Data networks

DR.Pakravan

electrical engineering department

Ahmadreza Majlesara 400101861

project

July 8, 2024

Data networks

project

Ahmadreza Majlesara 400101861

Part 2: FTP Protocol Understanding (15 points)

Answer the following questions thoroughly:

1. Investigate other protocols that are used for file transfer and compare them with FTP.

solution

- **FTP**: Simple and widely supported but lacks security features.
- **SFTP**: Secure and reliable, ideal for secure file transfers over a network.
- **FTPS**: Adds security to FTP but can be complicated to configure.
- **HTTP/HTTPS**: Great for web-based file transfers, with HTTPS providing security.
- **SCP**: Secure and straightforward for simple file transfers, especially in command-line environments.
- **SMB**: Advanced file sharing with support for complex operations and permissions, but more complex to set up and secure.

2. What is the commonly used transport protocol for file transfers? Is it possible to use UDP as the transport layer protocol?

solution

The most commonly used transport protocol for file transfers is the **Transmission Control Protocol (TCP)**. TCP is favored because it ensures reliable, ordered, and error-checked delivery of data. Key features include connection-oriented communication, flow control, congestion control, and error detection.
User Datagram Protocol (UDP) can be used for file transfers, but it is less common due to the following reasons:

- **Lack of Reliability**: UDP does not guarantee packet delivery, ordering, or error correction.
- **No Flow Control**: The sender can overwhelm the receiver without any control mechanisms.
- **Application Complexity**: Applications must handle reliability and order themselves.

UDP is used in scenarios where speed and low latency are more critical than reliability, such as:

- Streaming audio and video
- Online gaming
- Voice over IP (VoIP)
- Real-time applications where occasional data loss is acceptable

3. What are the drawbacks of FTP that have made it fairly obsolete on the modern web?

solution

Lack of Security:
- No encryption: Data, including passwords, is sent in plain text.
- Vulnerable to attacks: Susceptible to man-in-the-middle and packet sniffing attacks.

Authentication Issues:
- Weak authentication: Uses basic, insecure authentication methods.

Firewall and NAT Issues:
- Complex port management: Uses multiple ports, complicating firewall and NAT configurations.

Performance Limitations:
- Inefficiency: Not optimized for high-speed, high-volume transfers.

Limited Features:
- Lacks advanced features: No support for resumable downloads or file integrity checks.

User Experience:
- Complexity: Difficult to set up and use compared to modern protocols.

4. What are the disadvantages of using active mode FTP? How the passive mode can handle these problems?

solution

Disadvantages of Using Active Mode FTP:

Firewall Issues:
- **Client-Side Firewall**: Active mode FTP requires the client to open a random port to receive data connections from the server, which can be blocked by client-side firewalls.

NAT (Network Address Translation) Issues:
- **Client Behind NAT**: When a client is behind a NAT, the external server's attempt to initiate a connection back to the client can fail because the NAT may block the incoming connection.

Security Concerns:
- **Exposing Ports**: Active mode requires clients to open ports, which can expose the client to security risks and attacks on those ports.

How Passive Mode FTP Handles These Problems:

Firewall Compatibility:
- **Client-Side Firewall**: In passive mode FTP, the client initiates both the command and data connections. This ensures compatibility with client-side firewalls that block incoming connections, as only outbound connections are used.

NAT Compatibility:
- **Client Behind NAT**: Passive mode is more NAT-friendly because the client establishes both connections, eliminating the need for the server to initiate a connection back to the client.

Improved Security:
- **Reduced Exposure**: By having the client initiate all connections, passive mode reduces the need for clients to open ports, thus decreasing the attack surface and enhancing security.

5. In FTP, how is data transfer security guaranteed? Is there even a default security measure for FTP? Investigate SFTP, FTPS, and FTP over SSH protocols in terms of their security.

solution

Default FTP Security
- FTP does not provide inherent security.
- Data, including usernames and passwords, is sent in plain text, making it vulnerable to interception.

Enhanced Security Protocols

SFTP (SSH File Transfer Protocol)
- Operates over SSH.
- Provides encryption, robust authentication, and data integrity.
- Uses port 22.

FTPS (FTP Secure)
- Extends FTP with TLS/SSL encryption.
- Supports client and server certificates for authentication.
- Ensures data integrity.
- Uses port 21 (explicit) or a different port (implicit).

FTP over SSH
- Tunnels FTP through an SSH connection.
- Encrypts data and ensures integrity using SSH.
- Uses port 22.

6. With Wireshark, you can observe network packets traversing any network interface. Use Wireshark to investigate packets while you are downloading a file from your FTP server. Note that you should run FTP on the loopback interface. What is the maximum size of

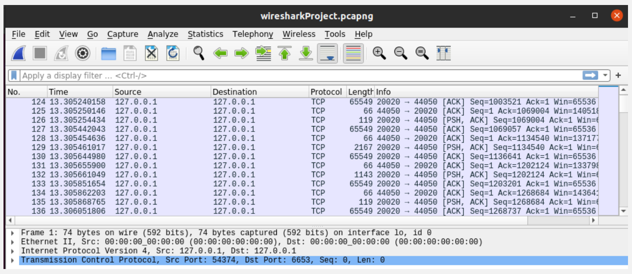
a TCP packet containing data? Use the following command on Linux-based machines to create an arbitrary large file:

```
# create a 10G file named file.txt
fallocate -l 10G file.txt
```

Include screenshots of packets captured by Wireshark in your report.

solution

the maximum packet size using TCP is **65549 byte**. the screenshot of captured packets are shown in figure 1


wiresharkProject.pcapng
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>

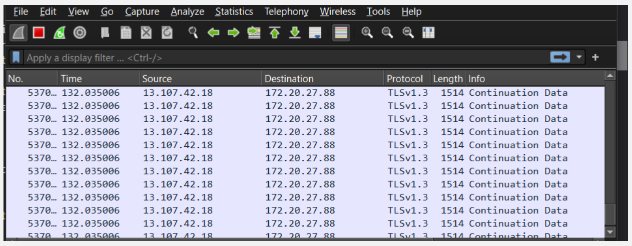
```
No.  Time          Source     Destination  Protocol Length Info
 124 13.305240158  127.0.0.1  127.0.0.1    TCP      65549 20020 → 44050 [ACK] Seq=1003521 Ack=1 Win=65536
 125 13.305250146  127.0.0.1  127.0.0.1    TCP         66 44050 → 20020 [ACK] Seq=1 Ack=1069004 Win=14051
 126 13.305254434  127.0.0.1  127.0.0.1    TCP        119 20020 → 44050 [PSH, ACK] Seq=1069004 Ack=1 Win=
 127 13.305442043  127.0.0.1  127.0.0.1    TCP      65549 20020 → 44050 [ACK] Seq=1069057 Ack=1 Win=65536
 128 13.305454636  127.0.0.1  127.0.0.1    TCP         66 44050 → 20020 [ACK] Seq=1 Ack=1134540 Win=13717
 129 13.305461017  127.0.0.1  127.0.0.1    TCP       2167 20020 → 44050 [PSH, ACK] Seq=1134540 Ack=1 Win=
 130 13.305644980  127.0.0.1  127.0.0.1    TCP      65549 20020 → 44050 [ACK] Seq=1136641 Ack=1 Win=65536
 131 13.305655900  127.0.0.1  127.0.0.1    TCP         66 44050 → 20020 [ACK] Seq=1 Ack=1202124 Win=13379
 132 13.305661049  127.0.0.1  127.0.0.1    TCP       1143 20020 → 44050 [PSH, ACK] Seq=1202124 Ack=1 Win=
 133 13.305851654  127.0.0.1  127.0.0.1    TCP      65549 20020 → 44050 [ACK] Seq=1203201 Ack=1 Win=65536
 134 13.305862203  127.0.0.1  127.0.0.1    TCP         66 44050 → 20020 [ACK] Seq=1 Ack=1268684 Win=14364
 135 13.305868765  127.0.0.1  127.0.0.1    TCP        119 20020 → 44050 [PSH, ACK] Seq=1268684 Ack=1 Win=
 136 13.306051806  127.0.0.1  127.0.0.1    TCP      65549 20020 → 44050 [ACK] Seq=1268737 Ack=1 Win=65536

Frame 1: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface lo, id 0
Ethernet II, Src: 00:00:00_00:00:00 (00:00:00:00:00:00), Dst: 00:00:00_00:00:00 (00:00:00:00:00:00)
Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
Transmission Control Protocol, Src Port: 54374, Dst Port: 6653, Seq: 0, Len: 0
```

Figure 1. screenshot of packets captured by wireshark

7. Did you know that Sharif University hosts an FTP server where you can download useful content like engineering programs, drivers, and so on? To visit it, first, you need to set up your Sharif VPN to access it when you are not on campus. Then, refer to this website and you can download tools you might need. While downloading a file from Sharif FTP, run Wireshark and observe the received packets. Are the captured packets similar to the previous part? If not, explain the difference.

solution


File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>

```
No.     Time        Source        Destination   Protocol Length Info
5370… 132.035006  13.107.42.18  172.20.27.88  TLSv1.3  1514 Continuation Data
5370… 132.035006  13.107.42.18  172.20.27.88  TLSv1.3  1514 Continuation Data
5370… 132.035006  13.107.42.18  172.20.27.88  TLSv1.3  1514 Continuation Data
5370… 132.035006  13.107.42.18  172.20.27.88  TLSv1.3  1514 Continuation Data
5370… 132.035006  13.107.42.18  172.20.27.88  TLSv1.3  1514 Continuation Data
5370… 132.035006  13.107.42.18  172.20.27.88  TLSv1.3  1514 Continuation Data
5370… 132.035006  13.107.42.18  172.20.27.88  TLSv1.3  1514 Continuation Data
5370… 132.035006  13.107.42.18  172.20.27.88  TLSv1.3  1514 Continuation Data
5370… 132.035006  13.107.42.18  172.20.27.88  TLSv1.3  1514 Continuation Data
5370… 132.035006  13.107.42.18  172.20.27.88  TLSv1.3  1514 Continuation Data
5370… 132.035006  13.107.42.18  172.20.27.88  TLSv1.3  1514 Continuation Data
5370… 132.035006  13.107.42.18  172.20.27.88  TLSv1.3  1514 Continuation Data
```

Figure 2. screenshot of packets captured by wireshark downloading from ftp Sharif

as we can see in the above figure the packets are limited to 1514 bytes using ftp sharif. also the length of each packet is the same but in our server the length of packet is sometimes lower than the maximum size. it means that ftp sharif have a lower fixed bandwith to prevent getting so much transfer rate from it and make it vulnerable. on the other hand it gives fixed byte rate for the purpose of QOS.

Part 3: Setting Up a Local FTP Server on Ubuntu (20 points)

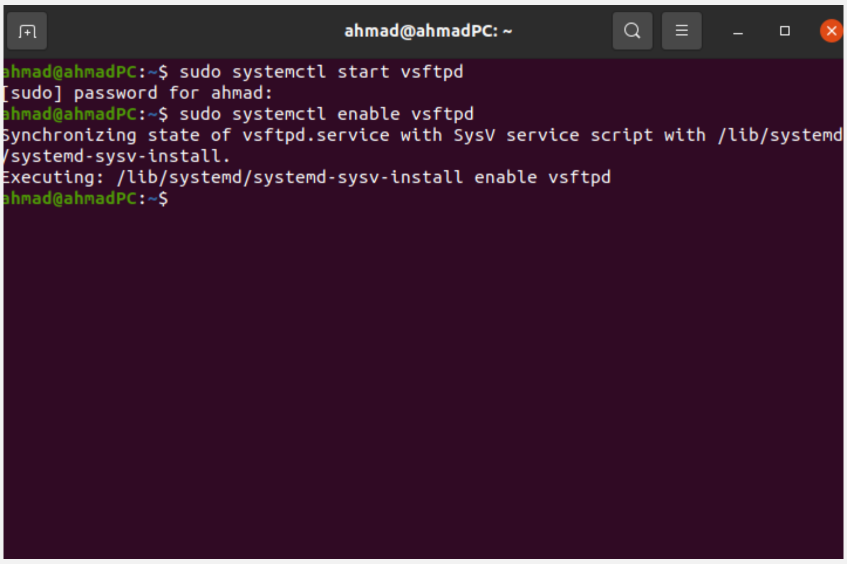
In this part, we will set up an FTP server using `vsftpd` on Ubuntu, configure users, and secure the connection.

Part A: Install and Configure FTP Server

1. Install the `vsftpd` package. After installing `vsftpd`, configure it to start the FTP server.
2. Create three dedicated FTP users and set up passwords for them.
3. Connect to the FTP server using an FTP client or command line, and take screenshots of your commands and results.

solution

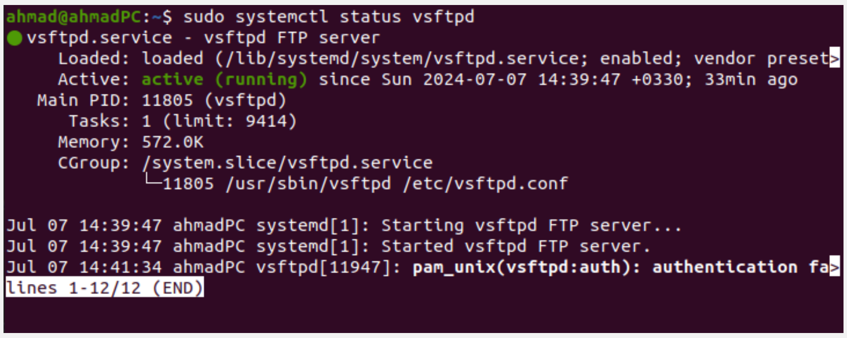
first, we need to start the `vsftpd` package. To do this, we run the following command:



```
ahmad@ahmadPC: ~
ahmad@ahmadPC:~$ sudo systemctl start vsftpd
[sudo] password for ahmad:
ahmad@ahmadPC:~$ sudo systemctl enable vsftpd
Synchronizing state of vsftpd.service with SysV service script with /lib/systemd
/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable vsftpd
ahmad@ahmadPC:~$
```

Figure 3. starting `vsftpd` package

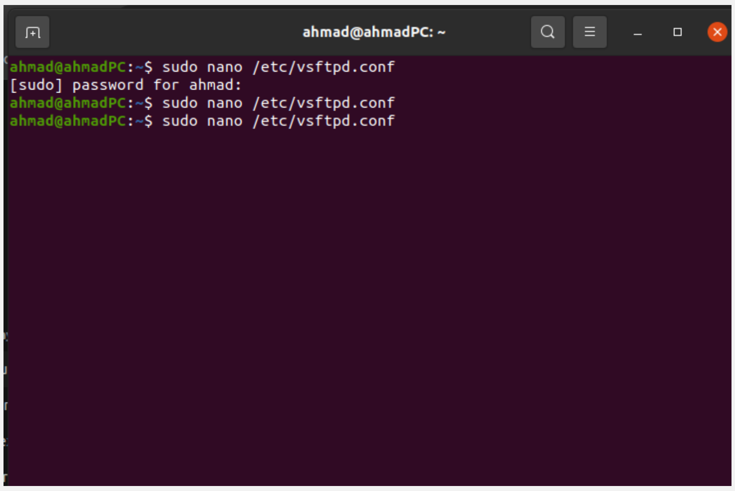
then to verify if the server is running we use the following command:



```
ahmad@ahmadPC:~$ sudo systemctl status vsftpd
● vsftpd.service - vsftpd FTP server
     Loaded: loaded (/lib/systemd/system/vsftpd.service; enabled; vendor preset>
     Active: active (running) since Sun 2024-07-07 14:39:47 +0330; 33min ago
   Main PID: 11805 (vsftpd)
      Tasks: 1 (limit: 9414)
     Memory: 572.0K
     CGroup: /system.slice/vsftpd.service
             └─11805 /usr/sbin/vsftpd /etc/vsftpd.conf

Jul 07 14:39:47 ahmadPC systemd[1]: Starting vsftpd FTP server...
Jul 07 14:39:47 ahmadPC systemd[1]: Started vsftpd FTP server.
Jul 07 14:41:34 ahmadPC vsftpd[11947]: pam_unix(vsftpd:auth): authentication fa>
lines 1-12/12 (END)
```

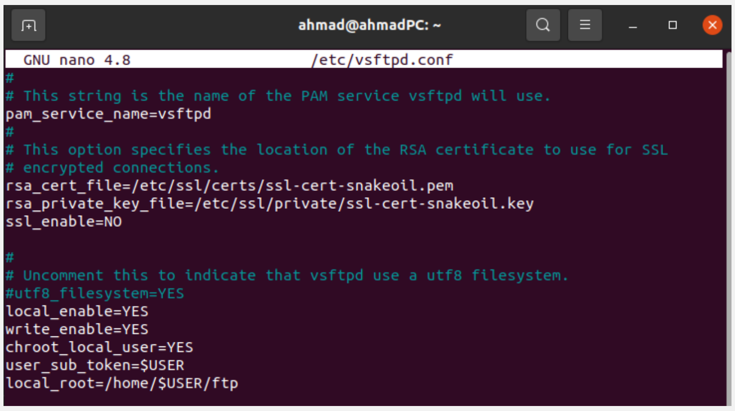
Figure 4. verifying the server is running after that we have to configure the server to start the FTP server. To do this, we run the following command:



```
ahmad@ahmadPC: ~
ahmad@ahmadPC:~$ sudo nano /etc/vsftpd.conf
[sudo] password for ahmad:
ahmad@ahmadPC:~$ sudo nano /etc/vsftpd.conf
ahmad@ahmadPC:~$ sudo nano /etc/vsftpd.conf
```

Figure 5. configuring the server to start the FTP server

the config file will look like this after configuration:

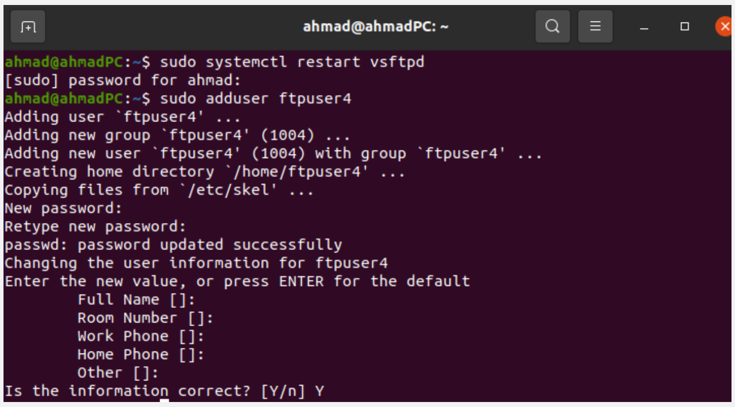


```
ahmad@ahmadPC: ~
  GNU nano 4.8                    /etc/vsftpd.conf
#
# This string is the name of the PAM service vsftpd will use.
pam_service_name=vsftpd
#
# This option specifies the location of the RSA certificate to use for SSL
# encrypted connections.
rsa_cert_file=/etc/ssl/certs/ssl-cert-snakeoil.pem
rsa_private_key_file=/etc/ssl/private/ssl-cert-snakeoil.key
ssl_enable=NO

#
# Uncomment this to indicate that vsftpd use a utf8 filesystem.
#utf8_filesystem=YES
local_enable=YES
write_enable=YES
chroot_local_user=YES
user_sub_token=$USER
local_root=/home/$USER/ftp
```

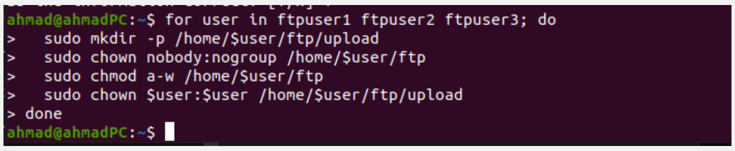
Figure 6. config file after configuration

now to create 3 dedicated FTP users we run the following command:



```
ahmad@ahmadPC: ~
ahmad@ahmadPC:~$ sudo systemctl restart vsftpd
[sudo] password for ahmad:
ahmad@ahmadPC:~$ sudo adduser ftpuser4
Adding user `ftpuser4' ...
Adding new group `ftpuser4' (1004) ...
Adding new user `ftpuser4' (1004) with group `ftpuser4' ...
Creating home directory `/home/ftpuser4' ...
Copying files from `/etc/skel' ...
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for ftpuser4
Enter the new value, or press ENTER for the default
        Full Name []:
        Room Number []:
        Work Phone []:
        Home Phone []:
        Other []:
Is the information correct? [Y/n] Y
```

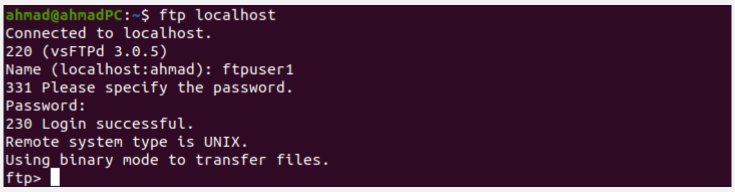
Figure 7. creating 3 dedicated FTP users after that we run the following code to create ftp directories and set permissions:



```
ahmad@ahmadPC:~$ for user in ftpuser1 ftpuser2 ftpuser3; do
>   sudo mkdir -p /home/$user/ftp/upload
>   sudo chown nobody:nogroup /home/$user/ftp
>   sudo chmod a-w /home/$user/ftp
>   sudo chown $user:$user /home/$user/ftp/upload
> done
ahmad@ahmadPC:~$ 
```

Figure 8. creating ftp directories and setting permissions

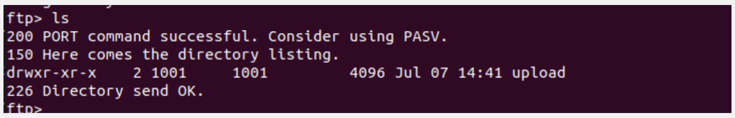
after that we are ready to connect to the FTP server using an FTP client or command line. To do this we run the following command:



```
ahmad@ahmadPC:~$ ftp localhost
Connected to localhost.
220 (vsFTPd 3.0.5)
Name (localhost:ahmad): ftpuser1
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> 
```

Figure 9. connecting to the FTP server

at last we can see the result of ls command in the following figure:



```
ftp> ls
200 PORT command successful. Consider using PASV.
150 Here comes the directory listing.
drwxr-xr-x    2 1001     1001         4096 Jul 07 14:41 upload
226 Directory send OK.
ftp>
```

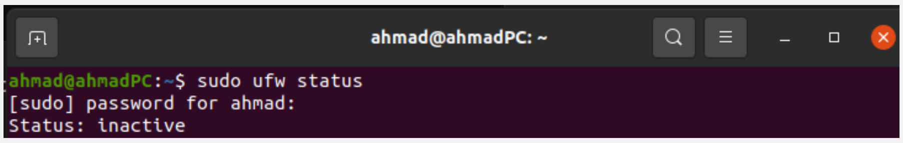
Figure 10. result of ls command

Part B: Configure Firewall

1. Verify if the firewall is active on your system.
2. Configure the firewall to allow FTP traffic.

solution

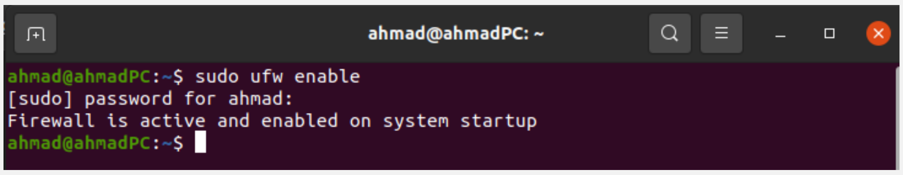
first we use the following code to check if the firewall is active:



```
ahmad@ahmadPC:~$ sudo ufw status
[sudo] password for ahmad:
Status: inactive
```

Figure 11. checking if the firewall is active

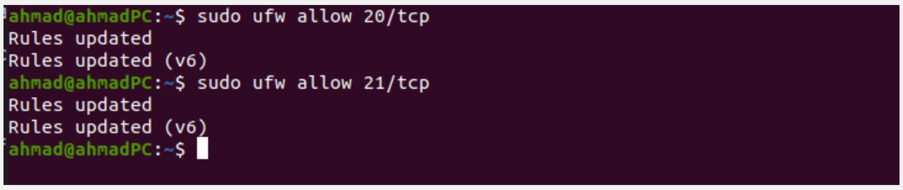
as we can see the firewall is inactive. so first we activate that using the following code:



```
ahmad@ahmadPC:~$ sudo ufw enable
[sudo] password for ahmad:
Firewall is active and enabled on system startup
ahmad@ahmadPC:~$
```

Figure 12. activating the firewall

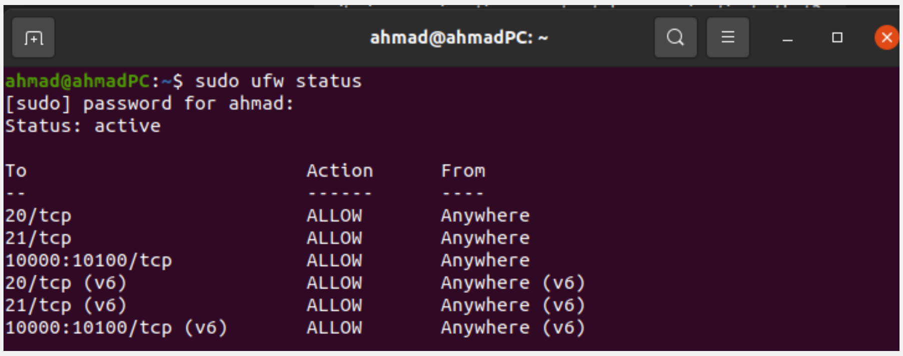
then as we can see the firewall is active. now we have to configure the firewall to allow FTP traffic. To do this we run the following code:



```
ahmad@ahmadPC:~$ sudo ufw allow 20/tcp
Rules updated
Rules updated (v6)
ahmad@ahmadPC:~$ sudo ufw allow 21/tcp
Rules updated
Rules updated (v6)
ahmad@ahmadPC:~$
```

Figure 13. configuring the firewall to allow FTP traffic

and now as we can see in the firewall status the FTP traffic is allowed:



```
ahmad@ahmadPC:~$ sudo ufw status
[sudo] password for ahmad:
Status: active

To                         Action      From
--                         ------      ----
20/tcp                     ALLOW       Anywhere
21/tcp                     ALLOW       Anywhere
10000:10100/tcp            ALLOW       Anywhere
20/tcp (v6)                ALLOW       Anywhere (v6)
21/tcp (v6)                ALLOW       Anywhere (v6)
10000:10100/tcp (v6)       ALLOW       Anywhere (v6)
```

Figure 14. FTP traffic is allowed

Questions:

- Why is it important to configure firewall rules for FTP traffic?
- What are the specific ports used for FTP, and how do you open them in the firewall?

solution

Why is it important to configure firewall rules for FTP traffic?

- **Security**: Protects the server from unauthorized access and attacks.
- **Functionality**: Ensures the FTP server can communicate effectively with clients.
- **Network Management**: Manages and controls network traffic to allow only legitimate FTP traffic.

What are the specific ports used for FTP, and how do you open them in the firewall?

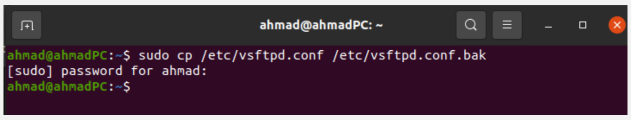
- **Port 21**: Control port for FTP commands.
- **Port 20**: Data port for active mode transfers.
- **Passive Mode Ports**: Range (e.g., 10000-10100) for passive mode transfers.

Part C: Change Default Directory

1. Before making any changes, back up your `vsftpd` configuration files. How can you back up the configuration files?

solution

first we use the below code to backup the configuration files:



```
ahmad@ahmadPC:~$ sudo cp /etc/vsftpd.conf /etc/vsftpd.conf.bak
[sudo] password for ahmad:
ahmad@ahmadPC:~$
```

Figure 15. backing up the configuration files

Question: Why is it important to back up configuration files before editing them?

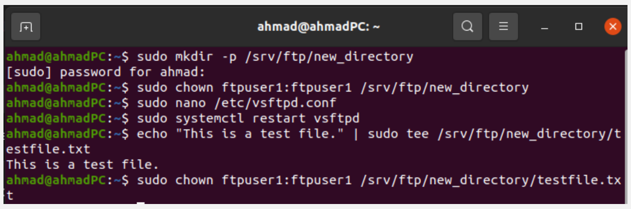
solution

- **Reversibility**: Allows reverting to the original configuration if something goes wrong.
- **Stability**: Ensures quick restoration to a known good state, minimizing downtime.
- **Safety**: Protects against accidental data loss or misconfiguration.

2. Modify the `vsftpd` configuration to change the default directory for FTP users. Create a new directory for the FTP server.

solution

we use the following code to change the default directory for FTP users and creating a test file in that directory:



```
ahmad@ahmadPC:~$ sudo mkdir -p /srv/ftp/new_directory
[sudo] password for ahmad:
ahmad@ahmadPC:~$ sudo chown ftpuser1:ftpuser1 /srv/ftp/new_directory
ahmad@ahmadPC:~$ sudo nano /etc/vsftpd.conf
ahmad@ahmadPC:~$ sudo systemctl restart vsftpd
ahmad@ahmadPC:~$ echo "This is a test file." | sudo tee /srv/ftp/new_directory/t
estfile.txt
This is a test file.
ahmad@ahmadPC:~$ sudo chown ftpuser1:ftpuser1 /srv/ftp/new_directory/testfile.tx
t
```

Figure 16. changing the default directory for FTP users and creating a test file

Question: Why is it beneficial to change the default directory for FTP users?

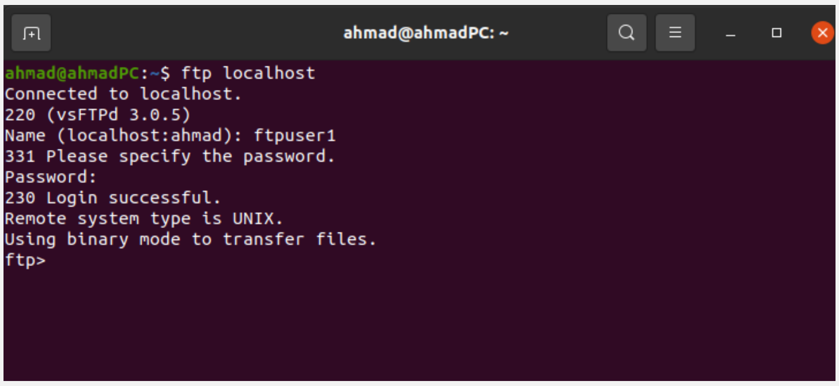
solution

- **Security**: Limits access to specific directories, reducing the risk of unauthorized access.
- **Organization**: Helps organize files and directories for better management.
- **Resource Management**: Allows tailoring the FTP server to specific user needs or requirements.

3. Create a new file in the new directory and check if it is accessible from the FTP client.

solution

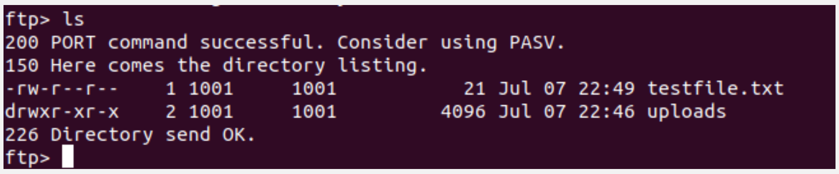
first we use the following code to run the ftp protocol on our local host which we configured in a new directory::



```
ahmad@ahmadPC:~$ ftp localhost
Connected to localhost.
220 (vsFTPd 3.0.5)
Name (localhost:ahmad): ftpuser1
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp>
```

Figure 17. running the ftp protocol on our local host

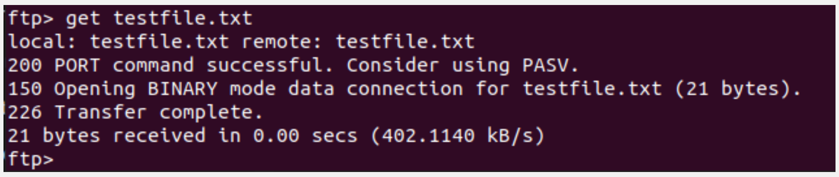
then we use the following code to check if the new file is accessible from the FTP client:



```
ftp> ls
200 PORT command successful. Consider using PASV.
150 Here comes the directory listing.
-rw-r--r--    1 1001     1001           21 Jul 07 22:49 testfile.txt
drwxr-xr-x    2 1001     1001         4096 Jul 07 22:46 uploads
226 Directory send OK.
ftp> 
```

Figure 18. checking if the new file is accessible from the FTP client

and at last we use get command to get the file from the server:



```
ftp> get testfile.txt
local: testfile.txt remote: testfile.txt
200 PORT command successful. Consider using PASV.
150 Opening BINARY mode data connection for testfile.txt (21 bytes).
226 Transfer complete.
21 bytes received in 0.00 secs (402.1140 kB/s)
ftp>
```

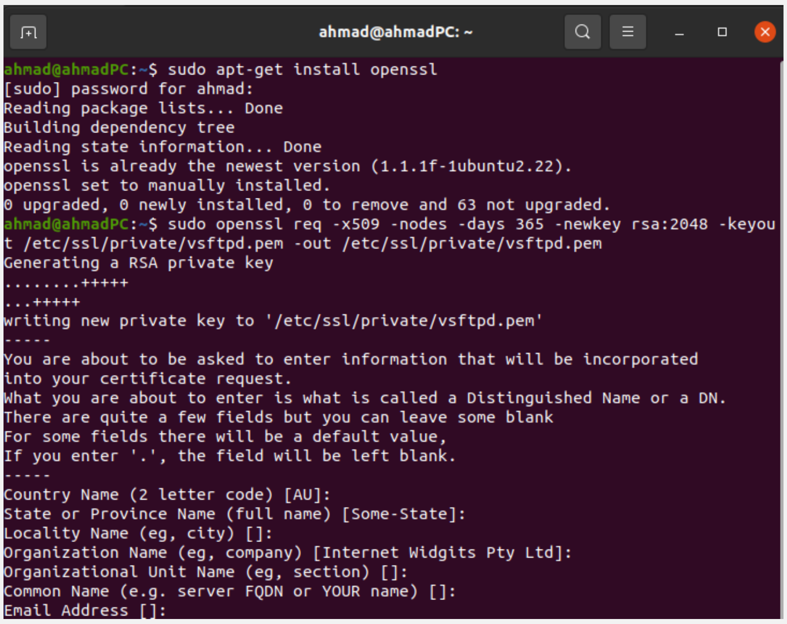
Figure 19. getting the file from the server

Part D: Securing FTP

1. **Encrypt FTP Traffic:** Configure the FTP server to use SSL/TLS encryption to secure the connection (provide code, but do not implement it).

solution

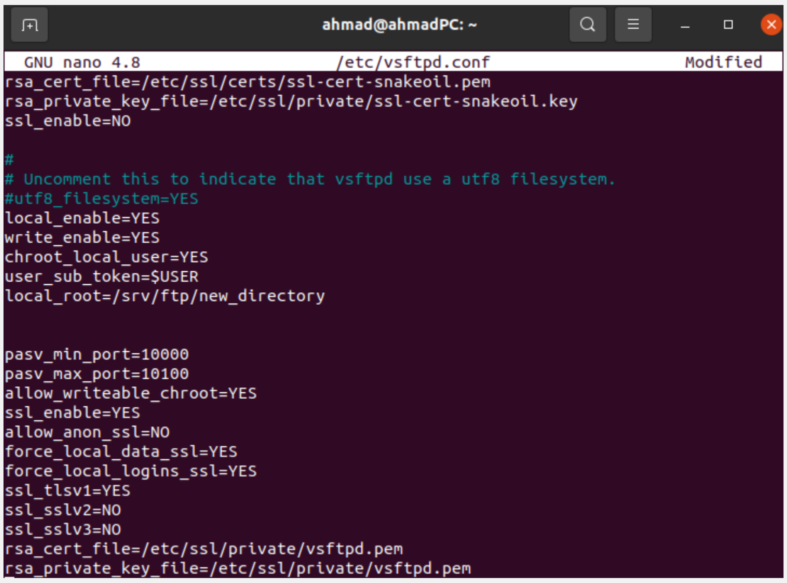
first we Install the SSL/TLS package and generate certificates using the following code:



```
ahmad@ahmadPC:~$ sudo apt-get install openssl
[sudo] password for ahmad:
Reading package lists... Done
Building dependency tree
Reading state information... Done
openssl is already the newest version (1.1.1f-1ubuntu2.22).
openssl set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 63 not upgraded.
ahmad@ahmadPC:~$ sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyou
t /etc/ssl/private/vsftpd.pem -out /etc/ssl/private/vsftpd.pem
Generating a RSA private key
........+++++
...+++++
writing new private key to '/etc/ssl/private/vsftpd.pem'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:
State or Province Name (full name) [Some-State]:
Locality Name (eg, city) []:
Organization Name (eg, company) [Internet Widgits Pty Ltd]:
Organizational Unit Name (eg, section) []:
Common Name (e.g. server FQDN or YOUR name) []:
Email Address []:
```

Figure 20. installing the SSL/TLS package and generating certificates

then we configure the FTP server to use SSL/TLS encryption using the following code:



```
  GNU nano 4.8                /etc/vsftpd.conf                    Modified
rsa_cert_file=/etc/ssl/certs/ssl-cert-snakeoil.pem
rsa_private_key_file=/etc/ssl/private/ssl-cert-snakeoil.key
ssl_enable=NO

#
# Uncomment this to indicate that vsftpd use a utf8 filesystem.
#utf8_filesystem=YES
local_enable=YES
write_enable=YES
chroot_local_user=YES
user_sub_token=$USER
local_root=/srv/ftp/new_directory


pasv_min_port=10000
pasv_max_port=10100
allow_writeable_chroot=YES
ssl_enable=YES
allow_anon_ssl=NO
force_local_data_ssl=YES
force_local_logins_ssl=YES
ssl_tlsv1=YES
ssl_sslv2=NO
ssl_sslv3=NO
rsa_cert_file=/etc/ssl/private/vsftpd.pem
rsa_private_key_file=/etc/ssl/private/vsftpd.pem
```

Figure 21. configuring the FTP server to use SSL/TLS encryption

Questions:

- Why is it important to encrypt FTP traffic?
- What are the benefits of using SSL/TLS encryption for FTP?

solution

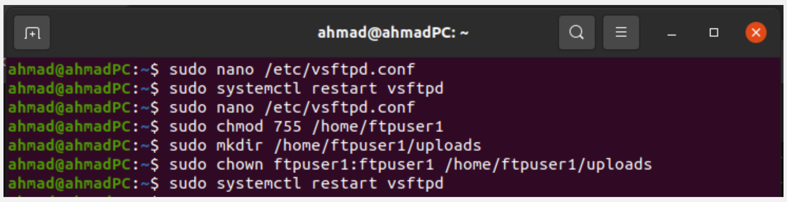
Why is it important to encrypt FTP traffic?
- **Data Protection**: Protects sensitive data from eavesdropping and unauthorized access.
- **Integrity**: Ensures data is not tampered with during transmission.
- **Authentication**: Verifies the identity of the server and client

What are the benefits of using SSL/TLS encryption for FTP?
- **Data Protection**: Protects data from eavesdropping and tampering.
- **Authentication**: Prevents man-in-the-middle attacks.
- **Compliance**: Meets regulatory requirements for data protection.

2. **Limit User Access:** Implement measures to limit user access to specific directories and functionalities to enhance security.

solution

for this part we use chroot jail to limit user access to specific directories and functionalities. we use the following code to implement this:



```
ahmad@ahmadPC:~$ sudo nano /etc/vsftpd.conf
ahmad@ahmadPC:~$ sudo systemctl restart vsftpd
ahmad@ahmadPC:~$ sudo nano /etc/vsftpd.conf
ahmad@ahmadPC:~$ sudo chmod 755 /home/ftpuser1
ahmad@ahmadPC:~$ sudo mkdir /home/ftpuser1/uploads
ahmad@ahmadPC:~$ sudo chown ftpuser1:ftpuser1 /home/ftpuser1/uploads
ahmad@ahmadPC:~$ sudo systemctl restart vsftpd
```

Figure 22. limiting user access to specific directories and functionalities

Questions:

- Why do we need to secure FTP servers?
- What are the methods to limit user access on an FTP server?

solution

Why do we need to secure FTP servers?
- **Prevent Unauthorized Access**: Protects sensitive data from unauthorized users.
- **Protect Data Integrity**: Ensures data is not altered or corrupted during transfer.
- **Comply with Regulations**: Meets legal and regulatory requirements for data protection.

What are the methods to limit user access on an FTP server?

- **Chroot Jail**: Restricts users to their home directories.
- **User Permissions**: Sets strict permissions on files and directories.
- **Restrict FTP Commands**: Limits the FTP commands available to users.

Part 4: Bandwidth and Transfer Rate Control (15 points Bonus)

In this part you will Implement bandwidth and transfer rate control in your FTP server using Python to ensure that file transfers do not exceed a specified rate. This will help in managing network resources more effectively and prevent the server from being overwhelmed by high-speed data transfers.

Question: Why is bandwidth control important in network applications?

solution

Bandwidth control is crucial in network applications for several reasons:

- **Network Stability and Performance**: Prevents network congestion and ensures consistent performance by managing bandwidth usage.
- **Quality of Service (QoS)**: Allows prioritization of critical applications and guarantees service levels for different types of traffic.
- **Security and Fair Usage**: Prevents network abuse, ensures fair usage policies, and provides equitable access to resources.
- **Improved User Experience**: Reduces latency and jitter for real-time applications, ensuring smooth data transfers.

Tasks

1. **Measurement of Transfer Rate:**
   - Track the amount of data being transferred and the time taken to calculate the current transfer rate.
   - Use this information to monitor the transfer speed in real-time. (print the progress percentage of transmission and transmission rate)
2. **Limiting Bandwidth:**
   - Introduce a maximum bandwidth limit (e.g., 100 KB/s).
   - If the transfer rate exceeds this limit, introduce a delay to throttle the speed of data transfer.
3. **Implementation Steps:**
   - Modify the server's file sending function to incorporate transfer rate monitoring and bandwidth throttling.
   - Ensure the client handles potentially slower data transfer rates smoothly.

Implementation Details

Server Side:

- Track the number of bytes sent and the elapsed time during file transfer.

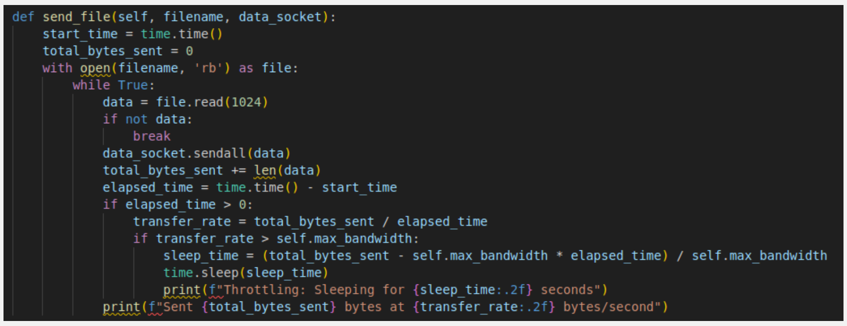
- Calculate the transfer rate and introduce delays if it exceeds the maximum bandwidth limit.

Client Side:

- The client will request a file and handle data reception as normal. No significant changes are required on the client side, but it should handle slower data reception gracefully.

solution

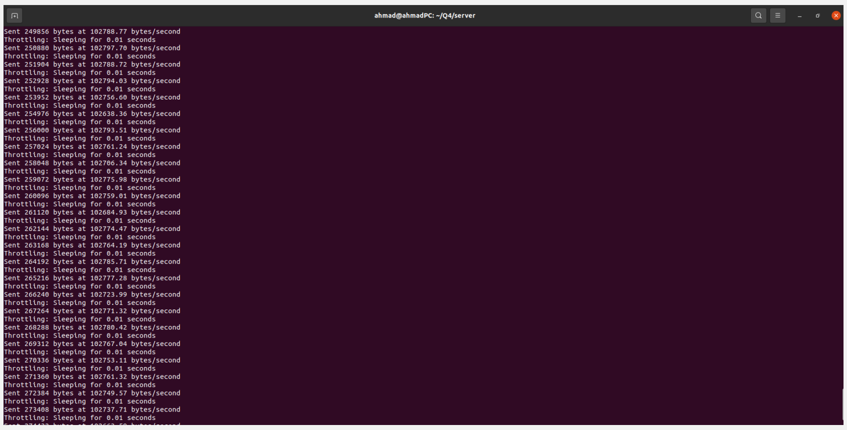
in this part as mentioned in the question i just updated the server side code to limit the bandwidth and transfer rate control. the client side code is the same as the previous part. the server side code is as follows:



```python
def send_file(self, filename, data_socket):
    start_time = time.time()
    total_bytes_sent = 0
    with open(filename, 'rb') as file:
        while True:
            data = file.read(1024)
            if not data:
                break
            data_socket.sendall(data)
            total_bytes_sent += len(data)
            elapsed_time = time.time() - start_time
            if elapsed_time > 0:
                transfer_rate = total_bytes_sent / elapsed_time
                if transfer_rate > self.max_bandwidth:
                    sleep_time = (total_bytes_sent - self.max_bandwidth * elapsed_time) / self.max_bandwidth
                    time.sleep(sleep_time)
                    print(f"Throttling: Sleeping for {sleep_time:.2f} seconds")
            print(f"Sent {total_bytes_sent} bytes at {transfer_rate:.2f} bytes/second")
```

Figure 23. Server side code for bandwidth and transfer rate control

as it can bee seen in this figure, the updated code get an argument as maxbandwidth. in each data transfer, it calculates the time taken to transfer the data and the amount of data transferred. then it calculates the transfer rate and if it exceeds the maxbandwidth, it introduces a delay to throttle the speed of data transfer. this process ensures that if the data transfer rate exceeds the specified limit, the server will slow down the data transfer to maintain the bandwidth limit. the result of this code is as follows:



```
Sent 249856 bytes at 102788.77 bytes/second
Throttling: Sleeping for 0.01 seconds
Sent 250880 bytes at 102797.70 bytes/second
Throttling: Sleeping for 0.01 seconds
Sent 251904 bytes at 102788.72 bytes/second
Throttling: Sleeping for 0.01 seconds
Sent 252928 bytes at 102794.03 bytes/second
Throttling: Sleeping for 0.01 seconds
Sent 253952 bytes at 102756.60 bytes/second
Throttling: Sleeping for 0.01 seconds
Sent 254976 bytes at 102638.36 bytes/second
Throttling: Sleeping for 0.01 seconds
Sent 256000 bytes at 102793.51 bytes/second
Throttling: Sleeping for 0.01 seconds
Sent 257024 bytes at 102761.24 bytes/second
Throttling: Sleeping for 0.01 seconds
Sent 258048 bytes at 102706.34 bytes/second
Throttling: Sleeping for 0.01 seconds
Sent 259072 bytes at 102775.98 bytes/second
Throttling: Sleeping for 0.01 seconds
Sent 260096 bytes at 102759.01 bytes/second
Throttling: Sleeping for 0.01 seconds
Sent 261120 bytes at 102684.93 bytes/second
Throttling: Sleeping for 0.01 seconds
Sent 262144 bytes at 102774.47 bytes/second
Throttling: Sleeping for 0.01 seconds
Sent 263168 bytes at 102764.19 bytes/second
Throttling: Sleeping for 0.01 seconds
Sent 264192 bytes at 102785.71 bytes/second
Throttling: Sleeping for 0.01 seconds
Sent 265216 bytes at 102777.28 bytes/second
Throttling: Sleeping for 0.01 seconds
Sent 266240 bytes at 102723.99 bytes/second
Throttling: Sleeping for 0.01 seconds
Sent 267264 bytes at 102771.32 bytes/second
Throttling: Sleeping for 0.01 seconds
Sent 268288 bytes at 102780.42 bytes/second
Throttling: Sleeping for 0.01 seconds
Sent 269312 bytes at 102767.04 bytes/second
Throttling: Sleeping for 0.01 seconds
Sent 270336 bytes at 102753.11 bytes/second
Throttling: Sleeping for 0.01 seconds
Sent 271360 bytes at 102761.32 bytes/second
Throttling: Sleeping for 0.01 seconds
Sent 272384 bytes at 102749.57 bytes/second
Throttling: Sleeping for 0.01 seconds
Sent 273408 bytes at 102737.71 bytes/second
Throttling: Sleeping for 0.01 seconds
```

Figure 24. Result of the server side code for bandwidth and transfer rate control

the files are transfered properly to the client but the time for that is longer than Q1.