Installation - Linux Source

Version 13, last updated by meijer at 2014-10-09

This is the 12 step recipe for a manual installation of FileSender version 1.6+

- Client and Server Requirements
- Step 1 - Install Apache2
- Step 2 - Configure Apache2 with SSL
- Step 3 - Install PHP5
- Step 4 - Install and configure PostgreSQL
- Step 5 - Install and configure SimpleSAMLphp
- Step 6 - Install the FileSender package
- Step 7 - Create the FileSender user and database
- Step 8 - Configure PHP5
- Step 9 - Configure your Apache virtual host
- Step 10 - Configure your FileSender installation
- Step 11 - Configure the FileSender clean-up cron job
- Step 12 - Start using FileSender
- Support and Feedback

Please help us with the FileSender one hour installation time guarantee

We've all used more time than we wanted on a first time install of a piece of open source software, just to see whether it does something interesting. We do not want FileSender to be like that.

If you follow this documentation you should be up and running within an hour with a working FileSender installation that uses Feide RnD's OpenIdP to authenticate users. We assume:

- you install on Debian Linux or Ubuntu
- you are a reasonably skilled sysadmin
- you follow these installation instructions

NOTE: you'll spend even less time if you use the Debian packags or the RPMs.

If you you notice mistakes in this documentation, or if it took you more than an hour to install FileSender, please let us know on filesender-dev@filesender.org and help us improve the documentation for those that come after you!

About this documentation

This is the installation documentation for installing **FileSender 1.6(.x)** on Linux. This guide is written for installation

from source on the Debian Linux platform but any Linux variant should work with some modifications (most notably about installing the required additional software packages).

This documentation was tested with:

- standard Debian/GNU Linux 7.5 (wheezy) as released by Debian
- PostgreSQL 9.1
- Apache 2.2.22
- PHP 5.4.4-14+deb7u11
- SimpleSAMLphp 1.12

Debian and RPM packages

Debian and RPM packages are also available to automate most of the steps below. Please see Installation - Debian Ubuntu and Installation - RPM for instructions on how to get and use them. The Debian and RPM packages will install the filesender and simplesamlphp software in the /usr/share/ directory tree. In the examples below /usr/local/filesender is used as base directory. Please adapt the examples below where appropriate when using the packages.

Client and Server Requirements

See Requirements.

Step 1 - Install Apache2

Install Apache2 from the Debian package repository:

```
apt-get install apache2
```

Step 2 - Configure Apache2 with SSL

The default FileSender configuration is for an installation that only works over SSL. Make your Apache SSL-enabled with the following commands:

```
a2ensite default-ssl
a2enmod ssl
service apache2 restart
```

This will give you a working SSL-enabled Apache server with a self-signed SSL-certificate, allowing you to verify your FileSender installation works with HTML5 compatible browsers. Uploads in older browsers use a Flash component. They will not work until you install an SSL certificate issued by a CA recognised by popular browsers. This is a Flash specific issue explained in more detail in the FAQ entry on SSL certificates and Flash

If you want to test Flash uploads by running FileSender over HTTP-only, change *$config['forceSSL']* in config.php to false. This will constitute a security risk. Disabling SSL means all file uploads and downloads are unprotected and any data transferred to and from your FileSender installation can be stolen. Disabling SSL is not advisable.

Step 3 - Install PHP5

Install PHP5 from the Debian package repository:

```
apt-get install php5 libapache2-mod-php5
```

Step 4 - Install and configure PostgreSQL

Note: FileSender also supports MySQL. If you prefer to use MySQL for FileSender, please refer to the MySQL installation manual for your version of MySQL. If you would like to see detailed instructions for how to install MySQL for use with FileSender, please consider contributing documentation.

Step 4a - Install PostgreSQL and the PostgreSQL module for PHP:

```
apt-get install postgresql php5-pgsql
```

Step 4b - Verify the PostgreSQL configuration

FileSender uses password based database logins and by default assumes that PostgreSQL is configured to accept password based sessions on 'localhost'. You should check and when needed change the relevant settings in the PostgreSQL pg_hba.conf configuration file. This file should have the following entries with **md5** listed as METHOD for local IPv4 and IPv6 connections:

```
# Database administrative login by UNIX sockets
local   all         postgres                    peer
# TYPE  DATABASE    USER        CIDR-ADDRESS        METHOD
# "local" is for Unix domain socket connections only
local   all         all                         peer
# IPv4 local connections:
host    all         all         127.0.0.1/32        md5
# IPv6 local connections:
host    all         all         ::1/128             md5
```

On Debian based systems this file will be in `/etc/postgresql/<version>/main/pg_hba.conf`. On Red Hat/Fedora based systems this file will be in `/var/lib/pgsql/data/pg_hba.conf`. When changing the pg_hba.conf file you'll have to restart the database server with (version number may be different or not needed depending on your system):

```
service postgresql reload
```

Step 5 - Install and configure SimpleSAMLphp

SimpleSAMLphp helps you use nearly any authentication mechanism you can imagine. Following these instructions will set you up with a SimpleSAMLphp installation that uses Feide RnD's OpenIdP to authenticate users. When you move to a production service you probably want to change that to only support authentication sources of your choice.

- **Step 5a: Download SimpleSAMLphp 1.12.0.** Other (later or older) versions will probably work but we tested with version 1.12.0.

```
cd /root
mkdir downloads
wget https://simplesamlphp.org/res/downloads/simplesamlphp-1.12.0.tar.gz
```

NOTE: you will of course remember to check the sha1 hash of the tar file, right?

- **Step 5b - Extract it in a suitable directory and create symlink:**

SECURITY NOTE: we only want *the user interface files* to be directly accessible for the world through the web server, not any of the other files. We will not extract the SimpleSAMLphp package in the /var/www directory (the standard Apache document root) but rather in a specific /usr/local tree. We'll point to the SimpleSAML webroot with a web server alias

```
mkdir /usr/local/filesender/
cd /usr/local/filesender
tar xvzf /root/downloads/simplesamlphp-1.12.0.tar.gz
ln -s simplesamlphp-1.12.0 simplesaml
```

- **Step 5c - Copy standard configuration files to the right places:**

```
cd /usr/local/filesender/simplesaml
cp -r config-templates/*.php config/
cp -r metadata-templates/*.php metadata/
```

To tailor your SimpleSAMLphp installation to match your local site's needs please check its installation and configuration documentation. When connecting to an Identity provider make sure all the required attributes are sent by the identity provider. See the section on IdP attributes in the Reference Manual for details.

Step 6 - Install the FileSender package

- **Step 6a**: **Download the FileSender software** and compare the hash with the one published on the FileSender 1.6 download page.

```
cd /root/downloads
wget http://downloads.filesender.org/filesender-1.6.tar.gz
openssl dgst -sha1 filesender-1.6.tar.gz
```

Verify the hash digest with the one on the FileSender download page for your release

- **Step 6b - Extract the FileSender tarbal**

SECURITY NOTE: we only want the *user interface files* to be directly accessible for the world through the web server, not any of the other files. We will not extract the FileSender package in the /var/www directory (the standard Apache document root) but rather in a specific /usr/local tree. We'll point to the FileSender webroot with a web server alias

```
cd /usr/local/filesender
tar xvzf /root/downloads/filesender-1.6.tar.gz
ln -s  filesender-1.6 filesender
```

This will create a directory 'filesender-1.6' with a symlink 'filesender' pointing to it. Using the symlink makes upgrading easier.

- **Step 6c - initialise config file and set permissions right**. Make the files, tmp and log directories writable by the web daemon user (www-data), copy the config file in place from the template and allow the web deamon user to read the config.php configuration file:

```
cd /usr/local/filesender/filesender
cp config/config-dist.php config/config.php
chown www-data:www-data tmp files log
chmod o-rwx tmp files log
chgrp www-data config/config.php
```

NOTE: we ship the FileSender tarball with config-dist rather than config.php to make life easier when building Debian packages and RPMs.

- The directory structure and permissions should now be as follows, carefully check the entries marked in **bold and orange background**:

```
root@ip-10-227-179-45:/usr/local/filesender/filesender# ls -l config
total 40
-rw-r----- 1 root root    16431 Apr  3 13:33 config-dist.php
-rw-r----- 1 root www-data 16431 Jul  1 15:18 config.php

root@ip-10-227-179-45:/usr/local/filesender/filesender# ls -la
total 92
drwxr-xr-x 14 root    root     4096 Apr  4 19:44 .
drwxr-sr-x  4 root    staff    4096 Jul  1 15:17 ..
-rw-r--r--  1 root    root    22495 Apr  4 16:43 CHANGELOG.txt
drwxr-xr-x  2 root    root     4096 Apr  4 19:44 classes
drwxr-xr-x  2 root    root     4096 Jul  1 15:18 config
drwxr-xr-x  2 root    root     4096 Apr  4 19:44 config-templates
drwxr-xr-x  2 root    root     4096 Apr  4 19:44 cron
drwxr-x---  2 www-data www-data 4096 Apr  4 19:44 files
drwxr-xr-x  2 root    root     4096 Apr  4 19:44 includes
-rw-r--r--  1 root    root     2547 Apr  4 19:19 INSTALL.txt
drwxr-xr-x  2 root    root     4096 Apr  4 19:44 language
-rw-r--r--  1 root    root     1658 Apr  4 19:43 LICENCE.txt
drwxr-x---  2 www-data www-data 4096 Apr  4 19:44 log
drwxr-xr-x  2 root    root     4096 Apr  4 19:44 pages
-rw-r--r--  1 root    root     2262 Apr  4 19:19 README.txt
drwxr-xr-x  2 root    root     4096 Apr  4 19:44 scripts
drwxr-x---  2 www-data www-data 4096 Apr  4 19:44 tmp
drwxr-xr-x  7 root    root     4096 Apr  4 19:44 www
```

Step 7 - Create the FileSender user and database

- **Step 7a -** Create the PostgreSQL user and database to be used by FileSender

Create the database user `filesender` without special privileges and with a password. The command will prompt you to specify and confirm a password for the new database user. *This is the password you need to configure in the FileSender configuration file later on.*

```
$ sudo -u postgres createuser -S -D -R -P filesender
Enter password for new role: <secret>
Enter it again: <secret>
```

This will create a database user **filesender** without special privileges, and with a password. This password you will have to configure in the filesender config.php later on.

NOTE: FileSender also supports MySQL. Please consult the MySQL manuals on how to create a MySQL database user. Please help us improve the documentation and send us the MySQL equivalent of this PostgreSQL instruction

- **Step 7b -** Create the filesender database with UTF8 encoding owned by the newly created filesender user:

```
$ sudo -u postgres createdb -E UTF8 -O filesender filesender
$ psql -h localhost filesender filesender < /usr/local/filesender/filesender/scripts/
Password for user filesender: <secret>
NOTICE:  CREATE TABLE will create implicit sequence "files_fileid_seq" for serial col
NOTICE:  CREATE TABLE / PRIMARY KEY will create implicit index "files_pkey" for table
CREATE TABLE
CREATE SEQUENCE
NOTICE:  CREATE TABLE / PRIMARY KEY will create implicit index "logs_pkey" for table
CREATE TABLE
```

NOTE: when using MySQL the database initialisation script is in scripts/mysql_filesender_db.sql

Step 8 - Configure PHP5

NOTE: a sample settings file is provided with FileSender in `config-templates/filesender-php.ini`. If you don't feel like manually editing your php.ini file, copy the filesender-php.ini file to your `/etc/php.d/` or `/etc/php5/conf.d/` directory to activate those settings. Please adapt the sample settings file where needed.

For manual edits, edit /etc/php5/apache2/php.ini and execute the following changes:

Step 8a: to allow for max. 2 GB Flash uploads change these settings to the values indicated:

```
max_input_time = 3600 ; in seconds
upload_max_filesize = 2047M ; in M, the default value is 2MB
post_max_size = 2146446312 ; in M, 2047M + 10K
```

NOTE: when you edit your FileSender config.php remember to change $config['max_flash_upload_size'] to match your upload_max_filesize. If they are not the same FileSender will use the lowest value as the actual maximum upload size for Flash uploads.

Step 8b - ensure the php temporary upload directory points to a location with enough space:

```
upload_tmp_dir = /your/temporarylocation
```

NOTE: You probably want to point this to the same directory you will use as your HTML5 upload temp directory ($config['site_temp_filestore']).
NOTE: that this setting is for all PHP-apps, not only for filesender.

Step 8c - Turn on logging:

```
log_errors = on
error_log = syslog
```

Step 8d - enable secure cookie handling to protect sessions

```
session.cookie_secure = On
session.cookie_httponly = On
```

Step 8e - Reload your Apache server to activate the changes to your php.ini:

```
service apache2 reload
```

Step 9 - Configure your Apache virtual host

Step 9a: make sure all traffic to http is redirected to https, to make things easy for your users

edit **/etc/apache2/sites-enabled/000-default** and add this line to the virtual host definition:

```
<VirtualHost *:80>
    ...

    Redirect / https://<your filesender site>

    ...
</VirtualHost>
```

Step 9b: create the URL aliases to your simplesamlphp and your FileSender web trees. This will make them accessible through your web server.

edit **/etc/apache2/sites-enabled/default-ssl** and add these lines to the virtual host definition:

```
<VirtualHost _default_:443>

...

    Alias /simplesaml /usr/local/filesender/simplesaml/www
     <Directory "/usr/local/filesender/simplesaml/www">
         AllowOverride None
         Order deny,allow
         Allow from all
    </Directory>

    Alias /filesender /usr/local/filesender/filesender/www
    <Directory "/usr/local/filesender/filesender/">
         Options FollowSymLinks
         DirectoryIndex index.php
         AllowOverride None
         Order deny,allow
         Allow from all
    </Directory>

...

</VirtualHost>
```

Step 9c - Reload your Apache server to activate the changes to the Apache config:

```
    service apache2 reload
```

Step 10 - Configure your FileSender installation

Edit your newly created config.php:

/usr/local/filesender/filesender/config/config.php

Carefully check and adapt the following settings. The **minimum required changes** to config.php are marked in **bold and orange background**:

```
$config['admin'] = '';
$config['adminEmail'] = 'Your.Address@example.org';
$config['Default_TimeZone'] = 'Europe/Berlin';
$config['site_defaultlanguage'] = 'en_AU'; // for available languages see the ./language directory
$config['site_name'] = '<my site> FileSender'; // Friendly name for your FileSender instance

$config['download_confirmation_to_downloader'] = true ; // DO NOT send copy of download confirmatic


$config['site_url'] = $prot . $SERVER['SERVER_NAME'] . '/filesender/'; // URL to Filesender
$config['site_simplesamlurl'] = $prot . $SERVER['SERVER_NAME'] . '/simplesaml/';
$config['forceSSL'] = true;



$config['site_filestore'] = '/usr/local/filesender/filesender/files/';
$config['site_temp_filestore'] = '/usr/local/filesender/filesender/tmp/';
$config['site_simplesamllocation'] = '/usr/local/filesender/simplesaml/';
$config['log_location'] = '/usr/local/filesender/filesender/log/';



$config["db_type"] = "pgsql";// pgsql or mysql
$config['db_host'] = 'localhost'; // postgres database host
$config['db_database'] = 'filesender';  // name of database on postgres
$config['db_port'] = '5432'; // default port
$config['db_username'] = 'filesender'; // username to connect to postgres database
$config['db_password'] = '<secret>';  // password to connect to postgres database
```

Detailed information about the configuration settings of FileSender can be found in the Administrator reference manual

Step 11 - Configure the FileSender clean-up cron job

Schedule the FileSender clean-up cron job to run once a day:

```
echo '#!/bin/sh
php -q /usr/local/filesender/filesender/cron/cron.php' > /etc/cron.daily/filesender
```

Filesender uses a cron job to remove files that have expired, close any vouchers that have expired and remove any stale entries of cancelled uploads from the database. Typically you would run the cron job every night at a set time.

NOTE: the Debian and RPM packages will install the required cronjob for you.

NOTE: although the cronjob is responsible for the *actual removal* of expired or deleted files, any files deleted by a user through MyFiles will be no longer be available for download.

Step 12 - Start using FileSender

https://<your site>/filesender/

NOTE: If you want your site to be available on https://<your site>/, without the /filesender, point your **Apache DocumentRoot** to /usr/local/filesender/filesender/www and remember to update your **$config['site_url']** accordingly

Support and Feedback

See Support and Mailinglists and Feature requests .

Comments are disabled for this space. In order to enable comments, Messages tool must be added to project.
You can add Messages tool from Tools section on the Admin tab.