دانشگاه تحصیلات تکمیلی علوم پایه زنجان

دانشکده علوم کامپیوتر و فناوری اطلاعات

شبکه های کامپیوتری پیشرفته

FTP Server Programming

استاد:

دکتر پیمان پهلوانی

دانشجو:

فاطمه فلاح

14034105

پاییز 1403

معرفی پروژه

این پروژه یک سیستم مدیریت فایل ساده ولی مؤثر برای شبکه های مبتنی بر پروتکل TCP ایجاد می‌کند؛ که در آن دو جزء اصلی سرور و کلاینت، به کاربران امکان مدیریت فایل‌ها از راه دور را می‌دهند. این سیستم اجازه می‌دهد که فایل‌ها را در سرور آپلود، دانلود و حذف کنید، به جستجوی فایل‌های خاص بپردازید و لیست فایل‌های موجود را مشاهده کنید. با این امکانات، یک محیط کارآمد برای مدیریت فایل‌ها فراهم می‌شود که می‌تواند در شبکه‌های آموزشی، سازمانی و حتی به عنوان پروژه‌های آموزشی برای آشنایی با شبکه‌های کامپیوتری مفید باشد. همچنین، برای افزایش سطح امنیت انتقال داده‌ها، از الگوریتم Fernet برای رمزنگاری و رمزگشایی استفاده می‌شود. این الگوریتم رمزنگاری، تضمین می‌کند که داده‌های ارسال شده به صورت ایمن منتقل شده و از دسترسی‌های غیرمجاز محافظت می‌شوند. Fernet مبتنی بر الگوریتم رمزنگاری AES عمل می‌کند و به دلیل سادگی و کارایی در رمزنگاری و رمزگشایی، یکی از گزینه‌های مناسب برای این سیستم است.

نصب و راه‌اندازی

برای اجرای این پروژه، ابتدا باید زبان برنامه‌نویسی Python نسخه ۳.۶ یا بالاتر را نصب کرد. سپس به کتابخانه‌ی cryptography نیاز خواهیم داشت که می‌توان آن را با دستور زیر نصب کرد:

```
-pip install cryptography
```

کتابخانه cryptography ابزارها و توابع مورد نیاز برای رمزنگاری و رمزگشایی داده‌ها را در اختیار ما قرار می‌دهد. در این سیستم، از ماژول Fernet که یکی از کلاس‌های این کتابخانه است، استفاده می‌شود. در استفاده از رمزنگاری، نگهداری کلید امنیتی بسیار اهمیت دارد، زیرا دسترسی به این کلید به معنی دسترسی به همه داده‌های رمزگذاری شده است. بنابراین، در صورت از دست دادن کلید یا دسترسی غیرمجاز به آن، امنیت داده‌ها به خطر می‌افتد.

نحوه ساختار کدهای برنامه

این برنامه شامل دو فایل است:

1- ftp_server.py:

این فایل کدهای مرتبط با سرور را در خود دارد و به‌طور کلی به درخواست‌های کلاینت پاسخ می‌دهد. وظیفه اصلی سرور، مدیریت فایل‌ها، پردازش درخواست‌ها و ارسال پاسخ‌ها به کلاینت است. همچنین سرور مسئولیت رمزنگاری و رمزگشایی فایل‌ها را نیز برعهده دارد.

2- ftp_client.py:

این فایل کدهای مربوط به کلاینت را شامل می‌شود. کلاینت امکان ارسال درخواست‌های آپلود، دانلود، حذف و جستجو را فراهم می‌کند. همچنین کلاینت لیست فایل‌های موجود در سرور را دریافت و نمایش می‌دهد.

ارتباط بین این دو ماژول بر اساس پروتکل TCP برقرار شده است که به عنوان یک پروتکل انتقال قابل اعتماد و پرکاربرد در شبکه‌های کامپیوتری شناخته می‌شود. در این سیستم، سرور روی یک آدرس IP و پورت خاص به درخواست‌های کلاینت گوش می‌دهد. این آدرس می‌تواند به صورت پیش‌فرض 0.0.0.0 باشد که دسترسی به سرور را از هر کجا در شبکه فراهم می‌کند، اما برای امنیت بیشتر می‌توان این آدرس را به یک IP مشخص محدود کرد.

نحوه راه‌اندازی سرور و کلاینت

مرحله اول: تولید و ذخیره کلید رمزنگاری

ابتدا نیاز به تولید یک کلید رمزنگاری Fernet است که هم در سرور و هم در کلاینت مورد استفاده قرار گیرد. این کلید به‌عنوان یکی از حساس‌ترین اطلاعات امنیتی برنامه باید به‌صورت امن ذخیره شود. برای تولید کلید می‌توان از کد زیر استفاده کرد:

```python
import socket
import os
from cryptography.fernet import Fernet


2 usages
def load_key():
    return b'your_key_here'  # Replace this with a secure key

```

مرحله دوم: راه‌اندازی سرور

این سرور به‌طور پیش‌فرض روی آدرس IP 0.0.0.0 و پورت 5557 تنظیم شده است که می‌تواند درخواست‌ها را از تمامی کلاینت‌های موجود در شبکه بپذیرد. پورت انتخاب شده باید در تنظیمات شبکه و فایروال شما باز باشد تا مشکلی در ارتباط به وجود نیاید.

نحوه اجرای هر دستور در سیستم

این سیستم از طریق کلاینت به کاربر امکان می‌دهد که با انتخاب عدد مربوط به هر دستور، درخواست موردنظر را به سرور ارسال کند. در ادامه توضیح دقیق‌تری درباره هر دستور آورده شده است.

دستور UPLOAD (آپلود فایل به سرور)

در این حالت، کاربر در کلاینت نام فایل را وارد کرده و کلاینت فایل را به سرور ارسال می‌کند. سرور فایل را دریافت کرده، آن را رمزگشایی و در پوشه‌ای که برای فایل‌ها مشخص شده ذخیره می‌کند. برای جلوگیری از مشکلاتی مانند خرابی داده در حین انتقال، فایل به بلوک‌های ۱۰۲۴ بایتی تقسیم شده و به تدریج ارسال و دریافت می‌شود. پس از اتمام عملیات، سرور پیامی مبنی بر موفقیت‌آمیز بودن آپلود به کلاینت باز می‌گرداند.

```
32          if command.startswith("UPLOAD"):
33              filename = command.split()[1]
34              filesize = int(client_socket.recv(1024).decode())
35              with open(filename, "wb") as f:
36                  bytes_received = 0
37                  while bytes_received < filesize:
38                      data = client_socket.recv(1024)
39                      if not data:
40                          break
41                      f.write(data)
42                      bytes_received += len(data)
43              decrypt_file(filename)
44              client_socket.send("UPLOAD_SUCCESS".encode())
45
```

دستور DOWNLOAD (دانلود فایل از سرور)

کاربر نام فایل موردنظر برای دانلود را وارد می‌کند و کلاینت درخواست دانلود را به سرور ارسال می‌کند. سرور پس از رمزگذاری فایل، آن را به کلاینت ارسال می‌کند. کلاینت فایل را دریافت و پس از رمزگشایی، آن را ذخیره می‌کند. همچنین در صورت عدم وجود فایل در سرور، سرور پیامی مبنی بر عدم وجود فایل ارسال می‌کند.

```
elif command.startswith("DOWNLOAD"):
    filename = command.split()[1]
    if os.path.exists(filename):
        client_socket.send("FILE_EXISTS".encode())
        filesize = os.path.getsize(filename)
        client_socket.send(str(filesize).encode())
        encrypt_file(filename)
        with open(filename, "rb") as f:
            while (chunk := f.read(1024)):
                client_socket.send(chunk)
        client_socket.send("DOWNLOAD_SUCCESS".encode())
    else:
        client_socket.send("FILE_NOT_FOUND".encode())
```

دستور DELETE (حذف فایل در سرور)

کاربر می‌تواند با وارد کردن نام فایل، دستور حذف را به سرور ارسال کند. سرور پس از دریافت دستور، اگر فایل مورد نظر وجود داشته باشد آن را حذف کرده و پیامی مبنی بر موفقیت‌آمیز بودن حذف فایل به کلاینت ارسال می‌کند.

```
elif command.startswith("DELETE"):
    filename = command.split()[1]
    try:
        os.remove(filename)
        client_socket.send("FILE_DELETED".encode())
    except FileNotFoundError:
        client_socket.send("FILE_NOT_FOUND".encode())
```

دستور SEARCH (جستجوی فایل در سرور)

با این دستور، کلاینت می‌تواند بررسی کند که آیا فایل مشخصی در سرور موجود است یا خیر. سرور فایل‌های موجود را جستجو کرده و اگر فایل پیدا شود، پیامی مبنی بر وجود آن و در غیر این صورت، پیامی مبنی بر عدم وجود آن ارسال می‌کند.

```
elif command.startswith("SEARCH"):
    filename = command.split()[1]
    if os.path.exists(filename):
        client_socket.send("FILE_FOUND".encode())
    else:
        client_socket.send("FILE_NOT_FOUND".encode())
```

دستور LIST (نمایش لیست فایل‌ها)

این دستور به کلاینت امکان مشاهده‌ی تمامی فایل‌های موجود در سرور را می‌دهد. سرور لیستی از فایل‌ها را به کلاینت ارسال کرده و کلاینت این لیست را نمایش می‌دهد. این دستور برای مشاهده وضعیت کلی فایل‌های موجود در سرور مفید است.

```
75        elif command == "LIST":
76            files = os.listdir('.')
77            files_list = '\n'.join(files)
78            client_socket.send(files_list.encode())
79
```

دستور EXIT (قطع ارتباط)

با انتخاب این دستور، ارتباط کلاینت و سرور به‌طور ایمن قطع می‌شود.

امنیت و عیب‌یابی

موارد امنیتی

در این سیستم، رمزنگاری نقش مهمی ایفا می‌کند و کلید رمزنگاری Fernet بخش حیاتی امنیت است. این کلید باید در جای امن نگهداری شود. علاوه بر این، توصیه می‌شود که تنظیمات فایروال طوری تنظیم شود که تنها کلاینت‌های مشخص به سرور دسترسی داشته باشند. همچنین، هرچند Fernet امنیت خوبی برای انتقال داده‌ها فراهم می‌کند، در صورت استفاده از این سیستم در محیط‌های حساس، بررسی‌های امنیتی بیشتری نیز لازم است.

عیب‌یابی

خطای پورت مشغول (WinError 10048): این خطا به دلیل اشغال بودن پورت توسط برنامه‌ای دیگر رخ می‌دهد. برای رفع آن می‌توان پورت دیگری انتخاب کرد.

خطای قطع اتصال ناگهانی (WinError 10053): این خطا می‌تواند به دلیل مشکلات شبکه یا تنظیمات فایروال رخ دهد. همچنین باید مطمئن شد که آدرس IP و پورت در هر دو طرف کلاینت و سرور یکسان باشند.

خطای دسترسی به فایل‌ها: در صورت بروز مشکلاتی در دسترسی به فایل‌ها، باید مطمئن شد که مسیر فایل‌ها و مجوزهای دسترسی در سیستم صحیح تنظیم شده‌اند.

نتیجه‌گیری

این فایل داکیومنت، شامل تمامی توضیحات و جزئیات مورد نیاز برای پیاده‌سازی است. این سیستم به‌خوبی می‌تواند به عنوان یک پروژه‌ی آموزشی در زمینه شبکه‌های کامپیوتری، رمزنگاری و امنیت اطلاعات مورد استفاده قرار گیرد و برای کاربردهای عملی در محیط‌های سازمانی و تیم‌های پروژه‌ای نیز مفید واقع شود.