Technical Report

Name Services Best Practices Guide

ONTAP 9.3

Chris Hurley, NetApp
March 2018 | TR-4668

Abstract

This document provides a comprehensive list of best practices, limits, recommendations, and considerations when implementing network-attached storage (NAS) solutions such as CIFS/SMB and NFS in NetApp® ONTAP®.

Information Classification

Public

NetApp®

Version History

Version	Date	Document Version History
Version 1.0	March 2018	Chris Hurley: Initial commit. Complete update to TR-4379. Older document can be found: TR-4379

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF BEST PRACTICES

1 Overview

The NetApp ONTAP operating system provides the ability to unify clients under a single namespace by way of storage virtual machines (SVMs). These SVMs can live on clusters that are up to 24 nodes in size. Each SVM provides the ability to offer individualized LDAP, NIS, DNS, and local file configurations for authentication purposes. These features are also known as “name services.”

External servers can provide replicated copies of databases containing user information, such as UID, GID, group membership, home directory, and other information, as well as netgroup and name resolution capabilities. These external servers make it possible to manage large environments that span global locations without extra administrative overhead and with the ability to reduce WAN latency by providing localized copies of databases to clients and servers.

1.1 Scope

This document covers the following topics:

- ONTAP NAS overview
- Name service overview
- Supported configurations
- Benefits of using name services with NAS
- Configurations and best practices

Note: This document covers only versions of ONTAP later than 9.3. There are some references to other versions on ONTAP.

1.2 Intended Audience and Assumptions

This technical report is for storage administrators, system administrators, and data center managers. It assumes basic familiarity with the following:

ONTAP and any supported platform it runs on (FAS, AFF, Select, Cloud)

Network file-sharing protocols

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

1.3 What Is a Name Service?

Name services are objects that process name requests from NetApp storage systems. Name requests can be for users, groups, netgroups, or host names and can be retrieved from local or external resources. These resources include:

Local files (hosts, passwd, netgroup, and so on)

- DNS
- NIS
- LDAP
- Active Directory

1.4 Storage Virtual Machine(SVM) Types

ONTAP includes multiple types of SVMs:

- **Data SVMs** are used for data access.
- **Cluster SVMs** are used for cluster administration

2 NAS in ONTAP

The ONTAP 9 operating system provides world-class enterprise-level storage to clients running NAS operations for use cases including, but not limited to:

- Home directories
- Application database hosting
- Archiving and staging
- Software source control
- Log file storage
- Video streaming
- Unstructured data sharing

The ONTAP operating system supports cutting-edge technologies in both CIFS and NFS so that the latest and greatest feature sets can be leveraged in data centers across the globe.

Supported protocol versions include:

- NFSv3, NFSv4, and NFSv4.1
- SMB 1.0, SMB 2.x, and SMB 3.x

For more information about supported features for these protocols, see TR-4067 and TR-4191.

Note: SMB1.0 is mentioned as supported, but should be deprecated and disabled in your environment due to the insecurity and vulnerabilities is has.

2.1 How NAS Requests in the ONTAP Operating System Work

A cluster can contain up to 24 nodes for NAS operations in ONTAP. Each physical node can own virtual objects such as volumes or data LIFs. An SVM spans all nodes in a cluster and allows interaction of logical storage entities under a single namespace. When a NAS client attempts to connect to an ONTAP system by using CIFS or NFS, that request can potentially reach any node in a 24-node cluster based on DNS, client settings and which node hosts the LIF that owns the IP address. If a node hosting a data LIF is used in a NAS operation that does not own the data volume being requested for access, then the NAS request is parsed into an optimized ONTAP protocol for disk access and that traffic passes through a dedicated high speed Ethernet (10GbE or 40GbE) cluster back-end network.

NAS Basics

The following section covers the basic interaction of NAS requests at a high level.

Volumes

All user data in ONTAP systems lives in flexible volumes (NetApp FlexVol® volumes). These volumes are located locally to the node that hosts the physical disk space (aggregate). In ONTAP, data can be accessed anywhere in a cluster, regardless of on which node it physically lives.

Note: ONTAP also has the ability to store data in a container called a FlexGroup® . This concept is not discussed in this Technical Report. For more information on how data is stored in FlexGroup volumes, please refer to TR-4557.

Logical Interfaces

Each SVM owns storage objects, such as volumes and logical interfaces (LIFs). LIFs can host management, data, or cluster traffic, depending on the assigned role and data protocols allowed. The option `-data-protocol` allows a storage administrator to specify which data protocols are allowed on the data LIF. When a data protocol is allowed on a data LIF, the LIF then listens on a specific list of ports

for the protocol. For NAS protocols (CIFS and NFS), the ports listed in Table 1 are opened when the protocol is allowed on a data LIF.

Table 1) Ports for NFS and CIS traffic on protocol-enabled data LIFs

Protocol	Ports
NFS	2049: NFS 2049 (program version 400010): vStorage 111: portmapper 635: mountd 4045: Network Lock Manager (NLM) 4046: Network Status Monitor (NSM) 4049: rquota
CIFS	135: RPC 139: NetBIOS 445: SMB 40001: SMB Witness
FlexCache	2050: FlexCache (As origin volume to 7-mode FlexCache)

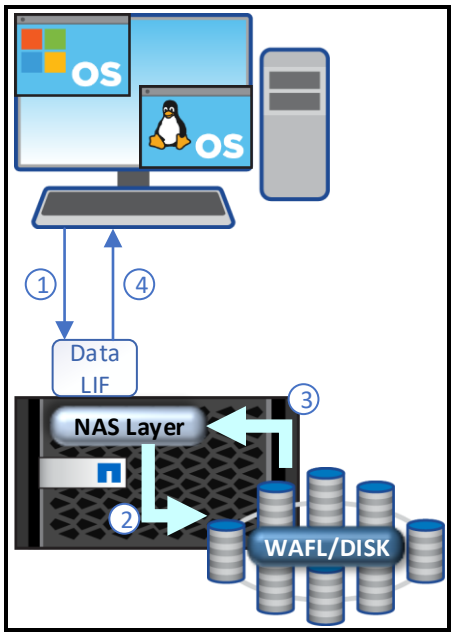
When a NAS request is made to an SVM, the request always arrives on a data LIF (including calls such as `showmount`). The data LIF chosen depends on the client's name resolution settings. ONTAP has a DDNS client that can automatically register a LIF address into the DNS server. More information on this can be found in the DDNS section of this document. The ONTAP on-box DNS load balancer can assist in directing the client to the least-used LIF and node combination. For more information about DNS load balancing in ONTAP, see TR-4073, TR-4182 or TR-4523.

When a data LIF receives a NAS request, it passes through the NAS layer for processing and handling

NAS Layer

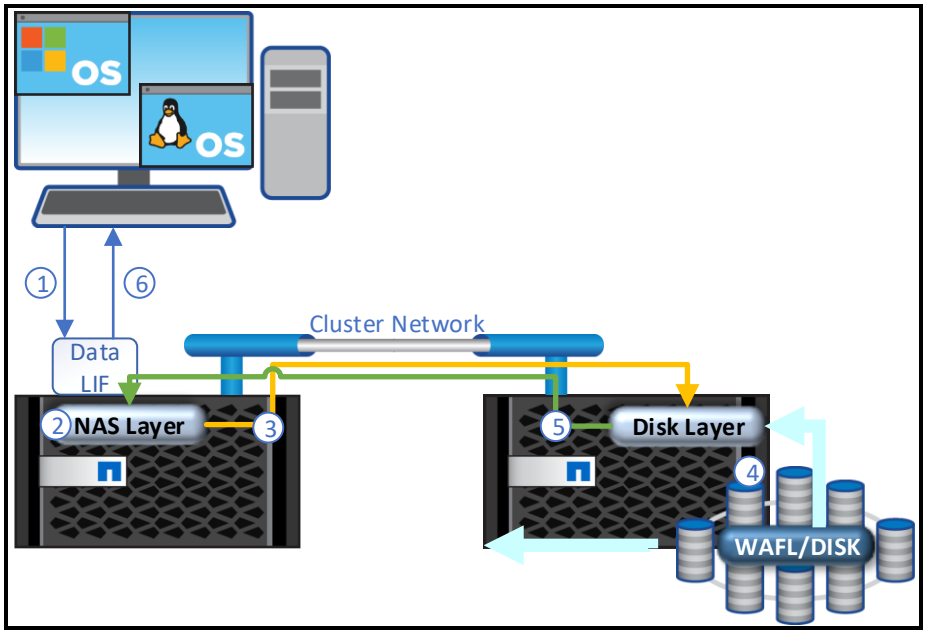
When a NAS request arrives on a physical interface, that request is forwarded to the NAS layer for processing. The NAS layer sends RPC calls to cluster processes, such as security daemon (SecD) and the volume location database (VLDB), to determine user credentials, data locality, and other NAS configuration aspects so that the request passes through the appropriate paths in the cluster. If the data being requested is local to the node that currently hosts the data LIF that received the request from the client, then the request goes straight to disk by way of an ONTAP direct I/O path mechanisms. If the data is on another node in the cluster, then the request traverses the cluster network.

Figure 1) NAS protocol path in ONTAP; LIF & volume on same node



1. NAS request is sent from the client to the SVM data LIF.
2. NAS layer receives all NAS requests and processes them. If authentication is needed, an RPC call is sent to SecD to perform authentication.
3. If local, disk request is sent right to disk.
4. Response passes back through the stack and is sent back to the NAS client.

Figure 2) NAS Protocol path in ONTAP; LIF & volume on different nodes



1. NAS request is sent from the client to the SVM data LIF
2. NAS layer receives all NAS requests and processes them. If authentication is needed, an RPC call is sent to SecD to perform authentication
3. If remote, the disk request is sent over the cluster network to the node local to the data.
4. Disk layer processes the request and reads from/writes to disk.
5. Disk layer response passes passes back through the stack and back to the node that processed the NAS layer request.
6. Response passes back through the stack and is sent back to the NAS client.

Security Daemon (SecD)

SecD is an application that runs on a per-node basis. The SecD application handles name service lookups such as Active Directory, DNS NIS, and LDAP, as well as credential queries and name mapping. SecD is node-specific, which means that a SecD process exists on each node in a cluster. When a NAS request arrives on a data LIF, the node that hosts that data LIF also hosts the SecD application used for authentication of the user and/or host.

SecD communicates with external name services on the node that the NAS request was received on, so there needs to be at least one LIF in the SVM that is routable to the name service servers. SecD is able to intelligently forward name service requests to remote nodes when necessary, if no access to name services servers is possible from the node processing the NAS request.

Management Gateway Daemon (mgwd)

The management gateway in the ONTAP operating system is exactly what it sounds like: It is the gateway into managing a cluster. It is responsible for maintaining and reporting cluster health/quorum; receiving SSH logins, SNMP, and NetApp Manageability SDK calls from management software (such as NetApp OnCommand® System Manager); processing export rules; and maintaining an exports cache. In addition, it interacts with all of the other cluster applications through RPC to send and receive requests for configuration reads and writes.LIFs

Best Practice 1) SecD and Data LIFs

Best practice is to have a data LIF on each node for the SVM that routes properly to name services to enable data locality. If this is not possible, it is still a requirement to have at least one data LIF per SVM that can route to name services for NAS environments.

3 Name Services

Name services are external servers that contain user, group, netgroup. and host information in an enterprise environment. This definition includes NIS, LDAP, and DNS, as well as local files and Microsoft Windows Active Directory.

3.1 Benefits of Using Name Services

In large environments with thousands of users and hosts, managing individual flat files for users, groups, netgroups, and host resolution on each individual machine is virtually impossible. Name service servers allow administrators to keep a database of current information for business-critical objects with which client machines and storage devices can interact for consistency of these objects across an enterprise environment. When all clients and storage access the same servers with the same databases, there can be no mistake in credential retrieval or host name resolution.

Other benefits of using name service servers include:

- Consolidation of users, groups, netgroups, and host names
- Disaster recovery through site replication of name service server databases
- Reduction in WAN latency by way of site replication to produce localized copies of server databases
- Load balancing and failover functionality

3.2 Name Services in ONTAP Operating Systems

In ONTAP beginning with version 9.3, configuration of name services functionality has been moved to its own command set called `vserver services name-service`. This allows for a single entry into the configuration of all associated name services for the SVM and cluster.

```
cluster::> vserver services name-service>
    dns         ldap        netgroup    nis-domain ns-switch  unix-group    unix-user
```

Additional name services diagnostic commands, such as `getXXbyYY`, exist at the **advanced** privilege level. These commands can help in confirming configuration integrity and troubleshooting issues. More detail on this command set is given throughout this document where it applies.

```
cluster::> vserver services name-service> set advanced
Warning: These advanced commands are potentially dangerous; use them only when directed to do so
by NetApp personnel.
Do you want to continue? {y|n}: y

cluster::vserver services name-service*>
    dns         getxxbyyy  ldap        netgroup    nis-domain ns-switch
    unix-group unix-user
```

What Is ns-switch?

Ns-switch is the *name service* switch. This controls which name service sources are used and the order in which the name service sources are used by the SVM for user/group, host, namemap, and netgroup lookups.

To view the current ns-switch configuration:

```
cluster:> vserver services name-service*> ns-switch show -vserver svm1
                                                Source
Vserver          Database       Enabled         Order
---------------- ------------   ---------       ---------
svm1             hosts          true            files,dns
svm1             group          true            files,ldap
svm1             passwd         true            files,ldap
svm1             netgroup       true            files
svm1             namemap        true            files
```

Note that in the preceding, ONTAP has support for granular control over `passwd, group, netgroup`, and so on, making the ns-switch functionality comparable to standard UNIX `nsswitch.conf` files.

What Is nm-switch?

Nm-switch is the *name mapping* switch. This controls which name mapping source is used by the SVM for mapping UNIX users to Windows users and vice versa. This only applies to multiprotocol environments. Name mapping rules can exist in local tables in the cluster or on LDAP servers.

To modify the current nm-switch configuration:

```
cluster::> name-service ns-switch modify -vserver svm1 -database namemap -sources
    files ldap
```

Order of Operations for ns-switch and nm-switch

When specifying multiple sources for ns-switch and nm-switch, it is important to consider what happens in the event of successful or unsuccessful lookups.

If using multiple name service sources in ns-switch and/or nm-switch, the following are true:

- When the object queried exists in the first source and the response is successful, the operation is finished. The next name service source is not tried, even if the object exists in both places.
- If the object doesn't exist in any name service source, the operation fails.

Therefore, it's important to list name service sources in order of priority.

When does ONTAP use namemap?

ONTAP does not always namemap (usermap). When there are NFS clients accessing a volume or qtree with a UNIX security style, the credentials passed are the ones used. The namemap process is not triggered in this scenario. ONTAP **always** usermaps when the volume or qtree security style is NTFS. This is because at its roots, ONTAP is a UNIX-based OS. By default, all users are mapped to the local ONTAP UNIX user pcuser and in SMB-only deployments this default map is sufficient. ONTAP also **always** usermaps when the protocol does not match the volume or qtree security style. A more detailed matrix is found in Table 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 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 svm1 -database namemap

                    Vserver: svm1
Name Service Switch Database: namemap
   Name Service Source Order: files, ldap
```

When using LDAP for name mapping, ONTAP will use whatever attribute the LDAP client schema is configured to use. In other words, for NFS clients accessing NTFS volumes or qtrees, the `windows-account-attribute` field is looked up for the NFS user and then used to produce Windows credentials. For SMB clients accessing UNIX volumes or qtrees, the `windows-to-unix-attribute` field is looked up in Active Directory.

Best Practice 2) Specifying external services in namemap

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

If no explicit name mapping can be found in the name services entries for the user, then ONTAP will try to implicitly map based on username. The particular username passed in the protocol's authentican process will be looked up in the other protocol's directory. This means that a username passed via NFS will be looked up in Active Directory. If the same username is found, those Windows credentials are attached to the NFS user's credentials. A username passed via SMB will be looked up via the SVM's ns-switch sources. If the same username is found, those UNIX credentials will be attached to the SMB user's credentials. If ONTAP cannot find any implicit mapping for the username, there is a 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.

Best Practice 3) Specifying local files

Specify "files" as the first name service database option to avoid situations where the root user or other unix users fail to resolve when external name services are unavailable.

Table 2) Name mapping/default user considerations for multiprotocol NAS access

Accessing Protocol	Volume/qtree security style	Name mapping direction	Default user
NFS	UNIX	N/A (v3: UID lookup only) (v4: username lookup)	N/A
NFS	NTFS	UNIX -> Windows	Default Windows user (NFS server option default-win-user)
CIFS/SMB	UNIX	Windows -> UNIX	Default UNIX user (CIFS server option default-unix-user; pcuser by default)
CIFS/SMB	NTFS	Windows -> UNIX (initial authentication) NTFS ACLs used after initial entry.	Default UNIX user (CIFS server option default-unix-user; pcuser by default)

4 Whats New?

New features have been added to name services in recent ONTAP versions. The following is a partial list:

- DNS and NIS statistics
- `getXXbyYY` support
- Improved NIS troubleshooting tools (tracing and showing bound servers)
- Name service queue status
- Name service configuration mirroring and repair
- Name service caching and management (ONTAP 9.3)
  See the Caching in ONTAP section of this document for detailed information
- Name service connectivity checking (LDAP and DNS)

5 Supported NS-Switch Configurations

Table 3 shows a list of the supported name service switch databases in the ONTAP operating system.

Table 3) Supported name service sources in ONTAP

NS-Switch Database	Supported Name Service sources
Hosts	DNS, local files
Passwd (users)	NIS, LDAP, local files
Group	NIS, LDAP, local files
Netgroup	NIS, LDAP, local files
Namemap	LDAP, local files

5.1 Hosts

In the ONTAP operating system, host name lookups (such as for use with export policy rules) are supported for use in both DNS and local files for SVMs.

Note: Host name resolution through LDAP and NIS is currently not supported.

Checking DNS Server Configurations

Starting in version 9.2, ONTAP provides a method to check the DNS configuration of the individual SVMs. This checker is invoked upon reconfiguration of the DNS settings or on demand. If upon reconfiguration of the DNS servers, the checker finds that a DNS server is non-responsive, the result will be that the new configuration is discarded. This behavior can be skipped with the modifier `-skip-config-validation`. Invoking the DNS checker will result in a DNS query sent for the name "example" plus the domain name of the particular SVM. If your SVM's domain name is "example.com", then ONTAP will query each DNS server configured for the SVM for "example.example.com". ONTAP will also guage the response time of the reply and display the result.

Example:

```
cluster::> vserver services name-service dns check -vserver svm1
                                Name Server
Vserver       Name Server     Status       Status Details
------------- --------------- ------------ --------------------------
svm1          203.0.113.44    down         Operation timed out.
svm1          203.0.113.93    up           Response time (msec): 2
svm1          198.51.100.200  up           Response time (msec): 2
3 entries were displayed.
```

Upgrade Considerations

If you intend to upgrade a cluster from 8.2.x or earlier, ONTAP can no longer use the cluster SVM to fallback for DNS lookups. First, ensure that the data SVMs have the appropriate DNS setting prior to upgrade. ONTAP will no longer fallback to the cluster's DNS servers for hostname resolution. Next, if local host names are being used in data SVMs, make sure that the host names and IP addresses exist in both the cluster admin SVM and the data SVMs. These steps reduce the chance of having outages. Ideally, these host names are sourced from DNS rather than from local files.

5.2 User and Group Information

User and group information (such as UID/GID) can be stored in files, NIS, or LDAP in all versions of ONTAP. In Data ONTAP 8.3 and later, users and groups can leverage different name service databases in the same SVM (such as local files only for groups, LDAP then files for users).

Example:

```
cluster::> name-service ns-switch show -vserver svm1 -database group,passwd
  (vserver services name-service ns-switch show)
                                Source
Vserver         Database        Order
--------------- ------------    ---------
svm1            group           files
svm1            passwd          ldap,
                                files
2 entries were displayed.
```

NetApp recommends that users and groups use LDAP for security and scalability, because LDAP can provide encrypted lookups and supports use with more servers than a NIS configuration. This is also recommended over using local files since many other applciations can use the same LDAP source for authentication other than ONTAP.

Local User and Group Limits

As local users and groups are created, the replicated database tables that make the ONTAP operating system run properly grow in size and memory allocation. If these databases grow to the point of memory exhaustion when reading/writing the tables, cluster outages can occur. Therefore, ONTAP has a hard limit on local users and groups. This limit is clusterwide and affects all SVMs.

Best Practice 4) Local UNIX Users and Groups

In versions earlier than the Data ONTAP 8.2.3 operating system, there was no hard limit on local users and groups. However, that does not mean that there is no actual limit. NetApp highly recommends not exceeding the local UNIX user and group limits as defined in **Error! Reference source not found.** when using ONTAP operating system versions earlier than 8.2.3.

Note: This limit is for local UNIX users and groups. Local CIFS users and groups (vserver cifs users-and-groups) have an separate limit

Table 4) Limits in local users and groups in ONTAP

	Local UNIX User Limit (Default and Maximum)	Local UNIX Group Limit (Default and Maximum)
Local files without scaled/file-only mode	32,768 (default) 65,536 (maximum)	32,768 (default) 65,536 (maximum)
Local files in scaled/file-only mode	Passwd file size: 10MB*	Group file size: 25MB
	Note: group and passwd file sizes can be overridden with `-skip-file-size-check` but larger file sizes have not been tested Users: 400K Groups: 15k Group memberships: 3000k SVMs: 6	

As previously mentioned, the local UNIX user and group limits are clusterwide and affect clusters with multiple SVMs. Thus, if a cluster has four SVMs, then the maximum number of users in each SVM must add up to the maximum limit set on the cluster. If more local entries are required and an external name service is not an option, see the section on Scaled/File-Only mode.

For example:

- SVM1 has 2,000 local UNIX users.
- SVM2 has 40,000 local UNIX users.
- SVM3 has 20 local UNIX users.
- SVM4 would then have 23,516 local UNIX users available to create.

Any UNIX user or group creation attempted beyond the limit results in an error message.

Example:

```
cluster::> unix-group create -vserver svm1 -name test -id 12345

Error: command failed: Failed to add "test" because the system limit of {limit number}
       "local unix groups and members" has been reached.
```

The limits are controlled by the following commands at the advanced privilege level:

```
cluster::*> unix-user max-limit
    modify show
```

Upgrade Considerations for UNIX user and group limits

When upgrading to a ONTAP version with the hard limits set, there is no check for existing users and groups. Therefore, if the limit is already exceeded on the cluster, the upgrade succeeds, but no new users and groups can be created. Also, if problems occur while the limit is exceeded, support issues might arise (for example, support deems your configuration as “unsupported”). Reducing the number of users and groups to below the limits is highly recommended. You can take this action before or after upgrades.

5.3 Netgroups

Netgroups are supported for use with files, NIS, and LDAP in all versions of ONTAP systems. NetApp recommends that netgroups use LDAP for security and scalability. When using netgroups, NetApp highly recommends leveraging the netgroup.byhost functionality (available in Data ONTAP 8.2.3 and later) for faster lookups and, thus, better performance. For information about configuring netgroups and netgroup.byhost maps in LDAP, see TR-4073: Secure Unified Authentication.

LDAP Netgroups

It is possible to leverage netgroup functionality in LDAP as opposed to NIS. Netgroups give storage administrators control of access to a series of hosts using a group, rather than needing to create a number of different rules per host. Using LDAP as an NIS server is covered in RFC-2307.

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 ONTAP and other LDAP clients query LDAP for netgroup-related objects. For netgroups, the nisNetgroup object class is used by default

Table 5) Object classes and attributes for NIS Objects in LDAP

Object Class	Used For	NIS Attributes Used
nisMap	NIS map abstraction	nisMapName
nisObject	Netgroup-by-host entries	nisMapName
		nisMapEntry
nisNetgroup	Netgroup members Nested Netgroup Members	nisNetgroupTriple
		memberNisNetgroup

NIS Object Terminology

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

Term	Definition
NIS map	NIS maps were designed to centralize and replace common files found in the /etc directory of Linux and UNIX clients. ONTAP currently supports the following NIS map types: passwd.byname and passwd.byuid group.byname and group.bygid netgroup netgroup.byhost (as of 8.2.3) Host name resolution in NIS is not currently supported. For more information about NIS maps, see http://docs.oracle.com/cd/E19683-01/817-4843/anis1-24268/index.html.
Netgroup	A netgroup is a set of (host,user,domain) triples (also known as tuples) used for permission and export access checking. ONTAP currently supports only hosts in netgroup entries. For more information about 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 (tuple) refers to the series of entries in a netgroup file consisting of (host,user,domain). A valid triple used in ONTAP consists of (host,,). Host names used in netgroup triples require DNS resolution in ONTAP. For best results in netgroup translation, see the name services best practices in TR-4067.
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 can reduce lookup time drastically and improve performance.

ONTAP Operating System Interaction with Active Directory LDAP for Netgroups

In the LDAP client schemas provided in the ONTAP operating system (for example, AD-IDMU, RFC-2307, and so on), 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 versions of ONTAP 8.2.3, 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 only if the default schemas are copied into new schemas. Default schemas in ONTAP are read only. For more information about default schemas, see the section in this document about LDAP schemas.

When Windows Server for NIS is installed in Active Directory, the container `DefaultMigrationContainer30` is created. This container is the default 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 ONTAP are 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 by using the utilities nis2ad and nismap or by 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 are creating 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 for UNIX is installed with server for NIS, a Windows 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. For examples, see TR-4073: Secure Unified Authentication.

Note: Microsoft has deprecated IDMU but the RFC2307bis schema extensions needed for configuration of the LDAP client in ONTAP will remain in Active Directory. Please see Microsoft's documentation on how to manage the RFC2307 attributes in your specific version of Windows AD.

Third-Party Schema Extensions

Active Directory provides an LDAP back end for use with directory services in Microsoft Windows. It also provides additional schema extensions to allow Active Directory to act as a UNIX identity management server. There are free schema extensions, such as services for UNIX (Windows 2003 and earlier) and identity management for UNIX (Windows 2003R2 and later), that allow LDAP clients to bind and search for UNIX attributes. ONTAP provides default read-only schemas for AD-SFU, AD-IDMU, and RFC-2307 schema types to help make the configuration simpler.

In addition to Microsoft Active Directory's integrated tools, there are third-party tools, such as Centrify's Vintela application suite, that extend the schema and provide a GUI for management. ONTAP supports any and all schema extensions that comply with RFC-2307 standards. To use third-party schema extensions with ONTAP, consult the vendor's product documentation on which schema attributes are leveraged and modify the client schema in ONTAP accordingly. TR-4073: Secure Unified Authentication covers how to create custom LDAP schemas for use with third-party vendors. The same general best practices for LDAP servers apply regardless of who is providing the schema.

LDAP Netgroup Optimization

LDAP servers can be optimized to allow faster lookups from storage systems running the ONTAP operating system. The following covers some general best practices that can be used. For specific best practices or steps to implement these best practices, contact the LDAP vendor.

Best Practice 5) LDAP Optimization

- Use LDAP servers that have fast WAN or LAN connections.
- Load balance LDAP servers to alleviate CPU, memory, and network pressure.
- Verify that LDAP servers have service records (SRVs) in DNS. Microsoft Active Directory does this by default for all LDAP servers that are also domain controllers.
- When the LDAP server database is extremely large, employ DN filtering through the base, user, group, and netgroup DN settings. Keep in mind that large is a subjective term and depends on factors such as network, LDAP server size, LDAP server load, number of objects, and so on.
- Searching LDAP at a lower level of the folder structure speeds up queries.
- If possible, try reducing the number of objects by deleting unused users, groups, and so on.
- If attempting to use multiple domains in an Active Directory forest for querying UNIX objects in LDAP, global catalog LDAP searches are required. LDAP referrals are currently not supported in ONTAP.
- Verify that all LDAP servers contain accurate and complete information in each object's schema attributes. For instance, every user should have a GID number assigned.
- Verify that all LDAP servers have a consistent copy of the schema. Active Directory does this by default on a 15-minute replication interval.
- Monitor the server's CPU, memory usage, and so on so that the server is not overworked.
- Remove slow or misbehaving LDAP servers from the client configuration as soon as possible to correct any issues.
- Always remove LDAP servers undergoing maintenance from the configuration.

Note: For details about LDAP referrals, global catalog searches, and other LDAP configurations, see TR-4073: Secure Unified Authentication.

Local Files

This section covers local files for use as a name service. In the ONTAP operating system, local files are entries in the replicated database (such as `unix-users`, `unix-groups`) that replace flat files seen in other operating systems, including Data ONTAP 7-mode (such as `/etc/passwd`, `/etc/group`), which allows a cluster to have current information about all member nodes.

Line Length Limitations

Netgroup files in the ONTAP operating system have limitations for line lengths and the number of nested netgroups. This information is covered in the limits section of the netgroup best practices.

Typo Handling

When using the `-load-from-uri` functionality in the ONTAP operating system to import netgroups into the cluster locally, great care must be taken so that no typos exist in the file to prevent access issues. Starting in Data ONTAP 8.2.3, the system checks the file for you prior to uploading and warns about potential typo errors. Typos in netgroup files can cause access issues, such as denying access to clients that should have access. This can affect local and remote netgroup host resolution.

Loading Netgroups from URI

When loading netgroups from URI to a local file on the cluster, the netgroup caches are not flushed out automatically. As a result, if a host name or IP is already in the cache (positive or negative) and the

netgroup file is changed to reflect that particular host name or IP, the change does not take effect until the cache is manually flushed or the TTL for the entry is expired. Although the global cache does limit the need for querying external netgroup sources when bringing up nodes or migrating LIFs, manually flushing caches requires repopulation, which could take a while and/or create a situation in which a flood of requests eats up resources and prevents access. Therefore, it is best to make this sort of change in a maintenance window. For more information about which caches store netgroup information, see the cache tunables section of this document. Remember that there are file size limits for loading files from URI.

Local File Sync Issues

In rare cases, local file entries might be out of sync with entries in the cluster's RDB. For instance, if a netgroup file was loaded recently and something happened during the load of the entries, then the netgroups in the cluster may not properly represent what was in the local file that was loaded.

Each time a file is loaded, it gets assigned a file version. Each time RDB is updated, it gets assigned the same version number as the file. To check if the versions are in sync, use the following command at diag privilege:

```
cluster::*> name-service file-version ?
  (vserver services name-service file-version show)
  show                        *Display the DB and file version
```

To rectify file version mismatches, use the following command in diag privilege:

```
cluster::*> name-service repair-configs ?
  (vserver services name-service repair-configs)
-node <nodename>                                                    *Node
-vserver <vserver name>                                             *Vserver (default: svm1)
-configuration {ns-switch|hosts|unix-user|unix-group|dns|netgroup|nis-domain|all} *Configuration
```

Netgroup.byhost

Netgroup.byhost entries are used to vastly speed up netgroup entry lookups by querying NIS and LDAP for the group membership by host rather than downloading the entire netgroup. The name service source, either NIS or LDAP server, creates a table of netgroups a particular host is a part of and that is automatically maintained on membership changes by the server. This allows the cluster to avoid needing to query every entry in a netgroup for access and instead allowing the name server to efficiently look up the single host. In large environments with netgroups that have many entries, this can drastically speed up the time for lookups and avoid access issues caused by timeouts on queries. Support for netgroup.byhost was added to ONTAP 8.2.3.

Note: Netgroup.byhost is enabled by default for NIS servers. No configuration change is needed. Netgroup.byhost in LDAP may require configuration changes, depending on the existing schema

Best Practice 6) Netgroup.byhost Considerations

When using netgroup.byhost, the following must be in place to achieve desired access results for hosts:

- Forward and reverse DNS records for host names
- Host triple entry in netgroup file (for example, host,,)
- Netgroup specification for the host's netgroup.byhost entry

Note: NetApp highly recommends netgroup.byhost functionality for environments with large (>1,000 members) netgroups.

Enabling netgroup.byhost Support for LDAP in the ONTAP Operating System

Netgroup.byhost support is not enabled by default in the ONTAP operating system. Several options in the LDAP client configuration would need to be modified:

```
-is-netgroup-byhost-enabled [true]
-netgroup-byhost-dn [DN with netgroup.byhost entries] (optional)
-netgroup-byhost-scope [base|onelevel|subtree]
```

DN and scope are used to specify the filters desired for netgroup.byhost functionality. For more information, see the administration guides for your release of ONTAP. For examples of this, see TR-4073: Secure Unified Authentication.

How Netgroup Processing Works

The following describes how netgroup operations work in the ONTAP operating system at a high level.

1. First, a mount request arrives from an IP address.
2. A reverse lookup is performed by using the SVM's DNS configuration to retrieve the FQDN.

   Note: A PTR record is highly recommended for all hosts that use NAS. PTR records enable CIFS to leverage Kerberos and allow export policies and rules to be resolved properly and efficiently.
3. After the FQDN (for example, *hostname.domainname.com*) is retrieved, the netgroup.byhost cache is searched for an entry. If there is a valid non-expired entry in the cache, then that result is used.
4. Next the netgroup.byname cache is consulted. If there is a valid non-expired entry for the netgroup that is being searched, then that result is used.
5. ONTAP consults the ns-switch sources in order. If netgroup.byhost is enabled for the source, then perform step 5a on the source, if not perform step 5b. (Netgroup.byhost is not available for files)
   a. Database (LDAP or NIS) is searched with the following, in order. If the result is found, it it put into the netgroup ip-to-hostname cache. If not found, then move onto step 5b.
      i. *hostname.domainname.com*.*
      ii. *hostname.** (if the NFS setting `netgroup-dns-domain-search` is enabled and the DNS domain search is listed properly in the DNS configuration)
      iii. IP address

      Note: If reverse lookup fails, we go directly to this step

      iv. Wildcard search: *.* (last resort)
   b. Database (LDAP, NIS, or files) full netgroup membership is requested and put into netgroup.byname cache. The result is then searched in the same manner as netgroup.byhost for the client. The result is then cached into the netgroup.byhost cache for faster future access

6 Caching in ONTAP

6.1 Global Name Services Cache

ONTAP 9.3 offers a new caching mechanism that moves name service caches out of memory and into a persistent cache that is replicated asynchronously between all nodes in the cluster. This provides more reliability and resilience in the event of failovers, as well as offering higher limits for name service entries due to being cached on disk rather than in node memory.

The name service cache is enabled by default. If legacy cache commands are attempted in ONTAP 9.3 with name service caching enabled, an error will occur, such as the following:

```
Error: show failed: As name service caching is enabled, "Netgroups" caches no longer exist. Use
the command "vserver services name-service cache netgroups members show" (advanced privilege
level) to view the corresponding name service cache entries.
```

The name service caches are controlled in a centralized location, below the `name-service cache` command set. This provides easier cache management, from configuring caches to clearing stale entries.

The global name service cache can be disabled for individual caches using `vserver services name-service cache` commands in **advanced** privilege, but it is not recommended to do so. For more detailed information, please see later sections in this document.

ONTAP also offers the additional benefit of using the caches while external name services are unavailable. If there is an entry in the cache, regardless if the entry's TTL is expired or not, ONTAP will use that cache entry when external name services servers cannot be reached, thereby providing continued access to data served by the SVM.

Hosts Cache

There are two individual host caches; forward-lookup and reverse-lookup but the hosts cache settings are controlled as a whole. When a record is retrieved from DNS, the TTL of that record will be used for the cache TTL, otherwise, the default TTL in the host cache settings will be used (24 hours). The default for negative entries (host not found) is 60 seconds. Changing DNS settings does not affect the cache contents in any way.

Note: The `network ping` command does not use the name services hosts cache when it needs to lookup a hostname.

User and Group Cache

The user and group caches consist of three categories; passwd (user), group and group membership.

Note: Cluster RBAC access does no use the any of the caches

Passwd (User) Cache

User cache consists of two caches, passwd and passwd-by-uid. The caches only cache the name, uid and gid aspects of the user data to conserve space since the other data such as homedir and shell are irrelevant for NAS access. When an entry is placed in the passwd cache, the corresponding entry is created in the passwd-by-uid cache. By the same token, when an entry is deleted from one cache, the corresponding entry will be deleted from the other cache. If you have an environment where there are disjointed username to uid mappings, there is an option to disable this behavior.

Group Cache

Like the passwd cache, the group cache consists of two caches, group and group-by-gid. When an entry is placed in the group cache, the corresponding entry is created in the group-by-gid cache. By the same token, when an entry is deleted from one cache, the corresponding entry will be deleted from the other

cache. The full group membership is not cached to conseve space and is not necessary for NAS data access, therefore only the group name and gid are cached. If you have an environment where there are disjointed group name to gid mappings, there is an option to disable this behavior.

Group Membership Cache

ONTAP also stores group membership information. The In file and NIS environments, there is no efficient way to gather a list of groups a particular user is a member of, so for these environments ONTAP utilizes the group membership cache to provide these efficiencies but LDAP does also use this cache to store the group membership lookup results.. The group membership cache consists of a single cache and contains a list of groups a user is a member of. For NIS we have the `nis-domain group-database` command set to help control this behavior. ONTAP gathers the group information from NIS and creates a user-to-group-membership map every 24 hours. This is consulted when group-membership is looked up.

Netgroup Cache

Beginning in ONTAP 9.3, the various netgroup caches have been consolidated into 2 caches; a netgroup.byhost and a netgroup.byname cache. The netgroup.byhost cache is the first cache consulted for the netgroups a host is a part of. Next, if this information is not available, then the query reverts to gathering the full netgroup members and comparing that to the host. If the information is not in the cache, then the same process is performed against the netgroup ns-switch sources. If a host requesting access via a netgroup is found via the netgroup membership lookup process, that ip-to-netgroup mapping is always added to the netgroup.byhost cache for faster future access. This also leads to needing a lower TTL for the members cache so that changes in netgroup membership can be reflected in the ONTAP caches within the TTL timeframe. For those environments that have volitle, dynamic netgroups, please see the Netgroup Best Practices section in this document for more information on how to best configure these environments.

Viewing cache entries

Each of the above came service caches and be viewed. This can be used to confirm whether or not expected results are gotten from name services servers. Each cache has its own individual options that you can use to filter the results of the cache to find what you are looking for. In order to view the cache, the `name-services cache <cache> <subcache> show` command is used.

Caches are unique per vserver, so it is suggested to view caches on a per-vserver basis. Below are some examples of the caches and the options.

```
cluster::*> name-service cache hosts forward-lookup show  ?
  (vserver services name-service cache hosts forward-lookup show)
  [ -instance | -fields <fieldname>, ... ]
  [ -vserver <vserver name> ]                                          *Vserver
  [[-host] <text>]                                                     *Hostname
  [[-protocol] {Any|ICMP|TCP|UDP}]                                     *Protocol
(default: *)
  [[-sock-type] {SOCK_ANY|SOCK_STREAM|SOCK_DGRAM|SOCK_RAW}]            *Sock Type
(default: *)
  [[-flags] {FLAG_NONE|AI_PASSIVE|AI_CANONNAME|AI_NUMERICHOST|AI_NUMERICSERV}]  *Flags (default:
*)
  [[-family] {Any|Ipv4|Ipv6}]                                          *Family (default:
*)
  [ -canonname <text> ]                                                *Canonical Name
  [ -ips <IP Address>, ... ]                                           *IP Addresses
  [ -ip-protocol {Any|ICMP|TCP|UDP}, ... ]                             *Protocol
  [ -ip-sock-type {SOCK_ANY|SOCK_STREAM|SOCK_DGRAM|SOCK_RAW}, ... ]    *Sock Type
  [ -ip-family {Any|Ipv4|Ipv6}, ... ]                                  *Family
  [ -ip-addr-length <integer>, ... ]                                   *Length
  [ -source {none|files|dns|nis|ldap|netgrp_byname} ]                  *Source of the
Entry
  [ -create-time <"MM/DD/YYYY HH:MM:SS"> ]                             *Create Time
  [ -ttl <integer> ]                                                   *DNS TTL
```



```
cluster::*> name-service cache unix-user user-by-id show
  (vserver services name-service cache unix-user user-by-id show)
Vserver    UID         Name         GID            Source  Create Time
---------- ----------- ------------ -------------- ------- -----------
svm1       0           root         1              files   1/25/2018 15:07:13
svm2
           0           root         1              files   1/24/2018 21:59:47
2 entries were displayed.
If there are no entries in a particular cache, the following message will be shown:
cluster::*> name-service cache netgroups members show
  (vserver services name-service cache netgroups members show)
This table is currently empty.
```

Clearing Cache Entries

The following steps show how to clear out cache entries manually out of the various name service caches. This should be done only when needed, such as when loading a new netgroup file from URI or when troubleshooting access denied issues.

With the introduction of the global name service cache, clearing netgroup caches has been simplified and is now centrally managed from the vserver name-service cache command set. Cache entries can be cleared individually, as well as the entire cache being purged. Keep in mind that clearing a cache means that it has to be repopulated, so some delays may occur if caches are cleared, especially when cleared en masse.

To clear the hosts cache:

```
cluster::*> name-service cache hosts reverse-lookup ?
  (vserver services name-service cache hosts reverse-lookup)
  delete                      *Delete an entry
  delete-all                  *Delete all the entries for the vserver
```

```
cluster::*> name-service cache hosts forward-lookup ?
  (vserver services name-service cache hosts forward-lookup)
  delete                      *Delete an entry
  delete-all                  *Delete all the entries for the vserver
```

To clear the ip-to-netgroup cache:

```
cluster ::*> name-service cache netgroups ip-to-netgroup ?
  (vserver services name-service cache netgroups ip-to-netgroup)
  delete                      *Delete netgroup.byhost cache entry
  delete-all                  *Delete all the entries for the vserver
```

To clear the netgroup member cache:

```
cluster::*> name-service cache netgroups members ?
  (vserver services name-service cache netgroups members)
  delete                      *Delete netgroup cache entry
  delete-all                  *Delete all the entries for the vserver
```

To clear the user cache:

```
cluster::*> name-service cache unix-user user-by-id ?
  (vserver services name-service cache unix-user user-by-id)
  delete                      *Delete an entry
  delete-all                  *Delete all the entries for the vserver
```

```
cluster::*> name-service cache unix-user user-by-name ?
  (vserver services name-service cache unix-user user-by-name)
  delete                      *Delete an entry
  delete-all                  *Delete all the entries for the vserver
```

To clear the group cache:

```
cluster::*> name-service cache unix-group group-by-gid ?
  (vserver services name-service cache unix-group group-by-gid)
  delete                      *Delete an entry
  delete-all                  *Delete all the entries for the vserver
```

```
cluster::*> name-service cache unix-group group-by-name ?
  (vserver services name-service cache unix-group group-by-name)
  delete                      *Delete an entry
  delete-all                  *Delete all the entries for the vserver
```

To clear the group-membership cache:

```
cluster::*> name-service cache group-membership ?
  (vserver services name-service cache group-membership)
  delete                      *Delete an entry
  delete-all                  *Delete all the entries for the vserver
```

Cache Tunables

The following section covers the TTLs and other tunables for name services caches in ONTAP. All the name service tunables can be modified with the command `name-service cache <hosts/unix-user/unix-group/group-membership/netgroup> settings modify`. Below is an example for the hosts cache:

```
cluster::*> name-service cache hosts settings modify ?
  (vserver services name-service cache hosts settings modify)
    -vserver <vserver name>                                  *Vserver
  [[-is-enabled] {true|false}]                               *Is Cache Enabled?
  [ -is-negative-cache-enabled {true|false} ]                *Is Negative Cache Enabled?
  [ -ttl <[<integer>h][<integer>m][<integer>s]> ]            *Time to Live
  [ -negative-ttl <[<integer>h][<integer>m][<integer>s]> ]   *Negative Time to Live
```

TTLs

Each cache has its own tunables for TTLs. TTL is the amount of time an entry can live in cache before it needs looked up again against the name services sources. As mentioned before, if an entry is expired and ONTAP cannot contact any of the external sources for a particular entry, then the cached value will still be used so that access is not abruptly ended due to network errors or other unforeseen problems with name services servers. TTLs can be modified for each individual cache by using the command `name-service cache <hosts/unix-user/unix-group/group-membership/netgroup> settings modify`. Most of the caches have similar settings that can be changed. Below is a table of the default values for each of the caches:

Table 6) Global Name Service Cache TTL Defaults

Cache	Default TTL	Default Negative TTL
Hosts	24h	1m
Unix-user	24h	1m
Unix-group	24h	1m

Cache	Default TTL	Default Negative TTL
Group-membership	24h	N/A
Netgroup.byhost	24h	1m
Netgroup.byname	24h	N/A

Note: For Group-membership and Netgroup.byname caches, there are no negative entries. If the group or netgroup does not exist, no caching of that failure will occur.

Effect on caches by configuration changes

There are various reasons that configuration changes should affect when caches get flushed or reloaded. Below is a table of various configuration changes and what affects they have on caches.

Configuration	Action	Cache impact
DNS Hosts	dns hosts create	Delete the negative cache entry from the forward-lookup and reverse-lookup caches for the host, IP and aliases getting created
	dns hosts modify	Delete the positive cache entry from the forward-lookup and reverse-lookup caches for the IP and older host/aliases. Delete the negative cache entry from the forward-lookup cache for the newly configured host and aliases
	dns hosts delete	Delete the positive cache entry from the forward-lookup and reverse-lookup caches for the host,IP and aliases.
UNIX user	unix-user create	Delete the negative cache entry from the by-name and by-id caches for the new username and uid getting created.
	unix-user modify (Only uid can be modified)	Delete the negative cache entry from the by-id cache for the new uid getting created. Delete the positive cache entry from the by-name and by-id caches for the username and old uid.
	unix-user delete	Delete the positive cache entry from the by-name and by-id caches for the new username and uid getting created.

Configuration	Action	Cache impact
	load-from-uri (normal mode)	New user getting added - Follow 'unix-user create' behavior. uid getting modified for an existing user - Follow 'unix-user modify' behavior. username getting modified for an existing uid - Follow 'unix-user delete' behavior for the username getting changed. Follow 'unix-user create' behavior for the new username.
	load-from-uri (file-only mode)	No impact. Caches will not be modified in this case.
UNIX group	unix-group create	Delete the negative cache entry from the by-name and by-id caches for the new groupname and gid getting created.
	unix-group modify (only gid can be modified)	Delete the negative cache entry from the by-id cache for the new gid getting created. Delete the positive cache entry from the by-name and by-id caches for the groupname and old gid.
	unix-group delete	Delete the positive cache entry from the by-name and by-id caches for the new username and user-id getting created.
	unix-group adduser	Delete the entry for the user from the group-membership cache.
	unix-group addusers	Delete the entry for all users from the group-membership cache.
	load-from-uri (normal mode)	New group getting created - Follow 'unix-group create' behavior. gid getting modified for an existing group - Follow 'unix-group modify' behavior. Group getting removed - Follow 'unix-group delete' behavior. User being added to a new group - Follow 'unix-group adduser' behavior. User being removed from a group - Follow 'unix-group deluser' behavior.
	load-from-uri (file-only mode)	No impact. Caches will not be modified in this case.

Configuration	Action	Cache impact
Netgroups	netgroup load	No impact. Caches will not be modified in this case.
NS switch	ns-switch modify	No impact. Caches will not be modified in this case.
DNS config	dns modify	No impact. Caches will not be modified in this case.
LDAP config	ldap client modify	No impact. Caches will not be modified in this case.

6.2 NAS Layer Caches

The following describes the layers used at the NAS layer in the ONTAP operating system. For more information about the NAS layer, see the corresponding section in this document. NAS Layer caches are held at the node level and unlike the global name service caches, are not replicated within the cluster. Cache changes are made at the diag privilege level. As with any diag command, use caution.

Export-Policy Caches

Export caches have information on whether a client that requested mount access to a certain export was granted access or not based on the export policy and rules applied to that volume or qtree. This cache is managed with `vserver export-policy access-cache config` commands. The following describes the attributes of the caches.

- TTL for Positive Entries: This is the TTL for positive entries in the access cache. During client access, if the TTL for the access cache entry that is allowing access has expired, that access cache entry will be refreshed. While the refresh is in progress, client access will be evaluated with the existing information in the access cache entry.
- TTL for Negative Entries: This is the TTL for negative entries in the access cache. During client access, if the TTL for the access cache entry that is denying access has expired, that access cache entry will be refreshed. While the refresh is in progress, client access will be evaluated with the existing information in the access cache entry.
- TTL for Entries with Failure: This is the TTL for access cache entries for which a failure was encountered while trying to get matching rules.
- Harvest Timeout: If Data ONTAP does not use an entry that is stored in the access cache for this period of time, it deletes the entry..

Flushing Export Policy Caches

Export policy caches are flushed by making changes to export policy rules. In addition, ONTAP offers a set of commands to allow manual flushing of export caches without needing to change export policies. This is done on a per-SVM, per-node, per-export policy basis. An individual IP can also be flushed from the cache. The command is a diag level command.

```
cluster::*> diag exports nblade access-cache flush
    -vserver <vserver name>   *Vserver
   [-node] <nodename>         *Node
   [-policy] <text>           *Export Policy Name
  [ -address <IP Address> ]   *IP Address
```

Note: Never flush caches when name service servers are unavailable or experiencing higher than normal latencies. Doing so could cause client disruptions as the caches attempt to repopulate. For information about how to evaluate name service response times, see the section in this document regarding name service statistics.

Credential Caches

When an NFS user requests access to NFS exports on the storage system, ONTAP must retrieve the user credentials either from external name servers or from local files to authenticate the user. ONTAP then stores these credentials in an internal credential cache for later reference. Understanding how the NFS credential caches works enables you to handle potential performance and access issues. These NAS layer credential caches can be flushed and viewed with the `diag nblade credentials` command, which is at the **diagnostic** privilege level.

```
cluster::*> diag nblade credentials
    count flush show
```

For the NFS protocol, the cache timeout values can be modified. NetApp does not recommend modifying these times unless required. Need for modifying these values would be determined through support cases. If it is necessary to modify these caches, then the values can be adjusted using the following:

```
cluster::*> nfs modify -vserver svm1 -cached-?
  [ -cached-cred-positive-ttl {60000..604800000} ]  *Time To Live Value (in msecs) of a Positive
Cached Credential
  [ -cached-cred-negative-ttl {60000..604800000} ]  *Time To Live Value (in msecs) of a Negative
Cached Credential
  [ -cached-transient-err-ttl {30000..300000} ]     *Time To Live Value (in msecs) of a Cached
Entry for a Transient Error
```

Individual users can be viewed in these caches if desired:

```
cluster::*> diag nblade credentials show -node node-01 -vserver svm1 -unix-user-id 1301

Getting credential handles.
1 handles found....

Getting cred 0 for user.
          Global Virtual Server: 10
          Cred Store Uniquifier: 1
Cifs SuperUser Table Generation: 0
               Locked Ref Count: 0
                     Info Flags: 1
          Alternative Key Count: 0
        Additional Buffer Count: 0
                 Creation Time: 4214881195 ms
        Time Since Last Refresh: 20539 ms
Windows Creds:
        Flags: 0
        Primary Group: S-0-0
Unix Creds:
        Flags: 0
        Domain ID: 0
        Uid: 1301
        Gid: 1201
        Additional Gids:
                Gid 0: 1201
                Gid 1: 1203
                Gid 2: 1206
```

Additionally, manual flushing of the cache should be done only when necessary, because it can cause latency to access and outages as the cache gets repopulated. The credential cache also comes into play when using the `-auth-sys-extended-groups` option. When extended groups are enabled, the cluster will keep track of the list of groups a user is a member of via the group-membership cache.

NFS/Name Service Database (NSDB) Caches

In addition to NAS layer caches, ONTAP has the concept of NFS caches when NFSv4 ID to name mappings are involved. Rather than constantly needing to reach out to name service servers (such as NIS or LDAP) and fetch credentials, the NSDB cache will keep NFS credentials for 30 minutes. The

NSDB cache can also be cleared starting in ONTAP 8.3.1 with the **diagnostic** privilege command `diag nblade nfs nsdb-cache clear`. Starting in ONTAP 9.0, the cache can be viewed with `diag nblade nfs nsdb-cache show`.

```
cluster::> set diag
cluster::*> diag nsdb-cache show -node node-03 -vserver svm1 -unix-user-name nfs_user
  (diag nblade nfs nsdb-cache show)

           Node: node-03
        Vserver: svm1
  Unix user name: nfs_user
   Creation time: 2146204100
Last Access time: 2146261100
  Number of hits: 19
```

SecD Caches

SecD is another area in the ONTAP operating system that caches information retrieved from name services. Beginning in ONTAP 9.3, many of the SecD caches previously available have been deprecated in favor of the Global Name Service caches. Most of the remaining SecD caches deal with CIFS/SMB access. SecD caches are managed at a diag privilege level. Most of these caches age out after 24 hours.

The following caches are available for configuration and querying:

```
cluster::*> diag secd cache show-config -cache-name ? -node node-02
  ad-to-netbios-domain
  netbios-to-ad-domain
  ems-delivery
  log-duplicate
  name-to-sid
  sid-to-name
  schannel-key
  username-to-creds
  ad-sid-to-local-membership
  nis-group-membership
  groupname-to-info
  groupid-to-name
  userid-to-name
  username-to-info
  ldap-netgroupname-to-members
  ldap-groupname-to-info-batch
  ldap-username-to-info-batch
  name-mapping-windows-to-unix
  user-realmname-to-short-name
```

Note: Any time a cache is modified on a node, it should be modified on every node in a cluster.

Modifying caches can alter NAS behavior. More aggressive caches can mean more load on the system for cache refreshes. Less aggressive caches can lead to inconsistencies in name service requests (that is, hosts removed from the netgroup remain in cache until flushed).

To adjust a secd cache, use the following command:

```
cluster::> set diag
cluster::*> diag secd cache set-config -node [nodename] -cache-name [cache] -lifetime [in
seconds]
Example of modifying a cache:
cluster::*> diag secd cache set-config -node node-01 -cache-name sid-to-name -life-time 3600
```

Example of viewing the cache configuration:

```
cluster::*> diag secd cache show-config -node node-01 -cache-name sid-to-name
Current Entries: 0
    Max Entries: 2500
 Entry Lifetime: 3600
```

Note: SecD cache changes are not persistent across reboots.

Clearing SecD Caches

It is possible to clear SecD caches with the diag secd cache clear command.

```
cluster::*> diag secd cache clear -node node-02 -vserver svm1 -cache-name ?
  ad-to-netbios-domain
  netbios-to-ad-domain
  ems-delivery
  log-duplicate
  name-to-sid
  sid-to-name
  schannel-key
  ad-sid-to-local-membership
  nis-group-membership
  name-mapping-windows-to-unix
  user-realmname-to-short-name
```

Clearing SecD Kerberos Credentials

In the event that ONTAP Kerberos credentials become stale (such as if the domain time skew is outside of the 5 minute range for CIFS Kerberos authentication), it is possible to clear the Kerberos credential cache for one or more SVMs.

```
cluster::*> diag secd
      *Vserver cache clear-krb-creds ?
   [-node] <nodename>          *Node
  [ -vserver <vserver> ]
```

Table 7) SecD Cache Ages

Cache Name	Default Refresh Time	Recommended Refresh Time
ad-to-netbios-domain	0	0
ad-sid-to-local-membership	86400	86400
ems-delivery	300	300
lif-bad-route-to-target	14400	14400
log-duplicate	300	300
name-to-sid	86400	86400
netbios-to-ad-domain	0	0
schannel-key	0	0
sid-to-name	86400	86400

Note: NetApp recommends not changing SecD caches without guidance from NetApp Support

7 Best Practices

7.1 Name Service (ns-switch) and Name Mapping (nm-switch)

The following section covers best practices for name service (ns-switch) and name mapping (nm-switch) configuration in ONTAP operating systems.

Best Practice 7) Nm-switch and ns-switch configuration
Avoid configuring nm-switch or ns-switch to use external name services if the services are not actually in use. For example, if you aren't using NIS to serve user names, leave NIS out of ns-switch passwd and group databases. If you aren't using LDAP for asymmetric name mapping rules, then don't include LDAP in nm-switch or the namemap database in ns-switch.

7.2 Name Server Configuration Best Practices

The following section covers name server best practices for use with ONTAP.

What Is a Name Server?

A name server is any external server that provides a database for name services. Name servers can include, but are not limited to:

- DNS
- LDAP
- NIS

Name Server Transport Information

Because name servers are external servers, they leverage standard network transport protocols and are subject to the same issues to which any protocol running over an Ethernet network is subject, such as latency, retransmissions, and so on.

UDP or TCP?

Some name servers can leverage both TCP and/or UDP for network transport, such as DNS. DNS uses UDP by default. However, in the ONTAP operating system, if a DNS response packet is greater than 512 bytes, then TCP is used to retrieve the remaining information.

Service	Transport
NIS	UDP with the management gateway (mgwd) TCP with the security daemon (SecD)
LDAP	TCP
DNS	UDP by default; falls back on TCP

UDP is short for User Datagram Protocol and is generally regarded as the lesser of the two protocols because of its unreliability and limitations. TCP is short for Transmission Control Protocol and is used when a network connection requires a method to guarantee that a packet arrived between two networked entities. Most name servers use TCP for their transport. However, DNS servers still make use of UDP calls for host name lookups because it does provide some advantages over TCP, such as speed and lack of a need for a complete network conversation (that is, no retransmissions). NIS servers also still make use of the UDP calls because of the same advantages over TCP.

Name server connectivity information

Often times enterprises may place their name services servers in segregated networks from their regular user data networks. Security is paramount in these types of configurations but can impact the availability of the data being served if the name service servers become unreachable. When ONTAP needs to make name service calls, they originate from the node that the inbound NAS (NFS or SMB) packet was received on. If the LIF cannot reach the name services servers from that node, then the NAS connectivity could fail. Consider the following example:

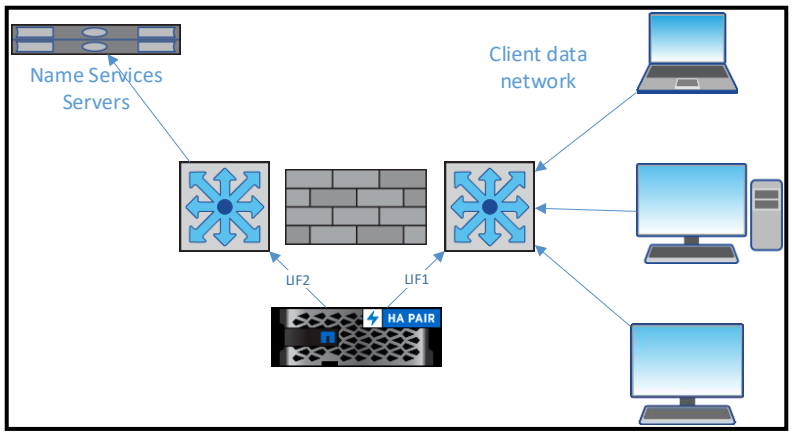
Name services servers are in the 192.0.2.0/24 network

Regular client data network is via the 203.0.113.0/24 network.

No internal route exists that the 203.0.113.0/24 network can get to the 192.0.2.0/24 network.

ONTAP has an SVM with a LIF on the 203.0.113.0/24 network for clients to access.

ONTAP needs to have a LIF on the same SVM in the 192.0.2.0/24 network in order to communicate properly with the name service servers.



In the above scenario, there is no need to create specific routes for name services since the two networks are separated by classes. If the case is that the name services also sit in the 203.0.113.0/24 network, but only in a specific range, a route may need to be created in the SVM if the ONTAP LIF and the name services servers sit in separate layer 2 broadcast domains. This is to ensure that the name services traffic originated by ONTAP will egress the correct interface to reach the name services server.

Best Practice 8) Name Services Connectivity
It is a best practice to have a LIF on each node that can communicate with name services servers.

LAN Versus WAN

Enterprise NAS environments often have sites at locations across the globe. In many cases, these sites are smaller and do not have resources such as name service servers local to the site. In these scenarios, it is important to make certain that WAN latencies to name service servers do not exceed 2 seconds (2000ms) for name services. For information about viewing latencies for name services, see the section in this document covering name service statistics.

Best Practice 9) Name Services over WAN
It is a best practice to determine that a name service server is local to any site where NAS access is desired or at least at a site close enough to prevent latencies over a WAN from exceeding 2 seconds.

Note: If using a WAN is required, enable latency monitoring for best results.

General Name Server Best Practices

The following provides a general list of name server best practices for the best possible resiliency and performance for name servers.

Best Practice 10) General Name Service Best Practices

- Always configure multiple servers for redundancy and load balancing.
- Verify that all name servers are in sync.
- Verify that forward and reverse lookup records exist for all hosts, including name servers.
- Verify that name servers are not overloaded (CPU, RAM, network connections, and so on) and can handle the load generated by ONTAP storage needs.
- Avoid using virtual machines for production name servers when possible.
- If using virtual machines for name servers, do not host them on NFS datastores that are dependent on those name servers.
- Never specify name services on SVMs (ns-switch, nm-switch) that are not configured with functional name service servers and configurations.
- For best results, use the name services cache in ONTAP 9.3.

Note: Misconfiguration of name services on SVMs can result in hangs, particularly with NFSv4.x, because the SVM attempts to use the services in the list to resolve nonexistent UID/GID mappings. If no servers are configured but external name services are specified (such as LDAP or NIS), requests for name lookups run indefinitely, and commands such as `ls` appear to hang.

LDAP as a Name Services Source

When configuring LDAP as a name services source, you can create multiple LDAP configurations and they can be associated with a particular SVM or the Cluster. When the settings seen with the below command are the same across multiple SVMs, then the LDAP client should be created to the cluster and shared across SVMs. This provides ease of management of a single LDAP configuration across multiple SVMs.

```
cluster::*> ldap client modify ?
  [ -vserver <vserver name> ]                          Vserver (default: cluster)
   [-client-config] <text (size 1..32)>                Client Configuration Name
  { [[-ldap-servers] <text>, ...]                      LDAP Server List
  | [ -ad-domain <TextNoCase> ]                        Active Directory Domain
    [ -preferred-ad-servers <IP Address>, ... ]        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
  [ -query-timeout {0..10} ]                           Query Timeout (sec)
  [ -min-bind-level {anonymous|simple|sasl} ]          Minimum Bind Authentication Level
  [ -bind-dn <LDAP DN> ]                               Bind DN (User)
  [ -base-dn <LDAP DN> ]                               Base DN
  [ -base-scope {base|onelevel|subtree} ]              Base Search Scope
  [ -user-dn <LDAP DN> ]                               *User DN
  [ -user-scope {base|onelevel|subtree} ]              *User Search Scope
  [ -group-dn <LDAP DN> ]                              *Group DN
  [ -group-scope {base|onelevel|subtree} ]             *Group Search Scope
  [ -netgroup-dn <LDAP DN> ]                           *Netgroup DN
  [ -netgroup-scope {base|onelevel|subtree} ]          *Netgroup Search Scope
  [ -use-start-tls {true|false} ]                      Use start-tls Over LDAP Connections
  [ -is-netgroup-byhost-enabled {true|false} ]         *Enable Netgroup-By-Host Lookup
  [ -netgroup-byhost-dn <LDAP DN> ]                    *Netgroup-By-Host DN
  [ -netgroup-byhost-scope {base|onelevel|subtree} ]   *Netgroup-By-Host Scope
  [ -session-security {none|sign|seal} ]               Client Session Security
  [ -skip-config-validation [true] ]                   Skip Configuration Validation
```

Best Practice 11) LDAP Client Configurations

As a best practice, a single LDAP client configuration should be created against the cluster SVM and shared across multiple SVMs for ease of management.

Name Service Servers Hosted in Virtual Machines

In some cases, name servers (for example, DNS and LDAP) are hosted on virtual machines running in virtualized environments such as ESXi, Hyper-V, and so on. This configuration is perfectly fine, but you should also consider the following:

- Name servers on VMs should always have dedicated resources (RAM, CPU, and so on) allocated to their operating systems so that the name servers can respond appropriately.
- Name servers on datastores should not have interdependencies on the devices that they intend to service. For example, if a VM that serves DNS is hosted on an NFS datastore that requires DNS for proper export policy client resolution, that datastore should not be hosted on the same storage that has the dependency on the DNS server.
- For NFS datastores that host VMs with critical servers, such as name servers, it is a best practice to employ a dedicated export policy rule that uses local host entries or IP addresses to remove DNS dependency on those exports.

Best Practice 12) Virtualized Name Services

As a best practice, verify that the name services servers configured for the SVMs in the ONTAP cluster are not dependant on hypervisor datastores that are hosted by an SVM in the cluster. If these name service servers are the optimal ones, ensure that there is at least one entry for a name server that is either a physical machine or where the hypervisor datastore is not hosted by this ONTAP cluster.

7.3 Host Name Resolution Best Practices

The following section covers host name resolution best practices in the ONTAP operating system.

DNS Load Balancers

In some environments, DNS Load Balancers are a common way of ensuring the resiliency of DNS services. Certain load balancing configurations can cause the response of the DNS query to be from a different IP address than the one that ONTAP queried. This scenario is rejected by ONTAP for security reasons but can be modified if necessary. If this is the configuration in your environment, use the following command to change ONTAP to allow for this behavior:

```
cluster::*> dns modify -vserver svm1 -require-source-address-match false
```

Forward and Reverse Lookup Names

When configuring name resolution, it is always best practice to have both forward-lookup (A records) and reverse lookups (PTR records) for each hostname configured that match. In export policy processing, ONTAP uses forward lookups to evaluate the export rules against the incoming client. In netgroup processing, the PTR record is evaluated when determining netgroup membership. There are certain cases where aliases (CNAME records) are convenient in the environment but in some cases, cannot be used for granting access to NAS resources. When using CNAMEs, consider what method of authentication will be used for NAS access.

Best Practice 13) Host forward and reverse records

- Ensure that all hosts have forward (A record) and reverse (PTR record) in the DNS system that match. For information about how to verify this, see the appendix in this document.
- Where possible, do not use CNAMEs for granting access to NAS resources.

Multiple DNS Search Domains

In the ONTAP operating system, it is possible to configure SVMs to use multiple DNS search domains for host name resolution. However, doing this can cause issues with export policy rules because DNS host name resolution can take a considerable amount of time traversing multiple DNS domains if the first name in the list is not valid for the host. When a hostname is unresolvable, ONTAP will retry the DNS request with each DNS suffix until it either exhausts the list with no results or gets an actual result from the DNS server.

Best Practice 14) Multiple DNS Search Domains

- Use only the search domains applicable to an environment in the configuration. If possible, use only one search domain.
- If multiple search domains are needed, verify that the most commonly used DNS search domain is listed first in the DNS configuration.
- Verify that all DNS servers properly forward to the DNS zones listed in the configuration to avoid failures in host name resolution.
- When possible, use fully qualified domain names (FQDNs) in netgroups, export policies, and so on so that the cluster does not try to resolve a host name with the search domain list.
- Verify that PTR/reverse lookup records exist in DNS. This is a requirement for fully functional export policy rule name resolution. For information about how to do this, see the appendix in this document.

Known DNS Issues

This is a list of some known DNS issues in ONTAP. This list is intended to help avoid scenarios that could cause problems, but is not comprehensive.

- ONTAP 9 introduces a new “bad” DNS caching mechanism. If a DNS server request experiences a timeout, the DNS server gets marked as “bad” for 10 minutes and is not used during this time period. The cache is flushed when DNS configuration is modified using “dns modify.” This timeout value is not modifiable.
- Currently, ONTAP supports only the use of DNS or local files for host name mappings. LDAP and NIS are not supported for host names.
- Local file host names for SVMs are supported only in versions of Data ONTAP 8.3 and later.
- If DNS records do not have both forward and reverse lookups, lookups for access in exports might fail.
- If using fully qualified domain names (FQDNs) in host names, netgroups, and so on, RFC-1535 states that it is best to append a dot (.) to the end of the FQDN to denote an “absolute rooted” FQDN. For example: hostname.example.com. For more information, see the section about rooted vs. nonrooted FQDNs.

Best Practice 15) General DNS and Host Name Resolution

- Use only relevant DNS search domains in DNS configurations for fast DNS lookups.
- Have multiple DNS servers with replicating databases (such as Active Directory DNS) to avoid single points of failure.
- Verify that local and/or the fastest DNS servers are listed first.
- Remove DNS servers undergoing maintenance to avoid issues with NAS clients.
- Verify that all DNS servers contain the same information.
- Only specify internal DNS servers as public DNS servers will not have the needed information to serve NAS data properly.

Note: IPv6 is disabled by default.

7.4 User and Group Best Practices

Name Mapping

Name mapping is used when accessing UNIX style volumes/qtrees via SMB and when accessing NTFS style volumes/qtrees via NFS. In most environments, it is possible to achieve a one-to-one mapping of usernames in both Windows AD and in Unix LDAP. The easiest way to achieve this is to use the same Active Directory domain to serve as CIFS authentication for the particular SVM and to configure the AD DCs as LDAP client servers in the SVM. There are some instances when a separate Unix LDAP implementation, such as RedHat's Identity Manager, is necessary. In those cases, you want to ensure that the usernames defined for the same user in the LDAP server and in the Windows AD environment use the same username to east the name mapping. If this cannot be achieved, the next-best thing to to employ a pattern where one can be transformed into the other so that the regex-compliant rule can be put into the name-mapping rules. Avoid at all costs custom name-mappings that have no pattern. If this is the case, leverage LDAP and the SVM's LDAP client schema's attributes to use asymmetric name-mapping for maximum manageability and scalability.

Best Practice 16) User and Group Name Mapping

Ensure usernames match between Windows and Unix environments for the quickest and most reliable name-mapping results for multi-protocol access.

Limits

In the ONTAP operating system, there are limits to the number of users and groups allowed locally on a system. Table 8 covers these limits.

Table 8) User and group limits in ONTAP; non-scaled mode

Limit	Value
Number of characters per user and group (that is, length of name)	64 characters
File size for -load-from-uri (unix-user, unix-group)	UNIX users: 2.5MB UNIX groups: 1MB **Note:** If the limit is exceeded, the load fails.

Limit	Value
Clusterwide local UNIX users and groups and members	UNIX users: 32,768 (default) 65,536 (maximum) UNIX groups and members: 32,768 (default) 65,536 (maximum)
UNIX groups single line (-load-from-uri)	32,768 characters
Name mapping rules	1,024 per SVM

Scaled Mode/File-Only Mode

Scaled Mode/File-Only Mode

Scaled mode/file-only mode for local users and groups beginning in ONTAP 9.1 allows storage administrators to expand the limits of local users and groups by enabling a **diagnostic**-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. Table 9 outlines those new limits. 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 done per cluster, adding an extra layer of complexity.

Best Practice 17) Using File Only Mode for Local Unix Users and Groups
Be sure to evaluate your options at length and make the appropriate decision for your environment and 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 Pracrice and Implementation Guide.

Table 9) User and group limits in ONTAP; scaled/file-only mode

Limit	Value
File size for -load-from-uri (unix-user, unix-group)	UNIX users: 10MB UNIX groups: 25MB **Note:** The limit can be exceeded by setting the option `-skip-file-size-check`, but larger file sizes have not been tested
Clusterwide local UNIX users and groups and members	Users: 400K Groups: 15k Group memberships: 3000k SVMs: 6

7.5 Netgroup Best Practices

Rooted vs Nonrooted FQDNs

When configuring netgroups either in LDAP, NIS or in local files, there are multiple ways to define the hostname in the triple. You can use a shortname, nonrooted FQDN or rooted FQDN. RFC-1535 states there is a difference in FQDNs depending on how the name is specified. From the RFC:

Current Domain Name Server clients are designed to ease the burden of remembering IP dotted quad addresses. As such they translate human-readable names into addresses and other resource records. Part of the translation process includes understanding and dealing with hostnames that are not fully qualified domain names (FQDNs).

An absolute "rooted" FQDN is of the format {name}{.} A non "rooted" domain name is of the format {name}

A domain name may have many parts and typically these include the host, domain, and type. Example: foobar.company.com or fooschool.university.edu.

In ONTAP, rooted FQDNs (FQDNs with the trailing dot) are handled in an efficient manner for scenarios where entries do not exist in DNS servers. When a rooted FQDN is used, the trailing dot is dropped (DNS only), and the cluster looks the FQDN up “as is” and only tries the lookup once. FQDNs are tried “as is” only if configured as rooted FQDNs (with the trailing dot). If the entry has no trailing dot and the FQDN is not found in DNS, the search domains are added to the FQDN, and DNS is queried with this combined name until all search domains are tried or a match is found.

In ONTAP, a nonrooted FQDN (any hostname entry with a dot in it) attempts to resolve “as is” first, and then the cluster attempts to append the search domains to the end of the FQDN. For example, a nonrooted FQDN might look like this:

```
hostname.example.com
```

If the search domains in the DNS configuration include “example.com” and others, then the retried lookup would look like this:

```
hostname.example.com.example.com
```

The cluster then cycles through all search domains, which adds unnecessary latencies to failed records. Using a rooted FQDN prevents this behavior.

When a hostname without a dot is encountered, the attempts to resolve are first tried by appending the search domains to the end of the hostname and then attempting to resolve it “as-is’.

In the event of a SERVFAIL error or DNS server timeout, the next server is tried.

Note: When a shortname is looked up “as-is” on certain DNS servers, the response is “SERVFAIL” and not “NXDOMAIN”. Windows DNS servers are one of the types of DNS servers that respond in this way. “NXDOMAIN” is an authoritative “name not found” answer and a “SERVFAIL” is considered a transient error. ONTAP will lookup that shortname “as-is” through all DNS servers configured for the SVM. This can add unneeded latencies to the lookups. Best practice is to either not use shortnames or prune shortnames that cannot be resolved from netgroups on a regular basis.

Best Practice 18) Netgroup hosts

It is always best practive to use a rooted FQDN for the host string of a netgroup triple.

Netgroups and DHCP/Dynamic DNS

In some scenarios, host names are granted IP address leases from servers running Dynamic Host Configuration Protocol (DHCP) and/or dynamic DNS (DDNS).

What Is DHCP?

DHCP is a protocol that automatically provides IP addresses to clients and servers based on a lease model. This is done both for ease of management and for the ability to work around IP number limitations.

What Is DDNS?

DDNS is a method that allows the client to automatically update the name records for itself on the DNS servers. It is often used in conjunction with DHCP to allow automation of host name resolution to help reduce management and administration overhead. In some cases the DHCP server will perform the update to the DNS server on the client's behalf.

Why Does This Affect Netgroups?

When a netgroup is created, it's done by adding a static host name or IP address. Because of the nature of DHCP and DDNS, using IP addresses in netgroups is a nonstarter. Thus, host names must be used, and DNS lookups would be leveraged by ONTAP when attempting to resolve netgroup members for export policy rule verification. ONTAP makes heavy use of caching to improve overall NAS performance, so when a netgroup member's IP address changes through DCHP/DDNS, the cluster does not update with that new information until a cache is refreshed. Cache TTLs are covered in this document and can be adjusted, and caches can be manually flushed.

Note: Manually flushing caches requires the caches to repopulate, which could take a while and/or create a situation in which a flood of requests eats up resources and prevents access.

Netgroups and DHCP/DDNS improvements in ONTAP 9.3

ONTAP 9.3 offers caching improvements that improves the ability to use DHCP/DDNS and netgroups together in a dynamic environment to dynamically scale up/down your client pool as needed. More information on the improvements can be found in the Caching section of this document. In order to optimize your environment to account for a dynamic client pool like this, please consider the following:

- Ensure that your TTLs for the hostname entries registered on the DNS server are low enough to expire before a DDNS hostname is reused. This will reflect into the ONTAP DNS cache.
- Ensure that the netgroup TTLs are properly set according to your DDNS hostname and netgroup policies and turnover.
- The netgroup.byhost cache is by IP address. If a particular IP address is reused within 24 hours of access, the same netgroups will apply to that IP address regardless of other changes. This TTL can be modified.
- If there is no entry in the netgroup.by host cache, then the netgroup members cache is consulted for a match. The netgroup members cache has a default TTL of 20 minutes and it is only populated if the source does not have byhost enabled.
- If netgroup.byhost is enabled for the source, the netgroup members query is not used for that source.
- The TTLs for each cache is listed in the cache tunables section of this document.

General Netgroup Best Practices

There are two categories of best practices for netgroups. The first is general netgroup pest practices and the second is best practices when using external servers for netgroups. The best practices are also split into two separate tables below.

Best Practice 19) General Netgroup Best Practices

- Use the same netgroup configuration across sources.
- Leverage netgroup.byhost mapping when using netgroups with LDAP.
- Leave blank the “domain” and “user” parts of netgroup triples. NetApp supports only host-based netgroup entries (for example, host name).
- Verify that forward and reverse DNS entries exist for host names in netgroups.
- Clean up netgroups periodically to eliminate stale entries to speed up access.Use multiple name service servers for redundancy.

Best Practice 20) Netgroup on external servers

- Verify that all servers in the configuration contain the same information.
- Remove servers from the list when undergoing maintenance.
- Enable LDAP netgroup.byhost mappings when possible (available in 8.2.3 and later).
- Verify that forward and reverse (PTR) DNS records exist for all hosts in netgroups. This is a necessary requirement for fully functional host name/export policy name resolution. For information about how to do this, see the appendix in this document.
- When loading netgroups from a file, be prepared to either wait for the cache to refresh organically or manually flush caches.
- When using DHCP and/or DDNS, netgroup caches might need to be manually flushed to reflect accurate host-ip information in netgroup caches.
- To secure LDAP binds and searches, consider using LDAP signing and sealing, which is available as of ONTAP 9.0. For more information, see TR-4073: Secure Unified Authentication.

Limits

The following table covers the limits for netgroups in versions of the ONTAP 8.3 and later operating system.

Limit	Value
Nesting limit for netgroups	1,000
Limits for file size for -load-from-uri	5MB **Note:** If the limit is exceeded, the load fails.
Single netgroup line limit (local file)	4096
Number of NIS servers	10
Line limit for NIS databases (NIS maps): external NIS servers	1024

7.6 Export Policy and Rule Best Practices

Preupgrade Considerations

It is always a good idea to prune export policy rules and client matches prior to upgrade. If there are un-resolvable hostnames in a rule’s client match, there is always the chance that NAS access to volumes that use that export policy could be compromised.

Best Practice 21) General Export Policy Best Practices

- If using host names in export policy rules or in netgroups, make sure that all host names resolve in DNS (forward and reverse lookup).
- Avoid using short names. Fully qualified domain names (FQDNs) are much faster to resolve and result in better export policy rule evaluation performance. In some instances, short names that do not resolve can cause outages for other export rules.
- Never use CNAMEs. At best, export evaluation is slow. At worst, access is denied to hosts that should be allowed access.
- Avoid making frequent changes to export policy rules if possible. Each change requires the cache to be repopulated, which means that the cluster needs to spend time and resources on that function. This can add latency to access requests.
- If using large netgroups or a large number of host names in export policy rules (that is, thousands of hosts), be sure that all hosts resolve properly in DNS and that FQDNs or IP addresses are used.
- If netgroups/host names see a large amount of churn (that is, things change often), then make sure the appropriate caches are set accordingly to reflect that. For more information, see the section about cache tunables in this document.

Predeployment

Prior to rolling out an export policy and rule set, be sure to check all hosts and netgroups being used so that they have proper name resolution in DNS and the netgroup lookups work properly.

Checking host and netgroup entries

The command `vserver services name-service getXXbyYY` (**advanced** privilege) is there to perform name-service lookups from the command line to ensure external servers are giving the expected response to the queries to provide the intended permissions to exports and files. By default, this command set does not use or warm the cache. If the intention is to query and warm the cache, the `-use-cache` option can be changed to do that, but some subqueries, such as netgroup do not support warming the cache. In addition, the `-show-source` option can be used to indicate which source returned the result. An example of this command is:

```
cluster::*> name-service getxxbyyy gethostbyaddr  192.0.2.100 -node node-02 -vserver svm1 -show-
source true -use-cache true
  (vserver services name-service getxxbyyy gethostbyaddr)
Source used for lookup: DNS
IP address: 192.0.2.100
Host name: hostname.ntap.local
```

You can also check to see if a certain IP address is part of a netgroup. Use the following `getXXbyYY` command:

```
cluster::*> name-service getxxbyyy netgrpcheck -node node-01 -vserver svm1 -netgroup netgroup1 -
clientIP 198.51.100.233 -show-source true
  (vserver services name-service getxxbyyy netgrpcheck)
Success. Client 198.51.100.233 is member of netgroup netgroup1
Searched using NETGROUP_BYNAME
Source used for lookup: LDAP
```

The diagnosing and troubleshooting section of this document has more information on using this command.

Postdeployment

After deploying export policies and rules, be sure to check client access by using the command `export-policy check-access`.

Export Policy Rule Access Verification

ONTAP has the ability to check access for a certain host, volume/qtree and export policy by using the `export-policy check-access` command. This function does not take hostname resolution into account, so only IP addresses are allowed to be checked. For name resolution, use the `name-services getXXbyYY` command set to verify the correct resolution prior to using this command. The following is the output:

```
NAME
vserver export-policy check-access -- Given a Volume And/or a Qtree, Check to See If the Client
Is Allowed Access

AVAILABILITY
This command is available to cluster and Vserver administrators at the admin privilege level.

DESCRIPTION
The vserver export-policy check-access command checks whether a specific client is allowed access
to a specific export path. This enables you to test export policies to ensure they work as
intended and to troubleshoot client access issues.

The command takes the volume name (and optionally the qtree name) as input and computes the
export path for the volume/qtree. It evaluates the export policy rules that apply for each path
component and displays the policy name, policy owner, policy rule index and access rights for
that path component. If no export policy rule matches the specified client IP address access is
denied and the policy rule index will be set to 0. The output gives a clear view on how the
export policy rules are evaluated and helps narrow down the policy and (where applicable) the
specific rule in the policy that grants or denies access. This command is not supported on
Infinite Volumes.
```

Example of export-policy check-access:

```
cluster::*> vserver export-policy check-access -vserver svm1 -client-ip 1.2.3.4 -volume flex_vol
-authentication-method sys -protocol nfs3 -access-type read

                                        Policy    Policy       Rule
Path                          Policy    Owner     Owner Type  Index Access
----------------------------- --------- --------- ---------- ------ ----------
/                             default   vs1_root  volume          1 read
/dir1                         default   vs1_root  volume          1 read
/dir1/dir2                    default   vs1_root  volume          1 read
/dir1/dir2/flex1              data      flex_vol  volume         10 read
4 entries were displayed.
```

8 Name Service Statistics

8.1 Global Cache Statistics

Included into the new Global Name Service caches are statistics about how fast the caches are and how fast the offbox resolution is taking on a per-api or per-db basis. These counter manager objects are:

```
nscc_api
nscc_db
nscc_rpc
mcached_mdb
```

The above objects are mostly used for cache performance issues. They should not be used to diagnose external name server issues. There are other counters to assist with that.

8.2 External Services Statistics

In ONTAP 9, new counter manager values for external services were added. This allows for the performance monitoring of external name service servers and how ONTAP perceives their response times. The objects are:

```
external_service_op
external_service_op_error
external_service_server
```

DNS Statistics

The DNS server statistics are only kept in the external_service_op and the external_service_op_error objects. To leverage the statistics, you must start the statistics gathering job. Use a pipe symbol (|) to include multiple objects in the capture:

```
cluster::*> statistics start –object external_service_op|external_service_op_error
```

After the gathering interval completes, stop the statistics:

```
cluster::*> statistics stop
```

Statistics collection is being stopped for sample-id: sample_91613

To view the statistics:

```
cluster::*> statistics show -sample-id [sample ID]
```

Sample output of the statistics gathered for external_service_op:

```
Object: external_service_op
Instance: svm1:DNS:Query:198.51.100.181
Start-time: 7/13/2016 11:48:41
End-time: 7/13/2016 11:49:34
Elapsed-time: 53s
Scope: svm1

    Counter                                                     Value
    -------------------------------- --------------------------------
    instance_name                       svm1:DNS:Query:198.51.100.181
    last_modified_time                       Wed Jul 13 11:48:48 2016
    node_name                                                 node-01
    num_not_found_responses                                         0
    num_request_failures                                            0
    num_requests_sent                                               1
    num_responses_received                                          1
    num_successful_responses                                        1
    num_timeouts                                                    0
    operation                                                 DNS Query
    request_latency                                             1892us
    request_latency_hist                                            -
                              <20us                                 0
                              <40us                                 0
                              <60us                                 0
                              <80us                                 0
                             <100us                                 0
                             <200us                                 0
                             <400us                                 0
                             <600us                                 0
                             <800us                                 0
                               <1ms                                 0
                               <2ms                                 0
                               <4ms                                 0
                               <6ms                                 0
                               <8ms                                 0
                              <10ms                                 0
                              <12ms                                 0
                              <14ms                                 0
                              <16ms                                 0
                              <18ms                                 0
                              <20ms                                 1
                              <40ms                                 0
                              <60ms                                 0
                              <80ms                                 0
                             <100ms                                 0
                             <200ms                                 0
                             <400ms                                 0
                             <600ms                                 0
                             <800ms                                 0
                                <1s                                 0
                                <2s                                 0
                                <4s                                 0
                                <6s                                 0
                                <8s                                 0
                               <10s                                 0
                               <20s                                 0
                               <30s                                 0
                               <60s                                 0
                               <90s                                 0
                              <120s                                 0
                              >120s                                 0
    server_ip_address                                 198.51.100.181
    server_name                                                     -
    service_name                                                  DNS
    vserver_name                                                 svm1
    vserver_uuid                                  05e7ab78-2d84-11e6-a796-
                                                          00a098696ec7
```


Sample output of the statistics gathered for external_service_op_error:

```
Object: external_service_op_error
Instance: svm1:DNS:Query:NXDOMAIN:198.51.100.181
Start-time: 7/13/2016 11:48:41
End-time: 7/13/2016 11:49:34
Elapsed-time: 53s
Scope: svm1

    Counter                                                     Value
    -------------------------------- --------------------------------
    count                                                           0
    error_string                                             NXDOMAIN
    instance_name              svm1:DNS:Query:NXDOMAIN:198.51.100.181
    last_modified_time                        Thu Jun 30 09:46:14 2016
    node_name                                                 node-01
    operation_name                                          DNS Query
    server_ip_address                                   198.51.100.181
    server_name                                                     -
    service_name                                                  DNS
    vserver_name                                                 svm1
    vserver_uuid                              05e7ab78-2d84-11e6-a796-
                                                          00a098696ec7
    count                                                           0
    error_string                                             NXDOMAIN
    instance_name              svm1:DNS:Query:NXDOMAIN:198.51.100.181
    last_modified_time                        Thu Jun 30 10:10:20 2016
    node_name                                                 node-02
    operation_name                                          DNS Query
    server_ip_address                                   198.51.100.181
    server_name                                                     -
    service_name                                                  DNS
    vserver_name                                                 svm1
    vserver_uuid                              05e7ab78-2d84-11e6-a796-
                                                          00a098696ec7
```

LDAP/NIS/Active Directory External Statistics

The LDAP, NIS and Active Directory server statistics are only kept in all the external_service objects. To leverage the statistics, you must start the statistics gathering job. Use a pipe symbol (|) to include multiple objects in the capture:

To leverage the statistics, you must start the statistics gathering job. Use a pipe symbol (|) to include multiple objects in the capture:

```
cluster::*> statistics start –object
external_service_server|external_service_op|external_service_op_error
```

After the gathering interval completes, stop the statistics:

```
cluster::*> statistics stop
```

Statistics collection is being stopped for sample-id: sample_91613

To view the statistics:

```
cluster::*> statistics show -sample-id [sample ID]
```

Note: Because the output of these statistics can be extensive, no examples are shown here.

In order to get overall server responsiveness from the statistics, the external_service_server object has the following counters:

```
cluster::*> statistics show -object external_service_server -counter ?
  connect_failures
  connect_latency_hist
  instance_name
  instance_uuid
  last_modified_timestamp
  node_name
  node_uuid
  num_connect_attempts
  process_name
  server_ip_address
  server_name
  service_name
  vserver_name
  vserver_uuid
```

If you want to understand the latency of individual calls, then you can use the external_service_op object. This object provides the following counters:

```
cluster::*> statistics show -object external_service_op -counter ?
  instance_name
  instance_uuid
  last_modified_timestamp
  node_name
  node_uuid
  num_not_found_responses
  num_request_failures
  num_requests_sent
  num_responses_received
  num_successful_responses
  num_timeouts
  operation
  process_name
  request_latency
  request_latency_hist
  server_ip_address
  server_name
  service_name
  vserver_name
  vserver_uuid
```

If you want to understand the errors, then use the external_service_op_error object.

```
cluster::*> statistics show -object external_service_op_error -counter ?
  count
  error_string
  instance_name
  instance_uuid
  last_modified_timestamp
  node_name
  node_uuid
  operation_name
  process_name
  server_ip_address
  server_name
  service_name
  vserver_name
  vserver_uuid
```

SecD External Server Statistics

Additionally, ONTAP offers SecD statistics at the diag privilege level to give information about server connections, failures, etc. for connections that are specific to the SecD process. Most of these statistics are only provided for recent queries so they will not always be shown.

Example of SecD connection statistics:

```
cluster::*> diag secd connections show -node node2 -vserver svm1
[ Cache: LSA/domain.example.com ]
  Queue> Waiting: 0, Max Waiting: 1, Wait Timeouts: 0, Avg Wait: 0.00ms
  Performance> Hits: 4, Misses: 1, Failures: 0, Avg Retrieval: 8.20ms

 + Rank: 01 - Server: 198.51.100.120 (2k8-dc-1.domain.example.com)
              Connected through the 198.51.100.9 interface, 0.0 mins ago
              Used 5 time(s), and has been available for 2 secs
              RTT in ms: mean=1.00, min=1, max=1, med=1, dev=0.00 (0.0 mins of data)

[ Cache: LDAP (Active Directory)/domain.example.com ]
  Queue> Waiting: 0, Max Waiting: 1, Wait Timeouts: 0, Avg Wait: 0.00ms
  Performance> Hits: 1, Misses: 1, Failures: 0, Avg Retrieval: 6.00ms

 + Rank: 01 - Server: 198.51.100.120 (2k8-dc-1.domain.example.com)
              Connected through the 198.51.100.9 interface, 0.0 mins ago
              Used 2 time(s), and has been available for 2 secs
              RTT in ms: mean=4.00, min=1, max=7, med=7, dev=3.00 (0.0 mins of data)

[ Cache: LDAP (NIS & Name Mapping)/<no key> ]
  Queue> Waiting: 0, Max Waiting: 1, Wait Timeouts: 0, Avg Wait: 0.00ms
  Performance> Hits: 4, Misses: 1, Failures: 0, Avg Retrieval: 0.60ms

 + Rank: 01 - Server: 198.51.100.120 (198.51.100.120)
              Connected through the 198.51.100.9 interface, 0.0 mins ago
              Used 5 time(s), and has been available for 2 secs
              RTT in ms: mean=8.60, min=2, max=22, med=4, dev=7.58 (0.0 mins of data)
```

9 Diagnosing and Troubleshooting Name Service Issues

The following sections give some tips on how to diagnose name service issues.

Common Causes of Name Service Issues

Some common causes of high RTT or incrementing errors include:

- Slow LAN or WAN links
- Name service servers that must travel great distances to the clients and storage system
- Network disconnects/drops/outages
- Busy or overloaded name service servers (for example, too many TCP connections, CPU maxed out, not enough servers to balance load)
- Firewall rules blocking TCP or UDP connections to name services

Common Symptoms of Name Service Outages

- Slow or failing user and group name resolution
- Permission/mount/CIFS share access issues
- Slow or hanging listing of NFS files
- Errors in logs concerning DNS or SecD processes (event log show on the cluster)

Check hosts for export policy access

Use the following command:

```
cluster::*> vserver export-policy check-access -vserver svm1 -client-ip 1.2.3.4 -volume flex_vol
-authentication-method sys -protocol nfs3 -access-type read

                                        Policy    Policy       Rule
Path                          Policy    Owner     Owner Type  Index Access
----------------------------- --------- --------- ---------- ------ ----------
/                             default   vs1_root  volume          1 read
/dir1                         default   vs1_root  volume          1 read
/dir1/dir2                    default   vs1_root  volume          1 read
/dir1/dir2/flex1              data      flex_vol  volume         10 read
```

Modify the NFS export policy access cache

You can also modify the NFS export policy access cache attributes in the NAS layer that control the refresh intervals of access caches. To see the default values, you have to go into **advanced** privilege.

```
cluster::*> export-policy access-cache config show -vserver svm1

                          Vserver: svm1
    TTL For Positive Entries (Secs): 3600
    TTL For Negative Entries (Secs): 3600
TTL For Entries with Failure (Secs): 5
            Harvest Timeout (Secs): 86400
```

Note: For more information about the values, use `man export-policy access-cache config modify`.

Check external name service responses

ONTAP uses standard libc functions for name services in versions 8.3 and later. The following getXXbyYY standard calls are available at the cluster shell of ONTAP.

Table 10) GetXXbyYY functions explained

Function	What It Does
Getaddrinfo	Get complete IP address information by using the host name
Getgrbygid	Get the group name, gid and members by using the group identifier, or GID
Getgrbyname	Get the group name, gid and members by using the group name
Getgrlist	Get the group membership list (gids) of a username
Gethostbyaddr	Get the hostname information from the IP address
Gethostbyname	Get the IP address information from the hostname
Getnameinfo	Get the hostname information from the IP address
Getpwbyname	Get the passwd entry information by using the user name
Getpwbyuid	Get the passwd entry information by using the user identifier, or UID
Netgrp	Check if a client is part of a netgroup

Function	What It Does
Netgrpbyhost	Check if a client is part of a netgroup using netgroup-by-host query

Example of netgroup lookup by using `getXXbyYY`:

```
cluster::*> getxxbyyy netgrpbyhost -node node-01 -vserver svm1 -netgroup netgroup2 -clientIP
198.51.100.140
  (vserver services name-service getxxbyyy netgrpbyhost)
Netgroup.byhost not enabled in all the configured sources
Hostname resolved to: centos65.domain.example.com
```

Example of user lookup by using `getXXbyYY`:

```
cluster::*> getxxbyyy getpwbyuid -node node-01 -vserver svm1 -userID 1107
  (vserver services name-service getxxbyyy getpwbyuid)
pw_name: ldapuser2
pw_passwd:
pw_uid: 1107
pw_gid: 10005
pw_gecos:
pw_dir:
pw_shell: /bin/sh
```

Troubleshooting by Using getXXbyYY

The `getXXbyYY` command also has a flag that allows an admin to show what name service source is being used during a request. This is useful to troubleshoot issues.

```
     [-show-source {true|false}] - Source used for Lookup
         Use this parameter to specify if source used for lookup needs to be
         displayed
```

In addition, there is a hidden flag that provides more granularity of the name services used `called show-granular-err`.

Example of user lookup by using getXXbyYY with troubleshooting flags provided:

```
cluster::*> getxxbyyy getpwbyname -node node-01 -vserver svm1 -username root -show-source true
-show-granular-err true
  (vserver services name-service getxxbyyy getpwbyname)
Source used for lookup: NIS
pw_name: root
pw_passwd: ABCD!efgh12345$67890
pw_uid: 0
pw_gid: 1
pw_gecos:
pw_dir:
pw_shell: /bin/sh
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_NONE
Error message: No error
Deterministic Result: Success
```

In the following examples, we can easily see what name service sources failed during a lookup.

Example of failed user and group lookup using getXXbyYY with troubleshooting flags provided:

```
cluster::*> getxxbyyy getgrbyname -node node-01 -vserver svm1 -groupname group1 -show-source true
-show-granular-err true
  (vserver services name-service getxxbyyy getgrbyname)
NIS:
Error code:    NS_ERROR_NONE
Error message: No error
LDAP:
Error code:    NS_ERROR_CONN_ERR
Error message: Connection error
DNS:
Error code:    NS_ERROR_NONE
Error message: No error
FILES:
Error code:    NS_ERROR_NOT_FOUND
Error message: Entry not found
Deterministic Result: Transient Error
```

```
cluster::*> getxxbyyy getpwbyname -node node-01 -vserver svm1 -username ldapuser -show-source
true
-show-granular-err true
  (vserver services name-service getxxbyyy getpwbyname)
NIS:
Error code:    NS_ERROR_CONN_ERR
Error message: Connection 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_NOT_FOUND
Error message: Entry not found
Deterministic Result: Transient Error
```

Best Practice 22) Number of Name Service Servers

It is a best practice to have multiple name service servers that are on fast Ethernet connections for all name service servers (such as LDAP, DNS, NIS, Active Directory, and so on). Multiple servers provide redundancy and load balancing and eliminate single points of failure in NAS environments.

Name Server Timeout Values

The following table shows the various timeouts for name services in the ONTAP operating system:

Table 11) Name Server Timeouts

Name Service Timeout Type	Timeout Value
LDAP server bind	5 seconds (nonconfigurable)
LDAP queries	3 seconds (default); 10 seconds (maximum)
SecD RPC call	23 seconds (nonconfigurable)
DNS query	2 seconds (default); 5 seconds (maximum)
SecD server connection	2 second ping response (nonconfigurable)
“Bad” DNS server cache	10 minutes (nonconfigurable)

Appendix

The following section covers topics that are not included in the main sections of this technical report. This includes troubleshooting, useful commands, and other topics. This section is subject to change over the lifespan of this document and does not cover all use cases

DNS Terminology

The following table is intended to define commonly used DNS terminology.

A	Resource record for IPv4 addresses performing host name–to-IP resolution
AAAA	Resource record for IPv6 addresses performing host name–to-IP resolution
DAD	Duplicate address detection
DDNS	Dynamic DNS: dynamic updates of DNS records
DNS	Domain Name System: maps host names to IP addresses and vice versa
FQDN	Fully qualified domain name: host name appended with DNS suffix; for example, host.example.com is an FQDN
PTR	Pointer record for IP-to–host name resolution
RR	DNS resource record
SOA	Start of authority record: designates which DNS server is the authoritative source for records
TTL	Time to live: how long a DNS record remains in the cache before being updated

Querying Host Names from Clients to Test DNS Entries

As per best practices for host names in netgroups, any host name should be able to resolve to DNS with forward and reverse lookups. To test this functionality from a client, two main tools can be used: dig (domain information groper) and nslookup (name service lookup):

- Dig man pages
- Nslookup man pages
- Windows nslookup reference

You can find a list of DNS error types in RFC-2929

Dig Examples

Dig example: forward lookup of host name:

```
# dig centos64.example.com -t any

; <<>> DiG 9.8.2rc1-RedHat-9.8.2-0.30.rc1.el6_6.1 <<>> centos64.example.com -t any
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 15976
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 0

;; QUESTION SECTION:
;centos64.example.com. IN  ANY

;; ANSWER SECTION:
centos64.example.com. 3600 IN A    198.51.100.140

;; Query time: 0 msec
;; SERVER: 198.51.100.120#53(198.51.100.120)
;; WHEN: Mon Apr 13 16:06:33 2015
;; MSG SIZE  rcvd: 67
```

Dig example: reverse lookup:

```
# dig -x 198.51.100.140

; <<>> DiG 9.8.2rc1-RedHat-9.8.2-0.30.rc1.el6_6.1 <<>> -x 198.51.100.140
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 14692
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 0

;; QUESTION SECTION:
;140.100.51.198.in-addr.arpa.   IN      PTR

;; ANSWER SECTION:
140.100.51.198.in-addr.arpa. 3600 IN    PTR     centos64.example.com.

;; Query time: 0 msec
;; SERVER: 198.51.100.120#53(198.51.100.120)
;; WHEN: Mon Apr 13 16:08:19 2015
;; MSG SIZE  rcvd: 92
```

Dig example: SRV record:

```
# dig _ldap._tcp.example.com -t srv

; <<>> DiG 9.8.2rc1-RedHat-9.8.2-0.30.rc1.el6_6.1 <<>> _ldap._tcp.example.com -t srv
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 41418
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 2

;; QUESTION SECTION:
;_ldap._tcp.example.com. IN        SRV

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

;; ADDITIONAL SECTION:
2k8-dc-1.example.com. 3600 IN A    198.51.100.120
2k8-dc-1.example.com. 3600 IN AAAA fd20:8b1e:b255:8599:5457:61d9:fc87:423f

;; Query time: 1 msec
;; SERVER: 198.51.100.120#53(198.51.100.120)
;; WHEN: Mon Apr 13 16:01:34 2015
;; MSG SIZE  rcvd: 150
```

Dig example: nonexistent record:

```
# dig fail.example.com

; <<>> DiG 9.8.2rc1-RedHat-9.8.2-0.30.rc1.el6_6.1 <<>> fail.example.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NXDOMAIN, id: 64486
;; flags: qr rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 1, ADDITIONAL: 0

;; QUESTION SECTION:
;fail.example.com.              IN      A

;; AUTHORITY SECTION:
example.com.            60      IN      SOA     ns1.example.com. ops.support.example.com.
1272525332 14400 20000 3600000 60

;; Query time: 132 msec
;; SERVER: 198.51.100.120#53(198.51.100.120)
;; WHEN: Mon Apr 13 16:02:21 2015
;; MSG SIZE  rcvd: 85
```

Nslookup Examples

Nslookup example: forward lookup:

```
# nslookup -type=any centos64.example.com.
Server:         198.51.100.120
Address:        198.51.100.120#53

Name:   centos64.example.com
Address: 198.51.100.140
Nslookup example: reverse lookup:
# nslookup -type=ptr 198.51.100.140
Server:         198.51.100.120
Address:        198.51.100.120#53

140.100.51.198.in-addr.arpa     name = centos64.example.com.
```

Nslookup example: SRV record:

```
# nslookup -type=srv _ldap._tcp.example.com
Server:         198.51.100.120
Address:        198.51.100.120#53

_ldap._tcp.example.com     service = 0 100 389 2k8-dc-1.example.com.
```

Nslookup example: nonexistent record:

```
# nslookup -type=any fail.example.com
Server:         198.51.100.120
Address:        198.51.100.120#53

** server can't find fail.example.com: NXDOMAIN
```

References

TR-4073: Secure Unified Authentication

TR-4182: Ethernet Storage Best Practices for ONTAP Configurations

TR-4191: Best Practices Guide for ONTAP 8.2.x and 8.3 Windows File Services

TR-4523: DNS Load Balancing in ONTAP

TR-4557: NetApp FlexGroup Technical Overview

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

Trademark Information