Gerald on IT
All about my opinions on IT and much, much more ...

Enabling SFTP-only access on Linux

Recently I had the need to share a zip file with a bunch of people that was big enough not to fit into email anymore. So I wanted to get it onto my server so that folks could grab it via SFTP from there. SFTP is setup by default on my Linux environment, so them accessing the machine was trivial. However, I didn't want to give them full access to the entire machine where they could randomly up- and download files anywhere. What I needed was some way of giving them a user which was self-contained, with no SSH privileges and bound to a single location on the filesystem. Luckily, setting something like this up was much easier than I thought, and here is how you can do it yourself. Note, all commands below are executed as the `root` user:

tl;dr

1. `useradd <your sftp user> -s /sbin/nologin -M`
2. `passwd <your sftp user>`
    1. `Enter your sftp user password and confirm`
3. `vi /etc/ssh/sshd_config`
4.
```
Match User <your sftp user>
   ChrootDirectory <your sftp user directory>
   ForceCommand internal-sftp
   AllowTcpForwarding no
   X11Forwarding no
```
5. `service sshd restart`

Creating the SFTP user

The first step is to create a dedicated Linux user that people can use to `sftp` into the server. Creating one is rather simple with the `useradd` command. The `-s` option allows me to specify which shell the user should get when logging on into the machine via `ssh`. In this case, as I only want to allow people to `sftp` into

the machine, I define the shell as `/sbin/nologin`. This is not a shell but a command that "*politely refuses a login*", as the man page for this program says. That way, I disable SSH access into my server for that user. The other option I pass on is the `-M` option. This tells `useradd` not to create a home directory for the user:

```
# useradd sftp_download -s /sbin/nologin -M
#
```

Next I specify a password for the user. This is easily done via the `passwd` command:

```
# passwd sftp_download
Changing password for user sftp_download.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
#
```

Configuring the SFTP location

Once the user itself is setup, it is time to think about the specific location that I want the user to have access to via `sftp`. In my case, I just want the user to be able to get files from a new directory `/download`, so I just go ahead and create it:

```
# mkdir /download
#
```

In order to allow `sftp` access for the user, I have to change the SSH configuration file located at `/etc/ssh/sshd_config`. There are five instructions that I add to the configuration file: `Match User`, `ChrootDirectory`, `ForceCommand`, `AllowTcpForwarding` and `X11Forwarding`. The addition as a whole to the configuration file looks like this:

```
Match User sftp_download
   ChrootDirectory /download
   ForceCommand internal-sftp
   AllowTcpForwarding no
   X11Forwarding no
```

- `Match User`: This introduces a conditional block that is only executed when the condition is matched, in this case if the user is `sftp_download`.
- `ChrootDirectory`: *Specifies the pathname of a directory to chroot to after authentication. At session startup sshd checks that* ***all components of the pathname are root-owned directories which are not writable by any other user or group.*** *After the chroot, sshd changes the working directory to the user's home directory.* This is the location that I have just created, `/download`. **Note: chroot only accepts directories that are owned by root and not writeable by any other user or group! If you want to enable write access for your SFTP user, you will have to create a subdirectory that is owned by the user, without referencing that subdirectory in the ChrootDirectory directive.**

- `ForceCommand internal-sftp` : *Forces the execution of the command specified by ForceCommand, ignoring any command supplied by the client. Specifying a command of internal-sftp will force the use of an in-process SFTP server that requires no support files when used with ChrootDirectory.*
- `AllowTcpForwarding no` : *Specifies whether TCP forwarding is permitted.*
- `X11Forwarding no` : *Specifies whether X11 forwarding is permitted.*

After the changes are added it is best to verify whether the `sshd_config` file is still valid.

Note: if the `sshd_config` file is invalid and you restart your sshd service, the sshd daemon won't startup anymore. I highly recommend that you verify the `sshd_config` file before restarting the service!

Validation can be done by running `sshd` in test mode, via the `-t` option. If `sshd` returns with no errors, the configuration file is fine and I am good to go to restart the service via `service sshd restart`.

```
# vim /etc/ssh/sshd_config
# sshd -t
# service sshd restart
Redirecting to /bin/systemctl restart sshd.service
#
```

Now that the service is restarted, I'm all set. So let's go ahead and test it. I create a new file `mytestfile.txt` and then `sftp` into as the `sftp_download` user and see whether I can break out into the parent directory:

```
# touch /download/mytestfile.txt
# sftp sftp_download@localhost
sftp_download@localhost's password:
Connected to localhost.
sftp> ls
mytestfile.txt
sftp> cd ..
sftp> ls
mytestfile.txt
sftp> bye
#
```

As you can see, I cannot go up a level in the directory structure but am self-contained to the folder I specified in the `sshd_config` file.

Conclusion

Enabling SFTP-only access on a Linux machine can be done in just a few steps. This is a great way if you want to allow other people to download files from your server without giving them full access to the server.

Share this:

Like

One blogger likes this.

Related

[Creating an Oracle Database Vagrant box]

In "Technology"

[Deploying a Kubernetes cluster with Vagrant on Virtual Box]

In "Technology"

[Creating an Oracle Database Docker image]

In "Technology"

Author: Gerald

Developer, Oracle expert, performance enthusiast and genuine technology geek.
View all posts by Gerald

Gerald / May 2, 2018 / Technology / Linux, sftp

This site uses Akismet to reduce spam. Learn how your comment data is processed.

Gerald on IT /