FTP Server

Previous Next

File Transfer Protocol (FTP) is a TCP protocol for downloading files between computers. In the past, it has also been used for uploading but, as that method does not use encryption, user credentials as well as data transferred in the clear and are easily intercepted. So if you are here looking for a way to upload and download files securely, see the section on *OpenSSH* in Remote Administration instead.

FTP works on a client/server model. The server component is called an *FTP daemon*. It continuously listens for FTP requests from remote clients. When a request is received, it manages the login and sets up the connection. For the duration of the session it executes any of commands sent by the FTP client.

Access to an FTP server can be managed in two ways:

1. Anonymous
2. Authenticated

In the Anonymous mode, remote clients can access the FTP server by using the default user account called "anonymous" or "ftp" and sending an email address as the password. In the Authenticated mode a user must have an account and a password. This latter choice is very insecure and should not be used except in special circumstances. If you are looking to transfer files securely see SFTP in the section on OpenSSH-Server. User access to the FTP server directories and files is dependent on the permissions defined for the account used at login. As a general rule, the FTP daemon will hide the root directory of the FTP server and change it to the FTP Home directory. This hides the rest of the file system from remote sessions.

vsftpd - FTP Server Installation

vsftpd is an FTP daemon available in Ubuntu. It is easy to install, set up, and maintain. To install *vsftpd* you can run the following command:

```
sudo apt-get install vsftpd
```

Anonymous FTP Configuration

By default *vsftpd* is *not* configured to allow anonymous download. If you wish to enable anonymous download edit `/etc/vsftpd.conf` by changing:

```
anonymous_enable=Yes
```

During installation a *ftp* user is created with a home directory of `/srv/ftp`. This is the default FTP directory.

If you wish to change this location, to `/srv/files/ftp` for example, simply create a directory in another location and change the *ftp* user's home directory:

```
sudo mkdir /srv/files/ftp
sudo usermod -d /srv/files/ftp ftp
```

After making the change restart *vsftpd*:

```
sudo restart vsftpd
```

Finally, copy any files and directories you would like to make available through anonymous FTP to `/srv/files/ftp`, or `/srv/ftp` if you wish to use the default.

User Authenticated FTP Configuration

By default *vsftpd* is configured to authenticate system users and allow them to download files. If you want users to be able to upload

By default *vsftpd* is configured to authenticate system users and allow them to download files. If you want users to be able to upload files, edit `/etc/vsftpd.conf`:

```
write_enable=YES
```

Now restart *vsftpd*:

```
sudo restart vsftpd
```

Now when system users login to FTP they will start in their *home* directories where they can download, upload, create directories, etc.

Similarly, by default, anonymous users are not allowed to upload files to FTP server. To change this setting, you should uncomment the following line, and restart *vsftpd*:

```
anon_upload_enable=YES
```

Enabling anonymous FTP upload can be an extreme security risk. It is best to not enable anonymous upload on servers accessed directly from the Internet.

The configuration file consists of many configuration parameters. The information about each parameter is available in the configuration file. Alternatively, you can refer to the man page, `man 5 vsftpd.conf` for details of each parameter.

Securing FTP

There are options in `/etc/vsftpd.conf` to help make *vsftpd* more secure. For example users can be limited to their home directories by uncommenting:

```
chroot_local_user=YES
```

You can also limit a specific list of users to just their home directories:

```
chroot_list_enable=YES
chroot_list_file=/etc/vsftpd.chroot_list
```

After uncommenting the above options, create a `/etc/vsftpd.chroot_list` containing a list of users one per line. Then restart *vsftpd*:

```
sudo restart vsftpd
```

Also, the `/etc/ftpusers` file is a list of users that are *disallowed* FTP access. The default list includes root, daemon, nobody, etc. To disable FTP access for additional users simply add them to the list.

FTP can also be encrypted using *FTPS*. Different from *SFTP*, *FTPS* is FTP over Secure Socket Layer (SSL). *SFTP* is a FTP like session over an encrypted *SSH* connection. A major difference is that users of SFTP need to have a *shell* account on the system, instead of a *nologin* shell. Providing all users with a shell may not be ideal for some environments, such as a shared web host. However, it is possible to restrict such accounts to only SFTP and disable shell interaction. See the section on OpenSSH-Server for more.

To configure *FTPS*, edit `/etc/vsftpd.conf` and at the bottom add:

```
ssl_enable=Yes
```

Also, notice the certificate and key related options:

```
rsa_cert_file=/etc/ssl/certs/ssl-cert-snakeoil.pem
rsa_private_key_file=/etc/ssl/private/ssl-cert-snakeoil.key
```

By default these options are set to the certificate and key provided by the *ssl-cert* package. In a production environment these should be replaced with a certificate and key generated for the specific host. For more information on certificates see Certificates.

Now restart *vsftpd*, and non-anonymous users will be forced to use *FTPS*:

```
sudo restart vsftpd
```

To allow users with a shell of `/usr/sbin/nologin` access to FTP, but have no shell access, edit `/etc/shells` adding the *nologin* shell:

```
# /etc/shells: valid login shells
/bin/csh
/bin/sh
/usr/bin/es
/usr/bin/ksh
/bin/ksh
/usr/bin/rc
/usr/bin/tcsh
/bin/tcsh
/usr/bin/esh
/bin/dash
/bin/bash
/bin/rbash
/usr/bin/screen
/usr/sbin/nologin
```

```
/usr/sbin/nologin
```

This is necessary because, by default *vsftpd* uses PAM for authentication, and the `/etc/pam.d/vsftpd` configuration file contains:

```
auth    required        pam_shells.so
```

The *shells* PAM module restricts access to shells listed in the `/etc/shells` file.

Most popular FTP clients can be configured to connect using FTPS. The *lftp* command line FTP client has the ability to use FTPS as well.

References

1. See the vsftpd website for more information.
2. For detailed `/etc/vsftpd.conf` options see the vsftpd.conf man page.

Previous Next