



Technical FAQ

Multiprotocol Technical FAQ

Justin Parisi, NetApp
July 2015

TABLE OF CONTENTS

1	Purpose	3
1.1	Definitions	3
	What does multiprotocol mean?	3
	What is security style?	3
	What is name mapping?	3
	What is WCC?	3
	What is authentication?	3
	What is authorization?	3
	What is the user space?	3
	What is the security daemon (SecD)?	4
	What is the NAS module?	4
	What is the replicated database?	4
	What is an ACL?	4
	What is a CIFS share?	4
	What is an NFS export?	4
2	Decision Points	4
	When is it necessary to use multiprotocol capabilities?	4
3	7-Mode	5
3.1	How It Works	5
	NFS Access of NTFS Security Style: 7-Mode	5
	CIFS Access of UNIX Security Style: 7-Mode	8
4	Clustered Data ONTAP	12
4.1	How It Works	12
	NFS Access of NTFS Security Style: Clustered Data ONTAP	13
	CIFS Access of UNIX Security Style: Clustered Data ONTAP	14
	References	17
	Version History	17
	Acknowledgements	17

No table of figures entries found.LIST OF FIGURES

Figure 1)	NFS access of NTFS security style data	6
Figure 2)	CIFS access of UNIX security style data	8

1 Purpose

The purpose of this FAQ is to provide clear direction as to when your customer might want to utilize the multiprotocol capabilities that are built into the NetApp Data ONTAP® operating system.

1.1 Definitions

The following are definitions of commonly used terms in this FAQ.

What does multiprotocol mean?

Answer: NetApp storage systems are designed to provide access to stored data in both Network File System (NFS) and Common Internet File System protocol (CIFS) networking environments. This access is transparent to the client because of native multiprotocol support, providing a bridge between UNIX® file security style and Windows® file security style by mapping the UNIX identities and Windows identities. This report provides a summary of how security styles and user mapping work together to provide a user transparent access to data.

What is security style?

Answer: Security style refers to the style of ACL used on a volume in Data ONTAP. If a volume has UNIX security style, then mode bits or NFSv4 style ACLs are used. If a volume has NTFS security style, then only Windows style ACLs are used. If a volume has mixed security style, then the ACL used will depend on what type of client last modified the ACL.

What is name mapping?

Answer: The process by which one style of user account is mapped to another. Valid name mappings include:

- UNIX user to Windows user
- Windows user to UNIX user
- Kerberos Service Principal Name (SPN) to UNIX user

What is WCC?

Answer: This is the WAFL® credential cache. It is used in 7G/7-Mode to cache the authorization for mapped users to access the “other” security style file system to improve performance by reducing the number of name lookups that need to be made. The cache times out after 20 minutes and is configurable using the `waf.lwcc_minutes_valid` option.

What is authentication?

Answer: The process of identifying an individual, usually based on a username and password. In security systems, authentication is distinct from authorization. Authentication makes sure that the individual is who he or she claims to be, but says nothing about the access rights of the individual.

What is authorization?

Answer: Authorization takes place after authentication. After a user is determined to be valid, the user’s credentials, permissions, group memberships, and so on are gathered to determine what level of access is allowed on a volume or file.

What is the user space?

Answer: The user space is a concept in clustered Data ONTAP where dedicated cluster applications run to provide end-user services, such as access, volume location, and command line availability.

What is the security daemon (SecD)?

Answer: This is the user space application responsible for authentication and authorization requests in clustered Data ONTAP. SecD makes connections to domain controllers and name service servers, sends and receives authentication and authorization requests, translates name mappings, works over RPC to commit configuration changes to the replicated database (RDB), and caches credentials to make sure that there is efficient access to NAS protocols. SecD is a per-node process, where requests will use the SecD application on the node that owns the data LIF on which the request arrived.

What is the NAS module?

Answer: The NAS module is a per-node entity in clustered Data ONTAP that translates CIFS and NFS requests into packets the cluster can understand, as well as back into CIFS and NFS packets after the request has been processed by the cluster. All NAS traffic passes through the NAS module local to the data LIF on which the request arrived. The NAS module also provides caching mechanisms similar to SecD to make sure of efficiency in NAS requests.

What is the replicated database?

Answer: In clustered Data ONTAP, this is what makes a cluster a cluster. The replicated database is a series of tables that get replicated across all nodes in a cluster through the cluster network. These tables remain in sync by way of a set of four user space RDB applications (mgwd, vifmgr, vldb, and bcomd) to make sure that each node in the cluster is aware of all entries in the cluster while providing a single entry point for management.

What is an ACL?

Answer: Access control list. Used to provide a list of users and their access level to a file system or share.

What is a CIFS share?

Answer: Used to share files and folders in an NTFS environment. In 7-Mode, CIFS shares are stored in a flat file called `/etc/cifsconfig_share.cfg`, which gets read after every boot. In clustered Data ONTAP, CIFS shares are stored in RDB tables.

What is an NFS export?

Answer: Used to share files and folders in an NFS environment. In 7-Mode, exports can be stored in two places: a flat file (`/etc/exports`) or memory. Because exports can live in memory, a reboot can clear exports unless they are explicitly added to the exports file. The `-p` (persistent) option in the command `exportfs` adds exports to the exports file. In clustered Data ONTAP, exports are stored in the RDB only and never in memory.

2 Decision Points

When is it necessary to use multiprotocol capabilities?

Answer: Some sites have pure Windows or pure UNIX environments in which all data is accessed using either CIFS and NT file system (NTFS) file security or using Network File System (NFS) and UNIX file security. However, many customer sites must enable data to be accessed from both Windows and UNIX clients. For these environments, Data ONTAP has native multiprotocol support. After the user is authenticated on the network and has both appropriate share or export permissions and the necessary file permissions, all data can be accessed by the user from UNIX hosts using NFS or from Windows hosts using CIFS. Leveraging multiprotocol access does not require using mixed mode volumes and/or qtrees.

3 7-Mode

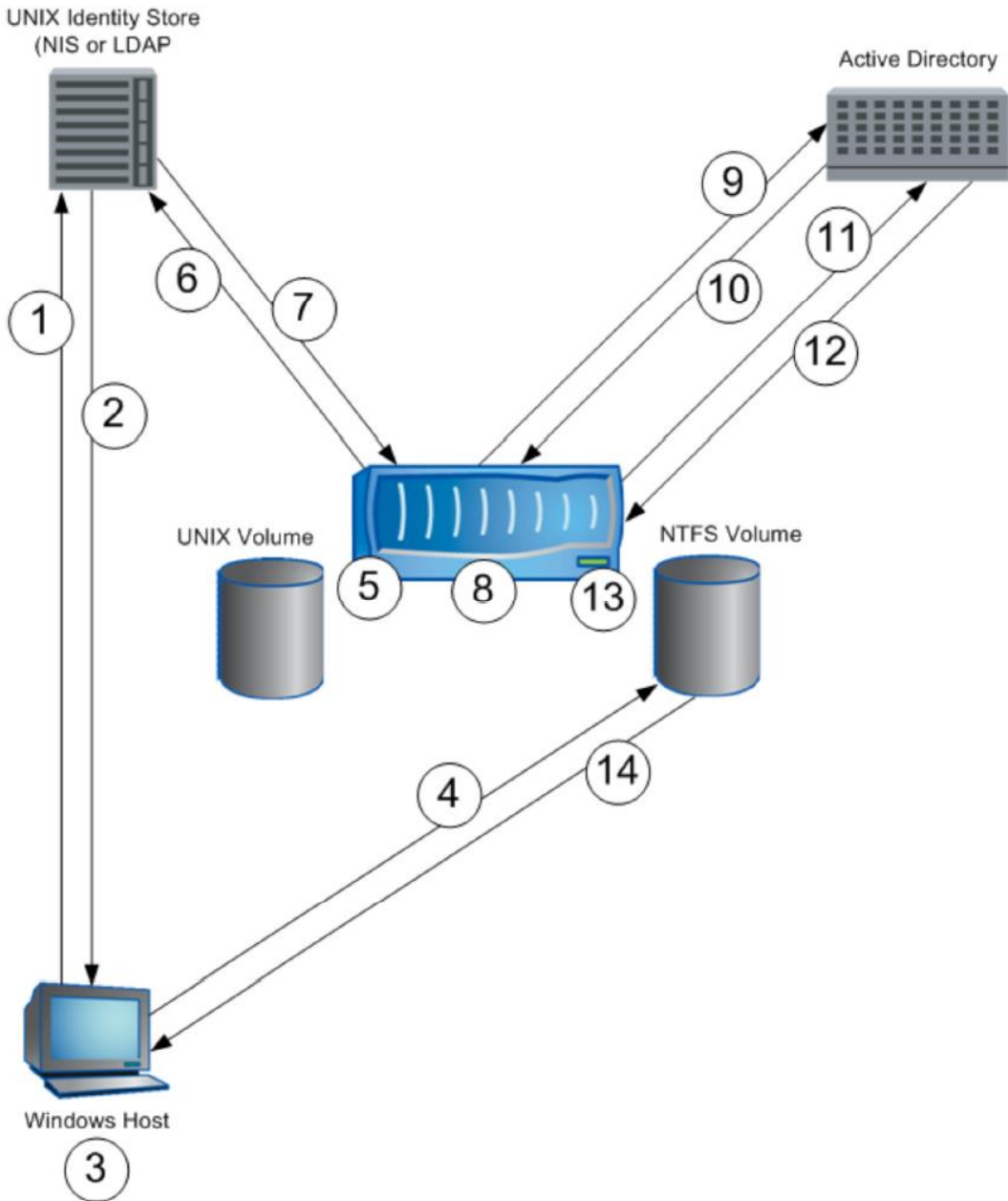
3.1 How It Works

NetApp has implemented user mapping to provide a bridge between the two security styles. When an NFS user requests access to a file that uses nonnative NTFS file security, the user is mapped to his or her corresponding CIFS identity, and the CIFS credentials are used for access checks. File access is granted or denied after the mapped user's credentials are evaluated against the file's Windows ACL. Any time a CIFS client accesses data stored on a NetApp storage system, a mapping process takes place. This is true even if the access request is to data with NTFS Windows ACLs.

NFS Access of NTFS Security Style: 7-Mode

The following section discusses how NFS access of NTFS security style data takes place in 7-Mode. For detailed information, see [TR-3490: "NetApp Storage System Multiprotocol User Guide."](#)

Figure 1) NFS access of NTFS security style data.



1. The user logs in from the UNIX host.

As part of the login process, the host requests user and group information for the user from the name services configured in the client's `/etc/nsswitch.conf` file. The data can be retrieved from local files, a NIS server, or an LDAP server.

2. The configured name service returns user and group information to the UNIX host.

All user information needed to log in, including UID, GID, and the user shell and home directory, is returned. Additionally, the user's secondary group GIDs are returned.

3. Using the user information retrieved from the identity store, the user is authenticated and allowed access to the UNIX host.

The user and group information retrieved in step 1 is cached and used to determine access rights to local and remote resources.

4. The user requests access to a mounted NetApp file system.

The storage system checks export options at the time the file system is mounted. Export options might affect the user's ability to perform a requested action. For instance, if the file system is mounted read only, the user cannot write to the file system even if file permissions allow it. The UNIX client sends an access request for a storage system mount that contains the user's UID and GIDs, including secondary GIDs.

5. The user requests access to data on the storage system that uses NTFS-style permissions, but the request contains UNIX style UID and GIDs.

The storage system cannot use the UID and GIDs sent in the access request for the access check, because NTFS security style leverages NTFS ACLs rather than UNIX mode bits. Instead, the storage system must compare Windows user and group information to the file's Windows ACL to determine access for the user. Therefore, the storage system maps the UNIX user to the corresponding Windows user.

6. The system queries the configured UNIX identity store, supplying the user's UID, and requests the user name.

When the user requests access from a UNIX host, the user identification contents of the request will depend on the version of NFS being used and the NFS options configured on the storage system. If using NFSv3, the request contains the user's UID but does not contain the user name. If using NFSv4.x, the request will use the `username@nfsv4iddomain`.

7. The UNIX name service returns the name of the UNIX user to the storage system.

8. The storage system uses the UNIX user name to begin the user-mapping process.

The storage system checks `/etc/usermap.cfg` and the LDAP user-mapping entries, if configured, to see if there is a specific mapping entry for the UNIX user. If there is a specific mapping entry, the storage system uses this entry for the user-mapping process. If there is not a specific mapping entry, the storage system assumes that the Windows user name is the same as the UNIX user name and automatically uses this name during the mapping process (implicit mapping).

Note: If there is not a specific mapping entry in `/etc/usermap.cfg` or in the LDAP store and LDAP is configured, the Windows user name is assumed to be the same as the UNIX user name, and this name is automatically used in the mapping process. In addition to correlating a Windows user name with a UNIX user name, the mapping process is important in determining whether the mapped user is a valid user. If the mapped name is not a valid user, access might be allowed as the default NT user or completely disallowed, depending on how the WAFL option `waf1.default_nt_user` is configured.

9. The storage system queries Active Directory® to determine if the mapped user name is a valid Windows user.

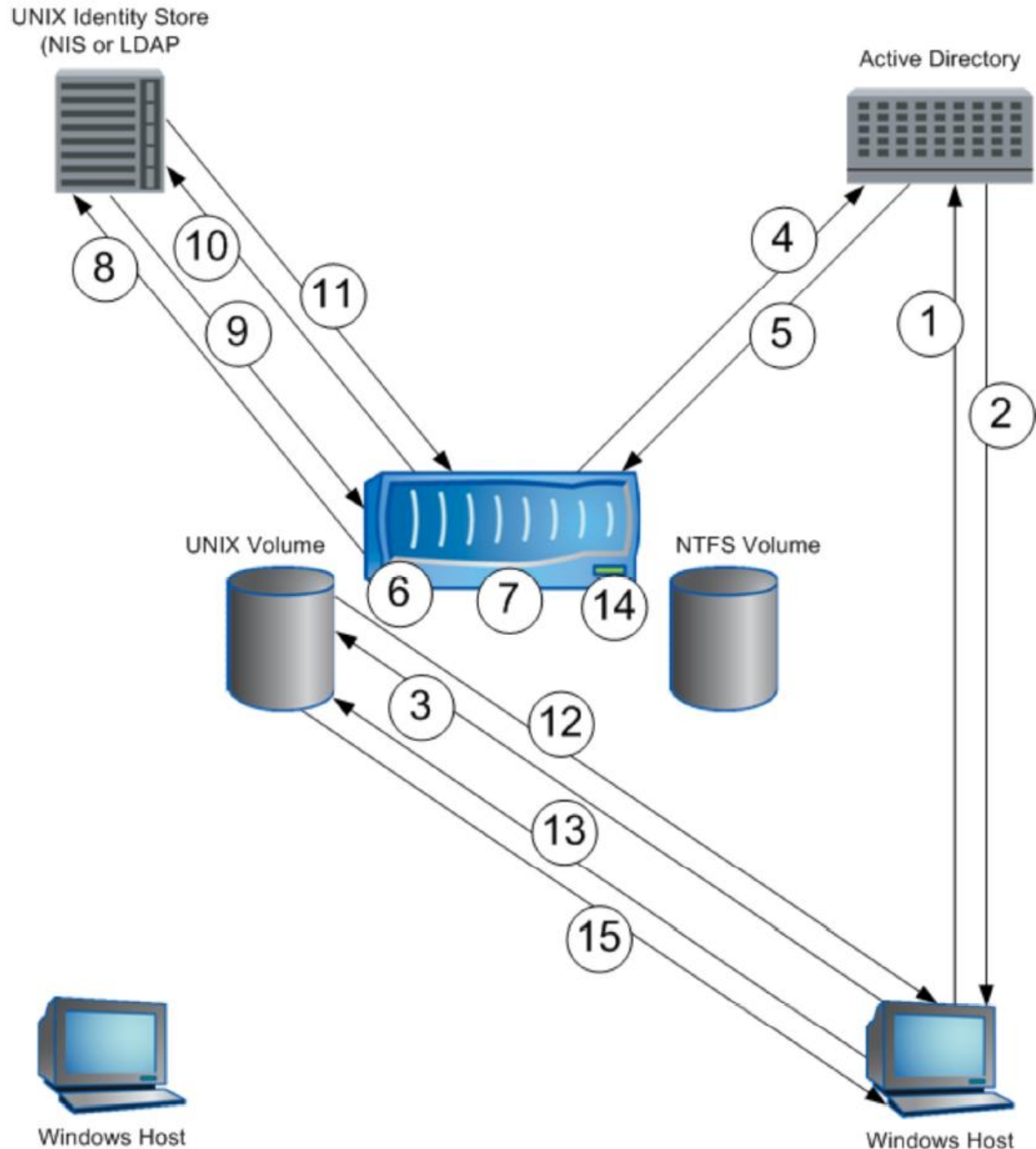
If the user name is a valid Windows user, the mapping process continues. If the user name is not a valid Windows user, either the user is mapped to the generic NT user or access is denied, depending on how the WAFL option `waf1.default_nt_user` is configured.

10. Active Directory replies to the query, either supplying information on the user or returning an error indicating that the user was not found.
11. The storage system queries Active Directory for the mapped Windows user's SIDs (user SID and all group SIDs).
12. Active Directory replies to the query, supplying information on the user, including the user's SIDs.
13. The storage system uses the user's SIDs and compares them to the ACLs of files and directories for which access has been requested.
The system then determines whether the desired action is allowed.
14. The system replies to the access request, either permitting the requested action if the file permissions allow it or denying the action if the permissions do not allow it.
Note: The complete credential retrieved in steps 5 through 12 is stored in the WAFL credential cache. With subsequent access requests, the WCC is checked instead of repeating steps 5 through 12.
Note: The WCC entries expire after a default of 20 minutes. After the entry expires, the user-mapping step is required again, with the results again being cached.

CIFS Access of UNIX Security Style: 7-Mode

The following section discusses how CIFS access of UNIX security style data takes place in 7-Mode. For detailed information, see [TR-3490: "NetApp Storage System Multiprotocol User Guide."](#)

Figure 2) CIFS access of UNIX security style data.



1. The user begins a login from the Windows host.
As part of the login process, the host communicates with the domain controller or the local security database if the user is logging in as a local user.
2. The user's credential is returned to the Windows host.
A credential is returned that contains the user SID and the user's group SIDs. If the user is from an untrusted domain and the domain guest account is enabled, the domain controller authenticates the user with the guest credentials. If a user is logging in locally to a Windows workstation where the local

guest account is enabled and does not have a local account, the user is authenticated with the local guest credentials.

3. The user requests access to data stored on a NetApp file system through the session setup and connect requests, through either mapping a drive or accessing through a UNC path.

When a Windows user requests access to data on a NetApp file system, an authenticated session must first be set up, and then the connection to the share must be made. The steps for session setup depend on the authentication protocol negotiated between the storage system and the client. The Windows client determines which protocol is used. The storage system supports NTLMv1, NTLMv2, Kerberos, and clear text password authentication.

[NTLM authentication](#) is used when the storage system is a member of an NT 4 domain, when operating in local workgroup mode, or if the Windows client requests that NTLM authentication be used. After the NTLM-level authentication is negotiated, the client sends a session setup request. The request contains the user name along with NTLM authentication information and other information used in session setup.

[Kerberos authentication](#) is used when the storage system is a member of an Active Directory domain, has a valid service principal name (SPN), and Kerberos authentication is requested by the client. After the Kerberos-level authentication is negotiated, the client sends a session setup request. Kerberos uses tickets for authenticating user requests to network services. The session setup request contains a ticket granting service (TGS) ticket for the storage system's CIFS service. This ticket is embedded in the session setup request and contains a security blob with all of the information necessary for the storage system to authenticate the user. The TGS ticket also contains the user SID and the SIDs for all of the user's domain groups.

4. The storage system processes the session setup request.

In a domain environment, the storage system does not store information on the user's Windows password; therefore, with NTLM authentication, the storage system must send the user information and NTLM authentication information to the domain controller, which performs the actual user authentication.

With Kerberos, even though the request contains all of the information needed to authenticate the user session, the user name is not contained in the security blob; therefore, the storage system first consults the system's SID cache to see if the user name for that SID is in cache. If it is not, the storage system queries the user's domain controller and does a SID lookup.

5. The domain controller processes the request sent by the storage system and provides the information that the storage system needs to complete the session setup.

In NTLM, the domain controller returns success or failure for the user authentication. If the user is authenticated, the domain controller also returns SIDs for the user and the user's groups. The storage system then includes the SIDs of any local groups to which the user belongs and stores this information in the user's CIFS session cache. If the user who is requesting session setup is a local storage system user, the session authentication is handled locally by the storage system. The SIDs of the local groups to which the user belongs are added to the user's session cache.

In Kerberos authentication, the domain controller returns the user name to the storage system, which already has the user SID and the user's domain group SIDs. The storage system then includes the SIDs of any local groups to which the user belongs, and all SID information is stored in the user's session cache.

Note: In a domain environment, before the storage system can query the domain controllers, the storage system must have a machine account in the domain, authenticate to the domain, and establish a Netlogon pipe. In a Kerberos environment, the storage system must obtain a valid Ticket Granting Ticket (TGT) for the machine account.

If the user is connecting as a guest, the storage system denies access unless the option to allow guest account connection is set:

```
filer> option cifs.guest_account unix_name
```

Unix_name is a user created in the UNIX identity store: `/etc/passwd`, NIS, or LDAP. If you do not want to have guest access to the storage system, set this option to a null value. All guest access requests will be denied. The default for this option is NULL.

6. The user requests access to data on the storage system that uses UNIX style file permissions, but the request contains Windows style SIDs.

The storage system does not use the SIDs sent in the file access request for the access check. Instead, the storage system must compare UNIX user and group information to the file UNIX permission to determine access for the user. To do this, the storage system maps the Windows user to the corresponding UNIX user. Additionally, current Data ONTAP implementations store the UNIX credentials with the user's session cache. Therefore, the Windows to UNIX user mapping occurs prior to completion of the session setup and connection to the share.

7. The storage system uses the Windows user name to begin the user-mapping process.

The storage system checks `/etc/usermap.cfg` or the name service user-mapping entries, if configured, to see if there is a specific mapping entry for the Windows user. If there is a specific mapping entry, the storage system uses this entry for the user-mapping process. If there is not a specific mapping entry, the storage system assumes that the UNIX user name is the same as the Windows user name and automatically uses this name during the mapping process (implicit mapping). If there is no name-mapping rule and no implicit mapping, the storage system will attempt to use the WAFL option `waf1.default_unix_user` to find a valid UNIX user name. If no name is specified, or if the name specified does not exist, the request will fail.

8. The storage system queries the configured UNIX name services to determine if the mapped user name is a valid UNIX user.

If the user name is a valid UNIX user, the mapping process continues. If the user name is not a valid UNIX user, either the user is mapped to the generic UNIX account or access is denied, depending on how the WAFL option `waf1.default_unix_user` is configured.

9. The configured UNIX name service replies to the query, either supplying information on the user or returning an error indicating that the user is not found.

10. The storage system queries the UNIX name services for the mapped UNIX user's user and group information.

11. The UNIX name service replies to the query, supplying information on the UNIX user's credential, including the user's UID and GIDs, and secondary group GIDs.

The UNIX credential is stored with the user's cached authenticated session information. The user-mapping information in this process is not stored in the WCC. If the same user establishes a new CIFS connection, the process is reexecuted.

12. After the UNIX credentials have been added to the authenticated session cache, the storage system can reply to the session setup request with either a success or a failure.

If the session request was successful, the connection request to the share can be processed and is either allowed or denied based on evaluation of the user and group share permissions.

13. The Windows user requests access through the mapped drive to a file or folder in the UNIX security style volume.
14. The storage system uses the UNIX user information stored in the user's session cache and compares it to the UNIX permissions on files and directories for which access has been requested.

The system then determines whether the desired action is allowed. This will depend on the mode security bit set on the file system. For example, if the volume has 0755 permissions, then the owner has full access to the volume and groups, and others have read-only access. The owner and group are used in access checks and can be modified from an NFS client. The storage system uses basic UNIX permission logic in its security models, with hooks for NFSv4 ACLs on UNIX style volumes. For information on how NFSv4 ACLs interact with CIFS, see [TR-3580: NFSv4 Enhancements and Best Practices Guide](#).
15. The system replies to the access request, either permitting the requested action if the file permissions allow it or denying the action if the permissions do not allow it.

4 Clustered Data ONTAP

4.1 How It Works

The Data ONTAP storage system is designed to support either CIFS credentials (NTFS), UNIX credentials (UNIX), or both (mixed), depending on the volume security type. Both of the supported network file systems—CIFS and NFS—can be used to access the files on any of the volume types. In a multiprotocol environment, the name-mapping mechanism plays a key role in controlling the behavior that allows a user with credentials from one network file system type to be mapped to a user with credentials on another network file system type.

Mistakes in configuring name mapping can cause the following problems:

- If no name-mapping entry was found, there is a high probability that the user will be denied access or will not even be able to log into the CIFS server.
- The user might be granted undesired access rights to the system.

Note: Group mapping is not supported in Data ONTAP. Even if you define a group mapping, the mapping entry is ignored under all conditions.

The name-mapping procedure is accomplished in three sequential steps:

1. Explicit name mapping: Explicit name mapping can be defined either locally through the cluster-wide name-mapping table or through LDAP. The name-mapping service switch can be defined in 8.2.x by using the following command:

```
cluster::> vservers modify -vservers <vservers> -fields nm-switch {file|ldap}
```

And in 8.3.x with:

```
cluster::> vservers services name-service ns-switch modify
```

2. Implicit name mapping: When explicit name mapping does not find a matching entry, the system tries implicit name mapping.
3. Default name mapping: When both explicit name mapping and implicit name mapping fail, the default name mapping is used, if defined, as the last resort.

NFS Access of NTFS Security Style: Clustered Data ONTAP

The following section discusses how NFS access of NTFS security style data takes place in clustered Data ONTAP. For a diagram, see Figure 1) NFS access of NTFS security style data.

1. The user begins a login from the UNIX host.
As part of the login process, the host requests user and group information for the user from the name services configured in the client's `/etc/nsswitch.conf` file. The data can be retrieved from local database entries, a NIS server, or an LDAP server.
2. The configured name service returns user and group information to the UNIX host.
All user information needed to log in, including UID, GID, and the user shell and home directory, is returned. Additionally, the user's secondary group GIDs are returned.
3. Using the user information retrieved from the identity store, the user is authenticated and allowed access to the UNIX host.
The user and group information retrieved in step 1 is cached and used to determine access rights to local and remote resources.
4. The user requests access to a mounted NetApp file system.
The storage system checks export rules for the volume at the time the file system is mounted. Export rules might affect the user's ability to perform a requested action. For instance, if the file system is mounted read only, the user cannot write to the file system even if file permissions allow it. The UNIX client sends an access request for a storage system mount that contains the user's UID and GIDs, including secondary GIDs.
5. The user requests access to data on the storage system that uses NTFS-style permissions, but the request contains UNIX style UID and GIDs.
The storage system cannot use the UID and GIDs sent in the access request for the access check, because NTFS security style leverages NTFS ACLs rather than UNIX mode bits. Instead, the storage system must compare Windows user and group information to the file's Window ACL to determine access for the user. Therefore, the storage system maps the UNIX user to the corresponding Windows user.
6. The system queries the configured UNIX identity store, supplying the user's UID, and requests the user name.
When the user requests access from a UNIX host, the user identification contents of the request will depend on the version of NFS being used and the NFS options configured on the storage system. If using NFSv3, the request contains the user's UID but does not contain the user name. If using NFSv4.x, the request will use the `username@nfsv4iddomain`.
Note: If a valid NFSv4 domain is not specified, or the requested user does not exist in the NFSv4 domain, users will map to the "nobody" user specified in the `/etc/ldapd.conf` file on the host. For more information on NFSv4.x in clustered Data ONTAP, see [TR-4067: Clustered Data ONTAP NFS Implementation Guide](#).
7. The UNIX name service returns the name of the UNIX user to the storage system.
8. The storage system uses the UNIX user name to begin the user-mapping process.

The storage system checks the Storage Virtual Machine (SVM) name-mapping rules if local files are specified in nm-switch and ns-switch SVM options. If LDAP is used, LDAP is queried for a valid mapping. If there is a specific mapping entry, the storage system uses this entry for the user-mapping process. If there is not a specific mapping entry, the storage system assumes that the Windows user name is the same as the UNIX user name and automatically uses this name during the mapping process (implicit mapping). If there is no specific mapping entry in either name-mapping rules/LDAP or implicit mapping, the storage system will attempt to use the default Windows user set in the NFS options for the SVM.

9. The storage system queries Active Directory to determine if the mapped Windows user name is a valid Windows user.

If the user name to which the UNIX name maps is a valid Windows user, the mapping process continues. If the user name is not a valid Windows user, either the user is mapped to the default Windows user or access is denied, depending on how the NFS option `-default-win-user` is set.

10. Active Directory replies to the query, either supplying information on the user or returning an error indicating that the user was not found.

11. The storage system queries Active Directory for the mapped Windows user's credentials (user SID and all group SIDs).

12. Active Directory replies to the query, supplying information on the user, including the user's SIDs.

13. The storage system uses the user's SIDs and compares them to the ACLs of files and directories for which access has been requested.

The system then determines whether the desired action is allowed.

14. The system replies to the access request, either permitting the requested action if the file permissions allow it or denying the action if the permissions do not allow it.

Note: The complete credential retrieved in steps 5 through 12 is stored in the NAS blade and SecD cache. With subsequent access requests, these caches are checked instead of repeating steps 5 through 12.

Note: These caches expire after a set period of time. After the entry expires, the user-mapping step is required again, with the results again being cached. The effect of this is negligible.

CIFS Access of UNIX Security Style: Clustered Data ONTAP

The following section discusses how CIFS access of UNIX security style data takes place in clustered Data ONTAP. For a diagram, see Figure 2) CIFS access of UNIX security style data.

1. The user begins a login from the Windows host.

As part of the login process, the host communicates with the domain controller or the local security database if the user is logging in as a local user.

2. The user's credential is returned to the Windows host.

A credential is returned that contains the user SID and the user's group SIDs. If the user is from an untrusted domain and the domain guest account is enabled, the domain controller authenticates the user with the guest credentials. If a user is logging in locally to a Windows workstation where the local

guest account is enabled and does not have a local account, the user is authenticated with the local guest credentials.

3. The user requests access to data stored on a NetApp file system through the session setup and connect requests, through either mapping a drive or accessing through a UNC path.

When a Windows user requests access to data on a NetApp file system, an authenticated session must first be set up, and then the connection to the share must be made. The steps for session setup depend on the authentication protocol negotiated between the storage system and the client. The Windows client determines which protocol is used. The storage system supports NTLMv1, NTLMv2, Kerberos, and clear text password authentication.

[NTLM authentication](#) is used in clustered Data ONTAP when the Windows client requests that NTLM authentication be used. Clustered Data ONTAP does not currently support workgroup mode. After the NTLM-level authentication is negotiated, the client sends a session setup request. The request contains the user name along with NTLM authentication information and other information used in session setup.

[Kerberos authentication](#) is used when the storage system's CIFS machine account has a valid service principal name (SPN) and Kerberos authentication is requested by the client. After the Kerberos-level authentication is negotiated, the client sends a session setup request. Kerberos uses tickets for authenticating user requests to network services. The session setup request contains a ticket granting service (TGS) ticket for the storage system's CIFS service. This ticket is embedded in the session setup request and contains a security blob with all of the information necessary for the storage system to authenticate the user. The TGS ticket also contains the user SID and the SIDs for all of the user's domain groups.

4. The storage system processes the session setup request.

In a domain environment, the storage system does not store information regarding the user's Windows password; therefore, with NTLM authentication, the storage system must send the user information and NTLM authentication information to the domain controller, which performs the actual user authentication.

With Kerberos, even though the request contains all of the information needed to authenticate the user session, the user name is not contained in the security blob; therefore, the storage system first consults the system's SID cache to see if the user name for that SID is in cache. If it is not, the storage system queries the user's domain controller and does a SID lookup.

5. The domain controller processes the request sent by the storage system and provides the information that the storage system needs to complete the session setup.

In NTLM, the domain controller returns success or failure for the user authentication. If the user is authenticated, the domain controller also returns SIDs for the user and the user's groups. The storage system then includes the SIDs of any local groups to which the user belongs and stores this information in the user's CIFS session cache. If the user who is requesting session setup is a local storage system user, the session authentication is handled locally by the storage system. The SIDs of the local groups to which the user belongs are added to the user's session cache.

In Kerberos authentication, the domain controller returns the user name to the storage system, which already has the user SID and the user's domain group SIDs. The storage system then includes the SIDs of any local groups to which the user belongs, and all SID information is stored in the user's session cache.

Note: Before the storage system can query the domain controllers, the storage system must have a machine account in the domain, authenticate to the domain, and establish a Netlogon pipe. In a Kerberos environment, the storage system must obtain a valid ticket granting ticket for the machine account. This is accomplished using `cifs server create`.

CIFS guest accounts are not currently supported in clustered Data ONTAP.

6. The user requests access to data on the storage system that uses UNIX style file permissions, but the request contains Windows style SIDs.

The storage system does not use the SIDs sent in the file access request for the access check. Instead, the storage system must compare UNIX user and group information to the file UNIX permission to determine access for the user. To do this, the storage system maps the Windows user to the corresponding UNIX user. Additionally, current Data ONTAP implementations store the UNIX credentials with the user's session cache. Therefore, the Windows to UNIX user mapping occurs prior to completion of the session setup and connection to the share.

7. The storage system uses the Windows user name to begin the user-mapping process.

The storage system checks for name-mapping rules in the SVM or for the name service user-mapping entries on name services specified in the SVM options for nm-switch and ns-switch, if configured, to see if there is a specific mapping entry for the Windows user. If there is a specific mapping entry, the storage system uses this entry for the user-mapping process. If there is not a specific mapping entry, the storage system assumes that the UNIX user name is the same as the Windows user name and automatically uses this name during the mapping process (implicit mapping). If there is no name-mapping rule and no implicit mapping, the storage system will attempt to use the default UNIX user specified in CIFS server options to find a valid UNIX user name. If no name is specified, or if the name specified does not exist, the request will fail.

8. The storage system queries the configured UNIX name services to determine if the mapped user name is a valid UNIX user.

If the user name is a valid UNIX user, the mapping process continues. If the user name is not a valid UNIX user, either the user is mapped to the generic UNIX account or access is denied, depending on how the CIFS server option `-default-unix-user` is configured.

9. The configured UNIX name service replies to the query, either supplying information on the user or returning an error indicating that the user is not found.

10. The storage system queries the UNIX name services for the mapped UNIX user's user and group information.

11. The UNIX name service replies to the query, supplying information on the UNIX user's credential, including the user's UID and GIDs, and secondary group GIDs.

The UNIX credential is stored with the user's cached authenticated session information. The user-mapping information in this process is not stored in the WCC. If the same user establishes a new CIFS connection, the process is reexecuted.

12. After the UNIX credentials have been added to the authenticated session cache, the storage system can reply to the session setup request with either a success or a failure.

If the session request was successful, the connection request to the share can be processed and is either allowed or denied based on evaluation of the user and group share permissions.

13. The Windows user requests access through the mapped drive to a file or folder in the UNIX security style volume.

14. The storage system uses the UNIX user information stored in the user's session cache and compares it to the UNIX permissions on files and directories for which access has been requested.

The system then determines whether the desired action is allowed. This will depend on the mode security bit set on the file system. For example, if the volume has 0755 permissions, then the owner has full access to the volume and groups, and others have read-only access. The owner and group are used in access checks and can be modified from an NFS client. The storage system uses basic UNIX permission logic in its security models, with hooks for NFSv4 ACLs on UNIX style volumes. For information on how NFSv4 ACLs interact with CIFS, see [TR-4067: Clustered Data ONTAP NFS Implementation Guide](#).

15. The system replies to the access request, either permitting the requested action if the file permissions allow it or denying the action if the permissions do not allow it.

References

The following references are good for further reading:

- File Access and Protocol Management Guide
- Local Storage Management Guide - Clustered Data ONTAP
- Storage Management Guide - 7-Mode
- TR-4067: Clustered Data ONTAP NFS Implementation Guide
media.netapp.com/documents/tr-4067.pdf
- TR-3580: NFSv4 Enhancements and Best Practices Guide
media.netapp.com/documents/tr-3580.pdf
- TR-3490: NetApp Storage System Multiprotocol User Guide
media.netapp.com/documents/tr-3490.pdf
- TR-4379: Name Services Best Practice Guide
media.netapp.com/documents/tr-4379.pdf

Version History

Version	Date	Document Version History
Version 1.0	October 2012	First publishing
Version 2.0	June 2013	Update
Version 2.1	July 2015	Minor updates

Acknowledgements

Bing Cai, Bikash Roy Choudhury, Steve Lawler.

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

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

Go further, faster®



www.netapp.com

© 2013 NetApp, Inc. All rights reserved. No portions of this document may be reproduced without prior written consent of NetApp, Inc. Specifications are subject to change without notice. NetApp, the NetApp logo, Go further, faster, Data ONTAP, and WAFL are trademarks or registered trademarks of NetApp, Inc. in the United States and/or other countries. UNIX is a registered trademark of The Open Group. Windows and Active Directory are registered trademarks of Microsoft Corporation. All other brands or products are trademarks or registered trademarks of their respective holders and should be treated as such.