Apache+DSO+mod_ssl+mod_perl+php+mod_auth_nds+mod_auth_mysql+mod_fastcgi

Table of Contents

Apache+DSO+mod_ssl+mod_perl+php+mod_auth_nds+mod_auth_mysql+mod_fastcgi

Ray Van Dolson, `rayvd@firetail.org`

v0.91, 5 April 2000

Details the installation of an Apache based webserver suite configured to handle DSO, and various useful modules including mod_perl, mod_ssl and php.

1.Legal Stuff

Apache+mods mini−HOWTO for Linux Systems

2.Introduction

This document outlines the process used to install Apache & modules onto the web−server at Walla Walla College (www.wwc.edu). While it will be, for the most part, system specific−−hopefully it can serve as a useful reference for performing other installations.

This document will attempt to outline the exact process used to install the server. Notes will be made when things should have been gone about differently, but the original steps will be given (assuming they worked).

2.1 Description of the Components

The platform on which the web−server was set up is a Red Hat 6.1 based system. Linux kernel 2.2.14 (compiled from scratch) running on a Dual PIII 600 based system with RAID5 and lots of other goodies.

The web−server software is Apache 1.3.12. The following modules were added to the server:

- mod_fastcgi SNAP (also mod_rewrite), for use with Zope.
- Auth−MySQL 2.20
- mod_ssl 2.6.2 (Open−SSL 0.9.5)
- mod_perl 1.21
- PHP 3.0.15
- mod_auth_nds 0.3a

2.2 History

v0.91 (April 5, 2000)

- Updated mod_fastcgi to correct version.

v0.9 (April 4, 2000)

- Completed first draft
- Spelling/Grammar errors

v0.1 (March 2000)

- Initial draft

3.Component Installation

3.1 Preparations

You will need the following software:

- Apache 1.3.12
- PHP 3.0.15
  - GD 1.3 (to make use of GIF files)
    - Source
    - RPM
    - RPM−devel
  - GD 1.8.1 (to make use of PNG files)
    - Source
    - RPM
    - RPM−devel
  - IMAP 4.5+
    - Source
    - RPM
  - OpenLDAP 1.2.9+
    - Source
    - RPM
    - RPM−devel
- mod_perl 1.22+
  - *Perl5 Modules Required*
  - MIME::Base64
  - URI
  - HTML−Parser
  - Digest−MD5
  - libnet
  - libwww
- mod_ssl 2.6.2+
  - OpenSSL 0.9.5

Apache+DSO+mod_ssl+mod_perl+php+mod_auth_nds+mod_auth_mysql+mod_fastcgi

- ◊ Source
- ◊ RPM
- ◊ RPM−devel

  - ♦ RSAREF 2.0
  - ♦ MM 1.0.12
- MySQL 3.22.32
- mod_auth_nds 0.4
  - ♦ ncpfs 2.2.0.17 Note: the kernel must also be compiled with IPX support.
- mod_auth_mysql 2.20
- mod_fastcgi SNAP Oct06

This is the directory layout scheme I use and recommend:

```
+ /usr/src
|
+-+ apache
  |
  +-+ apache-1.3.12
  |
  +-+ modules
  | |
  | +-+ mod_perl
  | | |
  | | +- mod_perl-1.21
  | | |
  | | +-+ depend
  | |   |
  | |   +- <perl modules here>
  | |
  | +-+ mod_ssl
  | | |
  | | +- mod_ssl-2.6.2-1.3.12
  | | |
  | | +-+ depend
  | |   |
  | |   +- openssl-0.9.5
  | |   |
  | |   +- rsaref-2.0
  | |   |
  | |   +- mm-1.0.12
  | |
  | +-+ mod_fastcgi_SNAP
  | |
  | +-+ php
  | | |
  | | +- php-3.0.15
  | | |
  | | +-+ depend
  | |   |
  | |   +- gd-1.3
  | |   |
  | |   +- imap-4.5
  | |   |
  | |   +- openldap-1.2.9
  | |
  | +-+ mod_auth_nds
  | | |
  | | +- mod_auth_nds-0.4
```

```
| | |
| | +-+ depend
| |   |
| |   +- ncpfs-2.2.0.17
| |
| +-+ mod_auth_mysql
|
+-+ mysql
```

Check to see if some of the above modules/software packages are already installed on your system. It usually doesn't hurt, however, to download everything and install it just in case. You might have an older version installed on your system.

3.2 mod_ssl

Installing and Compiling OpenSSL

mod_ssl requires some sort of SSL engine be installed. OpenSSL is the natural choice for the Linux environment. You can either install it via RPM (as I did), or compile it from source. Since I did not compile it from source, you're on your own, although I would think it pretty straightforward. Most likely you'll either install it on the system (into /usr/local or something) or leave it in its directory and simply point whichever applications need OpenSSL to its directory.

RPM will install OpenSSL into system directories.

Installing and Compiling RSAREF 2.0

Create the rsaref−2.0 directory wherever you like. For me this is in /usr/src/apache/modules/mod_ssl−blah/depend/. Change to this directory.

```
cd rsaref-2.0
cp -rp install/unix local
cd local
make
mv rsaref librsaref.a
```

These commands should build you the rsaref library! Just leave the stuff here and when you need to link against it, just point the appropriate configure script to this location.

Installing and Compiling MM

Extract mm−1.0.12 (or whichever version is most current) to the depend directory of the mod_ssl−blah subdirectory. Perform the following steps:

```
cd mm-1.0.12
./configure --disable-shared
make
```

This should build your mm libraries. As above, reference this path when needed. You're on your own if you want to install this library to the system.

Installing and Compiling mod_ssl (at last!)

The normal procedure with apxs is to compile Apache first, and then, using apxs, compile the modules you want to use and insert them into the server. However, mod_ssl needs to be compiled into the server the normal way before you can use it via apxs. Once mod_ssl is in the server for the first time, you can then upgrade it via apxs without having to completely recompile Apache.

Enter the directory where you are compiling mod_ssl and execute the following configuration directive (here is the file I use) for the initial compile:

```
#!/bin/sh

./configure \
        --with-apache=/usr/src/apache/apache_1.3.12 \
        --with-ssl \
        --with-rsa=../depend/rsaref-2.0/local \
        --with-mm=../depend/mm-1.0.12 \
        --enable-shared=ssl
```

You don't need to run 'make' or anything here. When we compile Apache, it will do it all for us.

This configuration line gives two examples of how your system could be set up. In my case, OpenSSL was already installed somewhere in the system (probably in /usr/lib, /usr/include). Therefore, I didn't need to pass it any location parameters. However, rsa and mm were −not− on the system, and I compiled them myself and left them within their source trees (didn't run make install, et al). In that case, you need to point configure to the appropriate directory so it can find the headers/libraries.

From this point on, unless you upgrade Apache (in which case you'd need to perform the above step again for the new version of Apache), you can use apxs to upgrade and recompile mod_ssl. Here is the configure script I use for this:

```
./configure \
        --with-apxs=/apps/apache-1.3.12/bin/apxs \
        --with-ssl=../depend/openssl-0.9.4 \
        --with-rsa=../depend/rsaref-2.0/local \
        --with-mm=../depend/mm-1.0.12
```

Or some combination of the above. Then run:

```
make
make install
make distclean
```

to complete the installation.

Notes: *MM is −not− required to compile mod_ssl. If you're having problems getting it to work, simply omit compiling it and also from the ./configure line(s).*

When I compiled mod_ssl, I had errors regarding DBM. To fix this, I had to add −lndbm to the Makefile:

- Run the above configure script.
- cd to pkg.sslmod
- Edit the makefile and add −lndbm to LIBS_SHLIB. It should look like:
- `LIBS_SHLIB=-lm -lcrypt -lndbdm`

Hopefully that will save you some grief.

3.3 Apache

Extract apache−1.3.12.tar.gz to /usr/src/apache or wherever. Next we want to compile Apache enabling the following options:

- mod_ssl (In order to compile mod_ssl as a DSO, it has to be first compiled into the server normally. After doing this, the module can then be upgraded via apxs.)
- mod_proxy
- mod_so
- mod_rewrite (For use with Zope)

Here is the configuration file I used to initially compile Apache:

```
#!/bin/sh

SSL_BASE=../depend/openssl-0.9.4 \
RSA_BASE=../depend/rsaref-2.0/local \
EAPI_MM=../depend/mm-1.0.12 \
./configure \
        --enable-module=ssl \
        --enable-module=proxy\
        --enable-shared=proxy\
        --enable-module=rewrite \
        --enable-shared=rewrite \
        --prefix=/apps/apache-1.3.12 \
        --enable-shared=ssl \
        --enable-rule=SHARED_CORE \
        --enable-rule=SHARED_CHAIN \
        --enable-module=so
```

Then run

```
make
make certificate
make install
```

Apache should now be compiled and installed into whicever directory you specified with −−prefix. Test it out and make sure it starts up.

```
/path/to/apache/bin/apachectl start
```

Apache+DSO+mod_ssl+mod_perl+php+mod_auth_nds+mod_auth_mysql+mod_fastcgi

or

```
/path/to/apache/bin/apachectl startssl
```

Hopefully it all runs smoothly. If not, trace back over your steps and ensure you didn't forget anything.

3.4 MySQL

php as well as mod_auth_mysql and possibly mod_perl will require that MySQL be installed and running on your system. It is beyond the scope of this document to go into the details of installing MySQL, but download the archive and follow the directions in the INSTALL file(s). It is a fairly straight−forward procedure to get MySQL up and running. Something like:

```
./configure
make
make install
```

Should get everything installed so that you can compile the other Apache modules.

3.5 PHP 3.0.15

We will compile php−3.0.15 as a DSO which means that it is a separate module that can be loaded and unloaded from the server. This makes it easy to upgrade php without having to recompile everything (which can be a pain if you use a lot of modules with Apache).

GD

In our installation of Apache, php uses gd to create images, and such. I used an older version of gd (installed via RPM) to link php against. This way we can use output GIF files. This probably isn't too desirable do to copyright issues, and thus you may wish to use a version later than 1.3 which only supports PNG files.

Either install via RPM (rpm −i gd*.rpm) or compile from source and install to the system.

IMAP

If you want IMAP support, the procedure is similar to that of gd. I used the RPM since I'm on a Red Hat system, but installing from source should be a relatively simple procedure of ./configure;make;make install.

OpenLDAP

Once again you can install OpenLDAP either via RPM or source. I chose to do it via source since the latest version was not yet available via RPM at the time we were setting things up.

```
./configure
make
make install
```

should do the trick! (Or rpm –i openldap*.rpm)

Installing and Compiling PHP 3.0.15

Once the above items are installed and working, we can go ahead and compile PHP as a DSO. The process is very straightforward and simple.

```
cd /usr/src/apache/modules/php/php-3.0.15

./configure \
        --with-apxs=/apps/apache/bin/apxs \
        --with-config-file-path=/apps/etc \
        --with-gd \
        --with-imap \
        --with-mysql=/apps/mysql \
        --with-ldap=/apps \
        --with-zlib \
        --enable-track-vars
```

Make sure that if any of your −−with libraries are not installed in /usr/local or /usr, that you tack on an =/location/ line so that configure can find the stuff it needs!

```
make
make install
```

If everything completes properly, 'make install' will use apxs to install libphp3.so to /apache/libexec/libphp3.so and add the proper entries into httpd.conf and activate php3. Pretty slick.

3.6 mod_perl

This section documents the installation of mod_perl as a DSO for Apache. There are a number of perl modules (in addition, of course, to perl5, which I will assume you already have installed) that must be added before mod_perl will compile without complaining. If you don't install these modules, mod_perl should complain and tell you which ones you are missing.

There is a certain order in which the modules must be installed. Some depend on others and thus I've listed the install order that I used without any problems.

Required Perl Modules

The perl modules can be obtained from locations detailed further up in this document. Download them and put them wherever you like or in the location I used as depicted in the directory map (also above). Installing a module is fairly simple. After extracting the module to a directory (usually with tar xvfz), you simply change to that directory and execute the following commands:

```
perl Makefile.PL
make
make install
```

If everything goes as it should, this will configure, build and install the perl module for you. Of course, check the README for each module if things don't work quite as expected.

Here is the order I used to install the modules necessary for mod_perl:

1. MIME::Base64
2. URI
3. HTML::Parser
4. Digest−MD5
5. libnet
6. libwww

Installing and Compiling mod_perl 1.2x

After installing the perl modules, we're ready to compile and install mod_perl into Apache. Change to the directory where you extracted mod_perl to, and run the following script:

```
perl Makefile.PL \
        USE_APXS=1 \
        WITH_APXS=/path/to/apache/bin/apxs \
        EVERYTHING=1
```

This will set up your Makefile and tell mod_perl to compile itself as a DSO using apxs (the location of which you must specify). After this step, simply run

```
make
make install
```

And mod_perl will be moved to the appropriate directory and lines added to your httpd.conf file.

3.7 mod_auth_mysql

mod_auth_mysql lets the Apache web−server authorize against a MySQL user database. Installation of the module as a DSO isn't exactly documented in the README file, but it can be done.

First, change to the directory you extracted mod_auth_mysql to. I assume that you have MySQL installed somewhere (along with the headers, etc). Make sure you know the location of the MySQL libraries and header files. If in doubt, check /usr/lib/mysql and /usr/include/mysql.

In order to compile mod_auth_mysql, we'll first have to rename the 'config.h' file to 'auth_mysql_config.h'. I'm not sure why this file wasn't named correctly, but simply execute the following command:

```
cp config.h auth_mysql_config.h
```

Now for the final step:

```
/path/to/apache/bin/apxs -i -a -I/usr/include/mysql -L/usr/lib/mysql \
        -lmysqlclient -c mod_auth_mysql.c
```

You may need to run as root if you do not have read/write access to the Apache directory.

3.8 mod_auth_nds

At my school, the Windows network of choice is Netware. It's been in place for a long time, and although hopefully someday it will be retired, for now it is still the main network on campus for filesharing and email. Every student has a Netware account on which their personal files−−including their webpages are stored. We mount these directories on our linux server and it's nice to be able to password protect certain ones with the Netware username and password information. With this module, Apache can authenticate straight to the Netware server itself.

ncpfs

In order to compile mod_auth_nds, we need to have ncpfs installed (along with its headers of course). Before compiling ncpfs, you must ensure that your kernel has IPX support compiled in. If this is the case, simply running

```
./configure
make
make install (optional)
```

will compile (and install) the libraries.

Compiling and Installing mod_auth_nds

With ncpfs installed, running the following command should compile mod_auth_nds as a DSO:

```
/path/to/apache/bin/apxs -c -lncp -L/usr/lib -I/usr/include mod_auth_nds.c
/path/to/apache/bin/apxs -i mod_auth_nds.so
```

Then add the following lines to your httpd.conf (by hand):

```
LoadModule nds_auth_module libexec/mod_auth_nds.so
AddModule mod_auth_nds.c
```

Then, restart Apache!

3.9 mod_fastcgi

Installing mod_fastcgi is necessary if you want to allow access to your Zope server through Apache. This might be useful simply because Apache is inherently more secure and much more configurable than the Zope server itself.

The current stable version of mod_fastcgi is 2.2.2, however, this version does not work properly with Zope.

You must get the SNAP release which is dated Oct 06. The link is provided above.

Change to the mod_fastcgi directory and run the following commands:

```
/path/to/apache/bin/apxs -o mod_fastcgi.so -c *.c
/path/to/apache/bin/apxs -i -a -n fastcgi mod_fastcgi.so
```

See the mod_fastcgi documentation for a description of its use.

4.Final Words

Much of this information can be obtained by reading the README and INSTALL files included with the various modules. However, this document is useful in the cases which didn't work as expected for me, or else for which the installation procedure was not as well defined as I would have liked. It also has the added benefit of being one, sequential document, which should hopefully be easier to follow and understand than a slew of README files.

4.1 Credits

```
Phillip R. Wilson <pwilson@drew.edu>, author of mod_auth_nds, for helping me get
 mod_auth_nds to compile and install with apxs.

John Ash <jash@wwc.edu>, my boss, for all sorts of help and of course, a job.

Marcus Faure <marcus@faure.de>, author of the Apache SSL PHP/FI frontpage mini-
 HOWTO, whose document this one is loosely based on.
```

4.2 Contact Information

If you find any blatant errors in this document, spelling, grammatical, content or otherwise, please don't hesitate to drop me an email. You can get ahold of me via a number of means.

```
Ray Van Dolson
Email: <url url="mailto:rayvd@firetail.org" name="rayvd@firetail.org">
IRC: DALnet, #Bludgeon (nick Variant)
```

4.3 Anything Else

Everything mentioned in this document will eventually be available for ftp from ftp.wwc.edu/pub/apache. I will have everything laid out as described above, and hopefully installation scripts to install everything from scratch. (A very dumb script mind you).