Restricting and Opening Access to Your Site

With CUWA

Restricting Web Access

To restrict web (HTTP) access to any directory in your application using CUWA 2.0, create/upload a .htaccess file to that directory with the following contents:

```
AuthName Cornell
AuthType All
AuthBasicAuthoritative off
#
#Use the following to restrict access
#(w/o comments) use spaces not commas
#when multiple entries are needed
#
#require valid-user
#require netid netid_name
#require permit permit_name
```

Details for the lines in the (uncommented) preamble are available in the CUWA 2.0 directive refernece, so we will focus on:

```
#require valid-user
#require netid netid_name
#require permit permit_name
```

These are the customizable parts of the template and are the ones most useful to anyone interested in controlling access to their website. They are commented out; the # should be removed to make the directive applicable.

Any member of the Cornell Community

require valid-user grants access to **anyone** with a valid Cornell NetID.

With Permits

require permit permit1 permit2 permit3 grants access to anyone listed in those permits.

More information about permits, particularly root permits, is available here. The permit administration site is here.

With Lists of NetIDs

require netid1 netid2 netid3 grants access to the users associated with those NetIDs.

Using a combination of Permits and NetIDs

To restrict to both a set of permits and a set of specific users, simply use two require lines:

```
require cit.foo.bar
require netid ewe2 elr32 abc123
```

Access will then be granted both to members of cit.foo.bar and the users ewe2, elr32 and abc123.

Restricting DAV Access

The equivalent of htaccess files for WebDAV access is available in the form of .wdaccess files. They work exactly the same way except that they apply only to DAV access. An editor is provided linked from your instance's splash page to create and manage .wdaccess files within your environment.

The editor also contains a brief summary of the information available in this wiki, and the template/example configuration available above, which is applicable both to .wdaccess and .htaccess files.

Opening Access / Removing CUWA Requirements

To remove CUWA requirements for a directory and all its subdirectories, create an htaccess as described above with the following require line, instead of `require permit` or `require netid`:

```
require noprompt
```

Subdirectories of that directory that should be closed again can then have normal CUWA rules applied via htaccesses as described in earlier sections.

Note that while this directive should work in DAV, its use there is discouraged; using this directive in WebDAV would open your source code and files to be viewable, downloadable and editable to the world.

Opening Access Without `require noprompt`

If for some reason you wish to open access without the use of `require noprompt` and/or to bypass CUWA all together, there is a method that allows this to be done via htaccess. However, it will also **disable by default** any standard authentication you add via htaccess for any subdirectories of the opened directory unless you take extra steps, so use with care. For most cases, `require noprompt` should do what you need without lots of interesting side effects. For the rest of the time, there's `Satisfy`.

In the directory you wish to open, create an .htaccess with:

```
Satisfy any
Allow from all
Order allow,deny
```

This configuration - in particular, `Satisfy any` - will bypass CUWA altogether, **including** NetID logging (even for logged-in users and subdirectories of a "require noprompt"). `Allow from all` and `Order allow,deny` are there to ensure that any higher level Allow/Order configurations aren't inherited and place similar restrictions. For any subdirectories of this opened directory, you will need to add the line

```
Satisfy all
```

to cause any CUWA htaccess-based restrictions to be picked up once again.

Note that using Satisfy in combination with Allow/Order/Require can cause interesting and unusual auth behaviors, particularly if you are using several layers of htaccess. For most intents and purposes, `require noprompt` should do what you need (i.e. open access to those without Cornell NetIDs). If you need or choose to use this method, though, it is strongly advisable to at least be familiar with the Satisfy directive and how it works.

httpd.apache.org - Satisfy (Apache 2.2)
httpd.apache.org - Allow (Apache 2.2)
httpd.apache.org - Order (Apache 2.2)

More information on CUWA Access Control

- Integrating CUWA With Your Application
- CUWA 2.0 Directive Reference
- CUWA 2.0 Wiki

Restricting Access With Apache Basic/Digest Authentication

HTTP Basic Authentication and HTTP Digest Authentication are available in LAMP, but currently must be configured via a request to webservices@cornell.edu. If you need either of these methods set up to authenticate users to your site, please send us a request with the details.

Help! I created a .ht|.wdaccess file and now I get HTTP 500

Your logs, also available from your splash page (or your staging URL + /logs, e.g. lamp.cit.cornell.edu/logs) should contain clues on what went wrong. Usually it's a little typo that is readily fixed. If you are really stuck, feel free to contact webservices-l for assistance.

What about other .htaccess directives?

Just about any directive that the Apache documentation says is legal in .htaccess files is available to you in your environment. Use with caution and care, and enjoy.

Note that this does not apply to DAV/.wdaccess files, as most non-auth based directives don't have much meaning to WebDAV.