

vsftpd FTP Server

FTP is the user interface the internet standard File Transfer Protocol. The program allows a user to transfer files to and from a remote network site.

Service name:

vsftpd

Package:

vsftpd (Acronym for Very Secure File Transfer Protocol Daemon)

Config:

/etc/vsftpd/vsftpd.conf

Access Control:

/etc/hosts.allow

/etc/hosts.deny

iptables

SELinux

Built in access control

Log Files:

/var/log/xferlog

FTP is a very simple service, this is because it has a very simple default configuration that provides an out of the box configuration that works. FTP is a very complex program with many features. For now, lets just get FTP installed and running:

```
# yum -y install vsftpd
# service vsftpd start
# chkconfig vsftpd on
```

These three commands will install, enable, and start the vsftpd server in the default anonymous download configuration. Of course we are making the assumption that you are connected to a yum repository with the package vsftpd.

This package provides the directory /var/ftp. This can be learned before or after install by running the following command:

Advanced configuration of vsftpd

Create firewall rules to allow connectivity over the network to your FTP server:

vsftpd is a PASV FTP server allowing initial authenticating connections to be made on 21/tcp and data transfers via a high unprivileged port.

As we don't know what port is going to be assigned to users when they've authenticated we need some help from a kernel module called `nf_conntrack_ftp`.

This module needs to be loaded persistently in order to work. Use the following commands to load it system start-up:

```
# echo "modprobe nf_conntrack_ftp" >> /etc/sysconfig/modules/local.modules
# chmod +x /etc/sysconfig/modules/local.modules
```

Now that we have some sort of mechanism to track the high unprivileged port numbers we can create the firewall rules needed to allow connectivity. In this case, we will be doing state matching as well.

The 2nd command below allows any connection to a port which is related to an established connection already made. So if the user successfully authenticated to the FTP service on 21/tcp and was allocated 40000/tcp as their data port, then iptables would allow connectivity to 40000/tcp because that connection is related to the already established connection to 21/tcp for that user.

Without the 2nd command, users would be able to authenticate but will not be able to browse the contents of the vsFTPD server.

```
# iptables -I INPUT -p tcp --dport 21 -j ACCEPT
# iptables -I INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
# service iptables save
```

Adapt the `/etc/vsftpd/vsftpd.conf` to your needs:

Important parameters:

`anonymous_enable` specifies that the anonymous user (who is the Linux system user `ftp`) is allowed to login without using a password.

`local_enable` specified whether Linux users (other than those listed in `/etc/vsftpd/ftpusers`) are allowed to login to the FTP server. Be very careful when enabling this setting because FTP is a plain text protocol and is therefore susceptible to man-in-the-middle attacks.

`write_enable` defines if FTP writes (uploads) are allowed. Note that you would also need to allocate Linux system permissions.

`listen` enabled the FTP server to be used with IPv4 on all interfaces

`userlist_enable` specifies whether the users listed in `/etc/vsftpd/ftpusers` are allowed to login or whether they're defined. When the value is set to YES it means that the users listed should NOT be allowed access to FTP. If the value is set to NO then only the users listed in the file would be allowed to login.

`tcp_wrappers` enables or disabled support for TCP Wrappers for the vsFTPD

Notes:

- The basic configuration file allows anonymous users to download files from `/var/ftp`
- For the anonymous FTP user to see files, the ftp Linux user needs r-x permissions
- Even though `local_enable` is set to YES, SELinux prevents users seeing their home directories
- While `write_enable` is also set to YES, SELinux prevents any type of write action
- In addition, directories require the context `public_content_rw_t` in order to allow uploads
- `man ftpd_selinux` contains valuable information about fcontext types and booleans
- Don't forget to run `service vsftpd restart` after editing `/etc/vsftpd/vsftpd.conf`

Examples:

```
anonymous_enable=YES
local_enable=YES
write_enable=NO
local_umask=022
dirmessage_enable=YES
xferlog_enable=YES
connect_from_port_20=YES
xferlog_std_format=YES
listen=YES
pam_service_name=vsftpd
userlist_enable=NO
tcp_wrappers=YES
```

Explanation:

As a result of `anonymous_enable=YES` and `local_enable=YES`, users may authenticate as anonymous or as a normal Linux user with read only permissions as we've disabled any form of writes through `write_enable=NO`.

If write access were allowed, we would need the `fcontext public_content_rw_t` to be set on the directory; the boolean `allow_ftp_anon_write` would need to be enabled should the anonymous user be granted write access; and Linux filesystem permissions would need to be granted as well.

The FTP service is bound to all IPv4 interfaces and supports TCP Wrappers because of the parameters `listen=YES` and `tcp_wrappers=YES` respectively.

Because `userlist_enable=NO` only users listed in the file `/etc/vsftpd/user_list` would be allowed access to the FTP server.

Lab activity

Create an FTP server which allows anonymous uploads to `/var/ftp/pub/incoming`. Users cannot download files which are owned by the user `ftp`.

Make sure that your FTP server is protected by a firewall.