Securing Apache

Understanding and securing your Apache web server configuration

Allaire Security White Papers Series

(Version 1.0)

Abstract

Title	Securing Apache
Date	January 8, 2001
Product	Apache HTTP Server
Target Audience	Web Server Administrators
Abstract	Securing a web server can be a difficult process, considering the vast number of security advisories an administrator must keep track of. This document and other lockdown documents are Allaire's effort toward making this job a little easier.

Allaire Corporation • One Riverside Center • 275 Grove Street • Newton • MA • 02466
www.allaire.com • info@allaire.com • (617) 219-2000 •
security issues: secure@allaire.com
document feedback: lockdown@neohapsis.com

Table of Contents

Apache configuration overview

This document is intended for the system administrator who has been tasked with securing his or her implementation of the Apache web server. This document assumes that the server has been installed and is working properly, and the admin is capable of basic configuration. This document details the security considerations and options surrounding Apache's operation.

As of May 2000, the most recent version of Apache is 1.3.12. If you are not running v1.3.12, we suggest you upgrade because the latest version fixes an assortment of bugs and security issues. This document assumes you are using a version of Apache in the 1.3.0 series. Users of the 1.2.x series should make upgrading a priority since there are security concerns in those versions as well.

Location of configuration files

Depending on the configuration and server version, most of the options detailed in this document are found in srm.conf, httpd.conf and access.conf. The default source distribution uses /usr/local/apache/conf as the location of these files; however, this may be different depending on your distribution. For example, RedHat Linux uses /etc/httpd/conf. If you are unsure of the location of your configuration files, you can locate them by running the Unix 'find' command:

find / -name httpd.conf

Configuration checks

There are a number of configuration settings you should be aware of within the Apache web server. The following is set of initial checks you should perform when securing your Apache installation:

- The 'User' or 'Group' directive should **not** be a normal user. Typically the 'User' and 'Group' directives specify 'nobody'. Regardless of who the user is, it is important to make sure that this user only has read permission on web documents and execute permissions on CGI documents *in web directories*.

- Make note of your 'DocumentRoot' setting. Make sure this is set to an acceptable directory. For example, if 'DocumentRoot' is set to "/etc," people will be able to request files from your /etc directory.

- Logs. We do not recommend that you put logfiles in the web server root directory, especially in <server root>/logs. *Doing so may make information available to potential attackers*. If you must allow for access to your logfiles remotely via the web, use .htaccess authentication (described below) to protect them.

- 'AccessFileName' typically defaults to '.htaccess'. This is the value we recommend.

- Take care when configuring your 'Alias' and 'ScriptAlias' directives. Alias will allow access to other directories, and ScriptAlias will give CGI/execute permissions to files in that directory. Two typical examples:

  ```
  Alias /icons/ /home/httpd/icons/
  ScriptAlias /cgi-bin/ /home/httpd/cgi-bin/
  ```

  This lets a user request files out of /home/httpd/icons, as well as /home/httpd/cgi-bin. Review all aliases and disable any that aren't used or not required.

  ***Note**: the /icons/ alias is used for graphical representation of directory indexes. You won't disrupt anything by disabling it.*

  ***Note:** If you do not use /cgi-bin, we suggest removing the associated 'ScriptAlias' directive.*

- Handlers - Remove any handlers that aren't used. Handlers tell the server to process a file in a special way. For example:

  ```
  AddHandler cgi-script .cgi

  AddType text/html .shtml
  AddHandler server-parsed .shtml

  AddHandler send-as-is asis
  AddHandler imap-file map
  ```

  Disable server-side-includes (.shtml) if they're not used. Map and 'asis' should be disabled, and you should use ScriptAlias rather than .cgi handler for CGI scripts. ScriptAlias limits the CGI files to the one directory, where adding a handler for .cgi will let CGI files run anywhere in the web directory. **Do not** enable the server-status handler.

- Directory options - If you do not use symbolic links in your web directories, we suggest you remove the 'FollowSymLinks' option from the default document directory. However, note that 'FollowSymLinks' does not always exist by

default. Also, by removing the 'Indexes' option you can keep the server from giving directory listings or "directory indexes." A typical example is:

```
<Directory /home/httpd/html>
Options None
AllowOverride None
order allow,deny
allow from all
</Directory>
```

- Some Apache configurations ship with a set of default cgi-bin scripts. We suggest you delete *all* of them, unless specifically required, since many of the included scripts contain known security holes.

- The 'DirectoryIndex' directive includes a list of files to return if a user requests a directory (such as "http://your.server.com/"). Take note of the list's order. For instance, you may have:

  ```
  DirectoryIndex index.cgi index.shtml index.html
  ```

  Even though an index.html may exist, there's the possibility that an attacker can write an index.cgi file, which will be called instead of (or before) index.html (or index.shtml)

- Disallow web requests to your .htaccess files by adding the following configuration directives in your httpd.conf:

  ```
  <Files ~ "\.htaccess$">
          order deny,allow
          deny from all
  </Files>
  ```

- Remove the documentation alias (shipped with certain configurations). Access to package documentation allows attackers to remotely determine what software is installed. They may even be able to determine what version the software is. Comment out or delete the following lines from access.conf:

  ```
  Alias /doc /usr/doc
  <Directory /usr/doc>
          … various configuration directives …
  </Directory>
  ```

Loadable module configuration

Apache 1.3.0 introduced the concept of loading and unloading the features you want via external “shared modules.” Modules allow for greater control over specific features within the Apache web server. Unfortunately, modules also add complexity to the configuration process.

Try to include only the modules you need. This is usually the minimal set of modules required to run the Apache web server. Below you will find detailed information on each module, and the associated LoadModule command to enable them.

Also, keep in mind that if you disable a module, you may need to comment out any of the module’s associated directives from your configuration files. If you do not do this, it’s likely that your server won’t start. For example, if you disable the alias module, you’ll also need to comment out any ‘Alias’ or ‘ScriptAlias’ directives in the apache configuration file(s). When in doubt, try starting your server and view the HTTPD error log for misconfiguration messages.

Also note that for every ‘LoadModule’ statement, there is a corresponding ‘AddModule’ statement. You need to enable or disable them in pairs. To disable a directive, simply comment it out (by placing a ‘#’ at the beginning of a line).

Minimal modules needed to run Apache and ColdFusion

LoadModule config_log_module modules/mod_log_config.so

- Handles access and error logs, as well as log formatting.

LoadModule mime_module modules/mod_mime.so

- Determines the document type based on file extension.

LoadModule negotiation_module modules/mod_negotiation.so

- Allows for content negotiation/selection.

LoadModule dir_module modules/mod_dir.so

- Scans a directory for files specified in ‘DirectoryIndex’ (e.g. index.html) and returns them, if found.

LoadModule access_module modules/mod_access.so

- Host-based access control (allow/deny directives).

LoadModule setenvif_module modules/mod_setenvif.so

- Provides the ‘BrowserMatch*’ directive, allowing the server to respond differently to support legacy and quirky browsers.

LoadModule coldfusion_module modules/mod_coldfusion.so

- Enables the processing of ColdFusion templates.

Optionally useful modules

LoadModule autoindex_module modules/mod_autoindex.so

- Generates directory listing if a document specified by DirectoryIndex (typically index.html) does not exist. Note: this is overridden by not including the 'Indexes' option in the directory configuration. *Proceed with caution when enabling this module. It may allow an attacker to view directory listings if the directory lacks a default file (for example, index.html).*

LoadModule alias_module modules/mod_alias.so

- Provides 'Alias,' 'ScriptAlias' and 'Redirect' directive support. Use this if you have defined Aliases or ScriptAliases in your configuration. This is typically used to define your "cgi-bin" location, and inform the server that it will contain executable scripts (CGIs).

LoadModule auth_module modules/mod_auth.so

- Provides support for .htaccess files using the 'AuthUserFile' directive. If you need to limit access to files/directories based on authentication, you need this module.

LoadModule db_auth_module modules/mod_auth_db.so

- Provides a functionality similar to .htaccess, except this module uses the Berkeley DB format for quicker lookup/access.

LoadModule dbm_auth_module modules/mod_auth_dbm.so

- Similar to db_auth_module, except it uses DBM files rather than Berkeley DB files.

LoadModule cgi_module modules/mod_cgi.so

- Enables and handles the execution of CGI scripts. *Proceed with caution when enabling this module.* Poorly written CGI scripts, as well as user-uploaded scripts, can compromise security.

LoadModule env_module modules/mod_env.so

- Provides the 'PassEnv,' 'SetEnv' and 'UnsetEnv' directives for passing environment variables to CGI scripts. You may need to enable this if mod_cgi is enabled

LoadModule includes_module modules/mod_include.so

- Enables and handles all 'server-side-includes' (.shtml). It is not enabled by default unless the directory configuration has the 'Includes' option set. *The same security risk/potential exists as in mod_cgi*.

LoadModule agent_log_module modules/mod_log_agent.so
- Enables the logging of the client's 'User-Agent' (browser type).

LoadModule referer_log_module modules/mod_log_referer.so
- Enables the logging of the client's referring page (if supplied).

LoadModule userdir_module modules/mod_userdir.so
- Allows the requesting of pages from user directories. *This is a security concern!* Attackers can learn valid user names if this is enabled. Also, include a 'UserDir disabled root' configuration line to prevent an attacker from gaining read access to your whole system (or root's home directory).

Other modules (not likely to be used)

LoadModule mmap_static_module modules/mod_mmap_static.so
- Provides a way to 'map' static files into memory, decreasing the time needed to serve them. *Note: this is an experimental module. The security risks are still unknown at this time.*

LoadModule action_module modules/mod_actions.so
- Lets you define 'Script' and 'Action' directives that automatically invoke CGIs when certain request attributes (Method, etc) are met.

LoadModule mime_magic_module modules/mod_mime_magic.so
- Attempts to identify a file type by judging the contents and returning an appropriate MIME-type. *Note: this is very processor intensive*. Use standard mime_module, which assumes file-type from the file extension.

LoadModule status_module modules/mod_status.so
- Enables real-time status information via the 'server-status' handler. *Note: this may yield configuration information to an attacker*.

LoadModule info_module modules/mod_info.so
- Enables the 'server-info' handler that generates a report on how the server is configured. *Note: this may yield configuration information to an attacker*.

LoadModule asis_module modules/mod_asis.so

- Lets you send a file with pre-defined headers in the response, without using a CGI or other script. It is used in conjunction with the 'AddType' directive. *Proceed with caution when using this module. Improper implementation can provide an attacker with access to the source code scripts.*

LoadModule cern_meta_module modules/mod_cern_meta.so

- Similar to mod_asis, this module lets you define a file of headers to include with each request.

LoadModule imap_module modules/mod_imap.so

- Provides server-side image map support without the need for an imagemap CGI. *Note: this module will soon be depreciated.*

LoadModule speling_module modules/mod_speling.so

- Attempts to correct spelling mistakes and typographical errors in user-submitted URL requests. *It is possible for this module to miscalculate the misspelled requested filename and show the wrong file—which may be a file not intended for viewing by that user.*

LoadModule proxy_module modules/libproxy.so

- Enables support to make your server a HTTP/1.0 caching proxy server. *Proceed with caution, as it is possible for an attacker to 'bounce' requests off your server, take advantage of trusted-host relationships and bypass certain firewall restrictions.*

LoadModule rewrite_module modules/mod_rewrite.so

- Provides a rule-based engine used to rewrite URLs internally to different formats/URLs.

LoadModule anon_auth_module modules/mod_auth_anon.so

- Allows for 'anonymous' user logins, similar to FTP, where a user must specify the username 'anonymous' along with an email address to gain access to a web resource.

LoadModule auth_digest_module modules/mod_auth_digest.so

- An experimental module shipped with Apache 1.3.8 used for MD5-based authentication.

LoadModule digest_module modules/mod_digest.so

- Allows authentication by MD5 digests, similar to auth_digest module.

LoadModule expires_module modules/mod_expires.so

- Generates expiration headers based on your configuration defined by the 'ExpiresActive,' 'ExpiresByType' and 'ExpiresDefault' directives.

LoadModule headers_module modules/mod_headers.so

- Similar to mod_asis. It lets you specify headers to send in a request by using the 'Header' directive.

LoadModule usertrack_module modules/mod_usertrack.so

- Provides support to track users via cookies.

LoadModule example_module modules/mod_example.so

- An example of the Apache API and should NOT be loaded on production installations.

LoadModule unique_id_module modules/mod_unique_id.so

- Causes the server to generate a 'unique ID' for each request.

For more module information, see:
http://www.apache.org/docs/mod/index.html

SSL support

Apache does not natively come with SSL (encrypted transport) support. However, there are many free and commercial add-ons for Apache that do provide SSL capabilities.

Mod_SSL/OpenSSL project
http://www.modssl.org/

Covalent Raven SSL
http://www.ravenssl.com/

RedHat Secure Server
http://www.redhat.com/

Other resources

IBM Redbook
Chapter 6 in IBM's "IBM HTTP Server Powered by Apache on RS/6000" Redbook (SG24-5132-00) provides excellent information and examples of using the various available authentication mechanisms, including using .htaccess. The RedBook is available online at
http://www.redbooks.ibm.com/pubs/pdfs/redbooks/sg245132.pdf

Apache Week
Apache Week is an online newsletter focused on developments in the Apache web server. It is available at www.apacheweek.com. A good article on using user authentication (.htaccess) is available at
http://www.apacheweek.com/features/userauth

A 'hints and tips' article is also available at http://www.apacheweek.com/tips/tips

Apache security tips
The Apache team has a page of compiled security configuration tips available at
http://www.apache.org/docs/misc/security_tips.html