Restricting CF9 websites

The page "Restricting and Opening Access to Your Site" has general information about utilizing CU WebAuth: https://confluence.cornell.edu/x/hc9gBg.

Information about university data policy can be found at: http://www.it.cornell.edu/security/data/index.cfm

General information about application security can be found at: https://www.owasp.org

This page is meant to provide some technical details on options of how to secure your website while using the Hosting CF9 environment.

1. Application based authorization

This is the preferred method of performing authorization on an application is to either check the NetID or permit against a known authorization list. This ensure that if the application is ever placed into a data area that is not protected by CU WebAuth it will not inadvertently share privileged information (fail closed vs. fail open).

2. Public/Private sites

In some cases you may want to have non-SSL public site and a private SSL protected site that uses the same name. This can be accomplished by following the directions at https://confluence.cornell.edu/x/9QobCg. SPECIAL care should be taken to ensure private content is protected. It is possible to have content that IS protected by CU WebAuth on the SSL portion of a website to NOT be protected on the non-SSL portion of the website. This makes sense for a content management system where you want to have a public portion and restricted administrative function, but can be a DISASTER if you do not do application based authorization. This can be a very tricky configuration and special care should be taken to ensure that it is done properly.

3. .htaccess vs. Directory vs. Location Restrictions

In CF9 there are now 3 different ways that you can configure authorization! This may seem really confusing, but there is a reason that all 3 are supported.

- .htaccess
  This is the traditional method of restricting a directory and can be used to limit access to a website.
- Directory
  Using a Customer VirtualHost include it is possible to set a "Directory" level restriction in the Apache VirtualHost configuration. This can work effectively the same as a ".htaccess" file, but is stored in the server configuration.
- Location
  A Location block is independent of directory structure and may be necessary if you have a script/process that does not map to a physical directory. For ColdFusion this is less of a issue, but it is documented for completeness.

Be VERY VERY careful if you choose to mix any of the above. It is possible to have competing configurations that override each other. This is especially true if you have multiple virtualhosts that map to the same data directory. It is best to choose one of the above options and use it consistently to avoid issues.

4. Suggested layout for public site

If you are going to have a mixture of both public and private data it is recommended to not specify a top-level ".htaccess" file, but instead only explicitly protect content either through application authorization or protected sub-directories. A layout could look like:

```
htdocs/site1/index.cfm
htdocs/site1/secure_pdfs/.htaccess
htdocs/site1/secure_pdfs/private.pdf
htdocs/site2/.htaccess
```

In the above example the "secure_pdfs/.htaccess" file contains:

```
AuthName Cornell
AuthType All
AuthBasicAuthoritative off
require permit cit.coldfusion.support
```

The "index.cfm" contains:

```
<cfoutput>
#REMOTE_USER#
</cfoutput>
<cfif REMOTE_USER EQ "emc256">
Hello you are Eric
<cfelse>
You are        not authorized
</cfif>
```

When accessing the non-SSL portion of the site you'll only see (http://site1.cornell.edu):

```
You are not authorized
```

When I access the SSL portion I see (https://site1.cornell.edu):

```
emc256 Hello you are Eric
```

When you access the SSL portion you see:

```
You are not authorized
```

By placing the authentication/authorization decision in your application code it ensures that you do not inadvertently expose privileged information.

In this example this also allows you to have "site1.cornell.edu that is "public" and "site2.cornell.edu" that is restricted like below. Be careful not to place a top-level ".htaccess" file.

5. Suggested layout for a restricted site

If all your access is going to require authentication it is still a best practice to utilize application based authorization, but you have further options if you want to restrict all your sites. If you want to restrict all your sites you can place a top-level ".htaccess" file in your htdocs directory.

```
htdocs/.htaccess
htdocs/site1/index.cfm
htdocs/site2/index.cfm
```

In the above example the ".htaccess" file contains:

```
AuthName Cornell
AuthType All
AuthBasicAuthoritative off
require permit cit.coldfusion.support
```

Unlike in the previous example both site1/site2 will inherit the permission that is set in the top-level restricted directory.

6. Satsify any/all

If you need to open access to the top-level of your SSL site, but want to secure sub-directories in your site you need to do a careful mixture of "satisfy any" and "satisfy all".

To open up access at the top:

```
htdocs/site1/.htaccess:
satisfy any
```

to protect a sub directory:

```
htdocs/site1/secure/.htaccess:
satisfy all
require netid emc256
```

Note that if you do not add "satisfy all" the "secure" directory will not be protected!

7. Tests/Examples

The attached zip file. Provides some tests/examples of using CU WebAuth. To use the examples you will need to upload the contents of the zip file so that it can be accessed at: https://\[your website]/cuwebauth_test/