FluidFS in a Multi-protocol (SMB/NFS) Environment

Dell Fluid File System is an easy-to-use, secure solution for simultaneously sharing files in both SMB and NFS domains.

Revisions

Date	Description
October 2013	Greg Deffenbaugh, Nimrod Shavit — Initial release
July 2014	Nimrod Shavit – Minor updates

Contents

Tables

Figures

Executive summary

One of the strengths of modern Networked Attached Storage (NAS) systems is the ability to provide file storage and sharing services for clients using NFS, SMB, or both protocols. To effectively present a single data store to clients using different protocols, authorization paradigms, security, and locking mechanisms require a sophisticated solution. Dell's Fluid File System (FluidFS) is an easy-to-use solution that provides features for effectively sharing information with both SMB and NFS clients. This solution helps consolidate Windows and UNIX/Linux file serving functionality into a single system that provides storage efficiency, high performance, and scalability to petabytes.

This paper describes the network components that interface with FluidFS to provide a manageable, secure solution for sharing files in both SMB and NFS domains, and discusses best practices for deploying systems where both protocols need to access files simultaneously.

Introduction

FluidFS provides support for the SMB and NFS protocols. SMB is the network file sharing protocol deployed with the Windows operating system. NFS is the network file sharing protocol associated with the Linux and UNIX operating systems. There are three use cases for deploying network file sharing with regards to access protocols:

- Homogeneous use case – all file sharing is done through a single protocol, as in a Windows only file sharing environment
- Logically separated heterogeneous use case – both SMB and NFS clients perform file sharing, however, both protocols do not access individual files systems and NAS volumes
- Heterogeneous use case – both SMB and NFS clients access files and directories

FluidFS supports all three protocol deployment use cases:

- FluidFS support for the file system characteristics and networking features that differ per protocol:
  - Authentication
  - Directory Services
  - File Naming
  - Locking
  - File Permissions
- FluidFS protocol interoperability capabilities
- Considerations an administrator should take into account when implementing FluidFS in multi-protocol use case environments:
  - NAS Volume Security Style
  - Authorization
  - Cross Protocol Permissions Display
  - User Mapping

Authentication and Directory Services

Most file sharing deployments today use Directory Services to manage users and network resources. SMB deployments use Microsoft Active Directory (AD). NFS deployments use Lightweight Directory Access Protocol (LDAP) or Network Information Services (NIS) (formerly known as Yellow Pages). More recently, some NFS deployments are being performed using schema extension to Active Directory.

FluidFS is capable of operating efficiently in any of the environments described above. Furthermore, if a directory service is not deployed, FluidFS has the ability to provide a local repository of user and group information to control access to FluidFS-stored information.

SMB protocol

FluidFS can operate in Windows Workgroup mode or Windows Domain mode. Workgroup authentication uses the internal/local FluidFS directory of users/groups and does not rely on a domain controller.

In domain mode, the FluidFS cluster[1] is part of an Active Directory domain. AD domain controllers provide user/groups directory and authentication services. In domain authentication, the client negotiates the highest possible security level when establishing a connection with the FluidFS cluster. There are two primary levels of security:

- Basic security, based on NTLM (NT LAN Manager challenge/response)
- Extended security, based on Kerberos v5

Windows computers that are not part of an Active Directory domain use only NTLM-based authentication. By default, Windows clients (XP/2000 and later) that are part of an Active Directory domain will try to use Kerberos authentication first. If Kerberos is not available, the NTLM authentication is used.

Kerberos authentication

In an Active Directory environment, the Kerberos Key Distribution Center (KDC) service provides an authentication service (AS) and a ticket granting service (TGS). The Kerberos service runs on Windows domain controllers. A Windows client that wants to establish a session with another system, such as the FluidFS appliance, contacts the KDC directly to obtain session credentials, which are presented to the client in the form of a Kerberos ticket. For more information on Kerberos, see http://web.mit.edu/kerberos.

The Kerberos authentication process has three main phases:

1. The client authenticates with the KDC and receives a ticket to granting ticket (TGT) to be used in future authentication requests to access a service, such as a file share hosted on FluidFS.
2. The client issues a request to access a file share on FluidFS to the KDC (using its TGT), and receives a service ticket.
3. The client establishes a session with FluidFS using the service ticket. FluidFS decrypts the information using its long term key.

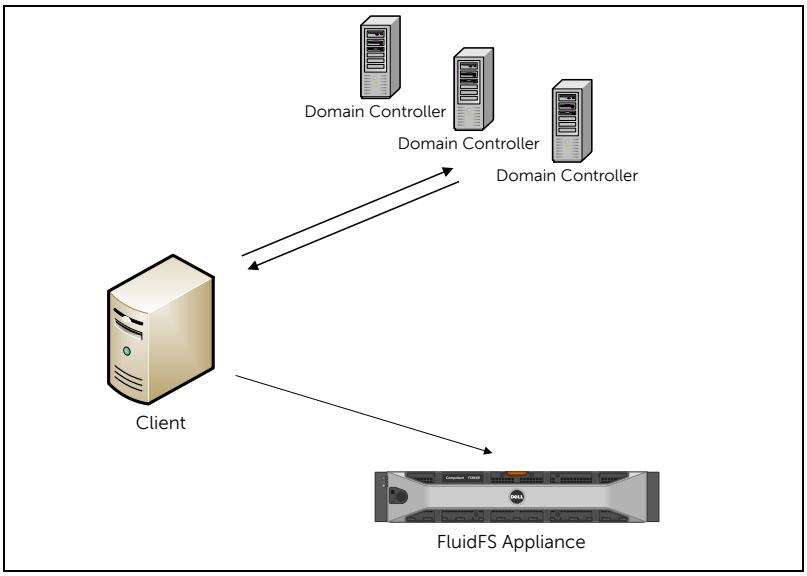

Figure 1. Kerberos Authentication Process

NTLM Authentication

Using NTLM, the FluidFS storage system contacts the Domain controller to verify user-supplied credentials. The Domain controller encrypts the user password with a challenge and compares the result with a response sent from the client. If these are identical, the authentication is successful.

The NTLM authentication process has three main phases:

1. Client contacts FluidFS system with logon details
2. FluidFS contacts Domain controller with the user name, challenge sent and received response
3. Domain controller validates authentication

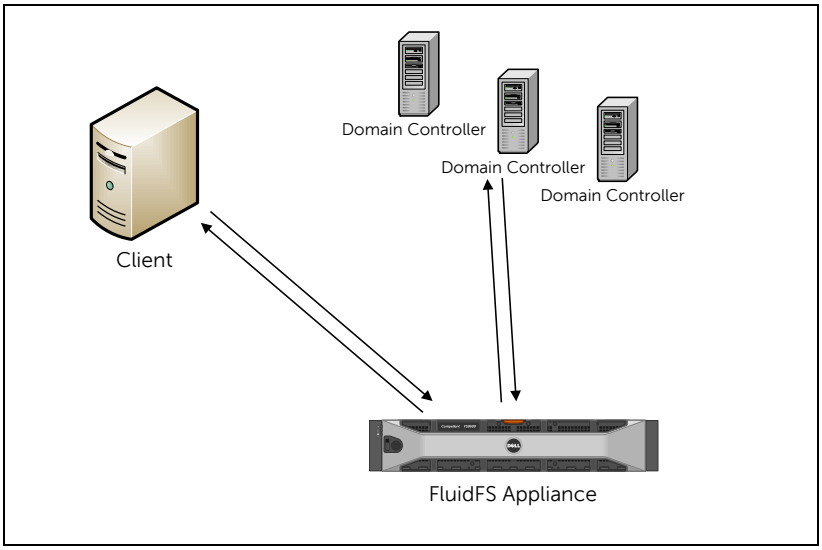

Figure 2. NTML Authentication Process

Directory Services and NFS v3/v4 protocol

NFS protocol traditionally relies on client authentication services (AUTH_SYS). For example, FluidFS trusts that a user on the system is authentic without verifying it. A user authenticates to the client machine and can then access through NFS mount to a FluidFS export. An NFS v3 session is stateless; each operation is sent along with the UID (user ID) and GID (group ID) of the user. FluidFS will query the directory services configured to obtain the group membership of the UID provided in the session.

FluidFS can integrate with various external user/group directory services for NFS v3/v4 protocol access in order to obtain user/group details.

- **NIS/YP**: Network Information Services/Yellow Pages is a client-server directory service protocol for distributing system configuration data such as users and groups between hosts in a network. A NIS/YP system maintains and distributes a central directory of user and group information and additional text-based information (such as hostnames). When configured to use NIS servers, the FluidFS queries the server for user/group details (UID/GID).
- **Open LDAP:** Lightweight Directory Access Protocol is used to access a hierarchical tree of entries. LDAP servers can store various attributes per entry such as name, surname, UID, GID, email address, and so on. When configured to use LDAP servers, the FluidFS queries the server for user/group details (UID/GID).
- **LDAP over Active Directory:** Active Directory schema can be extended so that each user/group entity can hold a UID and GID to allow central management of users and groups where you have both SMB and NFS users. SMB clients will use Active Directory natively, whereas NFS clients will use Active Directory as a directory service over LDAP protocol.

Authentication and NFS v3/v4 protocol

FluidFS v3 supports multiple methods for authenticating NFS clients/users. The authentication method is negotiated during the mount request, when the client specifies the desired authentication method, which should be enabled on the FluidFS export in order for the mount to succeed.

The following NFS exports Authentication method options can be enabled\disabled:

- **UNIX style**: AUTH_SYS security, FluidFS relies on client authentication services and trusts UID/GID information sent by the client.
- **Kerberos v5**: User Authentication support, FluidFS utilize Kerberos to verify user details are authentic
- **Kerberos v5 Integrity**: Enable cryptographic checksum of the data portion of each request and response.
- **Kerberos v5 Privacy**: Encrypt all packets bi-directionally including user data

FluidFS Kerberos authentication for NFS support relies on utilizing Active Directory in the environment.

File Naming

SMB and NFS protocols differ in their rules for file naming. New Technology File System (NTFS) does not permit the following characters in the file name: \/:*?"<>|

In a scenario where an NFS client has created a file with one or more of the "not permitted" characters and an SMB client accesses the file, FluidFS will modify the file name as follows:

- Translates the illegal characters to legal characters
- Maintains resemblance to original name
- Keeps original extension
- Ensures file name uniqueness and consistency

Note on case sensitivity

The NFS protocol is case sensitive. For example, two files with the same name can reside in the same folder even if their names only differ by a single uppercase/lowercase character, such as a.txt and A.txt.

SMB is case-aware, or case-preserving but not case sensitive. This means that when enumerating the contents of a folder, all files listed will maintain the original case with which they were created. Accessing a single file by name, however, is case-*insensitive*. For example, if both a.txt and A.txt exist in a directory, an SMB client will have to use a point-and-click method to select the file to be accessed.

File Locking

File locking is a method to limit access to a file to a single client process. File locking prevents the situation where two processes are modifying a file simultaneously, thereby eliminating the potential for data loss or dataset corruption.

SMB and NFS v3/v4 have different paradigms with regards to file locking as shown in Table 1.

Table 1. File locking in SMB and NFS v3/v4 protocols

	NFS v3	NFS v4	SMB
Share Mode lock	✖	✓	✓
Byte-range lock	✓	✓	✓
Leases/Delegations	✖	✓	✓
Enforcement	Advisory	Mandatory	Mandatory

NFS v4 and SMB have three types of locks:

- Share reservations/Share locks: declared when opening a file. Share locks limit access to a file to a single writer, but many clients can read the file; for example, while the client holds a write lock to a file, other clients can access the latest saved version of the file but cannot modify it.

- Byte-range locks: can be explicitly taken by an application. Byte range locks allow an application to lock a portion (byte range) of a file so that other applications can modify other ranges in a file.
- Delegations/Op-locks: not available to applications, but allow the operating system to cache file activity locally and improve responsiveness; for example, Microsoft Office makes updates to files on a local system rather than sends them to an SMB share provider until the user actively "saves" the file.

In scenarios of protocol interoperability, when users access the same file through SMB and NFS v4, FluidFS will handle locking in the following manner:

- NFS v4 has a comparable notion of Share reservations, and FluidFS will coordinate clients from these two protocols.
- NFS v4 has a comparable notion of Range locks, and FluidFS will coordinate clients between the two protocols.

If the FluidFS deployment is used where both SMB and NFS protocols have simultaneous access to a single file, it is recommended that the NFS clients use NFS v4.

File Permissions

Any file/directory stored in a FluidFS NAS volume includes metadata that (among other things) stores access permissions to the file/directory.

SMB protocol ownership and permissions are based on a Windows security identifier (SID) and Windows group SID for ownership along with a list of ACEs (access control entries) that comprise the ACL (access control list). The ACL defines the permissions applied to a given file or directory based on SMB Windows security descriptors.

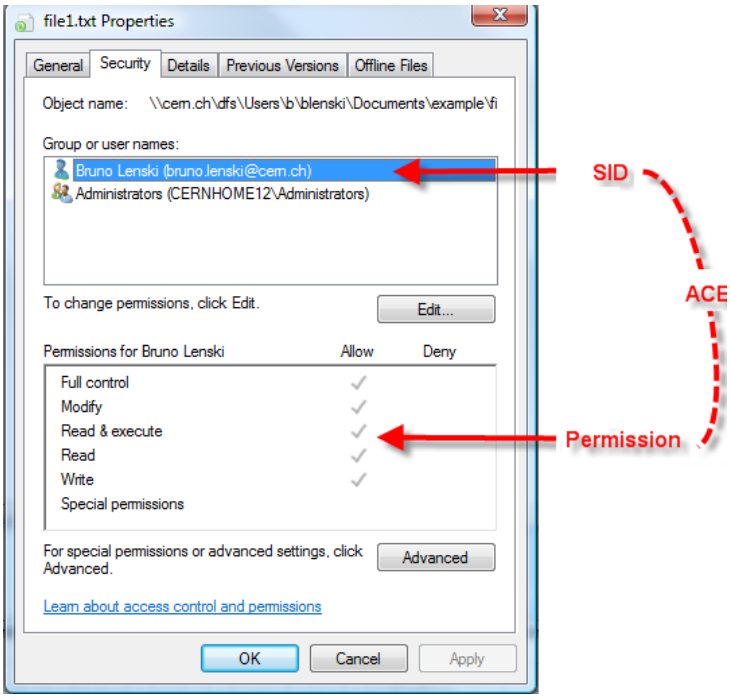

Figure 3. ACL/ACE example

NFS protocol file ownership and permissions are based on UID (User ID) and GID (Group ID) for ownership with POSIX style permissions described below:

- Each file/directory defines access for three entities (who):
  - User – owning user (UID); usually the creator of the file
  - Group – owning group (GID); usually the primary group of the User
  - Others – Users who are not any of the above
- For each entity, three types of actions are available:
  - Read (r) – if granted, allows reading from the file/directory
  - Write (w) – if granted, allows writing to the file/directory
  - Execute (x) – if granted, allows executing the file/directory
- UNIX file permissions are displayed in UNIX word notation; for example:

  rwxr-xr-- user1 group1

  Grants read, write and execute access to user1 (rwxr), read and execute access to members of group1 (r-x), and read access only to others (r--).

Note: NFS v4 protocol provides the ability to use POSIX based ACL/ACE security. FluidFS v3 support for NFS v4 does not include support for POSIX ACL/ACE authorization. Dell is planning to support POSIX ACL/ACE in the next major release of FluidFS.

User Mapping

User Mapping provides the ability to match users between different authorization domains (AD and LDAP/NIS). For example, Joe Smith's UNIX login name may be JSmith, while his Windows user name could be JoeS2. To make file access more user friendly for Joe, FluidFS needs to understand that JoeS2 and JSmith are the same person so that FluidFS can grant similar permissions to Joe independent of the domain that Joe uses to access the files and directories.

FluidFS supports manual and automatic user mapping. Automatic mapping is based on user name comparison; for example, if Joe has the same user name in both domains, such as Joe_Smith, FluidFS will automatically recognize that Joe is the same person in each domain.

If the user names are not syntactically identical between the LDAP/NIS and AD domains, FluidFS provides the ability to manually map the user names. The user map information is managed and stored by FluidFS. Manual user mapper data is stored with other system data on the FluidFS system. It is possible to copy the mapper data to a file for backup or to be transferred to a second FluidFS cluster. If AD provides LDAP service, each user will have a single name in the database, and manual user mapping will not be needed.

In scenarios where user mapping is required, the mapping mechanism is used to translate SID to UID and vice versa. If the system needs to map a user and cannot find a corresponding ID map, (either by

automatic or manual mapping, or if user mapping is not configured) it will map the user to a Guest account.

Note: Group names/IDs are never translated. The primary group for the mapped user is used for group level permissions.

NAS Volume Security Style

FluidFS NAS volumes are the virtual file systems that reside on top of the NAS pool. NAS volumes are accessed by clients through shares (for SMB protocol) and exports (for NFS protocol). Every NAS volume has a security style that dictates which file protocol governs the ownership and permission of data written to the NAS volume.

- **NTFS Security Style:** Files and directories are created with Windows security identifier ownership and with NTFS Windows security descriptors permissions. Only SMB clients can change ownership and permissions of files and directories stored in the volume. NFS clients can read and write data to the volume.
- **UNIX Security Style**: Files and directories are created with UID (user ID) and GID (group ID) ownership and POSIX style permissions. Only NFS clients can change ownership and permissions of files/directories stored in the volume. SMB clients can read and write data to the volume.
- **Mixed Security Style:** Files and directories are created with ownership and permission based on the file protocol that created them. A NAS volume with MIXED security style may end up having some files with SID owner and NTFS Windows security descriptors permissions (which were created by SMB protocol) and other files with UID/GID ownership and POSIX style permissions (which were created by NFS protocol). Any protocol can change file ownership or permission at any time. For example, a file that was created by an SMB client with NTFS permission can be changed by an NFS client later.

The security style for a NAS volume can be changed at any time. Changing the security style will not proactively affect files that are already stored in the NAS volume. Files created after a change in security style will reflect the new security style.

Authorization

Authorization is the process that decides if a certain user is allowed to perform certain actions on a file or directory based on the file/directory ownership and permissions. FluidFS authorizes operations on file or directory using native access check. File ownership and permission type are always evaluated against user and group entities from the same type, even if the original request came from a different type.

The following examples will show how authorization works with the three security styles

UNIX Security Style example

This example describes the behavior of a file with UNIX security style.

Table 2. UNIX security style

File Name	Ownership	Permissions
FileNFS.txt	UID/GID	Rwxrwxr-x

Native access

When a user accesses FileNFS.txt via NFS protocol, FluidFS authorizes access to the file using the UID/GID that is provided in the NFS access and the group membership of this UID in the NFS directory service. The user can change file permissions and ownership (chown, chmod).

Non-native access

When a SMB client accesses FileNFS.txt, the client presents FluidFS with its SID. In this scenario authorization cannot be done based on SID as the file has UID/GID ownership and POSIX permissions. Therefore, FluidFS will use the User Mapping mechanism to translate the SID to a UID. Then FluidFS will determine if the UID or the primary GID is authorized for the requested action.

Assuming appropriate permissions, the user can modify or delete the file, however cannot change the file ownership or permissions (ACEs).

NTFS Security Style example

This example describes the behavior of a file with NTFS security style.

Table 3. NTFS security style

File Name	Ownership	Permissions
FileSMB.txt	SID/GSID	ACL

Native access

FileSMB.txt ownership is SID (SMB protocol) and the permissions are NTFS based (ACL). When a user tries to access the file via SMB protocol, FluidFS will use the provided SID and the group membership of this SID in the AD to verify the requested operation is authorized. The user can change the ACEs for the file.

Non-native access

When a user tries to access FileSMB.txt via the NFS protocol, FluidFS will map the UID to a SID using the User Mapping mechanism. FluidFS will use the SID for the native authorization process (the actual SID and group membership for this SID are used).

Assuming appropriate permissions, user can modify or delete the file, however cannot change the file ownership or permissions of the file.

Mixed Security Style example

When mixed security style is used, the security style of a file will reflect the protocol used by the last client to save the file. When the file is accessed by a client, authorization will initially behave like the security style was either UNIX or NTFS.

For example, if the current security style for a file is NTFS and the file is accessed by an NFS client, file authorization will behave like the non-native access to NTFS. When the client saves the file, the file's security style will reflect the style of the client.

To continue the example, assuming the NFS client has write permissions to the file according to the NTFS ACEs, when the file is written, its security style will be changed to UNIX. Subsequently, if an SMB client accessed the file, access would be authorized through non-native access for UNIX.

Cross-protocol permissions display

When a client views the current permissions of a file, the permissions need to be displayed in the format that the client understands. When files are accessed in a non-native manner, FluidFS translates the native permissions to a permission format that matches the client expectation.

For example, if the security style for the file is NTFS and it is accessed by an NFS client, the NTFS ACEs will be mapped to the UNIX security word for display.

Note: NTFS ACEs provide a richer authorization data structure than UNIX permissions and therefore certain heuristics must be applied when mapping permissions between domains. When an ACL is displayed in UNIX word format, it is simplified. When UNIX permissions are displayed as an ACL, certain defaults are assumed.

In any case, the actual authorization process is based on permissions stored with the file, which may vary slightly from the presented translation of permissions.

Non-native access: NFS display guidelines

When NTFS file information is displayed for an NFS client, FluidFS maps the SID to a UID and sets the GID to the primary group of the UID.

When ACL permissions information is displayed for a UNIX client, FluidFS does the following:

- If the ACE is for file owner SID, FluidFS translates the ACE to UNIX owner unit.
- If the ACE is for file owner primary GSID, FluidFS translates the ACE to UNIX group unit.
- If the ACE is for Everyone, FluidFS translates it to all units.
- If none of the above, FluidFS translates to other UNIX unit.

Example: Dina Fine (dina@Dell-IDC.com) is the owner of a directory called cifs_dir. The following figures show three displays of ACEs for the directory cifs_dir.

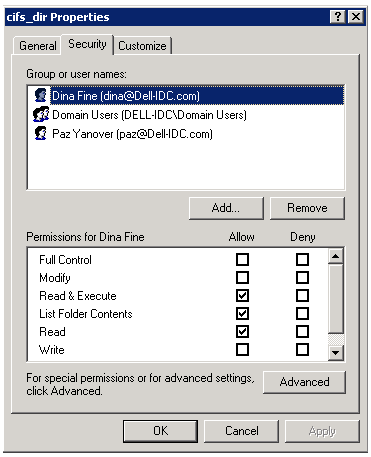

Figure 4. ACE for user Dina Fine

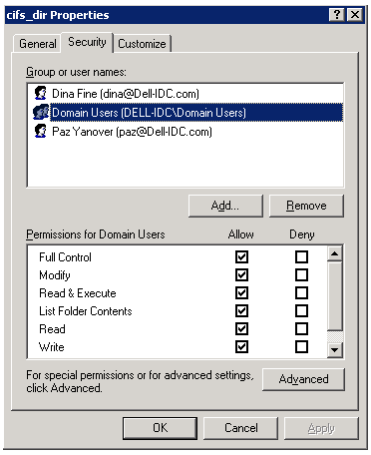

Figure 5. ACE for the group Domain Users

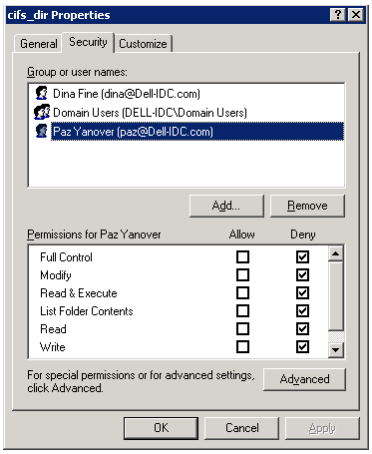

Figure 6. ACE for the User Paz Yanover

The UNIX word representation of this set of ACEs would be: dr-xrwx--- 1 dina fs 0 Jan 2 11:31 cifs_dir

Dina Fine's SID would be mapped to the UID for Dina Fine (dina). Since Dina did not have write permission to the directory in the ACE, her UID is only granted read and execute permissions in the UNIX display (r-x).

In this instance, Dina's primary group in the Windows domain is Domain Users. Dina's primary group in the UNIX domain is fs, which is given the same permissions as Domain Users have in the Windows domain. In this case, Domain Users have full access to the directory, and fs has read, write, and execute (rwx) permissions.

However, when the file is accessed, permissions are evaluated against Domain Users in the Windows domain, not the fs group in UNIX domain. In this case, even if Paz Yanover is a member of the fs group he cannot gain access to the directory from the UNIX domain because his credentials will be evaluated against the ACL for the directory which has an ACE that explicitly denies him access.

Since the ACL for the directory does not include an entry for Everyone, access to the directory for other is denied (---).

Non-native Access: SMB Display Guidelines

When NFS information is displayed for an SMB (Windows) client, FluidFS maps the UID to an SID and retrieves the primary group GSID of the SID.

When permissions based in a UNIX word are displayed as an ACL, FluidFS:

- Creates an ACE for the SID file owner (based on UNIX word user unit).
- Creates an ACE for the GSID file owner (based on UNIX word group unit).
- Creates an ACE for Everyone (based on UNIX word other unit).

Example: Dina Fine is the owner of a directory nfs_dir.

The UNIX Security Word for the directory is: drwxr-xr-- 1 dina fs 0 Jan 2 11:45 nfs_dir

Dina Fine's UID is mapped to her SID. Since Dina has full access to the directory in the UNIX domain, her ACE gives full permission:

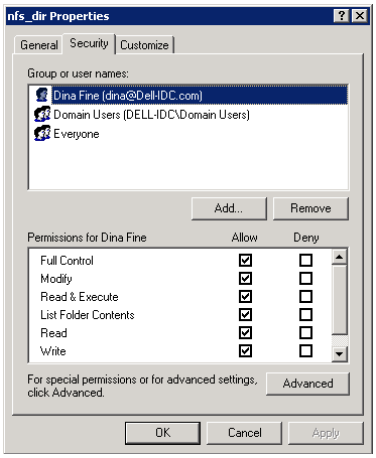

Figure 7. Dine Fine's ACE for nfs_dir

Since Dina owns the file, the permissions for Dina's primary group in the Windows domain are displayed as the same as the group permissions on the file. In this case, the UNIX group fs has read and execute (r-x) permissions so the Domain Users Windows group has the ACE shown below.

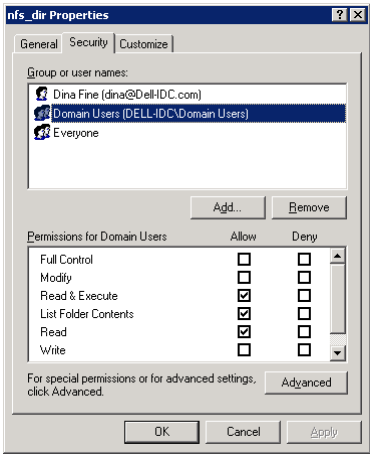

Figure 8. ACE for Domain Users group for nfs_dir

Since the UNIX other entry has read permissions, Everyone in the Windows domain has the ACE shown in Figure 9.

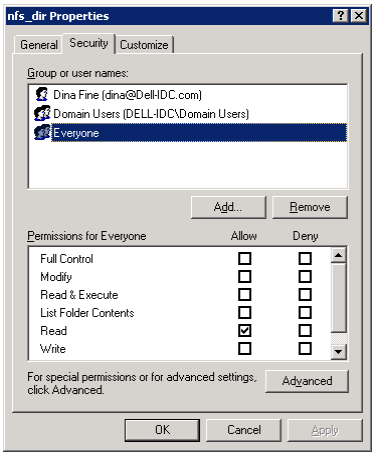

Figure 9. ACE for Everyone group

Best practices for choosing NAS Volume Security Style

If client create/modify access to a NAS volume is primarily from a single authorization domain, it is recommended that security style follows this domain's authorization model and domain. This provides efficient access to files from both domains and provides the simplest method for managing file authorization and file permissions display.

If client create/modify access to a volume is balanced between the domains, choosing NTFS or UNIX Volume security style will provide the most predictable environment since the security style for individual file will not reflect the "last writer's" authorization domain. However, there are use cases

where applications need to be able to manipulate file access attributes and therefore benefit from Mixed Security style.

As a reminder, security style is defined at the NAS volume or virtual file system level. If different datasets would benefit from using different security styles, unique NAS volumes should be created for each dataset. If there is uncertainty about the best security style for a dataset, creating a separate NAS volume for that dataset is recommended so that experimentation of security styles can be executed without affecting other dataset data stores.

Summary

The Fluid File System provides native support for the SMB and NFS protocols as well as a range of services for multi-protocol environments.

When working in a discrete manner, FluidFS supports services, such as Authentication and Directory Services in a pure, native manner. In other words, there is no need for the administrator to deviate from standard operating methods per protocol simply because both SMB and NFS protocols are in use in parallel in a FluidFS cluster.

When cross-protocol access to files within a NAS volume is deployed, FluidFS offers powerful methods to address file authorization and security to enable effective sharing of information between authentication and authorization domains.

Additional resources

Referenced or Recommended Dell Publications:

- "Dell Fluid File System Overview" white paper: http://en.community.dell.com/techcenter/extras/m/white_papers/20212904.aspx
- "Dell Fluid File System Migration Guide" white paper: http://en.community.dell.com/techcenter/extras/m/white_papers/20140531/download.aspx
- *Dell FluidFS NAS Solutions Administrator's Guide*: http://support.dell.com/support/edocs/stor-sys/nx3600/en/AG/AGen.pdf
- Additional Dell FluidFS NAS white papers: http://en.community.dell.com/techcenter/storage/w/wiki/4291.network-attached-storage.aspx

Referenced or recommended Microsoft publications:

- Kerberos Explained: http://technet.microsoft.com/en-us/library/bb742516.aspx
- How Domain and Forest Trusts Work: http://technet.microsoft.com/en-us/library/cc773178(v=WS.10).aspx
- Understanding ACLs in NTFS: http://blogs.msdn.com/b/brian_dewey/archive/2004/01/20/60902.aspx

- Access Control Entries:
  http://technet.microsoft.com/en-us/library/cc961995.aspx

Other Reference or recommend publications:

- Information on Kerberos Network Authentication Protocol
  http://web.mit.edu/kerberos
- Filesystem permissions:
  http://en.wikipedia.org/wiki/Filesystem_permissions

---

[1] A FluidFS cluster consists of one or more FluidFS appliances implemented in a single namespace.