My Kerberos Checklist

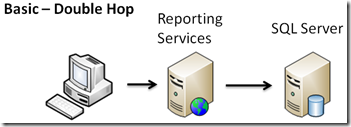
<http://blogs.msdn.com/b/psssql/archive/2010/06/23/my-kerberos-checklist.aspx>

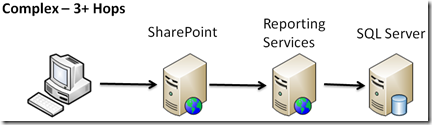
[http://blogs.msdn.com/resized-image.ashx/__size/32x32/__key/CommunityServer-Components-Avatars/00-00-28-00-92/4TFN0NLD9QI9.jpgAdam W. Saxton](http://blogs.msdn.com/members/awsaxton/)

23 Jun 2010 3:49 PM

I’ve had numerous questions regarding Kerberos, both internally within Microsoft and with Customers.  It continues to be a complicated topic and the documentation that is out there can be less than straight forward.  Based on some previous items I’ve worked on, I wanted to share my experience in regards

Let me start by looking at two scenarios for reference.  One that is basic and the other that is complex.

[](http://blogs.msdn.com/cfs-file.ashx/__key/CommunityServer-Blogs-Components-WeblogFiles/00-00-00-73-84-metablogapi/0624.image_5F00_1190B424.png)

[](http://blogs.msdn.com/cfs-file.ashx/__key/CommunityServer-Blogs-Components-WeblogFiles/00-00-00-73-84-metablogapi/0647.image_5F00_0F078266.png)

As you’ll find, once we figure out how to configure the basic scenario, the complex scenario ends up being very similar.

**Data Collection:**

The first thing when you try to tackle a Kerberos issue is to understand your environment.  I find that a lot of the Kerberos issues that I troubleshoot all come down to gathering the right information to make an informed analysis and identify the problem point.  The following data points relate to all servers involved.  We will circle back on the Client after we talk about the Servers.

1. Know your topology
2. What is the Service Account being used for the application in question?
3. What Service Principal Name (SPN) does your app require?
4. What SPNs are defined for that service account?
5. What are the delegation settings for the service account?
6. Local Policy related information
7. Additional application specific information

Consistent vs. Intermittent Kerberos Issues

The data collection points above should allow you to get Kerberos working in most cases.  I say most cases because the above refers specifically to configuration.  I typically break it down to consistent vs. intermittent issues.  If the issue is reproducable every time, it is a configuration issue.  If it is intermittent, then it is usually not a configuration issue.  If it was it would happen all the time.  Intermittent means it works most of the time.  In order to work at all, it has to be configured correctly.  The exception to this would be if you are in a Farm type situation and the configuration is not the same on every box in the farm.  Sometimes you may hit Server A which is configured properly, and another time you may hit Server B which is not and causes an error.  Which brings us to the first Data Collection Point…

Know your topology

Before you being, you should know what servers are involved in your application as a whole.  If we are talking about a single web application, you probably have at least two servers to consider and know about – the Web Server and the Backend (SQL for our purposes).  They both play a part.  This becomes even more important in a distributed environment where you may have 3+ servers.

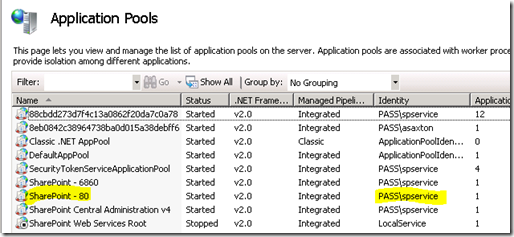
As you’ll see, with the data collection items, we basically will walk the line down your servers to check them one by one.

What is the Service Account?

For the particular server you are looking at, what is the service account that the application is using?  This is important, because this will tell us where the SPN needs to go.  It also plays a part in Delegation.  Not every service will be a Windows Service, so this could be dependent on the application in question.  Here are some examples:

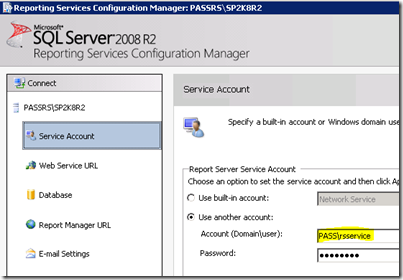
SharePoint

IIS – not a windows service

[](http://blogs.msdn.com/cfs-file.ashx/__key/CommunityServer-Blogs-Components-WeblogFiles/00-00-00-73-84-metablogapi/0647.image_5F00_7B7A22C4.png)

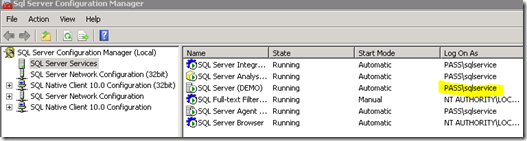
Reporting Services

Windows Service

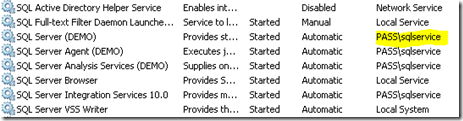
[](http://blogs.msdn.com/cfs-file.ashx/__key/CommunityServer-Blogs-Components-WeblogFiles/00-00-00-73-84-metablogapi/8463.image_5F00_20976D31.png)

SQL Server

Windows Service

[](http://blogs.msdn.com/cfs-file.ashx/__key/CommunityServer-Blogs-Components-WeblogFiles/00-00-00-73-84-metablogapi/6825.image_5F00_6A464521.png)

For windows services, you can also look in the Services MMC to get the information.  Again, you need to know what your application is doing:

[](http://blogs.msdn.com/cfs-file.ashx/__key/CommunityServer-Blogs-Components-WeblogFiles/00-00-00-73-84-metablogapi/5758.image_5F00_48E69F85.png)

What SPN does your app require?

We can look at all sorts of SPN listings, but before you do, we need to know what it is we are looking for.  I think this is one of the more complicated parts of Kerb configuration because the SPN is dependent on the application you are using.  The format of the SPN is consistent between applications, but what is required is dependent on the application, or from an SPN point of view, the service.  It is a Service Principal Name after all!

The SPN has the following format:  <service>/<host>:<port/name>

The port/name piece of this is optional and dependent on what the service will accept.

HTTP – For a default configuration, the port is never used for an HTTP SPN.  SPN’s are unique and if you add an HTTP SPN with a port on it, it will be ignored as it is not correct.  IIS and Internet Explorer do not affix the port number to the SPN request when they look for it.  From an Internet Explorer perspective, you can alter this behavior via a registry key to where it will, but I have yet to see anyone do that.  Most people aren’t aware of it from what I can tell.  From my experience, I would stay away from adding a port to an HTTP SPN.

MSSQLSvc – you can look at the following blog post to read more about how SQL determines the SPN needed.  <http://blogs.msdn.com/b/psssql/archive/2010/03/09/what-spn-do-i-use-and-how-does-it-get-there.aspx>

For the next couple of items, we will use the SharePoint service as the example – spservice.  In this case it is a web application, so we know it will use the HTTP service from an SPN perspective.  The host piece is dependent on how we are connecting to the web server.  This is true for any application really.  From an HTTP perspective it is the URL, for SQL it is the connection string.  Another thing to know is that both IIS and SQL will resolve a NetBIOS name to the Fully Qualified Domain Name if it can.  For example – <http://passp> will be resolved to passsp.pass.local.

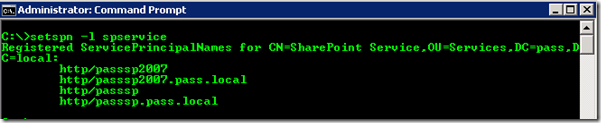
For our spservice example with a url of <http://passsp>, our SPN turns out to be http/passsp.pass.local and it is placed on the spservice account.

Another special note about HTTP SPNs.  If for example my SharePoint AppPool (service) was using Network Service, this is considered the machine context so the SPN would go on the machine account (PASSSP).  However, HTTP is considered a covered service for a special service type called HOST.  Every Machine account has a HOST entry for the FQDN as well as the NetBIOS name.  You don’t need to add an HTTP SPN on the machine account as long as your URL matches the machine name.

When adding an SPN, I also always recommend that you add both the FQDN SPN (i.e. http/passsp.pass.local) as well as the NetBIOS SPN (i.e. http/passsp).  The NetBIOS SPN is a safety measure in case the DNS resolution fails and it just submits the NetBIOS SPN request.

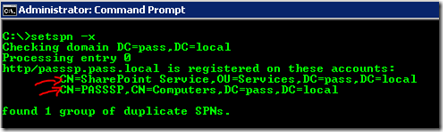
What SPN is defined?

Now that we know the service account and what our SPN should be, we can look at the SPNs that are defined on that account.  We can use SetSPN to do this, although there are other tools that can help get this information for you (ADSIEdit, LDAP queries, etc…).  SetSPN is nice though as it ships with the Operating System starting with Windows 2008.  Lets have a look at our SharePoint Service account – spservice:

[](http://blogs.msdn.com/cfs-file.ashx/__key/CommunityServer-Blogs-Components-WeblogFiles/00-00-00-73-84-metablogapi/8883.image_5F00_67BD1363.png)

Based on what we came up with above, we can see that the passsp SPN’s are in place.  You’ll also notice another SPN present, which means this Service Account is hosting two HTTP Services (could be two AppPools on the one server, or on two separate servers).

You could run into a situation where the SPN is defined on another account as well.  This may be a misplaced or a duplicate SPN.  Both will cause an issue for you.  Usually when I grab SPN information from an environment, I grab all SPN’s defined in the Domain so that I can look for misplaced or duplicate SPNs.  The SetSPN tool that comes with Windows 2008 and later (and can be downloaded for Windows 2003), contains a new switch that will look for Duplicates for you.  It is the –X switch.

[](http://blogs.msdn.com/cfs-file.ashx/__key/CommunityServer-Blogs-Components-WeblogFiles/00-00-00-73-84-metablogapi/0576.image_5F00_3F3E314F.png)

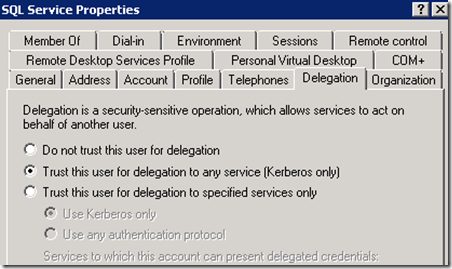
In the above, you can see two accounts that had the http/passsp.pass.local SPN.  You can then decide which one really needs to be there based on the Service Account being used.

What are the delegation settings?

Delegation only comes into play if you want the Client’s Windows credentials forwarded to another service.  For example, SharePoint to Reporting Services, Reporting Services to SQL, or even SQL to SQL in a Linked Server scenario.  NTLM does not allow for the forwarding of credentials.  This is accomplished through the process of delegation as part of the Kerberos Protocol. There are two main types of Delegation – Full Trust or Constrained Delegation.  Of note, you will not see the Delegation Tab on the Account within Active Directory unless an SPN has been assigned to that account.

Full Trust

This means that the given service can forward the Client’s credentials to any service.  You are non-discriminate in who you communicate to.  This is less secure option out of the two, but it is the easiest to configure out of the two (which I would expect being less secure – Secure always means complicated right?)

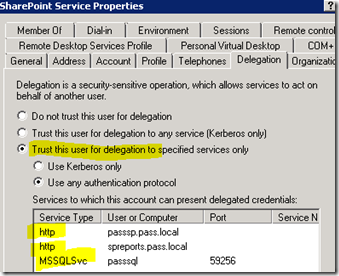
[](http://blogs.msdn.com/cfs-file.ashx/__key/CommunityServer-Blogs-Components-WeblogFiles/00-00-00-73-84-metablogapi/1072.image_5F00_3DF99870.png)

Constrained Delegation

Constrained means that you are going to specify which services you can actually delegate to.  The services are represented by SPN’s.  This is the more secure approach but has some drawbacks.  As mentioned before it is more complicated. The reason is that you have to know exactly what your application is trying to delegate to.  It may not be just the service you are interested.  For example, you may be configuring SharePoint for Delegation to go to Reporting Services, but then realize that you just broke a connection to SQL or maybe a connection to some web service that you are trying to hit that requires Kerberos.  It’s not really that bad as long as you understand everything that your application is going to reach out to and that would require passing on the Client’s credentials.

The other drawback to Constrained Delegation is that you lose the ability to cross a domain boundary.  Meaning a cross domain scenario will fail from a delegation perspective.  Users from another Domain can hit your application, but all of the services that you are communicating to need to be in the same domain.  For example, SharePoint (Domain A) cannot delegate to SQL (Domain B).  Under constrained delegation, that will fail.

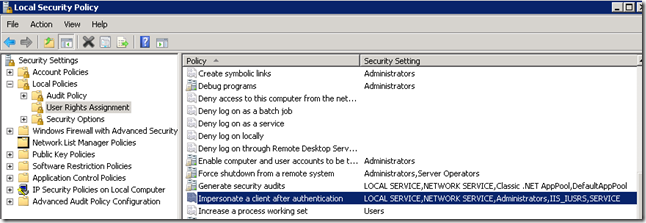
In the image below, the 3rd radio dial means that you want to use Constrained Delegation.  The sub radio dials define whether you want to use all Kerberos, or if you want to enable Protocol Transitioning.  I’m not going to get into Protocol Transitioning in this blog post as it is big enough, but you will have to deal with Protocol Transitioning if you are using the Claims to Windows Token Service.  This would come into the picture if you are doing anything with Excel Services in SharePoint or PowerPivot.

[](http://blogs.msdn.com/cfs-file.ashx/__key/CommunityServer-Blogs-Components-WeblogFiles/00-00-00-73-84-metablogapi/6862.image_5F00_2A6C38CF.png)

You will need to go back to your application’s topology to determine if enabling delegation is required. If we look at our Double Hop example from above, Reporting Services would need to have delegation enabled for it’s service account, but SQL would not as SQL isn’t going out to anything using the Client’s credentials.

Local Policy Settings

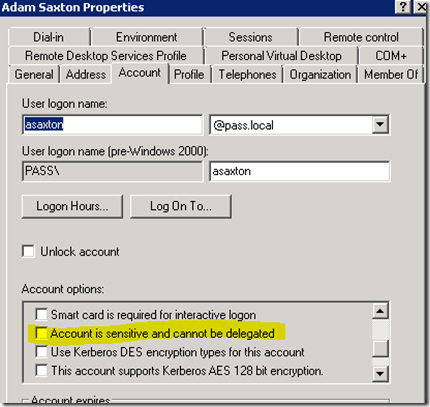
There is at least one Local Policy setting you’ll need to pay attention to when trying to delegate.  That is the “Impersonate a client after authentication” policy.

[](http://blogs.msdn.com/cfs-file.ashx/__key/CommunityServer-Blogs-Components-WeblogFiles/00-00-00-73-84-metablogapi/5488.image_5F00_7657997B.png)

If your middle server is a web server, you can take advantage of a build in group that has this permission.  For Windows 2003, the group is called IIS\_WPG.  For Windows 2008 and later it is the IIS\_USRS group.  By default, SharePoint and RS should place itself in that group.  So, you usually don’t have to worry about it.  I’m just mentioning it here as a step in the checklist.  I rarely see this as the issue though unless you are doing a customer application with a Domain User account for the service account.

Client

Let’s circle back on the Client.  You may be asking, all this is great for the application, but is there anything special I need to do for the User Account coming from the client.  Not really.  By default you should be good to go from the Client’s user account.  However, there is an account you should be aware of within Active Directory.  That is the “Account is sensitive and cannot be delegated” setting.  If that is checked, you will have issues with that specific user.  To this date, I have yet to see a customer actually have that checked.  Doesn’t mean people don’t do it.  I just haven’t seen it.

[](http://blogs.msdn.com/cfs-file.ashx/__key/CommunityServer-Blogs-Components-WeblogFiles/00-00-00-73-84-metablogapi/8831.image_5F00_18CC2837.png)

Application Specific Settings

When I started getting into Kerberos, I found that almost all of the issues were based on the Active Directory settings (SPN, Delegation, etc…).  Not to say that that has lessened, but I’ve also seen a shift in the complexity of getting specific applications up and running.  As applications become more complex, you should be aware of what settings may come into play within that app that could affect Kerberos.  If you have gone through everything above and it all looks good.  Chances are that there is an application specific setting that is interfering.

There is a lot to mention in this area, so I will spin up another blog post to discuss application specific settings to touch on IIS, SharePoint, Excel Services, PowerPivot and Reporting Services.  SQL doesn’t really have any Kerb specific settings as long as the SPN and delegation settings (if needed) are in place.

Tying it together…

So, we’ve looked at what my checklist is, but it was really focused on one service. What I’ve found is that it is as simple as that.  All I do is repeat the check list on each server that play a part in the application (topology).  Think of it as a wash, rinse, repeat.  When I help customers to get Kerberos configured, I just walk the line down each server to make sure everything lines up.  I have been fairly successful with that approach.  As I’ve had more experience with it (as I usually deal with it every day), I can usually target a specific segment depending on where the error is coming from.  Other times it may not be that straight forward.  Even when I target a specific area, if that doesn’t pan out, I just start from the beginning and apply the checklist to each server/service that is playing a part.

Once you approach it that way, it really doesn’t matter how many hops there are or what services are involved.  You just follow the checklist one more time.  The point where complications usually come into play are when Constrained Delegation is implemented and we didn’t account for everything or you hit up against an App Specific issue.  Outside of that, it is usually straight forward based on the above.  Just find out what the SPN needs to be and where it needs to go and you are 80% there.

I realize I’m making it sound simple when it can be very frustrating and complicated, but the above has worked well for me in the past. Hopefully the above is helpful to you as you try to implement Kerberos within your environment.

There is definitely way more to cover on this topic and I will continue to blog about those items.

Adam W. Saxton | Microsoft SQL Server Escalation Services   
<http://twitter.com/awsaxton>