Installing Apache on BSD

Remco Hobo

October 22, 2004

1 Some facts

With apache one can compile a vast array of modules. At the moment, there are a total of 356 modules available at modules.apache.org. These modules are patched against the source code before it is compiled. A number of modules are included in the default package, others have such a specific feature most users do not need it. Also, some modules might not work together with others.

A lot of arguments can be added to the .configure script. These arguments might instruct apache to install the files to a specific directory, tell it to include modules (e.g. with-mysel), specify the operating system, and change the execution of the configure script itself.

The enviroment variables can also be set as an argument of the .configure script. Paths to libraries can be added, the processor type can be set and things like multi-threading can be set. Normally these variables don't have to be changed. Most variables for compiling have been set in the /etc/make.conf file.

Packages are optional components that can be added to apache after compile time. These can be defined in the httpd.conf.

The main task of the .configure script is to find out what kind of environment the program is to be compiled for. What kind of operating system is in place, what kind of processor, what kind of compiler enviroment, etc.

After a .configure script has been executed, a Makefile will be generated. When this file is present a make kan be executed. The make command has ben created to make compiling a program less of a pain. Before it came to be, a lot of complicated compile options had to be executed to compile a program. The make file will go through the source recursively and will compile the source according to the Makefile file.

2 Downloading and compiling

1. First, download the sourcefile from www.apache.org and untar it.
2. Check the MD5 signature: md5sum httpd-%version%.tar.gz && cat httpd-%version%.tar.gz.md5
3. Check the PGP signature: gpg –verify httpd-%version%.tar.gz.asc
4. Configure the source tree: ./configure –prefix=/www –enable-so / –enable-header=shared –enable-rewrite=shared –enable-ssl=shared

- –prefix=/www points to the webroot.
- –enable-so will enable a static module, which can load dynamic modules.
- –enable-header=shared will load a dynamic module to control the HTTP headers
- –enable-rewrite=shared will load a dynamic module to rewrite URL‘s, a kind of alias (This is a security weakness)
- –enable-ssl=shared will load the ssl module, this is to make https possible, secure http.

3 Generating certificates

- Create a RSA private key for your Apache server (will be Triple-DES encrypted and PEM formatted): openssl genrsa -des3 -out server.key 1024
- Create a Certificate Signing Request (CSR) with the server RSA private key (output will be PEM formatted): openssl req -new -key server.key -out server.csr
- Create a RSA private key for your CA (will be Triple-DES encrypted and PEM formatted): openssl genrsa -des3 -out ca.key 1024
- Create a self-signed CA Certificate (X509 structure) with the RSA key of the CA (output will be PEM formatted): openssl req -new -x509 -days 365 -key ca.key -out ca.crt
- Prepare a script for signing which is needed because the “openssl ca” command has some strange requirements and the default OpenSSL config doesn’t allow one easily to use “openssl ca” directly. So a script named sign.sh is distributed with the mod_ssl distribution (subdir pkg.contrib/). Use this script for signing. Now you can use this CA to sign server CSR’s in order to create real SSL Certificates for use inside an Apache webserver (assuming you already have a server.csr at hand): ./sign.sh server.csr
- Edit the server.crt file and edit the thest two lines: SSLCertificateFile /path/to/this/server.crt SSLCertificateKeyFile /path/to/this/server.key
- Now run apachectl startssl
- The https server has been started with the certificates.

4 virtual hosts

Virtual hosts make it possible to make aliases for domains. For instance, if a server houses a lot of sites, this may be very handy. You can even rewrite whole domains: http://colocator.com/testShop is more difficult to remember then http://testShop.nl. When virtual hosts are configured correctly, apache will know which sites is being requested, and the proper source files will be read.

A virtual host entry can look like this:

```
NameVirtualHost 145.92.27.221

<VirtualHost 145.92.27.221:80>
 ServerName testShop.nl
 ServerAlias testShop.nl
 DocumentRoot /www/testShop
 ServerAdmin remco.os3.nl
</VirtualHost>
```