



Set up file access using NFS

ONTAP 9

NetApp
March 17, 2022

Table of Contents

- Set up file access using NFS 1
 - Set up file access using NFS overview 1
 - Secure NFS access using export policies 1
 - Using Kerberos with NFS for strong security 13
 - Configure name services 18
 - Configure name mappings 28
 - Enable access for Windows NFS clients 33
 - Enable the display of NFS exports on NFS clients 34

Set up file access using NFS

Set up file access using NFS overview

You must complete a number of steps to allow clients access to files on storage virtual machines (SVMs) using NFS. There are some additional steps that are optional depending on the current configuration of your environment.

For clients to be able to access files on SVMs using NFS, you must complete the following tasks:

1. Enable the NFS protocol on the SVM.

You must configure the SVM to allow data access from clients over NFS.

2. Create an NFS server on the SVM.

An NFS server is a logical entity on the SVM that enables the SVM to serve files over NFS. You must create the NFS server and specify the NFS protocol versions you want to allow.

3. Configure export policies on the SVM.

You must configure export policies to make volumes and qtrees available to clients.

4. Configure the NFS server with the appropriate security and other settings depending on the network and storage environment.

This step might include configuring Kerberos, LDAP, NIS, name mappings, and local users.

Secure NFS access using export policies

How export policies control client access to volumes or qtrees

Export policies contain one or more *export rules* that process each client access request. The result of the process determines whether the client is denied or granted access and what level of access. An export policy with export rules must exist on the storage virtual machine (SVM) for clients to access data.

You associate exactly one export policy with each volume or qtree to configure client access to the volume or qtree. The SVM can contain multiple export policies. This enables you to do the following for SVMs with multiple volumes or qtrees:

- Assign different export policies to each volume or qtree of the SVM for individual client access control to each volume or qtree in the SVM.
- Assign the same export policy to multiple volumes or qtrees of the SVM for identical client access control without having to create a new export policy for each volume or qtree.

If a client makes an access request that is not permitted by the applicable export policy, the request fails with a permission-denied message. If a client does not match any rule in the export policy, then access is denied. If an export policy is empty, then all accesses are implicitly denied.

You can modify an export policy dynamically on a system running ONTAP.

Default export policy for SVMs

Each SVM has a default export policy that contains no rules. An export policy with rules must exist before clients can access data on the SVM. Each FlexVol volume contained in the SVM must be associated with an export policy.

When you create an SVM, the storage system automatically creates a default export policy called `default` for the root volume of the SVM. You must create one or more rules for the default export policy before clients can access data on the SVM. Alternatively, you can create a custom export policy with rules. You can modify and rename the default export policy, but you cannot delete the default export policy.

When you create a FlexVol volume in its containing SVM, the storage system creates the volume and associates the volume with the default export policy for the root volume of the SVM. By default, each volume created in the SVM is associated with the default export policy for the root volume. You can use the default export policy for all volumes contained in the SVM, or you can create a unique export policy for each volume. You can associate multiple volumes with the same export policy.

How export rules work

Export rules are the functional elements of an export policy. Export rules match client access requests to a volume against specific parameters you configure to determine how to handle the client access requests.

An export policy must contain at least one export rule to allow access to clients. If an export policy contains more than one rule, the rules are processed in the order in which they appear in the export policy. The rule order is dictated by the rule index number. If a rule matches a client, the permissions of that rule are used and no further rules are processed. If no rules match, the client is denied access.

You can configure export rules to determine client access permissions using the following criteria:

- The file access protocol used by the client sending the request, for example, NFSv4 or SMB.
- A client identifier, for example, host name or IP address.
- The security type used by the client to authenticate, for example, Kerberos v5, NTLM, or AUTH_SYS.

If a rule specifies multiple criteria, the client must match all of them for the rule to apply.



Beginning with ONTAP 9.3, you can enable export policy configuration checking as a background job that records any rules violations in an error rule list. The `vserver export-policy config-checker` commands invoke the checker and display results, which you can use to verify your configuration and delete erroneous rules from the policy.

The commands only validate export configuration for host names, netgroups, and anonymous users.

Example

The export policy contains an export rule with the following parameters:

- `-protocol nfs3`

- `-clientmatch 10.1.16.0/255.255.255.0`
- `-rorule any`
- `-rwrule any`

The client access request is sent using the NFSv3 protocol and the client has the IP address 10.1.17.37.

Even though the client access protocol matches, the IP address of the client is in a different subnet from the one specified in the export rule. Therefore, client matching fails and this rule does not apply to this client.

Example

The export policy contains an export rule with the following parameters:

- `-protocol nfs`
- `-clientmatch 10.1.16.0/255.255.255.0`
- `-rorule any`
- `-rwrule any`

The client access request is sent using the NFSv4 protocol and the client has the IP address 10.1.16.54.

The client access protocol matches and the IP address of the client is in the specified subnet. Therefore, client matching is successful and this rule applies to this client. The client gets read-write access regardless of its security type.

Example

The export policy contains an export rule with the following parameters:

- `-protocol nfs3`
- `-clientmatch 10.1.16.0/255.255.255.0`
- `-rorule any`
- `-rwrule krb5,ntlm`

Client #1 has the IP address 10.1.16.207, sends an access request using the NFSv3 protocol, and authenticated with Kerberos v5.

Client #2 has the IP address 10.1.16.211, sends an access request using the NFSv3 protocol, and authenticated with AUTH_SYS.

The client access protocol and IP address matches for both clients. The read-only parameter allows read-only access to all clients regardless of the security type they authenticated with. Therefore both clients get read-only access. However, only client #1 gets read-write access because it used the approved security type Kerberos v5 to authenticate. Client #2 does not get read-write access.

Manage clients with an unlisted security type

When a client presents itself with a security type that is not listed in an access parameter of an export rule, you have the choice of either denying access to the client or mapping it to the anonymous user ID instead by using the option `none` in the access parameter.

A client might present itself with a security type that is not listed in an access parameter because it was authenticated with a different security type or was not authenticated at all (security type AUTH_NONE). By default, the client is automatically denied access to that level. However, you can add the option `none` to the access parameter. As a result, clients with an unlisted security style are mapped to the anonymous user ID instead. The `-anon` parameter determines what user ID is assigned to those clients. The user ID specified for the `-anon` parameter must be a valid user that is configured with permissions you deem appropriate for the anonymous user.

Valid values for the `-anon` parameter range from 0 to 65535.

| User ID assigned to <code>-anon</code> | Resulting handling of client access requests |
|--|---|
| 0 - 65533 | The client access request is mapped to the anonymous user ID and gets access depending on the permissions configured for this user. |
| 65534 | The client access request is mapped to the user <code>nobody</code> and gets access depending on the permissions configured for this user. This is the default. |
| 65535 | The access request from any client is denied when mapped to this ID and the client presents itself with security type AUTH_NONE. The access request from clients with user ID 0 is denied when mapped to this ID and the client presents itself with any other security type. |

When using the option `none`, it is important to remember that the read-only parameter is processed first. Consider the following guidelines when configuring export rules for clients with unlisted security types:

| Read-only includes <code>none</code> | Read-write includes <code>none</code> | Resulting access for clients with unlisted security types |
|--------------------------------------|---------------------------------------|---|
| No | No | Denied |
| No | Yes | Denied because read-only is processed first |
| Yes | No | Read-only as anonymous |
| Yes | Yes | Read-write as anonymous |

Example

The export policy contains an export rule with the following parameters:

- `-protocol nfs3`
- `-clientmatch 10.1.16.0/255.255.255.0`
- `-rorule sys,none`

- `-rwrule any`
- `-anon 70`

Client #1 has the IP address 10.1.16.207, sends an access request using the NFSv3 protocol, and authenticated with Kerberos v5.

Client #2 has the IP address 10.1.16.211, sends an access request using the NFSv3 protocol, and authenticated with AUTH_SYS.

Client #3 has the IP address 10.1.16.234, sends an access request using the NFSv3 protocol, and did not authenticate (meaning security type AUTH_NONE).

The client access protocol and IP address matches for all three clients. The read-only parameter allows read-only access to clients with their own user ID that authenticated with AUTH_SYS. The read-only parameter allows read-only access as the anonymous user with user ID 70 to clients that authenticated using any other security type. The read-write parameter allows read-write access to any security type, but in this case only applies to clients already filtered by the read-only rule.

Therefore, clients #1 and #3 get read-write access only as the anonymous user with user ID 70. Client #2 gets read-write access with its own user ID.

Example

The export policy contains an export rule with the following parameters:

- `-protocol nfs3`
- `-clientmatch 10.1.16.0/255.255.255.0`
- `-rorule sys,none`
- `-rwrule none`
- `-anon 70`

Client #1 has the IP address 10.1.16.207, sends an access request using the NFSv3 protocol, and authenticated with Kerberos v5.

Client #2 has the IP address 10.1.16.211, sends an access request using the NFSv3 protocol, and authenticated with AUTH_SYS.

Client #3 has the IP address 10.1.16.234, sends an access request using the NFSv3 protocol, and did not authenticate (meaning security type AUTH_NONE).

The client access protocol and IP address matches for all three clients. The read-only parameter allows read-only access to clients with their own user ID that authenticated with AUTH_SYS. The read-only parameter allows read-only access as the anonymous user with user ID 70 to clients that authenticated using any other security type. The read-write parameter allows read-write access only as the anonymous user.

Therefore, client #1 and client #3 get read-write access only as the anonymous user with user ID 70. Client #2 gets read-only access with its own user ID but is denied read-write access.

How security types determine client access levels

The security type that the client authenticated with plays a special role in export rules. You must understand how the security type determines the levels of access the client

gets to a volume or qtree.

The three possible access levels are as follows:

1. Read-only
2. Read-write
3. Superuser (for clients with user ID 0)

Because the access level by security type is evaluated in this order, you must observe the following rules when constructing access level parameters in export rules:

| For a client to get access level... | These access parameters must match the client's security type... |
|-------------------------------------|--|
| Normal user read-only | Read-only (<code>-rorule</code>) |
| Normal user read-write | Read-only (<code>-rorule</code>) and read-write (<code>-rwrule</code>) |
| Superuser read-only | Read-only (<code>-rorule</code>) and <code>-superuser</code> |
| Superuser read-write | Read-only (<code>-rorule</code>) and read-write (<code>-rwrule</code>) and <code>-superuser</code> |

The following are valid security types for each of these three access parameters:

- `any`
- `none`
- `never`

This security type is not valid for use with the `-superuser` parameter.

- `krb5`
- `krb5i`
- `krb5p`
- `ntlm`
- `sys`

When matching a client's security type against each of the three access parameters, there are three possible outcomes:

| If the client's security type... | Then the client... |
|--|--|
| Matches the one specified in the access parameter. | Gets access for that level with its own user ID. |
| Does not match the one specified, but the access parameter includes the option <code>none</code> . | Gets access for that level but as the anonymous user with the user ID specified by the <code>-anon</code> parameter. |

| If the client's security type... | Then the client... |
|---|--|
| Does not match the one specified and the access parameter does not include the option <code>none</code> . | Does not get any access for that level. This does not apply to the <code>-superuser</code> parameter because it always includes <code>none</code> even when not specified. |

Example

The export policy contains an export rule with the following parameters:

- `-protocol nfs3`
- `-clientmatch 10.1.16.0/255.255.255.0`
- `-rorule any`
- `-rwrule sys,krb5`
- `-superuser krb5`

Client #1 has the IP address 10.1.16.207, has user ID 0, sends an access request using the NFSv3 protocol, and authenticated with Kerberos v5.

Client #2 has the IP address 10.1.16.211, has user ID 0, sends an access request using the NFSv3 protocol, and authenticated with AUTH_SYS.

Client #3 has the IP address 10.1.16.234, has user ID 0, sends an access request using the NFSv3 protocol, and did not authenticate (AUTH_NONE).

The client access protocol and IP address matches all three clients. The read-only parameter allows read-only access to all clients regardless of security type. The read-write parameter allows read-write access to clients with their own user ID that authenticated with AUTH_SYS or Kerberos v5. The superuser parameter allows superuser access to clients with user ID 0 that authenticated with Kerberos v5.

Therefore, client #1 gets superuser read-write access because it matches all three access parameters. Client #2 gets read-write access but not superuser access. Client #3 gets read-only access but not superuser access.

Manage superuser access requests

When you configure export policies, you need to consider what you want to happen if the storage system receives a client access request with user ID 0, meaning as a superuser, and set up your export rules accordingly.

In the UNIX world, a user with the user ID 0 is known as the superuser, typically called root, who has unlimited access rights on a system. Using superuser privileges can be dangerous for several reasons, including breach of system and data security.

By default, ONTAP maps clients presenting with user ID 0 to the anonymous user. However, you can specify the `-superuser` parameter in export rules to determine how to handle clients presenting with user ID 0 depending on their security type. The following are valid options for the `-superuser` parameter:

- `any`
- `none`

This is the default setting if you do not specify the `-superuser` parameter.

- `krb5`
- `ntlm`
- `sys`

There are two different ways how clients presenting with user ID 0 are handled, depending on the `-superuser` parameter configuration:

| If the <code>-superuser</code> parameter and the client's security type... | Then the client... |
|--|---|
| Match | Gets superuser access with user ID 0. |
| Do not match | Gets access as the anonymous user with the user ID specified by the <code>-anon</code> parameter and its assigned permissions. This is regardless of whether the read-only or read-write parameter specifies the option <code>none</code> . |

If a client presents with user ID 0 to access a volume with NTFS security style and the `-superuser` parameter is set to `none`, ONTAP uses the name mapping for the anonymous user to obtain the proper credentials.

Example

The export policy contains an export rule with the following parameters:

- `-protocol nfs3`
- `-clientmatch 10.1.16.0/255.255.255.0`
- `-rorule any`
- `-rwrule krb5,ntlm`
- `-anon 127`

Client #1 has the IP address 10.1.16.207, has user ID 746, sends an access request using the NFSv3 protocol, and authenticated with Kerberos v5.

Client #2 has the IP address 10.1.16.211, has user ID 0, sends an access request using the NFSv3 protocol, and authenticated with `AUTH_SYS`.

The client access protocol and IP address matches for both clients. The read-only parameter allows read-only access to all clients regardless of the security type they authenticated with. However, only client #1 gets read-write access because it used the approved security type Kerberos v5 to authenticate.

Client #2 does not get superuser access. Instead, it gets mapped to anonymous because the `-superuser` parameter is not specified. This means it defaults to `none` and automatically maps user ID 0 to anonymous. Client #2 also only gets read-only access because its security type did not match the read-write parameter.

Example

The export policy contains an export rule with the following parameters:

- `-protocol nfs3`
- `-clientmatch 10.1.16.0/255.255.255.0`
- `-rorule any`
- `-rwrule krb5,ntlm`
- `-superuser krb5`
- `-anon 0`

Client #1 has the IP address 10.1.16.207, has user ID 0, sends an access request using the NFSv3 protocol, and authenticated with Kerberos v5.

Client #2 has the IP address 10.1.16.211, has user ID 0, sends an access request using the NFSv3 protocol, and authenticated with AUTH_SYS.

The client access protocol and IP address matches for both clients. The read-only parameter allows read-only access to all clients regardless of the security type they authenticated with. However, only client #1 gets read-write access because it used the approved security type Kerberos v5 to authenticate. Client #2 does not get read-write access.

The export rule allows superuser access for clients with user ID 0. Client #1 gets superuser access because it matches the user ID and security type for the read-only and `-superuser` parameters. Client #2 does not get read-write or superuser access because its security type does not match the read-write parameter or the `-superuser` parameter. Instead, client #2 is mapped to the anonymous user, which in this case has the user ID 0.

How ONTAP uses export policy caches

To improve system performance, ONTAP uses local caches to store information such as host names and netgroups. This enables ONTAP to process export policy rules more quickly than retrieving the information from external sources. Understanding what the caches are and what they do can help you troubleshoot client access issues.

You configure export policies to control client access to NFS exports. Each export policy contains rules, and each rule contains parameters to match the rule to clients requesting access. Some of these parameters require ONTAP to contact an external source, such as DNS or NIS servers, to resolve objects such as domain names, host names, or netgroups.

These communications with external sources take a small amount of time. To increase performance, ONTAP reduces the amount of time it takes to resolve export policy rule objects by storing information locally on each node in several caches.

| Cache name | Type of information stored |
|------------|--|
| Access | Mappings of clients to corresponding export policies |
| Name | Mappings of UNIX user names to corresponding UNIX user IDs |
| ID | Mappings of UNIX user IDs to corresponding UNIX user IDs and extended UNIX group IDs |

| Cache name | Type of information stored |
|------------|--|
| Host | Mappings of host names to corresponding IP addresses |
| Netgroup | Mappings of netgroups to corresponding IP addresses of members |
| Showmount | List of exported directories from SVM namespace |

If you change information on the external name servers in your environment after ONTAP retrieved and stored it locally, the caches might now contain outdated information. Although ONTAP refreshes caches automatically after certain time periods, different caches have different expiration and refresh times and algorithms.

Another possible reason for caches to contain outdated information is when ONTAP attempts to refresh cached information but encounters a failure when attempting to communicate with name servers. If this happens, ONTAP continues to use the information currently stored in the local caches to prevent client disruption.

As a result, client access requests that are supposed to succeed might fail, and client access requests that are supposed to fail might succeed. You can view and manually flush some of the export policy caches when troubleshooting such client access issues.

How the access cache works

ONTAP uses an access cache to store the results of export policy rule evaluation for client access operations to a volume or qtree. This results in performance improvements because the information can be retrieved much faster from the access cache than going through the export policy rule evaluation process every time a client sends an I/O request.

Whenever an NFS client sends an I/O request to access data on a volume or qtree, ONTAP must evaluate each I/O request to determine whether to grant or deny the I/O request. This evaluation involves checking every export policy rule of the export policy associated with the volume or qtree. If the path to the volume or qtree involves crossing one or more junction points, this might require performing this check for multiple export policies along the path.

Note that this evaluation occurs for every I/O request sent from an NFS client, such as read, write, list, copy and other operations; not just for initial mount requests.

After ONTAP has identified the applicable export policy rules and decided whether to allow or deny the request, ONTAP then creates an entry in the access cache to store this information.

When an NFS client sends an I/O request, ONTAP notes the IP address of the client, the ID of the SVM, and the export policy associated with the target volume or qtree, and first checks the access cache for a matching entry. If a matching entry exists in the access cache, ONTAP uses the stored information to allow or deny the I/O request. If a matching entry does not exist, ONTAP then goes through the normal process of evaluating all applicable policy rules as explained above.

Access cache entries that are not actively used are not refreshed. This reduces unnecessary and wasteful communication with external name serves.

Retrieving the information from the access cache is much faster than going through the entire export policy rule

evaluation process for every I/O request. Therefore, using the access cache greatly improves performance by reducing the overhead of client access checks.

How access cache parameters work

Several parameters control the refresh periods for entries in the access cache. Understanding how these parameters work enables you to modify them to tune the access cache and balance performance with how recent the stored information is.

The access cache stores entries consisting of one or more export rules that apply to clients attempting to access volumes or qtrees. These entries are stored for a certain amount of time before they are refreshed. The refresh time is determined by access cache parameters and depends on the type of access cache entry.

You can specify access cache parameters for individual SVMs. This allows the parameters to differ according to SVM access requirements. Access cache entries that are not actively used are not refreshed, which reduces unnecessary and wasteful communication with external name serves.

| Access cache entry type | Description | Refresh period in seconds |
|-------------------------|--|---|
| Positive entries | Access cache entries that have not resulted in access denial to clients. | Minimum: 300 Maximum: 86,400 Default: 3,600 |
| Negative entries | Access cache entries that have resulted in access denial to clients. | Minimum: 60 Maximum: 86,400 Default: 3,600 |

Example

An NFS client attempts to access a volume on a cluster. ONTAP matches the client to an export policy rule and determines that the client gets access based on the export policy rule configuration. ONTAP stores the export policy rule in the access cache as a positive entry. By default, ONTAP keeps the positive entry in the access cache for one hour (3,600 seconds), and then automatically refreshes the entry to keep the information current.

To prevent the access cache from filling up unnecessarily, there is an additional parameter to clear existing access cache entries that have not been used for a certain time period to decide client access. This `-harvest -timeout` parameter has an allowed range of 60 through 2,592,000 seconds and a default setting of 86,400 seconds.

Remove an export policy from a qtree

If you decide you do not want a specific export policy assigned to a qtree any longer, you can remove the export policy by modifying the qtree to inherit the export policy of the containing volume instead. You can do this by using the `volume qtree modify` command with the `-export-policy` parameter and an empty name string (`""`).

Steps

1. To remove an export policy from a qtree, enter the following command:

```
volume qtree modify -vserver vserver_name -qtree-path  
/vol/volume_name/qtree_name -export-policy ""
```

2. Verify that the qtree was modified accordingly:

```
volume qtree show -qtree qtree_name -fields export-policy
```

Validate qtree IDs for qtree file operations

ONTAP can perform an optional additional validation of qtree IDs. This validation ensures that client file operation requests use a valid qtree ID and that clients can only move files within the same qtree. You can enable or disable this validation by modifying the `-validate-qtree-export` parameter. This parameter is enabled by default.

About this task

This parameter is only effective when you have assigned an export policy directly to one or more qtrees on the storage virtual machine (SVM).

Steps

1. Set the privilege level to advanced:

```
set -privilege advanced
```

2. Perform one of the following actions:

| If you want qtree ID validation to be... | Enter the following command... |
|--|---|
| Enabled | <pre>vserver nfs modify -vserver vserver_name -validate-qtree-export enabled</pre> |
| Disabled | <pre>vserver nfs modify -vserver vserver_name -validate-qtree-export disabled</pre> |

3. Return to the admin privilege level:

```
set -privilege admin
```

Export policy restrictions and nested junctions for FlexVol volumes

If you configured export policies to set a less restrictive policy on a nested junction but a more restrictive policy on a higher level junction, access to the lower level junction might fail.

You should ensure that higher level junctions have less restrictive export policies than lower level junctions.

Using Kerberos with NFS for strong security

ONTAP support for Kerberos

Kerberos provides strong secure authentication for client/server applications. Authentication provides verification of user and process identities to a server. In the ONTAP environment, Kerberos provides authentication between storage virtual machines (SVMs) and NFS clients.

In ONTAP 9, the following Kerberos functionality is supported:

- Kerberos 5 authentication with integrity checking (krb5i)

Krb5i uses checksums to verify the integrity of each NFS message transferred between client and server. This is useful both for security reasons (for example, to ensure that data has not been tampered with) and for data integrity reasons (for example, to prevent data corruption when using NFS over unreliable networks).

- Kerberos 5 authentication with privacy checking (krb5p)

Krb5p uses checksums to encrypt all the traffic between client and the server. This is more secure and also incurs more load.

- 128-bit and 256-bit AES encryption

Advanced Encryption Standard (AES) is an encryption algorithm for securing electronic data. ONTAP now supports AES with 128-bit keys (AES-128) and AES with 256-bit keys (AES-256) encryption for Kerberos for stronger security.

- SVM-level Kerberos realm configurations

SVM administrators can now create Kerberos realm configurations at the SVM level. This means that SVM administrators no longer have to rely on the cluster administrator for Kerberos realm configuration and can create individual Kerberos realm configurations in a multi-tenancy environment.

Requirements for configuring Kerberos with NFS

Before you configure Kerberos with NFS on your system, you must verify that certain items in your network and storage environment are properly configured.



The steps to configure your environment depend on what version and type of client operating system, domain controller, Kerberos, DNS, etc., that you are using. Documenting all these variables is beyond the scope of this document. For more information, see the respective documentation for each component.

For a detailed example of how to set up ONTAP and Kerberos 5 with NFSv3 and NFSv4 in an environment using Windows Server 2008 R2 Active Directory and Linux hosts, see technical report 4073.

The following items should be configured first:

Network environment requirements

- Kerberos

You must have a working Kerberos setup with a key distribution center (KDC), such as Windows Active Directory based Kerberos or MIT Kerberos.

NFS servers must use `nfs` as the primary component of their machine principal.

- Directory service

You must use a secure directory service in your environment, such as Active Directory or OpenLDAP, that is configured to use LDAP over SSL/TLS.

- NTP

You must have a working time server running NTP. This is necessary to prevent Kerberos authentication failure due to time skew.

- Domain name resolution (DNS)

Each UNIX client and each SVM LIF must have a proper service record (SRV) registered with the KDC under forward and reverse lookup zones. All participants must be properly resolvable via DNS.

- User accounts

Each client must have a user account in the Kerberos realm. NFS servers must use “nfs” as the primary component of their machine principal.

NFS client requirements

- NFS

Each client must be properly configured to communicate over the network using NFSv3 or NFSv4.

Clients must support RFC1964 and RFC2203.

- Kerberos

Each client must be properly configured to use Kerberos authentication, including the following details:

- Encryption for TGS communication is enabled.

AES-256 for strongest security.

- The most secure encryption type for TGT communication is enabled.
- The Kerberos realm and domain are configured correctly.
- GSS is enabled.

When using machine credentials:

- Do not run `gssd` with the `-n` parameter.
- Do not run `kinit` as the root user.

- Each client must use the most recent and updated operating system version.

This provides the best compatibility and reliability for AES encryption with Kerberos.

- DNS

Each client must be properly configured to use DNS for correct name resolution.

- NTP

Each client must be synchronizing with the NTP server.

- Host and domain information

Each client's `/etc/hosts` and `/etc/resolv.conf` files must contain the correct host name and DNS information, respectively.

- Keytab files

Each client must have a keytab file from the KDC. The realm must be in uppercase letters. The encryption type must be AES-256 for strongest security.

- Optional: For best performance, clients benefit from having at least two network interfaces: one for communicating with the local area network and one for communicating with the storage network.

Storage system requirements

- NFS license

The storage system must have a valid NFS license installed.

- CIFS license

The CIFS license is optional. It is only required for checking Windows credentials when using multiprotocol name mapping. It is not required in a strict UNIX-only environment.

- SVM

You must have at least one SVM configured on the system.

- DNS on the SVM

You must have configured DNS on each SVM.

- NFS server

You must have configured NFS on the SVM.

- AES encryption

For strongest security, you must configure the NFS server to allow only AES-256 encryption for Kerberos.

- SMB server

If you are running a multiprotocol environment, you must have configured SMB on the SVM. The SMB server is required for multiprotocol name mapping.

- Volumes

You must have a root volume and at least one data volume configured for use by the SVM.

- Root volume

The root volume of the SVM must have the following configuration:

| Name | Setting |
|------------------|--------------|
| Security style | UNIX |
| UID | root or ID 0 |
| GID | root or ID 0 |
| UNIX permissions | 777 |

In contrast to the root volume, data volumes can have either security style.

- UNIX groups

The SVM must have the following UNIX groups configured:

| Group name | Group ID |
|------------|--|
| daemon | 1 |
| root | 0 |
| pcuser | 65534 (created automatically by ONTAP when you create the SVM) |

- UNIX users

The SVM must have the following UNIX users configured:

| User name | User ID | Primary group ID | Comment |
|-----------|---------|------------------|---|
| nfs | 500 | 0 | Required for GSS INIT phase The first component of the NFS client user SPN is used as the user. |

| User name | User ID | Primary group ID | Comment |
|-----------|---------|------------------|--|
| pcuser | 65534 | 65534 | Required for NFS and CIFS multiprotocol use Created and added to the pcuser group automatically by ONTAP when you create the SVM. |
| root | 0 | 0 | Required for mounting |

The nfs user is not required if a Kerberos-UNIX name mapping exists for the SPN of the NFS client user.

- Export policies and rules

You must have configured export policies with the necessary export rules for the root and data volumes and qtrees. If all volumes of the SVM are accessed over Kerberos, you can set the export rule options `-rorule`, `-rwrule`, and `-superuser` for the root volume to `krb5`, `krb5i`, or `krb5p`.

- Kerberos-UNIX name mapping

If you want the user identified by the NFS client user SPN to have root permissions, you must create a name mapping to root.

Related information

[NetApp Technical Report 4073: Secure Unified Authentication](#)

[NetApp Interoperability Matrix Tool](#)

[System administration](#)

[Logical storage management](#)

Specify the user ID domain for NFSv4

To specify the user ID domain, you can set the `-v4-id-domain` option.

About this task

By default, ONTAP uses the NIS domain for NFSv4 user ID mapping, if one is set. If an NIS domain is not set, the DNS domain is used. You might need to set the user ID domain if, for example, you have multiple user ID domains. The domain name must match the domain configuration on the domain controller. It is not required for NFSv3.

Step

1. Enter the following command:

```
vserver nfs modify -vserver vservice_name -v4-id-domain NIS_domain_name
```

Configure name services

How ONTAP name service switch configuration works

ONTAP stores name service configuration information in a table that is the equivalent of the `/etc/nsswitch.conf` file on UNIX systems. You must understand the function of the table and how ONTAP uses it so that you can configure it appropriately for your environment.

The ONTAP name service switch table determines which name service sources ONTAP consults in which order to retrieve information for a certain type of name service information. ONTAP maintains a separate name service switch table for each SVM.

Database types

The table stores a separate name service list for each of the following database types:

| Database type | Defines name service sources for... | Valid sources are... |
|---------------|---------------------------------------|----------------------|
| hosts | Converting host names to IP addresses | files, dns |
| group | Looking up user group information | files, nis, ldap |
| passwd | Looking up user information | files, nis, ldap |
| netgroup | Looking up netgroup information | files, nis, ldap |
| namemap | Mapping user names | files, ldap |

Source types

The sources specify which name service source to use for retrieving the appropriate information.

| Specify source type... | To look up information in... | Managed by the command families... |
|------------------------|------------------------------|--|
| files | Local source files | <code>vserver services name-service unix-user vserver services name-service unix-group</code> <code>vserver services name-service netgroup</code> <code>vserver services name-service dns hosts</code> |

| Specify source type... | To look up information in... | Managed by the command families... |
|------------------------|--|---|
| nis | External NIS servers as specified in the NIS domain configuration of the SVM | <code>vserver services name-service nis-domain</code> |
| ldap | External LDAP servers as specified in the LDAP client configuration of the SVM | <code>vserver services name-service ldap</code> |
| dns | External DNS servers as specified in the DNS configuration of the SVM | <code>vserver services name-service dns</code> |

Even if you plan to use NIS or LDAP for both data access and SVM administration authentication, you should still include `files` and configure local users as a fallback in case NIS or LDAP authentication fails.

Protocols used to access external sources

To access the servers for external sources, ONTAP uses the following protocols:

| External name service source | Protocol used for access |
|------------------------------|--------------------------|
| NIS | UDP |
| DNS | UDP |
| LDAP | TCP |

Example

The following example displays the name service switch configuration for the SVM `svm_1`:

```
cluster1::*> vserver services name-service ns-switch show -vserver svm_1
```

| Vserver | Database | Source Order |
|---------|----------|---------------|
| ----- | ----- | ----- |
| svm_1 | hosts | files, dns |
| svm_1 | group | files |
| svm_1 | passwd | files |
| svm_1 | netgroup | nis, files |

To look up IP addresses for hosts, ONTAP first consults local source files. If the query does not return any results, DNS servers are checked next.

To look up user or group information, ONTAP consults only local sources files. If the query does not return any results, the lookup fails.

To look up netgroup information, ONTAP first consults external NIS servers. If the query does not return any results, the local netgroup file is checked next.

There are no name service entries for name mapping in the table for the SVM svm_1. Therefore, ONTAP consults only local source files by default.

Related information

[NetApp Technical Report 4668: Name Services Best Practices Guide](#)

Use LDAP

LDAP Overview

An LDAP (Lightweight Directory Access Protocol) server enables you to centrally maintain user information. If you store your user database on an LDAP server in your environment, you can configure your storage system to look up user information in your existing LDAP database.

- Before configuring LDAP for ONTAP, you should verify that your site deployment meets best practices for LDAP server and client configuration. In particular, the following conditions must be met:
 - The domain name of the LDAP server must match the entry on the LDAP client.
 - The LDAP user password hash types supported by the LDAP server must include those supported by ONTAP:
 - CRYPT (all types) and SHA-1 (SHA, SSHA).
 - Beginning with ONTAP 9.8, SHA-2 hashes (SHA-256, SSH-384, SHA-512, SSHA-256, SSHA-384, and SSHA-512) are also supported.
 - If the LDAP server requires session security measures, you must configure them in the LDAP client.

The following session security options are available:

- LDAP signing (provides data integrity checking) and LDAP signing and sealing (provides data integrity checking and encryption)
- START TLS
- LDAPS (LDAP over TLS or SSL)
- To enable signed and sealed LDAP queries, the following services must be configured:
 - LDAP servers must support the GSSAPI (Kerberos) SASL mechanism.
 - LDAP servers must have DNS A/AAAA records as well as PTR records set up on the DNS server.
 - Kerberos servers must have SRV records present on the DNS server.
- To enable START TLS or LDAPS, the following points should be considered.
 - It is a NetApp best practice to use Start TLS rather than LDAPS.
 - If LDAPS is used, the LDAP server must be enabled for TLS or for SSL in ONTAP 9.5 and later. SSL is not supported in ONTAP 9.0-9.4.
 - A certificate server must already be configured in the domain.

- To enable LDAP referral chasing (in ONTAP 9.5 and later), the following conditions must be satisfied:
 - Both domains should be configured with one of the following trust relationships:
 - Two-way
 - One-way, where the primary trusts the referral domain
 - Parent-child
 - DNS must be configured to resolve all referred server names.
 - Domain passwords should be same to authenticate when `--bind-as-cifs-server` set to true.

The following configurations are not supported with LDAP referral chasing.



- For all ONTAP versions:
 - LDAP clients on an admin SVM
- For ONTAP 9.8 and earlier (they are supported in 9.9.1 and later):
 - LDAP signing and sealing (the `-session-security` option)
 - Encrypted TLS connections (the `-use-start-tls` option)
 - Communications over LDAPS port 636 (the `-use-ldaps-for-ad-ldap` option)

- You must enter an LDAP schema when configuring the LDAP client on the SVM.

In most cases, one of the default ONTAP schemas will be appropriate. However, if the LDAP schema in your environment differs from these, you must create a new LDAP client schema for ONTAP before creating the LDAP client. Consult with your LDAP administrator about requirements for your environment.

- Using LDAP for host name resolution is not supported.

For additional information, see [NetApp Technical Report 4835: How to Configure LDAP in ONTAP](#).

LDAP signing and sealing concepts

Beginning with ONTAP 9, you can configure signing and sealing to enable LDAP session security on queries to an Active Directory (AD) server. You must configure the NFS server security settings on the storage virtual machine (SVM) to correspond to those on the LDAP server.

Signing confirms the integrity of the LDAP payload data using secret key technology. Sealing encrypts the LDAP payload data to avoid transmitting sensitive information in clear text. An *LDAP Security Level* option indicates whether the LDAP traffic needs to be signed, signed and sealed, or neither. The default is `none`. `test`

LDAP signing and sealing on SMB traffic is enabled on the SVM with the `-session-security-for-ad-ldap` option to the `vserver cifs security modify` command.

LDAPS concepts

You must understand certain terms and concepts about how ONTAP secures LDAP communication. ONTAP can use START TLS or LDAPS for setting up authenticated sessions between Active Directory-integrated LDAP servers or UNIX-based LDAP servers.

Terminology

There are certain terms that you should understand about how ONTAP uses LDAPS to secure LDAP communication.

- **LDAP**

(Lightweight Directory Access Protocol) A protocol for accessing and managing information directories. LDAP is used as an information directory for storing objects such as users, groups, and netgroups. LDAP also provides directory services that manage these objects and fulfill LDAP requests from LDAP clients.

- **SSL**

(Secure Sockets Layer) A protocol developed for sending information securely over the Internet. It has been deprecated in favor of TLS. SSL is not supported in ONTAP 9.0-9.4.

- **TLS**

(Transport Layer Security) An IETF standards track protocol that is based on the earlier SSL specifications. It is the successor to SSL.

- **LDAPS (LDAP over SSL or TLS)**

A protocol that uses TLS or SSL to secure communication between LDAP clients and LDAP servers. The terms *LDAP over SSL* and *LDAP over TLS* are sometimes used interchangeably; TLS is supported by ONTAP 9 and later, SSL is supported by ONTAP 9.5 and later.

- In ONTAP 9.5-9.8, LDAPS can only be enabled on port 636. To do so, use the `-use-ldaps-for-ad -ldap` parameter with the `vserver cifs security modify` command.
- Beginning with ONTAP 9.9.1, LDAPS can be enabled on any port, although port 636 remains the default. To do so, set the `-ldaps-enabled` parameter to `true` and specify the desired `-port` parameter. For more information, see the `vserver services name-service ldap client create` man page



It is a NetApp best practice to use Start TLS rather than LDAPS.

- **Start TLS**

(Also known as *start_tls*, *STARTTLS*, and *StartTLS*) A mechanism to provide secure communication by using the TLS protocols.

ONTAP uses STARTTLS for securing LDAP communication, and uses the default LDAP port (389) to communicate with the LDAP server. The LDAP server must be configured to allow connections over LDAP port 389; otherwise, LDAP TLS connections from the SVM to the LDAP server fail.

How ONTAP uses LDAPS

ONTAP supports TLS server authentication, which enables the SVM LDAP client to confirm the LDAP server's identity during the bind operation. TLS-enabled LDAP clients can use standard techniques of public-key cryptography to check that a server's certificate and public ID are valid and have been issued by a certificate authority (CA) listed in the client's list of trusted CAs.

LDAP supports STARTTLS to encrypt communications using TLS. STARTTLS begins as a plaintext connection over the standard LDAP port (389), and that connection is then upgraded to TLS.

ONTAP supports the following:

- LDAPS for SMB-related traffic between the Active Directory-integrated LDAP servers and the SVM
- LDAPS for LDAP traffic for name mapping and other UNIX information

Either Active Directory-integrated LDAP servers or UNIX-based LDAP servers can be used to store information for LDAP name mapping and other UNIX information, such as users, groups, and netgroups.

- Self-signed root CA certificates

When using an Active-Directory integrated LDAP, the self-signed root certificate is generated when the Windows Server Certificate Service is installed in the domain. When using an UNIX-based LDAP server for LDAP name mapping, the self-signed root certificate is generated and saved by using means appropriate to that LDAP application.

By default, LDAPS is disabled.

Enable LDAP RFC2307bis support

If you want to use LDAP and require the additional capability to use nested group memberships, you can configure ONTAP to enable LDAP RFC2307bis support.

What you'll need

You must have created a copy of one of the default LDAP client schemas that you want to use.

About this task

In LDAP client schemas, group objects use the `memberUid` attribute. This attribute can contain multiple values and lists the names of the users that belong to that group. In RFC2307bis enabled LDAP client schemas, group objects use the `uniqueMember` attribute. This attribute can contain the full distinguished name (DN) of another object in the LDAP directory. This enables you to use nested groups because groups can have other groups as members.

The user should not be a member of more than 256 groups including nested groups. ONTAP ignores any groups over the 256 group limit.

By default, RFC2307bis support is disabled.



RFC2307bis support is enabled automatically in ONTAP when an LDAP client is created with the MS-AD-BIS schema.

For additional information, see [NetApp Technical Report 4835: How to Configure LDAP in ONTAP](#).

Steps

1. Set the privilege level to advanced:

```
set -privilege advanced
```

2. Modify the copied RFC2307 LDAP client schema to enable RFC2307bis support:

```
vserver services name-service ldap client schema modify -vserver vserver_name  
-schema schema-name -enable-rfc2307bis true
```

3. Modify the schema to match the object class supported in the LDAP server:

```
vserver services name-service ldap client schema modify -vserver vserver-name  
-schema schema_name -group-of-unique-names-object-class object_class
```

4. Modify the schema to match the attribute name supported in the LDAP server:

```
vserver services name-service ldap client schema modify -vserver vserver-name  
-schema schema_name -unique-member-attribute attribute_name
```

5. Return to the admin privilege level:

```
set -privilege admin
```

Configuration options for LDAP directory searches

You can optimize LDAP directory searches, including user, group, and netgroup information, by configuring the ONTAP LDAP client to connect to LDAP servers in the most appropriate way for your environment. You need to understand when the default LDAP base and scope search values suffice and which parameters to specify when custom values are more appropriate.

LDAP client search options for user, group, and netgroup information can help avoid failed LDAP queries, and therefore failed client access to storage systems. They also help ensure that the searches are as efficient as possible to avoid client performance issues.

Default base and scope search values

The LDAP base is the default base DN that the LDAP client uses to perform LDAP queries. All searches, including user, group, and netgroup searches, are done using the base DN. This option is appropriate when your LDAP directory is relatively small and all relevant entries are located in the same DN.

If you do not specify a custom base DN, the default is `root`. This means that each query searches the entire directory. Although this maximizes the chances of success of the LDAP query, it can be inefficient and result in significantly decreased performance with large LDAP directories.

The LDAP base scope is the default search scope that the LDAP client uses to perform LDAP queries. All searches, including user, group, and netgroup searches, are done using the base scope. It determines whether the LDAP query searches only the named entry, entries one level below the DN, or the entire subtree below the DN.

If you do not specify a custom base scope, the default is `subtree`. This means that each query searches the entire subtree below the DN. Although this maximizes the chances of success of the LDAP query, it can be inefficient and result in significantly decreased performance with large LDAP directories.

Custom base and scope search values

Optionally, you can specify separate base and scope values for user, group, and netgroup searches. Limiting the search base and scope of queries this way can significantly improve performance because it limits the search to a smaller subsection of the LDAP directory.

If you specify custom base and scope values, they override the general default search base and scope for user, group, and netgroup searches. The parameters to specify custom base and scope values are available at

the advanced privilege level.

| LDAP client parameter... | Specifies custom... |
|------------------------------|---|
| <code>-base-dn</code> | Base DN for all LDAP searchesMultiple values can be entered if needed (for example, if LDAP referral chasing is enabled in ONTAP 9.5 and later releases). |
| <code>-base-scope</code> | Base scope for all LDAP searches |
| <code>-user-dn</code> | Base DN for all LDAP user searchesThis parameter also applies to user name-mapping searches. |
| <code>-user-scope</code> | Base scope for all LDAP user searches This parameter also applies to user name-mapping searches. |
| <code>-group-dn</code> | Base DN for all LDAP group searches |
| <code>-group-scope</code> | Base scope for all LDAP group searches |
| <code>-netgroup-dn</code> | Base DN for all LDAP netgroup searches |
| <code>-netgroup-scope</code> | Base scope for all LDAP netgroup searches |

Multiple custom base DN values

If your LDAP directory structure is more complex, it might be necessary for you to specify multiple base DN values to search multiple parts of your LDAP directory for certain information. You can specify multiple DN values for the user, group, and netgroup DN parameters by separating them with a semicolon (;) and enclosing the entire DN search list with double quotes ("). If a DN contains a semicolon, you must add an escape character (\) immediately before the semicolon in the DN.

Note that the scope applies to the entire list of DN values specified for the corresponding parameter. For example, if you specify a list of three different user DN values and subtree for the user scope, then LDAP user searches search the entire subtree for each of the three specified DN values.

Beginning with ONTAP 9.5, you can also specify LDAP *referral chasing*, which allows the ONTAP LDAP client to refer look-up requests to other LDAP servers if an LDAP referral response is not returned by the primary LDAP server. The client uses that referral data to retrieve the target object from the server described in the referral data. To search for objects present in the referred LDAP servers, the base-dn of the referred objects can be added to the base-dn as part of LDAP client configuration. However, referred objects are only looked up when referral chasing is enabled (using the `-referral-enabled true` option) during LDAP client creation or modification.

Improve performance of LDAP directory netgroup-by-host searches

If your LDAP environment is configured to allow netgroup-by-host searches, you can configure ONTAP to take advantage of this and perform netgroup-by-host searches. This can significantly speed up netgroup searches and reduce possible NFS client access

issues due to latency during netgroup searches.

What you'll need

Your LDAP directory must contain a `netgroup.byhost` map.

Your DNS servers should contain both forward (A) and reverse (PTR) lookup records for NFS clients.

When you specify IPv6 addresses in netgroups, you must always shorten and compress each address as specified in RFC 5952.

About this task

NIS servers store netgroup information in three separate maps called `netgroup`, `netgroup.byuser`, and `netgroup.byhost`. The purpose of the `netgroup.byuser` and `netgroup.byhost` maps is to speed up netgroup searches. ONTAP can perform netgroup-by-host searches on NIS servers for improved mount response times.

By default, LDAP directories do not have such a `netgroup.byhost` map like NIS servers. It is possible, though, with the help of third-party tools, to import a NIS `netgroup.byhost` map into LDAP directories to enable fast netgroup-by-host searches. If you have configured your LDAP environment to allow netgroup-by-host searches, you can configure the ONTAP LDAP client with the `netgroup.byhost` map name, DN, and search scope for faster netgroup-by-host searches.

Receiving the results for netgroup-by-host searches faster enables ONTAP to process export rules faster when NFS clients request access to exports. This reduces the chance of delayed access due to netgroup search latency issues.

Steps

1. Obtain the exact full distinguished name of the NIS `netgroup.byhost` map you imported into your LDAP directory.

The map DN can vary depending on the third-party tool you used for import. For best performance, you should specify the exact map DN.

2. Set the privilege level to advanced: `set -privilege advanced`
3. Enable netgroup-by-host searches in the LDAP client configuration of the storage virtual machine (SVM):

```
vserver services name-service ldap client modify -vserver vserver_name -client  
-config config_name -is-netgroup-byhost-enabled true -netgroup-byhost-dn  
netgroup-by-host_map_distinguished_name -netgroup-byhost-scope netgroup-by-  
host_search_scope
```

`-is-netgroup-byhost-enabled {true|false}` enables or disables netgroup-by-host search for LDAP directories. The default is `false`.

`-netgroup-byhost-dn netgroup-by-host_map_distinguished_name` specifies the distinguished name of the `netgroup.byhost` map in the LDAP directory. It overrides the base DN for netgroup-by-host searches. If you do not specify this parameter, ONTAP uses the base DN instead.

`-netgroup-byhost-scope {base|onelevel|subtree}` specifies the search scope for netgroup-by-host searches. If you do not specify this parameter, the default is `subtree`.

If the LDAP client configuration does not exist yet, you can enable netgroup-by-host searches by specifying these parameters when creating a new LDAP client configuration using the `vserver services name-`

service ldap client create command.



Beginning with ONTAP 9.2, the field `-ldap-servers` replaces the field `-servers`. This new field can take either a hostname or an IP address for the LDAP server.

4. Return to the admin privilege level: `set -privilege admin`

Example

The following command modifies the existing LDAP client configuration named "ldap_corp" to enable netgroup-by-host searches using the `netgroup.byhost` map named "nisMapName="netgroup.byhost",dc=corp,dc=example,dc=com" and the default search scope `subtree`:

```
cluster1::*> vserver services name-service ldap client modify -vserver vs1
-client-config ldap_corp -is-netgroup-byhost-enabled true -netgroup-byhost
-dn nisMapName="netgroup.byhost",dc=corp,dc=example,dc=com
```

After you finish

The `netgroup.byhost` and `netgroup` maps in the directory must be kept in sync at all times to avoid client access issues.

Related information

[IETF RFC 5952: A Recommendation for IPv6 Address Text Representation](#)

Display LDAP statistics

Beginning with ONTAP 9.2, you can display LDAP statistics for storage virtual machines (SVMs) on a storage system to monitor the performance and diagnose issues.

What you'll need

- You must have configured an LDAP client on the SVM.
- You must have identified LDAP objects from which you can view data.

Step

1. View the performance data for counter objects:

```
statistics show
```

Examples

The following example shows the performance data for object `secd_external_service_op`:

```
cluster::*> statistics show -vserver vserverName -object
secd_external_service_op -instance "vserverName:LDAP (NIS & Name
Mapping):GetUserInfoFromName:1.1.1.1"
```

Object: secd_external_service_op

Instance: vserverName:LDAP (NIS & Name

Mapping):GetUserInfoFromName:1.1.1.1

Start-time: 4/13/2016 22:15:38

End-time: 4/13/2016 22:15:38

Scope: vserverName

| Counter | Value |
|--------------------------|--|
| instance_name | vserverName:LDAP (NIS & Name Mapping):GetUserInfoFromName: 1.1.1.1 |
| last_modified_time | 1460610787 |
| node_name | nodeName |
| num_not_found_responses | 1 |
| num_request_failures | 1 |
| num_requests_sent | 1 |
| num_responses_received | 1 |
| num_successful_responses | 0 |
| num_timeouts | 0 |
| operation | GetUserInfoFromName |
| process_name | secd |
| request_latency | 52131us |

Configure name mappings

Configure name mappings overview

ONTAP uses name mapping to map SMB identities to UNIX identities, Kerberos identities to UNIX identities, and UNIX identities to SMB identities. It needs this information to obtain user credentials and provide proper file access regardless of whether they are connecting from an NFS client or a SMB client.

There are two exceptions where you do not have to use name mapping:

- You configure a pure UNIX environment and do not plan to use SMB access or NTFS security style on volumes.
- You configure the default user to be used instead.

In this scenario, name mapping is not required because instead of mapping every individual client credential all client credentials are mapped to the same default user.

Note that you can use name mapping only for users, not for groups.

However, you can map a group of individual users to a specific user. For example, you can map all AD users that start or end with the word SALES to a specific UNIX user and to the user's UID.

How name mapping works

When ONTAP has to map credentials for a user, it first checks the local name mapping database and LDAP server for an existing mapping. Whether it checks one or both and in which order is determined by the name service configuration of the SVM.

- For Windows to UNIX mapping

If no mapping is found, ONTAP checks whether the lowercase Windows user name is a valid user name in the UNIX domain. If this does not work, it uses the default UNIX user provided that it is configured. If the default UNIX user is not configured and ONTAP cannot obtain a mapping this way either, mapping fails and an error is returned.

- For UNIX to Windows mapping

If no mapping is found, ONTAP tries to find a Windows account that matches the UNIX name in the SMB domain. If this does not work, it uses the default SMB user, provided that it is configured. If the default SMB user is not configured and ONTAP cannot obtain a mapping this way either, mapping fails and an error is returned.

Machine accounts are mapped to the specified default UNIX user by default. If no default UNIX user is specified, machine account mappings fail.

- Beginning with ONTAP 9.5, you can map machine accounts to users other than the default UNIX user.
- In ONTAP 9.4 and earlier, you cannot map machine accounts to other users.

Even if name mappings for machine accounts are defined, the mappings are ignored.

Multidomain searches for UNIX user to Windows user name mappings

ONTAP supports multidomain searches when mapping UNIX users to Windows users. All discovered trusted domains are searched for matches to the replacement pattern until a matching result is returned. Alternatively, you can configure a list of preferred trusted domains, which is used instead of the discovered trusted domain list and is searched in order until a matching result is returned.

How domain trusts affect UNIX user to Windows user name mapping searches

To understand how multidomain user name mapping works, you must understand how domain trusts work with ONTAP. Active Directory trust relationships with the SMB server's home domain can be a bidirectional trust or can be one of two types of unidirectional trusts, either an inbound trust or an outbound trust. The home domain is the domain to which the SMB server on the SVM belongs.

- *Bidirectional trust*

With bidirectional trusts, both domains trust each other. If the SMB server's home domain has a bidirectional trust with another domain, the home domain can authenticate and authorize a user belonging to the trusted domain and vice versa.

UNIX user to Windows user name mapping searches can be performed only on domains with bidirectional trusts between the home domain and the other domain.

- *Outbound trust*

With an outbound trust, the home domain trusts the other domain. In this case, the home domain can authenticate and authorize a user belonging to the outbound trusted domain.

A domain with an outbound trust with the home domain is *not* searched when performing UNIX user to Windows user name mapping searches.

- *Inbound trust*

With an inbound trust, the other domain trusts the SMB server's home domain. In this case, the home domain cannot authenticate or authorize a user belonging to the inbound trusted domain.

A domain with an inbound trust with the home domain is *not* searched when performing UNIX user to Windows user name mapping searches.

How wildcards (*) are used to configure multidomain searches for name mapping

Multidomain name mapping searches are facilitated by the use of wildcards in the domain section of the Windows user name. The following table illustrates how to use wildcards in the domain part of a name mapping entry to enable multidomain searches:

| Pattern | Replacement | Result |
|---------|------------------|--|
| root | *\\administrator | The UNIX user “root” is mapped to the user named “administrator”. All trusted domains are searched in order until the first matching user named “administrator” is found. |
| * | ** | <div>Valid UNIX users are mapped to the corresponding Windows users. All trusted domains are searched in order until the first matching user with that name is found.</div> <div> The pattern ** is only valid for name mapping from UNIX to Windows, not the other way around.</div> |

How multidomain name searches are performed

You can choose one of two methods for determining the list of trusted domains used for multidomain name searches:

- Use the automatically discovered bidirectional trust list compiled by ONTAP
- Use the preferred trusted domain list that you compile

If a UNIX user is mapped to a Windows user with a wildcard used for the domain section of the user name, the Windows user is looked up in all the trusted domains as follows:

- If a preferred trusted-domain list is configured, the mapped Windows user is looked up in this search list only, in order.
- If a preferred list of trusted domains is not configured, then the Windows user is looked up in all the bidirectional trusted domains of the home domain.
- If there are no bidirectionally trusted domains for the home domain, the user is looked up in the home domain.

If a UNIX user is mapped to a Windows user without a domain section in the user name, the Windows user is looked up in the home domain.

Name mapping conversion rules

An ONTAP system keeps a set of conversion rules for each SVM. Each rule consists of two pieces: a *pattern* and a *replacement*. Conversions start at the beginning of the appropriate list and perform a substitution based on the first matching rule. The pattern is a UNIX-style regular expression. The replacement is a string containing escape sequences representing subexpressions from the pattern, as in the UNIX `sed` program.

Create a name mapping

You can use the `vserver name-mapping create` command to create a name mapping. You use name mappings to enable Windows users to access UNIX security style volumes and the reverse.

About this task

For each SVM, ONTAP supports up to 12,500 name mappings for each direction.

Step

1. Create a name mapping:

```
vserver name-mapping create -vserver vserver_name -direction {krb-unix|win-unix|unix-win} -position integer -pattern text -replacement text
```



The `-pattern` and `-replacement` statements can be formulated as regular expressions. You can also use the `-replacement` statement to explicitly deny a mapping to the user by using the null replacement string " " (the space character). See the `vserver name-mapping create` man page for details.

When Windows-to-UNIX mappings are created, any SMB clients that have open connections to the ONTAP system at the time the new mappings are created must log out and log back in to see the new mappings.

Examples

The following command creates a name mapping on the SVM named `vs1`. The mapping is a mapping from UNIX to Windows at position 1 in the priority list. The mapping maps the UNIX user `johnd` to the Windows user `ENG\JohnDoe`.

```
vs1::> vsserver name-mapping create -vserver vs1 -direction unix-win
-position 1 -pattern johnd
-replacement "ENG\\JohnDoe"
```

The following command creates another name mapping on the SVM named vs1. The mapping is a mapping from Windows to UNIX at position 1 in the priority list. Here the pattern and replacement include regular expressions. The mapping maps every CIFS user in the domain ENG to users in the LDAP domain associated with the SVM.

```
vs1::> vsserver name-mapping create -vserver vs1 -direction win-unix
-position 1 -pattern "ENG\\(.+)"
-replacement "\\1"
```

The following command creates another name mapping on the SVM named vs1. Here the pattern includes "\$" as an element in the Windows user name that must be escaped. The mapping maps the windows user ENG\john\$ops to UNIX user john_ops.

```
vs1::> vsserver name-mapping create -direction win-unix -position 1
-pattern ENG\\john$ops
-replacement john_ops
```

Configure the default user

You can configure a default user to use if all other mapping attempts fail for a user, or if you do not want to map individual users between UNIX and Windows. Alternatively, if you want authentication of non-mapped users to fail, you should not configure a default user.

About this task

For CIFS authentication, if you do not want to map each Windows user to an individual UNIX user, you can instead specify a default UNIX user.

For NFS authentication, if you do not want to map each UNIX user to an individual Windows user, you can instead specify a default Windows user.


Step

1. Perform one of the following actions:

| If you want to... | Enter the following command... |
|------------------------------------|--|
| Configure the default UNIX user | <code>vsserver cifs options modify -default-unix-user user_name</code> |
| Configure the default Windows user | <code>vsserver nfs modify -default-win-user user_name</code> |

Commands for managing name mappings

There are specific ONTAP commands for managing name mappings.

| If you want to... | Use this command... |
|--|---|
| Create a name mapping | <code>vserver name-mapping create</code> |
| Insert a name mapping at a specific position | <code>vserver name-mapping insert</code> |
| Display name mappings | <code>vserver name-mapping show</code> |
| Exchange the position of two name mappings | <code>vserver name-mapping swap</code> |
| <div> A swap is not allowed when name-mapping is configured with an ip-qualifier entry.</div> | |
| Modify a name mapping | <code>vserver name-mapping modify</code> |
| Delete a name mapping | <code>vserver name-mapping delete</code> |
| Validate the correct name mapping | <code>vserver security file-directory show-effective-permissions -vserver vs1 -win-user-name user1 -path / -share-name sh1</code> |

See the man page for each command for more information.

Enable access for Windows NFS clients

ONTAP supports file access from Windows NFSv3 clients. This means that clients running Windows operating systems with NFSv3 support can now access files on NFSv3 exports on the cluster. To successfully use this functionality, you must properly configure the storage virtual machine (SVM) and be aware of certain requirements and limitations.

What you'll need

NFSv3 must be enabled on the SVM.

About this task

By default, Windows NFSv3 client support is disabled.

Windows NFSv3 clients do not support the network status monitor (NSM) protocol. As a result, Windows NFSv3 client sessions might experience disruptions during storage failover and volume move operations.

Steps

1. Enable Windows NFSv3 client support:

```
vserver nfs modify -vserver vserver_name -v3-ms-dos-client enabled
```

2. On all SVMs that support Windows NFSv3 clients, disable the `-enable-ejukebox` and `-v3-connection-drop` parameters: `vserver nfs modify -vserver vserver_name -enable-ejukebox false -v3-connection-drop disabled`

Windows NFSv3 clients can now mount exports on the storage system.

3. Ensure that each Windows NFSv3 client uses hard mounts by specifying the `-o mtype=hard` option.

This is required to ensure reliable mounts.

```
mount -o mtype=hard \\10.53.33.10\vol\vol1 z:\
```

Enable the display of NFS exports on NFS clients

NFS clients can use the `showmount -e` command to see a list of exports available from an ONTAP NFS server. This can help users identify the file system they want to mount.

Beginning with ONTAP 9.2, ONTAP allows NFS clients to view the export list by default. In earlier releases, the `showmount` option of the `vserver nfs modify` command must be enabled explicitly. For viewing the export list, NFSv3 should be enabled on the SVM.

Example

The following command shows the `showmount` feature on the SVM named `vs1`:

```
cluster1 : : > vserver nfs show -vserver vs1 -fields showmount
vserver showmount
-----
vs1      enabled
```

The following command executed on an NFS client displays the list of exports on an NFS server with the IP address 10.63.21.9:

```
showmount -e 10.63.21.9
Export list for 10.63.21.9:
/unix      (everyone)
/unix/unix1 (everyone)
/unix/unix2 (everyone)
/          (everyone)
```

Copyright Information

Copyright © 2022 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.

RESTRICTED RIGHTS LEGEND: Use, duplication, or disclosure by the government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.277-7103 (October 1988) and FAR 52-227-19 (June 1987).

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.