



Technical Report

# NFS Kerberos in ONTAP

Justin Parisi, NetApp  
June 2021 | TR-4616

## Abstract

This document covers NFS Kerberos support in NetApp® ONTAP® software and configuration steps with Active Directory and Red Hat Enterprise Linux (RHEL) clients.

## TABLE OF CONTENTS

<b>Overview .....</b>	<b>5</b>
Document scope.....	5
TL; DR – Just tell me the basic steps .....	5
High-level required components: What you need .....	5
Kerberos terminology .....	7
Supported encryption types .....	8
Supported Kerberos security modes .....	8
How Kerberos authentication works with NFS in ONTAP .....	9
KRB-UNIX name mapping behavior .....	12
<b>Benefits of using Kerberized NFS .....</b>	<b>16</b>
<b>ONTAP configuration .....</b>	<b>16</b>
Configure the NFS server .....	17
Configure DNS settings in ONTAP .....	17
Create the Kerberos realm .....	18
Enable Kerberos on the data LIFs .....	20
Modify export policy rules to allow Kerberos .....	22
Create a UNIX user or a name-mapping rule to map the NFS service principal.....	23
Create a UNIX user or a name-mapping rule to map the NFS client principal.....	25
Modify the NFS server machine account to allow only AES .....	28
<b>Red Hat Enterprise Linux client configuration .....</b>	<b>28</b>
Configure Network Time Protocol services.....	29
Verify DNS.....	29
Join the domain .....	29
Modify the machine account principal.....	29
<b>Best practices .....</b>	<b>30</b>
ONTAP best practices .....	30
NFS client best practices .....	31
Windows KDC best practices .....	31
<b>Sample configurations .....</b>	<b>32</b>
NetApp ONTAP .....	32
Windows (machine accounts and principals).....	33
RHEL 7.x client.....	36
<b>Corner cases .....</b>	<b>38</b>

Using the same machine account for CIFS/SMB and NFS Kerberos .....	38
Sharing keytabs on multiple clients .....	38
Using keytab files to kinit .....	39
Using local host files in place of DNS .....	39
Using non-Windows KDCs .....	40
DNS aliases/Canonical Names.....	41
NFS Kerberos in Cloud Volumes ONTAP .....	41
NFS Kerberos with storage Virtual Machine Disaster Recovery .....	42
Manual keytab configuration: Client and ONTAP .....	43
<b>Kerberos caches .....</b>	<b>44</b>
Initial client mount.....	44
Initial NFS mount access by a user .....	45
Subsequent mount access by users .....	45
Unmount impact on Kerberos context cache .....	46
NFS credential cache .....	47
Using -instance with the Kerberos context cache.....	47
Kerberos ticket lifetime – client cache .....	48
Kerberos ticket expiration behavior .....	49
<b>NFS Kerberos performance testing .....</b>	<b>52</b>
Observations .....	52
<b>Common issues .....</b>	<b>52</b>
Export policy troubleshooting.....	53
Errors during Kerberos interface enable, modify, or create in ONTAP .....	56
Errors during mounting of NFS Kerberos from a client .....	57
NFS Kerberos errors while attempting to access, read, or write.....	58
Common event log errors in ONTAP related to NFS Kerberos .....	59
Kerberos keytab troubleshooting .....	60
What information to collect before you contact NetApp Support .....	63
<b>Detailed configuration steps.....</b>	<b>64</b>
Rename NFS Kerberos machine accounts in Active Directory .....	64
Configure an NFS client to use Kerberos with net ads join.....	66
Configure an NFS client to use Kerberos with realm join .....	71
<b>Appendix A: Kerberos encryption types .....</b>	<b>75</b>
<b>Appendix B: Machine account attributes .....</b>	<b>76</b>

<b>Appendix C: Kerberos packet types, errors, and terminology .....</b>	<b>76</b>
<b>Disclaimer .....</b>	<b>78</b>
<b>Where to find additional information .....</b>	<b>78</b>
<b>Contact us .....</b>	<b>78</b>
<b>Version history.....</b>	<b>78</b>

## LIST OF TABLES

Table 1) Supported encryption types in ONTAP.....	8
Table 2) Supported Kerberos security modes in ONTAP. ....	8
Table 3) NFS Kerberos results. ....	52
Table 4) NFS Kerberos: Performance comparison versus nonencrypted baseline. ....	52
Table 5) Identifying and resolving issues while creating or modifying Kerberos interfaces in ONTAP. ....	56
Table 6) Identifying and resolving issues while mounting NFS Kerberos exports. ....	57
Table 7) Identifying and resolving issues in accessing Kerberos NFS exports in ONTAP.....	58
Table 8) Common event log errors in ONTAP.....	59
Table 9) Kerberos encryption types.....	75
Table 10) Valid msDS-SupportedEncryptionTypes attribute values.....	76
Table 11) Kerberos packets. ....	76
Table 12) Kerberos errors from network captures. ....	76
Table 13) Kerberos terminology from CentOS.org and IBM.com. ....	77

## LIST OF FIGURES

Figure 1) Kerberos AS-REQ conversation during mount – packet capture. ....	9
Figure 2) Kerberos TGS-REQ conversation during mount – packet capture.....	10
Figure 3) Kerberos AS-REQ conversation during kinit – packet capture. ....	11
Figure 4) Kerberos TGS-REQ conversation during kinit – packet capture. ....	11
Figure 5) Kerberos workflow between the client, the KDC, and the NFS server on NetApp storage.....	12
Figure 6) Kerberos interface configuration – System Manager prior to ONTAP 9.7. ....	21
Figure 7) Kerberos ticket lifetime management – Microsoft Windows Group Policy.....	49
Figure 8) Wireshark filter example.....	62
Figure 9) Kerberos packet capture – packet list. ....	63
Figure 10) TGS-REQ details – packet trace.....	63

# Overview

## Document scope

This document covers Kerberos configuration in NetApp ONTAP. The configuration scope is limited to an environment with the following components:

- Microsoft Windows 2016 Active Directory Key Distribution Center (KDC)
- RHEL versions 6.7 and later
- AES-256 encryption
- ONTAP 9.5 and later

**Note:** The RHEL configuration can be easily applied to CentOS clients.

In many cases, the concepts—and even the commands—can be applied to other clients and KDCs. For non-Windows KDC information, there is a short section in this document called “Using non-Windows KDCs.”

If you need to deviate from the preceding environment (such as the use of earlier ONTAP versions or different Linux clients) and this document is not getting the results you want, see the relevant client OS documentation and Windows documentation.

## TL; DR – Just tell me the basic steps

If you don't need or want the accompanying information for NFS Kerberos and prefer a shorter list of basic steps, this is the section for you. You can also use this list as a checklist of steps to follow and follow the “Common issues” section to troubleshoot. There is also a more detailed version of this list in the “Detailed configuration steps” section. If you get stuck with the shorter list, it's recommended to move on to the more detailed configuration steps.

## High-level required components: What you need

The following components are nonnegotiable for Kerberos configuration in ONTAP:

- Common DNS configurations.
- Forward and reverse DNS entries for client and server.
- SPNs (the common name used for NFS mounts) and DNS names matching.
- Common KDC servers and realm information.
- Time within five minutes on client, server, and KDC.
- Kerberos utilities installed on the client.
- Keytabs (created during domain join process).
- krb-unix name mapping rules.
- Export policy rules allowing Kerberos.
- UNIX user names matching on client and server that are the same as the incoming Kerberos UPNs.
- Kerberos allowed/running on client.

These items are recommended but are optional and not required:

- For NFSv4.x, matching ID domain on client and server.
- LDAP/SSSD for name services/UNIX identities.

## NFS client configuration

The steps listed are needed for each NFS Kerberos client:

- Configure the host name to a fully qualified domain name (FQDN).
- Configure the DNS to the KDC's DNS.
- Ensure that the client and KDC's time is within five minutes (normalizing for different time zones).
- Install the necessary Kerberos utilities:
  - For CentOS/RHEL 6.x and earlier, install `samba`, `samba-winbind`, `samba-winbind-clients`, `ntp`, `authconfig-gtk*`.
  - For CentOS/RHEL 7.x and later, install `krb5-workstation` (for `kinit`/`klist` commands).
  - For other NFS client operating systems, consult the vendor's product documentation.
- Join the NFS client to the Active Directory domain.
- Configure the `/etc/krb5.conf` file with the Kerberos realm information.
- Add the NFS client's host name/IP address to DNS as an A/AAAA record (or CNAME) and the IP address as a PTR (hostname and SPN should match) and ensure that the hostname resolves using `nslookup`.
- If desired, configure SSSD to use Active Directory as an LDAP server (see [TR-4835](#) for details).
- Ensure that the Kerberos services are started and/or the NFS client configuration allows secure NFS (depending on client operating system version).
- Modify the NFS client machine account to use the desired encryption types (no RC4-HMAC, as it is not supported by ONTAP for NFS Kerberos).
  - Alternately, modify the `/etc/krb5.conf` file to omit RC4-HMAC from the allowed encryption types.
- Test Kerberos ticket retrieval from the KDC (using `kinit` and a user name/password).

## NFS server configuration

In this case, the NFS server refers to the data LIF configuration on the NetApp ONTAP storage virtual machine (SVM). This configuration only needs to be performed once, provided it doesn't get disabled or deleted.

- Create the Kerberos realm in the SVM.
- Configure the DNS information for the SVM to the same information as the NFS client.
- Configure the NFS server option `permitted-enc-types` to the desired value.
- Ensure the date/time on the cluster is within 5 minutes of the KDC and NFS client's date/time (factoring in time zones).
- Add the NFS data LIF hostname/IP address to be used with Kerberos to DNS as an A/AAAA record (or CNAME) and the IP address as a PTR and ensure that they resolve using `nslookup`.
- Configure the data LIFs to be used for Kerberos by providing the SPN to use; this should match the DNS hostname with which clients access the NFS server (for example, `nfs/name.domain.com` = the A/AAAA record `name.domain.com` in DNS).
  - This step joins the data LIF to the domain (which is similar to CIFS/SMB servers) but uses a different machine account than the one used for CIFS/SMB.
- Check the NFS server machine account to ensure that only the desired encryption types are set.
- Ensure that the export policy rules for the volume allow `krb5` authentication.
  - Use `export-policy check-access` to verify if `krb5` is allowed access to the export.
- Create a `krb-unix` name mapping rule for the NFS client. See "Machine account SPN to UNIX name mapping" for details.
- If you plan on having non-root UNIX users access the Kerberos mount, ensure that ONTAP can resolve the UNIX user names to UIDs and UIDs to user names (see [TR-4835](#) and "User SPN to UNIX name mapping" for details).

- Alternately, configure local UNIX users and groups with the same user names to map the incoming UPNs by default.
- If you plan on using NFSv4.x, set the NFSv4 domain ID string on the NFS server options to match what is set on the NFS clients (see [TR-4067](#) for more information).
- Test an NFS mount using krb5 from the configured NFS client. If it fails, consult the “Common issues” section.

## Kerberos terminology

This section defines key terminology that is used when describing Kerberos processes. This section is meant to help clarify terms that might be unfamiliar to storage administrators.

### Key distribution center

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

### Realm (or Kerberos realm)

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

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

### Principal

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

A principal name has three parts:

- **Primary.** The primary part can be a user or a service such as the “nfs” service. It can also be the special service “host,” which signifies that this service principal is set up to provide various network services such as FTP, RSH, NFS, and so on.
- **Instance.** This part is optional in the case of a user. A user can have more than one principal. For example, Fred might have a principal that is for everyday use and a principal that allows privileged use such as a sysadmin account. The instance is required for service principals and designates the fully qualified domain name (FQDN) of the host that provides the service.
- **REALM.** A Kerberos realm is the set of Kerberos principals that are registered within a Kerberos server. By convention, the realm name is usually the same as the DNS name, but it is converted to uppercase letters. Uppercase letters are not obligatory, but the convention provides easy distinction between the DNS name and the realm name.

See the following example of principals:

```
user@DOMAIN.COM
user/admin@DOMAIN.COM
host/host.domain.com@DOMAIN.COM
root/host.domain.com@DOMAIN.COM
nfs/host.domain.com@DOMAIN.COM
```

## Tickets

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

## Secret keys

Kerberos uses a symmetric key system in which the secret key is used for both encryption and decryption. The secret key is generated from the principal's Kerberos password with a one-way hash function. The KDC stores the password for each principal and can thus generate the principal's secret key. For users who request a Kerberos service, the secret key is typically derived from a password that is presented to the `kinit` program. Service and daemon principals typically don't use a password; instead, the result of the one-way hash function is stored in a keytab.

## Keytab

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

## Supported encryption types

NetApp ONTAP technology supports NFS Kerberos with specific encryption types, depending on the operating mode and the version that you use.

To make sure that a client uses the appropriate encryption type, limit the valid encryption types on the object principal (such as the machine account) or in the keytab file rather than in the `krb5.conf` file, if possible. This approach is much more scalable in large enterprise environments, is easier to automate, and confirms that the client can use stronger encryption types when they are supported.

Table 1 shows the supported encryption type based on ONTAP version and operating mode. These types are for NFS Kerberos only and do not cover CIFS Kerberos support.

**Table 1) Supported encryption types in ONTAP.**

ONTAP version and mode	Supported encryption type
Data ONTAP operating in 7-Mode 7.x and later	DES and DES3 only <b>Note:</b> (RC4-HMAC works, but has no official support)
Data ONTAP 8.2.x and earlier (clustered)	DES and DES3
Data ONTAP 8.3.x	AES (128- and 256-bit), DES, and DES3
ONTAP 9.x	AES (128- and 256-bit), DES, and DES3

## Supported Kerberos security modes

In addition to the concept of encryption types, there are also levels of security and integrity checking in Kerberos to help prevent man-in-the-middle attacks by offering end-to-end encryption for NFS traffic. Table 2 shows which levels of Kerberos security mode are supported in various versions of ONTAP. The security modes for Kerberos are configured on the clients and in the KDCs. Export policy rules can then be configured to allow specific security modes.

**Table 2) Supported Kerberos security modes in ONTAP.**

ONTAP version and mode	Supported Kerberos security mode
Data ONTAP 7-Mode 7.x and later	krb5, krb5i, krb5p
Data ONTAP 8.2.x and earlier (clustered)	krb5
Data ONTAP 8.3.x	krb5, krb5i



ONTAP version and mode	Supported Kerberos security mode
ONTAP 9.x	krb5, krb5i, krb5p

## How Kerberos authentication works with NFS in ONTAP

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

KDCs, such as Windows Active Directory, maintain a database of principals and their Kerberos passwords. The secret key is just the principal's password converted into a cryptographic key format. In the case of NFS servers and clients, the secret key can be generated by using a random password and is stored in a keytab on the NFS server or client.

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

## Kerberos during the NFS mount process

When an NFS client mounts through Kerberos, the following process takes place:

- DNS is queried for the hostname/IP lookup. The DNS name is used to formulate the NFS SPN request.
- The NFS client SPN is used to perform an Authentication Service request (AS-REQ) from the KDC.
- If the NFS client SPN exists on the KDC and the password/keytab authentication succeeds, then a Ticket Granting Service request (TGS-REQ) is initiated from the client for the NFS SPN.
- The NFS client's SPN is used in a krb-unix name mapping operation in ONTAP. If the Kerberos SPN can map to a valid UNIX user, then the mount request is allowed. If there is no valid name mapping, access is denied. (For more information, see "Machine account SPN to UNIX name mapping.")
- The NFS server export-policy rules are checked to make sure that client access is allowed. If access is allowed to that client through NFS Kerberos, then the mount succeeds. If the export-policy rules do not have that client or allow Kerberos in the rules, access is denied. For more information, see "Export policy troubleshooting."

Figure 1 shows a packet capture of an NFS Kerberos mount request's AS-REQ conversation.

**Figure 1) Kerberos AS-REQ conversation during mount – packet capture.**

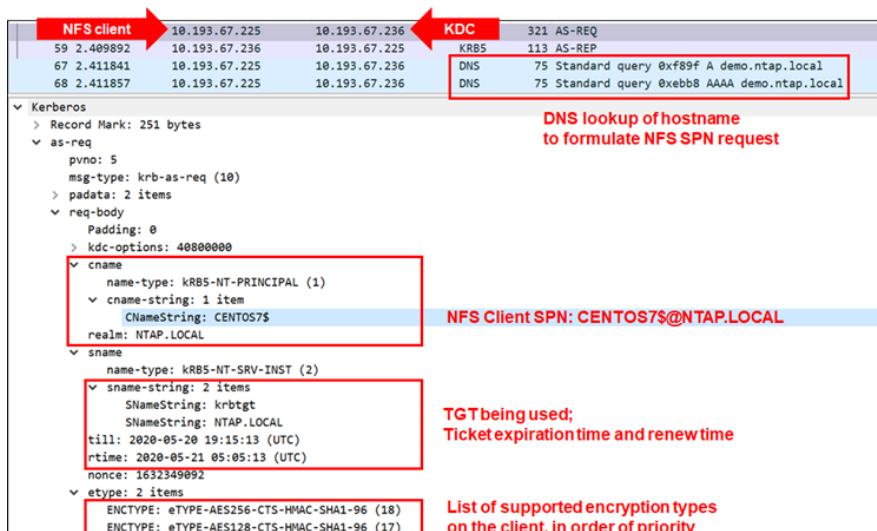
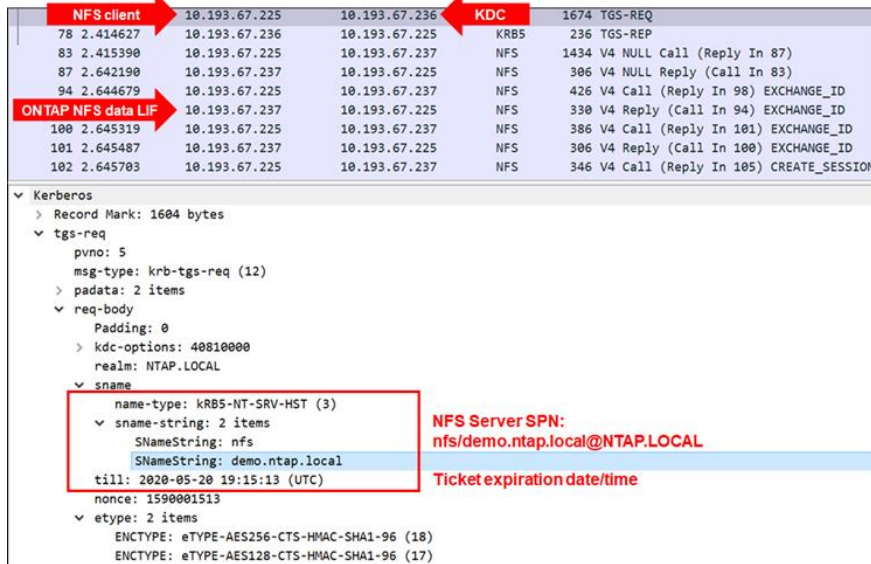


Figure 2 shows the TGS-REQ conversation during the NFS mount process.

**Figure 2) Kerberos TGS-REQ conversation during mount – packet capture.**



## Kerberos SPN formats

Kerberos SPNs can have several formats that are sent during the Kerberos mount and access process.

- `root/host.domain.com` is used by the NFS client for mount requests.
- `nfs/server.domain.com` is used by the NFS server (for example, `nfs/cluster.domain.com`).
- `host/host.domain.com` is used by the NFS client, usually for third-party applications such as SSSD.
- `CLIENT$` is used by the NFS client for mount requests, although this format is generally only used with Windows KDCs.

Any of the preceding types can be used to create a principal in Active Directory, but only one is required. In the example in Figure 1, the `CLIENT$` SPN format is used, which is the default when the NFS client joins a domain using `realm join`.

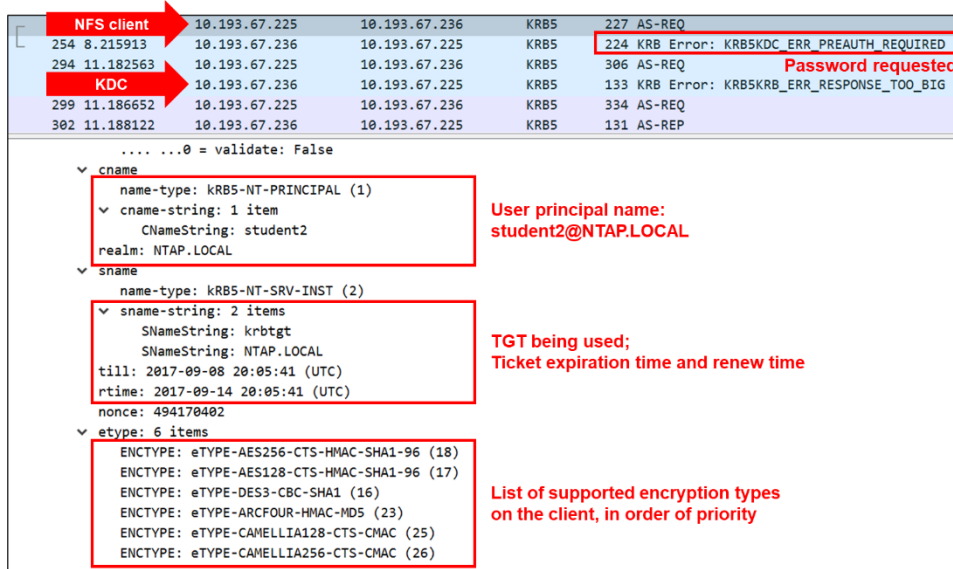
## Kerberos during NFS mount access

When a Kerberos principal such as a user or service logs in to the Kerberos realm using `kinit`, the principal sends a TGT request that contains the principal name, but not the password or secret key, to the `krb5kdc` daemon. This is known as the AS-REQ.

Upon receiving this request, the KDC looks up the principal in the KDC database and uses the associated password from the database to encrypt the TGT response.

If the principal exists, the encrypted TGT is sent to the requestor from the KDC. The principal decrypts the TGT response by using the secret key that was obtained from the password or from the keytab. This conversation is illustrated in Figure 3.

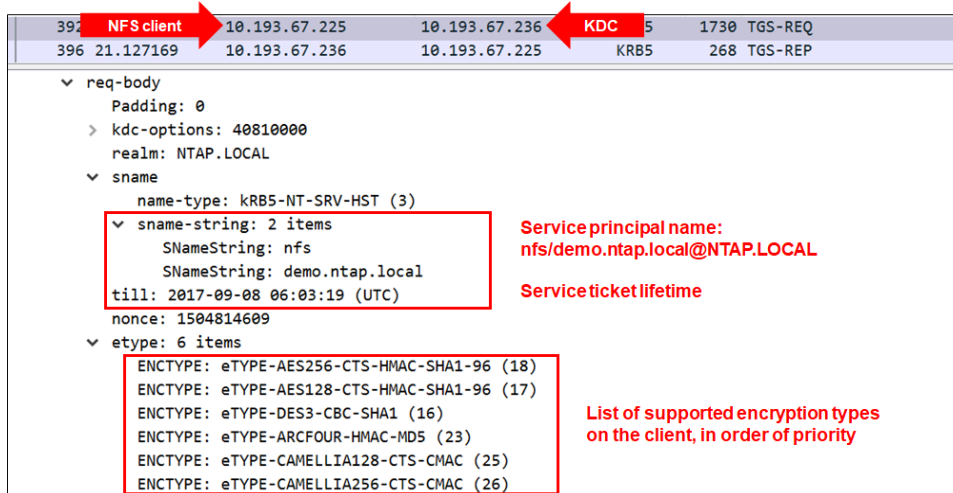
Figure 3) Kerberos AS-REQ conversation during kinit – packet capture.



The principal then requests authentication to the NFS server (in this case, ONTAP) by presenting the NFS server principal along with the encrypted TGT to the ticket-granting server (TGS). This occurs when the user attempts to access the NFS export (the TGS-REQ).

The TGS then issues a ticket for the NFS server. The ticket provides authentication to allow the principal to mount (for an NFS client SPN) or to use a specific file system that is mounted over NFS from the NetApp cluster (for a user principal). This TGS-REQ conversation is illustrated in Figure 4.

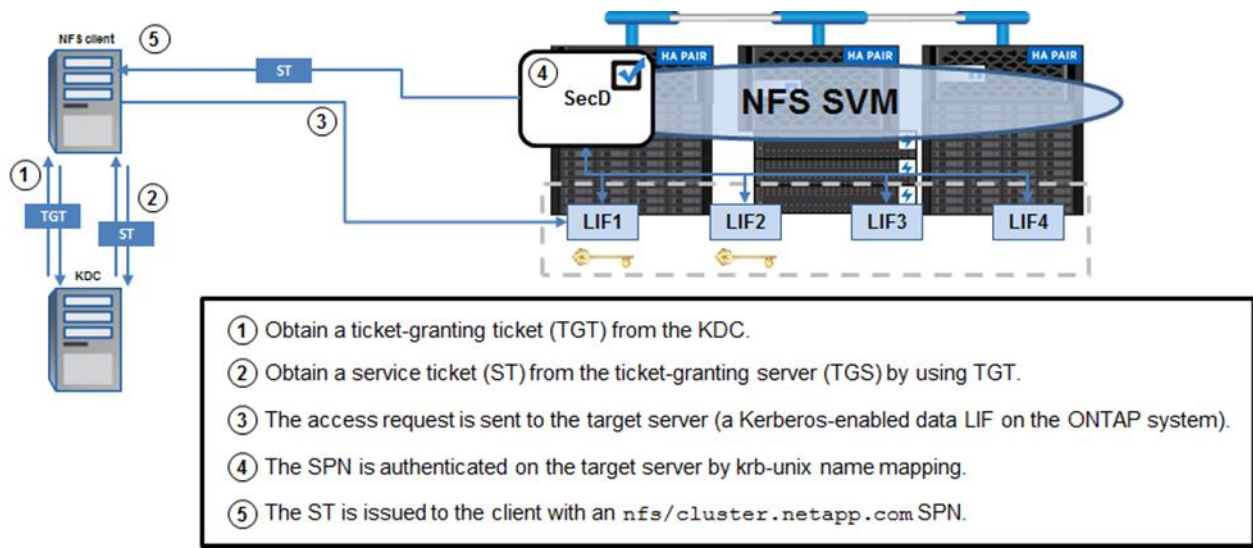
Figure 4) Kerberos TGS-REQ conversation during kinit – packet capture.



No Kerberos communication takes place between the ONTAP NFS server and the KDC because the NFS server decrypts its portion of the TGS by using its keytab entry. Figure 5 shows the Kerberos workflow between the client, the KDC, and the NFS server.

In ONTAP, the Kerberos ticket is cached until the cache is cleared (either by timeout or manual command) or the node is rebooted. For more information, see “Kerberos caches.”

Figure 5) Kerberos workflow between the client, the KDC, and the NFS server on NetApp storage.



## KRB-UNIX name mapping behavior

In ONTAP, there is a `krb-unix` name mapping rule that controls how Kerberos SPNs map into ONTAP for NFS Kerberos authentication. When a Kerberos SPN maps into ONTAP, it controls how the user is presented to the NFS export, which means that permissions depends on who the user authenticated as. With NFS Kerberos mounts, there are three `krb-spn` mappings that occur.

### Machine account SPN to UNIX name mapping

On the initial Kerberos mount request to ONTAP, the machine account SPN is used to authenticate. ONTAP attempts to map the SPN to a valid UNIX user by the same rules a regular user name has.

- 1:1 name mapping (`name == name`)
- `krb-unix` name mapping rule (explicit name mapping definition)

The machine account SPN is defined by the Kerberos keytab file on the client (`/etc/krb5.keytab`) and is dependent on how the client administrator configured the client for Kerberos. When `Samba/realm/net ads` is used to join an Active Directory domain, the keytab file contains SPNs in the following format:

```
root/fqdn.domain.com@DOMAIN.COM
host/fqdn.domain.com@DOMAIN.COM
SHORTNAME$@DOMAIN.COM
```

In the examples above, a 1:1 name mapping would map to the following UNIX users:

```
root
host
SHORTNAME$
```

If no UNIX users exist with those names, then ONTAP looks for a name mapping rule to determine initial mount access. If no valid UNIX users exist for the `krb-unix` name mapping, then the mount fails with “Permission denied,” and an EMS message is logged to ONTAP.

If a valid UNIX user exists (and export-policy rules allow it), then the NFS Kerberos mount succeeds. This authentication process controls how the `root` user is handled in Kerberos mounts.

**For example:**

- If the SPN that maps into ONTAP is `root/fqdn.domain.com@DOMAIN.COM`, then `root` is `root`. The `root` user gets the same permissions `root` always gets.
- If the SPN that maps into ONTAP is `host/fqdn.domain.com@DOMAIN.COM`, then `root` becomes the `host` user. Permissions/file ownership for `root` is determined by access allowed to `host`.
- If the SPN that maps into ONTAP is `SHORTNAME$@DOMAIN.COM`, then `root` becomes whatever user the machine account maps to. If a `SHORTNAME$` user exists, then `root` becomes the user `SHORTNAME$`. If a name mapping rule exists for all computer names, as shown in “Create a UNIX user or a name-mapping rule to map the NFS client principal”, then the `SHORTNAME$` SPN becomes the mapped user in the rule. Permissions/file ownership for `root` is determined by the mapped UNIX user.

**Note:** The way the NFS client maps into ONTAP can be used as an extra layer of security by managing `root` user access.

## User SPN to UNIX name mapping

When a user uses `kinit` to get a Kerberos TGT that is used to gain access to the NFS Kerberos mount by an NFS service ticket (ST), it is also controlling how that user gets presented to ONTAP for authorization purposes.

For example, if a UNIX user named `student1` uses `kinit` to a user named `student2@NTAP.LOCAL`, then that user maps into ONTAP with a `krb-unix` name mapping of `student2@NTAP.LOCAL`. Using the same name mapping logic as the machine account SPN, the following process applies:

- 1:1 name mapping (`name == name`)
- `krb-unix` name mapping rule (explicit name mapping definition)

This means that even though the UNIX user `student1` (`uidNumber 1301`) is accessing the mount, because the `kinit` was performed with the `student2` SPN (`uidNumber 1302`), then `student1` now becomes `student2` for access purposes.

If a user tries to `kinit` with a valid user in the KDC (with no UNIX user identity, then access to the mount fails. The following examples illustrate the two scenarios.

### Example 1: A User gets a Kerberos TGT from a user that has a valid UNIX user name

In the following example, `student1` uses the `student2` login credentials. Perhaps in this case, `student2` is trying to write to the `student1` homedir, but only has the `student2` login information.

```
# id student1
uid=1301(student1) gid=1101(group1) groups=1101(group1),1203(group3),1220(sharedgroup)
# id student2
uid=1302(student2) gid=1101(group1)
groups=1101(group1),1203(group3),1220(sharedgroup),10000(Domain Users),1202(group2)
```

`student1` and `student2` are the only users with write access to their own homedir:

```
# ls -la | grep student
drwxr-xr-x  2 student1          group1          4096 Apr 24 13:42 student1
drwxr-xr-x  2 student2          group1          4096 Apr 24 13:42 student2
```

You become `student1`, but you can `kinit/login` as `student2`:

```
# su student1
sh-4.2$ kinit student2
Password for student2@NTAP.LOCAL:
sh-4.2$ klist -e
Ticket cache: KEYRING:persistent:1301:1301
Default principal: student2@NTAP.LOCAL

Valid starting      Expires            Service principal
-----
```

```
04/24/2020 13:27:44 04/24/2020 23:27:44 krbtgt/NTAP.LOCAL@NTAP.LOCAL
renew until 05/01/2020 13:27:44, Etype (skey, tkt): aes256-cts-hmac-sha1-96, aes256-cts-
hmac-sha1-96
```

When you do this, you only have write access to the `student2` homedir, which is the expected result.

```
sh-4.2$ cd student1
sh-4.2$ touch newfile-student1-student2
touch: cannot touch 'newfile-student1-student2': Permission denied

sh-4.2$ cd student2
sh-4.2$ touch newfile-student1-student2
sh-4.2$ ls -la | grep newfile-student1-student2
-rw-r--r-- 1 student2 group1 0 Apr 24 13:28 newfile-student1-student2
```

The only way `student1` can truly become `student2` is if `student1` has the `student2`'s password. Using NFS Kerberos helps make sure that users accessing shares are indeed who they say they are.

## Example 2: A user gets a Kerberos TGT from a user that has *\*no\** valid UNIX user name

In this example, `kinit` is performed to a valid user in the KDC (for example, a Windows service account), but that user has no valid UNIX identity. In this case, the user is a `bind` user used for LDAP authentication (username is `bind`). For information on how to configure LDAP in ONTAP, see [TR-4835: How to Configure LDAP in ONTAP](#).

We check to see if the user exists as a UNIX user on the NFS client and if ONTAP can find a valid UNIX users.

```
# id bind
id: bind: no such user

cluster::*> getxxbyyy getpwbyname -node cluster-01 -vserver DEMO -username bind
(vserver services name-service getxxbyyy getpwbyname)

Error: command failed: Failed to resolve bind. Reason: Entry not found for "bind: bind".
```

In this case, `student2` is still trying to get access they don't have. This time, they use the LDAP `bind` user to try to get Kerberos access.

```
# su student2
sh-4.2$ kinit bind
Password for bind@NTAP.LOCAL:
sh-4.2$ klist
Ticket cache: KEYRING:persistent:1302:1302
Default principal: bind@NTAP.LOCAL

Valid starting    Expires          Service principal
04/24/2020 13:48:22 04/24/2020 23:48:22 krbtgt/NTAP.LOCAL@NTAP.LOCAL
renew until 05/01/2020 13:48:22
```

Because the `bind` user has no valid UNIX identity, the mount access fails, even though a valid NFS service ticket was issued. This is because we need a valid UNIX user to determine permissions. No user = no access.

```
sh-4.2$ cd /kerberos/
sh: cd: /kerberos/: Permission denied

sh-4.2$ klist
Ticket cache: KEYRING:persistent:1302:1302
Default principal: bind@NTAP.LOCAL

Valid starting    Expires          Service principal
04/24/2020 13:49:27 04/24/2020 23:48:22 nfs/demo.ntap.local@NTAP.LOCAL
renew until 05/01/2020 13:48:22
04/24/2020 13:48:22 04/24/2020 23:48:22 krbtgt/NTAP.LOCAL@NTAP.LOCAL
```



In the cluster event log, we can see the failure logged with `event log show`:

```
ERROR          secd.nfsAuth.problem: vserver (DEMO) General NFS authorization problem. Error: RPC
accept GSS token procedure failed
[ 0 ms] Using the NFS service credential for logical interface 1034
(SPN='nfs/demo.ntap.local@NTAP.LOCAL') from cache.
[ 0] GSS_S_COMPLETE: client = 'bind@NTAP.LOCAL'
[ 1] Trying to map SPN 'bind@NTAP.LOCAL' to UNIX user 'bind' using implicit mapping
[ 1] Unix User Name found in Name Service Negative Cache
[ 1] Unable to map SPN 'bind@NTAP.LOCAL'
**[ 1] FAILURE: Unable to map Kerberos NFS user 'bind@NTAP.LOCAL' to appropriate UNIX user
[ 1] Failed to accept the context: The routine completed successfully (minor: Unknown
error). Result = 6916
```

This shows that simply having access to a valid Kerberos ticket does not ensure access to Kerberos NFS mounts in ONTAP. A user SPN/UPN also needs to be able to resolve to a valid UNIX user name.

## Creating local UNIX users for user SPN mapping

In some cases, you might not have access to a name service such as LDAP (such as with Cloud Volumes ONTAP). In other cases, you might have a limited need for Kerberos, where only a few users need access—which doesn't justify standing up an entire LDAP server for identities. In most cases, applications such as [Apache](#) or the [NetApp NIPAM module](#) only need a single user, so it might be easier to simply create a local UNIX user for these use cases.

To create local UNIX users in ONTAP, ensure the user name and numeric ID match those on the client and that a UNIX group is also created by running the following commands:

```
cluster::> unix-group create
cluster::> unix-user create
```

After the user and group are created, then the Kerberos SPNs with that user name (user@REALM.COM) automatically map to the user for proper authentication into ONTAP.

## Creating an explicit name mapping rule for User SPNs

If you'd rather map a user SPN to a different UNIX user, or would like to map all user SPNs to the same UNIX user, you can use a name mapping rule. However, this option is not preferred in most cases because it defeats the purpose of identity verification of incoming Kerberos user SPNs. However, for some use cases (such as applications that only use a single UNIX user), this might be the preferred option.

To create a `krb-unix` name mapping rule, run the following commands:

```
cluster::> vserver name-mapping create -vserver NFS -direction krb-unix ?
[-position] {1..2147483647}      Position
[-pattern] <text (size 1..256)>   Pattern
[-replacement] <text (size 1..256)> Replacement
{ [[-address] <IP Address/Mask>] IP Address with Subnet Mask
| [ -hostname <text> ] }         Hostname
```

To create a global user SPN/UPN name mapping (for example, mapping all machine account SPNs or user SPNs to the "apache" user), see the following examples:

```
Vserver: DEMO
Direction: krb-unix
Position Hostname      IP Address/Mask
-----
1      -      -      Pattern: (.+)\$@NTAP.LOCAL << MACHINE$ SPN
Replacement: apache
2      -      -      Pattern: host/(.)@NTAP.LOCAL << host/ SPN
Replacement: apache
```

## NFS service SPN to UNIX name mapping

After a user logs in to become a valid user in the KDC to gain a TGT using `kinit`, the user can then access the Kerberos NFS mount, provided the NFS service SPN maps to a valid UNIX user. Failures in this process result in either “Permission denied” or in “Not a directory” errors when changing to the Kerberos export.

The NFS service SPN is defined when you enable Kerberos on data interfaces in ONTAP. Typically, the NFS service SPN uses the following format:

```
nfs/nfsservername.domain.com@DOMAIN.COM
```

In the above example, the NFS service SPN will attempt to map to a UNIX user named `nfs` first. If that fails, then ONTAP looks for explicit name mapping rules. If no valid UNIX user exists for that SPN mapping, then the request fails and an error is logged to ONTAP.

The SPN is queried in the KDC when a mount is accessed by a user. The way the NFS client requests the SPN depends on the name resolution of the NFS server specified in the mount. In most cases, DNS is preferred, but local host files could also be leveraged. For more information, see the section “Using local host files in place of DNS.”

## Benefits of using Kerberized NFS

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

- Authentication proves who you are.
- Authorization allows you to do what you need to do after you have been authenticated.

For example, if you buy a subway ticket, you are allowed through the turnstile (authentication). But after you are inside the station, you might not be allowed to travel to your destination if the ticket does not specify access (authorization).

The benefits of using NFS Kerberos in ONTAP include the following:

- Prevention of plaintext passwords from being passed over a network
- End-to-end, enterprise-class encryption through AES-256 and AES-128
- Control over SPN to user mappings through the `krb-unix` name-mapping rule
- Increased group membership limits (32 maximum) as compared with standard `AUTH_SYS` (16 maximum).

**Note:** In ONTAP 8.3 and later, the `AUTH_SYS` and `AUTH_GSS` limits can be raised to 1,024 for both `AUTH_SYS` and `AUTH_GSS`. For more information about extending the auxiliary group limits for NFS in ONTAP, see [TR-4067: NFS Best Practice and Implementation Guide](#).

## ONTAP configuration

This section covers how to configure NetApp ONTAP for the configuration of NFS Kerberos. The sections are ordered by the sequence of steps you should follow when configuring NFS Kerberos in ONTAP.

This section covers the following topics:

- Configure the NFS server



- Configure DNS settings in ONTAP
- Create the Kerberos realm
- Enable Kerberos on the data LIFs
- Modify export policy rules to allow Kerberos
- Create a UNIX user or a name-mapping rule to map the NFS service principal
- Create a UNIX user or a name-mapping rule to map the NFS client principal
- Modify the NFS server machine account to allow only AES

## Configure the NFS server

For the NFS server, you should enable and configure options to provide clients with the functionality that you need. You should make decisions about the NFS versions to use, which options to choose, and so on, before you configure NFS Kerberos. To help you make those NFS configuration decisions, see [TR-4067](#). For NFS Kerberos, you should also consider the following:

- **NFSv3 doesn't Kerberize everything.** NFSv3 has ancillary protocols such as mount, port mapper, NLM, and so on. Kerberos in ONTAP covers only the NFS portion of the protocol version. NFSv4.x can Kerberize the entire stack because it's all combined according to the standard.  
**Note:** If you want to use Kerberos for NFSv3 in ONTAP 8.2P5 and earlier, make sure that the export policy rules allow `sys` and `krb5*` as per [bug 756081](#) (a NetApp support login might be required to view the bug link).
- **NFSv3 can use `krb5i` and `krb5p`.** You can encrypt NFS packets when using NFSv3 but, as mentioned, NFSv3's ancillary protocols won't use Kerberos.
- **NFS Kerberos adds a performance hit.** There will be a performance penalty when you use Kerberos. See "NFS Kerberos pPerformance tTesting" for details.
- **Consider removing less secure encryption types.** By default, NFS servers in ONTAP allow the following encryption types (enctypes) on creation:

```
des,des3,aes-128,aes-256
```

DES and DES3 are much less secure enctypes. In fact, DES is disabled by default in modern Windows KDCs. If you don't need DES or DES3, remove them from the list. After Kerberos is enabled in an ONTAP SVM, removing enctypes later requires downtime. It is better to remove the enctypes before you enable NFS Kerberos in ONTAP.

To disable DES and DES3 in ONTAP SVMs:

```
cluster::> nfs modify -vserver [vserver] -permitted-enc-types aes-*
```

**Note:** ONTAP System Manager currently cannot be used to modify the permitted encryption types.

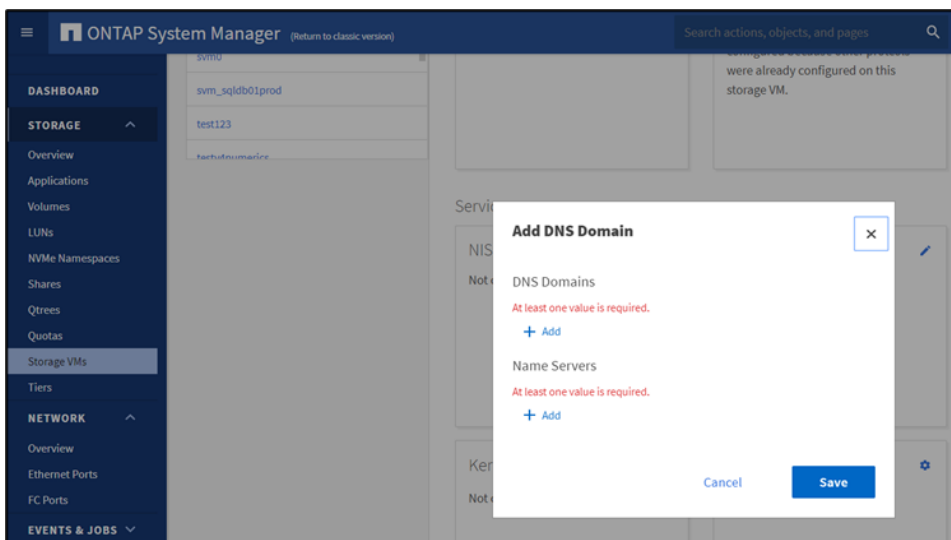
## Configure DNS settings in ONTAP

For DNS lookups to work properly for Active Directory connectivity and for Kerberos functionality with ONTAP, you must configure the DNS at the data SVM level. You can configure the DNS in ONTAP System Manager or through the command line. The DNS servers must be able to resolve the cluster data LIFs and the client's host name, either through A/AAAA records or through DNS forwarding/delegations.

To configure DNS settings in System Manager prior to ONTAP 9.7, go to SVMs > SVM Settings > DNS/DDNS.

Edit Refresh  
**DNS Service Configuration**  
 DNS Service: ● Enabled  
 DNS Domains: core-tme.netapp.com  
 Name Servers: 10.193.67.181  
 10.193.67.200

To configure DNS settings in ONTAP System Manager 9.7 and later, go to Storage > Storage VMs > DNS under the Services section and click the gear symbol.



To configure DNS settings in the CLI, run the following command:

```
cluster::> dns modify -vserver [SVM] -domains [domain1, domain2..] -name-servers [IP1, IP2..]
```

## Add DNS records or configure on-box DNS for the SVM data LIFs

You should add to DNS the data LIFs in the SVM that will be participating in NFS Kerberos. You can add the LIFs either through A/AAAA and PTR records or by leveraging the on-box DNS. Work with your DNS administrator to accomplish this task. For information about configuring on-box DNS or adding records to DNS, see [TR-4523](#).

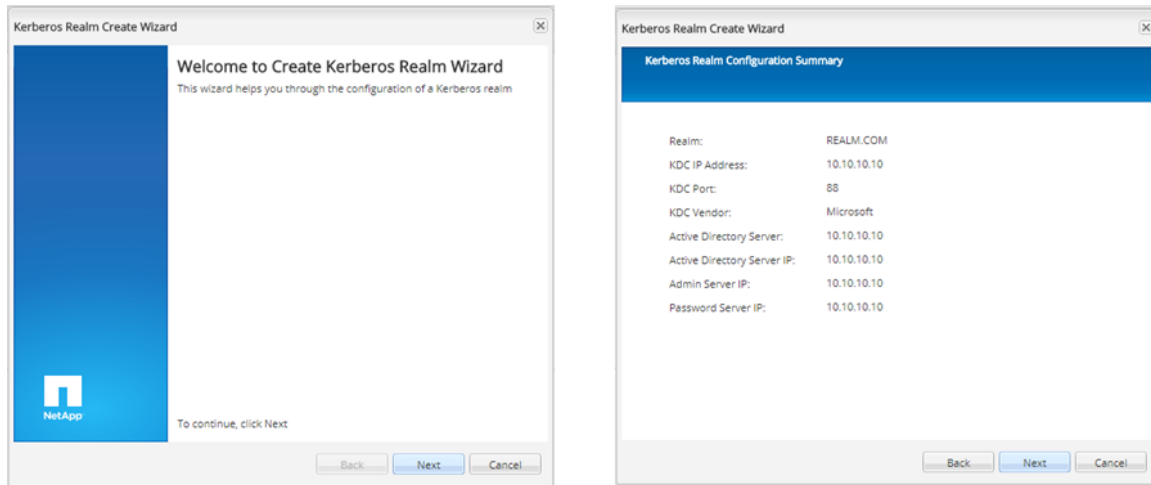
## Create the Kerberos realm

You need a Kerberos realm so that the cluster knows how to format Kerberos ticket requests properly. Creating the realm in ONTAP is similar to configuring `/etc/krb5.conf` on NFS clients. The IP

addresses that are specified in the Kerberos realm commands are used only during creation of the machine account object or SPN. These IP addresses are not used for actual Kerberized NFS traffic after Kerberos is enabled. Therefore, you do not need to worry about specifying KDCs for failover or DNS aliases with these commands. KDC failover for Kerberized traffic is handled by using DNS SRV records.

You can create Kerberos realms by using ONTAP System Manager or by using the CLI. To create a Kerberos realm in System Manager prior to ONTAP 9.7, complete the following steps:

1. Go to go to SVMs > SVM Settings > Services > Kerberos Realm.
2. The realm configuration appears as a wizard. Enter your values and click Next for each screen.



To create a Kerberos realm in System Manager in ONTAP 9.7 and later, complete the following steps:

1. Go to Storage > Storage VMs > Kerberos under the Services section.
2. Click Add and populate the fields.

**Add Kerberos Realm**

NAME: NTAP.LOCAL

KDC IP ADDRESS: 10.10.10.10

KDC PORT: 88

KDC VENDOR: ☒ Microsoft

ACTIVE DIRECTORY SERVER NAME: dc.ntap.local

ACTIVE DIRECTORY SERVER IP ADDRESS: 10.10.10.10

To create a Kerberos realm in the CLI, use the following command:

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

## Enable Kerberos on the data LIFs

To use Kerberos for NFS, you must enable Kerberos on a data LIF in the SVM. When Kerberos is enabled, the SPN is defined and a principal is created on the KDC you specified in the Kerberos realm configuration. By default, this machine account uses only the first 15 characters of the NFS SPN, including the `nfs/` portion. Therefore, if you want to have multiple Kerberos-enabled data LIFs, you should use names that are unique within the first 15 characters. To help avoid issues later with duplicate machine object names, you can specify the `-machine-account` option during the command or you can rename machine account objects after the fact. See the section “Rename NFS Kerberos machine accounts in Active Directory” for more information on how to rename machine accounts.

When Kerberos is enabled in ONTAP, the KDC is contacted and credentials are exchanged. The credentials that you provide must have the rights to create objects in the computer’s organizational unit (OU) in Active Directory. This user can be a [domain administrator or a user who has had rights delegated](#) to manage that OU. For non-Windows KDCs, the user also needs to be able to create and modify SPNs.

The SPN must use the format in the example of `primary/instance@REALM`, where `REALM` is always in ALL CAPS. If you don’t use this format, the command fails. For an example of this and other possible errors, see the section “Errors during Kerberos interface enable, modify, or create in ONTAP.”

Some other factors that you should consider include the following:

- This process is performed one data LIF at a time.
- After you enable Kerberos on a data LIF, no credential exchange is required if you use the same SPN on subsequent data LIFs.
- You can use the same SPN for multiple LIFs or use different SPNs for different data LIFs.
- For every new SPN that is specified, a new machine account is created in Active Directory with the default name `NFS-SPN-NAME` (up to 15 characters). To override this behavior, use the `-machine-account` option.
- For data LIFs with the same SPN, only one machine account is created.
- You need a domain user who has the permissions to create objects in the specified domain OU. The default OU is `DC=DOMAIN, DC=COM`.
- If you specify an OU, do not include `DC=DOMAIN, DC=COM`; the base DN is implied.
- The SPN is created as `nfs/[desired DNS name for access]@REALM_IN_CAPS.COM`.
- When manually creating the keytab and using the `keytab-uri` command option, the SPN in the Kerberos interface command is case sensitive; that is, if you specify the SPN in the KDC as `NFS/name` but try to use `nfs/name` as the SPN in the ONTAP command, the command will fail.

To create the user and group in ONTAP System Manager prior to ONTAP 9.7, go to SVM > SVM Settings, under Services > Kerberos Interface as shown in Figure 6.

**Figure 6) Kerberos interface configuration – System Manager prior to ONTAP 9.7.**

**Edit Kerberos Configuration**

Interface Name:

☒ **Enable Kerberos**

Kerberos Realm:

? Service Principal Name:   
example: nfs/<fqdn>@REALM

Keytab URI:  (optional)

Admin Username:

Admin Password:

To create the user and group in ONTAP System Manager 9.7 and later, complete the following steps:

1. Go to go to Storage > Storage VMs > Kerberos under the Services section.
2. Click Add or edit an existing Kerberos configuration.
3. Scroll down to the Add Network Interface to Realm section.
4. Click Add.

**Add Network Interface to Realm**

SELECT INTERFACE Filter

Interface Name	Service Principal Name	Kerberos Status
data2	nfs/demo.ntap.local@NT...	True
data	nfs/demo.ntap.local@NT...	True

+ Add

**Add Network Interface to Realm**

SELECT INTERFACE Filter

**Add Network Interface** x

KERBEROS INTERFACE

SERVICE PRINCIPLE NAME

ADMIN USERNAME

ADMIN PASSWORD

+ Add

To create the user and group in the CLI, run the following command:

```
kerberos interface enable -vserver [SVM] -lif data1 -spn [nfs/fqdn.domain.com@REALM.COM] -ou [CN=Servers] -machine-account [machineaccountname]
```

## Modify export policy rules to allow Kerberos

Export policies in ONTAP are containers for export policy rules. Export policy rules are the share-level permissions that are applied to NFS exports. Access is provided or is denied based on host identity, such as IP address, host name, netgroup, or Kerberos authentication.

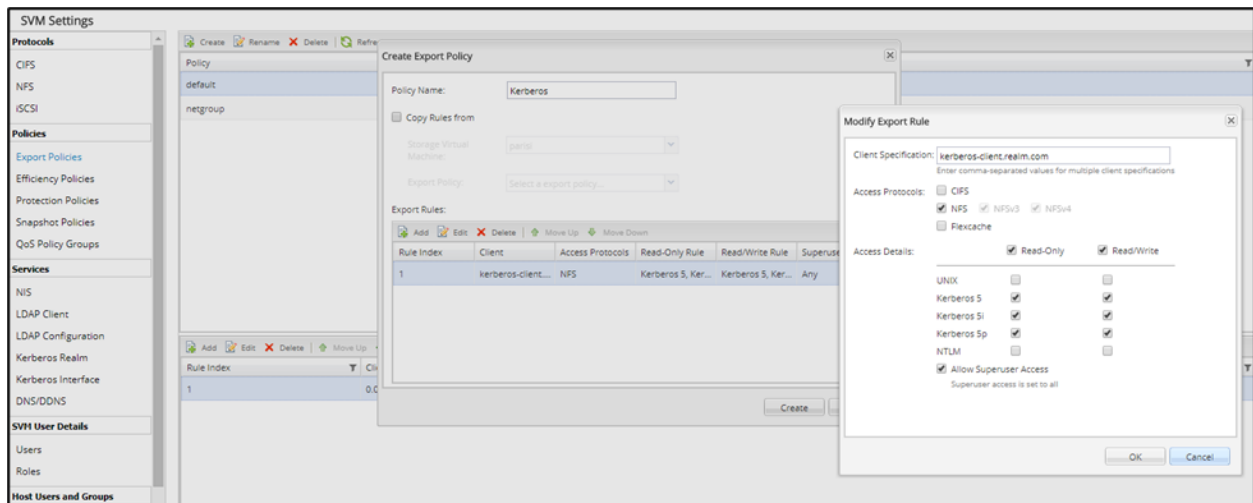
To allow Kerberos mounts, you must specify Kerberos security in the `rorule`, `rwrule`, and/or `superuser` fields of an export policy rule, depending on the level of access you wish to allow. Several different versions of Kerberos security are available in ONTAP 9 and later:

- **krb5.** Uses Kerberos V5 name strings and user principal names instead of local UNIX user IDs (UIDs) and group IDs (GIDs) to authenticate users.
- **krb5i.** Uses Kerberos V5 for user authentication and also performs integrity checking of NFS operations by using secure checksums to prevent data tampering and man-in-the-middle attacks.
- **krb5p.** Uses Kerberos V5 for user authentication and integrity checking and also encrypts all NFS traffic to prevent packet sniffing. This setting is the most secure, but it also creates the most performance overhead.

The Kerberos security options are negotiated between the client and the KDC. ONTAP export policies and rules simply provide a way to allow, or even require, a specific security option. If a krb5 security option is not specified in the export policy rule, attempts to mount NFS Kerberos exports fail, with access denied or permissions issues. You can check export-policy-rule access with the CLI, as detailed in the section called “Export policy troubleshooting.”

**Note:** NetApp does not recommend using `krb5i` or `krb5p` in ONTAP versions earlier than 9.2.

To create or modify export policies and rules in ONTAP System Manager prior to ONTAP 9.7, go to SVM > SVM Settings, under Policies > Export Policies.



To create the user and group in ONTAP System Manager 9.7 and later, complete the following steps:

1. Go to Storage > Volumes.
2. Select the volume for which you want to configure the export policies and rules.
3. Click Edit.
4. Scroll down to Export Settings and check Set export policies.
5. Select an existing policy and edit the rules or create a new policy and rules.

**Export Settings** [Export settings considerations](#)

MOUNT PATH  
/home

☒ Set export policies

☒ Select an existing policy

EXPORT POLICY  
home

This export policy is being used by 10 objects.

Rule Index	Clients	Access Protocols	Read-Only Rule	Read/Write Rule	Status
1	10.193.67....	Any	Any	Never	Any
2	10.193.67....	NFS	Krb5, Krb5i, KrbSp, Sys	Krb5, Krb5i, KrbSp, Sys	Krb

[+ Add](#)

☐ Add a new policy

**Save** **Cancel**

To modify export policy rules to allow krb5 in the CLI, run the following command:

```
cluster::> export-policy rule modify
```

For more details about export policies and rules, see [TR-4067](#).

## Create a UNIX user or a name-mapping rule to map the NFS service principal

When a client attempts to access a mount with NFS Kerberos, a service ticket is requested by using the SPN that was defined in the Kerberos configuration. This SPN attempts to map into ONTAP through a `krb-unix` name mapping, using the first portion of the SPN as the default UNIX user name. For Kerberos-enabled interfaces, that name is `nfs/fqdn.realm.com@REALM.COM`.

If no name mapping or valid UNIX user (such as `nfs`) exists, the Kerberos access attempt fails and the client reports access denied/permission denied. ONTAP logs the failure to the event management system (EMS) in the form of a name-mapping failure.

To see the EMS event that is logged, use the following command.

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

You can approach this task in either of two ways:

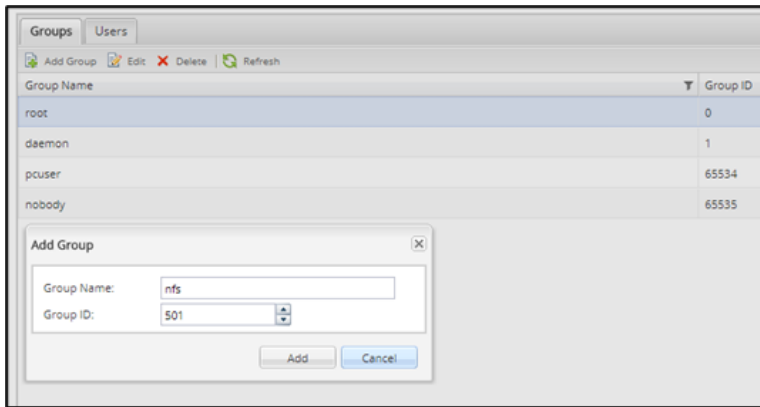
- Create a UNIX user named `nfs` for implicit name mapping either locally or in LDAP (if using LDAP).
- Create an explicit Vserver name-mapping rule for the SPN to map to an existing valid UNIX user.

For more information about Kerberos to UNIX name mapping, see the section “KRB-UNIX name mapping behavior.”

### Option 1: Creating a UNIX user and group

To create a UNIX user in ONTAP, use either ONTAP System Manager or the command line to create a user and a group named “`nfs`” with any UID and GID that you choose. In general, service accounts use a range between 1 and 1,024 for UIDs and GIDs. Before you define a numeric UID or GID, make sure that it is not in use elsewhere in your environment.

To create the user and group in ONTAP System Manager, go to SVM > SVM Settings, under Host Users and Groups.



**Note:** There's currently no way to create local UNIX users and groups in the new ONTAP System Manager view found in ONTAP 9.7 and later.

To create the user and the group in the CLI, run the following commands:

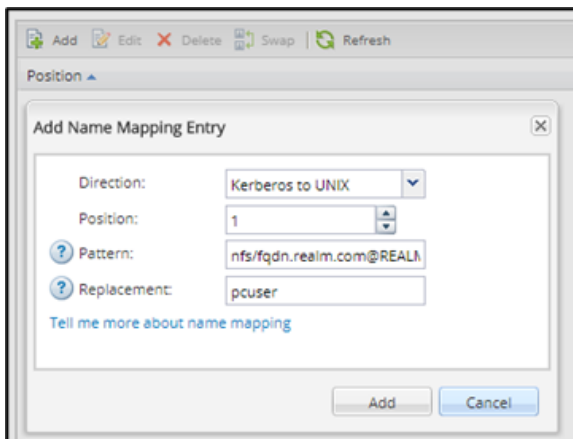
```
unix-user create -vserver [SVM] -user nfs -id [500] -primary-gid [500] -full-name "NFS Kerberos"
unix-group create -vserver [SVM] -name nfs -id [500]
```

Creating a UNIX user and group is the simplest way to handle NFS Kerberos SPN `krb-unix` authentication into the cluster. Alternatively, if you have LDAP in your environment, you can create a user named "nfs" in LDAP.

## Option 2: Creating a krb-unix name-mapping rule

If you do not want to create a UNIX user and group, you can create a name-mapping rule to handle NFS Kerberos SPN authentication. With this approach, the SPN `nfs/fqdn.realm.com@REALM.COM` (defined in the Kerberos interface commands) maps to the UNIX user of your choosing. In the following examples, we map the SPN to the "pcuser."

To create the name mapping in ONTAP System Manager prior to ONTAP 9.7, go to SVM > SVM Settings under Host Users and Groups.

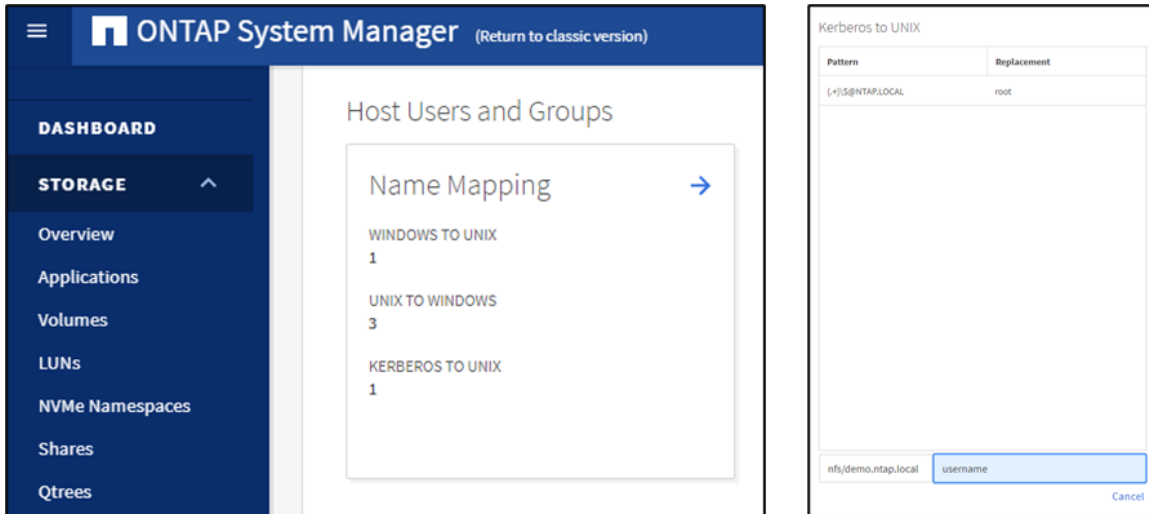


To create the user and group in ONTAP System Manager 9.7 and later, complete the following steps:

1. Go to Storage > Storage VMs and select the desired SVM.
2. Scroll down in the Settings tab to the Host Users and Groups section.



- Click the arrow.
- Click Add under the Kerberos to UNIX rules or click on existing name mapping rules to edit.



To create the name mapping in the CLI, run the following command:

```
vserver name-mapping create -vserver [SVM] -direction krb-unix -position 1 -pattern
nfs/fqdn.realm.com@REALM.COM -replacement pcuser
```

## Create a UNIX user or a name-mapping rule to map the NFS client principal

When an NFS client attempts to mount NFS exports in ONTAP through Kerberos, the client's principal is passed to ONTAP for authentication. The principal that the client uses for authentication depends on how Kerberos was configured on that NFS client. You can view which principals would potentially be used from the keytab on the client by using `klist -kte`.

When joining a domain by using `realm` or `net ads`, the principal is usually sent as `MACHINEACCOUNT$@REALM.COM` by default. In some cases, RHEL uses either `nfs/hostname` (in versions earlier than RHEL 6.x) or `root/hostname` (generally, with [manual keytab creation](#)) as the SPN.

When the `mount` command is issued, that principal is sent by the NFS client to ONTAP, which attempts to perform a `krb-unix` name mapping. The default behavior for clients that were joined to the domain is for ONTAP to look for a UNIX user named `MACHINEACCOUNT$`, which is a 1:1 mapping of the NFS client SPN `MACHINEACCOUNT$@REALM.COM`. If that user does not exist in local files or name services configured for the SVM, then ONTAP looks for an explicit name-mapping rule. If no explicit name-mapping rule exists, the NFS Kerberos mount attempt fails with a permission or access issue. ONTAP logs the failure in EMS as a `secd` error.

To see the EMS event that is logged, use the following command.

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

You can approach this task in one of two ways:

- Create an explicit name-mapping rule for the SPN/UPN to map to an existing valid UNIX user.
- Create a UNIX user named `MACHINEACCOUNT$` for implicit name mapping either locally or in LDAP (if LDAP is configured).

For more information about Kerberos to UNIX name mapping, see the section “KRB-UNIX name mapping behavior.”

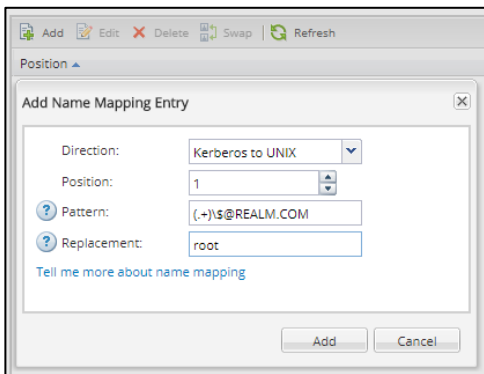
**Note:** In the cases where `nfs/hostname` or `root/hostname` are used as the SPNs, you should create UNIX users named “nfs” or “root.” Root is always a default user in ONTAP, so no action should be required in that case.

## Option 1: Creating a name-mapping rule (recommended)

Rather than creating multiple UNIX users for RHEL clients, it makes more sense to create a global name-mapping rule to map all Linux clients that attempt to authenticate as `MACHINEACCOUNT$@REALM.COM` to root. Mapping an account to root does not grant root access to anyone except for the root user. Other user principals that access the mount need to authenticate as themselves by using a username and password. You can create this global name-mapping rule in ONTAP System Manager or through the CLI.

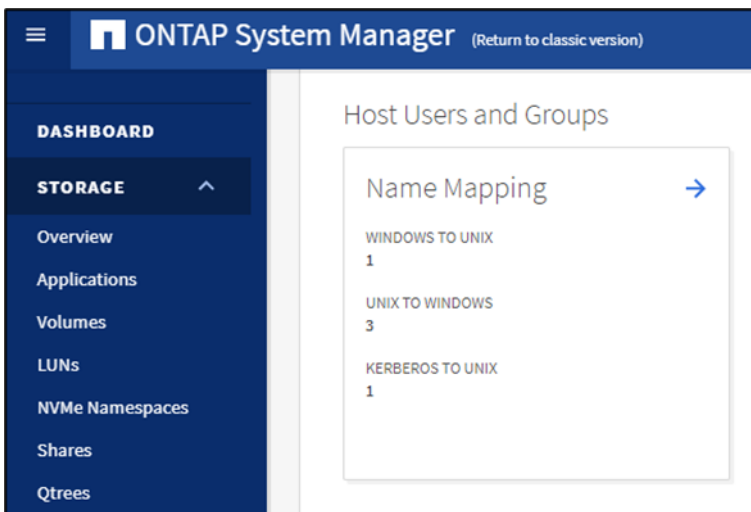
In the following examples, we create a rule that maps any computer account name that attempts Kerberos access to root through the regular expression of `(.+)\$`. This name-mapping rule does not map user principals to root; it only maps machine accounts (unless users are named `user$@REALM.COM`).

To create the name mapping in ONTAP System Manager prior to ONTAP 9.7, go to SVM > SVM Settings, under Host Users and Groups.



To create the user and group in ONTAP System Manager 9.7 and later, complete the following steps:

1. Go to Storage > Storage VMs and select the desired SVM.
2. Scroll down in the Settings tab to the Host Users and Groups section.
3. Click the arrow.
4. Click Add under the Kerberos to UNIX rules or click on existing name mapping rules to edit.



To create the name mapping in the CLI, run the following command:

```
vserver name-mapping create -vserver [SVM] -direction krb-unix -position 1 -pattern  
(.+)\\$@REALM.COM -replacement root
```

Using the “root” user here allows the client’s root user to behave as “root.” If you want to squash root access, you can map the incoming SPNs to a different UNIX user.

**Note:** The incoming NFS client principal is highly dependent on the client Linux version and how you configured Kerberos. Verify the incoming SPN (using packet captures or viewing `ktlist -kte`) when you determine how to create the name mapping. For example, some clients might use `host/name.realm.com` as their Kerberos principal. Event `log show` in the ONTAP CLI can also deliver details about which principal is trying to authenticate when failures occur.

To test the name mappings, run the following command:

```
set diag; diag secd name-mapping show -node [node] -vserver [SVM] -direction krb-unix -name  
[MACHINEACCOUNTNAME$@DOMAIN.COM]
```

## Option 2: Creating a UNIX user and group (not recommended)

To create a UNIX user in ONTAP for NFS client principals that come in as `MACHINEACCOUNTNAME$@REALM.COM`, use either ONTAP System Manager or the command line to create a user and a group named “`MACHINEACCOUNTNAME$`” with any UID and GID that you choose. In general, service accounts use a range between 1 and 1,024 for UIDs and GIDs. Before you define a numeric UID or GID, make sure that it is not in use elsewhere in your environment. You can also perform this task in LDAP by using the existing machine account object that is created by modifying LDAP attributes.

Keep in mind that when this is done, the root user no longer appears as “root” in NFS Kerberos mounts; instead, it reads and write files as whichever UID is assigned to the user you created.

For example, if I create a local UNIX user named `CENTOS7$` with UID 599, the SPN `CENTOS7$@NTAP.LOCAL` maps to that user.

```
cluster::*> access-check name-mapping show -vserver DEMO -direction krb-unix -name  
CENTOS7$@NTAP.LOCAL  
'CENTOS7$@NTAP.LOCAL' maps to 'CENTOS7$'  
cluster::*> unix-user show -vserver DEMO -user CENTOS7$  
  
Vserver: DEMO  
User Name: CENTOS7$  
User ID: 599  
Primary Group ID: 1  
User's Full Name:
```

When mounting with NFS Kerberos, the file owner shows as `nobody` when “root” writes a file because of how the SPN maps into ONTAP:

```
# id  
uid=0(root) gid=0(root) groups=0(root)  
# touch rootfile2  
# ls -la | grep rootfile2  
-rw-r--r-- 1 nobody daemon 0 May 21 13:53 rootfile2
```

The following example shows what ONTAP sees as the file owner—UID 599:

```
cluster::*> vserver security file-directory show -vserver DEMO -path /home/rootfile2  
  
Vserver: DEMO  
File Path: /home/rootfile2  
File Inode Number: 27980  
Security Style: unix  
Effective Style: unix
```

```
DOS Attributes: 20
DOS Attributes in Text: ---A----
Expanded Dos Attributes: -
    UNIX User Id: 599
    UNIX Group Id: 1
    UNIX Mode Bits: 644
    UNIX Mode Bits in Text: rw-r--r--
    ACLs: -
```

**Note:** Because an environment might contain hundreds of NFS clients that use Kerberos, NetApp does not recommend that you use this approach because it can interfere with scalability.

## Modify the NFS server machine account to allow only AES

Modifying the NFS server machine account to allow only AES prevents NFS clients from trying weaker (such as DES) or unsupported (such as RC4-HMAC) encryption types with ONTAP NFS mounts. When you enable Kerberos in ONTAP, an NFS-specific machine account is created in Active Directory (separate from any existing CIFS/SMB server machine accounts). In ONTAP 9.9.1 and later, ONTAP automatically populates this field with the values specified in the NFS server option permitted-enc-types, so modifying the machine account should no longer be necessary (see [bug 1316456](#) for more information).

**Note:** ONTAP 9.9.1RC1 has an issue with this fix, where the machine account will use unsupported encryption types (DES and RC4-HMAC) regardless of the NFS server settings. For that release, you will need to modify the encryption type attribute as described in the steps that follow. To avoid this issue, be sure to use the latest ONTAP 9.9.1 patch release.

Using Windows PowerShell is the easiest way to modify the NFS server's machine account:

```
PS C:\> Set-ADComputer NFS-KRB-NAME$ -KerberosEncryptionType AES256,AES128
```

If PowerShell is not an option, you can alternately use Active Directory Users and Computers to modify the msDs-SupportedEncryptionTypes field to the desired encryption type. For values you can use for this, see "Appendix B: Machine account attributes."

If modification of Active Directory machine accounts is not possible, you can modify the client's /etc/krb5.conf file to allow only AES by adding or modifying the following lines:

```
permitted_encetypes = aes256-cts-hmac-sha1-96 aes128-cts-hmac-sha1-96
default_tgs_encetypes = aes256-cts-hmac-sha1-96 aes128-cts-hmac-sha1-96
default_tkt_encetypes = aes256-cts-hmac-sha1-96 aes128-cts-hmac-sha1-96
```

## Red Hat Enterprise Linux client configuration

Before modern NFS clients added ways to join domains using a simple command, configuring NFS Kerberos with Active Directory KDCs was a manual process that required interaction from multiple teams. Client principals in Active Directory had to be manually created from the KDC, and keytab files had to be manually moved to clients, which then needed to be manually added to the main keytab file using `kutil`.

**Note:** If you need additional assistance with manual configuration steps for clients, refer to the OS vendor documentation.

Newer RHEL clients provide utilities that behave more like their Windows counterparts. The utilities allow the clients to automate the Kerberos configuration process by joining an Active Directory domain. When a domain is joined with `realmd`, the principal creation on the KDC, the client Kerberos configuration, and the keytab transfer are carried out automatically, without needing to touch the KDC. The following RHEL packages are recommended for joining clients to Active Directory domains:

- RHEL 6.x: Winbind/Samba (through net ads)
- RHEL 7.x and later: Realmd

As an added bonus, joining the Active Directory domain also configures the LDAP client SSSD to automatically use the Active Directory environment for UNIX Identity Management. However, some configuration might be required to make sure that SSSD is performing proper LDAP lookups.

For more information about configuring LDAP and SSSD considerations, see [TR-4835: How to Configure LDAP in ONTAP](#).

This configuration section makes the following assumptions about the RHEL clients:

- The RHEL client has forward (A/AAAA) and reverse (PTR) records in DNS.
- AES encryption is used.
- The RHEL client has the following packages installed (optional packages are denoted with \*):
  - nfs-utils, realmd, samba, samba-client, samba-winbind, autofs\*, ntp, bind-utils, tcpdump\*, sssd (or other LDAP client)\*, krb5-workstation, krb5-libs, auth-config-gtk

## Configure Network Time Protocol services

Configuring time services on the RHEL client helps prevent issues with [Kerberos time skew](#). To configure Network Time Protocol (NTP) services, run the following commands.

```
ntpdate [pool.ntp.org]
systemctl start ntpd.service
systemctl enable ntpd.service
```

## Verify DNS

This verification allows you to check that the client exists in DNS. DNS forward and reverse records are needed for proper Kerberos functionality. To verify DNS, run the following commands.

```
# nslookup [hostname/FQDN of SVM data LIFs]
# nslookup [IP address of SVM data LIFs]
# nslookup [hostname/FQDN of NFS client(s)]
# nslookup [IP address of NFS client(s)]
```

If the client is not in DNS, work with the DNS administrator to have it added or use the [dynamic DNS functionality](#) in RHEL.

## Join the domain

This step automatically creates a service principal/machine account for the NFS client in the KDC, a keytab file, configures SSSD, `/etc/nsswitch.conf`, and configures the client's `/etc/krb5.conf` file for Kerberos. Joining a domain requires a user account with access to create objects in the specified Active Directory container. The default container is `OU=Computers`, but you can specify it in the commands that you use:

- For RHEL 6.x, use `net ads`, because `realmd` doesn't exist on older clients.  
See the section "Configure an NFS client to use Kerberos with `net ads` join."
- For RHEL 7.x and later, use `realmd`.  
See the section "Configure an NFS client to use Kerberos with `realm` join."

## Modify the machine account principal

Although most client and KDC interaction is automated when you join the domain, there is a manual step of configuring the machine account principal to confirm that Kerberos works properly with NetApp ONTAP.

## Change the supported encyptes for the client machine account

This step is recommended to prevent the client from attempting RC4-HMAC Kerberos for NFS, which ONTAP does not support. For this step, use PowerShell to modify the `msDs-SupportedEncryptionTypes` value to use AES-256 and AES-128 only. This step is covered in the section “Modify the NFS server machine account to allow only AES.”

See the following example of failure when using RC4-HMAC:

```
6/29/2016 16:09:56 node03
WARNING      secd.nfsAuth.problem: vserver (parisi) General NFS authorization problem. Error:
RPC accept GSS token procedure failed
[ 0 ms] Using the NFS service credential for logical interface 1035 (SPN='nfs/parisi-nfs.core-
tme.netapp.com@CORE-TME.NETAPP.COM') from cache.
**[ 1] FAILURE: Failed to accept the context: Unspecified GSS failure. Minor code may
provide more information (minor: Encryption type ArcFour with HMAC/md5 not permitted).
```

## Optional: Add the machine account’s service principal to the userPrincipalName field

This step confirms that `kinit -k` works with the client, as well as any application (such as SSSD) that might need to use a machine account service principal for Kerberos.

See the following sample PowerShell command:

```
PS C:\> Set-ADComputer CENTOS7$ -KerberosEncryptionType AES256,AES128 -UserPrincipalName
HOST/centos7.ntap.local@NTAP.LOCAL
```

## Optional: Customize the `krb5.conf` file to bypass DNS canonicalization

If you want the client to avoid using DNS to create the NFS service principal (that is, you have multiple A records for the ONTAP SVM), then add the following option to `[libdefaults]` in `/etc/krb5.conf`.

```
dns_canonicalize_hostname = false
```

## Best practices

The following is a list of best practices for using NFS Kerberos in NetApp ONTAP. They are best practices, not requirements. By following these best practices, you can achieve optimal results, but not all the steps are necessary for Kerberos to work properly.

This list is not comprehensive. If you discover an issue with the best practices on this list or you want to suggest an addition, please send comments to us by following the instructions in the [Contact Us](#) section.

### ONTAP best practices

- Add the data LIFs that participate in NFS Kerberos to DNS with forward and reverse (PTR) records.
- Set up more than one data LIF per SVM for NFS Kerberos data access: preferably, one data LIF per node per SVM. This best practice is for performance and resiliency considerations. For more information on NAS data LIF best practices, see [TR-4067](#).
- If you are using more than one data LIF, consider creating A/AAAA records in DNS with the same FQDN to provide load balance functionality.
- If you want to use a DNS alias for client access, use a Canonical Name (CNAME).
- DNS records for the data LIFs in the SVM should match the name set for the NFS service principal that is used in the NFS Kerberos configuration of the data LIFs (through Kerberos interface commands).

- Before you configure NFS Kerberos, remove DES and DES3 encryption types from the `permitted-enc-types` option in the NFS server if you only intend on using AES encryption. Disabling DES and DES3 after you create principals requires an outage, because you have to re-create the machine accounts to generate new keytabs.
- If you use on-box DNS load balancing or off-box DNS load balancing with NFS Kerberos, enable NFS Kerberos on all data LIFs that participate in the DNS load balance zone.
- Create a local UNIX user or LDAP user named “nfs” to allow implicit `krb-unix` name mapping for the NFS service principal.
- Create a global name-mapping rule for `krb-unix` mapping of incoming NFS client machine accounts. Machine account principals attempt to map into ONTAP and should have a valid UNIX user to map to. For further information, see section 0 in this document.
- Keep the length of your machine account names to less than 15 characters, if possible. Remember, the machine account name is created using the SPN you specified in the `kerberos interface enable` command and can be overridden with the `-machine-account` option. If your machine accounts are not unique past 15 characters, machine account creation fails, because Active Directory does not allow duplicate machine account names. You can also [rename your machine accounts](#) after you create them.
- Configure ONTAP to use the same LDAP server as the NFS clients for identity management consistency. For LDAP configuration information, see [TR-4835: How to Configure LDAP in ONTAP](#).
- Verify that the SVM root volume (/) has an export policy rule that permits at least read access to clients. Read access is required to allow clients to traverse the top level of the namespace. See [TR-4067](#) for details.

## NFS client best practices

- Use NTP to keep NFS clients in sync with the time of the KDC and the cluster. Time skews outside of five minutes can cause outages for NFS Kerberos.
- Add forward and reverse (PTR) records to DNS for NFS clients that use Kerberos. The DNS fully qualified domain name (FQDN) should match the client principal and the Kerberos configuration contents for the Kerberos realm.
- Use `klist` and `kinit` commands to view keytabs and to test Kerberos functionality. Keep in mind that any non-root user who wants to access an NFS Kerberos mount must be able to `kinit` (log in) to the KDC before it can request tickets to access the mount.
- Set the timeout value for `rpcgssd` to `-T 60` for clients that hit timeout issues when mounting NFS Kerberos. For more information about how to set this value, see the NFS client OS guides.
- Using packet traces (`tcpdump`), `/var/log/messages`, and debug levels for `rpcgssd` and `mount` is the best way to troubleshoot most Kerberos issues. In many cases, access to the KDC and the ONTAP cluster is needed as well.
- On the KDC, make sure that the NFS client machine account has the appropriate encryption types enabled. For details about what the client can and cannot use, see the section called “Supported encryption types.”
- To avoid bugs in the NFS Kerberos stack, use the latest possible version of the client’s kernel.
- To configure NFS clients for Kerberos, use domain joins rather than manual Kerberos configuration.
- Be sure to consider Kerberos caches and ticket lifetimes when configuring or troubleshooting Kerberos; caches can affect Kerberos behavior when experiencing issues. See the section called “Kerberos caches” for more information.

## Windows KDC best practices

- Use `setspn /q` to search the KDC for duplicate SPNs. Duplicate SPNs cause access issues that can be hard to track down.



- Make liberal use of packet traces when you troubleshoot Kerberos issues.
- To avoid time skew issues, keep the KDC's time up-to-date and within five minutes of the ONTAP cluster and NFS clients.
- [Use PowerShell as a simple way to modify machine accounts.](#)
- Check the Event Viewer on the domain controllers for Kerberos errors and security errors when troubleshooting.
- Windows 2008 and later versions disable DES encryption by default. Use DES only if it is necessary. Use AES instead, which is enabled by default in Windows KDCs.
- Windows Active Directory currently defaults to RC4-HMAC as the encryption type for Kerberos. Because ONTAP does not support RC4-HMAC for NFS Kerberos, NetApp recommends removing RC4-HMAC as an option for NFS Kerberos clients and ONTAP servers. Section 0 explains how to modify the NFS client machine account. Section 0 covers how to modify the NFS server account.

## Sample configurations

This section presents a sample configuration for NFS Kerberos.

### NetApp ONTAP

#### Kerberos realm

```

KDC Vendor: Microsoft
KDC IP Address: x.x.x.y
KDC Port: 88
Clock Skew: 5
Active Directory Server Name: ONEWAY
Active Directory Server IP Address: x.x.x.y
Comment: -
Admin Server IP Address: x.x.x.y
Admin Server Port: 749
Password Server IP Address: x.x.x.y
Password Server Port: 464
Permitted Encryption Types: aes-256, aes-128

```

#### Kerberos interfaces

```

cluster::*> kerberos interface show -vserver DEMO -lif data*
(vserver nfs kerberos interface show)

```

Vserver	Logical Interface	Address	Kerberos SPN
DEMO	data	x.x.x.a	enabled nfs/demo.ntap.local@NTAP.LOCAL
DEMO	data2	x.x.x.b	enabled nfs/demo.ntap.local@NTAP.LOCAL

2 entries were displayed.

#### Pertinent NFS Server configuration options

```

cluster::*> nfs server show -vserver DEMO -fields permitted-enc-types
vserver permitted-enc-types
-----
DEMO      aes-256,aes-128

```

#### UNIX users and groups

```

cluster::*> unix-user show -vserver DEMO

```

Vserver	User Name	User ID	Group ID	Full Name
DEMO	nfs	500	500	
DEMO	nobody	65535	65535	



```

DEMO          pcuser          65534  65534
DEMO          root            0       1
4 entries were displayed.

cluster::*> unix-group show -vserver DEMO
Vserver       Name            ID
-----
DEMO          daemon          1
DEMO          nfs              500
DEMO          nobody           65535
DEMO          pcuser           65534
DEMO          root             0
5 entries were displayed.

```

## Name-mapping rules

```

cluster::*> vserver name-mapping show -vserver DEMO

Vserver:  DEMO
Direction: krb-unix
Position Hostname      IP Address/Mask
-----
1          -            -
                                Pattern: (.+)\$@NTAP.LOCAL
                                Replacement: root

Vserver:  DEMO
Direction: unix-win
Position Hostname      IP Address/Mask
-----
1          -            -
                                Pattern: root
                                Replacement: DEMO\\administrator
2 entries were displayed.

```

## Windows (machine accounts and principals)

### setspn

```

PS C:\> setspn /q nfs/demo.ntap.local
Checking domain DC=NTAP,DC=local
CN=KERBEROS,CN=Computers,DC=NTAP,DC=local
    nfs/KERBEROS
    HOST/KERBEROS
    HOST/nfs-demo-ntap-1.ntap.local
    nfs/nfs-demo-ntap-1.ntap.local
    nfs/demo.ntap.local

Existing SPN found!

```

**Note:** The machine account in the preceding sample was renamed from NFS-DEMO-NTAP-L to KERBEROS.

## NFS client machine account

```

PS C:\> Get-ADComputer -Properties * CENTOS7$

AccountExpirationDate      :
accountExpires              : 9223372036854775807
AccountLockoutTime         :
AccountNotDelegated        : False
AllowReversiblePasswordEncryption : False
AuthenticationPolicy        : {}
AuthenticationPolicySilo   : {}
BadLogonCount              : 0
badPasswordTime            : 0
badPwdCount                : 0
CannotChangePassword       : False
CanonicalName              : NTAP.local/Computers/CENTOS7

```

```

Certificates          : {}
CN                   : CENTOS7
codePage             : 0
CompoundIdentitySupported : {False}
countryCode          : 0
Created              : 5/15/2017 5:50:49 PM
createTimeStamp       : 5/15/2017 5:50:49 PM
Deleted              :
Description           :
DisplayName           :
DistinguishedName     : CN=CENTOS7,CN=Computers,DC=NTAP,DC=local
DNSHostName           : centos7.ntap.local
DoesNotRequirePreAuth : False
dSCorePropagationData : {12/31/1600 7:00:00 PM}
Enabled              : True
HomedirRequired       : False
HomePage              :
instanceType          : 4
IPv4Address           : x.x.x.x
IPv6Address           :
isCriticalSystemObject : False
isDeleted             :
KerberosEncryptionType : {AES128, AES256}
LastBadPasswordAttempt :
LastKnownParent       :
lastLogoff            : 0
lastLogon             : 131459819334568160
LastLogonDate          : 7/25/2017 1:40:51 PM
lastLogonTimestamp     : 131454780514971253
localPolicyFlags       : 0
Location              :
LockedOut              : False
logonCount             : 2402
ManagedBy             :
MemberOf              : {}
MNSLogonAccount        : False
Modified              : 7/25/2017 1:40:51 PM
modifyTimeStamp         : 7/25/2017 1:40:51 PM
msDS-SupportedEncryptionTypes : 24
msDS-User-Account-Control-Computed : 0
Name                  : CENTOS7
nTSecurityDescriptor   : System.DirectoryServices.ActiveDirectorySecurity
ObjectCategory          : CN=Computer,CN=Schema,CN=Configuration,DC=NTAP,DC=local
ObjectClass             : computer
ObjectGUID              : 3a50009f-2b40-46ea-9014-3418b8d70bdb
objectSid               : S-1-5-21-3552729481-4032800560-2279794651-1140
OperatingSystem         :
OperatingSystemHotfix   :
OperatingSystemServicePack :
OperatingSystemVersion  :
PasswordExpired         : False
PasswordLastSet         : 7/8/2017 12:06:54 AM
PasswordNeverExpires    : True
PasswordNotRequired     : False
PrimaryGroup            : CN=Domain Computers,CN=Users,DC=NTAP,DC=local
primaryGroupID          : 515
PrincipalsAllowedToDelegateToAccount : {}
ProtectedFromAccidentalDeletion : False
pwdLastSet              : 131439604148147009
SamAccountName : CENTOS7$
sAMAccountType          : 805306369
sDRightsEffective       : 15
ServiceAccount          : {}
ServicePrincipalName : {HOST/centos7.ntap.local, HOST/CENTOS7}
ServicePrincipalNames : {HOST/centos7.ntap.local, HOST/CENTOS7}
SID                     : S-1-5-21-3552729481-4032800560-2279794651-1140
SIDHistory              : {}
TrustedForDelegation    : False
TrustedToAuthForDelegation : False
UseDESKeyOnly           : False
userAccountControl      : 69632

```

```

userCertificate           : {}
UserPrincipalName      : HOST/centos7.ntap.local@NTAP.LOCAL
uSNChanged                : 95586
uSNCreated                : 77860
whenChanged               : 7/25/2017 1:40:51 PM
whenCreated               : 5/15/2017 5:50:49 PM

```

## NFS server machine account (ONTAP)

```
PS C:\> Get-ADComputer -Properties * KERBEROS
```

```

AccountExpirationDate    :
accountExpires            : 9223372036854775807
AccountLockoutTime       :
AccountNotDelegated      : False
AllowReversiblePasswordEncryption : False
AuthenticationPolicy      : {}
AuthenticationPolicySilo : {}
BadLogonCount            : 0
badPasswordTime          : 0
badPwdCount               : 0
CannotChangePassword     : False
CanonicalName             : NTAP.local/Computers/KERBEROS
Certificates              : {}
CN                        : KERBEROS
codePage                  : 0
CompoundIdentitySupported : {False}
countryCode              : 0
Created                   : 1/17/2017 4:24:36 PM
createTimeStamp           : 1/17/2017 4:24:36 PM
Deleted                   :
Description               :
DisplayName               : KERBEROS
DistinguishedName         : CN=KERBEROS,CN=Computers,DC=NTAP,DC=local
DNSHostName               : DEMO.NTAP.LOCAL
DoesNotRequirePreAuth     : False
dSCorePropagationData     : {12/31/1600 7:00:00 PM}
Enabled                   : True
HomedirRequired           : False
HomePage                  :
instanceType              : 4
IPv4Address               : x.x.x.b
IPv6Address               :
isCriticalSystemObject    : False
isDeleted                 :
KerberosEncryptionType : {AES128, AES256}
LastBadPasswordAttempt    :
LastKnownParent           :
lastLogoff                : 0
lastLogon                 : 0
LastLogonDate             :
localPolicyFlags          : 0
Location                  :
LockedOut                 : False
logonCount                : 0
ManagedBy                :
MemberOf                  : {}
MNSLogonAccount           : False
Modified                  : 7/13/2017 9:55:21 AM
modifyTimeStamp            : 7/13/2017 9:55:21 AM
msDS-SupportedEncryptionTypes : 24
msDS-User-Account-Control-Computed : 0
Name                      : KERBEROS
nTSecurityDescriptor      : System.DirectoryServices.ActiveDirectorySecurity
ObjectCategory            : CN=Computer,CN=Schema,CN=Configuration,DC=NTAP,DC=local
ObjectClass               : computer
ObjectGUID                : 2ade6c5d-1411-4cb1-ab84-e9a6228fd120
objectSid                 : S-1-5-21-3552729481-4032800560-2279794651-1116
OperatingSystem            : NetApp Release 9.1

```

```

OperatingSystemHotfix      :
OperatingSystemServicePack :
OperatingSystemVersion    :
PasswordExpired           : False
PasswordLastSet           : 1/17/2017 4:24:36 PM
PasswordNeverExpires      : False
PasswordNotRequired       : False
PrimaryGroup              : CN=Domain Computers,CN=Users,DC=NTAP,DC=local
primaryGroupID            : 515
PrincipalsAllowedToDelegateToAccount : {}
ProtectedFromAccidentalDeletion : False
pwdLastSet                : 131291618765754144
SamAccountName            : KERBEROS$
sAMAccountType            : 805306369
sDRightsEffective         : 15
ServiceAccount            : {}
servicePrincipalName       : {nfs/KERBEROS, HOST/KERBEROS, HOST/nfs-demo-ntap-
l.ntap.local, nfs/nfs-demo-ntap-l.ntap.local...}
ServicePrincipalNames      : {nfs/KERBEROS, HOST/KERBEROS, HOST/nfs-demo-ntap-
l.ntap.local, nfs/nfs-demo-ntap-l.ntap.local...}
SID                       : S-1-5-21-3552729481-4032800560-2279794651-1116
SIDHistory                : {}
TrustedForDelegation      : False
TrustedToAuthForDelegation : False
UseDESKeyOnly             : False
userAccountControl        : 4096
userCertificate           : {}
UserPrincipalName         :
uSNChanged               : 90841
uSNCreated               : 13490
whenChanged               : 7/13/2017 9:55:21 AM
whenCreated               : 1/17/2017 4:24:36 PM

```

## RHEL 7.x client

### DNS

```

cluster::*> dns show -vserver DEMO

Vserver: DEMO
Domains: NTAP.local
Name Servers: x.x.x.y
(DEPRECATED)-Enable/Disable DNS: enabled
Timeout (secs): 2
Maximum Attempts: 1
Is TLD Query Enabled?: true
Require Source and Reply IPs to Match: true
Require Packet Queries to Match: true

```

### krb.conf file

```

# cat /etc/krb5.conf
# Configuration snippets may be placed in this directory as well
includedir /etc/krb5.conf.d/

[logging]
default = FILE:/var/log/krb5libs.log
kdc = FILE:/var/log/krb5kdc.log
admin_server = FILE:/var/log/kadmind.log

[libdefaults]
dns_lookup_realm = false
ticket_lifetime = 24h
renew_lifetime = 7d
forwardable = true
rdns = false
default_ccache_name = KEYRING:persistent:%{uid}

```

```

default_realm = NTAP.LOCAL
[realms]

NTAP.LOCAL = {
}

[domain_realm]
ntap.local = NTAP.LOCAL
.ntap.local = NTAP.LOCAL

```

## Keytabs (using klist -k)

```

# klist -k
Keytab name: FILE:/etc/krb5.keytab
KVNO Principal
-----
  5 host/centos7.ntap.local@NTAP.LOCAL
  5 host/centos7.ntap.local@NTAP.LOCAL
  5 host/centos7.ntap.local@NTAP.LOCAL
  5 host/centos7.ntap.local@NTAP.LOCAL
  5 host/centos7.ntap.local@NTAP.LOCAL
  5 host/CENTOS7@NTAP.LOCAL
  5 host/CENTOS7@NTAP.LOCAL
  5 host/CENTOS7@NTAP.LOCAL
  5 host/CENTOS7@NTAP.LOCAL
  5 host/CENTOS7@NTAP.LOCAL
  5 CENTOS7$@NTAP.LOCAL
  5 CENTOS7$@NTAP.LOCAL
  5 CENTOS7$@NTAP.LOCAL
  5 CENTOS7$@NTAP.LOCAL
  5 CENTOS7$@NTAP.LOCAL
  5 HOST/centos7.ntap.local@NTAP.LOCAL
  5 HOST/centos7.ntap.local@NTAP.LOCAL
  5 HOST/centos7.ntap.local@NTAP.LOCAL
  5 HOST/centos7.ntap.local@NTAP.LOCAL
  5 HOST/centos7.ntap.local@NTAP.LOCAL

```

## Realm output

```

# realm list
NTAP.local
type: kerberos
realm-name: NTAP.LOCAL
domain-name: ntap.local
configured: kerberos-member
server-software: active-directory
client-software: sssd
required-package: oddjob
required-package: oddjob-mkhomedir
required-package: sssd
required-package: adcli
required-package: samba-common-tools
login-formats: %U@ntap.local
login-policy: allow-realm-logins

```

## Sample of working Kerberized homedir mount

### 1. Become a user.

```

# su profl
sh-4.2$ pwd
/root

```

### 2. Access is denied because you haven't "logged in" with kinit:

```

sh-4.2$ cd ~
sh: cd: /home/profl: Permission denied

```

### 3. Log in and view the TGT.

```
sh-4.2$ kinit
Password for prof1@NTAP.LOCAL:
sh-4.2$ klist -e
Ticket cache: KEYRING:persistent:1100:1100
Default principal: prof1@NTAP.LOCAL

Valid starting      Expires            Service principal
07/31/2017 11:32:31 07/31/2017 21:32:31  krbtgt/NTAP.LOCAL@NTAP.LOCAL
           renew until 08/07/2017 11:32:28, Etype (skey, tkt): aes256-cts-hmac-shal-96, aes256-cts-hmac-shal-96
```

#### 4. Navigate to the homedir, which is automounted to ONTAP by using NFSv4.1 and Kerberos.

```
sh-4.2$ cd ~
sh-4.2$ klist -e
Ticket cache: KEYRING:persistent:1100:1100
Default principal: prof1@NTAP.LOCAL

Valid starting      Expires            Service principal
07/31/2017 11:32:38 07/31/2017 21:32:31  nfs/demo.ntap.local@NTAP.LOCAL
           renew until 08/07/2017 11:32:28, Etype (skey, tkt): aes256-cts-hmac-shal-96, aes256-cts-hmac-shal-96
07/31/2017 11:32:31 07/31/2017 21:32:31  krbtgt/NTAP.LOCAL@NTAP.LOCAL
           renew until 08/07/2017 11:32:28, Etype (skey, tkt): aes256-cts-hmac-shal-96, aes256-cts-hmac-shal-96
sh-4.2$ pwd
/home/prof1
sh-4.2$ mount | grep prof1
demo:/home/prof1 on /home/prof1 type nfs4
(rw,nosuid,relatime,vers=4.1,rsize=1048576,wsiz=1048576,namlen=255,hard,proto=tcp,port=0,timeo=60,retrans=2,sec=krb5,clientaddr=x.x.x.x,local_lock=none,addr=x.x.x.b)
```

## Corner cases

The following section covers use cases that are outside the scope of the main document and yet are valid solutions to problems. If you have suggestions for issues to add to this section, please follow the steps in the [Contact Us](#) section of this document.

### Using the same machine account for CIFS/SMB and NFS Kerberos

When you create a CIFS server in ONTAP, a machine account is created in Windows Active Directory that holds the SPN information for that CIFS server and the keytab and machine account password periodically updated automatically for security purposes.

NFS Kerberos uses a separate dedicated machine account in Windows Active Directory. This is because the machine accounts cannot share the same keytab information with ONTAP. As a result, when the CIFS password is updated automatically, NFS Kerberos authentication will stop functioning. However, you can use the same hostnames and DNS entries for CIFS/SMB and NFS Kerberos, as CIFS/SMB Kerberos uses the `cifs/hostname` SPN, while NFS uses the `nfs/hostname` SPN.

The best practice is to leave NFS and CIFS/SMB Kerberos as separate machine accounts.

### Sharing keytabs on multiple clients

In the same style as using the same machine account for CIFS/SMB and NFS Kerberos, it is also not possible to use the same keytab file across multiple clients. This is a security feature of Kerberos. Clients must use unique keytabs to send the proper `padata-type: kRB5-PADATA-ENC-TIMESTAMP` information. If you try to use keytabs on multiple hosts, the `pa-data` doesn't get sent and authentication will fail.

## Using keytab files to kinit

In some cases, a service account needs to access a mount through NFS Kerberos. However, the service account, for a variety of reasons, might not be able to use a normal username and password to issue kinit logins to get Kerberos tickets.

In those scenarios, a Kerberos keytab file for authentication can be used, alongside scripts/cron jobs that renew the authentication periodically. With a keytab file entry, the `kinit` command can be run with the `-k` option to authenticate to the KDC periodically to refresh the credentials.

Creating a keytab on the KDC depends on the KDC in use; for Windows KDCs, use [ktpass](#). For other KDCs, refer to the KDC documentation.

When using keytab files for services, it might be necessary to add appropriate access to the [sudoers](#) file. Running `kinit` using keytab files requires elevated permissions on the client because the `/etc/krb5.keytab` file is set to 600 permissions with `root` as the owner.

```
[root@centos7 ~]# su oracle
sh-4.2$ kinit -k root/oracle@NTAP.LOCAL
kinit: Pre-authentication failed: Permission denied while getting initial credentials
```

After the user is added to `sudoers`, use `sudo` to `kinit` with the keytab file entry you created. Keytabs can be configured to not require a password.

```
sh-4.2$ sudo kinit -k root/oracle@NTAP.LOCAL
[sudo] password for oracle:
sh-4.2$ klist -e
Ticket cache: KEYRING:persistent:1999:krb_ccache_wii6eeV
Default principal: root/oracle@NTAP.LOCAL

Valid starting      Expires            Service principal
04/27/2020 22:39:52 04/27/2020 23:39:52  krbtgt/NTAP.LOCAL@NTAP.LOCAL
                    renew until 05/20/2020 22:39:52, Etype (skey, tkt): aes256-cts-hmac-shal-96, aes256-cts-hmac-shal-96
```

With the keytab entry, the service account can access NFS Kerberos mounts.

```
sh-4.2$ cd /kerberos
sh-4.2$ klist -e
Ticket cache: KEYRING:persistent:1999:krb_ccache_wii6eeV
Default principal: root/oracle@NTAP.LOCAL

Valid starting      Expires            Service principal
04/27/2020 22:47:01 04/27/2020 23:39:52  nfs/demo.ntap.local@NTAP.LOCAL
                    renew until 05/20/2020 22:39:52, Etype (skey, tkt): aes256-cts-hmac-shal-96, aes256-cts-hmac-shal-96
04/27/2020 22:39:52 04/27/2020 23:39:52  krbtgt/NTAP.LOCAL@NTAP.LOCAL
                    renew until 05/20/2020 22:39:52, Etype (skey, tkt): aes256-cts-hmac-shal-96, aes256-cts-hmac-shal-96
```

## Using local host files in place of DNS

In some cases, such as in [Cloud Volumes ONTAP](#) or [Azure NetApp Files](#) setups, DNS servers might not be readily available for use with host name resolution. Host name resolution is vital in Kerberos operations, because that is how the Kerberos SPN is passed to the KDC for ticket requests. For example, if a mount is performed to a FQDN of `SVM.domain.com`, then the Kerberos SPN that is requested would be `nfs/SVM.domain.com@DOMAIN.COM`. If DNS is not present, then `SVM.domain.com` is not able to find valid IP addresses. Additionally, reverse DNS lookups (IP to host name) would not be able to resolve an IP to a valid Kerberos SPN name for ticket retrieval.

In place of DNS, you can set local host file entries on the NFS client and ONTAP SVM.

The basic steps for this process include the following:

- Enable Kerberos on a data LIF in the SVM; at least one data LIF in the SVM must be able to contact the Windows KDC.
- Modify the NFS SPN machine account's `servicePrincipalName` to include the SPN `nfs/shortname`.
- Modify the NFS SPN machine account's `msDs-SupportedEncryptionTypes` value to use AES only (24).
- Add an entry to the NFS client's hosts file with the shortname and FQDN of the NFS SPN. For example, `nfs/svm.netapp.com` would need host entries for `svm` and `svm.netapp.com`.

When the local host file entries are present, the client can resolve hostnames to IP addresses, as well as IP addresses to hostnames, which can then formulate the NFS SPN request to the KDC.

See the following example:

```
[root@centos7 ~]# mount -o sec=krb5p DEM04:/home /kerberos
[root@centos7 ~]# umount /kerberos/
[root@centos7 ~]# mount -o sec=krb5p 10.x.x.y:/home /kerberos
[root@centos7 ~]# umount /kerberos/
[root@centos7 ~]# nslookup demo3
Server:      10.x.x.x
Address:     10.x.x.x#53

** server can't find demo3: SERVFAIL

[root@centos7 ~]# nslookup demo3.ntap.local
Server:      10.x.x.x
Address:     10.x.x.x #53

** server can't find demo3.ntap.local: NXDOMAIN
```

## Using non-Windows KDCs

When using a non-Windows KDC such as FreeIPA, MIT, Heimdal or another KDC, the Kerberos configuration process is essentially the same. ONTAP can be used to automate Kerberos configuration by interacting with the KDC to create the NFS SPN.

In place of automated interaction with the KDC, you can also manually transfer the NFS SPN keytabs from the KDC. The client configuration steps and DNS/hostname requirements remain the same as with Windows.

## FreeIPA Kerberos

FreeIPA KDC interaction with ONTAP does not currently use an automated process to configure Kerberos, because ONTAP uses `kadmin` commands, while FreeIPA uses the `ipa` command set. If you use FreeIPA as a KDC, then complete the following steps:

1. Add the NFS Kerberos SPN manually by using `ipa host-add` and specifying the AES encyptes.

```
ipa host-add demo-ipa.centos-ldap.local
ipa service-add nfs/demo-ipa.centos-ldap.local
ipa-getkeytab -p nfs/demo-ipa.centos-ldap.local -k ./nfs.keytab -e aes256-cts-hmac-sha1-96,aes128-cts-hmac-sha1-96
```

2. Copy the keytab file you created to a web server and then run the `kerberos interface enable` command by using the `-keytab-uri` option.

```
cluster::*> kerberos interface enable -vserver NFS -lif ipa-krb -spn nfs/demo-ipa.centos-ldap.local@CENTOS-LDAP.LOCAL -keytab-uri http://web-server/files/ipakrb-ontap.keytab
```



## DNS aliases/Canonical Names

When enabling Kerberos on a data LIF, the SPN is specified during the configuration. This SPN determines which host name is used to access Kerberized mounts. For example, if the SPN of `nfs/kerberos.domain.com` is used, then the mounts can be accessed with the FQDN of `kerberos.domain.com` or with the short host name of `kerberos`. DNS entries are needed because the host name used in the mount determines which SPN to pass to the KDC for authentication. If a DNS A/AAAA record is used to create an alias such as `nfskrb.domain.com`, then that host name is passed as the SPN to the KDC, and Kerberized mounts fail with an “access denied” error:

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

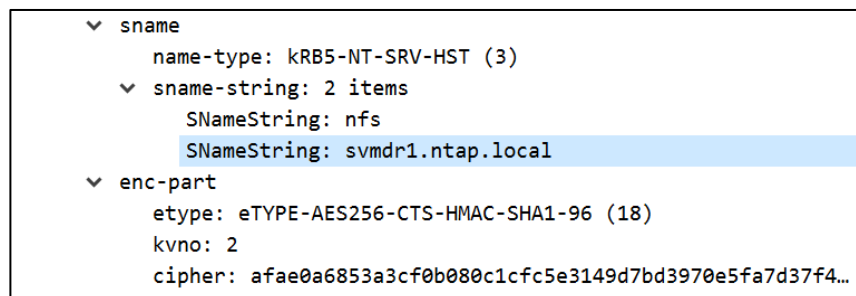
In a packet trace or corresponding Kerberos logs, you would see `PRINCIPAL_UNKNOWN` errors in Kerberos requests, because the DNS name being requested does not match the SPN. In the example above, `krb.domain.com != nfs/Kerberos.domain.com`, so access is denied.

To create a DNS alias properly, use a DNS CNAME that points to the DNS A/AAAA record associated with the NFS SPN. The DNS A/AAAA record the CNAME points to must use the same name used by the data LIF SPN. For example, if the Kerberos interface SPN is `nfs/kerberos.domain.com`, then the DNS A/AAAA record the CNAME points to needs to be `kerberos.domain.com`.

After you take this step, the DNS request is forwarded to the configured hostname, which is then used for Kerberos requests.

See the following example:

```
DNS 113 Standard query response 0x8e5f A svmdr.ntap.local CNAME svmdr1.ntap.local A x.x.x.x
DNS 97 Standard query response 0xeea2 AAAA svmdr.ntap.local CNAME svmdr1.ntap.local
DNS 77 Standard query 0x2632 A svmdr1.ntap.local
DNS 77 Standard query 0x8021 AAAA svmdr1.ntap.local
NFS 1438 V4 NULL Call
```



## NFS Kerberos in Cloud Volumes ONTAP

Cloud Volumes ONTAP can make use of NFS Kerberos to encrypt NFS communication over the wire for security-conscious storage administrators. There are no special configuration considerations for use with Cloud Volumes ONTAP because it is simply an ONTAP instance running in the cloud. If desired, you can configure the Cloud Volumes ONTAP instance to use NFS Kerberos without needing to connect to external name services such as LDAP or KDC by using manual keytab creation (as described in the section titled “Manual keytab configuration: Client and ONTAP”).

## IPSec: An alternative to encrypting your NFS packets

In ONTAP 9.8, IPSec support was introduced. This feature is able to encrypt any type of Ethernet communication, including NFS and iSCSI. IPSec offers the distinct advantage over NFS Kerberos of not requiring an elaborate name service and KDC setup just to achieve encryption over the wire. This is

particularly useful in Cloud Volumes ONTAP configurations where adding infrastructure can add billable hours to the solution. For more information on IPSec, see:

- [Episode 275: ONTAP 9.8 Security Updates \(featuring IPSec\)](#) (audio podcast)
- [How to configure multiple clients for IPsec for ONTAP 9.8 and higher](#)
- [TR-4569: Security Hardening Guide for NetApp ONTAP 9](#)

## NFS Kerberos with storage Virtual Machine Disaster Recovery

Storage Virtual Machine Disaster Recovery (SVM-DR) is an ONTAP feature that provides configuration replication of an SVM along with regular SnapMirror replication of data volumes. The steps needed to configure SVM-DR in your environment are covered in the NetApp Documentation.

SVM-DR replicates CIFS/SMB shares, DNS, name service configurations, and even Kerberos realms and interfaces. With SVM-DR, there are a few options for replication. The level of replication of SVM configuration directly affects how NFS Kerberos is configured on the source and destination systems. The configuration considerations are identical to the CIFS/SMB recommendations in the documentation.

### SVM-DR with identity-preserve set to true (all configuration is identical)

When you use NFS Kerberos with identity-preserve set to true (IP addresses and Kerberos realm is identical; hostnames stay the same), then you do not need to do anything different with NFS Kerberos. Failovers work as normal without any need for manual intervention.

### SVM-DR with identity-preserve set to true, but network interfaces are not replicated

In the case where IP addresses change from source to destination, NFS Kerberos must be enabled on both the source and destination SVMs with unique SPNs. This creates keytabs for each SVM that can be used for Kerberos interaction with clients. If you attempt to share the machine account/SPN in the KDC, then Kerberos mounts fail because clients are unable to find the proper authentication token on the destination systems.

Along with the unique SPNs, you also need to create DNS A/AAAA records for each SPN and then create a CNAME record (see the section DNS aliases/Canonical Names for more information) that points to the desired SVM's hostname.

When a failover occurs (either planned or unplanned), change the CNAME to point to the other SVM's hostname. DNS will redirect the hostname lookup to the appropriate A/AAAA record, which is then used for the Kerberos authentication.

See the following examples:

- SVMDR1 has IP `x.x.x.x` and Kerberos is enabled with the SPN `nfs/svmdr1.domain.com`.
- SVMDR2 has IP `y.y.y.y` and Kerberos is enabled with the SPN `nfs/svmdr2.domain.com`.
- DNS A/AAAA records for `svmdr1.domain.com` and `svmdr2.domain.com` are created.
- A CNAME record called `svmdr.domain.com` is created and pointed to `svmdr1.domain.com`.

When a CNAME record is queried, the following output is generated:

```
# nslookup svmdr.domain.com
Server:      x.x.x.z
Address:     x.x.x.z#53

svmdr.ntap.local canonical name = svmdr1.ntap.local.
Name:      svmdr1.ntap.local
Address:   x.x.x.x
```

When the CNAME is redirected to the other SVM, the CNAME changes to the configured IP:

```
# nslookup svmdr.domain.com
Server:      x.x.x.z
Address:     x.x.x.z#53

svmdr.ntap.local canonical name = svmdr2.ntap.local.
Name:   svmdr2.ntap.local
Address: y.y.y.y
```

NFS Kerberos mounts uses each unique SPN for mounts.

```
$ klist
Ticket cache: KCM:1587401110
Default principal: user@DOMAIN.COM

Valid starting     Expires            Service principal
06/10/2020 11:31:48 06/10/2020 11:41:44 nfs/svmdr1.domain.com@DOMAIN.COM
renew until 06/10/2020 21:31:44
06/10/2020 11:31:46 06/10/2020 11:41:44 krbtgt/DOMAIN.COM@DOMAIN.COM
renew until 06/10/2020 21:31:44
06/10/2020 11:34:47 06/10/2020 11:41:44 nfs/svmdr2.domain.com@DOMAIN.COM
renew until 06/10/2020 21:31:44
```

## SVM-DR with identity-preserve set to false

Setting identity-preserve to false must follow the same steps as the previous configuration (SVM-DR with identity-preserve set to true, but network interfaces are not ). Unique interfaces and NFS Kerberos SPNs must be configured, along with DNS entries.

The following examples mirror the examples from the previous section:

- SVMDR1 has IP `x.x.x.x` and Kerberos is enabled with the SPN `nfs/svmdr1.domain.com`.
- SVMDR2 has IP `y.y.y.y` and Kerberos is enabled with the SPN `nfs/svmdr2.domain.com`.
- DNS A/AAAA records for `svmdr1.domain.com` and `svmdr2.domain.com` are created.
- A CNAME record called `svmdr.domain.com` is created and pointed to `svmdr1.domain.com`.

## Manual keytab configuration: Client and ONTAP

In some cases, you might not be able to use automated client-side commands to create the keytab for Kerberos, such as with `realm join` or `net ads`.

Some reasons might include:

- Security restrictions on using Samba packages on Linux clients
- Airgapped name services (where only the client can access the KDC and DNS)
- Cloud Volumes ONTAP and on-premises KDC/DNS that cannot be reached

In these instances, manual keytab creation is needed, along with importing to the necessary client or ONTAP SVM:

- To create a keytab manually in Microsoft Active Directory, follow the steps in this: [Active Directory: Using Kerberos Keytabs to integrate non-Windows systems](#)
- To create a keytab manually in FreeIPA, follow the steps in “FreeIPA Kerberos.”
- To create a keytab manually in MIT or other Linux KDCs, use the process defined here: <https://web.mit.edu/kerberos/krb5-1.5/krb5-1.5.4/doc/krb5-install/The-Keytab-File.html>

## Manual keytab considerations

When creating a manual keytab, consider the following:

- Client keytabs should use either host/ or root/ as their SPNs (such as host/hostname.domain.com). Whichever you choose will require a corresponding UNIX user or name mapping rule (as described in “Machine account SPN to UNIX name mapping”).
- ONTAP keytabs should use `nfs/` for the SPN (such as `nfs/ontap.domain.com`). In older releases of ONTAP, you might need to create a UNIX user named `nfs` or a name mapping rule (as described in “NFS service SPN to UNIX name mapping”).
- User SPNs/UPNs accessing the mount does not require ONTAP to contact the KDC for ticket exchange; the Kerberos ticket exchange conversation occurs between the client and the KDC. Therefore, you could have a completely isolated ONTAP instance using Kerberos, provided the client can communicate with the KDC. However, user SPNs/UPNs requires a valid UNIX name mapping or resolvable UNIX user name as per the section called “User SPN to UNIX name mapping.”
- ONTAP only supports AES, DES3, and DES encryption types for NFS Kerberos; therefore, avoid using any encryption types other than those for the keytab.
- ONTAP can assign permitted encryption types in the NFS server. Even if you specify only AES\* encryption types in the keytab and NFS server, ONTAP still complains about other allowed encryption types.
- After a keytab file has been applied, if you want to modify encryption types later, you’ll need to create a new keytab and disable/enable the Kerberos interface; this will create an outage.
- If you need to disable kerberos on a data LIF in ONTAP but don’t have a connection to the KDC, use the `-force` option.

## Kerberos caches

When configuring and using NFS Kerberos, remember that things can get cached during the process, which can add to confusion when problems arise.

For example, if you are configuring NFS Kerberos and you hit an error condition, it is possible that the initial Kerberos ticket has been cached in the system and, even if you clear the cause of the error condition during troubleshooting, you might not see positive results until the Kerberos caches expire.

When a Kerberos NFS mount is performed to an ONTAP NFS server, ONTAP then caches a ticket in its subsystems and maintains that entry for the duration of the ticket expiry. This entry is kept in the Kerberos context cache, which is managed by the command `kerberos-context-cache` in `diag` privilege.

The following example shows what happens to the context cache during different points in the mount and NFS access process for mounts using Kerberos.

### Initial client mount

In the following example, a mount is performed to the ONTAP cluster by the root user at 15:22. The Kerberos ticket is set to expire in an hour, which leaves the expiration time as 16:22.

```
[root@centos7 ~]# mount -o sec=krb5p DEMO:/home /kerberos

cluster::*> kerberos-context-cache show -vserver DEMO -node node1
(diag nblade nfs kerberos-context-cache show)

Vserver      : DEMO
Node         : node1
```

Extent	Slot	Uid	Gid	Aux Gid Count	Idle Duration	Expiration	Enc Type	Key Data Length (Bytes)
0	1	0	1	1	0:0:3	4/27/2020 16:22:59	18	1192
0	2	0	1	1	0:0:3	4/27/2020 16:22:59	18	1192

## Initial NFS mount access by a user

In this example, the user `prof1` accesses the NFS mount using their Kerberos credentials at 15:24. Their ticket will expire at 16:24.

```
sh-4.2$ id
uid=1002(prof1) gid=10002(ProfGroup)
groups=10002(ProfGroup),1101(group1),1202(group2),1203(group3),1220(sharedgroup),10000(Domain
Users)
sh-4.2$ kinit
Password for prof1@NTAP.LOCAL:
sh-4.2$ cd /kerberos/
sh-4.2$ klist
Ticket cache: KEYRING:persistent:1002:1002
Default principal: prof1@NTAP.LOCAL
```

```
Valid starting      Expires      Service principal
04/27/2020 15:25:05 04/27/2020 16:24:59 nfs/demo.ntap.local@NTAP.LOCAL
        renew until 05/04/2020 15:24:59
04/27/2020 15:24:59 04/27/2020 16:24:59 krbtgt/NTAP.LOCAL@NTAP.LOCAL
        renew until 05/04/2020 15:24:59
```

```
cluster::*> kerberos-context-cache show -vserver DEMO -node node1
(diag nblade nfs kerberos-context-cache show)
```

```
Vserver      : DEMO
Node         : node1
```

Extent	Slot	Uid	Gid	Aux Gid Count	Idle Duration	Expiration	Enc Type	Key Data Length (Bytes)
0	0	0	1	1	0:0:21	4/27/2020 16:22:59	18	1192
0	1	0	1	1	0:0:32	4/27/2020 16:22:59	18	1192
0	2	0	1	1	0:0:52	4/27/2020 16:22:59	18	1192
0	3	1002	10002	6	0:0:8	4/27/2020 16:24:59	18	1192

## Subsequent mount access by users

Next, the user `student2` accesses the same NFS mount at 15:37. Their ticket expires in an hour.

```
sh-4.2$ id
uid=1302(student2) gid=1101(group1)
groups=1101(group1),1202(group2),1203(group3),1220(sharedgroup),10000(Domain Users)
sh-4.2$ kinit
Password for student2@NTAP.LOCAL:
sh-4.2$ cd /Kerberos
sh-4.2$ klist
Ticket cache: KEYRING:persistent:1302:1302
Default principal: student2@NTAP.LOCAL
```

```
Valid starting      Expires      Service principal
04/27/2020 15:37:21 04/27/2020 16:37:15 nfs/demo.ntap.local@NTAP.LOCAL
        renew until 05/04/2020 15:37:15
04/27/2020 15:37:15 04/27/2020 16:37:15 krbtgt/NTAP.LOCAL@NTAP.LOCAL
        renew until 05/04/2020 15:37:15
```

```
cluster::*> kerberos-context-cache show -vserver DEMO -node node1
(diag nblade nfs kerberos-context-cache show)
```

```
Vserver      : DEMO
Node         : node1
```

Extent	Slot	Uid	Gid	Aux Gid Count	Idle Duration	Expiration	Enc Type	Key Data Length (Bytes)
0	1	0	1	1	0:0:19	4/27/2020 16:22:59	18	1192
0	2	0	1	1	0:25:25	4/27/2020 16:22:59	18	1192

0	3	1002	10002	6	0:2:33	4/27/2020 16:24:59	18	1192
0	5	1302	1101	5	0:12:26	4/27/2020 16:37:15	18	1192

In the following example, another user accesses the Kerberos mount.

```
cluster::*> kerberos-context-cache show -vserver DEMO -node node1
(diag nblade nfs kerberos-context-cache show)
```

```
Vserver      : DEMO
Node         : node1
```

Extent	Slot	Uid	Gid	Aux Gid Count	Idle Duration	Expiration	Enc Type	Key Data Length (Bytes)
0	1	0	1	1	0:0:9	4/27/2020 16:22:59	18	1192
0	2	0	1	1	0:7:30	4/27/2020 16:22:59	18	1192
0	3	1002	10002	6	0:13:59	4/27/2020 16:24:59	18	1192
0	5	1302	1101	5	0:7:44	4/27/2020 16:37:15	18	1192
0	6	1301	1101	3	0:0:1	4/27/2020 16:59:42	18	1192

Note in the previous example that the slot numbers do not always show all entries. Those entries appear (or reappear) when the client requests specific operations. For example, when `student2` exits the prompt, we see two new entries that are actually older entries, given their slot table positions:

```
cluster::*> kerberos-context-cache show -vserver DEMO -node node1
(diag nblade nfs kerberos-context-cache show)
```

```
Vserver      : DEMO
Node         : node1
```

Extent	Slot	Uid	Gid	Aux Gid Count	Idle Duration	Expiration	Enc Type	Key Data Length (Bytes)
0	1	0	1	1	0:0:10	4/27/2020 16:22:59	18	1192
0	2	0	1	1	0:28:16	4/27/2020 16:22:59	18	1192
0	3	1002	10002	6	0:5:24	4/27/2020 16:24:59	18	1192
0	4	0	1	1	0:0:1	4/27/2020 16:22:59	18	1192
0	5	1302	1101	5	0:15:17	4/27/2020 16:37:15	18	1192
0	6	1302	1101	5	0:0:1	4/27/2020 16:37:15	18	1192

The enc type shows us the encryption type that the ticket has used. In this case, all our tickets are an enc type of 18, which maps to AES-256 as can be seen in the [list of Kerberos parameters](#).

The Kerberos cache in ONTAP can be cleared using the following commands:

```
cluster::*> kerberos-context-cache clear?
(diag nblade nfs kerberos-context-cache clear)
clear          *Clear the context cache entries
clear-all     *Clear the entire context cache
```

Clearing the context cache can get rid of stale entries that were populated during a Kerberos mount failure that might cause subsequent failures during initial configuration.

In addition to the context cache, credential caches need to be flushed to remove name-mapping entries for the `krb-unix` name mappings that occur during Kerberos authentication. However, caution should be taken during the process. See [bug1224820](#) for details.

**Note:** A NetApp support login might be required to view the bug link.

## Unmount impact on Kerberos context cache

When a client unmounts an NFS Kerberos mount, the context cache entries get removed. However, credentials from the `krb-unix` name mappings (such as `nfs/service` to UNIX user `nfs`) remain

cached. The default value is 24 hours and is controlled with the `name-service cache unix-user` commands.

```
cluster::*> name-service cache unix-user settings show -vserver DEMO
```

Vserver	Enabled	Negative-cache Enabled	TTL	Negative TTL	Propagation Enabled
DEMO	true	true	24h	1m	true

## NFS credential cache

When Kerberos mounts occur, several `krb-unix` name mappings are performed for authentication (as detailed in the section “KRB-UNIX name mapping behavior”). These mappings get cached in the NFS credential cache, which can impact Kerberos behavior during troubleshooting. For example, if you try to mount with Kerberos when no name mapping exists for the NFS service principal, caching can occur and cause failures even after you correct the initial issue.

Credential caches are managed via the `nfs credential` commands in advanced privilege.

```
cluster::*> nfs credentials ?
count          *Count credentials cached by NFS
flush          *Flush credentials cached by NFS
show           *Show credentials cached by NFS
```

NFS credential timeout settings are configured with the following NFS server options. These options can be modified to lower values if necessary.

```
cluster::*> nfs server show -vserver DEMO -fields cached-cred-positive-ttl,cached-cred-negative-
ttl,cached-cred-harvest-timeout
vserver cached-cred-positive-ttl cached-cred-negative-ttl cached-cred-harvest-timeout
-----
NFS      86400000                7200000                86400000
```

## Using -instance with the Kerberos context cache

When you specify `-instance` with the `Kerberos-context-cache`, you get a slew of useful information, such as Kerberos security flavor, encryption type strings, group lists and host information.

```
cluster::*> kerberos-context-cache show -vserver DEMO -slot-index 6 -extent-id 0
(diag nblade nfs kerberos-context-cache show)
```

```

      Vserver: DEMO
      Node: node1
      Extent ID: 0
      Slot Index: 6
      User ID: 1301
      Group ID: 1101
      Aux Gid Count: 3
      Expiration Time: 4/27/2020 16:59:42
      Last Used Time: 4/27/2020 16:42:40
      Idle Duration: 0:1:29
GSS Context (Network Bytes): c9eb619133980006
      Encryption Type: 18
      Encryption Type String: aes256-cts-hmac-sha1-96
      Key Data Length (Bytes): 1192
      Aux Gid List: 1101, 1203, 1220
      Reference Count: 1
      Is Marked For Deletion?: false
      Client IP Address: 10.x.x.x
      Logical Interface: data
      RPCSECGSS Service: krb5p
```

## Kerberos ticket lifetime – client cache

Some clients (such as Red Hat), cache Kerberos tickets for a period defined by the `ticket_lifetime` option in the `krb5.conf` file. The default value is 24 hours.

When Kerberos tickets are cached by the client, users that have authenticated already using Kerberos keep the ticket in local cache until it ages out. This is by design for performance considerations.

In those scenarios, even if the user runs a `kdestroy` command, Kerberos tickets remain cached and access is allowed. [Red Hat BugZilla 93891](#) covers this behavior.

If a lower ticket lifetime is desired, configure `krb5.conf` to use a lower value for `ticket_lifetime`.

**Note:** Clients cannot set the `ticket_lifetime` value higher than the KDC's value.

The Kerberos ticket lifetime when `ticket_lifetime` is not specified or commented out is whatever the KDC is set to. In this case, the ticket lifetime is one hour.

```
[libdefaults]
  dns_lookup_realm = false
  # ticket_lifetime = 24h
  # renew_lifetime = 24h

  # date
  Tue May 19 09:13:24 EDT 2020

  # klist
  Ticket cache: KEYRING:persistent:0:krb_ccache_V4nxJya
  Default principal: student1@NTAP.LOCAL

Valid starting      Expires              Service principal
05/19/2020 09:13:47 05/19/2020 10:13:47 krbtgt/NTAP.LOCAL@NTAP.LOCAL
    renew until 05/20/2020 09:13:44
```

When the file is set to 10 hours, the client still expires the ticket in one hour, as per the KDC's setting.

```
[libdefaults]
  dns_lookup_realm = false
  ticket_lifetime = 10h
  renew_lifetime = 10h

  # date
  Tue May 19 09:17:24 EDT 2020

  # klist
  Ticket cache: KEYRING:persistent:0:krb_ccache_V4nxJya
  Default principal: student1@NTAP.LOCAL

Valid starting      Expires              Service principal
05/19/2020 09:15:24 05/19/2020 10:15:24 krbtgt/NTAP.LOCAL@NTAP.LOCAL
    renew until 05/19/2020 19:15:21
```

If the ticket lifetime is set to 10 minutes, the expiration time changes to 10 minutes.

```
[libdefaults]
  dns_lookup_realm = false
  ticket_lifetime = 10m
  renew_lifetime = 10h

  # date
  Tue May 19 09:29:34 EDT 2020

  # klist
  Ticket cache: KEYRING:persistent:0:krb_ccache_V4nxJya
  Default principal: student1@NTAP.LOCAL

Valid starting      Expires              Service principal
05/19/2020 09:29:22 05/19/2020 09:39:20 krbtgt/NTAP.LOCAL@NTAP.LOCAL
```



**Note:** Changes to `krb5.conf` require a restart of the Kerberos service.

Ticket lifetimes can affect troubleshooting the Kerberos setup, so it might be prudent to change the ticket lifetime value on the client to a lower timeout during configuration efforts.

## Kerberos ticket expiration behavior

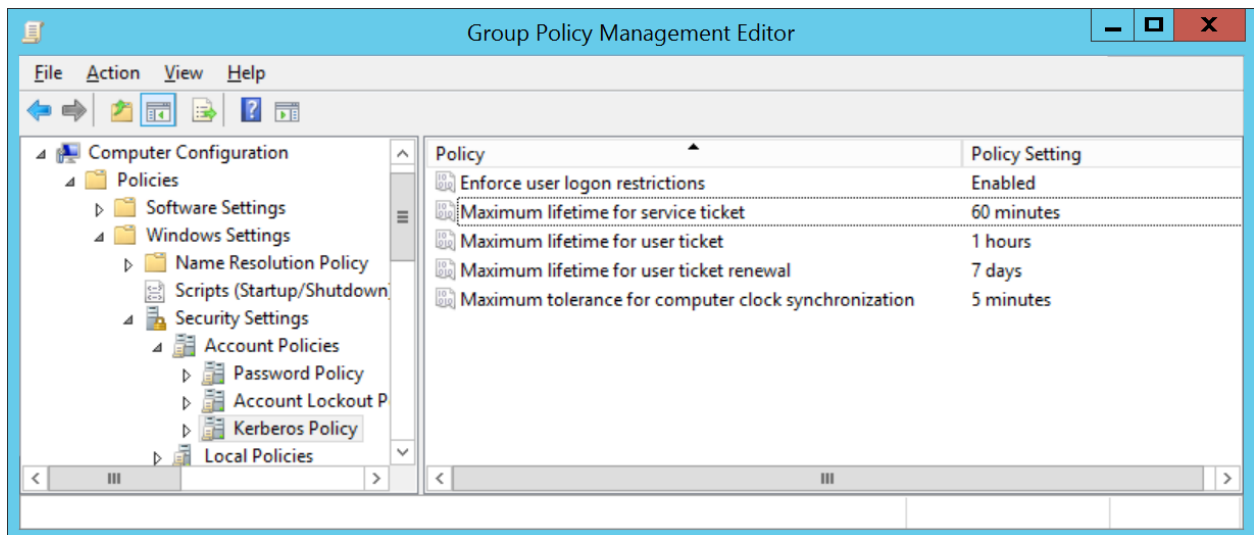
Kerberos ticket ages are configured on the KDCs and determine how long a Kerberos ticket remains valid. Ticket ages are set with the goal of balancing sensible security with load reduction on the KDC. For example, if you set a ticket age to a year, that might keep the load on the KDC lower, but it also increases security risks because your users only have to reauthenticate once a year. By default, Microsoft Kerberos tickets alive for 10 hours and requires renewal every 7 days. This can be overridden to a lower value with the client's `krb5.conf` file setting but cannot be set to a longer expiration time.

When a ticket expires, access to an NFS mount over Kerberos cannot succeed until the user re-authenticates to the KDC using `kinit`.

## Behavior when a Kerberos ticket expires

For example, the following user (`prof1`) requested a ticket that lasts an hour and accessed the NFS mount. We can see in Figure 7 that the user and service ticket both expire at 16:24:59. This is because our Windows KDC has been configured to use a one-hour ticket expiration policy. For information about configuring Kerberos ticket policies in Windows Active Directory, see: [Maximum lifetime for service ticket](#).

**Figure 7) Kerberos ticket lifetime management – Microsoft Windows Group Policy.**



Here we can see the list of Kerberos tickets and expiration times on the client.

```
sh-4.2$ klist
Ticket cache: KEYRING:persistent:1002:1002
Default principal: prof1@NTAP.LOCAL

Valid starting    Expires          Service principal
04/27/2020 15:25:05 04/27/2020 16:24:59 nfs/demo.ntap.local@NTAP.LOCAL
renew until 05/04/2020 15:24:59
04/27/2020 15:24:59 04/27/2020 16:24:59 krbtgt/NTAP.LOCAL@NTAP.LOCAL
renew until 05/04/2020 15:24:59
```

When the ticket expires, we can see that the ticket expiration date has been flushed:

```
sh-4.2$ date
```

```
Mon Apr 27 16:25:00 EDT 2020
sh-4.2$ klist
klist: Credentials cache keyring 'persistent:1002:1002' not found
```

And the NFS Kerberos mount is no longer accessible:

```
sh-4.2$ cd /kerberos
sh: cd: /kerberos: Not a directory
```

After we reauthenticate, we get access and new tickets with new expiration times.

```
sh-4.2$ kinit
Password for prof1@NTAP.LOCAL:
sh-4.2$ klist
Ticket cache: KEYRING:persistent:1002:1002
Default principal: prof1@NTAP.LOCAL

Valid starting      Expires            Service principal
04/27/2020 16:27:32  04/27/2020 17:27:32  krbtgt/NTAP.LOCAL@NTAP.LOCAL
        renew until 05/20/2020 16:27:32
sh-4.2$ cd /kerberos
sh-4.2$ klist
Ticket cache: KEYRING:persistent:1002:1002
Default principal: prof1@NTAP.LOCAL

Valid starting      Expires            Service principal
04/27/2020 16:27:39  04/27/2020 16:37:39  nfs/demo.ntap.local@NTAP.LOCAL
        renew until 05/20/2020 16:27:32
04/27/2020 16:27:32  04/27/2020 17:27:32  krbtgt/NTAP.LOCAL@NTAP.LOCAL
        renew until 05/20/2020 16:27:32
```

**Note:** We changed the service ticket expiration to 10 minutes to test the next sections.

## Behavior when only the service ticket expires

In some cases, the service ticket might expire before the user ticket does. When that happens, this is how the ticket appears on the client:

```
sh-4.2$ klist
Ticket cache: KEYRING:persistent:1002:1002
Default principal: prof1@NTAP.LOCAL

Valid starting      Expires            Service principal
klist: No credentials cache found while retrieving a ticket
```

If the user is in the NFS Kerberos mount, then `ls` fails until a new service ticket is acquired.

```
sh-4.2$ ls
ls: cannot open directory .: Permission denied
sh-4.2$ pwd
/kerberos
```

After a new ticket is acquired, access is restored:

```
sh-4.2$ kinit
Password for prof1@NTAP.LOCAL:
sh-4.2$ klist
Ticket cache: KEYRING:persistent:1002:1002
Default principal: prof1@NTAP.LOCAL

Valid starting      Expires            Service principal
04/27/2020 16:38:46  04/27/2020 17:38:46  krbtgt/NTAP.LOCAL@NTAP.LOCAL
        renew until 05/20/2020 16:38:46
sh-4.2$ ls
dir  dynamicuid  flexgroup  ftp  ftpuser  mtuser  nfs4  oracle  prof1  root  silly  student1
student2  test  unix
sh-4.2$ klist
Ticket cache: KEYRING:persistent:1002:1002
```

```

Default principal: prof1@NTAP.LOCAL

Valid starting      Expires      Service principal
04/27/2020 16:39:01 04/27/2020 16:49:01 nfs/demo.ntap.local@NTAP.LOCAL
        renew until 05/20/2020 16:38:46
04/27/2020 16:38:46 04/27/2020 17:38:46 krbtgt/NTAP.LOCAL@NTAP.LOCAL
        renew until 05/20/2020 16:38:46

```

## Behavior when a ticket is manually destroyed

When a ticket is manually destroyed using `kdestroy` on the client, access is allowed until the Kerberos ticket has expired or has been manually cleared from the ONTAP cache.

```

[root@centos7 /]# su student1
sh-4.2$ klist
Ticket cache: KEYRING:persistent:1301:1301
Default principal: student1@NTAP.LOCAL

Valid starting      Expires      Service principal
04/27/2020 16:01:12 04/27/2020 16:59:42 nfs/demo.ntap.local@NTAP.LOCAL
        renew until 05/04/2020 15:59:42
04/27/2020 15:59:42 04/27/2020 16:59:42 krbtgt/NTAP.LOCAL@NTAP.LOCAL
        renew until 05/04/2020 15:59:42
sh-4.2$ kdestroy
sh-4.2$ klist
klist: Credentials cache keyring 'persistent:1301:1301' not found

```

After the ticket is removed from the ONTAP SVM, access is denied for users that have destroyed the credentials until they reauthenticate.

```

cluster::*> kerberos-context-cache clear -vserver DEMO
(diag nblade nfs kerberos-context-cache clear)

Warning: This command removes all context cache entries for the Vserver "DEMO" on node "node1".
Do you want to continue? {y|n}: y

Successfully removed 3 context cache entries on node "cluster-01". The entries which were in use
will be removed when they are no longer used.

sh-4.2$ klist
klist: Credentials cache keyring 'persistent:1301:1301' not found
sh-4.2$ ls
ls: cannot open directory .: Permission denied
sh-4.2$ pwd
/kerberos

```

After the user reissues `kinit`, they regain access.

```

sh-4.2$ kinit
Password for student1@NTAP.LOCAL:
sh-4.2$ ls
dir dynamicuid flexgroup ftp ftpuser mtuser nfs4 oracle prof1 root silly student1
student2 test unix
sh-4.2$ klist
Ticket cache: KEYRING:persistent:1301:1301
Default principal: student1@NTAP.LOCAL

Valid starting      Expires      Service principal
04/27/2020 16:50:17 04/27/2020 17:00:17 nfs/demo.ntap.local@NTAP.LOCAL
        renew until 05/20/2020 16:50:15
04/27/2020 16:50:15 04/27/2020 17:50:15 krbtgt/NTAP.LOCAL@NTAP.LOCAL
        renew until 05/20/2020 16:50:15

```

## NFS Kerberos performance testing

The Customer Proof of Concept lab recently performed testing for NFSv3 and NFSv4.1 using NFS Kerberos (krb5, krb5i, krb5p). These numbers are not intended to show what the best possible performance is but, instead, to show what the impact of Kerberos is on NFS operations.

- The testing suite used was [vdbench](#) with a 50/50 read/write mix, 100% random, at 32K blocksize.
- A single CentOS 7.9 client was used to mount a NetApp A400 system running ONTAP 9.8 with an MTU size of 9000 and rsize/wsize of 64K. The average of 4 test runs were used to calculate these values.
- NFSv4.1 was using [pNFS](#).

Table 3 lists the MFS Kerberos results.

**Table 3) NFS Kerberos results.**

Test	Average IOPS	Average throughput (MB/s)	Average latency (ms)
NFSv3 - sys	~23839	~745	~0.5
NFSv3 – krb5	~12158	~380	~2.9
NFSv3 – krb5i	~10688	~334	~1.1
NFSv3 – krb5p	~5633	~176	~2.2
NFSv4.1 - sys	~24102	~753	~0.5
NFSv4.1 – krb5	~11351	~355	~3.2
NFSv4.1 – krb5i	~10856	~338	~1.1
NFSv4.1 – krb5p	~5579	~174	~2.1

Table 4 lists the MFS Kerberos performance comparison.

**Table 4) NFS Kerberos: Performance comparison versus nonencrypted baseline.**

Test	IOPS	Throughput (MB/s)	Latency (ms)
NFSv3 – krb5	-49%	-49%	+480%
NFSv3 – krb5i	-55%	-55%	+120%
NFSv3 – krb5p	-76%	-76%	+340%
NFSv4.1 – krb5	-53%	-53%	+540%
NFSv4.1 – krb5i	-55%	-55%	+120%
NFSv4.1 – krb5p	-77%	-77%	+320%

### Observations

Overall, NFS Kerberos substantially decreases performance for all NFS versions. IOPS and throughput are decreased since the storage will take longer to process encrypted packets and latency is increased during that time as well. Not all workloads will be impacted the same, so be sure to test in your environment.

## Common issues

This section covers some of the most common issues that arise in the process of configuring NFS Kerberos in NetApp ONTAP. It also explains how issues manifest and how to resolve them. This section is not comprehensive, but it does try to present some of the most commonly seen problems. Because

Kerberos has numerous moving parts, you might have to involve storage administrators, the Windows KDC administrator, DNS administrators, and NFS client administrators.

## Export policy troubleshooting

Export policies are typically the cause of issues in the following scenarios:

- Mounts fail.
- Reads or writes fail.
- Root access fails.
- Users do not show correct file ownership.
- Too much access is allowed.

These issues show up on clients as access denied, a read-only file system, or other errors that can span a variety of root causes. But, as a general rule, checking the export policy and rules is one of the first steps you should take when troubleshooting NFS access problems.

## Export-policy check-access

The easiest way to check access to NFS exports is to use `export-policy check-access`, which allows you to check an export policy's access rule set against a client's access to help determine if an export policy rule is working properly for predeployment as well as troubleshooting.

This command uses the normal name service communication and cache interaction that a standard mount from an NFS client would use.

See the following example of `export-policy check-access`:

```
cluster1::*> vserver export-policy check-access -vserver vsl -client-ip x.x.x.x -volume flex_vol
-authentication-method sys -protocol nfs3 -access-type read
```

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

This check also traverses the parent exports to make sure that all portions of the path have the intended access. After an export check fails, it stops at the volume or qtree it failed. In the following example, the vsroot export policy denies access to the volume flexvol. Vsroot must allow read access in the export policy for clients to traverse other volumes for mount. For information about locking down vsroot access, see [TR-4067: Network File Systems \(NFS\) in NetApp ONTAP](#).

See the following example of `export-policy check-access` failure at vsroot:

```
cluster::*> export-policy check-access -vserver DEMO -volume flexvol -client-ip x.x.x.x -
authentication-method sys -protocol nfs3 -access-type read
```

Path	Policy	Policy Owner	Policy Owner Type	Rule Index	Access
/	<b>empty</b>	<b>vsroot</b>	<b>volume</b>	<b>0</b>	<b>denied</b>

This command can also be used to check Kerberos access. This check verifies if the export policy and rules allow Kerberos access. It does not check if the entire Kerberos mount will succeed (such as name mapping, SPN lookups, passwords, and so on).

See the following example of `export-policy check-access` for Kerberos access:

```
cluster::*> export-policy check-access -vserver DEMO -volume flexvol -client-ip x.x.x.x -
authentication-method krb5p -protocol nfs4 -access-type read-write
```

Path	Policy	Policy Owner	Policy Owner Type	Rule Index	Access
/	default	vsroot	volume	2	read
/flexvol	default	flexvol	volume	2	read-write

## Export-policy caches

Export policy rules, client host names, and netgroup information are cached in ONTAP to reduce the number of requests made to the cluster. Caching helps improve performance of requests, as well as alleviating the load on networks and name service servers.

### Clientmatch caching

When a clientmatch entry is cached, it is kept local to the SVM and is then flushed after the cache timeout period is reached or if the export policy rule table is modified. The default cache timeout period is dependent on the version of ONTAP and can be verified using the command `export-policy access-cache config show` in admin privilege.

In ONTAP 9.7, these are the default values:

```
TTL For Positive Entries (Secs): 3600
TTL For Negative Entries (Secs): 3600
Harvest Timeout (Secs): 86400
```

To view a specific client in the `export policy access-cache`, run the following advanced privilege command:

```
cluster::*> export-policy access-cache show -node node2 -vserver NFS -policy default -address
x.x.x.x

Node: node2
Vserver: NFS
Policy Name: default
IP Address: x.x.x.x
Access Cache Entry Flags: has-usable-data
Result Code: 0
First Unresolved Rule Index: -
Unresolved Clientmatch: -
Number of Matched Policy Rules: 1
List of Matched Policy Rule Indexes: 2
Age of Entry: 11589s
Access Cache Entry Polarity: positive
Time Elapsed since Last Use for Access Check: 11298s
Time Elapsed since Last Update Attempt: 11589s
Result of Last Update Attempt: 0
List of Client Match Strings: 0.0.0.0/0
```

To flush an individual export policy cache entry, run the following command:

```
cluster::*> export-policy access-cache flush -vserver DEMO -node node1 -policy default -address
x.x.x.x
```

### Host name/DNS caching

When a clientmatch is set to a host name, the name is then resolved to an IP address. This happens based on the order the SVM's name service-switch (`ns-switch`) uses. For example, if the `ns-switch` host database is set to `files,dns`, then ONTAP searches for the client match in local host files and then searches DNS.

After a name lookup, ONTAP caches the result in the hosts cache. This cache's settings are configurable and can be queried and flushed from the ONTAP CLI in advanced privilege.

1. Query the cache.

```
cluster::*> name-service cache hosts forward-lookup show -vserver NFS
(vserver services name-service cache hosts forward-lookup show)
      IP      Address IP      Create
Vserver Host Protocol Family Address Source Time      TTL(sec)
-----
NFS      centos7.ntap.local
          Any      Ipv4      x.x.x.x  dns      3/26/2020 3600
                                   16:31:11
```

## 2. View the hosts cache settings.

```
cluster::*> name-service cache hosts settings show -vserver NFS -instance
(vserver services name-service cache hosts settings show)

      Vserver: NFS
      Is Cache Enabled?: true
Is Negative Cache Enabled?: true
      Time to Live: 24h
      Negative Time to Live: 1m
      Is TTL Taken from DNS: true
```

In some cases, if an NFS client's IP address changes, the hosts entry might need to be flushed to correct access issues.

## 3. Flush a host's cache entry.

```
cluster::*> name-service cache hosts forward-lookup delete -vserver NFS ?
      -host      -protocol -sock-type -flags      -family
```

## Netgroup caching

If you are using netgroups in the clientmatch field for export rules, then ONTAP performs additional work to contact the netgroup name service server to unpack the netgroup information. The netgroup database in ns-switch determines the order in which ONTAP queries for netgroups. In addition, the method ONTAP uses for netgroup support depends on whether netgroup.byhost support is enabled or disabled. For more information about netgroup.byhost, see [TR-4835: How to Configure LDAP in ONTAP](#).

- If netgroup.byhost is disabled, then ONTAP queries the entire netgroup and populates the cache with all netgroup entries. If the netgroup has thousands of clients, then that process could take some time to complete. Netgroup.byhost is disabled by default.
- If netgroup.byhost is enabled, then ONTAP queries the name service only for the host entry and the associated netgroup mapping. This greatly reduces the amount of time needed to query for netgroups, because we do not need to look up potentially thousands of clients.

These entries are added to the netgroup cache, which is found in `vserver services name-service cache` commands. These cache entries can be viewed or flushed, and the timeout values can be configured.

View the netgroups cache settings.

```
cluster::*> name-service cache netgroups settings show -vserver NFS -instance
(vserver services name-service cache netgroups settings show)

      Vserver: NFS
      Is Cache Enabled?: true
Is Negative Cache Enabled?: true
      Time to Live: 24h
      Negative Time to Live: 1m
      TTL for netgroup members: 30m
```

When an entire netgroup is cached, it is placed in the members cache.

```
cluster::*> name-service cache netgroups members show -vserver DEMO -netgroup netgroup1
(vserver services name-service cache netgroups members show)

      Vserver: DEMO
```

```
Netgroup: netgroup1
Hosts: sles15-1,x.x.x.x
Create Time: 3/26/2020 12:40:56
Source of the Entry: ldap
```

When only a single netgroup entry is cached, the ip-to-netgroup and hosts reverse-lookup caches are populated with the entry.

```
cluster::*> name-service cache netgroups ip-to-netgroup show -vserver DEMO -host x.x.x.z
(vserver services name-service cache netgroups ip-to-netgroup show)
Vserver   IP Address  Netgroup      Source  Create Time
-----
DEMO      x.x.x.z      netgroup1     ldap    3/26/2020 17:13:09

cluster::*> name-service cache hosts reverse-lookup show -vserver DEMO -ip x.x.x.z
(vserver services name-service cache hosts reverse-lookup show)
Vserver   IP Address  Host           Source  Create Time  TTL(sec)
-----
DEMO      x.x.x.z      centos8-ipa.centos-ldap.local
                                         dns      3/26/2020 17:13:09
                                         3600
```

## Cache timeout modification considerations

Cache configurations can be modified to different values if needed:

- Increasing the timeout values keeps cache entries longer but might result in inconsistencies in client access if a client changes its IP address (for example, if DHCP is used for client IP addresses and DNS does is not updated or if the export rule uses IP addresses).
- Decreasing the timeout values flushes the cache more frequently for more up-to-date information. However, this could add additional load to name service servers and add latency to mount requests from clients.

In most cases, leaving the cache timeout values intact is the best approach. For more information and guidance, see [TR-4668: Name Services Best Practices](#) and [TR-4835: How to Configure LDAP in ONTAP](#).

## Errors during Kerberos interface enable, modify, or create in ONTAP

If you see an error during the initial configuration of a data LIF for Kerberos or during modification of an existing data LIF, use Table 5 as a guide for resolving issues.

**Table 5) Identifying and resolving issues while creating or modifying Kerberos interfaces in ONTAP.**

Issue	How to view the error	Steps to resolution
The user who attempts to modify the interface does not have permissions on the KDC to create or modify machine accounts in the specified OU.	<ul style="list-style-type: none"><li>• Event log show</li><li>• Error output returned by the command when it fails</li></ul>	<ul style="list-style-type: none"><li>• Switch to a user who has access (such as a domain administrator).</li><li>• Delegate control of an OU to a user.</li><li>• Change the OU that is specified in the Kerberos configuration to an OU for which the user has create access.</li></ul>
Kerberos interface modification fails and cites the inability to connect to a valid KDC.	<ul style="list-style-type: none"><li>• Event log show</li><li>• Error output returned by the command when it fails</li></ul>	<ul style="list-style-type: none"><li>• Check the Kerberos realm configuration and ensure that it is configured properly.</li><li>• Check the DNS configuration for the SVM.</li><li>• Make sure that the data LIFs can route to the KDC.</li><li>• Confirm that the default route exists in the SVM.</li></ul>



Issue	How to view the error	Steps to resolution
Kerberos interface creation or modification fails and cites a time skew issue.	<ul style="list-style-type: none"> <li>Event log show</li> <li>Error output returned by the command when it fails</li> <li>Secd logs</li> </ul>	<ul style="list-style-type: none"> <li>Modify the cluster time to be within 5 minutes of the Windows KDC.</li> <li>Confirm that the time zone on the cluster matches the time zone on the KDC.</li> <li>Use NTP to sync the time across the environment.</li> </ul>
Kerberos interface creation fails with KRB5KDC_ERR_ETYPE_NOSUPP error.	<ul style="list-style-type: none"> <li>Event log show</li> <li>Error output returned by the command when it fails</li> <li>Secd logs</li> </ul>	<ul style="list-style-type: none"> <li>Confirm that the NFS server setting – <code>permitted-enc-types</code> matches the allowed Kerberos encryption types on the KDC.</li> <li>For example, if <code>-permitted-enc-types</code> is set to AES-256 and the KDC only allows DES, the command fails.</li> </ul>
Kerberos interface creation fails with cifs smb krb5 realm mismatch.	<ul style="list-style-type: none"> <li>Event log show</li> <li>Error output returned by the command when it fails</li> <li>Secd logs</li> </ul>	<ul style="list-style-type: none"> <li>Confirm that the username being used to enable Kerberos doesn't use a lowercase domain principal.</li> <li>For example, if the realm is NTAP.LOCAL, the user that is used to authenticate to the KDC should not use <code>user@ntap.local</code>, but instead should use <a href="#">user@NTAP.LOCAL</a> or just user.</li> </ul>

## Errors during mounting of NFS Kerberos from a client

If you see an error during the initial NFS mount through Kerberos from a client, Table 6 offers some potential issues for you to check. This information applies only to errors during the initial Kerberos mount attempt.

**Table 6) Identifying and resolving issues while mounting NFS Kerberos exports.**

Issue	How to view the error	Steps to resolution
Access/permission denied	<ul style="list-style-type: none"> <li>Mount command output</li> <li>Event log in ONTAP</li> <li>Packet trace</li> </ul>	<ul style="list-style-type: none"> <li>Check the export policy rules for the SVM root volume (/) and for the data volume (/path). If you're using qtree exports, check the policy for the qtree. Krb5 should be allowed in the <code>ro/rw</code> rules, and the NFS client should be allowed in the export policy rule's client match.</li> <li>Use <code>export-policy check-access</code> commands to verify that the specified client has access.</li> <li>Check the event log (<code>event log show</code>) in ONTAP for errors regarding <a href="#">krb-unix</a> name mapping for NFS clients. If there are errors, resolve the issue by creating local UNIX users or name-mapping rules. Generally, the NFS service principal/user does not apply to initial mounts. NFS principals authenticate when attempting to access Kerberos mounts.</li> <li>Check the event log (<code>event log show</code>) in ONTAP for errors regarding encryption types being unsupported. A common issue with Active Directory includes clients that are trying to use</li> </ul>

Issue	How to view the error	Steps to resolution
		RC4-HMAC to authenticate. ONTAP does not support RC4-HMAC with NFS Kerberos. To resolve this issue, <a href="#">modify the machine accounts</a> to remove RC4 from the list. After you modify the machine accounts, you might have to flush caches or disable/enable Kerberos to delete the keytab.
Protocol not supported	<ul style="list-style-type: none"> <li>Mount command output</li> <li>Packet trace</li> </ul>	<ul style="list-style-type: none"> <li>Verify which version of NFS is being mounted and compare it with the versions that are enabled in the ONTAP NFS server.</li> <li>Clients attempt to negotiate the highest NFS version that is enabled on a server.</li> <li>ONTAP supports NFSv3, NFSv4.0, and NFSv4.1 for NetApp FlexVol® volumes and supports NFSv3 for ONTAP FlexGroup volumes.</li> </ul>
No such file or directory	<ul style="list-style-type: none"> <li>Mount command output</li> <li>Packet trace</li> </ul>	<ul style="list-style-type: none"> <li>Verify that the path specified in the mount command exists in ONTAP as a junction path. This step can be performed in System Manager or through the CLI.</li> </ul>
Mount point "" does not exist	<ul style="list-style-type: none"> <li>Mount command output</li> </ul>	<ul style="list-style-type: none"> <li>Verify that the specified mount point folder exists on the local client.</li> </ul>
Incorrect mount option	<ul style="list-style-type: none"> <li>Mount command output</li> </ul>	<ul style="list-style-type: none"> <li>Check the mount command options that are specified. Do they actually exist as per client documentation?</li> <li>If specifying krb5, verify that the rpcgssd service is started and that SECURE_NFS is allowed on the NFS client.</li> </ul>
Mount hangs	<ul style="list-style-type: none"> <li>Mount command output</li> <li>Packet traces</li> </ul>	<ul style="list-style-type: none"> <li>If a mount is hanging, it means that the client or the server is failing to respond to a packet. Generally, this problem can be a network issue or a firewall or server configuration issue.</li> </ul>

## NFS Kerberos errors while attempting to access, read, or write

Table 7 covers issues that might occur after a Kerberos NFS export has successfully been mounted. With this category, traversal, reading, and/or writing issues an error.

**Table 7) Identifying and resolving issues in accessing Kerberos NFS exports in ONTAP.**

Issue	How to view the error	Steps to resolution
Access/permission denied when attempting to traverse the mount	Command-line output	<ul style="list-style-type: none"> <li>Check the event log (<code>event log show</code>) in ONTAP for errors regarding <a href="#">krb-unix</a> name mapping for the NFS service principal. If there are errors, resolve the issue by creating local UNIX users or name-mapping rules for the <code>nfs/name.realm.com</code> SPN.</li> <li>Use <code>export-policy check-access</code> commands to see whether the client is allowed to read and write through krb5 to the export.</li> <li>Verify that you actually mounted through Kerberos by issuing the mount command.</li> </ul>

Issue	How to view the error	Steps to resolution
		<ul style="list-style-type: none"> <li>• Verify that the export policy and rule allow <code>krb5</code> access to <code>ro</code> and <code>rw</code> rules.</li> <li>• Verify that you have used <code>kinit</code> to log in as a user to generate a Kerberos TGT.</li> <li>• Use <code>klist -e</code> to verify that the Kerberos ticket has not expired.</li> <li>• If you are root, check the ONTAP event logs to see who root is trying to authenticate as.</li> <li>• Use <code>vserver security file-directory show</code> to verify the file-level permissions to the export.</li> <li>• If the volume/qtree security style is NTFS, verify that the UNIX user who is attempting access has a valid UNIX-Windows name mapping.</li> </ul>
Issues when reading or writing to the export.	CLI output	<ul style="list-style-type: none"> <li>• Use <code>vserver security file-directory show</code> to verify the file-level permissions to the export.</li> <li>• If you're using NFSv4.x, verify that NFSv4.x is configured properly (see <a href="#">TR-4067</a>).</li> <li>• If you see Operation Not Permitted when you try to use <code>chown</code> or <code>chmod</code>, check the permissions on the folder and the export policy rule settings.</li> </ul>

**Note:** For other common NFS issues that are not listed in Table 7, see [TR-4067](#).

## Common event log errors in ONTAP related to NFS Kerberos

In some cases, viewing the event log in ONTAP can help isolate problems seen with Kerberos access and mounting. Table 8 shows a list of errors you can use to filter your event log, along with some common causes of the errors.

In the CLI, full descriptions of these errors can be seen with the command `event route show - messagename -instance`.

**Table 8) Common event log errors in ONTAP.**

EMS events	Common causes
secd.kerberos.clockskew secd.kerberos.lookupFailed secd.kerberos.noAuthdata secd.kerberos.preauth secd.kerberos.tktexpired secd.kerberos.tktnyv	In most cases, these errors pertain to Kerberos with SMB/CIFS servers.
secd.nfsAuth.noNameMap	This error pertains to a name mapping issue. With NFS Kerberos, that is generally the <code>krb-unix</code> name mapping for either the client SPN (such as <code>HOST\$@DOMAIN.COM</code> ) or a user SPN (such as <code>user@DOMAIN.COM</code> ). Review the error and add the appropriate name mappings. If an external name service like LDAP is used, check to see that LDAP is functioning properly and that LDAP queries from ONTAP are working.
secd.nfsAuth.problem	This error occurs for a variety of reasons in NFS Kerberos. Generally speaking, this is accompanied by a corresponding Kerberos-specific error, including the following:

EMS events	Common causes
	<ul style="list-style-type: none"> <li>• Unsupported encryption types</li> <li>• Name mapping errors</li> <li>• Decrypt integrity check</li> </ul> <p>Decrypt integrity check errors are fairly generic and require further troubleshooting, including packet traces and client logs. Typically, the error is caused when the ticket used by the client is not able to retrieve the proper NFS service credentials from ONTAP. In some cases, the issue can be resolved by restarting the client's Kerberos services or recreating the Kerberos configuration in the SVM.</p>
exports.*	Any exports* errors should be reviewed and investigated for potential misconfiguration issues. NFS Kerberos still uses the same configuration as NFS—a client must have access to the NFS export for mounts to work.
nfs.krb.lif.disabled	This error is fairly rare because it is only be triggered if the Kerberos credentials are accidentally shared across SVMs. If it does occur, Kerberos is disabled as a data LIF, which results in failed access from clients. Re-enabling Kerberos resolves the issue.

## Kerberos keytab troubleshooting

When troubleshooting whether your Kerberos keytab was configured correctly, there are a few things to consider:

- Is the SPN defined correctly?
- Did you create a principal on the KDC?
- Are the encryption types correct and supported?
- Is the keytab correctly imported?
- Are the keytab version numbers the same across all the client, server and KDC?

To help rule out these issues, there are a few troubleshooting tips you can use to verify keytab functionality.

## Event logs

Review the event logs. If a keytab file is the issue, you might receive an error about encryption types or about integrity checks. For details about errors you might see in the event log, see “Common event log errors in ONTAP related to NFS Kerberos.”.

## Key version number verification

Verify the [key version number \(kvno\)](#) on the client and on the ONTAP SVM.

To verify the kvno on the client/Linux KDC, run the following commands:

```
# kinit username
# kvno nfs/hostname.domain.com@REALM.COM
```

In ONTAP, the kvno is stored with the Kerberos keyblocks. They are created when you enable Kerberos on an interface. You can see the keyblock created for each encryption type in the keytab file. The command is `kerberos keyblocks show` at diag privilege.

**Note:** There are create, delete, and modify options for keyblocks; don't use these options unless directed by NetApp support.

```
cluster::*> kerberos keyblocks show ?
(vserver nfs kerberos keyblocks show)
[ -instance | -fields <fieldname>, ... ]
```

<code>[[[-service-type] {CIFS NFS}]</code>	<code>*Types of Service CIFS NFS</code>
<code>[ -vserver &lt;vserver&gt; ]</code>	<code>*Vserver ID</code>
<code>[[[-lif] &lt;integer&gt;]</code>	<code>*Logical Interface ID</code>
<code>[[[-key-version] &lt;integer&gt;]</code>	<code>*Key Version Number</code>
<code>[[[-encryption-type] &lt;integer&gt;]</code>	<code>*Encryption Type</code>
<code>[ -timestamp &lt;integer&gt; ]</code>	<code>*Time Stamp</code>
<code>[ -keyblock &lt;Hex String&gt; ]</code>	<code>*Keyblock</code>
<code>[ -spn &lt;text&gt; ]</code>	<code>*Service principal name</code>
<code>[ -machine-account &lt;text&gt; ]</code>	<code>*Machine Account Name</code>

What you want to see is that the kvno seen by the client matches what is seen in the keyblocks in ONTAP. If the kvno doesn't match, Kerberos authentication might not work ([Windows KDCs won't necessarily have this issue](#)). This is part of the reason that a CIFS/SMB machine account is unable to also be used for NFS Kerberos; CIFS/SMB constantly updates the machine password, which increments the keytab information. A packet trace can confirm if this is the root cause of the issue.

In the following example, we can verify that the kvno for the NFS service principal matches:

```
# kinit administrator
Password for administrator@NTAP.LOCAL:
[root@centos7 home]# kvno nfs/demo.ntap.local@NTAP.LOCAL
nfs/demo.ntap.local@NTAP.LOCAL: kvno = 1

cluster::*> kerberos keyblocks show -service-type NFS -vserver DEMO
(vserver nfs kerberos keyblocks show)
Service      Interface Key      Encryption
Type         Vserver  ID      Version  Type      Timestamp  SPN      Keyblock
-----
NFS          DEMO     1033    1        3         1588007407  nfs/demo.ntap.local@NTAP.LOCAL
a2a883ec540168f8
NFS          DEMO     1033    1        17        1588007407  nfs/demo.ntap.local@NTAP.LOCAL
70ec681fdc8f8183a893555901ff0385a
NFS          DEMO     1033    1        18        1588007407  nfs/demo.ntap.local@NTAP.LOCAL
4735f91db892917070c6e2ca1b1f4fdfa3d7dda0ace23226a3d28c82a37884a4
NFS          DEMO     1033    1        23        1588007407  nfs/demo.ntap.local@NTAP.LOCAL
701160a748ac404993b5fe4f0f388218
```

## Viewing the Kerberos context cache

As described in the “Initial NFS mount access by a user” section, you can see when a client uses a Kerberos ticket to gain access to a Kerberos mount in ONTAP. These cache entries can help you determine if the Kerberos authentication is working properly during access issues.

## Collecting and viewing packet traces for Kerberos

One of the most effective ways to troubleshoot Kerberos issues is with a packet capture. In small environments, simply collecting a trace on the NFS client during the Kerberos failure is enough. In larger environments, you might need to collect traces on the client, KDC, and ONTAP cluster. If there are multiple KDCs, then you might need to focus only on the ONTAP cluster and client at first.

### Collect packet traces

For information about collecting packet traces in ONTAP, see the following NetApp Knowledge Base articles:

- [How to Capture Packet Traces \(tcpdump\) on ONTAP 9.2+ Systems](#)
- [How to Collect Rolling Packet Traces Using pktt on ONTAP 9.1 and Below Systems](#)

On the NFS client, you can use `tcpdump` to collect a packet trace in a separate window. The following command would suffice:

```
# tcpdump -s 0 -i interfacename -w /filename.trc
```

**Note:** Try to only capture the failure to limit the size of the trace file.

For a Windows KDC, you can make use of Wireshark or your preferred packet capture method. In some cases, network traces on a Windows KDC won't be allowed due to security restrictions on which applications your environment allows to be installed on domain controllers.

## View packet traces

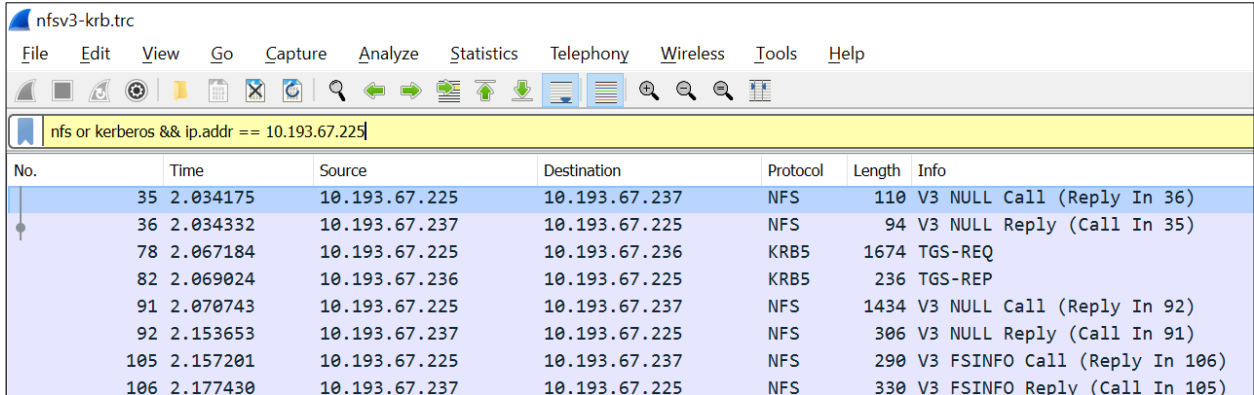
Any packet tracing application will work, but in this example, Wireshark was used to view Kerberos packet traces. A packet trace contains more information that you need, so you can make use of the filters provided.

For Kerberos, you generally need one or several of the following Wireshark filters:

- Kerberos
- IP.address
- DNS
- NFS
- Mount (NFSv3 only)

You can filter by one or several of the values above in the trace. For example, to filter out only NFS and Kerberos packets from the 10.193.67.225 client, use the filter in Wireshark shown in Figure 8.

**Figure 8) Wireshark filter example.**



The screenshot shows the Wireshark interface with the filter `nfs or kerberos && ip.addr == 10.193.67.225` applied. The packet list shows the following entries:

No.	Time	Source	Destination	Protocol	Length	Info
35	2.034175	10.193.67.225	10.193.67.237	NFS	110	V3 NULL Call (Reply In 36)
36	2.034332	10.193.67.237	10.193.67.225	NFS	94	V3 NULL Reply (Call In 35)
78	2.067184	10.193.67.225	10.193.67.236	KRB5	1674	TGS-REQ
82	2.069024	10.193.67.236	10.193.67.225	KRB5	236	TGS-REP
91	2.070743	10.193.67.225	10.193.67.237	NFS	1434	V3 NULL Call (Reply In 92)
92	2.153653	10.193.67.237	10.193.67.225	NFS	306	V3 NULL Reply (Call In 91)
105	2.157201	10.193.67.225	10.193.67.237	NFS	290	V3 FSINFO Call (Reply In 106)
106	2.177430	10.193.67.237	10.193.67.225	NFS	330	V3 FSINFO Reply (Call In 105)

In a Kerberos trace, you can collect a large amount of information, such as the Kerberos SPNs being used (NFS client, ONTAP NFS service, user SPN), the encryption types being used, the kvno, and whether the GSS payload is being sent and received properly. In addition, you can see Kerberos errors in the packet lists that can lead you in the appropriate direction for resolution.

As shown in Figure 9, the SSSD LDAP client is using an SPN that doesn't exist when trying to bind to the LDAP server using Kerberos. In the trace, you see these packets.

**Note:** You get a `KRB5KDC_ERR_S_PRINCIPAL_UNKNOWN` error, which tells us the client could not find the requested SPN.

**Figure 9) Kerberos packet capture – packet list.**

294	12.438849	10.193.67.225	10.193.67.236	KRB5	1657	TGS-REQ
300	12.439183	10.193.67.225	10.193.67.236	KRB5	1671	TGS-REQ
302	12.449098	10.193.67.236	10.193.67.225	KRB5	377	KRB Error: KRB5KDC_ERR_S_PRINCIPAL_UNKNOWN
318	12.450702	10.193.67.225	10.193.67.236	KRB5	1657	TGS-REQ
320	12.458210	10.193.67.236	10.193.67.225	KRB5	377	KRB Error: KRB5KDC_ERR_S_PRINCIPAL_UNKNOWN
344	12.461738	10.193.67.236	10.193.67.225	KRB5	377	KRB Error: KRB5KDC_ERR_S_PRINCIPAL_UNKNOWN

The TGS-REQ packet shown in Figure 10 show you the SPN being requested, the ticket lifetime, the encryption type, and much more.

**Figure 10) TGS-REQ details – packet trace.**

▼	sname
	name-type: kRB5-NT-SRV-HST (3)
▼	sname-string: 2 items
	SNameString: ldap
	SNameString: ntap.local
	till: 2020-10-30 15:12:30 (UTC)
	nonce: 1604067150
▼	etype: 2 items
	ENCTYPE: eTYPE-AES256-CTS-HMAC-SHA1-96 (18)
	ENCTYPE: eTYPE-AES128-CTS-HMAC-SHA1-96 (17)

Having that information allows you to look at the SPN as the possible issue, as well as investigating the client configuration. In the above case, the issue was with how SSSD was configured to bind to the LDAP server. For information about the types of Kerberos packets you might encounter, see “Appendix A: Kerberos encryption types.”

## What information to collect before you contact NetApp Support

If you encounter NFS Kerberos issues and cannot resolve them on your own, [NetApp Support](#) is available to assist. When you open a support case, the technical support engineer must gather some data to troubleshoot the issue. To expedite that process, see the following list of questions that you can answer and information that you can provide to help resolve support cases faster. This list is not exhaustive—you might be asked for more data from the technical support engineer—but it is a start:

- What date and time did the problem occur?
- What KDC server type and OS are you using?
- What Kerberos clients are you using?
- Which user or group is affected?
- Is the problem still occurring? Is it intermittent?
- Can the KDC and/or DNS administrator be present for the call in case the technical support engineer needs extra information from the servers?
- Does the issue occur on all nodes? Some nodes? On specific IP addresses?
- Can the KDC and DNS server be reached through the network from ONTAP?
- Generate a new AutoSupport report with `-type all` (`autosupport invoke * -type all`). This command gathers information about the Kerberos configuration, DNS, network, event logs, and so on.
- Packet traces during an issue from the client, KDC, and ONTAP system.

For information about collecting packet traces in ONTAP, see the following NetApp Knowledge Base articles:

- [How to Capture Packet Traces \(tcpdump\) on ONTAP 9.2+ Systems](#)

- [How to Collect Rolling Packet Traces Using pktt on ONTAP 9.1 and Below Systems](#)

## Detailed configuration steps

To declutter the earlier sections of this report and to provide a cleaner, easier-to-read document, we have presented only the main configuration steps so far. In this section, we present some of the more intricate configuration steps.

### Rename NFS Kerberos machine accounts in Active Directory

In some cases, the NFS-FQDN-FORMAT of the machine account name as detailed in the section “Enable Kerberos on the data LIFs” is not a preferred name for the Active Directory environment. For instance, some organizations require strict naming schemes for machine accounts. In ONTAP 9.5 and later, you can specify the machine account name during Kerberos configuration with the `kerberos interface enable` command option `-machine-account`. If you did not, or could not, specify a name for a machine account during its initial creation, you can rename it afterward without having to remount clients, reissue tickets, and so on.

You can easily rename it after creation because the display name of the machine account is not critical to the Kerberos operations. What matters in the Kerberos interaction between clients and KDCs are:

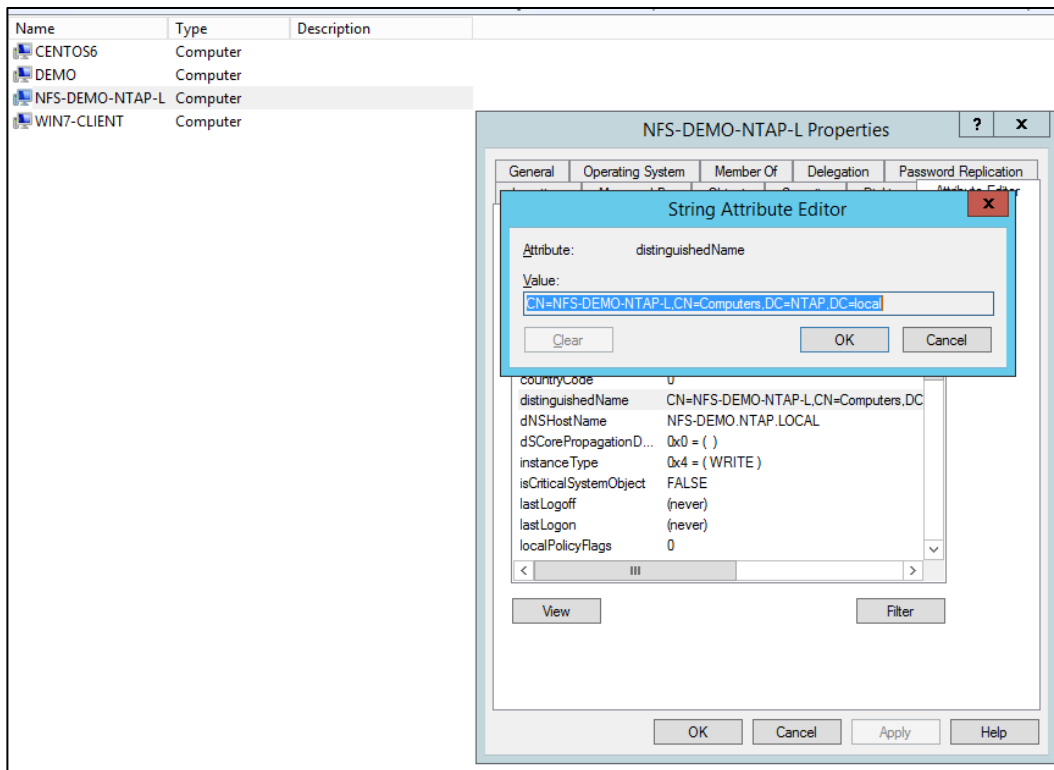
- SPNs on the machine account
- DNS host names
- Keytab files
- `sAMAccountName` on the machine account

With Active Directory, simply changing the display name (by highlighting and changing it in GUI) does not affect any of the preceding items. In some cases, Active Directory does not allow name changes through the GUI by default. Instead, you must use PowerShell. The following section guides you through machine account renaming.

To rename a machine account in Active Directory, complete the following steps:

1. First, locate the machine account of the object that you want to rename in Active Directory. Open the object in AD Users and Computers and find the DN value (you must [enable Advanced Features](#) for this step). You need this value for your PowerShell command.



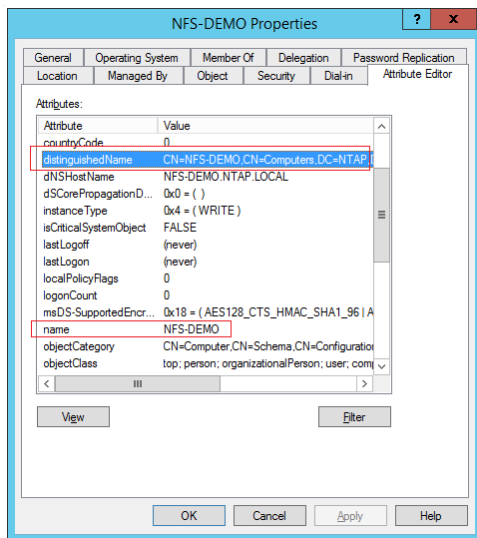


2. Open PowerShell as the domain administrator (or as another user with Active Directory renaming rights) and run the following command, replacing the objects in brackets with your desired values.

```
PS C:\> Rename-ADObject -Identity ["CN=NAME,CN=Computers,DC=DOMAIN,DC=local"] -NewName [NEW-NAME]
```

3. This command changes the DN and the Name value on the computer object, as well as the displayed name in AD Users and Computers.

	CENTOS6	Computer
	DEMO	Computer
	NFS-DEMO	Computer
	WIN7-CLIENT	Computer



4. Change the attributes for `dNSHostName` and add a new SPN with the machine account name's FQDN and short name. Use the PowerShell command [Set-ADComputer](#) for this step.

```
PS C:\> Set-ADComputer KERBEROS -DNSHostName demo.ntap.local -ServicePrincipalNames
@{Replace="nfs/KERBEROS", "HOST/KERBEROS", "HOST/nfs-demo-ntap-l.ntap.local", "nfs/nfs-demo-ntap-
l.ntap.local", "nfs/demo.ntap.local"}
```

5. Test your Kerberos access. Everything should still work fine, because the NFS SPN that is used by the data LIFs has not changed.

```
[root@centos6 ~]# mount home
[root@centos6 ~]# mount | grep home
demo:/home on /home type nfs (rw,hard,intr,sec=krb5,vers=4,addr=x.x.x.b,clientaddr=x.x.x.w)
[root@centos6 ~]# su student2
sh-4.1$ klist
klist: No credentials cache found (ticket cache FILE:/tmp/krb5cc_1302)
sh-4.1$ kinit
Password for student2@NTAP.LOCAL:
sh-4.1$ klist -e
Ticket cache: FILE:/tmp/krb5cc_1302
Default principal: student2@NTAP.LOCAL

Valid starting Expires Service principal
02/09/17 10:06:31 02/09/17 20:08:24 krbtgt/NTAP.LOCAL@NTAP.LOCAL
renew until 02/10/17 10:06:31, Etype(skey, tkt): aes256-cts-hmac-sha1-96, aes256-cts-hmac-sha1-96
sh-4.1$ cd ~
sh-4.1$ pwd
/home/student2
sh-4.1$ klist -e
Ticket cache: FILE:/tmp/krb5cc_1302
Default principal: student2@NTAP.LOCAL

Valid starting Expires Service principal
02/09/17 10:06:31 02/09/17 20:08:24 krbtgt/NTAP.LOCAL@NTAP.LOCAL
renew until 02/10/17 10:06:31, Etype(skey, tkt): aes256-cts-hmac-sha1-96, aes256-cts-hmac-sha1-96
02/09/17 10:08:35 02/09/17 20:08:24 nfs/demo.ntap.local@NTAP.LOCAL
renew until 02/10/17 10:06:31, Etype(skey, tkt): aes256-cts-hmac-sha1-96, aes256-cts-hmac-sha1-96
```

## Configure an NFS client to use Kerberos with net ads join

This section shows an example of how to configure NFS clients to use Kerberos after they join a domain by using `net ads join`. You can find net ads commands with the Samba and Winbind packages.

The NFS client that was used in this example is RHEL/CentOS 7.2. The `net ads` command is used to join the domain. The domain is Windows Server 2012 R2 Active Directory. Local UNIX users are used for name mappings.

To configure an NFS client to use Kerberos with `net ads join`, complete the following steps:

1. Install the necessary packages.

```
# yum install -y samba samba-winbind samba-winbind-clients ntp authconfig-gtk*
```

2. Check the time on the client and domain to ensure that you are within five minutes. This step also verifies that the client can find the domain controller.

```
# net time -S CORE-TME.NETAPP.COM
Mon Jul 11 16:08:00 2016

# date
Mon Jul 11 16:08:46 EDT 2016
```

3. Set up the NTP. If necessary, sync the time manually.

```
# net time set -S CORE-TME.NETAPP.COM
```

4. Make sure that the client is in the same DNS that Active Directory uses and that `nslookup` works for the client and for the domain controllers.

```
# nslookup centos7
Server:      x.x.x.c
Address:     x.x.x.c#53

Name:   centos7.core-tme.netapp.com
Address: x.x.x.x
Name:   centos7.core-tme.netapp.com
Address: 192.168.122.1

# nslookup core-tme.netapp.com
Server:      x.x.x.c
Address:     x.x.x.c#53

Name:   core-tme.netapp.com
Address: x.x.x.d
Name:   core-tme.netapp.com
Address: x.x.x.c
```

5. Modify the `/etc/krb5.conf` file to reflect the Active Directory domain.

```
[logging]
default = FILE:/var/log/krb5libs.log
kdc = FILE:/var/log/krb5kdc.log
admin_server = FILE:/var/log/kadmind.log

[libdefaults]
dns_lookup_realm = true
dns_lookup_kdc = true
ticket_lifetime = 24h
renew_lifetime = 7d
forwardable = true
rdns = false
default_realm = CORE-TME.NETAPP.COM
default_ccache_name = KEYRING:persistent:%{uid}

[realms]
# EXAMPLE.COM = {
#   kdc = kerberos.example.com
#   admin_server = kerberos.example.com
# }
CORE-TME.NETAPP.COM = {
    kdc = dc1.core-tme.netapp.com:88
    admin_server = dc1.core-tme.netapp.com:749
    default_domain = core-tme.netapp.com
}
```

```

}

[domain_realm]
# .example.com = EXAMPLE.COM
# example.com = EXAMPLE.COM
.core-tme.netapp.com = CORE-TME.NETAPP.COM
core-tme.netapp.com = CORE-TME.NETAPP.COM

```

## 6. Configure /etc/samba/smb.conf with the domain information.

```

[global]

    workgroup = CORE-TME
    password server = stme-infra02.core-tme.netapp.com:88
    realm = CORE-TME.NETAPP.COM
    security = ads
    idmap uid = 16777216-33554431
    idmap gid = 16777216-33554431
    template shell = /bin/bash
    winbind use default domain = false
    winbind offline logon = true

    log file = /var/log/samba/log.%m
    max log size = 50

    passdb backend = tdbsam

    load printers = yes
    cups options = raw

[homes]
    comment = Home Directories
    browseable = no
    writable = yes

[printers]
    comment = All Printers
    path = /var/spool/samba
    browseable = no
    guest ok = no
    writable = no
    printable = yes

```

## 7. Restart the SMB and rpcgssd services.

```

# service smb restart
# service rpcgssd restart

```

## 8. Get a Kerberos ticket for the administrator.

```

# kinit administrator
Password for administrator@CORE-TME.NETAPP.COM:
# klist
Ticket cache: KEYRING:persistent:0:0
Default principal: administrator@CORE-TME.NETAPP.COM

Valid starting          Expires              Service principal
07/12/2016 11:28:54    07/12/2016 21:28:54    krbtgt/CORE-TME.NETAPP.COM@CORE-TME.NETAPP.COM
        renew until 07/19/2016 11:28:49

```

## 9. Join the domain.

```

# net ads join -U administrator
Enter administrator's password:
Using short domain name -- CORE-TME
Joined 'CENTOS7' to dns domain 'core-tme.netapp.com'

```

**Note:** All normal Windows domain rules apply: The time skew is within five minutes, the user account has permissions to add computer objects to a domain, and the DNS can locate domain controllers.

## 10. Create a keytab file.

```
# net ads keytab create -U administrator
```

Warning: "kerberos method" must be set to a keytab method to use keytab functions.

Enter administrator's password:

## 11. Verify the keytab file.

When a machine account is added to Active Directory by using `net ads keytab`, the following SPNs are added to the `krb5.keytab` file automatically:

```
# ktutil
ktutil: rkt /etc/krb5.keytab
ktutil: list
slot KVNO Principal
-----
 1  3 host/centos7.core-tme.netapp.com@CORE-TME.NETAPP.COM
 2  3 host/centos7.core-tme.netapp.com@CORE-TME.NETAPP.COM
 3  3 host/centos7.core-tme.netapp.com@CORE-TME.NETAPP.COM
 4  3 host/centos7.core-tme.netapp.com@CORE-TME.NETAPP.COM
 5  3 host/centos7.core-tme.netapp.com@CORE-TME.NETAPP.COM
 6  3 host/centos7@CORE-TME.NETAPP.COM
 7  3 host/centos7@CORE-TME.NETAPP.COM
 8  3 host/centos7@CORE-TME.NETAPP.COM
 9  3 host/centos7@CORE-TME.NETAPP.COM
10  3 host/centos7@CORE-TME.NETAPP.COM
11  3 root/centos7.core-tme.netapp.com@CORE-TME.NETAPP.COM
12  3 root/centos7.core-tme.netapp.com@CORE-TME.NETAPP.COM
13  3 root/centos7.core-tme.netapp.com@CORE-TME.NETAPP.COM
14  3 root/centos7.core-tme.netapp.com@CORE-TME.NETAPP.COM
15  3 root/centos7.core-tme.netapp.com@CORE-TME.NETAPP.COM
16  3 root/centos7@CORE-TME.NETAPP.COM
17  3 root/centos7@CORE-TME.NETAPP.COM
18  3 root/centos7@CORE-TME.NETAPP.COM
19  3 root/centos7@CORE-TME.NETAPP.COM
20  3 root/centos7@CORE-TME.NETAPP.COM
21  3 CENTOS7$@CORE-TME.NETAPP.COM
22  3 CENTOS7$@CORE-TME.NETAPP.COM
23  3 CENTOS7$@CORE-TME.NETAPP.COM
24  3 CENTOS7$@CORE-TME.NETAPP.COM
25  3 CENTOS7$@CORE-TME.NETAPP.COM
```

No other SPNs should be required for the machine account. Notably, there are SPNs for `root/` in the keytab. Because there is a UNIX user named `root` in the SVM by default, you do not have to consider name mapping for the client unless you want a different mapping.

If you need a different mapping, a [KRB to UNIX name mapping must exist](#) for `machine$` either locally on the SVM (a name-mapping rule or a UNIX user) or on the Active Directory object (in the form of a `uidNumber/GidNumber` attribute in LDAP).

The easiest way to test a resolution for this issue is through the local `unix-user`:

```
::> unix-user create -vserver parisi -user CENTOS7$ -id 10001 -primary-gid 1
::> unix-user show -vserver parisi -user CENTOS7$
  Vserver: parisi
  User Name: CENTOS7$
  User ID: 10001
Primary Group ID: 1
User's Full Name:
```

**Note:** After testing the resolution with a local UNIX user, a long-term solution is to create the name mapping rule, as in the section “Create a UNIX user or a name-mapping rule to map the NFS client principal.”

## 12. Test the `krb-unix` mapping for the root SPN or for the machine account SPN if you prefer.

```
::> set diag
::> diag secd name-mapping show -node node03 -vserver parisi -direction krb-unix -name
CENTOS7$@CORE-TME.NETAPP.COM
```

```
CENTOS7$@CORE-TME.NETAPP.COM maps to CENTOS7$
```

```
::*> diag secd name-mapping show -node node03 -vserver parisi -direction krb-unix -name  
root/centos7.core-tme.netapp.com@CORE-TME.NETAPP.COM  
root/centos7.core-tme.netapp.com@CORE-TME.NETAPP.COM maps to root
```

**13. Verify that the following services are running and are enabled on boot:**

```
systemctl start ntpd  
systemctl enable ntpd  
systemctl start smb  
systemctl enable smb  
systemctl start winbind  
systemctl enable winbind  
systemctl start sssd  
systemctl enable sssd
```

**14. Test the domain connectivity.**

```
# net ads info  
LDAP server: x.x.x.c  
LDAP server name: stme-infra02.core-tme.netapp.com  
Realm: CORE-TME.NETAPP.COM  
Bind Path: dc=CORE-TME,dc=NETAPP,dc=COM  
LDAP port: 389  
Server time: Tue, 12 Jul 2016 11:33:29 EDT  
KDC server: x.x.x.c  
Server time offset: 0  
  
# wbinfo -t  
checking the trust secret for domain CORE-TME via RPC calls succeeded
```

**15. Verify that the NFS unix-user or equivalent name mapping is in place so that the service account (nfs/fqdn@REALM) can authenticate.**

```
::*> unix-user create -vserver parisi -user nfs -id 10002 -primary-gid 1  
::~*> unix-user show -vserver parisi -user nfs  
      Vserver: parisi  
      User Name: nfs  
      User ID: 10002  
Primary Group ID: 1  
User's Full Name:
```

**16. On the NFS client, make sure that SSSD (or the LDAP client equivalent) is configured. For details, see [TR-4835](#). Or you can use a local UNIX user in /etc/passwd and on the SVM.**

To test LDAP, run the following command:

```
# id ldapuser  
  
# getent passwd ldapuser
```

**17. Try to mount the SVM data interfaces with Kerberos. The SVM must already have the following created and configured:**

- Kerberos realm
- Kerberos interfaces
- DNS A/AAAA records in the DNS server (forward and reverse)
- Permitted encetypes for Kerberos
- Export policy rules on the NFS exports and parent directories that allow Kerberos

See the following mount example:

```
[root@centos7 /]# mount -o sec=krb5 parisi-nfs:/nfs /kerberos  
[root@centos7 /]#
```

**18. su as a different user and kinit. cd into the mount and check for your NFS service ticket:**

```
[root@centos7 /]# su test@CORE-TME.NETAPP.COM
```

```
[test@core-tme.netapp.com@centos7 /]$ kinit test@CORE-TME.NETAPP.COM
Password for test@CORE-TME.NETAPP.COM:
[test@core-tme.netapp.com@centos7 /]$ klist
Ticket cache: KEYRING:persistent:106003697:106003697
Default principal: test@CORE-TME.NETAPP.COM

Valid starting          Expires                Service principal
06/29/2016 16:38:26    06/30/2016 02:38:26    krbtgt/CORE-TME.NETAPP.COM@CORE-TME.NETAPP.COM
        renew until 07/06/2016 16:38:21

[test@core-tme.netapp.com@centos7 /]$ mount | grep kerberos
parisi-nfs:/nfs on /kerberos type nfs4
(rw,relatime,vers=4.0,rsize=65536,wsiz=65536,namlen=255,hard,proto=tcp,port=0,timeo=600,retrans=
2,sec=krb5,clientaddr=x.x.x.x,local_lock=none,addr=x.x.x.b)

[test@core-tme.netapp.com@centos7 /]$ cd /kerberos

[test@core-tme.netapp.com@centos7 /kerberos]$ klist -e
Ticket cache: KEYRING:persistent:106003697:106003697
Default principal: test@CORE-TME.NETAPP.COM

Valid starting          Expires                Service principal
06/29/2016 16:39:43    06/30/2016 02:38:26    nfs/parisi-nfs.core-tme.netapp.com@CORE-TME.NETAPP.COM
        renew until 07/06/2016 16:38:21, Etype (skey, tkt): aes256-cts-hmac-shal-96, aes256-cts-
hmac-shal-96
06/29/2016 16:38:26    06/30/2016 02:38:26    krbtgt/CORE-TME.NETAPP.COM@CORE-TME.NETAPP.COM
        renew until 07/06/2016 16:38:21, Etype (skey, tkt): aes256-cts-hmac-shal-96, aes256-cts-
hmac-shal-96
```

## Configure an NFS client to use Kerberos with realm join

This section shows an example of how to configure NFS clients to use Kerberos after they join a domain. The NFS client that we use in this example is RHEL/CentOS 7.2. We use the [realm](#) command to join the domain. You can find the packages that you need to perform these steps in the official Red Hat documentation for [Discovering and Joining Identity Domains](#). The domain is Windows Server 2012 R2 Active Directory. We use local UNIX users for name mappings.

To configure an NFS client to use Kerberos with `realm join`, complete the following steps.

1. Install the necessary packages.

```
yum -y install realmd sssd oddjob oddjob-mkhomedir adcli samba-common krb5-workstation ntp
```

2. Make sure that the DNS on the NFS client is configured to the Active Directory domain and that an A/AAAA record exists in the DNS for the Linux client. Test DNS lookups.

```
[root@centos7 /]# cat /etc/resolv.conf
# Generated by NetworkManager
search core-tme.netapp.com
nameserver x.x.x.c

[root@centos7 /]# nslookup centos7
Server:          x.x.x.c
Address:         x.x.x.c#53

Name:   centos7.core-tme.netapp.com
Address: x.x.x.x
```

3. Verify that [all firewall rules](#) allow Active Directory connectivity, LDAP, Kerberos, and so on.
4. Discover the Active Directory realm.

```
# realm discover core-tme.netapp.com
core-tme.netapp.com
type: kerberos
realm-name: CORE-TME.NETAPP.COM
domain-name: core-tme.netapp.com
configured: no
server-software: active-directory
```

```
client-software: sssd
required-package: oddjob
required-package: oddjob-mkhomedir
required-package: sssd
required-package: adcli
required-package: samba-common
```

## 5. Join the domain.

```
[root@centos7 ~]# realm join CORE-TME.NETAPP.COM
Password for Administrator:
```

**Note:** All normal Windows domain rules apply. The time skew is within 5 minutes, the user account has permissions to add computer objects to a domain, and the DNS can locate domain controllers. `realm join` automatically configures SSSD to a base level and configures the Kerberos keytab files.

## 6. Check connectivity to the domain by performing a name lookup (this action uses SSSD for LDAP connectivity):

```
[root@centos7 ~]# id CORE-TME\\test
uid=106003697(test@core-tme.netapp.com) gid=106000513(domain users@core-tme.netapp.com)
groups=106000513(domain users@core-tme.netapp.com)
```

**Note:** The preceding user created a UID and GID numeric based on an algorithm in SSSD by default to approximate a user and group ID based on the SID. If you want classic UNIX user attributes, be sure to configure SSSD as described in [TR-4835](#).

## 7. Run `kinit` to test Kerberos for a user.

```
[root@centos7 ~]# kinit test@CORE-TME.NETAPP.COM
Password for test@CORE-TME.NETAPP.COM:

[root@centos7 ~]# klist
Ticket cache: KEYRING:persistent:0:0
Default principal: test@CORE-TME.NETAPP.COM

Valid starting      Expires            Service principal
06/29/2016 15:23:54  06/30/2016 01:23:54  krbtgt/CORE-TME.NETAPP.COM@CORE-TME.NETAPP.COM
        renew until 07/06/2016 15:23:50
```

**Note:** As an option, you can configure `/etc/krb5.conf` with the realm information to avoid needing to append the realm to `kinit` requests.

See the following example:

```
[root@centos7 /]# cat /etc/krb5.conf
[logging]
default = FILE:/var/log/krb5libs.log
kdc = FILE:/var/log/krb5kdc.log
admin_server = FILE:/var/log/kadmind.log

[libdefaults]
dns_lookup_realm = true
dns_lookup_kdc = true
default_tkt_enctypes = aes256-cts-hmac-sha1-96
default_tgs_enctypes = aes256-cts-hmac-sha1-96
ticket_lifetime = 24h
renew_lifetime = 7d
forwardable = true
rdns = false
default_realm = CORE-TME.NETAPP.COM
default_ccache_name = KEYRING:persistent:%{uid}

[realms]
# EXAMPLE.COM = {
#   kdc = kerberos.example.com
#   admin_server = kerberos.example.com
# }
CORE-TME.NETAPP.COM = {
```



```

    kdc = dc1.core-tme.netapp.com:88
    admin_server = dc1.core-tme.netapp.com:749
    default_domain = core-tme.netapp.com
}

[domain_realm]
# .example.com = EXAMPLE.COM
# example.com = EXAMPLE.COM
.core-tme.netapp.com = CORE-TME.NETAPP.COM
core-tme.netapp.com = CORE-TME.NETAPP.COM

[root@centos7 ~]# kinit test
Password for test@CORE-TME.NETAPP.COM:

```

Make sure the [krb5.conf file is configured to allow only specific enctypees](#) or that the [machine account in the domain for the NFS client](#) allows only the desired enctypees. Be sure to disallow RC4-HMAC because NetApp ONTAP does not support it.

See the following example of failure when using RC4-HMAC:

```

6/29/2016 16:09:56 node03
WARNING      sec2.nfsAuth.problem: vserver (parisi) General NFS authorization problem. Error:
RPC accept GSS token procedure failed
[ 0 ms] Using the NFS service credential for logical interface 1035 (SPN='nfs/parisi-nfs.core-
tme.netapp.com@CORE-TME.NETAPP.COM') from cache.
**[ 1] FAILURE: Failed to accept the context: Unspecified GSS failure. Minor code may
provide more information (minor: Encryption type ArcFour with HMAC/md5 not permitted).

```

8. When a machine account is added to Active Directory by using `realm join`, the following SPNs are added to the `krb5.keytab` file automatically:

```

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

[root@centos7 /]# klist -kte
Keytab name: FILE:/etc/krb5.keytab
KVNO Timestamp Principal
-----
2 06/29/2016 15:16:49 host/centos7.core-tme.netapp.com@CORE-TME.NETAPP.COM (des-cbc-crc)
2 06/29/2016 15:16:49 host/centos7.core-tme.netapp.com@CORE-TME.NETAPP.COM (des-cbc-md5)
2 06/29/2016 15:16:50 host/centos7.core-tme.netapp.com@CORE-TME.NETAPP.COM (aes128-cts-hmac-
sha1-96)
2 06/29/2016 15:16:50 host/centos7.core-tme.netapp.com@CORE-TME.NETAPP.COM (aes256-cts-hmac-
sha1-96)
2 06/29/2016 15:16:50 host/centos7.core-tme.netapp.com@CORE-TME.NETAPP.COM (arcfour-hmac)
2 06/29/2016 15:16:50 host/centos7@CORE-TME.NETAPP.COM (des-cbc-crc)
2 06/29/2016 15:16:50 host/centos7@CORE-TME.NETAPP.COM (des-cbc-md5)
2 06/29/2016 15:16:50 host/centos7@CORE-TME.NETAPP.COM (aes128-cts-hmac-sha1-96)
2 06/29/2016 15:16:51 host/centos7@CORE-TME.NETAPP.COM (aes256-cts-hmac-sha1-96)
2 06/29/2016 15:16:51 host/centos7@CORE-TME.NETAPP.COM (arcfour-hmac)
2 06/29/2016 15:16:51 CENTOS7$@CORE-TME.NETAPP.COM (des-cbc-crc)
2 06/29/2016 15:16:51 CENTOS7$@CORE-TME.NETAPP.COM (des-cbc-md5)

```

```

2 06/29/2016 15:16:51 CENTOS7$@CORE-TME.NETAPP.COM (aes128-cts-hmac-shal-96)
2 06/29/2016 15:16:52 CENTOS7$@CORE-TME.NETAPP.COM (aes256-cts-hmac-shal-96)
2 06/29/2016 15:16:52 CENTOS7$@CORE-TME.NETAPP.COM (arcfour-hmac)

```

No other SPNs should be required for the machine account. The client attempts to get a ticket by using the machine account principal (machine\$@REALM.COM).

Therefore, a [KRB to UNIX name mapping must exist](#) for machine\$ either locally on the SVM (a name-mapping rule or a UNIX user) or on the Active Directory object (in the form of a uidNumber/GidNumber attribute in LDAP). Otherwise, the mount request fails with the following error:

```

6/29/2016 16:28:52 node03
WARNING      secd.nfsAuth.problem: vserver (parisi) General NFS authorization problem. Error:
RPC accept GSS token procedure failed
[ 0 ms] Using the NFS service credential for logical interface 1035 (SPN='nfs/parisi-nfs.core-tme.netapp.com@CORE-TME.NETAPP.COM') from cache.
[ 1] GSS_S_COMPLETE: client = 'CENTOS7$@CORE-TME.NETAPP.COM'
[ 2] Extracted KG_USAGE_ACCEPTOR_SIGN Derived Key
[ 2] Extracted KG_USAGE_INITIATOR_SIGN Derived Key
[ 2] Exported lucid context
[ 5] Trying to map SPN 'CENTOS7$@CORE-TME.NETAPP.COM' to UNIX user 'CENTOS7$' using implicit mapping
[ 6] Entry for user-name: CENTOS7$ not found in the current source: FILES. Ignoring and trying next available source
[ 7] Failed to initiate Kerberos authentication. Trying NTLM.
[ 11] Successfully connected to x.x.x.c:389 using TCP
**[ 91] FAILURE: User 'CENTOS7$' not found in UNIX authorization source LDAP.
[ 91] Entry for user-name: CENTOS7$ not found in the current source: LDAP. Entry for user-name: CENTOS7$ not found in any of the available sources
[ 91] Unable to map SPN 'CENTOS7$@CORE-TME.NETAPP.COM'
[ 91] Unable to map Kerberos NFS user 'CENTOS7$@CORE-TME.NETAPP.COM' to appropriate UNIX user
[ 91] Failed to accept the context: The routine completed successfully (minor: Unknown error). Result = 6916

```

The easiest way to resolve this issue is with the local unix-user:

```

::> unix-user create -vserver parisi -user CENTOS7$ -id 10001 -primary-gid 1
::> unix-user show -vserver parisi -user CENTOS7$
      Vserver: parisi
      User Name: CENTOS7$
      User ID: 10001
Primary Group ID: 1
User's Full Name:

```

## 9. Test the krb-unix mapping.

```

::> set diag
::> diag secd name-mapping show -node node03 -vserver parisi -direction krb-unix -name CENTOS7$@CORE-TME.NETAPP.COM
CENTOS7$@CORE-TME.NETAPP.COM maps to CENTOS7$

```

## 10. Make sure that the NFS unix-user or equivalent name mapping is in place so that the service account (nfs/fqdn@REALM) can authenticate.

```

::> unix-user create -vserver parisi -user nfs -id 10002 -primary-gid 1
::> unix-user show -vserver parisi -user nfs
      Vserver: parisi
      User Name: nfs
      User ID: 10002
Primary Group ID: 1
User's Full Name:

```

## 11. Try to mount the SVM data interfaces with Kerberos. The SVM must already have the following created and configured:

- Kerberos realm
- Kerberos interfaces

- DNS A/AAAA records in the DNS server (forward and reverse)
- Permitted encyptes for Kerberos
- Export policy rules on the NFS exports and parent directories that allow Kerberos

See the following mount example:

```
[root@centos7 ~]# mount -o sec=krb5 parisi-nfs:/nfs /kerberos
[root@centos7 ~]#
```

12. su as a different user and kinit. cd into the mount and check for your NFS service ticket.

```
[root@centos7 ~]# su test@CORE-TME.NETAPP.COM
[test@core-tme.netapp.com~]# kinit test@CORE-TME.NETAPP.COM
Password for test@CORE-TME.NETAPP.COM:
[test@core-tme.netapp.com~]# klist
Ticket cache: KEYRING:persistent:106003697:106003697
Default principal: test@CORE-TME.NETAPP.COM

Valid starting      Expires            Service principal
06/29/2016 16:38:26 06/30/2016 02:38:26 krbtgt/CORE-TME.NETAPP.COM@CORE-TME.NETAPP.COM
        renew until 07/06/2016 16:38:21

[test@core-tme.netapp.com~]# mount | grep kerberos
parisi-nfs:/nfs on /kerberos type nfs4
(rw,relatime,vers=4.0,rsize=65536,wsiz=65536,namlen=255,hard,proto=tcp,port=0,timeo=600,retrans=
2,sec=krb5,clientaddr=x.x.x.x,local_lock=none,addr=x.x.x.b)

[test@core-tme.netapp.com~]# cd /kerberos

[test@core-tme.netapp.com~]# klist -e
Ticket cache: KEYRING:persistent:106003697:106003697
Default principal: test@CORE-TME.NETAPP.COM

Valid starting      Expires            Service principal
06/29/2016 16:39:43 06/30/2016 02:38:26 nfs/parisi-nfs.core-tme.netapp.com@CORE-TME.NETAPP.COM
        renew until 07/06/2016 16:38:21, Etype (skey, tkt): aes256-cts-hmac-sha1-96, aes256-cts-
hmac-sha1-96
06/29/2016 16:38:26 06/30/2016 02:38:26 krbtgt/CORE-TME.NETAPP.COM@CORE-TME.NETAPP.COM
        renew until 07/06/2016 16:38:21, Etype (skey, tkt): aes256-cts-hmac-sha1-96, aes256-cts-
hmac-sha1-96
```

## Appendix A: Kerberos encryption types

Kerberos v5 supports multiple encyptes. The type used in a given instance is automatically negotiated between the client and the Kerberos KDC servers. The negotiation is based on client and server settings as well as encryption types used to encrypt the password for the user and service principals.

Table 9 defines the different encyptes used by Kerberos v5.

**Table 9) Kerberos encryption types.**

Entype	Cipher algorithm	Cipher mode	Key length	HMAC	Strength
aes256-cts aes256-cts-hmac-sha1-96	AES	CBC+CTS	256 bits	SHA-1 96 bits	Strongest
aes128-cts aes128-cts-hmac-sha1-96	AES	CBC+CTS	128 bits	SHA-1 96 bits	Strong
rc4-hmac	RC4		128 bits	SHA-1 96 bits	Strong
des3-cbc-sha1	3DES	CBC	168 bits	SHA-1 96 bits	Strong
des-cbc-crc	DES	CBC	56 bits	CRC 32 bits	Weak

Enctype	Cipher algorithm	Cipher mode	Key length	HMAC	Strength
des-cbc-md5	DES	CBC	56 bits	MD5 96 bits	Weak, but strongest single DES

## Appendix B: Machine account attributes

The attribute [msDS-SupportedEncryptionTypes](#) is used to specify which encryption types are used in Kerberos authentication. The values are specified by adding a series of values together.

The `msDS-SupportedEncryptionTypes` value is set to 27 (hex 0x19). That value translates to allowing only DES and AES encryption types. RC4 is omitted because ONTAP does not support RC4-HMAC for NFS Kerberos. Table 10 shows which values are valid. The value 27 is derived by adding the specified decimal values together for DES-CBC-CRC + DES-CBC-MD5 + AES128 + AES256 (1+2+8+16).

**Table 10) Valid `msDS-SupportedEncryptionTypes` attribute values.**

Property flag	Value in hexadecimal	Value in decimal
DES-CBC-CRC	0x01	1
DES-CBC-MD5	0x02	2
RC4-HMAC	0x04	4
AES128-CTS-HMAC-SHA1-96	0x08	8
AES256-CTS-HMAC-SHA1-96	0x10	16

## Appendix C: Kerberos packet types, errors, and terminology

Table 11, Table 12, and Table 13 show the type of Kerberos requests that take place over the wire, as well as which error codes can be returned during requests. This information is intended to help troubleshooting by explaining what each request does.

**Table 11) Kerberos packets.**

Kerberos packet	What it does
AS-REQ	Authentication Service request: looks up the user name and password to get the TGT; also requests the session key.
AS-REP	Authentication Service reply: delivers the TGT and session key.
AP-REQ	Application server request: Certifies to a server that the sender has recent knowledge of the encryption key in the accompanying ticket to help the server detect replays. The request also assists in the selection of a "true session key" to use with the particular session.
AP-REP	Application server reply: Includes the session key and sequence number.
TGS-REQ	Ticket-granting-server request: Uses the TGT to get the Service Ticket (ST).
TGS-REP	Ticket-granting-server reply: Delivers the ST.

**Table 12) Kerberos errors from [network captures](#).**

Kerberos error	What it means
KDC_ERR_S_PRINCIPAL_UNKNOWN	The SPN does not exist or there was a duplicate SPN on the KDC. Note the "S" in the error—this stands for "SPN" or "service."

Kerberos error	What it means
KDC_ERR_C_PRINCIPAL_UNKNOWN	The UPN does not exist or there was a duplicate UPN on the KDC. Note the “C” in the error—this stands for “client” and refers to the user principal rather than the service principal.
KDC_ERR_ETYPE_NOTSUPP	Encryption type requested by the client is not supported by the KDC. This is common with DES and Windows 2008 R2.
KDC_ERR_PREAUTH_REQUIRED	This error means that the KDC wants a password for the account attempting authentication; this is a benign error.
KDC_ERR_PREAUTH_FAILED	The preauthentication failed, generally because the password was incorrect.
KRB_AP_ERR_SKEW	The time is outside the allowed skew window. This is typically 5 minutes.
KRB_AP_ERR_REPEAT	This is the security mechanism to prevent replay attacks. If server name, client name, time, and microsecond fields from the Authenticator match recently seen entries in the cache, this error occurs.
KRB_AP_ERR_MODIFIED	This error indicates that the service was unable to decrypt the ticket that it was given. A common cause is because the SPN is registered to the wrong account. Another possible cause is a duplicate SPN in two different domains in the forest. This error can also occur if the KDC where the original ticket was issued is offline, causing the client to need to reauthenticate to a new KDC.

**Table 13) Kerberos terminology from [CentOS.org](http://CentOS.org) and [IBM.com](http://IBM.com).**

Term	Definition
KDC	Key Distribution Center: A service that issues Kerberos tickets, usually run on the same host as the ticket-granting server (TGS).
TGT	Ticket Granting Ticket: A special ticket that allows the client to obtain additional tickets without applying for them from the KDC. Example: krbtgt/domain@REALM. The principal for this exists as a user account named krbtgt in Microsoft Windows Active Directory.
TGS	Ticket Granting Server: A server that issues tickets for a desired service that are in turn given to users for access to the service. The TGS usually runs on the same host as the KDC.
SPN	Service Principal Name: Kerberos principal associated with service in the format of service/instance@REALM. Example: ldap/server.netapp.com@NETAPP.COM.
UPN	User Principal Name: Kerberos principal associated with a user name in the format of user@REALM. Example: ldapuser@NETAPP.COM.
Session key	A temporary encryption key used between two principals, with the lifetime limited to the duration of a single login session.
ST	Service Ticket: A ticket that is issued for a specific service; for example, nfs/instance@REALM for NFS services or ldap/instance@REALM for LDAP services.
AS	Authentication Server: A server that issues tickets for a desired service that are in turn given to users for access to the service. The AS responds to requests from clients who do not have or do not send credentials with a request. This server is usually used to gain access to the ticket granting server (TGS) service by issuing a TGT. The AS usually runs on the same host as the KDC.

Term	Definition
Realm	A network that uses Kerberos, composed of one or more servers called KDCs and a potentially large number of clients.
GSS-API	The Generic Security Service Application Program Interface (defined in RFC-2743 published by the Internet Engineering Task Force): A set of functions that provide security services. This API is used by clients and services to authenticate to each other without either program having specific knowledge of the underlying mechanism. If a network service (such as cyrus-IMAP) uses GSS-API, it can authenticate using Kerberos.

## Disclaimer

NetApp provides no representations or warranties regarding the accuracy, reliability, or serviceability of any information or recommendations provided in this publication, or with respect to any results that can be obtained by the use of the information or observance of any recommendations provided herein. The information in this document is distributed AS IS, and the use of this information or the implementation of any recommendations or techniques herein is a customer's responsibility and depends on the customer's ability to evaluate and integrate them into the customer's operational environment. This document and the information contained herein can be used solely in connection with the NetApp products discussed in this document.

## Where to find additional information

To learn more about the information that is described in this document, review the following documents and/or websites:

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

## Contact us

To let us know how we can improve this technical report, contact us at [docfeedback@netapp.com](mailto:docfeedback@netapp.com). Include TECHNICAL REPORT 4616 in the subject line.

## Version history

Version	Date	Document version history
Version 1.0	August 2017	Initial commit
Version 1.1	June 2020	Minor revisions; ONTAP 9.7 information
Version 1.2	February 2021	Minor revisions; ONTAP 9.8 information
Version 1.2.1	June 2021	Minor revisions; ONTAP 9.9.1 information

Refer to the [Interoperability Matrix Tool \(IMT\)](#) on the NetApp Support site to validate that the exact product and feature versions described in this document are supported for your specific environment. The NetApp IMT defines the product components and versions that can be used to construct configurations that are supported by NetApp. Specific results depend on each customer's installation in accordance with published specifications.

### **Copyright information**

Copyright © 2020—2021 NetApp, Inc. All Rights Reserved. Printed in the U.S. No part of this document covered by copyright may be reproduced in any form or by any means—graphic, electronic, or mechanical, including photocopying, recording, taping, or storage in an electronic retrieval system—without prior written permission of the copyright owner.

Software derived from copyrighted NetApp material is subject to the following license and disclaimer:

THIS SOFTWARE IS PROVIDED BY NETAPP "AS IS" AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, WHICH ARE HEREBY DISCLAIMED. IN NO EVENT SHALL NETAPP BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

NetApp reserves the right to change any products described herein at any time, and without notice. NetApp assumes no responsibility or liability arising from the use of products described herein, except as expressly agreed to in writing by NetApp. The use or purchase of this product does not convey a license under any patent rights, trademark rights, or any other intellectual property rights of NetApp.

The product described in this manual may be protected by one or more U.S. patents, foreign patents, or pending applications.

Data contained herein pertains to a commercial item (as defined in FAR 2.101) and is proprietary to NetApp, Inc. The U.S. Government has a non-exclusive, non-transferrable, non-sublicensable, worldwide, limited irrevocable license to use the Data only in connection with and in support of the U.S. Government contract under which the Data was delivered. Except as provided herein, the Data may not be used, disclosed, reproduced, modified, performed, or displayed without the prior written approval of NetApp, Inc. United States Government license rights for the Department of Defense are limited to those rights identified in DFARS clause 252.227-7015(b).

### **Trademark information**

NETAPP, the NETAPP logo, and the marks listed at <http://www.netapp.com/TM> are trademarks of NetApp, Inc. Other company and product names may be trademarks of their respective owners.

TR-4616-0621