Linux Apache SSL PHP/FI frontpage mini−HOWTO

Table of Contents

Linux Apache SSL PHP/FI frontpage mini−HOWTO

Marcus Faure, marcus@faure.de

v1.1, July 1998

This document is about building a ***multipurpose webserver*** *that will support dynamic web content via the* ***PHP/FI*** *scripting language, secure transmission of data based on Netscape's* ***SSL****, secure execution of* ***CGI's*** *and M$* ***Frontpage Server Extensions***

1.Introduction

2.Component installation

3.Putting it all together

1.Introduction

Before you start reading: I am not a native speaker, so there are probably spelling/grammatical errors in this document. Feel encouraged to inform me of mistakes.

1.1 Description of the components

The webserver you hopefully will get after having read this howto is composed of several parts, the original apache sources with some (well, many) patches and some external executables. I recommend using the software versions I tried, they will probably compile without greater problems and result in a fairly stable daemon. If you are courageous, you can try to compile all the latest−stuff−with−tons−of−new−features, but don't blame me if something fails ;−). However, you may report other working configurations to be included in future versions of this document. All of the steps were tested on a linux 2.0.35 box, so the howto is somewhat linux−specific, but you should be able to use it for other unixes as well.

You do not necesserily have to compile in all components. I tried to structure this howto so that you can skip the parts you are not interested in.

The document is neither a user manual to Apache, SSL, PHP/FI nor frontpage. Its prime intention is to save webservice providers some headaches when installing their server and to do my little contribution to the linux community.

PHP is a scripting language that supports dynamic HTML pages. It is a bit like Apache's SSI, but by far more complex and has database modules for many popular dbs. The GD libraries are needed by PHP.

SSL is an implementation of Netscape's Secure Socket Layer that allow secure connections over insecure networks, e.g. to transmit credit card numbers to web based forms.

frontpage is a wysiwyg web authoring tool that makes use of some server−specific extensions called webbots. Some people think frontpage is cool because you can create feedback forms and discussion webs without having to know a bit about html or cgi. It even protects the designer from uploading his/her site via ftp by using a builtin publisher. If you wish to support frontpage but do not like to setup a windows server, the apache server extensions are your choice.

1.2 Working configurations

Though this document has been downloaded some 100 times since I published it, I received only little feedback. In particular, noone told me of other working combinations. Combinations that work for me are:

- Linux 2.0.31, Apache 1.2.4, PHP 2.0.0, SSL 0.8.0, fp 98 3.0.3 (*)
- Linux 2.0.33, Apache 1.2.5, PHP 2.0.1, SSL 0.8.0, fp 98 3.0.3 (*)
- Linux 2.0.35, Apache 1.2.6, PHP 3, SSL 0.8.0, fp 98 3.0.4

(*) version 3.0.3 is not recommended

1.3 History

v0.0/Apr 98: Preview version

v1.0/Jun 98: Now using Apache 1.2.6, updated fp section, minor corrections

v1.1/Jul 98: Sgmlized and restructered version

You can find the latest version of this document at http://www.faure.de

---

2.Component installation

2.1 Preparations

You will need:

- Apache 1.2.6 http://www.apache.org/dist/apache_1_2_6.tar.gz
- PHP/FI Extensions http://php.iquest.net/files/download.phtml?/files/php−2.01.tar.gz
- GD Library http://siva.cshl.org/gd/gd.html
- SSL 0.8.0 ftp://ftp.ox.ac.uk/pub/crypto/SSL/SSLeay−0.8.0.tar.gz
- SSL patch for Apache 1.2.6 ftp://ftp.ox.ac.uk/pub/crypto/SSL/apache_1.2.6+ssl_1.17.tar.gz
- frontpage 98 server extensions and install script http://www.rtr.com/fpsupport/download.htm

Get the sources you want. Untar apche, php, gd and ssl to `/usr/src`. Untar the SSL patch to `/usr/src/apache_1.2.6`.

2.2 Adding PHP

`cd` to /usr/src/gd1.2 and type make. This will build the GD library `libgd.a`, that should be copied to `/usr/lib`. Now `cd` to `php−2.0.1` and run `./install`.

The relevant questions are:

```
Would you like to compile PHP/FI as an Apache module? [yN] y
Are you compiling for an Apache 1.1 or later server? [Yn] y
Are you using Apache−Stronghold? [yN] y
Does your Apache server support ELF dynamic loading? [yN] y
Apache include directory (which has httpd.h)? [/usr/local/include/apache] /usr/src/apache_1.2.6/s
Would you like to build an ELF shared library? [yN] y
Additional directories to search for .h files []: /usr/src/gd1.2
Would you like the bundled regex library? [yN] n
```

Like the frontpage extensions, phtml includes a security problem because it is run under the uid of the webserver. Be sure to turn on safe mode in src/php.h and restrict the search path to a save value. There are some other options in php.h you may want to edit. If you are very concerned about security, compile php as a cgi. However, this will be a performance loss and not as smart as the module version.

Type `make` to build all files. When the compilation is done, copy `mod_php.*` and `libphp.a` to `/usr/src/apache_1.2.6/src` Add a line

```
Module php_module mod_php.o
```

to the end of `/usr/src/apache_1.2.6/src/Configuration`, add

```
−lphp −lm −lgdbm −lgd
```

to the `EXTRA_LIBS` in the same file,

```
application/x−httpd−php phtml
```

to Apache's `mime.types` and

```
AddType  application/x−httpd−php .phtml
```

to Apache's `srm.conf`.

You may also want to add `index.phtml` to `DirectoryIndex` in that file so that a file index.phtml is automatically loaded when its directory is requested.

2.3 Adding SSL

`cd /usr/src/SSL−0.8.0; ./Configure linux−elf; make; make rehash` This will create libraries needed by apache. You may issue `make test` to verify the compilation. You have to apply a patch to apache. It is important that you apply it before the frontpage patch, otherwise frontpage will not work. `cd` to `/usr/src/apache_1.2.6/src` and issue `patch < /usr/src/apache_1.2.6/SSLpatch`. Set `SSL_BASE=/usr/src/SSLeay−0.8.0` in `Configuration`. Make sure that `Module proxy_module` is disabled otherwise Apache won't compile. If you are in need of a proxy, go for Squid http://squid.nlanr.net/

Now `make certificate` to generate `SSLconf/conf/httpsd.pem`.

2.4 Adding frontpage

Rename the `fp30.linux.tar.Z` file to `fp30.linux.tar.gz`, otherwise the install script will not find it. Run `./fp_install` to copy the extension files to `/usr/local/frontpage`. zcat can usually be invoked as /usr/bin/zcat.

You now have to apply the FP patch. `cd` to `/usr/src/apache_1.2.6/src` and type `patch < /usr/src/frontpage/version3.0/apache−fp/fp−patch−apache_1.2.5` This will create the `mod_frontpage.*` files and do some modifications to `Configuration` etc. The 1.2.5 patch will work with both apache 1.2.5 and 1.2.6. Skip the part about installing webs, you can do that later

3.Putting it all together

3.1 Apache modules to try

The modules I use besides SSL, PHP and frontpage are:

```
Module env_module           mod_env.o
Module config_log_module    mod_log_config.o
Module mime_module          mod_mime.o
Module negotiation_module   mod_negotiation.o
Module dir_module           mod_dir.o
Module cgi_module           mod_cgi.o
Module asis_module          mod_asis.o
Module imap_module          mod_imap.o
Module action_module        mod_actions.o
Module alias_module         mod_alias.o
Module rewrite_module       mod_rewrite.o
Module access_module        mod_access.o
Module auth_module          mod_auth.o
Module anon_auth_module     mod_auth_anon.o
Module digest_module        mod_digest.o
Module expires_module       mod_expires.o
Module headers_module       mod_headers.o
Module browser_module       mod_browser.o
```

3.2 Giving CGI's more security

If you are an ISP (you probably are when you read this) you will want to improve security. The suexec utility allows you to do so; it will execute cgi's under the UID of the webowner instead of executing it under the webservers UID. Go to `/usr/src/apache_1.2.6/support` and `make suexec`. `chmod 4711 suxec` and copy it to the location specified in `../src/httpd.h` which is `/usr/local/etc/httpd/sbin/suexec` by default. If the path seems a little cryptic to you − it did to me − edit `httpd.h` and set the path to a more comfortable value.

3.3 Compiling and installing the server daemon

Enter `/usr/src/apache_1.2.6/src` and edit `Configuration` to set all the Modules you want to include in your Apache daemon. When done, run `./Configure` and `make`. This is the last (and most complicated) compilation step, so cross your fingers. If it succeeds, `cp httpsd` to `/usr/sbin`. The daemon is somewhat big, consider this when assembling your webserver. Create the directory `/var/httpd` with subdirectories `cgi−bin`, `conf`, `htdocs`, `icons`, `virt1`, `virt2` and `logs`. In `/usr/src/apache_1.2.6/conf` edit `access.conf−dist`, `mime.types` and `srm.conf−dist` to suit your needs and copy them to `var/httpd/conf/access.conf`, `srm.conf` and `mime.types`. Copy the `httpsd.pem` you created with `make certificate` to `/var/httpd/conf`. Use the following `httpd.conf`:

```
ServerType standalone
```

```
Port 80
Listen 80
Listen 443
User wwwrun
Group wwwrun
ServerAdmin webmaster@yourhost.com
ServerRoot /var/httpd
ErrorLog logs/error_log
TransferLog logs/access_log
PidFile logs/httpd.pid
ServerName www.yourhost.com
MinSpareServers 3
MaxSpareServers 20
StartServers 3

SSLCACertificatePath /var/httpd/conf
SSLCACertificateFile /var/httpd/conf/httpsd.pem
SSLCertificateFile /var/httpd/conf/httpsd.pem
SSLLogFile /var/httpd/logs/ssl.log

<VirtualHost www.virt1.com>
SSLDisable
ServerAdmin webmaster@virt1.com
DocumentRoot /var/httpd/virt1
ScriptAlias /cgi−bin/ /var/httpd/virt1/cgi−bin/
ServerName www.virt1.com
ErrorLog logs/virt1−error.log
TransferLog logs/virt1−access.log
User virt1admin
Group users
</VirtualHost>

<VirtualHost www.virt1.com:443>
ServerAdmin webmaster@virt1.com
DocumentRoot /var/httpd/virt1
ScriptAlias /cgi−bin/ /var/httpd/virt1/cgi−bin/
ServerName www.virt1.com
ErrorLog logs/virt1−ssl−error.log
TransferLog logs/virt1−ssl−access.log
User virt1admin
Group users
SSLCACertificatePath /var/httpd/conf
SSLCACertificateFile /var/httpd/conf/httpsd.pem
SSLCertificateFile /var/httpd/conf/httpsd.pem
SSLLogFile /var/httpd/logs/virt1−ssl.log
SSLVerifyClient 0
SSLFakeBasicAuth
</VirtualHost>

<VirtualHost www.virt2.com>
SSLDisable
ServerAdmin webmaster@virt2.com
DocumentRoot /var/httpd/virt2
ScriptAlias /cgi−bin/ /var/httpd/virt2/cgi−bin/
ServerName www.virt2.com
ErrorLog logs/virt2−error.log
TransferLog logs/virt2−access.log
</VirtualHost>
```

Depending on the modules compiled in, not all directives may be available. You can retrieve a list of available directives with `httpsd −h`.

3.4 Adding frontpage support to a web

Enter `/usr/local/frontpage/version3.0/bin` and load `./fpsrvadm`. Choose `install` and `apache−fp`. The next questions should be answered the following way:

```
Enter server config filename: /var/httpd/conf/httpd.conf
Enter host name for multi−hosting []: www.virt2.com
Starting install, port: www.virt2.com:80, web: ""
Enter user's name []: virt2admin
Enter user's password:
Confirm password:
Creating root web
Recalculate links for root web
Install completed.
```

The user name must be the unix login of the webowner. The password does not necessarily have to match the system password. You have to manually add `sendmailcommand:/usr/sbin/sendmail %r` to `/usr/local/frontpage/www.virt2.com:80.conf`, otherwise your users will not be able to send web−generated eMails. `kill −HUP` your `httpsd` to make fp reread its config. You can now access `www.virt2.com` with your frontpage client.

Under some circumstances `fpsrvadm` complaints that a root web has to be installed first. This is pretty useless, but you should do so to silence `fpsrvadm`.

3.5 Starting the daemon

Start Apache with `httpsd −f /var/httpd/conf/httpd.conf`. You can now access `www.virt1.com` both through http and https which is pretty cool. Of course you have to pay for a real certificate if you want to offer webwide SSL or users might laugh at you.

Copy one of the demo files from the php examples directory to `virt1` to test phtml.

3.6 Some considerations left

Do not use frontpage 97 extensions. They do not work, at least under Linux. When installing specific versions of the c++ libraries, they appear to work but your logs will soon fill with `premature end of script headers` and your mailbox will fill with complaints. Do not use frontpage 98 extensions before version 3.0.2.1330. Do not be confused, version numbers are somewhat inheterogenous. When telnetting to port 80, typing "get / http/1.0" and hitting return twice, you get a version number 3.0.4 for frontpage.

You can find out the more specific version number by executing `/usr/local/frontpage/currentversion/exes/_vti_bin/shtml.exe −version`. Older versions have a nasty bug that requires httpd.conf to be writable by the gid of the webserver. This should make you scream if you are at all concerned about security. Versions since 3.0.2.1330 are more usable.

3.7 Known bugs

When touching `Recalculate Links` in the frontpage client, the server starts a process that consumes 99% cpu cycles and some 10 mb of memory. But even for medium−sized webs and fast machines, the client sometimes recieves a timeout message, though the calculation will be finished correctly. Inform frontpage users to be patient and not to hit `Recalculate Links` several times. Inform yourself to equip the server with at least 64MB.

Please note that at the time of writing both SSL and frontpage work, but not at the same time, that means you can neither publish your web using ssl nor make use of the webbots through https. You can publish your web on port 80 and access it encrypted on port 443, but your counters etc. will be broken. I consider this a bug. This problem shall be fixed in SSL 0.9.0.

3.8 The final word

For those who think the title of this howto is nearly as long as the document: Did you ever listened to Meat Loaf?

O.K. readers, you're done for today. Feel free to send me your feedback, eternal gratitude, flowers, ecash, cars, oil sources etc.