One Way Trust - MIT KDC to Active Directory

[**emaxwell**](https://community.hortonworks.com/users/98/emaxwell.html)  created · Oct 03, 2016 at 10:18 PM · [**edited**](https://community.hortonworks.com/revisions/59635.html) · Mar 16, 2017 at 03:18 PM

17

**Short Description:**

How to create a one-way trust from an MIT KDC to Active Directory

**Article**

**One Way Trust - MIT KDC to Active Directory**

Many security environments have strict policies on allowing administrative access to Active Directory. Some performance issues can also require that Hadoop cluster principals for Kerberos are not created directly in AD. To aid in these situations, it may be preferable to use a local MIT KDC in the Hadoop cluster to manage service principals while using a one-way trust to allow AD users to utilze the Hadoop environment. This tutorial describes the steps necessary to create such a trust.

The following assumptions are made for this tutorial:

* An existing HDP cluster
* Cluster has Kerberos enabled with an MIT KDC
* The MIT KDC realm name is [HDP.HORTONWORKS.COM](http://hdp.hortonworks.com/)
* The MIT KDC server is named [kdc-server.hdp.hortonworks.com](http://kdc-server.hdp.hortonworks.com/)
* The AD domain/realm is [AD.HORTONWORKS.COM](http://ad.hortonworks.com/)

**Step 1: Configure the Trust in Active Directory**

**Create a KDC definition in Active Directory**

On the AD server, run a command window ***with Administrator privileges*** and create a definition for the KDC of the MIT realm:

1. ksetup /addkdc HDP.HORTONWORKS.COM kdc-server.hdp.hortonworks.com

**Create the Trust in Active Directory**

On the AD server, create an entry for the one-way trust. The password used here will be used later in the MIT KDC configuration of the trust:

1. netdom trust HDP.HORTONWORKS.COM /Domain:AD.HORTONWORKS.COM /add /realm /passwordt:BadPass#1

**Step 2: Configure Encryption Types**

In order for the MIT realm to trust tickets generated by the AD KDC, the encryption types between both KDCs must be compatible. This means that there must be at least one encryption type that is accepted by both the AD server as well as the MIT KDC server.

**Specify Encryption Types in Active Directory**

On the AD server, specify which encryption types are acceptible for communication with the MIT realm. Multiple supported encryption types are specified on the command line separated by spaces:

1. ksetup /SetEncTypeAttr HDP.HORTONWORKS.COM AES256-CTS-HMAC-SHA1-96 AES128-CTS-HMAC-SHA1-96 RC4-HMAC-MD5 DES-CBC-MD5 DES-CBC-CRC

**Specify Encryption Types in MIT KDC**

By default, all of the encryption types are accepted by the MIT KDC. If security concerns require that the encryption types be limited, this is done in the /etc/krb5.conf file:

1. [libdefaults]
2. permitted\_enctypes = aes256-cts-hmac-sha1-96 aes128-cts-hmac-sha1-96 arcfour-hmac-md5 des-cbc-crc des-cbc-md5

**Step 3: Enable Trust in MIT KDC**

To complete the trust configuration, the trust must be added to the MIT KDC.

**Add Domain to MIT KDC Configuration**

In the /etc/krb5.conf file, add the AD domain to the [realms] section:

1. [realms]
2. HDP.HORTONWORKS.COM = {
3. kdc = kdc-server.hortonworks.com
4. admin\_server = kdc-server.hortonworks.com
5. default\_domain = hdp.hortonworks.com
6. }
7. AD.HORTONWORKS.COM = {
8. kdc = ad-server.hortonworks.com
9. admin\_server = ad-server.hortonworks.com
10. default\_domain = ad.hortonworks.com}

**Create Trust User**

In order for the trust to work, a principal combining the realms in the trust must be created in the MIT KDC. The password for this user must be the same as the password used to create the trust on the AD server:

1. kinit admin/admin@HDP.HORTONWORKS.COM
2. kadmin -q "addprinc krbtgt/HDP.HORTONWORKS.COM@AD.HORTONWORKS.COM"

**Step 4: Configure AUTH\_TO\_LOCAL**

The Hadoop auth\_to\_local parameter must be changed to properly convert user principals from the AD domain to usable usernames in the Hadoop cluster. In Ambari, add the following rules to the auth\_to\_local variable in HDFS -> Configs -> Advanced -> Advanced core-site.xml -> hadoop.security.auth\_to\_local

1. RULE:[1:$1@$0](^.\*@AD\.HORTONWORKS\.COM$)s/^(.\*)@AD\.HORTONWORKS\.COM$/$1/g
2. RULE:[2:$1@$0](^.\*@AD\.HORTONWORKS\.COM$)s/^(.\*)@AD\.HORTONWORKS\.COM$/$1/g