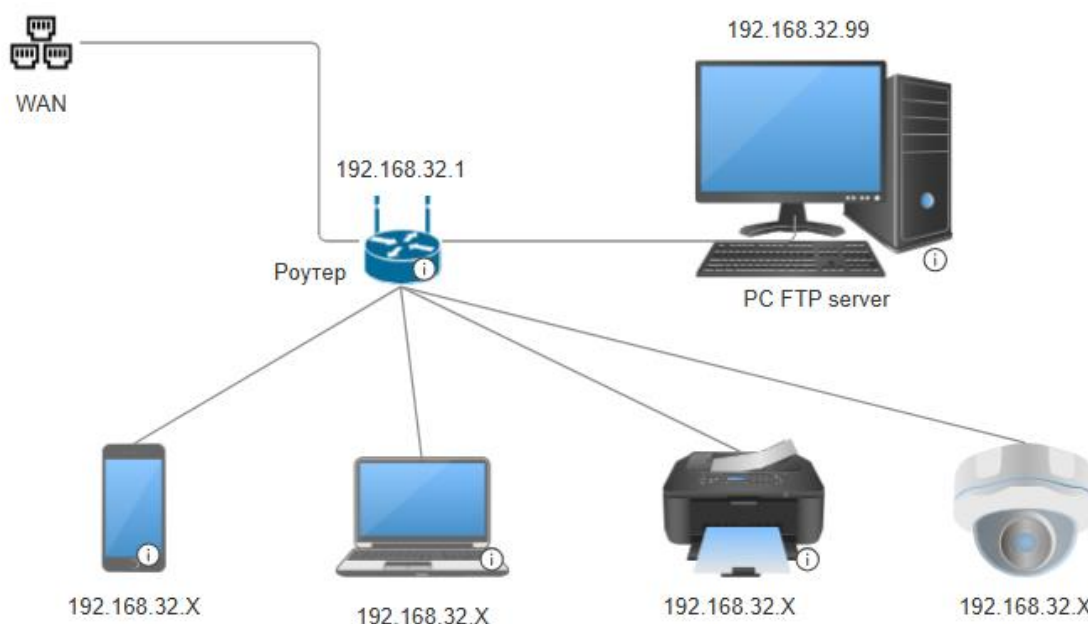


Рисунок 1 – Схема сети, в которой запускается приложение

Доступ к FTP-серверу может быть организован через авторизацию с использованием логина и пароля либо настроен в режиме гостевого доступа, не требующего учетных данных. Для повышения безопасности рекомендуется использовать сложные пароли и логины, так как FTP не обеспечивает шифрования данных.

3 Результаты и обсуждения

Внешний вид окна приложения изображён на рисунке (рис. 2).

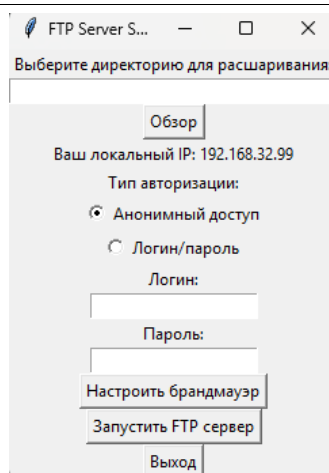


Рисунок 2 – Окно приложения

Программа обеспечивает возможность запуска сервера на локальной машине с настраиваемыми параметрами, созданными с помощью библиотеки Tkinter. Работа программы включает несколько взаимосвязанных компонентов, среди которых выделяются настройка сетевых параметров, управление авторизацией, обеспечение многозадачности и графический интерфейс для взаимодействия с пользователем.

Серверная часть построена на базе библиотеки pyftplib, которая предоставляет функциональность для работы с протоколом FTP. Пользователь через интерфейс может выбрать директорию для предоставления общего доступа, определить тип авторизации (анонимный доступ или логин и пароль) и запустить сервер. Для повышения удобства программа автоматически определяет локальный IP-адрес компьютера, что исключает необходимость ручной настройки сетевых параметров.

Особое внимание уделено многопоточности, что обеспечивает независимость работы интерфейса от работы сервера. Основной поток приложения управляет графическим интерфейсом, позволяя пользователю взаимодействовать с программой без задержек. Одновременно сервер запускается в отдельном потоке, что предотвращает блокировку интерфейса в процессе обслуживания FTP-клиентов. Это достигается использованием библиотеки `threading`, которая позволяет запускать сервер в фоне и обновлять статус работы в реальном времени.

Дополнительная функция программы включает автоматическую настройку брандмауэра для открытия порта 21, необходимого для работы FTP-сервера. Это реализуется через вызов системной команды с использованием модуля `subprocess`. Также предусмотрена обработка ошибок, например, в случае невозможности определить IP-адрес или проблемы с настройкой сетевых параметров, программа уведомляет пользователя через диалоговые окна.

Для проверки работоспособности приложения из-под ОС Android необходимо установить FTP-клиент (рис. 3).

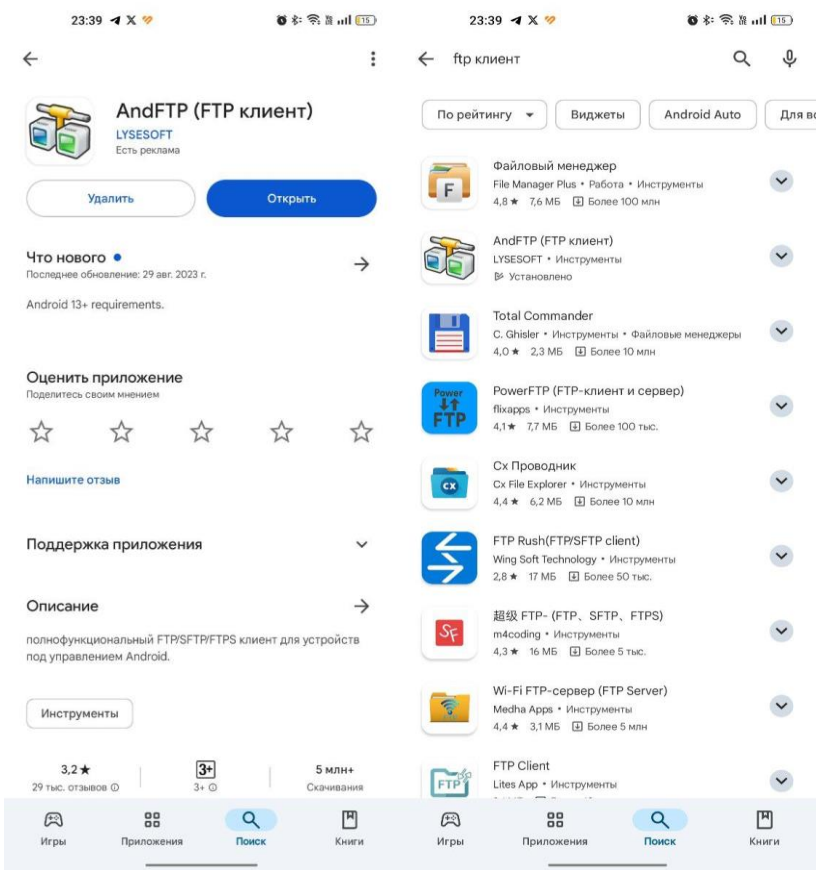


Рисунок 3 – Популярные FTP-клиенты в магазине приложений

После присоединения к локальной сети посредством WI-FI, можно подключиться к серверу (рис. 4).

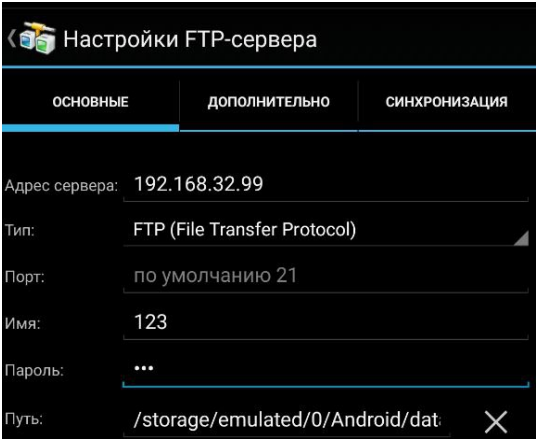


Рисунок 4 – Настройка подключения

В консоли выводятся сообщения об удачном подключении к серверу (рис. 5).

```
[I 2025-01-27 23:47:06] concurrency model: async
[I 2025-01-27 23:47:06] masquerade (NAT) address: None
[I 2025-01-27 23:47:06] passive ports: None
[I 2025-01-27 23:47:06] >>> starting FTP server on 192.168.32.99:21, pid=11904 <<<
[I 2025-01-27 23:47:22] 192.168.32.184:49232-[] FTP session opened (connect)
[I 2025-01-27 23:47:22] 192.168.32.184:49232-[123] USER '123' logged in.
[I 2025-01-27 23:47:22] 192.168.32.184:49232-[123] CWD C:\Users\Gavri\Downloads\Telegram Desktop 250
[I 2025-01-27 23:47:24] 192.168.32.184:49232-[123] FTP session closed (disconnect).
[I 2025-01-27 23:47:28] 192.168.32.184:49244-[] FTP session opened (connect)
[I 2025-01-27 23:47:28] 192.168.32.184:49244-[123] USER '123' logged in.
[I 2025-01-27 23:47:28] 192.168.32.184:49244-[123] CWD C:\Users\Gavri\Downloads\Telegram Desktop 250
```

Рисунок 5 – Сообщения в консоли приложения

Под управлением ОС Windows клиентом может выступать приложение FileZilla [6].

Подключившийся пользователь может копировать, просматривать, удалять, переименовывать файлы на сервере. Дополнительные уровни доступа можно настроить силами файловой системы ОС.

Получить исходный код приложения можно в GitHub репозитории проекта [7].

3. Выводы

В результате проведенной работы показана возможность создания и настройки FTP-сервера на Python для локальной сети.

Библиографический список

1. Конобрицкий Д. В. Выбор и настройка FTP-сервера на базе ОС Linux // Развитие современной науки и образования: анализ опыта и тенденций. 2022. С. 373-377.
2. Иванов А. В. Исследование безопасности протокола FTP // Научно-технический вестник информационных технологий, механики и оптики. 2005. №. 19. С. 167-171.
3. Акулов А. А. Управление доступом к ресурсам локальной сети с помощью ACL // Modern science. 2020. №. 11-3. С. 412-416.
4. Иванова Н. А., Савватеев м. Е. Использование FTP-сервера для хранения резервных копий системы // Вестник образовательного консорциума среднерусский университет. Информационные технологии. 2019. №. 1. С. 35-38.
5. Smartdraw. URL: <https://www.smartdraw.com/> (дата обращения: 27.01.2025).
6. FileZilla. URL: <https://filezilla-project.org/> (дата обращения: 27.01.2025).
7. GitHub Local_FTP_Server. URL: https://github.com/Gavriilbolt/Local_FTP_Server (дата обращения: 27.01.2025).