# TechTip: How to Secure the Web Query Login Page (or Any Other Web-Based Application)

Published Thursday, 05 March 2009 19:00 by MC Press On-line [Reprinted with permission from iTechnology Manager, published by MC Press, LP; http://www.mcpressonline.com.]

Written by Robert Andrews- robert.andrews@us.ibm.com

## Add another layer of security to all of your Web applications.

The IBM i is a great platform for running Web applications. These applications can take advantage of Apache, PHP, and MySQL, all running natively on IBM i. Or you may be using a purchased application such as DB2 Web Query for IBM i. One common thread amongst them is that they can use the IBM i user profiles and passwords. This is both a blessing and a curse. While it allows for fewer user profiles for the end user, it also exposes your administrative user profiles, such as QSECOFR. A user at the login screen could put in QSECOFR, make three wrong guesses at the password, and lock out the account. In this TechTip, I will provide a method of securing these Web-based login interfaces.

The method to accomplish this is to first present the users with a login and password box that does not authenticate against user profiles. Once they pass this first challenge, they will then be presented with the application login, which does go against their user profiles. As an example, I will be using the DB2 Web Query application.

The first step is to create a validation list. This is done via the Create Validation List command:

```
CRTVLDL VLDL(MYLIB/WEBQUERY)
```

You then need to make sure that the application can read the list. Therefore, you need to grant *PUBLIC *USE authority:

```
GRTOBJAUT OBJ(MYLIB/WEBQUERY) OBJTYPE(*VLDL) USER(*PUBLIC) AUT(*USE)
```

Working with validations from a command line is impossible. There are APIs that can be called, but those would require you to write a program. Luckily, the HTTP Administration tool provides a GUI to work with them. From a Web browser, go to http://your_system:2001/HTTPAdmin. Log in and click the Advanced section on top. From there, click Internet Users and Groups in the second row. In the task list on the left will be an option to Add Internet User. Here, you set the user name and password you want from the users before they get access to the real application login page. For the validation list parameter, enter mylib/webquery (as shown in Figure 1). Click Apply to add the user name. You can add as many user names as you like.

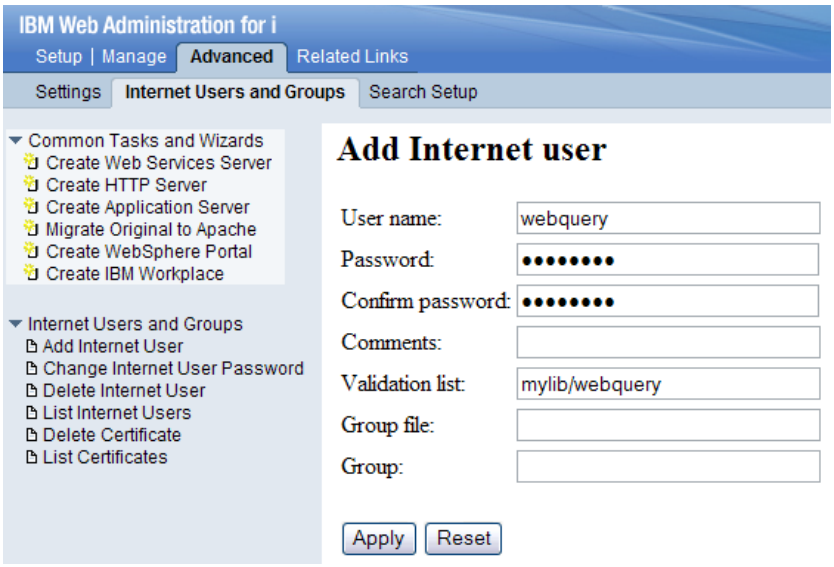


Figure 1: Add an Internet user.

Now that you have the validation list created, you need to set the Web application to use it. For this, go to the Apache configuration. On the top bar, select Setup. On the second bar, select HTTP Servers. From the drop-down box for Server, select the Web instance for your application. Here, for Web Query, I selected WQLWI7. Since security is set at the location level, select the location drop-down and change it to the Web server root, /. If you want more-granular control, you can add locations by selecting Container Management on the left and then the Locations tab.

Once you are at Location /, select Security on the left and choose the Authentication tab. Set the User authentication method to Internet users in the validation list. Enter the realm name for the application. This is used in the login prompt that is presented to the users. Keep in mind they will not see the Web application yet when they are prompted for this user name and password. You will see this in Figure 5 below. Click Add to get a line for a new authorization list. Again, point to your validation list as mylib/webquery and click Continue and Apply.

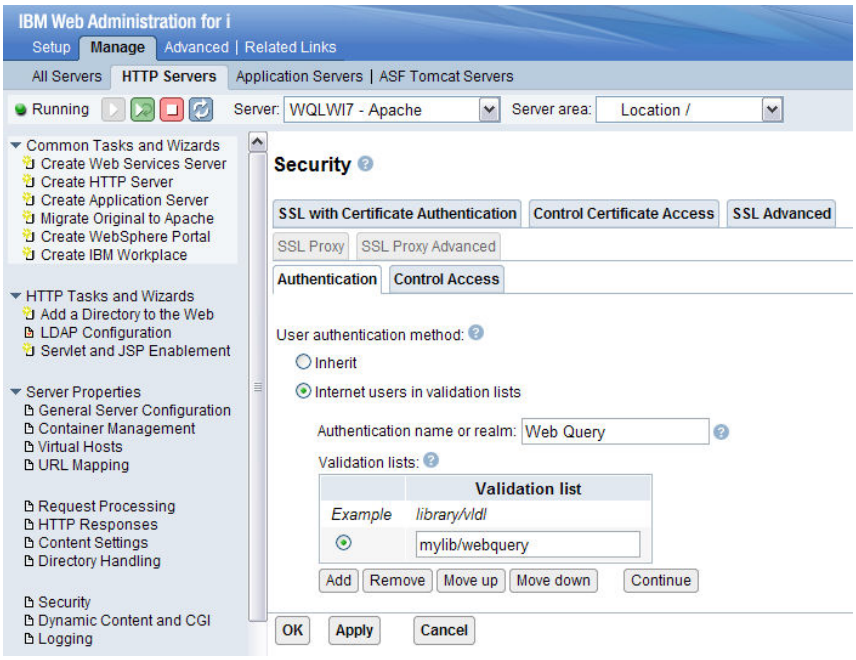


Figure 2: Choose authentication settings.

Click on the Control Access tab to put your new authentication settings to work. When we define Web security, it is based on two factors: who and where. Normally, these are set to anyone from anywhere. However, you now want to tell the application to allow access from anywhere provided the users can authenticate to the validation list. To do this, change the "who" section to require authentication: a valid user name and password. Remember from Figure 2, we told the application to look at the validation list for valid user authentication. This is shown in Figure 3.

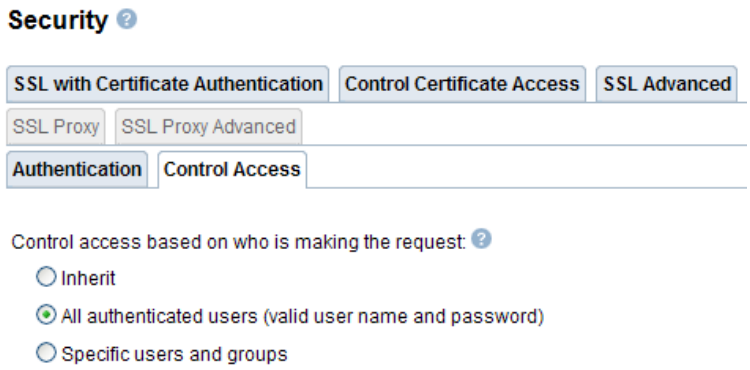


Figure 3: Control access for "who."

You don't need to adjust the "where" setting. By default, it is set to allow access from anywhere. However, if you wanted to further secure it by allowing only certain IP addresses, you could. You could use this to limit access only to users while in the office or to a set of certain IP addresses.

The last step is to tell the application to enforce both the who (someone on our validation list) and the where (anywhere) as shown in Figure 4. If you set the control access policy to who or where, then

everyone would pass the where test and not be prompted. Again, you could get fancy here by not prompting people in the office while prompting those coming from the public Internet. Apply your changes.

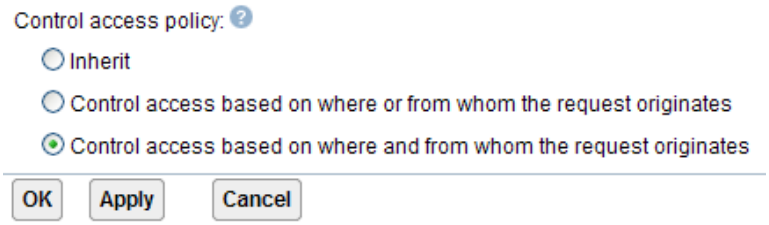


Figure 4: Choose your control access policy.

Since these changes update the Apache configuration file, you need to restart your Web server for the changes to be picked up. Once you do, when users attempt to log in to the Web application, they will first be presented with the pop-up login box shown in Figure 5. Here, you can see where the realm name is shown.


Figure 5: Internet users can now log in.

Once the users provide a valid login and password from the validation list, they are presented with the Web application login page. Here, we can see Web Query as a sample. This login then uses the IBM i user profiles and passwords.

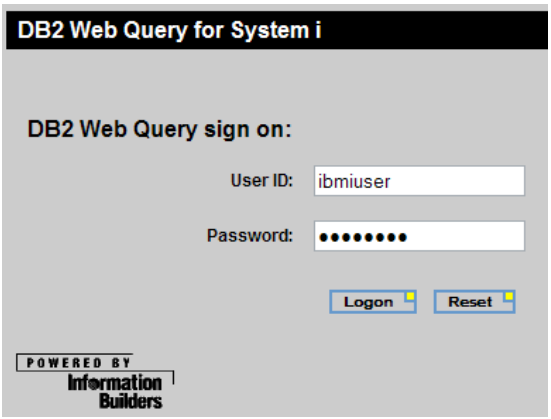


Figure 6: Users can log in to Web Query with their IBM i user profiles.

In this TechTip, I introduced you to an additional layer of security that you can add to your Web applications. While I provided a very basic example, I hope to have caught your interest in the various other ways you can add security to your Web-based applications.