Technical Report

Secure Unified Authentication with NetApp Storage Systems

Kerberos, NFSv4, and LDAP for User Authentication over NFS (with a Focus on Clustered Data ONTAP)

Justin Parisi, Bikash Roy Choudhury, NetApp
October 2013 | TR-4073

Abstract

This document explains how to configure NetApp® storage systems with the clustered Data ONTAP® operating system for use with UNIX®-based Kerberos version 5 (krb5) servers for NFS storage authentication and Microsoft® Windows Server® Active Directory® (AD) as the key distribution center (KDC) and Lightweight Directory Access Protocol (LDAP) identity provider. This document intends to help customers successfully integrate their NetApp storage systems with Kerberos version 5 to achieve secure NFS storage authentication. It briefly describes the basic terminology and concepts used in Kerberos version 5 authentication systems. This document does not include MIT Kerberos setup and configuration.

TABLE OF CONTENTS

1 Introduction 6

1.1 Overview 6

1.2 Intended Audience 6

2 Kerberos Overview 7

2.1 Kerberos Terminology 7

2.2 General Kerberized NFS Setup 8

2.3 How Kerberos Authentication Works 9

2.4 Client-Side Kerberos Commands 11

2.5 Client-Side Kerberos Files 12

3 Benefits of Using Kerberized NFS 13

4 Setting Up LDAP and Kerberized NFS with Microsoft Windows AD 13

4.1 Setting Up Kerberized NFS 14

4.1.1 Configuring the Clustered Data ONTAP System for Kerberos 14

4.1.2 Enabling DNS 14

4.1.3 Configuring NFS 15

4.1.4 Configuring a Kerberos Realm 15

4.1.5 Creating a CIFS Server (Optional) 15

4.1.6 Configuring Export Policies and Rules 15

4.1.7 Creating and Mounting Data Volumes 15

4.1.8 Enabling Kerberos on a Data LIF 15

4.1.9 Creating and Verifying Name Mapping for the NFS SPN 16

4.1.10 What Is SecD? 16

4.1.11 Name Mapping 17

4.1.12 Other Considerations 19

4.1.13 Configuring the Domain Controller for Kerberos 19

4.1.14 Adding an NFS Client to Windows Active Directory DNS 19

4.1.15 Adding the SVM Data LIFs to Windows Active Directory DNS 22

4.1.16 Adding SRV Records 50

4.1.17 Creating Principals/Keytab Files in Active Directory 57

4.1.18 Using Domain Trusts 71

4.1.19 Domain Controller Redundancy and Replication 71

4.1.20 Configuring the NFS Clients to Use Kerberos 73

4.1.21 Solaris Kerberos Configuration 73

4.1.22 Configuring Linux Clients 74

4.2 Setting Up LDAP 79
4.2.1 Overview of LDAP 79
4.2.2 Active Directory LDAP via SSSD Benefits 80
4.2.3 How SSSD Interacts with Active Directory 81
4.2.4 Configuring the Domain Controller as an LDAP Server 81
4.2.5 Configuring the Client to Use LDAP 88
4.2.6 RHEL/CentOS/Fedora Client Configuration 89
4.2.7 SUSE/SLES Client Configuration 91
4.2.8 Ubuntu Client Configuration 94
4.2.9 `sssd.conf` File Example 96
4.2.10 SSSD Configuration File Options 97
4.2.11 Configuring Solaris to Use LDAP 99
4.2.12 Configuring the clustered Data ONTAP System to Use LDAP 102
4.2.13 LDAP Schemas 102
4.2.14 Viewing SecD Logs 104
4.2.15 Creating a Custom LDAP Schema 105
4.2.16 LDAP Clients 105
4.2.17 LDAP Configuration 107
4.2.18 LDAPS (LDAP over SSL) 107
4.2.19 SVM Configuration 109
4.3 Setting Up NFSv4 109
4.3.1 Overview of NFSv4 109
4.3.2 NFSv4 Benefits 110
4.3.3 Configuring the Clustered Data ONTAP System for NFSv4.x 110
4.3.4 Configuring the Domain Controller for NFSv4.x 111
4.3.5 Configuring the NFS Clients for NFSv4.x 111

5 Quick Step Setup Guides 112

5.1 Quick Step Setup 112

Appendix **137**

References **156**

Version History **157**

Acknowledgements **157**

LIST OF TABLES

Table 1) Client-side Kerberos commands. 11

Table 2) Client-side Kerberos files. 12

Table 3) Adding NFS client to Windows DNS (GUI). 19

Table 4) Adding NFS client to Windows DNS (dnscmd). 22

Table 5) Setting up On-Box DNS load balancing on the cluster 26

Table 6) Setting up delegations. 27

Table 7) Setting up stub zones. 32

Table 8) Setting up primary zones. 39

Table 9) Setting up conditional forwarders – Windows 2008. 47

Table 10) Setting up conditional forwarders – Windows 2003. 48

Table 11) Creating SRV records for Kerberos-Master. 51

Table 12) Creating SRV records for Kerberos-Master (dnscmd). 54

Table 13) Allowing DES encryption types in Windows 2008. 55

Table 14) Creating machine accounts in Active Directory (GUI). 57

Table 15) Creating machine accounts in Active Directory (dsadd). 59

Table 16) Creating machine accounts in Active Directory (PowerShell). 59

Table 17) Modifying the NFS server machine account to use/support DES_CBC_MD5 (Attributes Editor). 61

Table 18) Modifying the NFS server machine account to use/support DES_CBC_MD5 (ADSIedit). 63

Table 19) Modifying the NFS server machine account to use/support DES_CBC_MD5 (import using ldifde). 66

Table 20) Creating machine accounts in Active Directory (PowerShell). 67

Table 21) Creating a keytab file. 68

Table 22) Setting the host name. 74

Table 23) Allowing secure NFS. 76

Table 24) Managing Kerberos services. 78

Table 25) Active Directory schema extensions per Windows version. 82

Table 26) Setting UID/GID in Active Directory LDAP (GUI). 83

Table 27) Setting UID/GID in Active Directory LDAP (`ldifde`). 85

Table 28) Difference in schema attributes before/after extending the schema using `ldifde`. 86

Table 29) Mapping users with LDAP 87

Table 30) /etc/sssd/sssd.conf file options. 97

Table 31) Default schemas available in clustered Data ONTAP. 103

Table 32) SecD scope definitions. 104
Table 33) Configuring LDAP over SSL in Clustered Data ONTAP 107
Table 34) Configuring the clustered Data ONTAP system for NFSv4.x (CLI). 110
Table 35) Services for NFSv4. 112
Table 36) Enctypes 137
Table 37) Valid userAccountControl attribute values. 139
Table 38) Valid msDS-SupportedEncryptionTypes attribute values. 141
Table 39) Kerberos packets. 141
Table 40) Kerberos errors from Kerberos errors in network captures. 142
Table 41) Kerberos terminology from CentOS.org and IBM.com. 142
Table 42) Configuring MIT Kerberos 144
Table 43) Common mount issues with Kerberized NFS. 148
Table 44) Common read/write issues with Kerberized NFS. 148
Table 45) Pre-setup Steps. 154

LIST OF FIGURES

Figure 1) Kerberos workflow between client, KDC, and NFS server on NetApp storage. 9
Figure 4) DES enabled. 56
Figure 5) Default value. 57

1 Introduction

Kerberos is a protocol, defined in RFC 1510, designed to provide strong authentication within a client/server environment. The basis of the protocol is a shared secret key cryptology system. MIT created the Kerberos authentication model in the early 1980s as a way of providing secure authentication in a networked environment.

Kerberos uses shared key encryption to ensure the confidentiality (no inappropriate access to data) of the data and uses hashing techniques to ensure the integrity of the data (no one modified the data who isn't allowed to modify the data).

Kerberos has been gaining acceptance as a secure network-based authentication service. Many companies are replacing local system authentication with Kerberos authentication because of its security and centralized management.

Microsoft implemented Kerberos as the primary authentication service in Windows® Active Directory starting in Windows 2000. The Microsoft Kerberos implementation is standards based, resulting in the ability to use Microsoft Active Directory Kerberos for UNIX and Linux® Kerberos authentication. This provides a method to unify authentication on networks based on UNIX and Windows. Using an existing Microsoft Windows Active Directory implementation as the KDC makes sense from an ease and cost perspective.

With the NetApp multiprotocol storage platform, through which clients based on UNIX or Windows can access data using CIFS or NFS, it is crucial to provide the ability to use standard network services for authentication and for identity storage.

NetApp storage systems fully support Kerberos 5 and Microsoft Active Directory–based Kerberos. However, Kerberos 5i (integrity) and Kerberos 5p (privacy) are currently not supported in clustered Data ONTAP.

Kerberos to KDCs over IPv6 are also currently not supported.

1.1 Overview

The following document covers the use of System Security Services Daemon (SSSD) LDAP on various Linux clients, leveraging secure technologies such as Kerberos/GSSAPI and NFSv4. This document is useful in multiprotocol environments in which a Windows Active Directory domain is in place, since this allows easy centralized management for all environments. The following environment was used.

- **Windows Active Directory domain with two domain controllers running Windows 2008 R2**
  Both domain controllers (DCs) were installed from scratch. Only the following roles were installed on each:
  - DNS
  - Identity Management
- **Various Linux clients**
  These clients were built from scratch and had no preexisting configuration.
- **Clustered Data ONTAP 8.2 Storage Virtual Machine (SVM)**
  The SVM that was used had only data LIFs and data volumes and was created using the vserver setup command. No configuration of protocol services had been done.

1.2 Intended Audience

This document will help administrators and architects implement Kerberized NFS for strong NFS authentication in their existing Microsoft Windows Active Directory environments leveraging clustered Data ONTAP for NAS storage. A working knowledge of NFS, Kerberos, and Windows Active Directory, as well as administrator access to these environments, is assumed.

Best Practice
There are "Quick Step Setup" guides at the end of this document, as well as customizable setup script examples if you are interested only in basic setup. However, NetApp highly recommends that you review and understand the concepts in this document before attempting to set up Kerberized NFS. After reviewing, use the Quick Step Setup guides for the actual setup procedures.

Note: This document contains advanced and diag-level commands; exercise caution when using them. If you have questions or concerns about using these commands, contact NetApp Support for assistance.

2 Kerberos Overview

2.1 Kerberos Terminology

Key Distribution Center

The key distribution center (KDC) is the authentication server that includes the ticket-granting service (TGS) and the authentication service (AS). KDC, AS, and TGS are used interchangeably. In Microsoft environments, an Active Directory domain controller is a KDC.

Realm (or Kerberos Realm)

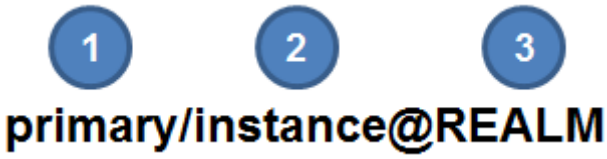
A realm (or Kerberos realm) can use any ASCII string. The convention is to use the domain name in uppercase; for example, domain.com becomes the realm DOMAIN.COM.

Administratively, each principal@REALM is unique. To avoid a single point of failure, each realm can have numerous KDCs sharing the same database (principals and their passwords) and have the same KDC master keys. Microsoft Windows Active Directory does this natively by way of domain replication, which takes place every 15 minutes by default.

Principal

The term principal refers to every entity within a Kerberos database. Users, computers, and services running on a client are all principals. Every principal is unique within the Kerberos database and is defined by its distinguished name. A principal can be a user principal (UPN) or a service principal (SPN).

A principal name has three parts:



The primary:

This can be a user or a service. The primary can be a service such as the "nfs" service. It can also be the special service "host," which signifies that this service principal is set up to provide various network services such as ftp, rsh, nfs, and so on.

The instance:

This is optional in case of a user. A user can have more than one principal. For example, Fred might have a principal that is for everyday use and a principal that allows privileged use such as a sysadmin account. The instance is required for service principals and designates the FQDN of the host providing the service.

The realm:

A Kerberos realm is the set of Kerberos principals that are registered within a Kerberos server. By convention, usually the realm name is the same as the DNS name, but it is converted to capital letters. Capital letters are not obligatory, but the convention allows easy distinction between the DNS name and the realm name.

Examples:

user@DOMAIN.COM

user/admin@DOMAIN.COM

host/host.domain.com@DOMAIN.COM

root/host.domain.com@DOMAIN.COM

nfs/host.domain.com@DOMAIN.COM

Tickets

A ticket is a temporary set of credentials that verifies the identity of a principal for a service and contains the session key. A ticket can be a service or application ticket or a ticket-granting ticket (TGT).

Secret Keys

Kerberos uses a symmetric key system in which the secret key is used for both encryption and decryption.

The secret key is generated from the principal's Kerberos password with a one-way hash function. The KDC stores the password for each principal and can thus generate the principal's secret key. For users requesting a Kerberos service, the secret key is typically derived from a password presented to the kinit program. Service and daemon principals typically don't use a password; instead, the result of the one-way hash function is stored in a keytab.

Keytab

A keytab contains a list of principals and their secret keys. The secret keys in a keytab are often created by using a random password and are mostly used for service or daemon principals.

2.2 General Kerberized NFS Setup

There are four main components in a Kerberized NFS environment:

- KDC (Microsoft Windows Active Directory in this document)
- DNS
- NetApp storage
- Clients (any flavor of UNIX or Linux)

2.3 How Kerberos Authentication Works

Kerberos is an authentication protocol that uses a secret key to validate the identity of principals.

Windows Active Directory maintains a database of principals and their Kerberos passwords. The secret key is nothing but the principal's password converted into a cryptographic key format. In the case of NFS servers and clients, the secret key can be generated using a random password and is stored in a keytab on the NFS server or client.

In Kerberos, the secret key is considered as proof of unique identity. Therefore, the KDC can be trusted to authenticate any principal to any other principal, such as authenticating an NFS client SPN to an NFS server SPN at mount or authenticating a user principal to an NFS server SPN for user access to the NFS mount point. Kerberos does not send clear-text passwords for authentication across the wire.

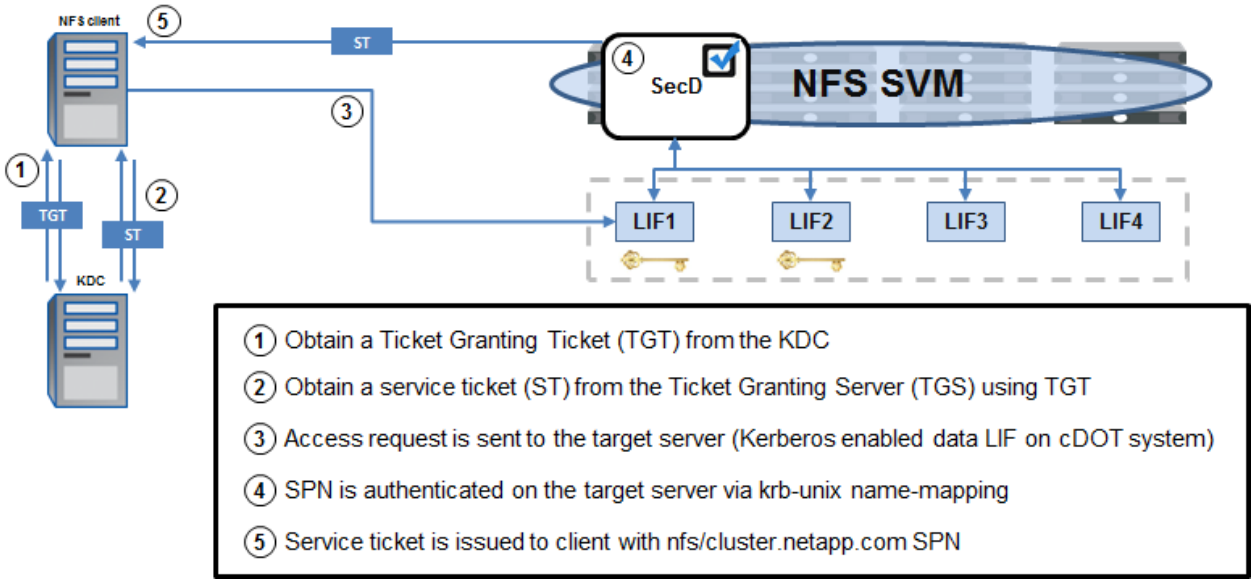
When a Kerberos principal logs in to the Kerberos realm, the principal sends a TGT request that contains the principal but not the password or secret key to the krb5kdc daemon. On receiving this request, the KDC looks up the principal in the KDC database and uses the associated password from the database to encrypt the TGT response.

From the KDC, the encrypted TGT is sent to the principal. The principal decrypts the TGT response by using the secret key obtained from the password or from the keytab and requests authentication to the NFS server (in this case, clustered Data ONTAP) by presenting the NFS server principal along with the encrypted TGT to the TGS. The TGS then issues a ticket for the NFS server that provides authentication to allow the principal to mount (in the case of an NFS client SPN), or to use a specific file system mounted over NFS from the NetApp cluster (in the case of a user principal). No Kerberos communication takes place between the NFS server and KDC since the NFS server decrypts its portion of the TGS using its keytab entry. The ticket is cached on the NetApp storage system until flushed. Figure 1 (below) shows the Kerberos workflow between the client, NFS server, and KDC.

In clustered Data ONTAP, the Kerberos ticket is cached until the node is rebooted or the SecD process is restarted. To see this cache, use the following command:

```
cluster::> set diag
cluster::> diag secd cache show-krb-creds -node [nodename] -vserver [vservername]
```

Figure 1) Kerberos workflow between client, KDC, and NFS server on NetApp storage.



When an object (in this case, an Active Directory machine or user account) is created for use by an SPN on the Active Directory DC, the user principal name (UPN) is also set when the ktpass utility is used. An object can have numerous SPNs, but only one UPN. When the NFS client attempts a Kerberos connection using a credential established via an SPN from a keytab file, Active Directory maps the incoming connection request to a UPN to find the appropriate account. In this document, only one machine account is needed, with one UPN/SPN. This is a departure from previous methods that created three separate accounts.

The rpc.gssd service on a Linux client will search for SPNs in a specific order, listed in the gssd man pages as well as below.

Three main Kerberos SPNs are leveraged for Kerberized NFS:

root/host.domain.com – used by the NFS client for mount requests

nfs/host.domain.com – required to be used by the NFS server (for example, nfs/cluster.domain.com)

host/host.domain.com – used by the NFS client, usually for third-party applications like SSSD

Any of the above may be used to create a principal in Active Directory, but only one is required. This document covers the use of only the root SPN.

2.4 Client-Side Kerberos Commands

Table 1) Client-side Kerberos commands.

Command	Description

kinit	To get and cache the initial Kerberos ticket. Essentially a "login" to the KDC. Leverages the krb5.conf file for realm information. If no name is specified, the user that is logged in to the NFS is used.
	Example:
	`[root@centos6 /]# kinit administrator`
	`Password for administrator@DOMAIN.NETAPP.COM:`

klist	List contents of the Kerberos ticket cache and keytabs. Default is to list the cached credentials. To specify the keytab file, use the `-k` option. To show encryption types, use `-e`. To show timestamps, use `-t`.
	Example:
	`[client] # klist -kte`
	`Keytab name: FILE:/etc/krb5.keytab`
	`KVNO Timestamp Principal`
	`---- ----------------- ----------------------------------------------------------`
	`4 05/16/13 11:57:56 root/nfsclient.nfsclient.domain.netapp.com@DOMAIN.NETAPP.COM (des-cbc-crc)`
	`4 05/16/13 11:57:56 root/nfsclient.nfsclient.domain.netapp.com@DOMAIN.NETAPP.COM (des-cbc-md5)`
	`4 05/16/13 11:57:56 root/nfsclient.nfsclient.domain.netapp.com@DOMAIN.NETAPP.COM (arcfour-hmac)`
	`4 05/16/13 11:57:56 root/nfsclient.nfsclient.domain.netapp.com@DOMAIN.NETAPP.COM (aes256-cts-hmac-sha1-96)`
	`4 05/16/13 11:57:56 root/nfsclient.nfsclient.domain.netapp.com@DOMAIN.NETAPP.COM (aes128-cts-hmac-sha1-96)`

ktutil	Used to manage the client-side keytab. Running ktutil will start an application that allows the reading (rkt) and writing (wkt) of keytab files.
	Example:
	`[root@centos6 /]# ktutil`
	`ktutil:  rkt /etc/krb5.keytab`
	`ktutil:  list`
	`slot KVNO Principal`
	`---- ---- ---------------------------------------------`
	`   1    2 root/nfsclient.nfsclient.domain.netapp.com@DOMAIN.NETAPP.COM`

kdestroy	Destroys a ticket cache.

kvno	Prints the key versions of Kerberos principals. The kvno must match across principals. Useful for troubleshooting SPNs. When using this command, a Kerberos session must be established between the client and KDC.
	Example:
	`[root@centos6 /]# kvno root/nfsclient.nfsclient.domain.netapp.com`
	`root/nfsclient.nfsclient.domain.netapp.com@DOMAIN.NETAPP.COM: kvno = 2`

2.5 Client-Side Kerberos Files

Table 2) Client-side Kerberos files.

File	Description
/etc/krb5.conf	This file must exist on clients wanting to use Kerberos. The file consists of several sections that are used in ticket services. • [libdefaults] sets defaults for Kerberos on your system; for example, default realm, default ticket lifetime, encryption types. • [realms] tells where to find the KDCs for each realm. • [instancemapping] maps client principal properly (for things like cron jobs that require a special principal). • [domain_realm] maps domains to realms. • [logging] tells Kerberos where and how to log errors. • [appdefaults] lists default settings for outgoing Kerberized network connection applications and for incoming portal mode connections. See the following for more about /etc/krb5.conf: http://web.mit.edu/kerberos/krb5-1.5/krb5-1.5/doc/krb5-admin/krb5.conf.html

/etc/krb5.keytab	An encrypted local copy of the host's key. Although the file is encrypted, it is still a point of entry and a potential security hole so the file must be readable only by root; otherwise Kerberos will fail. It should only exist on the local server's disk. See the following for more information about the krb5.keytab file: http://web.mit.edu/kerberos/krb5-1.5/krb5-1.5/doc/krb5-install/The-Keytab-File.html

3 Benefits of Using Kerberized NFS

Kerberos is a mode of authentication for users and hosts. Sometimes this is confused with authorization, which uses access control lists (ACLs) or mode bits on files and directories to determine a user's access. Authorization is performed after authenticating the user or host.

Authentication proves who you are; authorization allows you to do what you need to do once you've been authenticated.

For example, if someone buys a subway ticket, he is allowed through the turnstile (authentication). But once that person is inside the station, he may not be able to travel to his destination if the ticket does not allow him to get there (authorization).

Kerberos secures an infrastructure by preventing plain-text passwords from being communicated over the network. The Kerberos database maintains a centralized repository of user name and password information for enhanced security and better manageability. Since the password is encrypted and is not stored on local hosts, the chances of a single host getting compromised because of a local password information cache are reduced.

In addition to wire data security, Kerberized NFS uses RPCSEC_GSS (RFC 2203), which provides a higher group limit compared to the 16-group limitation inherent with AUTH_SYS. With RPCSEC_GSS the user can be part of 32 groups in clustered Data ONTAP.

Currently most NFS servers (7-Mode and clustered Data ONTAP) require a Kerberos service principal name (SPN) to UID mapping to allow a Kerberos principal access to exported data.

Best Practice

It's pointless to have Kerberos secure the NFS host to NFS server Kerberos GSS context establishment, only to have an NIS request to map the user Kerberos principal go on the wire in clear text. If the mapping is intercepted, it can be changed, giving a Kerberos user someone else's UID, and thus incorrect access to files. As such, NIS and NIS+ are not appropriate for use with Kerberos because they are not secure. LDAP using SASL authentication is the preferred method for identity management when using Kerberized NFS.

4 Setting Up LDAP and Kerberized NFS with Microsoft Windows AD

This section describes setting up Kerberized NFS using clustered Data ONTAP as the NFS server and storage. The KDC in this example is Microsoft Windows Active Directory 2008 R2, but the same steps can be used for Windows 2003/2003R2 and Windows 2012. MIT Kerberos and other KDC servers are outside the scope of this document. This section also covers setting up and configuring LDAP for identity management using the Microsoft Windows Active Directory 2008 R2 server.

The following Linux host configurations are covered, but this configuration may apply to earlier versions of each. The LDAP service used in this configuration is SSSD.

- Red Hat Enterprise Linux/CentOS 6.3 and 6.4
- Fedora 18
- SLES 11
- SUSE 12
- Ubuntu 12.1
- Solaris 10

Note: Solaris 10 does not have support for LDAP using SSSD. This document covers Solaris LDAP using the `ldapclient` utility.

4.1 Setting Up Kerberized NFS

The following section describes how to set up Kerberos for NFS clients. By the end of this section, NFS clients should be able to issue a successful kinit (login) to the KDC, as well as successfully mount an NFS export via krb5 security. These steps, outside of any clustered Data ONTAP-specific commands, can be used in conjunction with a 7-Mode system. The appendix of this document covers 7-Mode–specific configuration steps and supported features.

The following is an example of what mounting using Kerberos security looks like:

```
[root@centos64 ~]# mount -o sec=krb5 kerberos:/unix /test
[root@centos64 ~]# mount | grep krb
kerberos:/unix on /test type nfs (rw,sec=krb5,vers=4,addr=10.61.92.41,clientaddr=10.61.179.150)
```

Note: Quick Step Setup steps can be found at the end of this document.

4.1.1 Configuring the Clustered Data ONTAP System for Kerberos

To configure a clustered Data ONTAP system to use Kerberos for NFS, Kerberos must be enabled on a data LIF in the SVM that owns the NFS server. When Kerberos is enabled on a data LIF, an SPN is specified (must be nfs/hostname@REALM) and a principal is created in the KDC. In the case of Microsoft Windows Active Directory, the principal is a machine account.

Note: In order to properly support LIF migration, HA takeover, or pNFS, Kerberos must be enabled for all data LIFs in an SVM.

Before enabling Kerberos on a data LIF, the following must be done:

- DNS configured
- NFS licensed and configured
- Kerberos realm created
- CIFS server created (optional)
- Data volumes created and mounted
- Export policies and rules configured
- Kerberos enabled on data LIFs
- Valid name-mapping for the NFS SPNs exists

4.1.2 Enabling DNS

Enabling DNS must be done per SVM. DNS (forward and reverse lookup) is necessary for Kerberos to function properly. Without DNS, Kerberos is not possible.

To create/enable DNS, the following command should be used:

```
cluster::> dns create -vserver vs0 -domains domain.netapp.com
-name-servers 10.63.98.101,10.63.98.102 -state enabled
```


4.1.3 Configuring NFS

This document assumes that NFS has been licensed and configured on the SVM. If this has not taken place, refer to TR-4067, which covers NFS implementation in clustered Data ONTAP.

4.1.4 Configuring a Kerberos Realm

A Kerberos realm is needed so that the cluster knows how to format Kerberos ticket requests. This is similar to configuring /etc/krb5.conf on NFS clients.

To create a Kerberos realm, use the following example as a guide for the command to run on the SVM hosting the NFS server:

```
cluster::> kerberos-realm create -configname REALM -realm DOMAIN.NETAPP.COM -kdc-vendor Microsoft
-kdc-ip 10.63.98.101 -kdc-port 88 -clock-skew 5 -adminserver-ip 10.63.98.101 -adminserver-port
749 -passwordserver-ip 10.63.98.101 -passwordserver-port 464 -adserver-name WIN2K8-DC -adserver-
ip 10.63.98.101
```

Note: The IP addresses specified in the Kerberos-realm commands are used only during creation of the machine account object or SPN; these IP addresses are not used for actual Kerberized NFS traffic. Therefore there is no need to worry about failover or DNS aliases for these commands. KDC failover for Kerberized traffic is handled via DNS SRV records. For more information, see section 4.1.20, “Domain Controller Redundancy and Replication.”

4.1.5 Creating a CIFS Server (Optional)

Creating a CIFS server is not a necessary step, but it can impact how Kerberos is configured. To create a CIFS server, leverage the `vserver setup` command. For information on how a CIFS server can impact Kerberos configuration, see the section entitled “Configuring the domain controller.”

4.1.6 Configuring Export Policies and Rules

To be able to mount and access an NFS export, an export policy and rule must be created and applied to the data volume as well as its parent volume. This export policy and rule must permit krb5 access to the mount in the ro and/or rw portion of the export policy rule. Valid entries include “krb5” (plus additional options, if desired; for example, “krb5, sys”) and “any.” For NFSv3 mounts that use network lock manager, the export policy rule must include “sys” in addition to “krb5” to allow successful mounts. Additionally, the protocol portion of the export policy rule must allow NFS access. For more information on export policies and rules, refer to TR-4067, which covers NFS implementation in clustered Data ONTAP.

4.1.7 Creating and Mounting Data Volumes

Before accessing an NFS export, a data volume must be created and mounted to a junction path in the SVM’s namespace. For information on creating volumes and mounting them, refer to TR-4067, which covers NFS implementation in clustered Data ONTAP.

4.1.8 Enabling Kerberos on a Data LIF

To use Kerberos for NFS, Kerberos must be enabled on a data LIF in the SVM. When this is done, the SPN is defined and a principal is created on the KDC defined in the realm configuration. To enable Kerberos, use the following command as guidance:

```
cluster::> kerberos-config modify -vserver vs0 -lif data -kerberos enabled
-spn nfs/kerberos.domain.netapp.com@DOMAIN.NETAPP.COM
```

When this command runs, the KDC is contacted and a user name and password prompt are issued via CLI.

In versions of clustered Data ONTAP earlier than 8.2.1, the user name provided must have rights to create objects in the Computers OU in Active Directory. This can be a domain administrator or a user who has had rights delegated to manage that OU. In 8.2.1 and later, a custom OU can be specified when using the Kerberos-config modify command.

Note: The SPN must use the format in the example of primary/instance@REALM, where REALM is always in ALL CAPS. Failure to use this format will result in the command failing.

Best Practice

Kerberos should be enabled on **numerous data LIFs** using the same SPN, allowing redundancy across the SVM. If using DNS load balancing, Kerberos must be enabled on all data LIFs in the load balance set to prevent data access issues.

Example:

```
cluster::> kerberos-config show -vserver vs0
  (vserver nfs kerberos-config show)
              Logical
Vserver       Interface     Address          Kerberos SPN
------------- ------------- ---------------- -------- ----------------------
vs0            data          10.61.92.34     enabled
nfs/krbsn.domain.netapp.com@DOMAIN.NETAPP.COM
vs0            data2         10.61.92.37     enabled
nfs/krbsn.domain.netapp.com@DOMAIN.NETAPP.COM
vs0            vs_mgmt       10.61.92.36     disabled -
3 entries were displayed.
```

Note: The only time the data LIF contacts the Active Directory KDC is when the LIF has Kerberos enabled on it. Therefore, if an AD KDC goes down for any reason, the cluster will leverage other AD KDCs in the domain.

Non-Windows KDC Considerations—Keytab Files

The `Kerberos-config` command also has an option for `-keytab-uri`, in which a keytab file can be imported from a client.

```
cluster::> kerberos-config modify -keytab-uri
  {(ftp|http)://(hostname|IPv4 Address|'['IPv6 Address']')...}  Load keytab from URI
```

This is not necessary with KDCs running Windows Active Directory. However, when using non-Windows Active Directory servers (such as MIT or Heimdal), the keytab would need to be copied to the cluster.

4.1.9 Creating and Verifying Name Mapping for the NFS SPN

When Kerberos for NFS is enabled on a data LIF, an SPN is specified in the command structure:

```
cluster::> kerberos-config modify -vserver vs0 -lif data -kerberos enabled
-spn nfs/kerberos.domain.netapp.com@DOMAIN.NETAPP.COM
```

When a Kerberized mount request is made, the cluster's internal security daemon (SecD) will process the request and attempt to authenticate the SPN.

4.1.10 What Is SecD?

SecD is a user-space application that runs on a per-node basis. The SecD application handles name service lookups such as DNS, NIS, and LDAP, as well as credential queries, caching, and name mapping,

For more information on SecD, see TR-4067: Clustered Data ONTAP NFS Implementation Guide.

4.1.11 Name Mapping

When authentication occurs, SecD will attempt to map the SPN to a valid UNIX user by way of a krb-unix mapping. This mapping will use the first section of the SPN for the mapping rule if no name-mapping rules exist.

For example, **nfs/kerberos.domain.netapp.com** would map to the UNIX user **nfs** by default if no name-mapping rule is defined.

If the UNIX user does not exist in any of the name services listed for the SVM, then the authentication request will fail and Kerberos will be unable to mount. This manifests on a client as an access or permission-denied error.

If a mapping to a different user than the one defined in the SPN is required, then a krb-unix name-mapping rule can be created in the SVM.

Example:

```
cluster::> vserver name-mapping create -vserver vs0 -direction krb-unix -position 1
-pattern nfs/kerberos.domain.netapp.com -replacement krbuser
```

Once this is created, the SPN will then map to the UNIX user named “krbuser” instead of the SPN user.The clients will also attempt a Kerberos-to-UNIX mapping with their SPN. Clients such as RHEL, SUSE, and so on would use the SPN of **root/hostname@REALM** in most cases. The root user exists by default in clustered Data ONTAP 8.2 and later, but must be created in earlier versions. In Solaris, the client will generally use the SPN of **host/solaris@REALM**, so a user named host or an equivalent name-mapping rule should be created either locally on the SVM or in the NIS or LDAP server used for name mapping.

If it is unclear whether an SPN is properly mapping, the following diag-level commands in the cluster CLI can test the name mapping as seen by SecD:

```
cluster::> set diag
cluster::*> diag secd name-mapping show -node node1 -vserver vs0 -direction krb-unix -name
nfs/kerberos.domain.netapp.com
nfs/kerberos.domain.netapp.com maps to nfs

cluster::*> diag secd name-mapping show -node node1 -vserver vs0 -direction krb-unix -name
host/solaris.domain.netapp.com
host/solaris.domain.netapp.com maps to host
```

Name-Mapping Considerations in Active Directory Domain Trusts

Clustered Data ONTAP supports domain trusts. When using a domain trust with Kerberized NFS, the clustered Data ONTAP system must be able to resolve the user SPN from the trusted domain to a valid UNIX user. This can be done in one of several ways:

- LDAP user mapping
- Local user account created
- krb-unix name mapping rule

When a user SPN from a trusted domain arrives at the cluster node, SecD will attempt to map that user SPN to a valid UNIX user.

The following secd.log excerpt illustrates that:

```
GSS_S_COMPLETE: client = 'trust@TRUST.NETAPP.COM'
Querying config source 'NameMapping' (with 7 rows of data) by keys vserver id: '13', direction:
'krb-unix', position: '<no key specified>', type: 'user'
Attempting to map SPN trust@TRUST.NETAPP.COM using the cluster mapping store
Trying to map SPN trust@TRUST.NETAPP.COM to trust using implicit mapping
Could not find IDs for local unix user trust for vserver 13
```



```
IDS_FROM_USER_NAME ldapInfoType requested.
Querying config source 'Ldap' (with 3 rows of data) by keys vserver id: '13'
Querying config source 'LdapClientSchema' (with 6 rows of data) by keys schema: 'AD-IDMU' and
vserver id: '13'
 Searching LDAP for the "uidNumber, gidNumber" attribute(s) within base
"dc=domain,dc=netapp,dc=com" (scope: 2) using filter: (&(objectClass=User)(uid=trust))
 ERR  :  RESULT_ERROR_SECD_NAME_MAPPING_DOES_NOT_EXIST
 ERR  :    [  2 ms] Trying to map SPN 'trust@TRUST.NETAPP.COM' to UNIX user 'trust' using
implicit mapping
 ERR  :    [     2] Name 'trust' not found in UNIX authorization source LOCAL
 ERR  :    [     3] Using a cached connection to 10.61.179.152
 ERR  :    [     5] Name 'trust' not found in UNIX authorization source LDAP
 ERR  :    [     5] Could not get an ID for name 'trust' using any NS-SWITCH authorization source
 ERR  :    [     5] Unable to map SPN 'trust@TRUST.NETAPP.COM'
 ERR  :  **[     5] FAILURE: Unable to map Kerberos NFS user 'trust' to appropriate UNIX user
```

In the example above, the request first tries to find a local user named “trust” via implicit mapping. Since that failed, the request then looks for a name mapping in LDAP, which is the next preferred nm-switch/ns-switch specified on the SVM. Once it’s determined that the name doesn’t exist in LDAP, the cluster will then look for a name-mapping rule. If no name-mapping rule exists, the request fails. Since this is a krb-unix name mapping, the default UNIX user setting does not apply, since that is a win-unix attribute only.

The ways to resolve this would be:

- Create a UNIX user locally on the cluster with the same user name as the user SPN attempting access.
- Create an LDAP entry for the UNIX user.
- Create a name-mapping rule for the user SPN or for all user SPNs coming from the trusted domain.

Name-mapping rules support regular expressions, so it is possible to create a name-mapping rule to support all users in a trusted domain. For more information about regular expressions in name-mapping rules, consult the clustered Data ONTAP documentation for the version being used.

Example of SVM name-mapping rule for all SPNs coming from a designated domain:

```
cluster::> vserver name-mapping show -vserver nfs -direction krb-unix
Vserver        Direction Position
-------------- --------- --------
nfs            krb-unix  1        Pattern: (.+)@TRUST.NETAPP.COM
                                  Replacement: pcuser
```

Example of SVM setting for nm-switch and ns-switch:

```
cluster::> vserver show -vserver nfs -fields nm-switch,ns-switch
vserver ns-switch nm-switch
------- --------- ---------
nfs     file,ldap file,ldap
```

Example of default UNIX user option in `cifs options`:

```
cluster ::> cifs options show -vserver nfs -fields default-unix-user
vserver default-unix-user
------- -----------------
nfs     pcuser
```

Example of a working user in a domain trust accessing a mount in a different Kerberos realm:

```
sh-4.1$ klist -e
Ticket cache: FILE:/tmp/krb5cc_1234
Default principal: trust@TRUST.NETAPP.COM

Valid starting     Expires            Service principal
05/24/13 17:19:44  05/25/13 03:19:59  krbtgt/TRUST.NETAPP.COM@TRUST.NETAPP.COM
        renew until 05/25/13 17:19:44, Etype (skey, tkt): aes256-cts-hmac-sha1-96, aes256-cts-
hmac-sha1-96
```



```
05/24/13 17:20:02  05/25/13 03:19:59  krbtgt/DOMAIN.NETAPP.COM@TRUST.NETAPP.COM
        renew until 05/25/13 17:19:44, Etype (skey, tkt): arcfour-hmac, arcfour-hmac
05/24/13 17:20:02  05/25/13 03:19:59  nfs/kerberos.domain.netapp.com@DOMAIN.NETAPP.COM
        renew until 05/25/13 17:19:44, Etype (skey, tkt): des-cbc-crc, des-cbc-md5

sh-4.1$ pwd
/mnt
sh-4.1$ mount | grep mnt
kerberos:/unix on /mnt type nfs (rw,sec=krb5,vers=4,addr=10.61.92.42,clientaddr=10.61.179.150)
```

4.1.12 Other Considerations

- The data LIF in the SVM must be able to communicate with Active Directory. If the data LIF is not communicating, be sure to check the configuration of your data LIF and overall network.
- The only time the data LIF will communicate with Active Directory is during the machine account creation. After this, the cluster will store the Kerberos keytab locally.
- When creating the SPN with the `kerberos modify` command on the cluster, the machine account should be less than 15 characters long. Windows limits the creation of non-Windows machine accounts to 15 characters. Any name beyond 15 characters will get truncated. The machine account name is derived from the SPN specified, including the service portion.

  Example:
  SPN nfs/kerberos.netapp.com@NETAPP.COM becomes NFS-KERBEROS-NE
  SPN nfs/reallylongname.netapp.com@NETAPP.COM becomes NFS-REALLYLONGN
- By default, the machine account will be placed in the CN=Computers location on a Windows Active Directory domain controller. This currently cannot be changed. To work around this limitation, move the machine account manually in Active Directory Users and Computers after Kerberos has been enabled on the data LIF.
- Clustered Data ONTAP currently only supports DES style encryption types for Kerberized NFS. For information on configuring the NFS client to navigate this limitation, see the section entitled "Configuring the client." For information on how to navigate this limitation from the domain controller, see the section entitled "Configuring the domain controller."

4.1.13 Configuring the Domain Controller for Kerberos

The domain controller configuration will consist of the following:

- Allowing DES encryption types
- Modifying machine account objects to use DES
- Creating principals and keytab files
- Adding hosts to DNS
- Verifying/deleting duplicate SPNs

For condensed setup steps, see the "Quick Step Setup Guides" section in this document.

4.1.14 Adding an NFS Client to Windows Active Directory DNS

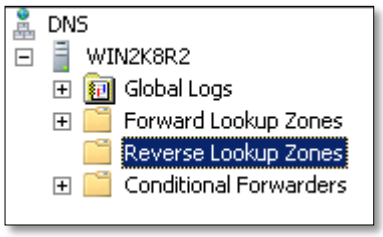
This section describes how to add an NFS client to a Windows DNS server. To utilize Kerberos, the NFS client will need to have forward and reverse lookup entries in DNS. This section applies to all Windows versions from 2003 onward. This section does not cover non-Windows DNS servers. This can be done via DNS GUI or via command line (cmd). This section covers both methods.

Table 3) Adding NFS client to Windows DNS (GUI).

1. Log in to the Windows server running DNS in Active Directory. In many cases, this server is a domain controller.

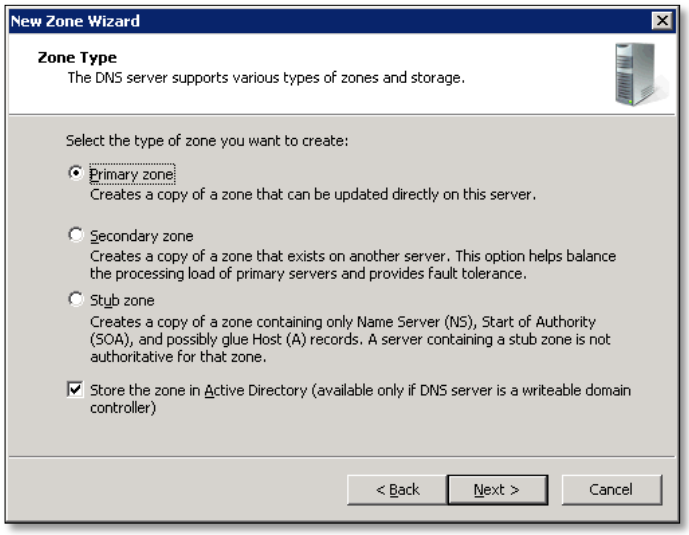
2. Navigate to Start -> Administrative Tools -> DNS.

3. Expand the DNS server and the “Reverse Lookup Zones” folder.

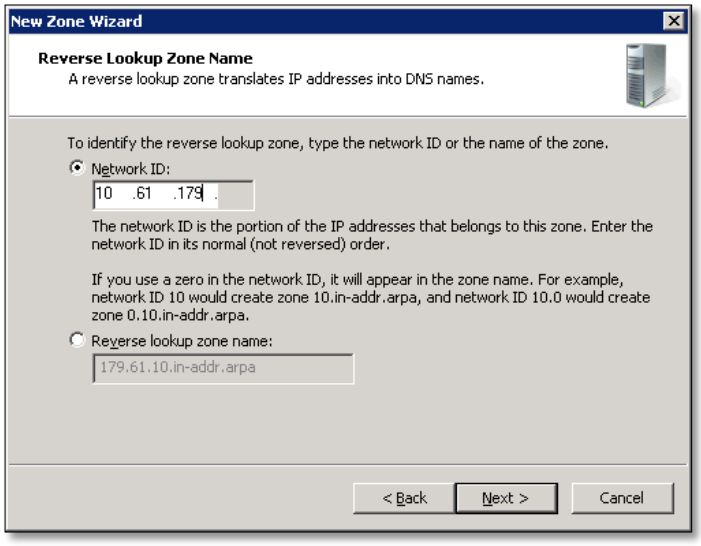


4. Right-click and select “New Zone” if the zone for the client does not already exist. If the zone already exists, move on to step 9.

5. Create a Primary Zone and verify that “Store the zone in Active Directory” is selected.

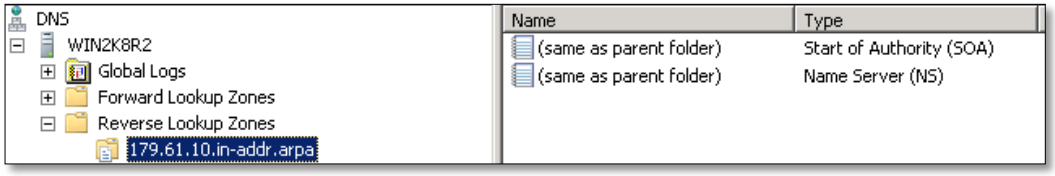


6. Accept the defaults until the “Reverse Lookup Zone Name” window appears. Enter the first three octets of the client’s IP address.

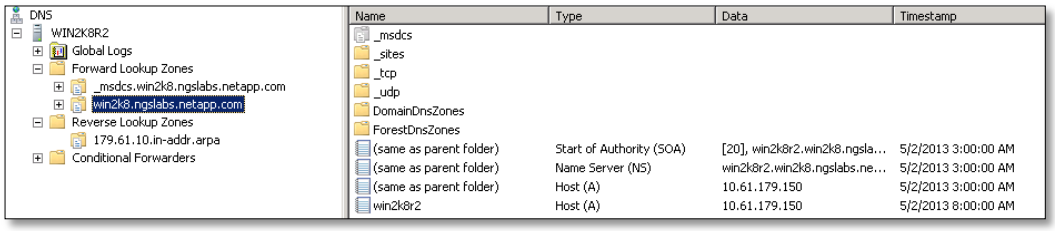


7. Select the desired option for Dynamic Updates and then click Next and Finish.

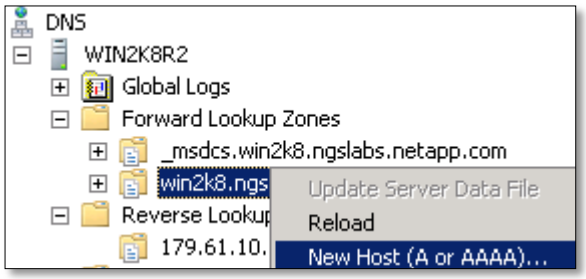
8. The reverse lookup zone is now created.



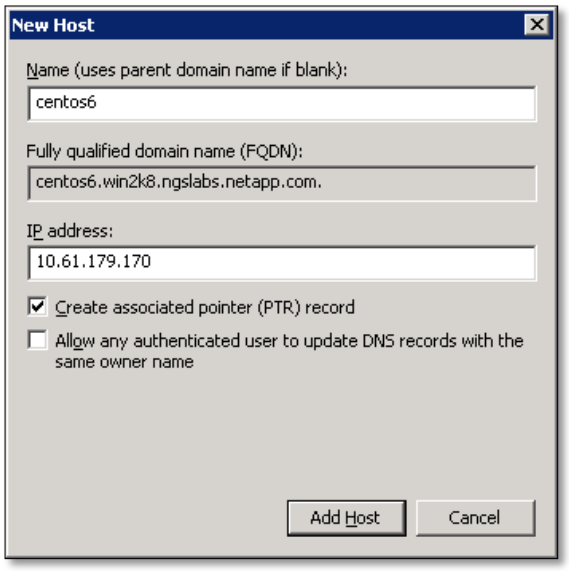
9. To create the client's "A" record, click on the "Forward Lookup Zone" and select the DNS domain.



10. Right-click on the domain and select New Host (A or AAAA).



11. Create the client's "A" record by filling in the necessary fields. Select "Create associated PTR record" to create the reverse lookup record.



12. Click Add Host and the client's DNS "A" record is created.

Table 4) Adding NFS client to Windows DNS (dnscmd).

1. Open the cmd prompt by going to Start -> Run and typing "cmd."
2. Run the following command on the DNS server:

```
C:\> dnscmd /RecordAdd [dnsdomain.com] [hostname] /CreatePTR A [clientIP]
```

Example:

```
C:\> dnscmd /RecordAdd domain.netapp.com centos6 /CreatePTR A 10.61.179.170
```

Note: Before running the commands above, verify that the reverse lookup zone exists for the subnet the client lives in. Steps to create reverse lookup zones are covered in Table 3) Adding NFS client to Windows DNS (GUI)..."

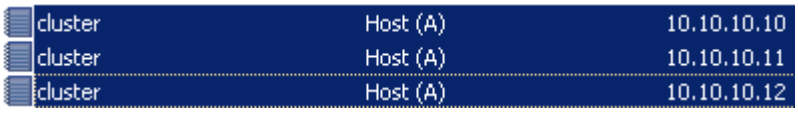
4.1.15 Adding the SVM Data LIFs to Windows Active Directory DNS

The SVM data LIF IP addresses will need to be added to DNS in addition to the NFS client's IP address. The same steps apply as above. However, if there are multiple data LIFs on the SVM, each data LIF should be added to DNS as a round-robin A record or by using the clustered Data ONTAP on-box DNS load-balancing feature. This prevents DNS lookup failures during Kerberos authentication attempts.

Using Round Robin DNS

To create a round-robin A record, simply create another A record with the same name as the original A record.

Example:



Note: Kerberos can be enabled on multiple data LIFs using the same SPN, allowing redundancy across the SVM. If using DNS load balancing, Kerberos must be enabled on all data LIFs in the load balance set to prevent data access issues.

For more information on round-robin DNS in Windows, see the following:

Configuring Round-Robin DNS in Windows

For information on clustered Data ONTAP networking best practices, see TR-4182: Best Practices for Clustered Data ONTAP Network Configurations.

Using DNS Aliases

When enabling Kerberos on a data LIF, the SPN is specified during the configuration. This SPN will determine which hostname is used to access Kerberized mounts. For example, if an SPN of "nfs/kerberos.domain.netapp.com" is used, then the mounts would be accessed with the hostname of "kerberos." This is because the hostname used in the mount will determine which SPN to pass to the KDC for authentication. If a DNS alias is used, then that alias would be passed as the SPN to the KDC and Kerberized mounts would fail with an "access denied" error if the DNS record isn't configured properly:

```
# mount -o sec=krb5 alias:/unix /mnt
mount.nfs: access denied by server while mounting alias:/unix
```

If an alias is to be used, a DNS Canonical name (CNAME) record should be created rather than an A record and pointed to the DNS record associated with the NFS machine account. Once this is done, the SPN request will be forwarded to the appropriate principal in the KDC.

Best Practice

When using multiple data LIFs for Kerberized NFS mounts, it is a best practice to use either round-robin or on-box DNS load balancing to make sure that name resolution of data LIFs returns multiple IP addresses to clients to prevent overloading a single node in the cluster with connections.

Using On-Box DNS Load Balancing

Clustered Data ONTAP offers the ability to leverage the named service on each node to service DNS requests from clients and to issue data LIF IP addresses based on an algorithm that calculates CPU and node throughput. This process provides the least-used data LIF to ensure proper load balancing across the cluster for mount requests. Once a mount is successful, the client continues to use that connection until remount.* This differs from round-robin DNS, because the external DNS server services all requests and has no insight into how busy a node in the cluster is. Rather, the DNS server simply issues an IP address based on which IP is next in the list. Use of DNS load balancing is not necessary when using NFSv4.x referrals, because the connection is made to the local node regardless of what IP address is returned from DNS.

Additionally, round-robin DNS issues IP addresses with a time to live (TTL). The TTL caches the DNS request in Windows for 24 hours by default. On-Box DNS issues a TTL of 0, which means that DNS is never cached on the client and a new IP is always issued based on load.

* This does not apply to pNFS, which redirects traffic for I/O consistently during mounts. For more information about pNFS, see TR-4067: NFS Implementation in Clustered Data ONTAP and TR-4063: Parallel Network File System Configuration and Best Practices for Clustered Data ONTAP 8.2.

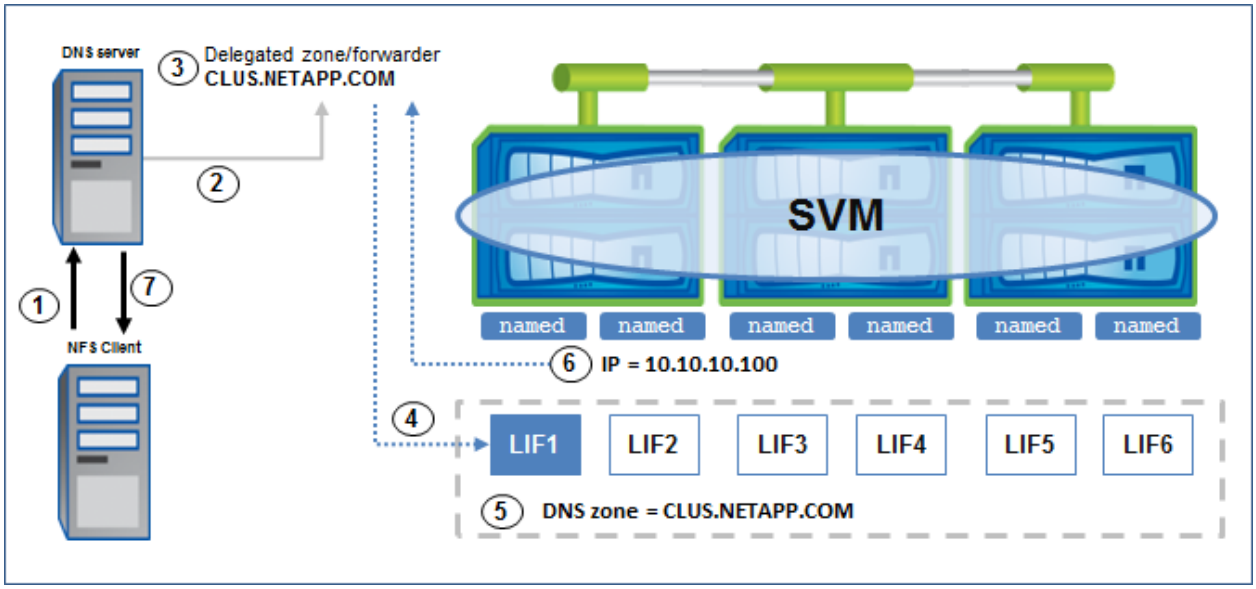
Best Practice
When using On-Box DNS in clustered Data ONTAP, make sure that the cluster is running version 8.1.3P1 or later or 8.2.1 or later.

How On-Box DNS Load Balancing Works

Each node in the cluster has a service running that handles incoming DNS requests from clients and issues IP addresses based on a calculated weight, which is determined by using an algorithm based on CPU utilization and node throughput.

When a client attempts to access the cluster by DNS hostname, the following process takes place:

Figure 2) On-Box DNS load balance example



1. The client issues a DNS request and uses the DNS server specified in its configuration.
2. The DNS server looks for the hostname in the request.
3. When using On-Box DNS, the hostname is either a DNS delegation or a conditional forwarder. The record contains a list of data LIF IP addresses to use for DNS requests.
4. The request is forwarded or delegated to one of the data LIF IP addresses on a round-robin basis.
5. The data LIF receives the request if the LIF has the DNS zone configured and is set to listen for DNS queries (which opens port 53 on the LIF).
6. The node receiving the request checks the DNS weights for each node and issues an IP address based on the calculated load.
7. The IP address is returned to the DNS server, which then returns the IP address to the client.

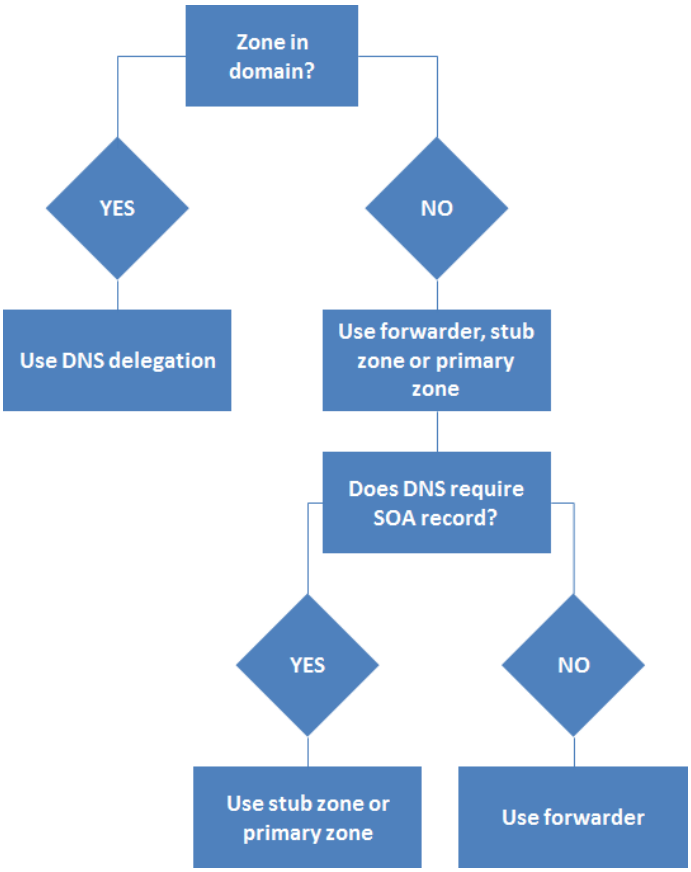
Note: In versions earlier than 8.2, On-Box DNS load balancing did not work with ifgrps or VLANs; with implementations that have those configurations, use external round-robin DNS.

How to Set Up On-Box DNS Load Balancing

This section covers how to set up on-box DNS load balancing by using a Windows DNS server to forward or delegate DNS queries to the clustered Data ONTAP cluster.

Before beginning this setup, a design decision needs to be made about whether to use conditional forwarding, a stub zone, a primary zone, or a DNS zone delegation. The design decision will be based on a variety of factors, as shown in Figure 3.

Figure 3) Factors to consider in setting up On-Box DMS load balancing.



What is a DNS forwarder?

A forwarder is a DNS server on a network that is used to forward DNS queries for external DNS names to DNS servers outside that network. You can also forward queries according to specific domain names by using conditional forwarders, which override regular DNS forwarders.

What is a conditional forwarder?

A conditional forwarder is a DNS server on a network that is used to forward DNS queries according to the DNS domain name in the query. For example, a DNS server can be configured to forward all the queries it receives for names ending with `example.newname.com` to the IP address of a specific DNS server, or to the IP addresses of multiple DNS servers. A conditional forward would be used when a DNS server's domain differs from the desired DNS domain name.

For example:

`example.newname.com` → `netapp.com`

A conditional forwarder requires the data LIFs to be added to DNS as name servers and to have a Start of Authority (SOA) record, as well as having a forward lookup zone and reverse lookup entries created. Clustered Data ONTAP does not provide SOA records, so if the DNS server requires them to configure conditional forwarding, then conditional forwarding is not possible. Windows 2008 and later may require SOA records. Windows 2003 DNS does not require SOA records.

What is a stub zone?

From the Microsoft TechNet article on stub zones:

"A stub zone is a copy of a zone that contains only those resource records necessary to identify the authoritative Domain Name System (DNS) servers for that zone. A stub zone is used to resolve names between separate DNS namespaces. This type of resolution may be necessary when a corporate merger requires that the DNS servers for two separate DNS namespaces resolve names for clients in both namespaces.

"A stub zone consists of:

- The start of authority (SOA) resource record, name server (NS) resource records, and the glue A resource records for the delegated zone.
- The IP address of one or more master servers that can be used to update the stub zone.

"The master servers for a stub zone are one or more DNS servers authoritative for the child zone, usually the DNS server hosting the primary zone for the delegated domain name.

"A stub zone would be required if conditional forwarding does not work because the name servers are not Start of Authority (SOA) servers and the DNS zone created is not a stub zone.

For a comparison between stub zones and conditional forwarders, see:

Contrasting stub zones and conditional forwarders

What is a primary zone?

A primary zone is a DNS zone that is the primary source of information for a zone and that stores a master copy of zone data in a local file or in the database. Unlike stub zones, primary zones allow records (A, AAAA, SRV, and so on) to be created.

What is a delegation?

A DNS delegation delegates requests in the same domain to the DNS servers specified in the delegation zone. For example, for `cdot.netapp.com` in the DNS domain of `netapp.com`, use a delegation.

For more information on zone delegations, see the Microsoft TechNet article on delegating zones and understanding zone delegation.

Table 5) Setting up On-Box DNS load balancing on the cluster

1. Enable DNS zones on the data LIF.

```
::> net int modify -vserver [SVM] -lif [LIF] -dns-zone
[cdot.domain.win2k8.netapp.com]
```

2. Set the LIF to listen for DNS queries. (8.2 and later only).

```
::> net int modify -vserver [SVM] -lif [LIF] -listen-for-dns-query true
```

Once this is done, proceed to "DNS Server Configuration for On-Box DNS Load Balancing," next.

DNS Server Configuration for On-Box DNS Load Balancing

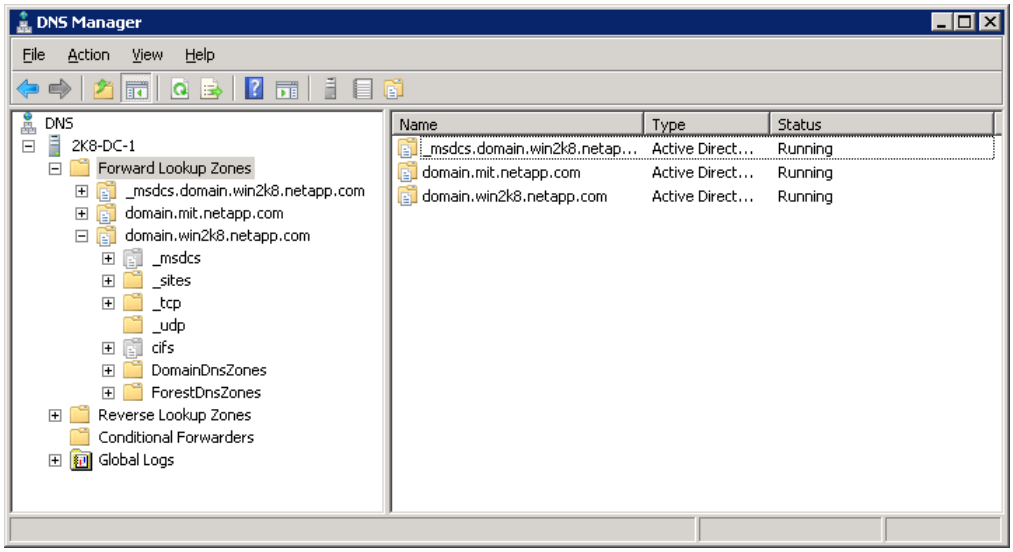
This section covers DNS server configuration for on-box DNS load balancing in clustered Data ONTAP for the following:

- Delegations
- Stub zones
- Primary zones
- Conditional forwarders

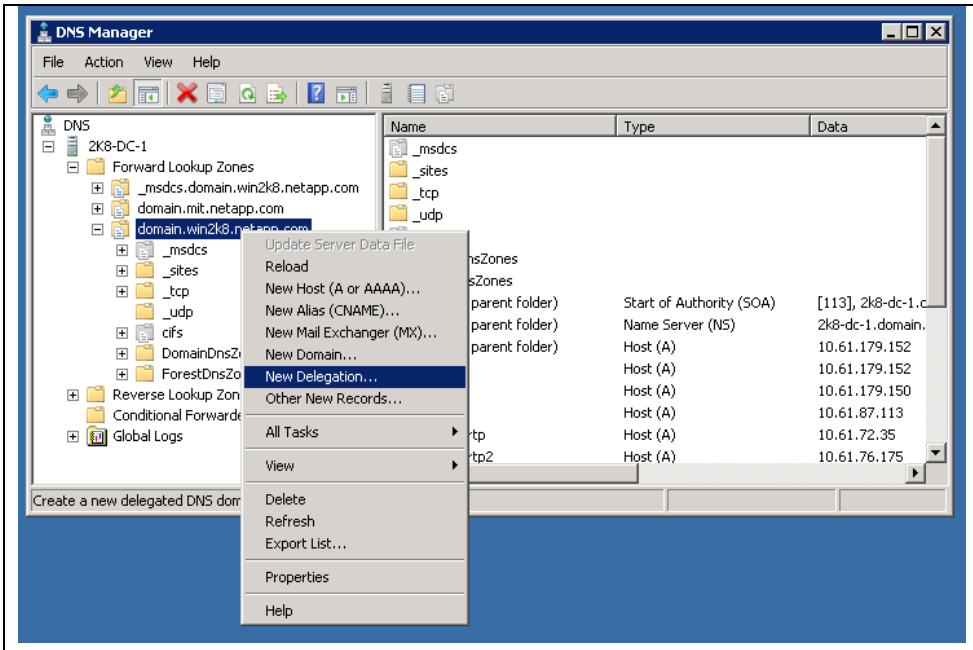
The steps in Table 6 are for Windows DNS servers. For non-Windows DNS servers, contact the vendor for guidance.

Table 6) Setting up delegations.

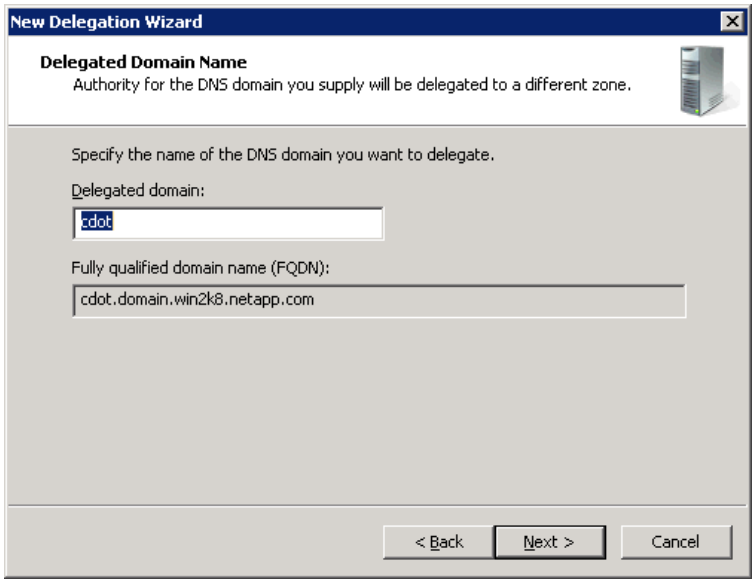
1. Open the DNS Manager console.



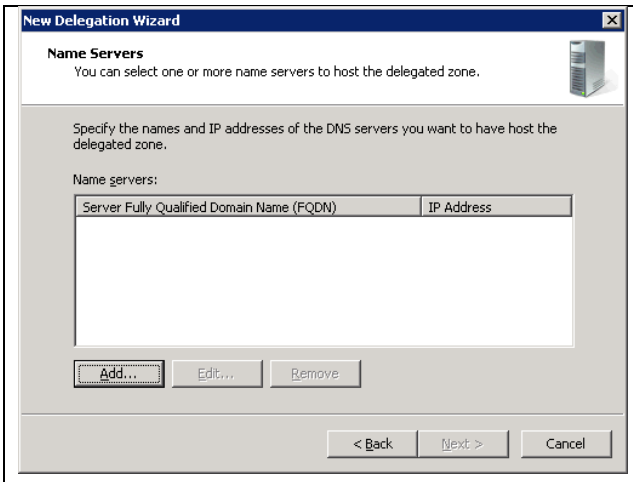
2. Right-click the DNS domain and select New Delegation.

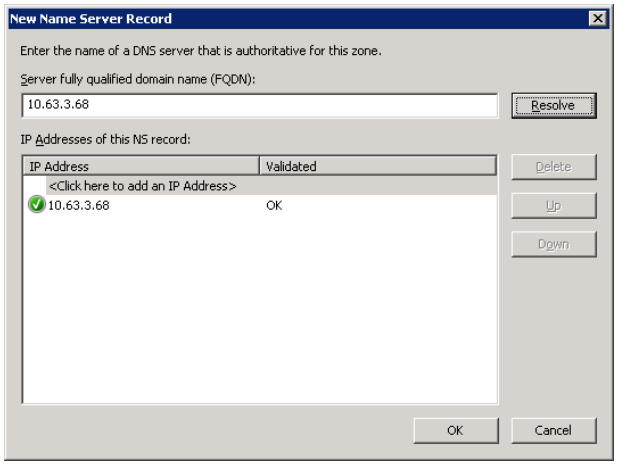


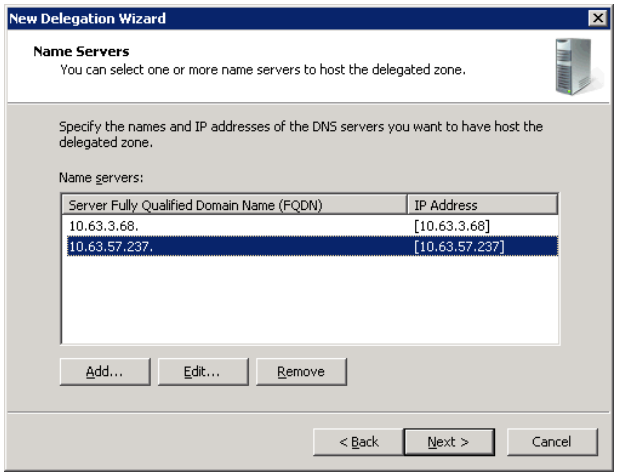
3. Enter the name of the delegated domain.



4. Add the data LIFs as name servers (one at a time).



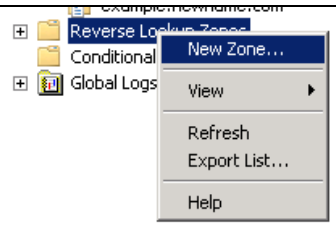




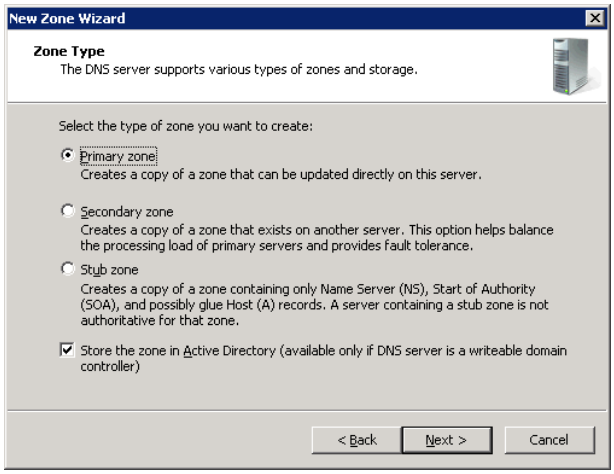
5. Create the reverse lookup zone and PTR records.

   On-Box DNS in clustered Data ONTAP does not support reverse lookups. If it is desired to force clients to use the hostname only for Kerberos, do not create PTR records. This prevents direct IP mounts and ensures that load balancing is enforced. However, in some cases, PTR records are required for Kerberized NFS to work at all.

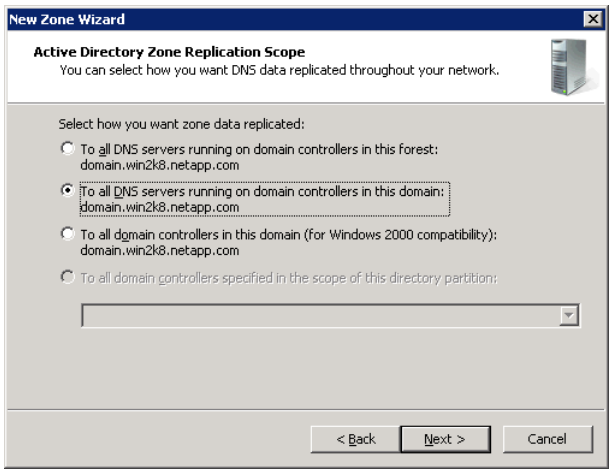
6. Create the reverse lookup zones for the data LIFs.



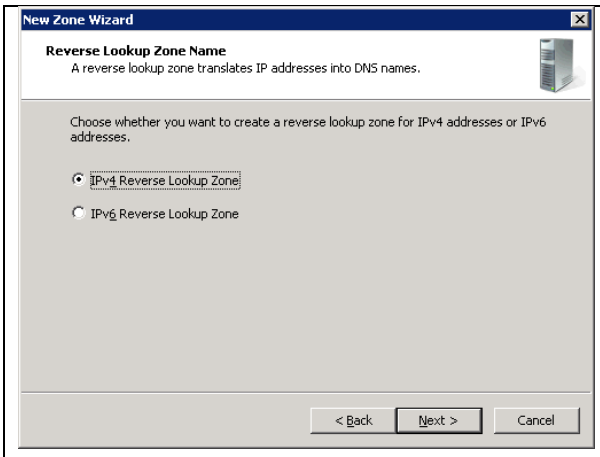
7. Select Primary Zone, because DNS in clustered Data ONTAP cannot service reverse lookups.



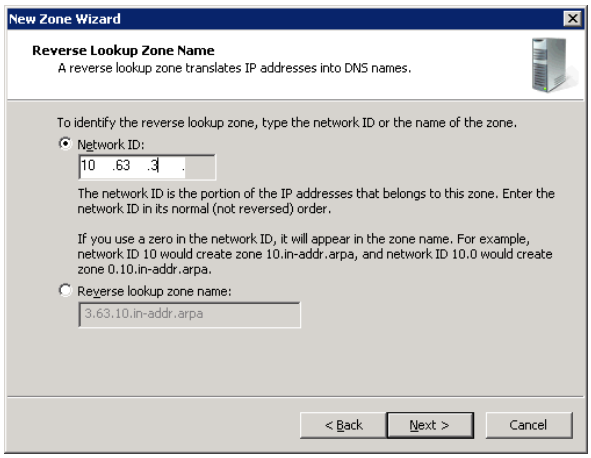
8. Select a zone replication policy to use.



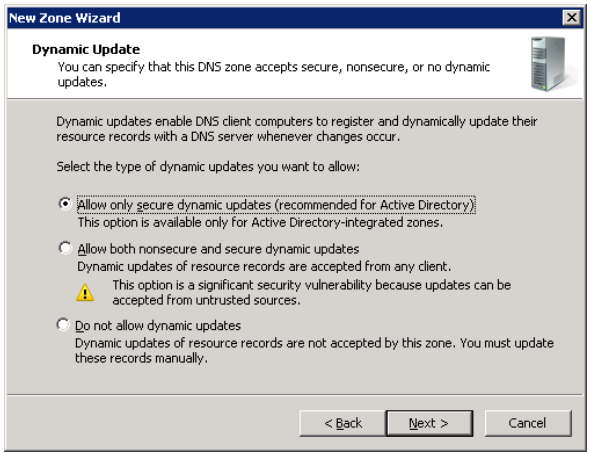
9. Select IPv4 or IPv6 for the lookup zone, depending on what the clustered Data ONTAP version supports and what the data LIFs are using.

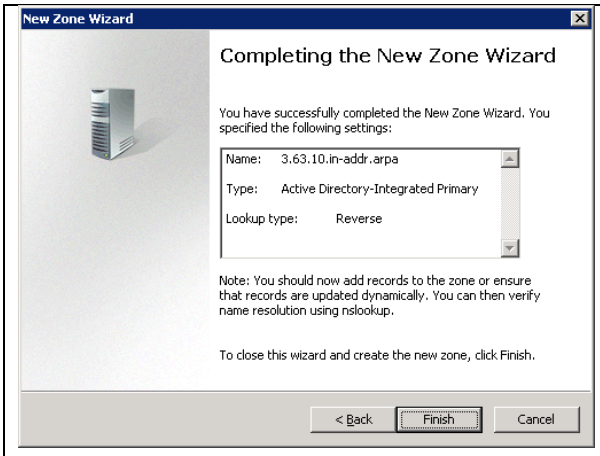


10. Enter the network ID/subnet (the first three octets of the IP address).



11. Select a dynamic update policy.



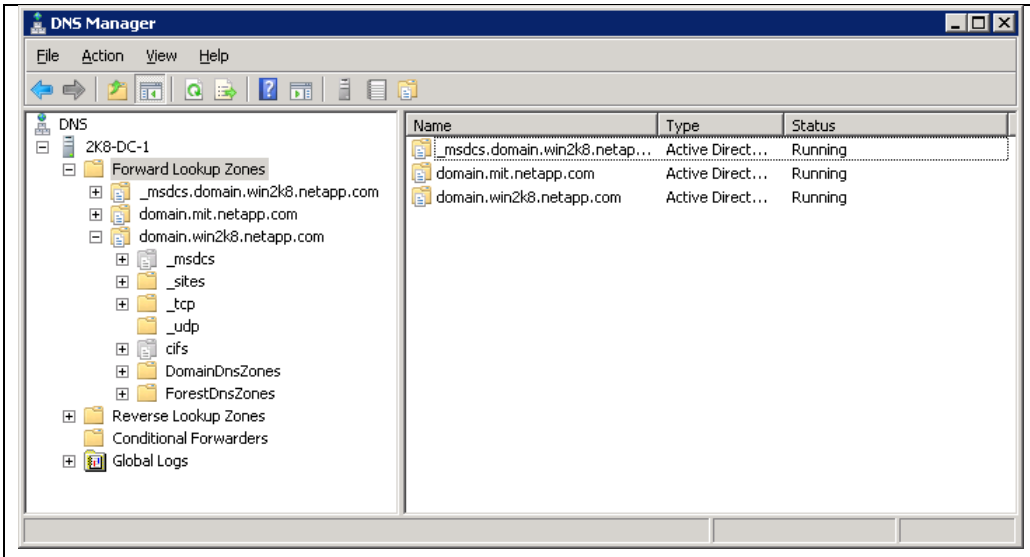


12. Repeat steps 6 through 11for other subnets.
13. Add the PTR records for the data LIFs, because clustered Data ONTAP does not support reverse name lookups.
14. Test DNS lookups for the new zone by using `nslookup` or `dig`.

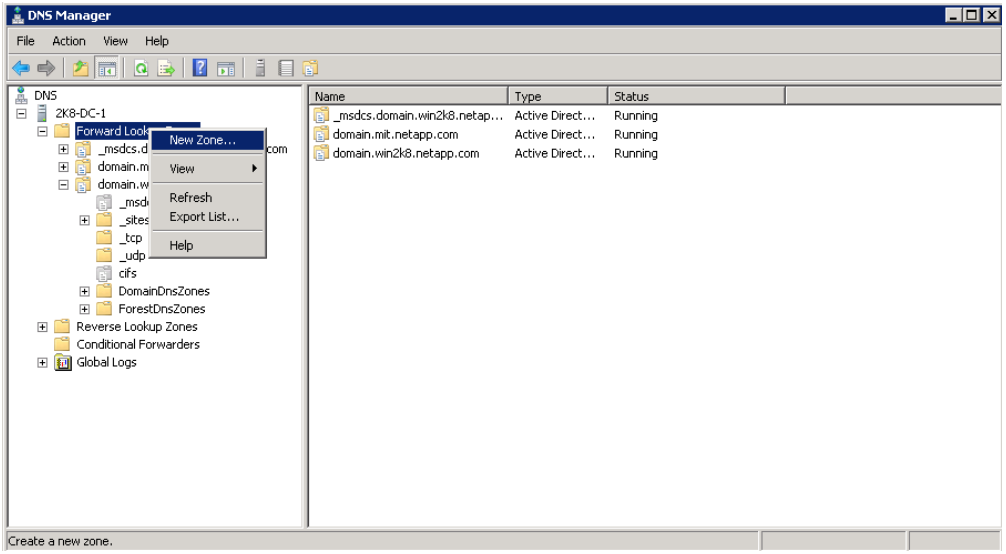
```
C:\>nslookup cdot
Server:  UnKnown
Address:  ::1

Non-authoritative answer:
Name:    cdot.domain.win2k8.netapp.com
Address:  10.63.57.237

C:\>nslookup cdot
Server:  UnKnown
Address:  ::1

Non-authoritative answer:
Name:    cdot.domain.win2k8.netapp.com
Address:  10.63.3.68
```

Table 7) Setting up stub zones.

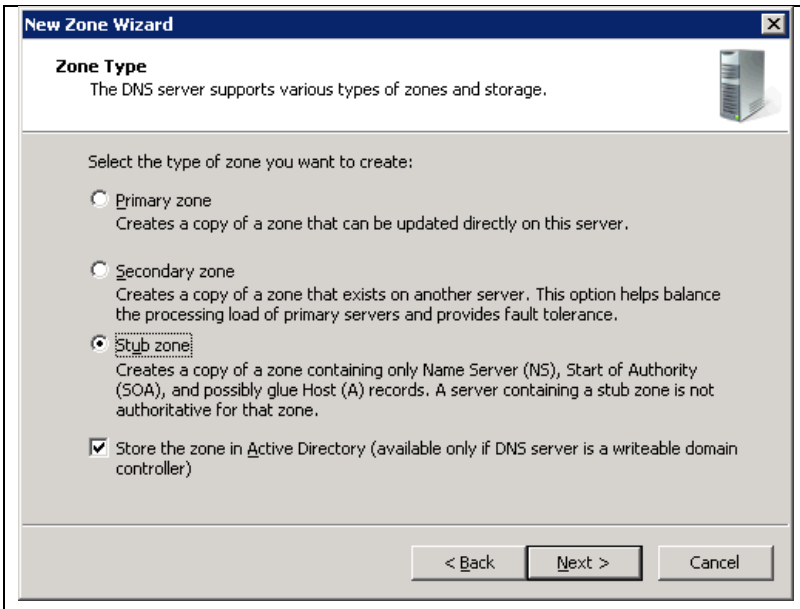
1. Open the DNS Manager console.



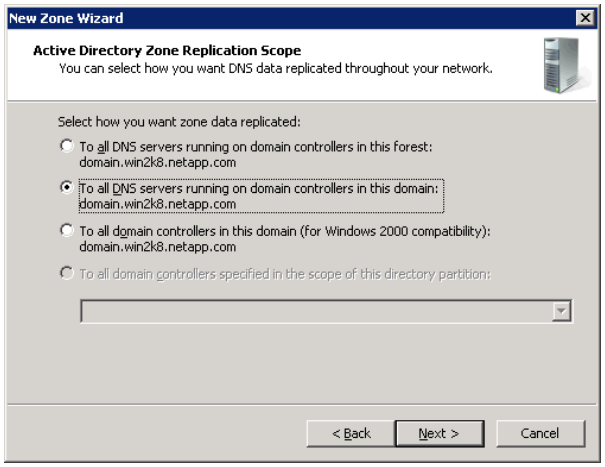
2. Right-click Forward Lookup Zones and select New zone.



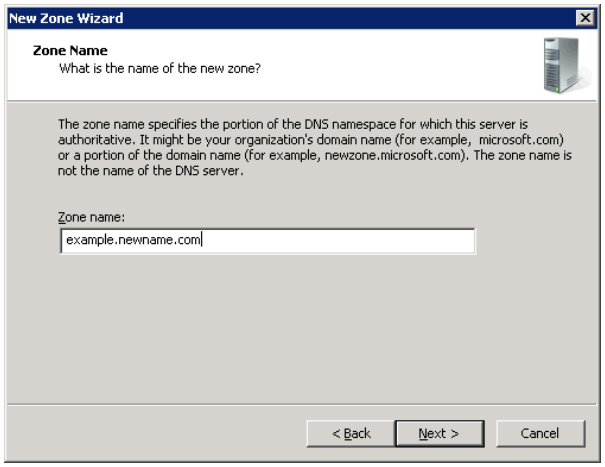
3. Select Stub Zone as the zone.



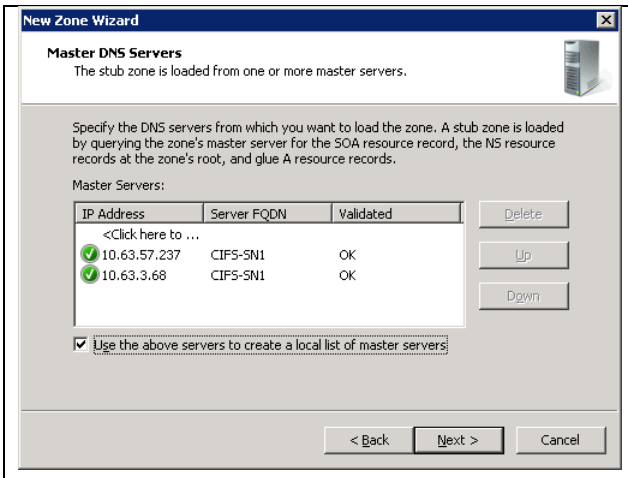
4. Select how zone replication should function.

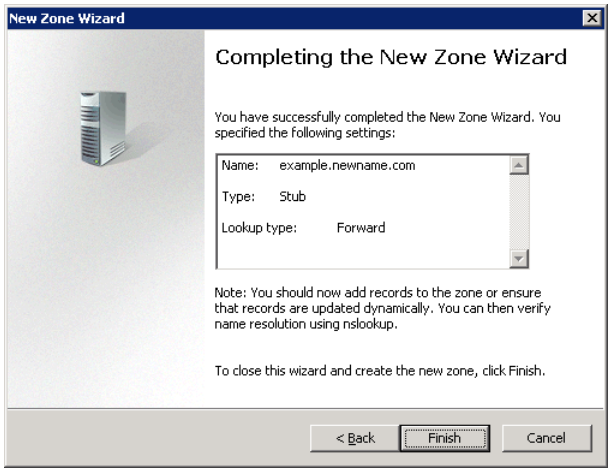


5. Specify the zone name.

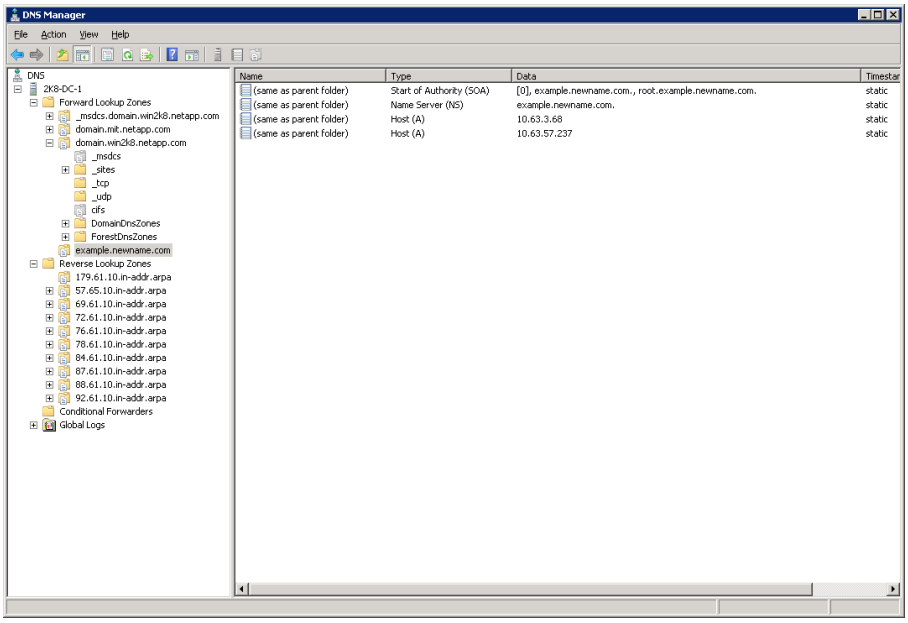


6. Add all data LIFs that are configured for On-Box DNS to the master DNS server list. Select the "Use the above servers to create a local list of master servers" check box.

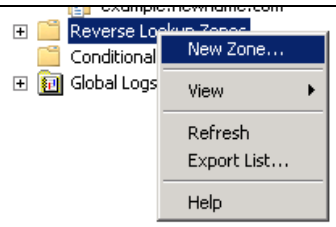




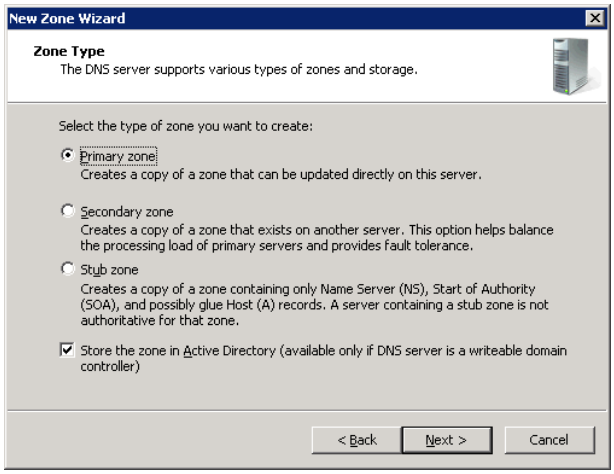
7. Verify that the stub zone has the SOA and NS records.



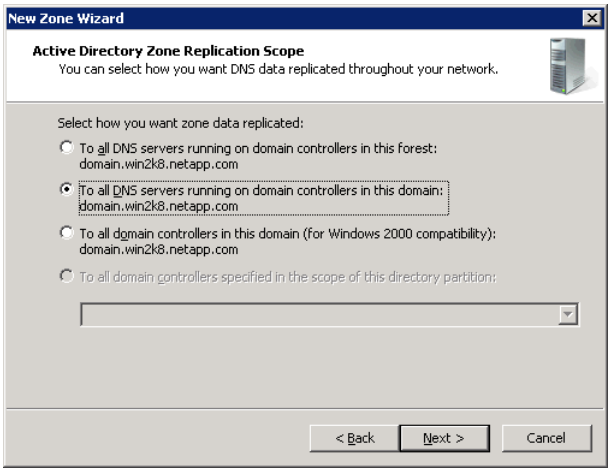
8. Create the reverse lookup zones for the data LIFs.



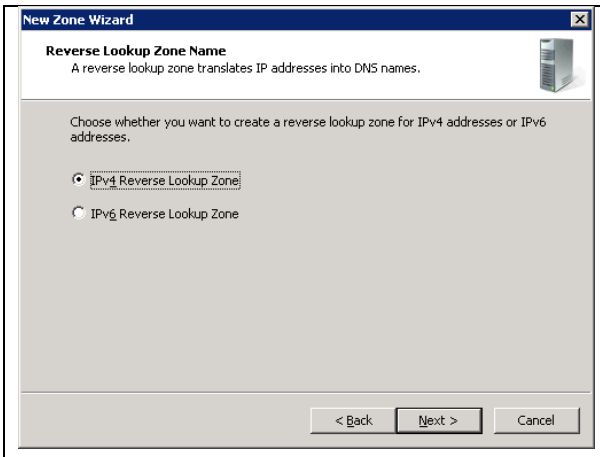
9. Select Primary Zone, because DNS in clustered Data ONTAP cannot service reverse lookups.



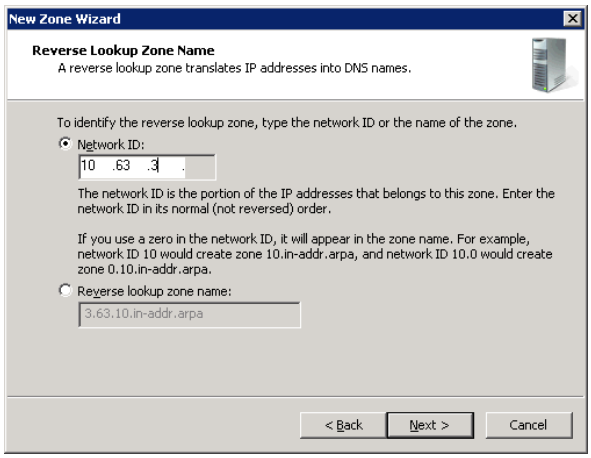
10. Select a zone replication policy to use.



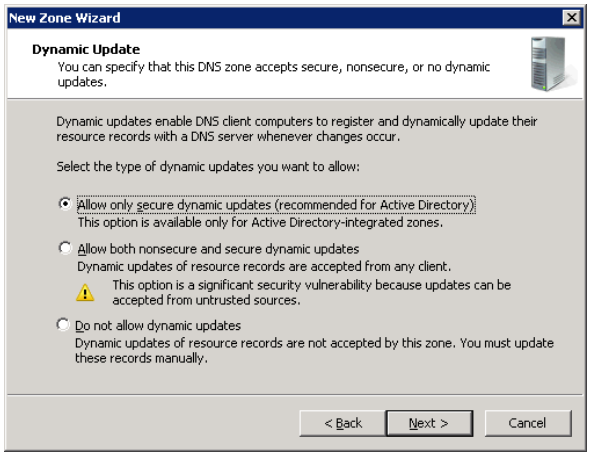
11. Select IPv4 or IPv6 for the lookup zone, depending on what the clustered Data ONTAP version supports and what the data LIFs are using.

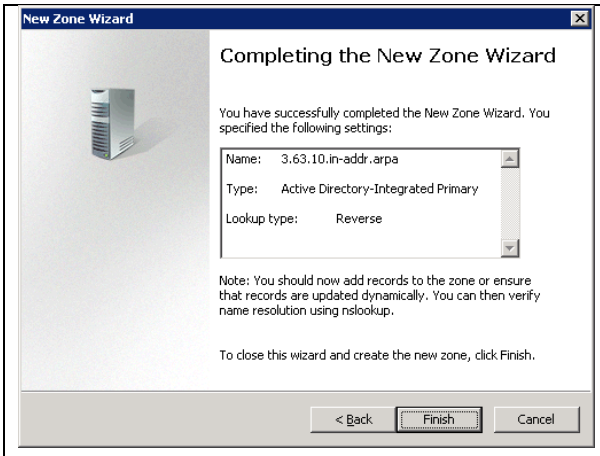


12. Enter the network ID/subnet (the first three octets of the IP address).

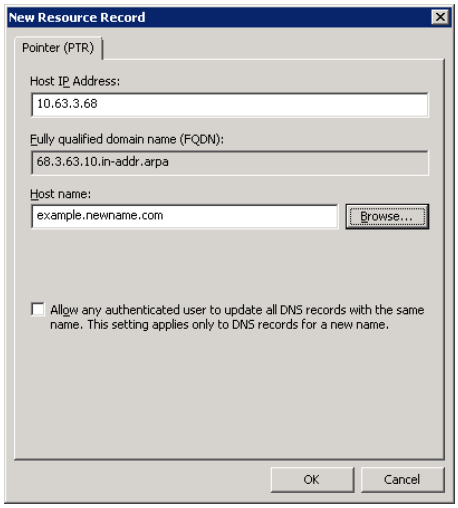


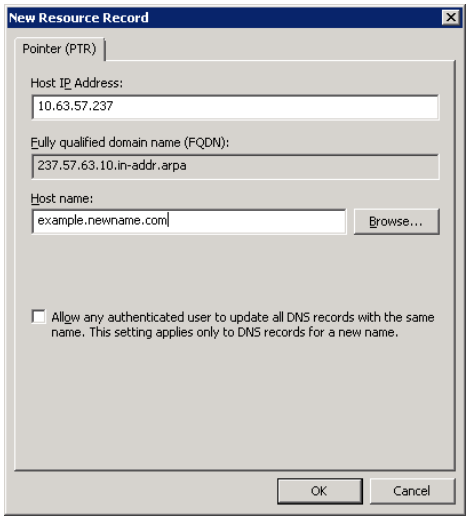
13. Select a dynamic update policy.





14. Repeat steps 8 through 13 for other subnets.
15. Add the PTR records for the data LIFs, because clustered Data ONTAP does not support reverse name lookups.





16. Use `nslookup` to test the forward and reverse lookups in DNS.

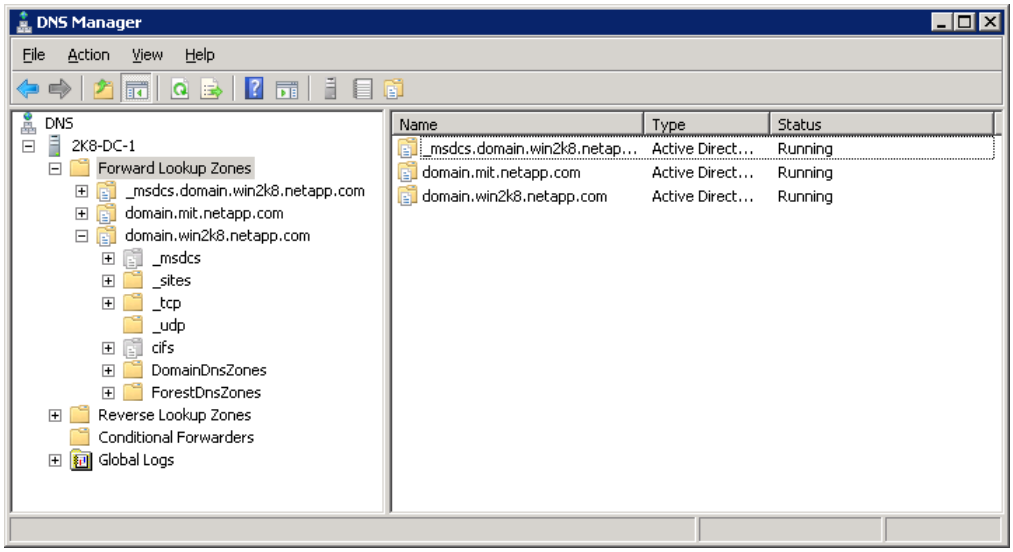
```
C:\>nslookup example.newname.com
Server:  localhost
```



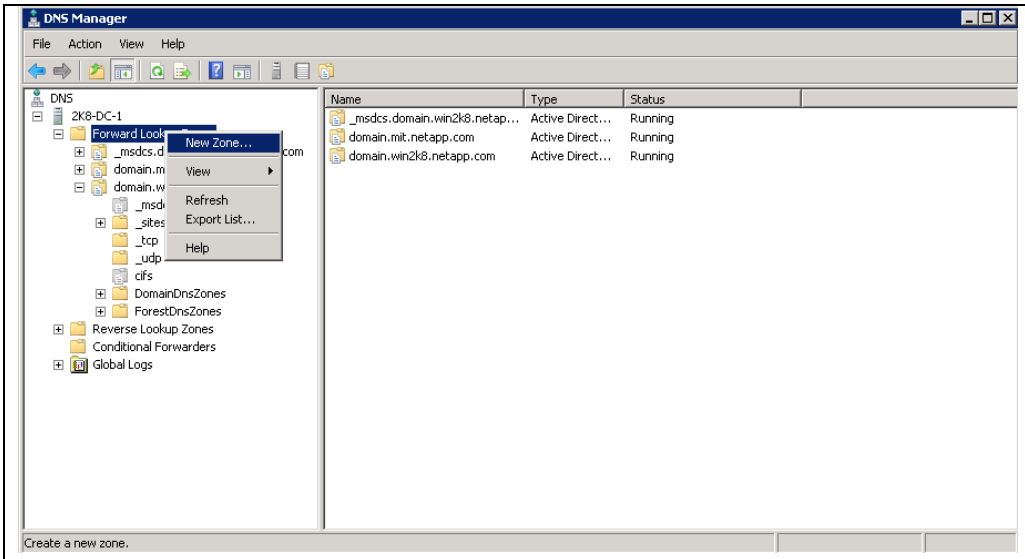
```
Address:  ::1

Name:    example.newname.com
Addresses:  10.63.57.237
          10.63.3.68

C:\>nslookup 10.63.57.237
Server:  localhost
Address:  ::1

Name:    example.newname.com
Address:  10.63.57.237


C:\>nslookup 10.63.3.68
Server:  localhost
Address:  ::1
```

Table 8) Setting up primary zones.

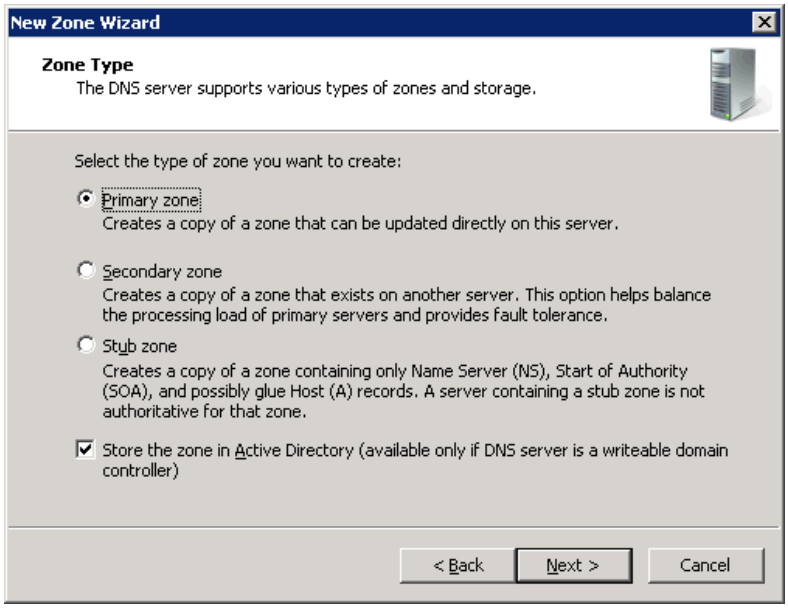
1. Open the DNS Manager console.



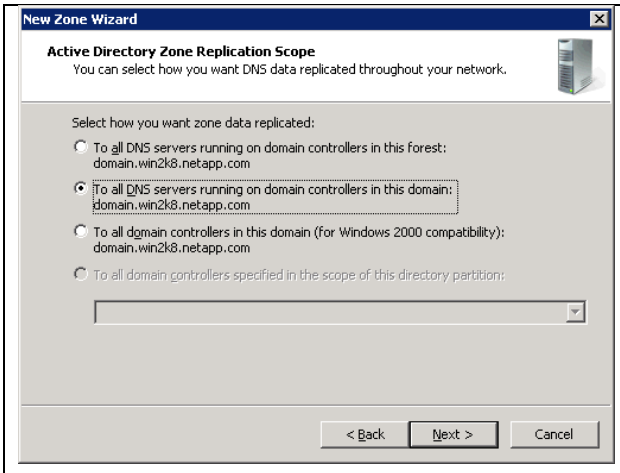
2. Right click Forward Lookup Zones and select New Zone.



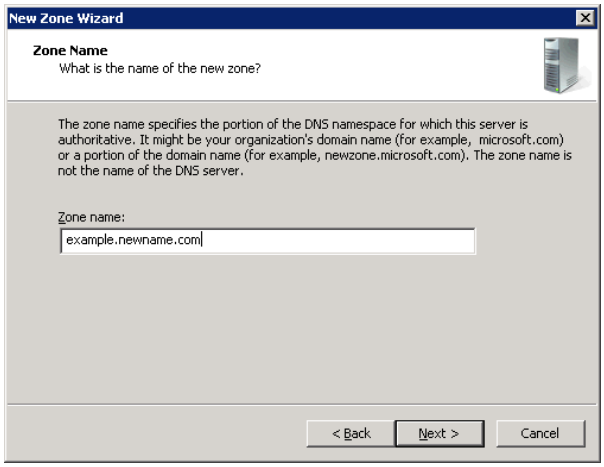
3. Select Primary Zone as the zone.



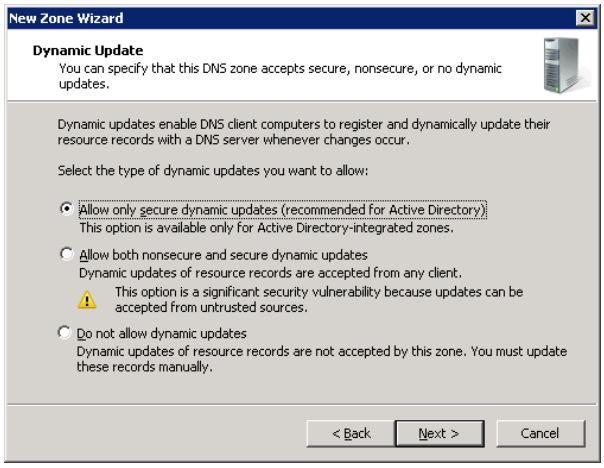
4. Select how zone replication should function.

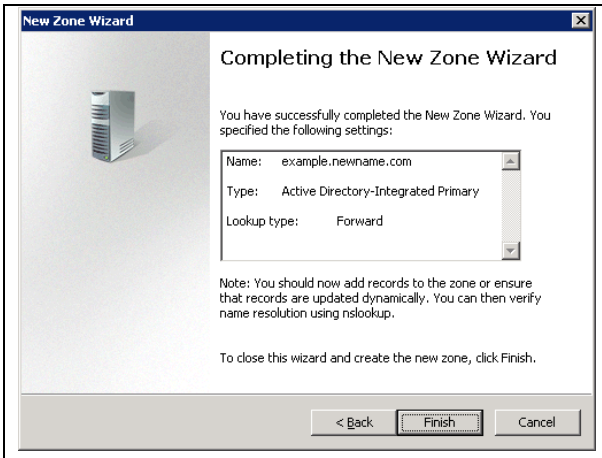


5. Specify the zone name.

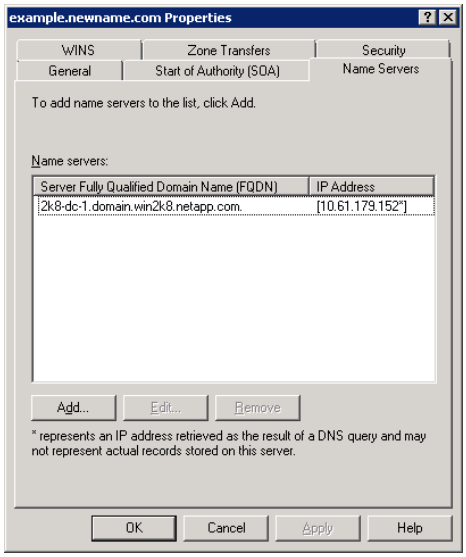


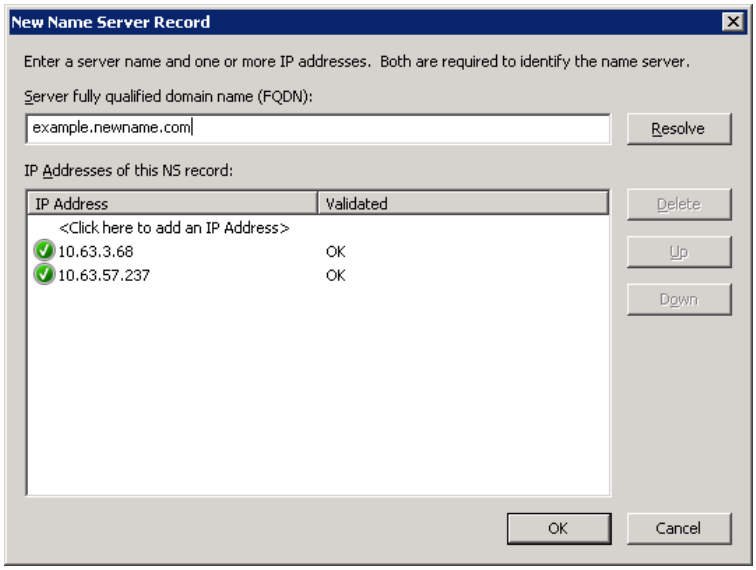
6. Specify how dynamic updates should occur. Note that clustered Data ONTAP does not currently support dynamic updates for DNS.

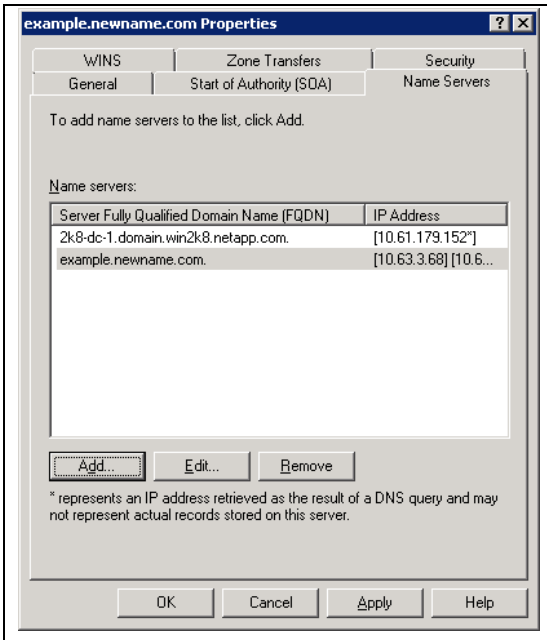




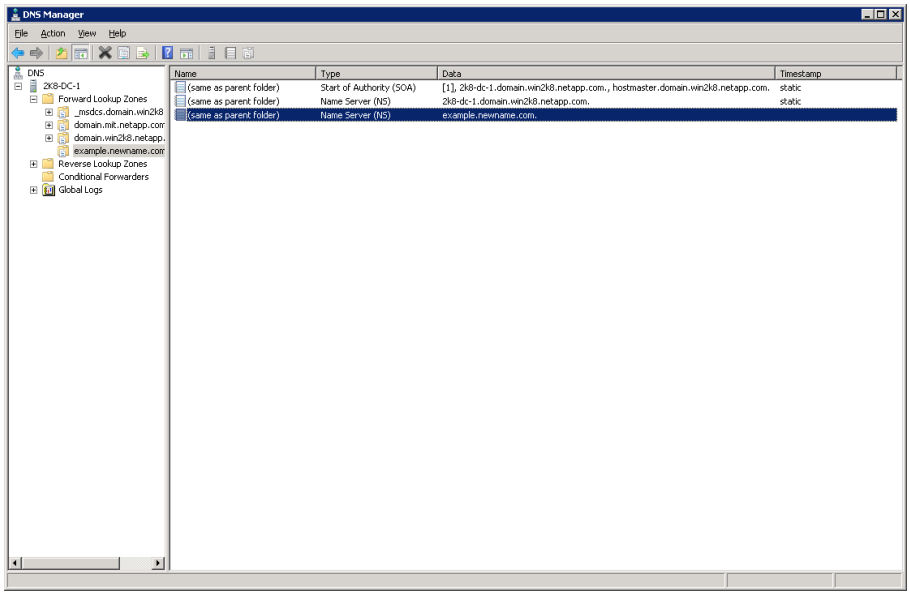
7. Right-click the NS record, add the data LIFs as DNS servers, and specify the zone name.



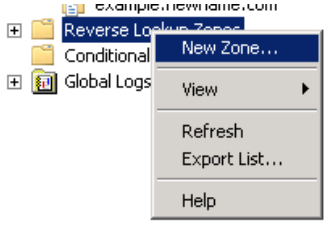




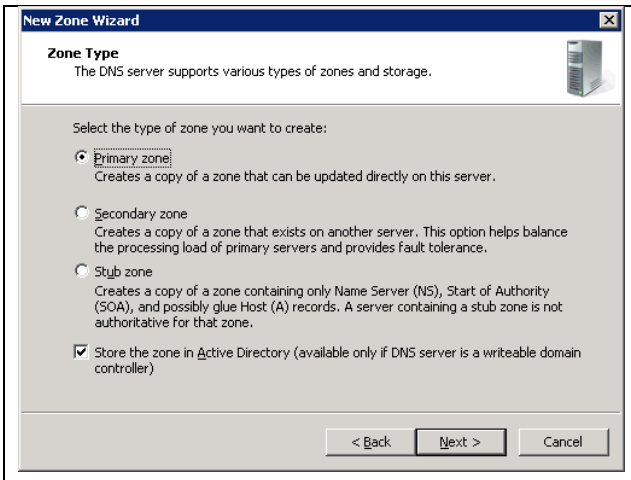
8. Verify that the stub zone has the SOA and NS records.



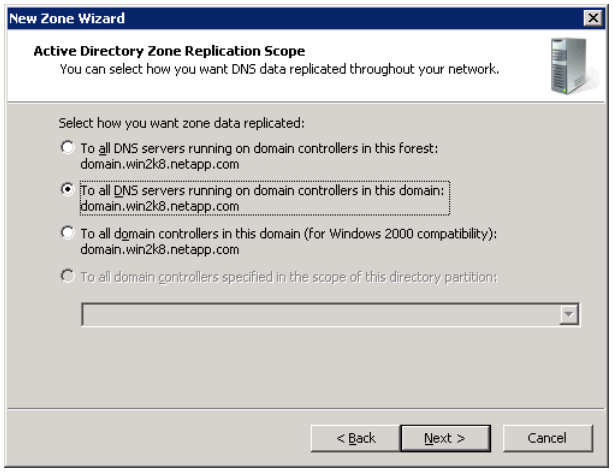
9. Create the reverse lookup zones for the data LIFs.



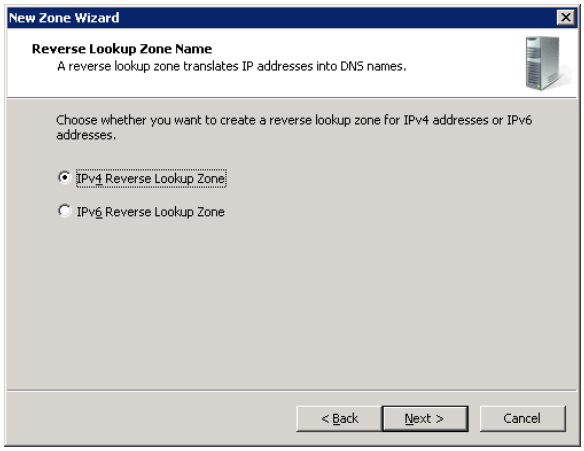
10. Select Primary Zone, because DNS in clustered Data ONTAP cannot service reverse lookups.



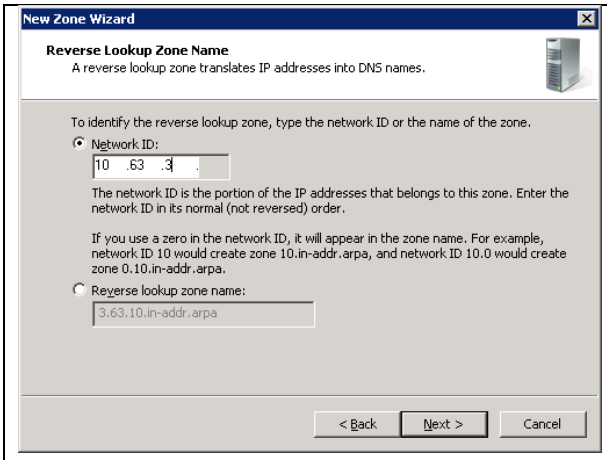
11. Select a zone replication policy to use.



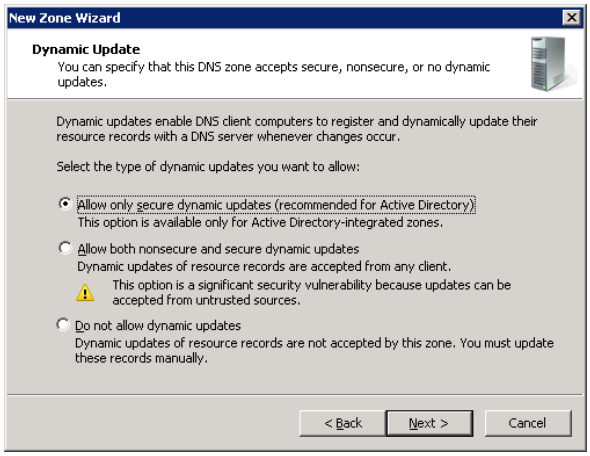
12. Select IPv4 or IPv6 for the lookup zone, depending on what the clustered Data ONTAP version supports and what the data LIFs are using.

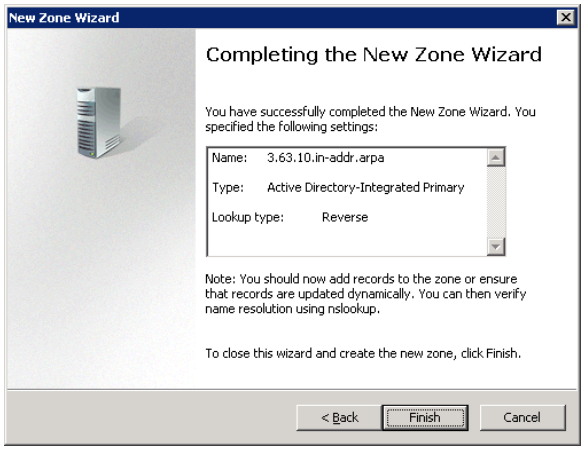


13. Enter the network ID/subnet (the first three octets of the IP address).

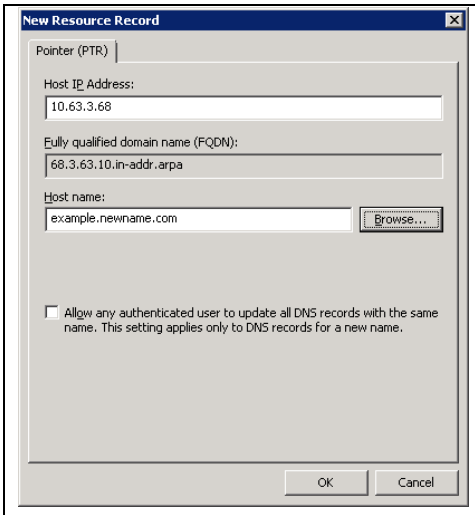


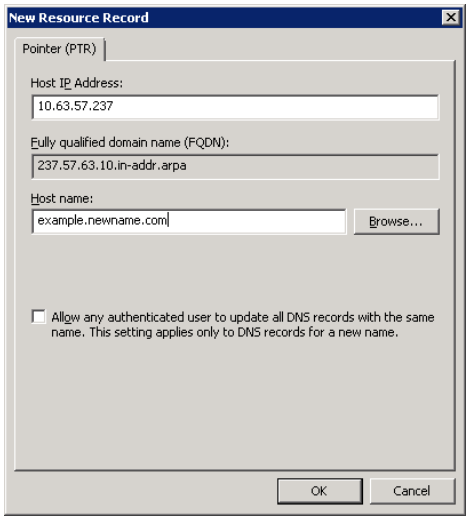
14. Select a dynamic update policy.





15. Repeat steps 9 through 14 for other subnets.
16. Add the PTR records for the data LIFs, because clustered Data ONTAP does not support reverse name lookups.



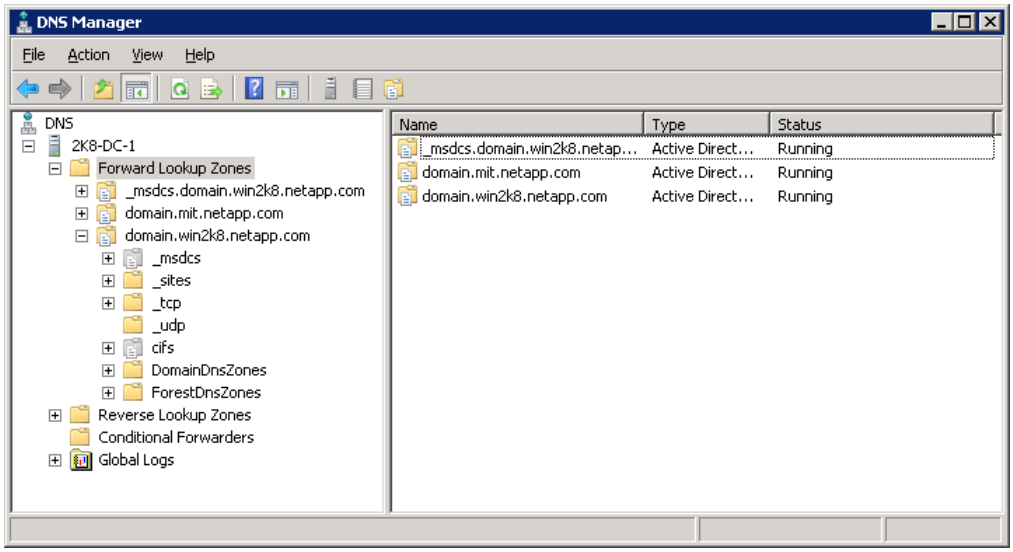


17. Use `nslookup` to test the forward and reverse lookups in DNS.

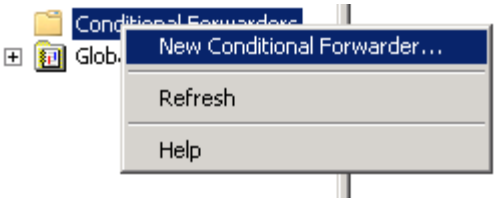
```
C:\>nslookup example.newname.com
Server:  localhost
Address:  ::1

Name:    example.newname.com
Addresses:  10.63.57.237
          10.63.3.68

C:\>nslookup 10.63.57.237
Server:  localhost
Address:  ::1

Name:    example.newname.com
Address:  10.63.57.237


C:\>nslookup 10.63.3.68
Server:  localhost
Address:  ::1
```


Table 9) Setting up conditional forwarders – Windows 2008.

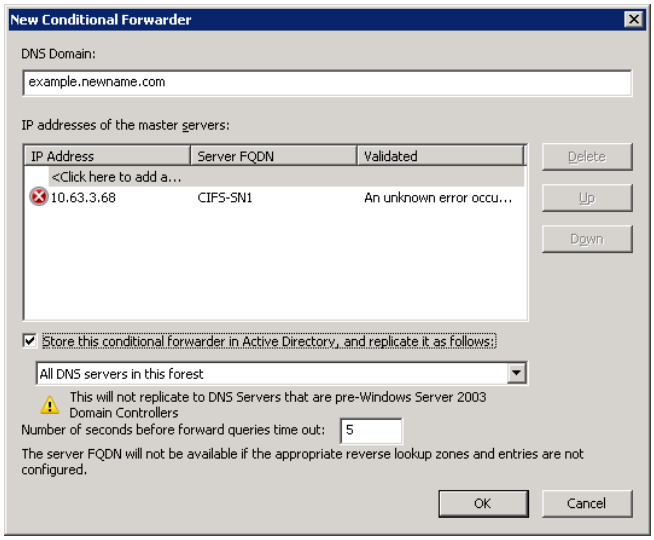
1. Open the DNS Manager console.



2. Right-click Conditional Forwarders and select New Conditional Forwarder.



3. Enter the DNS domain and data LIFs. If an error occurs, the server may be sending SOA record requests, and a stub zone or primary zone should be used.



4. Click OK and use `nslookup` to test the forwarded zone.

```
C:\>nslookup example.newname.com
Server:  localhost
Address:  ::1
```

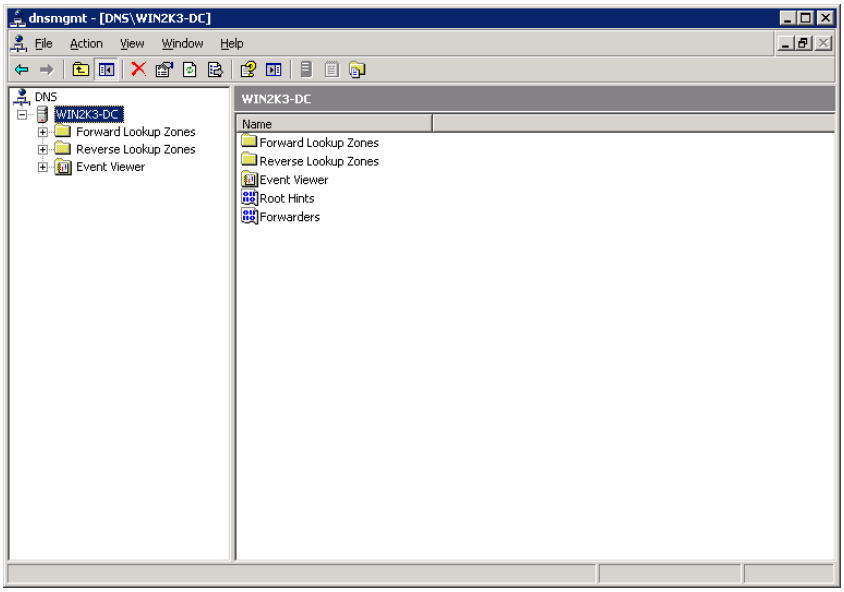


```
Name:    example.newname.com
Addresses:  10.63.57.237
          10.63.3.68

C:\>nslookup 10.63.57.237
Server:  localhost
Address:  ::1

Name:    example.newname.com
Address:  10.63.57.237


C:\>nslookup 10.63.3.68
Server:  localhost
Address:  ::1
```

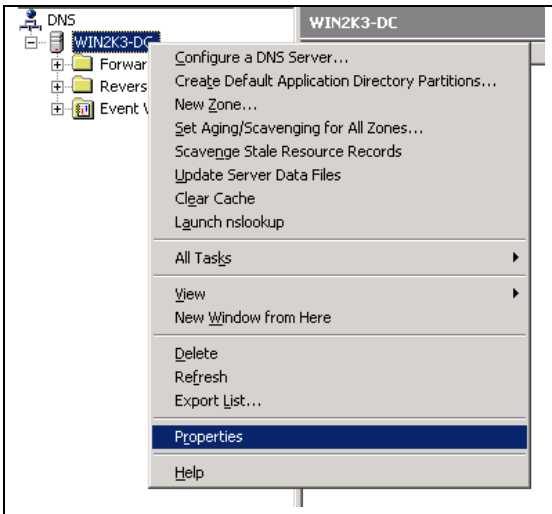
Table 10) Setting up conditional forwarders – Windows 2003.

Windows 2003 does not have a Conditional Forwarder folder in DNS. Instead, use the Forwarder tab on the DNS server.

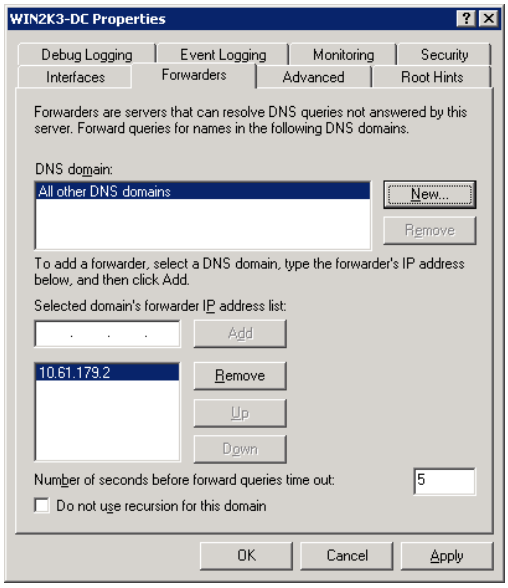
1. Open the DNS Manager console.

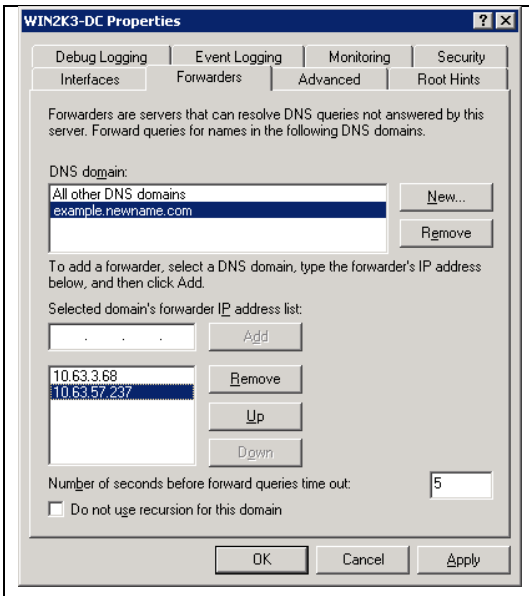


2. Right-click the DNS server and select Properties.



3. Click the Forwarders tab, then click New and add the DNS zone name and specified IP addresses.





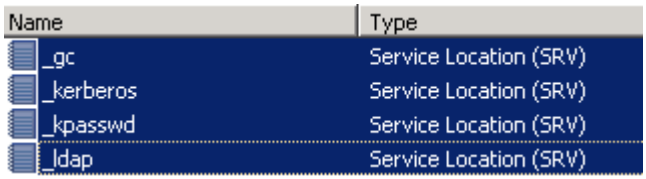
4. Click OK and use `nslookup` to test the new zone.

```
C:\>nslookup example.newname.com
Server:  localhost
Address:  127.0.0.1

Name:    example.newname.com
Address:  10.63.3.68

C:\>nslookup example.newname.com
Server:  localhost
Address:  127.0.0.1

Name:    example.newname.com
Address:  10.63.57.237
```

4.1.16 Adding SRV Records

Some NFS clients (such as SLES) will attempt to use _kerberos-master._udp and/or _kerberos-master._tcp in Kerberos requests. By default, those records do not exist in Active Directory DNS.

The following SRV records should exist by default:



If these SRV records do not exist in an Active Directory domain, contact Microsoft for assistance.

In the above, notice that _kerberos-master does not exist. Create this record for each DC.

A clue that the _kerberos-master SRV record is needed is if an NFS client can successfully kinit with a user, but cannot kinit –k with the machine SPN.

Example:

```
sles11:~# kinit -k root/sles11.domain.netapp.com
kinit(v5): Key table entry not found while getting initial credentials
sles11:~ # kinit administrator
Password for administrator@DOMAIN.NETAPP.COM:
sles11:~ # klist
Ticket cache: FILE:/tmp/krb5cc_0
Default principal: administrator@DOMAIN.NETAPP.COM

Valid starting     Expires            Service principal
05/07/13 10:27:45  05/07/13 20:27:42  krbtgt/DOMAIN.NETAPP.COM@DOMAIN.NETAPP.COM
        renew until 05/08/13 10:27:45


Kerberos 4 ticket cache: /tmp/tkt0
klist: You have no tickets cached
```

A packet trace would show requests for the SRV record failing:

```
10.61.179.162  10.63.98.101 DNS 110 Standard query
SRV _kerberos-master._udp.DOMAIN.NETAPP.COM

"10.63.98.101","10.61.179.162","DNS","195","Standard query response, No such name"
```

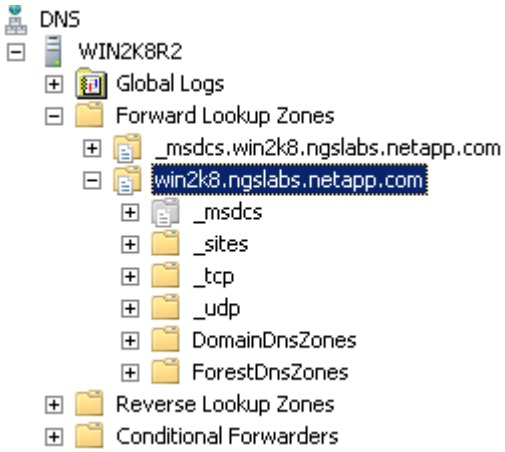
Creating SRV Records for Kerberos Master*

The Kerberos Master SRV records are used by some NFS clients to tell them that the server is a KDC. Most clients do not require these SRV records, but if an NFS client does, the following steps show how to create them.

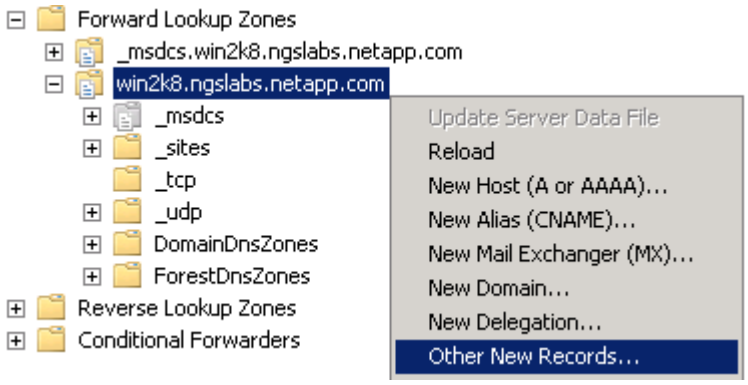
*This step is only required if the NFS client cannot use Kerberos without these records.

Table 11) Creating SRV records for Kerberos-Master.

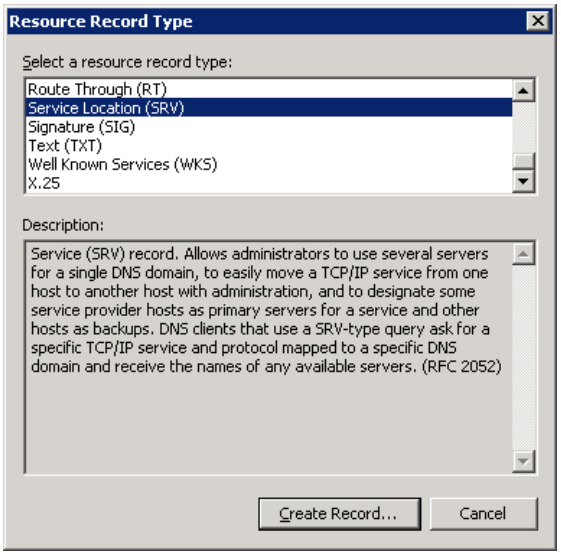
1. Log in to the DNS server.
2. Open the DNS GUI and navigate to the DNS domain folder under "Forward Lookup Zones."



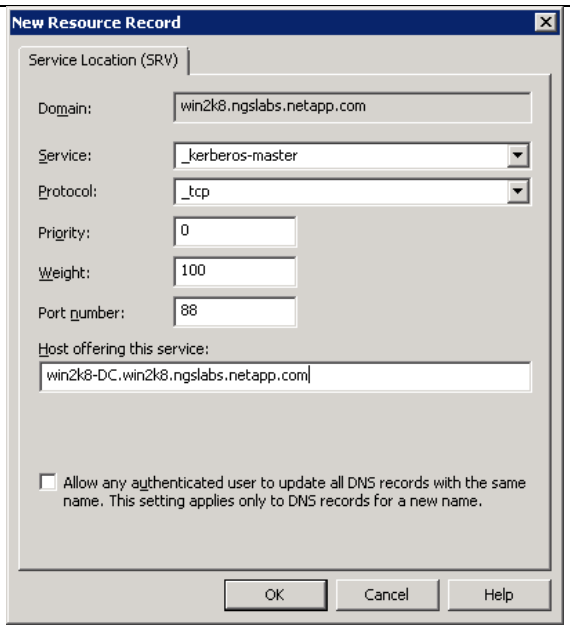
3. Click on the DNS domain. Right-click and select “Other New Records.”



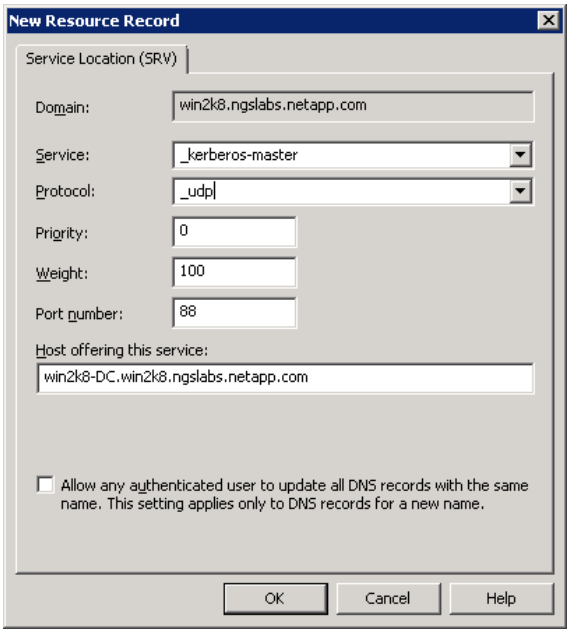
4. Select the Record Resource Type of “Service Location (SRV).”



5. Click “Create Record” and then fill in the fields. The “service” field for _kerberos-master does not exist; it must be manually typed in. The “priority,” “weight,” and “port” fields are the same as the normal _kerberos record. The “Host” field should be the FQDN of the KDC. See the following example for details.



Repeat the process for the _udp record:



Create records for each DC in the domain, since any of them could serve Kerberos requests.

6. Click “OK” and then test nslookup for the SRV records:

```
[client] # nslookup
> set type=SRV
> _kerberos-master._tcp.DOMAIN.NETAPP.COM
Server:         10.63.98.101
Address:        10.63.98.101#53

_kerberos-master._tcp.DOMAIN.NETAPP.COM
service = 0 100 88 win2k8-dc.domain.netapp.com.
```


Table 12) Creating SRV records for Kerberos-Master (dnscmd).

1. Create the TCP record.

```
C:\ >dnscmd /RecordAdd domain.netapp.com _kerberos-master._tcp SRV 0 100 88
win2k8DC.domain.netapp.com

Add SRV Record for _kerberos-master._tcp.domain.netapp.com at domain.netapp.com
Command completed successfully.
```

2. Create the UDP record.

```
C:\ >dnscmd /RecordAdd domain.netapp.com _kerberos-master._udp SRV 0 100 88
win2k8DC.domain.netapp.com

Add SRV Record for _kerberos-master._udp.domain.netapp.com at domain.netapp.com
Command completed successfully.
```

LDAP SRV Records

LDAP SRV records should exist in the domain. By default, Active Directory will create these records.

To see if the SRV records exist:

```
[root@centos64/]# dig SRV _ldap._tcp.domain.netapp.com

; <<>> DiG 9.8.2rc1-RedHat-9.8.2-0.17.rc1.el6 <<>> SRV _ldap._tcp.domain.netapp.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 43894
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 0, ADDITIONAL: 2

;; QUESTION SECTION:
;_ldap._tcp.domain.netapp.com. IN        SRV

;; ANSWER SECTION:
_ldap._tcp.domain.netapp.com. 600 IN SRV 0 100 389 2k8-dc-1.domain.netapp.com.
_ldap._tcp.domain.netapp.com. 600 IN SRV 0 100 389 2k8-dc-2.domain.netapp.com.

;; ADDITIONAL SECTION:
2k8-dc-1.domain.win2k8.com. 3600 IN A    10.61.179.152
2k8-dc-2.domain.win2k8.com. 3600 IN A    10.61.179.155

;; Query time: 0 msec
;; SERVER: 10.61.179.155#53(10.61.179.155)
;; WHEN: Wed May 22 18:00:25 2013
;; MSG SIZE  rcvd: 191
```

If LDAP SRV records do not exist, contact Microsoft to troubleshoot the issue. To use LDAP SRV records with SSSD, consult the SSSD configuration section of this document.

Allowing DES Encryption

Windows 2008 R2 servers disable DES encryption for Kerberos by default. Windows 2003 and Windows 2008 (non-R2) allow DES encryption by default, so no security policy configuration is required for Windows 2003, Windows 2003 R2, or Windows 2008 base servers. For more information, see the following on Windows Encryption Types.

If DES is not allowed, the following may be seen from a client trying to initiate a Kerberos ticket:

```
[root@centos6-4 sysconfig]# kinit -k root/centos6-4.domain.netapp.com@DOMAIN.NETAPP.COM
```



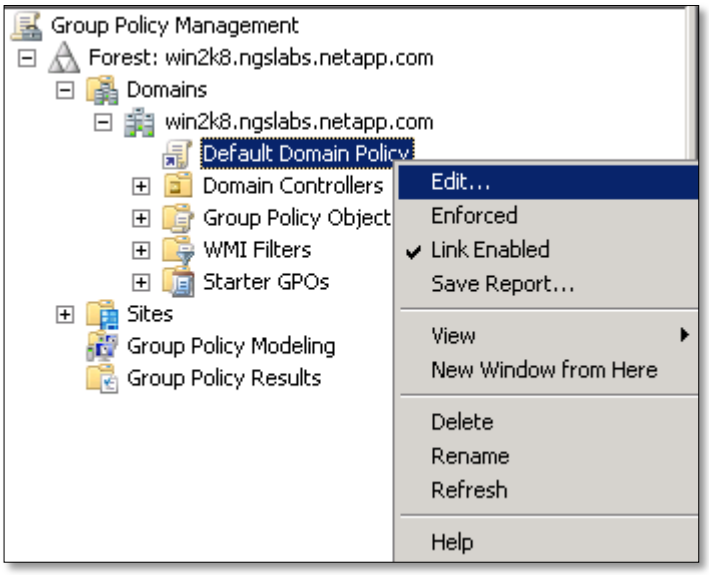
```
kinit: KDC has no support for encryption type while getting initial credentials
```

Enabling DES in Windows 2008 R2

This will allow DES encryption for the entire domain. Once this is done, the registry key `HKLM\Software\Microsoft\Windows\CurrentVersion\Policies\System\Kerberos\Parameters\SupportedEncryptionTypes` is changed. If this is not done, DES requests will fail, meaning NFS mounts using Kerberos with clustered Data ONTAP will fail.

Table 13) Allowing DES encryption types in Windows 2008.

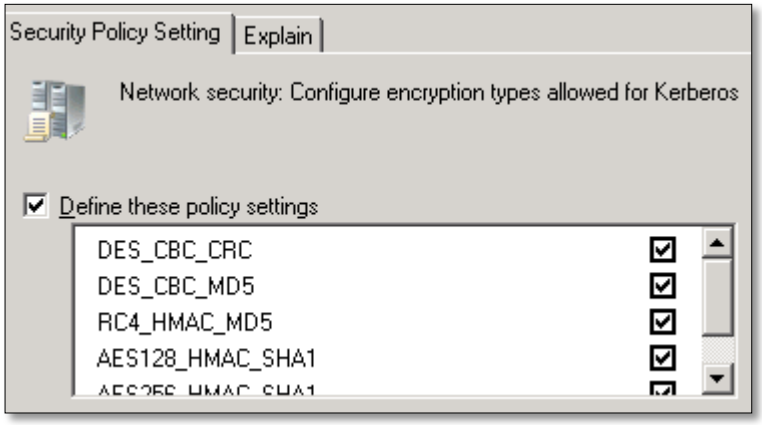
1. On the Windows 2008 R2 domain controller, go to Start -> Run and type “gpedit.msc.”
2. Expand “Domains” and expand the domain the DC belongs to.
3. Right-click on “Default Domain Policy” and select “Edit.”



4. Navigate to Computer Configuration -> Policies -> Windows Settings -> Security Settings -> Local Policies -> Security Options.

5. Select the “Network security: Configure encryption types allowed for Kerberos” option and double-click.

6. Select the encryption types desired. Since DES is needed and the rest are more secure than DES, select them all. Selecting DES only will prevent other enctypes entirely for the domain.



7. Click Apply. A reboot is not required to apply the change.

Disabling DES in Windows 2008 R2

To disable DES in Windows 2008 R2, it is not enough to modify the GPO by unchecking the policy settings. This does not change the registry value on the DC.

To disable DES, the registry value must be modified manually. The following figures show the value when DES is enabled and when it is the default setting. When DES is not enabled, this registry key does not exist. To disable DES, uncheck the DES boxes in the policy or delete the `HKLM\Software\Microsoft\Windows\CurrentVersion\Policies\System\Kerberos` key entirely.

Figure 4) DES enabled.

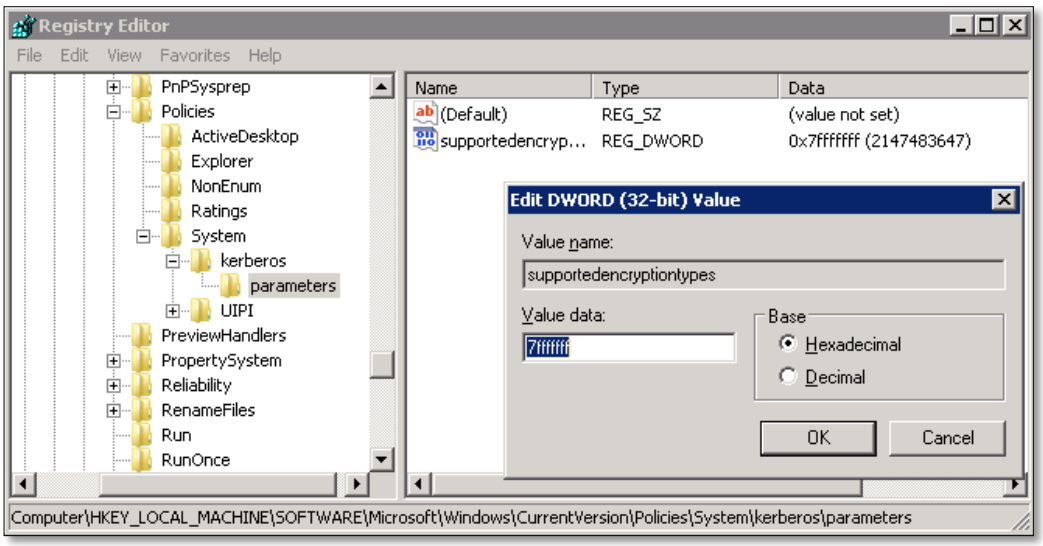
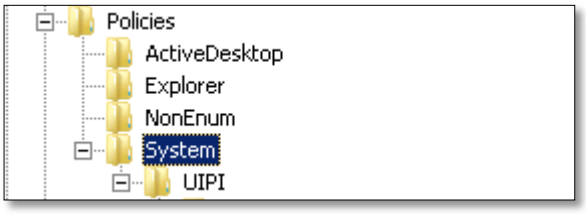


Figure 5) Default value.

Note that there is no Kerberos key.



4.1.17 Creating Principals/Keytab Files in Active Directory

This section describes how to create principals for the NFS clients to be used with Kerberos. Principals need to be created so that the KDC can verify the identity of the client requesting access. These principals will work in conjunction with keytab files to pass the encrypted secret password hash to allow Kerberos tickets to be granted. A machine object will be created but will not be a valid Kerberos principal until the keytab is created via ktpass or a SPN is manually defined.

Creating machine accounts can be done from the Active Directory Users and Computers GUI, from the command line prompt (cmd), or from Windows PowerShell®. User accounts can also be used as Kerberos principals.

This procedure applies for all domain controllers starting with Windows 2003.

Why Machine Accounts?

SPNs can be attached to user accounts or to machine accounts. This document uses machine accounts for the following reasons:

- No password entry is required; keytab authentication only.
- Logic: Machine accounts for machines make more sense.
- Multiple SPNs can be assigned to a machine account.

The downside of using machine accounts is that they are not as easily customizable as user accounts. User accounts in Active Directory provide GUI access to allow DES, forego preauthentication, and so on. Machine accounts require modification via ADSI or the advanced features in AD Users and Computers. Additionally, although machine accounts do not normally expire passwords, when a keytab file is created with ktpass, the machine account password can expire and the keytab file must be created.

Machine account password resets for accounts that have had ktpass run on them fall under the domain password policy, so right-clicking and selecting Reset Account fails unless the policy is set to not enforce password policies for the OU the machine accounts are located in. Rerun ktpass and re-create the krb5.keytab file to get around this limitation.

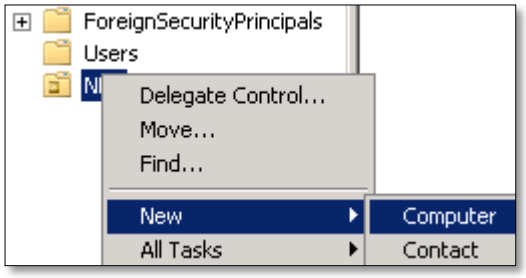
Creating the Machine Account

Table 14) Creating machine accounts in Active Directory (GUI).

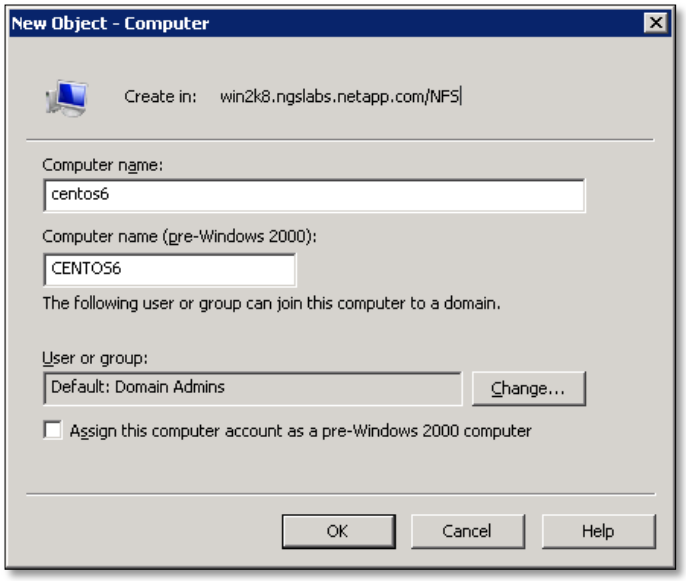
1. Go to Start -> Administrative Tools and select Active Directory Users and Computers. 2. In the GUI, select the Organizational Unit (OU) Computers or create a sub-OU. By default, all machine accounts live in “Computers.”

3. To create a sub-OU (optional), right-click on the domain and select New -> Organizational Unit.

4. In the selected OU, right-click and select New -> Computer.



5. Create the new computer object with the host name of the NFS client. Click OK to apply the change.



Note: After creating the machine account in this manner, the attributes will need to be modified to allow DES.

Table 15) Creating machine accounts in Active Directory (dsadd).

1. Open the cmd prompt by going to Start -> Run and typing “cmd.”
2. Run the following command on the domain controller to create a new OU (optional):

```
C:\> dsadd ou [OU=name,DC=domain,DC=netapp,DC=com]
```

Example:

```
C:\>dsadd ou OU=NFSClients,DC=domain,DC=netapp,DC=com
```

3. Run the following command on the domain controller to create a computer account in an OU:

```
C:\>dsadd computer [CN=name,OU=org_unit,DC=domain,DC=netapp,DC=com]
```

Example:

```
C:\>dsadd computer CN=centos6,OU=NFS,DC=domain,DC=netapp,DC=com
```

Note: After creating the machine account in this manner, the attributes will need to be modified to allow DES.

Table 16) Creating machine accounts in Active Directory (PowerShell).

1. Log in to the domain controller and open PowerShell.

2. Type the following (Click for more info on New-ADComputer):

```
PS C:\> import-module activedirectory
PS C:\> New-ADComputer -Name [computername] -SAMAccountName [computername] -DNSHostName
[computername.dns.domain.com] -OtherAttributes @{'userAccountControl'= 2097152;'msDS-
SupportedEncryptionTypes'=25}
```

Note: This object should not be assigned an SPN or a UPN. Ktpass will assign both once it is run to create the keytab file.

Modifying Machine Account Attributes

This section describes how to modify the machine accounts created for Kerberized NFS. Clustered Data ONTAP currently supports only DES/3DES, but Windows supports only DES. Thus, it is necessary to configure machine accounts used in the Kerberized NFS processes to allow DES encryption. Windows 2008 R2 disables DES by default, as it is considered less secure than other encryption types.

The following section will accomplish the following:

- Modifying the NFS server machine account to use DES only and allowing DES as a supported enctype
- Modifying the NFS client machine accounts to allow DES as a supported enctype

Why modify the machine accounts?

Modifying the machine account is necessary so that Kerberos authentication requests leverage DES when communicating with the cluster to avoid authentication failures.

Example of an authentication failure in SecD logs:

```
Wed May 22 2013 11:15:57 -04:00 [kern_secd:info:39727] | [000.002.049]  debug:  GSS_S_COMPLETE:
client = 'root/nfsclient.domain.netapp.com@DOMAIN.NETAPP.COM'  { in acceptGssToken() at
gss/secd_gss_accept_token.cpp:288 }
Wed May 22 2013 11:15:57 -04:00 [kern_secd:info:39727] | [000.002.124]  ERR  :  Unsupported
signing algorithm 17.  { in parseKrb5ContextToken() at gss/secd_gss_parsekrb5.cpp:106 }
```

Since NFS clients generally support most enctypes, modifying the NFS server machine account is the most logical approach—this avoids the need to modify scores of machine accounts when stronger enctypes become available. This also provides a hybrid security mechanism that allows stronger enctypes when they are available.

In the `klist -e` output below, the Kerberos ticket uses AES while the NFS mount uses DES. When this client mounts a clustered Data ONTAP data LIF, it obtains the TGT using AES 256, the strongest enctype available. Because the clustered Data ONTAP Active Directory machine account for Kerberos has been restricted to DES only and the client keytab has DES listed as a supported enctype, it uses AES for the client portion of the ST and DES for the server portion of the ST. This is much more secure than using DES for all Kerberos encryption. Furthermore, as support for stronger encryption types becomes available, the client configuration will **not** have to change.

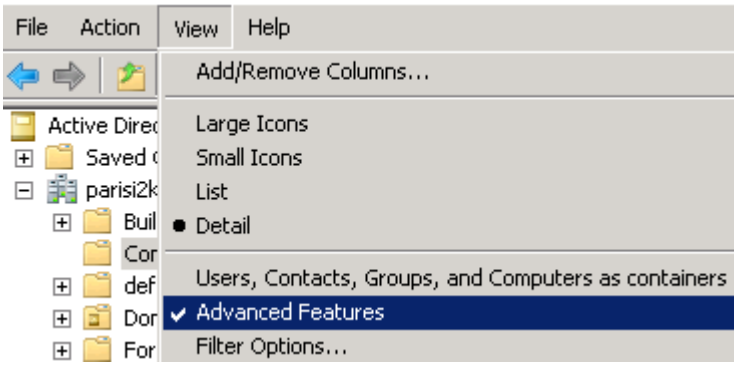
```
sh-4.1$ klist -e
Ticket cache: FILE:/tmp/krb5cc_50
Default principal: ldapuser@DOMAIN.NETAPP.COM

Valid starting     Expires            Service principal
05/21/13 11:16:37  05/21/13 21:16:12  krbtgt/DOMAIN.NETAPP.COM@DOMAIN.NETAPP.COM
        renew until 05/22/13 11:16:37, Etype (skey, tkt): aes256-cts-hmac-sha1-96, aes256-cts-
hmac-sha1-96
05/21/13 11:16:27  05/21/13 21:16:12  nfs/kerberos.domain.netapp.com@DOMAIN.NETAPP.COM
        renew until 05/22/13 11:16:37, Etype (skey, tkt): des-cbc-crc, des-cbc-md5
```

Configuring the machine account objects as well as creating a proper keytab file allows multiple encryption types to be supported on an NFS client.

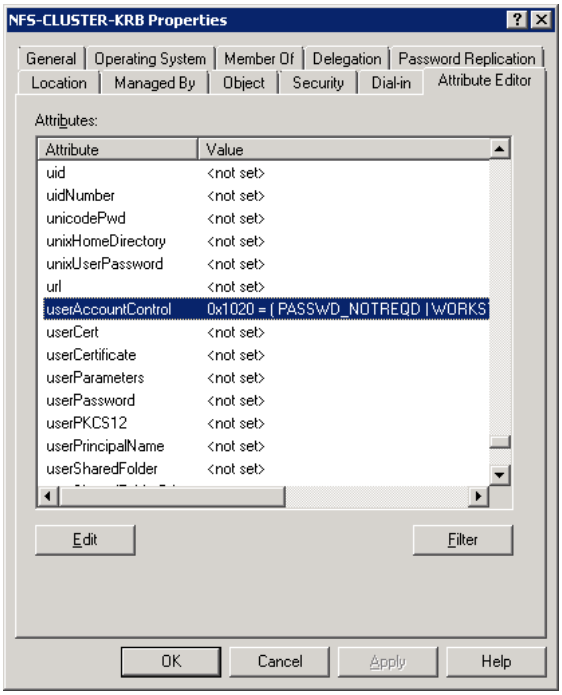
The following shows how to modify the account using the Attributes Editor tab as well as using ADSI Edit and `ldifde` commands. Using the Attributes Editor tab is the preferred method to do this, because it is the least dangerous. If ADSI Edit is used, exercise caution when modifying domain objects.

Table 17) Modifying the NFS server machine account to use/support DES_CBC_MD5 (Attributes Editor).

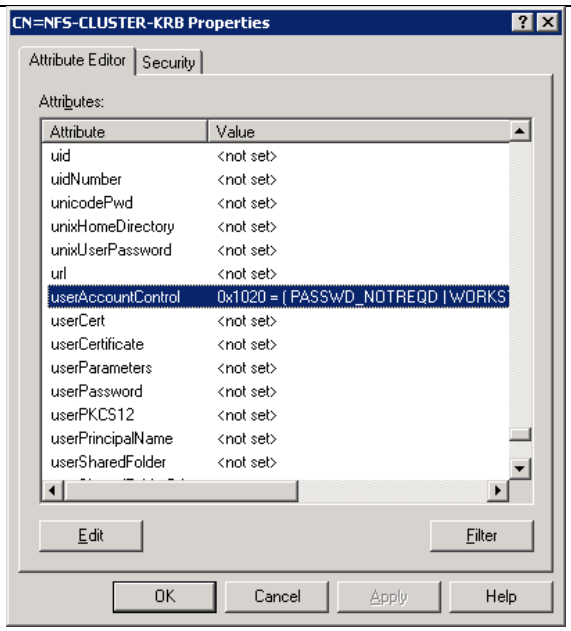
1. Log in to the domain controller and open “Active Directory Users and Computers.”
2. Click on “View” and select “Advanced Features.”



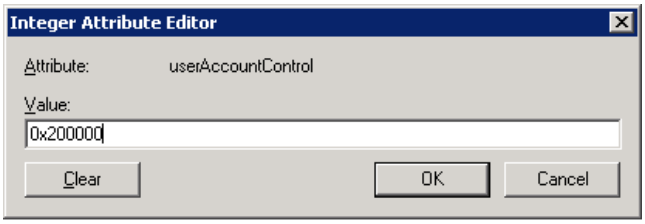
3. Once this is done, there will be a new tab under the machine account properties called “Attributes Editor.”



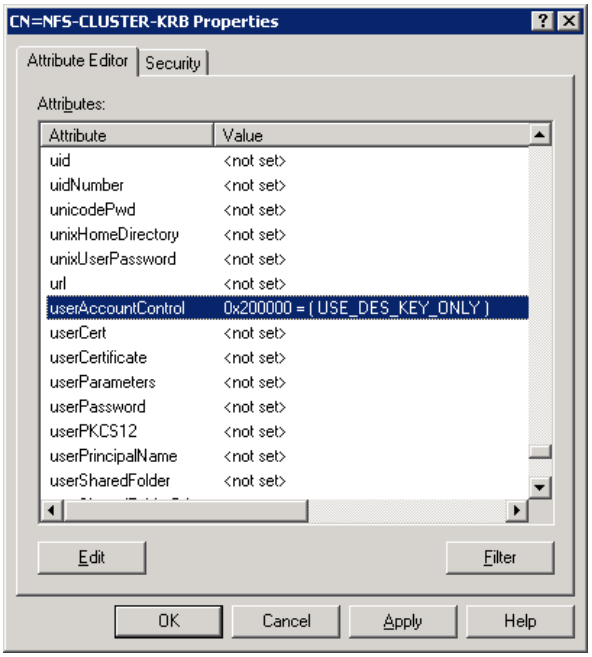
4. Navigate to the “userAccountControl” attribute.



5. Click on “Edit” and change the value to 0x200000 (USE_DES_KEY_ONLY).



6. Click “OK” and verify the change:



7. In Windows 2008 R2, an attribute called “msDS-SupportedEncryptionTypes” was added. This option should be set to allow all enctypes. Change this value to 25 (0x19 in hex) to allow all encryption types for the machine account. (This option did not exist prior to Windows 2008 R2.)

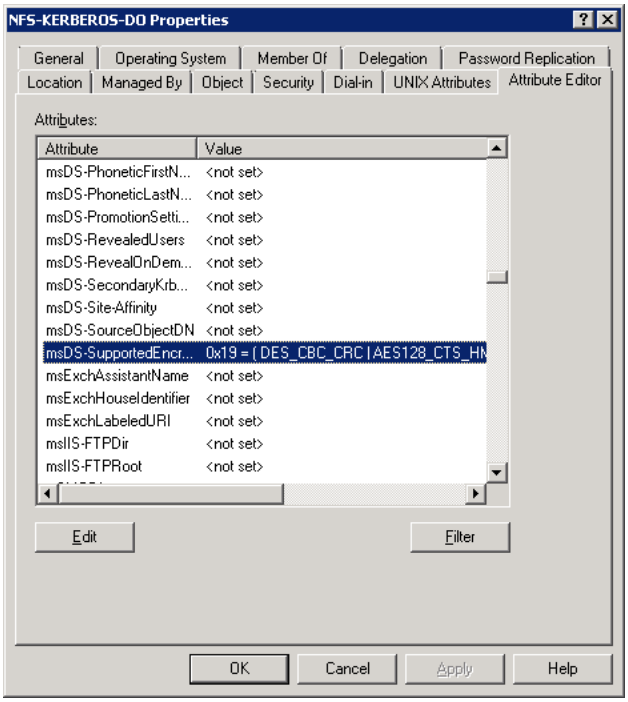
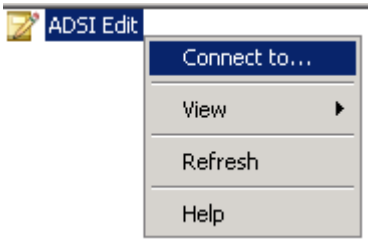
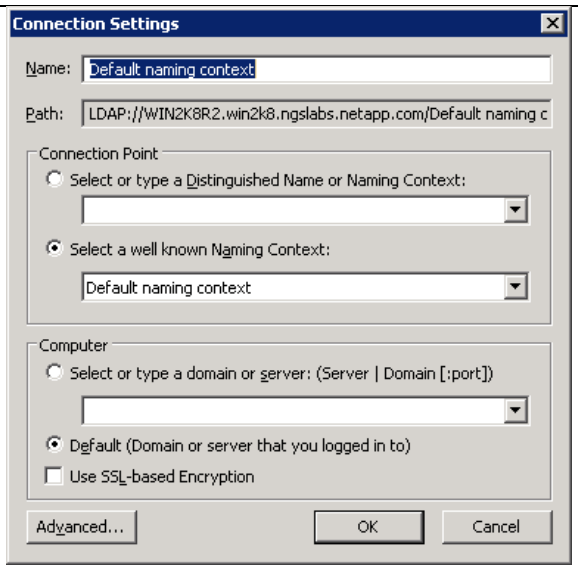


Table 18) Modifying the NFS server machine account to use/support DES_CBC_MD5 (ADSIedit).

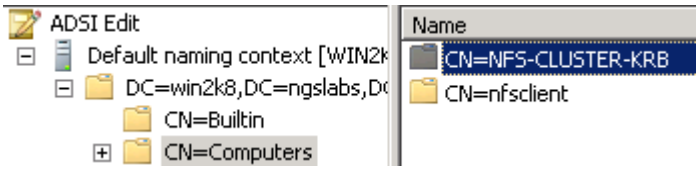
1. Log in to the domain controller; go to Start -> Run and type “adsiedit.msc.”
2. Connect to the Active Directory database by right-clicking and selecting “Connect to.”



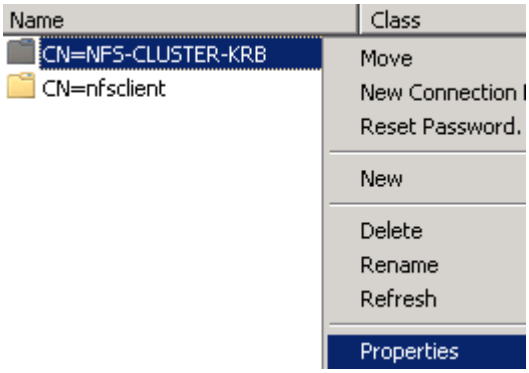
3. Leave the defaults and click “OK.”



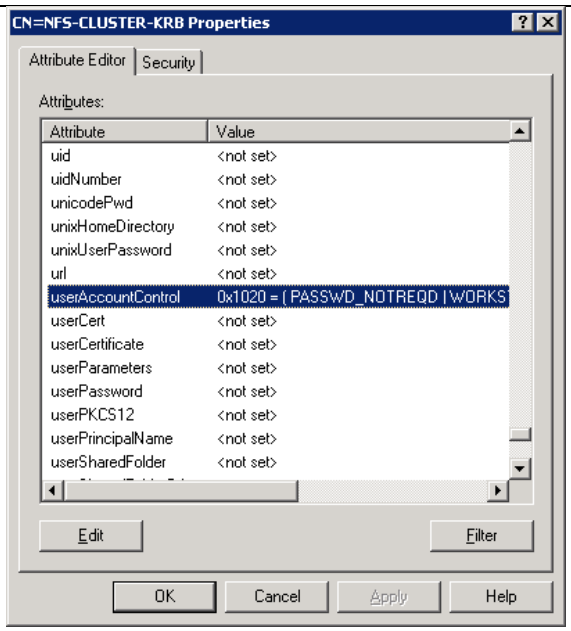
4. Navigate to the Computers container (where the NFS machine account was created).



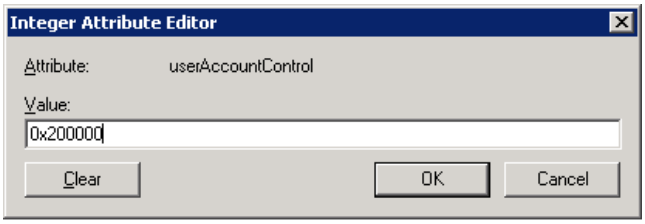
5. Right-click on the object and select “Properties.”



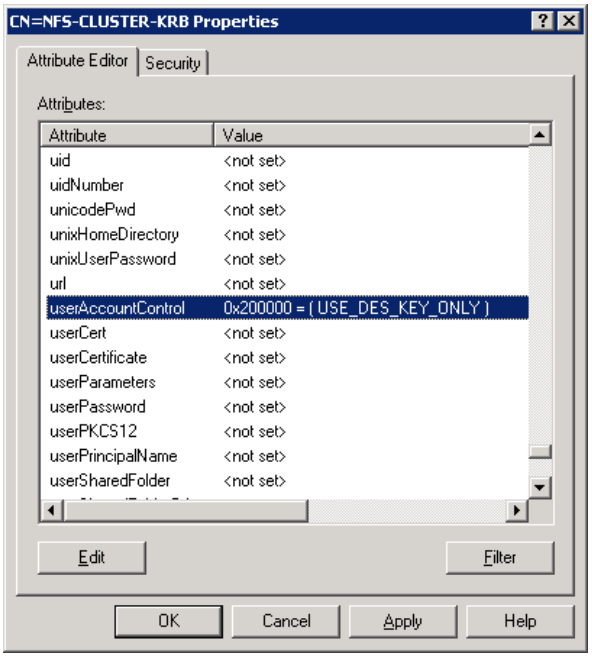
6. Navigate to the “userAccountControl” attribute.



7. Click on “Edit” and change the value to 0x200000 (USE_DES_KEY_ONLY).



8. Click “OK” and verify the change:



9. In Windows 2008 R2, an attribute called “msDS-SupportedEncryptionTypes” was added. This option should be set to allow all enctypes. Change this value to 25 (0x19 in hex) to allow all encryption types for the machine account. (This option did not exist prior to Windows 2008 R2.)

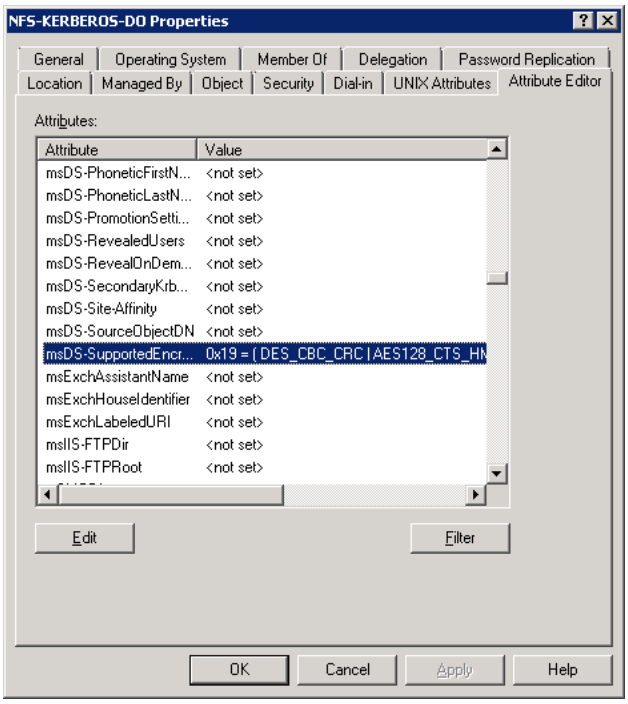


Table 19) Modifying the NFS server machine account to use/support DES_CBC_MD5 (import using ldifde).

1. Log in to the domain controller and open a text editor such as WordPad.

2. Create a file named account_name_des.ldf with the following entries (modified with the account information):

```
dn: CN=NFS-KERBEROS-DO,CN=Computers,DC=domain,DC=netapp,DC=com
changetype: modify
replace: userAccountControl
userAccountControl: 2097152
-

dn: CN= NFS-KERBEROS-DO,CN=Computers,DC=domain,DC=netapp,DC=com
changetype: modify
replace: msDS-SupportedEncryptionTypes
msDS-SupportedEncryptionTypes: 25
-
```

   NOTE: The above includes a dash and return carriage after each entry. These entries are required for the modification to work properly.

3. Save the file and open the cmd prompt by going to Start -> Run and typing “cmd.”

4. Run the following command to import the entry, replacing the file below with the name and location of the file that was created:


```
ldifde -i -f C:\account_name_des.ldf
```

5. Verify that the account has changed the attributes with the following command, replacing the [entries] with the LDAP server's entries:

```
C:\>ldifde -d "[DC=domain,DC=com]" -f DES_output.txt
-r "(&(objectCategory=computer)(objectClass=user)(name=[computername]))"
-l "msDS-SupportedEncryptionTypes"
```

Example:

```
C:\>ldifde -d "DC=domain,DC=netapp,DC=com" -f DES_output.txt
-r "(&(objectCategory=computer)(objectClass=user)(name=centos6))"
-l "msDS-SupportedEncryptionTypes"
```

Table 20) Creating machine accounts in Active Directory (PowerShell).

1. Log in to the domain controller and open PowerShell.

2. Type the following (Click for more info on set-ADComputer):

```
PS C:\> import-module activedirectory
PS C:\> Set-ADComputer -Identity [NFSservername] -Replace
@{'userAccountControl'=2097152;'msDS-SupportedEncryptionTypes'=25}
```

Note: For the NFS client machine account, repeat the above steps for **the msDS-SupportedEncryptionTypes value only**.

For more information about the userAccountControl and msDS-SupportedEncryptionTypes values, see the section entitled "About the machine account attributes" in the Appendix of this document.

For information on DES and other enctypes, see the section entitled "Kerberos encryption types."

Creating the NFS Client Keytab File

To be able to use a principal object for Kerberos with an NFS client, a keytab file must be created, mapped to an Active Directory account and copied to the NFS client. This task requires the following attributes:

- SPN/UPN in primary/instance@REALM format
- A mapped user/machine account
- The crypto method to be used
- A password (can be set to random)
- Principal type
- File name to dump contents

In this example, the SPN/UPN of root/hostname@REALM is used.

During the keytab creation process, a UPN and an SPN are assigned to the NFS client Active Directory machine account. **Until this is done, the computer object isn't actually a valid Kerberos principal.**

Note: When creating the keytab, use caution. If you run the ktpass on an existing account, the kvno will increase, causing existing clients to be unable to authenticate via Kerberos. The new keytab file will need to be migrated and applied to the NFS client. Verify the kvno in the keytab file with the kvno listed using the `kvno` command.

Keytabs are created on Windows domain controllers using the ktpass command and done only via the command line. This command is the same across all domain controllers from Windows 2003 on, but earlier Windows versions do not have ktpass by default. This and other utilities are included in the Windows 2003 support tools.

Table 21) Creating a keytab file.

1. Open the cmd prompt by going to Start -> Run and typing “cmd.”
2. Run the following command on the domain controller:

```
C:\> ktpass -princ primary/instance@REALM -mapuser DOMAIN\machine$ -crypto ALL +rndpass -
ptype KRB5_NT_PRINCIPAL +Answer -out [file:\location]
```

Example:

```
C:\>ktpass -princ root/nfsclient.domain.netapp.com@DOMAIN.NETAPP.COM -mapuser DOMAIN\suse12$
-crypto ALL +rndpass -ptype KRB5_NT_PRINCIPAL

 +Answer -out suse12-all.keytab

Targeting domain controller: win2k8DC.domain.netapp.com

Using legacy password setting method

Successfully mapped root/nfsclient.domain.netapp.com to SUSE12$.

WARNING: Account SUSE12$ is not a user account (uacflags=0x1021).

WARNING: Resetting SUSE12$'s password may cause authentication problems if SUSE12$ is being
used as a server.


Reset SUSE12$'s password [y/n]?  auto:

YES

WARNING: pType and account type do not match. This might cause problems.

Key created.

Key created.

Key created.

Key created.

Key created.

Output keytab to suse12-all.keytab:

Keytab version: 0x502

keysize 96 root/nfsclient.domain.netapp.com@DOMAIN.NETAPP.COM

 ptype 1 (KRB5_NT_PRINCIPAL) vno 4 etype 0x1 (DES-CBC-CRC) keylength 8 (0x1ae392970279eada)

keysize 96 root/nfsclient.domain.netapp.com@DOMAIN.NETAPP.COM

 ptype 1 (KRB5_NT_PRINCIPAL) vno 4 etype 0x3 (DES-CBC-MD5) keylength 8 (0x1ae392970279eada)

keysize 104 root/nfsclient.domain.netapp.com@DOMAIN.NETAPP.COM ptype 1 (KRB5_NT_PRINCIPAL)
vno 4 etype 0x17 (RC4-HMAC) keylength 16 (0xcb59b0528d99ec6c44e67ca7bee39e9f)

keysize 120 root/nfsclient.domain.netapp.com@DOMAIN.NETAPP.COM ptype 1 (KRB5_NT_PRINCIPAL)
vno 4 etype 0x12 (AES256-SHA1) keylength 32
(0x084e6030e9a0d3da3d1d5c5fa3ed8cd4c61c0c10ff0b02ed0c5999dc44498f1b)

keysize 104 root/nfsclient.domain.netapp.com@DOMAIN.NETAPP.COM ptype 1 (KRB5_NT_PRINCIPAL)
vno 4 etype 0x11 (AES128-SHA1) keylength 16 (0xdf2

1e49195aa835e57de9a382dfbc42e)
```

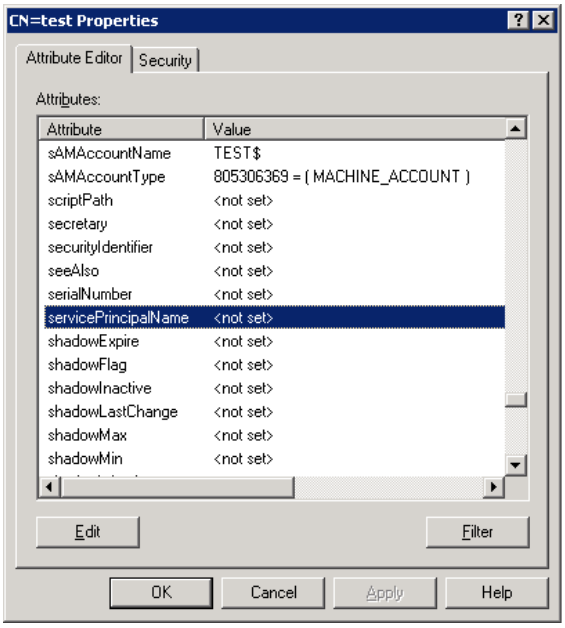
The above command creates a keytab file called “nfsclient.nfsclient.keytab” on the C:\ drive. It also assigns an SPN and a UPN to the account and allows all encryption types for the machine. Note that the command returns warnings, but these can be ignored. The reset password warnings can be avoided altogether by adding 65536 to the userAccountControl value to include DONT_EXPIRE_PASSWORD in the attribute. See “About the machine account attributes” for

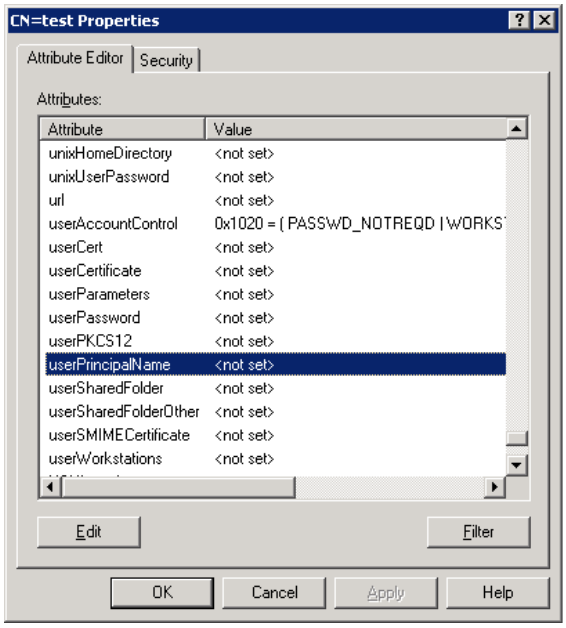
further details.

Sample `klist -kte` output from a Linux client after applying the keytab:

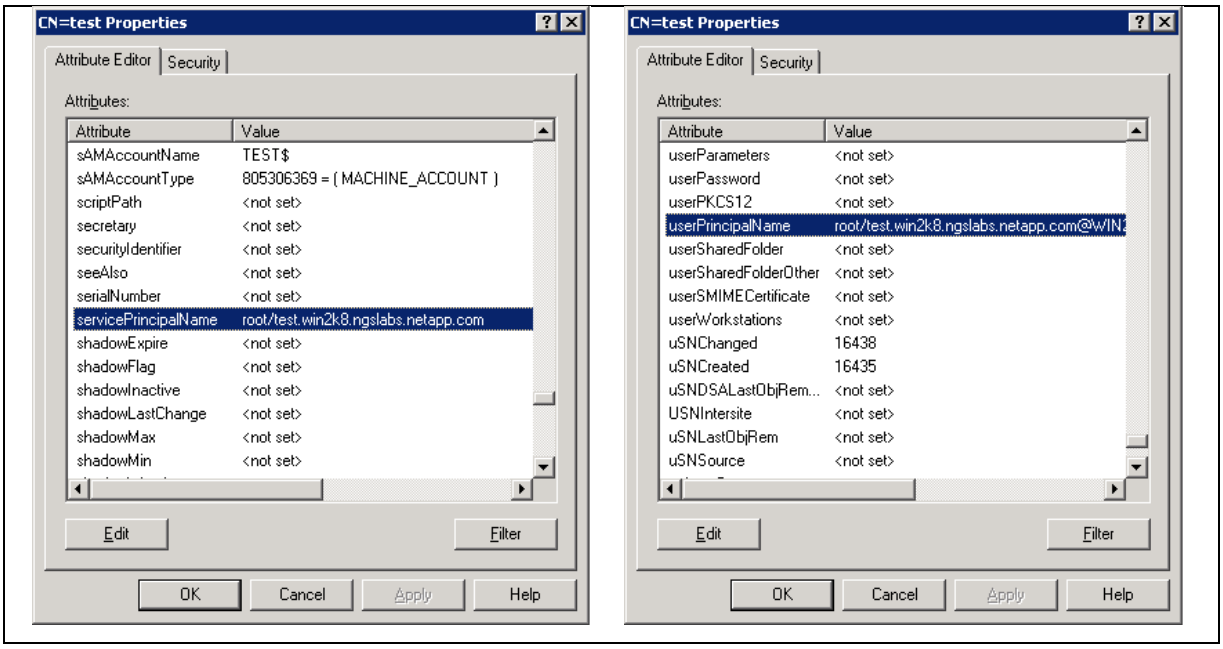
```
[root@centos64 ~]# klist -kte
Keytab name: FILE:/etc/krb5.keytab
KVNO Timestamp         Principal
---- ----------------- --------------------------------------------------------
3 05/21/13 10:23:24 root/nfsclient.domain.netapp.com@DOMAIN.NETAPP.COM (des-cbc-crc)
3 05/21/13 10:23:24 root/nfsclient.domain.netapp.com@DOMAIN.NETAPP.COM (des-cbc-md5)
3 05/21/13 10:23:24 root/nfsclient.domain.netapp.com@DOMAIN.NETAPP.COM (arcfour-hmac)
3 05/21/13 10:23:24 root/nfsclient.domain.netapp.com@DOMAIN.NETAPP.COM (aes256-cts-hmac-sha1-96)
3 05/21/13 10:23:24 root/nfsclient.domain.netapp.com@DOMAIN.NETAPP.COM (aes128-cts-hmac-sha1-96)
```

Machine account SPN and UPN before running ktpass:





Machine account SPN and UPN after running ktpass:



Moving Keytab Files to the NFS Client

Once the keytab file has been created, it needs to be moved to the NFS client. This can be done in a variety of ways. The easiest way is to download an FTP or SCP application to connect to the NFS client. There are plenty of free applications available for this, but covering this process is outside the scope of this document.

Verifying SPNs

After configuring the domain controller, verify that the machine account created by enabling Kerberos on the SVM data LIF(s) has the proper SPNs. If a CIFS server exists in the same SVM and domain, it's imperative that Active Directory is searched for duplicated SPNs. Duplicate SPNs can cause authentication failures with Kerberos.

Duplicate SPNs will log errors in the Windows event log and Kerberos attempts will fail. To find duplicate SPNs in Windows 2008 and later, the `setspn` utility can be used with the /Q flag to query for SPNs.

In the example below, note that there are two CNs (sles11 and centos6) that have the SPN root/nfsclient.nfsclient.domain.netapp.com assigned:

```
C:\>setspn /Q root/nfsclient.nfsclient.domain.netapp.com
Checking domain DC=domain,DC=netapp,DC=com
CN=sles11,CN=Computers,DC=domain,DC=netapp,DC=com
        root/nfsclient.nfsclient.domain.netapp.com
        root/sles11.domain.netapp.com
CN=centos6,CN=Computers,DC=domain,DC=netapp,DC=com
        root/nfsclient.nfsclient.domain.netapp.com

Existing SPN found!
```

Note: A common scenario in which duplicate SPNs may occur is if a CIFS server has been created and Kerberos has been enabled for the same SVM. Be sure to check the nfs/cluster@REALM SPN. However, this can occur on any machine account in the domain if a misconfiguration occurs.

If more than one CN is listed for the SPN that is queried, then one of the SPNs should be deleted using either the setspn tool or the Attributes Editor tab in the Active Directory Users and Computers GUI.

To delete a duplicate SPN:

```
C:\>setspn /D root/nfsclient.nfsclient.domain.netapp.com sles11
Unregistering ServicePrincipalNames for CN=sles11,CN=Computers,DC=domain,DC=netapp,DC=com
        root/nfsclient.nfsclient.domain.netapp.com
Updated object
```

For Windows 2003 servers, consult the following Microsoft KB to query for duplicate SPNs:

Finding Duplicate SPNs in Windows 2003 Servers

Common Kerberos and LDAP Errors

For a list of common Kerberos and LDAP errors as seen in packet traces, see the Microsoft TechNet article called "Kerberos and LDAP Error Messages."

For details on using packet traces for Kerberos troubleshooting, see "Kerberos errors in network captures." Also see Table 40 in this document for common Kerberos errors in network captures.

4.1.18 Using Domain Trusts

Trusted domains are domains that the local system trusts to authenticate users. In other words, if a user or application is authenticated by a trusted domain, this authentication is accepted by all domains that trust the authenticating domain.

For example, if Company A merges with Company B, those companies can set up a trust between their Active Directory domains so that all users can authenticate across the domains. That way, Company A's users can access files on Company B's NetApp storage systems via CIFS or Kerberized NFS. This is known as a two-way (or bidirectional) trust.

An alternative to this is a one-way trust. In this setup, Company A's domain can authenticate to Company B's domain, but Company B cannot authenticate to Company A's domain (or vice versa).

If both domains contain LDAP information, then care must be exercised so that UIDs and GIDs aren't duplicated across both domains.

For more information on Domain Trusts, see Microsoft's TechNet article on trusts.

NetApp clustered Data ONTAP storage systems support domain trusts, which means that LDAP and Kerberos will work with trusted domains, provided they are configured correctly. For configuration details and considerations, see the section regarding domain trusts in cluster configuration.

4.1.19 Domain Controller Redundancy and Replication

Active Directory, by default, replicates its databases to domain controllers every 15 minutes. That means that all objects in the domain are copied across multiple locations. This includes user and computer objects and their attributes, so it is possible to eliminate single points of failure in LDAP and Kerberos simply by having more than one domain controller in a domain.

Additionally, it is not a requirement to install Active Directory Services for UNIX or Identity Management on every domain controller. Only one domain controller needs the schema extended, because the new attributes will replicate across all domain controllers.

For example, the following SVM is connected to IP 10.61.179.152 for LDAP requests:

```
cluster::*> diag secd connections show -node node1 -vserver vs0 -type ldap-nis-namemap
[ Cache: LDAP (NIS & Name Mapping)/<no key> ]
  Queue> Waiting: 0, Max Waiting: 1, Wait Timeouts: 0, Avg Wait: 0.00ms
  Performance> Hits: 9, Misses: 4, Failures: 1, Avg Retrieval: 552.33ms

+ Rank: 01 - Server: 10.61.179.155 (10.61.179.152)
              Connected through the 10.61.92.34 interface, 10.0 mins ago
              Used 1 time(s), and has been available for 310 secs
```



```
RTT in ms: mean=1.00, min=1, max=1, med=1, dev=0.00 (11.9 mins of data)
```

And the IP is not reachable:

```
cluster::*> network ping -lif data -lif-owner vs0 -destination 10.61.179.152 -verbose true -show-
detail true -count 1
PING 10.61.179.152 (10.61.179.152) from 10.61.92.34: 56 data bytes

--- 10.61.179.152 ping statistics ---
1 packets transmitted, 0 packets received, 100.0% packet loss

C:\>ping 10.61.179.152

Pinging 10.61.179.152 with 32 bytes of data:
Request timed out.

Ping statistics for 10.61.179.152:
    Packets: Sent = 1, Received = 0, Lost = 1 (100% loss),
```

However, the SVM has two LDAP servers configured:

```
cluster::*> ldap client show -client-config LDAP -vserver vs0 -fields servers
  (vserver services ldap client show)
vserver client-config servers
------- ------------- --------------------------
vs0    LDAP          10.61.179.152,10.61.179.155
```

The other LDAP server is reachable from the SVM:

```
cluster::*> network ping -lif data -lif-owner vs0 -destination 10.61.179.155 -verbose true -show-
detail true -count 1
PING 10.61.179.155 (10.61.179.155) from 10.61.92.34: 56 data bytes
64 bytes from 10.61.179.155: icmp_seq=0 ttl=125 time=1.646 ms

--- 10.61.179.155 ping statistics ---
1 packets transmitted, 1 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 1.646/1.646/1.646/0.000 ms
```

After a set amount of time, the bad LDAP server will age out of cache:

```
cluster::*> diag secd connections show -node node1 -vserver vs0 -type ldap-ad
[ Cache: LDAP (Active Directory)/domain.domain.netapp.com ]
  Queue> Waiting: 0, Max Waiting: 1, Wait Timeouts: 0, Avg Wait: 0.00ms
  Performance> Hits: 0, Misses: 1, Failures: 0, Avg Retrieval: 10.00ms

  (No connections active or currently cached)
```

However, LDAP requests will still work due to the backup LDAP server:

```
cluster::*> diag secd authentication show-creds -node node1 -vserver vs0 -win-name ldapuser -
list-name true -list-id true

 UNIX UID: 1011 (ldapuser) <> Windows User: S-1-5-21-4188149759-3327341225-292728556-1011
(CIFS\ldapuser (Local User))

 GID: 513 (Domain Users)
 Supplementary GIDs: <None>

 Windows Membership:
 User is also a member of Everyone, Authenticated Users, and Network Users

 Privileges (0x0):
```

The new LDAP connection will show up in cache:

```
cluster::*> diag secd connections show -node node1 -vserver vs0 -type ldap-nis-namemap
[ Cache: LDAP (NIS & Name Mapping)/<no key> ]
  Queue> Waiting: 0, Max Waiting: 1, Wait Timeouts: 0, Avg Wait: 0.00ms
```



```
  Performance> Hits: 9, Misses: 4, Failures: 1, Avg Retrieval: 552.33ms

 + Rank: 01 - Server: 10.61.179.155 (10.61.179.155)
              Connected through the 10.61.92.34 interface, 2.0 mins ago
              Used 1 time(s), and has been available for 118 secs
              RTT in ms: mean=1.00, min=1, max=1, med=1, dev=0.00 (11.9 mins of data)
```

Once the other LDAP server is available again, the cluster will start using it since it is listed first in the configuration:

```
cluster::*> network ping -lif data -lif-owner vs0 -destination 10.61.179.152 -verbose true -show-
detail true -count 1
PING 10.61.179.152 (10.61.179.152) from 10.61.92.34: 56 data bytes
64 bytes from 10.61.179.152: icmp_seq=0 ttl=125 time=0.588 ms

--- 10.61.179.152 ping statistics ---
1 packets transmitted, 1 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 0.588/0.588/0.588/0.000 ms

cluster::*> diag secd authentication show-creds -node node1 -vserver vs0 -win-name ldapuser -
list-name true -list-id true

 UNIX UID: 1011 (ldapuser) <> Windows User: S-1-5-21-4188149759-3327341225-292728556-1011
(CIFS\ldapuser (Local User))

 GID: 513 (Domain Users)
 Supplementary GIDs: <None>

 Windows Membership:
 User is also a member of Everyone, Authenticated Users, and Network Users

 Privileges (0x0):


cluster::*> diag secd connections show -node node1 -vserver parisi -type ldap-nis-namemap
[ Cache: LDAP (NIS & Name Mapping)/<no key> ]
  Queue> Waiting: 0, Max Waiting: 1, Wait Timeouts: 0, Avg Wait: 0.00ms
  Performance> Hits: 9, Misses: 5, Failures: 1, Avg Retrieval: 510.46ms

 + Rank: 01 - Server: 10.61.179.152 (10.61.179.152)
              Connected through the 10.61.92.37 interface, 0.0 mins ago
              Used 1 time(s), and has been available for 2 secs
              RTT in ms: mean=2.00, min=0, max=4, med=3, dev=1.41 (18.2 mins of data)
```

For Kerberos, the same idea applies. For more information on SecD caches, see TR-4067.

Note: Because Active Directory replicates every 15 minutes, changes to machine accounts may not apply to all DCs until replication occurs. Keep this in mind when troubleshooting NFS Kerberos issues. If necessary, force replication in the domain.

NFS clients and LDAP applications will leverage this as well. For more information, see the section regarding setup of NFS clients to use LDAP.

4.1.20 Configuring the NFS Clients to Use Kerberos

Kerberos configuration on multiple client platforms is covered in the sections that follow. These steps can be transferred to 7-Mode implementations as well.

For condensed setup steps, see the "Quick Step Setup Guides" section in this document.

4.1.21 Solaris Kerberos Configuration

Solaris generally uses a utility called `kclient` to configure Kerberos. However, this utility has issues when configuring Kerberos with a Windows 2008 R2 DC due to the format the utility expects to use (username/admin). Attempts to use the utility may result in the following error:


```
kinit(v5): Client not found in Kerberos database while getting initial credentials
```

Instead, follow the same steps for other clients to configure Kerberos.

- Configure the /etc/krb5/krb5.conf file.
- Verify that DNS is working properly.
- Verify that the principals have been created.
- Verify that the time is within 5 minutes of the KDC.
- Verify that the keytab file has been created and exported to the client.

For MIT KDCs, `kclient` is the preferred method. When using `kclient`, the utility will create the principals for the client on the KDC.

If "Do you plan on doing Kerberized nfs" is answered with "yes," then the client will attempt to create the SPNs for the client on the KDC using the principal specified in the administrative principal section of the script. This will create the following SPNs:

```
nfs/hostname@REALM
root/hostname@REALM
host/hostname@REALM
```

To see the principals on an MIT KDC, use listprincs from kadmin:

```
[root@mit-kdc ~]# kadmin
Authenticating as principal root/admin@DOMAIN.MIT.NETAPP.COM with password.
Password for root/admin@DOMAIN.MIT.NETAPP.COM:
kadmin:  listprincs *solaris*
host/solaris-mit.domain.mit.netapp.com@DOMAIN.MIT.NETAPP.COM
nfs/solaris-mit.domain.mit.netapp.com@DOMAIN.MIT.NETAPP.COM
root/solaris-mit.domain.mit.netapp.com@DOMAIN.MIT.NETAPP.COM
```

For more information on `kclient`, see the Solaris documentation on configuring Kerberos clients.

For information on configuring LDAP with Solaris, see "Configuring Solaris to use LDAP."

4.1.22 Configuring Linux Clients

Many clients share commonalities in Kerberos configurations. The following configurations cover the following NFS clients:

- Red Hat Enterprise Linux/CentOS 6.3 and 6.4
- Fedora 18
- SLES 11
- SUSE 12
- Ubuntu 12.1

Host Name

For Kerberos to work properly, it's important that the client's host name be set via the network configuration. Host names are set differently depending on the client.

The following table lists where the host name is set for various Linux clients.

Table 22) Setting the host name.

OS	File to Modify
RHEL/CentOS/Fedora	`/etc/sysconfig/network`

SLES/SUSE	`/etc/HOSTNAME`
Ubuntu	`/etc/hostname`

For more information on setting the client's host name, see the vendor documentation.

Kerberos/LDAP Packages

Many clients have the necessary Kerberos components installed by default. However, if the Kerberos components are missing from the client, the necessary packages need to be installed.

Each client uses a different method to install packages. Installing packages is outside the scope of this document. Acquire any assistance needed with package installation from the client vendor.

Date and Time

Clients may manage their date and times differently, but the main consideration with Kerberos is that the date and time on the NFS clients must be within a five-minute window of the KDC and the storage system. If the clock skew is outside this five-minute window, Kerberos requests will fail.

DNS

Clients will get their DNS information either from a DHCP server or the static network configuration on the client. All clients covered in this document leverage the `/etc/resolv.conf` file for static DNS configuration. DHCP and automatic network configuration for clients is outside the scope of this document.

To check DNS resolution for the client, leverage the `nslookup` command for the host name and the IP to check for forward and reverse entries in DNS. Also check that the NFS client can look up the cluster's data LIF by name and IP.

Example:

```
[root@centos6 ~]# nslookup centos6
Server:         10.63.98.101
Address:        10.63.98.101#53

Name:   nfsclient.nfsclient.domain.netapp.com
Address: 10.61.179.164

[root@centos6 ~]# nslookup 10.61.179.164
Server:         10.63.98.101
Address:        10.63.98.101#53

164.179.61.10.in-addr.arpa      name = nfsclient.nfsclient.domain.netapp.com.

[root@centos6 ~]# nslookup krb5server
Server:         10.63.98.101
Address:        10.63.98.101#53

Name:   krb5server.domain.netapp.com
Address: 10.61.92.34

[root@centos6 ~]# nslookup 10.61.92.34
Server:         10.63.98.101
Address:        10.63.98.101#53

34.92.61.10.in-addr.arpa        name = krb5server.domain.netapp.com.
```

NFSv4 Domain

NFSv4 domains are set in the same file for every client listed in this document except Solaris.

Those clients all leverage the `/etc/idmapd.conf file`. To set the NFSv4 domain, modify the `[General]` section.

Example:

```
[General]
#Verbosity = 0
# The following should be set to the local NFSv4 domain name
# The default is the host's DNS domain name.
#Domain = local.domain.edu
Domain = domain.netapp.com
```

Note: There are other sections of this file that are not covered in the scope of this document. For more information on the /etc/idmapd.conf file, see http://linux.die.net/man/5/idmapd.conf.

The Solaris NFSv4 domain configuration is covered in the Solaris documentation.

NFSv4 implementation for all other clients is covered in detail in the NFSv4 section of this document.

Note: NFSv4 is only needed for Kerberos if a higher level of security is desired.

Allowing Secure NFS

Every NFS client disables secure NFS (Kerberos) by default. If secure NFS is disabled, Kerberos services will not start properly. Enabling this is different on each type of client. The table below describes which file for each OS needs to be modified and which value needs to be changed.

Table 23) Allowing secure NFS.

OS	File to Modify	Value to Change
RHEL/CentOS/Fedora	`/etc/sysconfig/nfs`	SECURE_NFS=”yes”
SLES/SUSE	`/etc/sysconfig/nfs`	NFS_SECURITY_GSS="yes"
Ubuntu	`/etc/default/nfs-common`	NEED_GSSD="yes"
Solaris	`/etc/nfssec.conf`	Uncomment the desired krb values

Note: If the configuration files are missing, it’s likely that the correct packages are not installed. Install the correct packages and try again.

krb5.conf

The `krb5.conf` file is where the client will get its Kerberos configuration information. This file must exist on clients wanting to use Kerberos.

- In Linux, the file is located at `/etc/krb5.conf`
- In Solaris, the file is located at `/etc/krb5/krb5.conf`

The file consists of several sections that are used in ticket services. The following sample `krb5.conf` file shows how to configure NFS clients for Kerberos.

```
[libdefaults]
 default_realm = DOMAIN.NETAPP.COM
 dns_lookup_realm = true
 dns_lookup_kdc = true
 allow_weak_crypto = true

[realms]
       DOMAIN.NETAPP.COM = {
  kdc = windows-KDC.domain.netapp.com:88
```



```
            default_domain = domain.netapp.com
        }

[logging]
        kdc = FILE:/var/log/krb5kdc.log
        admin_server = FILE:/var/log/kadmin.log
        default = FILE:/var/log/krb5lib.log

[domain_realm]
        .netapp.com = DOMAIN.NETAPP.COM
        .domain.netapp.com = DOMAIN.NETAPP.COM
```

Note: To control which encryption types the client uses first, there are settings under [libdefaults] that can be modified to control the order of enctypes requested.

To modify the order from what the client defaults to use, add the following lines under [libdefaults] so that the config looks as follows:

```
[libdefaults]
 default_realm = DOMAIN.NETAPP.COM
 default_tkt_enctypes = des-cbc-md5 des-cbc-crc aes256-cts des3-cbc-sha1 arcfour-hmac
 default_tgs_enctypes = des-cbc-md5 des-cbc-crc aes256-cts des3-cbc-sha1 arcfour-hmac
 dns_lookup_realm = true
 dns_lookup_kdc = true
 allow_weak_crypto = true
```

NetApp does not recommend this method for controlling encryption types due to its lack of scalability.

Note: Clustered Data ONTAP currently supports only DES and 3DES encryption types, which is why the *[libdefaults] allow_weak_crytpo = true* stanza is required.

Note: If there are a large number of clients or if you don't want to change the NFS client setting, it is possible to control this behavior from the KDC. See the section called "Configuring the domain controller" for details.

Once the `krb5.conf` file is configured and the DNS/time is confirmed correct, a `kinit` should work:

```
[root@centos6 /]# kinit administrator
Password for administrator@DOMAIN.NETAPP.COM:
```

Check the ticket with klist:

```
[root@centos6 /]# klist
Ticket cache: FILE:/tmp/krb5cc_0
Default principal: administrator@DOMAIN.NETAPP.COM

Valid starting     Expires            Service principal
05/03/13 15:18:00  05/04/13 01:19:10  krbtgt/ domain.netapp.com@DOMAIN.NETAPP.COM
        renew until 05/04/13 15:18:00
```

In RHEL/CentOS 6.4, DES_MD5 support was removed by default. To reenable it, add the following line to the /etc/environment file and reboot the client:

```
[root@centos6 /]#  /etc/environment
NSS_HASH_ALG_SUPPORT=+MD5
```

For more information, see Bugzilla report 895513.

krb5.keytab

This file is an encrypted local copy of the host's key. While the file is encrypted, it is still a point of entry and a potential security hole. The file should be readable only by root and should only exist on the local server's disk. The file is created on the KDC and then moved to the NFS client. Once on the client, the application called `ktutil` is used to read and write the `krb5.keytab` file. This file does not exist by default and must be created.

- In Linux, the file should be created at `/etc/krb5.keytab`.
- In Solaris, the file should be created at `/etc/krb5/krb5.keytab`.

Below is an example of a krb5.keytab file being created using `ktutil`:

```
[root@centos6 ~]# ktutil
ktutil:  rkt /nfsclient.nfsclient.keytab
ktutil:  list
slot KVNO Principal
---- ---- ---------------------------------------------------------------------
   1    3 root/nfsclient.nfsclient.domain.netapp.com@DOMAIN.NETAPP.COM
ktutil:  wkt /etc/krb5.keytab
ktutil:  q
```

The KVNO in the keytab file needs to match the KVNO returned from the KDC. To verify the KVNO, do the following after configuring `krb5.conf`:

```
[root@centos6 ~]# klist -kte
Keytab name: FILE:/etc/krb5.keytab
KVNO Timestamp         Principal
---- ----------------- --------------------------------------------------------
4 05/16/13 11:57:56 root/nfsclient.nfsclient.domain.netapp.com@DOMAIN.NETAPP.COM (des-cbc-crc)
)← KVNO is 4
4 05/16/13 11:57:56 root/nfsclient.nfsclient.domain.netapp.com@DOMAIN.NETAPP.COM (des-cbc-md5)
4 05/16/13 11:57:56 root/nfsclient.nfsclient.domain.netapp.com@DOMAIN.NETAPP.COM (arcfour-hmac)
4 05/16/13 11:57:56 root/nfsclient.nfsclient.domain.netapp.com@DOMAIN.NETAPP.COM (aes256-cts-
hmac-sha1-96)
4 05/16/13 11:57:56 root/nfsclient.nfsclient.domain.netapp.com@DOMAIN.NETAPP.COM (aes128-cts-
hmac-sha1-96)
[root@centos6 /]# kinit administrator
Password for administrator@DOMAIN.NETAPP.COM:
[root@centos6 /]# kvno root/nfsclient.nfsclient.domain.netapp.com@DOMAIN.NETAPP.COM
root/nfsclient.domain.netapp.com@DOMAIN.NETAPP.COM: kvno = 3 ←- KVNO doesn't match
```

Note: If the KVNO does not match between the DC and the keytab file, the `krb5.keytab` file must be recreated to update the KVNO.

Managing Kerberos Services

Once the `krb5.keytab` file and `krb5.conf` file are configured, the client-side Kerberos configuration is done. All that is left is restarting the Kerberos client service on the NFS client to apply the configuration.

Each client has a different service to restart with a different command to use. The following table covers which service is restarted on which client.

Table 24) Managing Kerberos services.

OS	Commands
RHEL/CentOS/Fedora	`service rpcgssd [start|stop|restart|status]`
SLES/SUSE*	`service nfs [start|stop|restart|status]`
Ubuntu	`service gssd [start|stop|restart|status]`
Solaris	`svcadm [enable|disable|restart|refresh] gss` `svcs –l gss (to list)`

*Enable the following to start at each boot (SUSE only):


```
[client] # systemctl enable rpcbind.service
[client] # systemctl enable nfs.service
```

*In addition, verify that the services are running (SUSE only):

```
[client] # service rpcbind start
[client] # service nfs start
```

For condensed Kerberos setup steps, see the "Quick Step Setup Guides" section in this document.

4.2 Setting Up LDAP

This section covers setting up LDAP for use by NFS clients. The server being used is a Windows 2008 R2 server with Identity Management installed. SSSD for the Linux clients is used for LDAP (except for Solaris). The LDAP requests will leverage Kerberos for security, so the Kerberos setup must be completed prior to attempting to set up LDAP. By the end of the section, users should be able to get identity information from the LDAP server. Currently, clustered Data ONTAP does not support the following for LDAP:

- RFC-2307bis

Note: For Quick Step Setup steps can be found at the end of this document.

4.2.1 Overview of LDAP

Lightweight Directory Access Protocol (LDAP) is a standard directory access protocol that was developed by the international committee Internet Engineering Task Force (IETF). LDAP is intended to provide a general-purpose, network-based directory service that can be used across heterogeneous platforms to locate network objects. LDAPv3 is the standard currently implemented version.

LDAP models define how to communicate with the LDAP directory store, how to find an object within the directory, how to describe the objects within the store, and the security used to access the directory. LDAP allows customization and extension of the objects described within the store. Therefore an LDAP store can be used to store many types of diverse information. Many of the initial LDAP deployments focused on using LDAP as a directory store for applications such as e-mail and web applications and to store employee information. During the last several years, LDAP has been gaining acceptance as a directory store for information used in network-based authentication and authorization. Many companies are replacing NIS with LDAP as a network directory store.

Microsoft implemented LDAPv3 as a directory store starting in Windows 2000/2003 Active Directory. The Microsoft LDAP implementation is standards based, resulting in the ability to use Microsoft Active Directory LDAP for the storage of UNIX user and group information. This provides a method to unify the directory service and directory store of networks based on both Windows and UNIX. However, native Active Directory LDAP does not contain the definitions of attributes needed to hold information that is necessary for UNIX authentication and authorization; therefore, the Microsoft Active Directory schema needs to be extended with the necessary objects to hold this information.

Clients based on both Windows and UNIX can access data in clustered Data ONTAP using CIFS or NFS; providing the ability to use standard network services for name resolution and for identity storage is crucial. Clustered Data ONTAP also supports integration into an Active Directory environment for Windows user authentication and authorization. The ability to use Active Directory LDAP as a directory store for UNIX user and group information has been provided as well.

What Does LDAP Store?

Active Directory LDAP can store the following information used in multiprotocol access:

- Username
- UID or GID

- Homedirs
- Login shell
- Netgroups, DNS names, and IP addresses
- Group membership

What LDAP Is Not

- Active Directory Identity Management (LDAP) is not Server for NIS.
- LDAP is not Active Directory, but Active Directory does leverage LDAP.
- AD LDAP cannot be used as a Pluggable Authentication Module (PAM) for cluster management role-based access control (RBAC).
  - To use AD for RBAC, leverage domain tunneling, which is covered in the product documentation.

System Security Services Daemon (SSSD) Overview

SSSD is a system daemon developed by Red Hat/Fedora as a replacement for PADL, Samba WinBind, and other AD-based PAM and nss modules. SSSD provides access to different identity and authentication providers. A new PAM module called pam_sss was created to leverage the new LDAP interface. SSSD includes an AD provider type, allowing easy integration with Windows Active Directory 2003, 2008, and 2012. SSSD leverages TLS encryption as well as LDAP via GSSAPI, which allows more secure LDAP binding and lookups over the wire. The steps in this document cover setting up SSSD to use GSSAPI (Kerberos) for authenticated LDAP binds. SSSD will use the strongest Kerberos encryption type supported by the client and Active Directory Domain controller.

Pluggable Authentication Module (PAM) Overview

PAM is a mechanism used to integrate low-level authentication schemes with more complex environments such as LDAP, SSSD, Kerberos, and so on. PAM authentication allows a Linux client to leverage higher encryption setups and use them, as opposed to using classic UNIX-style authentication. By way of PAM, Single Sign-On (SSO) can be implemented, allowing centralized management of users and groups and reducing the amount of overhead for managing individual Linux clients. With PAM, a user can log in to a system using his or her Active Directory user name and password, authenticate via Kerberos, and access Kerberized NFS mounts. SSSD will not leverage PAM, but other functions such as SSH and su will use PAM modules.

Note: Exercise caution when modifying PAM on a Linux client. Misconfiguration of PAM can lock users out from login. Consult the client vendor for configuring PAM. PAM configuration is outside the scope of this document.

4.2.2 Active Directory LDAP via SSSD Benefits

LDAP provides a centralized user ID and group ID database. When used with Active Directory, this database can be replicated to multiple sites and provides redundancy in case one LDAP server fails by way of native Active Directory replication mechanisms. Active Directory also provides ease of use over some of its Linux counterparts by way of configuration wizards and GUI access to set UIDs and GIDs. In addition it provides the flexibility to script via batch file or PowerShell to automate tasks. By default, Active Directory does not include UNIX-type schema attributes. These are included in schema extensions when installing Microsoft Services for UNIX (Windows 2003), Microsoft Identity Management (Windows 2003 R2 and later), or third-party identity management tools like Centrify or VAS. For more information on Windows Active Directory LDAP, see TR-3458.

Having a centralized identity management server also makes life with NFSv4 infinitely simpler. That is because all names will map to the correct IDs, preventing access attempts from being squashed to nfsnobody provided the names map into the NFSv4 ID domain.

Leveraging SSSD on Linux clients with Active Directory provides ease of use, security, and stability that other LDAP tools do not provide.

Details on SSSD support.

4.2.3 How SSSD Interacts with Active Directory

The following section details how SSSD interacts with Active Directory to query it for users and groups via the GSSAPI security method.

1. DNS queries for the LDAP SRV record are made via the first DNS server in the /etc/resolv.conf file; SRV record returns a list of valid servers with the _ldap._tcp.domain.com record.
2. A DNS query for the A record of the valid LDAP servers is made; the first successful query is the server that will be used by SSSD.
3. A TCP connection to the LDAP server is established and a searchRequest is made to the baseObject. This starts the authentication process.
4. Since GSSAPI was specified in the `/etc/sssd/sssd.conf` file as the SASL mechanism, a Kerberos ticket needs to be granted.
5. A DNS query for the Kerberos server is made (A and AAAA records); if the krb5_server is not included in the configuration file, then the SRV record for Kerberos is used.
6. A valid IP is returned for the Kerberos server and a TCP session is established.
7. An AS-REQ with the strongest encryption type supported by the client is made using the SPN specified by the ldap_sasl_authid option in the configuration file.
8. If successful, a Kerberos TGT is granted to the client using the strongest encryption type supported.
9. After the TGT is granted, a DNS query takes place (forward and reverse lookups) for the KDC being used for the ticket.
10. If the server is available, the client makes a TGS-REQ call using the strongest encryption type supported for the LDAP SPN in the domain associated with the server DNS returned.
11. If successful, the TGS is granted to the client using the strongest available encryption type.
12. Using the LDAP Kerberos TGS, the client will attempt a SASL bind to the LDAP server.
13. Other DNS queries (A and AAAA records) for ForestDNSZones, DomainDNSZones, and reverse lookup zones/pointer records take place.
14. If successful, the bind will report as successful and deliver the LDAP query to the client via SASL GSS-API Privacy payload packets. These packets are encrypted using the strongest encryption supported by the client and domain controller.
15. Once the request is complete, the LDAP server will send a reset (RST) packet to the client to close the TCP connection.

4.2.4 Configuring the Domain Controller as an LDAP Server

As previously mentioned, Microsoft Active Directory does not act as an LDAP server natively. To use an Active Directory domain controller as an LDAP server, a schema extender must be installed to include the UNIX-style schema attributes necessary for mapping user names to UIDs. The schema extender will depend on the version of Windows being used. There are also third-party LDAP schema extenders. Table 3 shows the Microsoft offerings for LDAP schema extension depending on the Windows version. For third-party schema extension, please contact the vendor of the product.

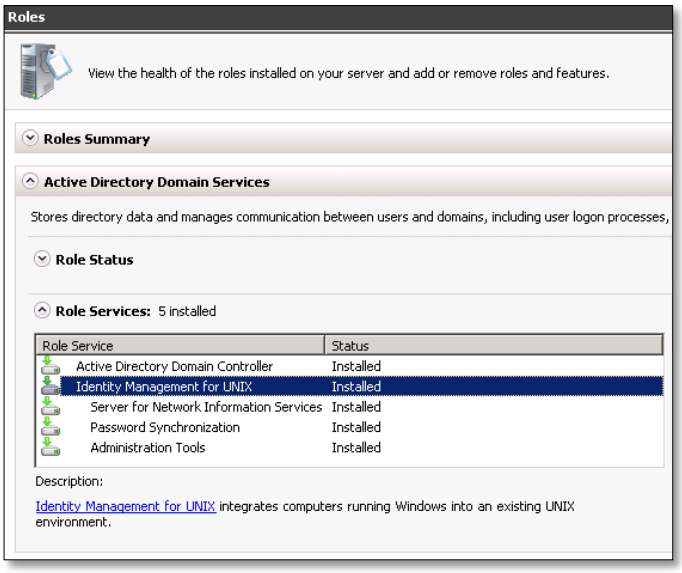
Note: When running commands on servers operating in Windows 2008 and later, User Account Control may prevent running certain commands for users that are not logged in as the administrator user. So that commands in this document run properly, either log in as the administrator user or use the “Run As Administrator” option.

Table 25) Active Directory schema extensions per Windows version.

Windows Version	Microsoft Schema Extender
Windows 2003	Windows Services for UNIX (SFU)
Windows 2003 R2	Windows Identity Management (IDMU)
Windows 2008 and later	Windows Identity Management (IDMU)

Extending the Schema in Windows 2008 R2

In Windows 2008 R2, Identity Management is included under the Role Service section of Server Manager.



To install a role in Windows 2008 R2, simply click on “Add Role Services” and follow the prompts. Once this finishes, the Active Directory schema will be extended and new attributes will be available for modification. In addition, new tabs will be available on user and group properties, such as the UNIX Attributes tab.

What Does “Extending the Schema” Mean?

By default, Active Directory has an LDAP schema with attributes used in directory lookups for AD tasks, such as Kerberos authentication, SID translation, and so on. However, AD does not have UNIX attributes in the schema by default, such as UID, UIDNumber, GID, GIDNumber, unixHomeDirectory, and so on. These attributes are added by installing AD-IDMU/AD-SFU or a third-party utility. Attributes can also be added manually, but this is not a straightforward endeavor.

When AD-IDMU or AD-SFU is installed, the default schema is extended with the new UNIX attributes to allow UNIX-style LDAP lookups for multiprotocol access.

For more information about schema extensions in Active Directory, see the TechNet Article on Extending the Schema.

Assigning UNIX Attributes

The UNIX Attributes tab allows an administrator to assign a UID and a default GID to user objects for use by LDAP. If no UID/GID is specified, then the object will not be able to be used for multiprotocol access.

The GID will be the default group for the user. To assign a GID to a user, the group must first be configured to have a GID. The GID assigned on the user object will be the user's default group. This group can be different than the Windows groups assigned in the "MemberOf" tab. Groups should be Global Security groups.

If secondary groups are desired, then the group object in the directory must be modified to include users as members under the "UNIX Attributes" tab for the group. These members can be different from the users in the "Members" tab. However, NetApp does not recommend assigning a user to a group that it already specified as its default GID because it creates a second entry for that group.

Best Practice

When choosing a UID or a GID, use the SID of the object for organizational purposes.

To get a SID for a user or group from a clustered Data ONTAP system, an existing CIFS server must be in place. If a CIFS server exists, use the following commands to get Windows SIDs:

```
cluster::> set diag

cluster::*> diag secd authentication translate -node [node] -vserver
[vserver] -win-name ldapuser

S-1-5-21-4188149759-3327341225-292728556-1011
```

Take the last set of digits and use those as the UID or GID. In the above example, the user "ldapuser" would be assigned a UID of 1011.

If a CIFS server does not exist, use the following to get a user SID:

Determining a SID for a User Account

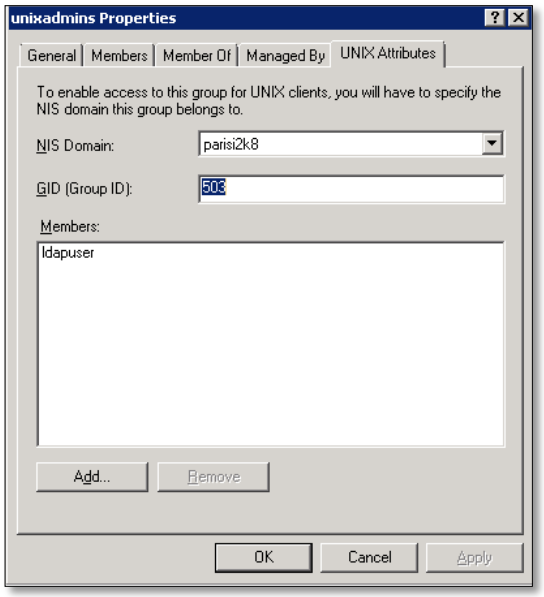
Note: In clustered Data ONTAP, there is a limit of 16 auxiliary GIDs per user for RPCGSS_SEC in 8.2 and earlier. In 8.2.1, that limit is increased to 32 auxiliary GIDs per user for RPCGSS_SEC. The limit for auxiliary GIDs for AUTH_SYS is 16, which is a limitation of the NFS standard. This limit was artificially extended to 256 in 7-Mode via the `nfs.max_num_aux_groups` option, introduced in Data ONTAP 7-Mode 7.3.2 and later.

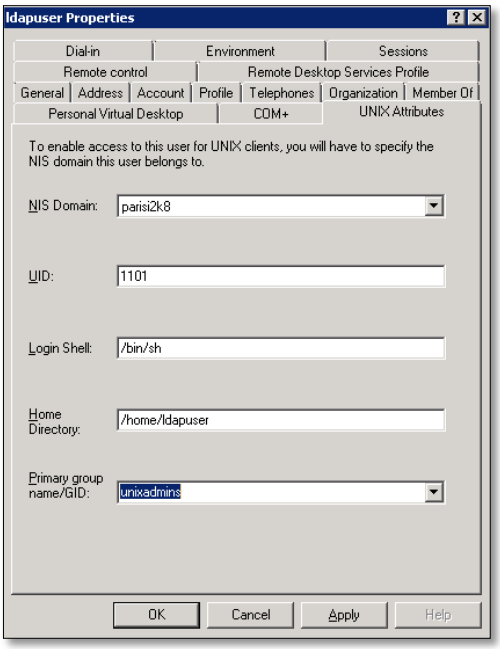
The following screenshots show different actions with UNIX attributes in Active Directory LDAP.

Table 26) Setting UID/GID in Active Directory LDAP (GUI).

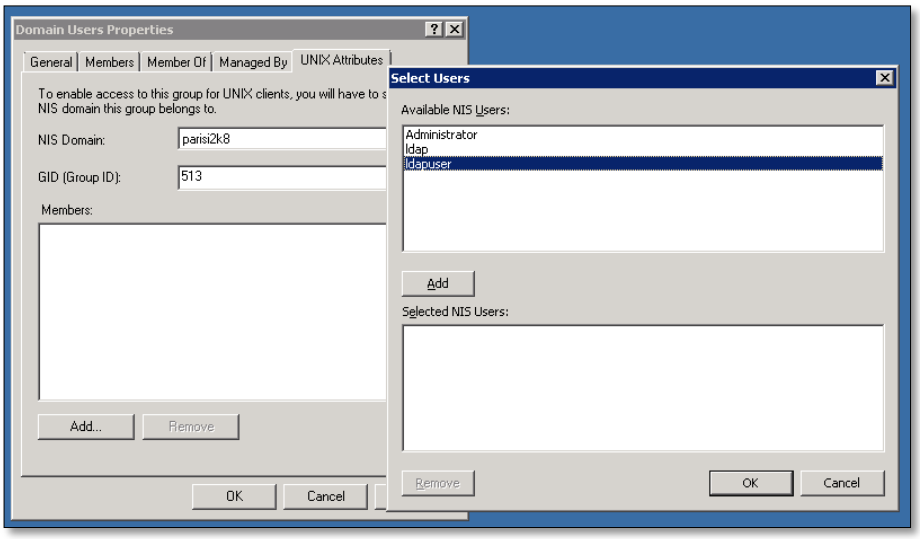
a) Setting a GID on an existing AD group.

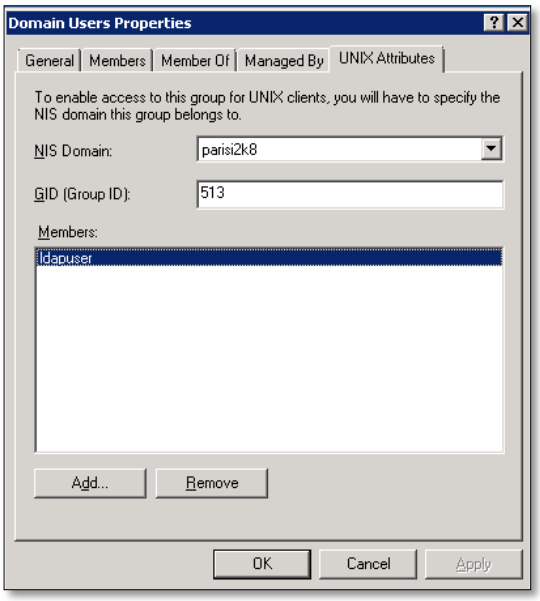
b) Setting a UID and GID on an existing AD user.





c) Adding a user to a secondary Active Directory group as a member.





Additionally, the attributes can be set using the **ldifde** utility. Below is an example of setting the UID and GID for an object using ldifde.

Table 27) Setting UID/GID in Active Directory LDAP (ldifde).

1. Log in to the domain controller and open a text editor such as WordPad.
2. Create a file named `account_name_unix.ldf` with the following entries (modified with the account info):

```
dn: CN=ldapuser,CN=Users,DC=domain,DC=netapp,DC=com
changetype: modify
replace: uidNumber
uidNumber: 1101
-
```



```
dn: CN=centos6,CN=Users,DC=win2k8,DC=netapp,DC=com
changetype: modify
replace: gidNumber
gidNumber: 513
-
```

NOTE: The above includes a dash and return carriage after each entry. These entries are required for the modification to work properly.

3. Save the file and open the cmd prompt by going to Start -> Run and typing “cmd.”

4. Run the following command to import the entry, replacing the file below with the name and location of the file that was created:

```
ldifde -i -f C:\account_name_unix.ldf
```

5. Verify that the account has changed the attributes with the following command, replacing the [entries] with the LDAP server’s entries:

```
C:\>ldifde -d "[DC=domain,DC=com]" -f unix_output.txt
-r "(&(objectCategory=person)(objectClass=user)(name=[username]))”
-l “uidNumber,gidNumber”
```

Example:

```
C:\>ldifde -d “DC=win2k8,DC=netapp,DC=com” -f DES_output.txt
-r “(&(objectCategory=person)(objectClass=user)(name=ldapuser))”
-l “uidNumber,gidNumber”
```

After configuring users and groups with UIDs and GIDs, there are several tools one can use to view the schema attributes for objects.

- Various built-in Microsoft tools
- LDAP Explorer
- LDAP Browser

The table below shows the difference in schema attributes on a user before and after IDMU is installed on a Windows 2008 R2 domain controller.

Note the UNIX attributes added after the schema was extended. Those attributes are used to determine UID/GID mappings in LDAP queries. The following uses `ldifde` to view the schema attributes. This tool can also be used to import and modify attributes. For examples, see the following:

- Import/Export from AD with LDIFDE
- Using LDIFDE to Import and Export
- TechNet Article on LDIFDE

Command used:

```
C:\>ldifde -d "CN=ldapuser,CN=Users,DC=netapp,DC=com" -f ldapuser.txt -r "(objectClass=user)"
```

Table 28) Difference in schema attributes before/after extending the schema using `ldifde`.

Before Schema Extend	After Schema Extend

dn: CN=ldapuser,CN=Users,DC=netapp,DC=com objectClass: top objectClass: person objectClass: organizationalPerson objectClass: user cn: ldapuser givenName: ldapuser distinguishedName: CN=ldapuser,CN=Users,DC=netapp,DC=com displayName: ldapuser name: ldapuser objectGUID:: pi7wE/AlKE+3rIspB9lAvQ== userAccountControl: 512 primaryGroupID: 513 objectSid:: AQUAAAAAAAUVAAAAWhBXSYAJJSTfIIw/TwQAAA== sAMAccountName: ldapuser sAMAccountType: 805306368 userPrincipalName: ldapuser@netapp.com objectCategory: CN=Person,CN=Schema,CN=Configuration, DC=netapp,DC=com	dn: CN=ldapuser,CN=Users,DC=netapp,DC=com objectClass: top objectClass: person objectClass: organizationalPerson objectClass: user cn: ldapuser givenName: ldapuser distinguishedName: CN=ldapuser,CN=Users,DC=netapp,DC=com displayName: ldapuser memberOf: CN=unixadmins,CN=Users,DC=netapp,DC=com name: ldapuser objectGUID:: kD6gtuDo9UKeZ5O/mqJLBg== userAccountControl: 66048 primaryGroupID: 513 objectSid:: AQUAAAAAAAUVAAAATaofhHZHSrXcjKDbYwQAAA== sAMAccountName: ldapuser sAMAccountType: 805306368 userPrincipalName: ldapuser@netapp.com objectCategory: CN=Person,CN=Schema,CN=Configuration, DC=netapp,DC=com **unixUserPassword: ABCD!efgh12345$67890** **uid: ldapuser** **msSFU30Name: ldapuser** **msSFU30NisDomain: netapp** **msSFU30PosixMemberOf:** **CN=unixadmins,CN=Users,DC=netapp,DC=com** **msSFU30PosixMemberOf:** **CN=Domain Users,CN=Users,DC=netapp,DC=com** **uidNumber: 1101** **gidNumber: 503** **unixHomeDirectory: /home/ldapuser** **loginShell: /bin/sh**

Setting Name Mapping Rules in LDAP

LDAP can map Windows user names to UNIX user names on a 1:1 basis, but it can also be used to map Windows user names that differ from their UNIX counterparts without the need to create name mapping rules on the storage system.

The LDAP schema defined in clustered Data ONTAP contains an attribute called “ONTAP Name Mapping windowsAccount Attribute,” which defines which LDAP schema attribute to use when mapping Windows names to UNIX names. The default value of this attribute is windowsAccount, which does not exist by default in Windows Active Directory LDAP schemas.

When mapping a Windows user name to a different UNIX user name in LDAP, follow these steps:

Table 29) Mapping users with LDAP

1. Copy the default schema to a new schema name, because default schemas are read-only. This is an advanced-level command.

Example:

```
::> set advanced
::*> ldap client schema copy -schema AD-IDMU -new-schema-name NEW -vserver [SVM]
```

2. Change “ONTAP Name Mapping windowsAccount Attribute” to uid. Modify is also an advanced- level command.

Example:


```
::> set advanced
::*> ldap client schema modify -schema NEW -windows-account-attribute uid -vserver [SVM]
```

3. Test the name mapping.

Example:

```
::> set diag
::*> diag secd name-mapping show -node [SVM] -vserver nfs -direction win-unix -name ldapuser
ldapuser maps to ldapuser2
```

DNS Considerations

SSSD can leverage Kerberos authentication for secure LDAP lookups. Because of this, DNS must be configured properly to include information about the LDAP URI being used in SSSD configuration. SSSD does not support the use of round-robin DNS entries for failover. Each entry needs to be unique and located in DNS for failover to work properly.

From the SSSD documentation:

The failover mechanism distinguishes between machines and services. The back end first tries to resolve the hostname of a given machine; if this resolution attempt fails, the machine is considered offline. No further attempts are made to connect to this machine for any other service. If the resolution attempt succeeds, the back end tries to connect to a service on this machine. If the service connection attempt fails, then only this particular service is considered offline and the back end automatically switches over to the next service. The machine is still considered online and might still be tried for another service.

The failover mechanism does not handle DNS A records with multiple IP addresses; instead it only uses the first address. DNS round-robin cannot be used for failover. Further, providing multiple A records does not provide failover. Only the first A record is used, and if a lookup attempt on the first record fails then the system attempts no further lookups. To find multiple servers with a single request, and thus implementing failover, SSSD relies on SRV resource records.

An additional limitation to SSSD is that failover also depends on the order of entries in `/etc/resolv.conf`. If the first DNS server in the file is inaccessible, SSSD will black hole the attempt until the DNS server is available or until the /etc/resolv.conf file is modified. For more information on this, see Red Hat bug 966757.

Best Practice

If the domain has multiple domain controllers, leave the ldap_uri and krb5_server options out of the configuration file. This will enable use of the built-in SRV records for Kerberos and LDAP, which will allow failover capability in the event a domain controller goes down. If you use a single domain controller, leave the options out for scalability in the event additional domain controllers are added at a later date.

4.2.5 Configuring the Client to Use LDAP

The following client configurations will leverage SSSD for LDAP and authentication. Verify that PAM is configured properly to avoid being locked out of the system, because SSSD modules get added to PAM configurations.

Note: This section assumes that Kerberos has been configured and a ticket can be issued to the NFS client via the `kinit` command. If this has not happened yet, SSSD configuration will fail because it uses Kerberos. Verify that `kinit` works for a valid domain user by reviewing the "Setting up Kerberized NFS" section of this document.

For condensed setup steps, see the "Quick Step Setup Guides" section in this document.

SSSD Configuration Information

SSSD config is done via the `/etc/sssd/sssd.conf` file on clients that support SSSD. Each time a configuration change is made, SSSD should be restarted.

SSSD can be configured to cache the name database on the client. For performance reasons, NetApp recommends doing this. However, caching can cause confusion in troubleshooting, so when restarting the service during troubleshooting, the cache can be cleared with the following:

```
[client] # service sssd stop
[client] # rm -f /var/lib/sss/db/*
[client] # service sssd start
```

Additionally, SSSD is case sensitive by default. NetApp recommends configuring SSSD to ignore case sensitivity, because Microsoft Active Directory does not care about case sensitivity.

4.2.6 RHEL/CentOS/Fedora Client Configuration

The following assumes that the kernel running supports the SSSD LDAP package. Some newer versions of Linux include SSSD by default in basic installations. If SSSD is not installed, install it.

To check for the application:

```
[client] # yum list | grep sssd
```

To install:

```
[client] # yum install sssd
```

If the application is already installed, it may be beneficial to upgrade:

```
[client] # yum update sssd
```

Configuring SSSD on RHEL/CentOS/Fedora

Once the application is installed, the `/etc/sssd/sssd.conf` file needs to be configured.

For an example of a working SSSD configuration, see the sample `sssd.conf` file at the end of this section.

The `sssd.conf` file is configured with specific parameters to leverage Kerberos. See the table at the end of this section for descriptions of important options in the file. For more detail on the `sssd.conf` file, see the SSSD documentation or the `/etc/sssd/sssd.conf` man pages.

Note: The `/etc/sssd/sssd.conf` file does not exist in some SSSD implementations by default and needs to be created. Once the file is created, set the permissions to 0600 and the owner to root:root.

```
[client] # chmod 0600 /etc/sssd/sssd.conf
[client] # chown root:root /etc/sssd/sssd.conf
```

Once the `/etc/sssd/sssd.conf` is configured, modify `/etc/nsswitch.conf` to use SSSD for name services. The “sss” entry will be used for passwd, group, and shadow.

Example:

```
[client] # cat /etc/nsswitch.conf

passwd:     files sss
shadow:     files sss
group:      files sss

hosts:      files dns
```



```
bootparams: nisplus [NOTFOUND=return] files

ethers:     files
netmasks:   files
networks:   files
protocols:  files
rpc:        files
services:   files

netgroup:   nisplus

publickey:  nisplus

automount:  files nisplus
aliases:    files nisplus
```

Alternately, use the following command (**preferred method**):

```
[client] # authconfig --enablesssd --enablesssdauth –updateall
```

Once `/etc/nsswitch.conf` is configured, the sssd service can be started:

```
[client] # service sssd restart
Stopping sssd:                                             [  OK  ]
Starting sssd:                                             [  OK  ]
```

SSSD Client Troubleshooting

After starting sssd, check that LDAP entries are returning information with the following commands:

```
[client] # getent passwd ldapuser
ldapuser:*:1101:503:ldapuser:/home/ldapuser:/bin/sh
[client] # getent group "Domain Users"
Domain Users:*:513:ldapuser
```

If entries are returned, then the configuration is complete.

If no entries are returned, or there are any errors on service restart, check the following:

- `/etc/sssd/sssd.conf` file is 0600 permissions and root:root owns the file.
- Kerberos ticket is issued (klist) and not expired; if not issued or expired, use `kinit` to get a ticket.
- `kinit –k root/hostname` succeeds.
- Configuration file is free of typos.
- `/etc/nsswitch.conf` is configured to use SSSD.
- SPN exists in the KDC and there are no duplicates.
- DNS is configured properly.
- All DNS servers in `/etc/resolv.conf` are functional, especially the first in the list.
- Client time is within 5 minutes of the KDC.

Keep in mind that when restarting SSSD, a database cache also needs to be cleared so that lookups work.

To clear the SSSD cache when restarting the service, do the following:

```
[client] # service sssd stop
[client] # rm –f /var/lib/sss/db/*
[client] # service sssd start
```

If the above are verified, the following log files can be useful:

```
/var/log/messages
```



```
/var/log/sssd/* (be sure to enable debugging in /etc/sssd/sssd.conf)
```

In addition to checking LDAP lookups, confirm that the client can su and ssh using the LDAP user:

```
[client] # su ldapuser
sh-4.1$ id
uid=1101(ldapuser) gid=503(unixadmins) groups=503(unixadmins),513(Domain Users)
context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
ldapuser@centos6-4's password:
-sh-4.1$
-sh-4.1$ id
uid=1101(ldapuser) gid=503(unixadmins) groups=503(unixadmins),513(Domain Users)
context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
```

If su or ssh fails:

- Check the PAM configuration files in `/etc/pam.d` for the `pam_sss.so`.
- If not included, rerun `authconfig --enablesssd --enablesssdauth --updateall`.
- Check the `/var/log/secure` log file for errors.
- Verify that the sssd service is running.
- Verify that the client firewall is not blocking SSH.

Best Practice

Before rebooting the client, verify that new SSH sessions work properly. Existing sessions will remain usable, but if PAM gets misconfigured and the server is rebooted, the server may need to be rebuilt.

4.2.7 SUSE/SLES Client Configuration

The following assumes that the kernel running supports the SSSD LDAP package. If SSSD is not installed, install it.

To check for the application (SLES/SUSE uses zypper by default):

```
[client] # zypper search sssd
```

To install:

```
[client] # zypper install sssd
```

If the application is already installed, it may be beneficial to upgrade:

```
[client] # zypper update sssd
```

Configuring SSSD on SUSE/SLES

Once the application is installed, the `/etc/sssd/sssd.conf` file needs to be configured.

For an example of a working SSSD configuration, see the sample `sssd.conf` file at the end of this section.

The `sssd.conf` file is configured with specific parameters to leverage Kerberos. See Table 30 in section 4.2.10 for descriptions of important options in the file. For more detail on the `sssd.conf` file, see the SSSD documentation or the `/etc/sssd/sssd.conf` man pages.

Note: The `/etc/sssd/sssd.conf` file does not exist in some SSSD implementations by default and needs to be created. Once the file is created, set the permissions to 0600 and the owner to root:root.


```
[client] # chmod 0600 /etc/sssd/sssd.conf
[client] # chown root:root /etc/sssd/sssd.conf
```

After `/etc/sssd/sssd.conf` is configured, modify `/etc/nsswitch.conf` to use SSSD for name services. The “sss” entry will be used for passwd and group.

Example:

```
[client] # cat /etc/nsswitch.conf

passwd: files sss compat
group:  files sss compat

hosts:          files dns
networks:       files dns

services:       files
protocols:      files
rpc:            files
ethers:         files
netmasks:       files
netgroup:       files nis
publickey:      files

bootparams:     files
automount:      files nis
aliases:        files
```

Once `/etc/nsswitch.conf` is configured, verify that PAM is configured to use the sss and krb5 modules:

```
[client] # pam-config --add --sss
[client] # pam-config --add --krb5
```

Note: The default settings in `/etc/pam.d/common-auth` and `/etc/pam.d/common-account` may be too restrictive and cause login issues. Review these files to verify that the `pam_sss` and `pam_krb5` modules are set to “sufficient” rather than “required” before the solution is completed.

Best Practice

Before rebooting the client, verify that new SSH sessions work properly. Existing sessions will remain usable, but if PAM gets misconfigured and the server is rebooted, the server may need to be rebuilt.

Sample `/etc/pam.d/common-auth` and `/etc/pam.d/common-account` files:

```
sles11:/etc/sssd # cat /etc/pam.d/common-auth
#%PAM-1.0
auth    required        pam_env.so
auth    sufficient      pam_unix2.so
auth    sufficient      pam_krb5.so     use_first_pass
auth    sufficient      pam_sss.so      use_first_pass
sles11:/etc/sssd # cat /etc/pam.d/common-account
#%PAM-1.0
account requisite       pam_unix2.so
account sufficient      pam_krb5.so     use_first_pass
account sufficient      pam_localuser.so
account sufficient      pam_sss.so      use_first_pass
```

Enable the following to start at each boot (SUSE only):

```
[client] # systemctl enable sssd.service
```

The sssd service can then be started:


```
[client] # service sssd restart
```

SSSD Client Troubleshooting

After starting sssd, check that LDAP entries are returning information with the following commands:

```
[client] # getent passwd ldapuser
ldapuser:*:1101:503:ldapuser:/home/ldapuser:/bin/sh
[client] # getent group "Domain Users"
Domain Users:*:513:ldapuser
```

If entries are returned, then the configuration is complete.

If no entries are returned, or there are any errors on service restart, check the following:

- `/etc/sssd/sssd.conf` file is 0600 permissions and root:root owns the file.
- Kerberos ticket is issued (klist) and not expired; if not issued or expired, use `kinit` to get a ticket.
- `kinit –k root/hostname` succeeds.
- Configuration file is free of typos.
- `/etc/nsswitch.conf` is configured to use SSSD.
- SPN exists in the KDC and there are no duplicates.
- DNS is configured properly.
- All DNS servers in `/etc/resolv.conf` are functional, especially the first in the list.
- Client time is within 5 minutes of the KDC.

Keep in mind that when restarting SSSD, a database cache also needs to be cleared to verify that lookups work.

To clear the SSSD cache when restarting the service, do the following:

```
[client] # service sssd stop
[client] # rm –f /var/lib/sss/db/*
[client] # service sssd start
```

If the above are verified, the following log files can be useful:

```
/var/log/messages
/var/log/sssd/* (be sure to enable debugging in /etc/sssd/sssd.conf)
```

In addition to checking LDAP lookups, confirm that the client can su and ssh using the LDAP user:

```
sles11:~ # su ldapuser
sles11:/root> exit
exit

sles11:~ # ssh ldapuser@sles11
The authenticity of host 'sles11 (127.0.0.2)' can't be established.
RSA key fingerprint is 0d:c8:9c:20:5c:cd:35:c5:15:c1:a1:a4:a7:00:23:db.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'sles11' (RSA) to the list of known hosts.
Password:
sles11:/>
```

If su or ssh fails:

- Check the PAM configuration files in `/etc/pam.d` for the `pam_sss.so` module.
- Verify that PAM is not configured to be too restrictive (“required” rather than “sufficient”).
- Check the logs for errors.
- Verify that the sssd service is running.

- Verify that the client firewall is not blocking SSH.

4.2.8 Ubuntu Client Configuration

The following assumes that the kernel running supports the SSSD LDAP package. Some newer versions of Ubuntu include SSSD by default in basic installations. If SSSD is not installed, install it.

To check for the application (Ubuntu uses apt-get by default):

```
[client] # yum list | grep sssd
```

To install:

```
[client] # yum install sssd
```

If the application is already installed, it may be beneficial to upgrade:

```
[client] # yum update sssd
```

Configuring SSSD on Ubuntu

Once the application is installed, the `/etc/sssd/sssd.conf` file needs to be configured.

For an example of a working SSSD configuration, see the sample `sssd.conf` file at the end of this section.

The `sssd.conf` file is configured with specific parameters to leverage Kerberos. See Table 30in section 4.2.10 for descriptions of important options in the file. For more detail on the `sssd.conf` file, see the SSSD documentation or the `/etc/sssd/sssd.conf` man pages.

Note: The `/etc/sssd/sssd.conf` file does not exist in some SSSD implementations by default and needs to be created. Once the file is created, set the permissions to 0600 and the owner to root:root.

```
[client] # chmod 0600 /etc/sssd/sssd.conf
[client] # chown root:root /etc/sssd/sssd.conf
```

After `/etc/sssd/sssd.conf` is configured, modify `/etc/nsswitch.conf` to use SSSD for name services. The “sss” entry will be used for passwd and group. Ubuntu may configure this by default when SSSD is installed.

Example:

```
[client] # cat /etc/nsswitch.conf
[client] # /etc/nsswitch.conf

passwd:         compat sss
group:          compat sss
shadow:         compat sss

hosts:          files dns
networks:       files

protocols:      db files
services:       db files
ethers:         db files
rpc:            db files

netgroup:       nis sss
```

Note: The default settings in `/etc/pam.d/common-auth, /etc/pam.d/common-session`, and `/etc/pam.d/common-account` may be too restrictive and cause login issues. Review these files to verify that the `pam_sss` and `pam_krb5` modules are set to something other than “required” before the solution is completed.

Best Practice

Before rebooting the client, verify that new SSH sessions work properly. Existing sessions will remain usable, but if PAM gets misconfigured and the server is rebooted, the server may need to be rebuilt.

Sample `/etc/pam.d/common-auth`, `/etc/pam.d/common-session`, and `/etc/pam.d/common-account` files:

```
root@ubuntu:/etc/init.d# cat /etc/pam.d/common-auth | grep -v "#"

auth    [success=2 default=ignore]      pam_unix.so nullok_secure
auth    [success=1 default=ignore]                      pam_sss.so use_first_pass
auth    requisite                       pam_deny.so
auth    required                        pam_permit.so

root@ubuntu:/etc/init.d# cat /etc/pam.d/common-session | grep -v "#"

session [default=1]                     pam_permit.so
session requisite                       pam_deny.so
session required                        pam_permit.so
session optional                        pam_umask.so
session required        pam_unix.so
session optional                                        pam_sss.so
session optional                        pam_ck_connector.so nox11

root@ubuntu:/etc/init.d# cat /etc/pam.d/common-account | grep -v "#"

account [success=1 new_authtok_reqd=done default=ignore]        pam_unix.so
account requisite                       pam_deny.so
account required                        pam_permit.so
account sufficient                                      pam_localuser.so
account [default=bad success=ok user_unknown=ignore]    pam_sss.so
```

The sssd service can then be started:

```
[client] # service sssd restart
```

SSSD Client Troubleshooting

After starting sssd, check that LDAP entries are returning information with the following commands:

```
[client] # getent passwd ldapuser
ldapuser:*:1101:503:ldapuser:/home/ldapuser:/bin/sh
[client] # getent group "Domain Users"
Domain Users:*:513:ldapuser
```

If entries are returned, then the configuration is complete.

If no entries are returned, or there are any errors on service restart, check the following:

- `/etc/sssd/sssd.conf` file is 0600 permissions and root:root owns the file.
- Kerberos ticket is issued (klist) and not expired; if not issued or expired, use `kinit` to get a ticket.
- `kinit -k root/hostname` succeeds.
- The configuration file is free of typos.
- `/etc/nsswitch.conf` is configured to use SSSD
- SPN exists in the KDC and there are no duplicates.
- DNS is configured properly.
- All DNS servers in `/etc/resolv.conf` are functional, especially the first in the list.
- Client time is within 5 minutes of the KDC.

Keep in mind that when restarting SSSD, a database cache also needs to be cleared to verify that lookups work.

To clear the SSSD cache when restarting the service, do the following:

```
[client] # service sssd stop
[client] # rm -f /var/lib/sss/db/*
[client] # service sssd start
```

If the above are verified, the following log files can be useful:

```
/var/log/messages
/var/log/sssd/* (be sure to enable debugging in /etc/sssd/sssd.conf)
```

In addition to checking LDAP lookups, confirm that the client can su and ssh using the LDAP user.

```
root@ubuntu:/etc/init.d# su ldapuser
$ exit
root@ubuntu:/etc/init.d# ssh ldapuser@ubuntu
The authenticity of host ubuntu (127.0.1.1)' can't be established.
ECDSA key fingerprint is 49:ae:ef:54:f4:7e:2c:45:f0:9e:24:ce:da:17:ee:53.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'ubuntu' (ECDSA) to the list of known hosts.
ldapuser@ubuntu's password:

$
```

If su or ssh fails:

- Check the PAM configuration files in `/etc/pam.d` for the `pam_sss.so` module.
- Verify that PAM is not configured to be too restrictive.
- Check the logs for errors.
- Verify that the sssd service is running.
- Verify that the client firewall is not blocking SSH.

4.2.9 `sssd.conf` File Example

Below is a sample file from a working configuration.

```
[domain/default]
cache_credentials = True
case_sensitive = False

[sssd]
config_file_version = 2
services = nss, pam
domains = DOMAIN
#debug_level = 0 – Set this to troubleshoot; 0-10 are valid values

[nss]
filter_users = root,ldap,named,avahi,haldaemon,dbus,radiusd,news,nscd
filter_groups = root

[pam]

[domain/YOURDOMAINNAME]
id_provider = ldap
auth_provider = krb5
# Case sensitive is specified to ensure NFSv4.x ID maps work properly.
case_sensitive = true
chpass_provider = krb5
cache_credentials = false

#ldap_uri = _srv_,ldap://ldap.netapp.com – leave out of the file to use LDAP SRV records
ldap_search_base = dc=domain,dc=netapp,dc=com
```



```
ldap_schema = rfc2307
ldap_sasl_mech = GSSAPI
ldap_user_object_class = user
ldap_group_object_class = group
ldap_user_home_directory = unixHomeDirectory
ldap_user_principal = userPrincipalName
ldap_account_expire_policy = ad
ldap_force_upper_case_realm = true
ldap_user_search_base = cn=Users,dc=domain,dc=netapp,dc=com
ldap_group_search_base = cn=Users,dc=domain,dc=netapp,dc=com
ldap_sasl_authid = root/nfsclient.domain.netapp.com@DOMAIN.NETAPP.COM
#entry_cache_timeout = 120 – useful for troubleshooting; omit otherwise

#krb5_server = domain.netapp.com – leave out of the file to use LDAP SRV records
krb5_realm = DOMAIN.NETAPP.COM
```

Note: In RHEL 6.3, add the following line in addition to the ones above:

```
krb5_canonicalize = False
```

For information on this, see the following:

What is krb5_canonicalize?

Best Practice

To avoid slow lookups for users and groups, specify the `ldap_user_search_base` and `ldap_group_search_base` options to help direct the LDAP lookups to the proper locations and avoid crawling large LDAP databases for entries.

4.2.10 SSSD Configuration File Options

Table 30) /etc/sssd/sssd.conf file options.

Option	Use Case
cache_credentials	Caches the LDAP credentials on the client for improved lookup performance.
case_senstive	Ignores case sensitivity in LDAP lookups.
devices	Services to start when SSSD starts.
domains	Defines the database in the config; SSSD can use multiple domains; will use in order of config listing.
debug_level	Sets the debug level for troubleshooting; can be commented out if desired.
filter_users	Exclude users from use with SSSD.
filter_groups	Exclude groups from use with SSSD.
id_provider	Identity provider.
auth_provider	Authentication provider.

chpass_provider	Password change provider.
ldap_uri	Address for LDAP queries; optional—leave this out if using more than one DC in a domain to leverage SRV records for failover.
ldap_search_base	Base DN for LDAP queries.
ldap_schema	Schema for use with LDAP; RFC-2307bis is the default. However, clustered Data ONTAP does not currently support RFC-2307bis.
ldap_sasl_mech	Auth mechanism for SASL; GSSAPI is used for Kerberos auth.
ldap_user_object_class ldap_group_object_class ldap_user_home_directory ldap_user_principal ldap_group_member ldap_group_name	LDAP schema attributes; these will determine how the client looks for LDAP information.
ldap_account_expire_policy	Specifies the account expiration policy rule.
ldap_force_upper_case_realm	Forces the realm to be in ALL CAPS; NetApp recommends setting this to “True.”
ldap_group_search_base ldap_user_search_base	Base DN for groups and users.
ldap_sasl_authid	Specifies the SPN for the client to authenticate; when GSSAPI is used, specify the client’s SPN. If not specified, the client will attempt to use host/hostname@REALM as the SPN and the request will fail if that SPN does not exist.
krb5_server krb5_realm krb5_kpasswd	Krb5 settings—kpasswd and server are optional; leave these out if using more than one DC in a domain to leverage SRV records for failover.
entry_cache_timeout	The number of seconds that nss_sss should consider entries valid before asking the back end again; useful for troubleshooting issues.
cache_credentials	Determines if user credentials are also cached in the local LDB cache. User credentials are stored in a SHA512 hash, not in plaintext.
krb5_canonicalize	Use with RHEL 6.3.

The [domain/YOURDOMAINNAME] Section

The `[domain/YOURDOMAINNAME` section tells SSSD the domain parameters to use. The `domains` option will tell SSSD which domain is used. Multiple domains can be specified. The `YOURDOMAINNAME` portion of the entry can be any value, provided that value is specified in the `domains` option. It is simply a placeholder for the domain name.

For example, the following are all valid values:

```
[domain/DOMAIN]
[domain/HELLO_WORLD]
[domain/NETAPP]
```

To use all of the above domains in order, set default_domain as such:

```
domains = DOMAIN,HELLO_WORLD,NETAPP
```

4.2.11 Configuring Solaris to Use LDAP

The following section covers configuration of Solaris to use Active Directory LDAP without the use of SSSD. The following needs to be done before configuring LDAP:

- Create the machine account and SPN/keytab file for the Solaris client as per the "Creating principals/keytabs" section of this document.
- Configure the LDAP server as per the "Configuring the domain controller as an LDAP server" section of this document.
- Configure Kerberos and perform a successful kinit to the Windows KDC as per the "Solaris Kerberos configuration" section of this document.

Example:

```
# kinit ldapuser@DOMAIN.NETAPP.COM
```

Once this is completed, the Solaris client can be configured for LDAP using the `ldapclient` utility.

LDAP can be configured for *simple* authentication or for *sasl/GSSAPI* leveraging Kerberos.

Simple authentication

When an LDAP query is made via simple authentication, the request is passed in plain text over the wire. To encrypt LDAP queries, use sasl/GSSAPI.

Example of simple authentication configuration:

```
ldapclient manual \
-a credentialLevel=proxy \
-a authenticationMethod=simple \
-a proxyDN=CN=ldapuser,CN=Users,DC=domain,DC=netapp,DC=com \
-a proxyPassword=P@ssw0rd \  <<<< optional
-a defaultSearchBase=dc=domain,dc=netapp,dc=com \
-a defaultSearchScope=sub \
-a domainName=domain.netapp.com \
-a defaultServerList=10.61.179.152 \
-a attributeMap=group:userpassword=userPassword \
-a attributeMap=group:memberuid=memberUid \
-a attributeMap=group:gidnumber=gidNumber \
-a attributeMap=passwd:gecos=cn \
-a attributeMap=passwd:gidnumber=gidNumber \
-a attributeMap=passwd:uidnumber=uidNumber \
-a attributeMap=passwd:homedirectory=unixHomeDirectory \
-a attributeMap=passwd:loginshell=loginShell \
-a attributeMap=shadow:shadowflag=shadowFlag \
-a attributeMap=shadow:userpassword=userPassword \
-a objectClassMap=group:posixGroup=group \
```



```
-a objectClassMap=passwd:posixAccount=user \
-a objectClassMap=shadow:shadowAccount=user \
-a serviceSearchDescriptor=passwd:CN=Users,DC=domain,DC=netapp,DC=com?sub \
-a serviceSearchDescriptor=group:CN=Users,DC=domain,DC=netapp,DC=com?sub
```

The bind password can be issued in the configuration command in plain text or it can be entered with a prompt. This is controlled by the `proxyPassword` attribute. If the attribute is left out, a password prompt is used.

```
credentialLevel requires proxyPassword
Proxy Bind Password:
System successfully configured
```

The credentials are then stored in the `ldap_client_cred` file in `/var/ldap`. The password is encrypted in the file.

```
bash-3.00# cat ldap_client_cred
#
# Do not edit this file manually; your changes will be lost.Please use ldapclient (1M) instead.
#
NS_LDAP_BINDDN= CN=ldapuser,CN=Users,DC=domain,DC=netapp,DC=com
NS_LDAP_BINDPASSWD= {NS1}414f88f3a9bfc411
```

sasl/GSSAPI authentication

sasl/GSSAPI configuration will leverage Kerberos tickets for LDAP queries. The ticket is obtained when the LDAP client is configured and will leverage the machine account's SPN found in the keytab file.

```
bash-3.00# klist -e
Ticket cache: FILE:/tmp/krb5cc_0
Default principal: root/solaris-mit.domain.netapp.com@DOMAIN.NETAPP.COM

Valid starting                Expires                Service principal
06/27/13 11:00:25  06/27/13 21:00:25  krbtgt/DOMAIN.NETAPP.COM@DOMAIN.NETAPP.COM
        Etype(skey, tkt): AES-256 CTS mode with 96-bit SHA-1 HMAC, AES-256 CTS mode with 96-bit
SHA-1 HMAC
06/27/13 11:00:56  06/27/13 21:00:25  ldap/2k8-dc-1.domain.netapp.com@
        Etype(skey, tkt): AES-256 CTS mode with 96-bit SHA-1 HMAC, AES-256 CTS mode with 96-bit
SHA-1 HMAC
```

Example of sasl/GSSAPI authentication configuration:

```
ldapclient manual \
-a credentialLevel=self \
-a authenticationMethod=sasl/GSSAPI \
-a defaultSearchBase=dc=domain,dc=netapp,dc=com \
-a defaultSearchScope=sub \
-a domainName=domain.netapp.com \
-a defaultServerList=10.61.179.152 \
-a attributeMap=group:userpassword=userPassword \
-a attributeMap=group:memberuid=memberUid \
-a attributeMap=group:gidnumber=gidNumber \
-a attributeMap=passwd:gecos=cn \
-a attributeMap=passwd:gidnumber=gidNumber \
-a attributeMap=passwd:uidnumber=uidNumber \
-a attributeMap=passwd:homedirectory=unixHomeDirectory \
-a attributeMap=passwd:loginshell=loginShell \
-a attributeMap=shadow:shadowflag=shadowFlag \
-a attributeMap=shadow:userpassword=userPassword \
-a objectClassMap=group:posixGroup=group \
-a objectClassMap=passwd:posixAccount=user \
-a objectClassMap=shadow:shadowAccount=user \
-a serviceSearchDescriptor=passwd:CN=Users,DC=domain,DC=netapp,DC=com?sub \
-a serviceSearchDescriptor=group:CN=Users,DC=domain,DC=netapp,DC=com?sub
```


The difference between a simple and a sasl/GSSAPI configuration is the attribute values for `credentialLevel` and `authenticationMethod` as well as the removal of the `proxyDN` and `proxyPassword` for binding. All binding in sasl/GSSAPI is done via Kerberos ticket authentication, so no passwords are required or stored.

How LDAP configuration in Solaris works

All LDAP configuration and logging for Solaris is stored in `/var/ldap`.

```
bash-3.00# ls -la
total 46
drwxr-xr-x   3 root     sys          512 Jun 26 19:55 .
drwxr-xr-x  46 root     sys         1024 Jun 26 12:53 ..
-rw-r--r--   1 root     root       17356 Jun 26 19:55 cachemgr.log
-r--------   1 root     root         216 Jun 26 19:55 ldap_client_cred
-r--------   1 root     root        1141 Jun 26 19:55 ldap_client_file
drwxr-xr-x   2 root     root         512 Jun 26 14:14 restore
```

If an error occurs during the configuration, the -v flag can be used with `ldapclient` to get verbose output. The logging of the configuration is done in the `/var/ldap/cachemgr.log` file.

```
bash-3.00# ldapclient -v mod -a authenticationMethod=simple
```

In the following example, an error occurs during the configuration due to the fact that the attributes `authenticationMethod` and `credentialLevel` are dependent on one another. When a failure occurs, the previous configuration will be restored.

```
bash-3.00# ldapclient mod -a authenticationMethod=simple
Error resetting system.
Recovering old system settings.
```

If the `authenticationMethod` is *simple*, `credentialLevel` must be *proxy*. If `authenticationMethod` is *sasl/GSSAPI*, the `credentialLevel` must be *self*. The `cachemgr.log` file will show the following error:

```
Error: Unable to read '/var/ldap/ldap_client_file': Configuration Error: Credential level self
requires authentication method sasl/GSSAPI
```

Once a valid configuration is applied, the `ldap_client_file` will be updated and can be viewed either with a text editor or using the `ldapclient list` command:

```
bash-3.00# ldapclient list
NS_LDAP_FILE_VERSION= 2.0
NS_LDAP_SERVERS= 10.61.179.152
NS_LDAP_SEARCH_BASEDN= dc=domain,dc=netapp,dc=com
NS_LDAP_AUTH= sasl/GSSAPI
NS_LDAP_SEARCH_SCOPE= sub
NS_LDAP_CACHETTL= 0
NS_LDAP_CREDENTIAL_LEVEL= self
NS_LDAP_SERVICE_SEARCH_DESC= passwd:CN=Users,DC=domain,DC=netapp,DC=com?sub
NS_LDAP_SERVICE_SEARCH_DESC= group:CN=Users,DC=domain,DC=netapp,DC=com?sub
NS_LDAP_ATTRIBUTEMAP= group:userpassword=userPassword
NS_LDAP_ATTRIBUTEMAP= group:memberuid=memberUid
NS_LDAP_ATTRIBUTEMAP= group:gidnumber=gidNumber
NS_LDAP_ATTRIBUTEMAP= passwd:gecos=cn
NS_LDAP_ATTRIBUTEMAP= passwd:gidnumber=gidNumber
NS_LDAP_ATTRIBUTEMAP= passwd:uidnumber=uidNumber
NS_LDAP_ATTRIBUTEMAP= passwd:homedirectory=unixHomeDirectory
NS_LDAP_ATTRIBUTEMAP= passwd:loginshell=loginShell
NS_LDAP_ATTRIBUTEMAP= shadow:shadowflag=shadowFlag
NS_LDAP_ATTRIBUTEMAP= shadow:userpassword=userPassword
NS_LDAP_OBJECTCLASSMAP= group:posixGroup=group
NS_LDAP_OBJECTCLASSMAP= passwd:posixAccount=user
NS_LDAP_OBJECTCLASSMAP= shadow:shadowAccount=user
```


Nsswitch files

In addition to the above, Solaris uses several nsswitch files:

```
bash-3.00# ls /etc | grep nsswitch
nsswitch.conf
nsswitch.dns
nsswitch.files
nsswitch.ldap
nsswitch.nis
nsswitch.nisplus
```

LDAP in particular uses the nsswitch.conf and nsswitch.ldap files. When LDAP is configured, the `nsswitch.conf` file will be replaced with the `nsswitch.ldap` file. By default, the `nsswitch.ldap` file will use LDAP and files **only** for all entries. This is problematic for environments using Windows as an LDAP server, since most Windows LDAP servers do not use a DN for hosts or ipnodes. As such, LDAP queries for these will fail and LDAP lookups may not work properly. To correct this, edit the `nsswitch.conf` and `nsswitch.ldap` files to include DNS for hosts and ipnodes:

```
bash-3.00# cat /etc/nsswitch.conf | grep hosts
# "hosts:" and "services:" in this file are used only if the
hosts:      dns files
# before searching the hosts databases.
bash-3.00# cat /etc/nsswitch.conf | grep ipnodes
# Note that IPv4 addresses are searched for in all of the ipnodes databases
ipnodes:    dns files
```

Verifying LDAP configuration

Once configuration is done, the configuration can be checked via the following command:

```
# ldapclient list
```

Restart the ldap service:

```
# svcadm restart network/ldap/client
```

Check the status:

```
# svcs network/ldap/client
```

Test lookups of users:

```
# getent passwd ldapuser
ldapuser:x:10000:503:ldapuser:/home/ldapuser:/bin/sh
```

For more information, see the Oracle "Solaris System Administration Guide" for naming and directory services.

4.2.12 Configuring the clustered Data ONTAP System to Use LDAP

The following section describes how to configure a clustered Data ONTAP system to use LDAP for its name mapping. This section assumes that a working LDAP server running on a Windows 2008 R2 domain controller exists and is reachable by the SVM data LIFs. If this is not the case, see the section entitled "Configuring the domain controller as an LDAP server" for details.

For condensed setup steps, see the "Quick Step Setup Guides" section in this document.

4.2.13 LDAP Schemas

In clustered Data ONTAP, built-in read-only schemas are available to admins. These schemas can be used to configure LDAP or can be copied to read-writable schemas to allow modification of the schema

attributes for LDAP servers that do not contain the same attributes as any of the default schemas. LDAP schemas must exist prior to configuring an LDAP client configuration.

Table 31) Default schemas available in clustered Data ONTAP.

Clustered Data ONTAP 8.2	Clustered Data ONTAP 8.0.x and 8.1
AD-SFU AD-IDMU AD-SFU-Deprecated RFC-2307	AD-SFU RFC-2307

Note: Examples of each schema can be found in the Appendix under "LDAP schema examples in clustered Data ONTAP 8.2."

Schemas are used in LDAP queries by the cluster to speed up name lookups by being able to use specific attributes to find information about a user, such as the UID. The schema attributes must exist in the LDAP server for the cluster to be able to find the entry. Otherwise, LDAP queries may return no data and authentication requests may fail.

For example, if a UID number (such as root=0) needs to be queried by the cluster, then the schema attribute "RFC 2307 uidNumber Attribute" will be used. The default schema for AD-IDMU uses the attribute msSFU30UidNumber for that query.

If the schema is configured correctly, the query will return the appropriate value.

```
cluster::> set diag
cluster::*> diag secd authentication translate -node node1 -vserver vs0 -unix-user-name ldapuser
1101
```

If an incorrect schema is specified (such as using RFC-2307 for Windows 2008R2 instead of AD-IDMU), queries would fail because incorrect attributes would be passed to the LDAP server.

LDAP queries in clustered Data ONTAP are passed through the SecD application. These queries use a simple ldapsearch to find information and can be seen in the SecD log on failed attempts, which can be useful for troubleshooting LDAP issues.

In the following example, SecD is asked to look for a user that does not have a valid UID number in LDAP.

```
cluster::> set diag
cluster::*> diag secd authentication translate -node node1 -vserver vs0 -unix-user-name nouser

Vserver: vs0 (internal ID: 8)

Error: Acquire UNIX credentials procedure failed
  [  0 ms] Name 'nouser' not found in UNIX authorization source LOCAL
  [     1] Using a cached connection to 10.63.98.101
  [     3] Name 'nouser' not found in UNIX authorization source LDAP
  [     3] Could not get a user ID for name 'nouser' using any
           NS-SWITCH authorization source
**[     3] FAILURE: Unable to retrieve UID for UNIX user nouser

Error: command failed: Failed to resolve user name to a UNIX ID. Reason:
"SecD Error: user not found".
```

In the SecD log, the following can be seen in the failure:

```
[kern_secd:info:10629] | [000.002.384]  debug:  Searching LDAP for the "uidNumber, gidNumber"
attribute(s) within base "cn=users,DC=domain,DC=netapp,DC=com" (scope: 2) using filter:
(&(objectClass=User)(uid=nouser))  { in searchLdap() at utils/secd_ldap_utils.cpp:257 }
```



```
 [kern_secd:info:10629] | [000.003.857]  ERR  :  RESULT_ERROR_SECD_UNIX_CRED_LOOKUP_FAILED:6987
in getFailureCode() at utils/secd_thread_task_journal.cpp:292
```

In the example, the specific LDAP filter used by SecD is specified:

```
(&(objectClass=User)(uid=nouser))
```

And the specific attributes are specified:

```
Searching LDAP for the "uidNumber, gidNumber" attribute(s)
```

As well as the base:

```
within base "cn=users,DC=domain,DC=netapp,DC=com"
```

Additionally, the scope type is specified:

```
(scope: 2)
```

The following table defines the scopes used in SecD logging:

Table 32) SecD scope definitions.

Scope	Definition
Scope -1	Invalid scope
Scope 0	Base
Scope 1	Onelevel
Scope 2	Subtree

The LDAP information in the error can be used to formulate an `ldapsearch` query to run manually on the LDAP server or an NFS client.

In Windows Active Directory, it is possible to search LDAP using the built-in `ldifde` utility and leveraging the attributes found in the SecD error:

```
C:\>ldifde -d "cn=users,DC=domain,DC=netapp,DC=com"
-r "(&(objectClass=User)(uid=nouser))" -l "uidNumber, gidNumber" -f filename.ldf
Connecting to "windowsDC.domain.netapp.com"
Logging in as current user using SSPI
Exporting directory to file filename.ldf
Searching for entries...
Writing out entriesldap://domain.netapp.com/cn=users,DC=domain,DC=netapp,DC=com

No Entries found

The command has completed successfully

C:\>more filename.ldf
C:\>
```

The above example shows that there are indeed no entries for the user “nouser” in LDAP.

4.2.14 Viewing SecD Logs

To view SecD logs, use the following commands:

```
cluster::> set diag
cluster::*> debug log files modify -incl-files secd
cluster::*> debug log show -timestamp >"Mon May 06 11:48:33 2013"
```


The timestamp needs to be specified in the format above. Use > or < to specify “before and after” to filter log files. A time range can also be specified. For details, use:

```
cluster::> set diag
cluster::*> man debug log show
```

4.2.15 Creating a Custom LDAP Schema

To create a custom LDAP schema, first copy an existing read-only schema as the base. Read-only schemas cannot be modified:

```
cluster::> ldap client schema modify -schema AD-SFU -vserver vs0 -comment modify
  (vserver services ldap client schema modify)

Error: command failed: You are not authorized to perform this operation

cluster::> set advanced
cluster::*> ldap client schema copy -schema AD-SFU -new-schema-name NEWSCHEMA -vserver vs0
```

Once the schema is copied, the new schema template can be modified:

```
cluster::> ldap client schema modify -schema NEWSCHEMA -vserver vs0 -comment modify
cluster::> ldap client schema show -schema NEWSCHEMA -vserver vs0 -fields comment
  (vserver services ldap client schema show)
vserver schema    comment
------- --------- -------
vs0    NEWSCHEMA modify
```

4.2.16 LDAP Clients

In clustered Data ONTAP, LDAP clients are needed to specify the configuration to be used by the SVM. **A schema must be defined before creating a client** or one of the default schemas should be used. If no valid schemas are specified, the command will fail.

When LDAP clients are created in admin mode, the following options are allowed:

```
cluster::> ldap client create ?
  (vserver services ldap client create)
  [ -vserver <vserver name> ]                  Vserver (default: cm6080-rtp2)
   [-client-config] <text (size 1..32)>        Client Configuration Name
   { [-servers] <IpAddress>, ...               LDAP Server List
   | [-ad-domain] <TextNoCase>                 Active Directory Domain
    [ -preferred-ad-servers <IpAddress>, ... ]  Preferred Active Directory Servers
    [ -bind-as-cifs-server {true|false} ] }     Bind Using the Vserver's CIFS Credentials
(default: false)
   [-schema] <text>                            Schema Template
  [ -port {1..65535} ]                         LDAP Server Port (default: 389)
  [ -query-timeout {0..10} ]                   Query Timeout (sec) (default: 3)
  [ -min-bind-level {anonymous|simple|sasl} ]  Minimum Bind Authentication Level (default:
anonymous)
  [ -bind-dn <LDAP DN> ]                       Bind DN (User)
  [ -base-dn <LDAP DN> ]                       Base DN (default: "")
  [ -base-scope {base|onelevel|subtree} ]      Base Search Scope (default: subtree)
```

When LDAP clients are created in advanced mode, the following options are allowed:

```
cluster::*> ldap client create ?
  (vserver services ldap client create)
  [ -vserver <vserver name> ]                  Vserver (default: cm6080-rtp2)
   [-client-config] <text (size 1..32)>        Client Configuration Name
   { [-servers] <IpAddress>, ...               LDAP Server List
   | [-ad-domain] <TextNoCase>                 Active Directory Domain
    [ -preferred-ad-servers <IpAddress>, ... ]  Preferred Active Directory Servers
    [ -bind-as-cifs-server {true|false} ] }     Bind Using the Vserver's CIFS Credentials
(default: false)
   [-schema] <text>                            Schema Template
```



```
  [ -port {1..65535} ]                          LDAP Server Port (default: 389)
  [ -query-timeout {0..10} ]                    Query Timeout (sec) (default: 3)
  [ -min-bind-level {anonymous|simple|sasl} ]   Minimum Bind Authentication Level (default:
anonymous)
  [ -bind-dn <LDAP DN> ]                        Bind DN (User)
  [ -base-dn <LDAP DN> ]                        Base DN (default: "")
  [ -base-scope {base|onelevel|subtree} ]       Base Search Scope (default: subtree)
  [ -user-dn <LDAP DN> ]                        *User DN
  [ -user-scope {base|onelevel|subtree} ]       *User Search Scope (default: subtree)
  [ -group-dn <LDAP DN> ]                       *Group DN
  [ -group-scope {base|onelevel|subtree} ]      *Group Search Scope (default: subtree)
  [ -netgroup-dn <LDAP DN> ]                    *Netgroup DN
  [ -netgroup-scope {base|onelevel|subtree} ]   *Netgroup Search Scope (default: subtree)
```

Best Practice

When a CIFS server is present in an SVM, it is a best practice to bind the LDAP client as a CIFS server. Doing this leverages Kerberos for LDAP queries and enables the LDAP traffic to be encrypted over the wire. This cannot be done until after the LDAP client is created and needs to be done via “ldap client modify.”

The following command example can be used as guidance for creating an LDAP client with a custom schema in an SVM that also has a CIFS server. The bind DN has to be a valid user in the LDAP server. This user only needs to have read-only access to LDAP. In Windows Active Directory, any valid domain user will work.

```
cluster::> set advanced
cluster::*> ldap client create -client-config LDAP -servers 10.63.98.101 -schema -min-bind-level
sasl -base-dn dc=domain,dc=netapp,dc=com -base-scope subtree -user-scope subtree -group-scope
subtree -netgroup-scope subtree -bind-dn ldapuser -user-dn cn=users,DC=domain,DC=netapp,DC=com -
group-dn cn=users,DC=domain,DC=netapp,DC=com -vserver vs0
  (vserver services ldap client create)
Please enter password:
Confirm password:
Cluster::*> ldap client show -client-config LDAP -instance
  (vserver services ldap client show)

                                  Vserver: vs0
                Client Configuration Name: LDAP
                         LDAP Server List: 10.63.98.101
                  Active Directory Domain: -
       Preferred Active Directory Servers: -
Bind Using the Vserver's CIFS Credentials: false
                           Schema Template: NEWSCHEMA
                         LDAP Server Port: 389
                       Query Timeout (sec): 3
        Minimum Bind Authentication Level: sasl
                            Bind DN (User): ldapuser
                                   Base DN: dn=domain,dc=netapp,dc=com
                        Base Search Scope: subtree
                                  User DN: cn=users,DC=domain,DC=netapp,DC=com
                        User Search Scope: subtree
                                 Group DN: cn=users,DC=domain,DC=netapp,DC=com
                       Group Search Scope: subtree
                              Netgroup DN: -
                    Netgroup Search Scope: subtree
                Vserver Owns Configuration: true
```

After the LDAP client is created, modify the client to bind as a CIFS server.

```
cluster::*> ldap client modify -client-config LDAP -vserver vs0 -bind-as-cifs-server true -
preferred-ad-servers 10.63.98.101 -ad-domain domain.netapp.com
  (vserver services ldap client modify)
```



```
cluster::*> ldap client show -client-config LDAP -instance
  (vserver services ldap client show)

                                 Vserver: vs0
               Client Configuration Name: LDAP
                         LDAP Server List: 10.63.98.101
                  Active Directory Domain: domain.netapp.com
       Preferred Active Directory Servers: 10.63.98.101
Bind Using the Vserver's CIFS Credentials: true
                          Schema Template: NEWSCHEMA
                         LDAP Server Port: 389
                      Query Timeout (sec): 3
        Minimum Bind Authentication Level: sasl
                           Bind DN (User): ldapuser
                                  Base DN: dn=domain,dc=netapp,dc=com
                        Base Search Scope: subtree
                                  User DN: cn=users,DC=domain,DC=netapp,DC=com
                        User Search Scope: subtree
                                 Group DN: cn=users,DC=domain,DC=netapp,DC=com
                       Group Search Scope: subtree
                              Netgroup DN: -
                    Netgroup Search Scope: subtree
               Vserver Owns Configuration: true
```

4.2.17 LDAP Configuration

Once the client configuration is created, the LDAP configuration can be created. This command simply applies the client configuration to the SVM and enables the use of the LDAP configuration.

```
cluster::> ldap create -vserver vs0 -client-config LDAP -client-enabled true
```

4.2.18 LDAPS (LDAP over SSL)

LDAPS (LDAP over SSL) for Active Directory was introduced in clustered Data ONTAP 8.2.1 to allow encrypted LDAP queries. This prevents plain text LDAP queries from traveling over the wire. To configure SSL encrypted LDAP queries, a Certificate Server must already be configured in the domain.

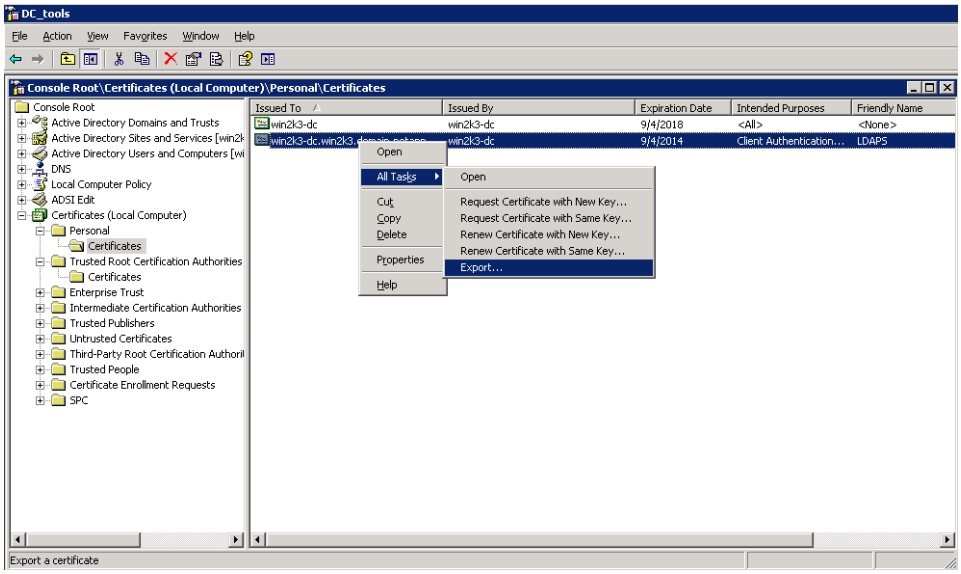
If a Certificate Server exists and is configured, then setting up LDAPS is a straightforward process.

Table 33) Configuring LDAP over SSL in Clustered Data ONTAP

1. Export the certificate to a CER format (DER encoded) file.

   This is done from the Certificates MMC snap-in on the Certificate Server.

   Example (Windows2k3):



2. Convert the DER file to a PEM format to get the human-readable text to allow copy and paste into clustered Data ONTAP.

   To do this, open a CLI, navigate to the location you exported the certificate to, and enter:

```
C:\> certutil -encode <exportedFileName> <PemFileName>
```

   This exports the certificate to text format, which is needed later to import into clustered Data ONTAP.

   Sample of resulting file:

```
-----BEGIN CERTIFICATE-----
MIIE6jCCA9KgAwIBAgIQGQUp+NqxOJhOWM+CvfbANjANBgkqhkiG9w0BAQUFADBx
MRMwEQYKCZImiZPyLGQBGRYDY29tMRYwFAYKCZImiZPyLGQBGRYGbmV0YXBwMRYw
FAYKCZImiZPyLGQBGRYGZG9tYWluMRYwFAYKCZImiZPyLGQBGRYGd2luMmszMRIw
EAYDVQQDEwl3aW4yazMtZGMwHhcNMTMwOTA0MTUzMTExWhcNMTgwOTA0MTUzODUz
WjBxMRMwEQYKCZImiZPyLGQBGRYDY29tMRYwFAYKCZImiZPyLGQBGRYGbmV0YXBw
MRYwFAYKCZImiZPyLGQBGRYGZG9tYWluMRYwFAYKCZImiZPyLGQBGRYGd2luMmsz
MRIwEAYDVQQDEwl3aW4yazMtZGMwggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEK
AoIBAQD41T8Blrrum1GML0Dy+dGu1P1ceL+0nkA6vA81xIy2CW/HYl8TWd7ZVq5n
IK9z86bKSnMaDKBZ8wpAhYAzkG2yA7AIaqHi3MZjUoty7+C/T/75O5bScxFgacKY
IMexOb1iLTVPx3a/jOHzy4a27TEMQig/YAHTOz/CKKBi0/u4/2KKCOKHhoTaUNes
NUIEVizKUwIbNRRDb9LDRvIMWm5zfzaZ2M6PWMkX5/ZwujFgd+GOTMjC4+H78SOA
CalhA1CLR364vGCWMEUWdi5lMIDSmZyR5Vpx6dLijWjFUpLcblGk2VuFj1jhvOs/
tMd89MIZgEEaULjXIgqTqt/BoZDNAgMBAAGjggF8MIIBeDALBgNVHQ8EBAMCAYYw
DwYDVR0TAQH/BAUwAwEB/zAdBgNVHQ4EFgQUrtBPBez/V6bDpN/TSWS/Azm7C7ww
ggElBgNVHR8EggEcMIIBGDCCARSgggEQoIIBDIaBxWxkYXA6Ly8vQ049d2luMmsz
LWRjLENOPXdpbjJrMy1kYyxDTj1DRFAsQ049UHVibGljJTIwS2V5JTIwU2Vydmlj
ZXMsQ049U2VydmljZXMsQ049Q29uZmlndXJhdGlvbixEQz13aW4yazMsREM9ZG9t
YWluLERDPW5ldGFwcCxEQz1jb20/Y2VydGlmaWNhdGVSZXZvY2F0aW9uTGlzdD9i
YXNlP29iamVjdENsYXNzPWNSTERpc3RyaWJ1dGlvblBvaW50hkJodHRwOi8vd2lu
MmszLWRjLndpbjJrMy5kb21haW4ubmV0YXBwLmNvbS9DZXJ0RW5yb2xsL3dpbjJr
My1kYy5jcmwwEAYJKwYBBAGCNxUBBAMCAQAwDQYJKoZIhvcNAQEFBQADggEBAOJP
jRgm0xlBIpGlbAsEdCzir3PNCBaiM9rcdcmk/NBbACjVoJX8X5uBkqBCfCSNeXSf
EteCFdkLFPxF/tJSuiNcgh5Ae7sSSok7tXRHWmzILKQ1tl63AihMXDdA5kWy7m3b
uMLenv7elPiEMgzvNmN21NEImd1RBJ/Y3O0aboNdMrrCHLTI+FGxtoeH74TwTg3W
ASgnhYjcIqVHlifFlY/l3dB/+hVzf27VzoczXq6X5Sl6v9/03ucsX9t8aVcOeQa+
3FBqyaiqNy3E8JWf/yfyYTZkoohXRjJ9/1FUkoWFxCUl6aVvkcoOoNUTklXwvRDz
wNh+9L85XYXRs6tmIDI=
-----END CERTIFICATE-----
```

3. Set up LDAPS in clustered Data ONTAP.
   a. Using Notepad, open the PEM output file (from step 2), then select and copy all of the text.
   b. Log in to the CLI for clustered Data ONTAP and enter:


```
cluster::> security certificate install –vserver <vServerName> -type server-ca
```

c. Paste in the text that you copied from the PEM file. Be sure to copy all of the text, including the first and last lines.

d. Press Enter twice. The following message appears: “You should keep a copy of the CA-signed digital certificate for future reference.”

e. Enable the TLS feature on the CIFS server.

```
cluster::> vserver cifs security modify –vserver <SVM> -use-start-tls-for-ad-ldap true
```

f. Enable TLS on the ldap client.

```
cluster::> vserver services ldap client create –vserver <SVM> … -use-start-tls true
```

Note: For more information, see section 4.2.16, “LDAP Clients.”

4. To test whether SSL is being used, capture a packet trace while running either of the following commands, depending on which service SSL is enabled for.

SSL for CIFS server LDAP traffic:

```
cluster ::> cifs server domain discovered-servers reset-servers –vserver <SVM>
```

Note: You should notice the storage controller connection to AD via LDAPs. DNS queries can also show up; however, those are not expected to be secure.

SSL for name-mapping or name-server LDAP traffic:

```
cluster::> set diag
cluster::*> diag secd authentication show-creds –node <node> -vserver <SVM> -unix-user-name
<unixUserName>
```

4.2.19 SVM Configuration

After the LDAP configuration is enabled for use, the SVM must be configured to use LDAP in its name switch and name service lookups.

```
cluster::> vserver modify -vserver vs0 -ns-switch file,ldap -nm-switch file,ldap
```

Best Practice

It is a best practice to include file as a secondary ns-switch and nm-switch. Then, even if communication to the LDAP server is broken, local file authentication can still occur.

4.3 Setting Up NFSv4

The following section describes how to set up NFSv4 for use with clustered Data ONTAP. This section covers client and cluster configuration. For more information on NFSv4 implementation in clustered Data ONTAP, see the Clustered Data ONTAP NFS Implementation Guide.

Note: Quick Step Setup steps can be found at the end of this document.

4.3.1 Overview of NFSv4

NFS has been the standard distributed file system for UNIX platforms and has been widely used for over two decades. It operates on a client/server basis and allows users to access files and directories on remote servers over the network. Users employ operating system commands on the client to create, delete, read, write, and set attributes of remote files and directories on the server. It is available on all

flavors and versions of UNIX, Linux, and other well-known operating systems and uses remote procedure calls (RPCs) to remain independent from machine types, operating systems, and network architectures. At a high level, the NFS architecture consists of these components:

- Network protocols
- Kernel extensions
- Client and server daemons

4.3.2 NFSv4 Benefits

Simplicity, reliability, and ease of manageability led to the wide adoption of NFS in the distributed file system landscape. As business complexity grew, customers started demanding stronger authentication, granular access control, multiplatform access, and better I/O performance than existing NFS versions could address. NFS version 4 inherits all essential features of versions 2 and 3 and goes a long way toward addressing the limitations of earlier versions of NFS.

The following improvements were included in NFSv4.

Enhanced built-in security:

- Better namespace handling
- Improved and integrated locking support
- Improved performance over the network
- Cross-platform interoperability, including the Microsoft Windows environment
- Protocol extension to support backward compatibility
- Movement toward an open standard, managed by the IETF

For detailed information on enhancements, refer to RFC3530.

For NFSv4 best practices, see TR-3580.

For NFS implementation in clustered Data ONTAP, see TR-4067.

For condensed NFSv4 setup steps, see the "Quick Step Setup Guides" section in this document.

4.3.3 Configuring the Clustered Data ONTAP System for NFSv4.x

The following section describes how to configure the NFS server for use with NFSv4.x in clustered Data ONTAP. For condensed setup steps, see the "Quick Step Setup Guides" section in this document.

Table 34) Configuring the clustered Data ONTAP system for NFSv4.x (CLI).

1. Log in to the cluster as the admin or vsadmin account.

2. Check that NFS is licensed and enabled.

```
cluster::> license show -description NFS*
cluster::> nfs status –vserver vs0
```

3. Enable NFSv4 and NFSv4.1 (optional).

```
cluster::> nfs modify -vserver vs0 -v4.0 enabled -v4.1 enabled
```

4. Enable the desired NFSv4.x options (described in TR-4067) such as referrals, ACLs, and so on.


```
cluster::> nfs modify -vserver vs0 ?
  [[-access] {true|false}]                        General NFS Access
  [ -v3 {enabled|disabled} ]                      NFS v3
  [ -v4.0 {enabled|disabled} ]                    NFS v4.0
  [ -udp {enabled|disabled} ]                     UDP Protocol
  [ -tcp {enabled|disabled} ]                     TCP Protocol
  [ -spinauth {enabled|disabled} ]                Spin Authentication
  [ -default-win-user <text> ]                    Default Windows User
  [ -v4.0-acl {enabled|disabled} ]                NFSv4.0 ACL Support
  [ -v4.0-read-delegation {enabled|disabled} ]    NFSv4.0 Read Delegation Support
  [ -v4.0-write-delegation {enabled|disabled} ]   NFSv4.0 Write Delegation Support
  [ -v4-id-domain <nis domain> ]                  NFSv4 ID Mapping Domain
  [ -v4.1 {enabled|disabled} ]                    NFSv4.1 Minor Version Support
  [ -rquota {enabled|disabled} ]                  Rquota Enable
  [ -v4.1-pnfs {enabled|disabled} ]               NFSv4.1 Parallel NFS Support
  [ -v4.1-acl {enabled|disabled} ]                NFSv4.1 ACL Support
  [ -vstorage {enabled|disabled} ]                NFS vStorage Support
  [ -default-win-group <text> ]                   Default Windows Group
  [ -v4.1-read-delegation {enabled|disabled} ]    NFSv4.1 Read Delegation Support
  [ -v4.1-write-delegation {enabled|disabled} ]   NFSv4.1 Write Delegation Support
  [ -mount-rootonly {enabled|disabled} ]          NFS Mount Root Only
  [ -nfs-rootonly {enabled|disabled} ]            NFS Root Only
```

5. Set the NFSv4.0 ID domain. This example assumes that LDAP was already installed and configured on the Windows Active Directory server. With Windows AD, the DNS domain name will be the NFSv4 ID domain. This also assumes that LDAP queries are working properly with the cluster.

```
cluster::> nfs modify -vserver vs0 -v4-id-domain domain.netapp.com
```

4.3.4 Configuring the Domain Controller for NFSv4.x

The domain controller configuration steps are covered in the section under LDAP: Configuring the domain controller as an LDAP server. These steps need to be completed before NFSv4 is set up for use with Windows Active Directory implementations, since this step creates the necessary attributes to gather UID and GID information from the server. It is also possible to use a separate server for NFSv4 domain IDs, but these should be in sync with the AD LDAP server.

4.3.5 Configuring the NFS Clients for NFSv4.x

The following section describes how to configure the NFS clients for use with NFSv4.x in clustered Data ONTAP. To configure the Linux client, simply modify `the /etc/idmapd.conf` file (all Linux clients covered in this document except Solaris, which uses `/etc/default/nfs`) to include the NFSv4 domain to use with NFS.

`/etc/idmapd.conf` sample:

```
sles11:~ # cat /etc/idmapd.conf
[General]
Verbosity = 0
Pipefs-Directory = /var/lib/nfs/rpc_pipefs
Domain = domain.netapp.com

[Mapping]

Nobody-User = nobody
Nobody-Group = nobody
```

`/etc/default/nfs` sample (mapID section only):

```
NFSMAPID_DOMAIN=domain.netapp.com
```


In addition, verify that LDAP lookups are working properly on the client and that the client is mounting via NFSv4 either from the mount option `-t nfs4` or from the client configuration.

Whenever you make a change to the `idmapd.conf` file, the necessary services need to be restarted. The table below covers which services are restarted on each client.

Table 35) Services for NFSv4.

OS	Commands
RHEL/CentOS/Fedora	`service rpcidmapd [start|stop|restart|status]`
SLES/SUSE	`service nfs [start|stop|restart|status]`
Ubuntu	`service idmapd [start|stop|restart|status]`
Solaris	`svcadm [enable|disable|restart|refresh] mapid` `svcs -l mapid (to list)`

For complete setup steps, refer to the client documentation.

Solaris NFSv4

CentOS NFSv4

RHEL NFSv4

SUSE/SLES NFSv4

Ubuntu NFSv4

5 Quick Step Setup Guides

Following are lists of condensed steps to set up Kerberos and LDAP for use with clustered Data ONTAP. These cover only the commands that should be run to complete the task. This section is intended for audiences that already understand the nuances of this solution and audiences that need to get the solution working quickly.

Sample shell scripts are included to make the process simpler. The Appendix includes steps on configuring noninteractive SSH on the cluster (passwordless) to make scripting easier.

NetApp highly recommends that you review the entire document for enterprise production solutions to fully understand the hows and whys of this setup.

Within the Quick Step Setup Guides there are references to previous portions of the document, denoted by a linked [?]. Simply click on the link to be redirected to the section in question.

5.1 Quick Step Setup

Cluster Configuration

The following commands can be copied and pasted into a script and modified to fit the environment. These assume an SVM and data LIFs that allow NFS and can route to the clients and that KDC is already present.

Kerberos [?]

1. DNS

g. Create DNS.

```
dns create -vserver [vserver] -domains [domain1,domain2..] -name-servers [ns1,ns2,..] -state
enabled
```

h. Modify DNS.

```
dns modify -vserver [vserver] -domains [domain1,domain2..] -name-servers [ns1,ns2,..] -state
enabled
```

2. NFS

a. Create NFS server (or modify existing).

```
nfs create -vserver [vserver] -access true -v3 [enabled|disabled] -v4.0 [enabled|disabled] -v4.1
[enabled|disabled]
```

b. Modify NFS server.

```
nfs modify -vserver [vserver] -access true -v3 [enabled|disabled] -v4.0 [enabled|disabled] -v4.1
[enabled|disabled]
```

c. Create UNIX users and groups (optional if already created).

```
unix-user create -vserver [vserver] -user root -id 0 -primary-gid 0
unix-user create -vserver [vserver] -user pcuser -id 65534 -primary-gid 65534
unix-user create -vserver [vserver] -user nobody -id 65535 -primary-gid 65535
unix-group create -vserver [vserver] -name root -id 0
unix-group create -vserver [vserver] -name pcuser -id 65534
unix-group create -vserver [vserver] -name nobody -id 65535
```

3. Create Kerberos realm.

```
kerberos-realm create -configname [realmname] -realm [REALM.ALL.CAPS] -kdc-vendor Microsoft -kdc-
ip [10.10.10.10] -kdc-port 88 -clock-skew 5 -adminserver-ip [10.10.10.10] -adminserver-port 749 -
passwordserver-ip [10.10.10.10] -passwordserver-port 464 -adserver-name [KDCname] -adserver-ip
[10.10.10.10]
```

4. Create CIFS server (optional).

```
vserver setup
```

5. Create export policy and rules (optional if already present).

```
export-policy create -vserver parisi -policyname Kerberos

export-policy rule create -vserver [vserver] -policyname [Kerberos] -clientmatch [0.0.0.0/0] -
rorule [sys,krb5] -rwrule [sys,krb5] -anon [65534] -superuser [any|krb5|sys|none] -ruleindex 1 -
protocol [any|nfs|nfs3|nfs4|cifs]
```

6. Create data volumes (optional if already present).

```
volume create -vserver [vserver] -volume [kerberosvol] -aggregate [aggrname] -size 1g -junction-
path [/kerberosvolpath] -security-style [unix|ntfs] -unix-permissions [755] -user [root] -group
[root] -policy [policyname]
```

7. Enable Kerberos on the data LIF(s).


```
kerberos-config modify -vserver [vserver] -lif [lif_name] -kerberos enabled -spn
[nfs/fqdn.netapp.com@REALM.NETAPP.COM]
```

8. Create name mapping or unix-user (only one or the other is needed).
   a. Create unix-user.

```
unix-user create -vserver [vserver] -user nfs -id [505] -primary-gid 65534
```

   b. Create name mapping.

```
vserver name-mapping create -vserver [vserver] -direction krb-unix -position 1 -pattern
[nfs/fqdn.netapp.com@REALM.NETAPP.COM] -replacement [validusername]
```

LDAP [?]

9. Create LDAP client.

```
set advanced; ldap client create -client-config [config_name] -servers [server IP] -schema [AD-
SFU|AD-IDMU|RFC-2307] -port 389 -query-timeout 3 -min-bind-level [anonymous|simple|sasl] -base-dn
[dc=domain,dc=netapp,dc=com] -base-scope subtree -user-scope subtree -group-scope subtree -
netgroup-scope subtree -bind-dn [username] -user-dn [cn=users,DC=domain,DC=netapp,DC=com] -group-
dn [cn=users,DC=domain,DC=netapp,DC=com] -vserver [vserver]
```

10. LDAP configuration.

```
ldap create -vserver [vserver] -client-config [config_name] -client-enabled true
```

11. SVM configuration.

```
vserver modify -vserver [vserver] -ns-switch file,ldap -nm-switch file,ldap
```

NFSv4.x [?]

12. NFSv4.x configuration.

```
nfs modify -vserver [vserver] -v4.0 enabled -v4.1 [enabled|disabled] -v4-id-domain
[domain.netapp.com]
```

Sample Shell Script—Cluster Configuration

- The following shell script can be run from any client that supports shell scripts via SSH key–based login.
- The script does not include CIFS setup.
- This script works for clustered Data ONTAP 8.1 and 8.2.
- The script requires some interaction (such as user name/password for the user to create the machine account).
- Replace the entries in <brackets> with the necessary information and save as Kerberos_setup.sh.
- The script can be modified to include commands to modify rather than create DNS/NFS and so on by commenting/uncommenting the line.
- This script is not supported by NetApp and does not cover every use case.

```
#!/bin/bash
# Linux/UNIX box with ssh key based login enabled
############################
# Define script variables #
############################
cluster="10.61.92.30"
# SSH User name
USR="ssh"
vserver="{vserver}"
```



```
#Kerberos
realm="{REALM.NETAPP.COM}"
fqdn="{host.domain.netapp.com}"
dns="{domain.netapp.com}"
domain="{domain.netapp.com}"
nameservers="{ns1,ns2..}"
v3enable="{enabled|disabled}"
v4enable="{enabled|disabled}"
v41enable="{enabled|disabled}"
realmconfigname="{Realm config name}"
kdcip="{KDC IP}"
adminserver="{Admin server IP}"
passwdserver="{Passwd server IP}"
kdcname="{DC Name}"
lif="{lif name}"
spn="{nfs/fqdn@REALM.NETAPP.COM}"
#Export policy
policyname="{Kerberos policy}"
clientmatch="{IP|host|subnet|netgroup}"
rorule="{sys,krb5|all|none}"
rwrule="{sys,krb5|all|none}"
anonid="{UID}"
superuser="{any|none|never|sys|krb5}"
protocol="{any|cifs|nfs|nfs3|nfs4}"
#name-mapping rule/unix user create
nfsid="{nfs UID}"
username="{Username to replace SPN}"
#LDAP config
ldapconfigname="{LDAP config name}"
ldapaddress="{LDAP1,LDAP2..}"
schema="{AD-IDMU|AD-SFU|AD-SFU-Deprecated|RFC-2307|custom schema}"
bindlevel="{anonymous|sasl|simple}"
basedn="{dc=domain,dc=netapp,dc=com}"
userdn="{dc=domain,dc=netapp,dc=com}"
groupdn="{dc=domain,dc=netapp,dc=com}"
binddn="{bind username}"
basescope="{base|onelevel|subtree}"
userscope="{base|onelevel|subtree}"
groupscope="{base|onelevel|subtree}"
netgroupscope="{base|onelevel|subtree}"
#CIFS config (optional)
cifserver="{CIFS server name}"
########################################
# DNS - remove comment (#) to include line
########################################
#ssh $USR@$cluster dns create -vserver $vserver -domains $dns -name-servers $nameservers -state
enabled
#ssh $USR@$cluster dns modify -vserver $vserver -domains $dns -name-servers $nameservers -state
enabled
########################################
# NFS - remove comment (#) to include line
########################################
#ssh $USR@$cluster nfs create -vserver $vserver -access true -v3 $v3enable -v4.0 $v4enable -v4.1
$enable
#ssh $USR@$cluster nfs modify -vserver $vserver -access true -v3 $v3enable -v4.0 $v4enable -v4.1
$v41enable
########################################
# Default Unix Users - remove comment (#) to include line
########################################
#ssh $USR@$cluster unix-user create -vserver $vserver -user root -id 0 -primary-gid 0
#ssh $USR@$cluster unix-user create -vserver $vserver -user pcuser -id 65534 -primary-gid 65534
#ssh $USR@$cluster unix-user create -vserver $vserver -user nobody -id 65535 -primary-gid 65535
#ssh $USR@$cluster unix-group create -vserver $vserver -name root -id 0
#ssh $USR@$cluster unix-group create -vserver $vserver -name pcuser -id 65534
#ssh $USR@$cluster unix-group create -vserver $vserver -name nobody -id 65535
########################################
# Kerberos - remove comment (#) to include line
########################################
#ssh $USR@$cluster kerberos-realm create -configname $realmconfigname -realm $realm -kdc-vendor
Microsoft -kdc-ip $kdcip -kdc-port 88 -clock-skew 5 -adminserver-ip $adminserver -adminserver-
```



```
port 749 -passwordserver-ip $passwdserver -passwordserver-port 464 -adserver-name $kdcname -
adserver-ip $kdcip
########################################
# Export Policy - remove comment (#) to include line
########################################
#ssh $USR@$cluster export-policy create -vserver $vserver -policyname $policyname
#ssh $USR@$cluster export-policy rule create -vserver $vserver -policyname $policyname -
clientmatch $clientmatch -rorule $rorule -rwrule $rwrule -anon $anonid -superuser $superuser -
ruleindex 1 -protocol $protocol
########################################
# Enable Kerberos - remove comment (#) to include line
########################################
#ssh $USR@$cluster kerberos-config modify -vserver $vserver -lif $lif -kerberos enabled -spn $spn
################################################################################
# Create Unix User or Name mapping rule - remove comment (#) to include line
################################################################################
#ssh $USR@$cluster unix-user create -vserver $vserver -user nfs -id $nfsid -primary-gid 65534
#ssh $USR@$cluster vserver name-mapping create -vserver $vserver -direction krb-unix -position 1
-pattern $spn -replacement $username
################################################################################
# Create LDAP client - remove comment (#) to include line
################################################################################
#ssh $USR@$cluster "set advanced; ldap client create -client-config $ldapconfigname -servers
$ldapaddress -schema $schema -port 389 -query-timeout 3 -min-bind-level $bindlevel -base-dn
$basedn -base-scope $basescope -user-scope $userscope -group-scope $groupscope -netgroup-scope
$netgroupscope -bind-dn $binddn -user-dn $userdn -group-dn $groupdn -vserver $vserver"
################################################################################
#Create CIFS server (optional) - remove comment (#) to include line
################################################################################
#ssh $USR@$cluster cifs server create -vserver $vserver -cifs-server $cifserver -domain $domain -
ou CN=Computers
################################################################################
# Modify LDAP client if using CIFS as well - remove comment (#) to include line
################################################################################
#ssh $USR@$cluster "set advanced; ldap client modify -client-config $ldapconfigname -vserver
$vserver -bind-as-cifs-server true -ad-domain $domain -preferred-ad-servers $ldapaddress"
################################################################################
# Create LDAP config - remove comment (#) to include line
################################################################################
#ssh $USR@$cluster ldap create -vserver $vserver -client-config $ldapconfigname -client-enabled
true
################################################################################
# Modify SVM to use LDAP - remove comment (#) to include line
################################################################################
#ssh $USR@$cluster vserver modify -vserver $vserver -ns-switch file,ldap -nm-switch file,ldap
################################################################################
# NFSv4 Config - remove comment (#) to include line
################################################################################
#ssh $USR@$cluster nfs modify -vserver $vserver -v4.0 $v4enable -v4.1 $v41enable -v4-id-domain
$domain
```

Windows Server Configuration

The following commands can be copied and pasted into a script and modified to fit the environment. These assume that the Active Directory Schema has already been extended and the necessary schema attributes for the users and computers already exist. The following were run on a Windows 2008 R2 domain controller but can be modified for use with other versions provided the version supports Windows PowerShell.

Use the following commands with caution because they are not supported by NetApp. Always test in a nonproduction environment before using them.

Kerberos [?]

These steps assume that DES was enabled on the domain controller via Group Policy.

1. Create the NFS server/client and the SRV DNS records for Kerberos-master.

a. Create NFS server and client records.

```
dnscmd /RecordAdd <domain.netapp.com> <host> /CreatePTR A {IP}
```

b. Create SRV record for Kerberos-master (if necessary).

```
dnscmd /RecordAdd <domain.netapp.com> _kerberos-master._tcp SRV 0 100 88
<win2k8DC.domain.netapp.com>
dnscmd /RecordAdd <domain.netapp.com> _kerberos-master._udp SRV 0 100 88
<win2k8DC.domain.netapp.com>
```

2. Allow DES (per machine account only; only the PowerShell step is mentioned here) for NFS server and NFS clients

Note: Import-module only needs to be run once.

a. Import the module for Active Directory (PowerShell).

```
import-module activedirectory
```

b. New machine account (PowerShell).

```
New-ADComputer -Name <computername> -SAMAccountName <computername> -DNSHostName
<computername.dns.domain.com> -OtherAttributes @{'userAccountControl'=2097152;'msDS-
SupportedEncryptionTypes'=25}
```

c. Modify the existing machine account for the NFS server (PowerShell).

```
Set-ADComputer -Identity <NFS server name> -Replace @{'userAccountControl'=2097152;'msDS-
SupportedEncryptionTypes'=25}
```

d. Modify the existing machine account for the NFS client (PowerShell).

```
Set-ADComputer -Identity <NFS server name> -Replace @{'msDS-SupportedEncryptionTypes'=25}
```

3. Create the keytab file (using cmd).

```
ktpass -princ [primary/instance@REALM] -mapuser [DOMAIN\machine$] -crypto ALL +rndpass -ptype
KRB5_NT_PRINCIPAL +Answer -out [file:\location]
```

a. Copy the keytab file to the NFS client.

Sample Batch Script—Windows

- Copy, paste, and modify the following into a file on a Windows KDC.
- Replace the entries in <brackets> with the necessary information. The first section defines variables.
- Save the file as “filename.bat.”
- Run the file from cmd.
- The file is not supported by NetApp and does not cover every use case.

```
PowerShell.exe -noninteractive -command "set-variable -name strhostnamelong -value
"<fqdn.domain.netapp.com>";  set-variable -name strhostnameshort -value "<shortname> "; set-
variable -name strdc -value "<dc name>"; $strfqdn; $strhostnameshort; $strdc; $strhostnamelong;
import-module activedirectory; New-ADComputer -Name $strhostnameshort -SAMAccountName
$strhostnameshort -DNSHostName $strhostnamelong -OtherAttributes
@{'userAccountControl'=2097152;'msDS-SupportedEncryptionTypes'=25}; exit"
ktpass -princ <root/host.domain.netapp.com@DOMAIN.NETAPP.COM> -mapuser <DOMAIN\host$> -crypto ALL
+rndpass -ptype KRB5_NT_PRINCIPAL +Answer -out <host>.keytab
dnscmd /RecordAdd <domain.netapp.com> {host} /CreatePTR A {IP}
dnscmd /RecordAdd <domain.netapp.com> _kerberos-master._tcp SRV 0 100 88
<win2k8DC.domain.netapp.com>
```



```
dnscmd /RecordAdd <domain.netapp.com> _kerberos-master._udp SRV 0 100 88
<win2k8DC.domain.netapp.com>
dnscmd /RecordAdd <domain.netapp.com> @ /CreatePTR A <DC1 IP>
dnscmd /RecordAdd <domain.netapp.com> @ /CreatePTR A <DC2 IP>
```

RHEL/CentOS/Fedora Configuration

The following covers RHEL/CentOS/Fedora client configuration. The commands below can be copied, pasted, and then modified to create a shell script for easy configuration. A sample shell script is included in this section. This section assumes that the KDC and the LDAP server have already been configured properly.

Network Configuration

1. Set the host name on the client in the /etc/sysconfig/network file and with the host name command.

Sample file:

```
NETWORKING=yes
HOSTNAME=centos.domain.netapp.com
```

Host name command:

```
hostname centos.domain.netapp.com
```

2. Update resolv.conf with DNS information (use only if DHCP is not used or if DHCP is not providing the correct DNS domain).

Sample file:

```
search domain.netapp.com
nameserver 10.63.98.101
```

   a. Prevent /etc/resolv.conf from being overwritten by networking scripts (optional).

```
chattr –i /etc/resolv.conf
```

Install Packages

3. Install/update Kerberos packages.

```
yum install krb5-workstation –y
yum install pam_krb5 –y
yum install krb5-appl-clients –y
```

4. Install/update the NFSv4 package.

```
yum install nfs-utils –y
```

5. Install/update SSSD.

```
yum install sssd –y
```

6. Remove nscd to avoid conflicts with SSSD (optional).

```
yum remove nscd –y
```


Kerberos [?]

7. Allow MD5 (RHEL 6.4 only).

```
echo NSS_HASH_ALG_SUPPORT=+MD5 > /etc/environment
```

8. Allow secure NFS.

```
sed -i 's/#SECURE_NFS/SECURE_NFS/g' /etc/sysconfig/nfs
```

9. Configure /etc/krb5.conf.

Sample file:

```
[libdefaults]
default_realm = DOMAIN.NETAPP.COM
dns_lookup_realm = true
dns_lookup_kdc = true
allow_weak_crypto = true

[realms]
DOMAIN.NETAPP.COM = {
kdc = domain.netapp.com:88
default_domain = domain.netapp.com
}

[logging]
kdc = FILE:/var/log/krb5kdc.log
admin_server = FILE:/var/log/kadmin.log
default = FILE:/var/log/krb5lib.log

[domain_realm]
.netapp.com = DOMAIN.NETAPP.COM
.domain.netapp.com = DOMAIN.NETAPP.COM
```

10. Create /etc/krb5.keytab (the keytab file must be copied from KDC first).

```
ktutil <<EOF
rkt /{hostname}.keytab
wkt /etc/krb5.keytab
list
exit
EOF
```

11. Restart Kerberos services.

```
service rpcgssd restart
```

NFSv4 [?]

12. Update the NFSv4 domain (if it is not already configured).

Sample file:

```
[General]
Domain = domain.domain.netapp.com
[Mapping]
```



```
Nobody-User = nobody
Nobody-Group = nobody
[Translation]
Method = nsswitch
```

13. Restart the NFSv4 daemon.

```
service rpcidmapd restart
```

SSSD [?]

14. Configure /etc/sssd/sssd.conf.

Sample file:

```
[domain/default]
cache_credentials = True
case_sensitive = False
[sssd]
config_file_version = 2
services = nss, pam
domains = DOMAIN
debug_level = 7
[nss]
filter_users = root,ldap,named,avahi,haldaemon,dbus,radiusd,news,nscd
filter_groups = root
[pam]
[domain/YOURDOMAINNAME]
id_provider = ldap
auth_provider = krb5
chpass_provider = krb5
case_sensitive = true
cache_credentials = false
ldap_search_base = dc=domain,dc=netapp,dc=com
ldap_schema = rfc2307
ldap_sasl_mech = GSSAPI
ldap_user_object_class = user
ldap_group_object_class = group
ldap_user_home_directory = unixHomeDirectory
ldap_user_principal = userPrincipalName
ldap_account_expire_policy = ad
ldap_force_upper_case_realm = true
ldap_group_search_base = cn=Users,dc=domain,dc=netapp,dc=com
ldap_user_search_base = cn=Users,dc=domain,dc=netapp,dc=com
ldap_sasl_authid = root/centos.domain.netapp.com@DOMAIN.NETAPP.COM
krb5_realm = DOMAIN.NETAPP.COM
#krb5_canonicalize = false – RHEL 6.3 only
```

a. Verify that the file has 0600 permissions and root:root owner.

```
chmod 0600 /etc/sssd/sssd.conf
chown root:root /etc/sssd/sssd.conf
```

15. Enable SSSD and SSSD auth.

```
authconfig --enablesssd --enablesssdauth –updateall
```

16. Restart the SSSD service.

```
service sssd restart
```


Sample Shell Script—RHEL/CentOS/Fedora Config

- Copy, paste, and modify the following into a file on an NFS client to be used with Kerberos.
- Replace the variables at the beginning of the script with the necessary values.
- Uncomment out the entries intended for use.
- The file is not supported by NetApp and does not cover every use case.
- Split the script into sections and run each section separately so that you can troubleshoot issues more easily.

```
##################################
####  Define variables below!  ####
##################################
#!/bin/bash
# Linux/UNIX box with ssh key based login enabled
linuxhost={hostname}
dnsIP1={IP}
###Add Additional DNS servers if desired
#dnsIP2={IP}
#dnsIP3={IP}
fqdn={domain.netapp.com}
domain={DOMAIN}
realm={DOMAIN.NETAPP.COM}
defaultdomain={netapp.com}
userdn={cn=Users,dc=domain,dc=netapp,dc=com}
basedn={dc=domain,dc=netapp,dc=com}
##################################
####  Define variables above!  ####
##################################
## Script backs files up to ensure config can be reverted easily
######################
### Network config ###
######################
### Modify the network config to include the hostname
## NOTE: Review the contents of the network file before modifying
#mv /etc/sysconfig/network /etc/sysconfig/network-original
#echo NETWORKING=yes > /etc/sysconfig/network; echo HOSTNAME=$linuxhost.$fqdn >>
/etc/sysconfig/network
#hostname $linuxhost.$fqdn
#echo ########################
#echo Hostname configured!
#echo ########################
#cat /etc/sysconfig/network
### Configure DNS
#mv /etc/resolv.conf /etc/resolv.conf-old
#echo search $fqdn > /etc/resolv.conf; echo nameserver $dnsIP1 >> /etc/resolv.conf
###Add additional DNS servers if desired
#echo nameserver $dnsIP2 >> /etc/resolv.conf
#echo nameserver $dnsIP3 >> /etc/resolv.conf
#echo ########################
#echo DNS is configured!
#echo ########################
#cat /etc/resolv.conf
#nslookup $linuxhost
### modify /etc/resolv.conf to prevent overwrite
#chattr -i /etc/resolv.conf
######################
###  Install pkgs  ###
######################
### Install/Update Kerberos packages
#yum install krb5-workstation -y
#yum install pam_krb5 -y
#yum install krb5-appl-clients -y
#### Install/update NFSv4
#yum install nfs-utils -y
#### Install/update SSSD
#yum install sssd -y
# yum remove nscd
######################
```



```
## Kerberos Config ###
######################
#### Allow MD5 in RHEL 6.4
#echo NSS_HASH_ALG_SUPPORT=+MD5 > /etc/environment
## Allow secure NFS
#sed -i 's/#SECURE_NFS/SECURE_NFS/g' /etc/sysconfig/nfs
#echo ########################
# echo Secure NFS configured!
#echo ########################
#cat /etc/sysconfig/nfs | grep SECURE_NFS
###Configure /etc/krb5.conf
#mv /etc/krb5.conf /etc/krb5.default
#echo [libdefaults]> /etc/krb5.conf
#echo  default_realm = $realm>> /etc/krb5.conf
#echo  dns_lookup_realm = true>> /etc/krb5.conf
#echo  dns_lookup_kdc = true>> /etc/krb5.conf
#echo  allow_weak_crypto = true>> /etc/krb5.conf
#echo  >> /etc/krb5.conf
#echo [realms]>> /etc/krb5.conf
#echo        $realm = {>> /etc/krb5.conf
#echo   kdc = $fqdn:88>> /etc/krb5.conf
#echo            default_domain = $fqdn>> /etc/krb5.conf
#echo        }>> /etc/krb5.conf
#echo  >> /etc/krb5.conf
#echo [logging]>> /etc/krb5.conf
#echo        kdc = FILE:/var/log/krb5kdc.log>> /etc/krb5.conf
#echo        admin_server = FILE:/var/log/kadmin.log>> /etc/krb5.conf
#echo        default = FILE:/var/log/krb5lib.log>> /etc/krb5.conf
#echo  >> /etc/krb5.conf
#echo [domain_realm]>> /etc/krb5.conf
#echo        .$defaultdomain = $realm>> /etc/krb5.conf
#echo        .$fqdn = $realm>> /etc/krb5.conf
#echo ########################
#echo Kerberos file is configured!
#echo ########################
#cat /etc/krb5.conf
#### Create Keytab file
### Keytab file must be moved from KDC before this step
#ktutil <<EOF
#rkt /$linuxhost.keytab
#wkt /etc/krb5.keytab
#list
#exit
#EOF
#### Restart Kerberos service
#service rpcgssd restart
########################
##### NFSv4 Config #####
########################
#### Configure /etc/idmpad.conf (if not already configured)
#mv /etc/idmapd.conf /etc/idmapd.default; echo [General] > /etc/idmapd.conf; echo Domain = $fqdn
>> /etc/idmapd.conf; echo [Mapping] >> /etc/idmapd.conf; echo Nobody-User = nobody >>
/etc/idmapd.conf; echo Nobody-Group = nobody >> /etc/idmapd.conf; echo [Translation] >>
/etc/idmapd.conf; echo Method = nsswitch >> /etc/idmapd.conf; chmod 0600 /etc/idmapd.conf; chown
root:root /etc/idmapd.conf
#echo ########################
#echo NFSv4 domain configured!
#echo ########################
#cat /etc/idmapd.conf
#### Restart idmapd
#service rpcidmapd restart
########################
##### SSSD Config #####
########################
#### Configure the /etc/sssd/sssd.conf file
#mv /etc/sssd/sssd.conf /etc/sssd/sssd.default
#echo [domain/default] > /etc/sssd/sssd.conf
#echo cache_credentials = True >> /etc/sssd/sssd.conf
#echo case_sensitive = False >> /etc/sssd/sssd.conf
#echo [sssd] >> /etc/sssd/sssd.conf
#echo config_file_version = 2 >> /etc/sssd/sssd.conf
```



```
#echo services = nss, pam >> /etc/sssd/sssd.conf
#echo domains = $domain >> /etc/sssd/sssd.conf
#echo debug_level = 7 >> /etc/sssd/sssd.conf
#echo [nss] >> /etc/sssd/sssd.conf
#echo filter_users = root,ldap,named,avahi,haldaemon,dbus,radiusd,news,nscd >>
/etc/sssd/sssd.conf
#echo filter_groups = root >> /etc/sssd/sssd.conf
#echo [pam] >> /etc/sssd/sssd.conf
#echo [domain/$domain] >> /etc/sssd/sssd.conf
#echo id_provider = ldap >> /etc/sssd/sssd.conf
#echo auth_provider = krb5 >> /etc/sssd/sssd.conf
#echo case_sensitive = false >> /etc/sssd/sssd.conf
#echo chpass_provider = krb5 >> /etc/sssd/sssd.conf
#echo cache_credentials = false >> /etc/sssd/sssd.conf
### Use ldap_uri only if there is a single DC
#echo ldap_uri = _srv_,ldap://$fqdn >> /etc/sssd/sssd.conf
#echo ldap_search_base = $basedn >> /etc/sssd/sssd.conf
#echo ldap_schema = rfc2307 >> /etc/sssd/sssd.conf
#echo ldap_sasl_mech = GSSAPI >> /etc/sssd/sssd.conf
#echo ldap_user_object_class = user >> /etc/sssd/sssd.conf
#echo ldap_group_object_class = group >> /etc/sssd/sssd.conf
#echo ldap_user_home_directory = unixHomeDirectory >> /etc/sssd/sssd.conf
#echo ldap_user_principal = userPrincipalName >> /etc/sssd/sssd.conf
#echo ldap_group_member = memberUid >> /etc/sssd/sssd.conf
#echo ldap_group_name = cn >> /etc/sssd/sssd.conf
#echo ldap_account_expire_policy = ad >> /etc/sssd/sssd.conf
#echo ldap_force_upper_case_realm = true >> /etc/sssd/sssd.conf
#echo ldap_group_search_base = $userdn >> /etc/sssd/sssd.conf
#echo ldap_sasl_authid = root/$linuxhost.$fqdn@$realm >> /etc/sssd/sssd.conf
### Use krb5_server and krb5_kpasswd only if there is a single DC
#echo krb5_server = $fqdn >> /etc/sssd/sssd.conf
#echo krb5_realm = $realm >> /etc/sssd/sssd.conf
#echo krb5_kpasswd = $fqdn >> /etc/sssd/sssd.conf
#echo ########################
#echo SSSD conf file created!
#echo ########################
#cat /etc/sssd/sssd.conf
#### Ensure /etc/sssd/sssd.conf is 0600 perms
#chmod 0600 /etc/sssd/sssd.conf
#chown root:root /etc/sssd/sssd.conf
#### Enable SSSD and SSSD auth
#authconfig --enablesssd --enablesssdauth --updateall
#### Restart SSSD
#service sssd restart
```

SLES/SUSE Configuration

Network Config

1. Set the host name on the client in the /etc/HOSTNAME and/or /etc/sysconfig/network files and with the `hostname` command.

Sample /etc/sysconfig/network file:

```
NETWORKING=yes
HOSTNAME=centos.domain.netapp.com
```

Sample HOSTNAME file:

```
suse.domain.netapp.com
```

`Hostname` command:

```
hostname suse.domain.netapp.com
```


2. Update resolv.conf with DNS information (use only if DHCP is not used or if DHCP is not providing the correct DNS domain).

Sample file:

```
search domain.netapp.com
nameserver 10.63.98.101
```

a. Prevent /etc/resolv.conf from being overwritten by networking scripts (optional).

```
chattr –i /etc/resolv.conf
```

Install Packages

3. Install/update Kerberos packages.

```
zypper --non-interactive install krb5
zypper --non-interactive install krb5-client
zypper --non-interactive install krb5-32bit
zypper --non-interactive install krb5-apps-clients
zypper --non-interactive install pam_krb5
zypper --non-interactive install pam_krb5-32bit
```

1. Install/update the NFSv4 package.

```
zypper --non-interactive install nfs-client
```

4. Install/update SSSD.

```
zypper --non-interactive install sssd
zypper --non-interactive install sssd-tools
```

5. Remove nscd to avoid conflicts with SSSD (optional).

```
zypper --non-interactive remove nscd
```

Kerberos [?]

6. Allow secure NFS.

```
sed -i 's/NFS_SECURITY_GSS="no"/NFS_SECURITY_GSS="yes"/g' /etc/sysconfig/nfs
```

7. Configure /etc/krb5.conf.

Sample file:

```
[libdefaults]
default_realm = DOMAIN.NETAPP.COM
dns_lookup_realm = true
dns_lookup_kdc = true
allow_weak_crypto = true

[realms]
DOMAIN.NETAPP.COM = {
kdc = domain.netapp.com:88
```



```
default_domain = domain.netapp.com
}

[logging]
kdc = FILE:/var/log/krb5kdc.log
admin_server = FILE:/var/log/kadmin.log
default = FILE:/var/log/krb5lib.log

[domain_realm]
.netapp.com = DOMAIN.NETAPP.COM
.domain.netapp.com = DOMAIN.NETAPP.COM
```

8. Create /etc/krb5.keytab (the keytab file must be copied from the KDC first).

```
ktutil <<EOF
rkt [/hostname.keytab]
wkt /etc/krb5.keytab
list
exit
EOF
```

9. Configure NFS services (SUSE only).

```
systemctl enable rpcbind.service
systemctl enable nfs.service
service rpcbind start
service nfs start
```

NFSv4 [?]

10. Update the NFSv4 domain (if it is not already configured).

```
mv /etc/idmapd.conf /etc/idmapd.default; echo Domain = [nfsv4domain.netapp.com] >
/etc/idmapd.conf; echo Nobody-User = nobody >> /etc/idmapd.conf; echo Nobody-Group = nobody >>
/etc/idmapd.conf; chmod 0600 /etc/idmapd.conf; chown root:root /etc/idmapd.conf
```

Sample file:

```
[General]
Domain = domain.domain.netapp.com
[Mapping]
Nobody-User = nobody
Nobody-Group = nobody
[Translation]
Method = nsswitch
```

11. Restart the NFSv4 daemon.

```
service rpcidmapd restart
```

SSSD [?]

12. Configure /etc/sssd/sssd.conf.

Sample file:

```
[domain/default]
cache_credentials = True
case_sensitive = False
[sssd]
config_file_version = 2
```



```
services = nss, pam
domains = DOMAIN
debug_level = 7
[nss]
filter_users = root,ldap,named,avahi,haldaemon,dbus,radiusd,news,nscd
filter_groups = root
[pam]
[domain/YOURDOMAINNAME]
id_provider = ldap
auth_provider = krb5
chpass_provider = krb5
case_sensitive = true
cache_credentials = false
ldap_search_base = dc=domain,dc=netapp,dc=com
ldap_schema = rfc2307
ldap_sasl_mech = GSSAPI
ldap_user_object_class = user
ldap_group_object_class = group
ldap_user_home_directory = unixHomeDirectory
ldap_user_principal = userPrincipalName
ldap_account_expire_policy = ad
ldap_force_upper_case_realm = true
ldap_user_search_base = cn=Users,dc=domain,dc=netapp,dc=com
ldap_group_search_base = cn=Users,dc=domain,dc=netapp,dc=com
ldap_sasl_authid = root/sles.domain.netapp.com@DOMAIN.NETAPP.COM
krb5_realm = DOMAIN.NETAPP.COM
```

a. Verify that the file has 0600 permissions and root:root owner.

```
chmod 0600 /etc/sssd/sssd.conf
chown root:root /etc/sssd/sssd.conf
```

13. Configure PAM.

```
pam-config --add --sss
pam-config --add --krb5
sed -i 's/required/sufficient/g' /etc/pam.d/common-auth
sed -i 's/required/sufficient/g' /etc/pam.d/common-account
sed -i 's/required/sufficient/g' /etc/pam.d/common-auth-pc
sed -i 's/required/sufficient/g' /etc/pam.d/common-account-pc
```

14. Enable SSSD to start at boot (SUSE only).

```
systemctl enable sssd.service
```

15. Restart the SSSD service.

```
service sssd restart
```

Sample Shell Script—SUSE/SLES

- Copy, paste, and modify the following into a file on an NFS client to be used with Kerberos.
- Replace the variables at the beginning of the script with the necessary values.
- Uncomment out the entries intended for use.
- The file is not supported by NetApp and does not cover every use case.
- Split the script into sections and run each section separately to troubleshoot issues more easily.

```
#################################
####  Define variables below!  ####
#################################
#!/bin/bash
# Linux/UNIX box with ssh key based login enabled
```



```
linuxhost={hostname}
dnsIP1={IP}
###Add Additional DNS servers if desired
#dnsIP2={IP}
#dnsIP3={IP}
fqdn={domain.netapp.com}
domain={DOMAIN}
realm={DOMAIN.NETAPP.COM}
defaultdomain={netapp.com}
userdn={cn=Users,dc=domain,dc=netapp,dc=com}
basedn={dc=domain,dc=netapp,dc=com}
###################################
####  Define variables above!  ####
###################################
## Script backs files up to ensure config can be reverted easily
######################
### Network config ###
######################
### Modify the network config to include the hostname
## NOTE: Review the contents of the network file before modifying
#mv /etc/sysconfig/network /etc/sysconfig/network-original
#echo NETWORKING=yes > /etc/sysconfig/network; echo HOSTNAME=$linuxhost.$fqdn >> /etc/HOSTNAME
#hostname $linuxhost.$fqdn
#echo ########################
#echo Hostname configured!
#echo ########################
#cat /etc/sysconfig/network
### Configure DNS
#mv /etc/resolv.conf /etc/resolv.conf-old
#echo search $fqdn > /etc/resolv.conf; echo nameserver $dnsIP >> /etc/resolv.conf
###Add additional DNS servers if desired
#echo nameserver $dnsIP2 >> /etc/resolv.conf
#echo nameserver $dnsIP3 >> /etc/resolv.conf
#echo ########################
#echo DNS is configured!
#echo ########################
#cat /etc/resolv.conf
#nslookup $linuxhost
### modify /etc/resolv.conf to prevent overwrite
#chattr -i /etc/resolv.conf
######################
###  Install pkgs  ###
######################
### Install/Update Kerberos packages
#zypper --non-interactive install krb5
#zypper --non-interactive install krb5-client
#zypper --non-interactive install krb5-32bit
#zypper --non-interactive install krb5-apps-clients
#zypper --non-interactive install pam_krb5
#zypper --non-interactive install pam_krb5-32bit
#### Install/Update NFSv4 packages
#zypper --non-interactive install nfs-client
#### Install/update SSSD
#zypper --non-interactive install sssd
#zypper --non-interactive install sssd-tools
#zypper --non-interactive remove nscd
######################
## Kerberos Config ###
######################
## Allow secure NFS
#sed -i 's/NFS_SECURITY_GSS="no"/NFS_SECURITY_GSS="yes"/g' /etc/sysconfig/nfs
#echo ########################
#echo Secure NFS enabled!
#echo ########################
#cat /etc/sysconfig/nfs | grep GSS
###Configure /etc/krb5.conf
#mv /etc/krb5.conf /etc/krb5.default
#echo [libdefaults]> /etc/krb5.conf
#echo  default_realm = $realm>> /etc/krb5.conf
#echo  dns_lookup_realm = true>> /etc/krb5.conf
#echo  dns_lookup_kdc = true>> /etc/krb5.conf
```



```
#echo  allow_weak_crypto = true>> /etc/krb5.conf
#echo  >> /etc/krb5.conf
#echo [realms]>> /etc/krb5.conf
#echo        $realm = {>> /etc/krb5.conf
#echo   kdc = $fqdn:88>> /etc/krb5.conf
#echo            default_domain = $fqdn>> /etc/krb5.conf
#echo        }>> /etc/krb5.conf
#echo  >> /etc/krb5.conf
#echo [logging]>> /etc/krb5.conf
#echo        kdc = FILE:/var/log/krb5kdc.log>> /etc/krb5.conf
#echo        admin_server = FILE:/var/log/kadmin.log>> /etc/krb5.conf
#echo        default = FILE:/var/log/krb5lib.log>> /etc/krb5.conf
#echo  >> /etc/krb5.conf
#echo [domain_realm]>> /etc/krb5.conf
#echo        .$defaultdomain = $realm>> /etc/krb5.conf
#echo        .$fqdn = $realm>> /etc/krb5.conf
#echo ########################
#echo Kerberos file is configured!
#echo ########################
#cat /etc/krb5.conf
### Set path to ktutil in SUSE only
# export PATH=$PATH:/usr/lib/mit/bin
#### Create Keytab file
### Keytab file must be moved from KDC before this step
#ktutil <<EOF
#rkt /$linuxhost.keytab
#wkt /etc/krb5.keytab
#list
#exit
#EOF
#### Ensure services are started properly (SUSE only)
#systemctl enable rpcbind.service
#systemctl enable nfs.service
#service rpcbind start
#service nfs start
########################
##### NFSv4 Config #####
########################
#### Configure /etc/idmpad.conf (if not already configured)
#mv /etc/idmapd.conf /etc/idmapd.default; echo [General] > /etc/idmapd.conf; echo Domain = $fqdn
>> /etc/idmapd.conf; echo [Mapping] >> /etc/idmapd.conf; echo Nobody-User = nobody >>
/etc/idmapd.conf; echo Nobody-Group = nobody >> /etc/idmapd.conf; echo [Translation] >>
/etc/idmapd.conf; echo Method = nsswitch >> /etc/idmapd.conf; chmod 0600 /etc/idmapd.conf; chown
root:root /etc/idmapd.conf
#echo ########################
#echo NFSv4 domain configured!
#echo ########################
#cat /etc/idmapd.conf
#### Restart idmapd
#service nfs restart
########################
##### SSSD Config #####
########################
#### Configure the /etc/sssd/sssd.conf file
#mv /etc/sssd/sssd.conf /etc/sssd/sssd.default
#echo [domain/default] > /etc/sssd/sssd.conf
#echo cache_credentials = True >> /etc/sssd/sssd.conf
#echo case_sensitive = False >> /etc/sssd/sssd.conf
#echo [sssd] >> /etc/sssd/sssd.conf
#echo config_file_version = 2 >> /etc/sssd/sssd.conf
#echo services = nss, pam >> /etc/sssd/sssd.conf
#echo domains = $domain >> /etc/sssd/sssd.conf
#echo debug_level = 7 >> /etc/sssd/sssd.conf
#echo [nss] >> /etc/sssd/sssd.conf
#echo filter_users = root,ldap,named,avahi,haldaemon,dbus,radiusd,news,nscd >>
/etc/sssd/sssd.conf
#echo filter_groups = root >> /etc/sssd/sssd.conf
#echo [pam] >> /etc/sssd/sssd.conf
#echo [domain/$domain] >> /etc/sssd/sssd.conf
#echo id_provider = ldap >> /etc/sssd/sssd.conf
```



```
#echo auth_provider = krb5 >> /etc/sssd/sssd.conf
#echo case_sensitive = false >> /etc/sssd/sssd.conf
#echo chpass_provider = krb5 >> /etc/sssd/sssd.conf
#echo cache_credentials = false >> /etc/sssd/sssd.conf
### Use ldap_uri only if there is a single DC
#echo ldap_uri = _srv_,ldap://$fqdn >> /etc/sssd/sssd.conf
#echo ldap_search_base = $basedn >> /etc/sssd/sssd.conf
#echo ldap_schema = rfc2307 >> /etc/sssd/sssd.conf
#echo ldap_sasl_mech = GSSAPI >> /etc/sssd/sssd.conf
#echo ldap_user_object_class = user >> /etc/sssd/sssd.conf
#echo ldap_group_object_class = group >> /etc/sssd/sssd.conf
#echo ldap_user_home_directory = unixHomeDirectory >> /etc/sssd/sssd.conf
#echo ldap_user_principal = userPrincipalName >> /etc/sssd/sssd.conf
#echo ldap_group_member = memberUid >> /etc/sssd/sssd.conf
#echo ldap_group_name = cn >> /etc/sssd/sssd.conf
#echo ldap_account_expire_policy = ad >> /etc/sssd/sssd.conf
#echo ldap_force_upper_case_realm = true >> /etc/sssd/sssd.conf
#echo ldap_group_search_base = $userdn >> /etc/sssd/sssd.conf
#echo ldap_sasl_authid = root/$linuxhost.$fqdn@$realm >> /etc/sssd/sssd.conf
### Use krb5_server and krb5_kpasswd only if there is a single DC
#echo krb5_server = $fqdn >> /etc/sssd/sssd.conf
#echo krb5_realm = $realm >> /etc/sssd/sssd.conf
#echo krb5_kpasswd = $fqdn >> /etc/sssd/sssd.conf
#echo ########################
#echo SSSD conf file created!
#echo ########################
#cat /etc/sssd/sssd.conf
#### Ensure /etc/sssd/sssd.conf is 0600 perms
#chmod 0600 /etc/sssd/sssd.conf
#chown root:root /etc/sssd/sssd.conf
#### Configure nsswitch file
#sed -i 's/passwd: compat/passwd: sss compat/g' /etc/nsswitch.conf
#sed -i 's/group:  compat/group: sss compat/g' /etc/nsswitch.conf
#echo ########################
#echo nsswitch.conf file set!
#echo ########################
#cat /etc/nsswitch.conf | grep sss
######################
#### PAM Config #####
######################
#pam-config --add --sss
#pam-config --add --krb5
#sed -i 's/required/sufficient/g' /etc/pam.d/common-auth
#sed -i 's/required/sufficient/g' /etc/pam.d/common-account
#sed -i 's/required/sufficient/g' /etc/pam.d/common-auth-pc
#sed -i 's/required/sufficient/g' /etc/pam.d/common-account-pc
#### Enable SSSD to start at boot (SUSE only)
#systemctl enable sssd.service
#### Restart SSSD
#service sssd restart
```

Ubuntu Configuration

Network Config

1. Set the host name on the client in /etc/hostname and with the `hostname` command.

Sample file:

```
ubuntu.domain.netapp.com
```

Sample command:

```
hostname ubuntu.domain.netapp.com
```


2. Update resolv.conf with DNS information (use only if DHCP is not used or if DHCP is not providing the correct DNS domain).

Sample file:

```
search domain.netapp.com
nameserver 10.63.98.101
```

a. Prevent /etc/resolv.conf from being overwritten by networking scripts (optional).

```
chattr –i /etc/resolv.conf
```

Install Packages

3. Install/update Kerberos packages.

```
apt-get install krb5-user –y
apt-get install nfs-common –y
```

4. Install/update SSSD.

```
apt-get install sssd –y
```

Kerberos [?]

5. Allow secure NFS in /etc/sysconfig/nfs.

```
sed –i 's/NEED_GSSD=/NEED_GSSD="yes"/g' /etc/default/nfs-common
```

6. Configure /etc/krb5.conf.

Sample file:

```
[libdefaults]
default_realm = DOMAIN.NETAPP.COM
dns_lookup_realm = true
dns_lookup_kdc = true
allow_weak_crypto = true

[realms]
DOMAIN.NETAPP.COM = {
kdc = domain.netapp.com:88
default_domain = domain.netapp.com
}

[logging]
kdc = FILE:/var/log/krb5kdc.log
admin_server = FILE:/var/log/kadmin.log
default = FILE:/var/log/krb5lib.log

[domain_realm]
.netapp.com = DOMAIN.NETAPP.COM
.domain.netapp.com = DOMAIN.NETAPP.COM
```

7. Create /etc/krb5.keytab (the keytab file must be copied from the KDC first).

```
ktutil <<EOF
rkt [/hostname.keytab]
wkt /etc/krb5.keytab
```



```
list
exit
EOF
```

8. Restart Kerberos services.

```
service gssd restart
```

NFSv4 [?]

9. Update the NFSv4 domain (if it is not already configured).

Sample file:

```
[General]
Domain = domain.domain.netapp.com
[Mapping]
Nobody-User = nobody
Nobody-Group = nobody
[Translation]
Method = nsswitch
```

10. Restart the NFSv4 daemon.

```
service idmapd restart
```

SSSD [?]

11. Configure /etc/sssd/sssd.conf.

Sample file:

```
[domain/default]
cache_credentials = True
case_sensitive = False
[sssd]
config_file_version = 2
services = nss, pam
domains = DOMAIN
debug_level = 7
[nss]
filter_users = root,ldap,named,avahi,haldaemon,dbus,radiusd,news,nscd
filter_groups = root
[pam]
[domain/YOURDOMAINNAME]
id_provider = ldap
auth_provider = krb5
chpass_provider = krb5
case_sensitive = true
cache_credentials = false
ldap_search_base = dc=domain,dc=netapp,dc=com
ldap_schema = rfc2307
ldap_sasl_mech = GSSAPI
ldap_user_object_class = user
ldap_group_object_class = group
ldap_user_home_directory = unixHomeDirectory
ldap_user_principal = userPrincipalName
ldap_account_expire_policy = ad
ldap_force_upper_case_realm = true
ldap_user_search_base = cn=Users,dc=domain,dc=netapp,dc=com
ldap_group_search_base = cn=Users,dc=domain,dc=netapp,dc=com
```



```
ldap_sasl_authid = root/ubuntu.domain.netapp.com@DOMAIN.NETAPP.COM
krb5_realm = DOMAIN.NETAPP.COM
```

a. Verify that the file has 0600 permissions and root:root owner.

```
chmod 0600 /etc/sssd/sssd.conf
chown root:root /etc/sssd/sssd.conf
```

12. Restart the SSSD service.

```
service sssd restart
```

Sample Shell Script—Ubuntu Config

- Copy, paste, and modify the following into a file on an NFS client to be used with Kerberos.
- Replace the variables at the beginning of the script with the necessary values.
- Uncomment out the entries intended for use.
- File is not supported by NetApp and does not cover every use case.
- Split the script into sections and run each section separately to troubleshoot issues more easily.

```
###################################
####  Define variables below!  ####
###################################
#!/bin/bash
# Linux/UNIX box with ssh key based login enabled
linuxhost={hostname}
dnsIP1={IP}
###Add Additional DNS servers if desired
#dnsIP2={IP}
#dnsIP3={IP}
fqdn={domain.netapp.com}
domain={DOMAIN}
realm={DOMAIN.NETAPP.COM}
defaultdomain={netapp.com}
userdn={cn=Users,dc=domain,dc=netapp,dc=com}
basedn={dc=domain,dc=netapp,dc=com}
###################################
####  Define variables above!  ####
###################################
## Script backs files up to ensure config can be reverted easily
######################
### Network config ###
######################
### Modify the network config to include the hostname
## NOTE: Review the contents of the network file before modifying
#mv /etc/sysconfig/network /etc/sysconfig/network-original
#echo NETWORKING=yes > /etc/sysconfig/network; echo HOSTNAME=$linuxhost.$fqdn >> /etc/hostname
#hostname $linuxhost.$fqdn
#echo ########################
#echo Hostname configured!
#echo ########################
#cat /etc/sysconfig/network
### Configure DNS
#mv /etc/resolv.conf /etc/resolv.conf-old
#echo search $fqdn > /etc/resolv.conf; echo nameserver $dnsIP >> /etc/resolv.conf
###Add additional DNS servers if desired
# echo nameserver $dnsIP2 >> /etc/resolv.conf
# echo nameserver $dnsIP3 >> /etc/resolv.conf
#echo ########################
#echo DNS is configured!
#echo ########################
#cat /etc/resolv.conf
#nslookup $linuxhost
### modify /etc/resolv.conf to prevent overwrite
#chattr -i /etc/resolv.conf
######################
```



```
###  Install pkgs  ###
######################
### Install/Update Kerberos packages
#apt-get install krb5-user -y
#apt-get install nfs-common -y
#### Install/update SSSD
#apt-get install sssd -y
######################
## Kerberos Config ###
######################
## Allow secure NFS
#sed -i 's/NEED_GSSD=/NEED_GSSD="yes"/g' /etc/default/nfs-common
#echo ########################
# echo Secure NFS configured!
#echo ########################
#cat /etc/default/nfs-common | grep SECURE_NFS
###Configure /etc/krb5.conf
#mv /etc/krb5.conf /etc/krb5.default
#echo [libdefaults]> /etc/krb5.conf
#echo  default_realm = $realm>> /etc/krb5.conf
#echo  dns_lookup_realm = true>> /etc/krb5.conf
#echo  dns_lookup_kdc = true>> /etc/krb5.conf
#echo  allow_weak_crypto = true>> /etc/krb5.conf
#echo  >> /etc/krb5.conf
#echo [realms]>> /etc/krb5.conf
#echo        $realm = {>> /etc/krb5.conf
#echo   kdc = $fqdn:88>> /etc/krb5.conf
#echo            default_domain = $fqdn>> /etc/krb5.conf
#echo        }>> /etc/krb5.conf
#echo  >> /etc/krb5.conf
#echo [logging]>> /etc/krb5.conf
#echo        kdc = FILE:/var/log/krb5kdc.log>> /etc/krb5.conf
#echo        admin_server = FILE:/var/log/kadmin.log>> /etc/krb5.conf
#echo        default = FILE:/var/log/krb5lib.log>> /etc/krb5.conf
#echo  >> /etc/krb5.conf
#echo [domain_realm]>> /etc/krb5.conf
#echo        .$defaultdomain = $realm>> /etc/krb5.conf
#echo        .$fqdn = $realm>> /etc/krb5.conf
#echo ########################
#echo Kerberos file is configured!
#echo ########################
#cat /etc/krb5.conf
#### Create Keytab file
### Keytab file must be moved from KDC before this step
#ktutil <<EOF
#rkt /$linuxhost.keytab
#wkt /etc/krb5.keytab
#list
#exit
#EOF
#### Restart Kerberos service
#service gssd restart
########################
##### NFSv4 Config #####
########################
#### Configure /etc/idmpad.conf (if not already configured)
#mv /etc/idmapd.conf /etc/idmapd.default; echo [General] > /etc/idmapd.conf; echo Domain = $fqdn
>> /etc/idmapd.conf; echo [Mapping] >> /etc/idmapd.conf; echo Nobody-User = nobody >>
/etc/idmapd.conf; echo Nobody-Group = nobody >> /etc/idmapd.conf; echo [Translation] >>
/etc/idmapd.conf; echo Method = nsswitch >> /etc/idmapd.conf; chmod 0600 /etc/idmapd.conf; chown
root:root /etc/idmapd.conf
#echo ########################
#echo NFSv4 domain configured!
#echo ########################
#cat /etc/idmapd.conf
#### Restart idmapd
#service idmapd restart
########################
##### SSSD Config #####
########################
#### Configure the /etc/sssd/sssd.conf file
```



```
#mv /etc/sssd/sssd.conf /etc/sssd/sssd.default
#echo [domain/default] > /etc/sssd/sssd.conf
#echo cache_credentials = True >> /etc/sssd/sssd.conf
#echo case_sensitive = False >> /etc/sssd/sssd.conf
#echo [sssd] >> /etc/sssd/sssd.conf
#echo config_file_version = 2 >> /etc/sssd/sssd.conf
#echo services = nss, pam >> /etc/sssd/sssd.conf
#echo domains = $domain >> /etc/sssd/sssd.conf
#echo debug_level = 7 >> /etc/sssd/sssd.conf
#echo [nss] >> /etc/sssd/sssd.conf
#echo filter_users = root,ldap,named,avahi,haldaemon,dbus,radiusd,news,nscd >>
/etc/sssd/sssd.conf
#echo filter_groups = root >> /etc/sssd/sssd.conf
#echo [pam] >> /etc/sssd/sssd.conf
#echo [domain/$domain] >> /etc/sssd/sssd.conf
#echo id_provider = ldap >> /etc/sssd/sssd.conf
#echo auth_provider = krb5 >> /etc/sssd/sssd.conf
#echo case_sensitive = false >> /etc/sssd/sssd.conf
#echo chpass_provider = krb5 >> /etc/sssd/sssd.conf
#echo cache_credentials = false >> /etc/sssd/sssd.conf
### Use ldap_uri only if there is a single DC
#echo ldap_uri = _srv_,ldap://$fqdn >> /etc/sssd/sssd.conf
#echo ldap_search_base = $basedn >> /etc/sssd/sssd.conf
#echo ldap_schema = rfc2307 >> /etc/sssd/sssd.conf
#echo ldap_sasl_mech = GSSAPI >> /etc/sssd/sssd.conf
#echo ldap_user_object_class = user >> /etc/sssd/sssd.conf
#echo ldap_group_object_class = group >> /etc/sssd/sssd.conf
#echo ldap_user_home_directory = unixHomeDirectory >> /etc/sssd/sssd.conf
#echo ldap_user_principal = userPrincipalName >> /etc/sssd/sssd.conf
#echo ldap_group_member = memberUid >> /etc/sssd/sssd.conf
#echo ldap_group_name = cn >> /etc/sssd/sssd.conf
#echo ldap_account_expire_policy = ad >> /etc/sssd/sssd.conf
#echo ldap_force_upper_case_realm = true >> /etc/sssd/sssd.conf
#echo ldap_group_search_base = $userdn >> /etc/sssd/sssd.conf
#echo ldap_sasl_authid = root/$linuxhost.$fqdn@$realm >> /etc/sssd/sssd.conf
### Use krb5_server and krb5_kpasswd only if there is a single DC
#echo krb5_server = $fqdn >> /etc/sssd/sssd.conf
#echo krb5_realm = $realm >> /etc/sssd/sssd.conf
#echo krb5_kpasswd = $fqdn >> /etc/sssd/sssd.conf
#echo ########################
#echo SSSD conf file created!
#echo ########################
#cat /etc/sssd/sssd.conf
#### Ensure /etc/sssd/sssd.conf is 0600 perms
#chmod 0600 /etc/sssd/sssd.conf
#chown root:root /etc/sssd/sssd.conf
#### Restart SSSD
#service sssd restart
```

Solaris Configuration

Network Config

1. Set the host name on the client in /etc/nodename and with the `hostname` command.

Sample file:

```
solaris.domain.netapp.com
```

Sample command:

```
hostname solaris.domain.netapp.com
```

2. Update resolv.conf with DNS information (use only if DHCP is not used or if DHCP is not providing the correct DNS domain).

Sample file:

```
search domain.netapp.com
nameserver 10.63.98.101
```

Kerberos [?]

3. Allow secure NFS in /etc/nfssec.conf by uncommenting the desired krb values. Keep in mind that clustered Data ONTAP supports only krb5 currently.

4. Configure /etc/krb5/krb5.conf.

Sample file:

```
[libdefaults]
default_realm = DOMAIN.NETAPP.COM
dns_lookup_realm = true
dns_lookup_kdc = true
allow_weak_crypto = true

[realms]
DOMAIN.NETAPP.COM = {
kdc = domain.netapp.com:88
default_domain = domain.netapp.com
}

[logging]
kdc = FILE:/var/log/krb5kdc.log
admin_server = FILE:/var/log/kadmin.log
default = FILE:/var/log/krb5lib.log

[domain_realm]
.netapp.com = DOMAIN.NETAPP.COM
.domain.netapp.com = DOMAIN.NETAPP.COM
```

5. Create /etc/krb5/krb5.keytab (the keytab file must be copied from the KDC first).

```
ktutil <<EOF
rkt [/hostname.keytab]
wkt /etc/krb5.keytab
list
exit
EOF
```

6. Restart Kerberos services.

```
svcadm restart gss
```

NFSv4 [?]

7. Update the NFSv4 domain (if it is not already configured).

Sample file:

```
[General]
Domain = domain.domain.netapp.com
[Mapping]
Nobody-User = nobody
Nobody-Group = nobody
```



```
[Translation]
Method = nsswitch
```

8. Restart the NFSv4 daemon.

```
svcadm restart mapid
```

LDAP [?]

9. Configure ldapclient with either simple or sasl/GSSAPI.

Sample configuration using simple auth:

```
ldapclient manual \
-a credentialLevel=proxy \
-a authenticationMethod=simple \
-a proxyDN=CN=ldapuser,CN=Users,DC=domain,DC=netapp,DC=com \
-a proxyPassword=P@ssw0rd \  <<<< optional
-a defaultSearchBase=dc=domain,dc=netapp,dc=com \
-a defaultSearchScope=sub \
-a domainName=domain.netapp.com \
-a defaultServerList=10.61.179.152 \
-a attributeMap=group:userpassword=userPassword \
-a attributeMap=group:memberuid=memberUid \
-a attributeMap=group:gidnumber=gidNumber \
-a attributeMap=passwd:gecos=cn \
-a attributeMap=passwd:gidnumber=gidNumber \
-a attributeMap=passwd:uidnumber=uidNumber \
-a attributeMap=passwd:homedirectory=unixHomeDirectory \
-a attributeMap=passwd:loginshell=loginShell \
-a attributeMap=shadow:shadowflag=shadowFlag \
-a attributeMap=shadow:userpassword=userPassword \
-a objectClassMap=group:posixGroup=group \
-a objectClassMap=passwd:posixAccount=user \
-a objectClassMap=shadow:shadowAccount=user \
-a serviceSearchDescriptor=passwd:CN=Users,DC=domain,DC=netapp,DC=com?sub \
-a serviceSearchDescriptor=group:CN=Users,DC=domain,DC=netapp,DC=com?sub
```

Sample configuration using sasl/GSSAPI:

```
ldapclient manual \
-a credentialLevel=self \
-a authenticationMethod=sasl/GSSAPI \
-a defaultSearchBase=dc=domain,dc=netapp,dc=com \
-a defaultSearchScope=sub \
-a domainName=domain.netapp.com \
-a defaultServerList=10.61.179.152 \
-a attributeMap=group:userpassword=userPassword \
-a attributeMap=group:memberuid=memberUid \
-a attributeMap=group:gidnumber=gidNumber \
-a attributeMap=passwd:gecos=cn \
-a attributeMap=passwd:gidnumber=gidNumber \
-a attributeMap=passwd:uidnumber=uidNumber \
-a attributeMap=passwd:homedirectory=unixHomeDirectory \
-a attributeMap=passwd:loginshell=loginShell \
-a attributeMap=shadow:shadowflag=shadowFlag \
-a attributeMap=shadow:userpassword=userPassword \
-a objectClassMap=group:posixGroup=group \
-a objectClassMap=passwd:posixAccount=user \
-a objectClassMap=shadow:shadowAccount=user \
-a serviceSearchDescriptor=passwd:CN=Users,DC=domain,DC=netapp,DC=com?sub \
-a serviceSearchDescriptor=group:CN=Users,DC=domain,DC=netapp,DC=com?sub
```


Appendix

Kerberos Encryption Types

Kerberos V5 supports multiple encryption types (*enctypes*). The type used in a given instance is automatically negotiated between the client and the Kerberos KDC servers, based on client and server settings as well as encryption types used to encrypt the password for the user and service principals.

The following section defines the different enctypes used by Kerberos V5. The information was gathered from this web site:

http://ait.its.psu.edu/services/identity-access-management/identity/kerberos/encryption-types.html

Table 36) Enctypes

Enctype	Cipher Algorithm	Cipher Mode	Key Length	HMAC	Strength
aes256-cts aes256-cts-hmac-sha1-96	AES	CBC+CTS	256 bit	SHA-1 96-bits	Strongest
aes128-cts aes128-cts-hmac-sha1-96	AES	CBC+CTS	128 bit	SHA-1 96-bits	Strong
rc4-hmac	RC4		128 bit	SHA-1 96-bits	Strong
des3-cbc-sha1	3DES	CBC	168 bit	SHA-1 96-bits	Strong
des-cbc-crc	DES	CBC	56 bit	CRC 32-bit	Weak
des-cbc-md5	DES	CBC	56 bit	MD5 96-bits	Weak, but strongest single DES

Cipher Algorithm and Mode Terminology

Block Cipher

A cipher mode that encrypts data at a fixed size, or *block,* at a time (for example, 64 bits). Contrast this with *stream cipher*.

Cipher

An encryption algorithm, or defined process, with which data is encrypted and decrypted.

3DES

Also known as “triple DES”; a method of using three separate 56-bit DES keys in three passes to make a stronger (but slower) encryption algorithm.

AES

Advanced Encryption Standard. This replaces DES and 3DES with stronger encryption and longer key lengths.

CBC

Cipher Block Chaining. A method by which the encrypted *cipher-text* from the last block of a *block cipher* is used to further strengthen the next block. Typically the next block's *plain-text* is XORed with the *cipher-text* of the previous block. This hides patterns of repeated plain-text blocks.

CRC

Cyclical Redundancy Check. A method for validating that data has not been corrupted by trivial medium noise (line noise, hard disk damage, and so on).

CTS

Cipher Text Stealing. A method similar to CBC in which the last plain-text block is better protected when it is shorter than other blocks (when the plain-text message does not end evenly on a block boundary).

DES

Data Encryption Standard. Designed to handle only 56-bit key lengths, which causes this to be a weak enctype.

HMAC

Hash-Based Message Authentication Code. A method used to simultaneously verify both the data integrity and the authenticity of a message.

MD5

A Message Digest hashing algorithm. A method of HMAC.

RC4

A symmetric stream cipher by Ron Rivest (hence, "Rivest Cipher"). It can handle several key sizes such as 40-bit and 128-bit keys.

SHA-1

Secure Hash Algorithm. A method of HMAC.

Stream Cipher

A stream cipher is designed to normally encrypt and decrypt on a single bit at a time. Contrast this with *block cipher*. Both block and stream ciphers can operate in block and stream modes.

Symmetric Cipher

A cipher is deemed *symmetric* when the same key is used to encrypt and decrypt the same data. When two keys are used, one to encrypt and another to decrypt (or one to sign and the other to verify the digital signature), it is called an *asymmetric* cipher. Kerberos can use asymmetric ciphers, but it was designed to need only symmetric ciphers.

About the Machine Account Attributes

The attributes userAccountControl and msDS-SupportedEncryptionTypes are used to specify how a machine will authenticate in the domain. The values are specified by adding a series of values together.

For example, the value 2097152 (hex 0x200000) is a default value for USE_DES_KEY_ONLY. However, Windows Active Directory will modify the value once it’s applied to 2097664 (hex 0x200200).

Example:

```
USE_DES_KEY_ONLY (2097152) + NORMAL_ACCOUNT (512) = 2097664
```

The following tables show the values available for each attribute. The source of the information is from http://support.microsoft.com/kb/305144 and

http://blogs.msdn.com/b/openspecification/archive/2011/05/31/windows-configurations-for-kerberos-supported-encryption-type.aspx.

Table 37) Valid userAccountControl attribute values.

Property Flag	Value in Hexadecimal	Value in Decimal
SCRIPT	0x0001	1
ACCOUNTDISABLE	0x0002	2
HOMEDIR_REQUIRED	0x0008	8
LOCKOUT	0x0010	16
PASSWORD_NOTRQD	0x0020	32
PASSWD_CANT_CHANGE **Note:** Cannot be modified via userAccountControl.	0x0040	64
ENCRYPTED_TEXT_PWD_ALLOWED	0x0080	128
TEMP_DUPLICATE_ACCOUNT	0x0100	256
NORMAL_ACCOUNT	0x0200	512
INTERDOMAIN_TRUST_ACCOUNT	0x0800	2048
WORKSTATION_TRUST_ACCOUNT	0x1000	4096
SERVER_TRUST_ACCOUNT	0x2000	8192
DONT_EXPIRE_PASSWORD	0x10000	65536
MNS_LOGON_ACCOUNT	0x20000	131072
SMARTCARD_REQUIRED	0x40000	262144
TRUSTED_FOR_DELEGATION	0x80000	524288
NOT_DELEGATED	0x100000	1048576
USE_DES_KEY_ONLY	0x200000	2097152

DONT_REQ_PREAUTH	0x400000	4194304
PASSWORD_EXPIRED	0x800000	8388608
TRUSTED_TO_AUTH_FOR_DELEGATION	0x1000000	16777216
PARTIAL_SECRETS_ACCOUNT	0x04000000	67108864

Property Flag Descriptions

- **SCRIPT** - The logon script will be run.
- **ACCOUNTDISABLE** - The user account is disabled.
- **HOMEDIR_REQUIRED** - The home folder is required.
- **PASSWD_NOTREQD** - No password is required.
- **PASSWD_CANT_CHANGE** - The user cannot change the password. This is a permission setting on the user's object. For information about how to programmatically set this permission, visit the following web site:

  http://msdn2.microsoft.com/en-us/library/aa746398.aspx
- **ENCRYPTED_TEXT_PASSWORD_ALLOWED** - The user can send an encrypted password.
- **TEMP_DUPLICATE_ACCOUNT** - This is an account for users whose primary account is in another domain. This account provides user access to this domain, but not to any domain that trusts this domain. This is sometimes referred to as a local user account.
- **NORMAL_ACCOUNT** - This is a default account type that represents a typical user.
- **INTERDOMAIN_TRUST_ACCOUNT** - This is a permit to trust an account for a system domain that trusts other domains.
- **WORKSTATION_TRUST_ACCOUNT** - This is a computer account for a computer that is running Microsoft Windows NT 4.0 Workstation, Microsoft Windows NT 4.0 Server, Microsoft Windows 2000 Professional, or Windows 2000 Server and is a member of this domain.
- **SERVER_TRUST_ACCOUNT** - This is a computer account for a domain controller that is a member of this domain.
- **DONT_EXPIRE_PASSWD** - This represents the password, which should never expire on the account.
- **MNS_LOGON_ACCOUNT** - This is an MNS logon account.
- **SMARTCARD_REQUIRED** - When this flag is set, it forces the user to log on by using a smart card.
- **TRUSTED_FOR_DELEGATION** - When this flag is set, the service account (the user or computer account) under which a service runs is trusted for Kerberos delegation. Any such service can impersonate a client requesting the service. To enable a service for Kerberos delegation, you must set this flag on the **userAccountControl** property of the service account.
- **NOT_DELEGATED** - When this flag is set, the security context of the user is not delegated to a service even if the service account is set as trusted for Kerberos delegation.
- **USE_DES_KEY_ONLY** (Windows 2000/Windows Server 2003) - Restrict this principal to use only Data Encryption Standard (DES) encryption types for keys.
- **DONT_REQUIRE_PREAUTH** (Windows 2000/Windows Server 2003) - This account does not require Kerberos preauthentication for logging on.
- **PASSWORD_EXPIRED** (Windows 2000/Windows Server 2003) - The user's password has expired.
- **TRUSTED_TO_AUTH_FOR_DELEGATION** (Windows 2000/Windows Server 2003) - The account is enabled for delegation. This is a security-sensitive setting. Accounts that have this option enabled

should be tightly controlled. This setting lets a service that runs under the account assume a client's identity and authenticate as that user to other remote servers on the network.

- **PARTIAL_SECRETS_ACCOUNT** (Windows Server 2008/Windows Server 2008 R2) - The account is a read-only domain controller (RODC). This is a security-sensitive setting. Removing this setting from an RODC compromises security on that server.

Table 38) Valid msDS-SupportedEncryptionTypes attribute values.

The msDS-SupportedEncryptionTypes value is set to 25 (hex 0x19). That value translates to allowing all supported encryption types. The table below shows which values are valid. The value 25 is derived by adding all decimal values together (1+2+4+8+10).

Property Flag	Value in Hexadecimal	Value in Decimal
DES-CBC-CRC	0x01	1
DES-CBC-MD5	0x02	2
RC4-HMAC	0x04	4
AES128-CTS-HMAC-SHA1-96	0x08	8
AES256-CTS-HMAC-SHA1-96	0x10	10

Kerberos Packet Types, Errors, and Terminology

The following tables show the type of Kerberos requests that take place over the wire, as well as what error codes can be returned during requests. This is intended to help troubleshoot by explaining what each request does.

Table 39) Kerberos packets.

Kerberos Packet	What It Does
AS-REQ	Authentication Service request – looks up the user name and password to get the Ticket Granting Ticket (TGT); also requests the session key.
AS-REP	Authentication Service reply – delivers the TGT and session key.
AP-REQ	Application server request – certifies to a server that the sender has recent knowledge of the encryption key in the accompanying ticket to help the server detect replays. It also assists in the selection of a "true session key" to use with the particular session.
AP-REP	Application server reply – includes the session key and sequence number.
TGS-REQ	Ticket Granting Server request – uses the TGT to get the Service Ticket (ST).
TGS-REP	Ticket Granting Server reply – delivers the ST.

Table 40) Kerberos errors from Kerberos errors in network captures.

Kerberos Error	What It Means
KDC_ERR_S_PRINCIPAL_UNKNOWN	The SPN does not exist or there was a duplicate SPN on the KDC. Note the “S” in the error—this stands for “SPN” or “service.”
KDC_ERR_C_PRINCIPAL_UNKNOWN	The UPN does not exist or there was a duplicate UPN on the KDC. Note the “C” in the error—this stands for “client” and refers to the user principal rather than the service principal.
KDC_ERR_ETYPE_NOTSUPP	Encryption type requested by the client is not supported by the KDC. This is common with DES and Windows 2008 R2.
KDC_ERR_PREAUTH_REQUIRED	This simply means that the KDC wants a password for the account attempting authentication; this is a benign error.
KDC_ERR_PREAUTH_FAILED	The preauthentication failed, generally because the password was incorrect.
KRB_AP_ERR_SKEW	The time is outside the allowed skew window. This is typically 5 minutes.
KRB_AP_ERR_REPEAT	This is the security mechanism to prevent replay attacks. If server name, client name, time, and microsecond fields from the Authenticator match recently seen entries in the cache, this error will occur.
KRB_AP_ERR_MODIFIED	This indicates that the service was unable to decrypt the ticket that it was given. A common cause is when the Service Principal Name (SPN) is registered to the wrong account. Another possible cause is a duplicate SPN in two different domains in the forest. This can also occur if the KDC where the original ticket was issued is offline, causing the client to need to reauthenticate to a new KDC.

Table 41) Kerberos terminology from CentOS.org and IBM.com.

Term	Definition
KDC	Key Distribution Center – A service that issues Kerberos tickets, usually run on the same host as the ticket-granting server (TGS).

Term	Definition
TGT	Ticket Granting Ticket – A special ticket that allows the client to obtain additional tickets without applying for them from the KDC. Example: krbtgt/domain@REALM. The principal for this exists as a user account named krbtgt in Microsoft Windows Active Directory.
TGS	Ticket Granting Server – A server that issues tickets for a desired service that are in turn given to users for access to the service. The TGS usually runs on the same host as the KDC.
SPN	Service Principal Name – Kerberos principal associated with service in the format of service/instance@REALM. Example: ldap/server.netapp.com@NETAPP.COM.
UPN	User Principal Name – Kerberos principal associated with a user name in the format of user@REALM. Example: ldapuser@NETAPP.COM.
Session key	A temporary encryption key used between two principals, with the lifetime limited to the duration of a single login session.
ST	Service Ticket – A ticket that is issued for a specific service; for example, nfs/instance@REALM for NFS services or ldap/instance@REALM for LDAP services.
AS	Authentication Server – A server that issues tickets for a desired service that are in turn given to users for access to the service. The AS responds to requests from clients who do not have or do not send credentials with a request. It is usually used to gain access to the ticket-granting server (TGS) service by issuing a ticket-granting ticket (TGT). The AS usually runs on the same host as the KDC.
Realm	A network that uses Kerberos, composed of one or more servers called KDCs and a potentially large number of clients.
GSS-API	The Generic Security Service Application Program Interface (defined in RFC-2743 published by the Internet Engineering Task Force) is a set of functions that provide security services. This API is used by clients and services to authenticate to each other without either program having specific knowledge of the underlying mechanism. If a network service (such as cyrus-IMAP) uses GSS-API, it can authenticate using Kerberos.

Non-Windows KDCs

This section covers setting up and configuring non-Windows KDCs. There are multiple offerings of non-Windows KDCs, and they will be added to this document in future iterations. This setup still leverages Windows DNS. These steps are not heavily detailed, but they cover the basic setup. Client vendors have plenty of documentation on their flavor of Kerberos server.

Setting Up MIT Kerberos

To configure MIT Kerberos, use the following steps.

Table 42) Configuring MIT Kerberos

1. Set up the MIT Kerberos server.

Example: http://www.centos.org/docs/5/html/5.1/Deployment_Guide/s1-kerberos-server.html

2. Add _Kerberos and _Kerberos Master DNS SRV records for TCP and UDP.

3. Disable iptables and set selinux to permissive on KDC/client, or allow port 88 in the firewall.

References:

http://www.cyberciti.biz/faq/turn-on-turn-off-firewall-in-linux/

http://www.crypt.gen.nz/selinux/disable_selinux.html

4. Enable secure NFS/rpcgssd.

5. Make sure that hosts are in DNS.

6. Add principals (for users and hosts).

Examples follow.

Adding a root or admin principal:

```
kadmin.local:  addprinc root/admin
WARNING: no policy specified for root/admin@DOMAIN.MIT.NETAPP.COM; defaulting to no policy
Enter password for principal "root/admin@DOMAIN.MIT.NETAPP.COM":
Re-enter password for principal "root/admin@DOMAIN.MIT.NETAPP.COM":
Principal "root/admin@DOMAIN.MIT.NETAPP.COM" created.
```

Adding a cluster NFS principal:

```
kadmin:  add_principal -e "des-cbc-crc:normal des-cbc-md5:normal des3-cbc-sha1:normal" -randkey
nfs/mitkerb.domain.mit.netapp.com@DOMAIN.MIT.NETAPP.COM
WARNING: no policy specified for nfs/mitkerb.domain.mit.netapp.com@DOMAIN.MIT.NETAPP.COM;
defaulting to no policy
Principal "nfs/mitkerb.domain.mit.netapp.com@DOMAIN.MIT.NETAPP.COM" created.
```

Note: -e is used because currently only DES and 3DES are supported for clustered Data ONTAP.

Adding host principals:

```
kadmin:  add_principal -randkey root/mitclient.domain.mit.netapp.com@DOMAIN.MIT.NETAPP.COM
WARNING: no policy specified for root/mitclient.domain.mit.netapp.com@DOMAIN.MIT.NETAPP.COM;
defaulting to no policy
```



```
Principal "root/mitclient.domain.mit.netapp.com@DOMAIN.MIT.NETAPP.COM" created.
```

7. Create keytab files.

Example:

```
ktadd -k /mitkerb.keytab -e "des-cbc-crc:normal des-cbc-md5:normal des3-cbc-sha1:normal"
nfs/mitkerb.domain.mit.netapp.com@DOMAIN.MIT.NETAPP.COM
ktadd -k /mitclient.keytab root/mitclient.domain.mit.netapp.com@DOMAIN.MIT.NETAPP.COM

[root@mit-kdc ~]# kadmin
Authenticating as principal root/admin@DOMAIN.MIT.NETAPP.COM with password.
Password for root/admin@DOMAIN.MIT.NETAPP.COM:
kadmin:  ktadd -k /mitkerb.keytab -e "des-cbc-crc:normal des-cbc-md5:normal des3-cbc-sha1:normal"
nfs/mitkerb.domain.mit.netapp.com@DOMAIN.MIT.NETAPP.COM
Entry for principal nfs/mitkerb.domain.mit.netapp.com@DOMAIN.MIT.NETAPP.COM with kvno 2,
encryption type des-cbc-crc added to keytab WRFILE:/mitkerb.keytab.
Entry for principal nfs/mitkerb.domain.mit.netapp.com@DOMAIN.MIT.NETAPP.COM with kvno 2,
encryption type des3-cbc-sha1 added to keytab WRFILE:/mitkerb.keytab.

kadmin:  ktadd -k /mitclient.keytab root/mitclient.domain.mit.netapp.com@DOMAIN.MIT.NETAPP.COM
Entry for principal root/mitclient.domain.mit.netapp.com@DOMAIN.MIT.NETAPP.COM with kvno 2,
encryption type aes256-cts-hmac-sha1-96 added to keytab WRFILE:/mitclient.keytab.
Entry for principal root/mitclient.domain.mit.netapp.com@DOMAIN.MIT.NETAPP.COM with kvno 2,
encryption type aes128-cts-hmac-sha1-96 added to keytab WRFILE:/mitclient.keytab.
Entry for principal root/mitclient.domain.mit.netapp.com@DOMAIN.MIT.NETAPP.COM with kvno 2,
encryption type des3-cbc-sha1 added to keytab WRFILE:/mitclient.keytab.
Entry for principal root/mitclient.domain.mit.netapp.com@DOMAIN.MIT.NETAPP.COM with kvno 2,
encryption type arcfour-hmac added to keytab WRFILE:/mitclient.keytab.
```

8. To use SSSD in a different KDC (such as a Windows KDC), add a second realm in `krb5.conf`.

Example:

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

[libdefaults]
 default_realm = DOMAIN.MIT.NETAPP.COM
 dns_lookup_realm = false
 dns_lookup_kdc = false
 ticket_lifetime = 24h
 forwardable = yes

[realms]
 DOMAIN.MIT.NETAPP.COM = {
  kdc = mit-kdc.domain.mit.netapp.com:88
  admin_server = mit-kdc.domain.mit.netapp.com:749
  default_domain = domain.mit.netapp.com
 }
 DOMAIN.WIN2K8.NETAPP.COM = {
 kdc = domain.win2k8.netapp.com:88
 default_domain = domain.win2k8.netapp.com
 }

[domain_realm]
 .domain.mit.netapp.com = DOMAIN.MIT.NETAPP.COM
 domain.mit.netapp.com = DOMAIN.MIT.NETAPP.COM
.domain.win2k8.netapp.com = DOMAIN.WIN2K8.NETAPP.COM
domain.win2k8.netapp.com = DOMAIN.WIN2K8.NETAPP.COM


[appdefaults]
 pam = {
   debug = false
   ticket_lifetime = 36000
   renew_lifetime = 36000
   forwardable = true
```



```
    krb4_convert = false
}
```

Note: When using two KDCs (Windows and non-Windows), create principals in both domains.

Troubleshooting Kerberos

When setting up Kerberized NFS, you may encounter issues in getting everything to work. This section is not intended to cover all scenarios, but it does capture most of the common issues.

Issues When Running Kinit

Kinit is similar to logging in on a Windows box. When kinit is being run, there are only three components at work during the login:

- The client
- The KDC
- DNS

Data ONTAP does not participate in the kinit process.

When attempting to run kinit and get a Kerberos ticket, the command can fail for a variety of reasons. In that case, check the following:

- Is the `krb.conf` file configured properly?
  - Is the encryption type being attempted allowed on the KDC?
  - Is the realm correct?
  - Is the port for Kerberos correct? Is the port allowed on the firewall?
- Is DNS configured properly?
  - Does the realm resolve from the client by using `nslookup` or `dig`?
  - Can the client resolve the KDC?
- Is the time in sync between the client and KDC?
- Is the `krb5.keytab` file configured properly?
- Does the user or SPN exist?
  - Is the password correct?
  - Is the account enabled?
  - Are there duplicate SPNs or UPNs?
- Is the gssd service running?
  - Is secure NFS allowed on the client via the NFS configuration file?

Things to Check When Troubleshooting Kinit

- System messages file on the client
- Event logs on the KDC
- Packet traces from the KDC and client

Issues When Enabling Kerberos on a Data LIF

Failures can occur when enabling Kerberos on a data LIF. Unfortunately, the errors returned on the cluster during failures aren't always the most informative. This subsection covers what happens when Kerberos is enabled on a data LIF. It also covers how to troubleshoot issues. Keep in mind that OnCommand® System Manager does not have the ability to run troubleshooting commands, so all troubleshooting must be done from the CLI.

- Has the Kerberos realm been created?
  - Is the Kerberos port that is specified in the `kerberos-realm` command allowed on the firewall?
- Is the SPN specified using the following format: nfs/hostname.domain.com@REALM.COM?
  - Is the realm in all caps?
- Is DNS configured for the SVM?
  - Can the SVM resolve the realm?
  - From the cluster, use `set diag; diag secd dns forward-lookup –node [nodename] –vserver [SVM] -hostname [realm]`.

  Note: The realm must be lowercase for DNS lookups, because the `secd dns` command does not recognize capital letters.
- Is the KDC reachable from the data LIF?
  - Does the data LIF have a proper route?
  - Ping from the KDC to the data LIF.
  - Ping from the cluster via IP using the following: `net ping –lif [lif name] –lif-owner [SVM] –destination [IP address]`.
- Is the time on the cluster within 5 minutes of the time on the KDC?
- Does the user have permissions to the specified OU?
  - Is the user password correct?
- The default OU is Computers, unless specified by the `–ou` option (in 8.2.1 and later only).

When Kerberos is enabled, action takes place only in the following places:

- The cluster
- The KDC
- DNS

Focus troubleshooting efforts on those locations.

Things to Check When Troubleshooting kerberos-config Failures

- From the cluster, run `event log show -messagename secd*`.
- `Secd` logs onto the node where the `kerberos-config` command was issued (access is via systemshell).
- Packet traces from the cluster and KDC.
- Event logs on the KDC.

Issues When Mounting a Kerberized Export

When mounting a Kerberized NFS export, the following factors come into play:

- Export policies and rules
- SPN existence, keytab configuration, and duplicated SPNs
- DNS configuration
- Client configuration

Volume permissions do not come into play until the export is actually mounted. Table 39 shows some common error messages and common causes. The table does not cover all causes and errors.

Table 43) Common mount issues with Kerberized NFS.

Symptom	Cause
Protocol not supported	• Machine account password expired (error seen on NFSv4.1 only)
Access denied by server while mounting	• Export policy doesn't allow krb5 • Export policy doesn't include the client, subnet, or netgroup in the clientmatch • Export policy doesn't allow the NFS version requested • NFS SPN does not exist or is a duplicate • NFS SPN does not map to a valid UNIX user
Requested NFS version or transport protocol is not supported	• NFS server is not created or is not running • NFS is disallowed on the data LIF • NFS version is not enabled on the server • NFS ports are blocked by firewall

For more mount troubleshooting scenarios, see TR-4067: Clustered Data ONTAP NFS Best Practice and Implementation Guide.

Issues When Changing Directories (cd) or Reading/Writing to a Kerberized Export

When attempting to read or write to a Kerberized export, the following factors come into play:

- Export policies and rules
- UNIX permissions
- NFS configuration
- Client configuration

Table 40 shows some common error messages and common causes. The table does not cover all causes and errors.

Table 44) Common read/write issues with Kerberized NFS.

Symptom	Cause
Permission denied	• Did not kinit to a valid user principal on the KDC • No permission to volume (mode bits or NFSv4.x ACL) • Export policy does not permit ro or rw access to the client • NFS service SPN incorrect, duplicated, or missing • DNS not configured properly
Not a directory	• Client issue; retry access or remount

For more read/write troubleshooting scenarios, see TR-4067: Clustered Data ONTAP NFS Best Practice and Implementation Guide.

LDAP Schema Examples in Clustered Data ONTAP 8.2

AD-IDMU schema

```
cluster::> ldap client schema show -schema AD-IDMU
  (vserver services ldap client schema show)

                                Vserver: vs0
                        Schema Template: AD-IDMU
                                Comment: Schema based on Active Directory Identity Management
for UNIX (read-only)
         RFC 2307 posixAccount Object Class: User
           RFC 2307 posixGroup Object Class: Group
          RFC 2307 nisNetgroup Object Class: nisNetgroup
                     RFC 2307 uid Attribute: uid
                RFC 2307 uidNumber Attribute: uidNumber
                RFC 2307 gidNumber Attribute: gidNumber
         RFC 2307 cn (for Groups) Attribute: cn
      RFC 2307 cn (for Netgroups) Attribute: name
            RFC 2307 userPassword Attribute: unixUserPassword
                   RFC 2307 gecos Attribute: name
           RFC 2307 homeDirectory Attribute: unixHomeDirectory
              RFC 2307 loginShell Attribute: loginShell
               RFC 2307 memberUid Attribute: memberUid
       RFC 2307 memberNisNetgroup Attribute: memberNisNetgroup
       RFC 2307 nisNetgroupTriple Attribute: nisNetgroupTriple
ONTAP Name Mapping windowsAccount Attribute: windowsAccount
                        Vserver Owns Schema: false
```

AD-SFU schema

```
cluster::> ldap client schema show -schema AD-SFU
  (vserver services ldap client schema show)

                                Vserver: vs0
                        Schema Template: AD-SFU
                                Comment: Schema based on Active Directory Services for UNIX
(read-only)
         RFC 2307 posixAccount Object Class: User
           RFC 2307 posixGroup Object Class: Group
          RFC 2307 nisNetgroup Object Class: msSFU30NisNetGroup
                     RFC 2307 uid Attribute: sAMAccountName
                RFC 2307 uidNumber Attribute: msSFU30UidNumber
                RFC 2307 gidNumber Attribute: msSFU30GidNumber
         RFC 2307 cn (for Groups) Attribute: cn
      RFC 2307 cn (for Netgroups) Attribute: name
            RFC 2307 userPassword Attribute: msSFU30Password
                   RFC 2307 gecos Attribute: name
           RFC 2307 homeDirectory Attribute: msSFU30HomeDirectory
              RFC 2307 loginShell Attribute: msSFU30LoginShell
               RFC 2307 memberUid Attribute: msSFU30MemberUid
       RFC 2307 memberNisNetgroup Attribute: msSFU30MemberNisNetgroup
       RFC 2307 nisNetgroupTriple Attribute: msSFU30MemberOfNisNetgroup
ONTAP Name Mapping windowsAccount Attribute: windowsAccount
                        Vserver Owns Schema: false
```

AD-SFU-Deprecated schema

```
cluster::> ldap client schema show -schema AD-SFU-Deprecated
  (vserver services ldap client schema show)

                                Vserver: vs0
                        Schema Template: AD-SFU-Deprecated
                                Comment: Schema based on Active Directory Services for UNIX
(read-only)
         RFC 2307 posixAccount Object Class: User
           RFC 2307 posixGroup Object Class: Group
          RFC 2307 nisNetgroup Object Class: nisNetgroup
                     RFC 2307 uid Attribute: uid
```



```
               RFC 2307 uidNumber Attribute: uidNumber
               RFC 2307 gidNumber Attribute: gidNumber
         RFC 2307 cn (for Groups) Attribute: cn
      RFC 2307 cn (for Netgroups) Attribute: name
            RFC 2307 userPassword Attribute: unixUserPassword
                   RFC 2307 gecos Attribute: name
           RFC 2307 homeDirectory Attribute: unixHomeDirectory
              RFC 2307 loginShell Attribute: loginShell
               RFC 2307 memberUid Attribute: primaryGroupID
       RFC 2307 memberNisNetgroup Attribute: memberNisNetgroup
       RFC 2307 nisNetgroupTriple Attribute: nisNetgroupTriple
ONTAP Name Mapping windowsAccount Attribute: windowsAccount
                        Vserver Owns Schema: false
```

RFC-2307

```
cluster::> ldap client schema show -schema RFC-2307
  (vserver services ldap client schema show)

                                    Vserver: vs0
                             Schema Template: RFC-2307
                                    Comment: Schema based on RFC 2307 (read-only)
         RFC 2307 posixAccount Object Class: posixAccount
           RFC 2307 posixGroup Object Class: posixGroup
          RFC 2307 nisNetgroup Object Class: nisNetgroup
                      RFC 2307 uid Attribute: uid
               RFC 2307 uidNumber Attribute: uidNumber
               RFC 2307 gidNumber Attribute: gidNumber
         RFC 2307 cn (for Groups) Attribute: cn
      RFC 2307 cn (for Netgroups) Attribute: cn
            RFC 2307 userPassword Attribute: userPassword
                   RFC 2307 gecos Attribute: gecos
           RFC 2307 homeDirectory Attribute: homeDirectory
              RFC 2307 loginShell Attribute: loginShell
               RFC 2307 memberUid Attribute: memberUid
       RFC 2307 memberNisNetgroup Attribute: memberNisNetgroup
       RFC 2307 nisNetgroupTriple Attribute: nisNetgroupTriple
ONTAP Name Mapping windowsAccount Attribute: windowsAccount
                        Vserver Owns Schema: falseFile Naming Convention
```

Setting Up Password-less SSH in Clustered Data ONTAP

To run noninteractive SSH commands, password-less SSH must be configured for use with clustered Data ONTAP. This is useful when running shell scripts, such as the ones mentioned in section 5.

The following describes how to configure password-less SSH on a Linux client for use with clustered Data ONTAP.

Create the SSH keypair

In the example below, ssh-keygen is used on a Linux box.

Note: If a ssh key pair already exists, there is no need to generate one using ssh-keygen.

```
monitor@linux:/$ ssh-keygen -q -f ~/.ssh/id_rsa -t rsa
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
monitor@linux:/$ ls -lsa ~/.ssh
total 16
4 drwx------ 2 monitor monitor 4096 2008-08-26 11:47 .
4 drwxr-xr-x 3 monitor monitor 4096 2008-08-26 11:47 ..
4 -rw------- 1 monitor monitor 1679 2008-08-26 11:47 id_rsa
4 -rw-r--r-- 1 monitor monitor 401 2008-08-26 11:47 id_rsa.pub
```


Create the user with a public key authentication method

```
netapp::> security login create -username monitor -application ssh -authmethod publickey -profile
admin
```

Create the public key on the cluster

Copy the public key contents of the id_rsa.pub file and place it between quotes in the security login `public key create` command. Take caution not to add carriage returns or other data that modifies the keystring; leave it in one line.

```
netapp::> security login publickey create -username monitor -index 1 -publickey "ssh-rsa
AAAAB3NzaC1yc2EAAAABIwAAAQEA5s4vVbwEO1sOsq7r64V5KYBRXBDb2I5mtGmt0+3p1jjPJrXx4/IPHFLalXAQkG7LhV5Dy
c5jyQiGKVawBYwxxSZ3GqXJNv1aORZHJEuCd0zvSTBGGZ09vra5uCfxkpz8nwaTeiAT232LS2lZ6RJ4dsCz+GAj2eidpPYMld
i2z6RVoxpZ5Zq68MvNzz8b15BS9T7bvdHkC2OpXFXu2jndhgGxPHvfO2zGwgYv4wwv2nQw4tuqMp8e+z0YP73Jg0T3jV8NYra
XO951Rr5/9ZT8KPUqLEgPZxiSNkLnPC5dnmfTyswlofPGud+qmciYYr+cUZIvcFaYRG+Z6DM/HInX7w==  monitor@linux"
```

Alternatively, you can use the load-from-uri function to bring the public key from another source.

```
netapp::> security login publickey load-from-uri -username monitor -uri http://linux/id_rsa.pub
```

Verify creation

```
netapp::> security login publickey show -username monitor
UserName: monitor Index: 1
Public Key:
ssh-rsa
AAAAB3NzaC1yc2EAAAABIwAAAQEA5s4vVbwEO1sOsq7r64V5KYBRXBDb2I5mtGmt0+3p1jjPJrXx4/IPHFLalXAQkG7LhV5Dy
c5jyQiGKVawBYwxxSZ3GqXJNv1aORZHJEuCd0zvSTBGGZ09vra5uCfxkpz8nwaTeiAT232LS2lZ6RJ4dsCz+GAj2eidpPYMld
i2z6RVoxpZ5Zq68MvNzz8b15BS9T7bvdHkC2OpXFXu2jndhgGxPHvfO2zGwgYv4wwv2nQw4tuqMp8e+z0YP73Jg0T3jV8NYra
XO951Rr5/9ZT8KPUqLEgPZxiSNkLnPC5dnmfTyswlofPGud+qmciYYr+cUZIvcFaYRG+Z6DM/HInX7w==monitor@linux
```

Test access from the host

```
monitor@linux:~$ ssh 10.61.64.150
The authenticity of host '10.61.64.150 (10.61.64.150)' can't be established.
DSA key fingerprint is d9:15:cf:4b:d1:7b:a9:67:4d:b0:a9:20:e4:fa:f4:69.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '10.61.64.150' (DSA) to the list of known hosts.
netapp::>
```

7-Mode

The following section describes how to configure a 7-Mode appliance for Kerberos, LDAP, and NFSv4.

Configuring Kerberos in 7-Mode

In Data ONTAP running in 7-Mode, to configure Kerberized NFS for an appliance or a vFiler® unit, simply enter the appropriate CLI context and type the following command:

```
nfs setup
```

Then follow the prompts to configure Kerberized NFS. For this to work properly, verify that a CIFS server has been created in the domain. If CIFS has already been configured and the Microsoft option is selected, the appliance will use the existing CIFS credentials and information to set up Kerberos for NFS.

Example (if CIFS is already running):

```
filer> nfs setup
Enable Kerberos for NFS? y
The filer supports these types of Kerberos Key Distribution Centers (KDCs):

        1 - UNIX KDC
        2 - Microsoft Active Directory KDC
```



```
Enter the type of your KDC (1-2):  2
Kerberos now enabled for NFS.
NFS setup complete.
```

If CIFS has not been configured, the attempt will fail.

Example:

```
filer> nfs setup
Enable Kerberos for NFS? y
The filer supports these types of Kerberos Key Distribution Centers (KDCs):

        1 - UNIX KDC
        2 - Microsoft Active Directory KDC

Enter the type of your KDC (1-2):  2
Unable to setup Kerberos for NFS.  An Active Directory KDC was
selected, but CIFS has been setup to not use Kerberos.
NFS setup complete.
```

If Kerberized NFS is desired without the use of a CIFS server, then Kerberos must be set up manually using appliance options. For manual steps, please see the **File Access and Protocols Management Guide** for the desired version of 7-Mode.

To disable Kerberos for NFS:

```
filer> nfs setup
Kerberos is presently enabled for NFS.
Disable Kerberos for NFS? y
Kerberos now disabled for NFS.
NFS setup complete.
```

For client setup steps, see these previous sections in this document:

Configuring Linux clients

Configuring Solaris

Configuring LDAP in 7-Mode

To configure LDAP for use with Windows Active Directory 2008 R2, use the following options:

```
ldap.ADdomain                {DOMAIN.NETAPP.COM}
ldap.base                    {DC=domain,DC=netapp,DC=com}
ldap.base.group              {cn=users,DC=domain,DC=netapp,DC=com}
ldap.base.netgroup           {DC=domain,DC=netapp,DC=com}
ldap.base.passwd             {cn=users,DC=domain,DC=netapp,DC=com}
ldap.enable                  on
ldap.fast_timeout.enable     on
ldap.minimum_bind_level      sasl
ldap.name                    {username}
ldap.nssmap.attribute.gecos  name
ldap.nssmap.attribute.gidNumber gidNumber
ldap.nssmap.attribute.groupname cn
ldap.nssmap.attribute.homeDirectory unixHomeDirectory
ldap.nssmap.attribute.loginShell loginShell
ldap.nssmap.attribute.memberNisNetgroup msSFU30PosixMemberOf
ldap.nssmap.attribute.memberUid memberUid
ldap.nssmap.attribute.netgroupname cn
ldap.nssmap.attribute.nisNetgroupTriple nisNetgroupTriple
ldap.nssmap.attribute.uid    sAMAccountName
ldap.nssmap.attribute.uidNumber uidNumber
ldap.nssmap.attribute.userPassword unixUserPassword
ldap.nssmap.objectClass.nisNetgroup nisNetgroup
ldap.nssmap.objectClass.posixAccount User
ldap.nssmap.objectClass.posixGroup Group
ldap.passwd                  {******}
```



```
ldap.port                       389
ldap.retry_delay                120
ldap.servers                    {10.63.98.101}
ldap.servers.preferred          {10.63.98.101}
ldap.ssl.enable                 off
ldap.timeout                    20
ldap.usermap.attribute.unixaccount gecos
ldap.usermap.attribute.windowsaccount sAMAccountName
ldap.usermap.base
ldap.usermap.enable             on
```

In addition to the above, the /etc/nsswitch.conf file would need to be modified on the appliance so that LDAP is used for name lookups:

```
filer> rdfile /etc/nsswitch.conf
hosts: files       nis     dns
passwd: files      nis     ldap
netgroup: files    nis     ldap
group: files       nis     ldap
shadow: files      nis
```

Also, DNS would need to be configured so that LDAP is reachable by name:

```
filer> options dns
dns.cache.enable                on
dns.domainname                  domain.netapp.com
dns.enable                      on
dns.update.enable               off
dns.update.ttl                  24h
```

Verify that DNS settings are listed in /etc/rc:

```
filer> rdfile /etc/rc
#Auto-generated by setup Mon Apr 29 15:25:00 GMT 2013
hostname filer
ifconfig e0M `hostname`-e0M mtusize 1500
ifconfig e0a `hostname`-e0a mediatype auto flowcontrol full netmask 255.255.255.0 mtusize 1500
route add default 10.61.84.1 1
routed on
options dns.domainname domain.netapp.com
options dns.enable on
options nis.enable off
savecore
```

For information on configuring the domain controller to be an LDAP server, see the section entitled "Configuring the domain controller as an LDAP server."

For information on configuring clients to use LDAP, see the section entitled "Configuring the client to use LDAP."

Configuring NFSv4 in 7-Mode

To set up NFSv4 on a 7-Mode appliance, use the following steps:

1) Set the NFSv4 domain.

```
options nfs.v4.id.domain domain.netapp.com
```

2) Enable NFSv4.

```
options nfs.v4.enable on
```

For more information on NFSv4 in 7-Mode, see TR-3580.

Setup Checklists

Following is a list of condensed setup steps to set up Kerberos and LDAP for use with clustered Data ONTAP. These steps do not cover explanations or describe anything beyond the simple “how to” of the specified task. These steps can be used as a checklist for setup and configuration verification. This section is intended for audiences that already understand the nuances of this solution and for audiences that need to get the solution working quickly.

NetApp highly recommends reviewing the entire document for enterprise production solutions to fully understand the hows and whys of this setup.

Within the Setup Checklists there are references to previous portions of this document, denoted by a linked [?]. Simply click on the link to be redirected to the section in question.

Table 45) Pre-setup Steps

Completed	Step
	Gather DNS information for the domain (DNS domain name, IPs of name servers, and so on).
	Gather cluster licenses.
	Plan data layout strategy.
	Plan security strategy.
	Create necessary scripts (optional).
	Download and install software packages for the NFS clients.
	Plan for/obtain domain administrator access for machine account/DNS settings.
	Verify that time services work properly and that all pieces are within a 5-minute time skew. Take time zones/daylight savings time into account.
	Plan the naming convention for machine accounts/SPN.
	Verify network connectivity between KDC and clients.
	Plan the kind of DNS load balance that will be used for data LIFs.

Cluster Configuration

Completed	Step

Completed	Step
	Configure DNS for the SVM using the `dns` commands. [?]
	Confirm that NFS is licensed and enabled. [?]
	Create the Kerberos realm for the SVM. [?]
	Create a CIFS server (optional). [?]
	Configure export policies and export policy rules. [?]
	Create and mount data volumes. [?]
	Enable Kerberos on the data LIF. [?]
	Verify name mapping for SPN. Create unix-user or name-mapping rule as needed. [?]

Domain Controller Configuration

Completed	Step
	Allow DES encryption in the domain. [?]
	Create the machine objects for the NFS clients and their keytabs. [?]
	Allow DES encryption for the machine accounts. [?]
	Add the NFS client to DNS in Windows AD. [?]
	Add the SVM data LIFs to DNS in Windows AD. [?]
	Add SRV records to DNS (specifically the Kerberos-master record). [?]
	Verify the SPNs. [?]
	Modify the NFS server machine account to allow DES only. [?]

Linux Client Configuration

Completed	Step
	Verify that the host name is set. [?]

Completed	Step
	Verify that software packages are installed for krb5/ldap. [?]
	Verify that the date and time are correct. [?]
	Verify that DNS is configured and working properly. [?]
	Verify that the NFSv4 domain is set (if using NFSv4). [?]
	Verify that secure NFS is allowed. [?]
	Configure /etc/krb5.conf. [?]
	Create the /etc/krb5.keytab file. [?]
	Restart the Kerberos service to apply the configuration. [?]
	Verify that NFS services are started (SUSE). [?]
	Set the environment variable to allow MD5 (RHEL 6.4 only). [?]
	Configure SSSD. [?]

References

The following references were used in this TR.

- Fedora SSSD documentation
  fedorahosted.org/sssd/
- RHEL SSSD documentation
  access.redhat.com/site/documentation/en-US/Red_Hat_Enterprise_Linux/5/html/Deployment_Guide/SSSD.html
- Microsoft TechNet
  technet.microsoft.com
- IETF
  www.ietf.org/
- MIT Kerberos
  web.mit.edu/kerberos/
- MSDN blogs
  blogs.msdn.com/
- Oracle documentation for Kerberos
  docs.oracle.com/cd/E23824_01/html/821-1456/setup-148.html
- Linux DIE.net
  linux.die.net/

- IBM
  publib.boulder.ibm.com/infocenter/zos/v1r12/index.jsp?topic=%2Fcom.ibm.zos.r12.euvmd00%2Feuva6a001200.htm
- Red Hat Bugzilla
  bugzilla.redhat.com
- Sourceforge
  sourceforge.net/
- Softerra
  www.ldapadministrator.com/
- Wikipedia
  wikipedia.org
- TR-3580: NFSv4 Enhancements and Best Practices
  media.netapp.com/documents/tr-3580.pdf
- TR-4067: NFS Implementation Guide in Clustered Data ONTAP
  media.netapp.com/documents/tr-4067.pdf
- TR-3457: Unified Windows and UNIX Authentication Using Microsoft Active Directory Kerberos
  media.netapp.com/documents/tr-3457.pdf
- TR-4847: Clustered Data ONTAP Networking Best Practice Guide
  media.netapp.com/documents/tr-4847.pdf
- TR-3458: Unified Windows and UNIX Authorization Using Microsoft Active Directory LDAP as a Directory Store
  media.netapp.com/documents/tr-3458.pdf

Version History

Version	Date	Document Version History
Version 1.0	May 2012	Initial release
Version 2.0	June 2013	Major updates to all sections
Version 2.1	October 2013	Major updates to some sections

Acknowledgements

Our thanks to Andy Adamson of NetApp and Kyle Payne of CDW.

Refer to the Interoperability Matrix Tool (IMT) on the NetApp Support site to validate that the exact product and feature versions described in this document are supported for your specific environment. The NetApp IMT defines the product components and versions that can be used to construct configurations that are supported by NetApp. Specific results depend on each customer's installation in accordance with published specifications.

NetApp provides no representations or warranties regarding the accuracy, reliability, or serviceability of any information or recommendations provided in this publication, or with respect to any results that may be obtained by the use of the information or observance of any recommendations provided herein. The information in this document is distributed AS IS, and the use of this information or the implementation of any recommendations or techniques herein is a customer's responsibility and depends on the customer's ability to evaluate and integrate them into the customer's operational environment. This document and the information contained herein may be used solely in connection with the NetApp products discussed in this document.

Go further, faster®

www.netapp.com