

## ***Secure Application on IBM WebSphere Portal with Secure Socket Layer (SSL)***

***Protect data in your applications by enabling SSL security  
between IBM HTTP Server and WebSphere Portal Server.***

*By Raghu Macha*

When making data available to the public on Web sites, it's crucial you protect that data, especially when working with sensitive applications such as those for the financial services industry. The mark-up WebSphere Portal generates can be transmitted between the server and a browser across a variety of public and private networks. To protect this data, you must secure the HTTP server and Portal servers.

This article shows you how to enable Secure Sockets Layer (SSL) security between IBM HTTP Server and WebSphere Portal Server because it's the most common place to use SSL, the mark-up generated by WebSphere Portal can be transmitted between the server and a browser across a variety of public and private networks. SSL can protect the data in case it is intercepted or cached on an insecure node. Configuring the portal-to-user's browser link for SSL will encrypt all communication starting with the portal login screen and the communication between user's browser and server will remain confidential.

*In this article, I show you how to:*

1. Generate a **Key database file** and a **Self Signed certificate** for HTTP Server: Where Key database file is used to store the certificate, and Self Signed Certificate is the temporary certificate that is used in this article to enable SSL security. For the production environment, you will be generating a Certificate Signing Request (CSR) instead of creating one and send it to the SSL certificate providers (e.g. Verisign, Thawte, Entrust and InstantSSL etc).

**Note:** When you are generating Certificate Signing Request, please ensure that the common name in your CSR is one of the following:

- Your fully qualified domain name (e.g. test.vic.com)
- Your public IP address.

When you Generate Certificate Signing Request (CSR), it creates a private key for your server, and once you send this key to Certificate Authority, they will create an official certificate matching to your private key and issue it to you, and then you can use that in your production environment.

*The CSR looks like this:*

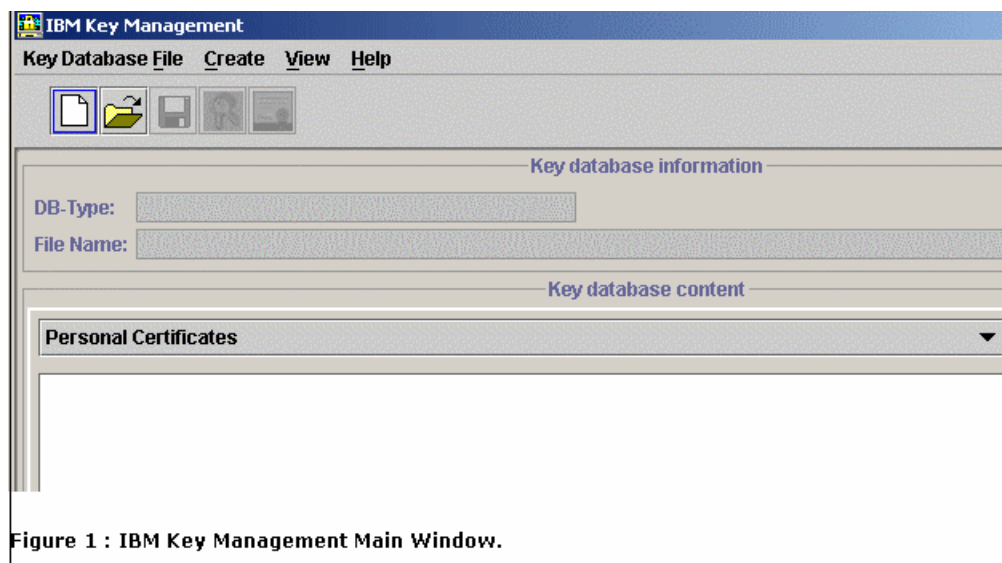
```
-----BEGIN NEW CERTIFICATE REQUEST-----  
MIIBcTCB2wIBADAYMQswCQYDVQQGEwJVUzEMMAoGA1UEChMDdmljMRUw  
EwYDVQQDD6raEakf/96ZG0BtkBu7I9k8GhNoK6IEuhyYx5u8lx2Ee/Uh9KTrdydK6  
th2ykqi  
9aVt7VZiwselJDAubMF/N2dDHYvh8D0FNszb4XMlzQbfYwU5QQ==  
-----END NEW CERTIFICATE REQUEST-----
```

2. Configure IBM HTTP Server for HTTPS support; this is the first place to protect because it's the first gateway to access your application.
3. Add new host entry in WebSphere Application Server.
4. Modify WebSphere Portal Server files to accept SSL connections.

This article describes a non-production environment installed on Windows; you might have to get certificates from Certificate Authority for your production environment. As I mentioned before, you need to generate Certificate Signing Request (CSR) from your server and send it to the Certificate Authority to get one.

*Follow these steps to generate a certificate to protect your HTTP Server:*

1. Open your Windows Explorer and navigate to C:\Program Files\IBMgsk5\bin.
2. Double click on gsk5ikm.exe to run the IBM Key management application (figure 1).



3. From the Key Management main menu, select Key Database File > New to create a new key database file.
4. Select CMS Key database file from the drop down box, enter the file name (I use IHSCerts.kdb for this example) and the Location as C:\Program Files\IBMHTTPServer\conf. Click on OK to create Key database file in the specified location.

5. In the pop-up window, enter the Password and Confirm password (e.g., password), check "Stash the password to a file?" and click on OK to save the password to key database file (figure 2).



Figure 2 : Set the Password for IHS Certificate.

The key management application saves the password to the location you entered in step number 3.

6. From the Key Management main menu, select Create > New Self-Signed Certificate to create a new certificate.
7. Enter the following values for your new certificate:  
**Key Label (optional):** IHS  
**Common Name:** your domain name, for this example I use test.vic.com  
**Organization:** Enter you organization name, for this example, I use VIC  
You can leave all other optional fields empty and click on OK to create the new certificate.
8. Close the IBM Key Management window.

Now, you configure HTTP Server to support HTTPS. You must make configuration changes to the httpd.conf file under HTTP Server install to handle requests on the HTTPS port (443). I'll show you how to do these edits and make the required changes.

1. Stop HTTP Server from services and make a backup copy of httpd.conf file in case anything goes wrong.
2. Edit httpd.conf file in a test editor to make changes for HTTPS support.
3. Add the following highlighted entry to the list of **LoadModule** directives near the start of the file:  
**#LoadModule usertrack\_module modules/ApacheModuleUserTrack.dll**  
**#LoadModule proxy\_module modules/ApacheModuleProxy.dll**  
**LoadModule ibm\_ssl\_module modules/IBMModuleSSL128.dll**

4. Comment Port 80 to allow Http server to use only HTTPS Port 443.

```
# Port: The port the standalone listens to.  
# Port 80
```

5. Enter the HTTPS port number as follows:

```
#Listen 3000  
#Listen 12.34.56.78:80  
Listen 443
```

6. Create a **VirtualHost** entry for the SSL port under the VirtualHost comments, entering your domain name instead of portal.hursley.ibm.com, for this example I entered test.vic.com:

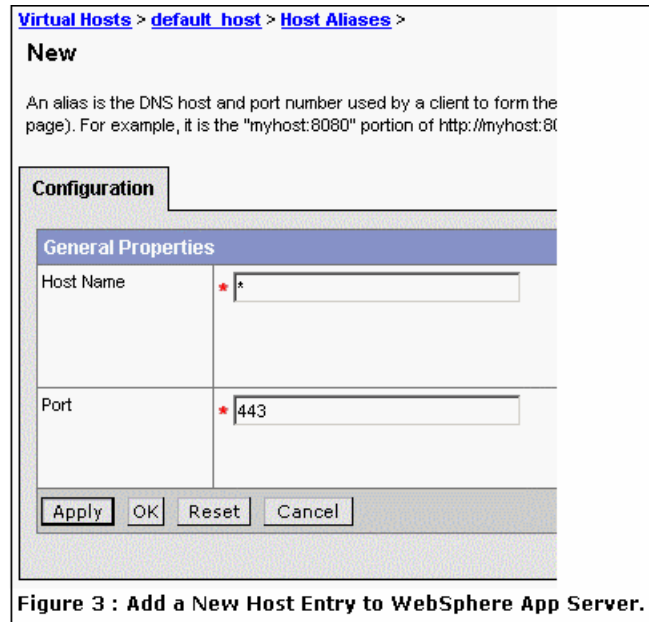
```
#<VirtualHost host.some_domain.com>  
#ServerAdmin webmaster@host.some_domain.com  
#DocumentRoot /www/docs/host.some_domain.com  
#ServerName host.some_domain.com  
#ErrorLog logs/host.some_domain.com -error.log  
#TransferLog logs/host.some_domain.com -access.log  
#</VirtualHost>
```

```
<VirtualHost test.vic.com:443>  
ServerName test.vic.com  
DocumentRoot "C:\IBMHttpServer\htdocs"  
SSLEnable  
</VirtualHost>  
SSLDisable  
Keyfile "C:\IBMHttpServer\conf\IHSCerts.kdb"  
SSLV2Timeout 100  
SSLV3Timeout 1000
```

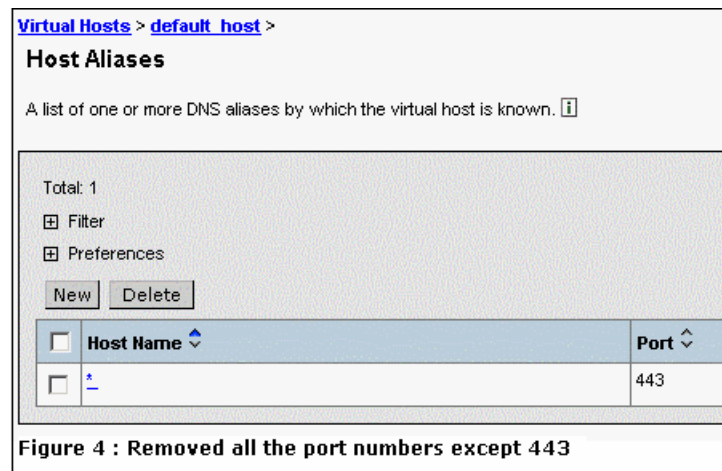
7. Make sure the values you entered above are correct and save the httpd.conf file.
8. Start HTTP Server from services to test the configuration changes you made.
9. Open your Web browser and enter your URL (https://test.vic.com, in this example), to confirm your configuration was successful.

*Now, follow these steps to add a new host entry with port number 443 and remove the rest to WAS:*

1. Log in to WebSphere Application Server Admin Console. You might have to provide a WAS admin user ID and password if you've enabled security with LDAP.
2. Navigate to Environment > Virtual Hosts > Default Host > Host Aliases to enter the HTTPS host information.
3. Click on New and enter Asterisk (\*) to take default hostname in Host Name field and 443 in Port field (figure 3).
4. Click on Apply and OK to save changes.



5. Now you are back to **Virtual Hosts > default host** page on WAS Admin console, select the check boxes for all the Ports except 443 and click on Delete button to remove all the ports, so that all the applications run on only HTTPS Port 443.
- 6.



*Now, follow these steps to modify WebSphere Portal Server files to accept the SSL Connection:*

1. Stop the WebSphere Portal and Application Server before you make any changes to their configuration files as they had started to make changes to add and remove host entry as shown in Figure 3 & 4.
2. Open your Windows Explorer and navigate to <wps-root>\shared\app\config\services and edit the ConfigService.properties file in the text editor.

3. Enter the following values to enable SSL connection:

```
redirect.login.ssl = true  
host.port.https = 443  
redirect.logout.ssl = true
```

4. Locate all the JSP files that provide the login button, the following JSP files contain the login button:

```
<was_root>installedApps\test\wps.ear\wps.war\themes\htm\ToolBarInclude.jsp  
<was_root>installedApps\test\wps.ear\wps.war\themes\htm\Corporate\ToolBarInclude.jsp  
<was_root>installedApps\test\wps.ear\wps.war\themes\htm\Engineering\ToolBarInclude.jsp  
<was_root>installedApps\test\wps.ear\wps.war\themes\htm\Finance\ToolBarInclude.jsp  
<was_root>installedApps\test\wps.ear\wps.war\themes\htm\Science\ToolBarInclude.jsp  
<was_root>installedApps\test\wps.ear\wps.war\themes\htm\YourCoFinancial\ToolBarInclude.jsp  
<was_root>installedApps\test\wps.ear\wps.war\themes\htm\YourCoFinancial2\ToolBarInclude.jsp
```

**Note:** There might be more JSP files that may contain login button if you have added your own custom themes.

Make the following modification as shown in the bold to the above JSP files:

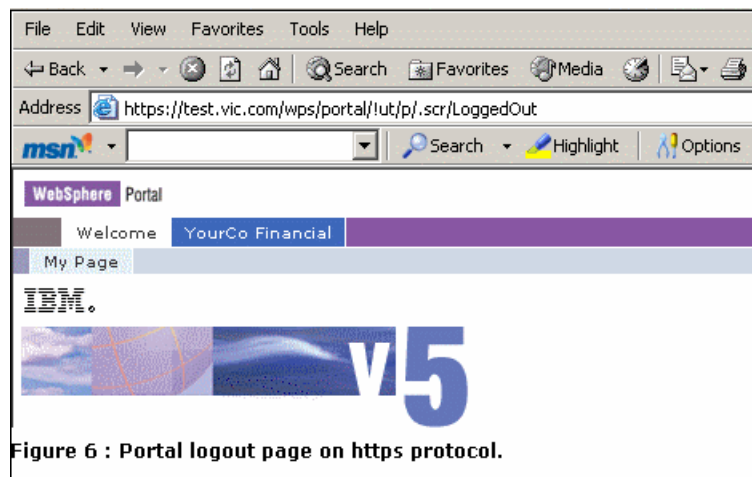
```
<%-- login button --%>  
<wps:if loggedIn="no" notScreen="Login">  
    <td class="wpsToolBar" valign="middle" align="<%=bidiAlignRight%>"  
  
    nowrap><a class="wpsToolBarLink" href='<wps:url home="public" screen="Login"  
ssl="true"/>'><wps:text key="link.login" bundle="nls.engine"/></a>  
    </td></wps:if>
```

5. Start WebSphere Portal and Application Server to apply the changes you made.
6. Log into WebSphere Application Server Admin Console to update Web Server Plug-in since you made changes to WAS configuration by adding a new host entry with port 443 and removing the other port numbers as shown in Figure 3 & 4. You have to regenerate Web Server Plug-in every time you make changes to your WebSphere configuration.
7. Expand Environment and click on Update Web Server Plug-in.
8. Click on OK to update the Web server plug-in.
9. Restart IBM HTTP Server from services to pick up the regenerated plug-in.
10. Open Internet Explorer and enter the portal URL to test the security, e.g. : <https://test.vic.com/wps/portal>.

11. **Notice:** The portal runs with only https protocol now. And now you'll get the security alert shown in figure 4 since we are using Self Signed Certificate for this example. You won't be seeing this Security Alert window once you start using the certificate that has been issued by Certificate Authority. Click on the Yes button to proceed.



12. Click on the Login button in the top right corner and login to portal with username and password, for this example I logged in as wpsadmin / wpsadmin, once you logged in click on Logout button and notice that it still keeps portal on https protocol (Figure 5).



***Congratulations! You've successfully enabled SSL Security between WebSphere Portal and Web browsers.***