**Developing a Secure File Transfer System Using SFTP on a Linux Server**

Project Title: Developing a Secure File Transfer System Using SFTP on a Linux Server

Prepared by: Sagar Chauhan

Date: 26-04-2025

**TABLE OF CONTENTS**

1. Project Title . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 5
2. Objective . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 6
3. Tools & Technologies Used. . . . . . . . . . . . . . . . . . . . . . . . . . . . 7
4. System Requirements. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 8
5. Steps for Configuration. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .

* 5.1 Install OpenSSH Server. . . . . . . . . . . . . . . . . . . . . . . . . . . . 9
* 5.2 Start & Check SSH Service. . . . . . . . . . . . . . . . . . . . . . . 9
* 5.3 Create a Restricted SFTP User. . . . . . . . . . . . . . . . . . . . . 10
* 5.4 Set Directory Structure & Permissions. . . . . . . . . . . . . . . 10
* 5.5 Configure SSH for SFTP (Chroot Jail) . . . . . . . . . . . . . . . . . 10
* 5.6 File Transfer using SFTP. . . . . . . . . . . . . . . . . . . . . . . . . . . . 11

1. Security Measures Implemented. . . . . . . . . . . . . . . . . . . . . . . . . . 13
2. Testing the System

* 7.1 IP Configuration. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 15
* 7.2 Kali → Ubuntu Connection Test. . . . . . . . . . . . . . . . . . . . . 15
* 7.3 Upload & Download File using SFTP. . . . . . . . . . . . 16

1. Conclusion. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 17

**Abstract**

In today's digital age, secure file transfer between systems is essential to protect sensitive data from unauthorized access and cyber threats. This project focuses on implementing a secure file transfer system using the **SFTP (Secure File Transfer Protocol)** on a Linux-based server environment. The configuration is done using **Ubuntu Server** as the host and **Kali Linux** as the client machine, both running within a virtualized environment (VirtualBox).

The goal of this project is to demonstrate the setup of a secure and restricted user environment for SFTP, ensuring that file access is limited and controlled through proper user permissions and directory structure. Key configurations include setting up the OpenSSH server, creating a chroot jail environment for the SFTP user, and testing the system through upload and download operations.

This project not only showcases technical skills in Linux administration and network configuration, but also emphasizes the importance of implementing secure protocols in real-world IT systems.

**1. Project Title**

Developing a Secure File Transfer System Using SFTP on a Linux Server

**2. Objective**

The objective of this project is to configure and implement a secure file transfer system using SFTP (Secure File Transfer Protocol) on a Linux server. The aim is to allow secure upload and download of files between systems while ensuring that user access is restricted and data remains protected from unauthorized access.

This includes:

* Setting up an OpenSSH server with SFTP support.
* Creating a restricted user environment (chroot jail).
* Testing the system using a client machine to verify secure file transfer operations.

**3. Tools & Technologies Used**

1. **Operating Systems:**
   * Ubuntu Server (for hosting the file server)
   * Kali Linux (as the client machine for accessing the server)
2. **Virtualization:**
   * Oracle VM VirtualBox (used to create virtual machines and simulate network configuration)
3. **Networking:**
   * Bridged Adapter Mode (network configuration to ensure both VMs can communicate over the same network)
4. **Protocols:**
   * SSH (Secure Shell) (used for encrypted communication between the client and server)
   * SFTP (Secure File Transfer Protocol) (for secure file transfer with authentication and encryption)
5. **Software:**
   * OpenSSH Server (installed on Ubuntu to enable SSH/SFTP services)
6. **Security:**
   * Chroot Jail (used to restrict user access to specific directories)
7. **Firewall:**
   * UFW (Uncomplicated Firewall) (configured to allow only necessary ports for secure communication)

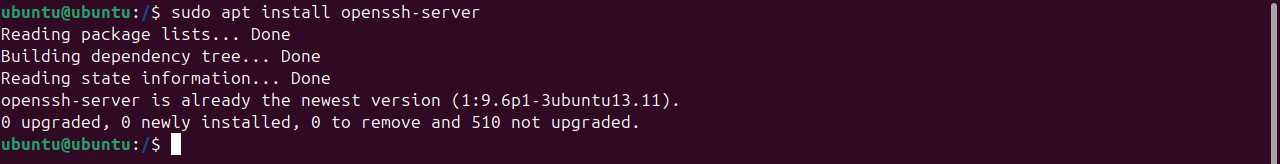
**4. System Requirements**

1. **Hardware Requirements:**
   * Processor: 2.0 GHz or higher (Intel/AMD)
   * RAM: Minimum 4 GB (8 GB or more recommended for smooth virtualization)
   * Hard Disk: 50 GB free space or more
   * Network: Local Network (wired or wireless) for virtual machines to communicate
2. **Software Requirements:**
   * Operating System:
     + Host OS: Any modern version of Linux, Windows, or macOS
     + Guest OS:
       - Ubuntu Server 20.04 (for the file server)
       - Kali Linux 2020/2021 (as the client for testing)
   * Virtualization Software:
     + Oracle VM VirtualBox (version 6.1 or higher)
   * SSH & SFTP Software:
     + OpenSSH Server (on Ubuntu for enabling SSH/SFTP)
   * Networking Software:
     + Bridged Adapter Mode (to ensure proper communication between VMs)
3. **Network Configuration:**
   * Virtual machines should be configured in Bridged Adapter mode to ensure they are on the same local network.
   * Both VMs should be able to ping each other using their assigned IP addresses.

**5. Step-by-Step Configuration**

**5.1 Install OpenSSH Server**

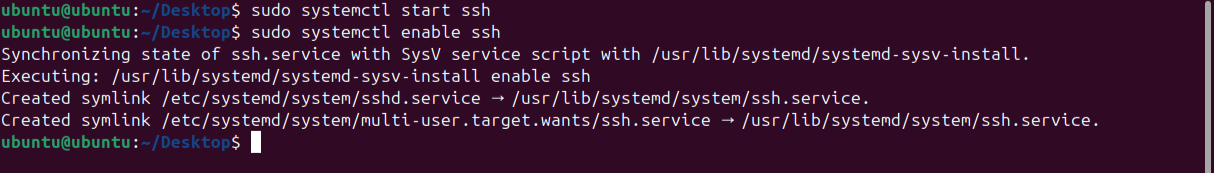
On the Ubuntu Server (within VirtualBox), install the OpenSSH server to enable SSH and SFTP functionalities:

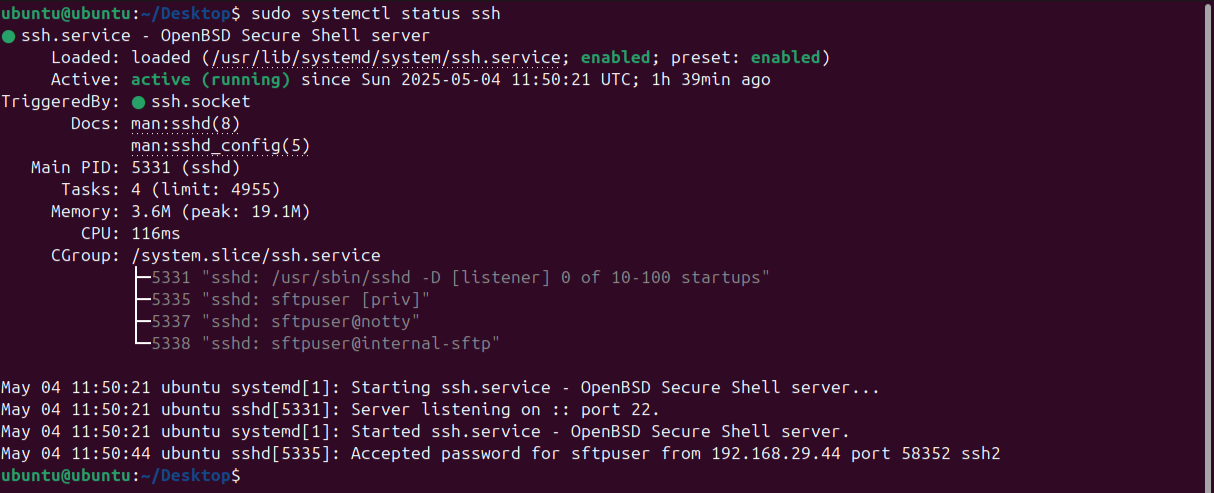


**Command:** sudo apt install openssh-server Installs the OpenSSH server to enable SSH and SFTP services.

**5.2 Start & Check SSH Service**

Ensure that the SSH service is running properly:





**Command:**

sudo systemctl start ssh: Starts the SSH service to allow remote connections.

sudo systemctl enable ssh: Ensures SSH starts automatically on boot.

sudo systemctl status ssh: Checks the status of the SSH service.

**5.3 Create a Restricted SFTP User**

Create a dedicated user for SFTP access:

**Command:**

sudo adduser sftpuser: Creates a new user sftpuser for restricted SFTP access.

**5.4 Set Directory Structure & Permissions**

Create a secure folder for the user and restrict access:

**Command:**

sudo mkdir -p /home/sftpuser/uploads: Creates the uploads directory for file storage.

sudo chown root:root /home/sftpuser: Sets the ownership of the home directory to root.

sudo chmod 755 /home/sftpuser: Provides the correct permissions to the home directory.

sudo chown sftpuser:sftpuser /home/sftpuser/uploads: Changes ownership of the uploads directory to the sftpuser.

**5.5 Configure SSH for SFTP (Chroot Jail)**

Edit the SSH configuration file:

**Command:**

sudo vim /etc/ssh/sshd\_config: Edits the SSH configuration file to set up the chroot jail for the user.



**Add the following lines at the end:**

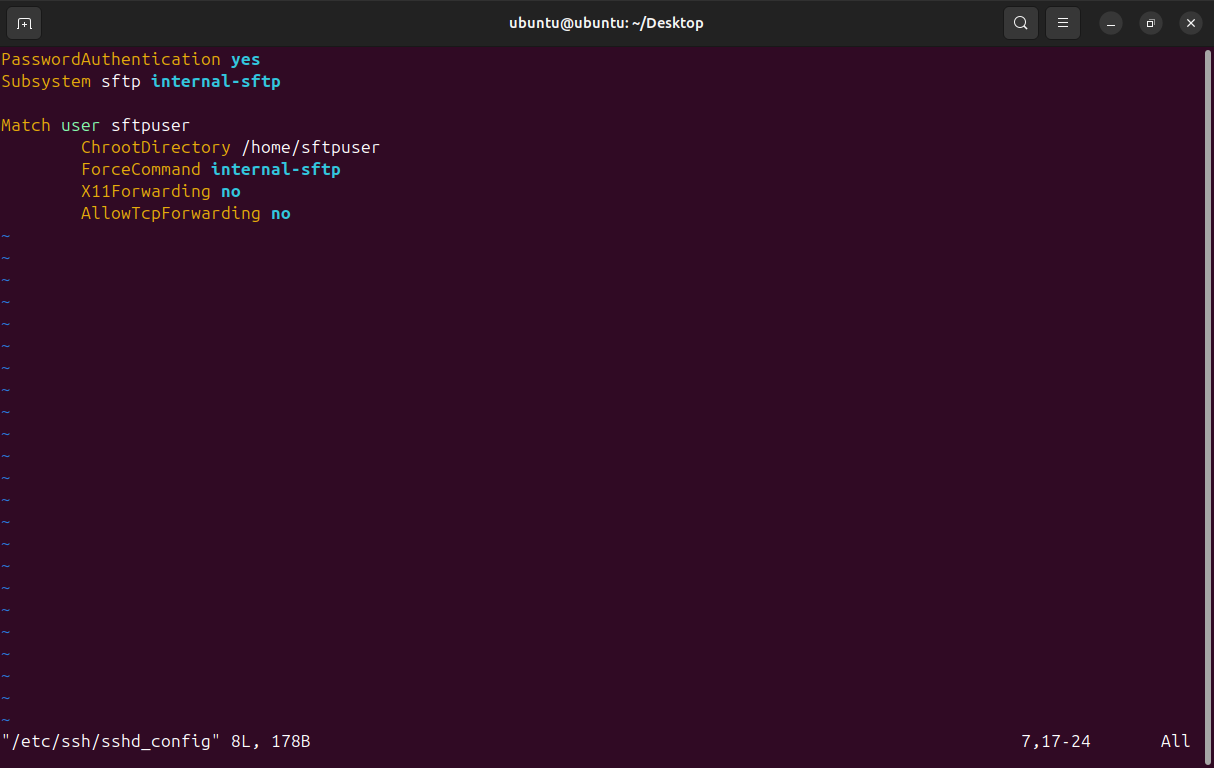
“Match User sftpuser

ChrootDirectory /home/sftpuser

ForceCommand internal-sftp

AllowTcpForwarding no

X11Forwarding no”



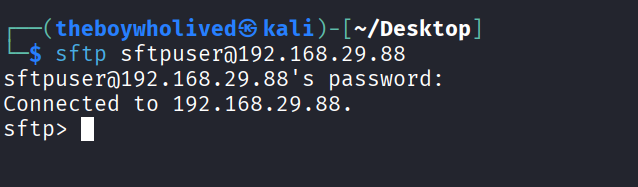
Save and exit. Then restart the SSH service:



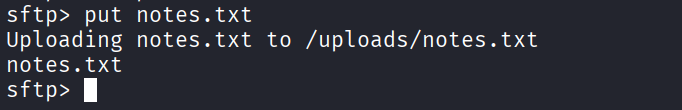
**5.6 File Transfer using SFTP**

**Command:**

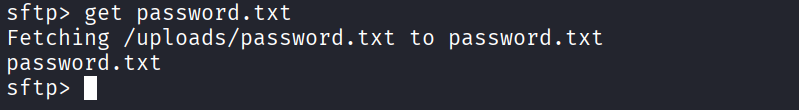
sftp sftpuser@192.168.29.88 Initiates an SFTP connection to the Ubuntu server.

From the **Kali Linux (client)** system, connect via SFTP:

To upload a file:



To download a file:



**6. Security Measures Implemented**

To ensure secure file transfer and prevent unauthorized access, the following security measures were implemented during the SFTP server setup:

**1. Use of SSH Protocol**

* The entire SFTP system operates over the **SSH (Secure Shell)** protocol, which provides encrypted communication.
* This ensures that credentials and file contents are not transmitted in plain text over the network.

**2. Chroot Jail for SFTP User**

* The sftpuser was restricted to a specific directory (/home/sftpuser) using a **chroot jail** configuration.
* This prevents the user from accessing other parts of the server's file system, enhancing isolation and security.

**3. Directory Permissions and Ownership**

* Proper directory structure and strict permissions were set:
  + Root owns /home/sftpuser
  + The upload directory (/home/sftpuser/uploads) is writable only by sftpuser
* This prevents privilege escalation and limits the user's capabilities.

**4. Disabled Unnecessary Services**

* For the sftpuser, services like **X11 forwarding** and **TCP port forwarding** were explicitly disabled using:

“AllowTcpForwarding no

X11Forwarding no”

**5. Dedicated SFTP-Only User**

* A separate user was created exclusively for SFTP access.
* The SSH configuration forces this user to use only the SFTP subsystem:

“ForceCommand internal-sftp”

**6. Password Protection**

* A strong password was set for the sftpuser account to prevent unauthorized login attempts.

**7. Testing the System**

**7.1 IP Configuration**

To ensure that the Kali Linux (client) can communicate with the Ubuntu Server (host), it is essential to confirm the **IP configuration** of both systems.

* On **Ubuntu Server**, check the IP address using:

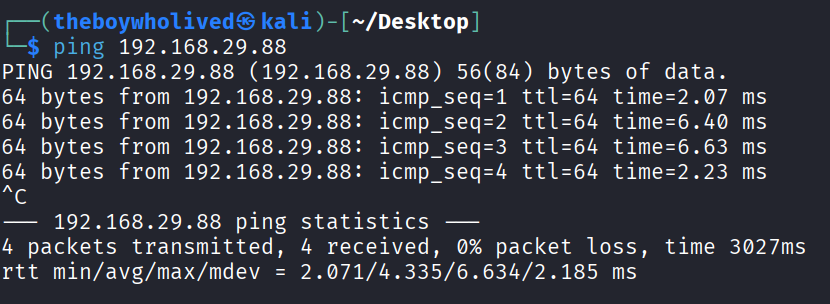
**Command:** ip a

ss

The IP address of the Ubuntu Server is 192.168.29.88

* On **Kali Linux**, check the IP address of the Ubuntu Server and confirm connectivity with:

ping 192.168.29.88



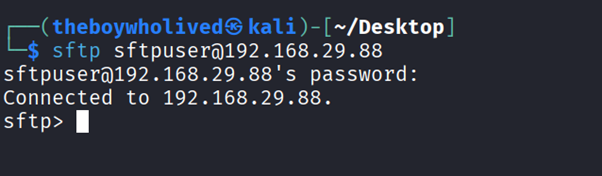
This ensures that both systems are on the same network and can communicate with each other.

**7.2 Kali → Ubuntu Connection Test**

Next, test the connection from **Kali Linux** to **Ubuntu Server** using the **SFTP** protocol.

From the Kali terminal, enter:

**Command:** sftp [sftpuser@192.168.29.88](mailto:sftpuser@192.168.29.88)

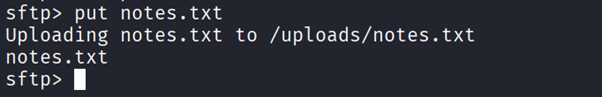


* You will be prompted to enter the password for sftpuser.
* Upon successful authentication, you should be in the SFTP shell (sftp>).

**7.3 Upload & Download File**

After establishing the connection, test the **file upload and download** functionality.

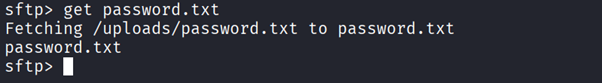
* **Uploading a file to the server**: On the Kali Linux terminal, use the put command:



**Command**: put notes.txt

This will upload notes.txt from Kali to the /uploads/ directory on the Ubuntu server.

* **Downloading a file from the server**: To download a file from the Ubuntu server to Kali, use the get command:



**Command:** get password.txt

This will download password.txt from the server to the local directory on Kali

**8. Conclusion**

In this project, we successfully implemented a **Secure File Transfer System** using **SFTP** on a Linux server, specifically with **Ubuntu Server** and **Kali Linux** as the client. The main goal was to establish a secure method for transferring files between two systems, ensuring confidentiality and preventing unauthorized access.

Key takeaways from the project:

* The **OpenSSH server** was configured to support **SFTP**, providing a secure channel for file transfers.
* A **restricted user environment** (chroot jail) was created for the SFTP user, ensuring that the user only had access to specific directories, preventing potential system breaches.
* File upload and download operations were successfully tested, confirming that the system was working as expected.
* Several **security measures** were implemented, including disabling unnecessary services, enforcing restricted file access, and using SSH for encrypted communication.

This project not only demonstrated the process of setting up a secure file transfer protocol but also provided insights into the importance of maintaining system security while managing file transfers in a real-world network environment.

Future enhancements could include implementing **SSL certificates**, setting up a **firewall** for added protection, and exploring automated **file synchronization** tools to make the system more versatile.