



Technical Report

Secure Unified Authentication

Kerberos, NFSv4, and LDAP in ONTAP

Justin Parisi, NetApp
May 2020 | TR-4073

Attention: This document is being deprecated. The content will remain intact, but will not be updated or maintained. Instead, use the following TRs:

- [TR-4067: Network File Systems \(NFS\) in NetApp ONTAP](#)
- [TR-4616: NFS Kerberos in ONTAP](#)
- [TR-4835: How to Configure LDAP in ONTAP](#)

Version History

Version	Date	Document Version History
Version 4.5	May 2020	Final revision; no updates or maintenance.
Version 4.4.2	August 2017	Minor updates for 9.2GA
Version 4.4.1	February 2017	Minor updates for 9.1GA
Version 4.4	October 2016	Updates for ONTAP 9.1 added
Version 4.3	July 2016	Updates for ONTAP 9.0 added
Version 4.2	May 2016	Updates for 8.3.2 added
Version 4.1	July 2015	Major updates; ONTAP 8.3.1 information added
Version 4.0	December 2014	Major updates; ONTAP 8.3 information added
Version 3.0	August 2014	Major updates
Version 2.1	October 2013	Major updates to some sections
Version 2.0	June 2013	Major updates to all sections
Version 1.0	May 2012	Initial release

This document is being deprecated. The content will remain intact, but will not be updated or maintained. Instead, use the following TRs.

- [TR-4067: Network File Systems \(NFS\) in NetApp ONTAP](#)
- [TR-4616: NFS Kerberos in ONTAP](#)
- [TR-4835: How to Configure LDAP in ONTAP](#)

TABLE OF CONTENTS

Version History	2
1 Introduction	9
1.1 Overview	9
1.2 Intended Audience	9
2 Kerberos Overview	10
2.1 Kerberos Terminology.....	10
2.2 Supported Encryption Types.....	11
2.3 Supported Kerberos Security Modes	12
2.4 How Kerberos Authentication Works	12
2.5 Client-Side Kerberos Commands.....	13
2.6 Client-Side Kerberos Files	15
3 Benefits of Using Kerberized NFS	15
4 Microsoft Windows Active Directory as the Key Distribution Center (KDC)	16
4.1 Setting Up Kerberized NFS.....	16
5 LDAP Overview	44
5.1 LDAP Using Microsoft Windows AD for Identity Management	48
5.2 LDAP Using Red Hat Directory Services for Identity Management.....	79
5.3 Setting Up LDAP Clients.....	87
5.4 How SecD Queries LDAP in Data ONTAP.....	106
5.5 Using SecD to Troubleshoot External Name Service Queries	106
5.6 Optimizing LDAP Searches: Best Practices.....	110
5.7 Configuring the Data ONTAP System to Use LDAP	111
5.8 Setting Up NFSv4	144
6 Configuration Steps	146
6.1 DNS Configuration	146
6.2 Kerberos Configuration—Manual Keytab Creation	153
6.3 Kerberos Configuration—Domain Join Method.....	171
6.4 LDAP Configuration Steps	187
6.5 NFSv4 Configuration.....	204

LIST OF BEST PRACTICES

Best Practices 1: Quick Step Setup Guides (see next: Best Practices 2)..... 10

Best Practices 2: Specifying Encryption Types (see next: Best Practices 3)..... 11

Best Practices 3: Use LDAP Instead of NIS (see next: Best Practices 4) 16

Best Practices 4: Data LIFs and Kerberos (see next: Best Practices 5)..... 16

Best Practices 5: Setting permitted encyptes (see next: Best Practices 6) 17

Best Practices 6: Kerberos Use with NFSv3 (see next: Best Practices 7)..... 18

Best Practices 7: Kerberos/Multiple Data LIFs/Same SPN/DNS LB (see next: Best Practices 8) 18

Best Practices 8: Client SPN Behavior (see next: Best Practices 9) 21

Best Practices 9: Using nm-switch and ns-switch (see next: Best Practices 10)..... 23

Best Practices 10: LIF Communication with Name Services (see next: Best Practices 11) 24

Best Practices 11: SPN Length (see next: Best Practices 12)..... 24

Best Practices 12: Machine Account OU Specification (see next: Best Practices 13)..... 24

Best Practices 13: Encryption Type Information (see next: Best Practices 14) 24

Best Practices 14: Name-Mapping Rule Limits (see next: Best Practices 15)..... 25

Best Practices 15: Kerberos NFS Clients and DNS (see next: Best Practices 16)..... 25

Best Practices 16: Prevent Lookup Failures with Multiple LIFs (see next: Best Practices 17)..... 26

Best Practices 17: On-Box DNS (see next: Best Practices 18) 26

Best Practices 18: Kerberos Encryption Type Recommendation (see next: Best Practices 19)..... 28

Best Practices 19: Disabling DES (see next: Best Practices 20)..... 28

Best Practices 20: Tools to Configure Machine Accounts (see next: Best Practices 21) 31

Best Practices 21: SPN Considerations for RHEL/CentOS 6.x (see next: Best Practices 22)..... 32

Best Practices 22: Kerberos Interface Command Usage (see next: Best Practices 23)..... 32

Best Practices 23: Kerberos Interface Command (see next: Best Practices 24) 33

Best Practices 24: NFS Kerberos SPN with CIFS Servers (see next: Best Practices 25) 35

Best Practices 25: Kerberos Setup with Solaris (see next: Best Practices 26)..... 37

Best Practices 26: NFS Client OS Recommendation (see next: Best Practices 27) 37

Best Practices 27: NFS Client OS with AES Encryption (see next: Best Practices 28) 41

Best Practices 28: Timeout Value for rpcgssd (see next: Best Practices 29) 42

Best Practices 29: LDAP Application for NFS Clients (see next: Best Practices 30)..... 45

Best Practices 30: NFSv4 ID Mapping: The Nobody User (see next: Best Practices 31)..... 46

Best Practices 31: Schema Extension Considerations (see next: Best Practices 32) 50

Best Practices 32: gidNumber Recommendations (see next: Best Practices 33)..... 53

Best Practices 33: UID/GID Selection Considerations (see next: Best Practices 34) 54

Best Practices 34: Considerations For Enabling Extended GIDs (see next: Best Practices 35) 56

Best Practices 35: Multiple Domains/Trusts: UNIX Identities (see next: Best Practices 36)..... 57

Best Practices 36: Multiple Domains/Trusts: Multiprotocol NAS (see next: Best Practices 37).....	57
Best Practices 37: Multiple Domains/Trusts: Name Map Search (see next: Best Practices 38).....	58
Best Practices 38: Specifying external services in namemap (see next: Best Practices 39).....	61
Best Practices 39: Data ONTAP Version: Netgroups (see next: Best Practices 40).....	68
Best Practices 40: Netgroups and DNS (see next: Best Practices 41).....	76
Best Practices 41: Netgroup Definition in Export Policy Rules (see next: Best Practices 42).....	78
Best Practices 42: Netgroup.byhost Considerations (see next: Best Practices 43).....	78
Best Practices 43: DNS Considerations for Use with SSSD (see next: Best Practices 44).....	79
Best Practices 44: Client PAM Configuration Recommendation (see next: Best Practices 45).....	90
Best Practices 45: LDAP Client Base DN Recommendations (see next: Best Practices 46).....	96
Best Practices 46: Automount Recommendations (see next: Best Practices 47).....	104
Best Practices 47: LDAP Client Schema Recommendation (see next: Best Practices 48).....	112
Best Practices 48: User/Computer Objects + Primary Groups (see next: Best Practices 49).....	112
Best Practices 49: RFC2307bis and Active Directory LDAP (see next: Best Practices 50).....	118
Best Practices 50: RFC2307bis and Active Directory Schema (see next: Best Practices 51).....	118
Best Practices 51: LDAP Group Attribute Best Practice (see next: Best Practices 52).....	119
Best Practices 52: UID/GID Configuration with Multiple Domains (see next: Best Practices 53).....	121
Best Practices 53: When to Create Custom Schemas (see next: Best Practices 54).....	122
Best Practices 54: LDAP Client Configuration with CIFS Servers (see next: Best Practices 55).....	125
Best Practices 55: SRV Record Lookups for LDAP Servers (see next: Best Practices 56).....	128
Best Practices 56: NS-Switch Best Practice (see next: Best Practices 57).....	144
Best Practices 57: File-Only Mode: Local UNIX Users and Groups (see next: Best Practices 1).....	144

LIST OF CONFIGURATION STEPS

Configuration Steps 1) Using the <code>nismap</code> command to create a netgroup in AD LDAP.....	71
Configuration Steps 2) Adding NFS client to Windows DNS (GUI).....	146
Configuration Steps 3) Adding NFS client to Windows DNS (<code>dnscmd</code>).....	150
Configuration Steps 4) Creating SRV records for Kerberos-Master.....	150
Configuration Steps 5) Creating SRV records for Kerberos-Master (<code>dnscmd</code>).....	153
Configuration Steps 6) Allowing DES encryption types in Windows 2008 R2 and later.....	153
Configuration Steps 7) Creating machine accounts in Active Directory (GUI).....	154
Configuration Steps 8) Creating machine accounts in Active Directory (<code>dsadd</code>).....	156
Configuration Steps 9) Creating machine accounts in Active Directory (Windows PowerShell).....	157
Configuration Steps 10) Modifying the NFS server machine account for DES_CBC_MD5 (Attributes Editor).....	157
Configuration Steps 11) Modifying the NFS server machine account for DES_CBC_MD5 (ADSI edit).....	160
Configuration Steps 12) Modifying the NFS server machine account for DES_CBC_MD5 (import using <code>ldifde</code>).....	163
Configuration Steps 13) Creating machine accounts for use with DES in Active Directory (Windows PowerShell).....	164
Configuration Steps 14) Modifying the NFS machine account to use/support AES (Attributes Editor).....	164
Configuration Steps 15) Creating a keytab file.....	168

Configuration Steps 16) Allowing DES encryption types in Windows 2008 R2 and later.	171
Configuration Steps 17) Modifying the NFS server machine account for DES_CBC_MD5 (Attributes Editor).	172
Configuration Steps 18) Modifying the NFS server machine account for DES_CBC_MD5 (ADSI edit).	175
Configuration Steps 19) Modifying the NFS server machine account for DES_CBC_MD5 (import using Ldifde).	178
Configuration Steps 20) Creating machine accounts for use with DES in Active Directory (Windows PowerShell). ...	179
Configuration Steps 21) Modifying the NFS machine account to use/support AES (Attributes Editor).	180
Configuration Steps 22) Using local UNIX users for authentication.	184
Configuration Steps 23) Configuring LDAP users for use with authentication.	184
Configuration Steps 24) Using Active Directory Users and Computers to Modify User/Computer Accounts.	185
Configuration Steps 25) Using PowerShell to Modify User/Computer Accounts.	186
Configuration Steps 26) Configuring krb-unix name mapping rules in the SVM for NFS service principals.	186
Configuration Steps 27) Configuring krb-unix name mapping rules in the SVM for client principals.	186
Configuration Steps 28) Setting UID/GID in Active Directory LDAP (GUI).	187
Configuration Steps 29) Setting UID/GID in Active Directory LDAP (ldifde).	189
Configuration Steps 30) Setting UID/GID in Active Directory LDAP (PowerShell).	190
Configuration Steps 31) Mapping users with LDAP.	190
Configuration Steps 32) Creating a container object with ADSI Edit.	191
Configuration Steps 33) Netgroup entry created using ADSI Edit and nisObject class.	193
Configuration Steps 34) Netgroup entry created using ADSI Edit and nisNetgroup class.	195
Configuration Steps 35) Creating netgroup.byhost entry.	199
Configuration Steps 36) Configuring LDAP over SSL in Data ONTAP.	202
Configuration Steps 37) Configuring the Data ONTAP system for NFSv4.x (CLI).	204
Configuration Steps 38) Configuring MIT Kerberos.	216
Configuration Steps 39) Configuring an NFS client to Use Kerberos with "realm join."	228
Configuration Steps 40) Configuring an NFS client to Use Kerberos with "net ads join"	233
Configuration Steps 41) Configuring Kerberos in ESXi 6.0	238

LIST OF TABLES

Table 1) Supported encryption types in Data ONTAP.	11
Table 2) Examples of nlttest in Windows trusted domains.	58
Table 3) Name mapping/default user considerations for multiprotocol NAS access	61
Table 4) Object class types for NIS objects in Active Directory.	68
Table 5) NIS object terminology.	68
Table 6) Caches and time to live (TTL).	76
Table 7) /etc/sss/sss.conf file options.	97
Table 8) Default schemas available in Data ONTAP.	111
Table 9) SecD scope definitions.	113
Table 10) Sample RFC-2307bis schema for LDAP servers in Active Directory:	118
Table 11) Example of multiple DN configuration:	120

Table 12) LDAP attributes in Data ONTAP operating in 7-Mode mapped to Data ONTAP	123
Table 13) Hidden options for LDAP in Data ONTAP operating in 7-Mode.	124
Table 14) Sample bind DN formats.	125
Table 15) Bind authentication order versus minimum bind level.	126
Table 16) Common RDN values in LDAP.....	128
Table 17) Search scope types.....	132
Table 18) Numeric representation of search scopes in secd logs.	135
Table 19) Supported LDAP bind levels.	140
Table 20) Services for NFSv4.	146
Table 21) Encyptes.	206
Table 22) Valid msDS-SupportedEncryptionTypes attribute values.....	210
Table 23) Kerberos packets.	214
Table 24) Kerberos errors from network captures.	214
Table 25) Kerberos terminology from CentOS.org and IBM.com.	215
Table 26) Common mount issues with Kerberized NFS.....	220
Table 27) Common read/write issues with Kerberized NFS.	220
Table 28) 7-Mode to Data ONTAP command translation for authentication.....	224
Table 29) 7-Mode to Data ONTAP log translation: NAS-specific logs.....	226
Table 30) What each SecD command does in Data ONTAP 8.3.x.....	227
Table 31) Presetup steps.	244

LIST OF FIGURES

Figure 1) Kerberos workflow between client, KDC, and NFS server on NetApp storage.	13
Figure 2) DES enabled.....	29
Figure 3) Example of addRequest in packet trace.....	34
Figure 4) User without gidNumber set in LDAP.....	53
Figure 5) Domain trust example with external LDAP server in separate forest.	57
Figure 6) Trusted domain using global catalog searches.	59
Figure 7) Modifying global catalog attributes to replicate.	60
Figure 8) Example of adding multiple UIDs to LDAP in Active Directory.	62
Figure 9) Example of msDs-PrincipalName field.....	65
Figure 10) Server for NIS MMC.....	70
Figure 11) Example of “hosts” netgroup created in AD LDAP.	72
Figure 12) Netgroup properties in AD LDAP.	73
Figure 13) Connecting to default naming context.....	73
Figure 14) Sample LDAP filter for a netgroup lookup using ldp.exe	75
Figure 15) LDAP schema structure examples.....	111
Figure 16) SRV record lookup for Active Directory domain.	128
Figure 17) LDAP DNs folder structure.....	129

Figure 18) LDAP DN containers.....	133
Figure 19) Packet capture of LDAP signing and sealing.....	141
Figure 20) LDAP search packet comparison: sealed versus unsealed.....	141

The following content is no longer being maintained, nor is it being updated. It may be out of date, as the last material update to this TR was in 2017. This TR's information is deprecated, but is being retained for archival purposes and reference.

The following technical reports should be referenced instead.

- TR-4067: NFS Best Practices and Implementation Guide
www.netapp.com/us/media/tr-4067.pdf
- TR-4523: DNS Load Balancing in ONTAP
www.netapp.com/us/media/tr-4523.pdf
- TR-4616: NFS Kerberos in ONTAP
www.netapp.com/us/media/tr-4616.pdf
- TR-4668: Name Services Best Practice Guide
www.netapp.com/us/media/tr-4668.pdf
- TR-4835: How to Configure LDAP in ONTAP
www.netapp.com/us/media/tr-4835.pdf

DEPRECATED

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 make sure of the confidentiality of the data (no inappropriate access to data). Kerberos uses hashing techniques to make sure of the integrity of the data (no one modifies the data who isn't allowed to).

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. Doing so 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 ease of use and cost perspectives.

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 Kerberos based on Microsoft Active Directory. However, Kerberos 5i (integrity) support was added in Data ONTAP 8.3. Kerberos 5p (privacy) is was added to ONTAP 9.0.

Kerberos to KDCs over IPv6 support was added in Data ONTAP 8.3.

Note: This document mainly covers Data ONTAP setup and interaction. However, the external client and server steps apply to all modes of Data ONTAP. Additionally, there are steps specific to [Z-Mode](#) for storage system setup for LDAP and Kerberos in the appendix of this document.

1.1 Overview

The following document covers the use of a System Security Services Daemon ([SSSD](#)) LDAP client 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 because it enables centralized management for all environments. The following environments were used:

- Windows Active Directory domains (Windows 2008R2 and Windows 2012R2)
- Various Linux clients
These clients were built from scratch and had no preexisting configuration.
- Data ONTAP 8.3 storage virtual machine (SVM)

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 Data ONTAP for NAS storage. A working knowledge of NFS, Kerberos, and Windows Active Directory and administrator access to these environments are assumed.

Best Practices 1: Quick Step Setup Guides (see next: Best Practices 2)

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 this document, 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

The following section describes key terminology when describing Kerberos processes. This section's intent is to help clarify terms that might be unfamiliar to storage administrators.

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 standard 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 [Active Directory 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:



primary/instance@REALM

The primary:

The primary part 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 part is optional in the 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 of principals:

```
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 an 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 Supported Encryption Types

Data ONTAP supports NFS Kerberos with specific encryption types, depending on the operating mode and OS version being used.

Best Practices 2: Specifying Encryption Types (see next: Best Practices 3)

To make sure a client uses the appropriate encryption type, limit the valid encryption types on the object principal or keytab file rather than in the krb5.conf file. This approach is much more scalable in large enterprise environments and makes sure that the client can leverage stronger encryption types when supported. For more information, see the section in this document on [setting up the NFS client principals](#).

The following table shows the supported encryption type based on OS and mode. These types are for NFS Kerberos only and do not cover CIFS Kerberos support.

Table 1) Supported encryption types in Data ONTAP.

Data ONTAP Version and Mode	Supported Encryption Type
Data ONTAP 7-Mode 7.x and later	DES and 3DES only Note: (RC4-HMAC works, but no official support)
Data ONTAP 8.2.x and earlier (clustered)	DES and 3DES

Data ONTAP 8.3 and later	AES (128 and 256 bit), DES, and 3DES
--------------------------	--------------------------------------

2.3 Supported Kerberos Security Modes

In addition to the concept of encryption types, there is also a level of security and integrity checking in Kerberos to help prevent “man in the middle” attacks. The following table shows which levels of Kerberos security modes are supported in various versions of ONTAP. The security modes for Kerberos are configured on the clients and KDCs. Export policy rules are then configured to allow specific security modes.

Table 2) Supported Kerberos security modes in Data ONTAP.

Data ONTAP Version and Mode	Supported Kerberos Security Mode
Data ONTAP 7-Mode 7.x and later	krb5, krb5i, krb5p
Data ONTAP 8.2.x and earlier (clustered)	krb5
Data ONTAP 8.3.x	krb5,krb5i
ONTAP 9	krb5, krb5i, krb5p

2.4 How Kerberos Authentication Works

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

KDCs such as Windows Active Directory maintain 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. It can also be trusted to authenticate 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.

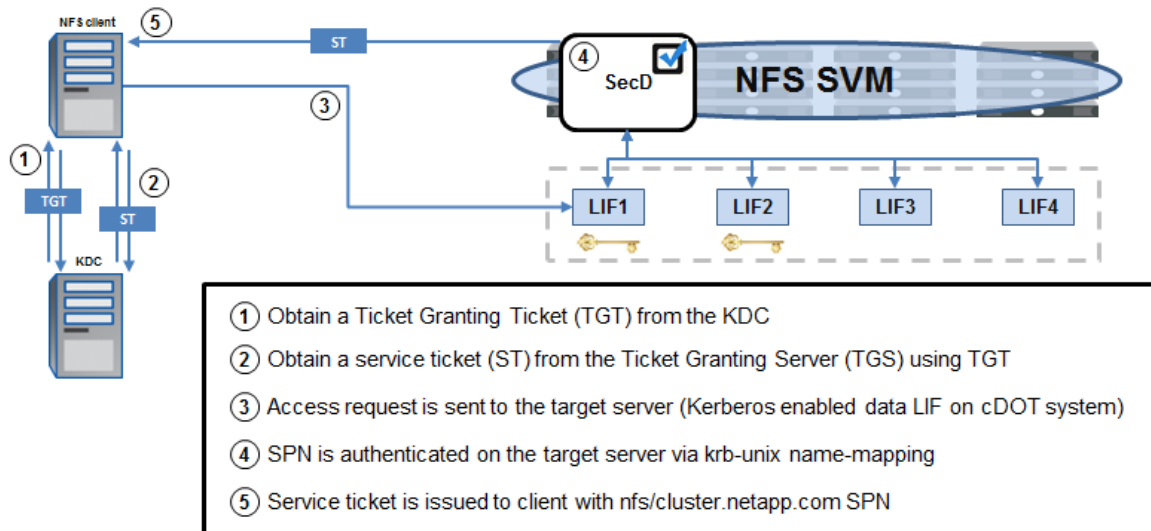
From the KDC, the encrypted TGT is sent to the principal. The principal decrypts the TGT response by using the secret key obtained from the password or from the keytab. The principal then requests authentication to the NFS server (in this case, Data ONTAP) by presenting the NFS server principal along with the encrypted TGT to the ticket-granting server (TGS). The TGS then issues a ticket for the NFS server. The ticket 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 because 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 shows the Kerberos workflow between the client, NFS server, and KDC.

In 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 using 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 searches for SPNs in a specific order, listed in the [rpc.gssd](#) man pages as well as the following table.

Kerberos SPN Types

<code>root/host.domain.com:</code>	used by the NFS client for mount requests
<code>nfs/host.domain.com:</code>	required to be used by the NFS server (for example, <code>nfs/cluster.domain.com</code>)
<code>host/host.domain.com:</code>	used by the NFS client, usually for third-party applications such as SSSD

Any of the preceding types can be used to create a principal in Active Directory, but only one is required. This document covers the use of only the root SPN in most cases, but other SPNs can be leveraged if desired. Some client operating systems require nonroot SPNs to leverage Kerberos, such as Red Hat 5.x and earlier versions.

2.5 Client-Side Kerberos Commands

Table 3) Client-side Kerberos commands.

Command	Description
---------	-------------

kinit	<p>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.</p> <p>Example:</p> <pre>[root@nfsclient /]# kinit administrator Password for administrator@DOMAIN.NETAPP.COM:</pre>
klist	<p>List contents of the Kerberos ticket cache and keytabs. Default is to list the cached credentials. To specify the keytab file, use the <code>-k</code> option. To show encryption types, use <code>-e</code>. To show timestamps, use <code>-t</code>.</p> <p>Example:</p> <pre>[client] # klist -kte Keytab name: FILE:/etc/krb5.keytab KVNO Timestamp Principal ----- 4 05/16/13 11:57:56 root/nfsclient.domain.netapp.com@DOMAIN.NETAPP.COM (des-cbc-crc) 4 05/16/13 11:57:56 root/nfsclient.domain.netapp.com@DOMAIN.NETAPP.COM (des-cbc-md5) 4 05/16/13 11:57:56 root/nfsclient.domain.netapp.com@DOMAIN.NETAPP.COM (aes256-cts-hmac-sha1-96) 4 05/16/13 11:57:56 root/nfsclient.domain.netapp.com@DOMAIN.NETAPP.COM (aes128-cts-hmac-sha1-96)</pre>
ktutil	<p>Used to manage the client-side keytab. Running <code>ktutil</code> starts an application that allows the reading (<code>rkt</code>) and writing (<code>wkt</code>) of keytab files.</p> <p>Example:</p> <pre>[root@nfsclient /]# ktutil ktutil: rkt /etc/krb5.keytab ktutil: list slot KVNO Principal ----- 1 2 root/nfsclient.domain.netapp.com@DOMAIN.NETAPP.COM</pre>
kdestroy	<p>Destroys a ticket cache.</p>
kvno	<p>Prints the key versions of Kerberos principals. The <code>kvno</code> must match across principals. Useful for troubleshooting SPNs. When using this command, a Kerberos session must be established between the client and KDC.</p> <p>Example:</p> <pre>[root@nfsclient /]# kvno root/nfsclient.domain.netapp.com root/nfsclient.domain.netapp.com@DOMAIN.NETAPP.COM: kvno = 2</pre>

2.6 Client-Side Kerberos Files

Table 4) Client-side Kerberos files.

File	Description
<code>/etc/krb5.conf</code>	<p>This file must exist on clients wanting to use Kerberos. The file consists of several sections that are used in ticket services.</p> <ul style="list-style-type: none">• <code>[libdefaults]</code> sets defaults for Kerberos on your system; for example, default realm, default ticket lifetime, encryption types.• <code>[realms]</code> tells where to find the KDCs for each realm.• <code>[instancemapping]</code> maps client principals properly (for things such as cron jobs that require a special principal).• <code>[domain_realm]</code> maps domains to realms.• <code>[logging]</code> tells Kerberos where and how to log errors.• <code>[appdefaults]</code> lists default settings for outgoing Kerberized network connection applications and for incoming portal mode connections. <p>See the following for more about <code>/etc/krb5.conf</code>: http://web.mit.edu/kerberos/krb5-1.5/krb5-1.5/doc/krb5-admin/krb5.conf.html</p>
<code>/etc/krb5.keytab</code>	<p>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 fails. It should only exist on the local server's disk.</p> <p>See the following for more information about the <code>krb5.keytab</code> file: http://web.mit.edu/kerberos/krb5-1.5/krb5-1.5/doc/krb5-install/The-Keytab-File.html</p>

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 after you've been authenticated.

For example, if someone buys a subway ticket, he is allowed through the turnstile (authentication). But after that person is inside the station, he might 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. Because 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 Data ONTAP 8.2.x and earlier (clustered). In Data ONTAP 8.3 and later, the `AUTH_SYS` and `AUTH_GSS` limits can be raised to 1,024 for both `AUTH_SYS` and `AUTH_GSS`. See

[TR-4067: NFS Best Practice and Implementation Guide](#) for more information on extending the auxiliary group limits for NFS in Data ONTAP.

NFS servers in ONTAP require a Kerberos service principal name (SPN) to UNIX UID mapping to allow a Kerberos principal access to exported data. In Data ONTAP (clustered), a specific name mapping rule type of `krb-unix` was added to allow manual configuration of this.

Note: 7-Mode does not allow custom name mapping of Kerberos SPNs to users.

Best Practices 3: Use LDAP Instead of NIS (see next: Best Practices 4)

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. Therefore, 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 Microsoft Windows Active Directory as the Key Distribution Center (KDC)

This section describes setting up Kerberized NFS using Data ONTAP as the NFS server and storage. The KDC in this example is Microsoft Windows Active Directory 2008 R2, but many of the same steps can be used for Windows 2003/2003R2 and Windows 2012 and beyond.

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 using `krb5` security. The appendix of this document covers [7-Mode-specific configuration steps](#) and supported features.

Note: [Quick Step Setup](#) steps can be found at the end of this document.

Configuring the Data ONTAP System for Kerberos

To configure a 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 in the format of `nfs/hostname@REALM`; `nfs` designates the principal as a service used for NFS), and a principal is created in the KDC. In the case of Microsoft Windows Active Directory, the principal is a machine account.

Best Practices 4: Data LIFs and Kerberos (see next: Best Practices 5)

To properly support LIF migration, HA takeover, and pNFS with Kerberos, Kerberos must be enabled for all data LIFs in the SVM.

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

- DNS configured (A, AAAA, or CNAME and PTR record)
- NFS licensed and configured
- Kerberos realm created
- CIFS server created (optional)
- Data volumes created and mounted into the namespace of the storage virtual machine (SVM)

- Export policies and rules configured
- Kerberos enabled on data LIFs
- Valid name mapping for the NFS SPNs exists

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 and enable DNS, use the following command:

```
cluster::> dns create -vserver vs0 -domains [search.domain1.com,search.domain2.com]
-name-servers [IP.address.1,IP.address.2] -state enabled
```

Note: Up to three DNS name servers and six search domains are allowed.

Configuring NFS

This document assumes that NFS has been licensed and configured on the SVM. If this has not taken place, see [TR-4067: NFS Best Practices and Implementation Guide](#).

Allowed Encryption Types

Data ONTAP 8.3 and later introduced [AES encryption type](#) support for Kerberos. To accommodate this, a new NFS server option (`permitted-enc-types`) was included to allow AES to be used:

```
cluster::> nfs server modify -permitted-enc-types
des      des3      aes-128 aes-256
```

```
[permitted-enc-types <NFS Kerberos Encryption Type>, ...] - Permitted Kerberos Encryption Types
This optional parameter specifies the permitted encryption types for Kerberos over NFS. The
default setting is des,des3,aes-128,aes-256.
```

Best Practices 5: Setting permitted encypes (see next: Best Practices 6)

Disable DES encryption types if AES is the only encryption used. By default, all encryption types are enabled. If reverting to a Data ONTAP version prior to 8.3, reenale DES encryption types to make sure of backward compatibility.

Configuring a Kerberos Realm

A Kerberos realm is needed so that the cluster knows how to format Kerberos ticket requests. Doing so 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 using DNS SRV records. For more information, see the section "[Domain Controller Redundancy and Replication](#)."

Creating a CIFS Server (Optional)

Creating a CIFS server is not a necessary step, but it can affect how Kerberos is configured. To create a CIFS server, use NetApp OnCommand® System Manager. For information on how a CIFS server can affect Kerberos configuration, see the section [“Configuring the Domain Controller.”](#)

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. If no rule is added to a policy, that lack effectively denies access to all clients. 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, krb5i, krb5p, sys”) and/or “any.”

Best Practices 6: Kerberos Use with NFSv3 (see next: Best Practices 7)

For NFSv3 mounts that use network lock manager, the export policy rule must include “sys” in addition to “krb5” to allow successful mounts in versions of ONTAP prior to 8.2P6. [See bug 756081 for details.](#) Additionally, the protocol portion of the export policy rule must allow NFS access. For more information on export policies and rules, see [TR-4067](#), which covers NFS implementation in Data ONTAP.

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. If this is not done, no export path is provided to clients. For information on creating volumes and mounting them, see [TR-4067](#), which covers NFS implementation in Data ONTAP.

Enabling Kerberos on a Data LIF

To use Kerberos for NFS, Kerberos must be enabled on a data LIF in the SVM. When Kerberos is enabled, the SPN is defined and a principal is created on the KDC defined in the realm configuration. To enable Kerberos in Data ONTAP before 8.3, 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 using the CLI. The user name provided must have rights to create objects in the computer’s organizational unit (OU) in Active Directory. This user can be a [domain administrator or a user who has had rights delegated](#) to manage that OU. In ONTAP 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 results in the command failing.

Best Practices 7: Kerberos/Multiple Data LIFs/Same SPN/DNS LB (see next: Best Practices 8)

Ideally, Kerberos should be enabled [on numerous data LIFs on multiple nodes in the cluster 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 zone 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/kerberos.domain.netapp.com@DOMAIN.NETAPP.COM
vs0      data2      10.61.92.37  enabled  nfs/kerberos.domain.netapp.com@DOMAIN.NETAPP.COM
```

```
vs0      vs_mgmt      10.61.92.36  disabled -
3 entries were displayed.
```

Note: If an AD KDC goes down for any reason, the cluster leverages other AD KDCs in the domain.

In Data ONTAP 8.3 and later, the Kerberos configuration commands changed. Use the following command to enable Kerberos for a data LIF in 8.3 and later:

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

The following command is also a valid method to enable Kerberos:

```
cluster::> nfs kerberos interface enable -vserver vs0 -lif data -kerberos enabled
-spns nfs/kerberos.domain.netapp.com@DOMAIN.NETAPP.COM
```

Multiple SPNs on Data LIFs in the Same SVM

As a best practice, NetApp recommends enabling Kerberos on multiple data LIFs to provide redundancy to Kerberized mount points, as well as using the same SPN on all data LIFs if possible. Doing so allows the data LIFs to all share the same machine accounts and Kerberos keyblocks, as well as avoiding exposure to issues in which keyblocks could get overwritten on SPNs that have identical names for the first 15 characters. Active Directory allows only 15 characters for machine account names, so when a SPN is created, the original machine account is reused and access is denied on the data LIF where Kerberos was originally enabled.

Note: This issue does not occur on non-Windows KDCs and is fixed in Data ONTAP 8.3.x and later.

It is possible to use multiple different SPNs on data LIFs provided the following rules are followed in Data ONTAP 8.2.x and earlier:

- SPNs are unique within the first 15 characters (including the nfs/ portion).
Example: nfs/uniquestpns1.domain.com
- [Manual Kerberos configuration](#) is done using multiple machine accounts on the Windows KDC as if it were a non-Windows KDC.
- [Use of CNAMEs](#) and [manually added SPNs](#) to create aliases to unique SPNs on the same machine account.

Using multiple machine accounts with multiple unique SPNs avoids having the Kerberos keyblocks being overwritten. However, if there is no specific reason to use multiple unique SPNs on data LIFs, then configure the data LIFs to share the same SPN.

Example of identical SPNs on multiple data LIFs in Data ONTAP 8.2.x and earlier:

```
cluster::> kerberos-config show
(vserver nfs kerberos-config show)
Logical
Vserver      Interface      Address      Kerberos SPN
-----
SVM          data1          10.63.57.237  enabled
nfs/svmdatalif.domain.netapp.com@DOMAIN.NETAPP.COM
SVM          data2          10.63.3.68    enabled
nfs/svmdatalif.domain.netapp.com@DOMAIN.NETAPP.COM
```

Example of unique SPNs on multiple data LIFs in Data ONTAP 8.2.x and earlier:

```
cluster::> kerberos-config show
(vserver nfs kerberos-config show)
Logical
Vserver      Interface      Address      Kerberos SPN
-----
SVM          data1          10.63.57.237  enabled
nfs/svmdatalif1.domain.netapp.com@DOMAIN.NETAPP.COM
```

```
SVM          data2          10.63.3.68      enabled
nfs/svmdatalif2.domain.netapp.com@DOMAIN.NETAPP.COM
```

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.

In ONTAP 8.2.x and earlier:

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

In ONTAP 8.3.x and later:

```
cluster::> kerberos interface 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 Active Directory servers other than Windows (such as MIT or Heimdal), the keytab needs to be copied to the cluster from the KDC. For more information on configuring non-Windows KDCs such as MIT, see the section in the appendix on [MIT KDC configuration](#).

Using AES When Kerberos Is Configured to Use DES

If Kerberos was configured for the SVM using DES encryption, then the existing machine account/principal needs to be destroyed and recreated. This action is needed because of the change in Kerberos libraries in Data ONTAP and the need to reestablish the Kerberos context on the SVM. Changing from DES to AES on NFS data LIFs *is a disruptive process*.

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) processes the request and attempts to authenticate the SPN. The authentication attempt maps the SPN using the SERVICE portion of the SPN. For example, if the SPN is `nfs/hostname@REALM`, then the cluster tries to do a `krb-unix` mapping for the user name "nfs." If that name does not exist in local files or name services such as LDAP, the authentication attempt fails.

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: NFS Best Practices and Implementation Guide](#).

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 preceding format. Use the greater than (>) or less than (<) symbols 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
```

By default, SecD logs errors only (severity ERR). These errors appear in the EMS log and can be viewed with the following command:

```
cluster::> event log show -messagename secd*
```

EMS can also log different levels of SecD messages if the log severity level is adjusted.

```
cluster::> set diag
cluster::*> secd log modify -node [nodename] -level
    emerg alert crit  err   warn  notice info  debug

cluster::*> secd log modify -node [nodename] -level debug
cluster::*> secd log show -node [nodename]
Log Options
-----
Log level:                               Debug
Function enter/exit logging:             OFF
```

After this logging is done, EMS shows all levels of SecD logging specified in the `-level` option.

Note: It is important to note that after the troubleshooting of SecD is complete, the log level should be set back to the normal ERR severity level to prevent the EMS logs from being spammed and rolling off too quickly.

Name Mapping

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

For example, `nfs/kerberos.domain.netapp.com` maps 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 fails and Kerberos is 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
```

After the preceding is created, the SPN then maps to the UNIX user named `krbuser` instead of the SPN user `nfs`. The clients also attempt a Kerberos-to-UNIX mapping with their SPN.

Best Practices 8: Client SPN Behavior (see next: Best Practices 9)

Clients such as RHEL, SUSE, and so on use the SPN of `root/hostname@REALM` in most cases. In Solaris, the client generally uses the SPN of `host/solaris@REALM`. Therefore, 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 when Solaris clients intending to leverage Kerberos are being used.

Note: The root user exists by default in Data ONTAP 8.2 and later, but must be created manually in earlier versions.

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

```
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
```

Example:

```
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
```

Then, run the following command to translate the user returned to a UID:

```
cluster::*> diag secd authentication translate -node node1 -vserver vs0 -unix-user-name nfs
500
```

The following shows a name mapping for the SPN failing:

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

Vserver: vs0 (internal ID: 4)

Error: Acquire UNIX credentials procedure failed
 [ 4 ms] Entry for user-name: host not found in the current
         source: FILES. Ignoring and trying next available source
 [ 5] Using a cached connection to 10.228.225.120
 [ 8] Source: LDAP not responding or corrupt. Ignoring and
         trying next available source for user-name: host
 [ 8] Entry for user-name:host not found in any of the
         available sources
**[ 8] FAILURE: Unable to retrieve UID for UNIX user host
```

Name-Mapping Considerations in Active Directory Domain Trusts

Data ONTAP [supports authentication using domain trusts](#). When using a domain trust with Kerberized NFS, the 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 attempts 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 preceding example, the request first tries to find a local user named `trust` using implicit mapping. Because that failed, the request then looks for a name mapping in LDAP, which is the next preferred `nm-switch/ns-switch` specified for the SVM. After it's determined that the name doesn't exist in LDAP, the cluster then looks for a name-mapping rule. If no name-mapping rule exists, the request fails.

Note: Because this is a `krb-unix` name mapping, the default UNIX user setting does not apply, because that is a `win-unix` attribute only.

The methods to resolve this challenge are:

- Create a local UNIX user on the SVM with the same user name as the user SPN attempting access.
Example: SPN `nfs/client.netapp.com` gets a local user named `nfs`.
- Create LDAP entry for the UNIX user.
Example: SPN `nfs/client.netapp.com` gets an LDAP user named `nfs`.
- 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 (regex), 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 Data ONTAP documentation for the version being used. For examples of using multidomain name mapping and regular expressions for trusted domains, see the section of this document entitled "[Name Mapping Across Multiple Domains.](#)"

Best Practices 9: Using `nm-switch` and `ns-switch` (see next: Best Practices 10)

When configuring name services in Data ONTAP, only add an external name service (such as LDAP or NIS) to name-mapping (`nm-switch`) or name-service (`ns-switch`) databases if external name services are in use and are properly configured. Adding external name services that don't exist in the environment can inject failures and delay authentication requests.

Example of name-mapping rule using regex for all machine account SPNs coming from the same 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` in 8.2.x and earlier:

```

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

```

Example of SVM name-service setting for `ns-switch` in 8.3 and later:

```

cluster::> vserver services name-service ns-switch show -vserver WIN2K12
Vserver      Database      Source
-----
WIN2K12      hosts         files,
              dns
WIN2K12      group         files,
              ldap
WIN2K12      passwd        files,
              ldap

```

```
WIN2K12      netgroup      files
WIN2K12      namemap        files,
              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
```

Other Considerations and Best Practices

Best Practices 10: LIF Communication with Name Services (see next: Best Practices 11)

At least one data LIF in the SVM must be able to communicate with Active Directory/name services. If the data LIF is not communicating, check the configuration of your data LIF and overall network. Starting in Data ONTAP 8.3, only one routable data LIF is needed per SVM, but the best practice is still to have one routable data LIF per node per SVM to make sure of data locality.

The only time the data LIF communicates with Active Directory is during the machine account creation. After that point, the cluster stores the Kerberos keytab locally.

Best Practices 11: SPN Length (see next: Best Practices 12)

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 longer than 15 characters gets truncated. The machine account name is derived from the SPN specified, including the service portion. If you wish to rename a machine account, see the [section in this document with the steps to do that](#).

Example:

SPN `nfs/kerberos.netapp.com@NETAPP.COM` becomes `NFS-KERBEROS-NE`
SPN `nfs/reallylongname.netapp.com@NETAPP.COM` becomes `NFS-REALLYLONGN`

Best Practices 12: Machine Account OU Specification (see next: Best Practices 13)

By default, the machine account is placed in the `CN=Computers` location on a Windows Active Directory domain controller. In versions before 8.2.1, this action could not be changed; 8.2.1 and later allowed the OU to be specified. To work around this limitation, move the machine account manually in Active Directory Users and Computers after Kerberos is enabled on the data LIF.

Best Practices 13: Encryption Type Information (see next: Best Practices 14)

Data ONTAP supports DES, 3DES, and AES encryption types for Kerberized NFS (depending in the ONTAP version). For specific information on what encyptes are supported in specific Data ONTAP versions, see the section "[Supported Encryption Types](#)." AES encryption was not added until Data ONTAP 8.3. For information on configuring the NFS client to navigate this limitation, see the section "[Configuring the Client](#)." For information on how to navigate this limitation from the domain controller, see the section "[Configuring the Domain Controller](#)."

Best Practices 14: Name-Mapping Rule Limits (see next: Best Practices 15)

Each SVM has a limit of 1,024 local name-mapping rules. If more than 1,024 name-mapping rules are needed, use LDAP to serve the rules or regex to consolidate the rules.

UNIX to Windows Authentication Behavior in Trusted Domains

UNIX to Windows name mappings might only be able to look up groups in the domain where the Windows user was authenticated. Therefore, NFS access might not behave as desired when using LDAP across trusted domains. Trusted domain groups are not queried by `secd`, so it is best to leverage Microsoft's best practice for domain trusts by way of [A-G-DL-P](#). This behavior affects only NTFS security-style volumes and `qtrees`, because NTFS requires a UNIX user to map to a valid Windows user. When using CIFS, trusted domain groups operate as expected. For more information on configuring LDAP servers in multidomain trusts, see the section "[Using Multiple Domains in a Forest for UNIX User and Group Identities](#)."

Configuring the Domain Controller for Kerberos

The domain controller configuration consists of the following:

- Choosing and allowing specific encryption types:
 - Allowing DES encryption types (Data ONTAP versions before 8.3)
 - Modifying machine account objects to use DES
 - Using AES encryption (Data ONTAP versions after 8.3)
- 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.

Adding NFS Clients to Windows Active Directory DNS

To properly leverage Kerberized NFS, the NFS clients should be added to DNS as `A/AAAA` records. `CNAME`s are supported, but they should be avoided when adding clients to `netgroups` or `export-policy` rules as host names. Additionally, all host names should have corresponding `PTR` records in DNS. For more information on host names in Data ONTAP, see [TR-4379: Name Services Best Practices](#).

Best Practices 15: Kerberos NFS Clients and DNS (see next: Best Practices 16)

To utilize Kerberos, the NFS client needs to have forward and reverse lookup entries in DNS. For more information on DNS entries used with Kerberos, see the [Recommended Practices for Deploying Kerberos by the Kerberos Consortium](#).

For steps on configuring this in Windows Active Directory DNS, see the [configuration steps](#) section in this document.

Adding the SVM Data LIFs to Windows Active Directory DNS

The SVM data LIF IP addresses need to be added to DNS in addition to the NFS client's IP address. The same steps apply as in the preceding section. As of ONTAP 9, both IPv4 and IPv6 are supported for DDNS. Previously, starting in ONTAP 8.3.1, only IPv4 was supported. See [TR-4379: Name Services Best Practices](#) for more information on DDNS.

Best Practices 16: Prevent Lookup Failures with Multiple LIFs (see next: Best Practices 17)

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 Data ONTAP [on-box DNS load-balancing feature](#). Doing so 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:

cluster	Host (A)	10.10.10.10
cluster	Host (A)	10.10.10.11
cluster	Host (A)	10.10.10.12

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 Data ONTAP networking best practices, see [TR-4182: Best Practices for Data ONTAP Network Configurations](#).

Using DNS Aliases/Canonical Names (CNAMEs)

When enabling Kerberos on a data LIF, the SPN is specified during the configuration. This SPN determines which host name is used to access Kerberized mounts. For example, if the SPN of `nfs/kerberos.domain.netapp.com` is used, then the mounts are accessed with the host name of `kerberos`. This occurs because the host name used in the mount determines which SPN to pass to the KDC for authentication. If a DNS alias is used, then that alias is passed as the SPN to the KDC, and Kerberized mounts 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. After this step is taken, the SPN request is forwarded to the appropriate principal in the KDC.

Best Practices 17: On-Box DNS (see next: Best Practices 18)

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. Doing so enables name resolution of data LIFs to return multiple IP addresses to clients to prevent overloading a single node in the cluster with connections.

Using on-Box DNS Load Balancing

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 make sure of proper load balancing across the cluster for mount requests. After a mount is successful, the client continues to use that connection until remount. This approach 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 which 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.

For more information on on-box DNS load balancing, as well as how to configure it, see [TR-4523: DNS Load Balancing in ONTAP](#).

Adding SRV Records

Some NFS clients (such as SLES) 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:

Name	Type
_gc	Service Location (SRV)
_kerberos	Service Location (SRV)
_kpasswd	Service Location (SRV)
_ldap	Service Location (SRV)

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

In the preceding, notice that `kerberos-master` does not exist by default. Create this record for each DC for clients that require this record.

A clue that the `_kerberos-master` SRV record is needed is if an NFS client can successfully run `kinit` with a user but cannot use `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 shows 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 steps in the corresponding section show how to create them.

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

Allowing DES and/or AES Encryption

Data ONTAP 8.2.x and earlier support only DES and 3DES encryption types for NFS Kerberos. Windows, however, does not support 3DES, so only DES encryption can be used when using Active Directory KDCs. Data ONTAP 8.3 and later introduced support for AES encryption for Kerberos. Windows 2008 R2 and later 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](#).

Best Practices 18: Kerberos Encryption Type Recommendation (see next: Best Practices 19)

If selecting a Kerberos encryption type, select the strongest encryption available. For Data ONTAP 8.3 and later, NetApp recommends AES 256-bit encryption strength.

If DES is not allowed on the KDC, the following might be seen from a client trying to obtain a ticket:

```
[root@linux-client sysconfig]# kinit -k root/linux-client-4.domain.netapp.com@DOMAIN.NETAPP.COM
kinit: KDC has no support for encryption type while getting initial credentials
```

Enabling DES and/or AES in Windows 2008 R2 and Later

This enabling allows DES and/or AES encryption for the entire domain. After enabling takes place, the registry key `HKLM\Software\Microsoft\Windows\CurrentVersion\Policies\System\Kerberos\Parameters\SupportedEncryptionTypes` is changed. If this is not done, AES and/or DES requests fail, meaning that NFS mounts using Kerberos with Data ONTAP fail, depending on how the Kerberos interface is configured. Make sure that the encryption types allowed match those specified on the [NFS server using the `nfs_server show -fields permitted-enc-types` command](#). For complete steps on how to enable DES and/or AES in Windows 2008 R2 and later, see the [corresponding section for configuration steps in this document](#).

Disabling DES and/or AES in Windows 2008 R2 and Later

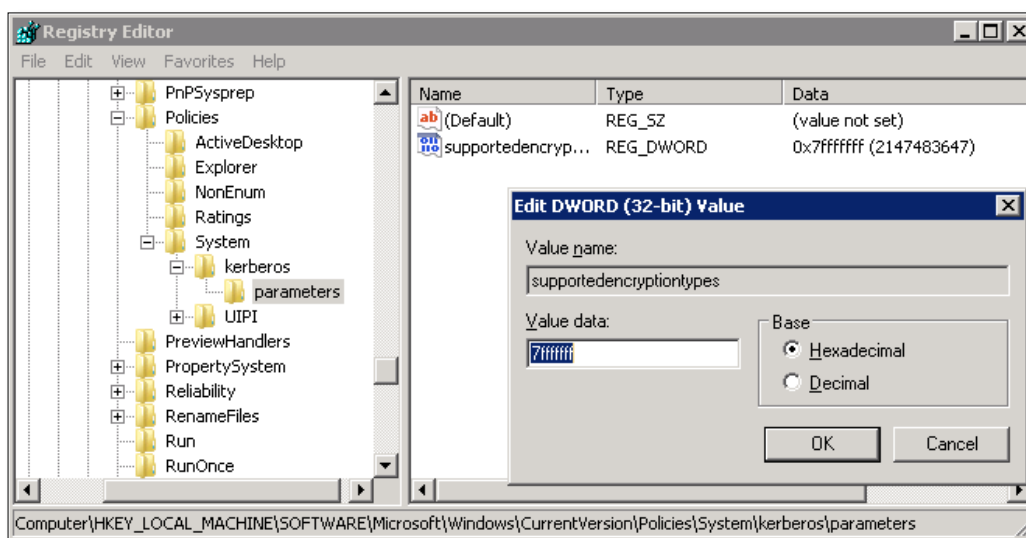
To disable DES and/or AES in Windows 2008 R2 and later, modifying the GPO by unchecking the policy settings is not enough. Doing so does not change the registry value on the DC.

Best Practices 19: Disabling DES (see next: Best Practices 20)

When using AES encryption for Kerberos, be sure to disable DES encryption on the KDC, provided no legacy clients still use DES.

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

Figure 2) DES enabled.



Using AES Encryption

AES encryption (both 128-bit and 256-bit) is supported in Data ONTAP 8.3 and later. Support for AES is controlled using the NFS server option `-permitted-enc-types`. After this option is set, Kerberos interfaces inherit the allowed encryption types. AES encryption is provided to include the strongest available encryption for Kerberized NFS.

What Is Advanced Encryption Standard (AES)?

Advanced Encryption Standard (AES) is covered in [RFC-3962](#). AES supersedes DES encryption and offers stronger encryption than any other encypte available with Kerberos. AES supports both 128-bit and 256-bit block encryption. AES is the chosen encryption method of the [U.S. National Institute of Standards and Technology \(NIST\)](#).

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 work in conjunction with keytab files to pass the encrypted secret password hash to allow Kerberos tickets to be granted. A machine object is created but is not a valid Kerberos principal until the keytab is created using [ktpass](#) or an 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.

Note: 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; only keytab authentication is needed.
- Logic: Machine accounts for machines make more sense.
- Multiple SPNs can be assigned to a machine account.

- For example, `nfs/nfsclient`, `root/nfsclient`, `host/nfsclient`.
- No need for multiple accounts, such as with user accounts.

The downside of using machine accounts is that they cannot be customized as easily as user accounts. User accounts in Active Directory provide GUI access to allow DES, forego preauthentication, and so on. Machine accounts require modification using 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 recreated. Although this might be tedious, it is more secure than never allowing the machine account password to expire.

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

For configuration steps to create machine accounts for Kerberos principals, see the corresponding section in this document.

Modifying Machine Account Attributes

This section describes how to modify the machine accounts created for Kerberized NFS. Data ONTAP 8.2.x and earlier (clustered) support 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, because DES is considered less secure than other encryption types. In Data ONTAP 8.3 and later, AES encryption is supported. Although AES is supported by default in Windows 2008 and later, it might be necessary to modify machine accounts to limit access to AES. That is because Linux clients might try to use other encryption types that are unsupported by NFS Kerberos (such as RC4-HMAC).

Machine account refers to the Kerberos principal. In this case, it is the NFS server account created by the Kerberos enablement in Data ONTAP. However, in some cases (such as when using DES encyptes), it might be necessary to perform the same steps on the NFS client machine accounts to make sure that they attempt to use the proper encryption types. For instance, in Windows Active Directory, RC4-HMAC is enabled by default. If a client attempts to use DES (which is a weaker enctype than RC4), then it might attempt to negotiate RC4 with the KDC and the cluster. However, Data ONTAP does not support RC4-HMAC for NFS clients, and authentication fails.

The following section details 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
- Modifying machine accounts to use AES only (for Data ONTAP 8.3 and later)

Why Modify the Machine Account?

Modifying the machine account is necessary so that Kerberos authentication requests leverage the supported encryption types 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_sec:info:39727] | [000.002.049] debug: GSS_S_COMPLETE:
client = 'root/nfsclient.domain.netapp.com@DOMAIN.NETAPP.COM' { in acceptGssToken() at
gss/sec_d_gss_accept_token.cpp:288 }
Wed May 22 2013 11:15:57 -04:00 [kern_sec:info:39727] | [000.002.124] ERR : Unsupported
signing algorithm 17. { in parseKrb5ContextToken() at gss/sec_d_gss_parsekrb5.cpp:106 }
```

Because NFS clients generally support most encyptes, modifying the Data ONTAP NFS server machine account is the most logical approach. Using this approach avoids the need to modify scores of machine

accounts when stronger encyptes become available. This approach also provides a hybrid security mechanism that allows stronger encyptes when they are available.

In the following `klist -e` output, the Kerberos ticket-granting ticket (krbtgt) uses AES while the NFS mount uses DES. When this client mounts a Data ONTAP data LIF using Kerberos, it obtains the TGT using the strongest encryption type available. In this case, that is AES-256. Because the Data ONTAP Active Directory machine account for Kerberos has been restricted to DES only and the client's `krb5.keytab` and `krb5.conf` files have DES and AES listed as supported encyptes, the account uses AES for the client portion of the service ticket (ST) and DES for the server portion of the ST. Doing so is much more secure than using DES for all Kerberos encryption. Furthermore, as support for stronger encryption types becomes available, the client configuration generally does not have to change.

Example of `klist` command:

```
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 NFS Kerberos machine account object and [creating a proper keytab file](#) allow multiple encryption types to be supported on an NFS client.

Note: To modify machine accounts in Active Directory, see the [configuration steps in this document](#).

Best Practices 20: Tools to Configure Machine Accounts (see next: Best Practices 21)

It is possible to modify the machine account using ADSI Edit, Idifde, or Windows PowerShell, such as in the DES examples, with the same steps. However, NetApp recommends using Active Directory Users and Computers unless scripting large numbers of machine accounts is required.

Creating the NFS Client Keytab File

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)
- A principal type
- A 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` increases, potentially causing the existing clients to be unable to authenticate using Kerberos. A new keytab file needs to be migrated and applied to the NFS clients. 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 only using the command line. The appendix in this document shows [command syntax for ktpass](#). 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.

Best Practices 21: SPN Considerations for RHEL/CentOS 6.x (see next: Best Practices 22)

In Red Hat versions before 6.x, the keytab must be created with an SPN in the format of `nfs/hostname@REALM`. If using the `root/hostname@REALM` format, the client does not read the keytab file properly:

```
# klist -kte
Keytab name: FILE:/etc/krb5.keytab
klist: No such file or directory while starting keytab scan
```

For steps on manually creating a keytab file for use with NFS Kerberos on Windows Active Directory KDC, see the [corresponding configuration steps in this document](#).

Setting the Keytab to Use AES Only

If the NFS client uses only AES-128 or AES-256, set the keytab file to use only those encryption types rather than using "ALL" in the `ktpass` command.

```
C:\> ktpass -princ primary/instance@REALM -mapuser DOMAIN\machine$ -crypto AES256-SHA1 +rndpass -
ptype KRB5_NT_PRINCIPAL +Answer -out [file:\location]
```

Example:

```
C:\>ktpass -princ root/nfsclient.domain.netapp.com@DOMAIN.NETAPP.COM -mapuser DOMAIN\nfsclient$ -
crypto AES256-SHA1 +rndpass -ptype KRB5_NT_PRINCIPAL +Answer -out C:\nfsclient-256only.keytab
```

Moving Keytab Files to the NFS Client

After the keytab file has been created, it needs to be moved to the NFS client. Moving the file 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.

Creating NFS Server Keytab Files

Windows KDCs allow automation of several of the tasks required when configuring Kerberos for NFS servers in Data ONTAP. The `kerberos-config` (8.2.x and earlier) or `kerberos interface` (8.3 and later) commands provide the ability to specify the following attributes:

- SPN
- OU
- Admin user name
- Keytab URI

Best Practices 22: Kerberos Interface Command Usage (see next: Best Practices 23)

The `kerberos interface` commands in Data ONTAP apply only to Kerberos for NFS. CIFS Kerberos is handled by the `cifs` server creation.

In some scenarios, manually creating the principals and keytab files might be required because of security restrictions in an environment. For example, there might be a restriction as to who has access to administrative passwords and user names that must be provided when configuring Kerberos on Windows KDCs. This section provides the steps to manually create NFS server keytab files for use cases such as these.

Best Practices 23: Kerberos Interface Command (see next: Best Practices 24)

If possible, use the on-box Kerberos configuration commands that interact with the Windows KDC through administrator user and password. Automating the steps in this process helps avoid mistakes in configuration and makes sure that the keytab file is transferred to the cluster in a secure, encrypted fashion.

If you use a KDC other than Windows, the only method of keytab transfer is `-keytab-uri`, which currently supports HTTP/HTTPS (as of Data ONTAP 8.3.1) and FTP transfers.

What Happens When Kerberos Is Configured from the Cluster Using Windows KDCs?

When a Windows KDC is used and a valid domain administrator user name and password are provided, the cluster interacts with the KDC and does the following when the command is issued:

1. After the command is issued, a user name and password are requested.
2. Data ONTAP initiates communication with the KDC listed in the specified Kerberos realm using a data LIF in the SVM that has the ability to route to the KDC.
3. A Kerberos AS-REQ is issued from the cluster SVM for the user name provided to the KDC. This process is known as the "AS Exchange," which is described in TechNet's article [Kerberos Explained](#).
4. The KDC responds to the AS-REQ with AS-REP if the provided principal exists and the provided password is valid.
5. The cluster SVM then issues a TGS-REQ to the KDC using the TGT that was issued during the AS Exchange.
6. If the KDC approves the TGS-REQ, it responds with a TGS-REP and issues a service ticket (ST) to the client.
7. A Kerberos keytab is created on the KDC, returned to the cluster, and added to the replicated database using encrypted Kerberos packets.
8. After the administrator account has logged in successfully using Kerberos, the SVM attempts to bind to the Windows LDAP server using the same credentials using SASL.
 - Given that a valid Kerberos ticket was issued in the previous steps, the SASL request succeeds.
9. After the bind, a search request is made to LDAP for a machine account using the `sAMAccountName` attribute. This machine account is limited to 15 characters and uses the NFS spn specified in the Kerberos command to form the name (for example, NFS-KERBEROS-MA is used for `nfs/kerberos.machine.netapp.com`).
 - If the machine account exists, the storage administrator is prompted to reuse the account. If the account does not exist, the request proceeds to create the machine account using an `addRequest` LDAP call.
 - If the user name provided by the storage administrator has permissions to the specified OU to create machine accounts, the `addRequest` succeeds.
10. The machine account is populated with HOST and NFS service principals (SPNs) using the `addRequest` call.

Figure 3) Example of addRequest in packet trace.

```
Lightweight Directory Access Protocol
└─ LDAPMessage addRequest (3) "cn=NFS-CDOTKRB-DOM,cn=Computers,dc=DOMAIN,dc=WIN2K8,dc=NETAPP,dc=COM"
  messageID: 3
  └─ protocolOp: addRequest (8)
    └─ addRequest
      entry: cn=NFS-CDOTKRB-DOM,cn=Computers,dc=DOMAIN,dc=WIN2K8,dc=NETAPP,dc=COM
      └─ attributes: 6 items
        └─ AttributeList item cn
          type: cn
          └─ vals: 1 item
            AttributeValue: NFS-CDOTKRB-DOM
        └─ AttributeList item sAMAccountName
          type: sAMAccountName
          └─ vals: 1 item
            AttributeValue: NFS-CDOTKRB-DOM$
        └─ AttributeList item objectClass
          type: objectClass
          └─ vals: 5 items
            AttributeValue: top
            AttributeValue: person
            AttributeValue: organizationalPerson
            AttributeValue: user
            AttributeValue: computer
        └─ AttributeList item servicePrincipalName
          type: servicePrincipalName
          └─ vals: 5 items
            AttributeValue: HOST/NFS-CDOTKRB-DOM
            AttributeValue: HOST/nfs-cdotkrb-dom.domain.win2k8.netapp.com
            AttributeValue: nfs/NFS-CDOTKRB-DOM
            AttributeValue: nfs/nfs-cdotkrb-dom.domain.win2k8.netapp.com
            AttributeValue: nfs/cdotkrb.domain.win2k8.netapp.com
        └─ AttributeList item userAccountControl
          type: userAccountControl
          └─ vals: 1 item
            AttributeValue: 4096
        └─ AttributeList item operatingSystem
          type: operatingSystem
          └─ vals: 1 item
            AttributeValue: ONTAP 8.1
```

Finally, the command completes and NFS Kerberos is configured for the NFS server.

How to Configure NFS Kerberos on Windows KDCs Manually

If an administrator user name and password are not known by the storage administrator, the following tasks can be assigned to the Windows KDC management team before configuring NFS Kerberos for an SVM:

1. Create/configure the NFS Kerberos realm information for the SVM.
2. Create a machine account for the NFS server with any machine account name desired.
3. Modify the NFS server machine account as per the section in this document called "Modifying Machine Account Attributes."
4. Create a keytab file for the NFS machine account using the ktpass command as described in the section "Creating the NFS Client Keytab File."

Note: Be sure to set the account to use DES only if configuring Kerberos in Data ONTAP 8.2.x and earlier.

5. Copy the keytab file to a web (HTTP) or FTP server.

Note: HTTPS/FTPS support for this operation was added in ONTAP 8.3.1. See [bug 816595](#) for details.

6. When running the Kerberos configuration commands on the cluster, specify the option `-keytab-uri` and use the web address that points to the desired NFS server keytab file.
7. The keytab is imported into the replicated database without needing to issue an administrator user name or password.
8. Complete the remaining steps and test NFS Kerberos access.

Verifying SPNs on the KDC

After configuring the domain controller, verify that the machine account created by enabling Kerberos on the SVM data LIF(s) has the proper SPNs.

Best Practices 24: NFS Kerberos SPN with CIFS Servers (see next: Best Practices 25)

If a CIFS server exists in the same SVM and domain, it's imperative that Active Directory is [searched for duplicate SPNs](#). Duplicate SPNs can cause authentication failures with Kerberos. If using a trusted domain setup, make sure that the same SPN does not exist in multiple domains by running the `setspn` utility on each domain's DC.

Duplicate SPNs [log errors in the Windows event log](#), and Kerberos attempts fail. To find duplicate SPNs in Windows 2008 and later, use the `setspn` utility with the `/Q` flag to query for SPNs.

In the following example, note that there are two CNs (CN=nfsclient and CN=linux-client) that have the SPN `root/nfsclient.domain.netapp.com` assigned:

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

Existing SPN found!

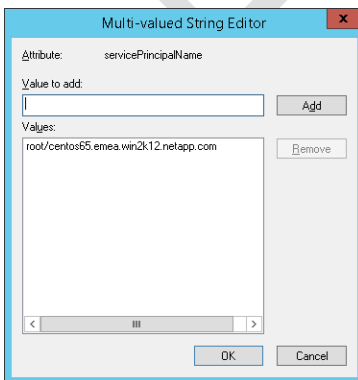
Note: A common scenario in which duplicate SPNs might occur is when a CIFS server was created and Kerberos was enabled for the same SVM. Be sure to check the `nfs/cluster@REALM` SPN. However, this can occur on any machine account in the domain if a misconfiguration occurs.

If more than one CN is listed for the SPN that is queried, then delete one of the SPNs using either the `setspn` tool or the [Attribute Editor](#) tab in the Active Directory Users and Computers GUI.

To delete a duplicate SPN:

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

SPNs can also be added/deleted through the GUI in Active Directory Users and Computers using the Attribute Editor tab:



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 "[Kerberos and LDAP Error Messages](#)."

For details on using packet traces for Kerberos troubleshooting, see "[Kerberos Errors in Network Captures](#)." Also see Table 24 in this document for common Kerberos errors in network captures.

Using Domain Trusts

Trusted domains are domains that the local system trusts to authenticate users. In other words, if a user or an application is authenticated by a trusted domain, this authentication is accepted by all domains that are configured to trust the authenticating domain. [For more information on domain trusts, see the TechNet article on the subject.](#)

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 using CIFS or Kerberized NFS. This process 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 are not duplicated across both domains.

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

Configuring NFS Clients to Use Kerberos—Manual Configuration

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 "[Quick Step Setup Guides](#)" in this document.

ESXi 6.0 Kerberos Configuration

ESXi 6.0 introduced support for Kerberos with NFS-mounted datastores (as well as NFSv4.1 support; no NFSv4.0 support). Currently, only DES encryption types are supported. For complete steps, see the section in the appendix of this document on [configuring Kerberos in ESXi 6.0](#).

Solaris Kerberos Configuration

Solaris generally uses a utility called `kclient` to configure Kerberos. However, this utility has issues when configuring Kerberos with a Windows KDC because of the format the utility expects to use (user name/admin). Attempts to use the utility might result in the following error:

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

For Windows KDCs, use the following best practice recommendations.

Best Practices 25: Kerberos Setup with Solaris (see next: Best Practices 26)

For Windows KDCs, use the [same steps for other clients](#) to configure Kerberos. For MIT KDCs, use `kclient`. Keep in mind the following general best practices for Kerberos setup:

- 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 five minutes of the KDC.
- Verify that the [keytab](#) file has been created and exported to the client.

Using `kclient` with MIT KDCs

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

If “Do you plan on doing Kerberized nfs?” is answered with “yes,” then the client attempts to create the SPNs for the client on the KDC using the principal specified in the administrative principal section of the script. This process creates 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](#).”

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
 - Later releases, such as 6.x and 7.x, should also be able to use these steps.
- Fedora 18
- SLES 11
- SUSE 12
- Ubuntu 12.1

Best Practices 26: NFS Client OS Recommendation (see next: Best Practices 27)

Use the latest available client release when possible for up-to-date bug fixes and interoperability.

Host Name

For Kerberos to work properly, it's important that the client's host name be set using 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 5) 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 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 might manage their date and times differently, but the main consideration with Kerberos is that the date and time on the NFS clients 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 fail.

DNS

Clients 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@nfsclient ~]# nslookup nfsclient
Server:      10.63.98.101
Address:     10.63.98.101#53

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

[root@nfsclient ~]# 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.domain.netapp.com.

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

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

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

NFS Client Behavior with Kerberos and DNS

By default, an NFS client using Kerberos will attempt to leverage DNS to formulate SPN requests. This means a mount request of `clientname:/` or `ipaddress:/` will be looked up in DNS to figure out what the SPN is. If this behavior is undesirable (for example, if the DNS name is different than the SPN name), then leverage the `krb5.conf` file option `dns_canonicalize_name` to bypass the default behavior.

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: Other sections of this file 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 needed for Kerberos only if a higher level of security is desired. NFSv3 can be used with Kerberos.

Allowing Secure NFS

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

Note: In most cases, RHEL/CentOS 7.x no longer needs this file to be changed.

Table 6) Allowing secure NFS.

OS	File to Modify	Value to Change
RHEL/CentOS/Fedora (prior to CentOS/RHEL 7)	<code>/etc/sysconfig/nfs</code>	<code>SECURE_NFS="yes"</code>
RHEL/CentOS 7.x	N/A	In 7.0, run: <code>systemctl enable nfs-secure &&</code> <code>systemctl start nfs-secure</code> In 7.1 and later, run: <code>systemctl enable nfs-client.target &&</code> <code>systemctl start nfs-client.target</code>
SLES/SUSE	<code>/etc/sysconfig/nfs</code>	<code>NFS_SECURITY_GSS="yes"</code>
Ubuntu	<code>/etc/default/nfs-common</code>	<code>NEED_GSSD="yes"</code>
Solaris	<code>/etc/nfssec.conf</code>	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 gets its Kerberos configuration information. This file must exist on clients wanting to use Kerberos.

- In Linux, the file is at `/etc/krb5.conf`.
- In Solaris, the file is 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 using *DES encryption*.

Note: DES and DES3 are allowed by way of the `allow_weak_crypto=true` option in the `krb5.conf` file. To disallow DES and DES3, set that value to "false."

```
[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, modify the settings under `[libdefaults]` that control the order of encyptes requested.

To modify the order from what the client defaults to, add the following lines under `[libdefaults]` so that the config looks like this:

```
[libdefaults]
default_realm = DOMAIN.NETAPP.COM
default_tkt_encytypes = des-cbc-md5 des-cbc-crc aes256-cts des3-cbc-sha1 arcfour-hmac
default_tgs_encytypes = 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 because of its lack of scalability.

Note: Data ONTAP 8.2 and earlier support 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 "[Configuring the Domain Controller](#)" for details.

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

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

Check the ticket with `klist`:


```
[root@nfsclient ~]# 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 reenale it, add the following line to the `/etc/environment` file and reboot the client:

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

For more information, see [Bugzilla report 895513](#).

The following `krb5.conf` file can be used for clients using *AES encryption*:

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

[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 that the `allow_weak_crypto` option is set to false. This setting helps to make sure that DES and other weak encryption methods are not used.

Best Practices 27: NFS Client OS with AES Encryption (see next: Best Practices 28)

If you use AES encryption with Linux clients, it is best to use the most recent updated version of the OS possible. Earlier Linux kernels have exhibited buggy behavior when attempting to leverage AES encryption.

krb5.keytab

This file is 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. The file should be readable only by root and should exist only on the local server's disk. The file is created on the KDC and then moved to the NFS client. After it is 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`.

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

```
[root@nfsclient ~]# ktutil
ktutil: rkt /nfsclient.keytab
ktutil: list
```

```
slot KVNO Principal
-----
  1   3 root/nfsclient.domain.netapp.com@DOMAIN.NETAPP.COM
ktutil: wkt /etc/krb5.keytab
ktutil: q
```

The key version number (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@nfsclient ~]# klist -kte
Keytab name: FILE:/etc/krb5.keytab
KVNO Timestamp          Principal
-----
--
 4 05/16/13 11:57:56 root/nfsclient.domain.netapp.com@DOMAIN.NETAPP.COM (des-
cbc-crc) ← KVNO is 4
 4 05/16/13 11:57:56 root/nfsclient.domain.netapp.com@DOMAIN.NETAPP.COM (des-
cbc-md5)
 4 05/16/13 11:57:56 root/nfsclient.domain.netapp.com@DOMAIN.NETAPP.COM
(arcfour-hmac)
 4 05/16/13 11:57:56 root/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@nfsclient /]# kinit administrator
Password for administrator@DOMAIN.NETAPP.COM:
[root@nfsclient /]# kvno root/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.

AES Encryption

Many modern Linux clients support AES encryption for Kerberos, which is a more secure encypte than DES or 3DES. To use AES encryption with Linux clients, check with the OS vendor to make sure the version being used supports AES encyptes. Generally, leaving the default `krb5.conf` file setup allows the client and KDC to negotiate AES encryption, because it is the strongest encypte available. In the case of Windows KDCs, the principal should be modified to make sure only AES encryption is used, because some Windows KDCs attempt to use RC4-HMAC, which is not supported by Data ONTAP. As a result, Kerberos mounts fail. See the section regarding [AES setup on Windows KDCs](#) for more information.

Best Practices 28: Timeout Value for rpcgssd (see next: Best Practices 29)

In some cases, a client might experience issues with negotiating Kerberos tickets because of network or client configurations. To avoid this problem, set the timeout value for `rpcgssd` to `-T 60` when mounting.

```
-T timeout
Timeout, in seconds, to create an RPC connection with a server while establishing an
authenticated gss context for a user. The default timeout is set to 5 seconds. If you get
messages like "WARNING: can't create tcp rpc_clnt to server %servername% for user with uid
%uid%: RPC: Remote system error- Connection timed out", you should consider an increase of
this timeout.
```

Managing Kerberos Services

After 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 7) Managing Kerberos services.

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

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

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

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

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

For condensed Kerberos setup steps, see the [“Quick Step Setup Guides”](#) section in this document.

Joining an NFS Client to a Windows Active Directory Domain

In addition to using the manual keytab process for NFS clients using Kerberos, it is also possible to join a client to an Active Directory domain. In some cases, it might be preferable to use this method to add clients to KDCs because there is no need to create a manual keytab file in those scenarios. With some tools (such as “realm join”), the domain join creates the keytab for you. With “net ads,” you can easily create a keytab and pull it over with a single command.

However, when using “realm join” the keytab file created might be a CIFS/SMB style of keytab with only “host” service entries.

For example, the `realm join` commands in RHEL/CentOS 7.x create keytabs that look like this:

Table 8) Sample keytab file from “realm join” on CentOS 7.2.

```
[root@centos7 ~]# ktutil
ktutil: rkt /etc/krb5.keytab
ktutil: list
slot KVNO Principal
-----
1 2 host/centos7.core-tme.netapp.com@CORE-TME.NETAPP.COM
2 2 host/centos7.core-tme.netapp.com@CORE-TME.NETAPP.COM
3 2 host/centos7.core-tme.netapp.com@CORE-TME.NETAPP.COM
4 2 host/centos7.core-tme.netapp.com@CORE-TME.NETAPP.COM
5 2 host/centos7.core-tme.netapp.com@CORE-TME.NETAPP.COM
6 2 host/centos7@CORE-TME.NETAPP.COM
7 2 host/centos7@CORE-TME.NETAPP.COM
8 2 host/centos7@CORE-TME.NETAPP.COM
9 2 host/centos7@CORE-TME.NETAPP.COM
10 2 host/centos7@CORE-TME.NETAPP.COM
11 2 CENTOS7$@CORE-TME.NETAPP.COM
12 2 CENTOS7$@CORE-TME.NETAPP.COM
13 2 CENTOS7$@CORE-TME.NETAPP.COM
14 2 CENTOS7$@CORE-TME.NETAPP.COM
15 2 CENTOS7$@CORE-TME.NETAPP.COM
```

As such, there are some things to consider when using realm joins for NFS clients using Kerberos.

- Ensure that the Active Directory users can obtain Kerberos tickets from the KDC on the client using `kinit`.
- Leverage [LDAP on the Active Directory server](#) and [configure the SVM as a client](#) for identity mapping. [Use SSSD on the NFS client](#) for identity retrieval.
- Configure the [NFS client's krb5.conf](#) file for the domain realm.
- Create a local unix-user or configure the NFS client's computer account in LDAP to have a UNIX UID and GID so that the host/machine\$ SPN can authenticate properly.

All other considerations remain the same when [configuring NFS clients](#) for Kerberos.

You can join the NFS client to an Active Directory domain in multiple ways. In the Configuration Steps section of this document, the [realm join](#) command in RHEL/CentOS 7.x and "[net ads join](#)" were used. Other methods of joining domains (such as using Centrify or other third-party utilities) might require assistance from the vendors that provide the tool.

5 LDAP Overview

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 (AD). 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. Doing so 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 Data ONTAP using CIFS or NFS. Providing the ability to use standard network services for name resolution and for identity storage is crucial. 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 is provided as well.

What Does LDAP Store?

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

- User name
- UID or GID
- Homedirs
- Login shell
- Netgroups, DNS names, and IP addresses

- Group membership

What Active Directory LDAP Is Not

- Active Directory Identity Management (LDAP) is not a 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.
 - Non-Windows LDAP can be used for RBAC to manage the cluster.

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 using 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 uses the strongest Kerberos encryption type supported by the client and Active Directory Domain controller.

Best Practices 29: LDAP Application for NFS Clients (see next: Best Practices 30)

SSSD is the recommended LDAP client to use when possible because of its ease of use, performance, and integration with Kerberos for secure LDAP binds.

Red Hat Directory Services Overview

From [Red Hat's documentation on Red Hat Directory Services](#):

Red Hat Directory Server provides the following key features:

- Multi-master replication — Provides a highly available directory service for both read and write operations. Multi-master replication can be combined with simple and cascading replication scenarios to provide a highly flexible and scalable replication environment.
- Chaining and referrals — Increases the power of the directory by storing a complete logical view of the directory on a single server while maintaining data on a large number of Directory Servers transparently for clients.
- Roles and classes of service — Provides a flexible mechanism for grouping and sharing attributes between entries dynamically.
- Efficient access control mechanisms — Provides support for macros that dramatically reduce the number of access control statements used in the directory and increase the scalability of access control evaluation.
- Resource-limits by bind DN — Grants the power to control the amount of server resources allocated to search operations based on the bind DN of the client.
- Multiple databases — Provides a simple way of breaking down the directory data to simplify the implementation of replication and chaining in the directory service.
- Password policy and account lockout — Defines a set of rules that govern how passwords and user accounts are managed in the Directory Server.
- TLS and SSL — Provides secure authentication and communication over the network, using the Mozilla Network Security Services (NSS) libraries for cryptography.

The major components of Directory Server include the following:

- An LDAP server — The LDAP v3-compliant network daemon.
- Directory Server Console — A graphical management console that dramatically reduces the effort of setting up and maintaining the directory service.
- SNMP agent — Can monitor the Directory Server using the Simple Network Management Protocol (SNMP).

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 using Kerberos, and access Kerberized NFS mounts. SSSD does not leverage PAM, but other functions such as SSH and su 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.

Active Directory LDAP Using 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, AD provides the flexibility to script using batch file or Windows PowerShell to automate tasks. By default, Active Directory does not include UNIX-type schema attributes. These attributes 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 such as 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 map to the correct IDs intuitively, preventing access attempts from being squashed to `nfsnobody`, because LDAP makes sure that the names match the UIDs and GIDs exactly.

Best Practices 30: NFSv4 ID Mapping: The Nobody User (see next: Best Practices 31)

In Data ONTAP, an NFS server option enables the NFS server to respond to UID/GID requests to behave more like NFSv3. If a client can't be configured to use a valid NFSv4 ID domain, this option can be leveraged:

```
cluster::> nfs server modify -vserver NAS -v4-numeric-ids
enabled disabled
```

For more information on this option, see [TR-4067: NFS Best Practice and Implementation Guide](#).

Leveraging SSSD on Linux clients with Active Directory provides ease of use, security, and stability that other LDAP tools do not provide. Leveraging SSSD also makes sure that NFSv4.x clients can leverage NFSv4 ID domains for enhanced security. Using non-Windows LDAP servers is also supported.

[Get details on SSSD support.](#)

How SSSD Interacts with Active Directory

The following section details how SSSD interacts with Active Directory to query it for users and groups using the GSSAPI security method. You can also configure SSSD to bind to Windows Active Directory LDAP using simple binds using a bind DN (user) and password, though this is not as secure.

1. DNS queries for the LDAP SRV record are made using the first DNS server in the `/etc/resolv.conf` file; the 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 is used by SSSD.
3. A TCP connection to the LDAP server is established and a searchRequest is made to the baseObject. These actions start the authentication process.
4. Because GSSAPI was specified in the `/etc/sss/sss.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 the request is 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 attempts 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 reports as successful and delivers the LDAP query to the client using SASL GSS-API Privacy payload packets. These packets are encrypted using the strongest encryption supported by the client and domain controller.
15. After the request is complete, the LDAP server sends a reset (RST) packet to the client to close the TCP connection.

Red Hat Directory Services Benefits

Red Hat Directory Services (DS) takes Linux-based LDAP and makes it simple to configure and use. Other LDAP servers require multiple configuration files and do not provide GUI or scripts to configure the server. Also, because Red Hat develops both DS and SSSD, interaction between the two is seamless. Red Hat DS offers a setup script and GUI to make setup easy and foolproof. In addition, Red Hat DS offers the following features:

- LDAP schema replication across master LDAP servers (similar to what Active Directory offers)
- AD user and group sync
- Secure authentication and transport (TLSv1, SSLv3, SASL)
- LDAPv3 support

For information on setting up Red Hat DS, see the section in this document entitled "[LDAP Using Red Hat Directory Services.](#)"

5.1 LDAP Using Microsoft Windows AD for Identity Management

This section covers setting up LDAP for identity management using Microsoft Windows Active Directory. The server being used is a Windows 2008 R2 server with Identity Management installed, but these steps can also be applied to Windows 2012 servers.

Note: [Quick setup](#) steps can be found at the end of this document.

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 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 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 nodel -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 cannot be reached:

```
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 can be reached 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 ages out of cache:

```
cluster::*> diag secd connections show -node nodel -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 still work because of the backup LDAP server:

```
cluster::*> diag secd authentication show-creds -node nodel -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 shows up in cache:

```
cluster::*> diag secd connections show -node nodel -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)
```

After the other LDAP server is available again, the cluster starts using it because it is listed first in the configuration:

```
cluster::*> diag secd authentication show-creds -node nodel -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 nodel -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 might 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 leverage this capability as well. For more information, see the section regarding [setup of NFS clients to use LDAP](#).

Using the Domain Controller as an Identity Management Server for UNIX

As previously mentioned, Microsoft Active Directory does not act as an LDAP Identity Management 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 depends on the version of Windows being used. There are also third-party LDAP schema extenders, such as [Vintela](#) and [Centrify \(now provided by Dell\)](#). These third-party LDAP schema extenders are supported and can be used with any LDAP client that supports RFC-2307 schemas, including Data ONTAP.

The following table shows the Microsoft offerings for LDAP schema extension depending on the Windows version. For third-party schema extension, contact the vendor of the product.

Note: When running commands on servers operating in Windows 2008 and later, [User Account Control](#) might 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, disable User Account Control, or use the Run As Administrator option.

Table 9) 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)

Schema Extension

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.

Best Practices 31: Schema Extension Considerations (see next: Best Practices 32)

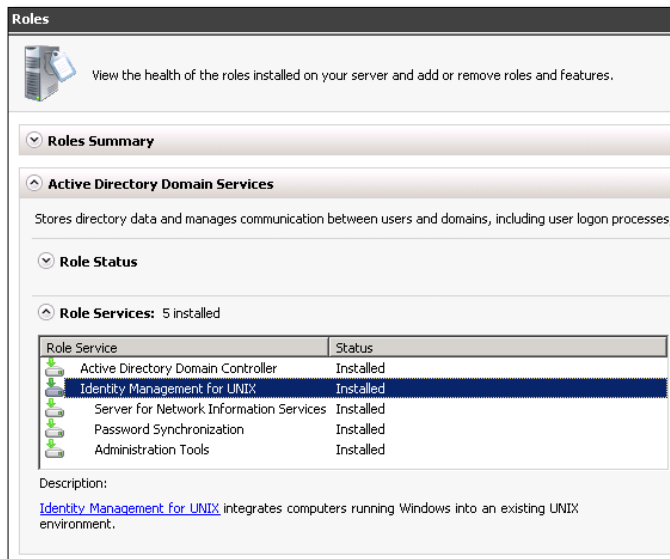
Extending the schema in Active Directory enables lookups of UNIX-style attributes. However, these attributes are not added to the global catalog for replication. For information on adding UNIX attributes to the global catalog, see the section in this document on [“Replicating New Attributes to the Global Catalog.”](#)

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](#).

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 Add Role Services and follow the prompts. After this installation finishes, the Active Directory schema is extended and new attributes are available for modification. In addition, new tabs are available on user and group properties, such as the UNIX Attributes tab.

Extending the Schema in Windows 2012

Windows 2012 allows schema extension only for UNIX user attributes using the command line. For more information on how that is done, see "[Installing Identity Management for UNIX by Using a Command Line](#)" on Microsoft's Technet site.

Example of installation:

```
Windows PowerShell
Copyright (C) 2013 Microsoft Corporation. All rights reserved.

PS C:\Users\Administrator> Dism.exe /online /enable-feature /featurename:adminui /all

Deployment Image Servicing and Management tool
Version: 6.3.9600.16384

Image Version: 6.3.9600.16384

Enabling feature(s)
[=====100.0%=====]
The operation completed successfully.
Restart Windows to complete this operation.
Do you want to restart the computer now? (Y/N) n

PS C:\Users\Administrator> Dism.exe /online /enable-feature /featurename:nis /all

Deployment Image Servicing and Management tool
Version: 6.3.9600.16384

Image Version: 6.3.9600.16384

Enabling feature(s)
[=====100.0%=====]
The operation completed successfully.
Restart Windows to complete this operation.
Do you want to restart the computer now? (Y/N) n

PS C:\Users\Administrator> Dism.exe /online /enable-feature /featurename:psync /all

Deployment Image Servicing and Management tool
Version: 6.3.9600.16384

Image Version: 6.3.9600.16384

Enabling feature(s)
[=====100.0%=====]
The operation completed successfully.
Restart Windows to complete this operation.
Do you want to restart the computer now? (Y/N) n
```

LDAP SRV Records

LDAP SRV records should exist in the domain. By default, [Active Directory creates these records](#).

To see if the SRV records exist:

```
[root@linux-client/]# dig SRV _ldap._tcp.domain.netapp.com
```

Note: 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.

Assigning UNIX Attributes

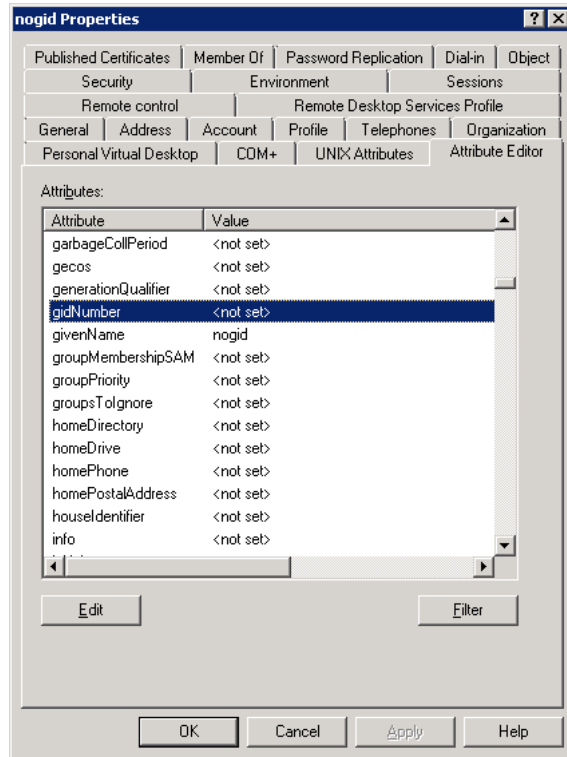
The UNIX Attributes tab allows an administrator to assign a UID and a default GID to user objects for use by LDAP. If no UID/GID is specified, then the object cannot be used for multiprotocol access.

The GID is 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 is 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.

Best Practices 32: gidNumber Recommendations (see next: Best Practices 33)

Every user in LDAP *must* have a gidNumber assigned. Otherwise, LDAP lookups fail. This point is true of all LDAP clients, including those that are not NetApp.

Figure 4) User without gidNumber set in LDAP.



```
cluster::*> diag secd authentication show-creds -node node2 -vserver SVM -unix-user-name nogid -
list-id true -list-name true

Vserver: SVM (internal ID: 3)

Error: Acquire UNIX credentials procedure failed
[ 0 ms] Connecting to LDAP (NIS & Name Mapping) server
        10.228.225.120
[  5] Using a new connection to 10.228.225.120
[  8] Failed to get a user ID for name 'nogid' using UNIX
        authorization source LDAP, Error: 1057
[  8] Name 'nogid' not found in UNIX authorization source LOCAL
[  8] Could not get a user ID for name 'nogid' using any
        NS-SWITCH authorization source
**[  8] FAILURE: Unable to retrieve UID for UNIX user nogid

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

This affects LDAP clients such as SSSD as well:

```
# getent passwd nogid
# id nogid
id: nogid: No such user
# id test
uid=10001(test) gid=513(domain users) groups=513(domain users),10011(ldifde-group)
```

If the gidNumber assigned to the user is present but cannot be resolved in LDAP, Data ONTAP behavior varies depending on the version of Data ONTAP used.

- In 8.2.1 and later, the lookup for the uidNumber succeeds, but there is a gidNumber error.
- In versions before 8.2.1, the authentication fails completely.

Note: See [bug 699947](#) for more information.

Example of user credential lookup with invalid gidNumber in 8.2.1 and later:

```
cluster::*> diag secd authentication show-creds -node node2 -vserver SVM -unix-user-name nogid -
list-id true -list-name true

UNIX UID: 1013787 (nogid) <> Windows User: S-1-5-21-3413584004-3312044262-250399859-1240
(DOMAIN\nogid (Domain User))

GID: 1111
Windows Membership:
  S-1-5-21-3413584004-3312044262-250399859-1118      DOMAIN\testgroup (Domain group)
  S-1-5-21-3413584004-3312044262-250399859-513      DOMAIN\Domain Users (Domain group)
  S-1-5-32-545      BUILTIN\Users (Alias)
User is also a member of Everyone, Authenticated Users, and Network Users

Privileges (0x80):

Vserver: SVM (internal ID: 3)

Error: Acquire UNIX credentials procedure failed
 [ 0 ms] Using a cached connection to 10.228.225.120
 [ 2] ID 1111 not found in UNIX authorization source LDAP
 [ 2] ID 1111 not found in UNIX authorization source LOCAL
 [ 2] Could not get a group name for ID 1111 using any
      NS-SWITCH authorization source
**[ 2] FAILURE: Unable to retrieve UNIX groupname for GID 1111
```

If gidNumber attributes cannot be resolved, there can be negative effects where permissions are controlled on a group basis. If a group name cannot be resolved, permission might be denied for users in groups assigned to the ACL. Be sure that all gidNumbers assigned to users are valid and can be resolved in LDAP to make sure of predictable user access and permissions.

If secondary groups are desired for use with RFC-2307 schemas, then the group object in the directory must be modified to include users as members under the UNIX Attributes tab for the group. This populates the memberUid field in LDAP. When using [RFC-2307bis](#), this is not necessary. These members can be different from the users in the Memberstab. 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 Practices 33: UID/GID Selection Considerations (see next: Best Practices 34)

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

To get a SID for a user or group from a 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 preceding example, the user ldapuser is assigned a UID of 1011.

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

[Determining an SID for a User Account](#)

For configuration steps to set UNIX attributes in Active Directory LDAP, see the [corresponding section in this document](#).

Note: In Windows 2012 R2, the UNIX attribute management using the GUI is being deprecated. See [this Microsoft blog for more information](#). Windows Active Directory LDAP can still leverage the UNIX-style attributes, but management needs to be done using PowerShell, CLI, or third-party LDAP management tools (such as Centrify).

Viewing Attributes

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 following table shows the difference in schema attributes on a user before and after IDMU is installed on a Windows 2008 R2 domain controller. This table also is relevant for Windows 2012 LDAP schemas.

Note the UNIX attributes added after the schema was extended. Those attributes 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](#)

Commands used (`ldifde`):

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

Commands used (PowerShell):

```
PS C:\> Get-ADUser [username] -Properties * | Select *
PS C:\> Get-ADGroup [groupname] -Properties * | Select *
```

Table 10) Difference in schema attributes before/after extending the schema using ldifde .

Before Schema Extend	After Schema Extend
<pre>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+3rIspB91AvQ== userAccountControl: 512 primaryGroupID: 513 objectSid: AQUAAAAAAAAUVAWAWhBXSAYAJJSTfIIw/TwQAAA== sAMAccountName: ldapuser sAMAccountType: 805306368 userPrincipalName: ldapuser@netapp.com objectCategory: CN=Person,CN=Schema,CN=Configuration, DC=netapp,DC=com</pre>	<pre>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:: kD6gtuDo9UKeZ50/mqJLBg== userAccountControl: 66048 primaryGroupID: 513 objectSid: AQUAAAAAAAAUVAWAATaofhHZHSrXcjKDbYwQAAA== 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</pre>

Auxiliary GIDs

In Data ONTAP, there is a limit of 16 auxiliary GIDs per user for AUTH_GSS in 8.2 and earlier. In 8.2.1, that limit is increased to 32 auxiliary GIDs per user for AUTH_GSS. 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 using the `nfs.max_num_aux_groups` option introduced in Data ONTAP 7-Mode 7.3.2 and later. Data ONTAP 8.3 and later introduced an option for NFS servers to leverage extended support for auxiliary groups. The maximum for both AUTH_GSS and AUTH_SYS is 1,024 GIDs. For more information on this functionality, see [TR-4067: NFS Best Practice and Implementation Guide](#).

Best Practices 34: Considerations For Enabling Extended GIDs (see next: Best Practices 35)

If enabling the extended GID feature in Data ONTAP, it's important to make sure that the users/UIDs being queried actually exist in the name service server. If a user does not exist in a name service and is queried in a NFS request, access might be denied. This feature should only be enabled if users belong to more than 16 extended groups.

Using Multiple Domains in a Forest for UNIX User and Group Identities

With Active Directory, it is possible to set up a trust between two domains in a forest, as well as have child domains below those parent domains. This configuration can create complications with LDAP requests, because the default behavior is to leverage LDAP referrals, which Data ONTAP does not currently support. However, the use of global catalog searches for UNIX users and groups is supported and can be used to perform LDAP lookups across multiple domains in a forest.

Best Practices 35: Multiple Domains/Trusts: UNIX Identities (see next: Best Practices 36)

If you attempt to use Active Directory domains with UNIX users in multiple domains, NetApp highly recommends leveraging global catalog searches for UNIX user and group lookups. If doing so is not possible, other options could include:

- Local UNIX users and groups (not in excess of the maximum of 65,536 clusterwide)
- Creating new users in the target LDAP domain with UID/GID information as placeholders for other domains

Neither of the preceding workarounds is as simple or scalable as using global catalog searches.

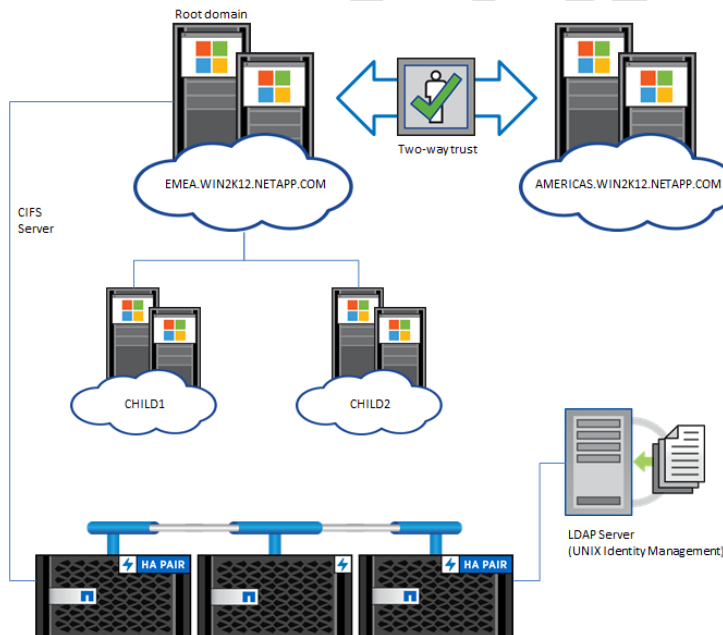
With global catalog searches, multiple domains in a forest can contain UNIX users and groups for the cluster to resolve.

Best Practices 36: Multiple Domains/Trusts: Multiprotocol NAS (see next: Best Practices 37)

If you attempt to leverage multiprotocol with Data ONTAP (NFS and CIFS access to the same SVM and data volumes) with LDAP serving the UID/GID identities and name mappings, consider child domains. If a domain trust has child domains, the CIFS server should be added to the domain that owns the child trusts.

In the following figure, the EMEA domain has child domains of FRANCE and GERMANY. The AMERICAS domain is trusted to EMEA.

Figure 5) Domain trust example with external LDAP server in separate forest.



Note: The CIFS server is added to EMEA, which is the parent domain to FRANCE and GERMANY

When the CIFS server lives in a parent domain, it is able to see the child and bidirectional trusts. This capability can be verified with the `vserver cifs domain trusts show` command.

Trusts show command: CIFS server in parent domain.

```
cluster::*> vserver cifs domain trusts show -vserver TRUST

Node: node01
Vserver: TRUST

Home Domain                Trusted Domains
-----
EMEA.WIN2K12.NETAPP.COM    FRANCE.EMEA.WIN2K12.NETAPP.COM,
                            GERMANY.EMEA.WIN2K12.NETAPP.COM,
                            AMERICAS.WIN2K12.NETAPP.COM,
                            EMEA.WIN2K12.NETAPP.COM
```

If the CIFS server does not live in the parent domain, then only the bidirectional trust is seen.

Trusts show command: CIFS server in trusted nonparent domain.

```
cluster::*> vserver cifs domain trusts show -vserver TRUST

Node: node01
Vserver: TRUST

Home Domain                Trusted Domains
-----
AMERICAS.WIN2K12.NETAPP.COM AMERICAS.WIN2K12.NETAPP.COM,
                            EMEA.WIN2K12.NETAPP.COM
```

This behavior is consistent with that of Windows domain controllers. This can be verified with the [nltest](#) command available in Active Directory.

Table 2) Examples of `nltest` in Windows trusted domains.

From the EMEA domain (listed as Native):

```
C:\>nltest /trusted_domains
List of domain trusts:
 0: FRANCE france.emea.win2k12.netapp.com (NT 5) (Forest: 3) (Direct Outbound)
    (Direct Inbound) ( Attr: 0x20 )
 1: GERMANY germany.emea.win2k12.netapp.com (NT 5) (Forest: 3) (Direct Outbound)
    (Direct Inbound) ( Attr: 0x20 )
 2: AMERICAS americas.win2k12.netapp.com (NT 5) (Direct Outbound)
    (Direct Inbound) ( Attr: 0x8 )
 3: EMEA emea.win2k12.netapp.com (NT 5) (Forest Tree Root) (Primary Domain) (Native)
The command completed successfully
```

From the AMERICAS domain (listed as Native):

```
C:\>nltest /trusted_domains
List of domain trusts:
 0: EMEA emea.win2k12.netapp.com (NT 5) (Direct Outbound) (Direct Inbound) (Attr: 0x8 )
 1: AMERICAS americas.win2k12.netapp.com (NT 5) (Forest Tree Root) (Primary Domain) (Native)
The command completed successfully
```

If the Windows domain cannot see a trust, then the CIFS server in the SVM cannot see the trust either. As a result, if Windows users exist in the child domain, they cannot be seen by the CIFS server when leveraging name mapping.

Best Practices 37: Multiple Domains/Trusts: Name Map Search (see next: Best Practices 38)

When using multiple domains for UNIX users and group identity sources with multiprotocol, [name mapping search rules](#) must be created to reflect the domains where the UNIX users exist.

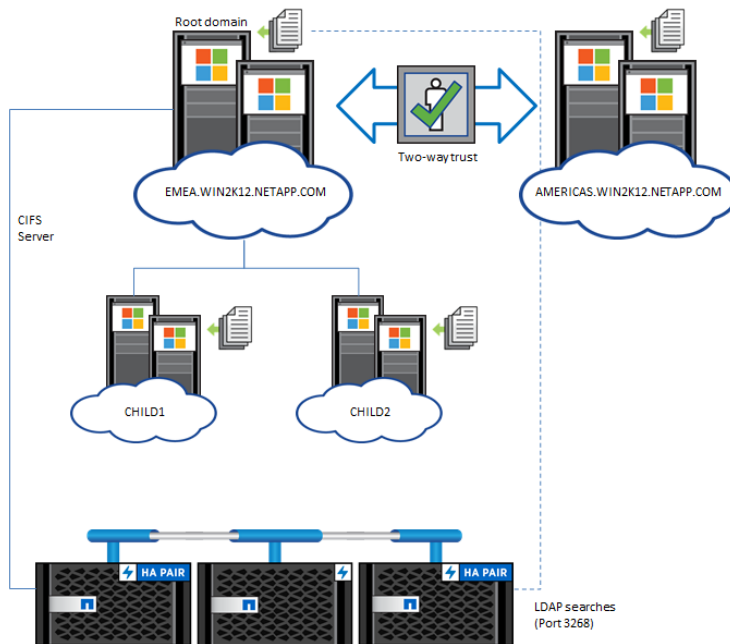
Replicating New Attributes to the Global Catalog

A [global catalog](#) is a domain controller that stores a copy of all Active Directory objects in a forest. The global catalog stores a full copy of all objects in the directory for its host domain and a partial copy of all objects for all other domains in the forest.

UNIX attributes are not inherently used by Active Directory, hence the need to use schema extensions. Additionally, these attributes are not replicated to the global catalog by default, and therefore cannot be searched in the global catalog. Instead, Active Directory [issues LDAP referrals](#) to find these objects. Data ONTAP does not support LDAP referrals, but it does support the use of a global catalog search. To configure Data ONTAP to use global catalog searches, see the section of this document called [“Configuring Data ONTAP LDAP Clients to Use Global Catalog Searches.”](#)

The following figure shows an example of a trusted domain setup that could be leveraged to use LDAP lookups using global catalog searches.

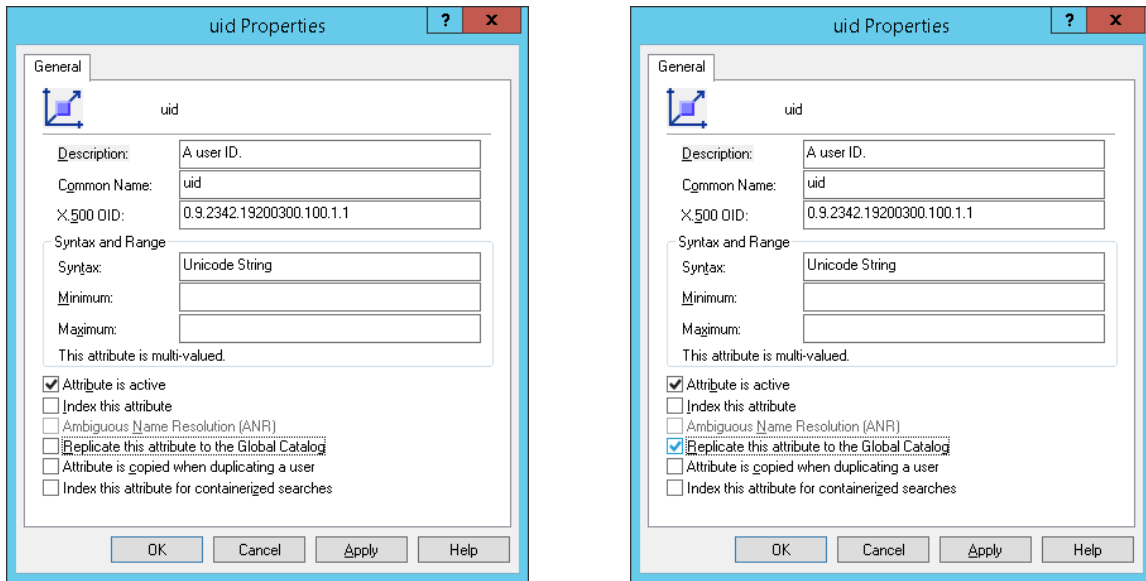
Figure 6) Trusted domain using global catalog searches.



How to Configure Attributes to Replicate to the Global Catalog

Active Directory uses a back-end schema to control how objects operate. This schema can be modified, but doing so requires special steps. It is best to contact Microsoft when modifying the schema, but documentation is available on how to [modify the partial attribute set to replicate](#). The following figure shows what a schema attribute might look like before and after setting it to replicate.

Figure 7) Modifying global catalog attributes to replicate.



The following attributes should be modified to replicate to enable global catalog LDAP searches to work with Data ONTAP:

```
gecos
gidNumber
memberUid
msSFU30Name
msSFU30NisDomain
msSFU30PosixMemberOf
any other populated msSFU30 attributes (Windows 2003 and prior - AD-SFU)
nisMapName (if using netgroups)
nisMapEntry (if using netgroups)
nisNetgroupTriple (if using netgroups)
uid
uidNumber
unixHomeDirectory
unixUserPassword
```

Keep in mind the following caveats:

- Using global catalog (GC) servers for searches can add significant load and traffic to these servers. If you use GC for LDAP searches, be sure there are enough servers available to handle the load.
- Modifying the Active Directory schema can be very dangerous. Do so with caution and document all changes in extreme detail.

After the change is made to replicate attributes to the global catalog, administrators can either wait for the 15-minute replication window or [force replication](#) using Active Directory Sites and Services.

Setting Name-Mapping Rules in LDAP

LDAP can map Windows user names to UNIX user names (and vice versa) on a 1:1 (symmetric) basis. It can also be used to map UNIX user names that differ from their Windows counterparts without the need to create name-mapping rules on the storage system.

Note: Data ONTAP supports asymmetric credential fetching from LDAP-based name mapping of Windows to UNIX user accounts starting in 8.3.2.

Order of Operations for Name Mappings in ONTAP

When a user attempts to authenticate to a NAS mount or share, ONTAP will use a specific order of name mapping mechanisms to look for valid users or name map entries. This will ultimately depend on the first name service database value specified for the `namemap` value in `vserver services name-service ns-switch`. In the following example, ONTAP will try local files first and then LDAP. “Local files” for `namemap` values means the entries in the SVM’s name mapping table in `vserver name-mapping`.

```
cluster::> vserver services name-service ns-switch show -vserver DEMO -database namemap

                Vserver: DEMO
Name Service Switch Database: namemap
Name Service Source Order: files, ldap
```

When using LDAP for name mapping, ONTAP will use whatever the LDAP server is configured to use. In most cases, this will be a symmetric name mapping (see below). But it’s also possible to use asymmetric values.

Best Practices 38: Specifying external services in `namemap` (see next: Best Practices 39)

Only specify an external service in the `namemap` database if one is actually being used for asymmetric name mappings. If you specify a server that does not have any name mapping rules configured, this will add latency to requests and create slow authentication or failures.

If no name mapping can be found in the name services entries for the user, then ONTAP will try to fall back on the default values set for the NFS or CIFS/SMB server. The use of this value will depend on the protocol attempting access, the volume security style and the name mapping direction requested. The following table shows the differences.

Table 3) Name mapping/default user considerations for multiprotocol NAS access

Protocol	Volume/mtree security style	Name mapping direction	Default user
NFS	UNIX	N/A (UID lookup only)	N/A
NFS	NTFS	UNIX -> Windows	Default Windows user (NFS server option <code>default-win-user</code>)
CIFS/SMB	UNIX	Windows -> UNIX	Default UNIX user (CIFS server option <code>default-unix-user</code> ; <code>pcuser</code> by default)
CIFS/SMB	NTFS	Windows -> UNIX (initial authentication) NTFS ACLs used after initial entry.	Default UNIX user (CIFS server option <code>default-unix-user</code> ; <code>pcuser</code> by default)

What Are Symmetric and Asymmetric Name Mappings?

Symmetric name mapping is implicit name mapping between UNIX and Windows users who leverage the same user name; for example, Windows user `DOMAIN\justin` maps to UNIX user `justin`.

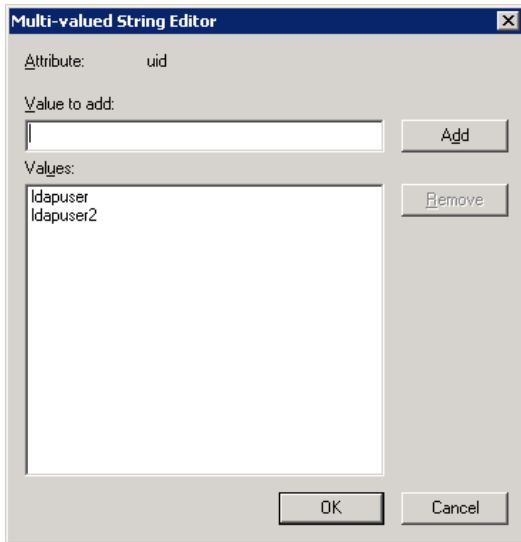
Asymmetric name mapping is name mapping between UNIX and Windows users who leverage different user names; for example, Windows user `DOMAIN\justin` maps to UNIX user `nfsdudeabides`.

What If I Need Asymmetric Name Mapping from Windows to UNIX Users in LDAP?

If an environment relies on bidirectional asymmetric name mapping from LDAP in Data ONTAP, name mapping rules should be created per SVM for the Windows-UNIX name mappings. However, there is a *limit of 1,024 rules per SVM*. If more rules are needed than are allowed by the cluster, then the LDAP server attributes must be modified to include a UNIX user name with the same value as the Windows user name. Clients still pick up the desired UID/GID in this case.

Before Data ONTAP versions 8.2.4 and 8.3.2, if my user DOMAIN\ldapuser needed to map to UNIX user ldapuser2 in both directions, I needed to add two names to the uid field in LDAP:

Figure 8) Example of adding multiple UIDs to LDAP in Active Directory.



This behavior was covered in [bug 913673](#). In the following example, even though Data ONTAP is mapping DOMAIN\ldapuser to ldapuser, the client translates the user based on its own passwd/LDAP settings.

```
cluster::*> diag secd name-mapping show -node node1 -vserver SVM -direction win-unix -name
DOMAIN\ldapuser
DOMAIN\ldapuser maps to ldapuser

cluster::*> diag secd name-mapping show -node node1 -vserver SVM -direction unix-win -name
ldapuser2
ldapuser2 maps to ldapuser
```

Why Didn't It Work?

Data ONTAP operating in 7-Mode had a series of [hidden LDAP options](#) that allowed specific functionality for certain use cases. When using LDAP for name mapping, the default behavior is to attempt a symmetric name mapping as defined by the hidden option `ldap.usermap.symmetriclookup`. This option is not present in current Data ONTAP versions.

Additionally, the hidden options to define the UNIX user attribute for Windows to UNIX users are also not available in Data ONTAP. These options include:

```
ldap.usermap.windows-to-unix.objectClass
ldap.usermap.attribute.unixaccount
```

[Bug 913673](#) was filed to address this issue and was fixed in Data ONTAP 8.2.4 and 8.3.2.

How Data ONTAP 8.2.4 and 8.3.2 Solve This Problem

Data ONTAP 8.2.4 and 8.3.2 (and later) implemented additional LDAP schema attributes to allow parity for LDAP name mapping with 7-Mode. The following new LDAP client schema options were added:

New LDAP Schema Attribute	What It Does
<code>-windows-to-unix-object-class</code>	Provides the LDAP attribute to define the Windows to UNIX name mapping object class. Object classes are used to group multiple LDAP objects to enable faster searches. The default value for this in AD-IDMU is User. For RFC-2037 schemas, the value is set to posixAccount.
<code>-windows-to-unix-attribute</code>	Provides the LDAP attribute for the value that will be used for mapping a Windows user to a UNIX user. The default value for AD-IDMU schemas in Data ONTAP is sAMAccountName. For RFC-2307 schemas, the value defaults to windowsAccount.
<code>-windows-to-unix-no-domain-prefix</code>	This option controls whether or not the attribute value in <code>-windows-to-unix-attribute</code> has the domain prefix added to it. (The default is false.) Because sAMAccountName is represented by a single user name (rather than DOMAIN\username) and because msDS-PrincipalName is not a value that can be used in LDAP search, domain prefixes might be necessary to enable functional asymmetric name mapping. The need for this value depends on the LDAP schema and attributes being used as well as if multiple domain name mappings are present for multiple unique Windows domains. See the section “Name Mapping Across Multiple Domains” for details.

These options allow bidirectional asymmetric name mappings from both Windows to UNIX and UNIX to Windows from LDAP servers. The attribute values for these options depend on what your environment looks like.

For most Active Directory LDAP servers, the values for asymmetric name mappings are as follows:

```
-windows-to-unix-object-class User
-windows-account-attribute sAMAccountName
-windows-to-unix-attribute sAMAccountName
-windows-to-unix-no-domain-prefix true
```

With the above values, Active Directory Identity Management works with name mappings out of the box. Any variations on the default schemas need to be accounted for.

Example of working UNIX -> Windows and Windows -> Unix name mappings:

```
cluster::*> diag secd authentication show-creds -node nodel -vserver SVM -win-name ldapuser

UNIX UID: ldapuser2 <> Windows User: DOMAIN\ldapuser (Windows Domain User)

GID: ldapgroup
Supplementary GIDs:
  ldapgroup

Windows Membership:
  DOMAIN\Domain Users (Windows Domain group)
  BUILTIN\Users (Windows Alias)
User is also a member of Everyone, Authenticated Users, and Network Users

Privileges (0x2080):
  SeChangeNotifyPrivilege

cluster::*> diag secd authentication show-creds -node nodel -vserver SVM -unix-user-name
ldapuser2

UNIX UID: ldapuser2 <> Windows User: DOMAIN\ldapuser (Windows Domain User)

GID: ldapgroup
Supplementary GIDs:
  ldapgroup

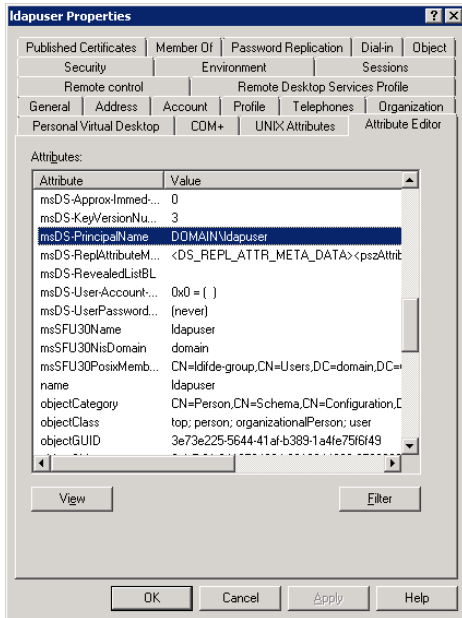
Windows Membership:
  DOMAIN\Domain Users (Windows Domain group)
  BUILTIN\Users (Windows Alias)
User is also a member of Everyone, Authenticated Users, and Network Users

Privileges (0x2080):
  SeChangeNotifyPrivilege
```

Asymmetric Mapping of UNIX Users to Windows Users

The LDAP schema defined in Data ONTAP contains an attribute called “ONTAP Name Mapping windowsAccount Attribute” that defines which LDAP schema attribute to use when mapping UNIX names to Windows names. The default value of this attribute is `windowsAccount` in 8.2.1 and earlier, which does not exist by default in Windows Active Directory LDAP schemas. In 8.2.2 and later, the value changes to `msDs-PrincipalName`, which is not a modifiable attribute in LDAP. However, according to [this Microsoft blog](#), this value is considered a “constructed attribute” and is not considered usable in LDAP search filters. Although UNIX to Windows name mappings still technically work, the attribute may be used in other SecD processes and fail with a `WRONG_MATCH_OPER` error. Therefore, it makes sense to change this attribute in schemas to `sAMAccountName` to prevent an incorrect LDAP query. Bug 914421 was opened for this issue and fixed in Data ONTAP versions 8.2.4 and 8.3.2 and later. For information on creating custom LDAP schemas, see the section in this document called “[Creating a Custom LDAP Schema](#).”

Figure 9) Example of msDs-PrincipalName field.



For [configuration steps concerning user mapping in LDAP](#), see the corresponding section in this document.

Name Mapping Across Multiple Domains

In Data ONTAP 8.2.1 and later (clustered), it is possible to configure multiple domains for use with implicit user mappings. This capability can be leveraged with Active Directory LDAP for UNIX and Windows user name lookups, provided that the following are in place:

- Functional domain with bidirectional trusts
- Active Directory global catalog domain controllers acting as LDAP servers with UNIX attributes
- [Global catalog searches for UNIX users](#) in Active Directory
- Data ONTAP 8.2.1 and later with the following configuration:
 - Properly configured DNS
 - CIFS server created in the forest
 - Domain trusts appearing properly with the `vserver cifs domain trusts show` command
 - Multidomain name mapping search configured to include the desired name mapping search domains
 - Unix-win usermapping rule of `* == */*` in the SVM

Example command to create the rule:

```
cluster::*> vserver name-mapping create -vserver TRUST -direction unix-win -pattern *  
-replacement */*
```

The following is a sample configuration in Data ONTAP in which multiple domain name mapping is configured.

```
cluster::*> vserver cifs server show -vserver TRUST -instance

                Vserver: TRUST
                CIFS Server NetBIOS Name: CIFS
                NetBIOS Domain/Workgroup Name: EMEA
                Fully Qualified Domain Name: EMEA.WIN2K12.NETAPP.COM
Default Site Used by LIFs Without Site Membership:
                Authentication Style: domain
                CIFS Server Administrative Status: up
                CIFS Server Description:
                List of NetBIOS Aliases: -

cluster::*> dns show -vserver TRUST -instance

                Vserver: TRUST
                Domains: emea.win2k12.netapp.com, americas.win2k12.netapp.com,
                        france.emea.win2k12.netapp.com
                Name Servers: 10.228.225.125, 10.228.225.122, 10.228.225.123
                Enable/Disable DNS: enabled
                Timeout (secs): 2
                Maximum Attempts: 1

cluster::*> vserver cifs domain trusts show -vserver TRUST

                Node: node01
                Vserver: TRUST

Home Domain          Trusted Domains
-----
EMEA.WIN2K12.NETAPP.COM  FRANCE.EMEA.WIN2K12.NETAPP.COM,
                        GERMANY.EMEA.WIN2K12.NETAPP.COM,
                        AMERICAS.WIN2K12.NETAPP.COM,
                        EMEA.WIN2K12.NETAPP.COM

cluster::*> name-mapping-search show
(vserver cifs domain name-mapping-search show)
Vserver      Trusted Domains
-----
TRUST        EMEA.WIN2K12.NETAPP.COM, AMERICAS.WIN2K12.NETAPP.COM,
                FRANCE.EMEA.WIN2K12.NETAPP.COM,
                GERMANY.EMEA.WIN2K12.NETAPP.COM

cluster::*> vserver name-mapping show -vserver TRUST -instance

                Vserver: TRUST
                Direction: unix-win
                Position: 1
                Pattern: *
                Replacement: *\\*
```

When the preceding is configured properly, the SVM can grab credentials from anywhere in a domain.

Note: It is crucial that user names/UIDs do not overlap in the domain.

The following are examples of cross-domain name mapping at work.

User in AMERICAS.WIN2K12.NETAPP.COM trusted domain:

```
cluster::*> diag secd authentication show-creds -node nodel -vserver TRUST -unix-user-name
ldapuser

UNIX UID: ldapuser <> Windows User: AMERICAS\ldapuser (Windows Domain User)

GID: unigroup-emea
Supplementary GIDs:
  unigroup-emea

Windows Membership:
  AMERICAS\Domain Users (Windows Domain group)
User is also a member of Everyone, Authenticated Users, and Network Users

Privileges (0x2000):
  SeChangeNotifyPrivilege
```

User in FRANCE.EMEA.WIN2K12.NETAPP.COM child domain:

```
cluster::*> diag secd authentication show-creds -node nodel -vserver TRUST -unix-user-name
unix-france

UNIX UID: unix-france <> Windows User: FRANCE\unix-france (Windows Domain User)

GID: unixgroup-france
Supplementary GIDs:
  unixgroup-france

Windows Membership:
  FRANCE\Domain Users (Windows Domain group)
User is also a member of Everyone, Authenticated Users, and Network Users

Privileges (0x2000):
  SeChangeNotifyPrivilege
```

User in EMEA.WIN2K12.NETAPP.COM parent domain:

```
cluster::*> diag secd authentication show-creds -node parisi-fs-01 -vserver TRUST -unix-user-name
euro-user

UNIX UID: euro-user <> Windows User: EMEA\euro-user (Windows Domain User)

GID: unigroup-emea
Supplementary GIDs:
  unigroup-emea

Windows Membership:
  EMEA\Domain Users (Windows Domain group)
  BUILTIN\Users (Windows Alias)
User is also a member of Everyone, Authenticated Users, and Network Users

Privileges (0x2080):
  SeChangeNotifyPrivilege
```

For more information on multiple domain name mappings, see the product documentation and [TR-4191: Best Practices Guide for Data ONTAP 8.2.x Windows File Services](#).

Netgroups in LDAP

It is possible to leverage netgroup functionality in LDAP as opposed to NIS. Netgroups allow storage administrators to control access to a series of hosts using a group, rather than needing to create a number of different rules per host. In LDAP, host names, IP addresses, and netgroup entries can be stored and queried using Data ONTAP. Using LDAP as a NIS server is covered in [RFC-2307](#). Currently, only host names and IP addresses are supported for use in Data ONTAP.

Best Practices 39: Data ONTAP Version: Netgroups (see next: Best Practices 40)

If you use netgroups (NIS or LDAP) in Data ONTAP leveraging host names, be sure to use Data ONTAP version 8.2.2 or later. Using this version is needed because versions before 8.2.2 did not handle host names in netgroups as efficiently.

About NIS Objects and Attributes in LDAP

NIS object types in LDAP are determined by way of the `objectClass` attribute. The `objectClass` attribute set on an object determines how Data ONTAP and other LDAP clients query LDAP for netgroup-related objects. For netgroups, the `nisNetgroup` object class is used by default.

Table 4) Object class types for NIS objects in Active Directory.

objectClass	Used For	NIS Attributes Used
nisMap	NIS Maps	nisMapName
nisNetgroup	Netgroups	nisMapName nisNetgroupTriple
nisObject	Netgroups Netgroup.byhost entries	nisMapEntry nisMapName

NIS Object Terminology

The following section describes terminology that defines specific aspects of NIS objects.

Table 5) NIS object terminology.

Term	Definition
NIS Map	<p>NIS maps were designed to centralize and replace common files found in the <code>/etc</code> directory of Linux and UNIX clients.</p> <p>Data ONTAP currently supports the following NIS map types:</p> <ul style="list-style-type: none">• <code>Passwd.byname</code> and <code>passwd.byuid</code>• <code>Group.byname</code> and <code>group.bygid</code>• <code>Netgroup</code>• <code>Netgroup.byhost</code> (as of 8.3) <p>Support for host name resolution in NIS is not currently supported.</p> <p>For more information on NIS maps, see http://docs.oracle.com/cd/E19683-01/817-4843/anis1-24268/index.html.</p>

Term	Definition
Netgroup	A netgroup is a set of (host,user,domain) triples used for permission and export access checking. Data ONTAP currently supports only hosts in netgroup entries. The netgroup must use only a comma (,) as the delimiter. For more information on netgroups, see: http://linux.die.net/man/5/netgroup and http://www.freebsd.org/cgi/man.cgi?query=netgroup&sektion=5 .
Triple	A netgroup triple refers to the series of entries in a netgroup file consisting of (host,user,domain). A valid triple used in Data ONTAP consists of (host,,). Be sure when designating a blank field to use only the format mentioned previously. Special characters, such as dashes, can cause lookups to fail and access to be denied. Host names used in netgroup triples require DNS resolution in Data ONTAP. For best results in netgroup translation, see the name services best practices in TR-4067 and TR-4379 .
Netgroup.byhost	Netgroup.byhost entries are used to speed up netgroup lookups by querying the name service for the group membership by host rather than querying the entire netgroup. For netgroups with many entries, this process can reduce lookup time drastically and improve performance. For more information on netgroup.byhost support, see the section regarding this feature in this document .

Data ONTAP Interaction with Active Directory LDAP for Netgroups

In the schemas provided in Data ONTAP, the following attributes control lookups for netgroups and their members:

```
-nis-netgroup-object-class
-nis-netgroup-triple-attribute
-member-nis-netgroup-attribute
-cn-netgroup-attribute
```

Starting in Data ONTAP 8.3, the following attributes are provided for [netgroup.byhost support](#):

```
-nis-object-class
-nis-mapname-attribute
-nis-mapentry-attribute
```

LDAP client schemas can be modified to change the default attributes. For more information on default schemas, see the section in this document on LDAP schemas.

When Server for NIS is installed, the container DefaultMigrationContainer30 is created. This container is the container to which NIS netgroups are migrated by default. To use a different container, create a new OU or container to host this information and specify it in your migrations.

The Active Directory schema has the following schema attributes added by default in Windows 2008 and later (default attributes used by Data ONTAP in bold):

```
memberNisNetgroup
msSFU-30-Netgroup-Host-At-Domain
msSFU-30-Netgroup-User-At-Domain
msSFU-30-Nis-Domain
msSFU-30-Nis-Map-Config
msSFU-30-Yp-Servers
NisMap
NisMapEntry
NisMapName
NisNetgroup
NisNetgroupTriple
NisObject
```

Creating Netgroups in AD-Based LDAP

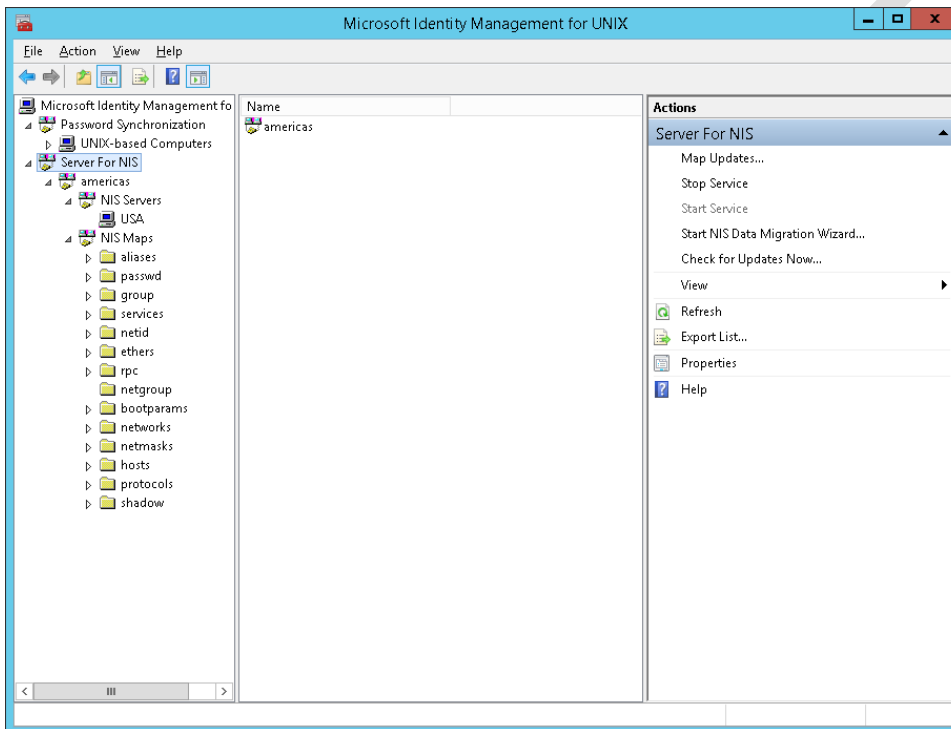
Active Directory netgroups can be controlled using the utilities “nis2ad” and “nismap” or using GUI tools such as [ADSI Edit](#).

Nis2ad allows migration of existing maps from NIS to AD, or the ability to create NIS maps from a local file. This utility is included in the Identity Management for UNIX feature in Windows 2008 and later. However, it generally is not needed unless you create new NIS maps outside of the default “netgroup” NIS map created by IDMU.

The `nismap` command allows granular management of NIS maps in addition to what `nis2ad` provides.

When Identity Management is installed with Server for NIS, an MMC is created to view and manage Server for NIS. The Server for NIS MMC cannot be used to create or delete NIS maps, however.

Figure 10) Server for NIS MMC.



By default, the NIS domain becomes the short name of the domain in which it was installed. In the preceding example, “americas” becomes the NIS domain. “Netgroup” is one of the default NIS maps.

Adding Netgroups in AD LDAP

Because AD LDAP creates a NIS server and domain when a NIS server is installed, all that remains is adding the netgroup entries.

There are multiple ways to create NIS objects such as netgroups in AD LDAP. One way is to leverage the `nismap` command.

Using nismap to Create NIS Objects

The following shows a sample command as well as the other options for syntax:

Configuration Steps 1) Using the nismap command to create a netgroup in AD LDAP.

```
C:\> nismap add -a [NIS domain] -s [NIS server] -c [C:\conflict_file] -e [netgroup_entry]
[nismap]

C:\>nismap add
Invalid arguments. No value for -a option. nis domain name under AD must be provided.

Usage: nismap [add | mod | del | create] [common_options] [specific_options] map

where
add          Add a map entry to NIS.
mod          Modify fields of a map entry.
del          Delete a map entry.
create       Create the configuration for a non-standard map.

map          Name of the NIS map.

common_options are :
-a AD_domain NIS domain name under AD. This option invalid for 'create'.
-f file      Name of the log file. Uses a default if not specified.
-s server    Name of the domain controller.
-u user      User name with administrative privileges.
             Uses current user if not specified.
-p password  User password. Prompts if required and not specified.
-h/-?       Displays this message.

add specific options :
-e map entry Quoted map entry string in the NIS map format.
-r yes/no    Replace existing object in AD with object to be migrated. Default is no.
-c file      File name to which conflict details are written.
             Uses default conflict file if not specified.

mod specific options :
-e map entry Quoted map entry string in the NIS map format.
-k key       Search key.

del specific options :
-k key       Search key.

create specific options :
-i fieldnum  Field number of the key field.
-g separator Field separator(single character other than '#').
-y          Remove key from value for this map.
```

The **-e** flag lists the netgroup/nismap entry. The format follows the same formats used in NIS netgroup files and covered in the [UNIX man pages](#).

The following is a sample netgroup entry:

```
netgroup (host1,,)
```

This host does not exist in DNS:

```
C:\>nslookup host1
DNS request timed out.
    timeout was 2 seconds.
Server: UnKnown
Address:  ::1

*** UnKnown can't find host1: Non-existent domain
```

Note: The ipHostNumber attribute is used when doing lookups of the NIS host IP information for most LDAP servers and clients. Data ONTAP does not use this attribute yet.

Example of netgroup named “hosts” created, in which “americas” is the NIS domain, which was created by default when the NIS server was installed:

```

C:\>nismap add -a americas -s USA -c C:\nisadd.txt -e "hosts
(host1,,) (host2,,)" netgroup
Activity = Adding map = 'netgroup'...
SUCCESS
Adding the object in Active Directory Domain Services.
Object = 'hosts'
Object class = 'NisNetgroup'
container =
'CN=netgroup,CN=americas,CN=DefaultMigrationContainer30,DC=americas,DC=win2k12,DC=netapp,DC=com'.

SUCCESS
adding NIS entries to AD

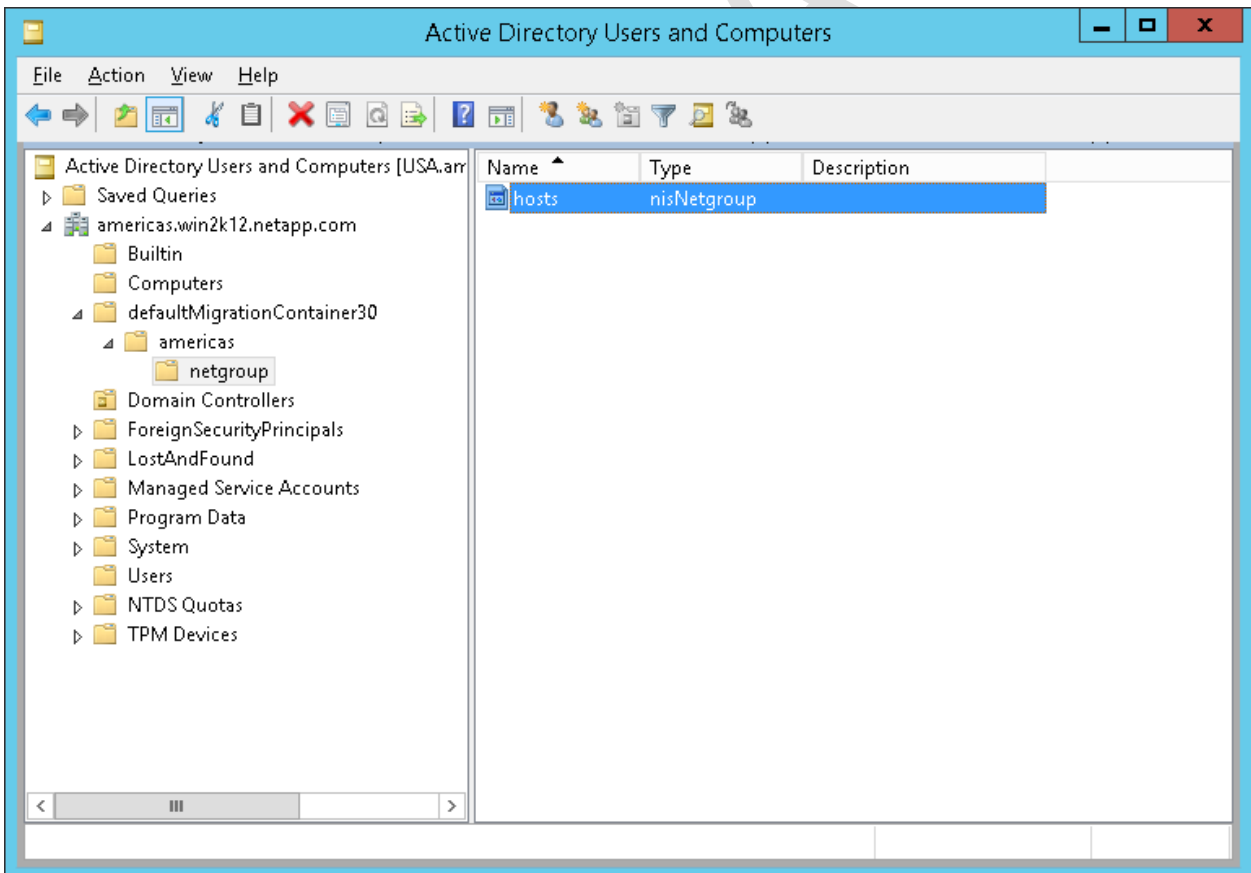
```

In the preceding, the following occurred:

- An object called “hosts” was created.
- The objectClass of “NisNetgroup” was applied to the object.
- The default container was
'CN=netgroup,CN=americas,CN=DefaultMigrationContainer30,DC=americas,DC=win2k12,DC=netapp,DC=com'.

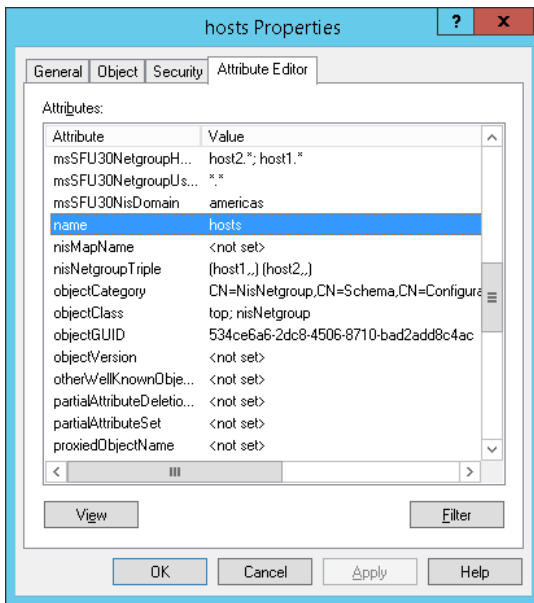
Note: The netgroup DN is used when configuring the LDAP client in Data ONTAP using the `-netgroup-dn` field.

Figure 11) Example of “hosts” netgroup created in AD LDAP.



The attributes can be viewed for the object by double-clicking and selecting Attribute Editor:

Figure 12) Netgroup properties in AD LDAP.



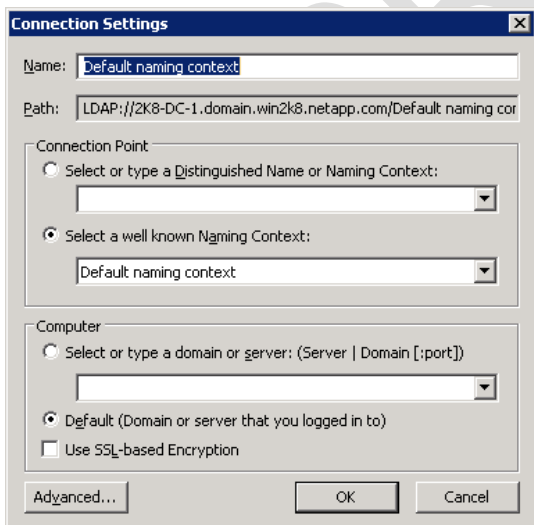
Using ADSI Edit to Create or Modify NIS Objects

Another way to create NIS objects is to use the ADSI Edit tool in Active Directory. This method is much simpler and more straightforward than using the CLI. To use ADSI Edit, make sure that it is installed. For more information, see the [TechNet article on ADSI Edit](#).

Note: ADSI Edit should be used with extreme caution, because serious damage to the Active Directory schema can be done if it is not used correctly. If you need assistance using ADSI Edit, contact Microsoft Technical Support.

After ADSI Edit is installed, open the ADSI Edit console and connect to the default Naming Context path.

Figure 13) Connecting to default naming context.



After you are connected, the entire Active Directory schema is shown. If a container for NIS objects does not already exist, it might make sense to create one for organizational purposes.

To create a container, right-click when the desired location is highlighted and select New -> Object.

For configuration steps to create netgroup objects in Active Directory, see the configuration section of this document on [creating netgroups in Active Directory LDAP](#).

Using Data ONTAP to Query Netgroup Information

When Data ONTAP does LDAP lookups for netgroups using the default AD-IDMU schema, it looks for the following attributes:

```
RFC 2307 nisNetgroup Object Class: nisNetgroup
RFC 2307 cn (for Netgroups) Attribute: name
RFC 2307 memberNisNetgroup Attribute: memberNisNetgroup
RFC 2307 nisNetgroupTriple Attribute: nisNetgroupTriple
```

Note: Before Data ONTAP 8.3.x, all netgroup queries were performed with SecD. After 8.3, netgroup queries take place through a standard libc call. As such, troubleshooting netgroups with the CLI varies depending on the version of Data ONTAP you run.

To query LDAP for the hosts in the netgroup using Data ONTAP versions before 8.3, use the `diag secd` command found at the *diag privilege* level:

```
cluster::> set diag
cluster::*> diag secd netgroup show-hosts -node node1 -vserver NFS -netgroup-name hosts
host1
```

To query LDAP for the host IP addresses, use the following `diag secd` command at the *diag* level:

```
cluster::*> diag secd netgroup show-host-addresses -node node2 -vserver NFS -netgroup-name hosts
10.10.10.10
```

If a name cannot be resolved using DNS or LDAP or the hosts do not have the correct attribute set, the lookup might fail with the following error:

```
cluster::*> diag secd netgroup show-host-addresses -node node2 -vserver NFS -netgroup-name hosts
Error: command failed: Failed to lookup host addresses for netgroup "hosts". Reason: invalid
Pointer
```

Be sure the hosts are either in DNS or LDAP and that the host entries are configured properly in LDAP to make sure that netgroup lookup works properly.

Expanding All Hosts or IPs in a Netgroup

In Data ONTAP 8.2.x and earlier, a cluster administrator could query a name service server for all hosts or host IP addresses in a netgroup with the following commands:

```
diag secd show-hosts
diag secd show-host-addresses
```

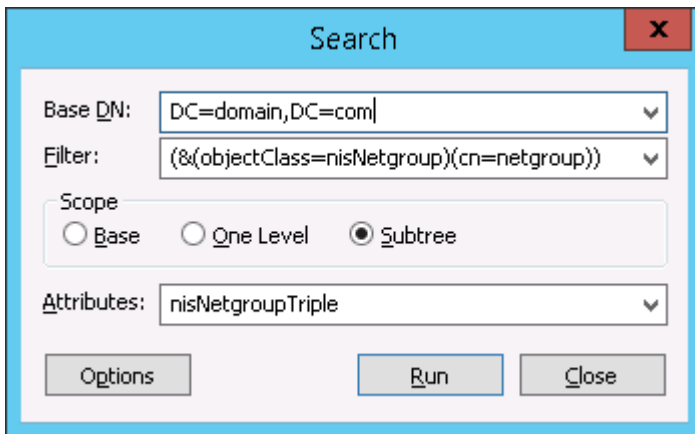
These commands expand a netgroup and allow the administrator to see the contents, which could be useful in troubleshooting netgroup issues.

One side effect of the changes made to netgroups in 8.3.x is that cluster administrators can no longer use the cluster CLI to dump all host names or IP addresses in a single netgroup with the `diag secd show-hosts` and `diag secd show-host-addresses` commands.

Instead, `getXXbyYY` is intended to be used to query individual netgroups and hosts. In addition, the command `export-policy check-access` can be used to verify if individual hosts have access to exports.

Dumping a netgroup's contents is no longer supported with the cluster CLI. Instead, name services need to be queried to find this information. This querying can be done using third-party tools in LDAP, such as `ldapsearch` or `ldp.exe`.

Figure 14) Sample LDAP filter for a netgroup lookup using ldp.exe



Note: Attributes and filter vary depending on the LDAP schema.

With the filter shown above, a dump of the netgroup would show this:

```
-----
***Searching...
ldap_search_s(ld, "DC=domain,DC=com", 2, "(&(objectClass=nisNetgroup)(cn=netgroup))", attrList, 0, &msg)
Getting 1 entries:
Dn: CN=netgroup,OU=Netgroups,DC=domain,DC=com
    nisNetgroupTriple (3): (10.10.10.12,,); (10.10.10.11,,); (10.10.10.10,,);
```

If it is not possible to query the external servers using third-party tools, contact NetApp Technical Support and reference [bug number 880614](#).

Netgroups in the Data ONTAP 8.3.x Operating System—GetXXbyYY Support

The Data ONTAP 8.3 operating system introduced the command `getXXbyYY` (advanced privilege) for use with name service lookups. The `diag secd` commands used for netgroup resolution were deprecated and are no longer supported. See the section detailing [changes between Data ONTAP 8.2.x and 8.3](#) operating systems for more information.

Example of netgroup lookup using `getXXbyYY`:

```
cluster83::*> getxxbyyy netgrpbyhost -node node1 -vserver SVM -netgroup netgroup2 -clientIP
10.228.225.140
(vserver services name-service getxxbyyy netgrpbyhost)
Netgroup.byhost not enabled in all the configured sources
Hostname resolved to: centos65.domain.netapp.com
```

Example of user lookup using `getXXbyYY`:

```
cluster83::*> getxxbyyy getpwbyuid -node node1 -vserver SVM -userID 1107
(vserver services name-service getxxbyyy getpwbyuid)
pw_name: ldapuser2
pw_passwd:
pw_uid: 1107
pw_gid: 10005
pw_gecos: ldapuser
pw_dir: /home/CDOT/ldapuser
pw_shell: /bin/sh
```

Example of `diag secd` command that is no longer supported in 8.3:

```
cluster83::*> diag secd netgroup show-hosts -node node1 -vserver SVM -netgroup-name netgroup2
This command is not supported in this release.
```

Best Practices 40: Netgroups and DNS (see next: Best Practices 41)

When a host name is specified in `nisNetgroupTriple`, Data ONTAP attempts to do a bulk DNS lookup for that host. If the host does not exist in DNS, the request fails. Make sure that all hosts are added to DNS (forward and reverse lookups) for proper NIS netgroups functionality and performance. This best practice applies to netgroups in both LDAP and NIS. For more information, see [TR-4379: Name Services Best Practices](#).

Netgroup Caches in Data ONTAP

Data ONTAP uses several caches to store information such as host names and netgroups locally. This method is faster than having to retrieve this information from an external source each time it is required.

Export policies and rules control access to NFS exports. Each export policy contains rules, and each rule contains parameters to control client access. Some of these parameters require Data ONTAP to contact an external source such as DNS or NIS servers to resolve objects such as domain names, host names, or netgroups. Communications with external sources can have associated latency because of load, network, and so on. To improve performance, Data ONTAP reduces the amount of time it takes to resolve export policy rule objects by storing information locally in several caches.

One main disadvantage to using caches to store information locally is that if the information on the external name server was changed after Data ONTAP retrieved and stored it locally, the caches might contain outdated information. As a result, client access requests that should succeed could fail, and client access requests that should fail could succeed. To help avoid such issues, Data ONTAP flushes caches automatically after certain time periods and provides commands that allow you to view and manually flush some of the export policy caches.

Table 6) Caches and time to live (TTL).

Cache Name	Type of Information	TTL (in Minutes)
Access	All export policy rules	5
Name	Name to UID	1
ID	ID to name	1
Host	Host to IP	1
Netgroup	Netgroup to IP	15
Showmount	Export paths	5

Flushing Export Policy Caches (and Other NFS-Related Caches)

In versions before Data ONTAP 8.3, flushing export policy caches could be done only by making changes to export policy rules. Now, Data ONTAP offers a set of commands to allow manual flushing of export caches without needing to change existing policies. This command set is similar to `exportfs -f` available in Data ONTAP operating in 7-Mode and is done on a per-node, per-SVM basis.

```
cluster::> vserver export-policy cache flush -vserver vs0 -node node1 -cache
all      access  host    id      name    netgroup showmount
```

Manual Cache Flush Considerations

Flushing the export policy caches manually removes information from them that might be outdated because of the following reasons:

- A recent change to export policy rules

- A recent change to host name records in name servers
- A recent change to netgroup entries in name servers
- Recovering from a network outage that prevented netgroups from being fully loaded

Flushing the caches removes the outdated information and forces Data ONTAP to retrieve current information from the appropriate external resources.

Note: Processing of netgroups can be resource intensive. You should flush the netgroup cache only if you are trying to resolve a client access issue that is caused by a stale netgroup.

Displaying Netgroup Caches

In addition to being able to flush various caches, Data ONTAP 8.3 and later offer the capability to display netgroup caches as well as check netgroup membership.

To view netgroup caches:

```
cluster::> vserver export-policy netgroup cache show ?

[ -instance | -fields <fieldname>, ... ]
-vserver <vserver name>                Vserver
[[-netgroup] <text>]                    Name of the Netgroup
[-record-id <integer> ]                 Record ID
[ -is-getting-hosts {true|false} ]      Hosts Being Retrieved
[ -is-ready {true|false} ]              Is Ready to Be Used
[ -is-notfound {true|false} ]           Is Not Found
[ -is-pending-notfound {true|false} ]   Is Pending Not Found
[ -is-wildcard {true|false} ]           Is Wildcard
[ -is-pending-wildcard {true|false} ]   Is Pending Wildcard
[ -is-abandoned {true|false} ]         Is Abandoned
[-member-count <integer> ]              Count of Members
[-hosts-count <integer> ]               Count of Hosts
[ -pending-addresses-count <integer> ]   Count of Addresses Pending
[ -pending-hosts-dropped <integer> ]    Count of Hosts Not Found in Pending
[ -retries-on-queue <integer> ]         Count of Times Retried in the Queue
[ -expanded-duration <[[<hours>:]:<minutes>:]:<seconds>> ] How Long it Took to Expand Netgroup
[ -pending-hosts-resolved <integer> ]   Count of Hosts Already Resolved
```

To check netgroup membership:

```
cluster::> man vserver export-policy netgroup check-membership ?

-vserver <vserver name>                Vserver
[-netgroup] <text>                     Name of the Netgroup
[-client-ip] <IP Address>              Client Address
```

To view a queue of unresolved netgroups:

```
cluster::> vserver export-policy netgroup queue show ?

[ -instance | -fields <fieldname>, ... ]
[ -vserver <vserver name> ]            Vserver
[ -netgroup <text> ]                    Name of the Netgroup
[ -queue-state {active|register|retry} ] State of Entry in the Queue
[-record-id <integer> ]                 Record ID
[ -is-getting-hosts {true|false} ]      Hosts Being Retrieved
[ -is-ready {true|false} ]              Is Ready to Be Used
[ -is-notfound {true|false} ]           Is Not Found
[ -is-pending-notfound {true|false} ]   Is Pending Not Found
[ -is-wildcard {true|false} ]           Is Wildcard
[ -is-pending-wildcard {true|false} ]   Is Pending Wildcard
[ -is-abandoned {true|false} ]         Is Abandoned
[-member-count <integer> ]              Count of Members
[-hosts-count <integer> ]               Count of Hosts
[ -pending-addresses-count <integer> ]   Count of Addresses Pending
[ -pending-hosts-dropped <integer> ]    Count of Hosts Not Found in Pending
[ -retries-on-queue <integer> ]         Count of Times Retried in the Queue
[ -age <[[<hours>:]:<minutes>:]:<seconds>> ] Age of Entry in the Queue
```

NIS Netgroup Strict (`nfs.netgroup.strict`)

In 7-Mode, the option `nfs.netgroup.strict` allowed the ability to control whether or not a netgroup entry required a `@` sign so that Data ONTAP recognized the netgroup as a netgroup.

Best Practices 41: Netgroup Definition in Export Policy Rules (see next: Best Practices 42)

In Data ONTAP, there currently is no equivalent to the `nfs.netgroup.strict` option. All netgroups in export policy rules must be designated with the `@` sign to be recognized as netgroups. If no `@` sign is present, Data ONTAP treats the entry as a host name and attempts to resolve the name in DNS or local hosts.

Netgroup.byhost Support

`Netgroup.byhost` entries can vastly speed up netgroup entry lookup by allowing the cluster to avoid the need to query every entry in a netgroup for access and instead fetch the netgroup by way of LDAP lookup per host. In large environments with netgroups that have many entries, this can drastically speed up the time for lookups and avoid access issues due to timeouts on LDAP queries. Support for `netgroup.byhost` was added to Data ONTAP 8.3.

Best Practices 42: Netgroup.byhost Considerations (see next: Best Practices 43)

When using `netgroup.byhost`, consider the following to enable the desired access results for hosts.

- Forward and reverse DNS records for host names.
- Host triple entry in netgroup file.
- Netgroup specification for the host's `netgroup.byhost` entry.
- Need to always use lowercase hosts to avoid case sensitivity issues.
- Syncing of DNS and `netgroup.byhost` entries, including case sensitivity.
- If using `netgroup.byhost` with NIS, be sure that the triples are configured to avoid using “-“ in the entries. For example, the entries should look like this: (host,,) and not (host,-,-). ONTAP supports only the host portion of the triple. NIS treats any entry in the other portions of the triple as an attempted entry.

NetApp highly recommends `netgroup.byhost` functionality for large environments with very large netgroups.

The `netgroup.byhost` and `netgroup` entries *must* be in sync to enable access to work properly.

Enabling `netgroup.byhost` Support in Data ONTAP

`netgroup.byhost` support is not enabled by default in Data ONTAP. There are several options in the LDAP client configuration that need to be modified:

```
-is-netgroup-byhost-enabled  
-netgroup-byhost-dn  
-netgroup-byhost-scope
```

Naturally, `-is-netgroup-byhost-enabled` needs to be enabled to allow the use of `netgroup.byhost` functionality.

DN and scope specify the filters desired for `netgroup.byhost` functionality. For more information, see the administration guides for your release of Data ONTAP. Keep in mind that support for this feature applies only to Data ONTAP 8.3 and later.

For configuration steps to create `netgroup.byhost` objects in Active Directory, see the configuration section of this document on [creating netgroup.byhost entries in Active Directory LDAP](#).

DNS Considerations

SSSD can leverage Kerberos authentication for secure LDAP binds. Therefore, 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 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 depends on the order of entries in `/etc/resolv.conf`. If the first DNS server in the file is inaccessible, SSSD black holes the attempt until the DNS server is available or until the `/etc/resolv.conf` file is modified. For more information on this point, see [Red Hat bug 966757](#).

Best Practices 43: DNS Considerations for Use with SSSD (see next: Best Practices 44)

If the domain has multiple domain controllers, leave the `ldap_uri` and `krb5_server` options out of the configuration file. Doing so enables use of the built-in SRV records for Kerberos and LDAP, which allows failover capability if a domain controller goes down. If you use a single domain controller, leave the options out for scalability if additional domain controllers are added at a later date.

5.2 LDAP Using Red Hat Directory Services for Identity Management

The following section covers how to set up a Red Hat Directory server as an Identity Management server for use with Data ONTAP.

Note: This process covers RHEL 6.x. For RHEL 7.x instructions, see Red Hat Product Documentation.

Prerequisites

- Valid FQDN with DNS entries for A record, PTR and SRV record for LDAP
- Firewall allowing LDAP ports (389 for normal LDAP, 636 for LDAP over SSL, 9830 for admin server)
- Correct repositories for package installation ([EPEL](#) and [REMI](#))

For information on allowing ports using the server's firewall (iptables and/or SELinux), see the vendor's OS documentation.

Example of DNS entries for the Directory Services LDAP server:

```
# nslookup ldap
Server:      10.228.225.120
Address:    10.228.225.120#53
Name:      ldap.linux.netapp.com
```

```
Address: 10.228.225.141
# nslookup
> set type=srv
> _ldap._tcp.linux.netapp.com
Server:      10.228.225.120
Address:     10.228.225.120#53

_ldap._tcp.linux.netapp.com    service = 0 100 389 ldap.linux.netapp.com.
```

Performance and Security Tuning in Red Hat Directory Services

As per the following [UnixMen LDAP configuration steps](#), some performance and security tuning options are recommended. Evaluate each value and determine if it fits your environment.

If the performance and security tuning options are not addressed, the following warning might be seen when you set up the server:

```
NOTICE : The net.ipv4.tcp_keepalive_time is set to 7200000 milliseconds
(120 minutes).  This may cause temporary server congestion from lost
client connections.

WARNING: There are only 1024 file descriptors (soft limit) available, which
limit the number of simultaneous connections.
```

Installing Red Hat Directory Services

Use `yum install` to install either the `fedora-ds` or the `389-ds` package on the server.

Example:

```
# yum install 389-ds-base* -y
```

Configuring Red Hat Directory Services

To configure the Red Hat Directory Services LDAP server, run the Perl script provided with the package. This script is added to the `/usr/sbin` directory when the package is installed.

```
# which setup-ds.pl
/usr/sbin/setup-ds.pl
```

The script can be run from anywhere on the server:

```
[root@ldap ~]# pwd
/root

[root@ldap ~]# setup-ds.pl

=====
This program will set up the 389 Directory Server.

It is recommended that you have "root" privilege to set up the software.
Tips for using this program:
- Press "Enter" to choose the default and go to the next screen
- Type "Control-B" or the word "back" then "Enter" to go back to the previous screen
- Type "Control-C" to cancel the setup program

Would you like to continue with set up? [yes]:
```


The rest of the setup looks like this if advanced setup is chosen:

Choose a setup type:

1. Express
Allows you to quickly set up the servers using the most common options and pre-defined defaults. Useful for quick evaluation of the products.
2. Typical
Allows you to specify common defaults and options.
3. Custom
Allows you to specify more advanced options. This is recommended for experienced server administrators only.

To accept the default shown in brackets, press the Enter key.

Choose a setup type [2]: 3

=====
Enter the fully qualified domain name of the computer on which you're setting up server software. Using the form <hostname>.<domainname>
Example: eros.example.com.

To accept the default shown in brackets, press the Enter key.

Warning: This step may take a few minutes if your DNS servers can not be reached or if DNS is not configured correctly. If you would rather not wait, hit Ctrl-C and run this program again with the following command line option to specify the hostname:

General.FullMachineName=your.host name.domain.name

Computer name [ldap.linux.netapp.com]:

=====
The server must run as a specific user in a specific group. It is strongly recommended that this user should have no privileges on the computer (i.e. a non-root user). The setup procedure will give this user/group some permissions in specific paths/files to perform server-specific operations.

If you have not yet created a user and group for the server, create this user and group using your native operating system utilities.

System User [nobody]:
System Group [nobody]:

=====
The standard directory server network port number is 389. However, if you are not logged as the superuser, or port 389 is in use, the default value will be a random unused port number greater than 1024. If you want to use port 389, make sure that you are logged in as the superuser, that port 389 is not in use.

Directory server network port [389]:

=====
Each instance of a directory server requires a unique identifier. This identifier is used to name the various instance specific files and directories in the file system, as well as for other uses as a server instance identifier.

Directory server identifier [ldap]:

=====
The suffix is the root of your directory tree. The suffix must be a valid DN. It is recommended that you use the dc=domaincomponent suffix convention.

For example, if your domain is example.com, you should use dc=example,dc=com for your suffix. Setup will create this initial suffix for you, but you may have more than one suffix. Use the directory server utilities to create additional suffixes.

Suffix [dc=linux, dc=netapp, dc=com]:

```
=====
Certain directory server operations require an administrative user.
This user is referred to as the Directory Manager and typically has a
bind Distinguished Name (DN) of cn=Directory Manager.
You will also be prompted for the password for this user. The password must
be at least 8 characters long, and contain no spaces.
Press Control-B or type the word "back", then Enter to back up and start over.
```

```
Directory Manager DN [cn=Directory Manager]:
Password:
Password (confirm):
```

```
=====
You may install some sample entries in this directory instance. These
entries will be installed in a separate suffix and will not interfere
with the normal operation of the directory server.
```

Do you want to install the sample entries? [no]: yes

```
=====
You may wish to populate your new directory instance with some data.
"You may already have a file in LDIF format to use or some suggested
entries can be added. If you want to import entries from an LDIF
file, you may type in the full path and filename at the prompt. If
you want the setup program to add the suggested entries, type the
word suggest at the prompt. The suggested entries are common
container entries under your specified suffix, such as ou=People and
ou=Groups, which are commonly used to hold the entries for the persons
and groups in your organization. If you do not want to add any of
these entries, type the word none at the prompt.
```

```
Type the full path and filename, the word suggest, or the word none [suggest]:
Your new DS instance 'ldap' was successfully created.
```

After configuring the server, make sure that the service starts every time on boot:

```
[root@ldap ~]# chkconfig dirsrv on
[root@ldap ~]# chkconfig | grep dirsrv
dirsrv          0:off  1:off  2:on   3:on   4:on   5:on   6:off
dirsrv-snmp    0:off  1:off  2:off  3:off  4:off  5:off  6:off
```

Testing the LDAP Server

The following command tests the LDAP server. Entries are already populated as a result of choosing sample entries in the configuration.

```
[root@ldap ~]# ldapsearch -x -b "dc=linux,dc=netapp,dc=com"
# extended LDIF
#
# LDAPv3
# base <dc=linux,dc=netapp,dc=com> with scope subtree
# filter: (objectclass=*)
# requesting: ALL
#
# linux.netapp.com
dn: dc=linux,dc=netapp,dc=com
objectClass: top
objectClass: domain
dc: linux

# Directory Administrators, linux.netapp.com
```

```
dn: cn=Directory Administrators,dc=linux,dc=netapp,dc=com
objectClass: top
objectClass: groupofuniqueNames
cn: Directory Administrators
uniqueMember: cn=Directory Manager

# Groups, linux.netapp.com
dn: ou=Groups,dc=linux,dc=netapp,dc=com
objectClass: top
objectClass: organizationalunit
ou: Groups

# People, linux.netapp.com
dn: ou=People,dc=linux,dc=netapp,dc=com
objectClass: top
objectClass: organizationalunit
ou: People

# Special Users, linux.netapp.com
dn: ou=Special Users,dc=linux,dc=netapp,dc=com
objectClass: top
objectClass: organizationalUnit
ou: Special Users
description: Special Administrative Accounts

# Accounting Managers, Groups, linux.netapp.com
dn: cn=Accounting Managers,ou=Groups,dc=linux,dc=netapp,dc=com
objectClass: top
objectClass: groupOfUniqueNames
cn: Accounting Managers
ou: groups
description: People who can manage accounting entries
uniqueMember: cn=Directory Manager

# HR Managers, Groups, linux.netapp.com
dn: cn=HR Managers,ou=Groups,dc=linux,dc=netapp,dc=com
objectClass: top
objectClass: groupOfUniqueNames
cn: HR Managers
ou: groups
description: People who can manage HR entries
uniqueMember: cn=Directory Manager

# QA Managers, Groups, linux.netapp.com
dn: cn=QA Managers,ou=Groups,dc=linux,dc=netapp,dc=com
objectClass: top
objectClass: groupOfUniqueNames
cn: QA Managers
ou: groups
description: People who can manage QA entries
uniqueMember: cn=Directory Manager

# PD Managers, Groups, linux.netapp.com
dn: cn=PD Managers,ou=Groups,dc=linux,dc=netapp,dc=com
objectClass: top
objectClass: groupOfUniqueNames
cn: PD Managers
ou: groups
description: People who can manage engineer entries
uniqueMember: cn=Directory Manager

# search result
search: 2
result: 0 Success

# numResponses: 1
```

Managing Users and Groups

There are two primary ways to manage users and groups in Red Hat Directory Services: the GUI and the CLI. Both of these methods are covered in the documentation.

The GUI is simple enough, so it is not covered in this document. However, it is important to note that the GUI is available only if the [FAQ on the Fedora Project wiki](#) is followed.

The following shows examples of creating users and groups using CLI.

Ldapadd is the command to use to add entries to Directory Services.

```
usage: ldapadd [options]
       The list of desired operations are read from stdin or from the file
       specified by "-f file".
Add or modify options:
  -a          add values (default)
  -c          continuous operation mode (do not stop on errors)
  -E [!]ext=extparam  modify extensions (! indicate s criticality)
  -f file     read operations from `file'
  -M          enable Manage DSA IT control (-MM to make critical)
  -P version  protocol version (default: 3)
  -S file     write skipped modifications to `file'
Common options:
  -d level   set LDAP debugging level to `level'
  -D binddn  bind DN
  -e [!]<ext>[=<extparam>] general extensions (! indicates criticality)
                        [!]assert=<filter>      (RFC 4528; a RFC 4515 Filter string)
                        [!]authzid=<authzid>    (RFC 4370; "dn:<dn>" or "u:<user>")
                        [!]chaining[=<resolveBehavior>[/<continuationBehavior>]]
                        one of "chainingPreferred", "chainingRequired",
                        "referralsPreferred", "referralsRequired"
                        [!]manageDSAIT        (RFC 3296)
                        [!]noop
                        ppolicy
                        [!]postread[=<attrs>]  (RFC 4527; comma-separated attr list)
                        [!]pread[=<attrs>]    (RFC 4527; comma-separated attr list)
                        [!]relax
                        abandon, cancel, ignore (SIGINT sends abandon/cancel,
                        or ignores response; if critical, doesn't wait for SIGINT.
                        not really controls)
  -h host    LDAP server
  -H URI     LDAP Uniform Resource Identifier(s)
  -I         use SASL Interactive mode
  -n         show what would be done but don't actually do it
  -N         do not use reverse DNS to canonicalize SASL host name
  -O props   SASL security properties
  -o <opt>[=<optparam>] general options
                        nettimeout=<timeout> (in seconds, or "none" or "max")
  -p port    port on LDAP server
  -Q         use SASL Quiet mode
  -R realm   SASL realm
  -U authcid SASL authentication identity
  -v         run in verbose mode (diagnostics to standard output)
  -V         print version info (-VV only)
  -w passwd  bind password (for simple authentication)
  -W         prompt for bind password
  -x         Simple authentication
  -X authzid SASL authorization identity ("dn:<dn>" or "u:<user>")
  -y file    Read password from file
  -Y mech    SASL mechanism
  -Z         Start TLS request (-ZZ to require successful response)
```

The bind user is the Directory Manager user defined in the DS setup.

```
Directory Manager DN [cn=Directory Manager]:
Password:
Password (confirm):
```

Adding users and groups is best done from a file with the `ldif` extension, which is covered in [RFC 2849](#). With this file, the `ldapadd` command can be used to create users and groups by reading from the file.

Example of `ldif` file (UID and GID attributes are intentionally left out):

```
## ADD a single entry to people level

dn: cn=RHEL user,ou=people,dc=linux,dc=netapp,dc=com
objectclass: inetOrgPerson
cn: RHEL User
sn: rheluser
uid: rheluser
userpassword: rheluser
mail: rheluser@linux.netapp.com

# create a new group

dn: cn=IT,ou=groups,dc=linux,dc=netapp,dc=com
objectclass: groupofnames
cn: IT
description: IT security group
member: cn=RHEL User,ou=people,dc=linux,dc=netapp,dc=com
```

To add the entries, simply run the command and point to the `ldif` file. Use the CN=Directory Manager login and password specified in setup:

```
[root@ldap ~]# ldapadd -x -D "CN=Directory Manager" -w P@ssw0rd -f /users_and_groups.ldif
adding new entry "cn=RHEL user,ou=people,dc=linux,dc=netapp,dc=com"

adding new entry "cn=IT,ou=groups,dc=linux,dc=netapp,dc=com"
```

Then check for the entries:

```
[root@ldap ~]# ldapsearch -x -b "dc=linux,dc=netapp,dc=com" | grep IT
dn: cn=IT,ou=Groups,dc=linux,dc=netapp,dc=com
cn: IT

[root@ldap ~]# ldapsearch -x -b "dc=linux,dc=netapp,dc=com" | grep rheluser
sn: rheluser
uid: rheluser
```

Modifying Existing Entries (Add, Replace, Delete)

In addition to adding entries using `ldif` files, existing entries can also be modified by including the line `changetype: modify` and a `replace: attribute_to_replace`, `add: attribute_to_add` or `delete: attribute_to_delete` line in the `ldif` file.

Example:

```
[root@ldap ~]# cat /modify.ldif
## Add a uidNumber, gidNumber, objectClass, homeDirectory, loginShell, userPassword; replace the
old uid with a new one; delete the old email address and add a new one

dn: cn=RHEL user,ou=people,dc=linux,dc=netapp,dc=com
changetype: modify
add: objectClass
objectClass: posixAccount
-
add: uidNumber
uidNumber: 10000
-
add: homeDirectory
homeDirectory: /home/newrheluser
-
add: gidNumber
gidNumber: 10001
-
add: loginShell
loginShell: /bin/sh
-
add: userPassword
userPassword: !@#md[opie
-
replace: uid
uid: newrheluser
-
delete: mail
-
add: mail
mail: newrheluser@linux.netapp.com

## Add gidNumber, objectClass to group

dn: cn=IT,ou=groups,dc=linux,dc=netapp,dc=com
changetype: modify
add: objectClass
objectClass: posixGroup
-
add: gidNumber
gidNumber: 10001
```

Then run the `ldapadd` command with the modify entries:

```
[root@ldap ~]# ldapadd -x -D "CN=Directory Manager" -w P@ssw0rd -f /modify.ldif
modifying entry "cn=RHEL user,ou=people,dc=linux,dc=netapp,dc=com"
modifying entry "cn=IT,ou=groups,dc=linux,dc=netapp,dc=com"
```

Confirm using `ldapsearch`:

```
# RHEL user, People, linux.netapp.com
dn: cn=RHEL user,ou=People,dc=linux,dc=netapp,dc=com
objectClass: inetOrgPerson
objectClass: organizationalPerson
objectClass: person
objectClass: top
objectClass: posixAccount
cn: RHEL User
sn: rheluser
uid: newrheluser
mail: newrheluser@linux.netapp.com
uidNumber: 10000
homeDirectory: /home/newrheluser
gidNumber: 10001
loginShell: /bin/sh

# IT, Groups, linux.netapp.com
dn: cn=IT,ou=Groups,dc=linux,dc=netapp,dc=com
objectClass: groupofnames
objectClass: top
objectClass: posixGroup
cn: IT
description: IT security group
member: cn=RHEL User,ou=people,dc=linux,dc=netapp,dc=com
gidNumber: 10001
```

Confirm using `getent passwd` or `id` from an LDAP client:

```
[root@ldapclient ~]# getent passwd newrheluser
newrheluser:*:10000:10001:RHEL User:/home/newrheluser:

[root@ldapclient ~]# id newrheluser
uid=10000(newrheluser) gid=10001(IT) groups=10001(IT)
```

Using Red Hat Directory Services with Data ONTAP

Red Hat DS uses the RFC-2307 schema by default. Therefore, when using Red Hat DS with Data ONTAP, the schema should be set to RFC-2307 unless a special configuration has been used. The rest of the steps remain the same as the steps in the section [“Configuring the Data ONTAP System to Use LDAP.”](#)

5.3 Setting Up LDAP Clients

The following section covers how to set up NFS clients for use with various LDAP servers. SSSD is used for LDAP on the Linux clients (except for Solaris). The LDAP requests leveraging Kerberos for security with Active Directory Identity Management, so the Kerberos setup must be completed before attempting to set up LDAP. For Red Hat Directory Services, simple binds are used. By the end of the section, users should be able to get identity information from the LDAP server.

The following Linux host configurations are covered, but this configuration might 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 (RHEL/CentOS 7.x will be added at a later date)
- 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.

Configuring the Client to Use LDAP

The following client configuration leverages 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 Pluggable Authentication Module (PAM) configurations.

Note: This section assumes that Kerberos has been configured and a ticket can be issued to the NFS client using the `kinit` command. If this has not happened yet, SSSD configuration fails 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 using the `/etc/sss/sss.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.

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 might be beneficial to upgrade:

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

Configuring SSSD on RHEL/CentOS/Fedora

After the application is installed, the `/etc/sss/sss.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 `sss.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 `sss.conf` file, see the [SSSD documentation](#) or the [/etc/sss/sss.conf man pages](#).

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

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

After the `/etc/sss/sss.conf` is configured, modify `/etc/nsswitch.conf` to use SSSD for name services. The “sss” entry is 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 (the preferred method):

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

After `/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 if there are errors on service restart, check the following:

- `/etc/sss/sss.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/host name` 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 five 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 preceding steps are verified, the following log files can be useful:

```
/var/log/messages
/var/log/sss/* (be sure to enable debugging in /etc/sss/sss.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@nfsclient'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 they are 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 Practices 44: Client PAM Configuration Recommendation (see next: Best Practices 45)

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

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 might be beneficial to upgrade:

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

Configuring SSSD on SUSE/SLES

After the application is installed, the `/etc/sss/sss.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 `sss.conf` file is configured with specific parameters to leverage Kerberos. See Table 7 in section 4.2.10 for descriptions of important options in the file. For more detail on the `sss.conf` file, see the [SSSD documentation](#) or the [/etc/sss/sss.conf man pages](#).

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

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

After `/etc/sss/sss.conf` is configured, modify `/etc/nsswitch.conf` to use SSSD for name services. The “sss” entry is 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
```

After `/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` might 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.

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

```
sles11:/etc/sss # 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/sss # 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 errors on service restart, check the following:

- `/etc/sss/sss.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 five 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 preceding steps are verified, the following log files can be useful:

```
/var/log/messages
/var/log/sss/* (be sure to enable debugging in /etc/sss/sss.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.

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 might be beneficial to upgrade:

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

Configuring SSSD on Ubuntu

After the application is installed, the `/etc/sss/sss.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 `sss.conf` file is configured with specific parameters to leverage Kerberos. See Table 7 in section 4.2.10 for descriptions of important options in the file. For more detail on the `sss.conf` file, see the [SSSD documentation](#) or the [/etc/sss/sss.conf man pages](#).

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

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

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

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 errors on service restart, check the following:

- `/etc/sss/sss.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 five 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 preceding steps are verified, the following log files can be useful:

```
/var/log/messages
/var/log/sss/* (be sure to enable debugging in /etc/sss/sss.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 `sss` service is running.
- Verify that the client firewall is not blocking SSH.

sssd.conf File Example

The following is a sample file from a working configuration with LDAP running on Windows Active Directory.

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

[sssd]
config_file_version = 2
services = nss, pam
domains = YOURDOMAINNAME
#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 preceding:

```
krb5_canonicalize = False
```

For information on this point, see the following:

[What is krb5_canonicalize?](#)

Best Practices 45: LDAP Client Base DN Recommendations (see next: Best Practices 46)

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.

The following is a sample from a working configuration with LDAP running on Red Hat Directory Services.

```
# cat /etc/sss/sss.conf
[domain/default]
cache_credentials = True
case_sensitive = False
[sss]
config_file_version = 2
services = nss, pam
domains = LINUX
debug_level = 7
[nss]
filter_users = root,ldap,named,avahi,haldaemon,dbus,radiusd,news,nscd
filter_groups = root
[pam]
[domain/LINUX]
id_provider = ldap
ldap_uri = ldap://ldap.linux.netapp.com
case_sensitive = false
cache_credentials = false
ldap_search_base = dc=linux,dc=netapp,dc=com
ldap_schema = rfc2307
ldap_user_object_class = posixAccount
ldap_group_object_class = posixGroup
ldap_user_home_directory = homeDirectory
ldap_force_upper_case_realm = true
ldap_user_search_base = ou=People,dc=linux,dc=netapp,dc=com
ldap_group_search_base = ou=Groups,dc=linux,dc=netapp,dc=com
```

SSSD Configuration File Options

Table 7) /etc/sss/sss.conf file options.

Option	Use Case
cache_credentials	Caches the LDAP credentials on the client for improved lookup performance.
case_sensitive	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; uses 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, Data ONTAP 8.2.x and earlier do not support RFC-2307bis. Support for RFC-2307bis schemas has been added to Data ONTAP 8.3 and later.
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 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 attempts to use host/hostname@REALM as the SPN, and the request fails 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 plain text.
krb5_canonicalize	Use with RHEL 6.3 .

The [domain/YOURDOMAINNAME] Section

The [domain/YOURDOMAINNAME] section in the sssd.conf file tells SSSD the domain parameters to use. The domains option tells SSSD which domain is used. Multiple domains can be specified. The YOURDOMAINNAME portion of the entry can be any value, provided that the 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 preceding domains in order, set default_domain as such:

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

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
```

After these steps are 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 using 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 \  
-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 choice 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 leverages Kerberos tickets for LDAP queries. The ticket is obtained when the LDAP client is configured and leverages 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` and the removal of the `proxyDN` and `proxyPassword` for binding. All binding in sasl/GSSAPI is done using Kerberos ticket authentication, so no passwords are required or stored.

How LDAP Configuration in Solaris Works

All LDAP configuration and logging for Solaris are 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 because the attributes `authenticationMethod` and `credentialLevel` depend on one another. When a failure occurs, the previous configuration is 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 shows the following error:

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

After a valid configuration is applied, the `ldap_client_file` is 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_ATTRIBUTE_MAP= group:userpassword=userPassword
NS_LDAP_ATTRIBUTE_MAP= group:memberuid=memberUid
NS_LDAP_ATTRIBUTE_MAP= group:gidnumber=gidNumber
NS_LDAP_ATTRIBUTE_MAP= passwd:gecos=cn
NS_LDAP_ATTRIBUTE_MAP= passwd:gidnumber=gidNumber
NS_LDAP_ATTRIBUTE_MAP= passwd:uidnumber=uidNumber
NS_LDAP_ATTRIBUTE_MAP= passwd:homedirectory=unixHomeDirectory
NS_LDAP_ATTRIBUTE_MAP= passwd:loginshell=loginShell
NS_LDAP_ATTRIBUTE_MAP= shadow:shadowflag=shadowFlag
NS_LDAP_ATTRIBUTE_MAP= shadow:userpassword=userPassword
NS_LDAP_OBJECTCLASS_MAP= group:posixGroup=group
NS_LDAP_OBJECTCLASS_MAP= passwd:posixAccount=user
NS_LDAP_OBJECTCLASS_MAP= shadow:shadowAccount=user
```

Nsswitch Files

In addition to the preceding, 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 is replaced with the `nsswitch.ldap` file. By default, the `nsswitch.ldap` file uses LDAP and files *only* for all entries. This fact is problematic for environments using Windows as an LDAP server, because most Windows LDAP servers do not use a DN for hosts or ipnodes. Therefore, LDAP queries for these fail, and LDAP lookups might not work properly. To correct these issues, 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

After configuration is done, the configuration can be checked using 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](#).

Autofs and Homedirs

NFS clients can leverage automounter and LDAP to point users to their home directories and mount the home directory using NFS whenever a user authenticates using LDAP.

In Windows Active Directory LDAP (Windows 2003R2 and later), the following UNIX attributes are added when a schema is extended:

```
unixUserPassword
uid
uidNumber
gidNumber
unixHomeDirectory
```

The attribute value in `unixHomeDirectory` can be used in conjunction with `automounter` to allow users to mount their home directories when logging in on NFS clients.

Example:

```
# id ldapuser
uid=1107(ldapuser) gid=513(domain users) groups=513(domain users),10011(ldifde-
group),2001(group1),10005(testgroup)
# getent passwd ldapuser
ldapuser:*:1107:513:ldapuser:/home/CDOT/ldapuser:/bin/sh
# mount | grep home
#
# su ldapuser
sh-4.1$ mount | grep home
10.63.57.237:/vol/home/ldapuser on /home/CDOT/ldapuser type nfs4
(rw,nosuid,minorversion=1,hard,tcp,timeo=60,sloppy,addr=10.63.57.237,clientaddr=10.228.225.140)
```

In the preceding, the user “ldapuser” mounts its home directory when it logs in and accesses the `~ share`, which is a symlink to the user’s `homedir`. These actions are all possible because of the following configuration:

- Exported `qtree` or volume on a Data ONTAP system with existing user folder
- Correctly configured `auto.master/auto.homedir` files
- Correctly configured NFS client with [connectivity to LDAP](#)
- Correctly configured user in LDAP with proper home directory attribute populated

The preceding example shows that users logging in and authenticating with LDAP query the `unixHomeDirectory` attribute in LDAP. For the user “ldapuser,” that attribute is populated as `/home/CDOT/ldapuser`.

Note: Keep in mind that the default entry for home directories in Windows LDAP is `/home/username`. That field might need to be modified depending on the junction path created in the cluster.

When the client authenticates using LDAP, it fetches the attributes associated with the user and implements them as a normal `passwd` entry.

Auto.master

This file is consulted when the `autofs` script runs on the NFS client and sets up the necessary mount points. This file can be configured to specify export path, map file, automounter options, and so on. The map file is a reference to another file in the format of `auto.[filename]`. `Auto.master` has default entries for `/misc` and `/net`. A new entry should be added for home directories. Additionally, the `+auto.master` entry should be commented out.

Example:

```
[client] # cat /etc/auto.master
#
# Sample auto.master file
# This is an automounter map and it has the following format
# key [ -mount-options-separated-by-comma ] location
# For details of the format look at autofs(5).
#
/misc    /etc/auto.misc
#
# NOTE: mounts done from a hosts map will be mounted with the
#       "nosuid" and "nodev" options unless the "suid" and "dev"
#       options are explicitly given.
#
/net     -hosts
#
# Include central master map if it can be found using
# nsswitch sources.
#
# Note that if there are entries for /net or /misc (as
# above) in the included master map any keys that are the
# same will not be seen as the first read key seen takes
# precedence.
#
##### SAMPLE HOME DIR ENTRY #####
/home/CDOT auto.homedir --timeout=50
#+auto.master
```

In the preceding, the mount point path is `/home/CDOT`, which is a folder created on the client. The mounts time out after 50 seconds (configurable), causing the client to need to remount.

Best Practices 46: Automount Recommendations (see next: Best Practices 47)

- Make sure that the specified mount point exists on the client.
- For security purposes, NetApp recommends automount timeouts.

Auto.homedir

The export path is referenced in the file named `auto.homedir`. In that file, the following entries are created:

```
[client] # cat /etc/auto.homedir
* -fstype=nfs4,minorversion=1,rw,nosuid,hard,tcp,timeo=60 10.63.57.237:/vol/home/&
```

The preceding output is explained as follows:

- The first entry is an asterisk. That entry tells the client to use the user name logged in when looking for the home directory name.
- The second entry specifies the mount options.
- The third entry is the mount path in the format of `[vserver data LIF or hostname]:/junction-path/&`.

The ampersand (&) signifies that the client should take the value grabbed by `*` and use it in its mount path. So if a user named "ldapuser" attempts an automount, then the path should be `/vol/home/ldapuser`.

Exported File System Paths

The export path specified in the `auto.homedir` file *must* exist. For every user that requires a home directory located on an exported file system on the cluster, a folder or qtree needs to be created in a volume on the cluster.

Note: Subdirectory exports (for example, exporting /volume/qtree/directory) are not currently supported.

In this example, qtrees are used, because they can be exported at a granular level starting in 8.2.1 for NFSv3 operations and for 8.3 for NFSv4.x operations.

```
cluster::> qtree status -vserver SVM -volume home
Vserver   Volume   Tree      Style     Oplocks   Status
-----
SVM       home     ""        unix      enable    normal
SVM       home     ldapuser  unix      enable    normal
SVM       home     test      unix      enable    normal
3 entries were displayed.
```

The qtrees live in a volume called “home,” which is mounted to /vol/home in the cluster namespace:

```
cluster::> vol show -vserver SVM -volume home -fields junction-path
(volume show)
vserver volume junction-path
-----
SVM     home   /vol/home
```

Thus, all home directories have a path of /mountpoint/username. Because the client is always mounting to the mount point of /home/CDOT (as per auto.master), the LDAP entries should be modified to /home/CDOT/username.

When a user has the correct home directory path specified, the users mount the correct exports:

```
# getent passwd test
test:*:10001:513:test:/home/CDOT/test:/bin/sh
# su test
sh-4.1$ mount | grep test
10.63.57.237:/vol/home/test on /home/CDOT/test type nfs4
(rw,nosuid,minorversion=1,hard,tcp,timeo=60,sloppy,addr=10.63.57.237,clientaddr=10.228.225.140)
```

When a user *does not* have the correct home directory path, the mount fails:

```
# getent passwd bob
bob:*:100067:513:bob:/home/bob:/bin/sh
# su bob
sh-4.1$ mount | grep bob
sh-4.1$
sh-4.1$ cd ~
sh: cd: /home/bob: No such file or directory
```

The preceding fails because the path /home/bob does not match the expected path of /home/CDOT/bob specified in auto.homedir.

After the LDAP entry is adjusted, the user is still unable to mount because the qtree/folder “bob” does not exist in the exported file system:

```
# getent passwd bob
bob:*:100067:513:bob:/home/CDOT/bob:/bin/sh
# su bob
sh-4.1$ cd ~
sh: cd: /home/CDOT/bob: No such file or directory
```

After adding a qtree/folder named “bob” to the home volume, the user sees a successful mount:

```
cluster::> qtree create -vserver SVM -volume home -qtree bob -security-style unix
cluster::> qtree status -vserver SVM -volume home
Vserver      Volume      Tree          Style         Oplocks      Status
-----
SVM          home        ""            unix          enable       normal
SVM          home        bob           unix          enable       normal
SVM          home        ldapuser     unix          enable       normal
SVM          home        test         unix          enable       normal
4 entries were displayed.

# su bob
sh-4.1$ mount | grep bob
10.63.57.237:/vol/home/bob on /home/CDOT/bob type nfs4
(rw,nosuid,minorversion=1,hard,tcp,timeo=60,sloppy,addr=10.63.57.237,clientaddr=10.228.225.140)
```

5.4 How SecD Queries LDAP in Data ONTAP

When a user or group lookup is required, the SVM looks the user up depending on how the `ns-switch` and `nm-switch` options are configured. The order of lookup depends on the order in which the name service and name-mapping switches are specified, with the first in the list used first in lookups. If the user/group/netgroup does not exist in the first name service database, then the query attempts to look up the user/group/netgroup in the second name service database. To perform these lookups, the authentication process called SecD (security daemon) sends requests to the configured name service servers to retrieve this information. These include, but are not limited to:

- User names and numeric user IDs (UIDs)
- Group names and numeric group IDs (GIDs)
- Group membership
- Netgroups and netgroup members
- Host name to IP and IP to host name lookups (DNS only)
- Active Directory–specific operations (SID, machine/user account attributes, domain add/remove, and so on)

5.5 Using SecD to Troubleshoot External Name Service Queries

Data ONTAP relies on the [SecD](#) process to serve authentication and lookup requests in NAS environments. On occasion, it might be necessary to try to isolate faults or failures. This section is intended to assist with that process, as well as provide considerations for SecD troubleshooting.

Data LIFs

In Data ONTAP 8.2.x and earlier, data LIFs were used for name services, but node management LIFs were also eligible for use by SecD. However, this capability came with the limitation of SecD being a node-specific process, which defeated the notion of a file system. Starting in Data ONTAP 8.3.x, SecD no longer has this limitation, because it can use any routable data LIF in an SVM. Node management LIFs are used for SecD requests.

Fault Isolation

Before Data ONTAP 8.3.x, SecD requests traveled only through a routable data LIF on the local node. Therefore, before 8.3.x, troubleshooting required that a storage admin be aware of the node experiencing issues to effectively troubleshoot SecD.

For example, if a cluster has four nodes and each node has a data LIF that can be used for NAS access, it is necessary to review logs, CLI commands, and packet traces to determine which node might be having issues.

In Data ONTAP 8.3.x, SecD requests can now leverage any data LIF in the SVM that can be routed to name services. To isolate faults in 8.3.x and later, `diag secd` commands can be used to determine which data LIF is in use, and, thus, which node's SecD process is affected.

It is important to determine which node is having issues, because that node's SecD process needs to be the one issued commands.

Caches

Each node's SecD process has caches that maintain information about users, groups, credentials, netgroups, connections, and so on to speed up authentication requests. These caches can sometimes interfere with the ability to accurately troubleshoot issues with SecD. For more information on SecD caches, see [TR-4067](#).

In some cases, it might be necessary to flush the SecD caches to properly troubleshoot so that stale information does not skew results. However, it is important to consider the impact of flushing the cache when issuing commands. Flushing caches on production systems can result in delays or failures as the caches are repopulated.

Commands

Troubleshooting authentication and other name service functionality in Data ONTAP is done primarily through the `diag secd` command set. The appendix of this report contains the section "[Commonly Used Commands and Logs for Troubleshooting NAS in Data ONTAP](#)" that includes a series of tables showing the translation from 7-Mode to Data ONTAP for troubleshooting commands and log files. The appendix also includes a description of what commonly used `diag secd` commands do. These tables cover only Data ONTAP 8.3.x and later.

Note: `diag secd` commands are available at *diag privilege* only. Exercise caution when using any diagnostic command, especially commands that flush caches or make changes. If you are unsure about running a command, contact NetApp Technical Support.

Note: These tables include only commands and logs commonly used and most useful to NAS/name service troubleshooting.

Statistics Available for SecD

In Data ONTAP, statistics are available to measure SecD performance. These statistics show successes, failures, and total time that a SecD operation took to process. These statistics are available at the advanced level in Data ONTAP 8.2.x and later.

```
NAME
    statistics secd show -- Display SecD Statistics

AVAILABILITY
    This command is available to cluster administrators at the advanced privilege level.

DESCRIPTION
    Attention:    This command is deprecated and will be removed in a future major release.

    The statistics secd show command displays information about SecD RPC usage statistics on the
    nodes in a cluster. You can view the following information:

    o Number of times an RPC was called
    o Number of successful RPC calls
    o Number of failed RPC calls
    o Maximum time taken to process an RPC
    o Minimum time taken to process an RPC
    o Total collective time spent on an RPC

    Use this command only as directed by support personnel to help analyze performance and
    diagnose problems.

PARAMETERS
    { [-fields <fieldname>, ...]
      If you specify the -fields <fieldname>, ... parameter, the command output also
      includes the specified field or fields. You can use '-fields ?' to display the fields to specify.

    | [-instance] }
      If you specify the -instance parameter, the command displays detailed information
      about all fields.

    [-node {<nodename>|local}] - Node
      Selects the nodes that match this parameter value.

    [-vserver <vserver>] - Vserver
      Selects the nodes that match this parameter value.

    [-secdstat-type <secdStatType>] - SecdStatType
      Selects the nodes that match this parameter value (SecD RPC type).

    [-count <Counter>] - Count
      Selects the nodes that match this parameter value (number of times an RPC was called).

    [-succeeded <Counter>] - Success
      Selects the nodes that match this parameter value (number of times an RPC succeeded).

    [-failed <Counter>] - Failure
      Selects the nodes that match this parameter value (number of times an RPC failed).

    [-total-time <Counter>] - TotalTime
      Selects the nodes that match this parameter value (total time for an RPC).

    [-max-time <Counter>] - MaxTime
      Selects the nodes that match this parameter value (maximum time for an RPC).

    [-min-time <Counter>] - MinTime
      Selects the nodes that match this parameter value (minimum time for an RPC).
```

The following is sample output from this command:

```

cluster::*> statistics secd show -node node-01 -vserver SVM
Node:                node-01
Vserver:             SVM
SecdStatType         Count    Success  Failure  TotalTime  MaxTime  MinTime
-----
auth_extended        0        0        0        0          0        0
auth_passthrough     0        0        0        0          0        0
ontap_admin_cifs_auth_extended 0 0        0        0          0        0
ontap_admin_cifs_auth_basic 0 0        0        0          0        0
auth_msrpc           0        0        0        0          0        0
auth_msrpc_decrypt   0        0        0        0          0        0
auth_msrpc_encrypt   0        0        0        0          0        0
auth_msrpc_samr      0        0        0        0          0        0
msrpc_witness_get_interface_list 0 0        0        0          0        0
msrpc_witness_register 0        0        0        0          0        0
msrpc_witness_unregister 0        0        0        0          0        0
msrpc_witness_async_notify 0 0        0        0          0        0
auth_get_creds       3        3        0        118741     80917    4493
auth_user_name_to_ontap_admin_unix_creds 1459 1459 0 15848904 504824 3359
auth_user_id_to_ontap_admin_unix_creds 0 0 0 0 0 0
auth_user_name_to_unix_creds 0 0 0 0 0 0
auth_user_id_to_unix_creds 459 459 0 215041 4886 184
auth_user_name_to_unix_ids 0 0 0 0 0 0
auth_user_id_to_unix_owner_names 0 0 0 0 0 0
auth_user_name_to_id 0 0 0 0 0 0
auth_user_id_to_name 11 9 2 14388 3558 177
group_name_to_id     0        0        0        0          0        0
group_id_to_name     22       20       2       91116     19639    170
auth_sid_to_name     0        0        0        0          0        0
auth_sid_to_unix_name 0        0        0        0          0        0
auth_name_to_sid     0        0        0        0          0        0
auth_sid_to_uid      0        0        0        0          0        0
auth_sid_to_uid_with_uuid 0 0        0        0          0        0
auth_uid_to_sid      0        0        0        0          0        0
auth_uid_to_sid_with_uuid 0 0        0        0          0        0
create_cifs_server   0        0        0        0          0        0
ds_change_password   0        0        0        0          0        0
ds_reset_password    0        0        0        0          0        0
ds_get_dc_info       0        0        0        0          0        0
ds_ad_account_delete 0        0        0        0          0        0
dce_rpc_passthrough 0        0        0        0          0        0
nmap_map_name        0        0        0        0          0        0
discover_servers     0        0        0        0          0        0
discover_service     0        0        0        0          0        0
server_information   24       24       0       4049     242     131
get_cifs_setup_server 0        0        0        0          0        0
nfs_krb_bind_spn    10       6        4       72842953 8136040 6032254
nfs_krb_change_key   0        0        0        0          0        0
nfs_krb_set_key      0        0        0        0          0        0
nfs_krb_get_key      0        0        0        0          0        0
netgroup_get_addrs  0        0        0        0          0        0
flush_netgroup_cache 0        0        0        0          0        0
accept_gss_token     107     57       50     399359   16300   518
handoff_gss_token    0        0        0        0          0        0
nmap_map_group       0        0        0        0          0        0
netgroup_get_host    0        0        0        0          0        0
get_hostname_from_ip 0        0        0        0          0        0
get_netgroup_cache_locks 0 0        0        0          0        0
gpo_get_list         0        0        0        0          0        0
54 entries were displayed.

```

SecD Log Analysis

SecD logs can give a wealth of information about specific SecD operations. The following section shows a SecD error and breaks it down in chunks to show the kind of information you can get.

The following error was from a query for the GID 0 in LDAP. The group did not exist, so the lookup failed.

```
00000048.003abf5e 0171fd0a Mon Jun 15 2015 12:36:00 -04:00 [kern_sec:info:4254] | [000.008.508]
ERR : RESULT_ERROR_SECD_GROUP_NOT_FOUND:6910 in getLdapGroupInfo() at
secd/authorization/secd_ldap_unix_authorization.cpp:1802
```

The following shows the most useful information in the errors.

Time Stamps

Time stamps show on which day and time/time zone the error occurred. Because SecD calls generally occur in milliseconds, the time stamp is most useful for narrowing down when an issue occurred.

```
Mon Jun 15 2015 12:36:00 -04:00
```

Time Spent in Process

This portion of the error is especially useful in seeing how long each part of the SecD error took to complete. The time is measured in *seconds.milliseconds.nanoseconds*.

In the preceding example, the error occurred around 8.5ms into the RPC:

```
[000.008.508]
```

Error Type

This portion of the log gives an idea of what the actual error was in the log and which call caused the error. In the preceding example, the error was `RESULT_ERROR_SECD_GROUP_NOT_FOUND` in the function `getLdapGroupInfo`, which is pretty straightforward: The group was not found.

```
RESULT_ERROR_SECD_GROUP_NOT_FOUND:6910 in getLdapGroupInfo()
```

Note: The remaining portions of the log messages are generally useful only for NetApp support, because they relate to internal code specifics.

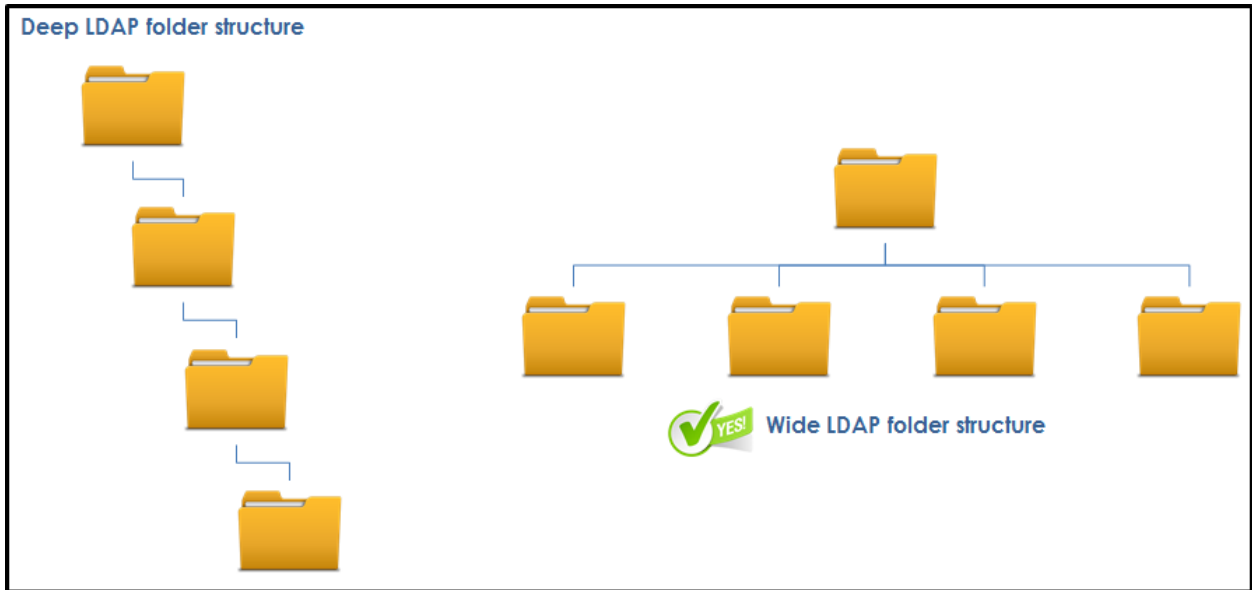
5.6 Optimizing LDAP Searches: Best Practices

When using Data ONTAP as an LDAP client for enterprise NAS, it is imperative to make sure that the LDAP searches perform as quickly as possible to eliminate delays in access. Although there is a copious amount of caching in Data ONTAP for NAS, there is still a cost associated with initial lookups. The following best practices should be followed to enable the best LDAP performance possible. For a complete list of name service best practices, see [TR-4379: Name Services Best Practice Guide](#).

- Make sure that LDAP servers and associated name service servers (such as DNS) are on low-latency network links.
- Ideally, LDAP servers and DNS servers are local to the Data ONTAP cluster.
- Make sure that LDAP servers are not overworked (high CPU and so on). Overworked LDAP servers return answers to queries more slowly. LDAP servers often have specific tools to measure performance, such as [ADTest](#). For more information on performance testing for LDAP, contact the LDAP server vendor.
- Use LDAP query tools such as `ldapsearch` or `ldp.exe` to troubleshoot search issues.
- Include multiple LDAP servers in any client configuration to allow load balancing and redundancy.
- Maintain your LDAP schemas to remove old records that are no longer in use.

- Build LDAP schema structures and distinguished names (DNs) with a wide design rather than a deep design. Wide schemas allow shorter DN.

Figure 15) LDAP schema structure examples.



5.7 Configuring the Data ONTAP System to Use LDAP

The following section describes how to configure a Data ONTAP system to use LDAP for its name mapping and UID/GID lookups. This section assumes that a working LDAP server exists and can be reached by the SVM data LIFs. If this is not the case, see the section [“Configuring the Domain Controller as an LDAP Server”](#) for details.

For condensed setup steps, see the [“Quick Step Setup Guides”](#) section in this document.

LDAP Schemas

In Data ONTAP, built-in read-only schemas are available to admins. These schemas can be used to configure LDAP. The schemas can also 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 before configuring an LDAP client configuration. This is because specifying the schema during LDAP client creation is required. The schemas can be changed any time after creation of the LDAP client, however.

Table 8) Default schemas available in Data ONTAP.

ONTAP 9.0 and Later	Data ONTAP 8.2.x and 8.3.x	Data ONTAP 8.0.x and 8.1
AD-SFU AD-IDMU RFC-2307 MS-AD-BIS	AD-SFU AD-IDMU RFC-2307	AD-SFU RFC-2307

Note: Examples of each schema can be found in the appendix under [“LDAP Schema Examples in Data ONTAP 8.2 and Later.”](#)

Schemas are used in LDAP queries by the cluster to find information about a user, such as the UID. The schema attributes must exist in the LDAP server for the cluster to find the entry. Otherwise, LDAP queries might return no data and authentication requests might 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` is used. The default schema for AD-IDMU uses the attribute `uidNumber` for that query. If no user in LDAP with a `uidNumber` attribute = 0 exists, then the lookup will fail.

Best Practices 47: LDAP Client Schema Recommendation (see next: Best Practices 48)

Most LDAP servers can leverage the default read-only schemas provided by Data ONTAP. It is best to use those schemas unless there is a direct requirement to do otherwise. Scenarios in which custom schemas are needed depend on the LDAP schema attributes in place. Consult your LDAP administrators to negotiate the proper LDAP schemas.

LDAP Schema Considerations

If the SVM LDAP client schema is configured correctly, the query returns 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 and later instead of AD-IDMU), queries fail because incorrect attributes are passed to the LDAP server. For instance, the objectClass is `posixAccount` in RFC-2307 schemas rather than `user` in AD-IDMU.

```
Schema Template: RFC-2307
RFC 2307 posixAccount Object Class: posixAccount

Schema Template: AD-IDMU
RFC 2307 posixAccount Object Class: User
```

LDAP queries in Data ONTAP are passed through the SecD application. In Data ONTAP 8.3 and later, SecD acts as a mediator and forwards LDAP queries to the new name services architecture based on LibC through RPC. For more information, see the section in this document on [getXXbyYY](#). These queries use `ldapsearch` to find information and can be seen in the SecD log as failed attempts, which can be useful for troubleshooting LDAP issues.

Best Practices 48: User/Computer Objects + Primary Groups (see next: Best Practices 49)

If leveraging LDAP to populate UNIX attributes for users and/or computer objects, make sure that the objects have a primary group ID set. Otherwise, credential fetching might not work properly.

In the following example, SecD is asked to look for a user that does not have a 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))
```

The specific attributes are also specified:

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

The base is also specified:

```
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 9) 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 preceding example shows that there are indeed no entries for the user “nouser” in LDAP.

It is also possible to leverage PowerShell in Windows Active Directory LDAP servers to perform similar searches for users and other objects in the directory. These commands can be used to filter searches, specify search DN's, and so on, and the commands are infinitely useful for troubleshooting LDAP queries.

Example:

```
PS C:\> Get-AdUser -Filter {UId -eq "ldapuser2"} -Properties uidNumber, gidNumber

DistinguishedName : CN=ldapuser,CN=Users,DC=domain,DC=com
Enabled            : True
gidNumber         : 1000
GivenName         : ldapuser
Name              : ldapuser
ObjectClass       : user
ObjectGUID        : 8d039512-d3ae-4662-a3bb-8d00296d6f47
SamAccountName    : ldapuser
SID               : S-1-5-21-56907238-3627968364-3018309926-1173
Surname           :
uidNumber         : 1234
UserPrincipalName : ldapuser@domain.com
```

Pulling Additional Information from LDAP Using SecD

With `diag secd` commands, it is possible to get more information about users than just the UID.

For instance, if CIFS is set up, `diag secd authentication show-creds` can be used to show the credentials, name mapping, and group membership for a user:

```
cluster::> set diag
cluster::*> diag secd authentication show-creds -node nodel-vserver flexvol -unix-user-name
ldapuser -list-name true -list-id true

UNIX UID: 1107 (ldapuser) <> Windows User: S-1-5-21-3413584004-3312044262-250399859-1107
(DOMAIN\ldapuser (Domain User))

GID: 513 (Domain Users)
Supplementary GIDs:
 10005 (testgroup)
 2001 (group1)
 10011 (ldifde-group)

Windows Membership:
S-1-5-21-3413584004-3312044262-250399859-1203    DOMAIN\domainlocal (Alias)
S-1-5-21-3413584004-3312044262-250399859-1204    DOMAIN\universalgroup (Domain group)
S-1-5-21-3413584004-3312044262-250399859-1169    DOMAIN\group13 (Domain group)
S-1-5-21-3413584004-3312044262-250399859-1166    DOMAIN\group10 (Domain group)
S-1-5-21-3413584004-3312044262-250399859-1162    DOMAIN\group6 (Domain group)
```

S-1-5-21-3413584004-3312044262-250399859-1172	DOMAIN\group16 (Domain group)
S-1-5-21-3413584004-3312044262-250399859-1173	DOMAIN\group17 (Domain group)
S-1-5-21-3413584004-3312044262-250399859-1161	DOMAIN\group5 (Domain group)
S-1-5-21-3413584004-3312044262-250399859-1171	DOMAIN\group15 (Domain group)
S-1-5-21-3413584004-3312044262-250399859-1163	DOMAIN\group7 (Domain group)
S-1-5-21-3413584004-3312044262-250399859-1170	DOMAIN\group14 (Domain group)
S-1-5-21-3413584004-3312044262-250399859-1118	DOMAIN\testgroup (Domain group)
S-1-5-21-3413584004-3312044262-250399859-1159	DOMAIN\group3 (Domain group)
S-1-5-21-3413584004-3312044262-250399859-1165	DOMAIN\group9 (Domain group)
S-1-5-21-3413584004-3312044262-250399859-1167	DOMAIN\group11 (Domain group)
S-1-5-21-3413584004-3312044262-250399859-1158	DOMAIN\group2 (Domain group)
S-1-5-21-3413584004-3312044262-250399859-1164	DOMAIN\group8 (Domain group)
S-1-5-21-3413584004-3312044262-250399859-1168	DOMAIN\group12 (Domain group)
S-1-5-21-3413584004-3312044262-250399859-1160	DOMAIN\group4 (Domain group)
S-1-5-21-3413584004-3312044262-250399859-1157	DOMAIN\group1 (Domain group)
S-1-5-21-3413584004-3312044262-250399859-513	DOMAIN\Domain Users (Domain group)
S-1-5-32-545	BUILTIN\Users (Alias)

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

Privileges (0x80):

If the SVM is not the member of a domain, this command fails, even if the SVM is pointed to the proper LDAP server:

```
cluster::*> cifs delete -vserver flexvol
```

In order to delete an Active Directory machine account for the CIFS server, you must supply the name and password of a Windows account with sufficient privileges to remove computers from the "DOMAIN.WIN2K8.NETAPP.COM" domain.

Enter the user name: administrator

Enter the password:

Warning: There are one or more shares associated with this CIFS server
Do you really want to delete this CIFS server and all its shares?
{y|n}: y

```
cluster::*> diag secd authentication show-creds -node node1 -vserver flexvol
-unix-user-name ldapuser -list-name true -list-id true
```

Vserver: flexvol (internal ID: 7)

```
Error: Get user credentials procedure failed
[ 6] Determined UNIX id 1107 is UNIX user 'ldapuser'
[ 7] Using a cached connection to 10.228.225.120
[ 9] 'CifsServerSecurity' configuration not available
**[ 9] FAILURE: 'CifsServer' configuration not available
```

Error: command failed: Failed to get user credentials. Reason: "SecD Error: configuration not found".

Using Vserver Security Commands to Look Up Permissions

Diag secd commands can be used in conjunction with vserver security file-directory commands to compare and contrast a user's group membership with the permissions on a specific file or folder.

These commands are available at the *admin privilege* level.

```
cluster::> man vserver security file-directory show
```

NAME

vserver security file-directory show -- Display file/folder security information

AVAILABILITY

This command is available to cluster and Vserver administrators at the admin privilege level.

DESCRIPTION

The vserver file-directory show command displays file/folder security information. The command output depends on the parameter or parameters specified with the command.

The -vserver and -path parameters are required for this command. If you do not specify any of the optional parameters, the command displays all security information in list format for the specified path.

You can specify the -fields parameter to specify which fields of information to display about files and folders security.

You can specify the -instance parameter to display all the security information in list format.

When using the vserver security file-directory show command, storage administrators can get a treasure trove of information on objects in the storage system, including:

- Security style
- Effective security style
- DOS attributes
- UNIX owner/group
- UNIX permissions
- NTFS and NFSv4 ACLs

This information provides what you need to troubleshoot basic permissions/access issues without ever having to contact the file owners or clients.

Example:

```
cluster::> vserver security file-directory show -vserver SVM -path /ntfs
```

```
      Vserver: SVM
      File Path: /ntfs
      File Inode Number: 64
      Security Style: ntfs
      Effective Style: ntfs
      DOS Attributes: 10
      DOS Attributes in Text: ----D---
      Expanded Dos Attributes: -
      UNIX User Id: 0
      UNIX Group Id: 0
      UNIX Mode Bits: 777
      UNIX Mode Bits in Text: rwxrwxrwx
      ACLs: NTFS Security Descriptor
            Control:0x9504
            Owner:BUILTIN\Administrators
            Group:BUILTIN\Administrators
            DACL - ACEs
              ALLOW-Everyone-0x1f01ff-OI|CI
              ALLOW-DOMAIN\ldapuser-0x1f01ff-OI|CI
```

Looking Up User Attributes with SecD for Non-Windows LDAP Servers

For non-Windows LDAP servers, such as Red Hat Directory Services, there is also a way to get additional information about a user with the `diag secd authentication show-ontap-admin-unix-creds` command. This command can be used only with non-Windows LDAP servers, because the `unixUserPassword` attribute does not pass to the cluster properly with Windows LDAP servers.

This command requests the following attributes:

- UID
- GID
- Home directory
- Login shell
- UNIX user password (requested, but not displayed)

If any of these attributes is missing from the user, the command fails:

```
cluster::~*> diag secd authentication show-ontap-admin-unix-creds -node node1 -vserver flexvol -
unix-user-name newrheluser

Vserver: flexvol (internal ID: 3)

Error: Acquire ontap admin UNIX credentials procedure failed
 [ 0 ms] No servers available for LDAP_NIS_AND_NAME_MAPPING,
        vserver: 3, domain: .
 [  0] Connecting to LDAP (NIS & Name Mapping) server
        10.228.225.141
 [  3] Using a new connection to 10.228.225.141
 [  5] Failed to get ontap admin credentials for name
        'newrheluser' using UNIX authorization source LDAP,
        Error: 1057
 [  5] Could not get ontap admin credentials for name
        'newrheluser' using any NS-SWITCH authorization source
**[  5] FAILURE: Unable to retrieve ontap admin credentials for
**   UNIX user with name newrheluser
```

Example of working query:

```
cluster::~*> diag secd authentication show-ontap-admin-unix-creds -node node1-vserver flexvol -
unix-user-name newrheluser
        User Id: 10000
        Group Id: 10001
Home Directory: /home/newrheluser
        Login Shell: /bin/sh
```

Using Microsoft Active Directory LDAP to Its Full Potential with RFC-2307bis

Microsoft Active Directory actually leverages LDAP schemas as its back end. All of the attributes in the directory are LDAP attributes. The schema is based on RFC-2307, with some modifications to fit the Windows model (often called the AD-IDMU schema). Because of this, LDAP for use with UNIX attributes in Windows Active Directory can be implemented relatively seamlessly.

However, one difference in the AD-IDMU schema that can trip up LDAP administrators is the way that secondary UNIX group memberships are handled. By default, Microsoft LDAP servers do not use the RFC-2307 standard way of populating secondary UNIX groups. Despite this, there is actually a more efficient way of handling secondary group memberships and providing support for nested group memberships that might not be available in non-Windows LDAP deployments: RFC-2307bis.

What Is RFC-2307bis?

[RFC-2307](#) is the request-for-comments memo entitled “An Approach for Using LDAP as a Network Information Service.” [RFC-2307bis](#) is an extension of RFC-2307 and adds support for `posixGroups`, which enables dynamic lookups for auxiliary/secondary groups using the `uniqueMember` attribute rather than the `memberUid` attribute in the LDAP schema. Instead of using just the name of the user for lookups (standard searches filter using the group attribute `memberUid=user` to crawl the LDAP schema for all groups in which a user is a member), this attribute contains the full distinguished name (DN) of another object in the LDAP database (such as another user or group). These values exist on the user object under the `Member` attribute. Once the values are queried, their UNIX attributes are returned in

subsequent queries. Therefore, groups can have other groups as members, which allows nesting of groups. Support for RFC-2307bis also adds support for the object class `groupOfUniqueNames`.

Best Practices 49: RFC2307bis and Active Directory LDAP (see next: Best Practices 50)

If using Windows Active Directory LDAP with Data ONTAP 8.3 and later, consider using RFC-2307bis support because of the natural fit with Active Directory default schema attributes for group memberships. With RFC-2307bis, no additional configuration steps are needed to add users to groups other than simply belonging to a Windows group.

To use RFC-2307bis functionality with Windows Active Directory, a [custom schema](#) should be created. The default attributes for the “RFC 2307bis groupOfUniqueNames Object Class” and “RFC 2307bis uniqueMember Attribute” on the [built-in schema templates](#) in Data ONTAP are not the common values in Windows Active Directory. The values might vary depending on the LDAP schema, but the following values are generally acceptable for most Windows Active Directory environments that wish to implement RFC-2307bis.

Note: If the user or group DN has a trailing slash (for example, User=test/), then BIS lookups might fail.

Best Practices 50: RFC2307bis and Active Directory Schema (see next: Best Practices 51)

ONTAP 9.0 introduces a new built-in schema template for RFC-2307bis environments, specifically with Active Directory in mind. This schema is called [MS-AD-BIS](#) and should be used with Microsoft Active Directory LDAP servers whenever possible.

Table 10) Sample RFC-2307bis schema for LDAP servers in Active Directory:

```
Vserver: -
Schema Template: 2307bis
Comment:
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
Enable Support for Draft RFC 2307bis: true
RFC 2307bis groupOfUniqueNames Object Class: Group
RFC 2307bis uniqueMember Attribute: Member
Data ONTAP Name Mapping windowsToUnix Object Class: User
Data ONTAP Name Mapping windowsAccount Attribute: sAMAccountName
Data ONTAP Name Mapping windowsToUnix Attribute: sAMAccountName
No Domain Prefix for windowsToUnix Name Mapping: true
Vserver Owns Schema: false
```

RFC-2307bis Support

Versions of Data ONTAP before 8.3 did not support RFC-2307bis schemas for LDAP. However, Data ONTAP has added support for this schema type. RFC-2307bis can be enabled on the LDAP client schema by copying a read-only schema (such as RFC-2307 or AD-IDMU) to a new schema, then modifying that schema to use RFC-2307bis, which is disabled by default.

Example:

```
cluster::> ldap client schema copy -schema RFC-2307 -new-schema-name BIS -vserver vs0
```

In addition, the following schema attributes have been added:

```
-enable-rfc2307bis  
-group-of-unique-names-object-class  
-unique-member-attribute
```

Best Practices 51: LDAP Group Attribute Best Practice (see next: Best Practices 52)

When using LDAP for UNIX attributes on users and groups, it's important to make sure that all groups that a UNIX user is a member of (whether it's Windows or UNIX) have a gidNumber specified. Failure to specify a gidNumber on a group can result in undesired or unexpected behavior in the enumeration of secondary GIDs for UNIX users. See bug 994736 for details.

To enable RFC-2307bis, modify the schema at the *advanced privilege* level.

Example:

```
cluster::> ldap client schema modify -schema BIS -enable-rfc2307bis true -vserver vs0  
Error: "modify" is not a recognized command  
  
cluster::*> set advanced  
cluster::*> ldap client schema modify -schema BIS -enable-rfc2307bis true -vserver vs0  
cluster::*> ldap client schema show -schema BIS  
  
Vserver: vs0  
Schema Template: BIS  
Comment:  
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  
Enable Support for Draft RFC 2307bis: true  
RFC 2307bis groupOfUniqueNames Object Class: Group  
RFC 2307bis uniqueMember Attribute: Member  
Data ONTAP Name Mapping windowsToUnix Object Class: User  
Data ONTAP Name Mapping windowsAccount Attribute: sAMAccountName  
Data ONTAP Name Mapping windowsToUnix Attribute: sAMAccountName  
No Domain Prefix for windowsToUnix Name Mapping: true  
Vserver Owns Schema: false
```

After the schema is copied and modified, it can then be [applied to the desired LDAP client configuration](#) for RFC-2307bis support.

Because of [bug 935722](#), using RFC2307bis when DN's contain commas can cause failures. For instance, if a user's DN is "DN=Smith,Bob," the cluster cannot look up that user. A fix for this issue will be available in a future release.

Multiple LDAP Server and Distinguished Name (DN) Support

Data ONTAP versions 8.2.2 and 8.3 and later offer support for semicolon-separated multiple [Distinguished Names](#) for user, group, and netgroup lookups as well as for base DN's. The following options under `ldap client` commands control this behavior:

```
-base-dn
-user-dn
-group-dn
-netgroup-dn
```

With this feature, searches can be customized and filtered to specific DN objects in the LDAP database. Doing so allows multiple CNs or OUs in the same LDAP server or across multiple LDAP servers.

To specify multiple DN's in 7-Mode, entries were designated with parentheses and separated by semicolons. However, in Data ONTAP, these entries should not use parentheses. If they do, LDAP lookups fail.

Table 11) Example of multiple DN configuration:

```
cluster::*> ldap client show -client-config WinLDAP -vserver WIN2K12 -instance
Vserver: WIN2K12
Client Configuration Name: WinLDAP
LDAP Server List: 10.228.225.120, 10.228.225.122
Active Directory Domain:
Preferred Active Directory Servers: -
Bind Using the Vserver's CIFS Credentials: false
Schema Template: AD-IDMU
LDAP Server Port: 389
Query Timeout (sec): 3
Minimum Bind Authentication Level: simple
Bind DN (User): ldapuser
Base DN:
dc=domain,dc=win2k8,dc=netapp,dc=com;dc=americas,dc=win2k12,dc=netapp,dc=com
Base Search Scope: subtree
User DN:
cn=users,dc=domain,dc=win2k8,dc=netapp,dc=com;cn=users,dc=americas,dc=win2k12,dc=netapp,dc=com
User Search Scope: subtree
Group DN:
cn=users,dc=domain,dc=win2k8,dc=netapp,dc=com;cn=users,dc=americas,dc=win2k12,dc=netapp,dc=com
Group Search Scope: subtree
Netgroup DN: -
Netgroup Search Scope: subtree
Vserver Owns Configuration: true
Use start-tls Over LDAP Connections: false
```

When this is done, the SVM will try the LDAP servers in the list for LDAP lookups in a round-robin fashion. If the user does not exist in the first DN listed, the next DN will be tried. If the first server does not contain the user or group in any of the specified DN's, then the next LDAP server is tried, then the next DN, and so on. Keep in mind that each failed search adds time to the overall query timeout value. If the LDAP search does not complete within the allotted timeout (3 seconds by default), then the request will fail. With multiple DN's, missed lookups can add up and potentially cause access issues. The LDAP timeout is controlled through the client option `-query-timeout`.

Note: Data ONTAP currently does not support multidomain LDAP referrals, but does support referrals to servers in the same domain. This support includes LDAP URI referral. [Global catalog searches are supported between domains in the same forest](#), however. For information on Microsoft LDAP referrals, see the [TechNet article on LDAP referrals](#).

The example below shows user lookups for two different LDAP servers in two different Windows AD domains/forests looking up users in entirely different DN's.

User from domain named AMERICAS:


```
cluster::*> diag secd authentication translate -node node2 -vserver WIN2K12 -unix-user-name
americas
10000
```

User from domain named DOMAIN:

```
cluster::*> diag secd authentication translate -node node2 -vserver WIN2K12 -unix-user-name
domain
100068
```

Best Practices 52: UID/GID Configuration with Multiple Domains (see next: Best Practices 53)

If using multiple LDAP DNSs, make sure that there are no duplicate user names or groups. This check is necessary because the cluster cannot discern the difference between them and returns the UID/GID based on “last server accessed” logic. All user and group names in the LDAP environment should be unique.

For example, the user “ldapuser” exists in both DOMAIN and AMERICAS. Thus, the UID returned would depend on the credentials found in the last server in cache.

In the following example, the last LDAP server accessed was 10.228.225.120:

```
cluster::*> diag secd authentication translate -node node1 -vserver WIN2K12 -unix-user-name
ldapuser
1107
cluster::*> diag secd connections show -node node1 -vserver WIN2K12 -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: 0, Misses: 2, Failures: 0, Avg Retrieval: 127.50ms

+ Rank: 01 - Server: 10.228.225.120 (10.228.225.120) <<< note the server used
    Connected through the 10.63.22.163 interface, 0.6 mins ago
    Used 1 time(s), and has been available for 33 secs
    RTT in ms: mean=2.50, min=0, max=5, med=5, dev=2.50 (6.0 mins of data)
```

In this example, the cluster is connected to the other LDAP server:

```
cluster::*> diag secd authentication translate -node node1 -vserver WIN2K12 -unix-user-name
ldapuser
10001
cluster::*> diag secd connections show -node node1 -vserver WIN2K12 -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: 1, Misses: 2, Failures: 0, Avg Retrieval: 74.00ms

+ Rank: 01 - Server: 10.228.225.122 (10.228.225.122) <<< note the server used
    Connected through the 10.63.22.163 interface, 0.7 mins ago
    Used 2 time(s), and has been available for 39 secs
    RTT in ms: mean=52.50, min=3, max=102, med=102, dev=49.50 (0.1 mins of data)
```

Because the user exists in both locations, the credentials for that user would depend only on what the last LDAP server used was.

Connections in secd are retained for five minutes and then flushed if inactive. Active connections remain in cache until they are inactive and age out, are manually flushed, or fail to connect. This is not configurable, but connections can be manually cleared:

```
cluster::*> diag secd connections clear ?
[-node] <nodename>      *Node
[-vserver] <vserver>    *Vserver
[[-type] <text>]        *Cache type (lsa,netlogon,ldap-ad,ldap-nis-namemap,nis)
[ -key <text> ]         *Connection key
```

Note: Flushing caches and connections requires time to repopulate the cache, so there might be some latency in authentication when performing these tasks.

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 NEWSHEMA -vserver vs0
```

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

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

Best Practices 53: When to Create Custom Schemas (see next: Best Practices 54)

In most cases, the default schema templates available in Data ONTAP are sufficient for LDAP configurations. However, there are occasions when a custom schema is needed:

- RFC-2307bis solutions
- Third-party LDAP solutions (such as Vintela, Centrify, QAS, and so on)
- Customized LDAP schema attributes are used
- LDAP is being used for name-mapping rules

As with any implementation, it's important to contact the owners of the LDAP environment to understand the schemas being used to make sure that configurations are correct.

Mapping 7-Mode LDAP Attributes to Data ONTAP

One of the benefits of using LDAP in Data ONTAP over 7-Mode is the inclusion of stock LDAP schema templates. In 7-Mode, there was a set of default attributes in options based on RFC-2307, but those did not cover use cases for LDAP built on Active Directory.

As a result, storage administrators were left with the daunting task of having to modify numerous options manually. However, one downside of the new templates is that existing 7-Mode customers have to map those options to those in Data ONTAP LDAP clients to make sure that everything works properly.

In the following output, the default LDAP schema attributes found in Data ONTAP LDAP templates are highlighted and in bold.

Default 7-Mode LDAP Options

```
ldap.ADdomain
ldap.base
ldap.base.group
ldap.base.netgroup
ldap.base.passwd
ldap.enable                off
ldap.fast_timeout.enable   on
ldap.minimum_bind_level    anonymous
ldap.name
```

```

ldap.nssmap.attribute.gecos gecos
ldap.nssmap.attribute.gidNumber gidNumber
ldap.nssmap.attribute.groupname cn
ldap.nssmap.attribute.homeDirectory homeDirectory
ldap.nssmap.attribute.loginShell loginShell
ldap.nssmap.attribute.memberNisNetgroup memberNisNetgroup
ldap.nssmap.attribute.memberUid memberUid
ldap.nssmap.attribute.netgroupname cn
ldap.nssmap.attribute.nisNetgroupTriple nisNetgroupTriple
ldap.nssmap.attribute.uid uid
ldap.nssmap.attribute.uidNumber uidNumber
ldap.nssmap.attribute.userPassword userPassword
ldap.nssmap.objectClass.nisNetgroup nisNetgroup
ldap.nssmap.objectClass.posixAccount posixAccount
ldap.nssmap.objectClass.posixGroup posixGroup
ldap.passwd *****
ldap.port 389
ldap.retry_delay 120
ldap.servers
ldap.servers.preferred
ldap.ssl.enable off
ldap.timeout 20
ldap.usermap.attribute.unixaccount unixaccount
ldap.usermap.attribute.windowsaccount windowsaccount
ldap.usermap.base
ldap.usermap.enable off

```

Table 15 details how the options found in 7-Mode map to LDAP client schema options in Data ONTAP. Mapping the options is done to make the transition from LDAP in 7-Mode to Data ONTAP easier.

Table 12) LDAP attributes in Data ONTAP operating in 7-Mode mapped to Data ONTAP.

7-Mode Attribute	Data ONTAP Attribute
ldap.nssmap.attribute.gecos	RFC 2307 geCOS Attribute -gecos-attribute
ldap.nssmap.attribute.gidNumber	RFC 2307 gidNumber Attribute -gid-number-attribute
ldap.nssmap.attribute.groupname	RFC 2307 cn (for Groups) Attribute -cn-group-attribute
ldap.nssmap.attribute.homeDirectory	RFC 2307 homeDirectory Attribute -home-directory-attribute
ldap.nssmap.attribute.loginShell	RFC 2307 loginShell Attribute -login-shell-attribute
ldap.nssmap.attribute.memberNisNetgroup	RFC 2307 memberNisNetgroup Attribute -member-nis-netgroup-attribute
ldap.nssmap.attribute.memberUid	RFC 2307 memberUid Attribute -member-uid-attribute
ldap.nssmap.attribute.netgroupname	RFC 2307 cn (for Netgroups) Attribute -cn-netgroup-attribute
ldap.nssmap.attribute.nisNetgroupTriple	RFC 2307 nisNetgroupTriple Attribute -nis-netgroup-triple-attribute
ldap.nssmap.attribute.uid	RFC 2307 uid Attribute -uid-attribute
ldap.nssmap.attribute.uidNumber	RFC 2307 uidNumber Attribute -uid-number-attribute
ldap.nssmap.attribute.userPassword	RFC 2307 userPassword Attribute -user-password-attribute

ldap.nssmap.objectClass.nisNetgroup	RFC 2307 nisNetgroup Object Class -nis-netgroup-object-class
ldap.nssmap.objectClass.posixAccount	RFC 2307 posixAccount Object Class -posix-account-object-class
ldap.nssmap.objectClass.posixGroup	RFC 2307 posixGroup Object Class -posix-group-object-class
ldap.usermap.attribute.unixaccount	N/A
ldap.usermap.attribute.windowsaccount	ONTAP Name Mapping windowsAccount Attribute -windows-account-attribute
N/A	RFC 2307 nisObject Object Class -nis-object-class
N/A	RFC 2307 nisMapName Attribute -nis-mapname-attribute
N/A	RFC 2307 nisMapEntry Attribute -nis-mapentry-attribute

Hidden 7-Mode LDAP Options

In Data ONTAP operating in 7-Mode, a number of hidden options also can be leveraged for LDAP configurations.

Table 13) Hidden options for LDAP in Data ONTAP operating in 7-Mode.

Hidden 7-Mode LDAP Option	Use Case
ldap.nssmap.attribute.uniqueMember	For RFC-2307bis use.
ldap.nssmap.objectClass.groupOfUniqueNames	For RFC-2307bis use.
ldap.rfc2307bis.enable	For RFC-2307bis use.
ldap.retry_delay	For specifying a wait time for the system to retry on LDAP server failures.
ldap.security.level	This specifies the level of security on the LDAP bind. 0 = SASL, 1 = Signing, 2 = Sealing
ldap.skip_cn_unescape.enable	Enables/disables the use of “unescape” for CNs (for attributes using special characters, such as “/”).
ldap.usermap.symmetriclookup	Used to specify if LDAP can be used for asymmetric name mappings (that is, not 1:1 mappings).
ldap.usermap.windows-to-unix.attribute	Specifies the attribute to be used for Windows to UNIX name mapping in LDAP.
ldap.usermap.windows-to-unix.objectClass	Specifies the objectClass of the Windows-to-UNIX name-mapping attribute.

Some of the preceding options, such as RFC-2307bis options (8.3 and later), are implemented in current Data ONTAP versions. Some options (such as LDAP signing and sealing) are the result of features that are currently not supported in Data ONTAP. For name mapping in Data ONTAP using LDAP and the differences from 7-Mode, see the section [“Setting Name Mapping Rules in LDAP.”](#)

Note: Before transitioning to Data ONTAP from Data ONTAP operating in 7-Mode, check the LDAP options (using the command `options ldap`) to make sure that these hidden options are not in use. A hidden option appears in the command output only if it has been modified from the default value.

LDAP Clients

In 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 fails.

When LDAP clients are created or modified in *admin privilege*, 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
[-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|sas} Minimum Bind Authentication Level
[ -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 or modified in *advanced privilege*, 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
[-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|sas} Minimum Bind Authentication Level
[ -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 Practices 54: LDAP Client Configuration with CIFS Servers (see next: Best Practices 55)

When a CIFS server is present in an SVM, it is a best practice to bind the LDAP client as a CIFS server. Doing so leverages CIFS/SMB authentication methods for LDAP binds (such as NTLM and Kerberos) and enables the LDAP bind to be encrypted over the wire. This action cannot be taken until after the LDAP client is created and it needs to be done using `ldap client modify`.

Bind DNs

The bind DN has to be a valid user in the LDAP server. Bind DNs can follow a variety of different formats. Table 17 shows a list of valid bind DN formats, but generally speaking, bind DNs can follow any standard format an LDAP server supports.

Table 14) Sample bind DN formats.

```
ldapuser
```

```
ldapuser@domain.com
DOMAIN\ldapuser
CN=ldapuser,CN=Users,DC=domain,DC=com
CN=ldapuser,o=Users,o=corp,c=us
```

The user/format being defined *must exist* in the LDAP server for the bind to take place properly. When defining a bind DN, a bind password must also be specified using the following command:

```
cluster::> ldap client modify-bind-password -client-config LDAP -vserver SVM
```

If the password is incorrect, the bind using the bind DN fails and the LDAP connection attempts the next highest security level of bind. If the minimum bind level is set to a value that is not available, the bind fails. For example, if the minimum bind level is set to SASL and the initial SASL authentication fails, no other bind levels is attempted and the authentication fails. Minimum bind level is the minimum bind level that can be attempted in LDAP binds.

The following table shows the order of bind authentication level attempted when the minimum bind level is specified at various values.

Table 15) Bind authentication order versus minimum bind level.

Minimum Bind Level	Bind Authentication Order
SASL	SASL
Simple	SASL, simple
Anonymous	SASL, simple, anonymous

This user needs to have read-only access only to LDAP. In Windows Active Directory, any valid domain user works. For more information on bind levels, including supported SASL mechanisms, see the section “Supported LDAP Bind Levels” in this document.

The following shows a sample LDAP client creation/configuration. This is not a catch-all configuration, because there are a number of variations that the client configuration can have. Contact your LDAP administrator to determine the proper configuration values for your environment.

```
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: NEWSHEMA
LDAP Server Port: 389
Query Timeout (sec): 3
Minimum Bind Authentication Level: sasl
Bind DN (User): ldapuser
Base DN: dc=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, if desired, and only if CIFS is configured. It is not necessary to bind as a CIFS server, but LDAP queries are passed using Kerberos if the CIFS server machine account is used. This process provides stronger encryption for queries. NetApp also recommends in Active Directory environments clearing the LDAP server list of entries, allowing DNS SRV records to determine the servers being used for binds. Clearing the LDAP servers would be done after adding the AD domain information or during the original client creation.

Example:

```
Vserver: vs0
Client Configuration Name: LDAP
LDAP Server List: -
Active Directory Domain: domain.netapp.com
Preferred Active Directory Servers: -
Bind Using the Vserver's CIFS Credentials: true
Schema Template: NEWSHEMA
LDAP Server Port: 389
Query Timeout (sec): 3
Minimum Bind Authentication Level: sasl
Bind DN (User): ldapuser
Base DN: dc=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
```

LDAP Client Configuration in Active Directory Environments

When using Active Directory LDAP servers, you can leverage the domain name in the LDAP client configuration to use [SRV records](#) for LDAP server lookups instead of specifying a list of LDAP servers. Proper DNS configuration on the DNS server as well as the Data ONTAP SVM is required for this to function properly.

LDAP client configuration to leverage LDAP SRV records:

```
cluster::*> ldap client show -client-config LDAP -instance
(vserver services ldap client show)

Vserver: vs0
Client Configuration Name: LDAP
LDAP Server List: -
Active Directory Domain: domain.netapp.com <<< only the domain is needed
Preferred Active Directory Servers: -
Bind Using the Vserver's CIFS Credentials: true
Schema Template: NEWSHEMA
LDAP Server Port: 389
Query Timeout (sec): 3
Minimum Bind Authentication Level: sasl
Bind DN (User): ldapuser
Base DN: dc=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
```

When the SVM is configured to use the domain name as the source for lookups, SRV records are used to determine LDAP servers.

The following output from a packet trace shows that SRV record lookups are used.

Figure 16) SRV record lookup for Active Directory domain.

Protocol	Length	Info
DNS	94	Standard query 0xfbf8 SRV _ldap._tcp.emea.win2k12.netapp.com
DNS	187	Standard query response 0xfbf8 SRV 0 100 389 italy.emea.win2k12.netapp.com

Best Practices 55: SRV Record Lookups for LDAP Servers (see next: Best Practices 56)

When using Active Directory for UNIX user and group lookup, NetApp recommends allowing DNS to look up the LDAP servers using the SRV records. All desired LDAP servers need to be in DNS with SRV records, which is the default for domain controllers in a domain. If you use this method, use the client configuration as seen in the preceding table.

Check DNS for SRV Records

It is possible to query DNS to see if the desired LDAP servers have SRV records using the following Microsoft KB article:

[How to verify that SRV DNS records have been created for a DC](#)

Example:

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

> set type=all
> _ldap._tcp.dc._msdcs
Server: UnKnown
Address: ::1

_ldap._tcp.dc._msdcs.emea.win2k12.netapp.com SRV service location:
    priority = 0
    weight = 100
    port = 389
    svr hostname = italy.emea.win2k12.netapp.com
italy.emea.win2k12.netapp.com internet address = 10.228.225.125
italy.emea.win2k12.netapp.com AAAA IPv6 address = fd20:8b1e:b255:8599:393b:aac6:9478:61ea
```

Distinguished Names (DNs)

A **DN** is a sequence of relative DN (RDN) separated by commas. In LDAP, a DN is essentially a folder structure that specifies locality of objects such as users, groups, machine accounts, netgroups, and so on.

In Active Directory, for instance, the domain itself can be represented as a DN. A domain of `domain.netapp.com` becomes a DN of `dc=domain,dc=netapp,dc=com`.

The following table shows which common RDN types are used in DN.

Table 16) Common RDN values in LDAP.

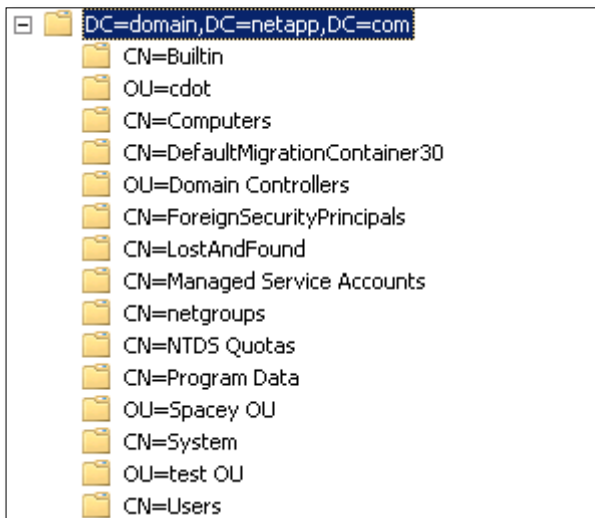
String	Attribute Type
DC	domainComponent

String	Attribute Type
CN	commonName
OU	organizationalUnitName
O	organizationName

In Data ONTAP, it is possible to specify a DN for base, user, group, and netgroup LDAP searches.

In the following figure, ADSI Edit is used to display the folder structure of the LDAP DNs.

Figure 17) LDAP DNs folder structure.



Why Specify a DN?

In the preceding figure, a number of folders could contain users, groups, or netgroups. If the base DN of `dc=domain,dc=netapp,dc=com` represents the domain of `domain.netapp.com`, the LDAP client could certainly be pointed to use the base DN. However, all queries potentially need to crawl each DN listed below `dc=domain,dc=netapp,dc=com`, depending on the search scope (in this case, subtree is the scope). Doing so could add latency to name service and authentication lookups, which could cause authentication failures, access issues, or client connectivity problems. It's critical that name service queries be returned in as little time as possible to avoid issues. Specifying DNs in the user, group, and netgroup lookups can aid in that by filtering requests to the LDAP server.

For example, if a user exists in the DN of `CN=Users,dc=domain,dc=netapp,dc=com`, then the user DN client configuration field could be specified to search in that DN for user objects, rather than at the base DN of `dc=domain,dc=netapp,dc=com`. Doing so eliminates the need to look in any of the other DNs that exist below the base. This is especially critical in large environments, where there could be hundreds or even thousands of DNs.

Multiple DNs

In Data ONTAP 8.2.2 and later, it is possible to specify multiple DNs for object searches. Versions before Data ONTAP 8.2.2 allowed only a single DN to be specified. This enhancement allows users, groups, and netgroups to exist in multiple locations in LDAP with the ability to still be able to filter searches by DN. When specifying multiple DNs, be sure to enclose the entry in double quotes; otherwise the command fails:

```
cluster::*> ldap client modify -client-config DOMAIN -vserver SVM -user-dn
cn=users,dc=domain,dc=netapp,dc=com;OU=cdot,dc=domain,dc=netapp,dc=com
(vserver services ldap client modify)
```

```
Error: "OU=cdot,dc=domain,dc=netapp,dc=com" is not a recognized command
```

Additionally, be sure that the entries use a semicolon separator between DNs. Using a comma to separate the values causes the cluster to see the entry as a single DN.

For example, if the preceding was separated by a comma, LDAP queries look for objects in a DN called *cn=users,dc=domain,dc=netapp,dc=com,OU=cdot,dc=domain,dc=netapp,dc=com* and requests fail because they are being queried in a DN that does not exist.

Example:

```
cluster::*> ldap client modify -client-config DOMAIN -vserver SVM -user-dn
"cn=users,dc=domain,dc=netapp,dc=com,OU=cdot,dc=domain,dc=netapp,dc=com"

cluster::*> diag secd authentication translate -node node2 -vserver SVM -unix-user-name test

Vserver: SVM (internal ID: 3)

Error: Acquire UNIX credentials procedure failed
[ 0 ms] Name 'test' not found in UNIX authorization source LOCAL
[ 0] Connecting to LDAP (NIS & Name Mapping) server
      10.228.225.120
[ 4] Using a new connection to 10.228.225.120
[ 6] Name 'test' not found in UNIX authorization source LDAP
[ 6] Could not get a user ID for name 'test' using any
      NS-SWITCH authorization source
**[ 6] FAILURE: Unable to retrieve UID for UNIX user test
```

When multiple DNs are specified correctly (semicolons, double quotes), users contained in multiple locations can be filtered and searches can be significantly faster.

```
cluster::*> ldap client modify -client-config DOMAIN -vserver SVM -user-dn
"cn=users,dc=domain,dc=netapp,dc=com;OU=cdot,dc=domain,dc=netapp,dc=com"

cluster::*> diag secd authentication translate -node node2 -vserver SVM -unix-user-name test
10001

cluster::*> diag secd authentication translate -node node2 -vserver SVM -unix-user-name cdot
100069
```

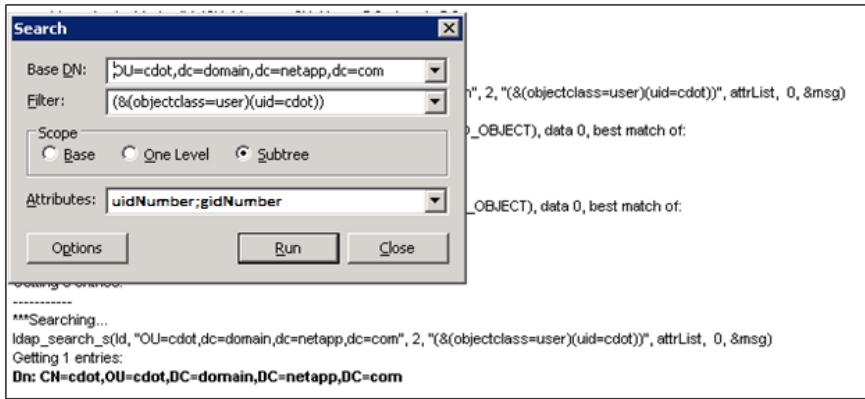
In the secd logs, the LDAP search attributes can be seen. Viewing these can be useful for troubleshooting LDAP issues, because the queries can be copied and pasted right into external LDAP search utilities, such as [ldp.exe](#) or the `ldapsearch` command.

```
[kern_sec:info:32181] | [000.008.557] 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=cdot)) { in searchLdap() at utils/secd_ldap_utils.cpp:262 }
[kern_sec:info:32181] | [000.010.169] debug: Searching LDAP for the "uidNumber, gidNumber"
attribute(s) within base "OU=cdot,dc=domain,dc=netapp,dc=com" (scope: 2) using filter:
(&(objectClass=User)(uid=cdot)) { in searchLdap() at utils/secd_ldap_utils.cpp:262 }
```

In the preceding, the attribute `uidNumber`, `gidNumber` and filter values `(&(objectClass=User)(uid=cdot))` are seen as well as the scope type `(scope: 2)` and which DN is being searched.

The following figures are screen captures of `ldp.exe` being used to search the same values as seen in the preceding.

Figure 20) Using ldap.exe to search for objects in LDAP.



Search Scopes

In addition to DN's, search scopes can be specified for LDAP queries. A scope is the starting point for LDAP queries and at what depth from the base DN the search should occur. The following search scopes are valid for LDAP queries in Data ONTAP.

Table 17) Search scope types.

Scopes
<p>Base</p> <p>A base search scope indicates that LDAP searches should occur only for the specified base DN.</p> <p>For example, if a user DN is set to <i>cn=users,dc=domain,dc=netapp,dc=com</i>, and a <i>base</i> search scope is specified, then an LDAP search occurs <i>only</i> for <i>cn=users,dc=domain,dc=netapp,dc=com</i>. It does not include objects inside of the DN. <i>Base is not a recommended search scope, because it is a literal search scope.</i></p> <p>For example, a base query for the user “ldapuser” succeeds only if the DN specified is the user’s DN:</p>
<pre>cluster::*> ldap client modify -client-config DOMAIN -vserver SVM -user-dn "cn=ldapuser,cn=users,dc=domain,dc=netapp,dc=com" -user-scope base (vserver services ldap client modify) cluster::*> diag secd authentication translate -node node2 -vserver SVM -unix-user-name ldapuser 1107</pre>
<p>When the DN is changed to the “Users” container, lookups fail:</p>
<pre>cluster::*> ldap client modify -client-config DOMAIN -vserver SVM -user-dn "cn=users,dc=domain,dc=netapp,dc=com" -user-scope base (vserver services ldap client modify) cluster::*> diag secd authentication translate -node node2 -vserver SVM -unix-user-name ldapuser Vserver: SVM (internal ID: 3) Error: Acquire UNIX credentials procedure failed [0 ms] Name 'ldapuser' not found in UNIX authorization source LOCAL [0] Connecting to LDAP (NIS & Name Mapping) server 10.228.225.120 [6] Using a new connection to 10.228.225.120 [8] Name 'ldapuser' not found in UNIX authorization source LDAP [8] Could not get a user ID for name 'ldapuser' using any NS-SWITCH authorization source **[8] FAILURE: Unable to retrieve UID for UNIX user ldapuser Error: command failed: Failed to resolve user name to a UNIX ID. Reason: "SecD Error: user not found".</pre>
<p>This is because the lookup is looking specifically for the specified DN.</p>

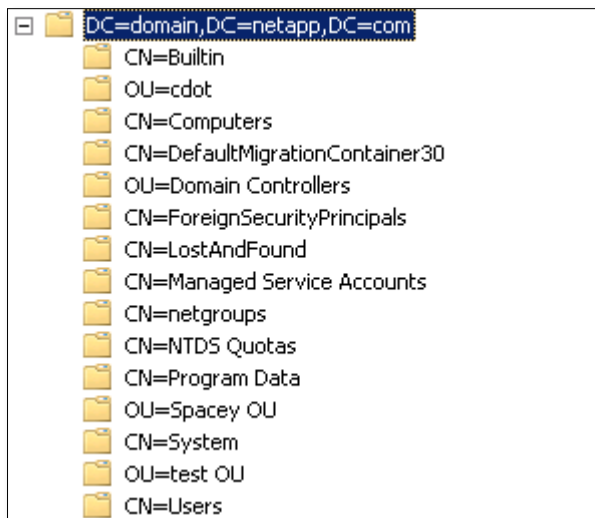
Subtree

A subtree search scope searches *all* levels below the specified DN.

For example, if a user DN is set to `cn=users,dc=domain,dc=netapp,dc=com`, and a *subtree* search scope is specified, then an LDAP search occurs for all objects below `cn=users,dc=domain,dc=netapp,dc=com`. These objects include other containers. *Subtree* is a recommended search scope in most cases, provided that the DN specified is at a level low enough to filter effectively.

In the following figure, the DN of `dc=domain,dc=netapp,dc=com` has several containers beneath it. If a search scope of subtree is used, then LDAP queries look in each DN below `dc=domain,dc=netapp,dc=com`. It might be more desirable to specify the DN at a more granular level for subtree searches, such as `cn=users,dc=domain,dc=netapp,dc=com`.

Figure 18) LDAP DN containers.



Onelevel

A onelevel search scope searches only at one level below the specified level, including the DN itself, but not entries below the one level.

For example, if a user DN is set to `cn=users,dc=domain,dc=netapp,dc=com`, and a *onelevel* search scope is specified, then LDAP search occurs only in DNs one level below `cn=users,dc=domain,dc=netapp,dc=com`.

```
cluster::*> ldap client modify -client-config DOMAIN -vserver SVM -user-dn
"cn=users,dc=domain,dc=netapp,dc=com" -user-scope onelevel
(vserver services ldap client modify)
```

```
cluster::*> diag secd authentication translate -node node2 -vserver SVM -unix-user-name
ldapuser
1107
```

If the DN is set at one level higher, the lookup fails:

```
cluster::*> ldap client modify -client-config DOMAIN -vserver SVM -user-dn
"dc=domain,dc=netapp,dc=com" -user-scope onelevel
(vserver services ldap client modify)
```

```
parisi-cdot::*> diag secd authentication translate -node node2 -vserver SVM -unix-user-name
ldapuser
```

Vserver: SVM (internal ID: 3)

Error: Acquire UNIX credentials procedure failed

```
[ 0 ms] Name 'ldapuser' not found in UNIX authorization source
LOCAL
[ 0] Connecting to LDAP (NIS & Name Mapping) server
10.228.225.120
[ 5] Using a new connection to 10.228.225.120
[ 7] Name 'ldapuser' not found in UNIX authorization source
LDAP
[ 7] Could not get a user ID for name 'ldapuser' using any
NS-SWITCH authorization source
**[ 7] FAILURE: Unable to retrieve UID for UNIX user ldapuser
```

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

Onelevel search scopes are ideal for filtering at a very granular level, but using them can result in lookup failures if the correct DNs are not specified.

Search Scopes as Seen in Logs

Search scopes can be found in the secD log as numeric representations. For example, if base is used as the search scope, the secD log shows `(scope: 0)`, because base has a numeric translation of 0.

The following table shows the numeric representations for all variations of search scopes in Data ONTAP.

Table 18) Numeric representation of search scopes in secd logs.

Search Scope	Numeric
Not set	-1
Base	0
Onelevel	1
Subtree	2

Using Host Names for LDAP Servers

In Data ONTAP, the specification of LDAP servers generally uses the `-servers` option in the client configuration. In this option, only server IP addresses are presently allowed. Attempting to use a host name results in the following error:

```
cluster::*> ldap client modify -client-config test2 -servers server.domain.netapp.com
Error: "domain.netapp.com" is an invalid value for field "-servers <IP Address>", ...
```

However, there are specific use cases in which host names need to be used for LDAP server specification. These include:

- Load balancers (IP addresses/host names that redirect to multiple LDAP servers)
- More servers in an environment than allowed by the `-servers` option
- Obfuscation of LDAP servers for simpler configuration and maintenance

In these instances, there is a way to add host name information to Data ONTAP client configurations and work around the `-servers` option limitation.

Using the `-ad-domain` Option

Although `-servers` does not allow the specification of host names, the `-ad-domain` option does. *Even in environments that do not use Active Directory for LDAP services, this option can be used.*

When `-ad-domain` is specified, the SecD process uses DNS and SRV records for LDAP services to locate any and all LDAP servers configured in the environment. However, when you use this method, the following considerations must be made:

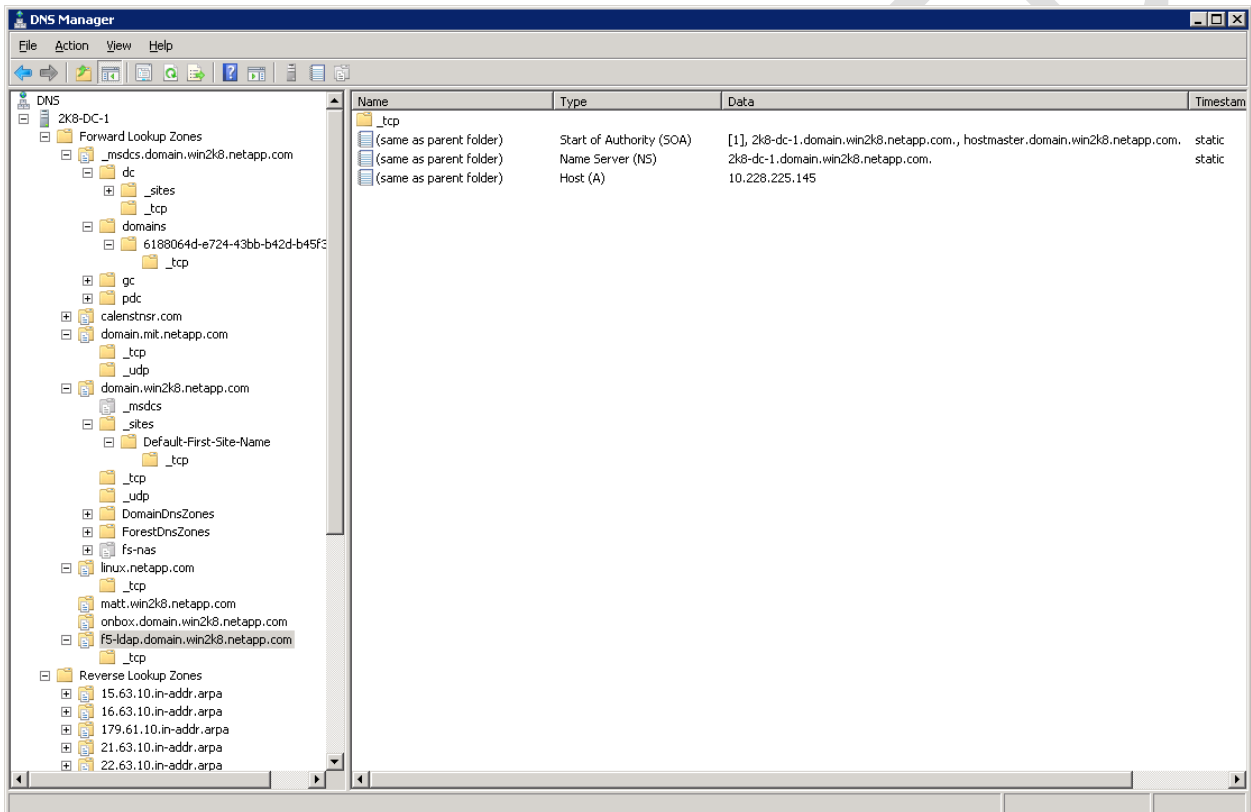
- The host name specified in the `-ad-domain` option *must* have an A/AAAA record and associated LDAP SRV records. For information on creating LDAP SRV records, see the section in this document on [LDAP SRV records](#).
- SRV records *must* point to an A/AAAA record. SRV records cannot point to CNAMEs.
- If you intend to use Kerberos to bind to LDAP servers, a corresponding SPN *must* be created for the DNS A/AAAA record being used.
- All LDAP servers that are intended for use with the specified host name *must* be added to the LDAP SRV record.
- When you specify host names, do not specify any LDAP servers in the `-servers` list.

The following is a sample client configuration in which a host name is used for the LDAP server:

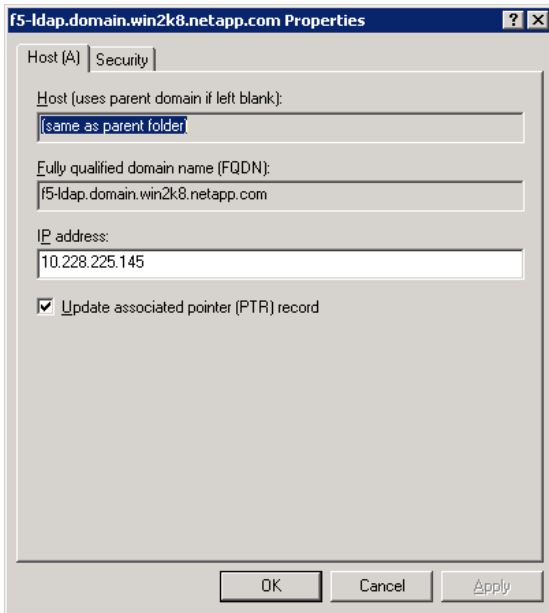
```
Vserver: NAS
Client Configuration Name: DIRSRV
LDAP Server List: -
Active Directory Domain: domain.mit.netapp.com
Preferred Active Directory Servers: -
```

If you intend to use a host name that is in the same DNS domain as the domain controllers, then a separate DNS stub zone with a corresponding A/AAAA and PTR record might be needed to prevent clashes between preexisting SRV records for the domain.

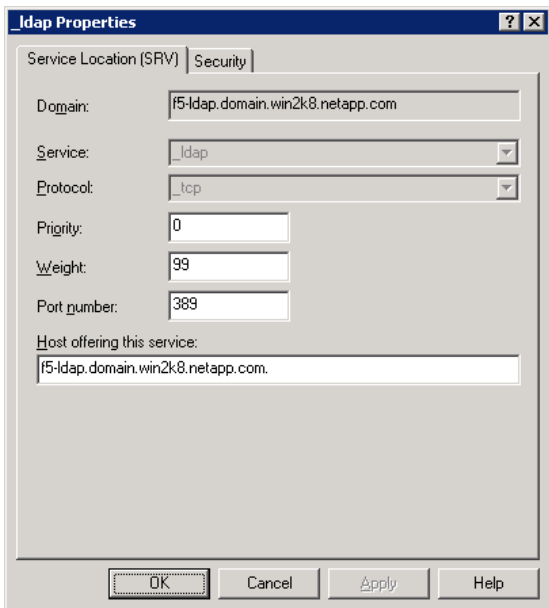
For example, if I want to use an F5 load balancer with an FQDN (in the following example, *f5-ldap.domain.win2k8.netapp.com*) in my domain to service LDAP requests, and it is in the same DNS domain as my domain controllers (*domain.win2k8.netapp.com*), my stub zone looks like this:



The A/AAAA record exists within the zone, but it is not an authoritative name server. The record maintains the same name as the parent folder:



An SRV record is created in that zone that points to that A/AAAA record:



When this is done, SRV records should be able to be queried from SecD on the cluster. Be sure to test for both the domain and the stub zone:

```
cluster::> set diag
cluster::*> diag secd dns srv-lookup -node nodel -vserver SVM -lookup-string
_ldap._tcp.domain.win2k8.netapp.com
10.228.225.120
fd20:8b1e:b255:8599:5457:61d9:fc87:423f

cluster::*> diag secd dns srv-lookup -node nodel -vserver SVM -lookup-string _ldap._tcp.f5-
ldap.domain.win2k8.netapp.com
10.228.225.145
```

The LDAP client configuration then uses that FQDN in the `-ad-domain:`

```
Vserver: SVM
      Client Configuration Name: F5
      LDAP Server List: -
      Active Directory Domain: f5-ldap.domain.win2k8.netapp.com
```

LDAP binds and searches come in using the load balancer and are forwarded to the LDAP servers.

Configuring Data ONTAP LDAP Clients to Use Global Catalog Searches

If you choose to use global catalog servers to honor LDAP lookups, Data ONTAP SVMs can be configured to leverage those servers to enable multiple domain lookups for UNIX users in LDAP searches.

The only cluster configuration requirement for using global catalog searches for LDAP lookups (aside from the [domain considerations](#)) is changing the LDAP client port to use the Active Directory port for global catalogs (port 3268).

Note: Port 3269 is used for global catalog searches over SSL. This is not currently supported in Data ONTAP.

After Active Directory sees a request come in on that specific port, it treats the query as a global catalog search rather than a basic LDAP search and bypasses the need for referrals. The following client configuration can be used to leverage global catalog searches with Data ONTAP.

```
cluster::*> ldap client show -client-config GlobalCatalog

      Vserver: SVM
      Client Configuration Name: GlobalCatalog
      LDAP Server List: [list of global catalogs,comma separated] MUST BE GC
      Active Directory Domain: [parent domain]
      Preferred Active Directory Servers: [list of global catalogs,comma separated]
      Bind Using the Vserver's CIFS Credentials: [true|false] If using CIFS, set to true
      Schema Template: AD-IDMU
      LDAP Server Port: 3268
      Query Timeout (sec): 3
      Minimum Bind Authentication Level: [simple|sasl]
      Bind DN (User): [user@parentdomain]
      Base DN: [can be blank or set to parent domain base]
      Base Search Scope: subtree
      User DN: [optional; can be multiple DNs]
      User Search Scope: subtree
      Group DN: [optional; can be multiple DNs]
      Group Search Scope: subtree
      Netgroup DN: [optional; only if using netgroups]
      Netgroup Search Scope: subtree
      Vserver Owns Configuration: true
      Use start-tls Over LDAP Connections: false
      Enable Netgroup-By-Host Lookup: false
      Netgroup-By-Host DN: -
      Netgroup-By-Host Scope: subtree
```

Example:

```
cluster::*> ldap client show -client-config GlobalCatalog

Vserver: SVM
Client Configuration Name: GlobalCatalog
LDAP Server List: 10.228.225.125
Active Directory Domain: emea.win2k12.netapp.com
Preferred Active Directory Servers: 10.228.225.125
Bind Using the Vserver's CIFS Credentials: false
Schema Template: AD-IDMU
LDAP Server Port: 3268
Query Timeout (sec): 3
Minimum Bind Authentication Level: sasl
Bind DN (User): ldapuser@emea.win2k12.netapp.com
Base DN:
Base Search Scope: subtree
User DN:
DC=emea,DC=win2k12,DC=netapp,DC=com;DC=france,DC=emea,DC=win2k12,DC=netapp,DC=com;DC=germany,DC=emea,DC=win2k12,DC=netapp,DC=com
User Search Scope: subtree
Group DN:
DC=emea,DC=win2k12,DC=netapp,DC=com;DC=france,DC=emea,DC=win2k12,DC=netapp,DC=com;DC=germany,DC=emea,DC=win2k12,DC=netapp,DC=com
Group Search Scope: subtree
Netgroup DN: -
Netgroup Search Scope: subtree
Vserver Owns Configuration: true
Use start-tls Over LDAP Connections: false
Enable Netgroup-By-Host Lookup: false
Netgroup-By-Host DN: -
Netgroup-By-Host Scope: subtree
```

When used in conjunction with use of multiple DNs, global catalog searches are ideal for large enterprise environments with multiple trusted domains in the same forest.

Should the Base DN Be Specified?

When using global catalog searches, a base DN is not needed, provided port 3268 is used. The Active Directory server negotiates the base DN.

From the TechNet article on [global catalog and LDAP searches](#):

For an LDAP search, you must supply a valid base distinguished name. For a Global Catalog search, the base distinguished name can be any value, including the value "NULL" (" "). A base distinguished name of NULL effectively scopes the search on the search computer to the Global Catalog. If you use a NULL base distinguished name with a scope of one level or subtree and specify port 389 (the default LDAP port), the search fails. Therefore, if you submit a NULL search to the Global Catalog port and then change the port to the LDAP port, you must change the base distinguished name for the search to succeed.

Supported LDAP Bind Levels

Bind levels are methods of authentication when a client attempts to perform LDAP queries. Bind levels can be one of the following types:

- **Anonymous.** No credentials are needed to perform queries. This option is disabled on most modern LDAP servers by default.
- **Simple.** The user name and password are delivered in plain text over the wire.
- **SASL.** Credentials are passed using a Kerberos ticket or encrypted packets. NetApp recommends this configuration.

When you set the `-min-bind-level` option on a Data ONTAP LDAP client, you are setting the *minimum allowed* authentication level. Data ONTAP always tries the strongest authentication type

possible first, but the `-min-bind-level` setting dictates what happens if stronger authentication types fail. For example, if the `-min-bind-level` is set to SASL, then only SASL authentication is attempted. If SASL authentication is not working properly, then LDAP queries fail and do not fall back to weaker bind levels.

The following table shows the supported bind levels offered in specific versions of Data ONTAP.

Table 19) Supported LDAP bind levels.

Data ONTAP Version	Supported Bind Levels
ONTAP 9.0 and earlier	Anonymous Simple SASL: GSS-SPNEGO (DES, RC4, AES-128, AES-256) Note: GSSAPI, DIGEST-MD5, Blowfish, and SHA1/SHA2 are currently not supported.

LDAP Configuration

After 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
```

LDAPS

LDAPS (LDAP over SSL) for Active Directory was introduced in Data ONTAP 8.2.1 to allow encrypted LDAP queries. Such encrypted queries prevent plain text LDAP queries from traveling over the wire. To configure SSL-encrypted LDAP queries, a Certificate Server must be configured in the domain.

Note: LDAP over SSL uses LDAP over [StartTLS](#). As a result, secure LDAP over ports other than 389 is currently not supported. The use of StartTLS in LDAP is covered in [RFC-2830](#).

If a Certificate Server exists and is configured, then setting up LDAPS is a straightforward process.

For information on [how to configure Active Directory to use LDAP over SSL in Data ONTAP](#), see the [configuration section in this document](#).

LDAP Signing and Sealing

In ONTAP 9.0, support for LDAP signing and sealing was added. Signing and sealing provide protection for LDAP operations throughout the process. Signing provides integrity checking of the origin making the connection and sealing provides encryption for the LDAP requests and responses. With Active Directory LDAP, that means that the bind (signing) will be done using Kerberos AES-256 encryption (Windows 2012 and later) and the LDAP request/response packets will be encrypted using GSS-API (sealing). In Microsoft Active Directory LDAP environments, this option can be an easier-to-configure option to use in lieu of [LDAP over SSL/start-TLS](#).

- Use of LDAP signing and sealing requires configuration on the Active Directory domain through Group Policy. For more information, see the [Domain Controller: LDAP server signing requirements on TechNet](#).
- To enable LDAP signing and sealing, use the `-session-security` option on the LDAP client for the SVM. There are three valid options: none, sign, and seal.

Figure 19) Packet capture of LDAP signing and sealing.

KRB5	349	AS-REQ
KRB5	1597	AS-REP
TCP	66	46435 > kerberos [ACK] Seq=284 Ack=1532 win=65024 Len=0 TSv
TCP	66	46435 > kerberos [FIN, ACK] Seq=284 Ack=1532 win=66560 Len=
TCP	66	kerberos > 46435 [ACK] Seq=1532 Ack=285 win=66560 Len=0 TS
TCP	54	kerberos > 46435 [RST, ACK] Seq=1532 Ack=285 win=0 Len=0
TCP	74	56471 > ldap [SYN] Seq=0 win=65535 Len=0 MSS=8960 WS=64 SA
TCP	74	ldap > 56471 [SYN, ACK] Seq=0 Ack=1 win=8192 Len=0 MSS=146
TCP	66	56471 > ldap [ACK] Seq=1 Ack=1 win=66560 Len=0 TSval=75302
DNS	86	Standard query 0xe9b4 PTR 181.67.193.10.in-addr.arpa
DNS	132	Standard query response 0xe9b4 PTR stme-infra02.core-tme.l
TCP	74	53418 > kerberos [SYN] Seq=0 win=65535 Len=0 MSS=8960 WS=64
TCP	74	kerberos > 53418 [SYN, ACK] Seq=0 Ack=1 win=8192 Len=0 MSS
TCP	66	53418 > kerberos [ACK] Seq=1 Ack=1 win=66560 Len=0 TSval=7
TCP	1514	[TCP segment of a reassembled PDU]
KRB5	107	TGS-REQ
TCP	66	kerberos > 53418 [ACK] Seq=1 Ack=1490 win=66560 Len=0 TSva
KRB5	1540	TGS-REP
TCP	66	53418 > kerberos [ACK] Seq=1490 Ack=1475 win=65088 Len=0 T
TCP	66	53418 > kerberos [FIN, ACK] Seq=1490 Ack=1475 win=66560 Le
TCP	66	kerberos > 53418 [ACK] Seq=1475 Ack=1491 win=66560 Len=0 T
TCP	54	kerberos > 53418 [RST, ACK] Seq=1475 Ack=1491 win=0 Len=0
LDAP	1447	bindRequest(1) "<ROOT>" sas1
LDAP	248	bindResponse(1) sas1bindInProgress
LDAP	88	bindRequest(2) "<ROOT>" sas1
LDAP	122	bindResponse(2) sas1bindInProgress
LDAP	122	bindRequest(3) "<ROOT>" sas1
LDAP	90	bindResponse(3) success
LDAP	308	SASL GSS-API Privacy: payload (178 bytes)
LDAP	735	SASL GSS-API Privacy: payload (605 bytes)

LDAP Signing and Sealing Process

The LDAP signing and sealing process is as follows:

- ONTAP uses LDAP client configuration to determine which LDAP server to use.
 - If ad-domain is used, uses SRV record lookup
 - If LDAP servers are specified, tries first in list
- ONTAP makes an AS-REQ request for a Kerberos ticket-granting ticket (krbtgt) to the KDC using the CIFS server/machine account UPN when “bind as CIFS server” is enabled (for example, [CIFS\\$@REALM.NETAPP.COM](mailto:CIFS$@REALM.NETAPP.COM)).
 - If “bind as CIFS server” is disabled, uses the LDAP Bind DN user
- If the request is granted, ONTAP issues a DNS PTR lookup for the domain pointer.
- ONTAP then uses the FQDN from the DNS request and the krbtgt to issue a TGS-REQ using the LDAP SPN ldap/server-from-ptr-request.
- Kerberos TGS-REQ is granted and strongest supported enctype is used to issue ticket to ONTAP.
- ONTAP uses the TGS to issue a SASL bind request to the LDAP server.
- If accepted, ONTAP binds to LDAP and issues the LDAP request.
- LDAP requests are encrypted and do not show LDAP-specific information.

Figure 20) LDAP search packet comparison: sealed versus unsealed.

LDAP with sealing:

```

429 21.8195420 10.193.67.220 10.193.67.181 LDAP 308 SASL GSS-API Privacy: payload (178 bytes)
430 21.8222210 10.193.67.181 10.193.67.220 LDAP 735 SASL GSS-API Privacy: payload (605 bytes)
Transmission Control Protocol, Src Port: 56471 (56471), Dst Port: ldap (389), Seq: 1460, Ack: 263, Len: 242
Lightweight Directory Access Protocol
SASL Buffer Length: 238
SASL Buffer
GSS-API Generic Security Service Application Program Interface
krb5_blob: 050406ff000000000000000001fa95079e3426c5332f6878b...
krb5_tok_id: KRB_TOKEN_CFX_WRAP (0x0405)
krb5_cfx_flags: 0x06
.... .1.. = AcceptorSubkey: Set
.... .1. = Sealed: Set
.... ...0 = SendByAcceptor: Not set
krb5_filler: ff
krb5_cfx_ec: 0
krb5_cfx_rrc: 0
krb5_cfx_seq: 531189881
krb5_sgn_cksum: e3426c5332f6878b620693ac004ad4ad0df00ff9b8eaff09...
GSS-API Encrypted payload (178 bytes)

```

LDAP without sealing:

```

47 2.09764900 10.193.67.181 10.193.67.220 LDAP 278 bindResponse(1) success
48 2.10129500 10.193.67.220 10.193.67.181 LDAP 244 searchRequest(2) "DC=CORE-TME,DC=NETAPP,DC=COM" wholesubtree
49 2.10394400 10.193.67.181 10.193.67.220 LDAP 671 searchResEntry(2) "CN=ldapuser,OU=Users,OU=Users and Groups,DC=core-tme,DC=netapp,DC=com"
Transmission Control Protocol, Src Port: 48812 (48812), Dst Port: ldap (389), Seq: 1437, Ack: 213, Len: 178
Lightweight Directory Access Protocol
LDAPMessage searchRequest(2) "DC=CORE-TME,DC=NETAPP,DC=COM" wholesubtree
messageID: 2
protocolOp: searchRequest(3)
searchRequest
baseObject: DC=CORE-TME,DC=NETAPP,DC=COM
scope: wholesubtree(2)
derefAliases: neverDerefAliases(0)
sizeLimit: 0
timeLimit: 3
typesOnly: False
Filter: (&(objectClass=user)(uid=ldapuser))
filter: and(0)
and: (&(objectClass=user)(uid=ldapuser))
and: 2 items
Filter: (objectClass=user)
and item: equalityMatch(3)
equalityMatch
attributeDesc: objectClass
assertionValue: user
Filter: (uid=ldapuser)
and item: equalityMatch(3)
equalityMatch
attributeDesc: uid
assertionValue: ldapuser
attributes: 7 items
AttributeDescription: uid
AttributeDescription: uidNumber
AttributeDescription: gidNumber

```

To sign *and* seal connections, use seal.

```

[-session-security {none|sign|seal}] - Client Session Security
This parameter specifies the level of security to be used for LDAP communications. If you do not
specify this parameter, the default is none.

LDAP Client Session Security can be one of the following:

o none - No Signing or Sealing.

o sign - Sign LDAP traffic.

o seal - Seal and Sign LDAP traffic.

```

LDAP signing and sealing should not be confused with SMB signing and sealing, which is covered in [TR-4543: SMB Protocol Best Practices](#).

Name Service Switch (ns-switch) Configuration

The following sections cover setting the ns-switch configuration to leverage LDAP for name and group lookups in Data ONTAP.

SVM Configuration in Versions Earlier Than 8.3

In Data ONTAP versions before 8.3, a different name service mechanism was used. As a result, name service configuration was done at the SVM configuration.

After the LDAP configuration is enabled for use, the SVM must be configured to use LDAP in its name mapping switch and/or name service lookups.

```
cluster::> vserver modify -vserver vs0 -ns-switch file,ldap -nm-switch file,ldap
```

Note: Only specify LDAP in `nm-switch` or `ns-switch` if LDAP is being used for that functionality. Specifying a name service for something when it is not in use can inject latency and failures in name service requests.

SVM Configuration in Data ONTAP 8.3 and Later

Data ONTAP 8.3 and later has changed how name services are implemented and managed. Rather than making changes on the SVM, name services have been moved to their own command set `vserver services name-service`. This change enables a more global approach to name services.

The following are included under this command set:

- DNS
- LDAP
- UNIX users and groups (local passwd and group entries)
- Netgroup
- NIS domain
- NS switch
- GetXXByYY

New ns-switch Functionality

In Data ONTAP 8.3, `ns-switch` functionality has been adjusted to more closely reflect the standard nsswitch.conf format found on Linux hosts. As a result, storage administrators can now set different source entries for:

- Groups
- Hosts
- Name mappings
- Netgroups
- Passwd (users)

Example:

```
cluster::> name-service ns-switch show -vserver vs0
(vserver services name-service ns-switch show)
Vserver      Database      Source
-----
vs0          hosts         files,
              dns
vs0          group         files,
              ldap
vs0          passwd        files,
              ldap
vs0          netgroup      files
vs0          namemap       files,
              ldap
5 entries were displayed.
```

This capability allows greater flexibility when configuring the SVM for name services.

Best Practices 56: NS-Switch Best Practice (see next: Best Practices 57)

It is a best practice to include `file` as a primary `ns-switch` and `nm-switch` entry, even if name service servers are in use. NetApp also recommends listing “file” first in the list. Doing so guarantees that local users and groups are always returned first and in a rapid manner and protects against LDAP server outages creating issues for SVMs authenticating with local files.

Scaled Mode/File-Only Mode

Scaled mode/file-only mode for local users and groups in ONTAP 9.1 enables storage administrators to expand the limits of local users and groups. It does so by enabling a **diag-level** name service option and then using the load-from-uri functionality to load files into the cluster to provide higher numbers of users and groups. Scaled mode/file-only mode also can add performance improvements to name service lookups, because there is no longer a need to have external dependencies on name service servers, networks, and so on. However, this performance comes at the expense of ease of management of the name services, because file management adds overhead to the storage management and introduces more potential for human error. Additionally, local file management must be performed per cluster, adding an extra layer of complexity.

Best Practices 57: File-Only Mode: Local UNIX Users and Groups (see next: Best Practices 1)

Be sure to evaluate your options at length and make the appropriate decision for your environment. Consider file-only mode only if you require a name service environment that needs more than 64k users/groups.

For more information on file-only mode for UNIX users and groups, see [TR-4067: NFS Best Practice and Implementation Guide](#).

5.8 Setting Up NFSv4

The following section describes how to set up NFSv4 for use with Data ONTAP. This section covers client and cluster configuration. For more information on NFSv4 implementation in Data ONTAP, see the [Data ONTAP NFS Implementation Guide](#).

Note: [Quick Step Setup](#) steps can be found at the end of this document.

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. NFS is available on all types 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

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 demanded 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 with the Microsoft Windows environment
- Protocol extension to support backward compatibility
- Movement toward an open standard, managed by the IETF

Additional information

- For detailed information on enhancements, see [RFC3530](#).
- For NFSv4 best practices, see [TR-3580](#).
- For NFS implementation in Data ONTAP, see [TR-4067](#).
- For condensed NFSv4 setup steps, see the “[Quick Step Setup Guides](#)” section in this document.
- For [complete configuration steps for NFSv4.x in Data ONTAP](#), see the [configuration section of this document](#). That section covers the cluster NFS configuration only. The external NFSv4.x configuration is covered in the following sections.

Configuring the Identity Management Server for NFSv4.x

The identity management 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, because 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.

Configuring the NFS Clients for NFSv4.x

The following section describes how to configure the NFS clients for use with NFSv4.x in 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 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 following table covers which services are restarted on each client.

Table 20) Services for NFSv4.

OS	Commands
RHEL/CentOS/Fedora	<code>service rpcidmapd [start stop restart status]</code>
SLES/SUSE	<code>service nfs [start stop restart status]</code>
Ubuntu	<code>service idmapd [start stop restart status]</code>
Solaris	<code>svcadm [enable disable restart refresh] mapid</code> <code>svcs -l mapid (to list)</code>

For complete setup steps, see the client documentation.

[Solaris NFSv4](#)

[CentOS NFSv4](#)

[RHEL NFSv4](#)

[SUSE/SLES NFSv4](#)

[Ubuntu NFSv4](#)

6 Configuration Steps

This section contains the configuration steps from the previous sections. These were moved to remove clutter from the informational portions of this document and to consolidate them all into one location.

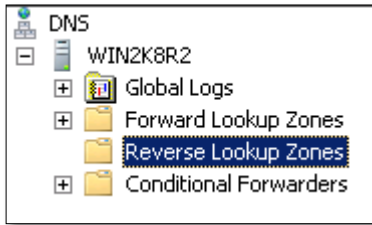
6.1 DNS Configuration

Adding NFS Clients to Windows DNS

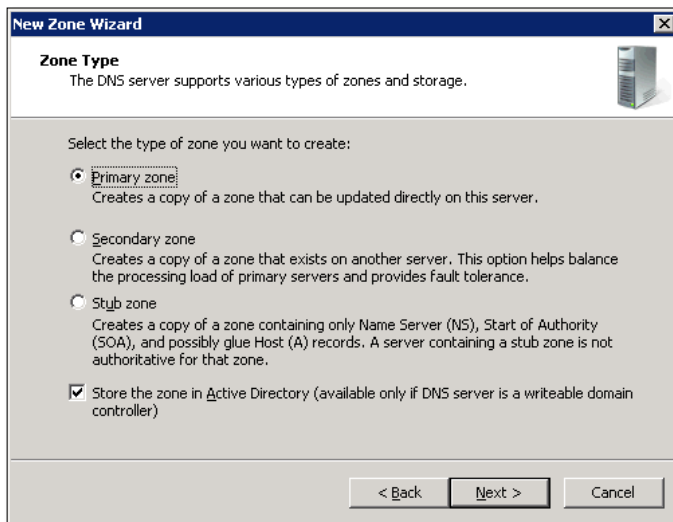
This section applies to all Windows versions from 2003 onward. This section does not cover non-Windows DNS servers. Adding clients can be done using the DNS GUI or using the command line (cmd). This section covers both methods.

Configuration Steps 2) Adding NFS client to Windows DNS (GUI).

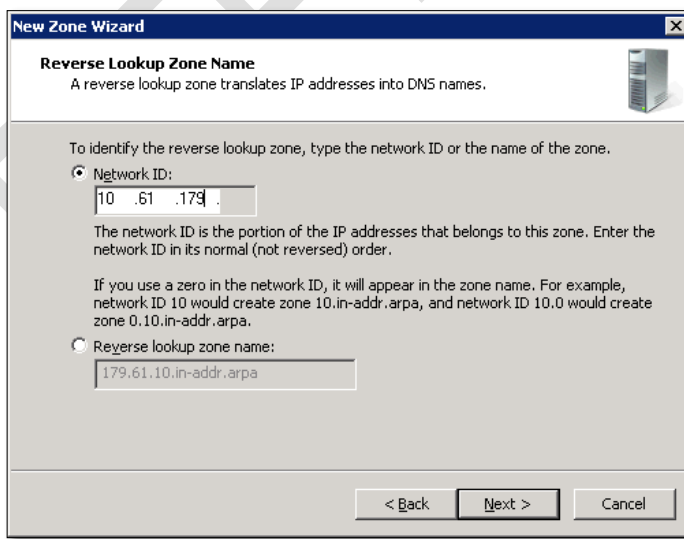
1. Log in to the Windows server running DNS in Active Directory. In many cases, this server is a domain controller.
2. Navigate to Start -> Administrative Tools -> DNS.
3. Expand the DNS server and the Reverse Lookup Zones folder.



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.

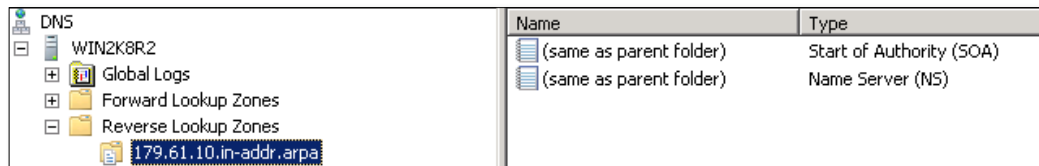


6. Accept the defaults until the Reverse Lookup Zone Name window appears. Enter the first three octets of the client's IP address.

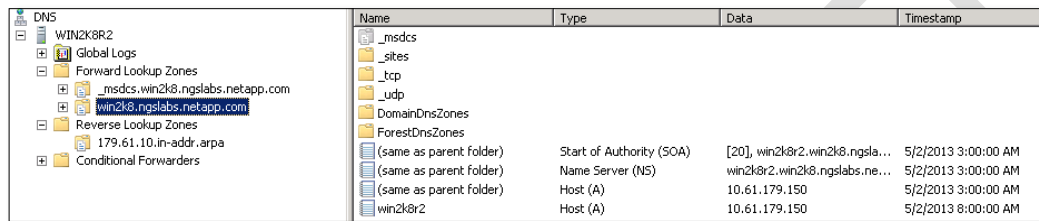


7. Select the desired option for Dynamic Updates and then click Next and Finish.

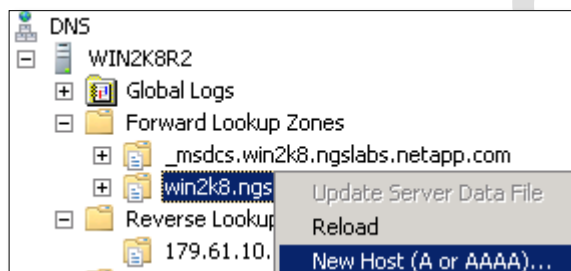
8. The reverse lookup zone is now created.



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



10. Right-click 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 Pointer Record (PTR) to create the reverse lookup record.

New Host [X]

Name (uses parent domain name if blank):
centos6

Fully qualified domain name (FQDN):
centos6.win2k8.ngslabs.netapp.com.

IP address:
10.61.179.170

Create associated pointer (PTR) record
 Allow any authenticated user to update DNS records with the same owner name

Add Host Cancel

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

centos6	Host (A)	10.61.179.170
---------	----------	---------------

Configuration Steps 3) 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:

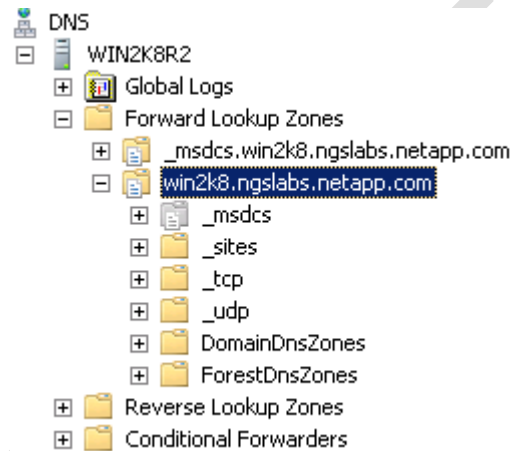
```
C:\> dnscmd /RecordAdd domain.netapp.com nfsclient /CreatePTR A 10.61.179.170
```

Before running the preceding commands, verify that the reverse lookup zone exists for the subnet in which the client lives. Steps to create reverse lookup zones are covered in [TR-4523: DNS Load Balancing in ONTAP](#).

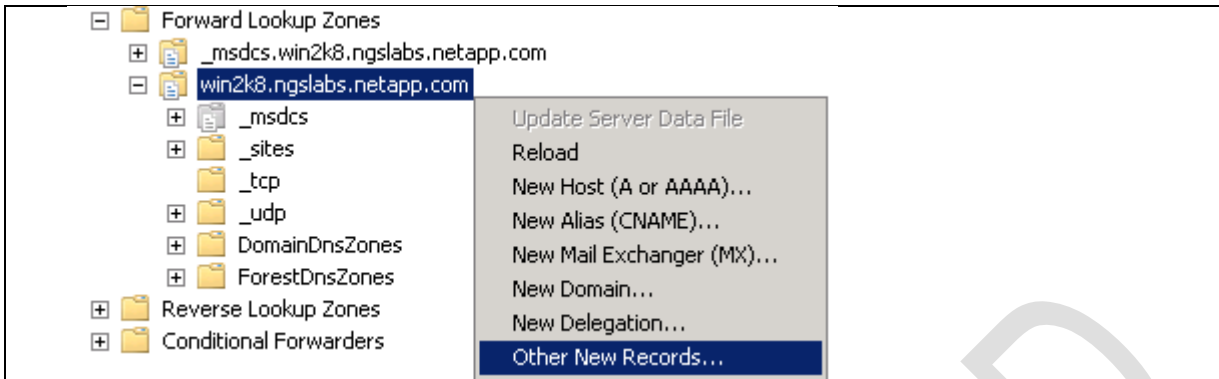
Creating SRV Records for Kerberos-Master

Configuration Steps 4) Creating SRV records for Kerberos-Master.

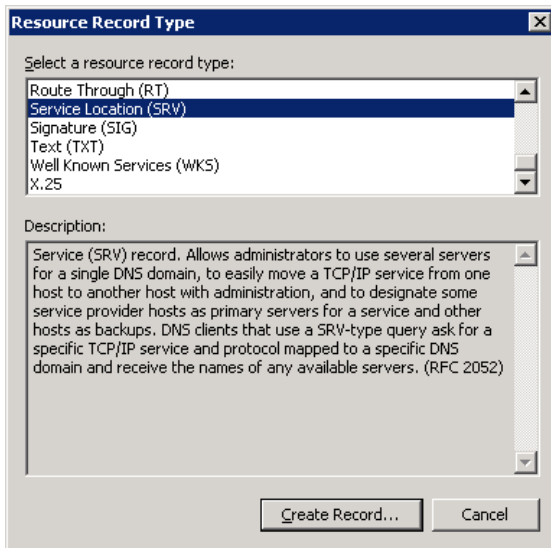
1. Log in to the DNS server.
2. Open the DNS GUI and navigate to the DNS domain folder under Forward Lookup Zones.



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



4. Select the Resource Record 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, because 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.
```


Configuration Steps 5) 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.

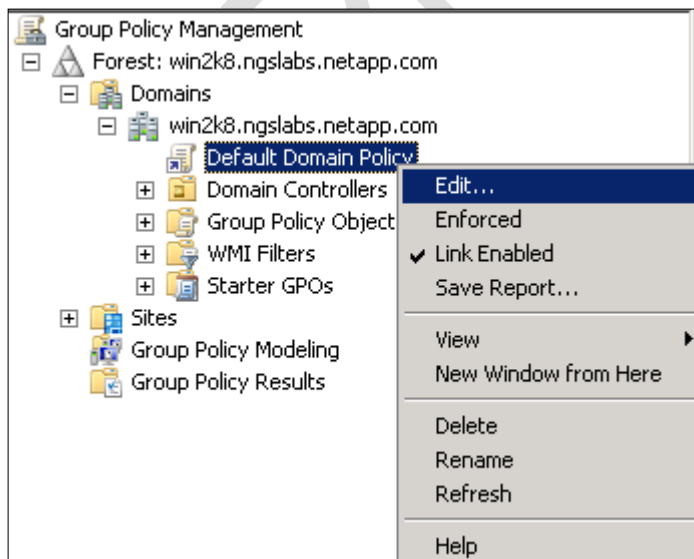
6.2 Kerberos Configuration—Manual Keytab Creation

The following section covers how to configure NFS clients for Kerberos in Active Directory domains. These steps use a manual process of machine account (principal) and keytab file creation. For a less manual configuration, consider using a [domain join method](#). Domain joins create the keytab and machine account (principals) for you.

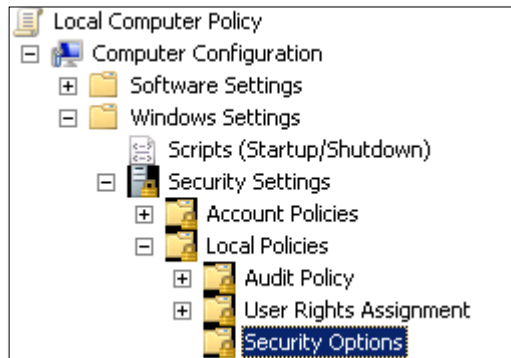
Enabling DES and/or AES in Windows 2008 R2 and Later

Configuration Steps 6) Allowing DES encryption types in Windows 2008 R2 and later.

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 Default Domain Policy and select Edit.



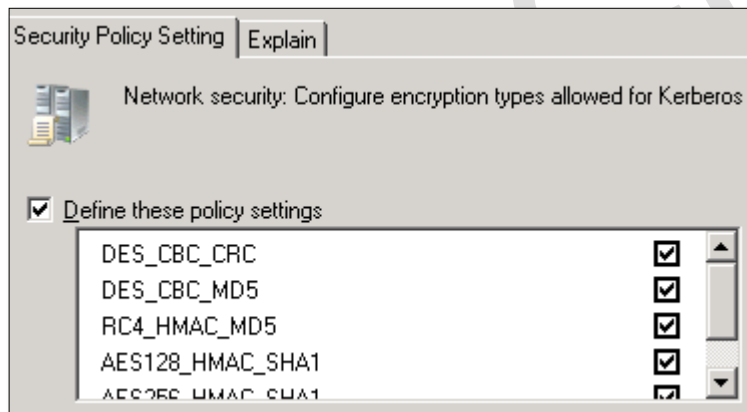
4. Navigate to Computer Configuration -> Policies -> Windows Settings -> Security Settings -> Local Policies -> Security Options.



5. Select Network Security: Configure Encryption Types Allowed for Kerberos and double-click.



6. Select the encryption types desired. If using a Data ONTAP version before 8.3, DES is needed and the rest of the types are more secure than DES, so select them all. Selecting only DES prevents other encytypes for the domain. In Data ONTAP 8.3 and later, do not select DES encryption types. Use only AES.



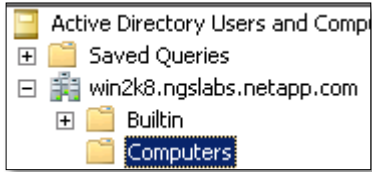
7. Click Apply. A reboot is not required to apply the change.

Creating the Principals as Machine Accounts

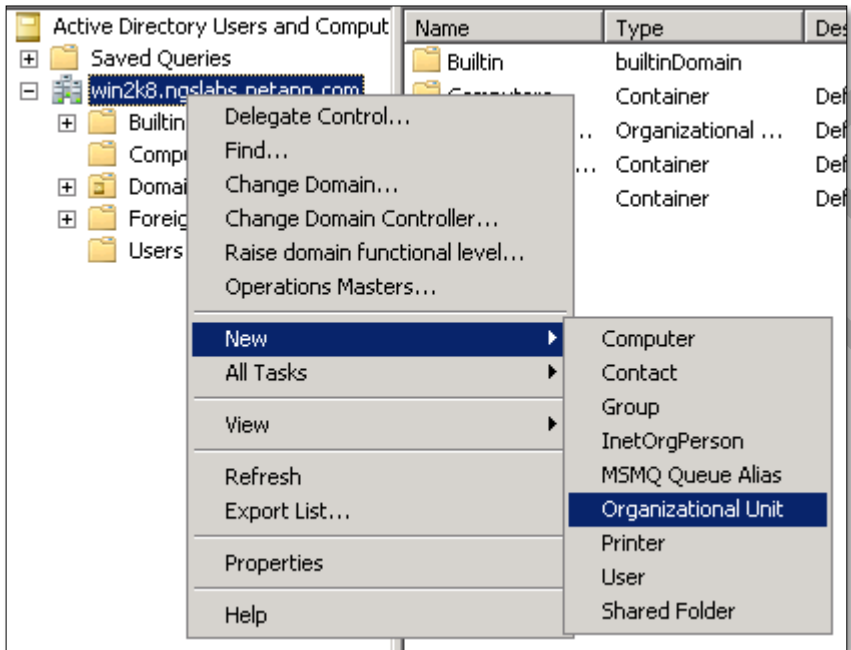
Configuration Steps 7) Creating machine accounts in Active Directory (GUI).

Note: Machine accounts do not need to be created for the NFS clients if you use a [domain join method](#) of Kerberos configuration.

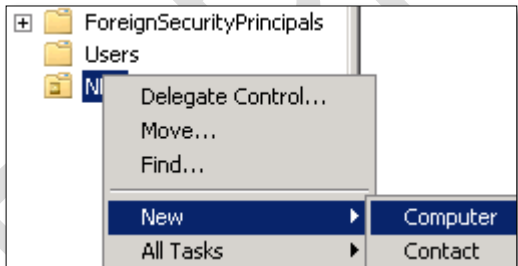
1. Go to Start -> Administrative Tools and select Active Directory Users and Computers.
2. In the GUI, select the organizational unit (OU) Computers or create a sub-OU. By default, all machine accounts live in Computers.



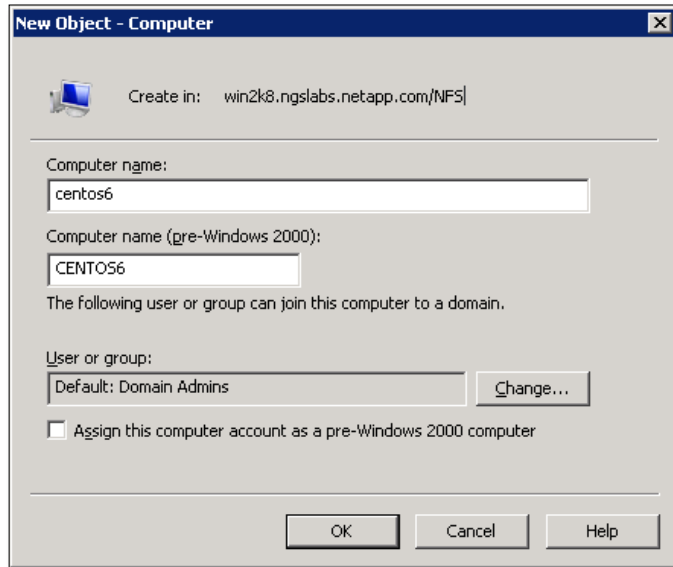
3. To create a sub-OU (optional), right-click the domain and select New -> Organizational Unit.



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 need to be modified to allow DES](#) (if using Data ONTAP 8.2.x or earlier).

Configuration Steps 8) Creating machine accounts in Active Directory (dsadd).

Note: Machine accounts do not need to be created for the NFS clients if you use a [domain join method](#) of Kerberos configuration.

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=nfsclient,OU=NFS,DC=domain,DC=netapp,DC=com
```

Note: After creating the machine account in this manner, the [attributes need to be modified to allow DES](#).

Configuration Steps 9) Creating machine accounts in Active Directory (Windows PowerShell).

Note: Machine accounts do not need to be created for the NFS clients if you use a [domain join method](#) of Kerberos configuration.

1. Log in to the domain controller and open Windows 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'=27}
```

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

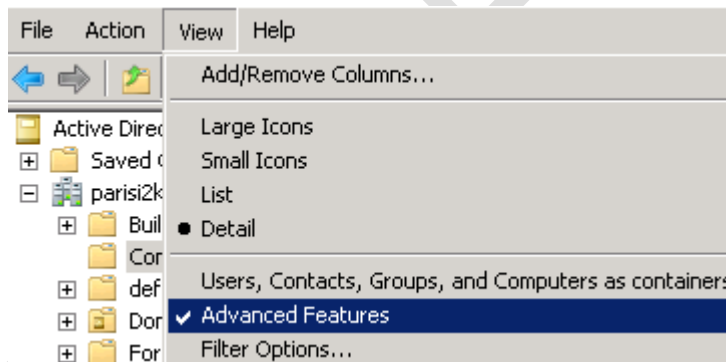
Modifying Machine Accounts in Windows Active Directory

The following tables show how to modify the account using the Attributes Editor tab as well as using ADSI Edit and `ldifde` commands. Windows 2008 was used for the DES examples, and Windows 2012 was used for the AES examples, but the steps are interchangeable for the operating system versions. 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.

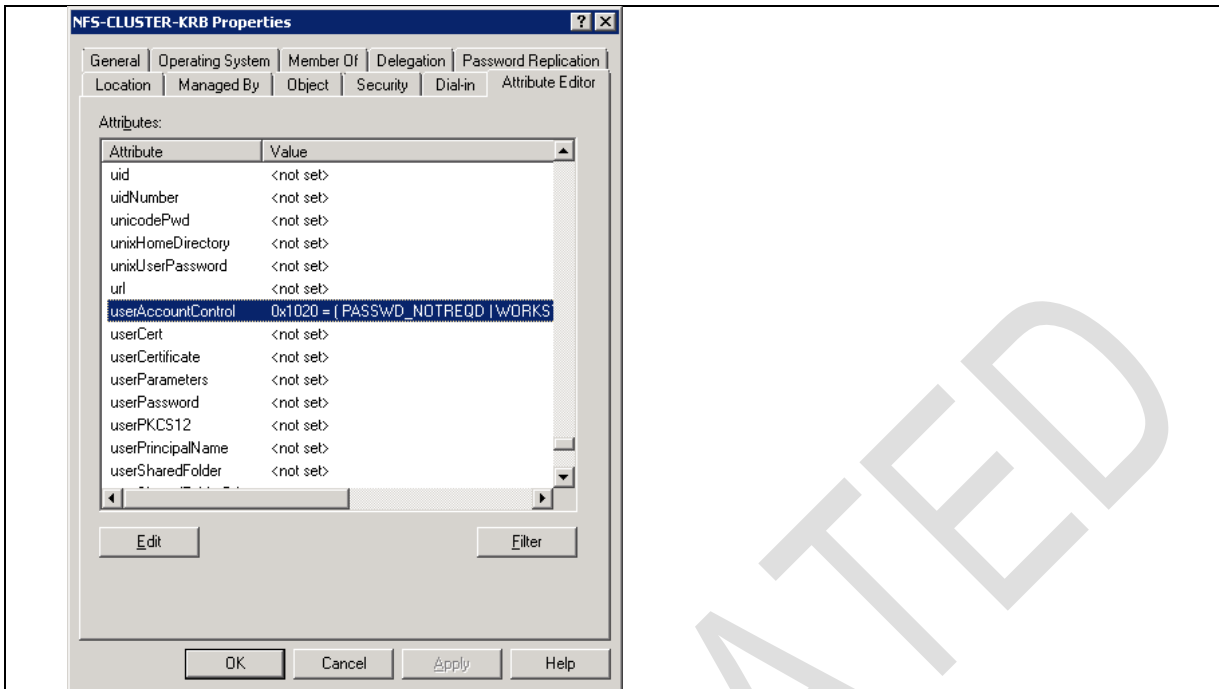
Configuration Steps 10) Modifying the NFS server machine account for DES_CBC_MD5 (Attributes Editor).

1. Log in to the domain controller and open Active Directory Users and Computers.

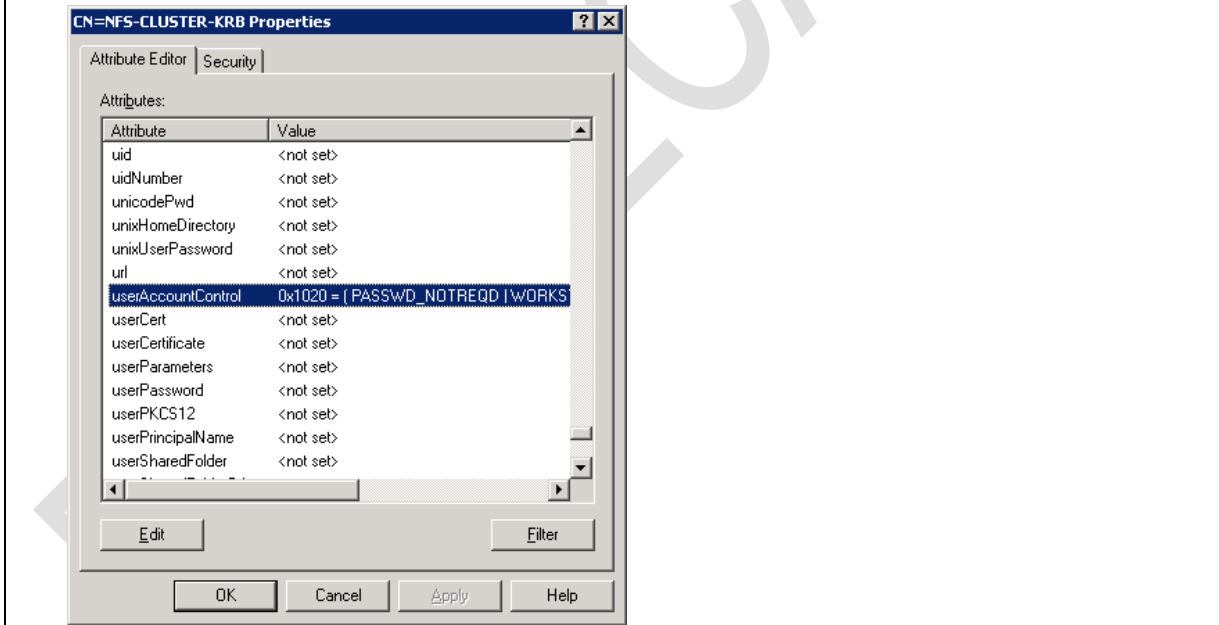
2. Click View and select Advanced Features.



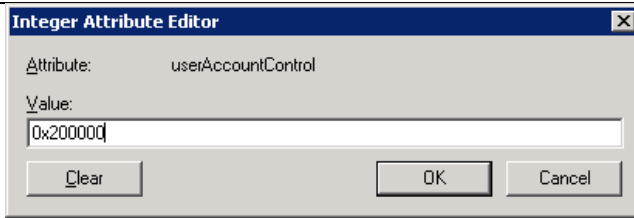
3. After this is done, a new tab called Attributes Editor appears under the machine account properties.



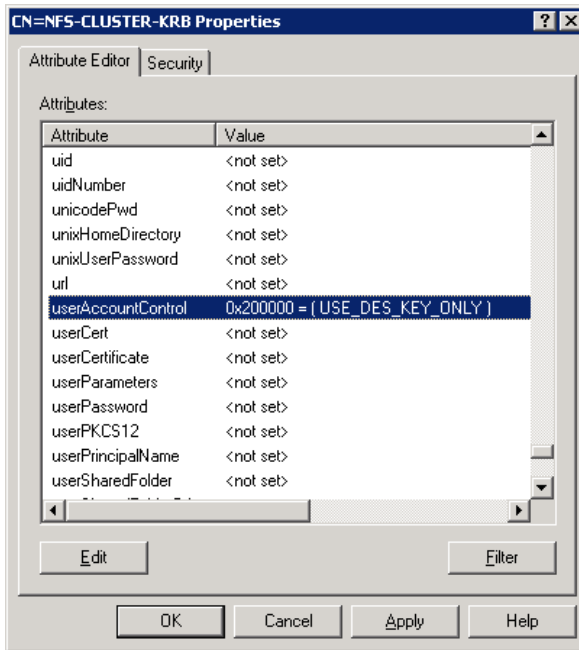
4. Navigate to the userAccountControl attribute.



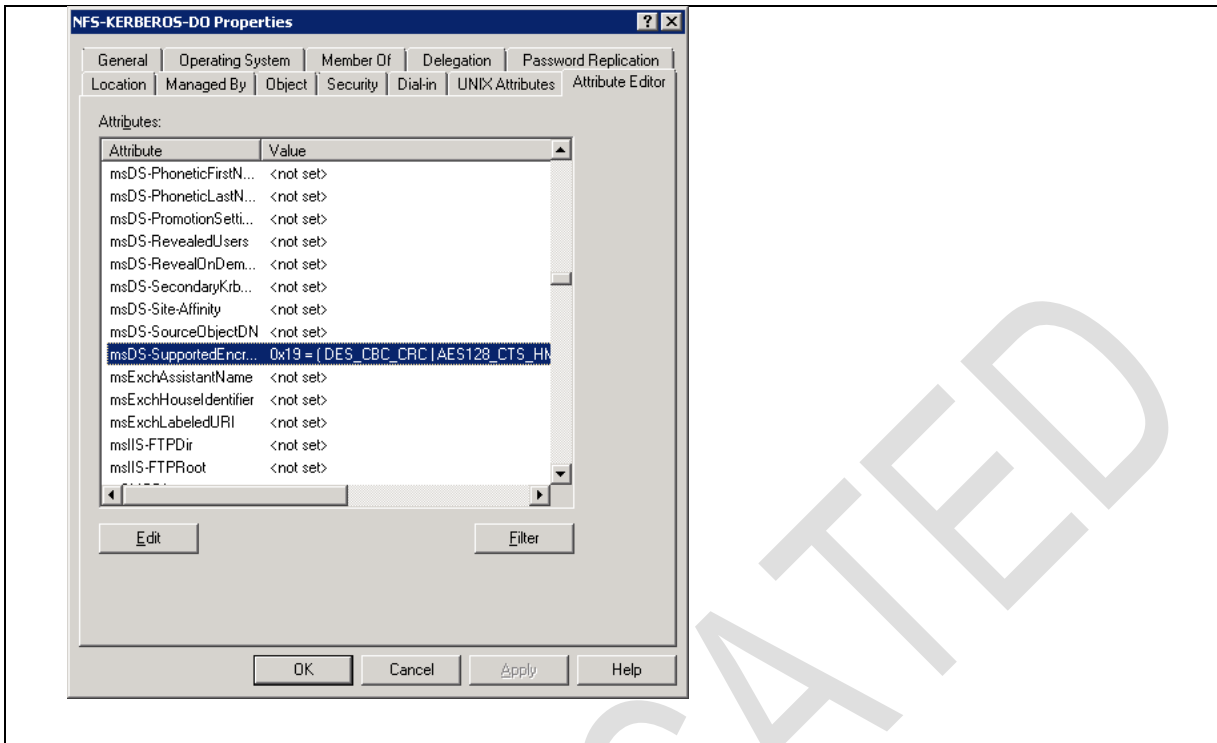
5. Click Edit and change the value to 0x200000 (USE_DES_KEY_ONLY). This action is required only in 8.2.x and earlier, because 8.3 adds support for AES encryption.



6. Click OK and verify the change:

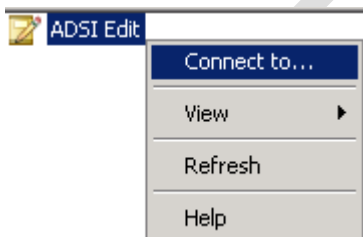


7. In Windows 2008 R2, an attribute called msDS-SupportedEncryptionTypes was added. This option should be set to allow all encyptypes. Change this value to 25 (0x19 in hex) to allow all encryption types for the machine account. (This option did not exist before Windows 2008 R2.)

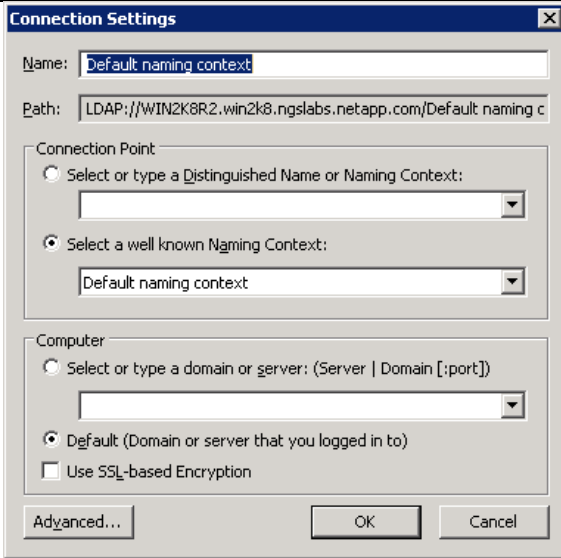


Configuration Steps 11) Modifying the NFS server machine account for DES_CBC_MD5 (ADSI edit).

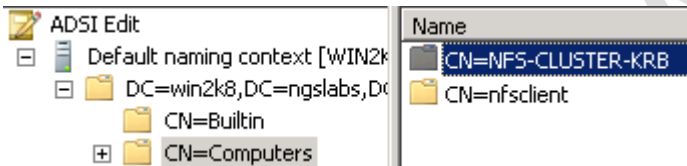
1. Log in to the domain controller; go to Start -> Run and type adsiedit.msc.
2. Connect to the Active Directory database by right-clicking and selecting Connect To.



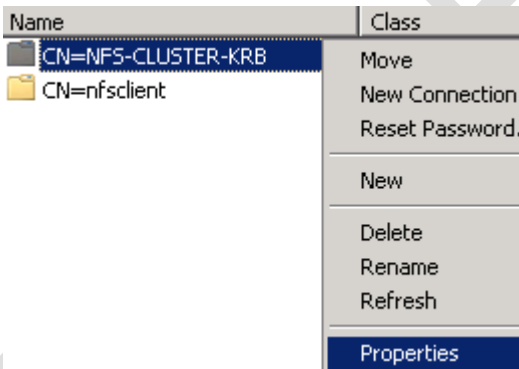
3. Leave the defaults and click OK.



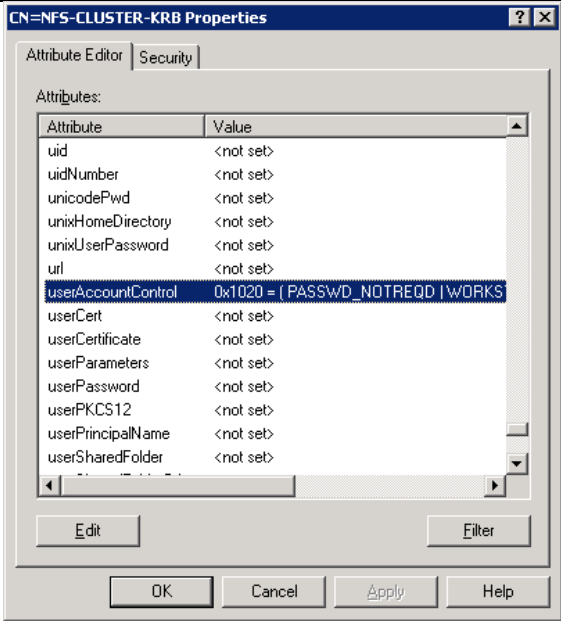
4. Navigate to the Computers container (where the NFS machine account was created).



5. Right-click the object and select Properties.



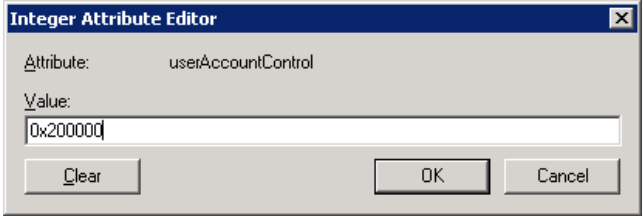
6. Navigate to the userAccountControl attribute.



Attributes:

Attribute	Value
uid	<not set>
uidNumber	<not set>
unicodePwd	<not set>
unixHomeDirectory	<not set>
unixUserPassword	<not set>
url	<not set>
userAccountControl	0x1020 = (PASSWD_NOTREQD WORKS
userCert	<not set>
userCertificate	<not set>
userParameters	<not set>
userPassword	<not set>
userPKCS12	<not set>
userPrincipalName	<not set>
userSharedFolder	<not set>

7. Click Edit and change the value to 0x200000 (USE_DES_KEY_ONLY). This step is required only in 8.2.x and earlier, because 8.3 adds support for AES encryption.

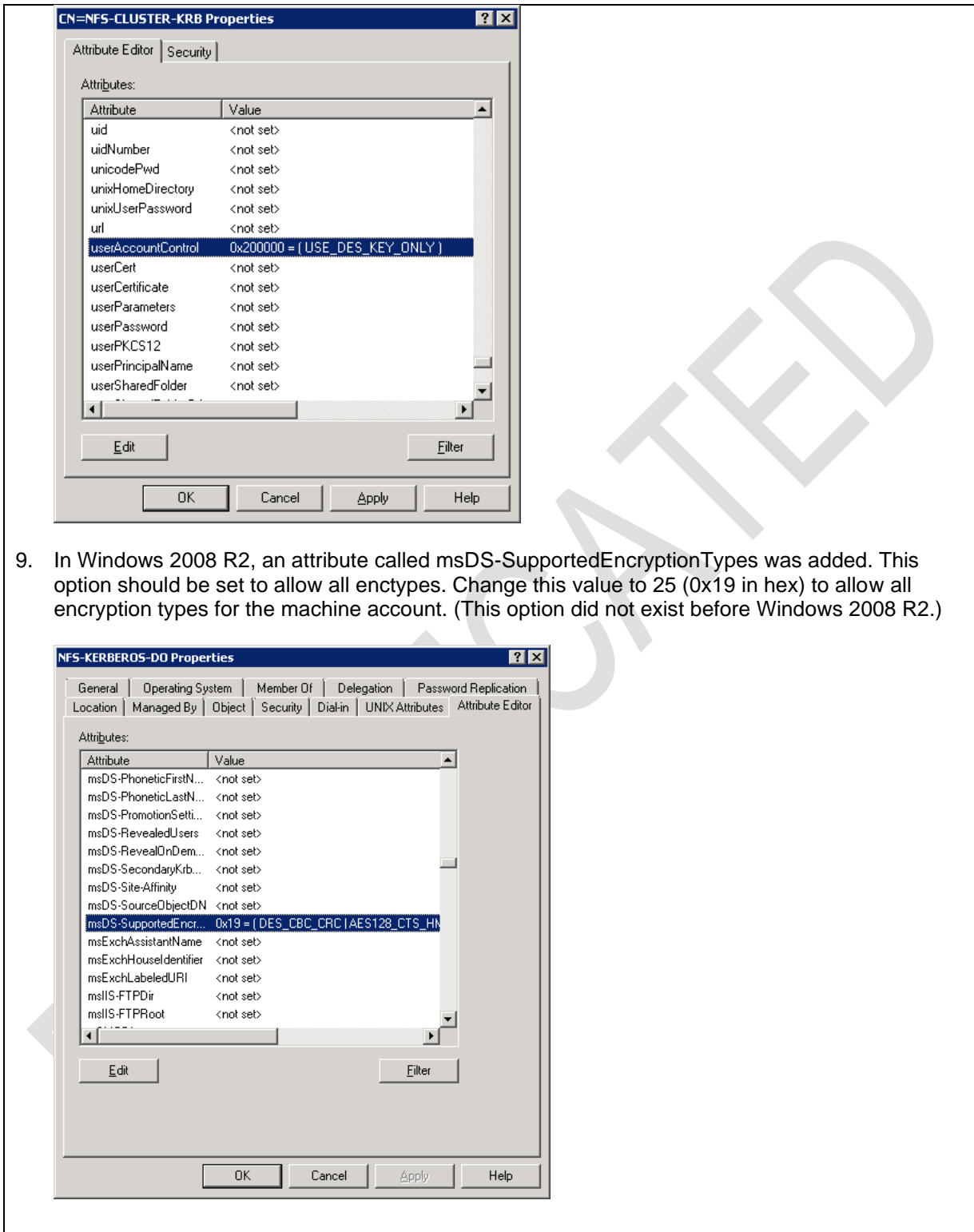


Integer Attribute Editor

Attribute: userAccountControl

Value: 0x200000

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 encrytpes. Change this value to 25 (0x19 in hex) to allow all encryption types for the machine account. (This option did not exist before Windows 2008 R2.)

Configuration Steps 12) Modifying the NFS server machine account for DES_CBC_MD5 (import using ldfidfe).

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: 27
-
```

Note: The preceding 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 following file 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=linux-client))"
-l "msDS-SupportedEncryptionTypes"
```

Configuration Steps 13) Creating machine accounts for use with DES in Active Directory (Windows PowerShell).

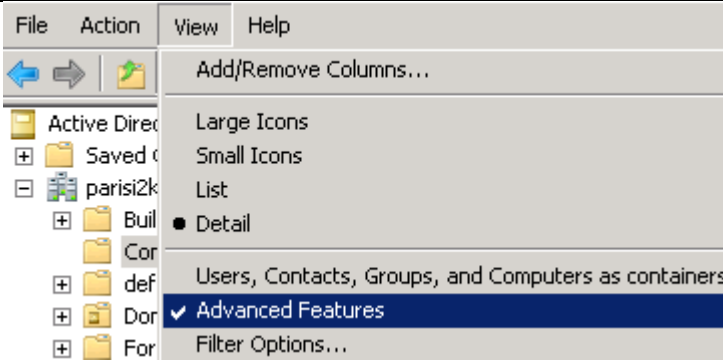
1. Log in to the domain controller and open Windows PowerShell.
2. Type the following ([click for more info on set-ADComputer](#)):

```
PS C:\> import-module activedirectory
PS C:\> Set-ADComputer -Identity [NFSservername] -Replace
@{'userAccountControl'=2097152;'msDS-SupportedEncryptionTypes'=27}
```

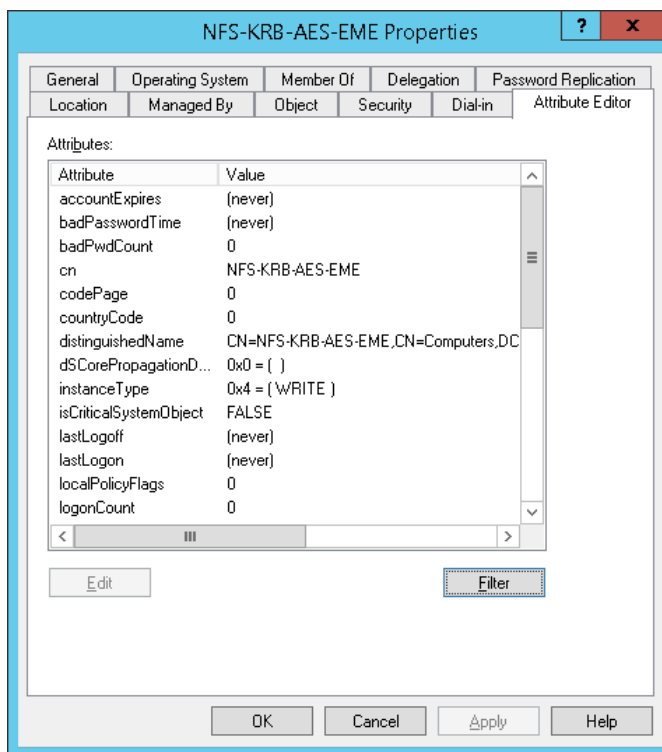
Note: For the NFS client machine account, repeat the preceding steps for *the msDS-SupportedEncryptionTypes* value only.

Configuration Steps 14) Modifying the NFS machine account to use/support AES (Attributes Editor).

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



3. After this is done, a new tab called Attributes Editor appears under the machine account properties.



4. Click Filter and clear the value that says Show Only Attributes That Have Values.

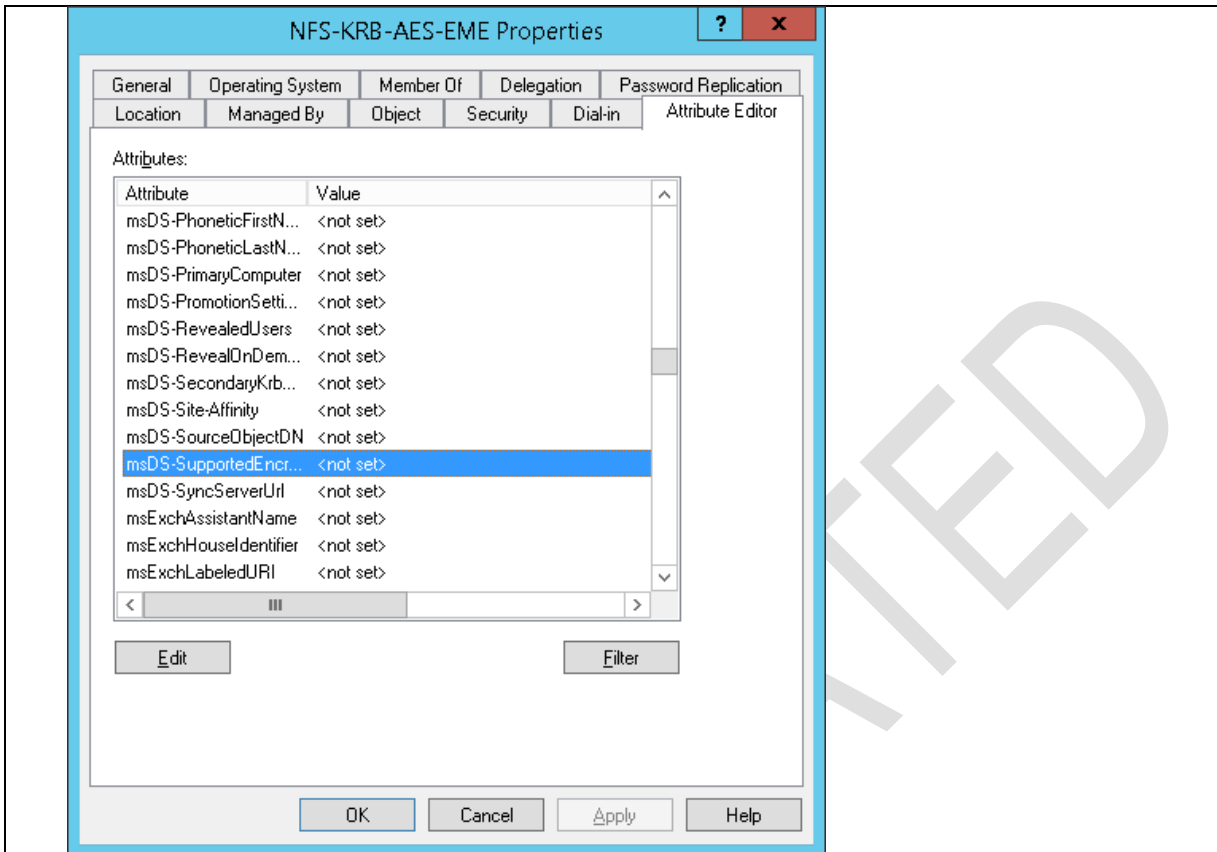
The screenshot shows the 'NFS-KRB-AES-EME Properties' dialog box with the 'Attribute Editor' tab selected. The 'Attributes' list is as follows:

Attribute	Value
accountExpires	(never)
badPasswordTime	(never)
badPwdCount	0
cn	NFS-KRB-AES-EME
codePage	0
countryCode	0
distinguishedName	CN=NFS-KRB-AES-EME,CN=Computers,DC
dSCorePropagationD...	0x0 = ()
instanceType	0x4 = (WRITE)
isCriticalSystemObject	FALSE
lastLogoff	(never)
lastLogon	(never)
localPolicyFlags	0
logonCount	0

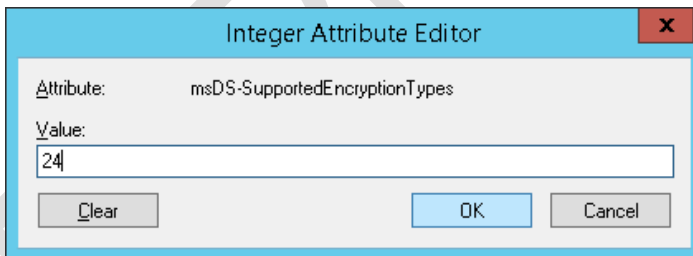
The 'Filter' menu is open, showing the following options:

- Show only attributes that have values
- Show only writable attributes
- Show attributes:
- Mandatory
- Optional
- Show read-only attributes:
- Constructed
- Backlinks
- System-only

5. After this is done, all values are displayed. Navigate to the value msDs-SupportedEncryptionTypes. Modifying UserAccountControl when using AES is not necessary.



- This value controls which supported encryption types are allowed for the machine account. The table in the appendix regarding [Kerberos property flags](#) shows which values are valid for this. Because AES-256 is 16, AES-128 is 8, DES MD5 is 2, and DES CRC is 1, the total value to allow all 4 is 27 (16 + 8 + 2 + 1). However, it's best to allow only the strongest encryption types for Kerberos. Thus, enable only AES with 24 as the value.



Note: It is important *not* to enable RC4-HMAC on this machine account, because the Kerberos requests might attempt to use RC4 regardless of the client configuration. RC4 is not supported in Data ONTAP for NFS Kerberos operations.

For more information about the [userAccountControl](#) and [msDS-SupportedEncryptionTypes](#) values, see the section “[About the Machine Account Attributes](#)” in the appendix of this document. For information on DES, AES, and other encyptes, see the section “[Kerberos Encryption Types](#).”

Creating Keytab Files

The following table shows the steps for creating a keytab file on an Active Directory KDC.

Note: Keytab files do not need to be created for the NFS clients if you use a [domain join method](#) of Kerberos configuration.

Configuration Steps 15) 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\nfsclient$ -crypto ALL +rndpass -ptype KRB5_NT_PRINCIPAL +Answer -out C:\nfsclient-
all.keytab

Targeting domain controller: win2k8DC.domain.netapp.com

Using legacy password setting method

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

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

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

Reset NFSCLIENT$'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 nfsclient-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 preceding command creates a keytab file called `nfsclient-all.keytab` on the `C:\` drive. It also assigns an SPN and 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. However, this is unnecessary and a potential security risk. See [About the Machine Account Attributes](#) for further details.

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

```

[root@nfsclient ~]# 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)

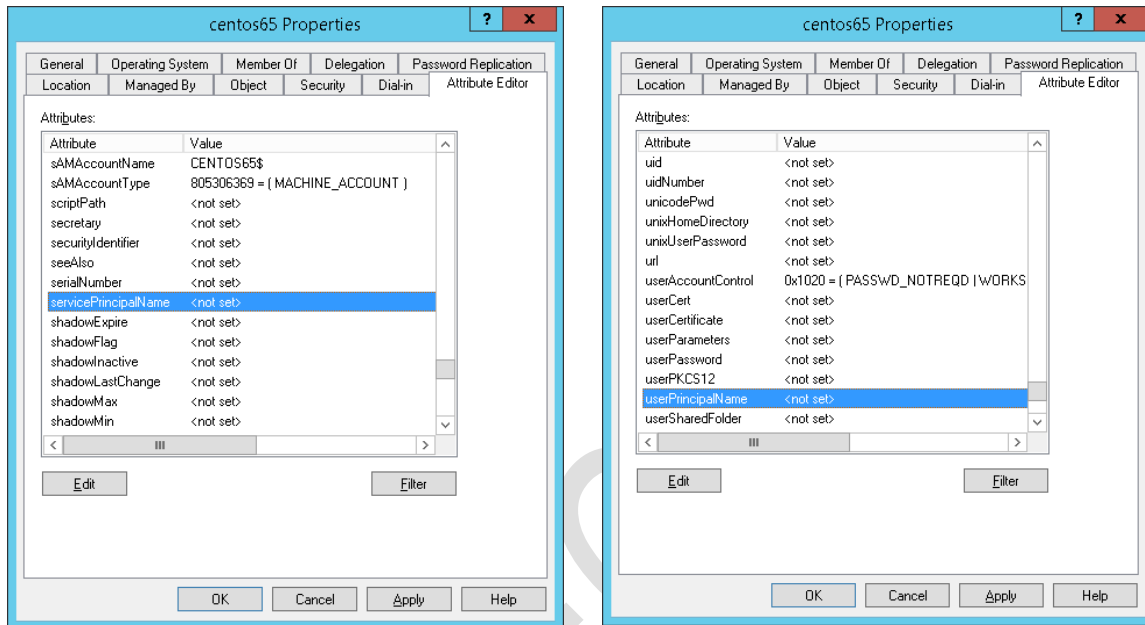
```

```

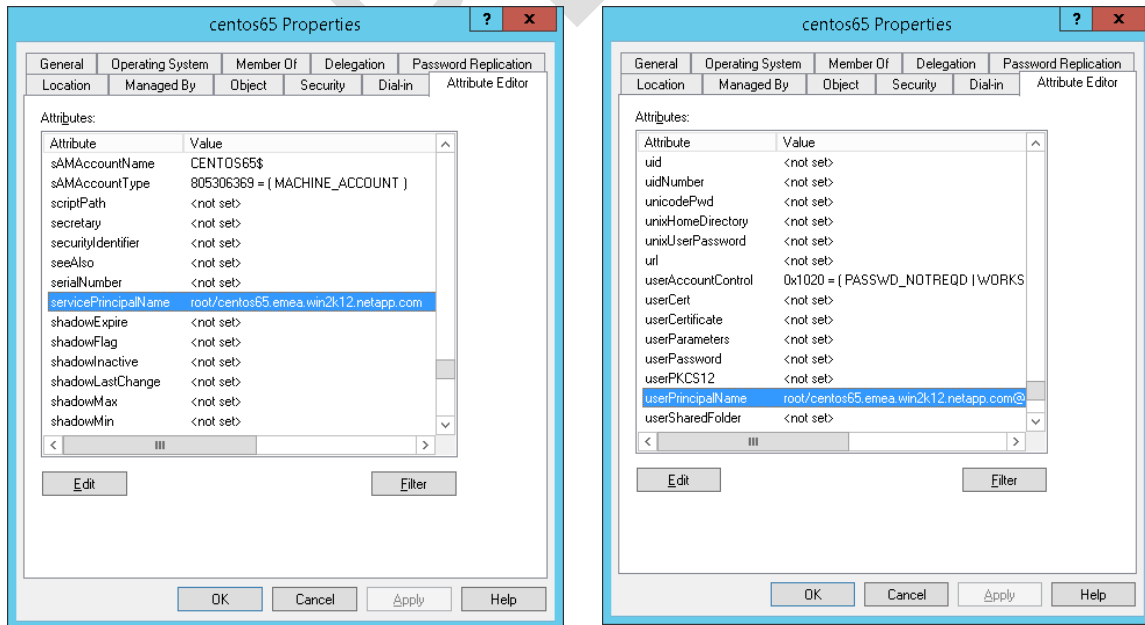
3 05/21/13 10:23:24 root/nfsclient.domain.netapp.com@DOMAIN.NETAPP.COM (arcfour-hmac)
3 05/21/13 10:23:24 root/nfsclient.domain.netapp.com@DOMAIN.NETAPP.COM (aes256-cts-hmac-shal-96)
3 05/21/13 10:23:24 root/nfsclient.domain.netapp.com@DOMAIN.NETAPP.COM (aes128-cts-hmac-shal-96)

```

Machine account SPN and UPN before running ktpass:



Machine account SPN and UPN after running ktpass:



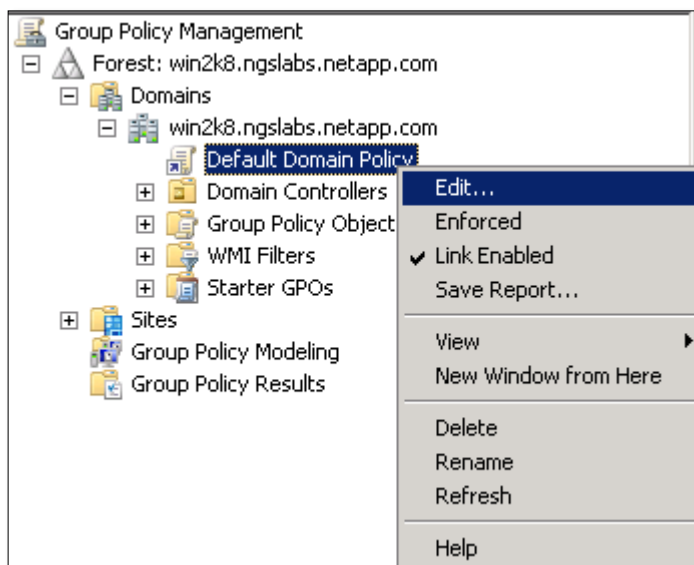
6.3 Kerberos Configuration—Domain Join Method

This section covers the domain join method for Kerberos client configuration. This method avoids the need to create machine accounts (principals) and keytab files manually. For an end-to-end example of this method, see Configuring an NFS Client to Use Kerberos with “realm join”.

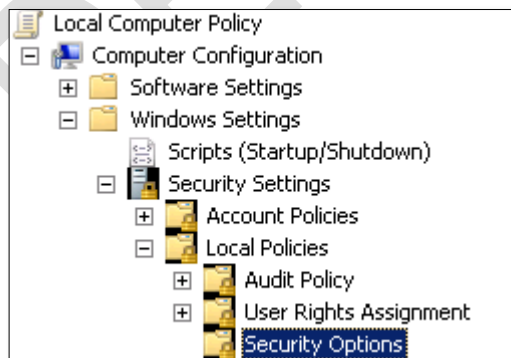
Enabling DES and/or AES in Windows 2008 R2 and Later

Configuration Steps 16) Allowing DES encryption types in Windows 2008 R2 and later.

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 Default Domain Policy and select Edit.



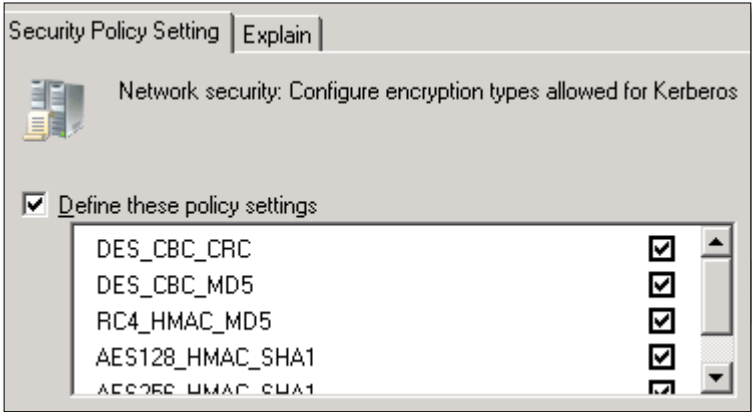
4. Navigate to Computer Configuration -> Policies -> Windows Settings -> Security Settings -> Local Policies -> Security Options.



5. Select Network Security: Configure Encryption Types Allowed for Kerberos and double-click.

Network security: Configure encryption types allowed for Kerberos Not Defined

- Select the encryption types desired. If using a Data ONTAP version before 8.3, DES is needed and the rest of the types are more secure than DES, so select them all. Selecting only DES prevents other encyptes for the domain. In Data ONTAP 8.3 and later, do not select DES encryption types. Use only AES.



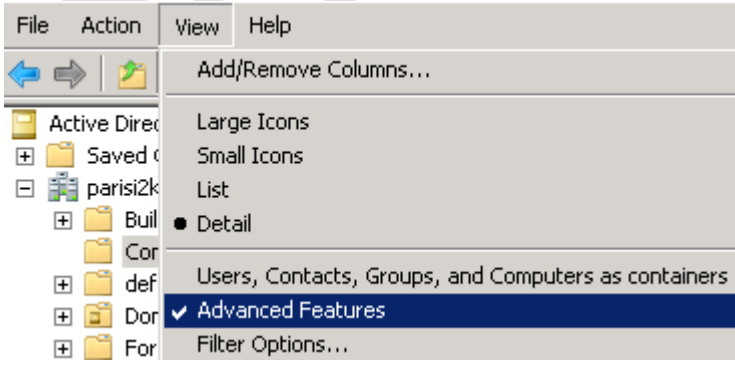
- Click Apply. A reboot is not required to apply the change.

Modifying Machine Accounts in Windows Active Directory

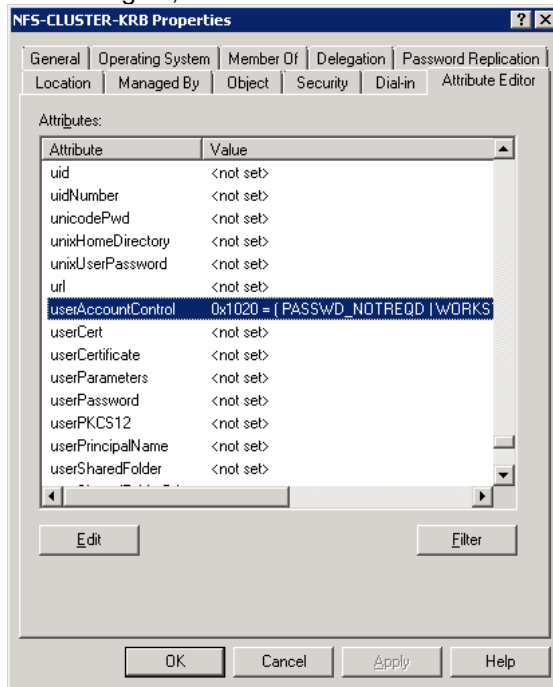
The following tables show how to modify the account using the Attributes Editor tab as well as ADSI Edit and `ldifde` commands. Windows 2008 was used for the DES examples and Windows 2012 was used for the AES examples, but the steps are interchangeable for the operating system versions. Using the Attributes 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.

Configuration Steps 17) Modifying the NFS server machine account for DES_CBC_MD5 (Attributes Editor).

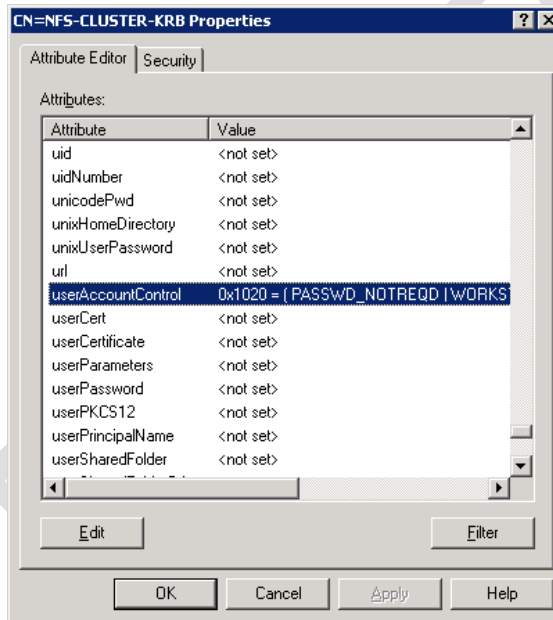
- Log in to the domain controller and open Active Directory Users and Computers.
- Click View and select Advanced Features.



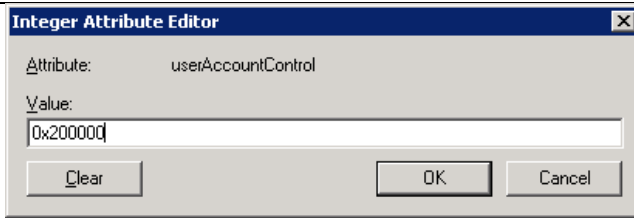
3. After doing so, a new tab called Attributes Editor appears under the machine account properties.



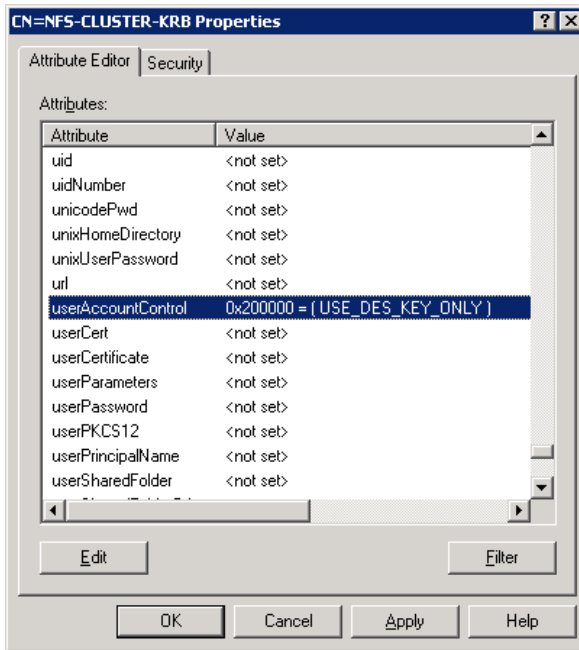
4. Navigate to the userAccountControl attribute.



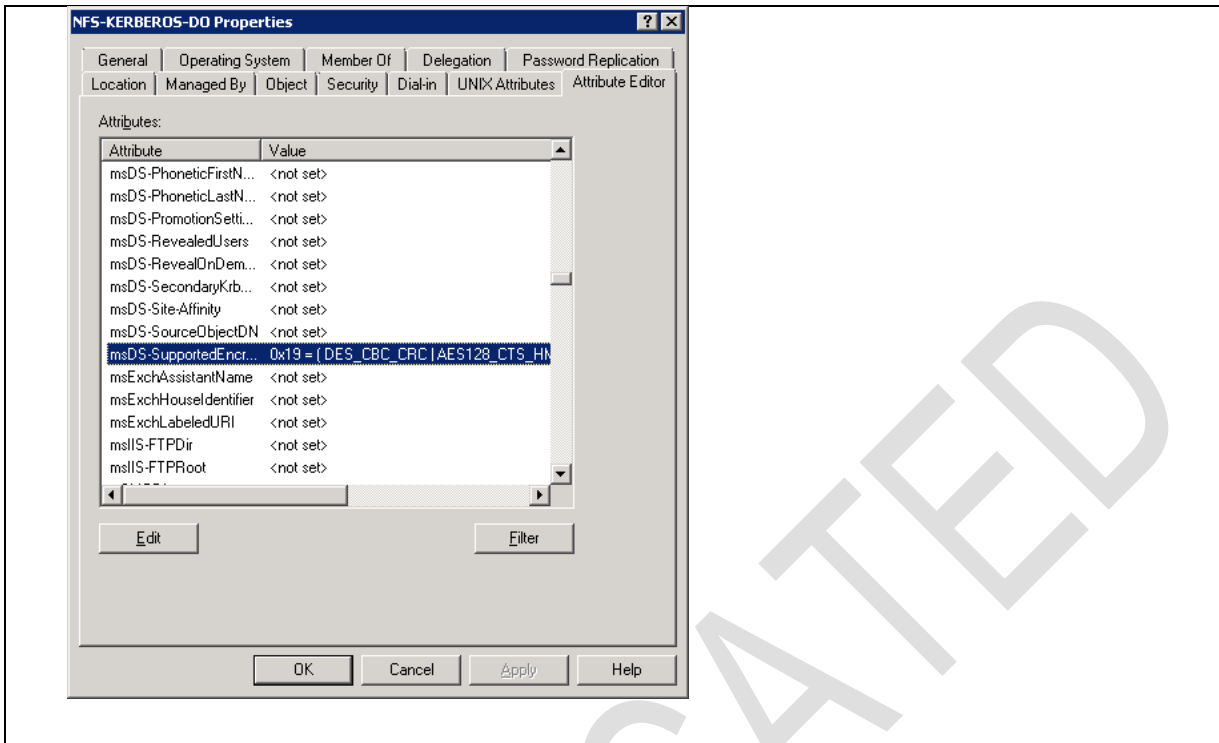
5. Click Edit and change the value to 0x200000 (USE_DES_KEY_ONLY). This action is required only in 8.2.x and earlier, because 8.3 adds support for AES encryption.



6. Click OK and verify the change.

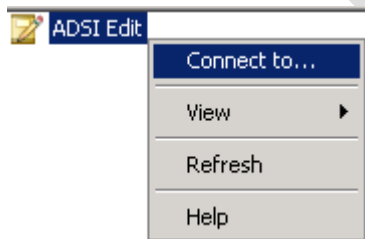


7. In Windows 2008 R2, an attribute called msDS-SupportedEncryptionTypes was added. This option should be set to allow all encyptypes. Change this value to 25 (0x19 in hex) to allow all encryption types for the machine account. (This option did not exist before Windows 2008 R2.)

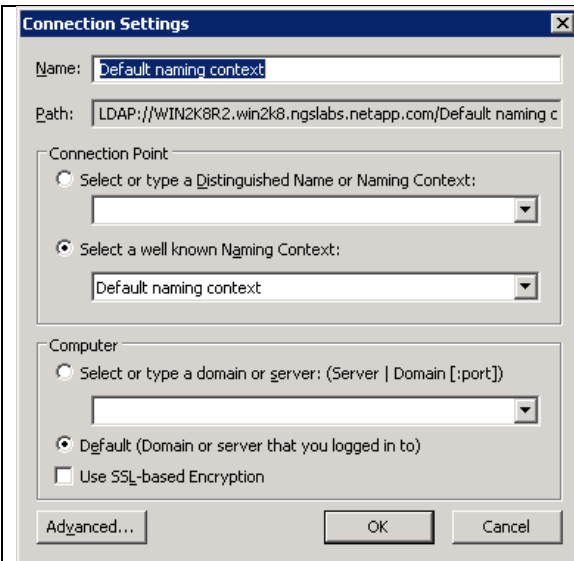


Configuration Steps 18) Modifying the NFS server machine account for DES_CBC_MD5 (ADSI edit).

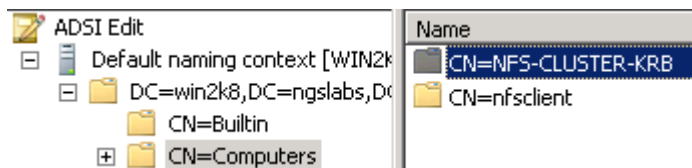
1. Log in to the domain controller; go to Start -> Run and type adsiedit.msc.
2. Connect to the Active Directory database by right-clicking and selecting Connect To.



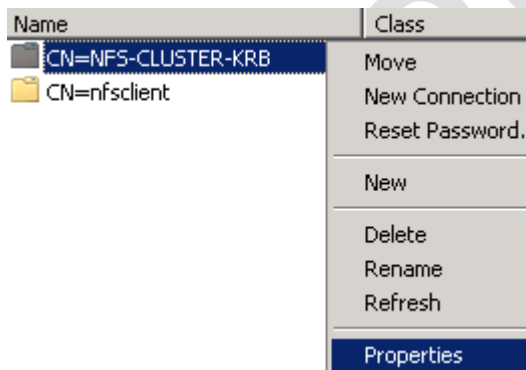
3. Leave the defaults and click OK.



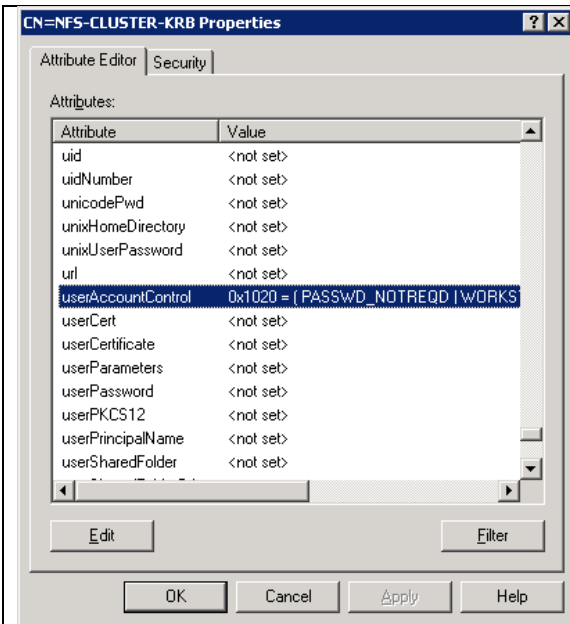
4. Navigate to the Computers container (where the NFS machine account was created).



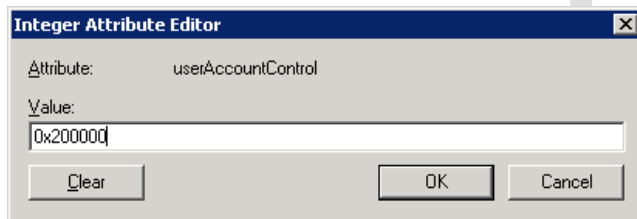
5. Right-click the object and select Properties.



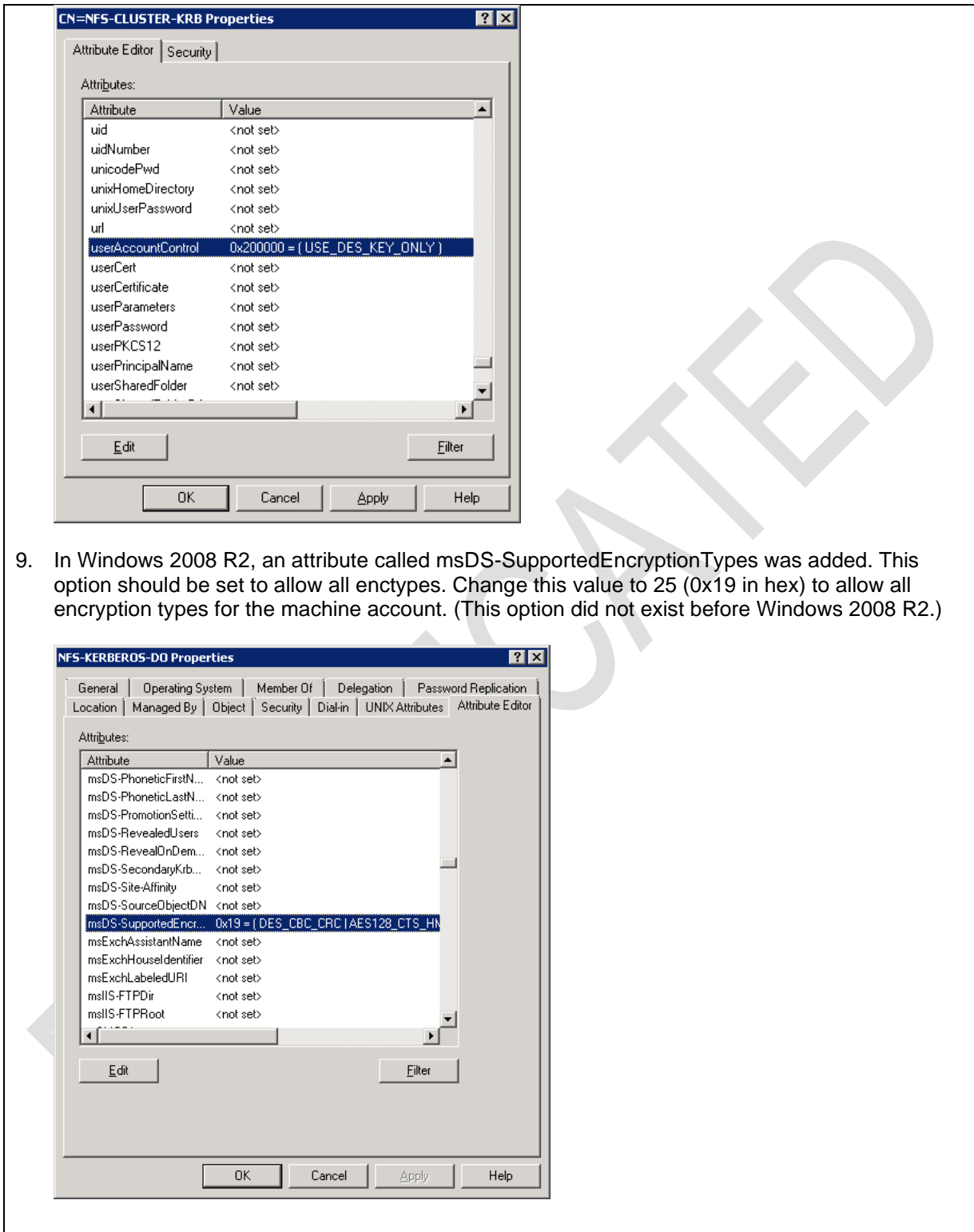
6. Navigate to the userAccountControl attribute.



7. Click Edit and change the value to 0x200000 (USE_DES_KEY_ONLY). This step is required only in 8.2.x and earlier, because 8.3 adds support for AES encryption.



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 encyptes. Change this value to 25 (0x19 in hex) to allow all encryption types for the machine account. (This option did not exist before Windows 2008 R2.)

Configuration Steps 19) Modifying the NFS server machine account for DES_CBC_MD5 (import using ldfidfe).

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: 27
-
```

Note: The preceding 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 following file 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=linux-client))"
-l "msDS-SupportedEncryptionTypes"
```

Configuration Steps 20) Creating machine accounts for use with DES in Active Directory (Windows PowerShell).

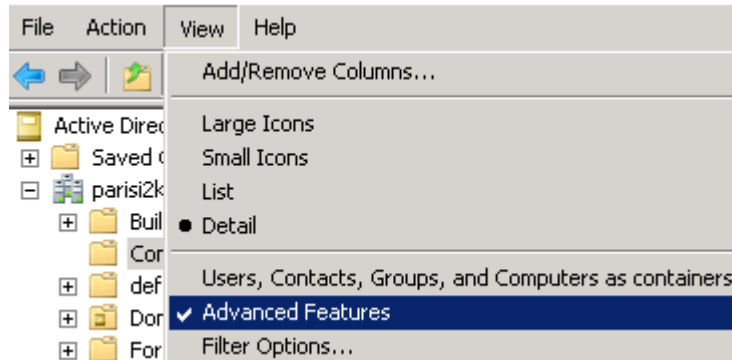
1. Log in to the domain controller and open Windows PowerShell.
2. Type the following ([click for more information on set-ADComputer](#)):

```
PS C:\> import-module activedirectory
PS C:\> Set-ADComputer -Identity [NFSservername] -Replace
@{'userAccountControl'=2097152;'msDS-SupportedEncryptionTypes'=27}
```

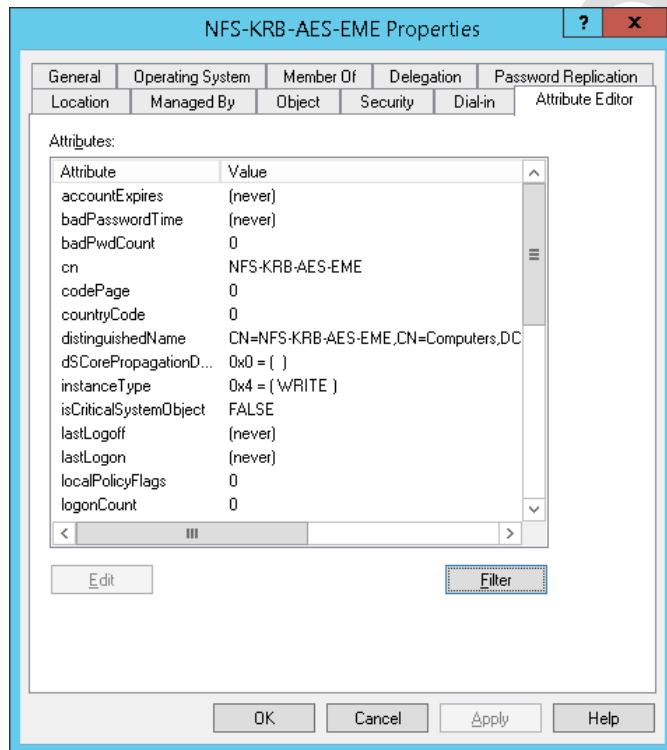
Note: For the NFS client machine account, repeat the preceding steps for *the msDS-SupportedEncryptionTypes* value only.

Configuration Steps 21) Modifying the NFS machine account to use/support AES (Attributes Editor).

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



3. After this is done, a new tab called Attributes Editor appears under the machine account properties.



4. Click Filter and clear the value that says Show Only Attributes That Have Values.

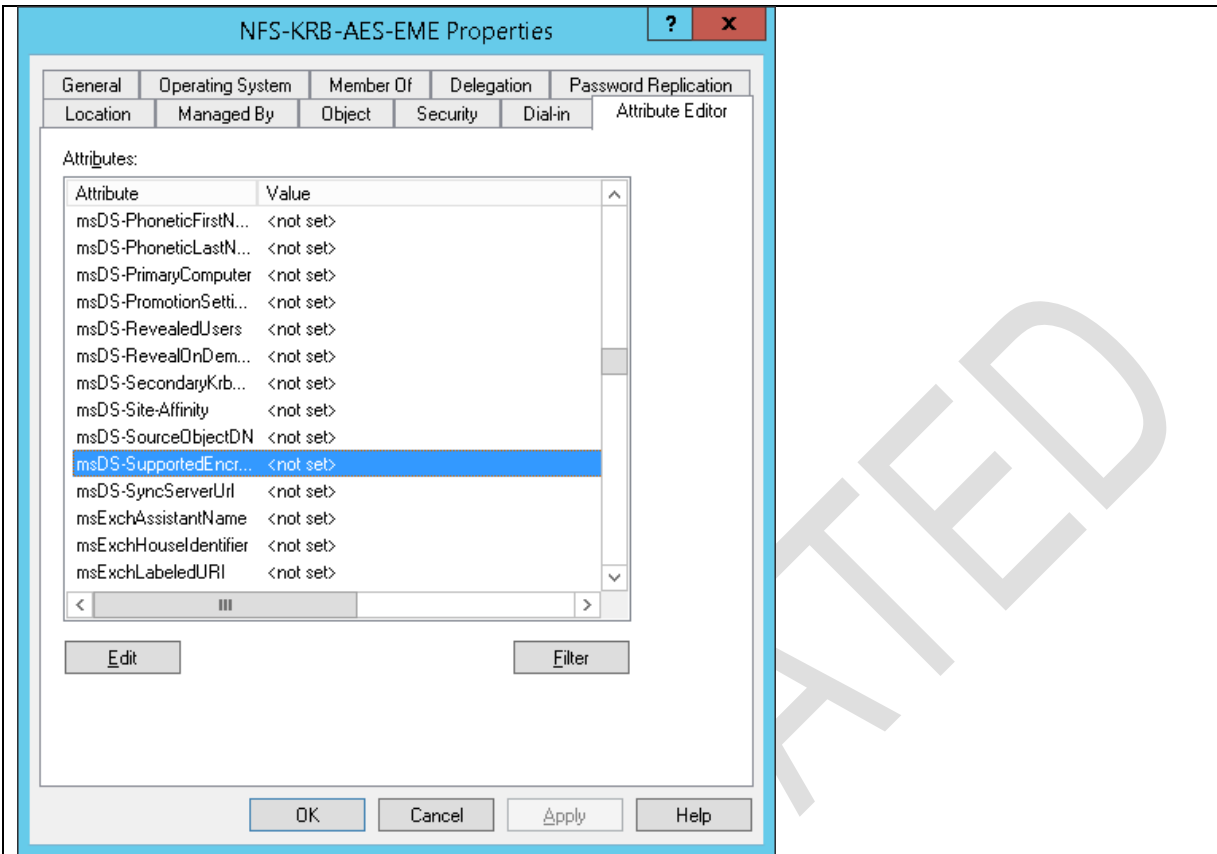
The screenshot shows the 'NFS-KRB-AES-EME Properties' dialog box. The 'Attributes' tab is active, displaying a table of attributes and their values. A 'Filter' dialog is overlaid on the table, showing the following checked options:

- Show only attributes that have values
- Show only writable attributes
- Show attributes:
- Mandatory
- Optional
- Show read-only attributes:
- Constructed
- Backlinks
- System-only

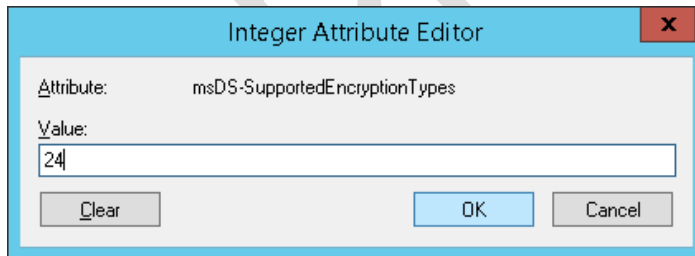
The attribute list in the background includes:

Attribute	Value
accountExpires	(never)
badPasswordTime	(never)
badPwdCount	0
cn	NFS-KRB-AES-EME
codePage	0
countryCode	0
distinguishedName	CN=NFS-KRB-AES-EME,CN=Computers,DC
dSCorePropagationD...	0x0 = ()
instanceType	0x4 = (WRITE)
isCriticalSystemObject	FALSE
lastLogoff	(never)
lastLogon	(never)
localPolicyFlags	0
logonCount	0

5. After this is done, all values are displayed. Navigate to the value msDs-SupportedEncryptionTypes. Modifying UserAccountControl when using AES is not necessary.



6. This value controls which supported encryption types are allowed for the machine account. The table in the appendix regarding [Kerberos property flags](#) shows which values are valid for this. Because AES-256 is 16, AES-128 is 8, DES MD5 is 2, and DES CRC is 1, the total value to allow all 4 is 27 (16 + 8 + 2 + 1). However, it is best to allow only the strongest encryption types for Kerberos. Thus, enable only AES with 24 as the value.



Note: It is important *not* to enable RC4-HMAC on this machine account because the Kerberos requests might attempt to use RC4 regardless of the client configuration. RC4 is not supported in Data ONTAP for NFS Kerberos operations.

For more information about the [userAccountControl](#) and [msDS-SupportedEncryptionTypes](#) values, see the section “[About the Machine Account Attributes](#)” in the appendix of this document. For information on DES, AES, and other encyptes, see the section “[Kerberos Encryption Types](#).”

Configuring Authentication for Kerberos SPNs/Mappings

When a Kerberos request is made to a cluster, ONTAP attempts to authenticate the requesting SPN to a valid UNIX user in the name services. This attempt is made with the new spn-unix name mapping methodology in Data ONTAP. As such, there are several options to control how SPNs map into the cluster.

An example of an SPN that would map into the cluster includes the NFS service SPN on the Kerberos data LIFs (that is, `nfs/fqdn.domain.com`). Since ONTAP uses implicit mappings by default, the system would attempt to look for a UNIX user named “nfs” for that mapping, then move on to name mapping rules. If no valid UNIX users map to that SPN, the Kerberos fails and is shown as “access denied” on the client.

For NFS clients, the SPNs generally are one of the following, depending on the setup and OS version:

```
root/fqdn.domain.com
nfs/fqdn.domain.com
host/fqdn.domain.com
machine$@domain.com
```

UNIX users or spn-unix name mapping rules would need to exist for those SPN values. These users can exist locally (through unix-user create) or in name services such as LDAP or NIS. Additionally, the spn-unix name mapping rules can granularly control which UNIX user maps to the SPNs.

For full details on this point, see the section

Creating and Verifying Name Mapping for the NFS SPN in this document.

Configuration Steps 22) Using local UNIX users for authentication.

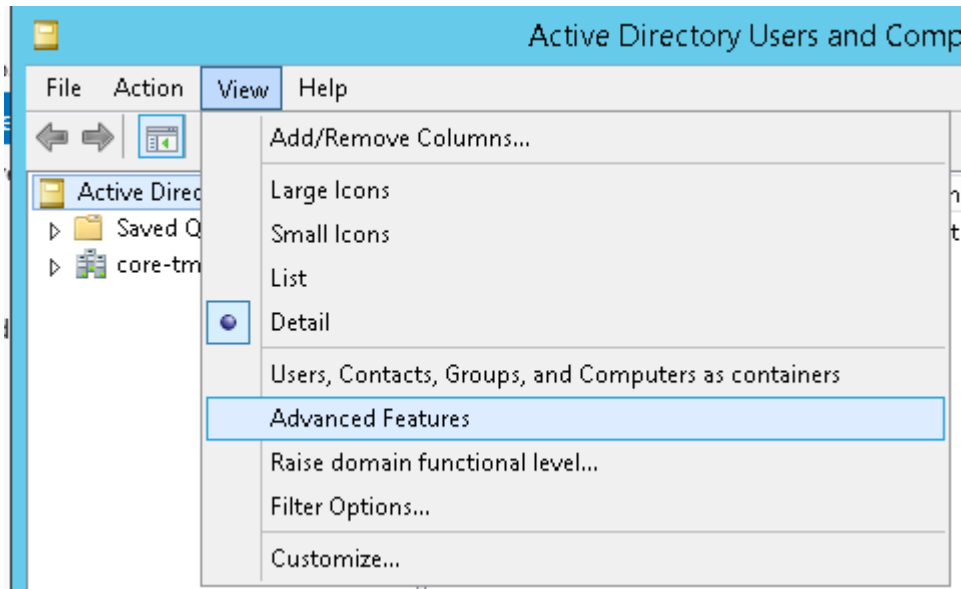
1. Verify the SPNs being used by the data LIFs and NFS clients. On the SVM:
<pre>cluster::> kerberos interface show -fields spn</pre>
On the NFS client:
<pre># ktutil ktutil: rkt /etc/krb5.keytab ktutil: list</pre>
Note: If the SPN in use is “root/fqdn,” then there is no need to create a UNIX user; it exists by default in the SVM.
2. Create the corresponding local UNIX user(s) on the SVM:
<pre>cluster::> unix-user create -vserver SVM -user [user] -id [id] -primary-gid [gid]</pre>
The user “nfs” is almost always needed. In configurations in which the client is joined to a domain, the “machine\$” user would be created.
3. Test the authentication for the SPN.
<pre>::> set diag ::*> diag sec2 name-mapping show -node [node] -vserver [SVM] -direction krb-unix -name [SPN]</pre>
Example:
<pre>::*> diag sec2 name-mapping show -node node3 -vserver SVM -direction krb-unix -name nfs/host.domain.com nfs/host.domain.com maps to nfs</pre>

Configuration Steps 23) Configuring LDAP users for use with authentication.

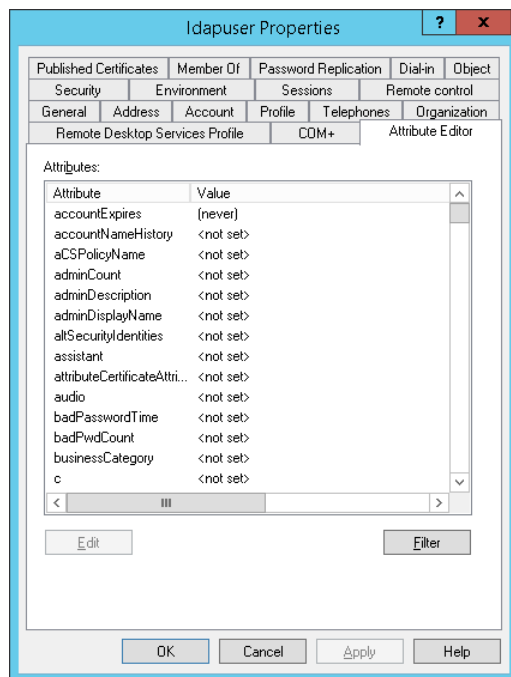
4. Verify the SPNs being used by the data LIFs and NFS clients. On the SVM:
<pre>cluster::> kerberos interface show -fields spn</pre>
On the NFS client:
<pre># ktutil ktutil: rkt /etc/krb5.keytab ktutil: list</pre>
Note: If the SPN in use is “root/fqdn,” then there is no need to create a UNIX user; it exists by default in the SVM.
5. Create the corresponding users in LDAP or configure the existing users in LDAP to have UNIX-style credentials (uidNumber, gidNumber, and so on). The following shows steps in the GUI and steps in PowerShell.

Configuration Steps 24) Using Active Directory Users and Computers to Modify User/Computer Accounts.

1. Set the view to enable "Advanced Features."



- Navigate to your user or computer account and right-click to choose "Properties." Select the Attribute Editor tab.



- Modify the necessary UNIX fields for the user or computer account. The following attributes must be modified:

```
uid (this will be the UNIX username)
uidNumber (numeric ID for UNIX user)
gidNumber (numeric ID for UNIX user - must exist in LDAP)
unixHomeDirectory (home directory path; for example, /home/user)
loginShell (shell for login shell; for example, /bin/sh)
gecos (attribute to define geocos name)
msSFU30Name (UNIX name)
```

```
msSFU30NisDomain (NIS domain; for example, domain.com)
```

Note: The `unixUserPassword` field is not needed for this, but it can be modified using third-party software.

- Test the user lookup in the cluster using `getXXbyYY` (if the cluster was configured for LDAP already):

```
::> set advanced
::*> getXXbyYY getpwbyname -node [node] -vserver [SVM] -username [user] -show-source true
      (vserver services name-service getxxbyyy getpwbyname)
Source used for lookup: LDAP
pw_name: ldapuser
pw_passwd:
pw_uid: 1107
pw_gid: 513
pw_gecos: ldapuser
pw_dir: /home/ldapuser
pw_shell: /bin/sh
```

2. Test the authentication for the SPN.

```
::> set diag
::*> diag secd name-mapping show -node [node] -vserver [SVM] -direction krb-unix -name [SPN]
```

Example:

```
::*> diag secd name-mapping show -node node3 -vserver SVM -direction krb-unix -name
nfs/host.domain.com
nfs/host.domain.com maps to nfs
```

Configuration Steps 25) Using PowerShell to Modify User/Computer Accounts.

Replace the values in [] with your values and run in PowerShell.

```
PS C:\ > Set-ADUser -Identity [user] -Replace @{uid="[user]";uidNumber="[numeric
ID]";gidNumber="[
Numeric GID]";unixHomeDirectory="/home/user";loginShell="/bin/sh";gecos="[name]";
msSFU30Name="[name]"; msSFU30NisDomain="[NIS domain]"}
```

Configuration Steps 26) Configuring krb-unix name mapping rules in the SVM for NFS service principals.

To map NFS server SPNs to different UNIX users than what an implicit name mapping would provide, use `krb-unix` name mappings. For example, if `nfs/fqdn.domain.com` should map to the UNIX user "pcuser" instead of "nfs," use a name mapping rule.

1. Verify the SPNs being used by the data LIFs.

On the SVM:

```
cluster::> kerberos interface show -fields spn
```

2. Check that a valid UNIX user exists either locally or in a name service.

```
::> set diag
::*> diag secd authentication show-ontap-admin-unix-creds -node [node] -vserver [SVM] -unix-
user-name [user]
```

3. Map the SPN to the user with a `krb-unix` name mapping rule.

```
::> vserver name-mapping create -vserver [SVM] -direction krb-unix -position 1 -pattern [SPN] -
replacement [valid UNIX user name]
```

Configuration Steps 27) Configuring krb-unix name mapping rules in the SVM for client principals.

To map NFS client SPNs to different UNIX users than what an implicit name mapping would provide, use `krb-unix` name mappings. For example, if `nfs/fqdn.domain.com` should map to the UNIX user "pcuser" instead of "nfs," use a name mapping rule.

1. Verify the SPNs being used by the NFS clients.

On the SVM:

```
cluster::> kerberos interface show -fields spn
```

On the NFS client:

```
# klist -k
```

2. Map the SPN to the user with a krb-unix name mapping rule. For NFS clients, it makes more sense to create a global name mapping rule for all machine accounts coming in to the cluster, rather than a mapping for each client. Event log show in ONTAP (after errors) during Kerberos mounts/access attempts or packet traces can show you the exact SPN being used for clients attempting Kerberos access.

For clients that were configured for Kerberos using realm join or net ads join:

```
::> vserver name-mapping create -vserver [SVM] -direction krb-unix -position 1 -pattern (.+)\$@REALM.COM -replacement root
```

For clients that were configured for Kerberos manually:

```
::> vserver name-mapping create -vserver [SVM] -direction krb-unix -position 1 -pattern service/fqnd.realm.com@REALM.COM -replacement [name]
```

6.4 LDAP Configuration Steps

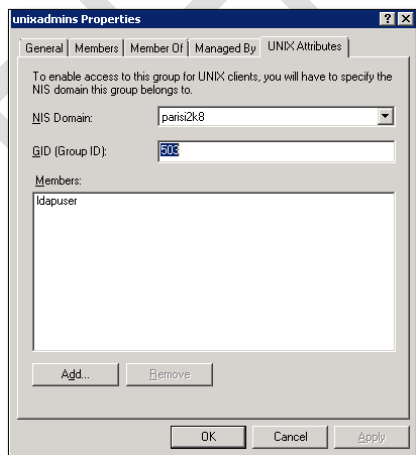
The following section covers LDAP configuration steps.

Changing UID/GID Using the GUI or CLI

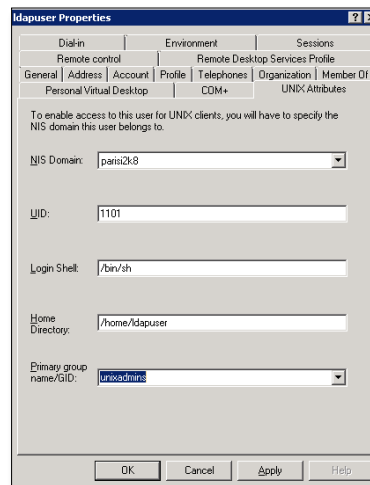
The following screenshots show different actions with UNIX attributes in Active Directory LDAP.

Configuration Steps 28) Setting UID/GID in Active Directory LDAP (GUI).

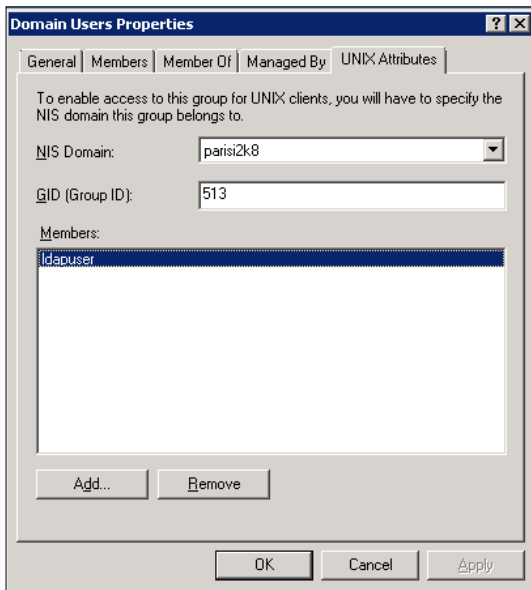
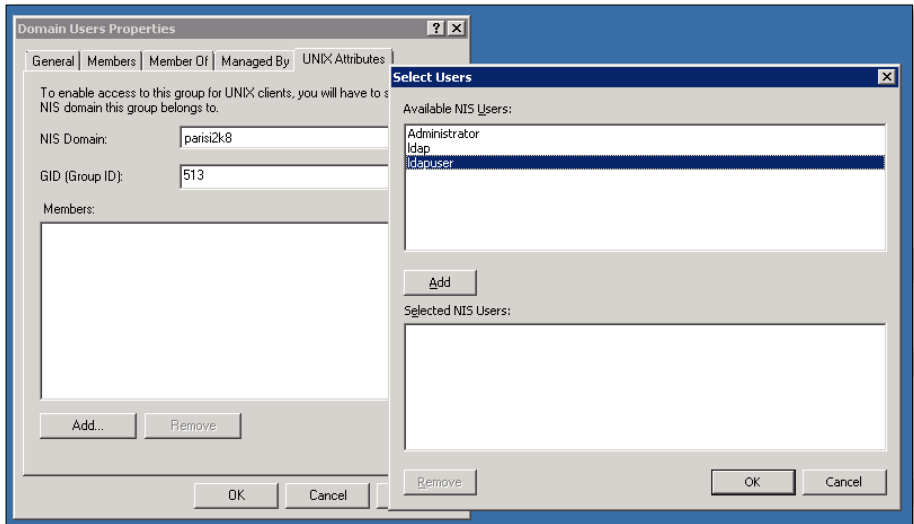
a) Setting a GID on an existing AD group.



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. The following is an example of setting the UID and GID for an object using `ldifde`.

Configuration Steps 29) 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=nfsclient,CN=Users,DC=win2k8,DC=netapp,DC=com
changetype: modify
replace: gidNumber
gidNumber: 513
-
```

Note: The preceding 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 following file 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"
```

Configuration Steps 30) Setting UID/GID in Active Directory LDAP (PowerShell).

UIDs and GIDs can also be set using PowerShell. The following is an example of setting the UID and/or GID for an object using PowerShell.

```
PS C:\Users\Administrator> Get-AdUser -Filter {Uid -eq "ldapuser2"} -Properties uidNumber,gidNumber

DistinguishedName : CN=ldapuser,CN=Users,DC=internaldomaina,DC=local
Enabled           : True
gidNumber         : 1000
GivenName        : ldapuser
Name             : ldapuser
ObjectClass       : user
ObjectGUID        : 8d039512-d3ae-4662-a3bb-8d00296d6f47
SamAccountName    : ldapuser
SID              : S-1-5-21-56907238-3627968364-3018309926-1173
Surname          :
uidNumber        : 1234
UserPrincipalName : ldapuser@internaldomaina.local

PS C:\Users\Administrator> Set-ADUser -Identity ldapuser -Replace @{uidNumber="1107"}
PS C:\Users\Administrator> Get-AdUser -Filter {Uid -eq "ldapuser2"} -Properties uidNumber,gidNumber

DistinguishedName : CN=ldapuser,CN=Users,DC=internaldomaina,DC=local
Enabled           : True
gidNumber         : 1000
GivenName        : ldapuser
Name             : ldapuser
ObjectClass       : user
ObjectGUID        : 8d039512-d3ae-4662-a3bb-8d00296d6f47
SamAccountName    : ldapuser
SID              : S-1-5-21-56907238-3627968364-3018309926-1173
Surname          :
uidNumber        : 1107
UserPrincipalName : ldapuser@internaldomaina.local
```

Configuring User Mapping in LDAP

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

Configuration Steps 31) 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 sAMAccountName. In Data ONTAP 8.3.2 and later, the following new options exist, but in most cases the options do not need to be adjusted.

```
Data ONTAP Name Mapping windowsToUnix Object Class
(-windows-to-unix-object-class)
Data ONTAP Name Mapping windowsToUnix Attribute
(-windows-to-unix-attribute)
No Domain Prefix for windowsToUnix Name Mapping
(-windows-to-unix-no-domain-prefix)
```

Contact your LDAP administrator for assistance to see if these values should be changed. For more information on these new options, see the [section in this document on name mapping in LDAP](#).

Note: Modify is an *advanced privilege* command.

Example:

```
::> set advanced
::*> ldap client schema modify -schema NEW -windows-account-attribute sAMAccountName -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
::*> diag secd name-mapping show -node [SVM] -vserver nfs -direction win-unix -name ldapuser2
ldapuser2 maps to ldapuser2
::*> diag secd name-mapping show -node [SVM] -vserver nfs -direction unix-win -name ldapuser
ldapuser maps to DOMAIN\ldapuser
::*> diag secd name-mapping show -node [SVM] -vserver nfs -direction unix-win -name ldapuser2
ldapuser2 maps to DOMAIN\ldapuser
```

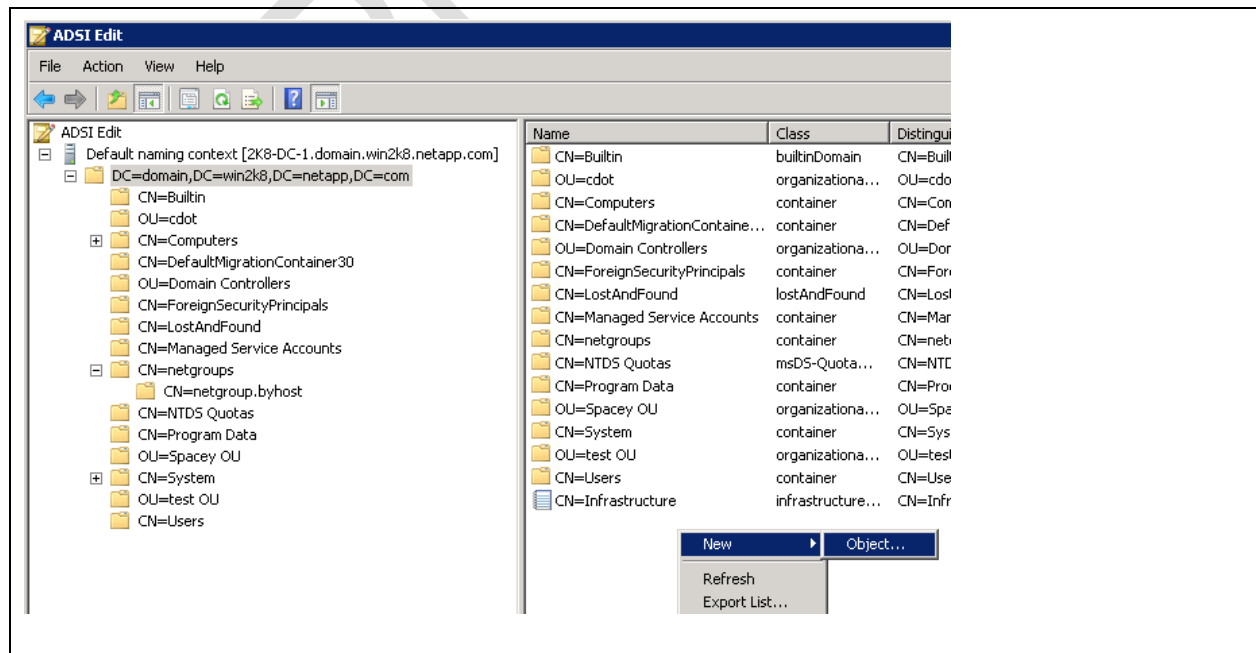
Note: Asymmetric credential fetching for Windows to UNIX name mappings served by LDAP is currently not supported in Data ONTAP without the inclusion of a 1:1 UNIX user name. See the section on [name mapping in LDAP](#) for more information.

Using LDAP for Netgroups

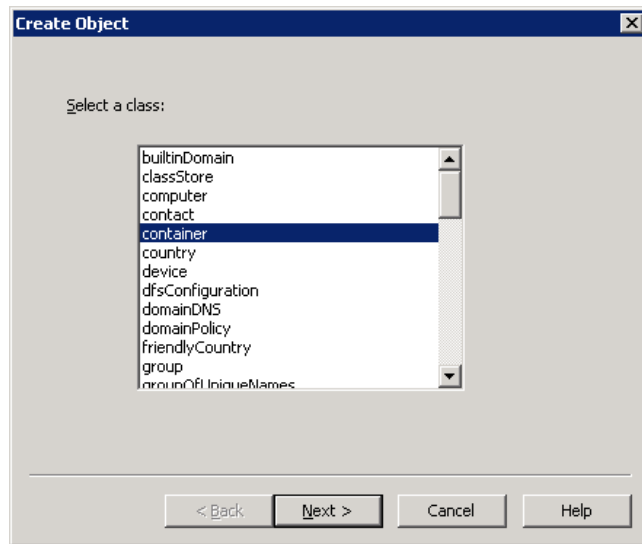
The following configuration steps show you how to create netgroups in Active Directory LDAP servers for use with Data ONTAP.

Creating Netgroups in Active Directory LDAP

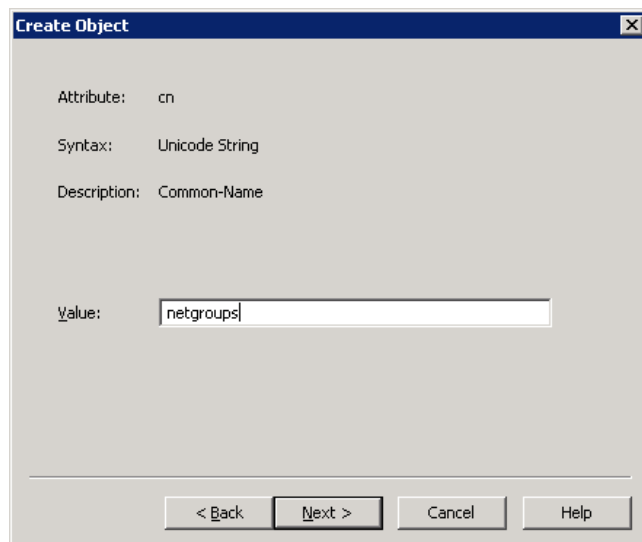
Configuration Steps 32) Creating a container object with ADSI Edit.



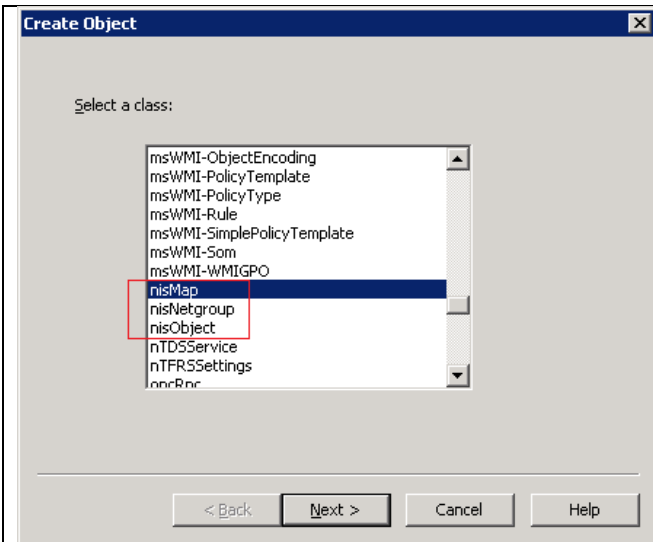
1. Select the container.



2. Give it a name.



3. After the container is created (or if a container already exists), NIS objects can be created in a similar manner. Select the desired container for the objects and create new objects. The NIS object classes are specified using the creation wizard.

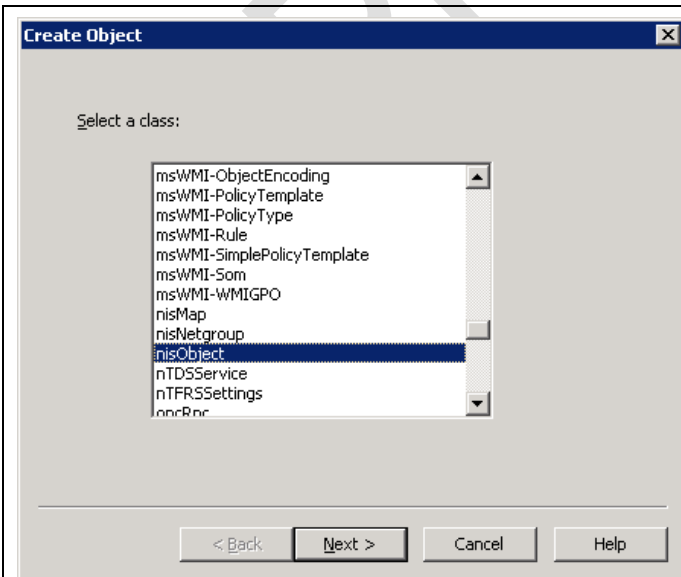


The object class for netgroups depends on which schema the attribute `-nis-netgroup-object-class` is set to in the Data ONTAP SVM's LDAP client configuration. For Windows Active Directory using Identity Management for UNIX (IDMU), that attribute is `nisNetgroup` by default:

```
cluster::> ldap client schema show -schema AD-IDMU -fields nis-netgroup-object-class
(vserver services ldap client schema show)
vserver schema nis-object-class
-----
SVM      AD-IDMU nisObject
```

Therefore, use `nisNetgroup` for netgroups unless you want a custom attribute. For information on customizing LDAP schemas, see the section in this document that covers [creating a custom LDAP schema](#). For `netgroup.byhost` functionality, use the `-nis-object-class` option and the default of `nisObject`. [For more information on netgroup.byhost functionality, see the appropriate section of this document.](#)

Configuration Steps 33) Netgroup entry created using ADSI Edit and `nisObject` class.



Create Object [X]

Attribute: cn

Syntax: Unicode String

Description: Common-Name

Value:

< Back Next > Cancel Help

Create Object [X]

Attribute: nisMapName

Syntax: IA5-String

Description: The attribute contains the name of the map to which the object belongs.

Value:

< Back Next > Cancel Help

Create Object [X]

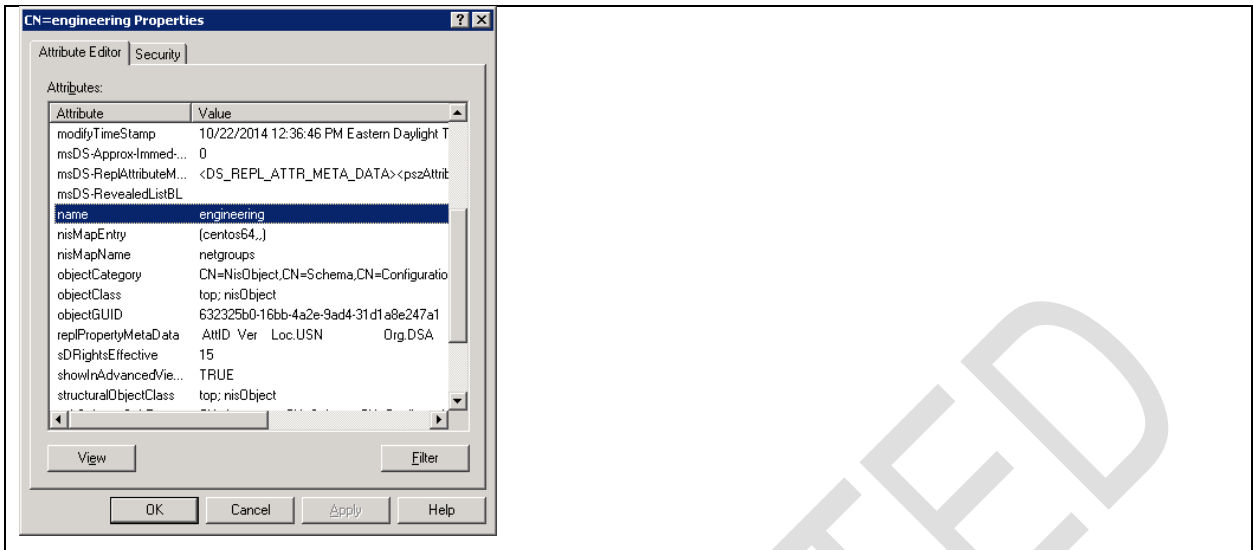
Attribute: nisMapEntry

Syntax: IA5-String

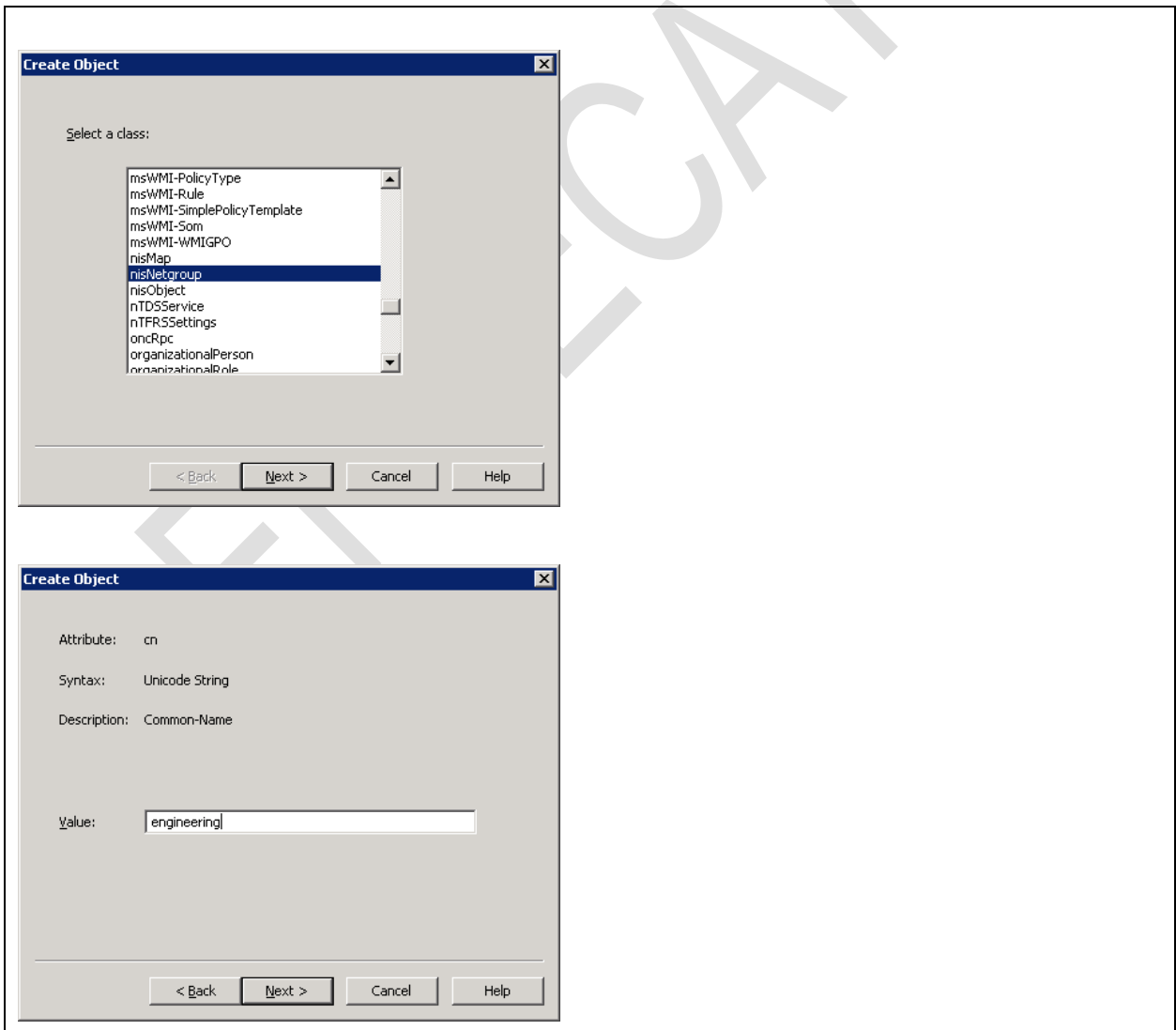
Description: This holds one map entry of a non standard map.

Value:

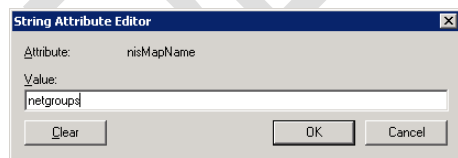
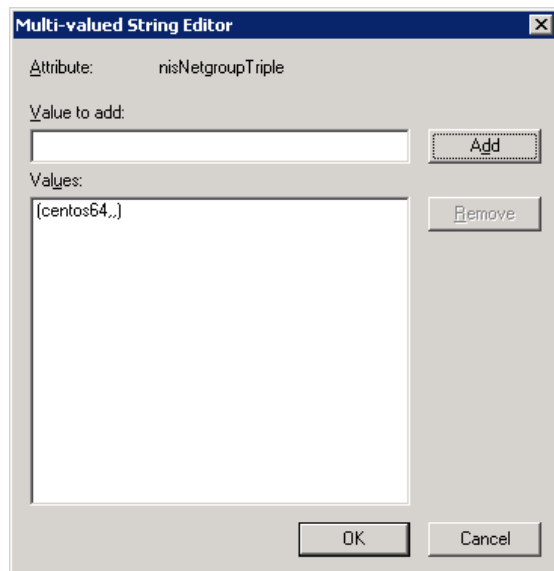
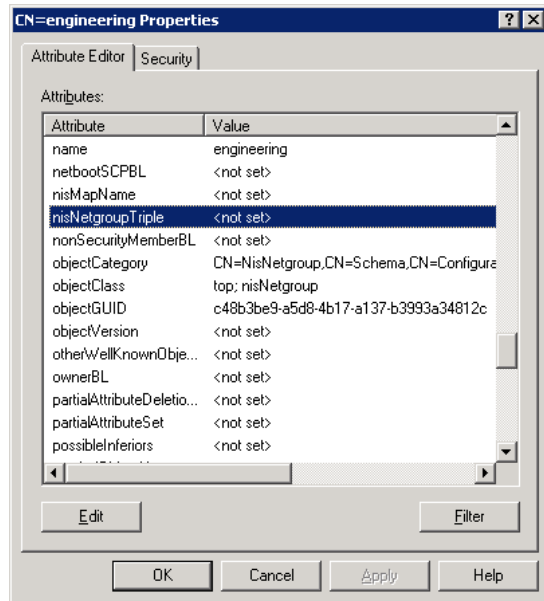
< Back Next > Cancel Help

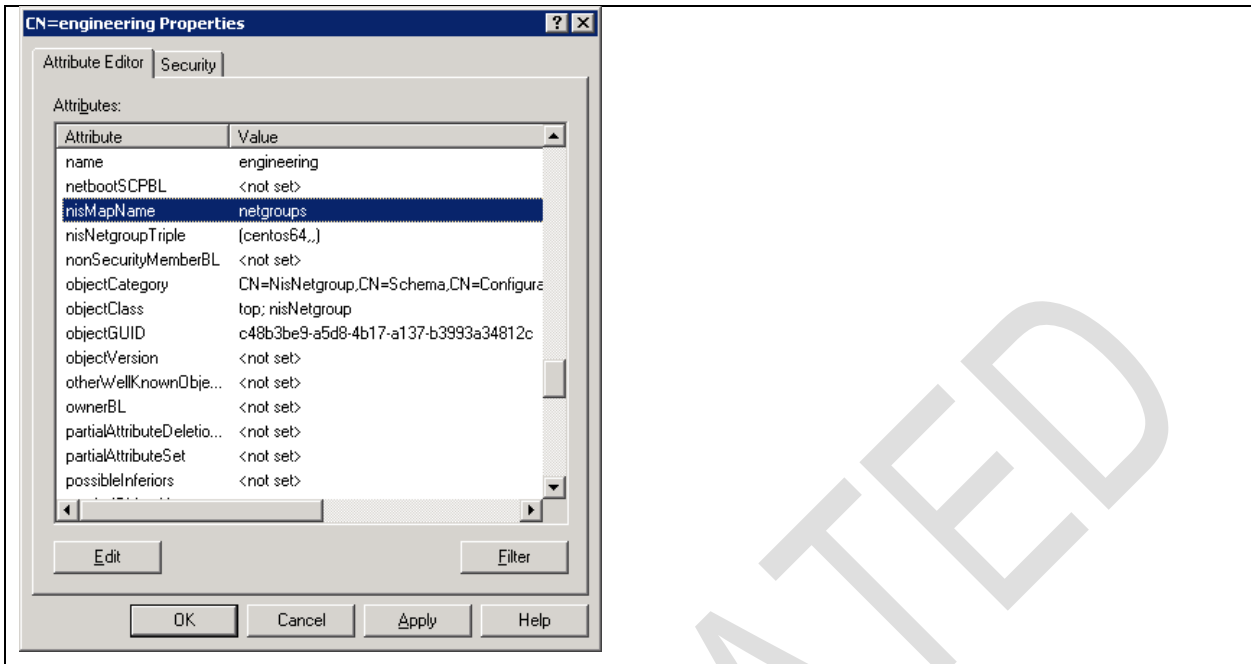


Configuration Steps 34) Netgroup entry created using ADSI Edit and nisNetgroup class.



nisNetgroup object types do not leverage the wizard to populate entries. Instead, edit the attributes in ADSI Edit:





After the entry is created, use Data ONTAP to check that the entry is being queried properly.

To check netgroups in Data ONTAP 8.2.x:

```
cluster::> set diag
cluster::*> diag secd netgroup show-hosts -node node2 -vserver SVM -netgroup-name engineering
centos64
cluster::*> diag secd netgroup show-host-addresses -node node2 -vserver SVM -netgroup-name
engineering
10.228.225.140
cluster::*> diag secd netgroup check-membership -node node2 -vserver SVM -netgroup-name
engineering -address 10.228.225.140 -instance

Node: node2
Vserver: SVM
Netgroup Name: engineering
IP Address: 10.228.225.140
cluster::*> diag secd netgroup show-triples -node node2 -vserver SVM -netgroup-name engineering
(centos64,,)
```

To check netgroups in Data ONTAP 8.3.x and later:

```
cluster::> set advanced
cluster::*> getxxbyyy netgrp -node node1 -vserver SVM -netgroup net
group -client 10.228.225.140 -show-source true -show-granular-err true
(vserver services name-service getxxbyyy netgrp)
Source used for lookup: LDAP
10.228.225.140 is a member of netgroup

NIS:
Error code:      NS_ERROR_NONE
Error message: No error
LDAP:
Error code:      NS_ERROR_NONE
Error message: No error
DNS:
Error code:      NS_ERROR_NONE
Error message: No error
FILES:
Error code:      NS_ERROR_FILES_NOTFOUND
Error message: Files not found
Deterministic Result: Success
```

To check netgroup.byhost in Data ONTAP 8.3.x and later:

```
cluster::*> getxxbyyy netgrpbyhost -node node1 -vserver SVM -netgroup netgroup -clientIP
10.228.225.140 -enable-domain-search-flag true -show-source true
(vserver services name-service getxxbyyy netgrpbyhost)
Success
Hostname resolved to: centos64.win2k8.netapp.com
Source used for lookup: LDAP
```

In addition, you can check export access:

```
cluster::> export-policy check-access -vserver SVM -volume unix -client-ip 10.228.225.140
-authentication-method sys -protocol nfs3 -access-type read-write
(vserver export-policy check-access)
```

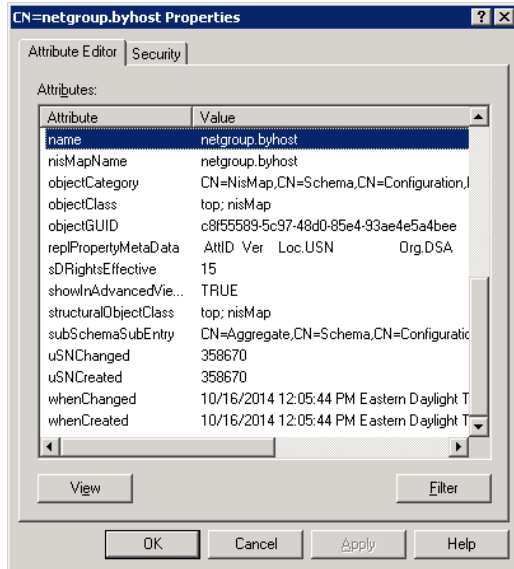
Path	Policy	Policy Owner	Policy Owner Type	Rule Index	Access
/	wideopen	rootvol	volume	1	read
/unix	netgroup	unix	volume	1	read-write

Creating netgroup.byhost Entries in Active Directory LDAP

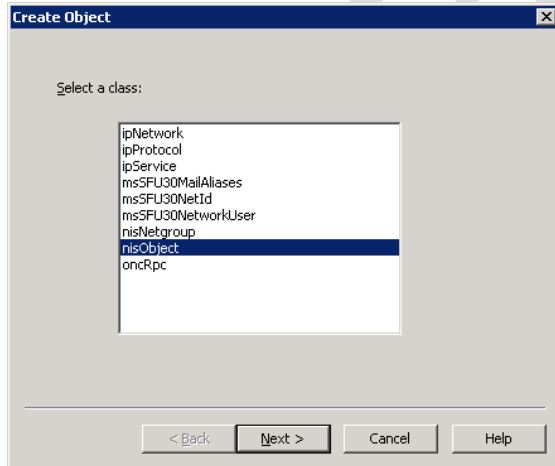
The following is an example of a netgroup.byhost file creation using ADSI Edit and subsequent queries in Data ONTAP. Netgroup.byhost entries can be created on a per-host basis.

Configuration Steps 35) Creating netgroup.byhost entry.

1. First, create a nisMap object to contain the netgroup.byhost entries.



2. Then, create the netgroup.byhost entry using the desired objectClass in that nisMap entry. The default for AD-IDMU is nisObject and is specified in the -nis-object-class field in Data ONTAP LDAP schemas. When specifying the name, be sure to use the FQDN and append .* to the end of the entry to allow lookups to work properly and efficiently.



Create Object [X]

Attribute: cn

Syntax: Unicode String

Description: Common-Name

Value:

< Back Next > Cancel Help

Create Object [X]

Attribute: nisMapName

Syntax: IA5-String

Description: The attribute contains the name of the map to which the object belongs.

Value:

< Back Next > Cancel Help

Create Object [X]

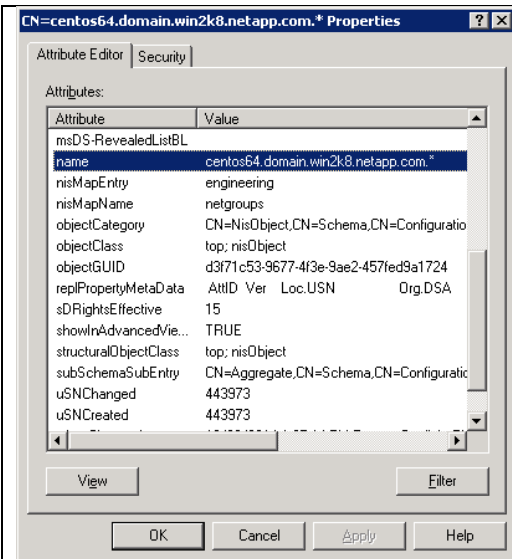
Attribute: nisMapEntry

Syntax: IA5-String

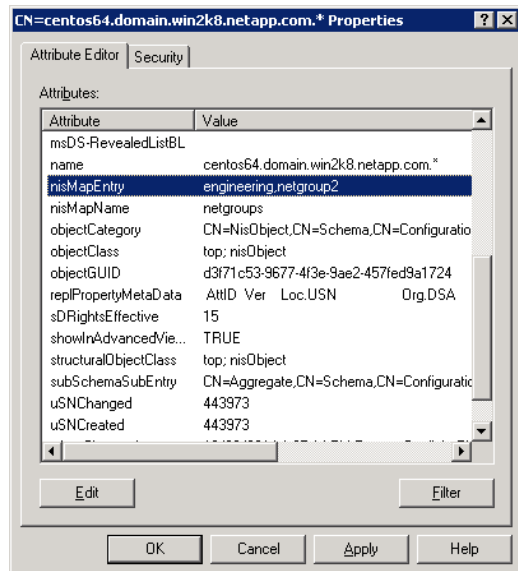
Description: This holds one map entry of a non standard map.

Value:

< Back Next > Cancel Help



To specify multiple netgroups, use comma-separated values.



3. Then test the `netgroup.byhost` lookup in Data ONTAP. By default, the query enables DNS search. To test in 8.2.x and earlier:

```
cluster::> set diag
cluster::*> diag secd netgroup query-netgroup-by-host -node node2 -vserver SVM -netgroup-name
engineering -address 10.228.225.140
Host IP : 10.228.225.140
Hostname : centos64.domain.win2k8.netapp.com
Netgroup : engineering
Member : yes

cluster::*> diag secd netgroup query-netgroup-by-host -node node2 -vserver SVM -netgroup-name
netgroup2 -address 10.228.225.140
Host IP : 10.228.225.140
Hostname : centos64.domain.win2k8.netapp.com
Netgroup : netgroup2
Member : yes
```

To test in 8.3.x and later:

```
cluster::*> getxxbyyy netgrpbyhost -node node1 -vserver SVM -netgroup netgroup -clientIP
10.228.225.140 -enable-domain-search-flag true -show-source true
(vserver services name-service getxxbyyy netgrpbyhost)
Success
Hostname resolved to: centos64.win2k8.netapp.com
Source used for lookup: LDAP
```

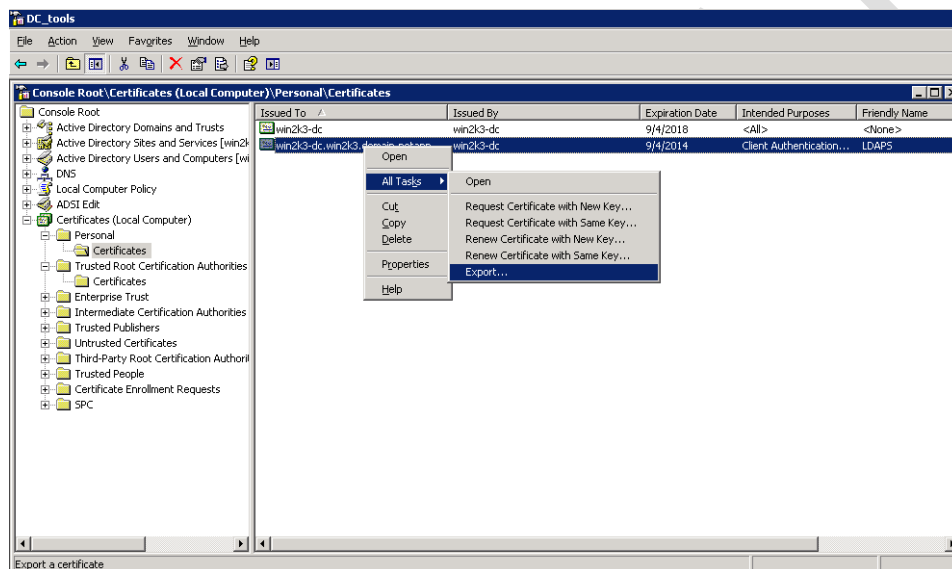
Configuring LDAP over SSL for Use with Active Directory LDAP

Configuration Steps 36) Configuring LDAP over SSL in Data ONTAP.

1. Export the certificate to a CER format (DER encoded) file.

Exporting is performed 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 copying and pasting into Data ONTAP.

To convert the file, open a CLI, navigate to the location to which you exported the certificate, and enter:

```
C:\> certutil -encode <exportedFileName> <PemFileName>
```

This command exports the certificate to text format, which is needed later to import into Data ONTAP.

Sample of resulting file:

```
-----BEGIN CERTIFICATE-----
MIIE6jCCA9KgAwIBAgIQGQUp+NqxoJhOWM+CvfbANjANBgkqhkiG9w0BAQUFADBx
MRMwEQYKCZImiZPyLGQBGRYDY29tMRYwFAFKCZImiZPyLGQBGRYGbmV0YXBwMRYw
FAFKCZImiZPyLGQBGRYGZG9tYWl1MRYwFAFKCZImiZPyLGQBGRYGd2luMmszMRIw
EAYDVQQDEw13aW40Y29tZGMwHhcNMTMwOTA0MTUzMTEwMTEwMTEwMTEwMTEwMTEw
WjBxMRMwEQYKCZImiZPyLGQBGRYDY29tMRYwFAFKCZImiZPyLGQBGRYGbmV0YXBw
MRYwFAFKCZImiZPyLGQBGRYGZG9tYWl1MRYwFAFKCZImiZPyLGQBGRYGd2luMmsz
```

```

MRIwEAYDVQQDEw13aW4yazMtZGMwggEiMA0GCSqGSIB3DQEBAQUAA4IBDwAwggEK
AoIBAQD41T8Blrrum1GML0Dy+dGu1P1ceL+0nkA6vA81xIy2CW/HY18TWd7ZVq5n
IK9z86bKSnMaDKBZ8wpAhYAzkG2yA7AIAqHi3MZjUoty7+C/T/7505bScxFgacKY
IMexOb1iLTVPx3a/jOHzy4a27TEMQig/YAHTOz/CKKBi0/u4/2KKCOKHhoTaUNes
NUIEViZkUwIbNRRDb9LDRvIMWm5zfzaZ2M6PwMkX5/ZwujFgd+GOTMjC4+H78SOA
CalhA1CLR364vGCWMEUWdi51MIDSmZyR5Vpx6dLijWjFUpLcb1Gk2VuFj1jhvOs/
tMd89MIzGEaULjXIgqTqt/BoZDNAGMBAAGjggF8MIIBeDALBgNVHQ8EBAMCAYYw
DwYDVR0TAQH/BAUwAwEB/zAdBgNVHQ4EFgQUrtBPBez/V6bDpN/TSWS/Azm7C7ww
ggElBgNVHR8EggEcMIIBGDCARSgggEQoIIBDIAxBxWxkYXA6Ly8vQ049d2luMmsz
LWRjLENOPXdpbjJrMy1kYyxDTj1DRFAsQ049UHvibG1jJTIwS2V5JTIwU2Yydm1j
ZXMsQ049U2Yydm1jZXMsQ049Q29uZmlndXJhdGlvbixEQz13aW4yazMsREM9ZG9t
YWluLERDPW5ldGFwcCxEQz1jb20/Y2VydG1maWNhdGVsZXZvY2F0aW9uTG1zdD9i
YXNlP29iamVjdENsYXNzPWNSTERpc3RyaWJldGlvb1BvaW50hkJodHRwOi8vd2lu
MmszLWRjLndpbiJrMy5kb21haW4ubmV0YXBwLmNmV39DZXJ0RW5yb2xsL3dpbjJr
My1kYy5jcmwEAYJKwYBBAGCNxUBBAMCAQAwDQYJKoZIhvcNAQEFBQADggEBAOJP
jRgm0x1BipG1bAsEdCzir3PNCBaiM9rcdcmk/NBbACjVoJX8X5uBkqBCfCSNeXSf
EteCFdkLFPxF/tJSuiNcgh5Ae7sSSok7tXRHWmzILKQ1t163AihMXDdA5kWy7m3b
uMLEnv7e1PiEMgzvNm21NEImd1RBJ/Y300aboNdMrrCHLTI+FGxtoeH74TwTg3W
ASgnhYjcIqVh1ifFlY/13dB/+hVzf27VzoczXq6X5S16v9/03ucsX9t8aVcOeQa+
3FBqyaiqNy3E8Jwf/yfyYTZkoohXRjJ9/1FUkoWFxCUL16aVvkc0o0NUTklXwvRDz
wNh+9L85XYXRs6tmIDI=
-----END CERTIFICATE-----

```

3. Set up LDAPS in 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 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 "[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 using LDAPS. DNS queries can also show up; however, those queries 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>
```

Note: Diag secd show-creds works only if a CIFS server is present because of the command's need to fetch both UNIX and Windows credentials.

6.5 NFSv4 Configuration

Configuring NFSv4 in Data ONTAP

Configuration Steps 37) Configuring the 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/or 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. With Windows AD, the DNS domain name is the NFSv4 ID domain. This example also assumes that LDAP queries are working properly with the cluster.

```
cluster::> nfs modify -vserver vs0 -v4-id-domain domain.netapp.com
```

Script samples

For samples of scripts used to configure a cluster for NFS Kerberos, see the following repository on GitHub:

<https://github.com/whyistheinternetbroken/TR-4073-setup>

Appendix

LDAP Terminology

The following list covers common terms that are used when describing LDAP configurations. More detailed and complete definitions are outside the scope of this report and can be found through web searches.

LDAP

Lightweight Directory Access Protocol. A client/server protocol used to manage directory information. LDAP leverages common ports 389, 636 (SSL), 3268 (Global Catalog), and 3269 (Global Catalog over SSL). LDAP servers contain user, group, and other information, including UIDs, GIDs, and user credentials.

LDAP client

LDAP clients are customers of LDAP servers. Clients have specific configurations that are based on what the LDAP server supports.

Schema

An LDAP schema is a container of relevant information within the LDAP architecture. Schemas consist of attribute syntaxes, matching rules, attribute types, object classes, and their subsequent values. These elements make the LDAP environment function as a directory service by providing locations and values for clients to query.

Attribute

LDAP attributes are essentially tags in a schema to help clients quickly look up values. For instance, an attribute in LDAP would be *uidNumber*, whose value would determine the numeric UID of a user.

Value

Values are tied to attributes. Values are what the LDAP query returns once an attribute is located.

UID/UID number

UID is a user identifier. In LDAP, that identifier can be a numeric or a friendly name. In Windows Active Directory LDAP, the UID is the user name (*uid*, *sAMAccountName*, or *name*). User numerics are generally *uidNumber* in Active Directory LDAP.

GID/GID number

GID is a group identifier. In LDAP, that identifier can be a numeric or a friendly name. In Windows Active Directory LDAP, the GID is generally represented by the *CN* or *name* attribute. Group numerics are *gidNumber* in Active Directory LDAP.

Distinguished name (DN)

A distinguished name is a series of relative distinguished names (RDNs) in the format of *attribute=value*. Distinguished names help establish a unique name for an object, as well as provide a way to filter queries to speed them up. A sample DN looks like:

CN=username,OU=users,DC=netapp,DC=com

Organizational unit (OU)

This unit is a type of container in Microsoft Active Directory that holds users, computers, and groups. As an attribute, an OU is usually depicted as *OU=* in distinguished names.

Bind

Basically, a bind is a login to an LDAP server to perform queries. A bind can be encrypted or unencrypted, depending on the server configuration.

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. The negotiation is 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 website:

<http://ait.its.psu.edu/services/identity-access-management/identity/kerberos/encryption-types.html>

Table 21) Enctypes.

Enctype	Cipher Algorithm	Cipher Mode	Key Length	HMAC	Strength
aes256-cts aes256-cts-hmac-sha1-96	AES	CBC+CTS	256 bits	SHA-1 96 bits	Strongest
aes128-cts aes128-cts-hmac-sha1-96	AES	CBC+CTS	128 bits	SHA-1 96 bits	Strong
rc4-hmac	RC4		128 bits	SHA-1 96 bits	Strong
des3-cbc-sha1	3DES	CBC	168 bits	SHA-1 96 bits	Strong
des-cbc-crc	DES	CBC	56 bits	CRC 32 bits	Weak
des-cbc-md5	DES	CBC	56 bits	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 standard replaces DES and 3DES with stronger encryption and longer key lengths.

CBC

Cipher Block Chaining. This method uses the encrypted *cipher text* from the last block of a *block cipher* to further strengthen the next block. Typically the next block's *plain text* is XORed with the *cipher text* of the previous block. This action hides patterns of repeated plain-text blocks.

CRC

Cyclical Redundancy Check. This method validates that data has not been corrupted by trivial medium noise (line noise, hard disk damage, and so on).

CTS

Cipher Text Stealing. This method is 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. This standard is designed to handle only 56-bit key lengths, which causes this type to be a weak encypte.

HMAC

Hash-Based Message Authentication Code. This method simultaneously verifies both the data integrity and the authenticity of a message.

MD5

A Message Digest hashing algorithm. This is an HMAC method.

RC4

This symmetric stream cipher is by Ron Rivest (hence, it is the "Rivest Cipher"). This cipher can handle several key sizes, such as 40-bit and 128-bit keys.

SHA-1

Secure Hash Algorithm. This is an HMAC method.

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 Machine Account Attributes

The attributes [userAccountControl](#) and [msDS-SupportedEncryptionTypes](#) are used to specify how a machine authenticates 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 modifies the value after 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 [Windows Configurations for Kerberos Supported Encryption Type](#). If you use AES, there is no need to modify userAccountControl.

Table 30) 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	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 is 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 website:
<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 account 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 account 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 runs 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 flag represents the password, which should never expire on the account.
- **MNS_LOGON_ACCOUNT** - This account 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_PEAUTH** (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.

The `msDS-SupportedEncryptionTypes` value is set to 27 (hex 0x19). That value translates to allowing only DES and AES encryption types. RC4 is omitted because Data ONTAP does not support RC4-HMAC for NFS Kerberos. The following table shows which values are valid. The value 27 is derived by adding the specified decimal values together for DES-CBC-CRC + DES-CBC-MD5 + AES128 + AES256 (1+2+8+16).

Table 22) Valid msDS-SupportedEncryptionTypes attribute values.

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	16

Renaming NFS Kerberos Machine Accounts in Active Directory

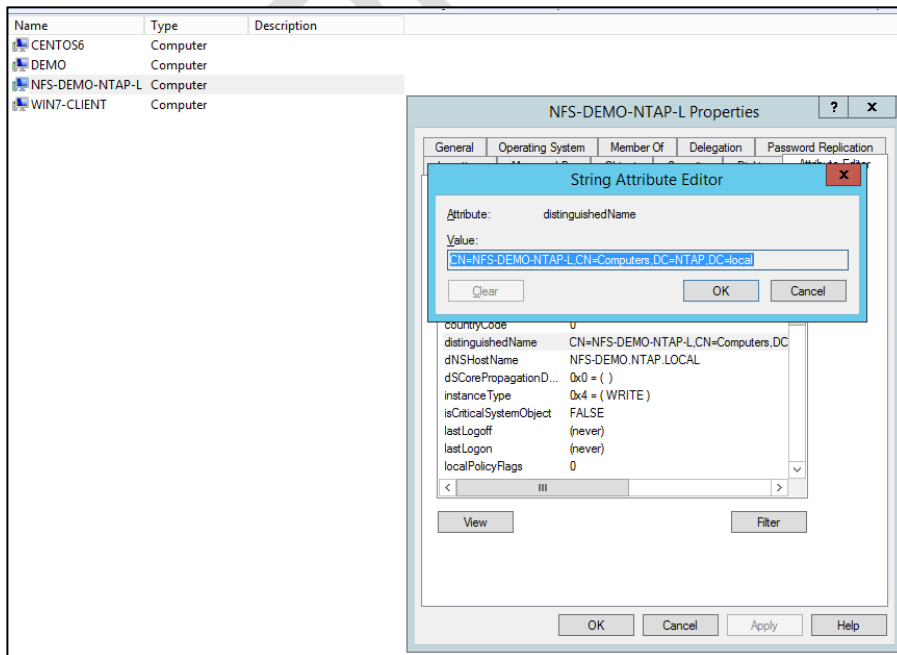
In some cases, the NFS-FQDN-FORMAT of the machine account name is not a preferred name for the Active Directory environment. For instance, some organizations require strict naming schemes for machine accounts. While it is not possible to specify a name for a machine account during its initial creation, it is possible to rename it afterwards without having to remount clients, re-issue tickets, etc.

This is because the display name of the machine account is not critical to the Kerberos operations. What matters in the Kerberos interaction between clients and KDCs are:

- SPNs on the machine account
- DNS hostnames
- Keytab files
- sAMAccountName on the machine account

With Active Directory, changing the display name does not affect any of the above. However, in some cases, Active Directory doesn't allow name changes via the GUI by default. Instead, you must use Powershell. The following steps will guide you through renaming a machine account.





1. First, locate the machine account of the object you wish to rename in Active Directory. Open up the object in AD Users and Computers and find the DN value (you need to [enable Advanced Features](#) to do this). This is the value you need for your Powershell command.

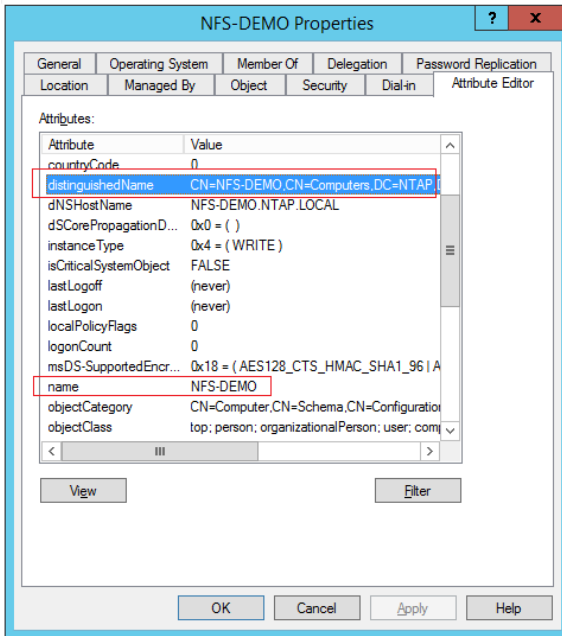


- Open up PowerShell as the domain administrator (or other user with AD rename rights) and run the following command, replacing the objects in brackets with your desired values.

```
PS C:\> Rename-ADObject -Identity ["CN=NAME,CN=Computers,DC=DOMAIN,DC=local"] -NewName [NEW-NAME]
```

- This will change the DN and the "name" value on the computer object, as well as the displayed name in AD Users and Computers.

	CENTOS6	Computer
	DEMO	Computer
	NFS-DEMO	Computer
	WIN7-CLIENT	Computer



- Next, change the attributes for dNSHostName and add a new SPN with the machine account name's FQDN and short name. Use PowerShell's [Set-ADComputer](#) to do this.

```
PS C:\> Set-ADComputer KERBEROS -DNSHostName demo.ntap.local -ServicePrincipalNames
@{Replace="nfs/KERBEROS", "HOST/KERBEROS", "HOST/nfs-demo-ntap-1.ntap.local", "nfs/nfs-demo-ntap-1.ntap.local", "nfs/demo.ntap.local"}
```

5. Next, test your Kerberos access. Everything should still work just fine, as the NFS SPN used by the data LIFs has not changed.

```
[root@centos6 /]# mount home
[root@centos6 /]# mount | grep home
demo:/home on /home type nfs
(rw,hard,intr,sec=krb5,vers=4,addr=10.193.67.219,clientaddr=10.193.67.211)
[root@centos6 /]# su student2
sh-4.1$ klist
klist: No credentials cache found (ticket cache FILE:/tmp/krb5cc_1302)
sh-4.1$ kinit
Password for student2@NTAP.LOCAL:
sh-4.1$ klist -e
Ticket cache: FILE:/tmp/krb5cc_1302
Default principal: student2@NTAP.LOCAL

Valid starting      Expires            Service principal
02/09/17 10:06:31  02/09/17 20:08:24  krbtgt/NTAP.LOCAL@NTAP.LOCAL
        renew until 02/10/17 10:06:31, Etype (skey, tkt): aes256-cts-hmac-sha1-96, aes256-cts-
hmac-sha1-96
sh-4.1$ cd ~
sh-4.1$ pwd
/home/student2
sh-4.1$ klist -e
Ticket cache: FILE:/tmp/krb5cc_1302
Default principal: student2@NTAP.LOCAL

Valid starting      Expires            Service principal
02/09/17 10:06:31  02/09/17 20:08:24  krbtgt/NTAP.LOCAL@NTAP.LOCAL
        renew until 02/10/17 10:06:31, Etype (skey, tkt): aes256-cts-hmac-sha1-96, aes256-cts-
hmac-sha1-96
02/09/17 10:08:35  02/09/17 20:08:24  nfs/demo.ntap.local@NTAP.LOCAL
        renew until 02/10/17 10:06:31, Etype (skey, tkt): aes256-cts-hmac-sha1-96, aes256-cts-
hmac-sha1-96
```

Ktpass Command Syntax

The following section covers the command syntax for ktpass on Windows KDCs.

```
C:\> ktpass /?
Command line options:

-----most useful args
[- /]      out : Keytab to produce
[- /]      princ : Principal name (user@REALM)
[- /]      pass : password to use
              use '*' to prompt for password.
[- +]      rndPass : ... or use +rndPass to generate a random password
[- /]      minPass : minimum length for random password (def:15)
[- /]      maxPass : maximum length for random password (def:256)
-----less useful stuff
[- /]      mapuser : map princ (above) to this user account (default: don't)
[- /]      mapOp : how to set the mapping attribute (default: add it)
[- /]      mapOp : is one of:
[- /]      mapOp :      add : add value (default)
[- /]      mapOp :      set : set value
[- +]      DesOnly : Set account for des-only encryption (default:don't)
[- /]      in : Keytab to read/digest
-----options for key generation
[- /]      crypto : Cryptosystem to use
[- /]      crypto : is one of:
[- /]      crypto : DES-CBC-CRC : for compatibility
[- /]      crypto : DES-CBC-MD5 : for compatibility
[- /]      crypto : RC4-HMAC-NT : default 128-bit encryption
[- /]      crypto : AES256-SHA1 : AES256-CTS-HMAC-SHA1-96
[- /]      crypto : AES128-SHA1 : AES128-CTS-HMAC-SHA1-96
[- /]      crypto : All : All supported types
[- /]      IterCount : Iteration Count used for AES encryption
              Default: ignored for non-AES, 4096 for AES
[- /]      ptype : principal type in question
[- /]      ptype : is one of:
[- /]      ptype : KRB5_NT_PRINCIPAL : The general ptype-- recommended
[- /]      ptype : KRB5_NT_SRV_INST : user service instance
[- /]      ptype : KRB5_NT_SRV_HST : host service instance
[- /]      ptype : KRB5_NT_SRV_XHST :
[- /]      kvno : Override Key Version Number
              Default: query DC for kvno. Use /kvno 1 for Win2K compat.
[- +]      Answer : +Answer answers YES to prompts. -Answer answers NO.
[- /]      Target : Which DC to use. Default:detect
[- /]      RawSalt : raw salt to use when generating key (not needed)
[- +]      DumpSalt : show us the MIT salt being used to generate the key
[- +]      SetUpn : Set the UPN in addition to the SPN. Default DO.
[- +]      SetPass : Set the user's password if supplied.
```

Kerberos Packet Types, Errors, and Terminology

The following tables show the type of Kerberos requests that take place over the wire, as well as which error codes can be returned during requests. This information is intended to help troubleshoot by explaining what each request does.

Table 23) 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. The request 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 24) Kerberos errors from [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 error 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 occurs.

Kerberos Error	What It Means
KRB_AP_ERR_MODIFIED	This error indicates that the service was unable to decrypt the ticket that it was given. A common cause is because 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 error 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 25) 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).
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. This server 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): 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 type of Kerberos server.

Setting Up MIT Kerberos

To configure MIT Kerberos, use the following steps.

Configuration Steps 38) 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 (DES only):

```
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 Data ONTAP 8.2.x and earlier. Data ONTAP 8.3 and later support AES, so those encyptes can be added to the `-e` option as well.

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-shal: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-
shal: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-shal 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-shal-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-shal-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-shal 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 might 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.

If you are looking for errors on the cluster with the CLI, run the following command to search for authentication-specific errors:

```
event log show -message-name secd*
```

Issues When Running Kinit

Kinit is similar to logging in on a Windows box. When kinit is being run, only three components are 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 using 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 capability 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 using 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 using `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 26 shows some common error messages and common causes. The table does not cover all causes and errors.

Table 26) Common mount issues with Kerberized NFS.

Symptom	Cause
Protocol not supported	<ul style="list-style-type: none"> • Machine account password expired (error seen on NFSv4.1 only)
Access denied by server while mounting	<ul style="list-style-type: none"> • 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	<ul style="list-style-type: none"> • 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: 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 27 shows some common error messages and common causes. The table does not cover all causes and errors.

Table 27) Common read/write issues with Kerberized NFS.

Symptom	Cause
Permission denied	<ul style="list-style-type: none"> • 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	<ul style="list-style-type: none"> • Client issue; retry access or remount

For more read/write troubleshooting scenarios, see [TR-4067: NFS Best Practice and Implementation Guide](#).

LDAP Schema Template Examples in Data ONTAP 8.3.2 and Later

For examples of schema templates in previous releases, see the product documentation/man pages.

AD-IDMU Schema

```
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
  Enable Support for Draft RFC 2307bis: false
  RFC 2307bis groupOfUniqueNames Object Class: groupOfUniqueNames
    RFC 2307bis uniqueMember Attribute: uniqueMember
Data ONTAP Name Mapping windowsToUnix Object Class: User
Data ONTAP Name Mapping windowsAccount Attribute: msDS-PrincipalName
Data ONTAP Name Mapping windowsToUnix Attribute: sAMAccountName
No Domain Prefix for windowsToUnix Name Mapping: true
  Vserver Owns Schema: false
Maximum groups supported when RFC 2307bis enabled: 256
  RFC 2307 nisObject Object Class: nisObject
  RFC 2307 nisMapName Attribute: nisMapName
  RFC 2307 nisMapEntry Attribute: nisMapEntry
```

AD-SFU Schema

```
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
  Enable Support for Draft RFC 2307bis: false
  RFC 2307bis groupOfUniqueNames Object Class: groupOfUniqueNames
    RFC 2307bis uniqueMember Attribute: uniqueMember
Data ONTAP Name Mapping windowsToUnix Object Class: User
Data ONTAP Name Mapping windowsAccount Attribute: windowsAccount
Data ONTAP Name Mapping windowsToUnix Attribute: windowsAccount
No Domain Prefix for windowsToUnix Name Mapping: false
  Vserver Owns Schema: false
Maximum groups supported when RFC 2307bis enabled: 256
  RFC 2307 nisObject Object Class: msSFU30NisObject
  RFC 2307 nisMapName Attribute: msSFU30NisMapName
  RFC 2307 nisMapEntry Attribute: msSFU30NisMapEntry
```

AD-MS-BIS

This is a new schema template available in ONTAP 9 for use with RFC-2307bis schemas.

```
Schema Template: MS-AD-BIS
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
  Enable Support for Draft RFC 2307bis: true
  RFC 2307bis groupOfUniqueNames Object Class: group
    RFC 2307bis uniqueMember Attribute: Member
Data ONTAP Name Mapping windowsToUnix Object Class: User
  Data ONTAP Name Mapping windowsAccount Attribute: sAMAccountName
  Data ONTAP Name Mapping windowsToUnix Attribute: sAMAccountName
  No Domain Prefix for windowsToUnix Name Mapping: true
    Vserver Owns Schema: true
Maximum groups supported when RFC 2307bis enabled: 256
  RFC 2307 nisObject Object Class: nisObject
  RFC 2307 nisMapName Attribute: nisMapName
  RFC 2307 nisMapEntry Attribute: nisMapEntry
```

RFC-2307

```
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
  Enable Support for Draft RFC 2307bis: false
  RFC 2307bis groupOfUniqueNames Object Class: groupOfUniqueNames
    RFC 2307bis uniqueMember Attribute: uniqueMember
Data ONTAP Name Mapping windowsToUnix Object Class: posixAccount
  Data ONTAP Name Mapping windowsAccount Attribute: windowsAccount
  Data ONTAP Name Mapping windowsToUnix Attribute: windowsAccount
  No Domain Prefix for windowsToUnix Name Mapping: false
    Vserver Owns Schema: false
Maximum groups supported when RFC 2307bis enabled: 256
  RFC 2307 nisObject Object Class: nisObject
  RFC 2307 nisMapName Attribute: nisMapName
  RFC 2307 nisMapEntry Attribute: nisMapEntry
```

For an example of RFC-2307bis schemas in Microsoft Active Directory LDAP, [see the section in this document on RFC-2307bis](#).

Setting Up Passwordless SSH in Data ONTAP

To run noninteractive SSH commands, password-less SSH must be configured for use with Data ONTAP. This configuration 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 Data ONTAP.

Create the SSH Keypair

In the following example, ssh-keygen is used on a Linux box.

Note: If an 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

```
cluster::> 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
AAAAB3NzaC1yc2EAAAABIWAAAQEAs4vVbwEO1sOsq7r64V5KYBRXBDb2I5mtGmt0+3p1jjPJrXx4/IPHFLalXAQkG7LhV5Dy
c5jyQiGKVawBYwxxSZ3GqXJNv1aORZHJEUcd0zvSTBGGZ09vra5uCFxkxz8nwaTeiAT232LS21Z6RJ4dsCz+GAj2eidpPYMld
i2z6RVoxpZ5Zq68MvNzz8b15BS9T7bvdHkC2OpXFXu2jndhgGxPHvfO2zGwgYv4wv2nQw4tuqMp8e+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.

```
cluster::> 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
AAAAB3NzaC1yc2EAAAABIWAAAQEAs4vVbwEO1sOsq7r64V5KYBRXBDb2I5mtGmt0+3p1jjPJrXx4/IPHFLalXAQkG7LhV5Dy
c5jyQiGKVawBYwxxSZ3GqXJNv1aORZHJEUcd0zvSTBGGZ09vra5uCFxkxz8nwaTeiAT232LS21Z6RJ4dsCz+GAj2eidpPYMld
i2z6RVoxpZ5Zq68MvNzz8b15BS9T7bvdHkC2OpXFXu2jndhgGxPHvfO2zGwgYv4wv2nQw4tuqMp8e+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.
```

Commonly Used Commands and Logs for Troubleshooting NAS Issues

This section covers some commonly used commands in Data ONTAP for troubleshooting NAS issues, as well as what their Data ONTAP operating in 7-Mode counterparts were.

Table 28) 7-Mode to Data ONTAP command translation for authentication.

7-Mode Command	Data ONTAP 8.3.x Command	What It Does
<code>cifs domaininfo</code>	<code>cifs domain discovered-servers show</code> <code>diag secd connections show -node [node] -vserver [SVM] -type ldap-ad</code>	Shows information about the CIFS server's domain.
<code>cifs lookup</code>	<code>diag secd authentication translate</code> <code>diag secd authentication sid-to-uid</code> <code>diag secd authentication sid-to-unix-name</code> <code>diag secd authentication uid-to-sid</code>	Translates CIFS users to SIDs and vice versa.
<code>cifs resetdc</code>	<code>cifs domain discovered-servers reset-servers</code> <code>diag secd connections clear -node [node] -vserver [SVM] -type ldap-ad</code>	Forces reconnection of discovered servers.
<code>cifs testdc</code>	<code>diag secd connections test</code> <code>diag secd server-discovery test</code>	Tests SecD connectivity to name services.
<code>exportfs -c</code>	<code>vserver export-policy check-access</code>	Verifies if a specific host has access to a specific mount and the level of access it has.
<code>exportfs -f</code>	<code>vserver export-policy cache flush</code>	Flushes the exports cache on a local node. Use with caution, because flushing cache causes it to need to be repopulated.
<code>fsecurity show</code>	<code>vserver security file-directory show</code>	Shows the file owner, ACLs, permissions, and so on from Data ONTAP CLI.
<code>getXXbyYY</code>	<code>getXXbyYY</code>	Allows simulation of external name service queries from Data ONTAP CLI (that is, LDAP, NIS).
<code>ifstat</code>	<code>node run <nodename> ifstat</code>	Shows physical network port statistics.
<code>lock status/break</code>	<code>vserver locks show/break</code>	Shows and breaks NFS or CIFS locks on files. Use with caution.
<code>nbtstat</code>	<code>vserver cifs nbtstat</code>	Shows NetBIOS information.

netstat	node run <nodename> netstat network connections active show network connections listening show	Shows network status, port information (listening, open, and so on).
nfs diag show access_cache nsdb_cache	diag exports nblade access-cache show diag nblade credentials show	Shows diagnostic-level NFS information. Use with caution.
nfs nsdb flush	diag nblade nfs nsdb-cache clear	Flushes the nsdb (Name Services Database) cache.
nfs_hist	N/A	Shows nfs histogram.
nfsstat	statistics start -object nfs*/statistics stop	Shows NFS-related statistics.
nis info	nis show-statistics nis show-bound-debug	Shows NIS information.
options nfs.mountd.trace	logger mgwd log modify -node <node> -module mgwd::exports - level debug	Shows trace output of mount requests. Use with caution and always disable after use. Logs to /mroot/etc/mlog/mgwd.log.
options cifs.trace_login	diag secd trace set -node <node> - trace-all yes diag secd log set -node <node> - level debug -enter-exit on	Shows trace output of name mapping, CIFS logins, and so on. Use with caution and always disable after use. Logs to /mroot/etc/mlog/secd.log.
ping	network ping	Runs ping.
pktt	node run [nodename] pktt	Collects a packet capture. For details on packet traces in Data ONTAP, see the following knowledge base article: How to collect packet traces in Data ONTAP
route	network route	Creates/shows/deletes routes.
sectrace	vserver security trace filter	Used to troubleshoot security/permissions issues. For more information, see the following knowledge base article: How to troubleshoot Microsoft Client permission issues on a NetApp Vserver running Data ONTAP
showfh	node run [nodename where file lives] "priv set diag; showfh"	Shows the file handle, FSID, and so on for files and folders.
traceroute	traceroute	Traces routes between devices.

wcc	diag nblade credentials diag secd authentication show- creds	Shows/flushes credential stores on local nodes. Use with caution.
ypcat	N/A	Dumps contents of entire NIS map.
ypmatch	diag secd authentication show- ontap-admin-unix-creds	Fetches NIS object's attributes.
ypwhich	diag secd connections show -node [nodename] -vserver [SVM] -type nis	Shows the NIS map server/master.

The following table shows where you can find logs in Data ONTAP that were originally located in Data ONTAP operating in 7-Mode.

Table 29) 7-Mode to Data ONTAP log translation: NAS-specific logs.

7-Mode Log	Data ONTAP 8.3.x Log/Command	What It Does
/etc/messages	EMS log event log show	Shows relevant errors on the system.
options cifs.trace_login	/mroot/etc/mlog/secd.log	Shows login failures and name mapping results.

In addition, check out the [7-Mode Options Map](#) and [7-Mode Configuration Files Map](#).

The following table shows a list of `diag secd` commands and what they are intended to simulate during troubleshooting, as well as specific use cases.

Table 30) What each SecD command does in Data ONTAP 8.3.x.

Diag secd command	What it does
<code>diag secd authentication login-cifs</code>	Tests logins to CIFS servers. Returns UNIX and CIFS information.
<code>diag secd authentication show-creds</code>	Tests secd for credential lookups/user mappings. Will leverage name services specified by configuration. Requires a CIFS server.
<code>diag secd authentication sid-to-uid</code> <code>diag secd authentication sid-to-unix-name</code> <code>diag secd authentication uid-to-sid</code>	Allows secd to look up a UNIX UID or user name based on a Windows SID and vice versa. Will leverage name services specified by configuration. Requires a CIFS server.
<code>diag secd authentication translate</code>	Allows secd to query specified name services to translate Windows and UNIX users/groups into UIDs, GIDs, and SIDs. Can be used in any NAS environment. Use this command instead of <code>show-creds</code> in NFS-only environments.
<code>diag secd authentication ontap-admin-login-cifs</code>	Tests login of cluster/SVM administrators. Tests Windows authentication only.
<code>diag secd authentication show-ontap-admin-unix-creds</code>	Tests name service lookups of cluster/SVM administrators. Fetches, UID, GID, home directory, login shell, and so on.
<code>diag secd cache clear</code>	Clears specific caches in SecD.
<code>diag secd cache dump</code>	Dumps cache contents to SecD log.
<code>diag secd connections clear show test</code>	Shows, tests, or clears connections to name services such as LDAP, NIS, and Active Directory.
<code>diag secd dns forward-lookup srv-lookup</code>	Tests SecD's ability to do DNS lookups of host names and SRV records.
<code>diag secd log set</code>	Allows setting of debug level logging in SecD.
<code>diag secd name-mapping show</code>	Tests SecD's capability to do a name mapping and returns the value a client would see when attempting to authenticate.
<code>diag secd netgroup</code>	Commands are deprecated in 8.3.x; use <code>getXXbyYY</code> instead.

Configuring an NFS Client to Use Kerberos with “realm join”

This section shows an example of configuring NFS clients to use Kerberos after joining a domain. The NFS client used is RHEL/CentOS 7.2. The `realm` command is used to join the domain. You can find the packages needed to perform these steps in the official Red Hat documentation for [Discovering and Joining Identity Domains](#). The domain is Windows 2012R2 Active Directory. Local UNIX users were used for name mappings.

Configuration Steps 39) Configuring an NFS client to Use Kerberos with “realm join.”

1. Install the necessary packages:

```
yum -y install realmd sssd oddjob oddjob-mkhomedir adcli samba-common krb5-workstation ntp
```

2. Ensure that the DNS on the NFS client is configured to the AD domain and that an A/AAAA record exists in DNS for the Linux client. Test DNS lookups:

```
[root@centos7 ~]# cat /etc/resolv.conf
# Generated by NetworkManager
search core-tme.netapp.com
nameserver 10.193.67.181

[root@centos7 ~]# nslookup centos7
Server:      10.193.67.181
Address:    10.193.67.181#53

Name:   centos7.core-tme.netapp.com
Address: 10.193.67.225
```

3. Ensure that [all firewall rules](#) allow Active Directory connectivity, LDAP, Kerberos, and so on.

4. Discover the AD realm:

```
# realm discover core-tme.netapp.com
core-tme.netapp.com
type: kerberos
realm-name: CORE-TME.NETAPP.COM
domain-name: core-tme.netapp.com
configured: no
server-software: active-directory
client-software: sssd
required-package: oddjob
required-package: oddjob-mkhomedir
required-package: sssd
required-package: adcli
required-package: samba-common
```

5. Join the domain:

```
[root@centos7 ~]# realm join CORE-TME.NETAPP.COM
Password for Administrator:
```

Note: All normal Windows domain rules apply: time skew within 5 minutes, user account with permissions to add computer objects to a domain, DNS able to locate domain controllers. Realm join automatically configures SSSD to a base level and the Kerberos keytab files.

6. Check connectivity to the domain by doing a name lookup (this uses SSSD for LDAP connectivity):

```
[root@centos7 ~]# id CORE-TME\\test
uid=106003697(test@core-tme.netapp.com) gid=106000513(domain users@core-tme.netapp.com)
groups=106000513(domain users@core-tme.netapp.com)
```

Note: The above user created a UID and GID numeric based on an algorithm in SSSD by default to approximate a user and group ID based on the SID. If classic UNIX user attributes are desired, be sure to [configure SSSD](#).

7. Run `kinit` to test Kerberos for a user:

```
[root@centos7 ~]# kinit test@CORE-TME.NETAPP.COM
Password for test@CORE-TME.NETAPP.COM:

[root@centos7 ~]# klist
Ticket cache: KEYRING:persistent:0:0
Default principal: test@CORE-TME.NETAPP.COM

Valid starting      Expires            Service principal
06/29/2016 15:23:54  06/30/2016 01:23:54  krbtgt/CORE-TME.NETAPP.COM@CORE-TME.NETAPP.COM
    renew until 07/06/2016 15:23:50
```

Note: If desired, configure `/etc/krb5.conf` with the realm information to avoid needing to append the realm to `kinit` requests.

Example:

```
[root@centos7 /]# cat /etc/krb5.conf
[logging]
default = FILE:/var/log/krb5libs.log
kdc = FILE:/var/log/krb5kdc.log
admin_server = FILE:/var/log/kadmind.log

[libdefaults]
dns_lookup_realm = true
dns_lookup_kdc = true
default_tkt_enctypes = aes256-cts-hmac-shal-96
default_tgs_enctypes = aes256-cts-hmac-shal-96
ticket_lifetime = 24h
renew_lifetime = 7d
forwardable = true
rdns = false
default_realm = CORE-TME.NETAPP.COM
default_ccache_name = KEYRING:persistent:%{uid}

[realms]
# EXAMPLE.COM = {
#   kdc = kerberos.example.com
#   admin_server = kerberos.example.com
# }
CORE-TME.NETAPP.COM = {
    kdc = dc1.core-tme.netapp.com:88
    admin_server = dc1.core-tme.netapp.com:749
    default_domain = core-tme.netapp.com
}

[domain_realm]
# .example.com = EXAMPLE.COM
# example.com = EXAMPLE.COM
.core-tme.netapp.com = CORE-TME.NETAPP.COM
core-tme.netapp.com = CORE-TME.NETAPP.COM

[root@centos7 ~]# kinit test
```

Password for test@CORE-TME.NETAPP.COM:

Ensure that the [krb5.conf file is configured to allow only specific enctypees](#) or that the [machine account in the domain for the NFS client](#) allows only the desired enctypees. Be sure to disallow RC4-HMAC because Data ONTAP does not support it.

Example of failure when using RC4-HMAC:

```
6/29/2016 16:09:56 ontap-tme-prod-03
WARNING      secd.nfsAuth.problem: vserver (parisi) General NFS authorization problem. Error:
RPC accept GSS token procedure failed
 [ 0 ms] Using the NFS service credential for logical interface 1035 (SPN='nfs/parisi-
nfs.core-tme.netapp.com@CORE-TME.NETAPP.COM') from cache.
**[ 1] FAILURE: Failed to accept the context: Unspecified GSS failure. Minor code may
provide more information (minor: Encryption type ArcFour with HMAC/md5 not permitted).
```

8. When a machine account is added to Active Directory using realm join, the following SPNs are added to the krb5.keytab file automatically:

```
[root@centos7 ~]# ktutil
ktutil: rkt /etc/krb5.keytab
ktutil: list
slot KVNO Principal
```

```
-----
 1  2 host/centos7.core-tme.netapp.com@CORE-TME.NETAPP.COM
 2  2 host/centos7.core-tme.netapp.com@CORE-TME.NETAPP.COM
 3  2 host/centos7.core-tme.netapp.com@CORE-TME.NETAPP.COM
 4  2 host/centos7.core-tme.netapp.com@CORE-TME.NETAPP.COM
 5  2 host/centos7.core-tme.netapp.com@CORE-TME.NETAPP.COM
 6  2 host/centos7@CORE-TME.NETAPP.COM
 7  2 host/centos7@CORE-TME.NETAPP.COM
 8  2 host/centos7@CORE-TME.NETAPP.COM
 9  2 host/centos7@CORE-TME.NETAPP.COM
10  2 host/centos7@CORE-TME.NETAPP.COM
11  2 CENTOS7$@CORE-TME.NETAPP.COM
12  2 CENTOS7$@CORE-TME.NETAPP.COM
13  2 CENTOS7$@CORE-TME.NETAPP.COM
14  2 CENTOS7$@CORE-TME.NETAPP.COM
15  2 CENTOS7$@CORE-TME.NETAPP.COM
```

```
[root@centos7 /]# klist -kte
Keytab name: FILE:/etc/krb5.keytab
KVNO Timestamp Principal
```

```
-----
 2 06/29/2016 15:16:49 host/centos7.core-tme.netapp.com@CORE-TME.NETAPP.COM (des-cbc-crc)
 2 06/29/2016 15:16:49 host/centos7.core-tme.netapp.com@CORE-TME.NETAPP.COM (des-cbc-md5)
 2 06/29/2016 15:16:50 host/centos7.core-tme.netapp.com@CORE-TME.NETAPP.COM (aes128-cts-hmac-
shal-96)
 2 06/29/2016 15:16:50 host/centos7.core-tme.netapp.com@CORE-TME.NETAPP.COM (aes256-cts-hmac-
shal-96)
 2 06/29/2016 15:16:50 host/centos7.core-tme.netapp.com@CORE-TME.NETAPP.COM (arcfour-hmac)
 2 06/29/2016 15:16:50 host/centos7@CORE-TME.NETAPP.COM (des-cbc-crc)
 2 06/29/2016 15:16:50 host/centos7@CORE-TME.NETAPP.COM (des-cbc-md5)
 2 06/29/2016 15:16:50 host/centos7@CORE-TME.NETAPP.COM (aes128-cts-hmac-shal-96)
 2 06/29/2016 15:16:51 host/centos7@CORE-TME.NETAPP.COM (aes256-cts-hmac-shal-96)
 2 06/29/2016 15:16:51 host/centos7@CORE-TME.NETAPP.COM (arcfour-hmac)
 2 06/29/2016 15:16:51 CENTOS7$@CORE-TME.NETAPP.COM (des-cbc-crc)
 2 06/29/2016 15:16:51 CENTOS7$@CORE-TME.NETAPP.COM (des-cbc-md5)
 2 06/29/2016 15:16:51 CENTOS7$@CORE-TME.NETAPP.COM (aes128-cts-hmac-shal-96)
 2 06/29/2016 15:16:52 CENTOS7$@CORE-TME.NETAPP.COM (aes256-cts-hmac-shal-96)
 2 06/29/2016 15:16:52 CENTOS7$@CORE-TME.NETAPP.COM (arcfour-hmac)
```

No other SPNs should be required for the machine account. The client will attempt to get a ticket using the machine account principal (machine\$@REALM.COM).

Because of this, a [KRB to UNIX name mapping must exist](#) for machine\$ either locally on the SVM (name mapping rule or UNIX user) or on the Active Directory object (in the form of a uidNumber/GidNumber attribute in LDAP).

Example:

```
cluster::*> vserver name-mapping show -vserver DEMO -direction krb-unix -position 1

      Vserver: DEMO
      Direction: krb-unix
      Position: 1
      Pattern: (.+)\$@NTAP.LOCAL
      Replacement: root
IP Address with Subnet Mask: -
      Hostname: -
```

Otherwise, the mount request will fail with the following error:

```
6/29/2016 16:28:52  ontap-tme-prod-03
WARNING          secd.nfsAuth.problem: vserver (parisi) General NFS authorization problem. Error:
RPC accept GSS token procedure failed
 [ 0 ms] Using the NFS service credential for logical interface 1035 (SPN='nfs/parisi-
nfs.core-tme.netapp.com@CORE-TME.NETAPP.COM') from cache.
 [ 1] GSS_S_COMPLETE: client = 'CENTOS7$@CORE-TME.NETAPP.COM'
 [ 2] Extracted KG_USAGE_ACCEPTOR_SIGN Derived Key
 [ 2] Extracted KG_USAGE_INITIATOR_SIGN Derived Key
 [ 2] Exported lucid context
 [ 5] Trying to map SPN 'CENTOS7$@CORE-TME.NETAPP.COM' to UNIX user 'CENTOS7$' using
implicit mapping
 [ 6] Entry for user-name: CENTOS7$ not found in the current source: FILES. Ignoring and
trying next available source
 [ 7] Failed to initiate Kerberos authentication. Trying NTLM.
 [ 11] Successfully connected to 10.193.67.181:389 using TCP
**[ 91] FAILURE: User 'CENTOS7$' not found in UNIX authorization source LDAP.
 [ 91] Entry for user-name: CENTOS7$ not found in the current source: LDAP. Entry for user-
name: CENTOS7$ not found in any of the available sources
 [ 91] Unable to map SPN 'CENTOS7$@CORE-TME.NETAPP.COM'
 [ 91] Unable to map Kerberos NFS user 'CENTOS7$@CORE-TME.NETAPP.COM' to appropriate UNIX
user
 [ 91] Failed to accept the context: The routine completed successfully (minor: Unknown
error). Result = 6916
```

The easiest way to resolve this is with the local unix-user:

```
::*> unix-user create -vserver parisi -user CENTOS7$ -id 10001 -primary-gid 1
::~*> unix-user show -vserver parisi -user CENTOS7$
      Vserver: parisi
      User Name: CENTOS7$
      User ID: 10001
Primary Group ID: 1
User's Full Name:
```

9. Test the krb-unix mapping:

```
::> set diag
::~*> diag secd name-mapping show -node ontap-tme-prod-03 -vserver parisi -direction krb-unix -
name CENTOS7$@CORE-TME.NETAPP.COM
CENTOS7$@CORE-TME.NETAPP.COM maps to CENTOS7$
```

10. Ensure that the NFS unix-user or equivalent name mapping is in place so that the service account (nfs/fqdn@REALM) can authenticate.

```
::*> unix-user create -vserver parisi -user nfs -id 10002 -primary-gid 1
::*> unix-user show -vserver parisi -user nfs
      Vserver: parisi
      User Name: nfs
      User ID: 10002
Primary Group ID: 1
User's Full Name:
```

11. Attempt to mount the SVM data interfaces with Kerberos. The SVM must already have the following created and configured:

- Kerberos realm
- Kerberos interfaces
- DNS A/AAAA records in the DNS server (forward and reverse)
- Permitted encyptes for Kerberos
- Export policy rules on the NFS exports and parent directories that allow Kerberos

All of the above are covered in the section called "[Configuring the ONTAP System for Kerberos](#)" in this document.

Mount example:

```
[root@centos7 /]# mount -o sec=krb5 parisi-nfs:/nfs /kerberos
[root@centos7 /]#
```

12. su as a different user and kinit. CD into the mount and check for your NFS service ticket:

```
[root@centos7 /]# su test@CORE-TME.NETAPP.COM
[test@core-tme.netapp.com@centos7 /]$ kinit test@CORE-TME.NETAPP.COM
Password for test@CORE-TME.NETAPP.COM:
[test@core-tme.netapp.com@centos7 /]$ klist
Ticket cache: KEYRING:persistent:106003697:106003697
Default principal: test@CORE-TME.NETAPP.COM

Valid starting    Expires          Service principal
06/29/2016 16:38:26 06/30/2016 02:38:26  krbtgt/CORE-TME.NETAPP.COM@CORE-TME.NETAPP.COM
    renew until 07/06/2016 16:38:21

[test@core-tme.netapp.com@centos7 /]$ mount | grep kerberos
parisi-nfs:/nfs on /kerberos type nfs4
(rw,relatime,vers=4.0,rsize=65536,wsiz=65536,namlen=255,hard,proto=tcp,port=0,timeo=600,retran
s=2,sec=krb5,clientaddr=10.193.67.225,local_lock=none,addr=10.193.67.226)

[test@core-tme.netapp.com@centos7 /]$ cd /kerberos

[test@core-tme.netapp.com@centos7 /kerberos]$ klist -e
Ticket cache: KEYRING:persistent:106003697:106003697
Default principal: test@CORE-TME.NETAPP.COM

Valid starting    Expires          Service principal
06/29/2016 16:38:43 06/30/2016 02:38:26  nfs/parisi-nfs.core-tme.netapp.com@CORE-
TME.NETAPP.COM
    renew until 07/06/2016 16:38:21, Etype (skey, tkt): aes256-cts-hmac-sha1-96, aes256-
cts-hmac-sha1-96
06/29/2016 16:38:26 06/30/2016 02:38:26  krbtgt/CORE-TME.NETAPP.COM@CORE-TME.NETAPP.COM
    renew until 07/06/2016 16:38:21, Etype (skey, tkt): aes256-cts-hmac-sha1-96, aes256-
cts-hmac-sha1-96
```


Configuring an NFS Client to Use Kerberos with “net ads join”

This section shows an example of configuring NFS clients to use Kerberos after joining a domain using “net ads join.” Net ads commands can be found with the samba and winbind packages.

The NFS client used is RHEL/CentOS 7.2. The net ads command is used to join the domain. The domain is Windows 2012R2 Active Directory. Local UNIX users were used for name mappings.

Configuration Steps 40) Configuring an NFS client to Use Kerberos with “net ads join”

1. Install the necessary packages:

```
# yum install -y samba samba-winbind samba-winbind-clients ntp authconfig-gtk*
```

2. Check the time on the client and domain to ensure that you are within 5 minutes. Doing so also verifies that the client can find the domain controller:

```
# net time -S CORE-TME.NETAPP.COM
Mon Jul 11 16:08:00 2016
```

```
# date
Mon Jul 11 16:08:46 EDT 2016
```

Set up ntp. If necessary, sync the time manually:

```
# net time set -S CORE-TME.NETAPP.COM
```

3. Ensure that the client is in the same DNS that Active Directory uses and that nslookup for the client and the domain controllers work.

```
# nslookup centos7
Server:      10.193.67.181
Address:     10.193.67.181#53

Name:   centos7.core-tme.netapp.com
Address: 10.193.67.225
Name:   centos7.core-tme.netapp.com
Address: 192.168.122.1
```

```
# nslookup core-tme.netapp.com
Server:      10.193.67.181
Address:     10.193.67.181#53

Name:   core-tme.netapp.com
Address: 10.193.67.200
Name:   core-tme.netapp.com
Address: 10.193.67.181
```

4. Modify the /etc/krb5.conf file to reflect the Active Directory domain:

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

[libdefaults]
dns_lookup_realm = true
dns_lookup_kdc = true
default_tkt_enctypes = aes256-cts-hmac-shal-96
default_tgs_enctypes = aes256-cts-hmac-shal-96
ticket_lifetime = 24h
```

```

renew_lifetime = 7d
forwardable = true
rdns = false
default_realm = CORE-TME.NETAPP.COM
default_ccache_name = KEYRING:persistent:%{uid}

[realms]
# EXAMPLE.COM = {
#   kdc = kerberos.example.com
#   admin_server = kerberos.example.com
# }
CORE-TME.NETAPP.COM = {
    kdc = dc1.core-tme.netapp.com:88
    admin_server = dc1.core-tme.netapp.com:749
    default_domain = core-tme.netapp.com
}

[domain_realm]
# .example.com = EXAMPLE.COM
# example.com = EXAMPLE.COM
.core-tme.netapp.com = CORE-TME.NETAPP.COM
core-tme.netapp.com = CORE-TME.NETAPP.COM

```

Ensure that the [krb5.conf file is configured to allow only specific enctypees](#) or that the [machine account in the domain for the NFS client](#) allows only the desired enctypees. Be sure to disallow RC4-HMAC because Data ONTAP does not support it.

Example of failure when using RC4-HMAC:

```

6/29/2016 16:09:56  ontap-tme-prod-03
WARNING          secd.nfsAuth.problem: vserver (parisi) General NFS authorization problem. Error:
RPC accept GSS token procedure failed
 [ 0 ms] Using the NFS service credential for logical interface 1035 (SPN='nfs/parisi-
nfs.core-tme.netapp.com@CORE-TME.NETAPP.COM') from cache.
**[ 1] FAILURE: Failed to accept the context: Unspecified GSS failure. Minor code may
provide more information (minor: Encryption type ArcFour with HMAC/md5 not permitted).

```

5. Configure `/etc/samba/smb.conf` with the domain information:

```

[global]

workgroup = CORE-TME
password server = stme-infra02.core-tme.netapp.com:88
realm = CORE-TME.NETAPP.COM
security = ads
idmap uid = 16777216-33554431
idmap gid = 16777216-33554431
template shell = /bin/bash
winbind use default domain = false
winbind offline logon = true

log file = /var/log/samba/log.%m
max log size = 50

passdb backend = tdbsam

load printers = yes
cups options = raw

[homes]
comment = Home Directories
browseable = no
writable = yes

[printers]

```

```
comment = All Printers
path = /var/spool/samba
browseable = no
guest ok = no
writable = no
printable = yes
```

6. Restart smb and rpcgssd services:

```
# service smb restart
# service rpcgssd restart
```

7. Get a Kerberos ticket for the administrator:

```
# kinit administrator
Password for administrator@CORE-TME.NETAPP.COM:
# klist
Ticket cache: KEYRING:persistent:0:0
Default principal: administrator@CORE-TME.NETAPP.COM

Valid starting      Expires            Service principal
07/12/2016 11:28:54 07/12/2016 21:28:54 krbtgt/CORE-TME.NETAPP.COM@CORE-TME.NETAPP.COM
    renew until 07/19/2016 11:28:49
```

8. Join the domain:

```
# net ads join -U administrator
Enter administrator's password:
Using short domain name -- CORE-TME
Joined 'CENTOS7' to dns domain 'core-tme.netapp.com'
```

Note: All normal Windows domain rules apply: time skew within 5 minutes, user account with permissions to add computer objects to a domain, DNS able to locate domain controllers.

9. Create a keytab file:

```
# net ads keytab create -U administrator

Warning: "kerberos method" must be set to a keytab method to use keytab functions.
Enter administrator's password:
```

10. Verify the keytab file.

When a machine account is added to Active Directory using “net ads keytab,” the following SPNs are added to the krb5.keytab file automatically:

```
# ktutil
ktutil: rkt /etc/krb5.keytab
ktutil: list
slot KVNO Principal
-----
 1  3 host/centos7.core-tme.netapp.com@CORE-TME.NETAPP.COM
 2  3 host/centos7.core-tme.netapp.com@CORE-TME.NETAPP.COM
 3  3 host/centos7.core-tme.netapp.com@CORE-TME.NETAPP.COM
 4  3 host/centos7.core-tme.netapp.com@CORE-TME.NETAPP.COM
 5  3 host/centos7.core-tme.netapp.com@CORE-TME.NETAPP.COM
 6  3 host/centos7@CORE-TME.NETAPP.COM
 7  3 host/centos7@CORE-TME.NETAPP.COM
 8  3 host/centos7@CORE-TME.NETAPP.COM
 9  3 host/centos7@CORE-TME.NETAPP.COM
10  3 host/centos7@CORE-TME.NETAPP.COM
11  3 root/centos7.core-tme.netapp.com@CORE-TME.NETAPP.COM
12  3 root/centos7.core-tme.netapp.com@CORE-TME.NETAPP.COM
```

```

13 3 root/centos7.core-tme.netapp.com@CORE-TME.NETAPP.COM
14 3 root/centos7.core-tme.netapp.com@CORE-TME.NETAPP.COM
15 3 root/centos7.core-tme.netapp.com@CORE-TME.NETAPP.COM
16 3 root/centos7@CORE-TME.NETAPP.COM
17 3 root/centos7@CORE-TME.NETAPP.COM
18 3 root/centos7@CORE-TME.NETAPP.COM
19 3 root/centos7@CORE-TME.NETAPP.COM
20 3 root/centos7@CORE-TME.NETAPP.COM
21 3 CENTOS7$@CORE-TME.NETAPP.COM
22 3 CENTOS7$@CORE-TME.NETAPP.COM
23 3 CENTOS7$@CORE-TME.NETAPP.COM
24 3 CENTOS7$@CORE-TME.NETAPP.COM
25 3 CENTOS7\$@CORE-TME.NETAPP.COM

```

No other SPNs should be required for the machine account. If you notice, there are SPNs for “root/” in the keytab. Since there is a UNIX user named root in the SVM by default, no name mapping considerations have to be made for the client unless a different mapping is desired.

If a different mapping is needed, a [KRB to UNIX name mapping must exist](#) for machine\$ either locally on the SVM (name mapping rule or UNIX user) or on the Active Directory object (in the form of a uidNumber/GidNumber attribute in LDAP).

The easiest way to resolve this is through the local unix-user:

```

::*> unix-user create -vserver parisi -user CENTOS7$ -id 10001 -primary-gid 1
::*> unix-user show -vserver parisi -user CENTOS7$
    Vserver: parisi
    User Name: CENTOS7$
    User ID: 10001
    Primary Group ID: 1
    User's Full Name:

```

11. Test the krb-unix mapping for the root SPN or for the machine account SPN if desired:

```

::> set diag
::*> diag secd name-mapping show -node ontap-tme-prod-03 -vserver parisi -direction krb-unix -
name CENTOS7$@CORE-TME.NETAPP.COM
CENTOS7$@CORE-TME.NETAPP.COM maps to CENTOS7$

::*> diag secd name-mapping show -node ontap-tme-prod-03 -vserver parisi -direction krb-unix -
name root/centos7.core-tme.netapp.com@CORE-TME.NETAPP.COM
root/centos7.core-tme.netapp.com@CORE-TME.NETAPP.COM maps to root

```

12. Ensure that the following services are running and enabled on boot:

```

systemctl start ntpd
systemctl enable ntpd
systemctl start smb
systemctl enable smb
systemctl start winbind
systemctl enable winbind
systemctl start sssd
systemctl enable sssd

```

13. Test the domain connectivity:

```

# net ads info
LDAP server: 10.193.67.181
LDAP server name: stme-infra02.core-tme.netapp.com
Realm: CORE-TME.NETAPP.COM
Bind Path: dc=CORE-TME,dc=NETAPP,dc=COM
LDAP port: 389
Server time: Tue, 12 Jul 2016 11:33:29 EDT

```

```
KDC server: 10.193.67.181
Server time offset: 0
```

```
# wbinfo -t
checking the trust secret for domain CORE-TME via RPC calls succeeded
```

14. Ensure that the NFS unix-user or equivalent name mapping is in place so that the service account (nfs/fqdn@REALM) can authenticate:

```
::*> unix-user create -vserver parisi -user nfs -id 10002 -primary-gid 1
::*> unix-user show -vserver parisi -user nfs
      Vserver: parisi
      User Name: nfs
      User ID: 10002
Primary Group ID: 1
User's Full Name:
```

15. On the NFS client, [ensure that SSSD \(or the LDAP client equivalent\) is configured](#). Or, use a local UNIX user in /etc/passwd and on the SVM.

To test LDAP:

```
# id ldapuser
# getent passwd ldapuser
```

16. Attempt to mount the SVM data interfaces with Kerberos. The SVM must already have the following created and configured:

- Kerberos realm
- Kerberos interfaces
- DNS A/AAAA records in the DNS server (forward and reverse)
- Permitted encyptes for Kerberos
- Export policy rules on the NFS exports and parent directories that allow Kerberos

All of the above are covered in the section called "[Configuring the ONTAP System for Kerberos](#)" in this document.

Mount example:

```
[root@centos7 ~]# mount -o sec=krb5 parisi-nfs:/nfs /kerberos
[root@centos7 ~]#
```

17. su as a different user and kinit. CD into the mount and check for your NFS service ticket:

```
[root@centos7 ~]# su test@CORE-TME.NETAPP.COM
[test@core-tme.netapp.com@centos7 ~]$ kinit test@CORE-TME.NETAPP.COM
Password for test@CORE-TME.NETAPP.COM:
[test@core-tme.netapp.com@centos7 ~]$ klist
Ticket cache: KEYRING:persistent:106003697:106003697
Default principal: test@CORE-TME.NETAPP.COM

Valid starting          Expires                Service principal
06/29/2016 16:38:26    06/30/2016 02:38:26    krbtgt/CORE-TME.NETAPP.COM@CORE-TME.NETAPP.COM
    renew until 07/06/2016 16:38:21

[test@core-tme.netapp.com@centos7 ~]$ mount | grep kerberos
```

```

parisi-nfs:/nfs on /kerberos type nfs4
(rw,relatime,vers=4.0,rsize=65536,wsiz=65536,namlen=255,hard,proto=tcp,port=0,timeo=600,retran
s=2,sec=krb5,clientaddr=10.193.67.225,local_lock=none,addr=10.193.67.226)

[test@core-tme.netapp.com@centos7 /]$ cd /kerberos

[test@core-tme.netapp.com@centos7 kerberos]$ klist -e
Ticket cache: KEYRING:persistent:106003697:106003697
Default principal: test@CORE-TME.NETAPP.COM

Valid starting          Expires                Service principal
06/29/2016 16:39:43   06/30/2016 02:38:26   nfs/parisi-nfs.core-tme.netapp.com@CORE-
TME.NETAPP.COM
    renew until 07/06/2016 16:38:21, Etype (skey, tkt): aes256-cts-hmac-sha1-96, aes256-
cts-hmac-sha1-96
06/29/2016 16:38:26   06/30/2016 02:38:26   krbtgt/CORE-TME.NETAPP.COM@CORE-TME.NETAPP.COM
    renew until 07/06/2016 16:38:21, Etype (skey, tkt): aes256-cts-hmac-sha1-96, aes256-
cts-hmac-sha1-96

```

Configuring Kerberos in ESXi 6.x

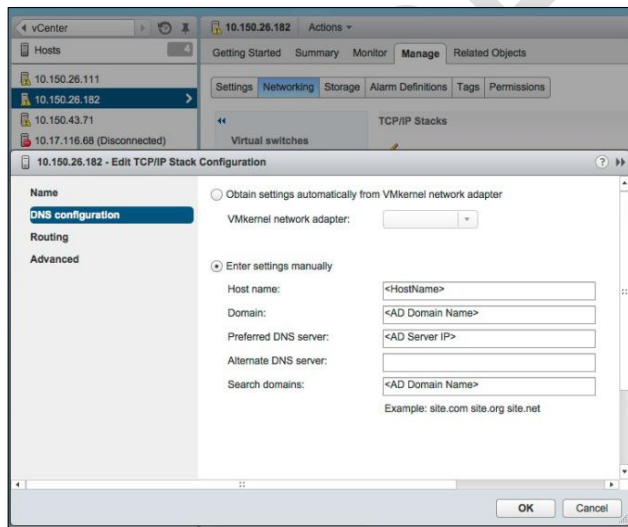
The configuration is all done using GUI and essentially joins an ESXi server instance to Active Directory. For information on using Kerberos with non-Windows KDCs, contact VMware support.

- ESXi 6.0 supports only DES encryption types.
- ESXi 6.5 and later removes support for DES and supports only AES encryption types.

Configuration Steps 41) Configuring Kerberos in ESXi 6.0

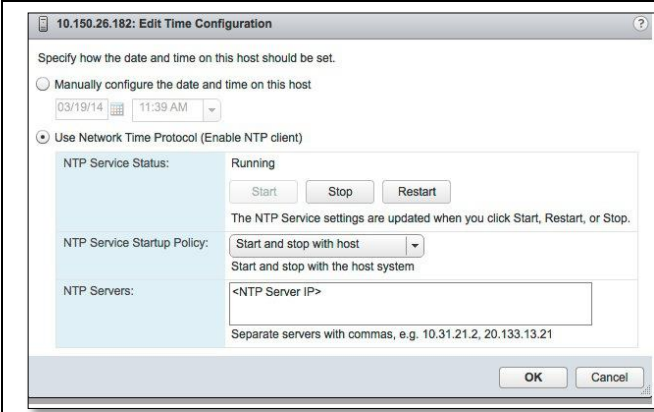
Configure DNS

Kerberos needs DNS to work properly. Forward (A/AAAA) and reverse (PTR) records are necessary for proper functionality.



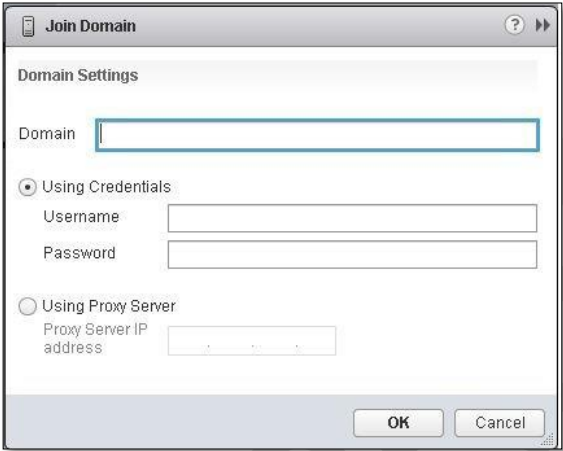
Configure Network Time Protocol (NTP)

Having NTP configured avoids issues with time skew, which can cause Kerberos authentication to fail.



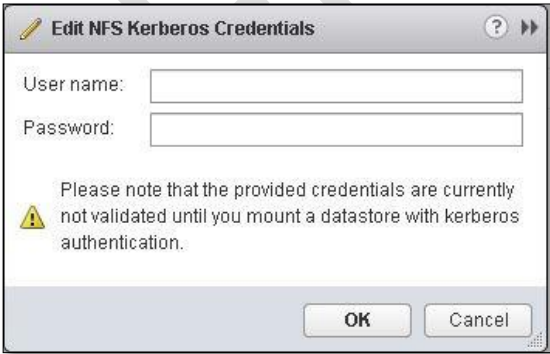
Join ESXi to the Domain

Creates the machine account, SPN, and so on. This process is similar to what Data ONTAP does with CIFS setup and NFS Kerberos configuration.



Specify a User Principal Name (UPN)

This is used by ESXi to kinit and grab a TGT.



Create Your NFS Datastore for Use with Kerberos Authentication

Ideally, you use NFSv4.1 for this.

The screenshot shows the 'New Datastore' wizard at Step 3, 'Select NFS version'. The left sidebar shows steps 1 through 6, with Step 3 highlighted. The main area has 'NFS Version' with radio buttons for 'NFS 3' and 'NFS 4.1'. 'NFS 4.1' is selected. A warning icon and text state: 'Use at most one NFS version to access a given datastore. Consequences of mounting one or more hosts to the same datastore using different versions can include data corruption.' Buttons for 'Back', 'Next', 'Finish', and 'Cancel' are at the bottom.

The screenshot shows the 'New Datastore' wizard at Step 4, 'Name and configuration'. The left sidebar shows steps 1 through 6, with Step 4 highlighted. The main area has 'Datastore name: Datastore-1'. Below is 'NFS Share Details' with a note: 'Enter the NFS share details. If the server that will back this datastore has trunking enabled, below you can enter additional IPs, which the ESX will use to achieve multipathing to this NFS server mount point.' There are fields for 'Folder:' (containing 'E.g. /datastore-001') and 'Server(s):' (containing 'E.g. nas, nas.it.com or 192.168.0.1'). A 'Server(s) to be added' section has a search filter and a table with 0 items. A note says: 'If you intend to configure an existing datastore on new hosts in the datacenter, it is recommended to use the "Mount to additional hosts" action instead.' There is an 'Access Mode' section with a checkbox for 'Mount NFS as read-only'. Buttons for 'Back', 'Next', 'Finish', and 'Cancel' are at the bottom.

The screenshot shows the 'New Datastore' wizard at Step 5, 'Configure Kerberos authentication'. The left sidebar shows steps 1 through 6, with Step 5 highlighted. The main area has the title 'Configure Kerberos authentication' and a note: 'The NFS 4.1 client supports Kerberos authentication of RPC headers. You can enable it here.' There is a checkbox for 'Enable Kerberos-based authentication' which is checked. A warning icon and text state: 'In order to use Kerberos authentication, each host that mounts this datastore has to be a part of an Active Directory domain and its NFS authentication credentials need to be set. This is done on the Authentication Services page on each host.' Buttons for 'Back', 'Next', 'Finish', and 'Cancel' are at the bottom.

Configuring Kerberos, LDAP, and NFSv4 in Data ONTAP Operating in 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 NetApp 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 uses 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 fails.

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, 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 geacos
ldap.usermap.attribute.windowsaccount sAMAccountName
ldap.usermap.base
ldap.usermap.enable   on
```

In addition to the preceding, the `/etc/nsswitch.conf` file needs 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 needs to be configured so that LDAP can be reached 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 “[Configuring the Domain Controller as an LDAP Server.](#)”

For information on configuring clients to use LDAP, see the section “[Configuring the Client to Use LDAP.](#)”

Configuring NFSv4 in 7-Mode

To set up NFSv4 on a 7-Mode appliance, use the following steps:

18. Set the NFSv4 domain.

```
options nfs.v4.id.domain domain.netapp.com
```

19. 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 Data ONTAP. These steps do not cover explanations or describe anything beyond the simple “how to” of the specified task. You can use these steps 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 are references to previous portions of this document, denoted by a linked [?]. Simply click the link to be redirected to the section in question.

Table 31) Presetup 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: AES or DES?
	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 is used for data LIFs.

Cluster Configuration: Kerberos

Completed	Step
	Configure DNS for the SVM using the <code>dns</code> 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. [?]
	Modify the NFS SVM to specify permitted enctypees. [?]

Cluster Configuration: LDAP

Completed	Step
	Choose the LDAP schema or copy an existing schema and modify it. [?]
	Create the LDAP client for the SVM. [?]
	Enable LDAP on the SVM. [?]
	Modify the ns-switch to leverage LDAP for name/groups lookups. [?]

Domain Controller Configuration: Kerberos

Completed	Step
	Allow DES encryption in the domain (8.2.x and earlier only). [?]
	Create the machine objects for the NFS clients and their keytabs. [?]
	Allow DES encryption for the machine accounts (required for 8.2.x and earlier only). [?]
	Add the NFS client to DNS in Windows AD. [?]
	Add the SVM data LIFs to DNS in Windows AD. [?]
	Add SRV records to DNS if needed (specifically the Kerberos-master record). [?]
	Verify the SPNs. [?]
	Modify the NFS server machine account to allow DES and AES only. [?]

Domain Controller Configuration: LDAP

Completed	Step
	Extend the AD schema (if not already done). [?]
	Add SRV records as needed. [?]
	Modify user and groups to include UNIX-style attributes. [?]
	Add users to auxiliary groups as needed. [?]
	Create name mapping rules if desired. [?]
	Create netgroups in LDAP if desired. [?]

Linux Client Configuration: Kerberos and LDAP

Completed	Step
	Verify that the host name is set. [?]
	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 <code>/etc/krb5.conf</code> . [?]
	Create the <code>/etc/krb5.keytab</code> 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
- 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
- UNIX Men
www.unixmen.com

Relevant Technical Reports

- TR-3580: NFSv4 Enhancements and Best Practices
www.netapp.com/us/media/tr-3580.pdf
- TR-4067: NFS Best Practices and Implementation Guide
www.netapp.com/us/media/tr-4067.pdf
- TR-4523: DNS Load Balancing in ONTAP
www.netapp.com/us/media/tr-4523.pdf
- TR-4557: NetApp FlexGroup Technical Overview
www.netapp.com/us/media/tr-4557.pdf
- TR-4616: NFS Kerberos in ONTAP
www.netapp.com/us/media/tr-4616.pdf
- TR-4668: Name Services Best Practice Guide
www.netapp.com/us/media/tr-4668.pdf
- TR-4835: How to Configure LDAP in ONTAP
www.netapp.com/us/media/tr-4835.pdf

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.

Copyright Information

Copyright © 2017–2020 NetApp, Inc. All Rights Reserved. Printed in the U.S. No part of this document covered by copyright may be reproduced in any form or by any means—graphic, electronic, or mechanical, including photocopying, recording, taping, or storage in an electronic retrieval system—without prior written permission of the copyright owner.

Software derived from copyrighted NetApp material is subject to the following license and disclaimer:

THIS SOFTWARE IS PROVIDED BY NETAPP "AS IS" AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, WHICH ARE HEREBY DISCLAIMED. IN NO EVENT SHALL NETAPP BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

NetApp reserves the right to change any products described herein at any time, and without notice. NetApp assumes no responsibility or liability arising from the use of products described herein, except as expressly agreed to in writing by NetApp. The use or purchase of this product does not convey a license under any patent rights, trademark rights, or any other intellectual property rights of NetApp.

The product described in this manual may be protected by one or more U.S. patents, foreign patents, or pending applications.

Data contained herein pertains to a commercial item (as defined in FAR 2.101) and is proprietary to NetApp, Inc. The U.S. Government has a non-exclusive, non-transferrable, non-sublicensable, worldwide, limited irrevocable license to use the Data only in connection with and in support of the U.S. Government contract under which the Data was delivered. Except as provided herein, the Data may not be used, disclosed, reproduced, modified, performed, or displayed without the prior written approval of NetApp, Inc. United States Government license rights for the Department of Defense are limited to those rights identified in DFARS clause 252.227-7015(b).

Trademark Information

NETAPP, the NETAPP logo, and the marks listed at <http://www.netapp.com/TM> are trademarks of NetApp, Inc. Other company and product names may be trademarks of their respective owners.