NFS INDUSTRY CONFERENCE

Multi-Protocol File sharing Environments

Jeff Purcell - EMC

Senior Corporate Systems Engineer

purcell_jeff@emc.com

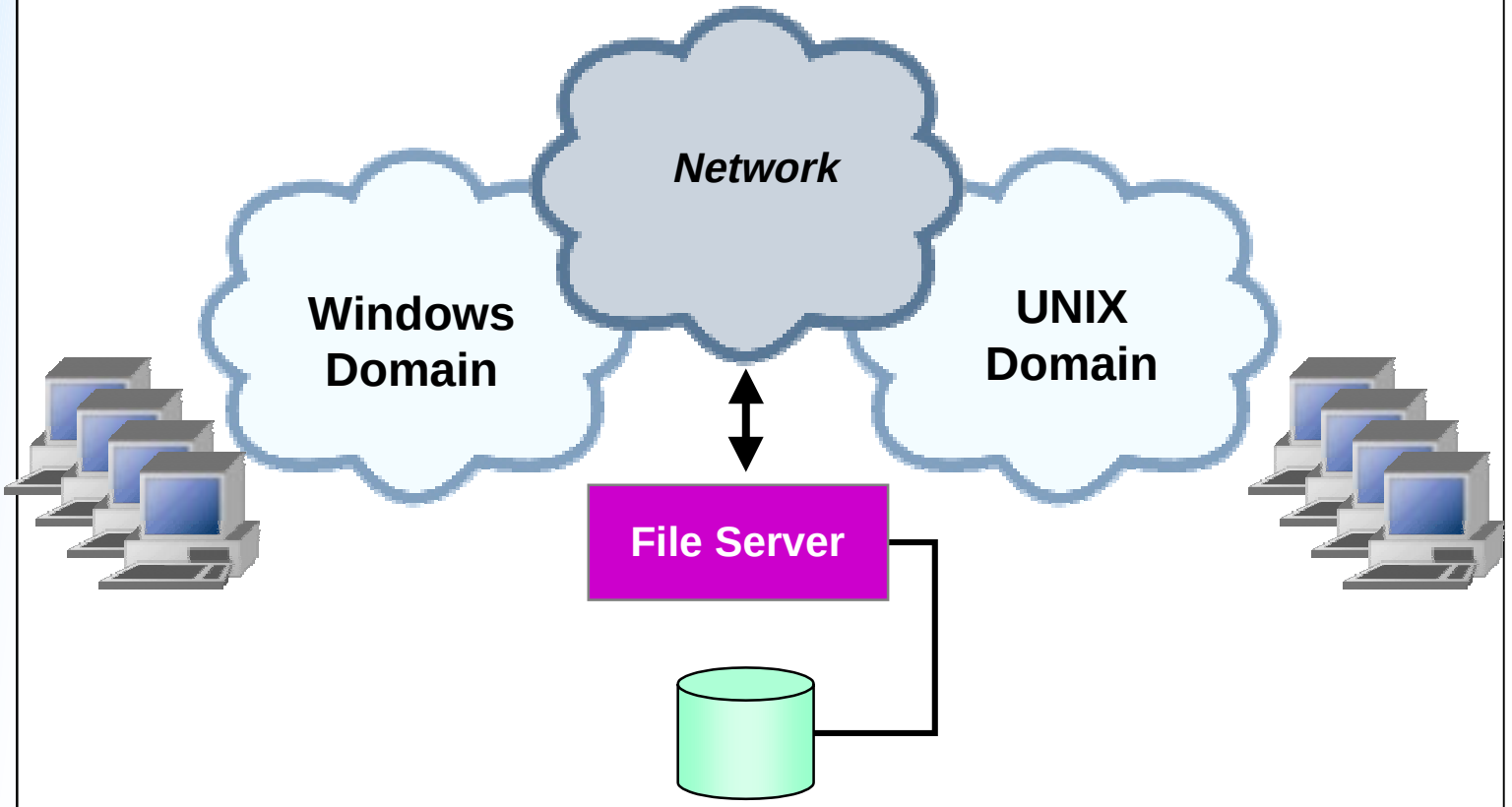
NFS INDUSTRY CONFERENCE

Agenda

- Understanding Multi-Protocol
- Architectural Challenges
  - User Considerations
  - Network Security
  - File and Directory Consistency
  - Filesystem Security
- Summary

Understanding Multi-Protocol

- Basic business goal is simple: share the same filesystem to both UNIX and Windows users concurrently



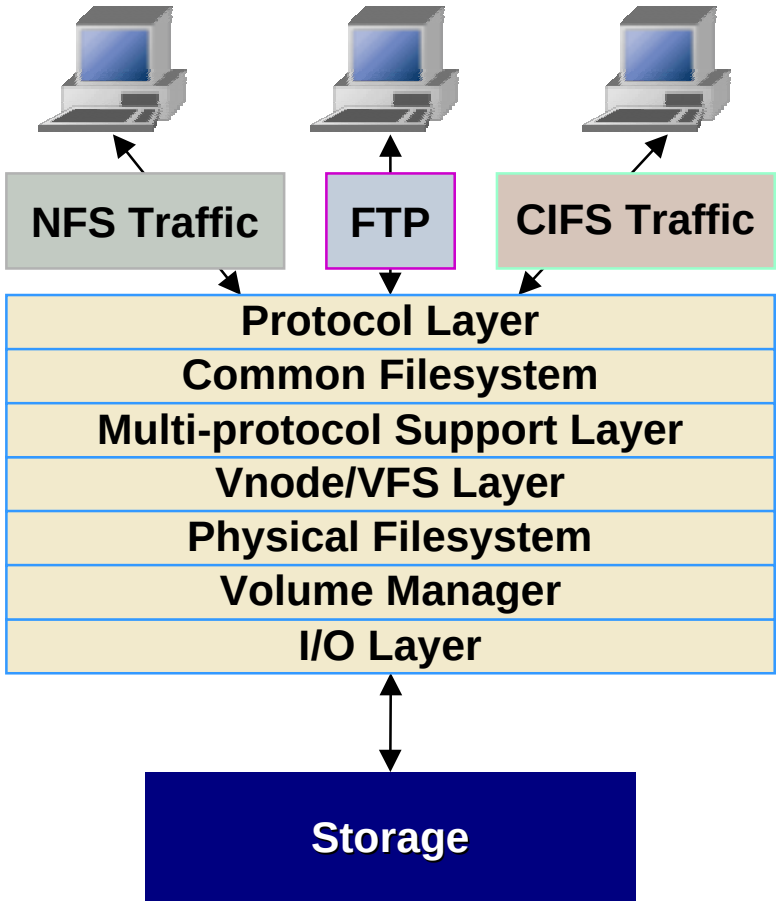
NFS INDUSTRY CONFERENCE

Understanding Multi-Protocol - Design Goals

- Efficient use of storage
- Support both NFS and CIFS Windows clients
- Share common set of Infrastructure resources
- Simplicity of Management
- Transparent to end users

NFS INDUSTRY CONFERENCE

Multi-Protocol: UNIX and Windows Information Sharing



NFS INDUSTRY CONFERENCE

Multi-Protocol Architectural Challenges

- Integration
- Different Security Semantics
- Different Locking Semantics
- Different User Information Repositories
- Different Network Topology Structures

NFS INDUSTRY CONFERENCE

Multi Protocol Challenges - Security Layers & Interaction

A way of describing the security layer:

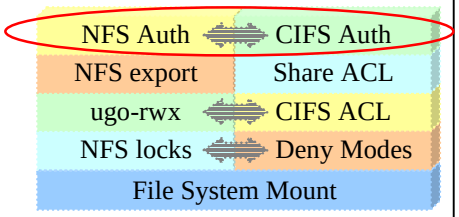
Layer	Security Entity
User	NFS Auth ⟷ CIFS Auth
Network	NFS export — Share ACL
File/Dir	ugo-rwx ⟷ CIFS ACL
Locking	NFS locks ⟷ Deny Modes
I/O	File System Mount

NFS INDUSTRY CONFERENCE

User Authentication

- Each protocol provides unique authentication methods
- Need to Support and conform to each
- NFS:
  - Users authenticated as part logon to the NFS client
  - In NFSv2 & v3, NFS clients provide UID and GID
  - Secure NFS provides additional security with KerberosV5
- CIFS:
  - Users are authenticated via Domain Controller in most cases
  - NTLMv0.12 or Kerberos (default in Windows 2000) and LDAP

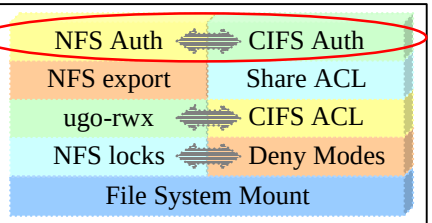
CIFS Authentication



- File server can participate in Windows Domains
  - Windows 2000 Member Server (Kerberos and LDAP)
  - NT4 Member Server (NTLM)
- Security = NT Style
  - User authentication handled by domain controller using
  - Access checking is done against user and group SIDs
- Security = SHARE
  - Access provided via global or no password protection
  - Limited Security provided

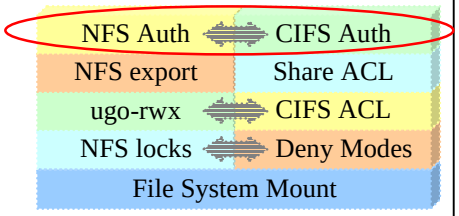
NFS INDUSTRY CONFERENCE

Mapping Different User Repositories



- Credential Repositories
  - Local passwd and group files
  - NIS
  - Windows 2000 Active Directory
  - Local Workgroup
- Basic concept
  - Match Windows credentials with UNIX credentials
  - Allow for cross-platform access, permissions, quotas, etc.
- User Mapping Services
  - Assigns persistent SID → UID/GID mappings for a windows environment

NFS INDUSTRY CONFERENCE



Mapping Issues:

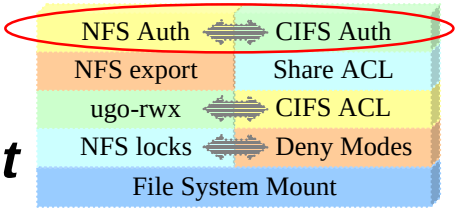
Primary Groups in a Bi-Lingual Environment

- Both UNIX and NT have the concept of a "Primary Group" for a user
  - It is compulsory in UNIX, optional in Windows
- All files have an associated owner (UID) and group (GID)
- When a file is created via NFS, the GID is taken from the GID as supplied by the client

NFS INDUSTRY CONFERENCE

Mapping Issues:

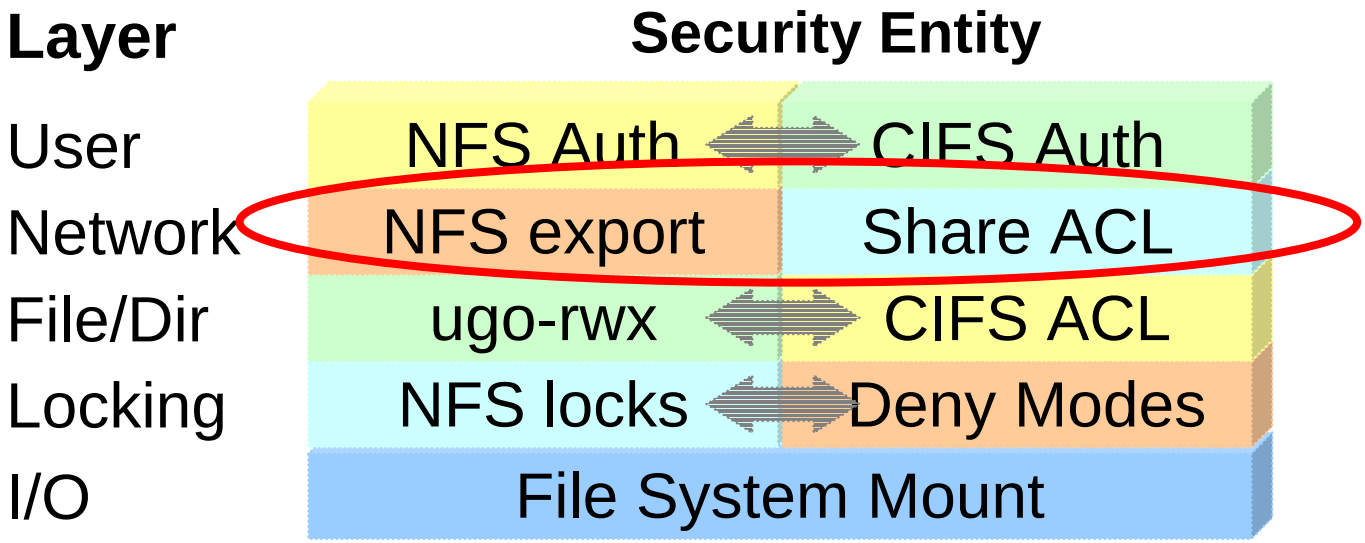
Primary Group in Bi-Lingual Environment



- When a file is created via CIFS, the GID is taken based on policy definition.
- The default behaviour is to use the GID associated with the NT user's primary group
  - Typically "Domain Users"
  - NFS users may see GIDs or group names on files that represent Windows Groups
- If this is not desired, you can also get GID from user's UNIX primary group as defined in the passwd file, NIS or AD

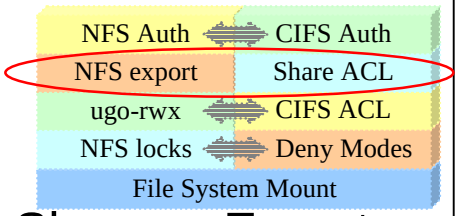
NFS INDUSTRY CONFERENCE

Network Exports and Shares



NFS INDUSTRY CONFERENCE

Network Interoperability Options



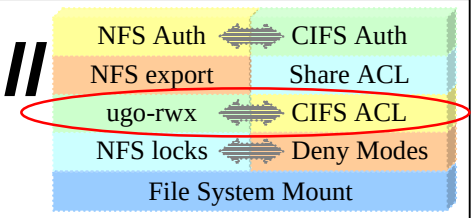
- Administrative restrictions implemented on Share or Export
- **NFS**
  - **ro/rw/access** – export read-only, read/write, or limit access to a set of hosts
  - **vlan** – limit access to a specific vlan
  - **anon** – set UID given to anonymous users, eg. Nobody (UID=65534) on clients.
  - **root** – set of list of hosts who can be root
- CIFS share security options include
  - **ro** – read-only access is allowed to the share
  - **rw** – grant read/write access to listed hosts
    - Everyone else is read-only
  - **umask** – works the same as UNIX, independent of NFS access
  - Share ACLs (the SD on the share)

NFS INDUSTRY CONFERENCE

File & Directory Security

Layer	Security Entity
User	NFS Auth ⇔ CIFS Auth
Network	NFS export — Share ACL
File/Dir	ugo-rwx ⇔ CIFS ACL
Locking	NFS locks ⇔ Deny Modes
I/O	File System Mount

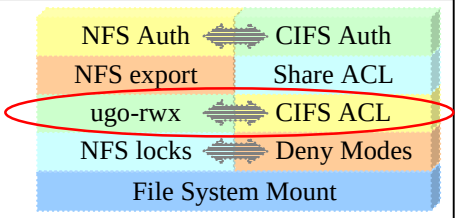
One Set of Permissions for All



- Goal
  - To have just one set of permissions for CIFS, NFS and FTP
- Restrictions
  - Permissions limited to the Lowest Common Denominator
    - Only Read/Write/Execute for Owner, Group and Other
    - Cannot use “Deny” ACEs
    - Only one Group ACE
    - Cannot use an ACE for any user except Owner

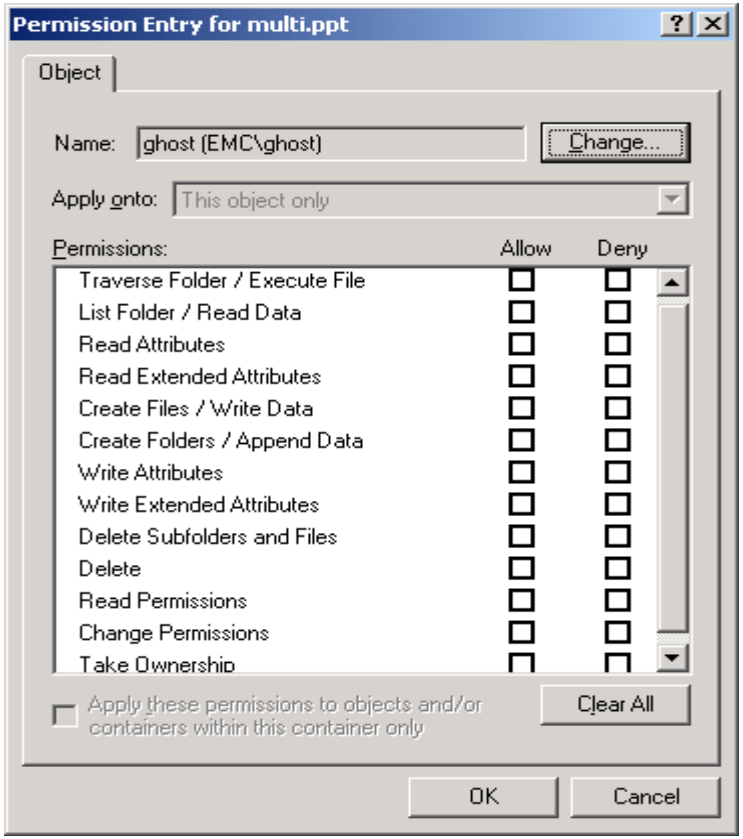
File & Directory Security

Windows and UNIX permissions are fundamentally different



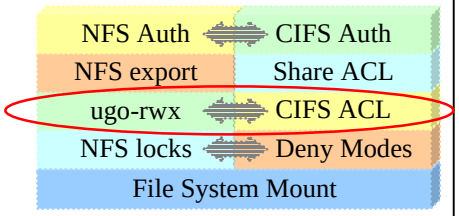
- UNIX/NFS
  - **Only three** entries
    - Owner
    - Group
    - Other (!= Everyone)
  - That **Allow (no Deny)**:
    - Read
    - Write
    - Execute
- Windows/CIFS
  - **Multiple** ACEs which Allow or Deny any of →

```
drwxr-xr-x   2 root     staff         80 Sep 16 09:27 eng
-r-xr-x---   1 jeff     other        305 Sep 16 09:30 jp
-r--r--r--   1 sys      sys         1001 Sep 16 09:29 test
#
```



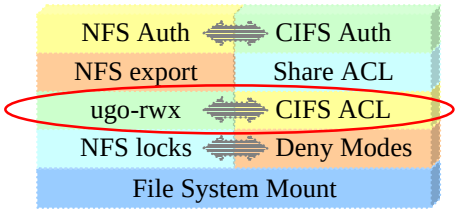
File & Directory Security

Umasks and ACL Inheritance



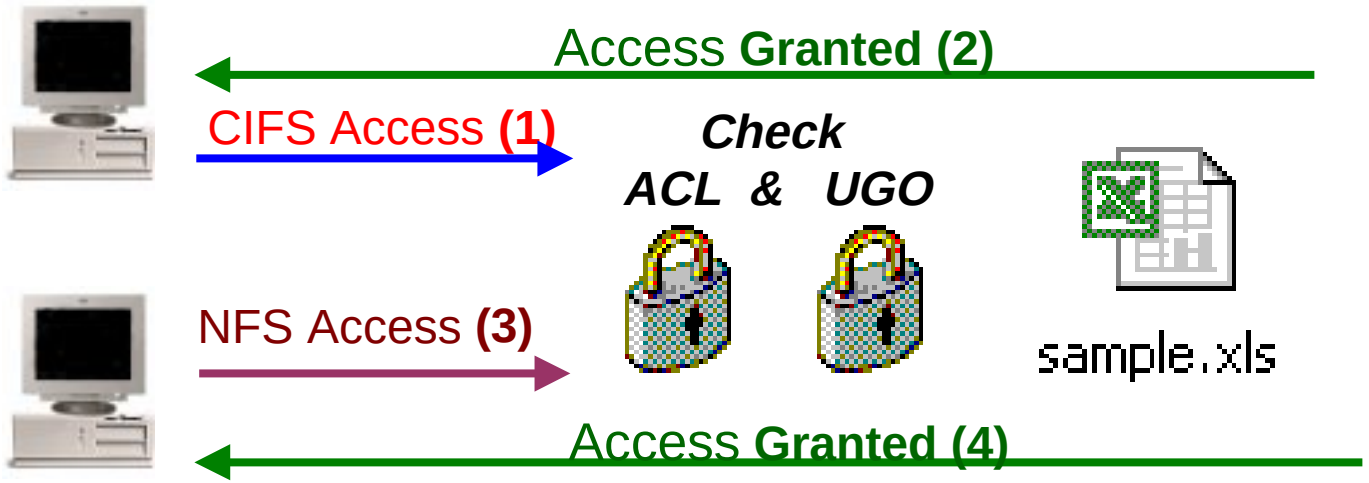
- When a new file or directory is created on a bi-lingual file system it needs both a UNIX permission string and an ACL
- For files and directories created by NFS clients
  - The users umask determines the UNIX permissions
  - The ACL is inherited from the enclosing directory (if it has one)
- For files and directories created by CIFS clients
  - The ACL is inherited from the enclosing directory where there is one
  - Each share has a umask that is used to apply UNIX permissions to files created through it

File & Directory Security



- Access Policy
  - Should access rights be determined by ACL and/or UGO?
  - Choose: Native, NT, UNIX, or Secure

Example: "Secure" Mode
(Always Check ACL and UGO)



NFS INDUSTRY CONFERENCE

Benefits with NFSv4 ACL's

- Take advantage of rich set of ACL's
  - Read, write, list, add, execute, etc.
- Extended Access Control Entries (ACE)
  - Interactive, anonymous, authenticated
- Provide better alignment with CIFS ACL's
- Benefits interoperability and integration between protocols.
- Improved control and end user experience

NFS INDUSTRY CONFERENCE

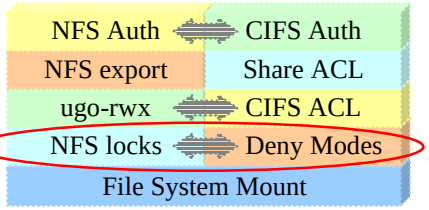
File & Directory Locking

Layer	Security Entity	
User	NFS Auth	CIFS Auth
Network	NFS export	Share ACL
File/Dir	ugo-rwx	CIFS ACL
Locking	NFS locks	Deny Modes
I/O	File System Mount	

NFS INDUSTRY CONFERENCE

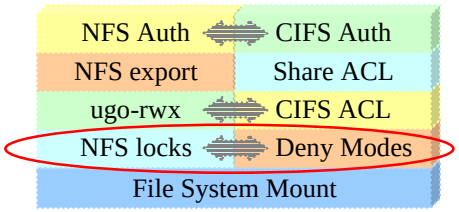
File & Directory Locking

CIFS and NFS Locking Semantics Differ



- File Locking provides a method for file integrity
- Coordinates user access to file
- NFS locks are advisory and seldom used in practice
- CIFS has
  - Opportunistic locks (exclusive and batch & Level II)
  - Range locking (equivalent to NFS Range locking)
  - Byte Range -
  - “Deny Modes”, e.g., “Deny Write”
- Application use of CIFS oplocks and deny modes is variable
- Samba uses Posix and strict locking

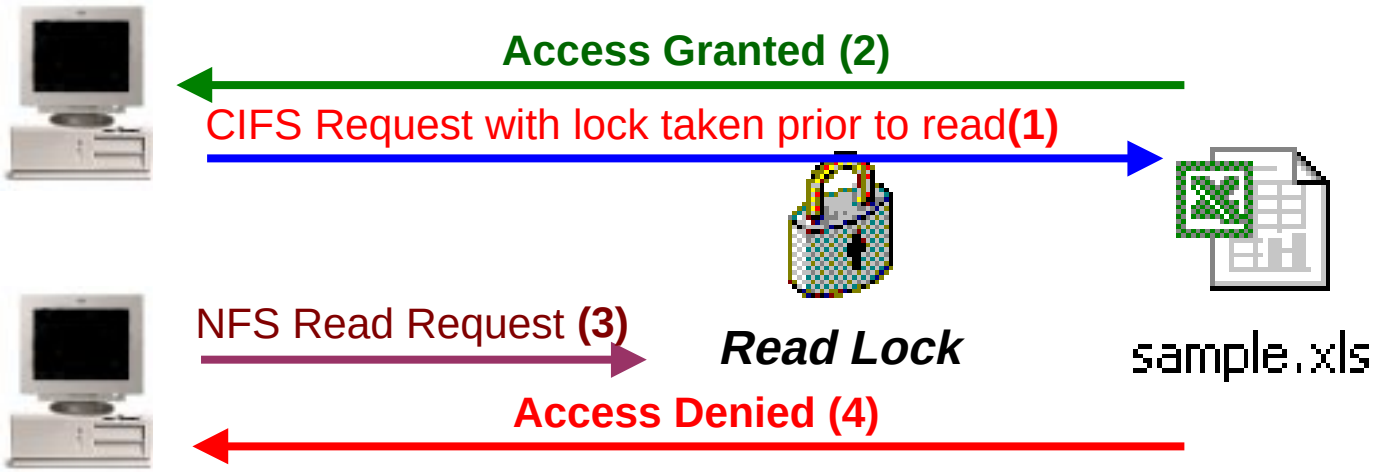
File & Directory Locking



- Locking Policy
  - How should locks be handled?
  - Choose: nolock, wlock, or rwlock

Example: "rwlock" Policy

(Read and/or write requests will be denied if a corresponding lock is already in place)



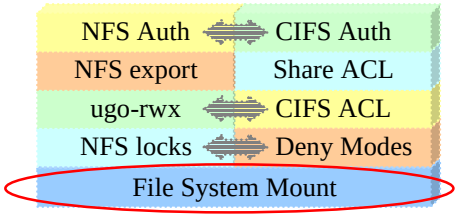
NFS INDUSTRY CONFERENCE

File System I/O Security

Layer	Security Entity
User	NFS Auth ⇔ CIFS Auth
Network	NFS export — Share ACL
File/Dir	ugo-rwx ⇔ CIFS ACL
Locking	NFS locks ⇔ Deny Modes
I/O	File System Mount

NFS INDUSTRY CONFERENCE

File System I/O Security



- A file system can be mounted in either of two modes
  - Read-Write (default for standard file systems)
    - Both reading and writing to the file system is allowed
  - Read-Only (default for certain fs types)
    - The OS will deny any attempts to write to the file system
- Read-only access provides ability to share data to a wider audience.

NFS INDUSTRY CONFERENCE

Applications of Multi Protocol Support

- Why would you want to use Multi Protocol Filesystem resources in your environment?
- We have seen countless uses of Multi Protocol Filesystems
  - User home directories
  - Collaborative environments, including software development
  - Manufacturing and design
    - One data set, multiple application platforms
  - Infrastructure systems, processing computers are UNIX based, clientele is Windows-based
  - Common set of permissions to manage

NFS INDUSTRY CONFERENCE

Summary

- Multi-protocol technology solutions continue to evolve
- NFSV4 will benefit key areas of multi-protocol
  - Security and ACL implementation
  - File Locking and policy integration
  - Improved efficiency in processing tasks
- Challenges lie ahead in the deployment and adaptation of supporting infrastructure
  - V3 and V4 support
  - Security integration
- User Mapping still required

NFS INDUSTRY CONFERENCE

EMC²

where information lives