4.0 Setup 0FS Apache

Apache HTTP Server is an open-source web server platform. This article will outline the steps to install, configure, harden a zero-footprint instance of Apache 2.2 & 2.4, with particular focus on the nuances between each.

Prerequisites

If you are building you zero-footprint for the first time you will need a C/C++ compiler available on the initial system. Once compiled, the resulting package is portable to other like-O/S servers. For the most part, most Unix/Linux distributions will come packaged with the gcc compiler.
Unix/Solaris

Check if gcc compiler is installed:

```
$ which gcc

# dependent on environment
variables being set correctly.
# Alternatively check the
/usr/bin and /usr/sfw/bin paths.
```

If no compiler found, install it:

```
$ pkg install gcc-3  # or
whatever version you need
```

Linux

Chech if gcc compilete is installed:

```
$ which gcc
```

If no compiler found, install it:

```
# Debian/Ubuntu
$ sudo apt-get install
build-essential

# RHEL/CentOS/Fedora
$ sudo yum group install
"Development Tools"
```

Initial Installation

1) Get Source Files

The first step is to retrieve the source files from Apache. Grab the compressed files pertinent to the O/S you are using, typically bzip2 for Unix and gunzip for Linux:

```
# Change dir to whichever working directory you want to use
$ cd /opt


# Change version number/archive type as required  - current version is
2.4.9
$ wget --no-check-certificate
https://archive.apache.org/dist/httpd/httpd-2.4.9.tar.bz2 [ -e
use-proxy=yes -e https_proxy=xxxxx ]


# Apache also provides MD5 hashes to verify your downloads, so you could do
the following to generate a local MD5 hash to compare
wget -O - https://archive.apache.org/dist/httpd-2.4.9.tar.bz2 | tee
httpd-2.4.9.tar.bz2 | md5sum > md5sum.local
```

Unpack the archive:

Unix/Solaris

```
# Use -k switch to preserve the
original archive
$ bzip2 -d[k]
httpd-2.4.9.tar.bz2
$ tar -xvf httpd-2.4.9.tar -C
/opt/httpd/
```

Linux

```
$ tar -xzvf httpd-2.4.9.tar.gz
-C /opt/httpd/
```

2) Compile Apache

Next, we will compile Apache. Different versions require different steps, so choose your version below:

Apache 2.2 and earlier Setup:

```
$ cd /opt/httpd

# First we configure the build using the following syntax
# ./configure --prefix=/opt/apache2 --enable-mods-shared=few
[--enable-{modname}] [--disable-{modname}] [with-apr=included]
[with-pcre=/opt/pcre]

# Here is the most common configuration
./configure --prefix=/opt/apache2 --enable-mods-shared=few
--enable-rewrite --enable-headers --enable-ssl --disable-userdir
--disable-autoindex --disable-status --disable-env --disable-setenvif
--disable-cgi --disable-actions --disable-negotiation --disable-alias
--disable-include --disable-filter --disable-version --disable-asis
--with-apr=included --with-pcre=/opt/pcre

$make
$make install
```

Apache 2.4 Setup:
Since Apache 2.4, the Apache Portable Runtime and the Perl Compatible Regex modules are no longer packaged with the original source. However, these modules are mandatory for Apache to compile and run.

Click here to find out why you need these libraries...

APR

The APR library provides a set of APIs that map to the underlying O/S and emulate functions if they are not available, making Apache platform-agnostic.

PCRE

The PCRE library provides more powerful and flexible regex expression functionality than other flavours and is used by mod_rewrite, etc.

Apache provides the flexibility to point to existing instances of these when compiling. If you do not have these modules you can add them as follows:

First, download the module source files:

```
$ wget http://archive.apache.org/dist/apr/apr-1.6.3.tar.bz2
$ wget http://archive.apache.org/dist/apr/apr-util-1.6.1.tar.bz2

# Apache 2 requires pcre, not pcre2
$ wget --no-check-certificate
https://ftp.pcre.org/pub/pcre/pcre-8.41.tar.bz2
```

Extract the source files:

```
# APR and APR utils can be compiled with Apache out of the box
provided they are in the srclib directory.   # NOTE, the contents of
the untarred folders must be copied to a folder under srclib with the
exact names   # below:
$ tar -x[z]vf apr-1.6.3.tar[.gz] --directory
/opt/httpd-2.4.x/srclib/apr
$ tar -x[z]vf apr-util-1.6.1.tar[.gz] -- directory
/opt/httpd-2.4.x/srclib/apr-util


# PCRE will not be automatically compiled in the srclib directory, so
either manipulate the build script or simply keep it separate.
$ tar -x[z]vf pcre-8.41.tar[.gz]
```

If you've placed PCRE in its own folder, you will have to build it first:

```
$ ./configure --prefix=/opt/pcre --enable-pcre16 --enable-pcre32
$ make
$ make install
```

Apache 2.4 requires the use of specific options for APR and APR utils to install. Here is a standard configuration for Apache 2.4:

```
$ cd /opt/httpd

# First we configure the build using the following syntax
# ./configure --prefix=/opt/apache2 --enable-mods-shared=few
[--enable-{modname}] [--disable-{modname}] [with-apr=included]
[with-pcre=/opt/pcre]

# Here is the most common configuration
$./configure --prefix=/opt/apache2 --enable-mods-shared=few
--enable-rewrite --enable-headers --enable-ssl --disable-userdir
--disable-autoindex --disable-status --disable-env --disable-setenvif
--disable-cgi --disable-actions --disable-negotiation --disable-alias
--disable-include --disable-filter --disable-version --disable-asis
--with-included-apr --with-included-apr-util --with-pcre=/opt/pcre


$make
$make install
```

Here it is important to understand what each switch is doing and the implications of each.

Configure Command Switch	What does it do?

--prefix	Sets the output directory for the build i.e. where Apache will reside. This direcory specification will have a direct impact on portability of the 0FS package. Read more in the Portability section.
--enable-mods-shared=value or --enable-mods-shared={module_names} (space-delimited)	Sets which modules will be compiled as DSOs (shared libraries). Options are "all" | "most" and in 2.4 and higher also "few" | "none" | "reallyall".
--enable-{module_name} or -- enable-modules={module_names} (space-delimited)	Enables the module for the build. Shared or static inclusion is determined by the underlying APR as will as the --enable-mods-shared directive. For example, with Apache 2.4, the standard APR supports DSOs, so it would compile the module as shared, unless the --enable-mods-shared is set to "none", which will force it to be compiled as static.
--disable-{module_name}	Disables the module for the build. The module will not be compiled at all, so you will not even be able to add it dynamically later through Apache configuration without either recompiling Apache in full or compiling the module itself and copying it into the modules directory of the Apahce install
--with-{module_name}=path|included	Used to specify specific path to find compiled modules if not using the defaults included with source. The included value will force the build to use the one included with Apache source.

Click here to see a description of each module

Here

Module	Min. Apache V2 Version	Included				What does it do?	Reasons to include/exclude
		Default	Most	Reallyall	Few		
mod_access_compat	2.4	Yes			YES	Control access based on client hostname, IP address or other characteristics of client request	
mod_actions	2.0	No				Lets you run CGI scripts when a particular file or method is used in a request	Exclude if not using CGI scripts or have no need to execute scripts conditionally based on requests. XSS vulnerability considerations. If included, ensure request parameters are not considered when making decisions based on content type
mod_alias	2.0					Used for simple URL manipulation tasks, including mapping URLs to filesystem paths and standard redirection.	
mod_allowmethods	2.4					Restricts what HTTP methods can be used on a server	
mod_asis	2.0					Allows you to send a document without adding the usual HTTP headers	
mod_auth_basic	2.2					Used to restrict access with HTTP Basic Auth. Should be combined with at least one authentication module and one authorization module.	If this type of authentication is required, it is nearly imperative to use SSL as passwords are sent as almost plain text (base4 encoded).
mod_auth_digest	2.0					Used to implement HTTP Digest Auth.	If this type of authentication is required, it is nearly imperative to use SSL as an attacker can force the browser to downgrade to basic auth. The passwords are stored unsecurely on the server.
mod_auth_form	2.4					Allows the use of an HTML login form to restrict access	Depends on mod_session modules and makes use of HTTP cookies, which is susceptible to XSS attacks.
mod_authn_anon	2.2					Authentication - Provides anonymous user access to authenticated areas	
mod_authn_core	2.4					Authentication - Provides core authentication capabilities	
mod_authn_dbd	2.2					Authentication - Provides authentication against SQL tables	
mod_authn_dbm	2.2					Authentication - Provides authentication against dbm password files	

mod_authn_file	2.2					Authentication - Provides authentication against plain text password files	
mod_authn_socache	2.4					Authentication - Maintains shared object cache of authentication credentials	
mod_authnz_fcgi	2.4[.10]					Authorization - FastCGI authorizer application	
mod_authnz_ldap	2.2					Authorization - Provides authorization through an LDAP directory	
mod_authz_core	2.4					Authorization - Provides core authorization capabilities	
mod_authz_dbd	2.4					Authorization - Provides group authorization based on SQL database	
mod_authz_dbm	2.2					Authorization - Provides group authorization based on dbm files	
mod_authz_groupfile	2.2					Authorization - Provides authorization against plain text files	
mod_authz_host	2.4[.19]					Authorization - Provides authorization based host (name or IP)	
mod_authz_owner	2.2					Authorization - Provides authorization based on file ownership	
mod_authz_user	2.2					Authorization - Provides authorization based on authenticated user	
mod_autoindex	2.0					Generates directory indexes	Exclude in most cases. Be sure to disable index generation in Apache configuration as shown in Hardering section below.
mod_brotli	2.4[.26]					Compresses content using Brotli before its delivered to the client	
mod_buffer	2.4					Support for request buffering	Exclude in most cases. Reads the request into RAM and then repacks into fewest memory buckets possible. However, at the cost of CPU time. If request/response is already efficiently packed, this could have adverse affects on processing time.
mod_cache	2.0					HTTP caching filter	If included be aware that CacheQuickHandler is on by default which circumvents Allow and Deny directives.
mod_cache_disk	2.4					Disk based storage for mod_cache	
mod_cache_socache	2.4					Implements a shared object cache storage for mod_cache	
mod_cern_meta	2.0					Emulate CERN HTTPD Meta file semantics	
mod_cgi		Yes				Allows execution of cgi scripts	Exclude if not required. Considerations for exploits including ShellShock, etc. If invoking bash scripts, ensure bash version is > 4.3
mod_cgid	2.0					Allows execution of cgi scripts (used for certain Unix multi-threaded environments only)	Ibid.
mod_charset_lite	2.0					Allows the server to change the character set of responses before sending them to the client i.e. if files are stored as EBCDIC, it can be translated to ISO	
mod_data	2.4					Converts response body into an RFC2397 data URL	Exclude if not required. XSS attacks have been reported in applications leveraging mod_data such as Moodle, etc.
mod_dav	2.0					Enables creating, moving, copying, and deleting of resources and collections on a remote web server	**This should be excluded unless absolutely necessary. DLL Hijack exploits, etc. are widely known/reported.** If including, ensure the server is secure before enabling with some type of authentication.
mod_dav_fs	2.0					Filesystem provider for mod_dav. Prerequisite is mod_dav.	Ibid.

mod_dav_lock	2.2					Generic locking API used by backend provider for mod_dav. Prerequisite is mod_dav and backend provider such as mod_dav_svn	Ibid.
mod_dbd	2.2					Enables APR to manage db connections	Exclude if not required. Considerations for SQL injection attacks especially when using third-party modules in conjunction.