This article describes how to configure Kerberos Constrained Delegation (KCD) on the NetScaler appliance version 10.1 build 120.13.
# Table of Contents

1. **Introduction** .......................................................................................................................................... 3
   1.1 Audience ........................................................................................................................................... 36
   1.2 2 Introduction .................................................................................................................................... 36

2. **Goal** ....................................................................................................................................................... 5

3. **Application of KPT and Kerberos Constrained Delegation (KCD) on a NetScaler Appliance** ............ 6
   3.1 Workflow ............................................................................................................................................. 6
   3.2 Explanation of Workflow Sequence .................................................................................................... 7
   3.3 3 Integration with Existing Authentication Methods ........................................................................... 8

4. **Configuration** ......................................................................................................................................... 9
   4.1 1 Active Directory Configuration ......................................................................................................... 9
       4.1.1 Create a Kerberos Constrained Delegation (KCD) User Account .................................................. 9
       4.1.2 Enable the Delegation Tab for the Created User .......................................................................... 9
       4.1.3 Add the Services ......................................................................................................................... 13
       4.1.4 Generate Keytab file ........................................................................................................................ 14
   4.2 Web Application Server Configuration .............................................................................................. 16
       4.2.1 Configuring the IIS Server ........................................................................................................... 17
       4.2.2 Web Server - Protocol Negotiation ............................................................................................. 17
   4.3 Setting up NetScaler for Constrained Delegation .............................................................................. 19
       4.3.1 Joining a NetScaler Appliance to the KDC .................................................................................. 19
       4.3.2 Create Backend Servers and Services ......................................................................................... 19
       4.3.3 Create TM and Authentication Virtual Server ............................................................................ 20
       4.3.4 Create KcdAccount ..................................................................................................................... 20
       4.3.5 Create Traffic/Session Profile ...................................................................................................... 22
       4.3.6 Enable Client Side Authentication (Form-based/401basic) ........................................................ 23
5. Troubleshooting ........................................................................................................................................... 24

5.1 Ensuring that Iwagent Process and All Likewise Daemons are Running.................................................... 24

5.2 How to Know if KCD is Working? ........................................................................................................... 27

5.3 Kinit string: Verifying if NetScaler appliance is Requesting a Forwardable Ticket: S4U2SELF ............. 30

5.4 Kinit String: Verifying if Kinit Request Fails or if you Get Any of the Following Errors ...................... 31

5.5 t_s4u request ......................................................................................................................................... 30

6. Firewall Ports Required to be Open for KCD Communication .............................................................. 35

7. Reference .................................................................................................................................................... 36
1. Introduction

1.1 Audience

This document describes configuring Single Sign On (SSO) using Kerberos. This document assumes that the reader has basic knowledge of Kerberos protocol and terms used in Kerberos authentication.

1.2 Introduction

Kerberos SSO is a variant of SSO mechanisms through which a proxy or Application Delivery Controller (ADC) provides a seamless access to a protected resource for a user who logged into ADC. Traditional SSO mechanisms such as Basic, NTLM or forms requires the user’s passwords in order to provide SSO. Kerberos SSO works with or without the user’s password. When user’s password is available, SSO is established through impersonation. When password is not available, SSO is established by delegating a user (say super user), who is allowed to request “tickets” for different users and services.

A major advantage of Kerberos SSO is that ADC need not be aware of user’s password. With growing adoption of One Time Password (OTP) analogous to password, this provides a lot of flexibility in resource access.

Impersonation is possible when user’s password is available. In that case, user would be impersonated and tickets are obtained on behalf of the user. It can also be done with user’s certificate but requires Citrix Receiver for remote certificate signing operations.

When user’s password is not available, Kerberos SSO is possible through delegation. With delegation, an administrator account will be employed to get tickets on behalf of users and services. This configuration is described in the following sections. Administrator account’s credentials are required in this method. However, there is a lot of flexibility in obtaining these credentials. One can use account password or keytab (containing encrypted password) or certificate (which is mapped to user in Active Directory). These are described in section 4.3.3 where configuration for kcdaccount is defined.

In short, Kerberos SSO is possible in the following ways:
### Kerberos SSO Types

<table>
<thead>
<tr>
<th>Kerberos SSO Types</th>
<th>Requirements</th>
</tr>
</thead>
<tbody>
<tr>
<td>Impersonation</td>
<td>User’s password</td>
</tr>
<tr>
<td></td>
<td>User’s certificate and specialized Citrix Receiver</td>
</tr>
<tr>
<td>Constrained Delegation</td>
<td>Delegated user’s password</td>
</tr>
<tr>
<td></td>
<td>Delegated user’s keytab</td>
</tr>
<tr>
<td></td>
<td>Delegated user’s certificate and Certificate Authority certificate</td>
</tr>
</tbody>
</table>

In order to obtain ticket on behalf of user, two new extensions in Keberos V5, S4U2Self and S4UProxy are employed. These are also called Protocol transition and Constrained Delegation. These are described in Addendum.
2. Goal

- To support Kerberos Protocol Transition and Constrained Delegation (Kerberos SSO) on nCore build of the NetScaler software release
- Integrated with all existing supported authentication methods
3. Application of KPT and Kerberos Constrained Delegation (KCD) on a NetScaler Appliance

3.1 Workflow
### 3.2 Explanation of Workflow Sequence

1. The client makes a request to a traffic management (TM) virtual server on a NetScaler appliance, which can be a Load Balancing (LB) or a Content Switching (CS) virtual server.

2. Depending on the authentication policy configured on the TM virtual server, the handshake of client side authentication starts and eventually the TM virtual server receives the complete credential from the client.

3. The TM virtual server verifies the client credential either directly, or through external authentication server.

4. If the authentication succeeds, the TM virtual server selects a backend service and forwards the request to the service.

5. If Kerberos authentication is required or enabled on the service, it responds with a 401 status code with WWW-Authenticate headers as follows:

   ```
   HTTP/1.1 401 Unauthorized
   ...
   WWW-Authenticate: Negotiate
   WWW-Authenticate: NTLM
   ```

   The two WWW-Authentication headers provide the receiver with the information that the more secured Negotiate or Kerberos authentication is preferred. If the receiver does not support Negotiate or Kerberos authentication, the receiver sends the authentication to NTLM.

   When the TM virtual server receives the 401 response and if SSO is enabled at the global level or on the virtual server, then the SSO handler for Kerberos is triggered.

6. NetScaler appliance sends a request to nskrb daemon to get Kerberos ticket.

7. The authentication service request/response (AS-REQ/REP) exchange is the Kerberos GT request and reply messages that the nskrb daemon sends to the KDC. If the exchange is successful, the nskrb gets a Kerberos ticket-granting ticket (TGT). Through the TGS-REQ/REP exchange with the KDC, the nskrb gets the S4U2Self ticket with the TGT, user principle, and TM virtual server principle.

8. nskrb returns the obtained token or an error to NetScaler appliance.
9. The service ticket is encoded by base64 and then sent to the requested service in the HTTP Authorization header.

   ```
   GET /
   Authorization: Negotiate 89a8742aa8729a8b028
   ```

10. The backend service verifies the ticket in the HTTP request. If it is valid, the service allows the user to access and reply with the right HTTP 200 OK response.

11. The NetScaler appliance receives the response, inserts session cookie, and forwards the response to the client.

12. All the subsequent HTTP requests to the same TM virtual server have the session cookie. As long as the cookie is still valid, no further client side authentication is required.

3.3 Integration with Existing Authentication Methods

Kerberos PT and CD are designed to support all existing authentication methods provided on a TM virtual server. The following matrix summarizes these authentication methods:

**Note:** Currently, Certificate-based authentication is not supported in 401-based TM VIP.

<table>
<thead>
<tr>
<th>Client Side Authentication</th>
<th>Server Side Authentication</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Kerberos</td>
</tr>
<tr>
<td></td>
<td>HTTP – Basic, Digest, NTLM</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Non-Kerberos</th>
<th>SAML</th>
<th>NTLM</th>
<th>Kerberos</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Version 1</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Version 2</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>CAC (Smart Card): at SSL/TLS Layer</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>HTTP Basic</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Form-based</td>
<td></td>
</tr>
<tr>
<td>Kerberos</td>
<td>Kerberos</td>
<td>Kerberos</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Kerberos</th>
<th>Kerberos</th>
<th>Kerberos</th>
<th>Kerberos</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
4. Configuration
This section outlines how to set up Kerberos Constrained Delegation with a NetScaler appliance. This involves setting up an account in the Active Directory and the Server hosting the services, and configuring a NetScaler appliance.

4.1 Active Directory Configuration
This section explains the configuration steps required in the Active Directory to enable Constrained Delegation with NetScaler.

4.1.1 Create a Kerberos Constrained Delegation (KCD) User Account
User account must be created for the Constrained Delegation to work. This account must have the rights to do the Protocol Transition, Delegation, and to request a Kerberos Ticket on behalf of a user logging into the NetScaler appliance.

Start by creating a new user in the Active Directory.

In this example, create the kcdtest as the account to provide Constrained Delegation Access to Web server.

4.1.2 Enable the Delegation Tab for the Created User
Delegation is not enabled by default for a User account, you must enable it. This involves the use of the SETSPN command-line tool and a standard Windows 2003 installation does not include this tool.

Check in Active Directory User properties if delegation tab is available, if not download the Windows package.

Install the Windows Server 2003 Support Tools from the product CD or from the Microsoft Download Center (http://go.microsoft.com/fwlink/?LinkId=100114).
The Windows Support Tools for Microsoft Windows 2003 are intended for use by Microsoft support personnel and experienced users to assist in diagnosing and resolving computer problems.

**Quick details**

<table>
<thead>
<tr>
<th>Version:</th>
<th>1.0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Language:</td>
<td>English</td>
</tr>
<tr>
<td>Date published:</td>
<td>3/22/2003</td>
</tr>
<tr>
<td>KB articles:</td>
<td>KB923127</td>
</tr>
</tbody>
</table>

**Files in this download**
The links in this section correspond to files available for this download. Download the files appropriate for you.

<table>
<thead>
<tr>
<th>File name</th>
<th>Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>support.sdb</td>
<td>3.4 MB</td>
</tr>
<tr>
<td>supptools.xml</td>
<td>232 KB</td>
</tr>
</tbody>
</table>

For more information about how to install Windows Support Tools from the product CD, see Install Windows Support Tools ([http://go.microsoft.com/fwlink/?LinkId=62270](http://go.microsoft.com/fwlink/?LinkId=62270)). If this is installed in Windows 2003 server, it can be found in the C:\Program Files\Support Tools folder. To get the latest version of setspn tool see [http://support.microsoft.com/kb/970536](http://support.microsoft.com/kb/970536).

Use the following command to run the setspn tool:

```
setspn -A host/kcdvserver.example.com example\kcdtest
```

**Note:** In this example, *example* is the Domain and *kcdtest* is the user account. In this command kcdtest user is with SPN: *host/kcdvserver.example.com*, and *kcdvserver.example.com* is the Fully Qualified Domain Name (FQDN) of the load balancing vserver. This enables the Delegation tab in the kcdtest Properties.
If the Delegation Tab does not appear, the Active Directory is running in mixed or native mode and must raise it to the Windows 2003 functional level.

**NOTE** that the following steps changes the Active Directory behavior and the support for older Windows clients. If you are uncertain, you should not raise the Domain Functional Level without checking if this has any impact to the environment because this step cannot be reversed.

After the Active Directory is at Windows 2003 functional level you can continue with the Configuration.
The Delegation tab must be visible now. Ensure that you enable the options **Trust this user for delegation to specified services only** and **Use any Authentication protocol**.
Even though other options might seem more accurate, the Kerberos options do not work because Protocol Transition and Constrained Delegation are not enabled.
4.1.3 Add the Services
You must specify the Services this applies to because this is a constrained delegation. Select Add.

Click Users or Computers to select the Computer hosting these services.

In this example, SPH07.example.com hosts the Web service. Therefore, in the preceding SPH07 is selected. This can be any other Server in the Domain.

Note that Constrained Delegation does not support Services hosted in other Domains even though there is a trust relationship to those Domains.
Add the Services on the selected Server.

**Add Services**

To allow services to be delegated for a user or computer, select the appropriate users or computers, and then click the services.

To select one or more user or computer names, click Users or Computers.

Available services:

<table>
<thead>
<tr>
<th>Service Type</th>
<th>User or Computer</th>
<th>Port</th>
<th>Service Name</th>
<th>Domain</th>
</tr>
</thead>
<tbody>
<tr>
<td>eventlog</td>
<td>sph07@com</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>eventsystem</td>
<td>sph07@com</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>exchangeAB</td>
<td>sph07@com</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>fax</td>
<td>sph07@com</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GC</td>
<td>sph07@com</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HOST</td>
<td>sph07@com</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>http</td>
<td>sph07@com</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>is</td>
<td>sph07@com</td>
<td></td>
<td>te</td>
<td></td>
</tr>
<tr>
<td>isadmin</td>
<td>sph07@com</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Select All

OK Cancel

The http service is selected because this example is about setting up Constrained Delegation to Web Server. This is the same for any other Web based applications running on this server such as Share Point/Outlook Web Access. Review the settings and click **Apply/OK**.

4.1.4 Generate Keytab file

Keytab file contains encrypted password for the user. For KCD, this is necessary only if KCDAccount (explained in the section 4.3.3) is configured by using a keytab file. Otherwise, this step can be skipped.

Create the Keytab file for the kcdtest user with SPN:
host/kcdvserver.example.com@example.COM

You can select any of the following two options available to generate a keytab file:

- NetScaler GUI has a utility that helps in creating a batch file which can be run on Domain Controller to generate a keytab file. Select **AAA-Application Traffic > Batch file to generate keytab** to open this utility.

  Copy the generated script and run it from a Domain controller cmd prompt.

  **Note:** This GUI option is only available in 10.1 release.
@echo off
set kcdusername=kcdtest
set kcdpassword=freebsd
echo %kcdusername%
setspn -A HTTP/%kcdusername% %USERDOMAIN%/%kcdusername%
setspn -A host/%kcdusername%@.com@.COM.%USERDNSDOMAIN% %USERDOMAIN%/%kcdusername%
ktpass -out c:\ns_kcd.keytab /princ
host/host/kcdserver = .com@.COM.%USERDNSDOMAIN%@%

Run this script in AD Domain Controller and copy the keytab file to /nsconfig/krb/ folder in NetScaler
- Run the following command to generate a keytab file:
  
  ```
  ktpass /princ host/kcdvserver.example.com@example.COM /ptype KRBS_NT_PRINCIPAL /mapuser example\kcdtest /pass freebsd -out C:\kcdvserver.keytab
  ```

  ![Command Output]

  **Note:** host/kcdvserver.example.com@example.COM is case sensitive.

  Copy the kcdvserver.keytab file to the /nsconfig/krb directory on the NetScaler appliance.

  Active Directory configuration is complete.

  **4.2 Web Application Server Configuration**

  This section describes the configuration required on the server hosting the Web application. In this example, the application is Internet Information Services (IIS) Web server SPH07. This could be any Web application hosted on any server within the domain.
4.2.1 Configuring the IIS Server

Open the Internet Information Services Manager and select the Web site you want to enable Constrained Delegation.

**Note:** The Authentication settings might be set up at the top level (Default Web Site) for all sub-services or it could be setup for each service such as Exchange, Exadmin, ExchWeb, and Public to meet the requirements to log on properly to the required applications.

Open Properties and select the Directory Security tab. Select to edit the authentication and access control.

Ensure that you disable anonymous access and enable Integrated Windows authentication (only).

4.2.2 Web Server - Protocol Negotiation

After you select Integrated Windows authentication for IIS, the Web Server protocol negotiation must be set to Negotiate, NTLM allowing Kerberos authentication with potential fallback to NTLM for non-Kerberos capable devices.
There might be scenarios where this is changed and Kerberos Authentication is disabled or fallback to NTLM does not work. If this is unknown, you can run the following command in C:\inetpub\AdminScripts to enable it:

```plaintext
C:\inetpub\AdminScripts>cscript adsutil.vbs
Microsoft (R) Windows Script Host Version 5.6
Copyright (C) Microsoft Corporation 1996-2001. All rights reserved.

Usage:
    ADSUTIL.VBS <cmd> [<path> [optional]]

Description:
IIS administration utility that enables the configuration of metabase properties.

Supported Commands:
    GET, SET, ENUM, DELETE, CREATE, COPY,
    APPCREATEINPROC, APPCREATEOUTPROC, APPCREATEPOOLPROC, APPDELETE, APPUNLOAD, APPGETSTATUS

Samples:
    adsutil.vbs GET W3SVC/1/ServerBindings
    adsutil.vbs SET W3SVC/1/ServerBindings "":81:"
    adsutil.vbs CREATE W3SVC/1/Root/MyVirtualDir "IisWebVirtualDir"
    adsutil.vbs START_SERVER W3SVC/1
    adsutil.vbs ENUM /P W3SVC

For Extended Help type:
    adsutil.vbs HELP

C:\inetpub\AdminScripts>cscript adsutil.vbs set w3svc/1/root/NTAuthenticationProviders "Negotiate,NTLM"
Microsoft (R) Windows Script Host Version 5.6
Copyright (C) Microsoft Corporation 1996-2001. All rights reserved.

NTAuthenticationProviders : <STRING> "Negotiate,NTLM"

C:\inetpub\AdminScripts>cscript adsutil.vbs get w3svc/1/root/NTAuthenticationProviders
Microsoft (R) Windows Script Host Version 5.6
Copyright (C) Microsoft Corporation 1996-2001. All rights reserved.

NTAuthenticationProviders : <STRING> "Negotiate,NTLM"
```

cscript adsutil.vbs get w3svc/WebSite/root/NTAuthenticationProviders

In this command, Website is a placeholder for the ID number of the Web site. The ID number of the default Web site is 1.
In the preceding screen shot, the Authentication Protocol Negotiation is set to the required Negotiate, NTLM. If the returned string was different like (STRING) "NTLM", only NTLM is enabled.

To set this back to its default, run the following command:
cscript adsutil.vbs set w3svc/1/root/NTAuthenticationProviders "Negotiate,NTLM"

4.3 Setting up NetScaler for Constrained Delegation

4.3.1 Create Backend Servers and Services
add server sph07 sph07.example.com
Or
add server sph07 10.217.28.93

Note: Here sph07.example.com resembles the backend server name, which should match the backend server real name.

add service s1 sph07 HTTP 80
4.3.2 Create TM and Authentication Virtual Server
There are new commands introduced for the NetScaler appliance to specify domain or realm for a LB or CS virtual server when you define the virtual server. These parameters are optional.

```
add lb vserver accesslb1 HTTP 10.217.28.20 80
Create Authentication Virtual Server
add authentication vserver auth1 SSL 10.217.28.20 443
set authentication vserver auth1 –authenticationdomain Example.COM
```

4.3.3 Create KcdAccount
KcdAccount is an entity which consists of the configuration necessary for doing delegation or impersonation. With delegation, configured user would get tickets on behalf of actual user for a service. Impersonation is used in cases where user password is available to NetScaler, and therefore delegation is not necessary. With impersonation, tickets will be obtained using actual user’s credentials.
There are multiple ways of creating a KcdAccount.

4.3.3.1 Create KcdAccount using Keytab file

To extract SPN from the keytab file use Kcdaccount, NetScaler appliance reads the keytab file and extracts SPN listed from the keytab file.

```
add kcdaccount kcdaccount1 –keytab kcdvserver.keytab
Or
add kcdaccount kcdaccount1 –keytab /nsconfig/krb/kcdvserver.keytab
```

**Note:** You must copy the Kcdvserver.keytab file to the /nsconfig/krb/ directory. If the file is not found in the /nsconfig/krb directory, the NetScaler appliance rejects it.

```
show kcdAccount kcdaccount2
1)   KCD Account : kcdaccount2
     Keytab : /nsconfig/krb/kcdvserver.keytab
```
SPN: host/kcdvserver.example.com@example.COM
Done

4.3.3.2 Create KcdAccount manually with password

Alternatively, KCDAccount can be created by giving the delegated username and password

`add kcdaccount kcdaccount1 --delegatedUser root --kcdPassword secret --realmStr TEST.COM`
where, delegatedUser is the user account to which services/users are delegated. kcdPassword is the password for this delegatedUser account. realmStr is the realm of the delegateduser, and can be omitted if delegatedUser name is in UPN (user@domain) format. If UPN is given, realm is taken from the domain portion of delegatedUser name.

Another example of creating kcdaccount using SPN:
`add aaa kcdAccount kcd4 --realmStr SQL2012.COM --delegatedUser "host/kcdvserver.sql2012.com" --kcdPassword freebsd`

NetScaler uses configured delegatedUser to get Ticket Granting Ticket (TGT). Active Directory accepts both ways of specifying names (that is UPN format and SPN format).

4.3.3.3 Create KcdAccount manually with certificate

This approach differs slightly from the preceding section (4.3.3.2). Instead of specifying delegatedUser’s password, one can use a client certificate which is mapped to this user in Active Directory. CA certificate is also required in this approach.
With this approach, NetScaler uses PKINIT using client’s certificate to obtain a TGT. This approach is highly secure as it does not involve specifying user’s password, and hence the password expiry problems.

`add aaa kcdAccount kcd4 --realmStr SQL2012.COM --delegatedUser "host/kcdvserver.sql2012.com" --usercert <string> --cacert <string>`

where usercert is the path to the user’s certificate file stored on NetScaler, and cacert is the path to the issuing authority’s certificate.

4.3.3.4 Create KcdAccount for Impersonation

Impersonation can be done if user’s password is available to NetScaler either through prior login. If session user’s name is in UPN format, user’s realm is obtained from UPN. Otherwise, username can be obtained from ssoDomain which is extracted from authentication to NetScaler or through configuring session profiles. However, kcdAccount needs a mandatory “realm” parameter which corresponds to the realm to which accessed service belongs.
This “realm” is also used as user’s realm if user’s realm cannot be obtained either through 
authentication to Netscaler or through session profiles.

```
add aaa kcdAccount kcd5 –realmStr SQL2012.COM
```

If the username during authentication is `user1@FOO.COM` or `FOO.COM\user1`, and the service
which is being accessed is server.fqdn.com, then FOO.COM is used as user’s realm and
SQL2012.COM as service’s realm.

If username during authentication is user1, and there’s a session profile to configure ssoDomain
to FOO.COM, this is equivalent to preceding case where FOO.COM is used for user’s realm, and
SQL2012.COM is used for service’s realm.

If username during authentication is user1 and ssoDomain is neither obtained through
authentication (either by user input or through external server) nor through session profile,
then SQL2012.COM is used both as user’s realm and also as service’s realm.

### 4.3.3.5 Create KcdAccount for Impersonation with user’s certificate

This is a special case of impersonation, where PKINIT is employed, along with Citrix Receiver, to
get TGT. In this approach NetScaler uses a proprietary protocol to communicate with the Citrix
Receiver over “back channel”. PKINIT involves data validation using client’s private key (which is
not available to NetScaler), and this happens over the back channel.

This configuration works only with Citrix Receiver as client as NetScaler and Citrix Receiver
negotiate PKINIT capabilities either during session creation or at runtime.

```
add aaa kcdAccount kcd6 –cacert <path to certificate>
```

where cacert is the path to the certificate authority certificate stored on NetScaler.

Based on the negotiation with Citrix Receiver, NetScaler determines whether user session is
PKINIT impersonation capable, and whether required information such as user’s public key is
already obtained.

### 4.3.5 Create Traffic/Session Profile

You can provide Kcdaccount in session_action or traffic_action, NetScaler Administrator can
select different rules. Rules are combinations of Expressions. Expressions are simple conditions,
such as a test for equality, applied to operands, such as a URL string or an IP address.
**Note:** When you specify kcdaccount in traffic/session profile with SSO: ON, NetScaler appliance completes a Kerberos constrained delegation to the backend servers. If KcdAccount is set to “none”, the NetScaler appliance does NTLM SSO.

**Setting in Session Profile**
add tm sessionAction myssO1 -SSO ON –kcdaccount kcdaccount1
add tm sessionPolicy myssO1 ns_true myssO1
bind authentication vserver auth1 -policy myssO1

**Setting in Traffic Profile**
add tm trafficAction mytraffic1 -SSO ON –kcdAccount kcdaccount1
add tm trafficPolicy myssO1 TRUE myssO1
bind lb vserver accesslb1 –policy myssO1 –priority 100

**Setting at Global Level**
set tm sessionparameter -sso on –kcdaccount kcdaccount1

**4.3.6 Enable Client Side Authentication (Form-based/401basic)**
Enable the authentication method that you want on the TM virtual server. The following are examples to enable 401 Basic and form-based authentication:

set lb vs accesslb1 –authentication on –authenticationhost authvs.example.com
Or
set lb vs accesslb1 –authn401 –authnvsname auth1
bind lb vserver accesslb1 s1

If there is any error associated with running the preceding commands, see Troubleshooting section 5.1.
5. Troubleshooting

5.1 Getting started

Netscaler’s Kerberos daemon, nskrb, is responsible for communicating to Active Directory on behalf of NetScaler. This daemon, along with serving as authentication daemon, also serves as a command line utility. This supports an extensible set of command line parameters. These options are described in section 5.2.2 below.

5.2.1 Ensure that nskrb daemons is running

nskrb daemon must be running all the time for KCD to work.

Run the following command at the BSD shell:

```
ps ax | grep nskrb
```

If you do not see nskrb daemon running, run the following command:

```
#/netscaler/nskrb nsauth
```

This operation runs nskrb as a daemon. For development purposes, you can also run nskrb as a regular process. In order to run nskrb as a normal process, issue the following command:

```
#/netscaler/nskrb nsauth nodae
```

5.2.2 Command line options for nskrb

nskrb, as mentioned in the preceding section, supports a variety of command line options. You can do operations such as reading and merging a keytab, inspecting or destroying cache, and also perform kcd operations through command line (as described in subsequent sections). The first argument to nskrb determines the top level direction of the arguments that follow.

At the top level, nskrb takes these command line arguments:

- `ktutil`: directive to operate on keytab files
- `klist`: to list keytab and ticket cache files
- `kdestroy`: to purge cache files
- `kinit`: to request a TGT using different options
- `kgetcred`: to request for TGS either via impersonation or through S4U2Self and S4U2Proxy

An easy way to get more information about these arguments is to issue `–help` or `–h` as the argument following these top level directives.
# /netscaler/nskrb ktutil -h
-k keytab to operate on, --keytab=keytab to operate on keytab
-v, --verbose verbose

#/netscaler/nskrb klist -h
     [-c string, --cache=string credential cache to list]
-k string, --keytab=string keytab file to parse
-f list flags
-t, --test test for having tickets
-T, --tokens display AFS tokens
-S, --v5 display v5 credential tokens
-A, --all-content List all caches with their content
-l, --list-all List all caches
-v, --verbose Verbose output
-i use default keytab
-z timestamp option for keytab
-K display keys for keytab
-e display encryption type
--hidden Verbose output

# /netscaler/nskrb kdestroy -h
     --credential=principal remove one credential
-c cache, --cache=cache cache to destroy
-A, --all destroy all caches
--no-unlog do not destroy tokens
--no-delete-v4 do not destroy v4 tickets

# /netscaler/nskrb kinit -h
     [--extra-addresses=addresses]... [-a addresses]... [--anonymous] [--request-pac] [-- password-file=string] [--password=string] [--canonicalize]
Usage:

```
# /netscaler/nskrb kgetcred -h
```

```
```

```
```

```
--afslog
```

```
get afs tokens
```

```
-c cachename, --cache=cachename
```

```
get credentials cache
```

```
--no-forwardable
```

```
get tickets not forwardable
```

```
-f
```

```
get forwardable tickets
```

```
-t keytabname, --keytab=keytabname
```

```
keytab to use
```

```
-l time, --lifetime=time
```

```
get lifetime of tickets
```

```
-p, --provisible
```

```
get proxiable tickets
```

```
-R, --renew
```

```
renew TGT
```

```
--renewable
```

```
get renewable tickets
```

```
-r time, --renewable-life=time
```

```
renewable lifetime of tickets
```

```
-S principal, --server=principal
```

```
server to get ticket for
```

```
-s time, --start-time=time
```

```
when ticket gets valid
```

```
-k, --use-keytab
```

```
get key from keytab
```

```
-v, --validate
```

```
validate TGT
```

```
-e enctypes, --enctypes=enctypes
```

```
encryption types to use
```

```
--fcache-version=integer
```

```
file cache version to create
```

```
-A, --no-addresses
```

```
request a ticket with no addresses
```

```
-a addresses, --extra-addresses=addresses
```

```
include these extra addresses
```

```
--anonymous
```

```
request an anonymous ticket
```

```
--request-pac
```

```
request a Windows PAC
```

```
--password-file=string
```

```
read the password from a file
```

```
--password=string
```

```
principal's password
```

```
--canonicalize
```

```
canonicalize client principal
```

```
--enterprise
```

```
parse principal as a KRB5-NT-ENTERPRISE name
```

```
--pk-enterprise
```

```
use enterprise name from certificate
```

```
-C id, --pk-user=id
```

```
principal's public/private/certificate identifier
```

```
-D directory, --x509-anchors=directory
```

```
directory with CA certificates
```

```
--pkinit
```

```
use pkinit without loading client cert
```

```
--pk-use-enckey
```

```
Use RSA encrypted reply (instead of DH)
```

```
--ntlm-domain=domain
```

```
```
NTLM domain
```

```
--no-change-default
```

```
switch the default cache to the new credentials cache
```

```
--ok-as-delegate
```

```
honor ok-as-delegate on tickets
```

```
--use-referrals
```

```
only use referrals, no dns canalisation
```

```
--windows
```

```
get windows behavior
```

```
# /netscaler/nskrb kgetcred -h
```

```
```

```
```

```
26
```

```
```
-c cache, --cache=cache    credential cache to use
--out-cache=cache         credential cache to store credential in
--delegation-credential-cache=cache where to find the ticket use for delegation
--canonicalize            canonicalize the principal
--forwardable             forwardable ticket requested
-e enctype, --enctype=enctype encryption type to use
--impersonate=principal   client to impersonate

5.3 How do I know if KCD is working?
The following is a sample aaad.debug log when you access LB virtual server enabled with KCD
and when KCD is successful:

lwagent.c[1198]: lw_start_get_s4u Call /opt/likewise/bin/kinit -k -t
/nsconfig/krb/kcdvserver.keytab -f 'host/kcdvserver.example.com@example.COM'
Sun Jun 10 20:34:20 2012
lwagent.c[993]: get_s4u Get S4U2Proxy for:
User: abc@example.COM; Target: http/sph07.example.com;
Keytab: /nsconfig/krb/kcdvserver.keytab; spnego:0

Sun Jun 10 20:34:20 2012
lwagent.c[1015]: get_s4u Protocol transition tests follow

Sun Jun 10 20:34:20 2012
lwagent.c[1016]: get_s4u ----------------------------

Sun Jun 10 20:34:20 2012
lwagent.c[1044]: get_s4u

The following is a trace at the Active Directory:
Pkt.514 shows Ticket request from NetScaler appliance to Active Directory and asking for a
Ticket.
Pkt.515 shows Ticket is granted.
Pkt.520 S4U2SELF Request for host SPN: kcdvserver.example.com
Pkt.530 Constrained Delegation request for backend service http\sph07.example.com

NSIP: 10.217.28.2 AD: 10.217.28.93
Kerberos AS-REQ

**MSG Type:** AS-REQ(0)

**B pdata:** PA-ENC-TIMESTAMP unknown:149

- **Type:** PA-ENC-TIMESTAMP 2
- **Type:** unknown(49)

**KDC_REQ_BODY**

**Padding:** 0

- **KDCOptions:** 400000(Forgivable, Renewable OK)
- **Client Name** Principal: `host/kcdvserver` I.com
  - **Name-type:** Principal 1
  - **Name:** host
  - **Realm:** FDM.COM
- **Server Name** and Instance krbtgt/*COM
  - **from:** 2012-06-11 03:37:42(UTC)
  - **till:** 2012-06-12 03:37:42(UTC)
  - **Nonce:** 76942123
- **Encryption Types:** RC4-hmac

**Otherlet H** src: 12:8.82:8:56:10:38:56:0:10:8:56:0:df

**User Datagram Protocol, src:** 10.217.28.2 10.217.28.93 (Ost: 10.217.28.93)

**Kerberos AS-REP**

**MSG Type:** AS-REP(1)

**B pdata:** PA-ENC-TIMESTAMP unknown:149

- **Type:** PA-ENC-TIMESTAMP 2
- **Type:** unknown(49)

**KDC_REP_BODY**

- **Padding:** 0

  - **KDCOptions:** 400000(Forgivable, Renewable OK)
  - **Client Name** Principal: `host/kcdvserver` I.com
    - **Name-type:** Principal 1
    - **Name:** host
    - **Realm:** FDM.COM
  - **Server Name** and Instance krbtgt/*COM
    - **from:** 2012-06-11 03:37:42(UTC)
    - **till:** 2012-06-12 03:37:42(UTC)
    - **Nonce:** 76942123
  - **Encryption Types:** RC4-hmac
5.4 Verifying if NetScaler appliance can do KCD

Successful KCD requires obtaining TGT for delegated user, S4U2Self on behalf of user and S4U2Proxy for obtaining service ticket. However, if impersonation is desired, which is the easiest, you need only TGT and TGS. These are described as follows.

5.4.1 Impersonation

5.4.1.1 Obtaining a TGT

Run the following command from the shell prompt of the NetScaler appliance:

```
# /netscaler/nskrb kinit --password=secret svc_kcd1@NSI-TEST.COM
```

In the preceding command, password can be omitted. Then, you would be prompted for password

```
# /netscaler/nskrb kinit svc_kcd1@NSI-TEST.COM
svc_kcd1@NSI-TEST.COM's Password: ...
```

If the preceding command is successful, TGT is obtained and is written into the default cachefile /tmp/krb5cc_0

```
# /netscaler/nskrb klist -c /tmp/krb5cc_0
Credentials cache: FILE:/tmp/krb5cc_0
    Principal: svc_kcd1@NSI-TEST.COM
    Issued   Expires    Principal
    Jun 12 23:35:09 2013  Jun 13 09:35:09 2013  krbtgt/NSI-TEST.COM@NSI-TEST.COM
```

5.4.1.2 Obtaining a TGT

Next step is obtaining a TGS, that is, the service ticket. Issue the following command to get TGS:

```
# /netscaler/nskrb kgetcred -c /tmp/krb5cc_0 http/nsi-dc1-2008.nsi-test.com@NSI-TEST.COM
```

If the preceding command is successful, TGS is obtained and stored in /tmp/krb5cc_0. This can be viewed by the command issued earlier.

```
# /netscaler/nskrb klist -c /tmp/krb5cc_0
Credentials cache: FILE:/tmp/krb5cc_0
    Principal: svc_kcd1@NSI-TEST.COM
    Issued   Expires    Principal
    Jul 8 22:17:13 2013  Jul 9 08:17:13 2013  krbtgt/NSI-TEST.COM@NSI-TEST.COM
```
5.4.2 Constrained Delegation
Constrained delegation, as described in the preceding involves 3 steps. The first one is similar to getting TGT for user. In this case, however, it is obtained for the delegated user.

5.4.2.1 Obtaining TGT for delegated user
In order to obtain TGT for delegated user, issue the following command:

```
# /netscaler/nskrb kinit --password=secret svc_kcd1@NSI-TEST.COM
```

This command, if successful, writes a TGT into the default cache file.

5.4.2.2 Obtaining S4U2Self on behalf of user
In the following command, try to obtain a ticket that can be forwarded on behalf of user “user1” using the TGT obtained in the previous command:

```
# /netscaler/nskrb kgetcred -c /tmp/krb5cc_0 --out-cache=/tmp/imper_cache --impersonate=user1@NSI-TEST.COM svc_kcd1@NSI-TEST.COM
```

In the preceding command, ticket is obtained for user1 using the delegated user svc_kcd1 and cached TGT in /tmp/krb5cc_0. The ticket is stored in output cache /tmp/imper_cache

If output cache is not specified, it writes into /tmp/krb5cc_0. These credentials can be viewed with klist command mentioned earlier.

```
/netscaler/nskrb klist -c /tmp/imper_cache
Credentials cache: FILE:/tmp/imper_cache
Principal: user1@NSI-TEST.COM

Issued       Expires       Principal
Jun 12 23:38:10 2013 Jun 13 09:35:09 2013 svc_kcd1@NSI-TEST.COM
```

5.4.2.3 Obtaining S4U2Proxy, the TGS for target service

The last step in obtaining TGS ticket is S4UProxy is described by following command:

```
# /netscaler/nskrb kgetcred --delegation-credential=/tmp/imper_cache http/hsni-dc1-2008.nsitest.com --outcache=/tmp/kcd_cache
```

In the preceding command, the delegation credentials obtained in previous step (the S4U2Self) are used , and request for TGS for the service “http/hsni-dc1-2008.nsitest.com”. If this
command is successful, TGS is written into outcache /tmp/kcd_cache specified in the preceding command.
The ticket can be viewed using the same klist command:

/netscaler/nskrb klist -c /tmp/krb5cc_0
Credentials cache: FILE:/tmp/krb5cc_0
    Principal: svc_kcd1@NSI-TEST.COM

<table>
<thead>
<tr>
<th>Issued</th>
<th>Expires</th>
<th>Principal</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jun 12 23:35:09 2013</td>
<td>Jun 13 09:35:09 2013</td>
<td>krbtgt/NSI-TEST.COM@NSI-TEST.COM</td>
</tr>
</tbody>
</table>
Fix: Ensure that you enable the **Use Any Authentication Protocol** option.

"Server not found in Kerberos database" when you run `/opt/likewise/bin/t_s4u user@example.COM http/nsi-dc1-2008.example.com /nsconfig/krb/kcdvserver.keytab` command.

**Fix:** Check `/etc/krb5.conf`, if it is pointing to the wrong domain, take a backup of the file and change it to correct domain and save it.

- `/opt/likewise/bin/t_s4u user@example.COM http/sharepoint.example.com /nsconfig/krb/kcdvserver.keytab` Protocol transition tests follow

-----------------------------

`gss_acquire_cred_impersonate_name`: Unspecified GSS failure. Minor code may provide more information

`gss_acquire_cred_impersonate_name`: Client not found in Kerberos database
**Fix:** Verify if ‘Do not require kerberos preauthentication’ option is enabled or disabled. You must disable this option.

If the procedure to join the appliance to the domain fails or takes longer time, as in the case of Cross Domain Forests, add nameserver to the /etc/hosts file.
For example; 10.217.22.241 nsi-dc1-2008.example.com nsi-dc1-2008

And add KDC to the /etc/resolve.conf file
For example; search example.com
nameserver 10.217.22.241
6. Firewall Ports required to be open for KCD communication

<table>
<thead>
<tr>
<th>Port</th>
<th>Protocol</th>
<th>Use</th>
</tr>
</thead>
<tbody>
<tr>
<td>53</td>
<td>UDP/TCP</td>
<td>DNS</td>
</tr>
<tr>
<td>88</td>
<td>UDP/TCP</td>
<td>Kerberos</td>
</tr>
<tr>
<td>123</td>
<td>UDP</td>
<td>NTP</td>
</tr>
<tr>
<td>135</td>
<td>TCP</td>
<td>RPC Endpoint Mapper</td>
</tr>
<tr>
<td>137</td>
<td>UDP</td>
<td>NetBIOS Name Service</td>
</tr>
<tr>
<td>139</td>
<td>TCP</td>
<td>NetBIOS Session (SMB)</td>
</tr>
<tr>
<td>389</td>
<td>UDP/TCP</td>
<td>LDAP</td>
</tr>
<tr>
<td>445</td>
<td>TCP</td>
<td>SMB over TCP</td>
</tr>
<tr>
<td>464</td>
<td>UDP/TCP</td>
<td>Machine password changes (typically after 30 days)</td>
</tr>
<tr>
<td>3268</td>
<td>TCP</td>
<td>Global Catalog Search</td>
</tr>
</tbody>
</table>
7. Reference


8. Addendum

8.1 Protocol Transition (S4U2Self)
The protocol transition extension allows a service to obtain a Kerberos service ticket to the service on behalf of a user or proxy without requiring the user or proxy to be part of the Kerberos domain, or restricts to use Internet Explorer. You do not require a user credentials for the transition. Applications might be switched into Kerberos even though you complete the actual authentication by using another authentication method, such as HTTP Basic, form-based, NTLM, Radius, LDAP, SAML, RSA SecureID, PKI, or Certificates and other OTP systems.

8.2 Constrained Delegation (S4U2Proxy)
The constrained delegation extension allows a service to obtain service tickets under the delegated users identity to a subset of other services after you present a service ticket that you get either through the TGS_REQ protocol, as defined in IETF RFC 1510, or in the protocol transition extension.

The constrained delegation extension is available in Windows Server 2003 to address limitations in the Windows 2000 implementation of Kerberos delegation. In the Windows 2000 Kerberos delegation model, the Kerberos Key Distribution Center (KDC) does not limit the scope of services to which you can delegate a Kerberos principal identity. That is, after a service account is trusted for delegation, it can request service tickets on behalf of an authenticated user to any other service accounts. This delegation method does not provide precise mechanisms for an application to specify a subset of service accounts that it determines to be trustworthy for delegation. Applications are exposed to broader impersonation risks that might span across resource domains that have different levels of security policy requirements. Some of the security policies might not be as strict as the applications security requirements. From the domain administrator point of view, it is risky to enable unconstrained Kerberos delegation in the enterprise, because you cannot exclude untrusted servers from participating in delegation. With constrained delegation, domain administrators can configure service accounts so that they delegate only to specific sets of service accounts.
References
For more information about RFC 1510, see the IETF Web site (http://www.ietf.org).

For more information about Kerberos and the two extensions, see
http://www.microsoft.com/msj/0899/kerberos/kerberos.aspx and
http://support.citrix.com/article/CTX135587