FTPChat Project

File Transfer Protocol Chat Project

Made & Presented by: Ahmed Omar Saad

CONTENTS

1. What is FTPChat?
2. What is an FTP server?
3. Advantages for FTP Chat
4. What makes it different?
5. Visualizing how it works
6. Sources

1. What is FTPChat?

FTPChat is a custom-built, encrypted messaging protocol that operates over the File Transfer Protocol (FTP) instead of traditional socket-based communication. It transforms FTP servers, hosted on routers like the ZTE ZXHN H188A SuperVectoring or via platforms like SFTPCloud.io, into secure, relay-based messaging hubs.

2. What is An FTP Server?

An FTP server is a system that uses the File Transfer Protocol (FTP) to store, share, and manage files across a network. According to Cisco, FTP operates on a client-server model using TCP/IP, Port 21 handles control commands (login, navigation).

In practical setups, FTP servers can be hosted on routers with USB and FTP capabilities, such as the ZTE ZXHN H188A transforming a simple flash drive into a customizable cloud-like system.

Unlike commercial cloud services (e.g., Google Drive or Dropbox), FTP servers offer:

- Full control over storage and access
- Local and remote file management
- And are more developer-friendly

3. FTPChat Advantages over web sockets

- Simplified Architecture

FTPChat replaces complex socket programming with FTP, an easy-to-learn protocol that handles transport, authentication, and session management natively.

- Router-Based Hosting

Easily deployable on routers with USB and FTP capabilities, such as the ZTE ZXHN H188A, turning local storage into a secure relay without custom server setups.

- Global Access via SFTPCloud.io

Eliminates the need for tunneling tools like Ngrok. Relays are hosted directly and securely, improving stability and scalability.

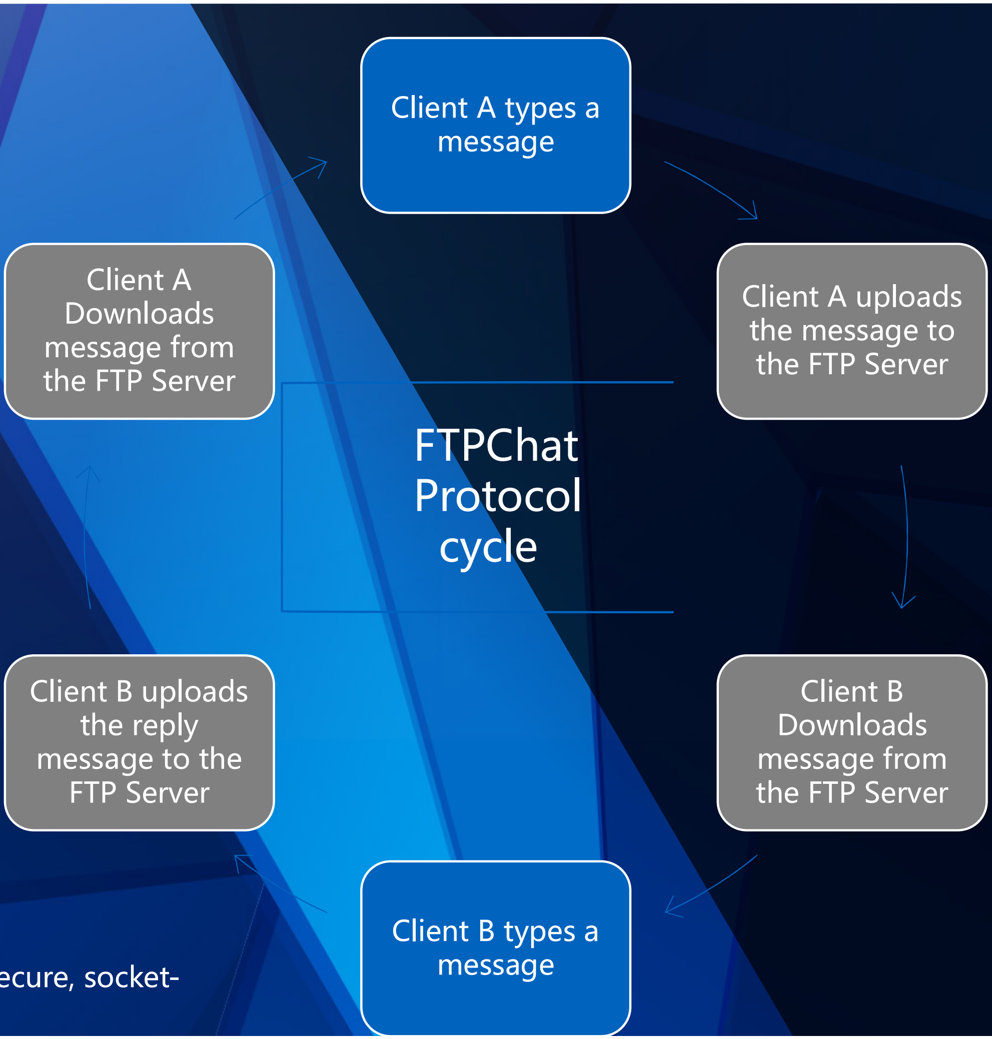
- Modular Encryption Layer

Built with Python, FTPChat`s encryption is flexible and secure, without relying on low-level socket handling. And it comes with 24-Layer Mono-Alphabetic encryption.

4. What makes it different

FTPChat replaces traditional socket-based messaging with a stable, FTP-driven architecture. This shift simplifies deployment, improves compatibility, and reduces development complexity.
It consumes fewer system resources, avoids common socket-related issues, Enables faster and more reliable deployment. FTPChat introduces features rarely found in conventional chat systems, making it ideal for secure communication, protocol innovation, and legacy-grade attribution.

5. Visualizing how it works


Client A types a message
Client A uploads the message to the FTP Server
Client B Downloads message from the FTP Server
Client B types a message
Client B uploads the reply message to the FTP Server
Client A Downloads message from the FTP Server
FTPChat Protocol cycle

Note: The FTP server acts as a relay node, secure, socket-free, and encrypted.

6. Sources

1.Cisco: "FTP uses TCP Port 21 for control and data transfer."
Link:
https://www.cisco.com/c/en/us/td/docs/ios/sw_upgrades/interlink/r2_0/user/ugftpc1.html

2.Microsoft Learn: "Port 22 is used for secure file exchange over SSH."
Link:
https://learn.microsoft.com/en-us/troubleshoot/azure/general/secure-file-exchange-transfer-files

3.IBM Docs: "Sockets require open ports and persistent connections, which may be blocked by firewalls."
Link:
https://www.ibm.com/docs/en/i/7.4.0?topic=programming-how-sockets-work

THANK YOU

Thank you for your time, I hope you enjoyed the project!