Some letters get all the love. Take ‘C’ for example. You’ve got collation, clustering, compression, constraints, change data capture… Colleen. And what about ‘A’? Atomnicity, authentication, automation, auditing, Azure, and so on. But not ‘K’. What does poor little ‘K’ get? Kill? Keywords? Meh.

So today I’m talking about Kerberos. I’ll tell you a little about what it is, how to use it with SQL Server, and why you use it.

**What is Kerberos?**

Kerberos is a highly secure network authentication protocol that is used to authenticate clients and servers (called security principals) on a network. It’s a preferred alternative to NTLM. Authentication is based upon encrypted session keys and authentication tokens which are issued by the Key Distribution Center (KDC). Typically the KDC is a domain controller using Active Directory. For this reason, all parties involved in Kerberos authentication must be part of the same domain or trusted domains.

**How does it work?**

When a client logs into the network, it contacts the KDC and presents its password to prove it is who it claims to be. The KDC then issues the client a ticket (TGT) valid for a certain amount of time. If the client wants to access a service on the network it contacts the KDC again with its ticket and the name of the service it wants access to (using the SPN, discussed below). The KDC will then issue the client a service ticket to be presented to the service in question. The service only needs to check out the service ticket to know that the client has already been authenticated on the network.

**Authentication, not Authorization**

A key principal (get it? “key” “principal”) to keep in mind is that Kerberos provides authentication, not authorization. This means that it only confirms that the client is who they claim to be. It does not validate whether the client has permission to access a particular service. That still needs to be handled by the target service.

**Kerberos and SQL Server**

So what does all this have to do with SQL Server? Well, as I stated earlier, Kerberos is an alternative to NTLM, so if you’re using Windows authentication you’d use Kerberos indirectly.

Another benefit to using Kerberos, besides security, is a little thing called a “double hop.” Let’s say you’re running Reporting Services and you want to use integrated Windows authentication. But you also want the report to connect to the database as that same Windows user. The connection from the client to Reporting Services is one “hop”. The connection from Reporting Services to the SQL Server instance is a second “hop”. NTLM doesn’t support double hops. Kerberos, however, does.

**Configuring SQL Server to use Kerberos**

In order for a SQL Server instance to use Kerberos, it has to register its Service Principal Name (SPN) with the Kerberos server. The SPN is a unique mapping between the service name and the Windows account that started the service. SQL Server will automatically try to register its SPN with Active Directory on startup. This SPN is what clients will use to specify which service threy want access to. If for some reason it can’t register the SPN, for example if the service account doesn’t have the required permissions, SQL Server will use NTLM instead. So you don’t have to worry about SQL Server not starting because of that.

A login requires Domain Admin rights to register an SPN, so if the service account isn’t a Domain Admin (and it shouldn’t be), SQL Server won’t be able to register the SPN itself. A Domain Admin can manually register the SPN, however, using the setspn utility.  
To determine whether or not your SQL Server instance is currently using Kerberos, you can check the Error Log for the instance. If SQL Server was unable to register an SPN, you’ll see a message like the following on startup:

The SQL Network Interface library could not register the Service Principal Name (SPN) for the SQL Server service. Error: 0x2098, state: 15. Failure to register an SPN may cause integrated authentication to fall back to NTLM instead of Kerberos. This is an informational message. Further action is only required if Kerberos authentication is required by authentication policies.

Alternatively, if you don’t feel like looking through the log, you can also use the following query to determine what authentication protocol you’re using.

|  |  |
| --- | --- |
| 1 | select auth\_scheme from sys.dm\_exec\_connections where session\_id=@@spid |

**Further Reading**

For more information on Kerberos, check out the following links:

[Using Kerberos Authtication with SQL Server](http://technet.microsoft.com/en-us/library/cc280745.aspx)

[How to: Enable Kerberos Authentication on a SQL Server Failover Cluster](http://msdn.microsoft.com/en-us/library/ms189585%28v=sql.90%29.aspx)

