Linux - Setup FTP and sFTP Server on Raspberry PI

Introduction

Are you looking to set up an FTP server on your Raspberry PI, Rocky, Ubuntu or Debian-based operating system? vsftpd is a reliable and secure solution that allows you to transfer files between your computer and a remote server.

In this comprehensive guide, I'll walk you through the process of installing and using vsftpd step by step. Whether you're a beginner or an experienced user, this guide has got you covered.

Installing VSFTPD

Before we proceed let us ensure that our Raspberry Pi OS is running the latest available packages.

To update all packages on the device you will need to run the following two commands on the terminal.

```
sudo apt update
sudo apt upgrade -y
```

Updating all packages ensures that we shouldn't run in to any weird issues when installing vsftpd on to our Raspberry Pi.

Once the update process has completed we can now install the software we require.

Install vsftpd to your Raspberry Pi by using the command below.

```
sudo apt install vsftpd
```

Before you can start using vsftpd, you need to configure it to suit your needs. To do this, we'll need to make a few changes to the vsftpd configuration file. Open the file using the following command:

```
sudo cp /etc/vsftpd.conf /etc/vsftpd.conf.orig
sudo nano /etc/vsftpd.conf
```

Inside the configuration file, you'll find various settings that you can customize. For example, you can enable or disable anonymous login, set up local user accounts, and define the directories accessible to each user.

Change the following:

```
# Change or enable these values
anonymous_enable=NO
local_enable=YES
write_enable=YES
chroot_local_user=YES
# Add the following to the very end
user_sub_token=$USER
local_root=/home/$USER/ftp

userlist_enable=YES
userlist_file=/etc/vsftpd.userlist
userlist_deny=NO
```

Once you've made the necessary changes, save the file and exit the editor. To apply the new configuration, restart the vsftpd service by running the following command:

```
sudo systemctl restart vsftpd
```

Creating a User

To make use of vsftpd, you'll need to create user accounts that can access the FTP server. This allows you to control who can upload and download files. To create a new user account, use the adduser command followed by the desired username.

Script to use

Create a file in your home folder and, eg: sudo nano addFTP.sh

```
#!/bin/bash

# Configuration
USERNAME="$1"
PASSWORD="$2"
LOG_IDENTIFIER="create_user"  # Identifier for syslog entries
USER_LIST_FILE="/etc/vsftpd.userlist" # File to append usernames

# Function to log messages to syslog
```

```
log() {
  logger -t "$LOG_IDENTIFIER" "$1"
}

# Check if arguments are provided
if [ -z "$USERNAME" ]; then
  log "Error: Username is missing."
  exit 1
fi

if [ -z "$PASSWORD" ]; then
  log "Error: Password is missing."
  exit 1
fi

user_id=$(id -u "$USERNAME" &>/dev/null)

# Check if the user already exists
if [ ! $user_id = "" ]; then
  log "Error: User '$USERNAME' already exists."
  exit 1
fi

# Create the user
log "Creating user '$USERNAME'..."
sudo useradd "$USERNAME"

if [ $? -ne 0 ]; then
  log "Error: Failed to create user '$USERNAME'."
  exit 1
fi

# Set the password (using a non-interactive method)
log "Setting password for user '$USERNAME'..."
echo "$USERNAME:$PASSWORD" | chpasswd

if [ $? -ne 0 ]; then
  log "Error: Failed to set password for user '$USERNAME'."
  sudo userdel "$USERNAME"
  log "Cleaning up user '$USERNAME' due to password failure."
```

```
  exit 1
fi

log "User '$USERNAME' created successfully."
# Append username to the user list file
#sudo echo "$USERNAME" >> "$USER_LIST_FILE"
echo "$USERNAME" | sudo tee -a "$USER_LIST_FILE"

if [ $? -ne 0 ]; then
  log "Error: Failed to append username to '$USER_LIST_FILE'."
  sudo userdel "$USERNAME" # if the file append fails, delete the created user.
  exit 1
fi

log "Creating the ftp folder"
sudo mkdir -p /home/$USERNAME/ftp

log "Change the owner to nobody:nogroup"
sudo chown nobody:nogroup /home/$USERNAME/ftp

log "Adding the correct permissions"
sudo chmod a-w /home/$USERNAME/ftp

log "Showing the permissions on the ftp folder"
sudo ls -la /home/$USERNAME/ftp

log "Making a files folder within the '$USERNAME' folder"
sudo mkdir /home/$USERNAME/ftp/files

log "Changing the ownership on the files folder for the user"
sudo chown $USERNAME:$USERNAME /home/$USERNAME/ftp/files

log "Showing the permissions within the ftp folder"
sudo ls -la /home/$USERNAME/ftp

log "Change the user to have FTP access only"
sudo usermod -s /bin/ftponly $USERNAME

exit 0 # Exit with a zero status to indicate success.
```

Change the permissions on the file

```
chmod 777 addFTP.sh
```

Run like this to add a FTP user, eg: [script] [username] YOUR_PASSWORD

```
addFTP.sh steve YOUR_PASSWORD
```

Enabling FTPS (FTP over SSL)

o add an extra layer of security to your FTP server, you can enable FTPS, which is FTP over SSL. This encrypts the data transferred between the client and the server, ensuring that it cannot be intercepted by unauthorized users.

To enable FTPS, we need to create an SSL certificate. Run the following command to generate the certificate:

```
sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /etc/ssl/private/vsftpd.pem -out /etc/ssl/private/vsftpd.pem
```

During the certificate generation process, you will be prompted to provide information such as the country code, state, and organization name. Once the certificate is generated, we can proceed with the configuration.

Open the vsftpd configuration file again:

```
sudo nano /etc/vsftpd.conf
```

Scroll down to the bottom of the file and locate the RSA certificate settings. Replace the paths with the following:

```
rsa_cert_file=/etc/ssl/private/vsftpd.pem
rsa_private_key_file=/etc/ssl/private/vsftpd.pem
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
require_ssl_reuse=NO
ssl_ciphers=HIGH
```

Save the file and exit the editor. Restart the vsftpd service to apply the changes:

```
sudo systemctl restart vsftpd
```

Your FTP server is now configured to accept FTPS connections. You can test this by connecting to the server using an FTP client that supports FTPS.

Securing FTP User Access

To further enhance security, you can limit FTP users to only FTP access and disable shell access. This prevents users from accessing the command line interface on your server.

To restrict FTP user access, we need to create a limited shell. Run the following command to create the shell script:

```
sudo nano /bin/ftponly
```

Inside the editor, add the following line:

```
#!/bin/sh
echo "Limited to FTP access only"
```

Save the file and exit the editor. Next, make the shell script executable:

```
sudo chmod a+x /bin/ftponly
```

To enforce the limited shell, open the /etc/shells file:

```
sudo nano /etc/shells
```

Scroll to the bottom of the file and add the following line:

```
/bin/ftponly
```

Save the file and exit the editor. Finally, modify the user’s account to use the limited shell:

```
sudo usermod -s /bin/ftponly ftpuser
```

Now, if the user tries to access the server via SSH, they will be denied access. However, they can still connect to the FTP server using their FTP client.

resource: https://hub.tcno.co/linux/debian/vsftpd/

Revision #3
Created 12 March 2025 23:59:29 by Admin
Updated 8 October 2026 18:18:38 by Admin