Setting Up a Basic File Server

This article is aimed at setting up a simple, networked file server with a small amount of *trusted* users. Typically, we would expect such users to not do anything intentionally malicious, but they may accidentally do something harmful. As with any data storage, **make frequent backups**.

These are *not* all considered to be best practices. If you are storing identifying, proprietary, sensitive, or otherwise private data, consult with a data security expert. Examples of such data include anything involving human subjects, medical records, privately-owned datasets, and more.

Setting up local directories

Before directories can be served over a network, the file system must be correctly set up.

Data Folder Hierarchy

When data is intended to be read by all users, I prefer the following file structure:

```
/srv
|-- data/
| |-- dataset_type1/
| |-- dataset_type2/
| .
| .
| .
|-- tmp/
| |-- dataset_type1/
| |-- dataset_type2/
| .
| .
| .
```

Here, the directory `/srv/data` contains the actual raw data files. Its permissions are `rwxr-xr-x`, and it is owned by `root`. Here, all users have read and execute access, but only root can write and modify files. These permissions can be set with:

```
chmod -R a+rx /srv/data/
```

However, you still want to allow non-root users to upload their data for later inclusion in `/srv/data`. This is the purpose of `/srv/tmp`. This directory acts as a volatile "holding space" for all users to have write access. Once a user uploads data to the temporary directory, the system administrator may move the data into the correct location in `/srv/data`. This directory is still owned by root, but its permissions are `rwxrwxrwt`, meaning that *all* users may read, write, and execute all files. For an extra safeguard, set the sticky bit for this directory, so that only file and directory owners can delete or rename their data. This can be done with the command:

```
chmod -R a+trwx /srv/tmp
```

Note that these are actually the same permissions set for the system's `/tmp` directory.

Mounting External Storage

It may be necessary to store or access data located on removable drives, such as a USB drive, or an external hard drive. In this case, I prefer to mount these drives to the directory `/mnt/ext/short_drive_name`, where "`short_drive_name`" is some easily recognizable name for this particular device /partition.

Then, make a symbolic link inside your data directory with:

```
sudo ln -s /mnt/short_drive_name/data_dir /srv/data/ext_dataset_name
```

Make sure the link itself is owned by root.

As for the actual mount configuration, I prefer the following style added in /etc/fstab:

```
UUID=partition_uuid /mnt/ext/short_drive_name filesystem_format umask=022,async,auto,rw,nofail 0 0
```

where `partition_uuid` is your storage device's UUID and `filesystem_format` is the format of your device (ntfs, ext4, etc.). To locate your device's UUID, try the `blkid` command. If that is not available, then the information given by the commands `lsblk` and `ls -l /dev/disk/by-uuid` should be enough to discover your UUID. This will only allow `root` to write to the disk, but other users may read from it.

Serving Files Over the Network

TODO