VIET NAM NATIONAL UNIVERSITY

HO CHI MINH UNIVERSITY OF SCIENCE

FACULTY OF INFORMATION TECHNOLOGY

Secure FTP Client with Virus Scanning via ClamAVAgent

Topic: Socket Programming

Course: Computer Networking

Authors:

Toan Nguyen Thanh (24127559)

Viet Doan The (24127590)

Huy Vo Minh (24127405)

Instructors:

MS. Son Le Ngoc

MS. Quan Nguyen Thanh

Ho Chi Minh City, August 1, 2025

Contents

List of Tables

List of Figures

1   Information

1.1   Authors

This report was written by the following four authors:

Student's name	Student's ID
Nguyen Thanh Toan	24127559
Doan The Viet	24127590
Vo Minh Huy	24127405

Table 1: Authors of the Report

Note: The authorship of this report will not change, except in unavoidable circumstances as required by the instructors.

1.2   Declaration

- The content of this report is based on our understanding and research on socket programming.

- The content in this report is original work.

- All referenced information is fully cited.

2   Overview

2.1   Target

- Demonstrate file transfer via FTP using a custom socket implementation, and scan for viruses with ClamAV Agent before uploading files.

- Protect the server from files containing malware.

- Practice Socket Programming, File Transfer Protocol, and ClamAV.

2.2   Overall Architecture

This system is designed based on a three-component model:

- **FTP Client**: A custom-developed application.

- **ClamAV Agent**: A dedicated agent deployed to interface with the ClamAV antivirus engine.

- **FTP Server**: A standard FTP server, such as FileZilla Server.

The core operational workflow is designed for security. When a user initiates a file upload, the file is not sent directly to the FTP Server. Instead, it follows a pre-scan procedure:

1. The **FTP Client** first sends the file to the **ClamAV Agent** over a dedicated TCP socket on **port 8888**.

2. The Agent scans the file using the ClamAV engine.

3. Based on the scan result, the upload is either approved or rejected:

   - If the Agent returns an **"OK"** status (indicating the file is clean), the FTP Client then proceeds to upload the file to the FTP Server.

   - If the result is **"INFECTED"**, the upload process is immediately aborted to prevent malware from reaching the server.

2.3   Main Components

- **FTP Client** (`ftp_client.exe`): This is a C++ application that provides the user interface, handles standard FTP commands (e.g., `put`, `get`, `ls`), and manages communication with the ClamAV Agent for security scans.

- **ClamAV Agent** (`clamav_agent.exe`): A background process that acts as a security gateway. It continuously listens on port 8888 for incoming files from the Client. Upon receiving a file, it calls the `clamscan.exe` command-line scanner to perform the virus check and returns the final result ("OK" or "INFECTED") back to the Client.

- **FTP Server** (`FileZilla Server`): A standard FTP server responsible for the secure storage and management of the uploaded files.

3 Screenshots of a successful session

3.1 System Requirements and Setup

- **Operating System:** Windows 10/11.

- **Software Prerequisites:**

  - **FileZilla Server:** Must be installed and running to act as the destination file server.

  - **ClamAV for Windows:** The ClamAV engine is required for scanning. The virus databases (`main.cvd`, `daily.cvd`, `bytecode.cvd`) must be downloaded manually from the official ClamAV website and placed in the appropriate database directory.

- **Server Configuration:** An FTP user account (e.g., `testuser`) must be created on the FileZilla Server with read/write permissions to the target directory.

3.2 Operating Instructions

To run the application, follow this specific order:

1. **Step 1:** Execute `clamav_agent.exe`. This will start the background agent, which will begin listening for incoming scan requests on TCP port 8888.

2. **Step 2:** Execute `ftp_client.exe`. This will launch the main client application, allowing you to connect to the FTP server and begin a session.

3.3 Key Commands

The client supports several important commands, including:

- `put`: Upload a single file (with pre-scan).

- `mput`: Upload multiple files.

- `get`: Download a single file.

- `mget`: Download multiple files.

- `ls`: List files on the server's current directory.

- `putall`: Recursively upload an entire local folder.

- `getall`: Recursively download an entire remote folder.

3.4 Screenshots for a Successful Session

4 Problems Encountered and Solutions

Problem	Cause	Solution	Result
Server cannot login	Connect() function don't provide login with user and password	Make ManualLogin() function	Now can input username and password
Server cannot change to passive mode	Client don't send command PASV	Make togglePassive() send command	Now server can change to passive mode and appear log on server
Recursive upload and download cannot run	Not include library yet	Include library string and filesystem	Now function can run and dow/up multiple file

5 Sử dụng tài liệu tham khảo

File BibTeX tài liệu tham khảo nằm ở đường dẫn `ref/ref.bib`. Sửa tên file `.bib` sẽ phải sửa lại nội dung file `ref.tex`.

Đây là ví dụ cite một tài liệu[1].

References

[1] George D. Greenwade. The Comprehensive Tex Archive Network (CTAN). *TUGBoat*, 14(3):342–351, 1993.

A Phụ lục

- Template này **không phải** là template chính thức của Khoa Công nghệ thông tin - Trường Đại học Khoa học Tự nhiên.

- Các hình ảnh, bảng biểu, thuật toán trong template chỉ mang tính chất ví dụ.

- Nhóm tác giả phân phối **miễn phí** template này trên GitHub và trên Overleaf với Giấy phép GNU General Public License v3.0. Nhóm tác giả không chịu trách nhiệm với các bản phân phối không nằm trong hai kênh phân phối chính thức nêu trên.