Apache Module mod_authz_host

Description:	Group authorizations based on host (name or IP address)
Status:	Base
Module Identifier:	authz_host_module
Source File:	mod_authz_host.c
Compatibility:	Available in Apache 2.1 and later

Summary

The directives provided by `mod_authz_host` are used in `<Directory>`, `<Files>`, and `<Location>` sections as well as `.htaccess`[1] files to control access to particular parts of the server. Access can be controlled based on the client hostname, IP address, or other characteristics of the client request, as captured in environment variables[2]. The `Allow` and `Deny` directives are used to specify which clients are or are not allowed access to the server, while the `Order` directive sets the default access state, and configures how the `Allow` and `Deny` directives interact with each other.

Both host-based access restrictions and password-based authentication may be implemented simultaneously. In that case, the `Satisfy` directive is used to determine how the two sets of restrictions interact.

In general, access restriction directives apply to all access methods (`GET`, `PUT`, `POST`, etc). This is the desired behavior in most cases. However, it is possible to restrict some methods, while leaving other methods unrestricted, by enclosing the directives in a `<Limit>` section.

Topics

Directives

See also

- `Satisfy`
- `Require`

Allow Directive

Description:	Controls which hosts can access an area of the server
Syntax:	`Allow from all|`*`host`*`|env=`*`env-variable`* `[`*`host`*`|env=`*`env-variable`*`] ...`
Context:	directory, .htaccess
Override:	Limit
Status:	Base
Module:	mod_authz_host

The `Allow` directive affects which hosts can access an area of the server. Access can be controlled by hostname, IP Address, IP Address range, or by other characteristics of the client request captured in environment variables.

The first argument to this directive is always `from`. The subsequent arguments can take three different

forms. If `Allow from all` is specified, then all hosts are allowed access, subject to the configuration of the `Deny` and `Order` directives as discussed below. To allow only particular hosts or groups of hosts to access the server, the *host* can be specified in any of the following formats:

A (partial) domain-name

Example:

```
Allow from apache.org
```

Hosts whose names match, or end in, this string are allowed access. Only complete components are matched, so the above example will match `foo.apache.org` but it will not match `fooapache.org`. This configuration will cause the server to perform a reverse DNS lookup on the client IP address, regardless of the setting of the `HostnameLookups` directive.

A full IP address

Example:

```
Allow from 10.1.2.3
```

An IP address of a host allowed access

A partial IP address

Example:

```
Allow from 10.1
```

The first 1 to 3 bytes of an IP address, for subnet restriction.

A network/netmask pair

Example:

```
Allow from 10.1.0.0/255.255.0.0
```

A network a.b.c.d, and a netmask w.x.y.z. For more fine-grained subnet restriction.

A network/nnn CIDR specification

Example:

```
Allow from 10.1.0.0/16
```

Similar to the previous case, except the netmask consists of nnn high-order 1 bits.

Note that the last three examples above match exactly the same set of hosts.

IPv6 addresses and IPv6 subnets can be specified as shown below:

```
Allow from fe80::a00:20ff:fea7:ccea
Allow from fe80::a00:20ff:fea7:ccea/10
```

The third format of the arguments to the `Allow` directive allows access to the server to be controlled based on the existence of an environment variable [2]. When `Allow from env=`*`env-variable`* is specified, then the request is allowed access if the environment variable *env-variable* exists. The server provides the ability to set environment variables in a flexible way based on characteristics of the client request using the directives provided by `mod_setenvif`. Therefore, this directive can be used to allow access based on such factors as the clients `User-Agent` (browser type), `Referer`, or other HTTP request header fields.

Example:

```
SetEnvIf User-Agent ^KnockKnock/2.0 let_me_in
<Directory /docroot>
  Order Deny,Allow
  Deny from all
  Allow from env=let_me_in
</Directory>
```

In this case, browsers with a user-agent string beginning with `KnockKnock/2.0` will be allowed access, and all others will be denied.

Deny Directive

Description:	Controls which hosts are denied access to the server
Syntax:	`Deny from all|`*`host`*`|env=`*`env-variable`* `[`*`host`*`|env=`*`env-variable`*`] ...`
Context:	directory, .htaccess
Override:	Limit
Status:	Base
Module:	mod_authz_host

This directive allows access to the server to be restricted based on hostname, IP address, or environment variables. The arguments for the `Deny` directive are identical to the arguments for the `Allow` directive.

Order Directive

Description:	Controls the default access state and the order in which Allow and Deny are evaluated.
Syntax:	`Order` *`ordering`*
Default:	`Order Deny,Allow`
Context:	directory, .htaccess
Override:	Limit
Status:	Base
Module:	mod_authz_host

The `Order` directive controls the default access state and the order in which `Allow` and `Deny` directives are evaluated. *Ordering* is one of

`Deny,Allow`
: The `Deny` directives are evaluated before the `Allow` directives. Access is allowed by default. Any client which does not match a `Deny` directive or does match an `Allow` directive will be allowed access to the server.

`Allow,Deny`
: The `Allow` directives are evaluated before the `Deny` directives. Access is denied by default. Any client which does not match an `Allow` directive or does match a `Deny` directive will be denied access to the server.

`Mutual-failure`
: Only those hosts which appear on the `Allow` list and do not appear on the `Deny` list are granted

access. This ordering has the same effect as `Order Allow,Deny` and is deprecated in favor of that configuration.

Keywords may only be separated by a comma; *no whitespace* is allowed between them. Note that in all cases every `Allow` and `Deny` statement is evaluated.

In the following example, all hosts in the apache.org domain are allowed access; all other hosts are denied access.

```
Order Deny,Allow
Deny from all
Allow from apache.org
```

In the next example, all hosts in the apache.org domain are allowed access, except for the hosts which are in the foo.apache.org subdomain, who are denied access. All hosts not in the apache.org domain are denied access because the default state is to deny access to the server.

```
Order Allow,Deny
Allow from apache.org
Deny from foo.apache.org
```

On the other hand, if the `Order` in the last example is changed to `Deny,Allow`, all hosts will be allowed access. This happens because, regardless of the actual ordering of the directives in the configuration file, the `Allow from apache.org` will be evaluated last and will override the `Deny from foo.apache.org`. All hosts not in the `apache.org` domain will also be allowed access because the default state will change to *allow*.

The presence of an `Order` directive can affect access to a part of the server even in the absence of accompanying `Allow` and `Deny` directives because of its effect on the default access state. For example,

```
<Directory /www>
  Order Allow,Deny
</Directory>
```

will deny all access to the `/www` directory because the default access state will be set to *deny*.

The `Order` directive controls the order of access directive processing only within each phase of the server's configuration processing. This implies, for example, that an `Allow` or `Deny` directive occurring in a `<Location>` section will always be evaluated after an `Allow` or `Deny` directive occurring in a `<Directory>` section or `.htaccess` file, regardless of the setting of the `Order` directive. For details on the merging of configuration sections, see the documentation on How Directory, Location and Files sections work[3].

URI References

[1] http://httpd.apache.org/docs-2.1/mod/core.html#accessfilename
[2] http://httpd.apache.org/docs-2.1/env.html
[3] http://httpd.apache.org/docs-2.1/sections.html