Instalando e Configurando Servidor FTP com vsftpd no Linux

Fala pessoal, hoje venho ensinar como instalar e configurar um servidor FTP no Linux, mais precisamente no Ubuntu 12.04.

Dentre as muitas opções de programas que temos por ai, irei instalar o vsftpd, que parece ser muito seguro e é utilizado pela RedHat e OpenSuse além de outras empresas..

Entendendo o FTP

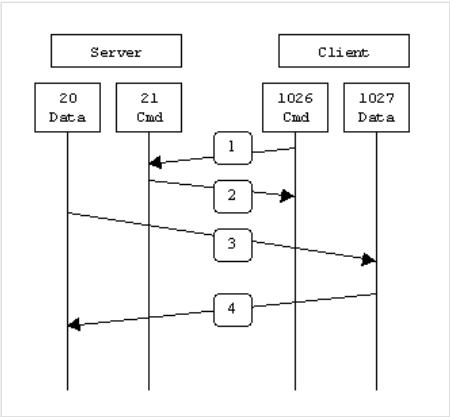
Por padrão o protocolo de transferência de arquivos utiliza duas portas para comunicação, a 21 para comandos e 20 para transferência dos dados.

Porém nem sempre a porta 20 é utilizada para transferência de dados, isso depende do modo que ele está configurado: ativou ou passivo.

Modo Ativo

No modo ativo, o cliente conecta-se através de uma porta randômica maior que 1023 (N > 1023) na porta 21 do servidor. Então o cliente começa ouvir na porta N + 1 e envia para o servidor essa porta, o servidor conecta-se de volta na porta especificada pelo cliente (N + 1) através da porta local 20.

Representação:

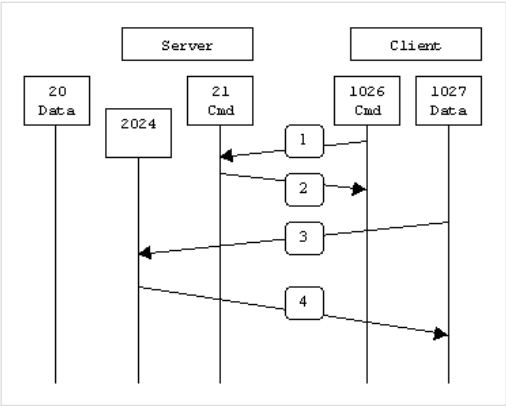


O problema com esse modo do FTP é que o servidor que começa uma conexão para transferência de dados, isso pode gerar problemas com o firewall do cliente pois está recebendo uma solicitação de fora e que geralmente é bloqueada.

Modo Passivo

Para resolver essa questão, foi desenvolvido o modo passivo (PASV). Nesse modo, o cliente abre duas portas randômicas maiores do que 1023 (N > 1023 e N+1). A primeira porta conecta-se com a porta 21 do servidor, mas invés de informar o servidor a porta para comunicação de dados, o cliente usará o comando PASV. Como resultado, o servidor vai abrir uma porta randômica (ou não, se configurarmos) P e então enviará essa porta P como resposta ao PASV para o cliente. Por fim, o cliente inicia o conexão através da porta N+1 na porta P do servidor para transferência de dados.

Representação:



Observe que no modo passivo não utilizamos a porta 20.

Essa foi uma breve abordagem sobre o funcionamento do FTP e dos modos ativo e passivo, recomendo esse excelente link caso você queira saber mais.

Instalação e Configuração

Agora que já temos uma base, vamos à instalação.

Primeiro, atualize a lista de pacotes e depois instale o vsftpd:

```
sudo apt-get update
```

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

Após a instalação, podemos testar acessando o servidor FTP localmente como anônimo (usuário sem senha), basta rodarmos o comando `ftp localhost` e digitarmos "anonymous" no usuário e deixar a senha vazia.

```
superuser@superman:~$ ftp localhost
Connected to localhost.
220 (vsFTPd 2.3.5)
Name (localhost:superuser): anonymous
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.
226 Directory send OK.
ftp>
```

Veja que estamos recebendo a mensagem "Consider using PASV" porque estamos no modo ativo.

Com nosso servidor instalado e funcionando, vamos alterar algumas configurações no arquivo que se encontra geralmente em `/etc/vsftpd.conf`.

Vamos começar desabilitando acesso anônimo de nosso servidor. Abra o arquivo de configuração para editar e defina `anonymous_enable=NO`.

```
sudo vim /etc/vsftpd.conf
```

```
# Allow anonymous FTP? (Beware - allowed by default if you comment this out).
anonymous_enable=NO
```

Se tentarmos acessar o servidor agora não conseguiremos pois estamos sem nenhuma forma de autenticação.

Toda vez que alterarmos o arquivo de configuração precisamos reiniciar o servidor FTP, use o comando `sudo service vsftpd restart` para isso.

Precisamos de usuário para acessar nosso servidor, vamos dizer ao vsftpd para permitir login com usuários locais (do sistema operacional). Abra o arquivo de configuração e descomente a linha `local_enable=YES`.

```
# Uncomment this to allow local users to log in.
local_enable=YES
```

Uma boa prática é criar um grupo para usuários de FTP no Linux, assim facilita a administração de permissões para esses indivíduos. Execute o comando abaixo para criar um grupo chamado "ftpusers".

```
sudo groupadd ftpusers
```

Já temos o grupo, vamos criar um usuário chamado "ftpuser", adicionar ele no grupo "ftpusers" e definir uma senha:

```
useradd -g ftpusers ftpuser
```

```
passwd ftpuser
```

Se tentarmos acessar nosso servidor FTP agora, veremos que precisamos criar uma pasta com o mesmo nome do usuário dentro de `/home`. Vamos criar esse diretório do usuário, mudar o owner para "ftpuser" e o grupo para "ftpusers". Também precisamos remover o acesso de escrita desse diretório, pois o vsftpd não permite que a pasta do usuário (somente ela, os subdiretórios podem ter) tenha acesso de escrita:

```
cd /home
```

```
sudo mkdir ftpuser
```

```
sudo chown ftpuser:ftpusers ftpuser
```

```
sudo chmod a-w ftpuser
```

Após as operações acimas, devemos ter isso:

```
superuser@superman:/home$ ls -l
total 8
dr-xr-xr-x 2 ftpuser   ftpusers  4096 Jun 22 14:59 ftpuser
drwxr-xr-x 7 superuser superuser 4096 Jun 22 14:46 superuser
superuser@superman:/home$
```

Como o diretório do usuário não tem acesso de escrita, iremos criar um diretório chamado `public` dentro de `/home/ftpuser` onde o usuário poderá armazenar seus arquivos e depois mudaremos o owner para o usuário e seu grupo (para que ele consiga ter acesso total)

```
sudo mkdir /home/ftpuser/public
```

```
sudo chown ftpuser:ftpusers /home/ftpuser/publi
```

A partir de agora já conseguimos acessar o servidor FTP com o usuário criado, só que tem algo de errado ai, o usuário consegue acessar qualquer diretório do sistema a partir do diretório dele, basta executar `cd /` para acessar a raiz do Linux por exemplo..

Descomente a linha `chroot_local_user=YES` para enjaular o usuário apenas no diretório dele, assim ele só poderá acessar subdiretórios de sua pasta.

```
# You may restrict local users to their home directories.  See the FAQ for
# the possible risks in this before using chroot_local_user or
# chroot_list_enable below.
chroot_local_user=YES
```

Até agora está tudo certo, porém ainda estamos utilizando o modo ativo, isso pode nos gerar problemas.. Vamos ativar o modo passivo, para isso precisamos definir `pasv_enable=YES` no arquivo de configuração.

```
# Enable passive mode
pasv_enable=YES
```

Nesse servidor não foi necessário, mas em alguns pode ser necessário definir o ip do servidor FTP no qual o cliente deve fazer a solicitação, se for seu caso, basta definir `pasv_address=XXX.XXX.XXX.XXX` (troque XXX.XXX.XXX.XXX pelo ip do seu servidor FTP) no arquivo de configuração.

Pode-se também definir uma faixa de portas na qual o servidor FTP irá usar para transferir arquivos, basta definir `pasv_min_port=9040` e `pasv_max_port=9050` no arquivo de configuração, nesse caso estariamos liberando as portas 9040 até 9050 para transferência no modo passivo.

Se você desejar pode mudar onde será o diretório inicial do usuário FTP (o padrão é `/home/nome_usuario`), basta definir `local_root=/var/$USER` no arquivo de configuração, caso você utilize a variável `$USER` do sistema no caminho (como nesse exemplo), defina também a linha `user_sub_token=$USER` para o vsftpd entender como uma variável. No exemplo dado, ao acessarmos o servidor FTP com o usuário "ftpuser" o vsftpd estaria procurando o diretório `/var/ftpuser`.

⚠ Não esqueça que o diretório inicial deve ser do usuário e grupo desejados e a raiz não pode ter permissão de escrita.

Para finalizar, muitas vezes podemos precisar que o diretório inicial do servidor FTP aponte para um outro diretório que não seja um subdiretório (que esteja fora da home), a melhor forma para fazer isso seria montar o diretório desejado dentro da pasta inicial do usuário. Temos aqui um exemplo: `sudo mount --bind /var/webapps/meu-blog /home/usuario_ftp/blog`, assim cada vez que acessarmos o diretório "blog" do usuário estaremos na verdade acessando o "meu-blog" dentro de "var".

Por hoje é só, não esqueça de deixar sua opinião nos comentários.

Até mais.

Written on June 22, 2013

IMPORTANTE
É necessário habilitar o modo de escrita para a pasta do usuário.
Execute:

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

É necessário habilitar o seguinte parâmetro:

```
write_enable=YES
```

Reinicie o vsftpd

```
sudo service vsftpd restart
```

Jayden Chua

Follow 134 Followers About

Setting up an FTP server on Ubuntu 18.04 on AWS

Jayden Chua Mar 26, 2020 · 6 min read ★

Here is me documenting my journey as I set up a temporary FTP server for quick access before removing it at the end of the project.

Photo by Taylor Vick on Unsplash

The quick overview as to why I needed this article was to serve as a reminder for my future self on how to get this running quickly. So, recently I needed to run an FTP server quick and dirty because I needed to get a huge amount of files quickly to another party. Unfortunately, other storage solution wasn't possible for this project, such as S3, so here we go…

quickly and set this up again in the blink of an eye.

Thanks for the many wonderful tutorials out in the wild, this should be a piece of cake, but there are some small little information that would be good to know once you're setting this up.

So I've split this article into 2 sections. Section 1 is how to setup pure insecure FTP, next, the second part is FTP with TLS. All these will be done on Amazon AWS, feel free to use this on other cloud providers and let me know if you face some issues.

. . .

Setup plain simple FTP

So assuming you already have have a working instance of the free tier EC2 instance on AWS, or some other cloud provider, let's start.

1. Install vsftpd

Just install from `apt-get` on ubuntu with the following commands

```
$ sudo apt-get update && sudo apt-get install vsftpd
```

After the installation, the FTP server service should up and running so just check it with

```
$ sudo service vsftpd status
```

So, usually, we would setup a firewall on the server and that would be the best practice, but since this is usually handled by some security group at the cloud provider level, this next step is optional.

2. Configure firewall

Here we are going to allow the following ports to pass through.

1. 22 for SSH (Important! Since without this will lock you out from SSH)

So, to do that we will use `ufw` and the commands are as follows

```
$ sudo ufw allow OpenSSH
$ sudo ufw allow 20:21/tcp
$ sudo ufw allow 12000:12100/tcp

$ sudo ufw enable
```

Now that we have enabled, the `ufw` let's just check to make sure everything is up and running.

```
$ sudo ufw status
```

This should show something similar to the below.

```
To                         Action      From
--                         ------      ----
OpenSSH                    ALLOW       Anywhere
20/tcp                     ALLOW       Anywhere
21/tcp                     ALLOW       Anywhere
12000:12100/tcp            ALLOW       Anywhere
OpenSSH (v6)               ALLOW       Anywhere (v6)
20/tcp (v6)                ALLOW       Anywhere (v6)
21/tcp (v6)                ALLOW       Anywhere (v6)
12000:12100/tcp (v6)       ALLOW       Anywhere (v6)
```

Finally, for these ports that you are allowing, remember to add them to your AWS security groups if you're using AWS as your cloud provider.

3. Create User

Next, we are going to to create a user with the required credentials and access rights, for this example, we will be creating a FTP user with the username `ftpuser`.

```
$ sudo adduser ftpuser
```

command.

```
$ sudo nano /etc/ssh/sshd_config
```

Add the following line to the `/etc/ssh/sshd_config` file.

```
DenyUsers ftpuser
```

Finally, save and restart the SSH service

```
$ sudo service sshd restart
```

4. Access Rights

There are different ways to create the access rights, but I will assume we are using the use case where this user will only be able to upload to his own home directory. If you are interested in other use cases such as uploading to a web directory, consider following the links shown at the bottom of the page.

Let's start by creating the user folder.

```
$ sudo mkdir /home/ftpuser/ftp
```

Set the ownership of the ftp directory to `nobody:nogroup`

$ sudo chown nobody:nogroup /home/ftpuser/ftp

Next, set the permissions so that everyone will not be able to have write permissions. We do this by using the following command, the flag `a-w` can be read as 'all/everyone remove write permissions'

```
$ sudo chmod a-w /home/ftpuser/ftp
```

Then, we will create a files sub folder where the user is allowed to upload files to.

```
$ sudo mkdir /home/ftpuser/ftp/files
```

And, assign ownership to him.

```
$ sudo chown ftpuser:ftpuser /home/ftpuser/ftp/files
```

5. Configure FTP server

This is the part that tripped me up, and mostly, it was because of the passive modes. So just be aware, and I'll explain where you need to take extra precautions when you're there.

Configure the `vsftpd` configuration file located in `/etc/vsftpd`. But first, create a backup.

```
$ sudo cp /etc/vsftpd.conf /etc/vsftpd.conf.bak

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

Next, ensure you turn on or have these flags within the configuration file. Remember to change `pasv_address=x.x.x.x` with the IP address of your server. Also, ensure `listen=YES` is set to YES. Without both of these, you might face the warning message from your FTP client such as "***Server sent passive reply with unroutable address. Using server address instead.***".

```
listen=YES
listen_ipv6=NO
write_enable=YES
chroot_local_user=YES
local_umask=022
force_dot_files=YES

pasv_enable=YES
pasv_min_port=12000
```

```
pasv_address=x.x.x.x

user_sub_token=$USER
local_root=/home/$USER/ftp
```

Finally, restart the ftp server and check that everything is up and running properly with the following commands.

```
$ sudo systemctl restart vsftpd

$ sudo service vsftpd status
```

A proper working example would be something like.

```
● vsftpd.service - vsftpd FTP server
   Loaded: loaded (/lib/systemd/system/vsftpd.service; enabled;
vendor preset: enabled)
   Active: active (running) since Thu 2020-03-26 04:16:27 UTC; 3s
ago
  Process: 26682 ExecStartPre=/bin/mkdir -p /var/run/vsftpd/empty
(code=exited, status=0/SUCCESS)
 Main PID: 26693 (vsftpd)
    Tasks: 1 (limit: 1152)
   CGroup: /system.slice/vsftpd.service
           └─26693 /usr/sbin/vsftpd /etc/vsftpd.conf
```

If there would be an error, you will know with something like the following.

```
● vsftpd.service - vsftpd FTP server
   Loaded: loaded (/lib/systemd/system/vsftpd.service; enabled;
vendor preset: enabled)
   Active: failed (Result: exit-code) since Thu 2020-03-26 04:13:21
UTC; 1min 1s ago
  Process: 26445 ExecStart=/usr/sbin/vsftpd /etc/vsftpd.conf
(code=exited, status=2)
  Process: 26434 ExecStartPre=/bin/mkdir -p /var/run/vsftpd/empty
(code=exited, status=0/SUCCESS)
 Main PID: 26445 (code=exited, status=2)
```

. . .

Setup FTP with TLS

The better approach would be to use FTP over TLS.

To do this we will follow all the steps above and more. We'll make the following small changes.

1. Allow TLS ports in Firewall

Add the following ports to `ufw`.

```
sudo ufw allow 990/tcp
```

Remember to add them in Security Group also.

Create new Certificate

Let's create a certificate with `openssl` with the following commands. You can hit enter for all the questions to use the default values.

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

Update FTP Configuration

Finally, we just need to add or uncomment the following lines to `/etc/vsftpd.conf`. To revert to using the insecure FTP, just change `ssl_enable=NO`.

```
ssl_enable=YES

rsa_cert_file=/etc/ssl/private/vsftpd.pem
rsa_private_key_file=/etc/ssl/private/vsftpd.pem
allow_anon_ssl=NO
force_local_data_ssl=YES
force_local_logins_ssl=YES
ssl_tlsv1=YES
```

```
ssl_ciphers=HIGH
```

Finally, restart server and check that it is up.

```
$ sudo systemctl restart vsftpd

$ sudo service vsftpd status
```

. . .

What's Next

I'll be creating a script to automate this, so I can forget how to do this completely. Also, I will want to try adding the steps to create sFTP server in this article in future. Leave me a comment if you would like me to finish this article sooner!

References

- https://devanswers.co/installing-ftp-server-vsftpd-ubuntu-18-04/
- https://www.gosquared.com/blog/fix-ftp-passive-mode-problems-on-amazon-ec2-instances
- https://stackoverflow.com/questions/28356796/aws-ec2-passive-ftp-server-sent-passive-reply-with-unroutable-address-using-s/43887600

Ftp Server Ubuntu Vsftpd AWS Aws Ec2