TDV Integration with Kerberos

<table>
<thead>
<tr>
<th>Project Name</th>
<th>TIBCO Data Virtualization</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release</td>
<td>V1.4</td>
</tr>
<tr>
<td>Date</td>
<td>January 23, 2018</td>
</tr>
<tr>
<td>Primary Authors</td>
<td>Sam Carroll</td>
</tr>
<tr>
<td></td>
<td>Sergei Sternin</td>
</tr>
<tr>
<td></td>
<td>Ravikiran Thummalapenta</td>
</tr>
<tr>
<td>Document Owner</td>
<td>TIBCO Software</td>
</tr>
<tr>
<td>Purpose</td>
<td>TDV Integration with Kerberos</td>
</tr>
</tbody>
</table>
Revision History

<table>
<thead>
<tr>
<th>Version</th>
<th>Date</th>
<th>Author</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>V1.1</td>
<td>October 31, 2016</td>
<td>Sam Carroll, Sergei Sternin, Ravikiran Thummalapenta</td>
<td>Initial Draft</td>
</tr>
<tr>
<td>V1.2</td>
<td>January 31, 2017</td>
<td>Sergei Sternin</td>
<td>Added more troubleshooting tips and information about UAC affecting TGT</td>
</tr>
<tr>
<td>V1.3</td>
<td>February 15, 2017</td>
<td>Sam Carroll</td>
<td>Fixed klist typo</td>
</tr>
<tr>
<td>V1.4</td>
<td>January 23, 2018</td>
<td>Venky Madireddi</td>
<td>Applied TIBCO branding</td>
</tr>
</tbody>
</table>

Distribution

This document has been distributed to:

<table>
<thead>
<tr>
<th>Name</th>
<th>Title</th>
<th>Company</th>
<th>Date of Issue</th>
<th>Version</th>
</tr>
</thead>
</table>

Related Documents

This document requires the following approvals. Signed approval forms are filed in the project files.

<table>
<thead>
<tr>
<th>Document</th>
<th>Date</th>
<th>Author</th>
</tr>
</thead>
<tbody>
<tr>
<td>TDV Administration Guide</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TDV User Guide</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Copyright Notice
COPYRIGHT© 2018 TIBCO Software Inc. This document is unpublished and the foregoing notice is affixed to protect TIBCO Software Inc. in the event of inadvertent publication. All rights reserved. No part of this document may be reproduced in any form, including photocopying or transmission electronically to any computer, without prior written consent of TIBCO Software Inc. The information contained in this document is confidential and proprietary to TIBCO Software Inc. and may not be used or disclosed except as expressly authorized in writing by TIBCO Software Inc. Copyright protection includes material generated from our software programs displayed on the screen, such as icons, screen displays, and the like.

Trademarks
All brand and product names are trademarks or registered trademarks of their respective holders and are hereby acknowledged. Technologies described herein are either covered by existing patents or patent applications are in progress.

Confidentiality
The information in this document is subject to change without notice. This document contains information that is confidential and proprietary to TIBCO Software Inc. and its affiliates and may not be copied, published, or disclosed to others, or used for any purposes other than review, without written authorization of an officer of TIBCO Software Inc. Submission of this document does not represent a commitment to implement any portion of this specification in the products of the submitters.

Content Warranty
The information in this document is subject to change without notice. THIS DOCUMENT IS PROVIDED "AS IS" AND TIBCO MAKES NO WARRANTY, EXPRESS, IMPLIED, OR STATUTORY, INCLUDING BUT NOT LIMITED TO ALL WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. TIBCO Software Inc. shall not be liable for errors contained herein or for incidental or consequential damages in connection with the furnishing, performance or use of this material.

Export
This document and related technical data, are subject to U.S. export control laws, including without limitation the U.S. Export Administration Act and its associated regulations, and may be subject to export or import regulations of other countries. You agree not to export or re-export this document in any form in violation of the applicable export or import laws of the United States or any foreign jurisdiction.

For more information, please contact:

TIBCO Software Inc.
3303 Hillview Avenue
Palo Alto, CA 94304
USA
Table of Contents

1 INTRODUCTION .................................................................................................................. 6
  1.1 Purpose ................................................................................................................................. 6
  1.2 Audience ............................................................................................................................... 6
  1.3 Kerberos Overview ................................................................................................................ 6
  1.4 Brief History .......................................................................................................................... 6
  1.5 Realm .................................................................................................................................. 6
  1.6 Principals ............................................................................................................................... 7
  1.7 Client Principals ..................................................................................................................... 7
  1.8 Service Principals .................................................................................................................. 7
  1.9 Keytabs ................................................................................................................................ 7
  1.10 krb5.conf or krb5.ini .......................................................................................................... 8
  1.11 Tickets ................................................................................................................................ 8
  1.12 Getting Tickets .................................................................................................................... 8
  1.13 Credential/Ticket Cache ...................................................................................................... 9
  1.14 How it Works ....................................................................................................................... 9
  1.15 SSPI versus JGSS mode in TDV configuration ................................................................... 9

2 Kerberos SSO Configuration in TDV – Prerequisites ......................................................... 10
  2.1 Active Directory Configuration Prerequisites ...................................................................... 10
  2.2 TDV Configuration Prerequisites ....................................................................................... 10

3 Kerberos SSO Configuration for TDV on Linux ................................................................. 12
  3.1 Active Directory Configuration ............................................................................................. 12
  3.2 TDV Server Configuration ................................................................................................... 13
  3.3 Configure TDV Settings via Studio ...................................................................................... 14
  3.4 Testing & Validation ............................................................................................................. 15

4 Kerberos SSO Configuration for TDV on Windows ......................................................... 17
  4.1 Configuration steps to use TDV in SSPI mode ..................................................................... 17
    4.1.1 Active Directory configuration to use TDV in SSPI mode ............................................. 17
    4.1.2 TDV Configuration for SSPI mode ............................................................................... 17
  4.2 Configuration steps to switch using TDV from SSPI to JGSS mode ..................................... 22
    4.2.1 TDV Server Configuration for JGSS mode ................................................................. 23
  4.3 Testing & Validation ............................................................................................................. 25

5 Kerberos SSO Configuration for TDV Studio ................................................................. 26
  5.1 Configuring Studio in SSPI mode ......................................................................................... 28
  5.2 Configuring Studio in JGSS mode ....................................................................................... 28
  5.3 Tips & Troubleshooting ....................................................................................................... 30

6 Kerberos SSO Configuration for TDV Datasources ....................................................... 32
  6.1 Datasource Configuration .................................................................................................... 32
  6.2 Testing & Validation ............................................................................................................. 34
  6.3 Tips & Troubleshooting ....................................................................................................... 34

7 Hive TDV Integration using Kerberos ............................................................................. 35
  7.1 Hive Configuration ............................................................................................................... 35
Table of Figures

No table of figures entries found.

Table of Tables

No table of figures entries found.
1 INTRODUCTION

1.1 Purpose
The purpose of this document is to provide step by step guidance on how to configure Kerberos Delegations in Tibco Data Virtualization (TDV).

1.2 Audience
This document is intended to provide guidance for the following users:

- TDV Administrators
- TDV Developers
- TIBCO TDV PS Consultants

1.3 Kerberos Overview
Kerberos is an authentication protocol designed to verify the identity of a user or a service, it consists of a client, a server, and a trusted third party (Key Distribution Center aka KDC) to mediate between them.

Kerberos provides a mechanism for a client to pass identity to a server without sending a password through the network. Instead Kerberos uses a shared key and a KDC to validate the identity of clients.

In Kerberos, users are known as principals. The KDC has a database of principals and their encrypted keys which are used to perform authentication.

Kerberos introduces the concept of a Ticket-Granting Server (TGS). A client that wishes to use a service has to receive a ticket, a time-limited cryptographic message giving it access to the server. Kerberos also requires an Authentication Server (AS) to verify clients. The two servers combined make up a KDC. For our purposes we will be using Active Directory as the specific implementation of KDC and Kerberos.

1.4 Brief History
Kerberos was originally developed at MIT as part of the Athena project. Many open source and commercial products implement Kerberos, here are few examples:

Cygnus’s KerbNet, OpenVision’s AXXION and Microsoft’s Active Directory provides Kerberos authentication, NTLM authentication, user and group management through LDAP, DNS, and a host of other services.

Below are few important Kerberos concepts and terminology

1.5 Realm
Realm simply means the set of users and application servers that the (KDC) covers, or has information about. Thus, for a user to join, or login to a Realm the KDC for that Realm must have knowledge of the user’s credentials (and other information) which are maintained in some form of secure database. In Microsoft terminology this would be called a Domain. Realms may trust other Realms.
1.6 Principals
Within Kerberos, everything that has an identity is a principal including users. These users can be real, physical people that login and make a request, or they can be generic users that make requests as well. Other principals are accounts running services that respond to requests.

1.7 Client Principals
A client principal is a unique identifier of a user account. Client principals are used by Kerberos authentication to associate a user with a user account.

When a principal is acting as a client, regardless of whether that principal is a real person or a generic user, the principal is normally identified through a name in the form username@DOMAIN.

For example, jsmith@CISCO.COM or dv_user.gen@CISCO.COM are typical ways that a client principal might appear.

1.8 Service Principals
A service principal name (SPN) is a unique identifier of a service instance. SPNs are used by Kerberos authentication to associate a service instance with a service account.

A SPN is represented as service-name/QDN@REALM, where QDN (Qualified Domain Name) is the domain name of the host on which the service runs and service-name is an application specific string which identifies the service on the host.

Certain service types use the keyword host. Thus, a service-name of ftp running on a host with a name fileserver.example.com in the Realm @CISCO.COM would have a Service-Principal name of ftp/fileserver.example.com@CISCO.COM.

1.9 Keytabs
A keytab is a file containing pairs of Kerberos principals and an encrypted copy of that principal's key. This file is used to authenticate a principal on a host to Kerberos without human interaction or storing a password in a plain text file.

Because having access to the keytab file for a principal allows one to act as that principal, access to the keytab files should be tightly secured. Access to a keytab file is equivalent to having access to a principal’s credentials.

In a Unix/Linux environment the propagation of principal doesn’t happen automatically. There is no linkage between the Unix account and the principals created in Active Directory, instead, the process in Unix can correctly interpret the authentication based on a keytab file.

Active Directory generates a keytab file for the SPN myservice/myhost@DOMAIN, any process running under any Unix account that has the keytab file can correctly interpret incoming authentication requests.

In a Windows Active Directory environment, the propagation of the principals associated with the service account happen automatically. If a service process is running under an account, the client principal can attempt to authenticate to any SPN associated with that service account and the service can correctly interpret that authentication.
1.10  krb5.conf or krb5.ini

krb5.conf(Unix) or krb5.ini(Windows) is a configuration file for Kerberos and contains configuration information needed by Kerberos. This includes information describing the default Kerberos realm, and the location of the KDCs for known realms. For example:

```plaintext
[logging]
default = FILE:/var/log/krb5libs.log
dc = FILE:/var/log/krb5kdc.log
admin_server = FILE:/var/log/kadmind.log

[libdefaults]
dns_lookup_realm = true
default_tkt_enctypes = RC4-HMAC
default_tgs_enctypes = RC4-HMAC
ticket_lifetime = 24h
renew_lifetime = 7d
forwardable = true
rdns = false
default_realm = CIS.CISCO
default_ccache_name = KEYRING:Persistent:%{uid}

[realms]
CIS.CISCO = {
    kdc = dc01.cis.cisco
    admin_server = dc01.cis.cisco
}
[domain_realm]
.cis.cisco = CIS.CISCO
cis.cisco = CIS.CISCO
```

krb5.conf is located at /etc/krb5.conf on Unix/Linux OS and krb5.ini is located at c:\windows on Windows OS.

1.11  Tickets

A Ticket is a data structure whose content is known only to the issuer of the Ticket and any party or parties to which the ticket is relevant.

Tickets within Kerberos may be either Ticket Granting Tickets (TGT) - essentially proof of successful authentication, or Service Tickets (ST) - issued by a Ticket Granting Service (TGS) enabling the user to access a desired Application Service.

1.12  Getting Tickets

The following information is needed to generate Kerberos tickets:

1.  Krb5.conf - create C:\windows\krb5.ini (or /etc/krb5.conf on Linux/Unix/MAC) using the contents from the provided krb5.conf
2.  Principal - Principal can be either 2 or 3 parts depending on the type of principle; client principals are in the format username@DOMAIN, and service principals are in the format servicename/host[:port]@DOMAIN
3.  Keytab - Keytab file is usually provided by a Windows (or Unix/Linux) system admin

On Linux/Unix run following command "kinit -k -t <path_to_keytab> < Principal Name>"
On Windows getting a Kerberos ticket is as easy as typing "kinit".

You will be prompted to enter the password for the user <username>@MYDOMAIN.COM. The kinit utility will communicate with the Kerberos Infrastructure (Active Directory) and if you entered the correct password kinit will write the ticket into the credential cache.

1.13 Credential/Ticket Cache

The command klist -e -f will display the tickets that are in the credential cache, but also the encryption algorithms and flags associated with those tickets. This can be useful in debugging issues with tickets.

1.14 How it Works

Below is a step by step process which describes how authentication works in Kerberos

1. Client with a principal “client1” wants to authenticate with the Server with principal “serverA”, the process starts by client1 generating a request “I want to authenticate with the serverA”.

2. Then client1 signs that request with its secret key and then client1 transmits it to the Kerberos Infrastructure(KDC). Kerberos checks the signature of the request. If the signature matches that of client1, then Kerberos knows the request is valid.

3. Kerberos generates a ticket with the contents “serverA, this is client1”. Kerberos encrypts that token with server A’s secret key, then encrypts it again with client1’s secret key, and sends it back to client1.

4. When “client1” receives the twice encrypted token, it uses its own secret key to decrypt the first layer of encryption. Then “client1” forwards the encrypted key to “serverA”. The “serverA” can then use its key to complete the decryption and verify that the incoming connection is from client1.

There are a few important things to note here. First, Kerberos does not specify how the secret keys are originally established between the principals and the Kerberos infrastructure. Second, Kerberos allows the propagation of identity, but does not provide for privileges. In other words, Kerberos provides authentication, but not authorization.

1.15 SSPI versus JGSS mode in TDV configuration

There are two Kerberos modes that TDV can be configured in: Security Support Provider Interface(SSPI) which is native for Windows and Java Generic Security Services (JGSS) non-native for Windows. Native Kerberos SSO allows simpler configuration between Windows components, such as KDC, Windows services and ODBC.

TDV Server and Studio are java applications, they are not native to Windows, so in some cases they have to use JGSS protocol. TDV setup in JGSS mode is complex, this will be discussed in detailed in Kerberos SSO configuration on Windows section, so when possible, SSPI mode should be used for TDV. Here are the cases when TDV must be configured in JGSS mode:

1. TDV is running on Linux/Unix.

2. Delegation of authentication from TDV client applications to TDV data sources is required when TDV itself is running on Windows. This requirement is caused by the fact that currently TDV does not support such delegation in SSPI mode.
2 Kerberos SSO Configuration in TDV – Prerequisites

2.1 Active Directory Configuration Prerequisites

1. Create an AD account that will be used by TDV to connect to AD and authenticate external (to TDV) users.
   a. This step can be done also if Kerberos SSO is not configured for TDV, but external (AD) users still need to be explicitly logged into TDV with their user name and password.
   b. When Kerberos SSO is configured in TDV, this step becomes mandatory, because SSO implies that an AD user that has already been authenticated in AD prior to connecting to TDV now needs to be logged into TDV; so by definition it is an external to TDV user (i.e. AD user).

2. Create new or use existing AD groups for users that will be logging into TDV. Add the external users to those groups.
   a. When testing the setup, it is sufficient to create a small number of groups with a couple of users in them.

3. Create a non-personal account (sometimes also called service account) in AD that the TDV server will run under
   a. This can be the same account as in the above step.

4. If Kerberos delegation of credentials from TDV clients to TDV data sources is required, then delegation should be also configured in AD for the service account running TDV.

5. The next two steps will be described in detail in the further sections since they are run from command line, so the exact syntax is OS-specific.
   a. Create Kerberos SPN for the AD account that will be running TDV.

If TDV needs to be configured in JGSS mode (which is needed in two cases: TDV is running on Linux or TDV is running on Windows, but also requires Kerberos delegation of credentials from TDV clients to TDV data sources), then a keytab file must be created for the service account running TDV.

2.2 TDV Configuration Prerequisites

1. TDV server has been successfully installed and any necessary service packs/hotfixes have been applied. Please refer to the TDV Installation and Upgrade Guide for installation instructions

2. TDV server license has been applied to the installation. Please refer to the TDV Administration Guide for guidance for deployment instructions

   Please note: Only the base TDV server license is required to configure Kerberos authentication. Additional licenses for modules such as TDV Active Cluster or PPA adapters can be applied at a later time.

3. Configure ldap.properties file to specify the search pattern for the external groups in AD.
Requires that these groups already exist in AD with at least one user (for testing) and also requires knowledge of where these groups and users are created in AD tree, please refer to LDAP domain administration section in TDV Admin guide. This step is done in TDV Manager.

4. Configure an LDAP Domain in TDV Manager.

Requires AD account created for this first (as described in the previous section). An external LDAP domain is created in TDV Manager, a URL is saved along with user name and password for this AD account.

5. Log into Studio with an external user account.

This step is performed with explicit credentials (user name and password).
3 Kerberos SSO Configuration for TDV on Linux

Following are the high level steps needed to integrate TDV with Kerberos when the TDV server is running in a Linux/Unix OS. These instructions assume that all steps from the above prerequisites section have been completed.

1. Active Directory
   a. Create AD (Active Directory/Domain Controller) Account
   b. Set SPNs (Service Principal Name)
   c. Generate Keytab for TDV Server

2. Tibco Data Virtualization
   a. Copy Keytab (verify TDV’s service account has read access to the keytab file)
   b. Verify Keytab using klist and TestKerberos.jar

3. Change TDV’s configuration settings via Studio to enable Kerberos/specify the location of the pertinent Kerberos files on the TDV Server.

3.1 Active Directory Configuration

1. Create an Active Directory account that TDV will be using for Kerberos authentication, in our example, we will create an account called krbcis.

2. Generate SPN’s for the krbcis account created in the step above using commands below

   ```
   setspn -a HTTP/RedCis.cis.cisco@CIS.CISCO cis/krbcis
   ```

3. These SPNs are extremely important and need to be set properly in order for Kerberos integration to work. Once set, a keytab file needs to be generated from an active directory server using the ktpass command line utility, you can verify the SPN’s were set correctly by looking them up using the Setspn –l command,

4. Generate the Keytab for the newly created account

   ```
   ktpass -princ HTTP/RedCis.cis.cisco@CIS.CISCO -mapUser cis/krbcis -mapOp set -pass MyPassword -crypto RC4-HMAC-NT -pType KRB5_NT_PRINCIPAL -out MyKeytabName.keytab
   ```

   If user password contains special characters make sure to enclose the value with single quotes.

   Part of the process that occurs when running ktpass is that the SPN name gets mapped to the account. What this means is that the AD account’s login name will be changed from cissvc to HTTP/RedCis.Cis.Cisco. If this doesn’t get updated, you will get an error saying that the server is not found in the Kerberos database.

5. Copy the Keytab to UNIX/Linux/Windows TDV Server

   Copy the generated keytab file from the Active Directory server to a suitable directory on the server where TDV is installed (e.g. `<TDV_HOME>/conf/server/<keytab.file>`)
3.2 TDV Server Configuration

Verify and test the Kerberos artifacts using steps mentioned below

1. Check KRB5.conf file to ensure Domain/Realm configuration is correct
2. Double check encryption types: default_tkt_enctypes = RC4-HMAC default_tgs_enctypes = RC4-HMAC
3. Copy the Keytab to server filesystem (validate that the TDV Service Account can read the directory/file).
4. Double check the contents of the Keytab by using klist.
5. klist –k –t /path/to/keytab.keytab

![Keytab example]

6. Download TestKerberos.jar from https://cisco.box.com/s/t5t0twuojumnk12ip2eszofu0e0an7y1
7. Run TestKerberos.jar using java –jar TestKerberos.jar [keytab] [SPN] [krb5Conf file] to validate keytab, krb5conf and SPN
8. When Unix/Linux servers are provisioned in most cases they are already provisioned in AD and configured to work with KDC, if not please join the TDV server to Realm using instructions below

Required packages for joining the realm, please download and install these packages:

a. realmd
b. oddjob
c. oddjob-mkhomedir
d. sssd
e. adcli

Once the packages/dependencies are installed successfully you need to join the TDV server to the domain by running this following command.

```
[ciscRedCis ~]$ realm join cts.cisco -U userId
```

The console will prompt for the AD account password.

After joining the domain, you can validate by running realm list. The output should resemble the following:
3.3 Configure TDV Settings via Studio

1. Map Windows domain name (key) to External domain name (value) created in TDV Manager

Login to TDV Studio and navigate to Configuration > Security > Authentication. Choose attribute “Windows Domain Mapping”

Create a new entry and map the external domain name created in TDV Manager in pre-requisites step 3 to actual Windows domain name in the current environment. Apply this change.
2. Login to TDV Studio and navigate to -> Admin -> Configuration -> Kerberos

3. Set Kerberos Authentication: True

4. Set Debug Output Kerberos: False - Setting this to True will make Kerberos connections (via ODBC, ADO.net, etc. return useful Kerberos error messages when encountered).

5. Set path to KeyTab File: /Path/To/Keytab.keytab

6. Set the location of Kerberos Configuration File: /etc/krb5.conf

7. Leave Kerberos Ticket Cache Name: Blank

8. Set Native: False

9. Required Principal Name: HTTP/RedCis.cis.cisco@CIS.CISCO - Set in step 1 in Active Directory configuration

10. Save, Exit studio and Restart TDV

3.4 Testing & Validation
The TestKerberos utility can optionally be used to further validate the TDV server’s Kerberos integration. This tool can be downloaded here.

Please note: If running from a linux server that does not have a gui installed, you can manually pass the properties into the jar file.

The utility accepts the following input file format:
**TKserverprops.properties**

type=server
krb5Conf=/etc/krb5.conf
keytab=/apps/Cisco_Systems/CIS_7.0.3/cissvc.keytab
spn=http/redcis.cis.cisco
port=9800
verbose=true

**TKclientprops.properties**

type=client
hostname=redcis.cis.cisco
krb5Conf=/etc/krb5.conf
keytab=/tmp/keytab/cisclient.keytab
spn=HTTP/RedCis.cis.cisco
port=9800
verbose=true
username=cisclient

**TestKerberos Commandline**

If testing on the same box, make sure you run with 2 separate user accounts.

Run Server:

```
java -jar /tmp/keytab/TestKerberosUI.jar /tmp/keytab/TKserverprops.properties
```

Run Client:

```
java -jar /tmp/keytab/TestKerberosUI.jar /tmp/keytab/TKclientprops.properties
```

A successful test should resemble the following:

```
The Kerberos subject was initialized successfully
Found Keytab /home/cis/cis/keytab/cissvc.keytab for HTTP/RedCis.cis.cisco@CISCO
Found Keytab /home/cis/cis/keytab/cissvc.keytab for HTTP/RedCis.cis.cisco@CISCO
Found ticket for HTTP/RedCis.cis.cisco@CISCO to go to krbtgt/CISCO/CISCO expiring on Tue May 17 13:30:37 UTC 2016
The GSS Context was created successfully.
Now listening for client connections...
```
4 Kerberos SSO Configuration for TDV on Windows

If delegation of authentication from TDV client applications to TDV data sources is not required for the given TDV environment, then TDV installations on Windows should be configured in Windows native (SSPI) mode, since this option is simpler and more reliable and takes advantage of internal Windows Kerberos implementation. If delegation is required, then we recommend that TDV still be configured and tested successfully in SSPI mode first, before taking the steps to re-configure TDV in JGSS mode. The sections below assume that the recommended steps are followed.

4.1 Configuration steps to use TDV in SSPI mode

Please note that the below sections also assume that the steps from the above prerequisites section have been completed.

4.1.1 Active Directory configuration to use TDV in SSPI mode

1. The only step required in AD at this point is creation of the SPN for the service account running TDV and host name where it is running. Windows AD Domain administrator would run

   ```
   setspn -s HTTP/<TDVHostName>.<WindowsDomainName>@<REALM> <serviceAccountName>
   ```

2. Once this is done, TDV administrator can list SPNs created for the given service account from his/her own machine like this:

   ```
   setspn -l <serviceAccountName>
   ```

   The above command will list all SPNs in the current domain for the given AD service account.

3. Only one SPN should be defined for TDV service account and it should have format

   a. HTTP/<TDVHOSTNAME>.<WindowsDomainName>@<REALM>

   Note: KDC will allow creation of SPNs in different formats, for example without @<REALM> at the end, so make sure you keep this format. REALM is always in upper case. After SPN is created, it is treated by AD as a string in case sensitive manner.

4.1.2 TDV Configuration for SSPI mode

Modify the TDV windows service, to run under the Windows service account created above

By default, new Windows installations of TDV are configured to run as a service under the credentials of the user account that performed the installation. You will need to change this account in order to configure Kerberos integration successfully.

On the Windows server running TDV open Window Services window and find service running TDV.

Before change it looks like this (runs under Local System account):
Select TDV Server 7.0 service and click on properties, to get to dialog box below

Select “This Account”, option and then click Browse

In the Object Name text box (see below), specify `<domainName>`\`<serviceAccountName>`, in the example below it is namIDataVtp and click on Check Names button. DataVtp account should be found with the SPN in brackets. Click OK.

In these screenshots wcltciscop01 is host name for the machine running TDV. Actual Windows domain name and REALM are replaced with `<domainName>` and `<REALM>`.
Specify the password for DataVrtp account. The Properties Window should look like the figure above.

Click Ok to get a confirmation message that this Service changed the account and that it will become effective on the next restart of the Service.

After the change TDV Service runs under the service account Datavrtp that we designated for this service earlier: you can see that the “Log On As” is now NAMIDataVrtp:
Restart TDV Server 7.0 Service.

There is no need to make the same change for the other two TDV-related Windows services: “TDV Repository 7.0” and “TDV Database Cache 7.0”.

Use Studio to configure Kerberos SSO in SSPI (Native) mode. After TDV server is started, from Studio launch Administration > Configuration and then open Server > Configuration > Security > Authentication > Kerberos to make following changes:

a. Change Allow Kerberos Authentication from false to true.
b. Change Native from false to true.

c. Map Windows domain name (key) to External domain name (value) created in TDV Manager as part of prerequisites section steps. Names are case sensitive here. In the configuration screen, navigate to Configuration > Security > Authentication. Choose attribute “Windows Domain Mapping”. Create a new entry and map the external domain name created in TDV Manager in prerequisites step to actual Windows domain name in the current environment. Apply this change.

The configuration of Kerberos SSO is now complete. You can now test Kerberos SSO from a TDV client in the next step via ODBC DSN.
Configure ODBC DSN and test it in Kerberos mode.

Configure ODBC DSN and test it successfully against a published TDV data source with explicit user name and password. TDV Client Interfaces guide has the required instructions about configuring ODBC DSN. Select Integrated Authentication option for the DSN as Kerberos and then provide the SPN that was created on the previous steps.

Click on the Test button to make sure Kerberos SSO is working correctly.

4.2 Configuration steps to switch using TDV from SSPI to JGSS mode

Create a file krb5.ini in C:\Windows directory of the TDV Server based on the contents for the Kerberos REALM provided by Windows Domain Admin:

```ini
[libdefaults]
default_realms = <REALM>
dns_lookup_realm = true
dns_lookup_kdc = true
ticket_lifetime = 4h
renew_lifetime = 24h
forwardable = true
default_tkt_enctypes = RC4-HMAC
default_tgs_enctypes = RC4-HMAC

[realms]
<REALM> = {
kdc = cltnamdc1.<domainName>.com:88
admin_server = cltnamdc1.<domainName>
}

[domain_realm]
.<realm> = <REALM>
<realm> = <REALM>
```
Request Windows domain admin to generate a Kerberos keytab file for DataVrtp account using the following command:

```
ktpass -princ HTTP/wcltciscop01.<domainName>@<REALM> -crypto all -ptype KRB5_NT_PRINCIPAL -mapuser DataVrtp -pass ********* -out DataVrtp.keytab
```

where ******** should be replaced by the domain password for DataVrtp account.

Copy keytab file into `<TDV_INSTALL_DIR>\keytab\DataVrtp.keytab`

That is a configurable folder path via Studio, we will discuss this configuration in the next section.

**Note:** `ktpass` command not only creates a keytab file, but it also creates (or re-creates if it existed) an SPN. That SPN has a version number associated with it, and it matches with the keytab generated by the same `ktpass` command execution. So after running `ktpass` command don’t run `setspn –s` command, because it will re-create the SPN with a different version number that will not match the keytab.

### 4.2.1 TDV Server Configuration for JGSS mode.

Make the following additional three configuration settings in TDV server via Studio configuration menu (all three have to be done for delegation to work):

In Studio open menu Administration > Configuration > Server > Configuration > Security > Authentication

- Under Kerberos folder provide path to the keytab file. Key/Tab File: C:\Program Files\Tibco\keytab\DataVrtp.keytab
- Under Kerberos folder provide name of the principal (SPN for DataVrtp account). Required Principal Name: `HTTP/wcltciscop01.<domainName>@<REALM>`
- Create a new Windows Domain mapping (also keep the existing mapping NAM -> nam). Key: `<REALM>` Value: nam
Here we did not explicitly provide path to krb5.ini that we earlier put in C:\Windows, this file is being found without such explicit path. Also, the Kerberos Native flag is set to false to use JGSS. Configure pass-through data source if required.

In the example below we use a SQL Server data source that provides data in this delegation scenario and makes sure Pass-Through is set to Enabled. Section on configuring SSO for data sources covers data source configuration in more detail.

There is no need to restart TDV Server, these configuration settings take effect immediately. But you may still want to restart and retest to make sure that above settings were written to disk without any issues (if DataVrtp account does not have sufficient privileges in Windows to write into TDV installation folder, the configuration settings will not be saved on server restart without any error).
4.3 Testing & Validation
Test delegation via Excel that uses ODBC DSN. Make sure you can get data. If you can, the delegation is successfully tested.
5 Kerberos SSO Configuration for TDV Studio

Studio, like TDV server, can be configured in either SSPI or JGSS mode. The main requirement for Studio is that its mode should match the mode of TDV server, i.e. if TDV is configured in SSPI mode, then Studio should be configured in SSPI mode; if TDV is configured in JGSS mode, then Studio should be configured in JGSS mode.

Once TDV server is configured for SSO and tested (via ODBC, for example), then it does not need any additional configuration to allow Studio authenticate with TDV server via Kerberos, i.e. all such configuration is performed on the Studio side. Since in most cases Studio is installed on separate client Windows machines (used by TDV developers), such configuration is performed there.

By default, Studio is not enabled for SSO. Once it is enabled, an additional SSO checkbox appears in Studio login dialog window. Once this checkbox is checked, the username, password and domain fields in this dialog window become greyed out and current Windows users' credentials are being used to log into TDV (indirectly, by using Kerberos ticket that already exists by then in that client Windows machine).

The following steps outline how to configure TDV studio for SSO. Please refer to the TDV Administration Guide for more details.

1. In Studio installation directory find file C:\<StudioInstallationDir>\conf\studio, make a copy of krb5_sample.properties and rename it krb5.properties. After a file with this name appears in that directory, next time you start Studio, you will see an SSO checkbox in the Studio login window.
2. Open an editor such as Wordpad to edit krb5.properties. This file contains two sections: one for SSPI mode and one for JGSS mode. Depending on which mode you need, you will need to uncomment one section, keeping the other one commented out. In the further sections of this document we will provide details on how to configure TDV for each of the two modes. Here are contents of the krb5_sample.properties file, that you will be starting with:

```ini
## SSPI and JGSS Switch.
## "true" for SSPI, "false" for JGSS
## NOTICE: Please make sure studio and server are using same facilities (SSPI -> SSPI or JGSS -> JGSS) before you set "native" value
#native=[true|false]

###############################
#                SSPI              
###############################
##Service Principal Name or Service account
#spn=HTTP/FullyQualified_HostName@Realm
#spn=[domain name]\[account name]
#spn=[account name]@[domain name]

###############################
# Default User -- JGSS    
###############################
##Service Principal Name
#spn=HTTP@dev-krb5-win.support.net
#native=false
#doNotPrompt=true
#useKeyTab=false
#debug=true
#useTicketCache=true
#renewTGT=true
#krb5.conf=c:/krb5.conf

###############################
# Specific User -- JGSS   
###############################
##Service Principal Name
#spn=HTTP@dev-krb5-win.support.net
#native=false
#principal=principalName
#doNotPrompt=true
#storeKey=true
#debug=true
#useKeyTab=true
#keyTab=keytab file
#krb5.conf=c:/krb5.conf
```
5.1 Configuring Studio in SSPI mode

In order to configure Studio in SSPI mode, you only need to know one parameter: SPN that was created by Windows domain administrator for the TDV server.

1. Uncomment the line that says native and set it to true. This line should look like this:
   
   native = true

2. Set SPN in the SSPI section of the file; uncomment line spn=HTTP/FullyQualified_HostName@REALM and replace the value of the SPN with the actual SPN string defined for the current environment using the pattern above.

The value of the actual SPN should be the same as the one tested via ODBC DSN when Kerberos configuration was tested for TDV server in SSPI mode. If you take a look above at the required SPN pattern for SSPI and JGSS modes, you will notice, that it is different: SSPI pattern is

   HTTP/<hostName.domainName>@<REALM>

the JGSS pattern is

   HTTP@<hostName.domainName.REALM>

Note: In ODBC DSN SSPI pattern should always be used, no matter which mode TDV Server is configured in.

Once changes are made and saved in krb5.properties file, restart Studio (close all opened Studio windows and open a new one). Log into TDV using SSO checkbox in Studio instead of explicit username and password (also provide correct server name and port for TDV server previously configured and tested for Kerberos). If authentication is successful and main Studio window opens, it means Kerberos SSO configuration for Studio was done correctly.

5.2 Configuring Studio in JGSS mode

Studio configuration in JGSS mode is more challenging as it requires configuration of more explicit parameters that just an SPN. For some Windows environments there is a known issue when local Windows LSA does not allow Ticket Granting Ticket (TGT) for Studio. This is caused by Windows UAC that zeroes-out the Kerberos session key if the user account has some permissions or belongs to some groups that are considered as too generous by UAC to keep the original session key. Such restriction by the UAC is called ‘split token’. The solution to this problem is either to investigate Windows permissions and group memberships of the user account of disable UAC altogether. Below we will show how to check for the ‘split token’ condition and how to disable the UAC.

Below are the configuration steps:

1. On Windows machine running Studio add a Windows Registry Key to enable Ticket-Granting-Ticket (TGT) Session Keys :

   Using regedit utility, change the allowtgtsessionkey registry REG_DWORD value to 1 to include a session key in the TGT.

   For Windows XP and Windows 2000, the registry location of allowtgtsessionkey is:

   HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\Lsa\Kerberos
For Windows 2003 and Windows Vista, the registry location of allowtgtsessionkey is:

```
HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\Lsa\Kerberos\Parameters
```

2. A value of 1 requires that a session key be returned with the TGT, and enables use of Kerberos TGT sessions. Open studio configuration file krb5.properties (see section above on configuring Studio in SSPI mode) and make the following changes there.

   a. Change native to false for JGSS: `native = false`
   
   b. Comment out SSPI section of the document (if SSPI was configured previously, SSPI SPN should be the only uncommented value there)

   c. JGSS section of the document contains two sub-sections: Default User -- JGSS and Specific User – JGSS. Specific user section contains an additional parameter – user principal. Since our goal is to be able to authenticate any currently logged in user on the given Windows machine running Studio, we usually don’t need to configure a specific user and its principal, which is mostly used for specific troubleshooting with TDVco Support, so we will keep this section commented out and do the changes in the Default User section of the configuration file. Here are those changes, uncomment corresponding parameters:

   ```
   spn= HTTP@<hostName.domainName.REALM>
doNotPrompt=true
useKeyTab=false
debug=true
useTicketCache=true
renewTGT=false
krb5.conf=C:/Windows/krb5.ini
   ```

3. You may have noticed the last configuration parameter. It points to a file that is similar to the one we created on the TDV server side when configuring JGSS there. Copy this file from the TDV server to the Studio machine (if they are different), following above path (you can also use a different location for the file and different path above). This file should be created on every Windows client machine that will be using SSO. Also notice single forward slashes in the path: Studio is a java application, so it expects this style of slashes.

4. Restart Studio (close all opened Studio windows and open a new one). Log into TDV using SSO checkbox in Studio instead of explicit username and password (also provide correct server name and port for TDV server previously configured and tested for Kerberos). If authentication is successful and main Studio window opens, it means Kerberos SSO configuration for Studio was done correctly.

5. Make sure the user is not a local administrator on the client machine, because it will create a 'split token' situation described above.

6. Verify that after all above steps we are not getting a ‘split token’ situation by running the following command from regular (not as administrator) command prompt: `klist tgt`

Take a look at the Session Key field. If it shows the following:

```
Session Key : KeyType 0x12 - AES-256-CTS-HMAC-SHA1-96 : KeyLength 32 - 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

It means the session key is zeroed out by the UAC. Now run Command Prompt as Administrator from Window start menu. If prompted, provide the same end user credentials, not local Windows administrator credentials. After the
command prompt is opened, it will be called in its header “Administrator: Command Prompt” instead of just “Command Prompt”:

![Command Prompt](image)

Re-run `klist tgt` command again and check the Session Key:

![Command Prompt](image)

As you can see, this time the key does not contain all zeros, so Kerberos SSO should work.

Running any application in Windows, including Command Prompt, as Administrator (with the same end user credentials, not local admin credentials) effectively disables Windows UAC. So if you can’t configure permissions and group memberships for the end user account that would allow using UAC and not getting a split token (i.e. zeroed out Kerberos Session Key), you may have to disable UAC when running Studio by running it as Administrator. There is also a way to disable UAC system-wide on Windows, in which case Command Prompt will always get opened as Administrator as shown above.

### 5.3 Tips & Troubleshooting

Before using SSO in Studio for the first time, run `klist tgt` command from regular command prompt and from ‘as Administrator’ command prompt to see if the session key is zeroed out or not. It will also get zeroed out if you don’t set registry entry `allowtgsessionkey`.

You may also want to run TestKerberos utility in client mode to check if you get any issues. TestKerberos provides SSO handshake details and can help troubleshoot any issues.

There are also some TDV debug settings in `log4j.properties` that allow providing additional SSO-related logging. Here they are:

```properties
log4j.logger.com.compositesw.server.security.authenticator=DEBUG
log4j.logger.com.compositesw.server.soa.deployment.AuthenticationFilter=DEBUG
```

Once a user connects via studio with SSO he/she can now run queries against SSO enabled data sources. The main caveat here is that currently TDV only supports Kerberos delegation (which in turn allows using data sources in pass-through mode) when it is configured in JGSS mode, which in turn requires Studio to be configured in JGSS mode for such scenario.

If a user connects to Studio with explicit credentials or in SSPI mode, he/she can’t work with pass-through data sources (‘Test Connection’ and all queries will fail for such data sources).
One workaround for the scenario when Studio can’t be successfully configured in JGSS mode, is to temporarily change a corresponding data source from KERBEROS and Pass-Through to default Basic authentication and Disable Pass-through during development/testing phase of the project, then enable Pass-through/Kerberos on a data source again and do end-end testing via ODBC DSN client (like Excel).

Another workaround is to keep KERBEROS mode for the data source, disable pass-through and provide some Windows credentials for the data source that it can successfully authenticate with. Those would be some user’s Windows credentials that this data source will be saved with, so this approach would be a short-term solution, since all connections to that data source will be done as this specific user but shared by all TDV users. See more details below.
6 Kerberos SSO Configuration for TDV Datasources

6.1 Datasource Configuration

In order for Kerberos delegations to be passed through TDV, the source system needs to be configured to support Kerberos delegations. Kerberos configuration with in TDV is applicable for all datasources that support Kerberos authentication, once TDV and Studio are configured in correct mode, we should be able to connect to any datasource that supports and is configured for Kerberos authentication by choosing Kerberos as the authentication scheme when we create the datasource connection.

In example below, we are using SQL Server. But the configuration in TDV is same for any other Kerberos supported datasources such as Oracle, DB2 or Sybase.

In general, the steps for validating the active directory portion of the configuration would be similar across all datasource platforms.

1. When TDV connects to SQL Server with explicit credentials and in Basic authentication mode, it uses an account that is created directly in SQL Server. When TDV connects to SQL Server in pass-through mode, it uses an AD account. Before SQL Server data source is configured in TDV with KERBEROS authentication and in pass-through mode, we need to make sure that SQL Server itself is configured for Kerberos SSO, i.e. that it is possible to connect to SQL Server from a client (e.g. SQL Server Development Studio) with an AD account.

This concept is similar to logging into TDV with an account from an external AD domain instead of composite domain, when TDV delegates authentication to AD while authorization for that external account is configured in TDV. SQL Server has similar functionality to delegate authentication to AD while permissions to that account are configured in SQL Server (usually on AD group level).

So the first step here is to connect to SQL server from SQL Server Development Studio with an AD account that we are planning to test TDV pass-through with and make sure we have access to SQL Server tables/data that we expect to also access from TDV later.

2. SQL Server can be configured in either NTLM or Kerberos mode. At this step we need to make sure it is configured in Kerberos mode. This can be done by running the following SQL statement against the system database on the SQL instance. For more information on this configuration, please visit the link here.

```
select auth_scheme from sys.dm_exec_connections where session_id=@@spid
```

Make sure that this query returns 'Kerberos', not 'ntlm' value. SQL Server DBA should be able to reconfigure this if necessary.

3. Ensure that appropriate SPN’s (Service Principal Names) have been set for SQL Server. In the below screenshot you can see that I am looking up the service principal names by using the active directory account that is running the SQL Service on the Windows Server. Please note, you will need to know the AD account’s name in order to look up this service principal reference.

The command for the below looks like setspn –l domain\username. In this case, I ran setspn –l cis\sql
Usually MSSQLSvc is the name of SQL Server service and it becomes the first part of the SPN, as you can see from the above. In this example you can also see that this service is configured for two hosts: CisKerberosSql and Sql2014.

When configuring the datasource in TDV it is important that your data source's host name includes domain name as specified in the SPN above i.e. part of the SPN string after forward slash and before port number. For example, if you tried to use a hostname of CisKerberosSql instead of CisKerberosSql.cis.cisco, you would get an error stating that hostname was not found in the Kerberos Database. More details on this error are provided in the Tips & Troubleshooting section of this document.

4. Delegation of SSO credentials via TDV to SQL Server may require SQL Server-specific Windows .dll file sqljdbc_auth.dll to be added to TDV installation directory. This dll file comes together with the SQL Server JDBC driver (from MS download site) and should be copied into the following TDV folder: 
<TDV_Server_install>/apps/common/lib/win64. If that specific folder does not exist, it should be created manually.

5. Finally, configure SQL Server data source in CDV Studio.

a. Configure and test connection to SQL Server data source using explicit credentials. If you use direct SQL Server account (that has access to that SQL Server instance), you should use Basic authentication mode in the data source configuration in Studio. If you use AD account (that has access to that SQL Server instance via Windows authentication), then use KERBEROS authentication mode in the data source configuration in Studio. In both cases, you should temporarily save these credentials in Studio before you use ‘Test Connection’ button. Pass-through setting should be in ‘Disabled’ mode.

b. Once connection test is successful and you can introspect data from SQL Server instance, change the connection properties: remove user name and password, make sure authentication mode is set to KERBEROS and Enable Pass-through. Then in order to successfully test connection to that data source using ‘Test Connection’ button, you need to make sure you open Studio and login using SSO checkbox. You would also need to be logged in to the machine running Studio with the same AD account that you use to test delegation to SQL Server. If you log in to Studio with explicit credentials instead of using SSO checkbox, then pass-through data source will not work (‘Test Connection’ button on the data source will return an error and also all queries against that data source will fail).

Studio should be configured for SSO in JGSS mode, because its mode (SSPI vs. JGSS) should match the mode of TDV Server and currently SSO delegation in TDV server to data sources in not supported when TDV is in SSPI mode.
6.2 Testing & Validation


You can test end-end (Windows workstation -> TDV -> SQL Server) SSO delegation via Studio by using SSO checkbox on Studio login.

   a. Use ‘Test Connection’ button on SQL Server data source.
   b. If above works, execute a view that gets data from an introspected SQL Server table.

2. Testing via ODBC.

   a. Create an ODBC DSN and configure it in Kerberos mode, specify SPN for TDV. Click on ‘Test’ button.
   b. If above works, connect to that DSN via Excel and make sure you can get data.

6.3 Tips & Troubleshooting

Common errors and resolutions are provided below:

- The client credential cannot be delegated
  You are not connected to studio / TDV drivers with Kerberos and are trying to access a SSO datasource.

- KrbException: Server not found in Kerberos database (7)
  Double check the hostname you are using. Typically this error occurs when you are referencing a hostname that doesn’t have an SPN registered. IE. MySqlServer instead of MySqlServer.Domain.Name. You can double check this by running setspn -l Domain/AccountRunningTheSqlServerService. If this does not help, ask Windows Domain administrator to re-generate the keytab that is being used by TDV server using ktpass command; this should also recreate the SPN that matches the keytab. After that do not re-create SPN again by setspn -A command, because that may create a mismatch between keytab and SPN. Copy the newly created keytab file to the location that TDV server is pointed to in Administration > Configuration > Server > Configuration > Security > Authentication > Kerberos > Keytab File:
7  Hive TDV Integration using Kerberos

7.1  Hive Configuration
HiveServer2 supports authentication of the Thrift client using either Kerberos or LDAP

If you configure HiveServer2 to use Kerberos authentication, HiveServer2 acquires a Kerberos ticket during start-up. HiveServer2 requires a principal and keytab file specified in the configuration. The client applications (for example JDBC or beeline) must get a valid Kerberos ticket before initiating a connection to HiveServer2.

7.2  Enabling Kerberos Authentication for HiveServer2
To enable Kerberos Authentication for HiveServer2, add the following properties in the /etc/hive/conf/hive-site.xml file:

```xml
<property>
    <name>hive.server2.authentication</name>
    <value>KERBEROS</value>
</property>

<property>
    <name>hive.server2.authentication.kerberos.principal</name>
    <value>hive/_HOST@YOUR-REALM.COM</value>
</property>

<property>
    <name> hive.server2.authentication.kerberos.keytab</name>
    <value>/etc/hive/conf/hive.keytab</value>
</property>
```

where:

1. The _HOST@YOUR-REALM.COM value in the example above is the Kerberos principal for the host where HiveServer2 is running.

   The special string _HOST in the properties is replaced at run-time by the fully-qualified domain name of the host machine where the daemon is running. This requires that reverse DNS is properly working on all the hosts configured this way. Replace YOUR-REALM.COM with the name of the Kerberos realm your Hadoop cluster is in.

2. The /etc/hive/conf/hive.keytab value in the example above is a keytab file for that principal.

   If you configure HiveServer2 to use both Kerberos authentication and secure impersonation, JDBC clients and BeeLine can specify an alternate session user. If these clients have proxy user privileges, HiveServer2 will impersonate the alternate user instead of the one connecting. The alternate user can be specified by the JDBC connection string proxyUser=userName

7.3  Configuring JDBC Clients for Kerberos Authentication with HiveServer2
JDBC-based clients must include principal=<HiveServer2-Kerberos-Principal> in the JDBC connection string. For example:

```java
String url = "jdbc:hive2://node1:10000/default;principal=hive/HiveServer2Host@YOUR-REALM.COM"
Connection con = DriverManager.getConnection(url);
```

where HiveServer2Host is the host where HiveServer2 is running.
7.4 TDV Configuration

Please follow instructions below to connect to HiveServer2 with Kerberos authentication.

Obtain the JARS that are needed for Hive JDBC adapter.

Few important points to consider:

- Different CDH/HDP versions might have a slight variation in the names.
- On some versions, we noticed there might be additional jars dependency.
- This list pertains to CDH 5.1 (Hive 0.14).
- The jars in general are not forward or backward compatible.
- It is recommended to have the same set of jars for the client as on the server.

Below is the list of jars required by the Hive JDBC Driver to create Hive data source (e.g., Hive 0.14). You would need to obtain these jars from target Hadoop cluster.

```
/usr/lib/hadoop-mapreduce/hadoop-mapreduce-client-core-2.3.0-cdh5.1.2.jar
/usr/lib/hadoop/hadoop-common-2.3.0-cdh5.1.2.jar
/usr/lib/hadoop/hadoop-auth-2.3.0-cdh5.1.2.jar
/usr/lib/hadoop/lib/commons-configuration-1.6.jar
/usr/lib/hadoop/lib/commons-logging-1.1.3.jar
/usr/lib/hadoop/lib/commons-collections-3.2.1.jar
/usr/lib/hadoop/lib/slf4j-log4j12.jar
/usr/lib/hadoop/lib/slf4j-api-1.7.5.jar
/usr/lib/hive/lib/hive-exec-0.12.0-cdh5.1.2.jar
/usr/lib/hive/lib/hive-jdbc-0.12.0-cdh5.1.2.jar
/usr/lib/hive/lib/hive-service-0.12.0-cdh5.1.2.jar
/usr/lib/hive/lib/libthrift-0.9.0.cloudera.2.jar
/usr/lib/hive/lib/libfb303-0.9.0.jar
/usr/lib/hive/lib/log4j-1.2.16.jar
/usr/lib/hive/lib/httpclient-4.2.5.jar
/usr/lib/hive/lib/httpcore-4.2.5.jar
```

For Kerberos:

If your realm is not the default_realm in krb5.conf file, you would also need Core-Site.jar. If you can set the default_realm in your krb5.conf to the hive Kerberos realm, then Core-Site.jar is not required.

Before you attempt to test your Hive data source with Kerberos, make sure that you test your SPN and keytab by running KINIT in a command prompt window.

```
kinit -k -t <path_to_hive.service.keytab> <Hive Service Principal Name>
```

Steps to set up your Hive data source:

1. Copy the JARs to the folder `<TDV_INSTALL_DIR>\adapters\system\hive_0_14_hiveserver2`
2. Create a Hive data source

**Basic Tab**

- **Host**: <your host>
- **Database**: <your db>
- **Port**: <your port>
- **Pass-through Login**: Disabled
- **Authentication**: KERBEROS
- **If Keytab File is enabled**: <path_to_hive.service.keytab>
- **Service Principle Name**: <your SPN name>

If you want to customize the JDBC connection string, modify the advanced property "Connection URL Pattern" (example, to include the principal as in the example below).

```
jdbc:hive2://<HOST>:<PORT>/<DATABASE_NAME>;principal=hive/host-192-168-1-18.cdh.com@CDH_51_KERBEROS.COM;
```

3. Specify the krb5.conf location

a. Set the location of the krb5.conf file in Studio, using the setting Administration->Configuration->Server->Configuration->Security-Authentication->Kerberos->Kerberos Configuration File

b. Restart TDV for the setting to take effect.

Sample krb5.conf:

```
[logging]
default = FILE:/var/log/krb5libs.log
kdc = FILE:/var/log/krb5kdc.log
admin_server = FILE:/var/log/kadmind.log

[libdefaults]
default_realm = EXAMPLE.COM
dns_lookup_kdc = false
dns_lookup_realm = false
ticket_lifetime = 24h
renew_lifetime = 7d
forwardable = true

[realms]
EXAMPLE.COM =
{ kdc = 178.168.255.224 admin_server = 178.168.255.224 }
CDH_51_KERBEROS.COM =
{ kdc = cdh51.com admin_server = cdh51.com }

[domain_realm]
.support.net = SUPPORT.NET
support.net = SUPPORT.NET
host-178-168-1-8.cisco.com = CDH_51_KERBEROS.COM
```
4. If your Kerberos is using AES encryption, then
   a. Download local_policy.jar and US_export_policy.jar from
      http://www.oracle.com/technetwork/java/javase/downloads/jce-7-download-432124.html
   b. Paste these jars in &lt;TDV_INSTALL_DIR&gt;/jre/lib/security

7.5 Testing & Validation
Use the following TestHive_ProxyUser class to test the hive connection, edit the JDBC_DB_URL to provide the correct connection string

```java
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.ResultSetMetaData;
import java.sql.SQLException;

public class TestHive_ProxyUser {
    // JDBC credentials
    static final String JDBC_DRIVER = "org.apache.hive.jdbc.HiveDriver";
    static final String JDBC_DB_URL = "jdbc:hive2://vm-dvbu-as-bdl-003.cisco.com:10000/default;principal=hive/vm-dvbu-as-bdl-003.cisco.com@BDL.AS.DVBU.CISCO.COM;auth=kerberos;kerberosAuthType=fromSubject";
    static final String USER = null;
    static final String PASS = null;
    static Connection getConnection() throws Exception {
        Connection con = null;
        try {
            Class.forName(JDBC_DRIVER);
            con = DriverManager.getConnection(JDBC_DB_URL, USER, PASS);
        } catch (SQLException e) {
            e.printStackTrace();
        } catch (ClassNotFoundException e) {
            e.printStackTrace();
        }
        return con;
    }
    // Print the result set.
    private static int traverseResultSet(ResultSet rs, int max) throws SQLException {
        ResultSetMetaData metaData = rs.getMetaData();
        int rowIndex = 0;
        while (rs.next()) {
            for (int i=1; i<=metaData.getColumnCount(); i++) {
                System.out.print("  " + rs.getString(i));
            }
            System.out.println();
            rowIndex++;
            if(max > 0 && rowIndex >= max )
                break;
        }
        return rowIndex;
    }
    public static void main(String[] args) {
        System.out.println("--- Test started ---");
        Connection conn = null;
        try {
            conn = getConnection();
            ResultSet rs = conn.getMetaData().getTables(null, null, "%", null);
            traverseResultSet(rs, 10);
        } catch (Exception e) {
            e.printStackTrace();
        } finally {
            try { if (conn != null) conn.close(); } catch(Exception e) { e.printStackTrace();
        }
        System.out.println("Test ended ");
    }
}```
Use the command below to test the java class, please provide your keytab, SPN and set the required class path as necessary

```
TDV_HOME=/opt/Tibco/TDV_7.0.4/
javac TestHive_ProxyUser.java
jar -cvf testhive.jar TestHive_ProxyUser.class
kinit -k -t hive.service.keytab hive@BDL.AS.DVBU.CISCO.COM
$TDV_HOME/jre/bin/java -Djavax.security.auth.useSubjectCredsOnly=false -Dsun.security.jgss.debug=true -Dsun.security.krb5.debug=true -Djava.security.krb5.conf=/etc/krb5.conf -cp "$<YOUR_HOME_DIR>*/$TDV_HOME/apps/dlm/cis_ds_hive2/lib/*" TestHive_ProxyUser
```

### 7.6 Tips & Troubleshooting

1. When integrating with Hadoop Stack never download the jars from open source repos, always retrieve correct version of jars from the target Hadoop cluster

2. Admin Guide has the list of jars when connecting to non-secure Hadoop cluster

3. For list of jars when connecting to secure cluster see the TDV Configuration section above, sometimes jars could be named differently across Hadoop flavors and versions

4. Never integrate with TDV directly, test with test program before integrating with TDV

5. Remember that you are working with open source software, things don’t work as expected, so if you hit an error don’t try to figure it out, Google the issue, chances are there might be an outstanding bug open for the issue

6. When debugging an error you will save lot of time by searching using the error stack trace, instead of figuring it out