Coldfusion 9.0 - SSL

Note that these directions only apply for sites that were created AFTER January 15th, 2011. Sites created prior to this date may be using a "legacy" configuration that may differ from the following.

Sites that utilize CU WebAuth are required to use SSL.

There are two options for having an SSL configured. One is referred to as "HTTPS" and one as "BOTH".

If you wish to have ALL traffic protected by SSL we recommend that you request "HTTPS" for the protocol. This will create a redirect so that any traffic that goes to http://\[my site]/ will be redirected to https://\[my site]/.

In some cases you may wish to only have a portion of your site protected by SSL while the option portion is not. In this case if you specify "BOTH" then you will be responsible for determining whether a page can be displayed using SSL or non-SSL.

When using "BOTH" non-SSL connections have NO restrictions, that means that all files are publicly available unless you explicitly put in a ".htaccess" to protect the directory using CU WebAuth.

Some examples of restricting access via a ".htaccess" file:

Require SSL (will just cause a 403 error)

```
SSLRequireSSL
```

Redirect to SSL

```
RewriteEngine On
RewriteCond %{HTTPS} off
RewriteRule (.*) https://%{HTTP_HOST}%{REQUEST_URI}/$1
```

Note that if you protect a directory with CU WebAuth and you go to the non-SSL site you will get the following error "CUWebAuth error... Server is not properly configured. Check the Kerberos principal.". In this case you need to be sure to specify the "https" URL and not the "http" URL to avoid getting the error. If you wish to setup a redirect from the non-SSL site to the SSL site for the ".htaccess" protected directory you'll need to specify the redirect in a custom virtualhost include.

Migrating from a "legacy" configuration

If your website was created prior to January 15th, you may need to update your ".htaccess" files to conform with the requirement that CUWebAuth is only configured for SSL sites. The need to update configurations can likely occur if you previously had both public/private portions of your website and used CUWebAuth to protect portions of your site. If ALL of your content is CUWebAuth protected then we recommend a configuration of "HTTPS" which will redirect all traffic to SSL and preserve the behavior that previously existed. If you have the need to have both public/private portions of your site then you'll likely need to following the following.

The following is technical information on how to update existing configurations.

Prior to January 15th the configuration for CUWebAuth took the position of all content was CUWebAuth protected and you would need to explicitly disable CUWebAuth to enable access to the public. The new configuration takes that reverse position that CUWebAuth is disabled for non-SSL traffic and SSL traffic is configured similar to the previous configuration of CUWebAuth is enabled by default.

This can create problems with existing configurations and you may need to update your sites ".htaccess" files to conform to the new configuration.

Example of a problem configuration #1 (require noprompt)

Some websites, in particular CommonSpot sites, have a ".htaccess" file that contains the following:

```
AuthName Cornell
AuthType All
require noprompt
```

This will case a problem for new "HTTP" configurations since CUWebAuth is not loaded for non-SSL traffic. One possible fix is to remove "require noprompt" if you wish to have no CUWebAuth protection. Note that special care should be taken to ensure that there are no directories beneath that were expecting cuwebauth to be enabled. This can be tricky configuration to verify that there is no risk of exposing files that should be protected.

Example of a problem configuration #2 (require valid-user)

If you have a directory that only contains:

```
htdocs/mysite/secure/.htaccess:
AuthName Cornell
AuthType All
require valid-user
```

One option that you may do is to remove the "require valid-user" and just replace it with:

```
htdocs/mysite/secure/.htaccess:
RewriteEngine On
RewriteCond %{HTTPS} off
RewriteRule ^(.*) https://%{HTTP_HOST}%{REQUEST_URI}
```

By default all SSL traffic has "require valid-user" set. As long as you do NOT have "satisfy any" set at the top-level directory this should work (this assumes that you want all SSL traffic to require valid-user). If your application performs its own authorization checks (i.e. verifies that netid "xyz123" is listed in your "users" database table or checks the permit by examining the CUWA_GROUPS environment variable) this solution may work.

Example of a problem configuration #3 (require netid or permit)

Some websites will have a sub-directory that contains something like the following:

```
htdocs/mysite/secure/.htaccess:
AuthName Cornell
AuthType All
require permit my.group
```

In this case this configuration will NOT work with non-SSL traffic. You will need to ensure that an SSL certificate is obtained for your site and that CUWebAuth is enabled. Once you have SSL enabled you may notice that requests to the secure directory using http will generate a 500 error while requests to the https directory will work.

This case provides 3 possible solutions:

Solution A (recommended): VirtualHost Include

With new virtualhost configuration it is possible to request "Custom VirtualHost Include" files. This will provide you a configuration file in your webdav area that will get included in the Apache VirtualHost configuration for your website. This is a very powerful feature and should be only utilized by advanced users.

Using the VirtualHost include configuration you can specify that your non-SSL website redirect to the SSL website for your protected directories, i.e.:

```
Redirect /secure https://www.\[mysite\].cornell.edu/secure
```

Specifying this in the VirtualHost configuration will treat HTTP separate from HTTPS (doing the same in a ".htaccess" file would redirect ALL traffic and create a loop/broken configuration).

Solution B: HTTPS

The "HTTPS" configuration will mimic the exact same behavior of the previous site, but it will require that ALL traffic is SSL protected. This can be a problem for sites that may be serving content like RSS feeds that might produce errors when SSL/non-SSL resources are linked. This may be suitable for an administrative application that is only used by authenticated users and could also provide for a more secure configuration.

Solution C: custom 403 error page

A very kludgy solution is to override the 403 error page and have it perform the redirect. This assumes that you add "SSLRequireSSL" to the existing ".htaccess" file that is producing the 500-error and override the top-level ErrorDocument 403 document. This description is intentionally vague and it is left as an exercise to implement. The major problem with this plan is that it will override a default error page that is also used when directory listings are prohibited and/or cuwebauth authorized denied errors.