Technical White Paper

Dell EMC PowerScale: Integrating OneFS with Kerberos Environment for Protocols

A practical guide for Implementation

Abstract

This white paper covers basic Kerberos concepts and introduces Dell EMC™ PowerScale™ OneFS™ supported Kerberos types for protocols. This document also provides practical procedures to integrate Kerberos authentication into OneFS 8.0 and later for SMB and NFS.

June 2020

H17769

Revisions

Date	Description
April 2019	Initial release
June 2020	PowerScale rebranding

Acknowledgements

This paper was produced by the following members of the Dell EMC:

Author: Lieven Lin (lieven.lin@dell.com)

Dell EMC and the author of this document welcome your feedback along with any recommendations for improving this document.

Table of contents

Executive summary

For business security compliance, organizations usually require a more secure and centralized authentication infrastructure to fulfill their goals. In a network-attached storage (NAS) environment that uses all types of network storage protocols, Kerberos™ provides mutual authentication between servers and clients to prevent a man-in-the-middle attack. This document explains basic concepts of Kerberos components and how it works in theory. This documents also explores key procedures to set up a Kerberized environment including the Dell EMC™ OneFS™ operating system, SMB protocol, and NFS protocol.

Audience

This document is intended for Dell EMC™ PowerScale™ storage administrators who want to integrate OneFS with Kerberos for protocols. The document assumes readers have basic knowledge of the following:

- NAS systems
- Microsoft® Windows® operating system and SMB protocol
- Linux® operating system and NFS protocol
- OneFS distributed file system and multi-protocol access feature
- LDAP server and KDC server implementation

For more information on the topics discussed in this paper, Dell EMC recommends reviewing the following publications:

- Dell EMC PowerScale OneFS: A Technical Overview
- PowerScale OneFS Web Administration Guide
- PowerScale OneFS CLI Administration Guide
- Current PowerScale Software Releases

1 Kerberos authentication overview

1.1 Kerberos

Authentication is a set of actions to verify the validity of users. To prove a user's identities, most computer systems use a password by transferring plaintext or encrypted password directly through network. The password would be stolen by man-in-the-middle attack which will result in security vulnerabilities.

Kerberos is a ticket-based authentication protocol to prove identities in a very secure manner instead of transferring password over network. It allows mutual authentication between clients and servers over unsecure networks and prevents eavesdropping and replay attacks. With Kerberos, organizations can authenticate their users to use any service at any time. A ticket in Kerberos contains identity of a client, the desired server to access, timestamp, and other information. So that the client who holds the ticket can be authenticated to the server. Refer to appendix A for details about the Kerberos authentication.

1.2 Kerberos and OneFS

This document will focus on SMB and NFS integration with the Kerberos environment. For the HDFS protocol used in a Hadoop solution, refer to PowerScale OneFS with Hadoop Kerberos and Identity Management Approaches.

1.2.1 OneFS authentication provider

Most authentication providers in OneFS provide two functions:

- **Authentication (Are you who you say you are):** A user provides a name and a password to prove itself. This verifies that the user is really the user and not an imposter who is pretending to be the user.
- **Identity management (Who are you):** This function provides information about users by looking up the validated user in the provider to determine their group membership, user identifiers (UID, SID, UPN) and other relevant identity information.

Kerberos provider only provides authentication function. It has no concept of identity to contain user information. A Kerberos server can be thought of as a very simple key/value database where keys are names and values are secret keys (passwords). OneFS supports Microsoft Kerberos through Microsoft Active Directory authentication provider and MIT Kerberos authentication providers on a OneFS cluster.

1.2.1.1 MIT Kerberos authentication provider

In OneFS implementation, MIT Kerberos works independently of Active Directory (AD) and supports portion of network protocols such as NFS, HDFS, and HTTP.

A user that has been authenticated through MIT Kerberos will need a source for identity. The user may exist anywhere, but it is recommended to store the user in an LDAP server. In this case, the LDAP authentication provider should be set to not perform authentication (`--authentication=False`).

1.2.1.2 Microsoft Kerberos through Active Directory authentication provider

OneFS provides Microsoft Kerberos authentication using Active Directory (AD) and supports protocols including NFS, SMB, HDFS, and HTTP. Because the AD service is composed of LDAP, the Microsoft version of Kerberos, and DNS. When adding an AD to a OneFS cluster as an authentication provider, it can act as a

Kerberos server for authentication and a LDAP server for identity management at the same time. The Microsoft Kerberos is available automatically when configuring AD provider.

1.2.2 Service principal name

A service principal name (SPN) represents a service within a cluster and it has a specific secret key stored in the Kerberos server. The SPN identifies not only the user or service, but also the realm that the entity belongs to. A SPN is formed with the identifier and the realm: `<identifier>@<KERBEROS_REALM>`.

- For a user, the identifier is only the Kerberos user name, like `krbuser@EXAMPLE.COM`.
- For a service, the identifier is a combination of the service name and the host name of the machine it runs on with a format of `<service>/<fqdn>@<KERBEROS_REALM>`. Use NFS service as an example here: `nfs/sc01.example.com@EXAMPLE.COM`
- The service name is a case-sensitive string that is specific to the service type, like `host`, `hdfs`, `HTTP`, `ldap`. Not all services have dedicated principal identifiers, for example, the `sshd` service uses the `host` service principal.

In OneFS, SPNs creation are based on the SmartConnect zone name and the cluster FQDN hostname. If the SmartConnect zone names are changed, administrators should update the SPNs to apply the changes.

1.2.3 Domain to realm mapping

In a Kerberos environment, maybe more than one Kerberos realm are implemented, and clients and servers must know which realm they should contact according to service FQDN. An example below defines the domain to realm mapping in `/etc/krb5.conf` file:

```
[domain_realm]
example.com = EXAMPLE.COM
.example.com = EXAMPLE.COM
```

The configuration contains two mappings. The first mapping `example.com = EXAMPLE.COM` indicates that a system with the exact name `example.com` belongs to the `EXAMPLE.COM` realm. The second mapping `.example.com = EXAMPLE.COM` indicates that any system name in the `example.com` DNS domain belongs to the `EXAMPLE.COM` realm.

OneFS configures the mapping automatically for AD providers. When using MIT Kerberos in OneFS, OneFS provides a command to create the domain to realm mapping, shown as below:

```
#isi auth krb5 domain create --realm=EXAMPLE.COM --domain=.example.com
```

2 SMB Kerberos authentication

This section will introduce how Kerberos authentication is used on OneFS for SMB, and list the key considerations and configurations on OneFS cluster.

2.1 SMB authentication methods

As Figure 1 shows, the OneFS cluster will authenticate SMB users in the SMB session setup phase, then users can create SMB sessions to access data stored on the cluster. OneFS supports two types of authentication methods for SMB: Kerberos and NTLM. Kerberos authentication is the first option in the SMB session setup. When Kerberos authentication is not available or failed, authentication method will fall back to NTLM authentication.

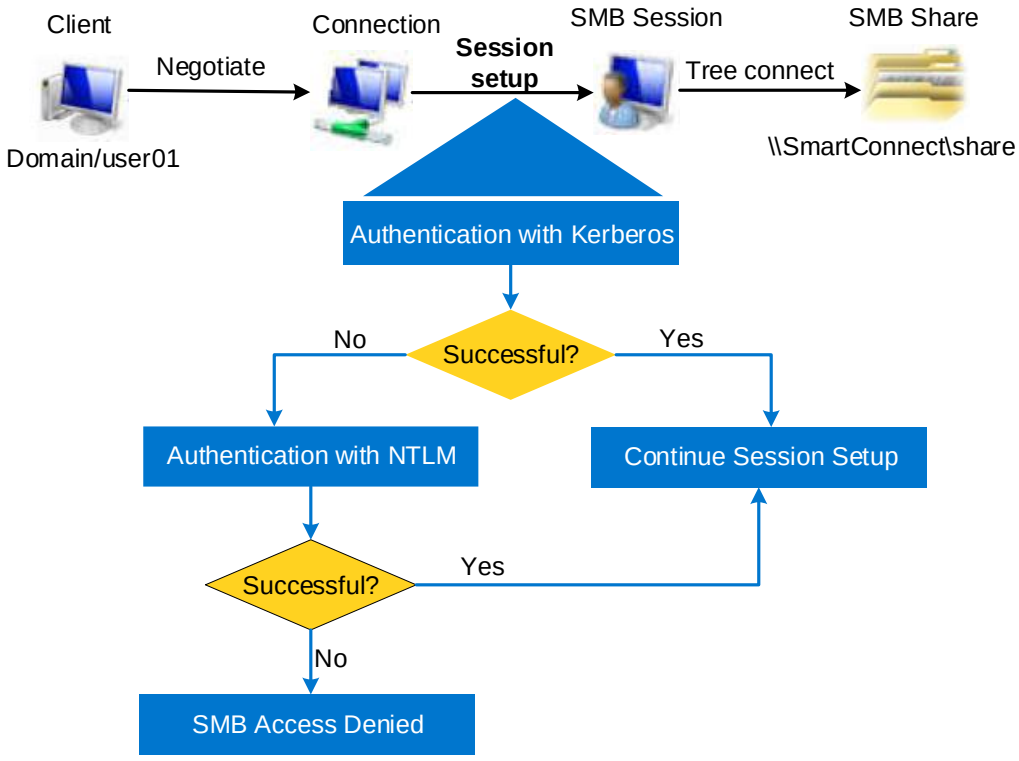

Figure 1 SMB authentication process

2.2 Considerations for Kerberized SMB access

When configuring SMB Kerberos authentication on OneFS, take the following items into considerations:

- The time must be synchronized across SMB clients, OneFS cluster nodes, and Kerberos server (Active Directory in this case); it is recommended to use a NTP server in a Kerberos environment.
- Kerberos relies on being able to resolve host names. Thus, it requires a DNS for host resolution.
- OneFS cluster joins to a domain by creating Active Directory authentication provider.
- Add the Active Directory authentication provider to an access zone.
- Configure SmartConnect for the access zone and create SPNs for SmartConnect zone names.
- Clients should use SmartConnect zone name and domain user for accessing SMB share.
- Use IP address will fall back to NTLM authentication directly.

2.3 Configurations for Kerberized SMB access

The following configurations illustrate key steps to use SMB Kerberos authentication from both cluster side and Windows client side.

1. Create Active Directory authentication provider using the **isi** CLI command.

   ```
   # isi auth ads create --name=example.com --user=administrator --
   groupnet=groupnet0
   ```

2. Add the Active Directory authentication provider to access zone.

   ```
   # isi zone zones modify --name=zone01 --add-auth-providers=lsa-
   activedirectory-provider:EXAMPLE.COM
   ```

3. Configure SmartConnect on OneFS cluster and DNS server. Refer to the Technical Demo: EMC PowerScale SmartConnect to finish the configuration.
4. Check whether SmartConnect zone name SPNs are created on Active Directory by using the isi CLI command as below.

   ```
   # isi auth ads spn check --provider-name=EXAMPLE.COM
   Possible missing SPNs:
             HOST/sc01
             HOST/sc01.example.com
             nfs/sc01.example.com
   ```

   For SMB Kerberos, `HOST/SmartConnect_zone_name` SPNs are required. From the output above, create the following SPNs using the isi CLI command.

   ```
   # isi auth ads spn create --provider-name=EXAMPLE.COM --user=administrator
   --spn=HOST/sc01.example.com
   # isi auth ads spn create --provider-name=EXAMPLE.COM --user=administrator
   --spn=HOST/sc01
   ```

   Alternatly, fix all missing SPNs with the following command quickly.

   ```
   # isi auth ads spn fix --provider-name=EXAMPLE.COM --user=administrator
   ```

If you have access rights to log on your Active Directory domain controller server, you can find the associated SPNs under the advanced feature view in the attribute editor tab as shown in Figure 2.

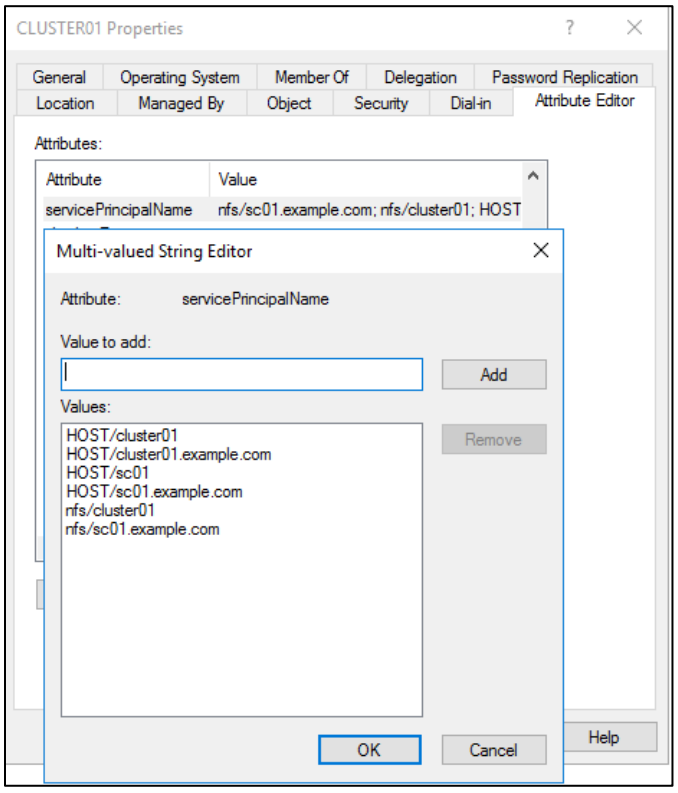

Figure 2 Check SPNs in AD

5. After making the above configuration on cluster, to leverage SMB Kerberos authentication, clients in AD domain need to access SMB share using SmartConnect FQDN along with an AD user account. If the current Windows login user is a domain user who has access right to the SMB share. The Windows client will transparently authenticate using its logon credentials and will prompt for credentials if the Windows credentials failed. For NetBIOS Name Service (NBNS) enabled Windows environment, SMB Kerberos authentication may still be effective even though a Windows client is not a domain member.

6. To verify Kerberos is being used, capture network packets during the SMB connection or using `klist` command. Figure 3 shows an example for `klist` command.

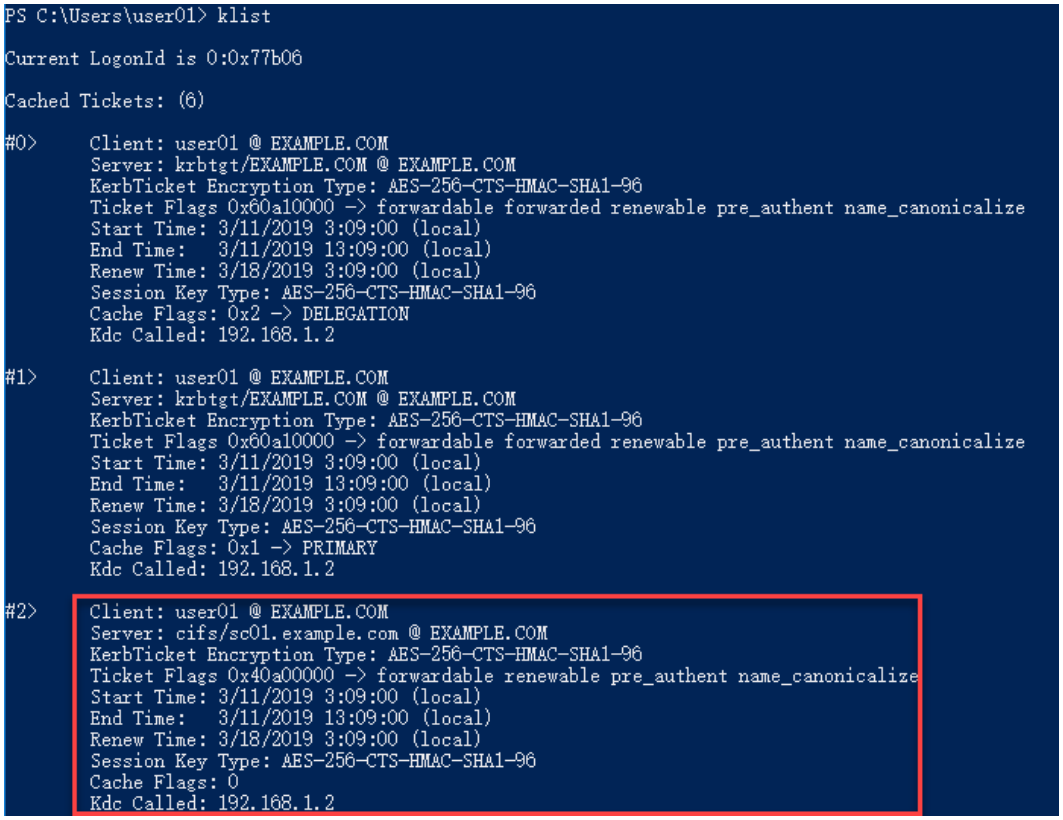

Figure 3 Confirm Kerberos is used

3 NFS Kerberos authentication

There are four authentication methods (also known as security types) supported by OneFS: UNIX, Kerberos5, Kerberos5 Integrity, and Kerberos5 Privacy. Use the `sec` mount option on NFS client to specify a security type. Table 1 shows the details of `sec` option. `sec=sys` is the default mount option; each NFS operation includes the UID/GID of users and authenticate to servers. This authentication method has a risk of spoofing by a different user with same UID/GID. It also has a vulnerability of being tampered with a third party between the client and server on the network. For NFSv3, using Kerberos authentication would mitigate this situation, but is still not completely secure, because Kerberos is only applied to the NFS packets and not the auxiliary services like Network Lock Manager (NLM), Network Status Monitoring (NSM), or mountd. For NFSv4, it is designed as a single protocol without any auxiliary service required, and all NFS operations are authenticated and protected by Kerberos.

In an environment that requires high security for NFS, it is recommended to use NFSv4 instead of NFSv3 and integrate Kerberos authentication with NFS.

Table 1 Mount security types

Options	Authentication methods	Description
sec=sys	UNIX (also known as AUTH_SYS)	The default setting, which uses local UNIX UIDs and GIDs by means of AUTH_SYS to authenticate NFS operations.
sec=krb5	Kerberos only	Use Kerberos V5 instead of local UNIX UIDs and GIDs to authenticate users.
sec=krb5i	Kerberos with integrity	Use Kerberos V5 for user authentication and performs integrity checking of NFS operations using secure checksums to prevent data tampering.
sec=krb5p	Kerberos with integrity and encryption	Use Kerberos V5 for user authentication, integrity checking, and encrypts NFS traffic to prevent traffic sniffing. This is the most secure setting, but it also has the most performance overhead involved.

To use NFSv4, you must have an identical NFSv4 domain name configured on OneFS cluster and NFSv4 clients. With NFSv4 domain, the NFSv4 represents users and groups in the form of `user@domain` or `group@domain` in the results of a get attribute (GETATTR) operation and in the arguments of a set attribute (SETATTR) operation. Figure 4 is a capture of NFSv4 GETATTR operation. As Figure 4 shows, the user/group names have an NFSv4 domain suffix `@vlab.local` in the GETATTR operation.

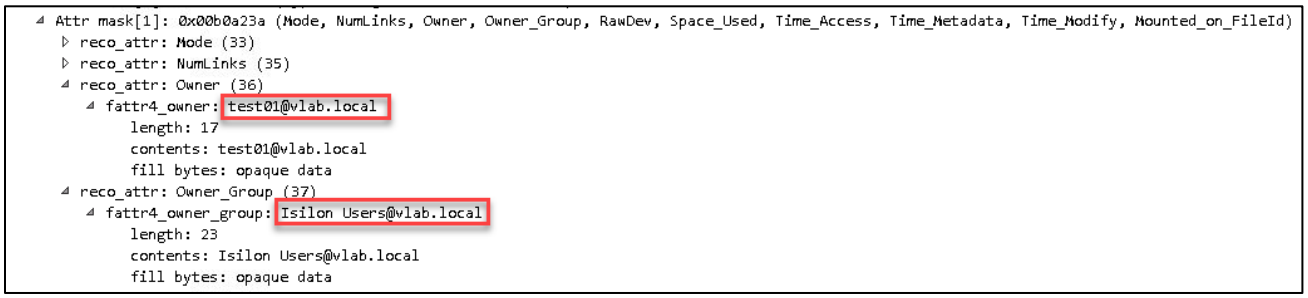

Figure 4 NFSv4 user and group format

The Kerberos-related configurations for NFSv3 and NFSv4 are same. This section will show the key configuration on OneFS when using Microsoft Kerberos or MIT Kerberos and will provide a sample configuration for CentOS 7 Linux clients as an example in appendix B.

3.1 Microsoft Kerberos using AD

3.1.1 Considerations for Kerberized NFS with AD

When configuring NFS on OneFS with Microsoft AD Kerberos, take the following items into considerations:

- The time must be synchronized across NFS clients, OneFS cluster nodes, and Kerberos server (Windows Active Directory in this case), it is recommended to use a NTP server in a Kerberos environment.
- Kerberos relies on being able to resolve host names. Thus, it requires a DNS for host resolution.
- Consistent UID/GID information for domain user accounts from NFS clients and OneFS. It is recommended to enable RFC2307 attributes in AD for this purpose. Refer to the article here for details. Required attributes include `uidNumber`, `gidNumber`, `loginshell`, `unixHomeDirectory`.
- OneFS cluster joins to a domain by creating Active Directory authentication provider with RFC2307 enabled.
- Add the Active Directory authentication provider to an access zone.
- Configure SmartConnect for the access zone in advance. SPNs for SmartConnect zone names must be created.
- OneFS NFS exports with Kerberos security type enabled.
- Clients should use SmartConnect zone name and domain user for accessing NFS export.
- SPNs for NFS for OneFS and client must exist.

3.1.2 Configurations for Kerberized NFS with AD

The following configurations illustrate key steps to use NFS Kerberos authentication.

1. Create Active Directory authentication provider using the **isi** CLI command. NFS clients cannot recognize the auto-generated UID/GID assigned by OneFS, so we need to enable Service for UNIX with `---sfu-support` option for consistent UID/GID information. This will specify OneFS retrieve UID/GID from AD RFC2307 attributes instead of using auto-generated UID/GID.

   ```
   # isi auth ads create --name=example.com --user=administrator --
   groupnet=groupnet0 --sfu-support=rfc2307 --kerberos-nfs-spn=true
   ```

2. Add the Active Directory authentication provider to access zone.

   ```
   # isi zone zones modify --name=KrbZone --add-auth-providers=lsa-
   activedirectory-provider:EXAMPLE.COM
   ```

3. Configure SmartConnect on OneFS cluster and DNS server. Refer to the Technical Demo: EMC PowerScale SmartConnect to finish the configuration.

4. Check whether SmartConnect zone name SPNs are created on Active Directory by using the isi CLI command as below.

```
# isi auth ads spn check --provider-name=EXAMPLE.COM
Possible missing SPNs:
          HOST/sc01
          HOST/sc01.example.com
          nfs/sc01.example.com
```

When joining an Active Directory domain, the cluster will register cluster name and any SmartConnect zone names as Service Principal Names (SPNs) configured on the cluster with the machine account on the domain. Any additional SmartConnect zone names created after joining the cluster to the domain must have SPNs manually added.

Fix all the missing SPNs with the following command quickly.

```
# isi auth ads spn fix --provider-name=EXAMPLE.COM --user=administrator
```

If you have access rights to log on your Active Directory domain controller server, you can find and edit the associated SPNs under the advanced feature view in the attribute editor tab for the OneFS cluster machine account as shown in Figure 5

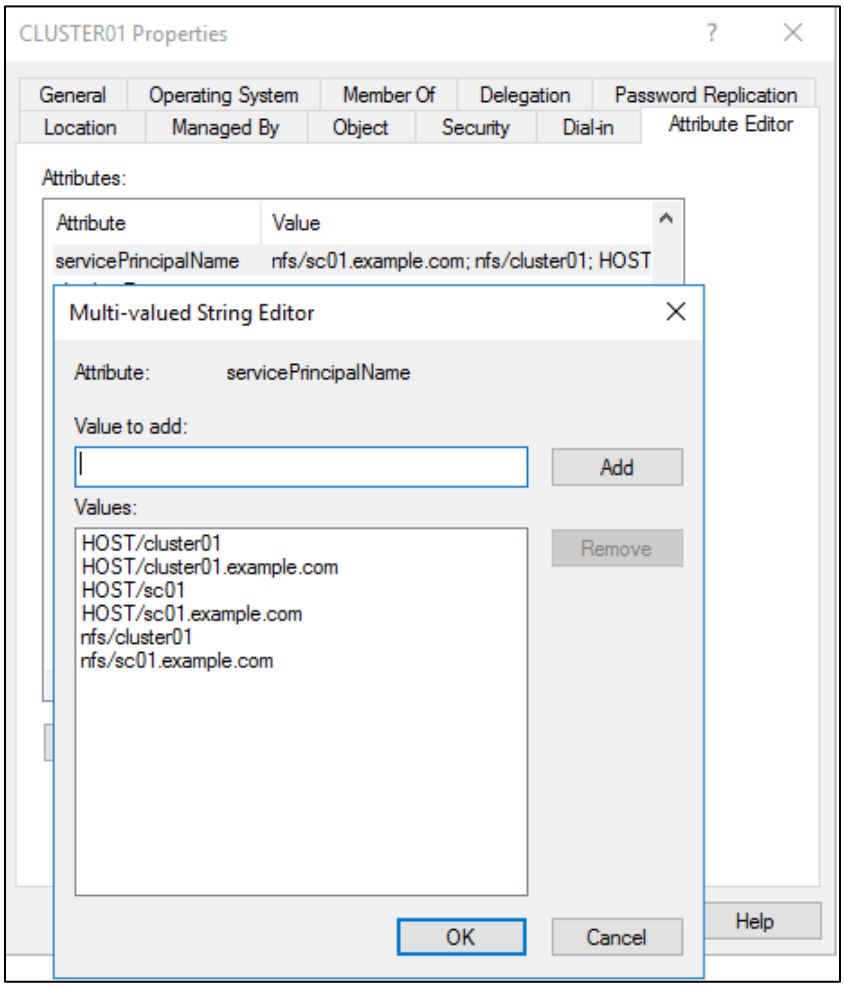

Figure 5 Check SPNs in AD

5. For NFSv4, enable the NFSv4 service and configure NFSv4 domain name in the specific access zone.

```
# isi nfs settings global modify --nfsv4-enabled=true
# isi nfs settings global view
      NFSv3 Enabled: Yes
      NFSv4 Enabled: Yes
NFS Service Enabled: Yes
# isi nfs settings zone modify --zone=KrbZone --nfsv4-domain=example.com
# isi nfs settings zone view --zone=KrbZone
           NFSv4 Domain: example.com
   NFSv4 Replace Domain: Yes
        NFSv4 No Domain: No
   NFSv4 No Domain UIDs: Yes
         NFSv4 No Names: No
NFSv4 Allow Numeric Ids: Yes
```

6. Create a NFS export with Kerberos security type enabled on OneFS cluster. The below command enables all supported security type. Refer to Table 1 for details about the security types.

```
# isi nfs exports create --paths=/ifs/nfs --zone=KrbZone --security-
flavors=unix,krb5,krb5i,krb5p
```

7. Kerberize NFS client by integrating with AD and mount NFS export with Kerberos authentication using `sec` option as shown in Table 1. Refer to appendix B.1 for a CentOS 7 sample configuration.

3.2 MIT Kerberos

3.2.1 Considerations for Kerberized NFS with MIT Kerberos

When configuring NFS on OneFS with MIT Kerberos, take the following items into considerations:

- The time must be synchronized across NFS clients, OneFS cluster nodes, and Kerberos server. It is recommended to use a NTP server in a Kerberos environment.
- Kerberos relies on being able to resolve host names. Thus, it requires a DNS for host resolution.
- Use consistent UID/GID information for users from NFS clients and OneFS. It is recommended to use a central identity store, such as LDAP server for this purpose.
- Add the MIT Kerberos authentication provider to an access zone.
- Configure SmartConnect for the access zone in advance. SPN for SmartConnect zone names must be created.
- OneFS NFS exports with Kerberos security type enabled.
- Clients should use SmartConnect zone name and Kerberos user for accessing NFS export.
- SPNs for NFS for OneFS and client must exist.

3.2.2 Configurations for Kerberized NFS with MIT Kerberos

The following configurations illustrate key steps to use NFS Kerberos authentication.

1. Create the realm using the following command:

   ```
   # isi auth krb5 realm create --realm=EXAMPLE.COM --kdc=kdc.example.com --
   admin-server=kdc.example.com
   ```

2. Create domains for the realm. This step configures the domain to realm mapping described in section 1.2. It is recommended to add both the following two domain-realm mappings for a domain. The first one specifies that any system in the `example.com` domain belongs to the `EXAMPLE.COM` realm. The second specifies that a system with the exact name `example.com` is also in the realm. (The distinction between a domain and a specific host is marked by the presence or lack of an initial "."). Note that the realm is case sensitive and must always be used with the same case.

   ```
   # isi auth krb5 domain create --realm=EXAMPLE.COM --domain=.example.com
   # isi auth krb5 domain create --realm=EXAMPLE.COM --domain=example.com
   ```

3. Create the actual MIT Kerberos provider with the user who has the permission to create SPNs in the Kerbers realm, for example, `root/admin`. This command will add SPNs for OneFS cluster into Kerberos server.

   ```
   # isi auth krb5 create --realm=EXAMPLE.COM --user=kadmin/admin --
   groupnet=groupnet
   ```

4. Add the LDAP provider to OneFS. Kerberos user accounts are followed by realm name, for example, `user01@EXAMPLE.COM`. To make OneFS recognize the user from LDAP followed by realm name, we need to specify the `--provider-domain` option with Kerberos realm name. If configure the option incorrectly, OneFS will map all incoming Kerberized NFS requests to a special `nobody` user and result in permission issues.

   ```
   # isi auth ldap create --name=ldap01 --server-uris=ldap://ldap_fqdn --
   base-dn=dc=example,dc=com --bind-dn=cn=admin,dc=example,dc=com --
   groupnet=groupnet0 --provider-domain=EXAMPLE.COM --authentication=false
   ```

5. Add the LDAP and MIT Kerberos authentication provider to access zone. The NFS file permission is tightly associated with users' UID and GID information, and inconsistent mapping between user name and UID/GID will cause unexpected file access issues. Thus, when integrating OneFS cluster into MIT Kerberos environment, it is recommended to prepare a MIT Kerberos server with the LDAP backend and add both the LDAP server and MIT Kerberos server as OneFS authentication providers. In this way, administrators can maintain a central identity and authentication source to provide a consistent user information (UID/GID) between NFS clients and OneFS cluster.

   ```
   # isi zone zones modify --name=zone01 --add-auth-providers=lsa-ldap-
   provider:ldap01,lsa-krb5-provider:EXAMPLE.COM
   ```

6. Configure SmartConnect on the OneFS cluster and DNS server. Refer to the Technical Demo: EMC PowerScale SmartConnect to finish the configuration.
7. Check whether SmartConnect zone name SPNs are created on MIT Kerberos server using the isi CLI command as follows. And fix it if there are any missing SPNs. This is similar to AD Kerberos.

   ```
   # isi auth krb5 spn check –provider-name=EXAMPLE.COM
   # isi auth krb5 spn fix –provider-name=EXAMPLE.COM --user=root/admin
   ```

8. For NFSv4, enable the NFSv4 service and configure the NFSv4 domain name in the specific access zone.

```
# isi nfs settings global modify --nfsv4-enabled=true
# isi nfs settings global view
      NFSv3 Enabled: Yes
      NFSv4 Enabled: Yes
NFS Service Enabled: Yes
# isi nfs settings zone modify --zone=mitZone --nfsv4-domain=example.com
# isi nfs settings zone view --zone=mitZone
           NFSv4 Domain: example.com
   NFSv4 Replace Domain: Yes
        NFSv4 No Domain: No
   NFSv4 No Domain UIDs: Yes
         NFSv4 No Names: No
NFSv4 Allow Numeric Ids: Yes
```

9. Create an NFS export with the Kerberos security type enabled on the OneFS cluster. The following command enables all supported security type. Refer to Table 1 for details about the security types.

```
# isi nfs exports create --paths=/ifs/nfs --zone=mitZone --security-
flavors=unix,krb5,krb5i,krb5p
```

10. Kerberize the NFS client by integrating with LDAP and MIT Kerberos, and mount the NFS export with Kerberos authentication using `sec` option as shown in Table 1. Refer to appendix B.2 for a CentOS 7 sample configuration.

4 Common issues

This section shows several common issues when using NFS with Kerberos and provide troubleshooting tips.

4.1 Access denied by server while mounting

When mounting an NFS export with Kerberos authentication, the OneFS cluster does not allow the mount operation and you will see the errors in Figure 6.

```
krbuser@RDUVNODE402265:~ $ sudo mount -t nfs -vo nfsvers=4.0,sec=krb5 sc01.demo.local:/ifs/nfs /mnt/
mount.nfs: timeout set for Wed Apr  3 06:39:00 2019
mount.nfs: trying text-based options 'nfsvers=4.0,sec=krb5,addr=192.168.1.111,clientaddr=192.168.1.15'
mount.nfs: mount(2): Permission denied
mount.nfs: access denied by server while mounting sc01.demo.local:/ifs/nfs
```

Figure 6 Access denied error

There are two possible reasons for the issue as listed below:

- Missing client SPNs on Kerberos server and local keytab file. You can use `klist -k` command to list all local SPNs on client. And check whether the host SPN for the client is existing. If not, follow Appendix B.2 to add it.
- The `rpcgssd` service is not available. For example, in CentOS 7, you can use command `systemctl status rpcgssd` to check the status of the service. And start it using command `systemctl start rpcgssd`.

4.2 Ownership shows as nobody

When listing file details in NFS export from client, all existing and newly created files' ownership are always shown as `nobody`. The `nobody` is a special account in most Linux environments. When the NFS client cannot resolve UID/GID to a corresponding username/groupname, the UID/GID will be translated into the `nobody` account.

```
krbuser@RDUVNODE402265:/mnt $ ll
total 7
drwxr-xr-x. 2 nobody nobody 0 Apr  3 06:57 dir1
drwxr-xr-x. 2 nobody nobody 0 Apr  3 06:57 dir2
-rw-r--r--. 1 nobody nobody 0 Apr  3 06:57 file1
-rw-r--r--. 1 nobody nobody 0 Apr  3 06:57 file2
```

When using Kerberos for NFS, this issue could be caused by the following items:

- If NFSv4 is used to mount an export, check whether the NFSv4 domain is configured consistently on OneFS cluster and NFSv4 client. Check the NFSv4 domain setting on OneFS using command `isi nfs settings zone view --zone=zone_name` and check the `Domain=nfsv4_domain_name` option under file `/etc/idmapd` on NFS client. The setting must be identical on both sides.
- If LDAP is used as the authentication provider in OneFS. The `--provider-domain` option must be configured as Kerberos realm name. To check the setting on OneFS, using the following command:
- `# isi auth ldap view –provider-name=ldap_provider_name | grep "Provider Domain"`

A Simplified Kerberos authentication process

The following terms are used in this section:

- **User:** A user represents a person who needs to access a network service, like file share service.
- **Client:** A client is also an entity uses service in the network, but it could be a person or a computer.
- **Server:** A server stands for an entity that provides service to users or clients. Usually, a client will request service to a server on behalf of a user.

When it comes to Kerberos environment, clients and servers both act as Kerberos clients to the Kerberos server. The Kerberos server hosts a database to store the secret keys of its Kerberos clients. For a user, the secret key is the password of the user. For a host, the secret key is a random string generated and stored by Kerberos server, the client also stores the secret key in a keytab file.

The Kerberos server, also known as Key Distribution Center (KDC), contains two services:

- **Authentication Service (AS):** Responsible for initializing a Ticket-Granting Ticket (TGT) to client, the TGT is a special ticket used to authenticate a Kerberos client to a Kerberos server.
- **Ticket-Granting Service (TGS):** Responsible for granting tickets to clients for accessing network services on servers.

There are three phases when a client authenticates to a service using Kerberos. A simplified process is described as follows. Refer to RFC4120 for details on the Kerberos authentication.

Phase 1: Getting the initial Ticket-Granting Ticket (TGT)

This phase typically used at the initiation of a login session to get TGT from AS. As mentioned above, Kerberos uses TGT to authenticate Kerberos clients who are requesting to access other servers. The TGT will subsequently be used to request tickets for other servers without requiring further use of the client's secret key.

Phase 2: Requesting for a server ticket

After getting the initial TGT, the client sends a request to TGS. The request contains the name of the server, the TGT received at phase 1, and an authenticator to prove the client itself. The TGS will check the TGT and authenticator. If valid, the TGS grants a ticket to the client, and the ticket contains the client's name, the server's name, timestamp, and other information along with a newly generated random session key to be used between the client and the server.

Phase 3: Requesting to access a service

In order to get access to the service in the server, the client builds an authenticator encrypted with session key. Then the client sends the authenticator and the server ticket to the server in a way defined by application, for example, via a SMB session setup message.

Figure 7 shows the messages exchange in a Kerberos process.

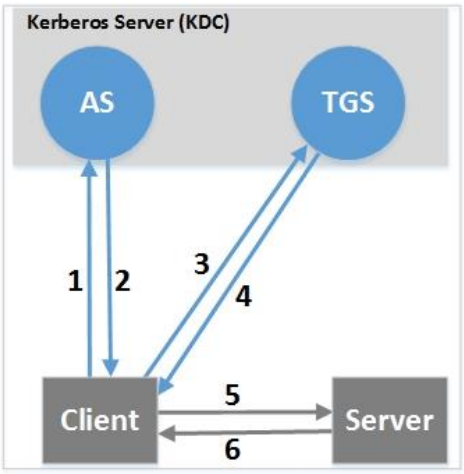

Figure 7 Simplified Kerberos authentication process

1. The client sends an as_req message to AS for requesting a TGT.
2. AS replies an as_rep message to client. The message contains a session key between client and TGS, and a TGT encrypted with TGS secret key.
3. The client sends a tgs_req message to TGS for requesting a ticket for the server. The message contains an authenticator encrypted with the session key, a TGT, and other information.
4. TGT replies a tgs_rep message to client. The message contains a new session key for the client and the server, a ticket for the target server encrypted with the server's secret key.
5. The client sends an ap_req message to the server. The message contains an authenticator encrypted with the new session key, the encrypted server ticket.
6. This response is optional and only used when the user requires mutual-authentication by the server.

B Sample configuration for Kerberizing CentOS 7

In this sample configuration, we use the System Security Services Daemon (SSSD) to connect CentOS client to external identity and authentication providers, including an LDAP directory, an Active Directory (AD), or a Kerberos realm. The SSSD service access remote identity and authentication providers through a common framework which provides local cache and offline authentication support to the system. This is a recommended solution with the following advantages:

- Reduced load on identity and authentication servers by using local cache. Clients request information from servers only when the user information is not available in the SSSD local cache.
- Offline authentication. SSSD optionally keeps a cache of user identities and credentials retrieved from external services. Thus, they can still be successfully authenticated even if remote servers are offline.

When it comes to introduce a Linux system to an AD environment, the most convenient deployment way is to use `realmd` service. The `realmd` service provides a standard method to configure authentication and domain membership. It automatically discovers available domain information and join a domain without complicate manual configuration.

NFS mount uses rpc.gssd for the Kerberos authentication process; the rpc.gssd uses the keys found in keytab (`/etc/krb5.keytab`) to obtain machine credentials. In the old version of rpc.gssd, it used only `nfs/<hostname>@<REALM>` SPN keys found within the keytab. In newer versions of rpc.gssd, `host/<hostname>@<REALM>` SPN keys can also be used. Therefore, it is recommended to add both host SPN and nfs SPN when configuring Kerberos authentication for NFS. Refer to the man page of rpc.gssd for more details. When the user accesses resources on the mount, their TGT will be used to get a TGS for the NFS service which will be used for access checks. Note that the NFS mount does **not** use the users TGT or TGS. The client machines credentials are used for mount.

B.1 Kerberize CentOS 7 with Active Directory

There are different methods to integrate Linux systems with AD environment, for example, when using native LDAP and Kerberos PAM and NSS modules, Samba Winbind, or SSSD. In this sample, we use SSSD as the component to Kerberize CentOS 7 with AD. In the following steps, we use CentOS 7.5 and Windows 2016 Active Directory.

1. Install the System Security Services Daemon (SSSD).

   ```
   # yum install sssd
   ```

   All supported providers packages for AD, LDAP, and Kerberos will also be installed. Using Active Directory as a provider for SSSD is a complex task, there are a number of different configuration parameters for each underlying service (NSS, PAM, Kerberos) and for SSSD itself. Thus, we use `realmd` system in step 4 to simplify the task.
2. Install the required packages to use `realmd` system. It is recommended to install the `oddjob`, `oddjob-mkhomedir`, and `adcli` packages for management purpose using `realmd`.

   ```
   # yum install realmd samba-common-tools
   ```

3. Install the required packages for a Kerberos client.

   ```
   # yum install krb5-workstation
   ```

4. Join the client to a domain using `realm join` command.

   ```
   # realm join example.com -U administrator --automatic-id-mapping=no
   ```

   The `realmd` system provides a clear and simple method to discover and join domains to achieve AD integration. It configures underlying Linux system services automatically to connect to the domain, such as some of key configurations files: `/etc/sssd/sssd.conf`, `/etc/nsswitch`, `/etc/pam.d/system-auth`, `/etc/krb5.conf`, `/etc/krb5.keytab`.
   By default, SSSD maps Windows SIDs to UIDs/GIDs in local system. The mapping information is valid on local only. Thus, use consistent domain users UIDs/GIDs across OneFS cluster and all NFS clients. It is recommended to use `realm` utility to disable ID mapping in SSSD with `--automatic-id-mapping=no` option. This option configures SSSD to use POSIX attributes (RFC2307) defined in AD. If a client already joins a domain without disable ID mapping, an alternative method is to add `ldap_id_mapping=False` setting in SSSD configuration file `/etc/sssd/sssd.conf`.
5. Use `klist -k` to verify that the client's SPNs exist in keytab. If this does not exist, refer to article here to generate keytabs in AD and import into client using `ktutil` tool.
6. Confirm the client can retrieve RFC2307 attributes for users from AD using `id` command.

   ```
   # id user01@EXAMPLE.COM
   uid=10001 (user01@example.com) gid=10000 (domain users@example.com)
   groups=10000 (domain users@example.com), 10001 (linuxusers@example.com)
   ```

   Use `kinit` to request a Kerberos ticket from AD for an AD user. And check the ticket using `klist` command.

   ```
   # kinit user01@EXAMPLE.COM
   Password for user01@EXAMPLE.COM:
   # klist
   Ticket cache: KEYRING:persistent:0:0
   Default principal: user01@EXAMPLE.COM

   Valid starting     Expires            Service principal
   26/02/19 02:53:29  26/02/19 12:53:29   krbtgt/EXAMPLE.COME@EXAMPLE.COM
           renew until 05/03/19 02:53:24
   ```

7. To make a client works correctly in an NFS environment, ensure these configurations are applied.

   a. For NFSv4 only, the NFSv4 user representation is based on NFSv4 domain name in the format of `user@domainname` for name resolution, thus we need to modify the configuration file `/etc/idmad.conf` on the client to contain following settings:

      > Add NFSv4 domain setting: `Domain=example.com`
      > Add SSSD as the method of NFSv4 user ID <=> Name mapper: `Method=nsswitch,sss`

      Start `rpcidmapd` service using `systemctl start rpcidmapd`

   b. To use NFS Kerberos authentication, the kernel needs to load the `rpcsec_gss_krb5` and `auth_rpcgss` modules. To configure the modules, using these commands:

      ```
      # modprobe auth_rpcgss
      # modprobe rpcsec_gss_krb5
      # depmod –a
      ```

c. Add `SECURE_NFS="yes"` to file `/etc/sysconfig/nfs` on the client. And restart the `rpcgssd` service using commands:

```
# systemctl restart rpcgssd.
```

8. Log in to the client with an AD user. Mount the NFS export and access the directory to your data.

a. Mount the NFS export using the mount command.

```
# sudo mount -t nfs -vo nfsvers=4.0,sec=krb5 sc01.example.com:/ifs/nfs
mnt/nfs
```

b. Mount the NFS export using fstab. Add an entry to file `/etc/fstab`, an example shown below. The client will mount the export automatically after reboot.

```
sc01.example.local:/ifs/nfs /mnt/nfs  nfs4  rw,vers=4.0,sec=krb5 0 0
```

You can view the NFS ticket using `klist` command. This NFS ticket indicates the NFS client has connected to OneFS cluster NFS service through SmartConnect name `sc01.example.com` with user account `user01@EXAMPLE.COM`.

```
# klist
Ticket cache: KEYRING:persistent:10001:krb_ccache_Ew4cwW9
Default principal: user01@EXAMPLE.COM

Valid starting     Expires             Service principal
26/02/19 03:26:34  26/02/19 13:24:18   nfs/sc01.example.com@EXAMPLE.COM
        renew until 05/03/19 03:24:18
26/02/19 03:24:18  26/02/19 13:24:18   krbtgt/EXAMPLE.COME@EXAMPLE.COM
        renew until 05/03/19 03:24:18
```

B.2 Kerberize CentOS 7 with MIT Kerberos

The following steps configure a CentOS 7 client integrated into a MIT Kerberos server with an LDAP backend. Similar to AD Kerberos, we use SSSD to complete the configuration.

1. Install the required packages for the System Security Services Daemon (SSSD) and Kerberos client.

```
# yum install sssd krb5-workstation
```

2. Enable SSSD as the identity and authentication provider.

```
# authconfig --update --enablesssd --enablesssdauth --enablemkhomedir
```

The `authconfig` tool will configure the related service automatically, such as NSS and PAM services. Configure LDAP as identity provider and Kerberos as authentication provider for SSSD. Modify `/etc/sssd/sssd.conf` file to contain the settings for domain (an example is shown below). Make sure the file is accessible only by the owner and owned by root.

```
[sssd]
domains = example.com
services = nss, pam
config_file_version = 2
```

```
[domain/example.com]
id_provider = ldap
ldap_uri = ldap://ldap.example.com
ldap_search_base = dc=example,dc=com

auth_provider = krb5
krb5_server = kerberos.example.com
krb5_realm = EXAMPLE.COM
```

3. Enable the SSSD service and start it.

```
# systemctl enable sssd
# systemctl start sssd
```

4. Edit the `/etc/krb5.conf` file to contain the following settings under `[libdefaults]`, `[realms]` and `[domain_realm]` sections.

```
[libdefaults]
  …
  default_realm = EXAMPLE.COM
  …
[realms]
   EXAMPLE.COM = {
    kdc = kdc.example.com
    admin_server = kdc.example.com
  }

[domain_realm]
 .example.com = EXAMPLE.COM
 example.com = EXAMPLE.COM
```

5. Confirm the client can get information for users from LDAP server using the `id` command.

```
# id user01@EXAMPLE.COM
uid=10001(user01) gid=10000(ldapusers) groups=10000(ldapusers)
```

Use `kinit` to request a Kerberos ticket for an LDAP user. And check the ticket using `klist` command.

```
# kinit user01@EXAMPLE.COM
Password for user01@EXAMPLE.COM:
# klist
Ticket cache: KEYRING:persistent:0:0
Default principal: user01@EXAMPLE.COM

Valid starting     Expires            Service principal
02/03/19 12:31:24  03/03/19 12:31:24   krbtgt/EXAMPLE.COME@EXAMPLE.COM
```

6. During the NFS mount process, the host and nfs principal name for NFS client must exist on both the Kerberos server and local client in keytab. Thus, we log in to the remote Kerberos server principal database using the `kadmin` tool and create principal names for NFS client with `addprinc` command.

Within the same `kadmin` session, export the newly created principal names into local NFS client file named `krb5.keytab` along with multiple supported encryption type.

```
# kadmin -r EXAMPLE.COM -p kadmin/admin@EXAMPLE.COM
Authenticating as principal kadmin/admin@EXAMPLE.COM with password.
Password for kadmin/admin@EXAMPLE.COM:
kadmin:  addprinc -randkey host/nfsclient.example.com
WARNING: no policy specified for host/nfsclient.example.com@EXAMPLE.COM;
defaulting to no policy
Principal "host/nfsclient.example.com@EXAMPLE.COM" created.
kadmin:  addprinc -randkey nfs/nfsclient.example.com
WARNING: no policy specified for nfs/nfsclient.example.com@EXAMPLE.COM;
defaulting to no policy
Principal "nfs/nfsclient.example.com@EXAMPLE.COM" created.
kadmin:  ktadd -k /etc/krb5.keytab host/nfsclient.example.com
Entry for principal host/nfsclient.example.com with kvno 2, encryption
type aes256-cts-hmac-sha1-96 added to keytab WRFILE:/etc/krb5.keytab.
Entry for principal host/nfsclient.example.com with kvno 2, encryption
type aes128-cts-hmac-sha1-96 added to keytab WRFILE:/etc/krb5.keytab.
Entry for principal host/nfsclient.example.com with kvno 2, encryption
type des3-cbc-sha1 added to keytab WRFILE:/etc/krb5.keytab.
Entry for principal host/nfsclient.example.com with kvno 2, encryption
type arcfour-hmac added to keytab WRFILE:/etc/krb5.keytab.
Entry for principal host/nfsclient.example.com with kvno 2, encryption
type camellia256-cts-cmac added to keytab WRFILE:/etc/krb5.keytab.
Entry for principal host/nfsclient.example.com with kvno 2, encryption
type camellia128-cts-cmac added to keytab WRFILE:/etc/krb5.keytab.
Entry for principal host/nfsclient.example.com with kvno 2, encryption
type des-hmac-sha1 added to keytab WRFILE:/etc/krb5.keytab.
Entry for principal host/nfsclient.example.com with kvno 2, encryption
type des-cbc-md5 added to keytab WRFILE:/etc/krb5.keytab.
kadmin:  ktadd -k /etc/krb5.keytab nfs/nfsclient.example.com
Entry for principal nfs/nfsclient.example.com with kvno 2, encryption type
aes256-cts-hmac-sha1-96 added to keytab WRFILE:/etc/krb5.keytab.
Entry for principal nfs/nfsclient.example.com with kvno 2, encryption type
aes128-cts-hmac-sha1-96 added to keytab WRFILE:/etc/krb5.keytab.
Entry for principal nfs/nfsclient.example.com with kvno 2, encryption type
des3-cbc-sha1 added to keytab WRFILE:/etc/krb5.keytab.
Entry for principal nfs/nfsclient.example.com with kvno 2, encryption type
arcfour-hmac added to keytab WRFILE:/etc/krb5.keytab.
Entry for principal nfs/nfsclient.example.com with kvno 2, encryption type
camellia256-cts-cmac added to keytab WRFILE:/etc/krb5.keytab.
Entry for principal nfs/nfsclient.example.com with kvno 2, encryption type
camellia128-cts-cmac added to keytab WRFILE:/etc/krb5.keytab.
Entry for principal nfs/nfsclient.example.com with kvno 2, encryption type
des-hmac-sha1 added to keytab WRFILE:/etc/krb5.keytab.
Entry for principal nfs/nfsclient.example.com with kvno 2, encryption type
des-cbc-md5 added to keytab WRFILE:/etc/krb5.keytab.
```

7. To make sure a client works correctly in a NFS environment, ensure these configurations are applied.

   a. For NFSv4 only, the NFSv4 user representation is based on NFSv4 domain name in the format of `user@domainname` for name resolution. Thus, we need to modify the configuration file `/etc/idmad.conf` on the client to contain following settings:

      > Add NFSv4 domain setting: `Domain=example.com`
      > Add SSSD as the method of NFSv4 ID <=> Name mapper: `Method=nsswitch,sss`

      Start `rpcidmapd` service using `systemctl start rpcidmapd`

   b. The kernel needs to load the `rpcsec_gss_krb5` and `auth_rpcgss` modules. To configure the module, use these commands:

      ```
      # modprobe auth_rpcgss
      # modprobe rpcsec_gss_krb5
      # depmod -a
      ```

   c. Add `SECURE_NFS="yes"` to file `/etc/sysconfig/nfs` on the client, and restart the `rpcgssd` service using command:
      `# systemctl restart rpcgssd`.

8. Log in to the client with LDAP user. Mount the NFS export and access the directory to your data.

   a. Mount the NFS export using the mount command.

      ```
      # sudo mount -t nfs -vo nfsvers=4.0,sec=krb5 sc01.example.com:/ifs/nfs
      /mnt/nfs
      ```

   b. Mount the NFS export using fstab. Add an entry to file `/etc/fstab` (an example is shown below). The client will mount the export automatically after reboot.

      ```
      sc01.example.local:/ifs/nfs /mnt/nfs  nfs4  rw,vers=4.0,sec=krb5 0 0
      ```

   You can view the NFS ticket using `klist` command.

   ```
   # klist
   Ticket cache: KEYRING:persistent:40681:krb_ccache_2QXSkuj
   Default principal: user01@EXAMPLE.COM

   Valid starting     Expires            Service principal
   03/03/19 06:46:34  04/03/19 06:46:20  nfs/sc01.example.com@EXAMPLE.COM
   03/03/19 06:46:20  04/03/19 06:46:20  krbtgt/EXAMPLE.COME@EXAMPLE.COM
   ```

C Technical support and resources

Dell.com/support is focused on meeting customer needs with proven services and support.

Storage technical documents and videos provide expertise that helps to ensure customer success on Dell EMC storage platforms.