Technical Report

NetApp Storage System Multiprotocol User Guide

Ellie Berriman and Binguxe Cai, NetApp
May 2011 | TR-3490

EXECUTIVE SUMMARY

NetApp® storage systems are designed to provide access to stored data in both UNIX® 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 the security styles and user mapping work together to provide seamless access to data.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1 INTRODUCTION

1.1 OVERVIEW

Some customer 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.

NetApp supports multiple security models. When a client requests access to a file with UNIX security, the client must present UNIX credentials. Similarly, if a client requests access to a file with NTFS security, the client must present CIFS credentials. Because the UNIX security model and the Windows security model do not directly correlate, one security style's credentials cannot be used when accessing data with a nonnative security style.

Because both models use the concept of a user for authentication and authorization, 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 Access Control List (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.

1.2 PURPOSE AND SCOPE

This technical report provides a summary of the practical application of multiprotocol access of data stored on a NetApp storage system. The purpose of this report is to help storage administrators determine which type of data access configuration is appropriate for their environment.

TR-3014: Multiprotocol Data Access: NFS, CIFS, and HTTP provides additional information on multiprotocol concepts and theory.

2 MULTIPROTOCOL ON THE NETAPP STORAGE SYSTEM

2.1 AUTHENTICATION IN A MULTIPROTOCOL ENVIRONMENT

Before users request access to data on the network, they must be authenticated by whatever method is configured in the customer environment. Users who log in from a Windows client are typically authenticated in a Windows domain; less frequently, they are authenticated locally on the workstation. UNIX users are typically authenticated locally by the system or by Kerberos, a secure network-based authentication service. After the drive is mapped or an export is mounted, the user can request access to data.

UNIX/NFS and Windows/CIFS differ in how they authenticate users:

- **UNIX users.** On UNIX systems, an export can be mounted manually by root, or, more typically, is mounted at boot time or mounted on demand by the automounter. After the export is mounted, the user can request access to data. NFS is a connectionless protocol, and each NFS request includes the user ID (UID) and group IDs (GIDs) of the user making the request. If the system authenticates the user, the UNIX client determines the UID, primary GID, and secondary/auxiliary GIDs when the user first logs in. This is done by retrieving the information from `/etc/passwd` and `/etc/groups`,

Network Information System (NIS), or the Lightweight Directory Access Protocol (LDAP) identity store. If Kerberos authenticates the user, the UNIX client gets only the user's primary credential, including UID and primary GID. Upon receiving an NFS request, the NetApp storage system determines the secondary/auxiliary GIDs by retrieving the information from the local `/etc/passwd` and `/etc/groups`, NIS, or the LDAP identity store.

- **Windows users.** Before the Windows user can access data on a storage system, the user must create an authenticated session with the storage system (mapping a drive). Because CIFS is session-based, the identity of the user can be determined just once, when the session is first set up. Data ONTAP always maps the user's Windows identity to the user's UNIX identity during a CIFS session setup process. Failures in identity mapping might disrupt the establishment of a CIFS session.
- **Data ONTAP.** On the contrary, Data ONTAP does not map the user's UNIX identity to the user's Windows identity during the UNIX/NFS authentication process. Data ONTAP maps only the user's UNIX identity to the user's Windows identity when a UNIX user tries to access files/folders with an effective Windows NT ACL.

2.2 EXPORT AND SHARE SECURITY

In addition to file security, access to data over a network is secured by share or export access checks.

When data is accessed from a Windows client from CIFS, a share to the data must exist and the user must have sufficient share permissions to perform the requested action. This is true whether the file security style is UNIX or NTFS. Windows share permissions and NTFS Windows ACLs are both user based. When you evaluate whether a specific action is permitted on the file, both share and file permissions must allow it.

When data is accessed from a UNIX client with NFS, an export to the data must exist and the UNIX client must have sufficient export permissions to perform the requested action. This is true whether the file security style is UNIX or NTFS. UNIX export rules are host-based; therefore, when a user wants to perform an action on a file or folder over NFS, the UNIX host must be permitted the requested action and the user's credential must have the required file access permissions.

SHARE-LEVEL ACLS

The following list contains guidelines for share-level ACLs:

- Share-level ACLs include Full Control, Change, Read, and No Access.
- The effective rights of a user are the combination of share and Windows ACLs.
- The most restrictive permissions take effect.
- No access takes precedence over any granted permissions, file, or share.
- CIFS share ACLs can be managed through the Windows Computer Management snap-in, through FilerView®, or through the `cifs access` command.

EXPORT OPTIONS

The following list contains guidelines for export options:

- Export options are set on the storage system with the `/etc/exports` file.
- Exports can be changed manually or added with the `exportfs` command.
- Export options apply to the host; however, the effective permissions for the user are based on the combined export and file access controls.
- The most restrictive permission takes effect. If a file system is exported as read only, the user has read-only access even if the file permission would grant write access.
- The root export option is used to grant root access to the NetApp file system to the root user on UNIX hosts.

2.3 DATA ONTAP FILE SECURITY STYLES AND MULTIPROTOCOL ACCESS

All files and folders stored on NetApp storage systems are protected by one of two file security schemes, UNIX or NTFS. The effective security schemes are represented by three security styles: UNIX, NTFS, and mixed.

- Every volume and qtree is given a default security style at creation time. The default security style is set with the option `options wafl.default_security_style [unix | ntfs | mixed] (default: unix).`
- The security style of a volume or qtree can be changed after creation with the following command: `qtree security <path> [unix | ntfs | mixed]`

QTREE SECURITY STYLES AND THE NETAPP STORAGE SYSTEM

The following list contains brief explanations of qtree security styles:

- **UNIX qtree style** indicates that the file security style of the qtree is UNIX based.
- **NTFS qtree style** indicates that the file security style of the qtree is Windows based.
- **Mixed qtree style** indicates that either the UNIX or the NTFS file security style could be the effective security style for each file or folder within the qtree.

  With mixed volumes or qtrees, some files in the qtree or volume might have UNIX security style and some might have NTFS security style. Each file or folder has only one security style—either NTFS or UNIX—but not both. A file's security style depends on whether the permission was last set from CIFS or NFS. For example, if a file currently uses the UNIX security style and a CIFS user sends a set-ACL request to the file, the file's security style is changed to NTFS. If a file currently uses the NTFS style and an NFS user sends a set-permission request to the file, the file's security style is changed to UNIX.

Referring to security style does not refer to the type of client used to access the data. Data in all three security styles can be accessed from both Windows and UNIX hosts if the appropriate user mapping is in place and if the share, export, and file access permissions allow it.

Security style refers to the style of file permissions and the type of authorization needed to access the directories and files within a volume or qtree.

- **UNIX security style.** For the UNIX security style, authorization to access directories and files is based on access allowed to the user's UNIX UID and GIDs, with access rules following the UNIX style of file permissions through mode bits (rwxrwxrwx) or through NFSv4 ACL.
- **Windows security style.** For the Windows security style, authorization to access folders and files is based on access allowed to the user's Windows security identifier (SID) and Windows group SIDs, with access rules based on NTFS Windows security descriptors.

2.4 SECURITY STYLES AND USER MAPPING

User mapping between a Windows user and a UNIX user is a fundamental part of NetApp multiprotocol access. Multiprotocol access depends on user mapping between a user's UNIX identity and Windows identity to evaluate the user's rights to perform file and folder operations within volumes and qtrees.

The following sections contain access logic for the qtree security styles.

UNIX QTREES

Use the following guidelines for accessing a UNIX-style qtree:

- A UNIX-style qtree is always accessed with a user's UNIX credentials.
- If a user accesses data through NFS, the user's UID and GIDs are used to determine access rights.

- If a user accesses data in a UNIX volume through CIFS, Data ONTAP first maps the Windows user name to its corresponding UNIX UID. If there is no corresponding UNIX user, by default the Windows user is mapped to the default UNIX user. The default user is designated with the following storage system option: `options wafl.default_unix_user` *`user_name`* `(default: pcuser)`

  The designated default UNIX user may be any valid UNIX user designated by the storage administrator, but the default user must exist in the storage system's `/etc/passwd` file, the NIS database, or the LDAP database.

  If this option is set to the null string, Windows users who are not mapped to a UNIX user are not allowed to log in. Therefore, if this option is set to the null string, it is vital that each Windows user can map successfully to a specific UNIX user.

NTFS QTREES

Use the following guidelines for accessing an NTFS-style qtree:

- An NTFS-style file system is accessed with a user's Windows credentials.
- Data ONTAP always maps the user's Windows identity to the user's UNIX identity during the CIFS authentication process. Hence, a CIFS user credential also contains its mapping UNIX credential.
- If a CIFS user accesses NTFS security style files or folders with Windows ACLs, the SIDs of the CIFS user and groups are used to determine NTFS access rights. The user and group SIDs are compared to the file's Windows security descriptor.
- If a UNIX user accesses files and folders with Windows ACLs in an NTFS qtree, Data ONTAP maps the UNIX user to the Windows user based on the user mapping policy. Upon failed identity mapping, Data ONTAP rejects the access request. Upon successful mapping, Data ONTAP grants access based on the user's mapped Windows user. The user's Windows user SID and Windows group SIDs are used to determine NTFS access rights.
- There is an exception. A user's Windows credential is not used if a user requests access to data in an NTFS security style qtree or volume. When a mixed or UNIX security style volume or qtree is changed to NTFS security style, the files and folders contained within the qtree or volume do not automatically have Windows ACLs placed on them. In this situation, if the file or folder has existing UNIX-style permissions, the UID and GIDs are used for the access check. Section 4.2, Changing a UNIX or Mixed Volume with UNIX-Effective Security to an NTFS Volume, contains more information on this exception.
- By default, if there is no corresponding Windows user, access is denied to the NTFS qtree from a UNIX host. This is because the option that is used to configure a generic Windows account is, by default, set to null.

  `wafl.default_nt_user` is a storage system option that can be used to allow mapping of Windows users with no corresponding UNIX account to a generic Windows account. The default for this option is null.

  To enable access from a default user, add a valid Windows account to this option: `options wafl.default_nt_user corp\ntuser`

MIXED QTREES

Use the following guidelines for accessing a mixed-style qtree:

- **For mixed-style volumes and qtrees,** access is based on the effective security style on the volume and the qtree, folders, and files within the volume.
- **A mixed volume or qtree** can have either UNIX- or NTFS-style security in place. However, only one security style is applied to any one file or folder. A file or folder has either NTFS security style or UNIX security style in place, but not both.

- **Particular folders or files within a mixed volume or qtree** can have a security style that differs from the root of the volume or qtree, but both security styles don't apply at the same time to any particular folder or file.

Consider the following qtree and a folder that was created within it:

- `eddie:/vol/vol1/qtree_mixed`
- `eddie:/vol/vol1/qtree_mixed/ntfs_folder`

In this example, the root of `qtree_mixed` has UNIX-style security—it uses UID and GIDs to determine access rights. However, the `folder ntfs_folder` was created by a Windows administrator using a mapping on a Windows machine. The Windows administrator specified that this folder should not inherit parent permissions. Instead, the administrator gave the folder its own specific NTFS permissions.

The addition of specific NTFS permissions changed the security style of this folder and all data subsequently placed within it to NTFS security style.

2.5 MULTIPROTOCOL ACCESS AND GROUP MAPPING

Data ONTAP user mapping maps user names. It does not map groups. Due to the fundamentally different way that Windows groups and UNIX groups are configured, it is not possible to perform a consistent, exact, and simple mapping between UNIX groups and Windows groups. However, because group membership is critically important when determining file access, as part of the mapping process the mapped user's group membership is retrieved and cached along with the user mapping information.

If a user is supposed to have access to a file or folder based on a group membership, determine that the user is a member of a group that correlates to the file's security style. Also determine that this group has appropriate access rights to the file.

For instance, suppose that a CIFS user is a member of a Windows group called engineer and requests access to a UNIX security style file with a group owner of engineer:

```
-rwxr-xr-x    1 fred     engineer        0 Jul  7 07:19 file1
```

Also, the user is not the file owner and expects access to be granted based on rights granted to the UNIX engineer group; the mapped UNIX user must be a member of the UNIX engineer group. Access is not granted based on the Windows user's membership in the Windows engineer group.

2.6 CACHING USER MAPPING INFORMATION: THE WAFL CREDENTIAL CACHE

After performing UNIX-to-Windows user mapping, the results, including group membership for both the UNIX user and the Windows user, are stored in the storage system's WAFL® credential cache (wcc).

Windows-user-to-UNIX-user mapping is not stored in the wcc. Instead, with Windows-to-UNIX-user mapping, the UNIX user information is kept as part of the CIFS session credential. A fresh Windows-to-UNIX user mapping is needed only when a new CIFS session is established for that user.

By default, the information is cached for 20 minutes. The cache time can be increased with the option `wafl.wcc_minutes_valid`.

`options wafl.wcc_minutes_valid` *`minutes`* (Valid range is 1 through 20,160.)

Use the storage system command `wcc` to determine the effective user mapping between UNIX and Windows user.

The `wcc` command simulates NFS-to-Windows-user mapping; however, there is no CIFS authentication step when using the `wcc` command. This command cannot be used to troubleshoot CIFS authentication issues but it is useful for troubleshooting user mapping issues. The `wcc` command does not look in the

wcc. Each time the command is run, the user map operation is performed, providing current user map information.

EXAMPLE: MAPPING A WINDOWS USER TO A UNIX USER

```
jeckle*> wcc -s sunny\enid
(NT - UNIX) account name(s):  (SUNNY\enid - enid2)
        ***************
        UNIX uid = 136
        user is a member of group staff (10)

        NT membership
                SUNNY\enid
                SUNNY\Domain Users
        User is also a member of Everyone, Network Users,
        Authenticated Users
        ***************
```

EXAMPLE: MAPPING A UNIX USER TO A WINDOWS USER

```
jeckle*> wcc -u anne
(NT - UNIX) account name(s):  (ELLIE\anne - anne)
        ***************
        UNIX uid = 118
        user is a member of group perform (110)

        NT membership
                ELLIE\anne
                ELLIE\Domain Users
                BUILTIN\Users
        User is also a member of Everyone, Network Users,
        Authenticated Users
        ***************
```

2.7 MANAGING FILE AND FOLDER SECURITY

The following list contains guidelines for managing file and folder security:

- CIFS file and folder-level ACLs can be managed with the Security tab of the file or folder Properties tab. NTFS ACLs also can be managed from a UNIX client with the `smbcacls` program.
- UNIX file and directory-level permissions can be managed by the UNIX `chmod` command. Special permissions such as changing the file owner or group are managed with the `chown` or `chgrp` command.
- Starting with Data ONTAP 7.2, UNIX permissions can also be managed from the Windows Security tab, which is accessed through a CIFS share.
- Starting with Data ONTAP 7.2.2, an additional layer of security, Storage-Level Access Guard, was introduced to guard the security (permissions and auditing) on volumes and qtrees levels through the `fsecurity` console command. This new security cannot be set or modified by normal CIFS or NFS administrative clients. This prevents such clients from overriding policies that should be managed at the storage level, not the protocol level. TR-3596: Storage-Level Access Guard Quick Start Guide and TR-3597: Bulk Security Quick Start Guide provide detailed information.

3 DISPLAY AND EFFECTIVE PERMISSIONS IN A MULTIPROTOCOL ENVIRONMENT

Requests from users are evaluated using the real file permissions that are in effect on the file or folder. However, Data ONTAP must provide a method for viewing display permissions from nonnative clients.

There are two ways that Data ONTAP manages display permissions, depending on the Data ONTAP version. All versions of Data ONTAP prior to 7.2 handle display permissions as outlined in the following section. Starting with Data ONTAP 7.2, if you enable a specific option, display permissions are displayed differently than in previous versions. Section 3.2, Display and Effective Permissions Beginning with Data ONTAP 7.2, describes the new permission display method and information about enabling this feature. Section 3.3, Interaction Among Mode Bits, NFSv4 ACL, and NT ACL with Data ONTAP 7.3, describes the interaction among UNIX-mode bits, NFSv4 ACLs, and NT ACLs on Data ONTAP 7.3.

3.1 DISPLAY AND EFFECTIVE PERMISSIONS PRIOR TO DATA ONTAP 7.2

When a host views the displayed security information associated with a file or directory that has nonnative file permissions (that is, the Windows host displays permissions on a UNIX file system or a UNIX host displays permissions on an NTFS file system), a set of display permissions is created but they are generally not the same as the effective permissions. They are intended to represent as closely as possible the semantics of the nonnative security information. The permissions are for display purposes only and are not used for actual permission checking.

NFS CLIENT REQUESTS PERMISSIONS FROM AN NTFS-STYLE SECURITY VOLUME

An NFS client requesting permissions for a file with NTFS-style security gets a UNIX permission set derived from the ACL. Because some NFS clients actually use the display permissions to prescreen certain kinds of file access, the returned display permissions show the maximum access allowed to any user in the ACL. This often results in a displayed UNIX permission that appears to be granting read, write, and execute access to all. However, this representation is for display purposes only. Regardless of the display permissions, file access is still granted based on the ACL.

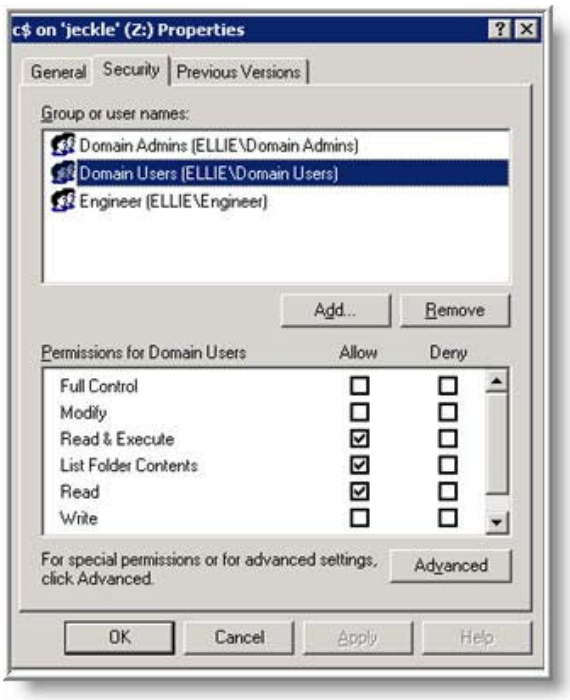
The following example shows a UNIX display and NTFS effective permissions of an NTFS volume.

The display permissions appear to show that all users have read, write, and execute permissions:

```
drwxrwxrwx    8 root     root         4096 Jul  5 08:38 /mnt
```

However, the effective permissions show that not all users may write.

Figure 1) Effective permissions from a Windows client of a volume with NTFS security style.

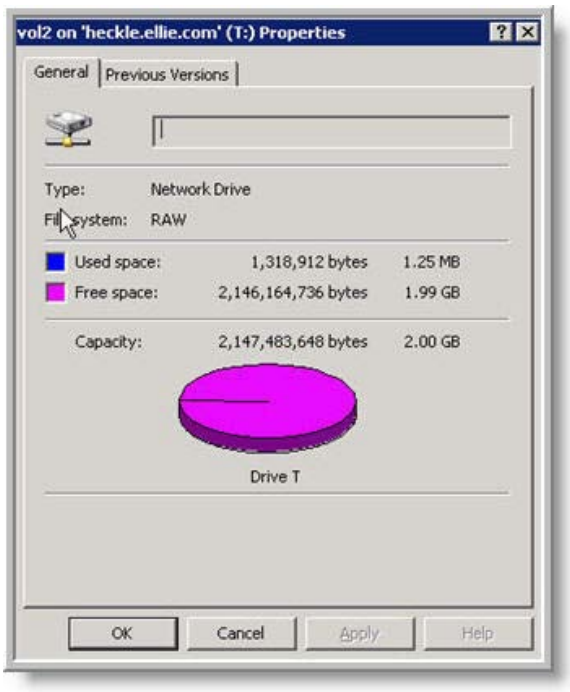


CIFS CLIENT REQUESTS PERMISSIONS FROM A UNIX-STYLE SECURITY VOLUME

The display permissions in this case differ depending on whether the volume is UNIX security style or mixed security style with an effective security style of UNIX.

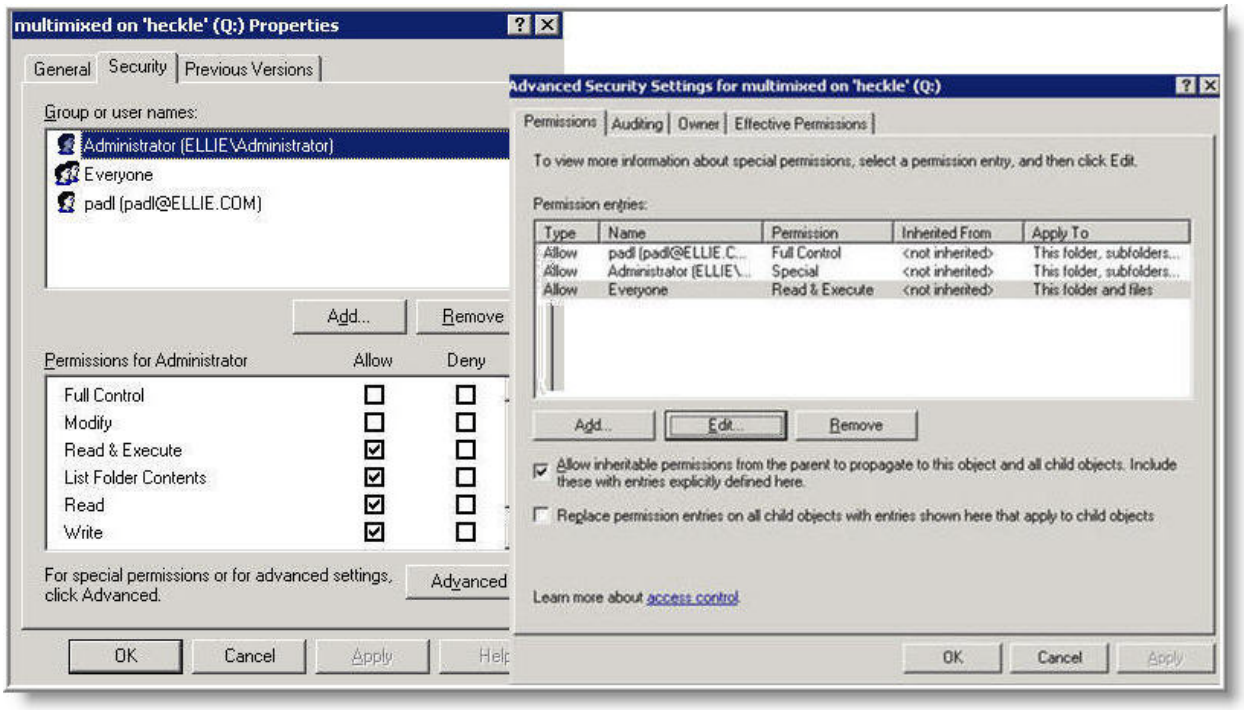
If the security style of the volume is UNIX, display permissions are not displayed. The Security tab is not available when a UNIX security style volume is mapped on a Windows machine. The mapped drive appears to be formatted with a file allocation table (FAT), which has no file permissions. In this case, any representation of the permissions must be viewed from UNIX mounts. This behavior changes with Data ONTAP 7.2. Section 3.2, Display and Effective Permissions Beginning with Data ONTAP 7.2, provides a description of the changes to display permissions in Data ONTAP 7.2.

Figure 2) Display permissions from a Windows client of a volume with UNIX security style.



If the security style of the volume is mixed with an effective security style of UNIX, the Security tab is available when the volume is mapped on a Windows machine. A set of display permissions is available for viewing. A CIFS client requesting the security descriptor for a file with UNIX-style security gets a display ACL that is based on the rights granted to the owner of the file, the rights granted to the requester, and the rights granted to everyone. Again, this ACL is for display only, and is not used for permission checking.

Figure 3) Display permissions from a Windows client of a volume with mixed security using UNIX effective security.



3.2 DISPLAY AND EFFECTIVE PERMISSIONS BEGINNING WITH DATA ONTAP 7.2

NFS CLIENT REQUESTS PERMISSIONS FROM AN NTFS-STYLE SECURITY VOLUME

There is no change between the way previous versions of Data ONTAP and Data ONTAP 7.2 and later display NTFS security style permissions on a UNIX host.

CIFS CLIENT REQUESTS PERMISSIONS FROM A UNIX-STYLE SECURITY VOLUME

By default, there is no change in displaying UNIX permissions from a CIFS client. However, by enabling the following option, the new multiprotocol feature is implemented and there is a change between the way previous versions of Data ONTAP and Data ONTAP 7.2 and later display UNIX-style permissions on a Windows host.

Enable the option:

```
cifs.preserve_unix_security  on    (default is off)
```

If the security style of the volume is either UNIX or mixed with an effective security style of UNIX when this option is enabled, the Windows Security tab is available in the file or the folder Properties tab and now displays the effective UNIX owner, group, and other access permissions on the file or folder. The special UNIX permissions, `suid`, `sgid`, and `svtx`, are also displayed.

In addition to viewing effective UNIX permissions, UNIX permissions can also be changed from the Windows Property Security tab when the new security feature is available. The new functionality is limited to changing permissions on the existing owner and group and on the other `suid`, `sgid`, and `svtx` entities. Ownership and group cannot be changed from the Windows Security tab.

Because Windows file permissions do not directly correlate to the standard Windows file permissions, UNIX permissions are viewed by clicking the Advanced button in the Security Properties tab. Permissions can be viewed and edited by selecting a UNIX entity and clicking the Edit button in the Advanced window.

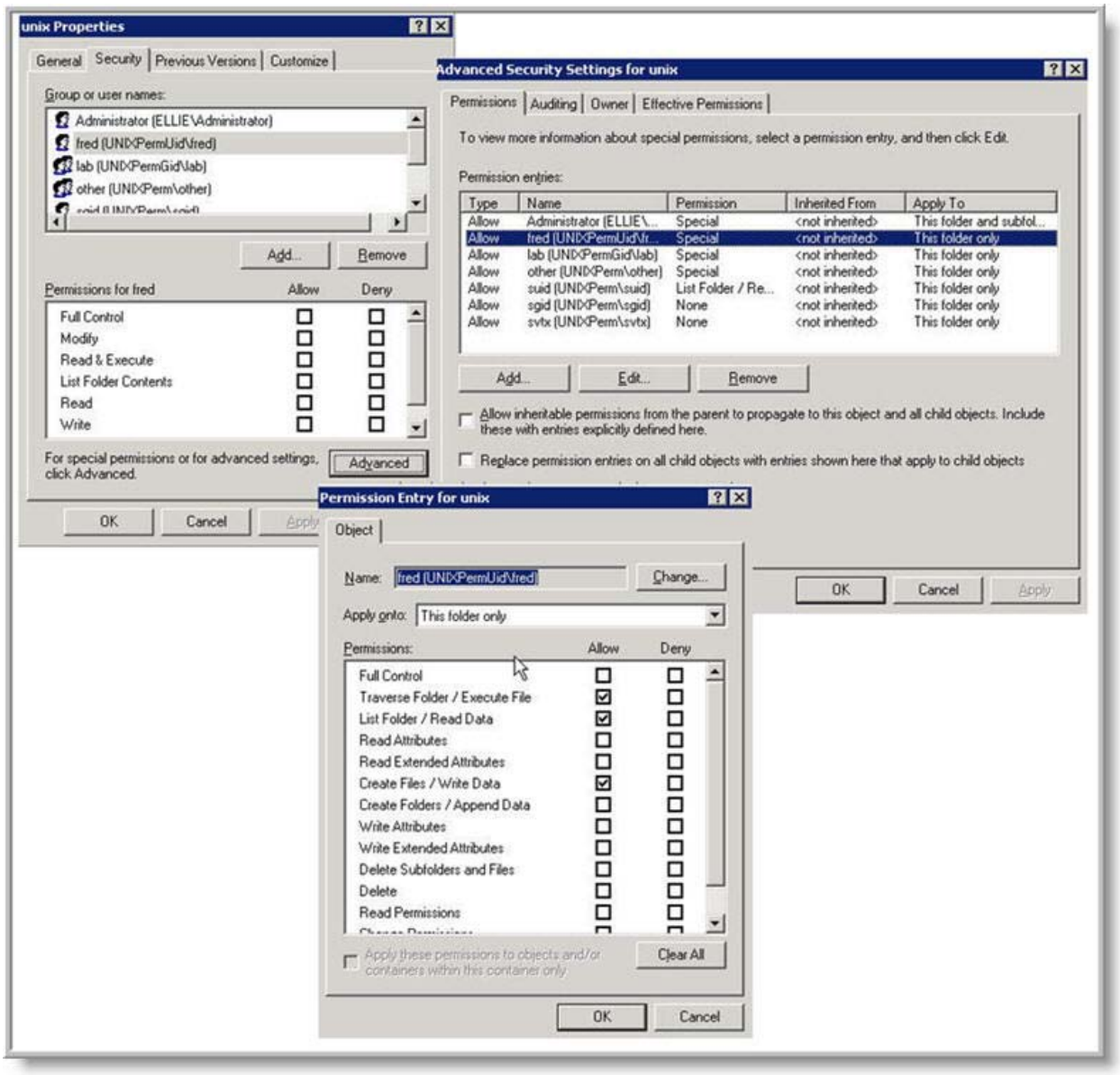
The following example shows the UNIX permissions for a UNIX security style file, test doc.

```
-rwsrwsr-t  1 fred lab 160 May 10 08:20 test.doc
```

Fred and the lab group have read, write, and execute permissions on the file. Other has read and execute permissions. The special permissions `suid`, `guid`, and `svtx` are set on the file. The following figures illustrate how these permissions are displayed in the Windows Security tab.

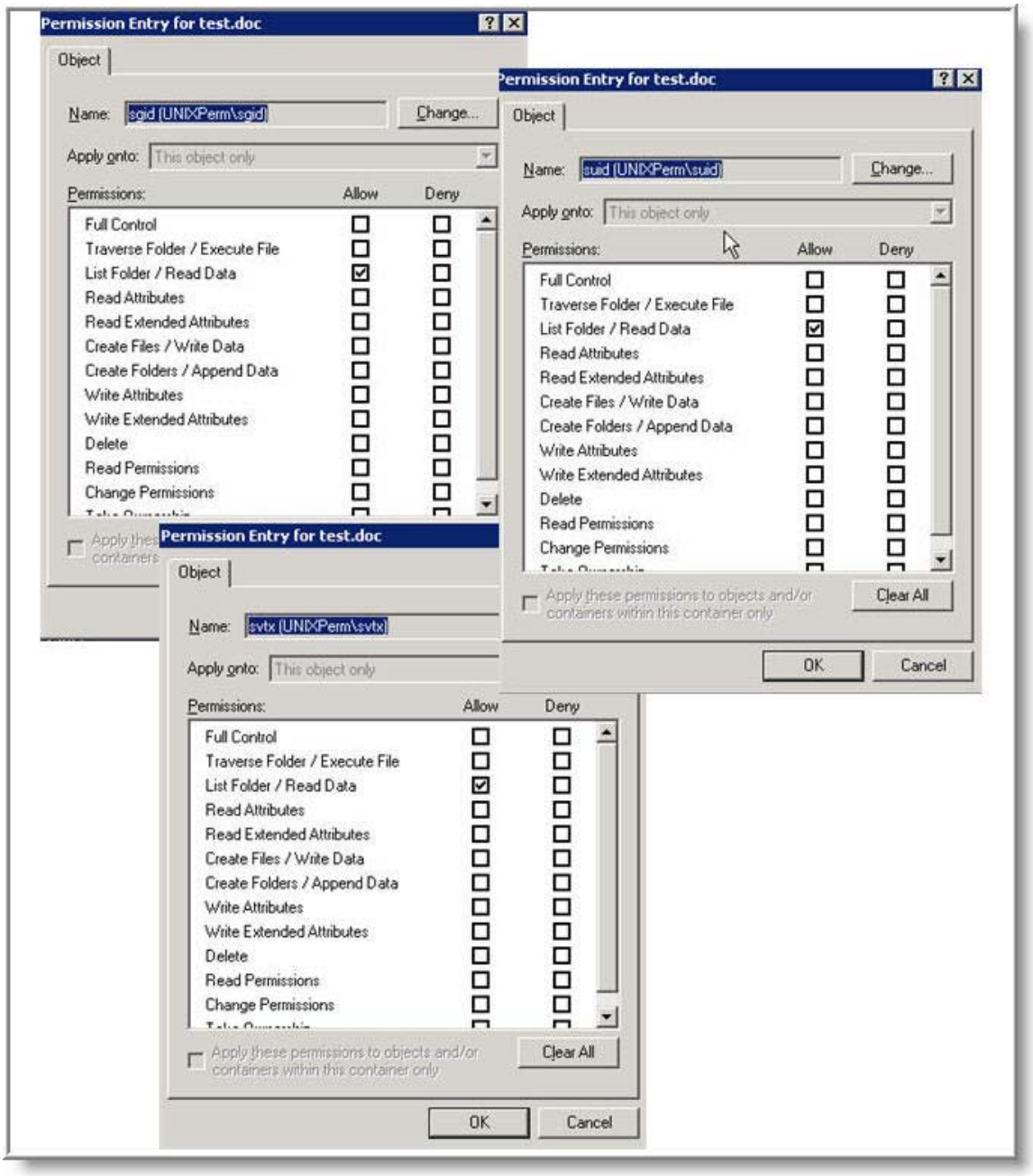
To set UNIX permissions from the Windows Security tab, select the appropriate permission, click OK, and then click Apply.

Figure 4) Data ONTAP 7.2 display permissions from a Windows client of a UNIX security style volume with the new multiprotocol feature enabled.



Traverse Folder/Execute File corresponds to the UNIX execute permission; List Folder/Read Data corresponds to the UNIX read permission; and Create Files/Write Data corresponds to the UNIX write permission.

Figure 5) Data ONTAP 7.2 `suit`, sgid, and svtx display permissions from a Windows client of a UNIX security style volume with the new multiprotocol feature enabled.



List Folder/Read Data corresponds to the UNIX permissions `suid`, `sgid`, and `svtx`. To set the permissions, choose this permission for each of the special permission entries—`suid`, `sgid`, and `svtx`. Click OK, and then click Apply.

3.3 INTERACTION AMONG MODE BITS, NFSV4 ACL, AND NT ACL WITH DATA ONTAP 7.3

With Data ONTAP 7.3, you can use the following option to enable/disable NFSv4 ACLs. This option does not affect whether an ACL is enforced and does not affect the existing ACLs.

`options nfs.v4.acl.enable     on | off [off]`

UNIX-STYLE SECURITY VOLUME

When a file or directory is placed in the volume of UNIX-style security, only UNIX-mode bits and NFSv4 ACLs are considered for permission checking. NT ACLs are ignored.

CIFS is not allowed to change permission on a file in a UNIX-style security volume into NT ACL. However, you are able to change some UNIX-mode bits through a new feature introduced in Data ONTAP 7.2 by setting:

`options cifs.preserve_unix_security  on    (default is off)`

If a file has effective NT ACL, it always has UNIX-mode bits attached for display purposes. Upon NFS GETATTR request, the display UNIX-mode bits is returned.

If a file has effective NT ACL and a UNIX client performs a `chmod` on permission, `chgrp`, or `chown`, the NT ACL is dropped. Effective UNIX-mode bits then take effect instead.

If a file has effective NT ACL, it always has UNIX-mode bits attached for display purpose. Upon a get NFSv4 ACL request, a display NFSv4 ACL is calculated based on display UNIX-mode bits. Data ONTAP does not support direct mapping from NT ACL to NFSv4 ACL.

If a file has effective NT ACL and NFSV4 sets ACL on the file, the NT ACL is dropped. NFSv4 ACL is set and takes effect. The UNIX-mode bits are calculated accurately based on the effective NFSv4 ACL and get set on the file.

If a file has effective mode bits and NFSV4 sets ACL on the file, the NFSv4 ACL is created and takes effect immediately. The mode bits are calculated accurately based on the effective NFSv4 ACL and get set on the file as well. The new mode bits do not affect the SUID/SGID/STICKY bits that might be set originally on that file. All subsequent permission checks are done against the NFSv4 ACL.

If a file has effective mode bits upon a get NFSv4 ACL or NT ACL request, a display NFSv4 ACL or NT ACL is calculated based on the mode bits and returned to the client.

If a file has effective NFSv4 ACL or NT ACL and a UNIX client performs a `chmod` to change the permissions on the file, the NFSv4 ACL is dropped and the mode bits get set and take effect. All subsequent permission checks are done from the mode bits.

If a file has effective NFSv4 ACL or NT ACL and a UNIX client performs a `chmod` to do a SUID/SGID/STICKY operation, the NFSv4 ACL is not dropped and these special permission bits are added to the mode bits.

If a file has effective NFSv4 ACL and a UNIX client performs a `chown` or `chgrp`, the NFSv4 ACL is not affected.

If a file has effective NFSv4 ACL, it always has mapping UNIX-mode bits attached for display purposes. Upon a GETATTR request, the display UNIX-mode bits are returned to the client.

NTFS-STYLE SECURITY VOLUME

When a file or a directory is placed in the NTFS-style security volume, only UNIX-mode bits and NT ACLs are considered for permission checking. NFSv4 ACLs are ignored.

NFS is not allowed to change permissions on a file in an NTFS-style security volume.

If a file has effective NT ACL, it always has UNIX-mode bits attached for display purposes. Upon an NFS GETATTR request, the display UNIX-mode bits are returned.

If a file has effective NT ACL, it always has UNIX-mode bits attached for display purposes. Upon a get NFSv4 request, a display NFSv4 ACL is calculated based on display UNIX-mode bits. Data ONTAP does not support direct mapping from NT ACL to NFSv4 ACL.

If a file has effective UNIX-mode bits upon a set NT ACL request, NT ACL is set and takes effect. Display UNIX-mode bits are calculated to have the maximum access allowed to any user in NT ACL and are set to the file as well.

If a file has effective UNIX-mode bits upon a get NT ACL or NFSv4 ACL request, a display NT ACL or NFSv4 ACL is calculated on the fly based on the effective UNIX-mode bits.

If a file has effective NFSv4 ACL upon a set NT ACL request, NT ACL is set and takes effect. Display UNIX-mode bits are calculated to have the maximum access allowed to any user in NT ACL and are set to the file as well.

If a file has effective NFSv4 ACL, it always has a mapping UNIX-mode bit attached for display purposes. Upon a GETATTR request, the display UNIX-mode bits are returned to the client.

If a file has effective NFSv4 ACL upon a get NT ACL request, a display NT ACL is mapped based on effective NFSv4 ACL. Note that this ACL mapping might be a PARTIAL mapping for the following reasons:

- An NFSV4 ACE containing an NFS user is mapped to an NT ACE with mapping Windows user.
- If an NFSV4 ACE contains an NFS user that cannot be mapped, it is ignored by Data ONTAP during the ACL mapping process.
- Any NFSV4 ACEs containing NFS group information is ignored as well because Data ONTAP does not support group mapping.

MIXED-STYLE SECURITY VOLUME

When a file or a directory is placed in the volume of mixed-style security, the attached effective permission (UNIX-mode bits, NFSv4 ACL, or NT ACL) is considered for the permission checking.

If a file has effective mode bits, either NFSV4 or CIFS requests to get ACL, and display NFSv4 ACL or display NT ACL is calculated on the fly and returned to the client.

If a file has effective mode bits, either NFSV4 or CIFS requests to set ACL, and the NFSv4 ACL or NT ACL is set on the file and becomes effective. Then display-mode bits are calculated and also get set on the file:

- The display-mode bits are calculated accurately based on the NFSv4 ACL.
- The display-mode bits are calculated to have the maximum access allowed based on the NT ACL.

If a file has effective NFSv4 ACL or NT ACL and a UNIX client tries to perform a `chmod` to change the permission on the file, the NFSv4 ACL or NT ACL is dropped. The mode bits are set and take effect. All subsequent permission checks are done from the mode bits.

If a file has effective NFSv4 ACL or NT ACL and a UNIX client performs a `chmod` to perform a SUID/SGID/STICKY operation, the NFSv4 ACL or NT ACL is dropped and these special permission bits are added to the mode bits.

If a file has effective NFSv4 ACL upon a get NT ACL request, a display NT ACL is mapped based on effective NFSv4 ACL. Such ACL mapping might be a partial mapping; the NTFS-Style Security Volume section contains reasons.

If a file has effective NFSv4 ACL, it always has mapping UNIX-mode bits attached for display purposes. Upon a GETATTR request, the display UNIX-mode bits are returned to the client.

If a file has effective NT ACL, it always has UNIX-mode bits attached for display purposes. Upon an NFS GETATTR request, the display UNIX-mode bits are returned.

If a file has effective NT ACL, it always has UNIX-mode bits attached for display purposes. Upon a get NFSv4 ACL request, a display NFSv4 ACL is calculated based on display UNIX-mode bits. Data ONTAP does not support direct mapping from NT ACL to NFSv4 ACL.

If a file has effective NT ACL and NFSV4 sets ACL on the file, the NT ACL is dropped. NFSv4 ACL is set and takes effect. The UNIX-mode bits are calculated accurately based on the effective NFSv4 ACL and get set on the file.

If a file has effective NT ACL and a UNIX client performs a `chmod` on permission, `chgrp`, or `chown`, the NT ACL is dropped. Effective UNIX-mode bits then take effect instead.

4 CHANGING SECURITY STYLES

4.1 CHANGING AN NTFS OR MIXED VOLUME WITH NTFS-EFFECTIVE SECURITY TO A UNIX VOLUME

If an NTFS or mixed qtree is changed to UNIX, the permissions used revert to a minimally permissive set such that the allowed permissions are at least as strict as the ACL. Because the full range of possibilities in an ACL cannot be represented in the UNIX security model, some access is necessarily denied by the synthesized UNIX permissions that would have been granted by the ACL. For example, assume that file `ABC` is in an NTFS or mixed qtree and has the following ACL entries:

```
Joe (owner)     Full Control (All)
Bill            Change (RWXD)
Everyone        Read (RX)
```

If the qtree is changed to UNIX, both Bill and Everyone belong to the Other category. The synthesized UNIX permissions for Other are set to the minimally permissive setting, which is r-x. Therefore, Bill loses the ability to write to or delete the file `ABC`.

Even though UNIX permissions are used when accessing the files, the ACLs are not removed when the qtree security style is changed to UNIX. As long as the files do not have their security modified with `chmod`, `chown`, `chgrp`, and so on, the original ACLs continue to be stored. If the volume security style is changed back to NTFS, the original ACLs are restored.

4.2 CHANGING A UNIX OR MIXED VOLUME WITH UNIX-EFFECTIVE SECURITY TO AN NTFS VOLUME

If a volume or qtree with UNIX-effective security style is changed to NTFS security style, the root volume or qtree directory gets an ACL that allows everyone full control if it did not already have an ACL. But it does not add ACLs to directories and files within the volume. On the root volume, `/etc/` does not get an ACL.

Files created before the security style is changed do not have Windows NT® permissions unless the volume or qtree previously had NTFS security style and already has an ACL in place. For these files, the storage system uses only the UNIX-style permission bits to determine access. ACLs must be placed on existing files and folders within the volume or qtree, either on a file-by-file basis or by setting ACLs on a parent folder and propagating them down to child objects. New files or folders receive ACLs depending on the rules of inheritance and propagation set up by the Windows administrator on that file system.

5 ROOT AND ADMINISTRATIVE ACCESS

Root has special privileges on UNIX file systems and Windows administrators have special privileges on Windows file systems; therefore, root and administrative file and folder access requests require additional access checks.

5.1 ROOT PRIVILEGED ACCESS

For UNIX-style file permissions, if the root user is on a client that is granted root access in the `exportfs` file, root always has file access permissions.

If a root user on a client that is not granted root access in the `exports` file attempts to perform an action on a file or folder, the user is mapped to the nobody user.

In the following example, root from a client without root export rights creates a file.

```
 [root@orion]# touch test1
[root@orion]# ls -l
-rw-r--r--    1 nfsnobody nfsnobody        0 Jul 10 10:52 test1
```

For UNIX security style volumes, Windows administrators always have access to files as if they were root if the appropriate mapping configuration exists. There are two ways to map a Windows user to root:

- Turn on the option that allows all Windows administrators to map to root:
  `options wafl.nt_admin_priv_map_to_root on`
- Configure the `/etc/usermap.cfg` file to allow specific users to map to root:
  In the following example, the following entries perform one-way mapping of specific Windows users to root:
  `ellie\rodrigo => root`
  `sunny\enid => root`

Prior to Data ONTAP 7.2, permissions on UNIX security style volumes could not be configured through the Windows Security tab. The permissions are configured from a UNIX host with the `chmod` command.

Data ONTAP 7.2 and later releases allow UNIX permissions to be configured from the Windows Security tab as long as that user maps to a UNIX user who has the appropriate rights to change permissions on that object. Special UNIX permissions (`chown` and `chgrp` commands) cannot be configured from the Windows interface even with Data ONTAP 7.2 or later releases.

5.2 WINDOWS ADMINISTRATOR PRIVILEGED ACCESS

Windows administrators do not always have access to NTFS-style files. The NTFS permissions can be configured so that an administrator cannot access the files.

Windows administrators can always take ownership of files and grant themselves access.

The option `cifs.nfs_root_ignore_acl` controls how Data ONTAP handles ACLs placed on an NTFS-style file system when a root user has been granted root privileges to the exported NTFS security style volume.

`options cifs.nfs_root_ignore_acl on` allows root to access files with ACLs, even if the ACLs would not allow access to the mapped Windows user.

`options cifs.nfs_root_ignore_acl off` requires that the Windows user to whom root maps has appropriate access rights in the ACLs to perform the requested action.

In addition to setting the option `cifs.nfs_root_ignore_acl` to on, the export options must also be set to allow root access to the UNIX host from which the root user accesses data.

Even if the option `cifs.nfs_root_ignore_acl` is set to on, a root user cannot perform privileged commands such as changing ownership on files or changing permissions on files with NTFS security style volumes. These actions must be performed by Windows administrators.

Windows file security can also be managed with the Samba tool, smbcacls.

Because security style determines what type of file permissions is used to control access, choose NTFS security style if the file system is to be administered by Windows administrators and access is to be controlled using Windows users and groups.

5.3 MIXED SECURITY STYLE AND WINDOWS ADMINISTRATOR AND ROOT PRIVILEGED ACCESS

Files and folders in mixed security style volumes can have either Windows-style security or UNIX-style security, but not both at the same time. Both Windows administrators and UNIX administrators can administer file security within the volume, with the effective security style of the file or folder determined by which type of administrator last changed permissions or the owner on the file or folder.

If mixed security style volumes are configured, it is important to have documentation that outlines which areas of the volume are controlled by NTFS-style security and which areas are controlled by UNIX-style security. Administrators must be careful not to change the security style of files or folders inadvertently. Doing so might cause access issues for users, especially when a folder has been changed from UNIX security style to NTFS security style. Windows, by default, propagates changes, and carefully crafted UNIX permissions on subfolders might be changed inadvertently.

Prior to Data ONTAP 7.2, areas of a mixed volume that have UNIX security style files and folders need to be administered from UNIX hosts. With Data ONTAP 7.2 and later, UNIX permissions can be changed from the Windows Security tab, but changes to the owner and group ownership must still be performed from a UNIX host.

Areas of a mixed volume that have NTFS-style files and folders should be administered from the Windows Security tab.

A file's security model can be flipped from one style to another by NFS set attribute or CIFS set ACL requests. Only a file's owner can flip the style.

6 MULTIPROTOCOL ACCESS PROCESS

This section describes four main methods used to access data:

- A UNIX user on a UNIX or Linux host accesses data on a UNIX security style volume.

  This type of file access does not require a user mapping operation.
- A UNIX user on a UNIX or Linux host accesses data on a Windows security style volume.

  This type of file usually requires a user mapping operation.

  There is an exception. If a mixed or UNIX security style qtree is changed to an NTFS security style qtree, the files and folders contained within the qtree or volume do not automatically have Windows ACLs placed on them. In this situation, if the data being accessed has existing UNIX permissions, UNIX UIDs and GIDs are used for the access check. In this case, the mapping does not need to be done. Section 4.2, Changing a UNIX or Mixed Volume with UNIX-Effective Security to an NTFS Volume, contains detailed information on this data access method.
- A Windows user authenticating either with domain authentication or local workgroup authentication accesses data on a UNIX security style volume.

This type of file access requires a user mapping operation.

- A Windows user authenticating either with domain authentication or local workgroup authentication accesses data on an NTFS security style volume.

  This type of file access requires a user mapping operation.

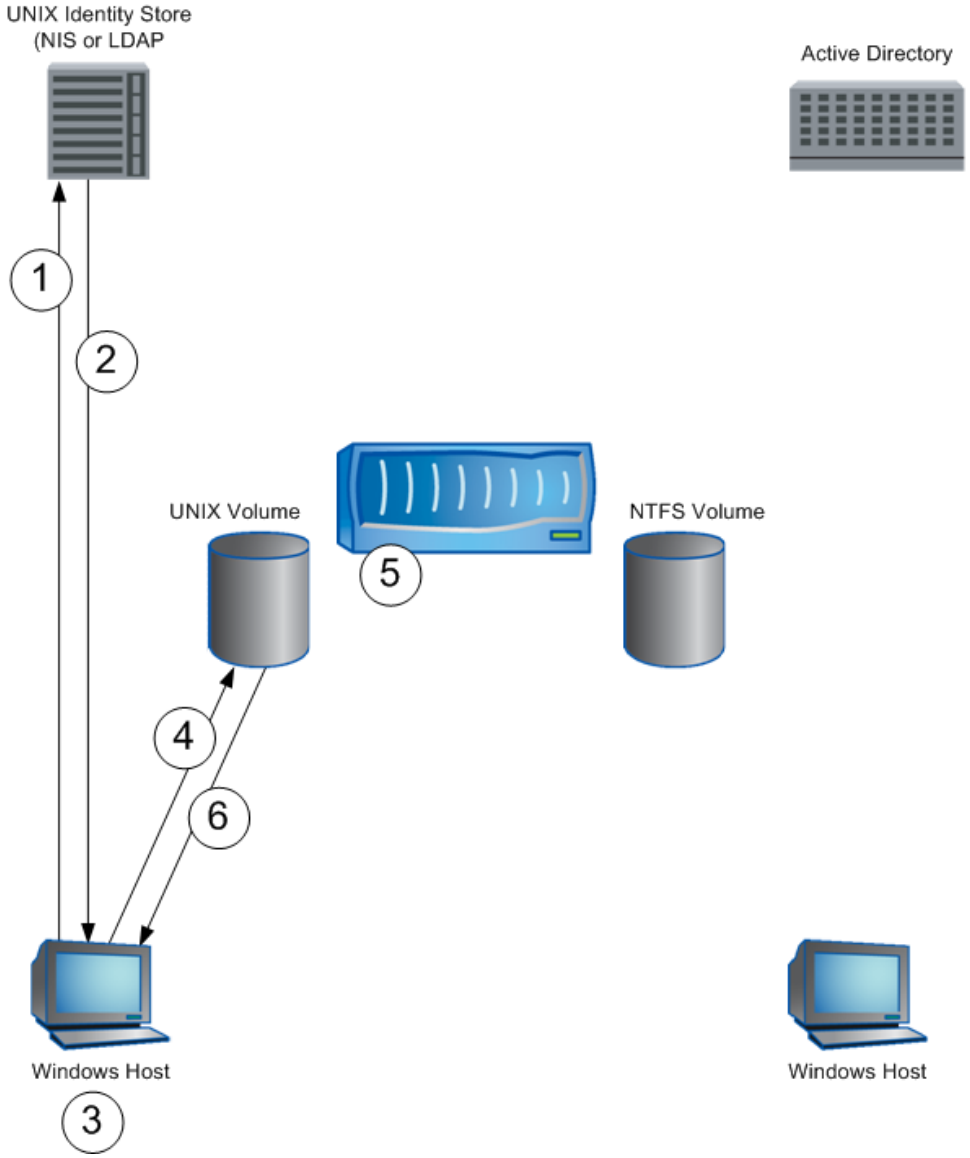
  Even though this is a Windows user accessing NTFS security style data, user mapping still occurs. Current implementations of Data ONTAP always perform a user mapping operation when data is accessed by CIFS.

The four main access methods are described in detail in the following sections. These descriptions are valid for NFS v2 and NFS v3. Special mapping operations that are used when the user is root or a Windows administrator are not included in the mapping process described in this section. See section 5, Root and Administrative Access, for information on how Data ONTAP handles access requests from root or Windows administrators.

6.1 NFS ACCESS OF UNIX SECURITY STYLE DATA

Figure 6 shows the process used when a user on a UNIX host accesses data with UNIX security style using NFS.

Figure 6) NFS access of UNIX security style data.



NFS ACCESS OF UNIX SECURITY STYLE DATA

The following list describes NFS access of UNIX 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 `/etc/nsswitch.conf` file. The data can be retrieved from local files, an 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 file or folder access request contains the user's UID and GIDs, including secondary GIDs.

5. The storage system uses the UID and GIDs sent in the access request for the access check.

   Because the file system has UNIX security style, with UNIX rwxrwxrwx type of file permissions the storage system uses the UID and GIDs with the access check. The system determines if the UID is the owner of the file. If not, the system determines if any of the user's groups are the objects group owner. If not, then other permissions apply. The system then determines whether the desired action is allowed.

6. 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.

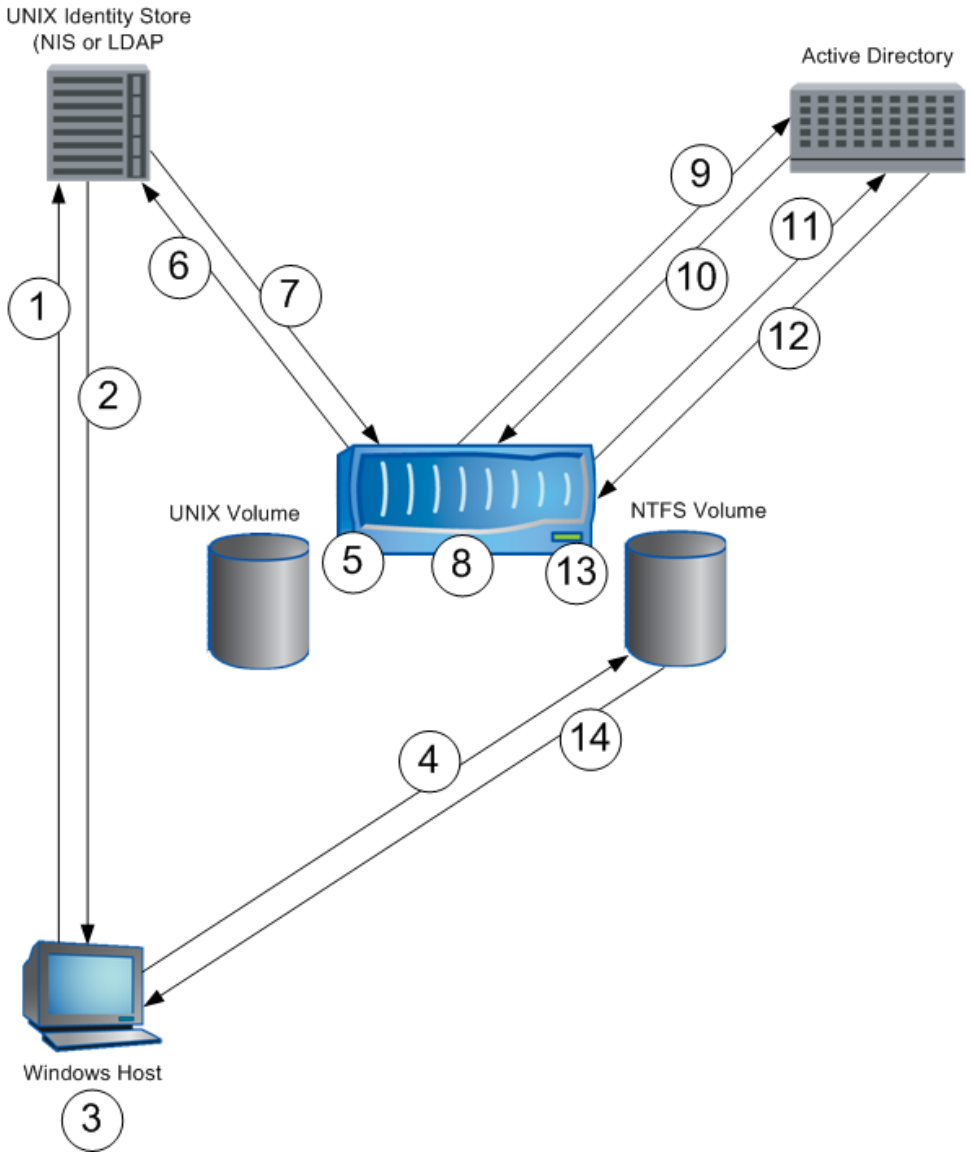
Note: NFS access of UNIX security style data does not include a user mapping step.

Note: This section does not describe how the process handles access requests by root. See section 5 for information on root access.

6.2 NFS ACCESS OF NTFS SECURITY STYLE DATA

Figure 7 shows the process for a user on a UNIX host to access data with NTFS security style using NFS.

Figure 7) NFS access of NTFS security style data.



The following list describes NFS access of NTFS security style data:

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

   As part of the log-in process, the host requests user and group information for the user from the name services configured in the `/etc/nsswitch.conf` file. The data can be retrieved from local files, an 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. 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 request contains the user's UID but does not contain the user name. After mapping is done by user name, the system must determine the UNIX user name before the mapping process can proceed.

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.

   Note: If there is not a specific mapping entry in `/etc/usermap.cfg` or in the LDAP store and LDAP user mapping 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 `wafl.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, the user is either mapped to the generic NT user or access is denied, depending on how the `wafl` option `wafl.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.

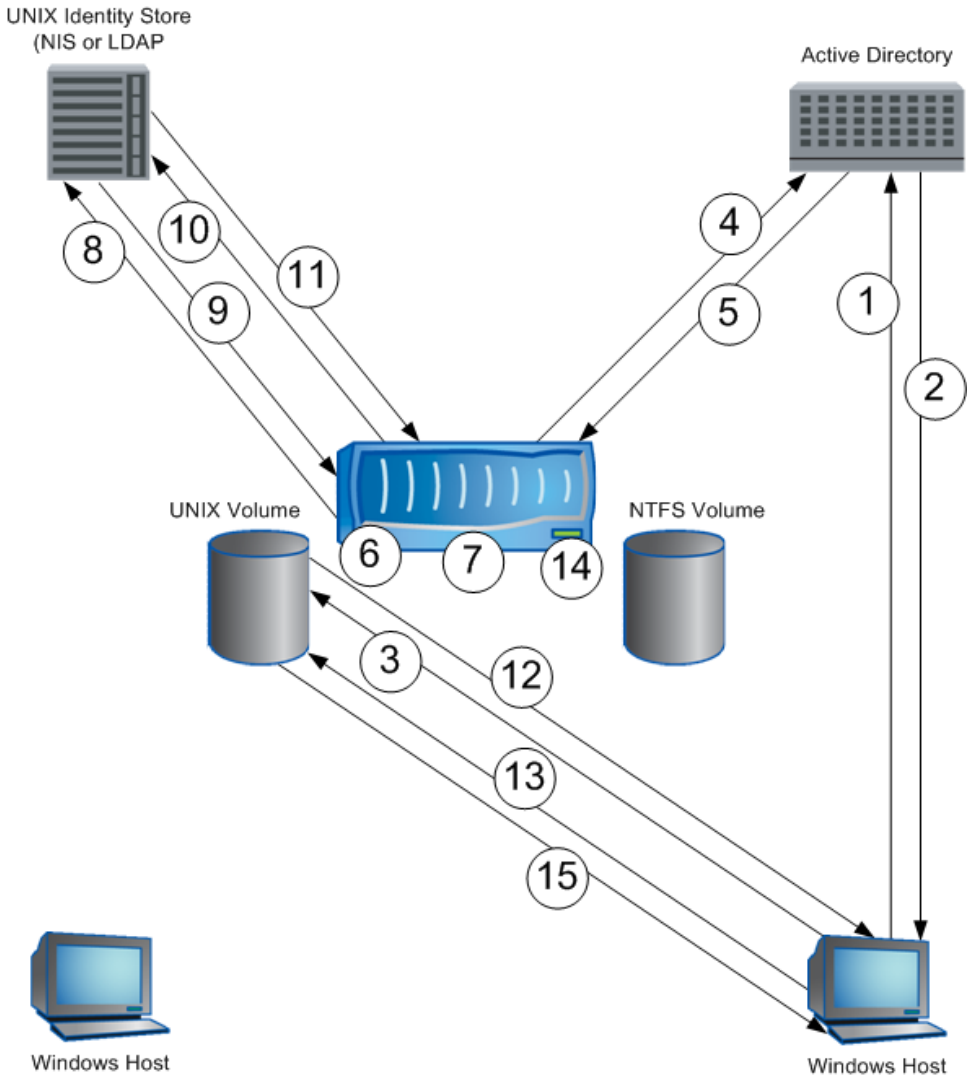
The wcc entries expire after a default of 20 minutes. After the entry expires, the user mapping step is required again, with the results again being cached.

Note: This section does not describe how the process handles access requests by root. Root access is described in section 5.

6.3 CIFS ACCESS OF UNIX SECURITY STYLE DATA

Figure 8 shows the process used when a user on a Windows host accesses data with UNIX security style using CIFS.

Figure 8) CIFS access of UNIX security style data.



The following list describes 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 user's domain controller or the local security database if the user is logging in as a local user. The process might vary depending on whether the user is part of an Active Directory domain or an NT domain, or is a local user.

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

The result is a credential that contains the user SID and the user's group SIDs.

If the user is from a nontrusted 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, either through 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

   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 set-up request. The request contains the user name along with NTLM authentication information and other information used in session setup.

   Kerberos Authentication

   Kerberos authentication is used when the storage system is a member of an Active Directory domain and Kerberos authentication is negotiated between the client and the storage system.

   After the Kerberos-level authentication is negotiated, the client sends a session set-up request. Kerberos uses tickets for authenticating user requests to network services. The session set-up request contains a Ticket Granting Service (TGS) ticket for the storage system's CIFS service. This ticket is embedded in the session set-up 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 set-up request.

   NTLM Authentication

   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.

   Kerberos Authentication

   Even though the request contains all of the information needed to authenticate the user session, the user name is not 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.

   NTLM Authentication

   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.

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 for the machine account. There might be additional traffic between storage system and domain controllers during the session setup; however, the basic process is as outlined here.

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

`options cifs.guest_account unix_name`

`unix_name` is a user created in the UNIX identity store: `/etc/passwd`, NIS, or LDAP.

If guest access to the storage system is not desired, set this option to a null value. All guest access requests will be denied. The default for this option is NULL. Guest access is denied by default.

`options cifs.guest_account ""`

If the user is connecting to the storage system as a local user, the option cifs.guest_account determines whether access as a guest user is allowed or denied.

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 cannot 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. Therefore, 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 ldap 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.

   Note: If there is not a specific mapping entry in `/etc/usermap.cfg` or in the ldap store, ldap user mapping is configured, and the Windows user name is automatically used, the mapping process is still performed. It is important to determine whether the UNIX user is a valid user. If the mapped name is not a valid user, access might be as the default user or it might be completely disallowed, depending on how the `wafl` option `wafl.default_unix_user` is configured.

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, the user is either mapped to the generic UNIX account or access is denied, depending on how the `wafl` option `wafl.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. Now that the UNIX credentials have been added to the authenticated session cache, the storage system can reply to the session set-up 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.

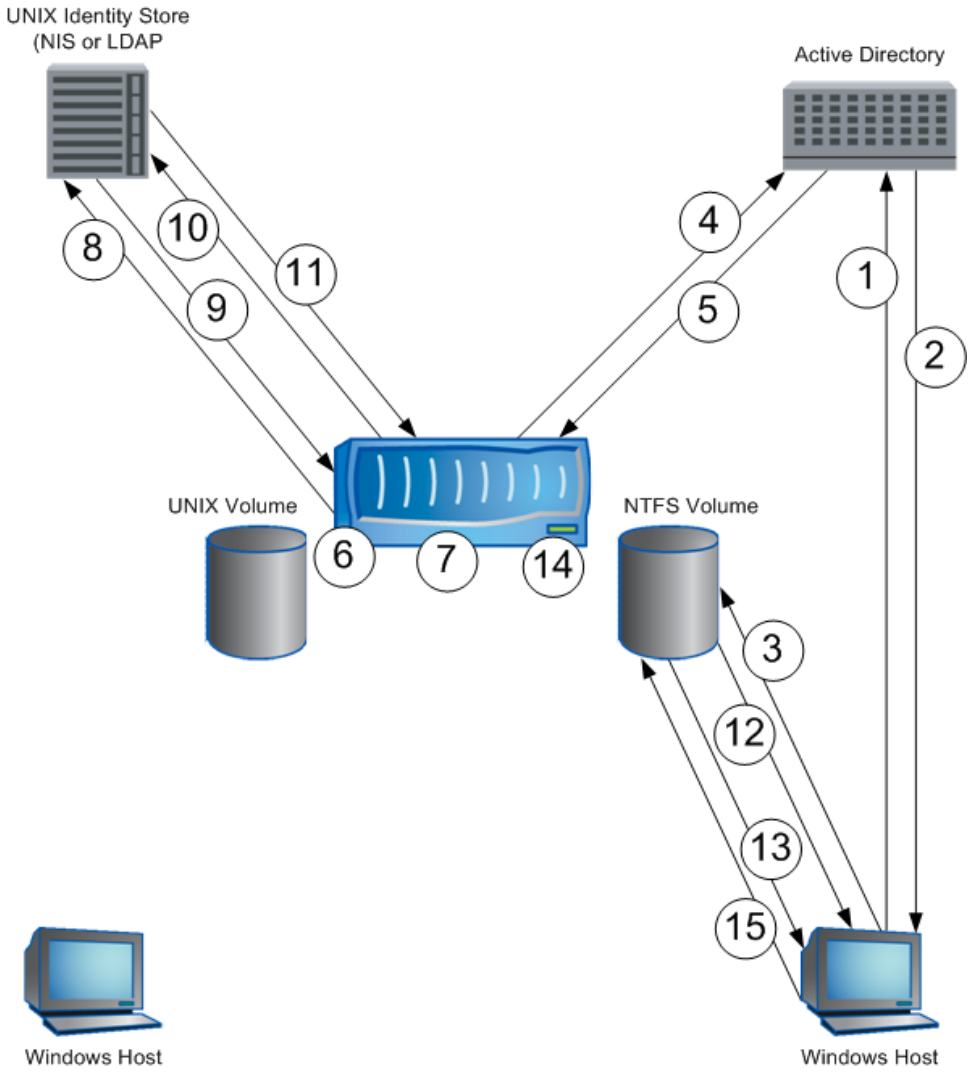
    Because the file system has UNIX security style with UNIX rwxrwxrwx type of file permissions, the storage system uses the UID and GIDs with the access check. The system determines if the UID is the owner of the file. If not, the system determines if any of the user's groups shall be the object group owner. If not, then other permissions apply if the default user is configured. The system then determines whether the desired action is allowed.
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.

Note: This section does not describe how the process handles access requests by Windows administrators. Section 5, Root and Administrative Access, contains additional information on data access by administrators.

6.4 CIFS ACCESS OF NTFS SECURITY STYLE DATA

Figure 9 shows the process used when a user on a Windows host accesses data with NTFS security style using CIFS.

Figure 9) CIFS access of NTFS security style data.



The following list describes CIFS access of NTFS security style data.

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

   As part of the login process, if the user is logging in as a local user, the host communicates with the user's domain controller or the local system's local security database. The process might vary, depending on whether the user is part of an Active Directory domain or an NT domain, or is a local user.

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

   The result is a credential that contains the user SID and the user's group SIDs.

   If a user is from a nontrusted 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, either through mapping a drive or accessing through a UNC path.

   When a 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

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 set-up request. The request contains the user name, along with NTLM authentication information and other information used in session setup.

Kerberos Authentication

Kerberos authentication is used when the storage system is a member of an Active Directory domain and Kerberos authentication is negotiated between the client and the storage system.

After the Kerberos-level authentication is negotiated, the client sends a session set-up request. Kerberos uses tickets for authenticating user requests to network services. The session set-up request contains a TGS ticket for the storage system's CIFS service. This ticket is embedded in the session set-up 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 set-up request.

   NTLM Authentication

   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.

   Kerberos Authentication

   Even though the request contains all of the information needed to authenticate the user session, the user name is not in the security blob; therefore, the storage system consults the system's SID cache to see if the user name for that SID is in the 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.

   NTLM Authentication

   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, and the SIDs of the local groups to which the user belongs are added to the user's session cache.

   Kerberos Authentication

   The domain controller returns the user name to the storage system, which already has the user SID and all of 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 and must 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. There might be additional traffic between storage system and domain controllers during the session setup; however, the basic process follows what is outlined here.

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

`options cifs.guest_account unix_name`

`unix_name` is a user created in the UNIX identity store—`/etc/passwd`, NIS, or LDAP.

If guest access to the storage system is not desired, set this option to a null value. All guest access requests will then be denied. The default for this option is NULL. Guest access is denied by default.

`options cifs.guest_account ""`

If the user is connecting to the storage system as a local user, the option `cifs.guest_account` determines if access as a guest user is allowed or denied.

6. The user requests access to data on the storage system that uses NTFS-style file permissions. The storage system uses the Windows-style SIDs when evaluating file access rights; however, current implementations of Data ONTAP always perform a user mapping when data is requested by CIFS.

   The UNIX credentials, containing all of the mapped UNIX user's UID and GIDs, are stored with the Windows user's session authentication cache; therefore, a Windows-to-UNIX user mapping must be done before the session setup is complete.

   Therefore, the storage system maps the Windows user to the corresponding UNIX user even when a Windows user is accessing data stored in an NTFS volume.

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

   The storage system checks `/etc/usermap.cfg` or the ldap 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.

   Note: If there is not a specific mapping entry in `/etc/usermap.cfg` or in the ldap store, the ldap user mapping is configured, and the Windows user name is automatically used, the mapping process is still performed. It is important to determine if the UNIX user is a valid user. If the mapped name is not a valid user, access might be as the default user, or it might be completely disallowed, depending on how the `wafl` option `wafl.default_unix_user` is configured.

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, the user is either mapped to the generic UNIX account or access is denied, depending on how the `wafl` option `wafl.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 user, including the user's UID and GIDs, and including secondary group GIDs.

    The UNIX credential is stored with the user's cached authentication 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. Now that the UNIX credentials have been added to the authenticated session cache along with the Windows user SID information, the storage system can reply to the session set-up request with either a success or a failure.

If the session request is 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 NTFS security style volume.
14. The storage system uses the Windows user information stored in the user's session cache and compares it to the NTFS permissions on files and directories for which access has been requested. The system uses the Windows user and group SIDs to determine whether the desired action is allowed. The system then determines whether the desired action is allowed.
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.

Note: This section does not describe how the process handles access requests by Windows administrators. Section 5, Root and Administrative Access, contains information on data access by administrators.

7 APPENDIXES

7.1 CIFS ACCESS IN WORKGROUP MODE USING /ETC/PASSWORD, NIS, OR LDAP FOR AUTHENTICATION

A NetApp storage system supports four CIFS access methods; the desired method is chosen during CIFS setup. The access methods for options 1, joining an Active Directory domain; 2, joining an NT domain; and 3, joining a local Windows workgroup, were outlined in this technical report. Option 4, joining a workgroup using `/etc/password`, NIS, or LDAP for authentication, offers a substantially different method for managing file and folder access and is outlined in this appendix.

When CIFS access is based on a workgroup using `/etc/password`, NIS, or LDAP, session authentication is done on the basis of user names and passwords that are stored in the UNIX directory stores. Even if local Windows users are created on the storage system using the `useradmin` command, they are not used for session authentication. All authentications are done based on UNIX user information stored in the UNIX identity stores.

If the volume being accessed is NTFS security style, ACLs are not used during the access check, even if the file or folders has valid ACLs. File access is determined by share-level permissions. From the end user's point of view, an NTFS security style volume is a FAT volume.

If the volume being accessed is UNIX security style, UNIX file permissions in conjunction with share permissions are used to determine access.

1. The Windows client requests access to the data.

   The user who is logged in to the Windows client might be a domain user or a local user. However, when mapping the drive, the user must use one of the following methods:

   - Map the drive using a different user name in which the user name and password are the name and the password of a UNIX user is stored in the UNIX identity store.
   - Use pass-through authentication in which the logged-in user is using the same user name and password as a user stored in the UNIX identity store of the storage device.
2. The storage device maps the Windows account name to the UNIX account name.
3. The storage device checks `/etc/password` and `/etc/group`, NIS, or LDAP to retrieve the UNIX UID and GIDs.
4. The storage system compares account information with share-level permissions.
5. The storage system compares account information with UNIX permissions or DOS attribute bit.
6. If the user has both share and file-level access, then access is granted.

7.2 TROUBLESHOOTING WITH PERMISSION TRACING TOOL

A new permission tracing tool called sectrace was introduced beginning with Data ONTAP 7.3. This tool can be used to troubleshoot access control problems; that is, to determine why a client or a user is given or denied access to a file on the storage system. The sectrace tool adds filtering criteria for incoming file access requests.

8 REFERENCES

- "Data ONTAP 7.1.1 File Access and Protocols Management Guide" http://now.netapp.com/NOW/knowledge/docs/ontap/rel711/html/ontap/filesag/index.htm
- "Data ONTAP 7.2 File Access and Protocols Management Guide" http://now.netapp.com/NOW/knowledge/docs/ontap/rel72rc/pdfs/ontap/filesag.pdf
- TR-3014: "Multiprotocol Data Access: NFS, CIFS, and HTTP" http://media.netapp.com/documents/wp_3014.pdf
- TR-3580: "NFSv4 Enhancements and Best Practices Guide: Data ONTAP Implementation" http://media.netapp.com/documents/tr-3580.pdf
- TR-3596: "Storage-Level Access Guard Quick Start Guide" http://media.netapp.com/documents/tr-3596.pdf
- TR-3597: "Bulk Security Quick Start Guide" http://media.netapp.com/documents/tr-3597.pdf

9 REVISION HISTORY

Table 1) Revision information.

Date	Comments
February 2011	Updated with interaction behavior of NFSv4 ACL
July 2006	Creation