|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | **10/6/2010**   |  | | --- | | [**Step by Step Kerberos Authentication for SharePoint 2010**](http://www.thesharepointblog.net/Lists/Posts/Post.aspx?ID=43) |   **By: Christian Holslin**  **Introduction**  **Kerberos authentication**, created at MIT and named after Hades’ three-headed guard dog Cerberus (according to Wikipedia), has been around for decades.  The latest version 5, implemented currently by **Active Directory**, was released in 1993.  The protocol is designed to provide rapid, secure authentication to users on a multi-system network, or **“farm”** as we like to call them.  **Advantages over Traditional Windows Authentication**  The main advantage of **Kerberos** over NTLM or forms-based authentication is the ability for a user’s identity to **securely traverse multiple servers** **without requiring a re-key** of the user’s credentials.  This concept is referred to as **single sign-on**: login once to access everything.  A secondary advantage is speed.  Authenticating connections with Kerberos tokens is considerably faster than other methods.  **Platform Uniformity**  Another advantage is platform uniformity.  **Any application, that you wrote, or Microsoft wrote, or anyone wrote, which uses Windows Authentication can automatically use Kerberos**.  It’s built in to Windows and Active Directory.  It doesn’t require custom code like a forms-based or claims-aware provider.  Enabling it is as simple as telling the web.config to use it.  **Necessity**  Many farm scenarios do not warrant Kerberos authentication.  How can you tell if yours does?  There is a simple test: **the double-hop**.  Draw a quick diagram of your farm topology.  If you have any servers which are more than two degrees of separation away from your client, you will need Kerberos authentication **only** **if you need to delegate access to those resources**.  The figure below shows the double-hop scenario.  Step_by_Step_Kerberos_Authentication_for_SharePoint_2010  **Figure 1: The Double-Hop**  Each connection, or “hop,” must be authenticated.  Thus, the SharePoint server must establish a secure, authenticated connection to SQL in order to return data for the user.  If the data connections above need to **impersonate the user**, the connections must use **delegation**.  Kerberos authentication allows SharePoint and SQL Server to implement delegation.  **Real-World Examples**  The most common example of Kerberos in practice involves Reporting Services.  A user browses to a SharePoint document library to run a Report with data in a SQL Server database.  SharePoint and SQL Server both implement Kerberos authentication to allow the user to view the Report using the user’s own credentials.  No login prompts, no proxy accounts, **no stored credentials**.  **Setup**  Setting up Kerberos authentication for SharePoint and SQL Server takes only a few minutes.  Follow the steps below to get it running in your farm.  We will assume that SharePoint requires **classic mode authentication** for the Web Application.  (Obviously, you will need to change CONTOSO to your Domain name and use your actual service accounts.)  **1. Configure SQL Server**  Configuring SQL Server to use Kerberos is easy.  Create a Service Principal Name for your SQL Server by running the setspn.exe utility from the command-line.  NOTE: you will need to be a Domain Administrator to do this:  Step_by_Step_Kerberos_Authentication_for_SharePoint_2010  **Figure 2: setspn.exe Syntax**  **Service Principal Names**  You will need to become familiar with **Service Principal Names** to setup Kerberos.  They are composed of the following pieces:   |  |  |  |  |  | | --- | --- | --- | --- | --- | | Service | | | Principal | | | Service Class | Endpoint | Port | Domain | User | | **MSSQLSvc** | **DB-SRV-01** | **1433** | **CONTOSO** | **SqlServer** | | This is the unique class name of the service. It differs between different types of services. | This is the DNS address where the service is accessed. In this case, it’s the server name, but it can also be the fully-qualified domain name like:  db-srv-01.contoso.local  - or an alias like -  database.contoso.local | The port is needed if it is not a standard port for the Service Class. | This is the NetBIOS domain name of the Active Directory where the service account resides. | This is the login name for the service account itself. |     **As far as I know, the Service Class is case-sensitive.**  For good measure, Microsoft recommends creating multiple Service Principal Names.  The reason why: **the client application creates the Service Principal Name when it sends it to the server**.  If the client application choses to include the port number, or not include the port number, you should be ready.  The solution: create all of the following SPNs for SQL Server:  ·         MSSQLSvc/DB-SRV-01 CONTOSO\SqlServer  ·         MSSQLSvc/DB-SRV-01:1433 CONTOSO\SqlServer  ·         MSSQLSvc/DB-SRV-01.contoso.local CONTOSO\SqlServer  ·         MSSQLSvc/DB-SRV-01.contoso.local:1433 CONTOSO\SqlServer  Note the **variation in the Endpoint and Port**.  We do this to ensure that we cover all the possible combinations that a client application could throw at SQL Server.  This is the best practice.  **2. Create a Web Application**  Create a new Web Application in SharePoint 2010 to use with Kerberos authentication.  Pick **Classic Mode Authentication** and make sure NTLM is used.  **This Web Application will be created as the Default Zone**.  We want to put this on a **non-standard port** and **use NTLM authentication** to **ensure that we can always access it** from the SharePoint server itself.  **Note: you must use a Domain Account for the application pool identity.**  Step_by_Step_Kerberos_Authentication_for_SharePoint_2010  **Figure 3: New Web Application**    **3. Extend the Web Application to use Kerberos Authentication**  Extend the Web Application you just created.  Set the **Zone** to **Intranet** and put the site on **Port 80**.  Use the host header **intranet.contoso.local**:  Step_by_Step_Kerberos_Authentication_for_SharePoint_2010  **Figure 4: Web Application Extension**  When you click OK you will get a warning about Kerberos.  Don’t worry: the Service Principal Name can be created before or after the Web Application Extension.  **6. Create the DNS Record**  Your server needs a static IP address and a DNS record to be accessed by users.  When Kerberos is involved, you must be sure that you create an **A** (for address) record and not a **CNAME** (canonical name, or alias) record for the SharePoint Web Application Extension:  Step_by_Step_Kerberos_Authentication_for_SharePoint_2010  **Figure 5: New DNS Record**  Enter the **IP address** of the SharePoint server and **hostname of the Web Application Extension** into the box and click **Add Host** to save the new DNS record.  The automatically generated FQDN should read **intranet.contoso.local**.  **4. Create a Service Principal Name**  Just like we did for SQL Server, create a Service Principal Name for the SharePoint Web Application Extension:  Step_by_Step_Kerberos_Authentication_for_SharePoint_2010  **Figure 6: SharePoint SPN**  The SharePoint Service Principal Name breakdown is as follows:   |  |  |  |  |  | | --- | --- | --- | --- | --- | | Service | | | Principal | | | Service Class | Endpoint | Port | Domain | User | | **HTTP** | **intranet.contoso.local** |  | **CONTOSO** | **SP\_WebApp** | | HTTP works for http and https connections. | This is the DNS address where SharePoint is accessed. In this case, it’s the URL of the Web Application Extension | 80 is a standard port, therefore we don’t need to include it. | This is the NetBIOS domain name of the Active Directory where the service account resides. | This is the login name for the SharePoint Application Pool account. |     **5. Enable Constrained Delegation**  If this were SharePoint 2007, we’d be done.  But SharePoint 2010 requires **Constrained Delegation**.  In order to enable constrained delegation you have to connect to the Domain Controller and enable Delegation on the account used to host the SharePoint Web Application Pool.  Remote Desktop into the Domain Controller, open **Active Directory Users and Computers**, then locate the SharePoint Web Application Pool account.  Double-click on the account and locate the **Delegation** tab:  Step_by_Step_Kerberos_Authentication_for_SharePoint_2010  **Figure 7: Delegation**  Pick **Trust this user for delegation to any service** and click OK**.**  SharePoint will now authenticate clients using Kerberos authentication to [**http://intranet.contoso.local**](http://intranet.contoso.local/)  **Workarounds**  A common work-around to the Real-World Scenario above, when Kerberos authentication is not involved, is a proxy account: hard-code the Report Server credentials into the Report itself.  When the user accesses the Report, SharePoint connects to SQL using the stored credentials.  This is also what the Secure Store service does.  This is also a form a delegation, but does not pass the user’s actual credentials to the data store: it uses a proxy account.  Thus, all users get the same rights on the data store and the password is saved in clear-text in the Report’s connection string.  If this doesn’t meet your requirements, you need to call in Kerberos to handle the connection.  **Looking Ahead**  Even though Kerberos is not always needed, or possible like with extranets, the introduction of External Content Types in SharePoint 2010 as a reporting tool will greatly increase the need for it.  The increased maturity and new features in PerformancePoint, PowerPivot, and Reporting Services in SharePoint mode, if your data is not on the SharePoint server itself you will need to use delegation.  The best choice which provides the lowest maintenance overhead, the highest level of security, and the lowest processor overhead, is Kerberos authentication.  Try it out in a VM farm on your local computer.  It’s a great tool to have in your SharePoint architect’s toolbox.  **By: Christian Holslin, Solutions Architect, Gig Werks** | | |

# Configuring Kerberos in SharePoint 2010

When installing SharePoint for the first time, there is an option to use either NTLM or Kerberos.  Kerberos is recommended, but the caveat that they give you is that additional steps need to be taken by an administrator to make it work.

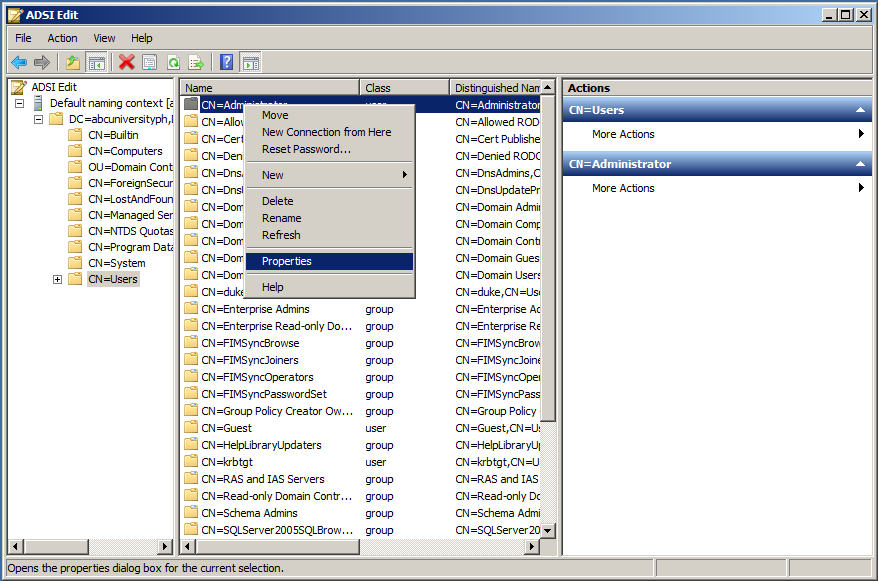
On older server versions (Windows Server 2003 R2, for instance) you could pick Kerberos from the get-go and continue setting everything up as long as you were logged in as an Administrator.  Later on you’d find that nobody else could log in until an administrator set up the SPNs, and at that time you’d be setting them up (typically via command line).

If you’re installing SharePoint 2010 onto Server 2008 R2, though, Central Administration won’t even load until those same SPNs are set up.  This post is intended as a quick walkthrough of how to do it if what I just said made no sense.

In our case (for our test environment) we’re using the domain abcuniversityph.edu (does not exist – we just use this sample for class), have gone through the steps to install SharePoint, and have told it to use Kerberos for authentication.  We then tried to load Central Administration and it wouldn’t allow us to log in.

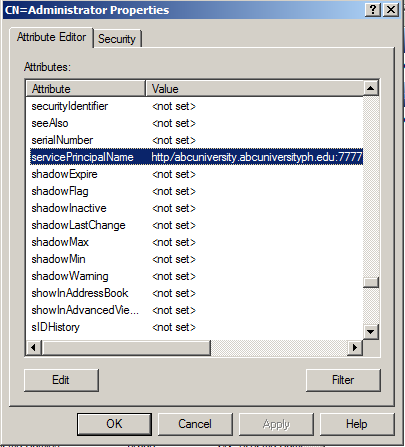
Our next step is to use the ADSI Edit utility, which can be launched by typing **adsiedit.msc** in the search/run textbox from the start menu.

When that comes up, we need to locate the Administrator Container.

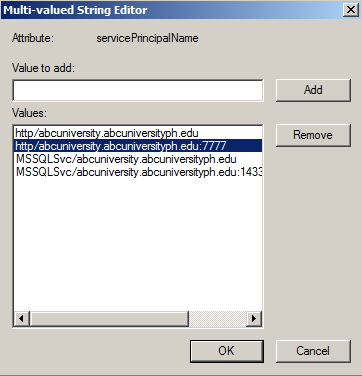
[](http://www.sharepointrating.com/wp-content/uploads/2010/08/ADSIEdit2.png)

As you can see, we had to go to our domain, then choose the Users Container, and found Administrator in there.

At this point we’ll right click on the Administrator container, and choose Properties.  From there, we locate the servicePrincipalName property and edit it.

[](http://www.sharepointrating.com/wp-content/uploads/2010/08/ADSIEdit3.png)

For our example, the necessary line to add (just for Central Administration) was the **http/abcuniversity.abcuniversity.edu:7777** line, as our Central Administration Web Application happens to be running on the lucky port 7777.  Note that it takes the form of **protocol/fully qualified computer name:port**. We also added a line for the computer name on port 80 (by leaving the port off) so that the demo web applications that we create are also accessible.

[](http://www.sharepointrating.com/wp-content/uploads/2010/08/ADSIEdit4.png)  
At this time it’s enough to “OK” our way out of the utility, saving our changes along the way, and continue to set up the environment in your now more-secure environment.

**Configuring Kerberos**

May 28, 2010 [Shannon Bray](http://shannonbray.wordpress.com/author/shannonbray/) [Leave a comment](http://shannonbray.wordpress.com/2010/05/28/configuring-kerberos/#respond) [Go to comments](http://shannonbray.wordpress.com/2010/05/28/configuring-kerberos/#comments)

i

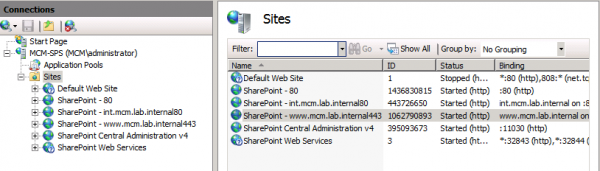
5 Votes

Quantcast

In this session, we will now configure Kerberos on our farms so that we can build closer to the completion of the series. If you are unsure where to start in the process, please refer back to one of the previous posts:

* [SharePoint 2010 Machine Configurations](http://shannonbray.wordpress.com/2010/05/24/sharepoint-2010-machine-configurations/)
* [Laptop Configurations](http://shannonbray.wordpress.com/2010/05/28/laptop-configurations/)
* [Configuring Windows Server 2008 for Virtualization](http://shannonbray.wordpress.com/2010/05/25/configuring-windows-server-2008-for-virtualization/)
* [Configuring RRAS for Windows Server 2008 R2](http://shannonbray.wordpress.com/2010/05/25/configuring-rras-for-windows-server-2008-r2/)
* [Building Your SharePoint 2010 Infrastructure](http://shannonbray.wordpress.com/2010/05/26/building-your-sharepoint-2010-infrastructure/)
* [Creating Your Web Application](http://shannonbray.wordpress.com/2010/05/26/creating-your-web-application/)

In order to follow the following directions, you will need to create a web application and extend it. Please refer back to ‘[Creating Your Web Application](http://shannonbray.wordpress.com/2010/05/26/creating-your-web-application/)‘ for more details. Navigate over to your MCM-SPS1 image and launch the Internet Information Services (IIS) Manager. Click on the ‘Sites’ in the left pane so that is displays a list of the sites in IIS. Make note of the site id for ‘SharePoint – www.mcm.lab.internal443′. The one in this diagram is ’1062790893′.

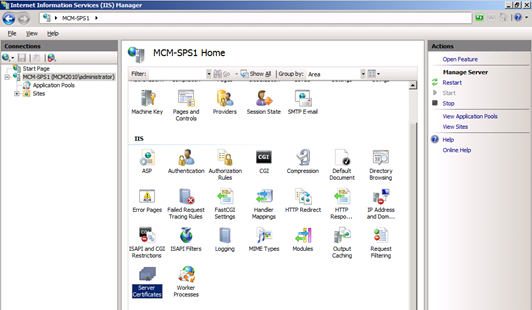


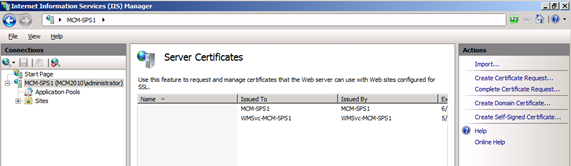
  You will need to download a copy of SelfSSL and execute it on MCM-SPS1.

Notice the syntax: SelfSSL.exe /S: 1062790893. 🡨 IIS ID for the web site!!!!

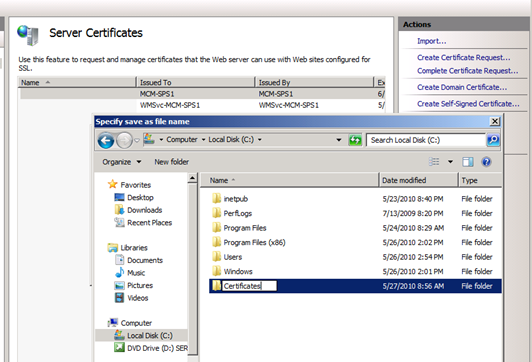


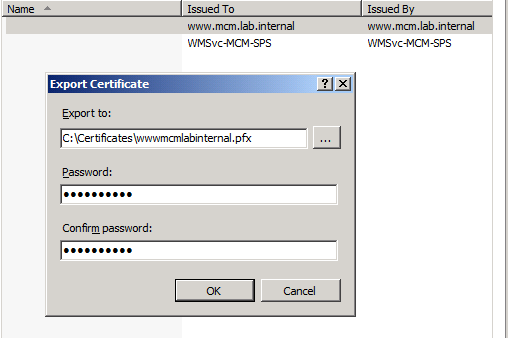
You will see a question asking, “Do you want to replace the SSL setting for site 1062790893 (Y/N)?”. Type “Y”. You should see a success message. We will now open Internet Information Services (IIS) Manager and highlight ‘MCM-SPS1′ in the left node, and then select ‘Server Certificates’ in the center pane.



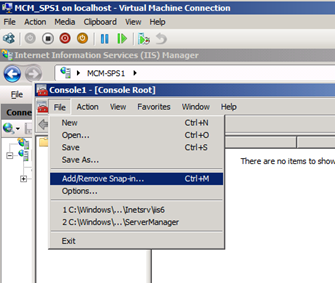


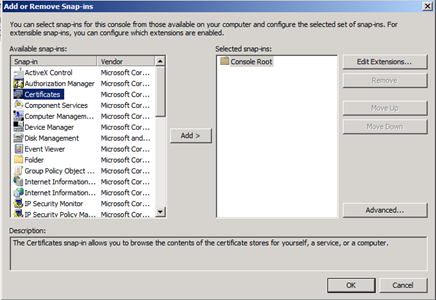
  We now need to export the certificate. We will have a few certificates along the way, so I like to create a centralized location to keep them in one place. I will create a ‘Certificates’ folder on the C drive of MCM-SPS. The next couple of articles will use this same location.

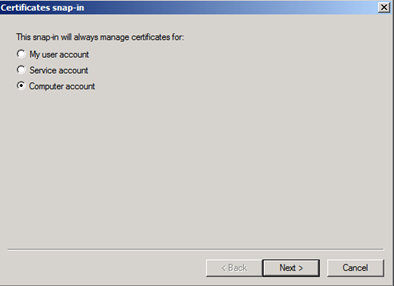


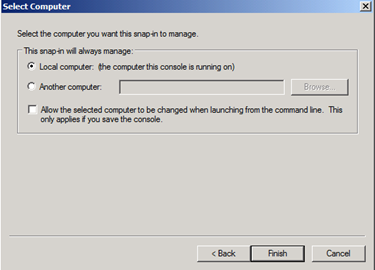


Click “Start” and in the search box, type “MMC” and hit enter. From here, click on ‘File’ and select ‘Add/Remove Snap-in…’. This will launch the wizard to select the certification snap-in. We will be referring back to this several times throughout the process. Follow the images below to complete this task.

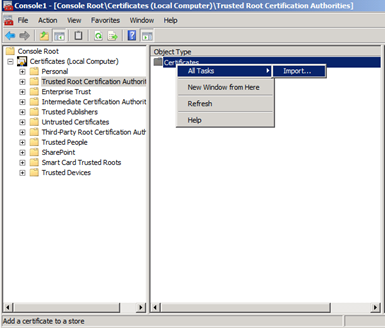


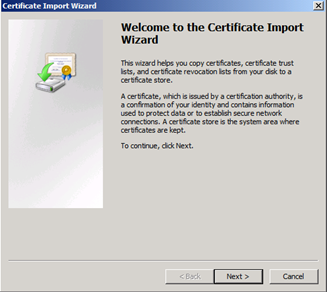


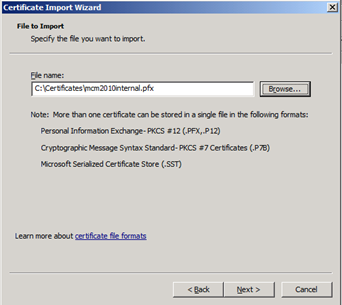




To import the certificate, make sure you click on the ‘Trusted Root Certification Authority’ folder. You will then be able to right click, select ‘All Tasks’ and then ‘Import…’.







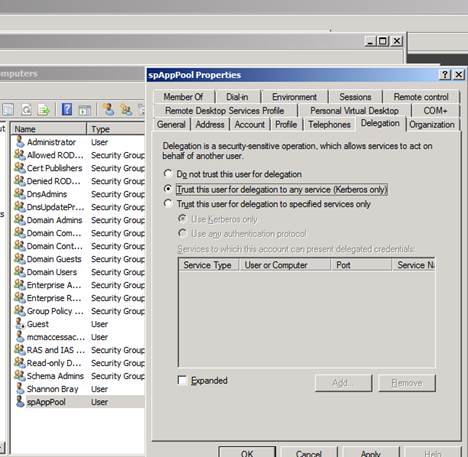
Now that we have completed the certificates for this section, we will bounce back to the Internet Information Services (IIS) Manager. Click on ‘Application Pools’. This will help us identify the application pool we are using for your application. It should be similar to the one I have here: SharePoint – int.mcm.lab.internal80.

Launch a command prompt. Here we will set the spn. Note that even though our site is using HTTPS, we will still use http. You syntax should look something like Setspn –A http {site} {app pool account}:

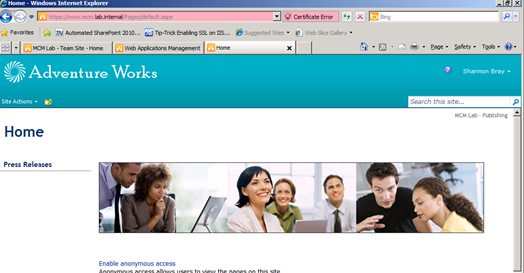
    Setspn –A http/www.mcm.lab.internal mcm\spAppPool

At this point, Kerberos is officially set up. All we really need is the SPN set up. We will now set up delegation which will enable us to pass our credentials forward. This may be needed for advanced configurations.

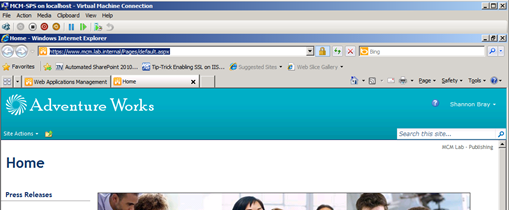
  After we set the spn, we now need to bring up our domain controller. For the sake of this demo, we will refer to it as MCM-DC. We will now modify the user account that we use for our SharePoint application pool: spAppPool. Click on properties, find the ‘Delegation’ tab, and select ‘Trust this user for delegation to any service (Kerberos only).



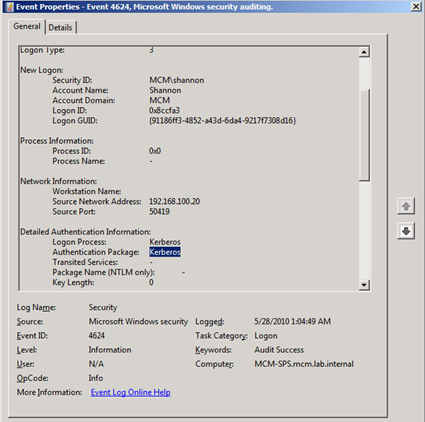
  You can now test Kerberos. Type in your HTTPS address. If you get a certificate error similar to the one below, you have made a mistake in the process. Here I typed HTTPS instead of HTTP when I set my SPN.



  When done correctly, everything should appear normal.



  We have one final test we need to verify. Open up the event properties and verify that the Authentication Package is set to Kerberos instead of NTML. If not, you will need to review the previous steps and look for typos.



ADVERTISEMENT

Categories: [Authentication](http://en.wordpress.com/tag/authentication/), [Configuration](http://en.wordpress.com/tag/configuration/)

[Like](http://shannonbray.wordpress.com/2010/05/28/configuring-kerberos/?like=1&_wpnonce=e8f219d372)

One blogger likes this post.

[Ranjeet Bhargava](http://gravatar.com/ranjeetbhargava)

[Comments (4)](javascript:void(0);) [Trackbacks (8)](javascript:void(0);) [Leave a comment](http://shannonbray.wordpress.com/2010/05/28/configuring-kerberos/#respond) [Trackback](http://shannonbray.wordpress.com/2010/05/28/configuring-kerberos/trackback/)

1. http://1.gravatar.com/avatar/fe7a3a9491eeeaffd1c6469a1e9aaac0?s=32&d=identicon&r=G

Brad Thurber

June 16, 2010 at 6:39 am | [#1](http://shannonbray.wordpress.com/2010/05/28/configuring-kerberos/#comment-85)

[Reply](http://shannonbray.wordpress.com/2010/05/28/configuring-kerberos/?replytocom=85#respond) | [Quote](javascript:void(0);)

Possible error.

Make note of the site id for ‘SharePoint – <http://www.mcm.lab.internal443′>. The one in this diagram is ’1361630023′.

Should this be ’1062790893′ instead?

* + http://0.gravatar.com/avatar/e02bdad8a5dd9f66a9d6ebcbb4330097?s=32&d=identicon&r=G

[Shannon Bray](http://shannonbray.wordpress.com)

June 16, 2010 at 10:04 am | [#2](http://shannonbray.wordpress.com/2010/05/28/configuring-kerberos/#comment-87)

[Reply](http://shannonbray.wordpress.com/2010/05/28/configuring-kerberos/?replytocom=87#respond) | [Quote](javascript:void(0);)

Hello Brad,

You are correct. I missed this edit. I will correct it now. Thank you for your comments!!! :)

Shannon

1. http://0.gravatar.com/avatar/26ca092146993a841fc874838482c470?s=32&d=identicon&r=G

[spence](http://www.harbar.net)

July 8, 2010 at 1:44 pm | [#3](http://shannonbray.wordpress.com/2010/05/28/configuring-kerberos/#comment-103)

[Reply](http://shannonbray.wordpress.com/2010/05/28/configuring-kerberos/?replytocom=103#respond) | [Quote](javascript:void(0);)

you don’t need to modify apphost.config and delegation is not required

* + http://0.gravatar.com/avatar/e02bdad8a5dd9f66a9d6ebcbb4330097?s=32&d=identicon&r=G

[Shannon Bray](http://shannonbray.wordpress.com)

July 8, 2010 at 2:28 pm | [#4](http://shannonbray.wordpress.com/2010/05/28/configuring-kerberos/#comment-104)

[Reply](http://shannonbray.wordpress.com/2010/05/28/configuring-kerberos/?replytocom=104#respond) | [Quote](javascript:void(0);)

Thanks Spence. I will make that correction.