دانشکده مهندسی برق

شبکه مخابرات داده

دکتر پاکروان

پروژه FTP

پارسا حاتمی

400100962

Part 1: Client and Server Implementation

با توجه به اینکه در فایل پروژه نوشته شده است این بخش نیازی به نوشتن گزارش ندارد.

کد های لازم برای این بخش شامل کد سرور و کلاینت در پوشه پروژه قرار داده شده است.

تمامی کامند های ذکر شده مطابق فایل پروژه پیاده سازی شده است و به درستی عمل میکند.

Part 2: FTP Protocol Understanding

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

Answer:

FTP is one of the oldest and most well-known file transfer protocols, but it has some significant drawbacks, especially regarding security. some other protocols:

- *SFTP (SSH File Transfer Protocol): This protocol operates over SSH and is known for its robust security features. It encrypts both commands and data, which makes it highly secure. SFTP typically uses port 22 and is favored for its straightforward configuration and single port usage, making it easier to navigate through firewalls compared to FTP.*
- *FTPS (FTP Secure): FTPS enhances the traditional FTP protocol by adding support for SSL/TLS encryption. It can use ports 21 for control and 990 for implicit mode, but it can be more complicated to configure due to the need for multiple ports. Nevertheless, FTPS is widely supported and adds a necessary layer of security to FTP.*
- *HTTP/HTTPS (Hypertext Transfer Protocol Secure): HTTPS is particularly useful for web-based file transfers and API interactions. Using port 443, it provides a secure and encrypted way to transfer data over the web. HTTP/HTTPS are easy to use and integrate well with modern web applications.*
- *SMB (Server Message Block): Commonly used for network file sharing, SMB supports complex operations and detailed permissions. SMB3 includes encryption for data in transit, providing robust security. However, SMB can be more challenging to set up and is primarily used within local networks, particularly in Windows environments.*

Each of these protocols offers unique advantages over FTP, particularly in terms of security and ease of configuration.

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

Answer:

The Transmission Control Protocol (TCP) is the predominant transport protocol used for file transfers. It ensures reliable, ordered, and error-checked delivery of data, which is crucial for the integrity of file transfers. TCP's features such as connection-oriented communication, flow control, congestion control, and error detection make it ideal for this purpose.

While it is technically possible to use the User Datagram Protocol (UDP) for file transfers, it is not commonly done due to several limitations. UDP lacks built-in reliability mechanisms, meaning there is no guarantee that packets will be delivered in order, or even at all. This can result in data loss or corruption, making UDP unsuitable for most file transfer needs. However, UDP is preferred in scenarios where speed and low latency are more critical than reliability, such as in streaming, online gaming, and VoIP.

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

Answer:

***Security Vulnerabilities**: FTP transmits data in plaintext, including usernames and passwords, making it susceptible to eavesdropping and man-in-the-middle attacks.*

***Firewall and NAT Issues**: FTP's requirement for multiple ports (control and data) complicates firewall and NAT configurations, leading to connectivity problems.*

***Complex Configuration**: Setting up and managing FTP servers and clients can be difficult, especially when considering the differences between active and passive modes.*

***Lack of Advanced Features**: FTP does not support advanced features such as resumable downloads or file integrity checks, which are standard in more modern protocols.*

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

Answer:

***Active Mode FTP** has several disadvantages:*

- ***Firewall and NAT Issues**: Active mode requires the client to open a random port to receive data connections from the server. This can be problematic as client-side firewalls and NAT devices may block these incoming connections.*
- ***Security Risks**: The necessity for clients to open ports exposes them to potential security vulnerabilities.*

***Passive Mode FTP**, on the other hand, mitigates these issues:*

- ***Improved Firewall Compatibility**: In passive mode, the server opens a port and the client initiates the connection, making it easier to pass through firewalls and NAT devices.*
- ***Enhanced Security**: By having the client initiate all connections, passive mode reduces the exposure of client-side ports, thereby decreasing security risks.*

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.

Answer:

***Default FTP Security**: FTP lacks inherent security measures, transmitting data in plaintext without encryption. This makes it vulnerable to interception and unauthorized access.*

***Enhanced Security Protocols**:*

- ***SFTP (SSH File Transfer Protocol)**: SFTP provides secure file transfer through SSH, offering encryption and robust authentication. It operates over port 22 and is known for its strong security features and ease of use.*
- ***FTPS (FTP Secure)**: By extending FTP with SSL/TLS encryption, FTPS secures data transfers and supports the use of client and server certificates for authentication. It typically uses ports 21 and 990.*
- ***FTP over SSH**: This method tunnels FTP through an SSH connection, encrypting the data and ensuring integrity. Although it uses port 22, it can be more complex to set up compared to SFTP and FTPS.*

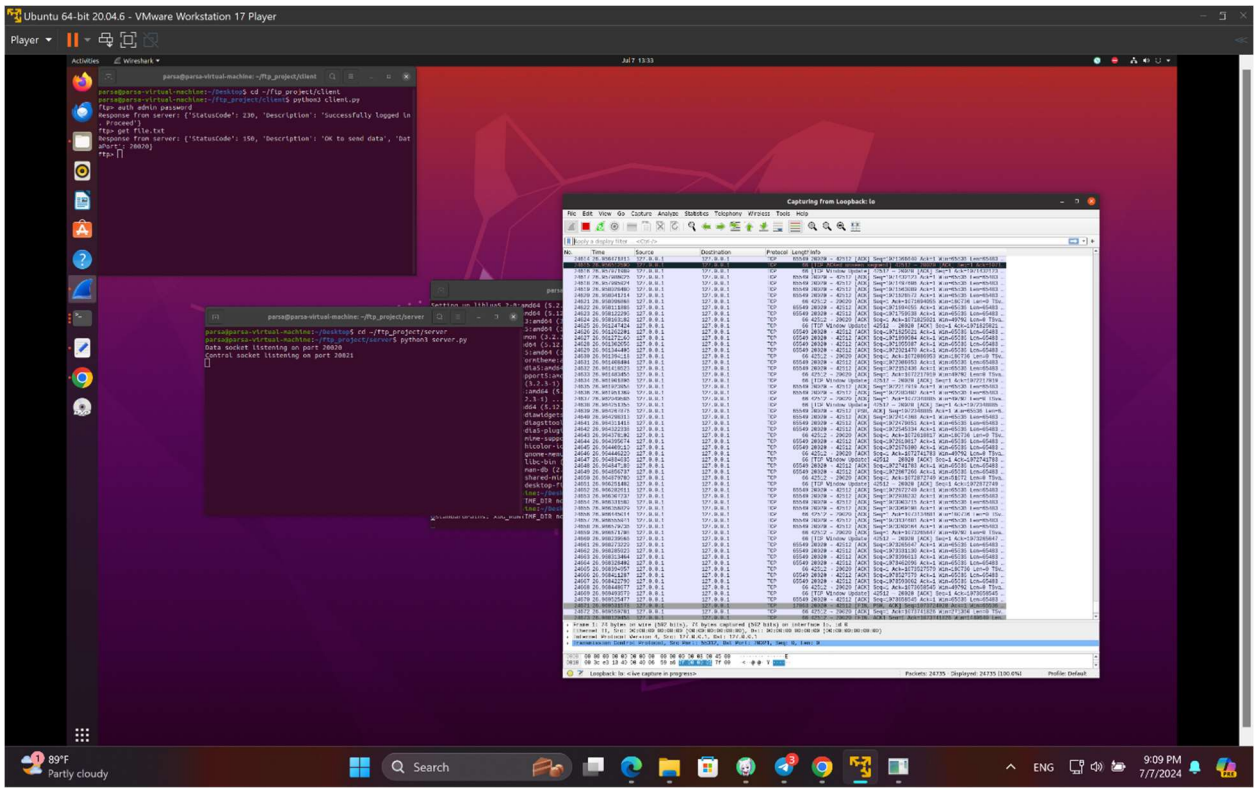
6. Wireshark:

برای انجام این بخش ابتدا برنامه مورد نظر را نصب کردم. سپس با استفاده از ترمینال برنامه را اجرا میکنیم. حال با استفاده از دستور داده شده یک فایل *1GB* تولید کرده. (در بخش کلاینت)

دستور : *fallocate -l 1GB file.txt*

سپس با استفاده از دستور *put* این فایل را در سرور قرار میدهیم. حال میتوان آن را با دستور *get* نیز از سرور دریافت کرد. فرایند دانلود فایل از سرور با استفاده از *wireshark* مشاهده میکنیم. برای اینکار در صفحه اول *loopback interface* را انتخاب میکنیم.

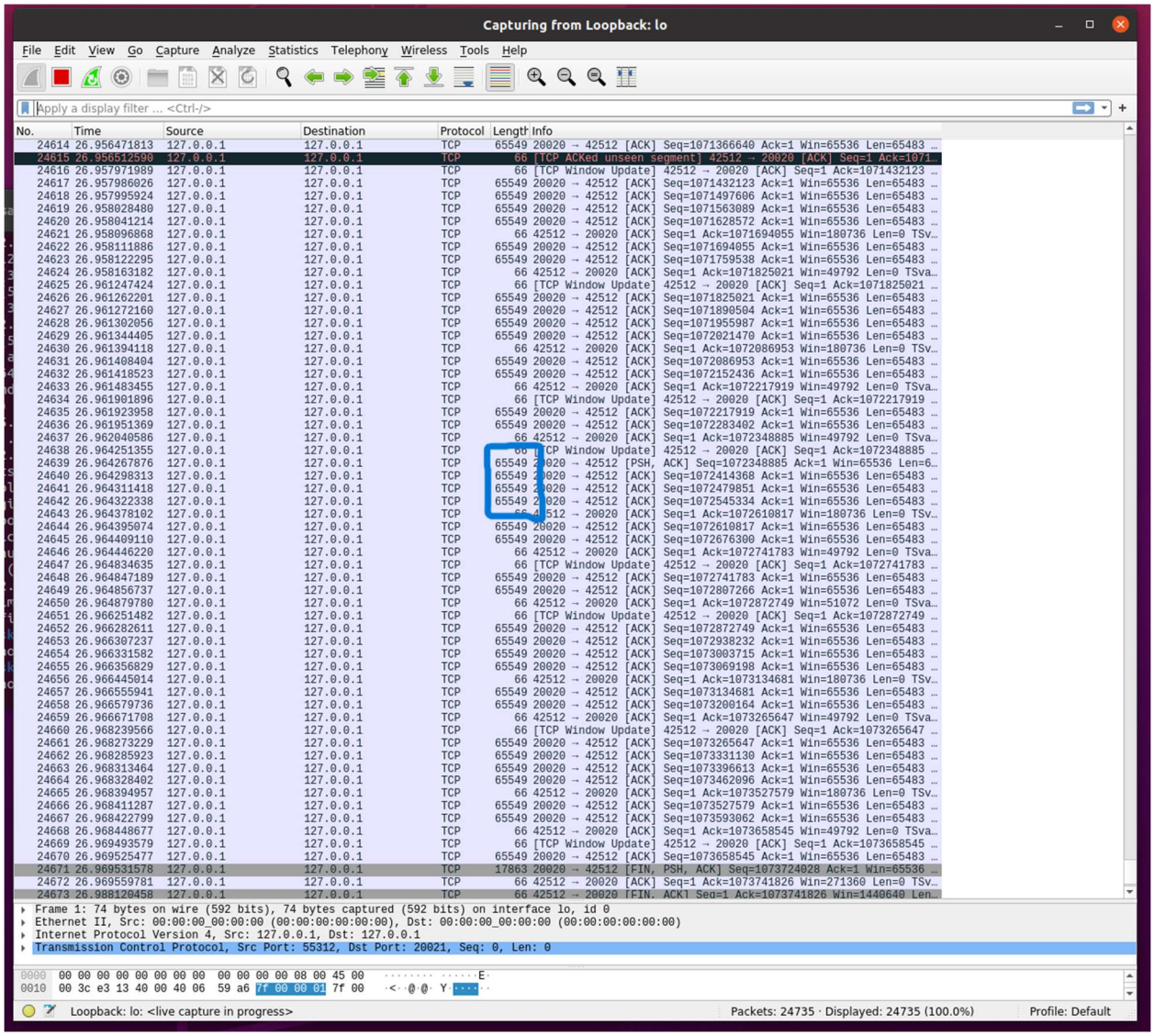
به صورت زیر :



به دلیل بالا بودن حجم فایل لاگ های زیادی در صفحه ثبت شده است و یک صفحه از آن به عنوان نمونه در گزارش آورده شده است تا صحت عمل انجام شده بررسی گردد.

تصویر زوم شده :

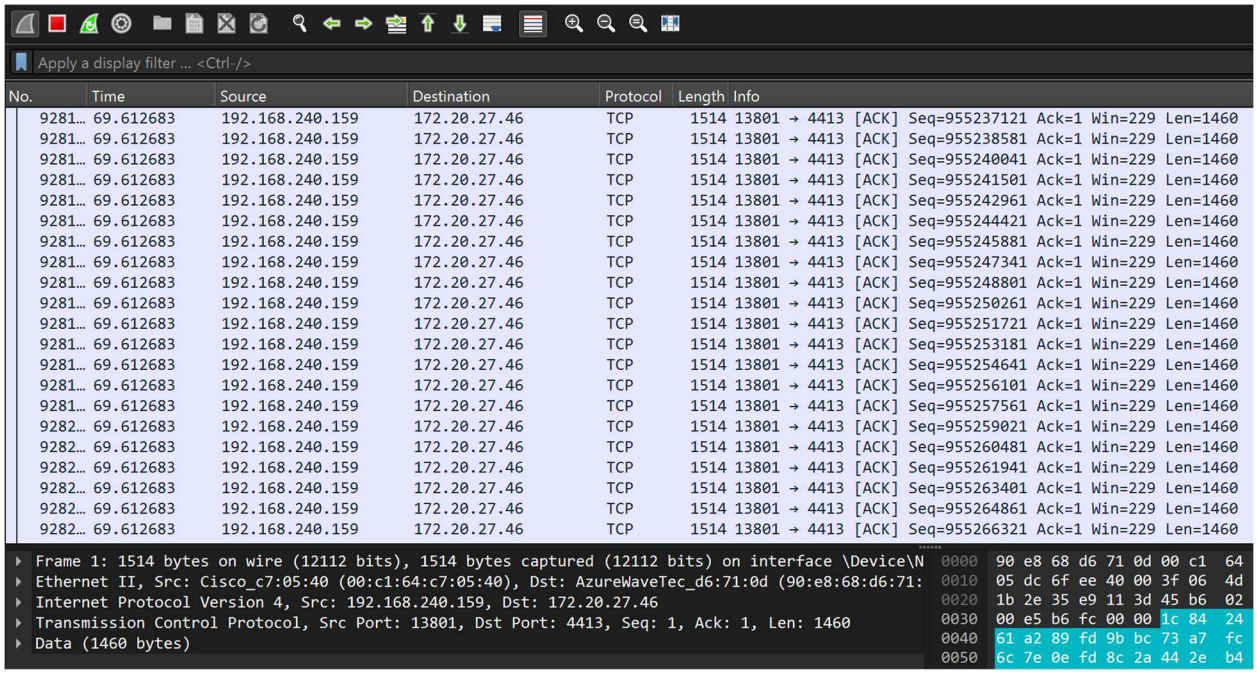
(صفحه بعدی)



همانطور که از عکس بالا مشخص است و با اسکرول کردن هم متوجه میشویم که عددی بزرگتر از این عدد وجود ندارد ، **ماکسیمم سایز پکت های *TCP* برابر با *65549* بایت** است.

7. Sharif FTP:

ابتدا با آدرس */ftp://ftp.sharif.edu* یکی از فایل های موجود را انتخاب میکنیم. سپس فایل را دانلود میکنیم و با استفاده از *wireshark* انتقال پکت ها را مانیتور میکنیم. که به صورت زیر میشود:



همانطور که از تصویر بالا مشخص است. ماکسیمم سایز پکت های *TCP* برابر با *1514* بایت است.

علت تفاوت :

به صورت خلاصه میتوان گفت که اتصال به سرور شریف ماکسیمم سایز پکت ها را محدود میکند ولی در پیاده سازی لوکال ماکسیمم سایز پکت مورد استفاده میگیرد. که عدد 1514 (محدودیت قرار داده شده برای این بخش) مربوط به استاندارد اترنت است: ماکسیمم سایز پکت ها برابر 1500 بایت است به علاوه 14 بایت هدر.

توضیحات بیشتر و جزیی تر:

***Ethernet Standard**: For most Ethernet networks, the standard MTU is 1500 bytes. Adding Ethernet headers and trailers results in a maximum packet size of 1514 bytes. This is commonly seen in typical internet and network traffic, such as our download from Sharif University FTP.*

***Standard TCP/IP**: In standard internet traffic, TCP segments are designed to fit within the MTU of the network path. Hence, we'll typically see packet sizes up to around 1514 bytes.*

***Sharif University FTP**: When downloading files from the FTP server, the packets are likely constrained by the MTU settings of the network path, typically 1514 bytes.*

***Network Layer Constraints**: The network layer (specifically the Data Link layer in the OSI model) limits packet sizes to fit within the MTU of each link in the network path. This ensures efficient routing and prevents fragmentation.*

***Transport Layer Segmentation**: The TCP layer handles segmentation, breaking down larger chunks of data into smaller packets that fit within the MTU constraints. Each segment is then encapsulated in an IP packet for transmission.*

***Standard MTU**: You observe 1514-byte packets due to standard Ethernet MTU settings.*

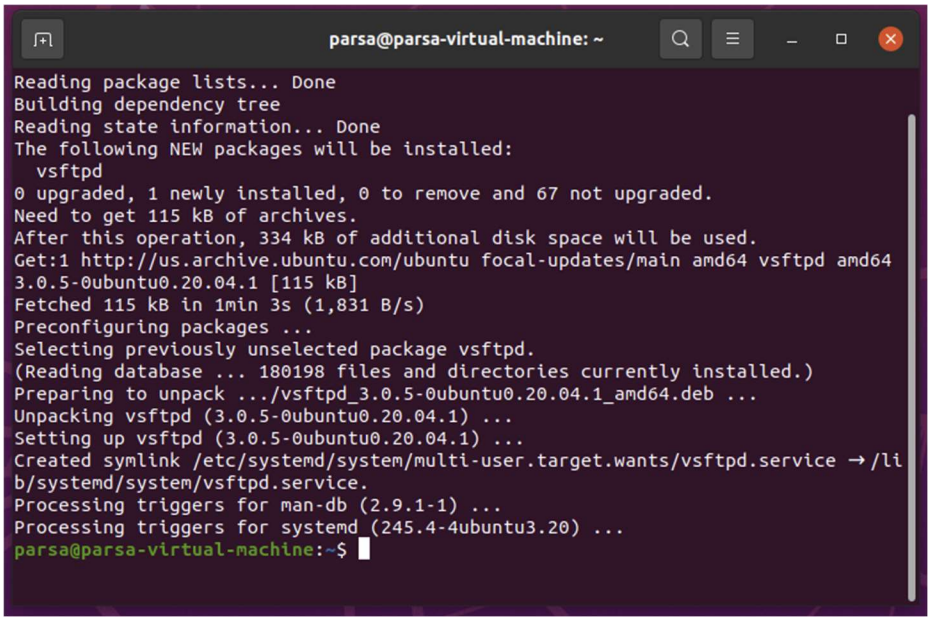
***Larger Chunks**: In a local or controlled environment with different configurations, you might see larger packets if the data is observed before it is segmented or if jumbo frames are in use.*

Part 3: Setting Up a Local FTP Server on Ubuntu

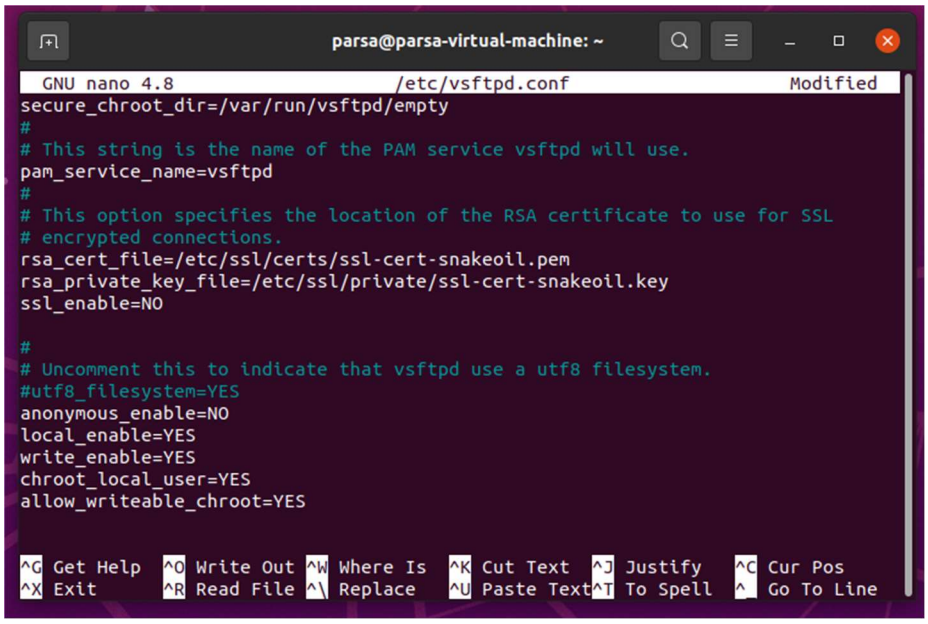
Part A: Install and Configure FTP Server

ابتدا با استفاده از دستور زیر نصب را انجام میدهیم:

sudo apt install vsftpd



نصب انجام شده است. حالا با استفاده از دستور پایین وارد صفحه کانفیگ میشویم تا تغییرات لازم در کانفیگ انجام دهیم.



همانظور که در عکس بالا قابل مشاهده است تغییرات زیر را در انتها اضافه میکنیم و سپس با استفاده از دستور *ctrl+O* و سپس *ctrl+X* تغییرات انجام شده را *write* و سپس خارج میشویم. تغییرات :

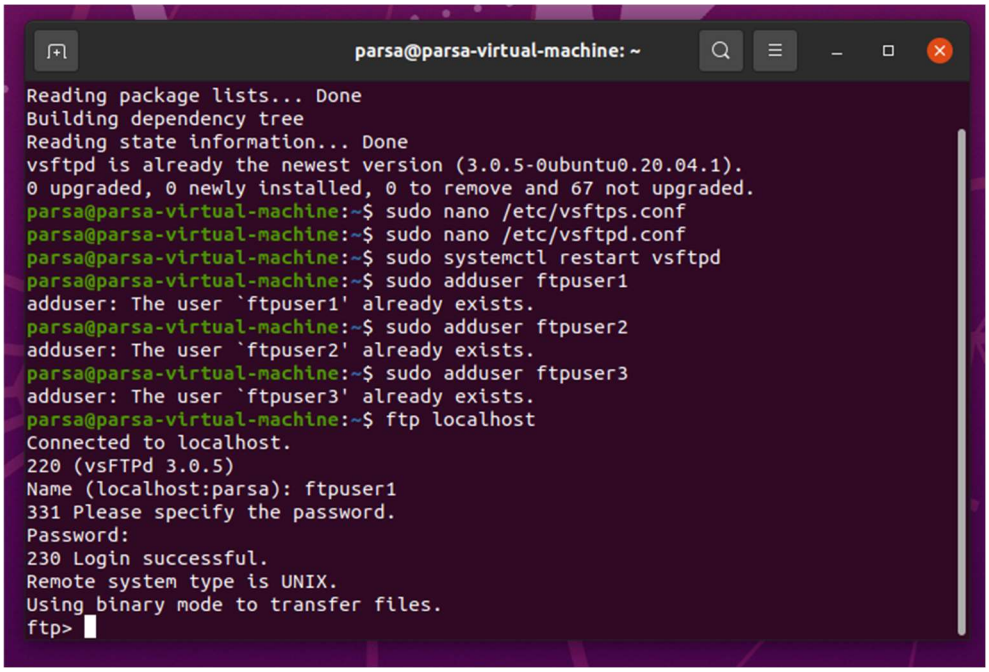
write_enable=YES *local_enable=YES* *anonymous_enable=NO*
allow_writeable_chroot=YES *chroot_local_user=YES*

ابتدا یک بار ری استارت میکنیم و سپس سه یوزر را میسازیم و برای هرکدام پسوورد مشخص میکنیم. به صورت زیر :

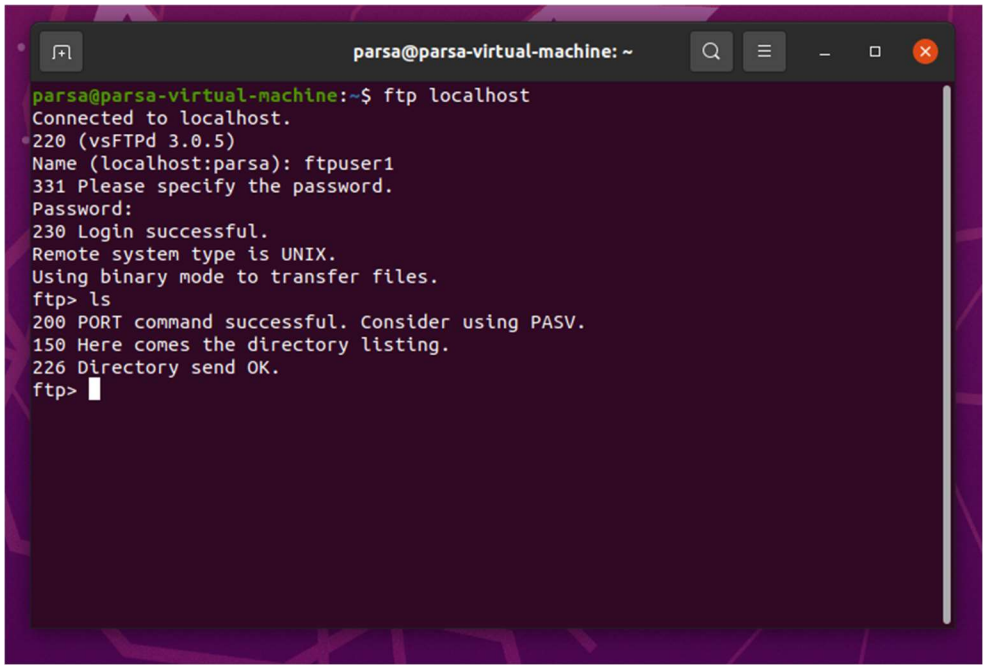
```
parsa@parsa-virtual-machine:~$ sudo systemctl restart vsftpd
parsa@parsa-virtual-machine:~$ sudo adduser ftpuser1
Adding user `ftpuser1' ...
Adding new group `ftpuser1' (1001) ...
Adding new user `ftpuser1' (1001) with group `ftpuser1' ...
Creating home directory `/home/ftpuser1' ...
Copying files from `/etc/skel' ...
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for ftpuser1
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
parsa@parsa-virtual-machine:~$ sudo adduser ftpuser2
Adding user `ftpuser2' ...
Adding new group `ftpuser2' (1002) ...
Adding new user `ftpuser2' (1002) with group `ftpuser2' ...
Creating home directory `/home/ftpuser2' ...
Copying files from `/etc/skel' ...
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for ftpuser2
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
parsa@parsa-virtual-machine:~$ sudo adduser ftpuser3
Adding user `ftpuser3' ...
Adding new group `ftpuser3' (1003) ...
Adding new user `ftpuser3' (1003) with group `ftpuser3' ...
Creating home directory `/home/ftpuser3' ...
Copying files from `/etc/skel' ...
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for ftpuser3
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
parsa@parsa-virtual-machine:~$
```

حال با استفاده از یکی از کلاینت ها (در اینجا من از کلاینت 1 استفاده کردم) به سرور متصل میشویم.

دستور : *ftp localhost*



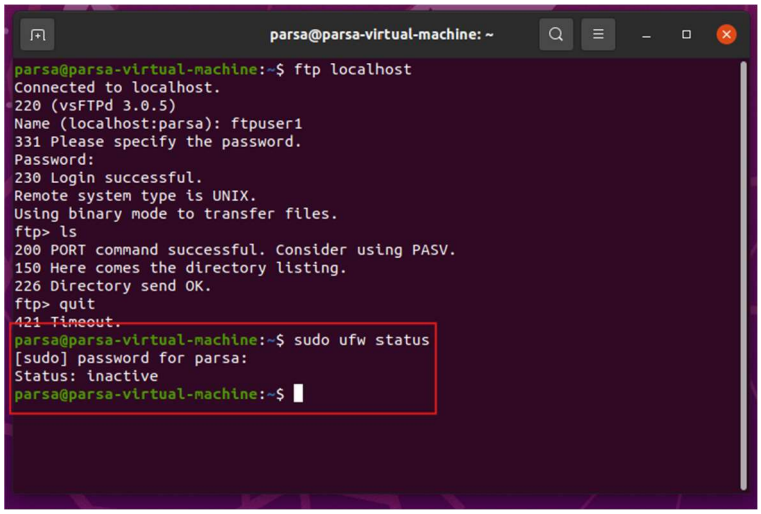
تست دستور *ls* و خروجی صحیح



Part B: Configure Firewall

با استفاده از دستور زیر وضعیت فایروال را بررسی میکنیم:

sudo ufw status

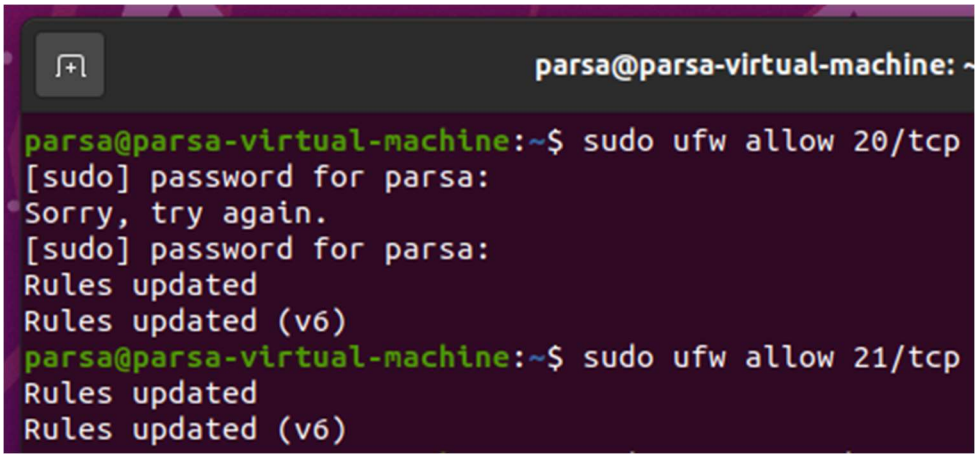


همانطور که در تصویر بالا مشخص است فایروال *inactivate* است.

حالا در گام بعدی فایروال را فعال میکنیم.

```
parsa@parsa-virtual-machine:~$ sudo ufw enable
Firewall is active and enabled on system startup
```

اجازه دادن ترافیک *ftp* بر روی پورت 20 و 21:



چک کردن وضعیت فایروال: (قبل از آن یکبار ریلود انجام میدهیم)

```
parsa@parsa-virtual-machine:~$ sudo ufw enable
Firewall is active and enabled on system startup
parsa@parsa-virtual-machine:~$ sudo ufw allow 20/tcp
Skipping adding existing rule
Skipping adding existing rule (v6)
parsa@parsa-virtual-machine:~$ sudo ufw allow 21/tcp
Skipping adding existing rule
Skipping adding existing rule (v6)
parsa@parsa-virtual-machine:~$ sudo ufw reload
Firewall reloaded
parsa@parsa-virtual-machine:~$ sudo ufw status
Status: active

To                         Action      From
--                         ------      ----
20/tcp                     ALLOW       Anywhere
21/tcp                     ALLOW       Anywhere
20/tcp (v6)                ALLOW       Anywhere (v6)
21/tcp (v6)                ALLOW       Anywhere (v6)

parsa@parsa-virtual-machine:~$
```

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

1. Security:

- ***Protection Against Unauthorized Access:*** *Configuring firewall rules ensures that only legitimate FTP traffic is allowed, preventing unauthorized access to the FTP server. Without proper firewall rules, malicious actors could exploit open ports to gain unauthorized access or launch attacks on the server.*
- ***Mitigation of Attack Vectors:*** *Firewalls help mitigate various network attack vectors, such as brute force attacks and denial-of-service (DoS) attacks, by controlling and monitoring the traffic that reaches the FTP server.*

2. Functionality:

- ***Ensuring Reliable Communication:*** *Proper firewall configuration is essential to ensure that the FTP server can communicate effectively with clients. It allows the necessary control and data connections to be established, enabling smooth and reliable file transfers.*
- ***Avoiding Connection Issues:*** *Incorrect or missing firewall rules can lead to connection issues, such as dropped connections or inability to establish a data connection. Configuring the firewall correctly helps prevent such issues and ensures stable FTP operations.*

3. Network Management:

- ***Traffic Management:*** *Firewalls help manage network traffic by allowing only necessary and legitimate FTP traffic. This control can help prevent network congestion and ensure that bandwidth is used efficiently.*
- ***Resource Allocation:*** *By controlling access to the FTP server, firewalls help in allocating network resources appropriately, ensuring that critical services are prioritized and not overwhelmed by excessive or malicious traffic.*

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

Port 21: *This is the control port used by FTP for sending commands from the client to the server.*

Port 20: *This is the data port used for active mode FTP data transfer.*

Passive Mode Ports: *A range of ports (e.g., 10000-10100) is often used for passive mode FTP data transfer. The exact range can be configured on the FTP server.*

Part C: Change Default Directory

ابتدا از فایل کانفیگ بک آپ میگیریم. با استفاده از دستور پایین:

```
parsa@parsa-virtual-machine:~$ sudo cp /etc/vsftpd.conf /etc/vsftpd.conf.bak
parsa@parsa-virtual-machine:~$
```

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

Reversibility: *If something goes wrong during the editing process or if the new configuration causes issues, having a backup allows you to quickly revert to the original state. This ensures that you can restore the system to its previous working condition without losing any critical settings.*

Stability: *Backing up configuration files ensures that you have a known good state to revert to, minimizing downtime and disruptions. This is especially important for systems that need to remain operational without extended periods of troubleshooting.*

Safety: *Editing configuration files can sometimes lead to accidental misconfigurations or data loss. A backup provides a safety net, protecting against accidental errors and ensuring that you can recover from mistakes without significant impact on the system.*

سپس دو خط انتهایی را در فایل کانفیگ اضافه میکنیم:

```
anonymous_enable=NO
local_enable=YES
write_enable=YES
chroot_local_user=YES
allow_writeable_chroot=YES
user_sub_token=$USER
local_root=/home/$USER/ftp
```

با استفاده از دستورات قابل مشاهده در پایین ، ابتدا دایرکتوری های لازم را تشکیل میدهیم.
سپس مالکیت دایرکتوری ها را اختصاص میدهیم. سرویس را ری استارت میکنیم و فایل های تست در هر دایرکتوری ایجاد میکنیم. سپس کانکشن را با سرور برقرار میکنیم و با یوزر پس کاربر اول وارد میشویم.

```
parsa@parsa-virtual-machine:~$ sudo mkdir -p /home/ftpuser1/ftp
parsa@parsa-virtual-machine:~$ sudo mkdir -p /home/ftpuser2/ftp
parsa@parsa-virtual-machine:~$ sudo mkdir -p /home/ftpuser3/ftp
parsa@parsa-virtual-machine:~$ sudo chown -R ftpuser1:ftpuser1 /home/ftpuser1/ftp
parsa@parsa-virtual-machine:~$ sudo chown -R ftpuser2:ftpuser2 /home/ftpuser2/ftp
parsa@parsa-virtual-machine:~$ sudo chown -R ftpuser3:ftpuser3 /home/ftpuser3/ftp
parsa@parsa-virtual-machine:~$ sudo systemctl restart vsftpd
parsa@parsa-virtual-machine:~$ sudo touch /home/ftpuser1/ftp/textfile.txt
parsa@parsa-virtual-machine:~$ sudo touch /home/ftpuser2/ftp/textfile.txt
parsa@parsa-virtual-machine:~$ sudo touch /home/ftpuser3/ftp/textfile.txt
parsa@parsa-virtual-machine:~$ ftp localhost
Connected to localhost.
220 (vsFTPd 3.0.5)
Name (localhost:parsa): ftpuser1
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls
```

سپس دستور ls را تست میکنیم:

```
parsa@parsa-virtual-machine:~$ ftp localhost
Connected to localhost.
220 (vsFTPd 3.0.5)
Name (localhost:parsa): ftpuser1
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls
200 PORT command successful. Consider using PASV.
150 Here comes the directory listing.
drwxr-xr-x    2 1001     1001         4096 Jul 07 17:48 ftp
-rw-r--r--    1 0        0               0 Jul 07 18:11 textfile.txt
226 Directory send OK.
```

همانطور که مشخص است فایل ذکر شده در دایرکتوری مد نظر ساخته شده است.

یک فایل دیگر در کنار آن با نام *testfile.txt* ایجاد میکنیم و سپس تلاش میکنیم این فایل را دیلیت کنیم. همچنین فایل *textfile.txt* را با استفاده از دستور *get* دریافت میکنیم.

```
parsa@parsa-virtual-machine:~$ ftp localhost
Connected to localhost.
220 (vsFTPd 3.0.5)
Name (localhost:parsa): ftpuser1
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls
200 PORT command successful. Consider using PASV.
150 Here comes the directory listing.
drwxr-xr-x    2 1001     1001         4096 Jul 07 17:48 ftp
-rw-r--r--    1 0        0              33 Jul 07 18:35 testfile.txt
-rw-r--r--    1 0        0              33 Jul 07 18:32 textfile.txt
226 Directory send OK.
ftp> dele testfile.txt
250 Delete operation successful.
ftp> ls
200 PORT command successful. Consider using PASV.
150 Here comes the directory listing.
drwxr-xr-x    2 1001     1001         4096 Jul 07 17:48 ftp
-rw-r--r--    1 0        0              33 Jul 07 18:32 textfile.txt
226 Directory send OK.
ftp> get textfile.txt
local: textfile.txt remote: textfile.txt
200 PORT command successful. Consider using PASV.
150 Opening BINARY mode data connection for textfile.txt (33 bytes).
226 Transfer complete.
33 bytes received in 0.00 secs (111.5106 kB/s)
ftp>
```

همانطور که از بالا قابل مشاهده است ، فایل ذکر شده به درستی حذف شده و سپس فایل دیگر نیز به درستی انتقال داده شده است. قابلیت دسترسی توسط کلاینت به شکل بالا بررسی شد.

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

Security: *By setting specific directories for FTP users, you can limit their access to only the areas of the filesystem that they need. This reduces the risk of unauthorized access to sensitive files and directories outside the designated FTP area.*

Organization: *Assigning a dedicated directory for FTP users helps in organizing files and managing the FTP server more efficiently. It keeps the FTP environment tidy and makes it easier to manage user access and file permissions.*

Resource Management: *Customizing the FTP home directories allows you to allocate resources more effectively. You can tailor the FTP environment to meet the specific needs of different users or groups, optimizing storage and access based on usage patterns and requirements.*

Part D: Securing FTP

در ابتدای این بخش با استفاده از دستور زیر *openssl* را نصب میکنیم.

```
parsa@parsa-virtual-machine:~$ sudo apt install openssl
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 67 not upgraded.
parsa@parsa-virtual-machine:~$
```

همانطور که مشاهده میشود. نصب شده است و جدیدترین ورژن است.

Generate SSL/TLS Certificates:

```
parsa@parsa-virtual-machine:~$ sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /etc/ssl/private/vsftpd.key -out /etc/ssl/cer
ts/vsftpd.crt
Generating a RSA private key
.....................................................+++++
.............+++++
writing new private key to '/etc/ssl/private/vsftpd.key'
-----
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 []:
```

حال تغییرات لازم در کانفیگ را به صورت زیر انجام میدهیم: (*Configure vsftpd to use SSL/TLS:*)

```
secure_chroot_dir=/var/run/vsftpd/empty
#
# 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
anonymous_enable=NO
local_enable=YES
write_enable=YES
chroot_local_user=YES
allow_writeable_chroot=YES
user_sub_token=$USER
local_root=/home/$USER/ftp
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/certs/vsftpd.crt
rsa_private_key_file=/etc/ssl/private/vsftpd.key
```

Restart the vsftpd service, Set user permissions to restrict access:

```
parsa@parsa-virtual-machine:~$ sudo nano /etc/vsftpd.conf
parsa@parsa-virtual-machine:~$ sudo systemctl restart vsftpd
parsa@parsa-virtual-machine:~$ sudo nano /etc/vsftpd.conf
parsa@parsa-virtual-machine:~$ sudo chown -R ftpuser1:ftpuser1 /home/ftpuser1/ftp
parsa@parsa-virtual-machine:~$ sudo chmod 755 /home/ftpuser1/ftp
parsa@parsa-virtual-machine:~$ sudo chown -R ftpuser2:ftpuser2 /home/ftpuser2/ftp
parsa@parsa-virtual-machine:~$ sudo chmod 755 /home/ftpuser2/ftp
parsa@parsa-virtual-machine:~$ sudo chown -R ftpuser3:ftpuser3 /home/ftpuser3/ftp
parsa@parsa-virtual-machine:~$ sudo chmod 755 /home/ftpuser3/ftp
parsa@parsa-virtual-machine:~$ lftp -u ftpuser1,ftpuser1 localhost
```

با استفاده از دستور زیر میتوان تست انجام داد. (در گروه گفته شد این بخش لازم نیست و صرفا قرار دادن کد ها کافیست)

```
parsa@parsa-virtual-machine:~$ lftp -u ftpuser1,ftpuser1 localhost

Command 'lftp' not found, but can be installed with:

sudo apt install lftp

parsa@parsa-virtual-machine:~$ sudo apt install lftp
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
  lftp
0 upgraded, 1 newly installed, 0 to remove and 67 not upgraded.
Need to get 563 kB of archives.
After this operation, 1,700 kB of additional disk space will be used.
Get:1 http://us.archive.ubuntu.com/ubuntu focal/main amd64 lftp amd64 4.8.4-2build3 [563 kB]
Fetched 563 kB in 3s (205 kB/s)
Selecting previously unselected package lftp.
(Reading database ... 180255 files and directories currently installed.)
Preparing to unpack .../lftp_4.8.4-2build3_amd64.deb ...
Unpacking lftp (4.8.4-2build3) ...
Setting up lftp (4.8.4-2build3) ...
Processing triggers for desktop-file-utils (0.24-1ubuntu3) ...
Processing triggers for mime-support (3.64ubuntu1) ...
Processing triggers for hicolor-icon-theme (0.17-2) ...
Processing triggers for gnome-menus (3.36.0-1ubuntu1) ...
Processing triggers for man-db (2.9.1-1) ...
parsa@parsa-virtual-machine:~$ lftp -u ftpuser1,ftpuser1 localhost
lftp ftpuser1@localhost:~> ls
```

مثال استفاده:

```
lftp ftpuser1@localhost:~> put localfile.txt
put: /home/parsa/localfile.txt: No such file or directory
```

Why is it important to encrypt FTP traffic?

Data Protection: Encryption ensures that the data being transferred between the client and the server is protected from eavesdropping. Without encryption, sensitive information such as usernames, passwords, and file contents can be intercepted and read by malicious actors.

Data Integrity: Encryption helps maintain the integrity of the data being transferred. It prevents unauthorized modifications during transit, ensuring that the data received is exactly what was sent.

Authentication: Encryption protocols like SSL/TLS provide mechanisms for authenticating the identities of the client and the server. This

verification process helps prevent man-in-the-middle attacks where an attacker might impersonate the server or the client.

Compliance: Many industries and regulatory bodies require data to be encrypted during transmission to protect sensitive information. Encrypting FTP traffic helps organizations comply with these regulatory requirements and avoid potential legal and financial penalties.

What are the benefits of using SSL/TLS encryption for FTP?

Enhanced Security: *SSL/TLS encryption provides a secure channel for transferring data, ensuring that the data cannot be easily intercepted or tampered with during transmission. This is especially important for protecting sensitive information such as login credentials and confidential files.*

Data Confidentiality: *With SSL/TLS, the data transferred between the client and the server is encrypted, making it unreadable to anyone who might intercept the traffic. This ensures that confidential information remains private.*

Data Integrity: *SSL/TLS includes mechanisms to detect any alterations to the data during transit. This means that if any data is modified in transit, the recipient can detect the tampering and take appropriate action.*

Authentication: *SSL/TLS supports the use of digital certificates for authenticating the server and, optionally, the client. This helps ensure that clients are connecting to a legitimate server and not an impostor, thereby preventing man-in-the-middle attacks.*

Regulatory Compliance: *Many regulatory standards and frameworks require encryption for data in transit. Implementing SSL/TLS for FTP helps organizations meet these compliance requirements and avoid potential legal and financial repercussions.*

Why do we need to secure FTP servers?

Prevent Unauthorized Access*: FTP servers often store sensitive and confidential data. Without proper security measures, unauthorized users can gain access to these files, leading to potential data breaches and information theft.*

Protect Data Integrity*: Securing the FTP server ensures that the data being transferred and stored is not altered or corrupted by unauthorized users. This is crucial for maintaining the accuracy and reliability of the information.*

Comply with Regulations*: Many industries are subject to regulatory requirements that mandate secure data transmission and storage. Securing FTP servers helps organizations comply with these regulations, avoiding legal penalties and maintaining their reputation.*

Mitigate Attacks*: FTP servers can be targets for various cyber attacks, including brute force attacks, malware, and denial-of-service (DoS) attacks. Implementing security measures helps protect the server from these threats, ensuring its availability and performance.*

Ensure Confidentiality*: By securing FTP servers, organizations can ensure that the data transferred between clients and the server remains confidential and is not intercepted by malicious actors. This is especially important for protecting sensitive information such as login credentials and personal data.*

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

Chroot Jail: *This method restricts users to a specific directory, preventing them from navigating to other parts of the filesystem. It effectively isolates users, limiting their access to only their designated directories.*

User Permissions: *Setting strict file and directory permissions ensures that users can only read, write, or execute files that they are authorized to access. This includes configuring ownership and permission settings appropriately for each user or group.*

Restrict FTP Commands: *Limiting the FTP commands available to users can enhance security. For example, disabling certain commands like DELETE or RMD (Remove Directory) for regular users can prevent accidental or malicious deletion of files and directories.*

Account Isolation: *Creating separate accounts for different users or groups with specific access rights ensures that each user can only access their own files and directories. This isolation helps prevent unauthorized access to other users' data.*

Use of Secure Protocols: *Implementing secure versions of FTP, such as FTPS or SFTP, can include additional access control features. For example, using SSH keys with SFTP can provide strong authentication, ensuring that only authorized users can connect to the server.*

Access Control Lists (ACLs): *ACLs provide fine-grained control over who can access specific files and directories. They allow administrators to define detailed access permissions for individual users or groups, beyond the standard UNIX permission model.*

Part 4: Bandwidth and Transfer Rate Control

Why is bandwidth control important in network applications?

1. Network Stability and Performance

- ***Preventing Congestion:*** *By controlling the bandwidth, network administrators can prevent any single application or user from consuming excessive network resources, which can lead to congestion. This ensures that all users experience consistent and reliable network performance.*
- ***Optimizing Traffic Flow:*** *Bandwidth control helps in managing the flow of data across the network, optimizing the use of available resources and preventing bottlenecks that could slow down critical applications.*

2. Quality of Service (QoS)

- ***Prioritizing Critical Applications:*** *Bandwidth control allows administrators to prioritize network traffic, ensuring that critical applications such as VoIP, video conferencing, and online transactions receive the necessary bandwidth to function effectively.*
- ***Guaranteeing Service Levels:*** *By allocating specific amounts of bandwidth to different types of traffic, network administrators can guarantee certain levels of service for various applications, meeting the performance expectations of users and business requirements.*

3. Security and Fair Usage

- ***Preventing Network Abuse:*** *Without bandwidth control, a few users or applications can monopolize the network, leading to unfair usage and potential network abuse. Implementing*

bandwidth limits ensures that all users have fair access to network resources.

- ***Protecting Against DoS Attacks:*** *Bandwidth control can help mitigate the impact of Denial of Service (DoS) attacks by limiting the amount of bandwidth any single connection can consume, thus protecting the network from being overwhelmed by malicious traffic.*

4. Improved User Experience

- ***Reducing Latency and Jitter:*** *By managing bandwidth effectively, network administrators can reduce latency and jitter, which are critical for real-time applications such as online gaming, video streaming, and VoIP. This leads to a smoother and more responsive user experience.*
- ***Ensuring Consistent Performance:*** *Bandwidth control helps maintain consistent network performance, ensuring that users can access applications and services without experiencing significant slowdowns or interruptions.*

5. Cost Management

- ***Efficient Use of Resources:*** *By optimizing the use of available bandwidth, organizations can avoid the need for expensive network upgrades or additional resources. This efficient use of network resources can lead to significant cost savings.*
- ***Budgeting for Bandwidth:*** *Bandwidth control allows organizations to plan and budget for their network usage more effectively, ensuring that they are not paying for more bandwidth than they actually need.*

برای انجام دادن این پارت کافی است مقداری تغییرات در کدی که برای بخش اول سوال پیاده کردیم انجام بدهیم. به صورت زیر :

تغییرات بخش سرور :

```python
def send_file(filename, data_socket, max_bandwidth_kbps=100):
    max_bandwidth_bps = max_bandwidth_kbps * 1024  # Convert KB/s to Bytes/s
    with open(filename, 'rb') as file:
        total_sent = 0
        start_time = time.time()
        while True:
            data = file.read(1024)
            if not data:
                break
            data_socket.sendall(data)
            total_sent += len(data)

            elapsed_time = time.time() - start_time
            if elapsed_time > 0:
                transfer_rate = total_sent / elapsed_time
                if transfer_rate > max_bandwidth_bps:
                    sleep_time = (total_sent / max_bandwidth_bps) - elapsed_time
                    time.sleep(sleep_time)

            file_size = os.path.getsize(filename)
            progress = (total_sent / file_size) * 100
            print(f"Progress: {progress:.2f}% - Transfer Rate: {transfer_rate / 1024:.2f} KB/s")
```

توضیحات :

این تابع برای ارسال فایل از سرور به کلاینت استفاده می‌شود و شامل کنترل پهنای باند و چاپ پیشرفت انتقال فایل است.

تبدیل پهنای باند به بایت بر ثانیه در ابتدای کد. در این خط، پهنای باند که به کیلوبایت بر ثانیه داده شده است، به بایت بر ثانیه تبدیل می‌شود. داده‌ها از فایل به اندازه ۱۰۲۴ بایت خوانده می‌شوند و سپس به کلاینت ارسال می‌شوند. همچنین تعداد بایت‌های ارسال شده به‌روز می‌شود. نرخ انتقال محاسبه می‌شود و اگر از حداکثر پهنای باند مشخص شده بیشتر باشد، تاخیر اعمال می‌شود تا سرعت انتقال کاهش یابد. میزان پیشرفت انتقال فایل و نرخ انتقال چاپ می‌شود.

تغییرات بخش کلاینت :

```python
def receive_data(filename, port):
    client = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    client.connect(('localhost', port))
    with open(filename, 'wb') as file:
        total_received = 0
        start_time = time.time()
        while True:
            data = client.recv(1024)
            if not data:
                break
            file.write(data)
            total_received += len(data)

            elapsed_time = time.time() - start_time
            if elapsed_time > 0:
                transfer_rate = total_received / elapsed_time
                file_size = os.path.getsize(filename)
                progress = (total_received / file_size) * 100
                print(f"Progress: {progress:.2f}% - Transfer Rate: {transfer_rate / 1024:.2f} KB/s")
    client.close()
```

توضیحات :

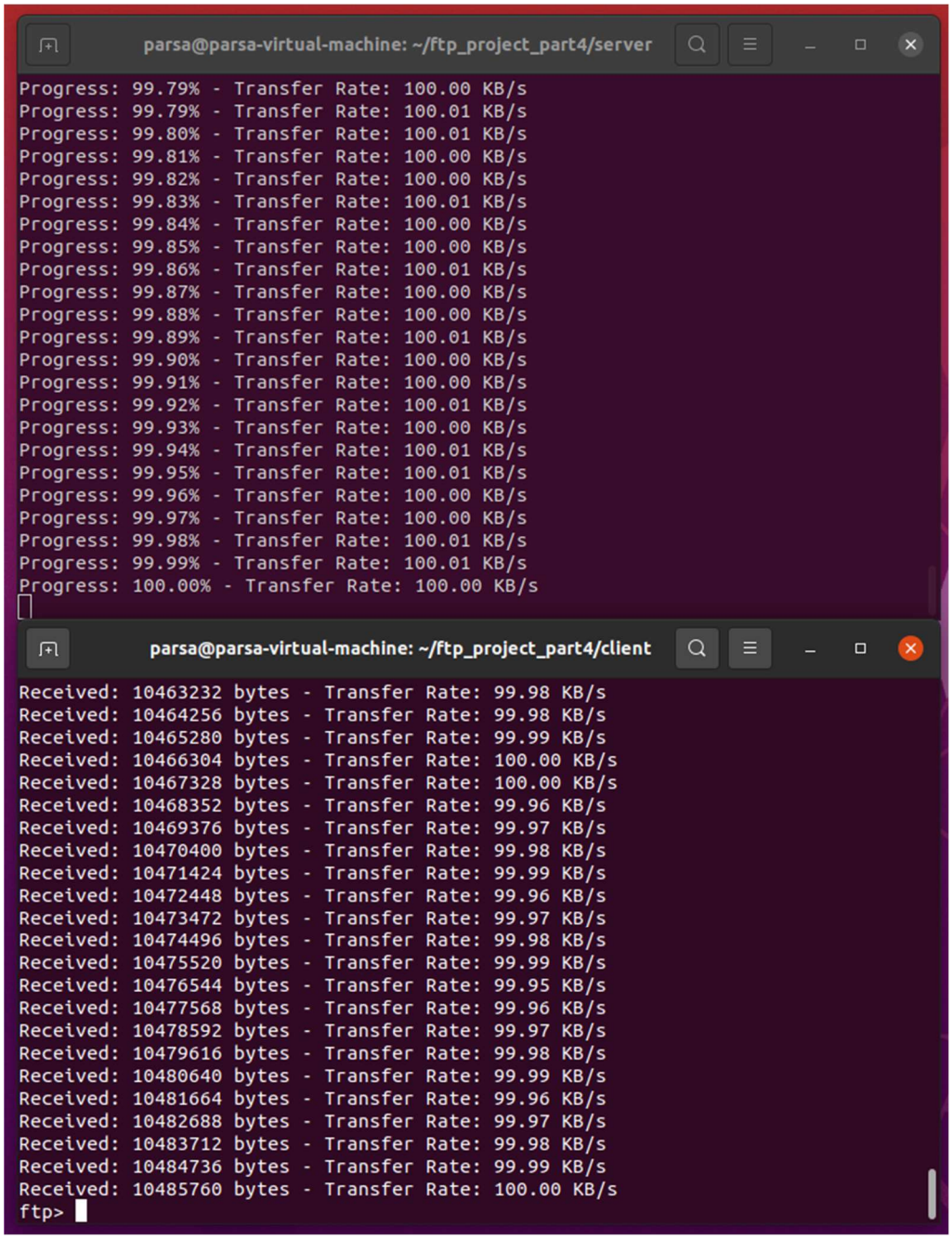
این تابع برای دریافت داده‌ها از سرور استفاده می‌شود و شامل چاپ پیشرفت و نرخ انتقال است.

داده‌ها از سرور به اندازه ۱۰۲۴ بایت دریافت و سپس در فایل ذخیره می‌شوند. همچنین تعداد بایت‌های دریافت شده به‌روز می‌شود. نرخ انتقال محاسبه و میزان پیشرفت انتقال فایل و نرخ انتقال چاپ می‌شود.

حال با استفاده از دستور زیر یک فایل *10M* ایجاد میکنیم :

fallocate -l 10M testfile.txt

سپس این فایل را توسط دستور *get testfile.txt* دریافت میکنیم. خروجی به صورت زیر است:



همانطور که مشخص است در هر دو سمت سرور و کلاینت مراحل انتقال دیتا ، تعداد بایت های انتقال داده شده تا کنون ، درصد پیشرفت انتقال و همچنین دیتا ریت مورد استفاده برای انتقال نمایش داده شده است. همانطور که مشخص است دیتا ریت مورد استفاده به $100KB/s$ محدود شده است. با توجه به این که حجم فایل $10M$ است، این انتقال حدود 100 ثانیه طول میکشید.

$$(10M)/(100KB/s) = 100\ s$$